داده کاوی و کشف دانش

داده کاوی و کشف دانش

تدوین و نالیف:

مهدی غضنفری، سمیه علیزاده، بابک تیمور پور

فهرست مطالب

فصل اول: مقدمه‌ای بر داده‌کاوی

فصل دوم: آماده‌سازی داده‌ها در داده‌کاوی

فصل سوم: تحلیل خوشه‌ای

فصل چهارم: قواعد تلازمی

فصل پنجم: دسته‌بندی و پیش‌بینی

فصل ششم: انباره داده‌ها

فصل هفتم: متدلوژی اجرا و پیاده‌سازی پروژه‌های داده‌کاوی

فصل هشتم: سریهای زمانی در داده‌کاوی

فصل نهم: تحلیل شبکه‌های اجتماعی

فصل دهم: کاربرد داده‌کاوی در مدیریت ارتباط با مشتری

پیش‌گفتار

ن و القلم و ما یسطرون

سوگند به قلم و هر آنچه می‌نگارد

داده‌کاوی همچون هر کاوش دیگری به‌دنبال گنجی است که از چشم نهان است. داده کاوی به‌عنوان رویکرد کشف دانش، در دریای داده‌ها می‌کاود تا مروارید ذیقمت دانش را به چنگ آورد. هرچند داده‌کاوی به‌شکل نوین خود شاخه جدیدی در حوزه علوم دانشگاهی محسوب می شود ولی برخی از روشها و ابزارهای آن دارای سابقه بسیار دیرینه‌ای هستند. این ابزارها که با آنها در این کتاب آشنا می‌شویم به فراخور نیازهای مدیران و تحلیلگران و نیز وضعیت بانکهای داده متنوع و متکثر شده‌اند. در کشور ما نیز چند سالی است که مکانیزه شدن سیستمها منجر به جمع‌آوری آرشیو بزرگی از داده‌ها شده است. با افزایش روزافزون داده‌های ذخیره شده، اکنون با انبار بزرگی از داده مواجه هستیم. استفاده از این داده‌ها بیشتر مربوط به عملیات روزمره سازمانها و شرکتها است. در سطوح بالاتر، گزارشات مدیریتی نیز تهیه می‌شود که برای تصمیم‌گیری مورد استفاده قرار می‌گیرد. به‌ندرت پیش می‌آید که الگوهای موجود در این داده‌ها جستجو و یافته شوند. سؤالات بسیاری برای مدیران مطرح است که جواب به آنها با داشتن الگوهای مفیدِ یافته شده در این داده‌ها ممکن است. برای مثال مدیران نیازمند شناخت گروه‌های متفاوت مشتریان خود هستند، یا علاقه‌مند هستند بدانند احتمال خرید کدام مشتریان بالقوه بیشتر است. دولت به‌دنبال گروه‌بندی مناطق مختلف کشور بر حسب شاخصهای توسعه یافتگی است. در این راستا می‌توان روشهای مختلف توصیف و پیش‌بینی را برای استخراج الگوها و قواعد مناسب از سوابق داده‌های موجود به‌کار گرفت. در حوزه‌های تصمیم‌گیری جواب به این سؤالات باید متکی بر داده‌ها و اطلاعات موجود باشد. این نتایج به‌همراه نظرات فرد خبره می‌توانند کمک مناسبی به افراد تصمیم‌گیرنده نمایند. روشهای موجود برای این کار تحت نام عمومی داده‌کاوی و کشف دانش مطرح هستند. این روشها که ترکیبی از آمار، هوش مصنوعی و پایگاه داده‌ها می‌باشند چند سالی است که در کشورهای توسعه یافته صنعتی رونق زیادی پیدا کرده‌اند و اخیراً نیز در ایران مورد توجه قرار گرفته است.

این کتاب سعی دارد مفاهیم و مبانی داده‌کاوی و روشهای آن را بیان نماید. در فصل اول تعاریف و مفاهیم اولیه داده‌کاوی مطرح شده است. فصل پیش‌پردازش شامل روشهای متعددی در مورد آماده‌سازی داده‌ها و پیش‌پردازش وجود دارد. بخشهایی از این فصل مانند بخش کاهش بُعد مفاهیم پیشرفته‌ای در زمینه پیش-

پردازش داده‌ها مطرح می‌کند که می‌توان مطالعه آن را به بعد از بخش دوم موکول کرد. در فصل قواعد تلازمی برای فهم مطلب اصلی می‌توان به مطالعه الگوریتم *Apriori* اکتفا نمود. فصل تحلیل خوشه‌ای و فصل دسته‌بندی و پیش‌بینی مهمترین فصول کتاب هستند و لازم است کاملاً درک شوند. مباحث داده‌کاوی سریهای زمانی و تحلیل شبکه‌های‌اجتماعی جزو مباحث تکمیلی محسوب می‌شوند. انباره داده‌ها نیز بیشتر برای کسانی که با پایگاه داده‌ها کار می‌کنند مناسب است. فصل انتهایی کتاب در مورد داده‌کاوی در بازاریابی و مدیریت روابط مشتری تا حد زیادی مستقل از فصول دیگر بوده و مناسب دانشجویان مدیریت بازرگانی، تجارت الکترونیک و MBA است. مخاطبان اصلی کتاب دانشجویان کارشناسی‌ارشد مهندسی و مدیریت می‌باشند. البته مطالب کتاب برای دانشجویان مستعد کارشناسی نیز قابل استفاده است.

کتاب حاضر با بهره‌گیری از منابع علمی متنوع (کتاب، مقاله، سایتهای اینترنتی وحتی *Help* نرم افزار) سعی در پر کردن بخشی از خلأ موجود در این زمینه کرده است. معهذا، با وجود همه تلاشهای صورت گرفته کتاب حاضر الزاما خالی از اشکال نیست. نظرات ارشادی شما خواننده اندیشمند می‌تواند در کشف اشکالات احتمالی و رفع آنها در چاپهای بعدی به نویسندگان کمک نماید. لذا خواهشمند است نظرات خود را در خصوص ابهامات و اشکالات متن کتاب به آدرس *dmbook.iust@gmail.com* ارسال فرمائید.

تدوین این کتاب، حاصل چندین سال تدریس و برخورداری از دیدگاه‌ها و تلاشهای دانشجویان ساعی در خلال ترمهای مختلف بوده است. در اینجا برخود لازم می‌دانیم از تلاشهای صادقانه خانمها سمیرا ملک-محمدی، نگار رستگار و بنت‌الهدی‌علی‌احمدی، گلاره توحیدی و آقایان فرزاد وزیری، عیسی چمبر، هیوا فاروقی، و سلمان هوشمند قدردانی نماییم. در انتها از همکاری کلیه کارکنان و مسئولین انتشارات دانشگاه علم وصنعت ایران که نهایت همکاری را در چاپ این کتاب با نویسندگان داشته‌اند صمیمانه سپاسگزاریم. با آرزوی موفقیت و به‌کارگیری عملی داده‌کاوی برای افزایش کارآیی تصمیمات و برنامه های اجرایی کشور.

مهدی غضنفری، سمیه علیزاده، بابک تیمور پور

تابستان۸۶

بخش اول

داده‌کاوی و آماده‌سازی داده‌ها

فصل اول: مقدمه‌ای بر داده‌کاوی

فصل دوم: پیش‌پردازش داده‌ها

فصل اول

مقدمه‌ای بر داده‌کاوی

«کشف دانش و داده‌کاوی[1]» یک حوزه جدید میان رشته‌ای[2] و در حال رشد است که حوزه‌های مختلفی همچون پایگاه داده، آمار، یادگیری ماشین و سایر زمینه‌های مرتبط را با هم تلفیق کرده تا اطلاعات و دانش ارزشمند نهفته در حجم بزرگی از داده‌ها را استخراج نماید. با رشد سریع کامپیوتر و استفاده از آن در دو دهه اخیر تقریباً همه سازمانها حجم عظیمی داده در پایگاه داده‌شان ذخیره کرده‌اند. این سازمانها نیاز به فهم داده‌های خود و یا کشف دانش مفید از داده‌ها به‌صورت الگو یا مدل دارند.

---

[1]- Knowledge Discovery and Data Mining (KDD)
[2]- Interdisciplinary

مروری بر کشف دانش و داده‌کاوی

کشف دانش و داده‌کاوی

همان‌طورکه الکترونها و امواج موضوع اصلی مهندسی برق شدند، داده‌ها[۱]، اطلاعات[۲] و دانش[۳] نیز موضوع اصلی حوزه جدیدی از تحقیق و کاربرد به نام «کشف دانش و داده‌کاوی» یا به اختصار *KDD* هستند.

به‌طور کلی، *داده‌ها* رشته‌ای از بیتها (صفر و یک) یا اعداد و نشانه‌ها و یا اشیاء[۴] هستند که وقتی در فرمتی مشخص به یک برنامه ارسال می‌شوند، معنا می‌یابند (ولی هنوز تفسیر نشده‌اند). اطلاعات، داده‌ای است که موارد افزونه یا زایدش[۵] حذف شده است و به حداقل ممکنی که برای تصمیم‌گیری لازم است، تقلیل یافته است (حال داده‌ها تفسیر شده‌اند). *دانش* اطلاعات تلفیق شده‌ای است که شامل حقایق[۶] و روابط میان آنها است. دانش در واقع به‌عنوان تصاویر ذهنی ما درک، کشف یا فراگیری شده است. به‌عبارت دیگر می‌توان دانش را همان داده‌هایی فرض کرده که در بالاترین سطح تعمیم قرار گرفته‌اند.

متخصصانی که از حوزه‌های مختلف به رشد این موضوع جدید کمک می‌کنند فهم متفاوتی از عبارات «کشف دانش» و«داده‌کاوی» دارند. تعریف مورد نظر در این فصل به شرح زیر است:

کشف دانش از پایگاه داده‌ها در واقع فرایند تشخیص الگوها[۷] و مدلهای موجود در داده‌هاست. الگوها و مدلهایی که معتبر، بدیع[۱]، بالقوه، مفید و کاملاً قابل فهم هستند.

---

[1]- Data
[2]- Information
[3]- Knowledge
[4]- Objects
[5]- Redundancy
[6]- Facts
[7]- Patterns

داده‌کاوی مرحله‌ای از فرایند کشف دانش است که با کمک الگوریتمهای خاص داده‌کاوی و با کارایی قابل قبول محاسباتی، الگوها یا مدلها را در داده‌ها پیدا می‌کند.
به‌عبارت دیگر، هدف کشف دانش و داده‌کاوی یافتن الگوها و یا مدلهای جالب موجود در پایگاه داده‌ها است که در میان حجم عظیمی از داده‌ها مخفی هستند.
در طول این فصل ایده‌های[۲] گوناگونی بر روی یک پایگاه داده واقعی (در مورد التهاب مغزی[۳] که در مؤسسه تحقیقات پزشکی دانشگاه پزشکی و دندانپزشکی توکیو از سال ۱۹۷۹ تا ۱۹۹۳ جمع‌آوری شده) طرح خواهند شد.
این پایگاه داده حاوی داده‌های بیمارانی است که دچار التهاب مغزی بوده و در بخش اورژانس و عصب شناسی بیمارستانهای مختلفی پذیرفته شده‌اند. جدول (۱-۱) ویژگیها یا فیلدهای این پایگاه داده را نشان می‌دهد. در ادامه دو رکورد مربوط به بیماران این پایگاه داده که ترکیبی از داده‌های عددی و طبقه‌ای[۴] و نیز مقادیر مفقوده[۵] (با علامت؟ مشخص شده‌اند) هستند، مشاهده می‌شوند.

,−,−,۱۵,۰,۱,۲,۳۷, *Subacute*,۰,۰,۰,۱۰,۱۰,۰, *M*, *Abscess*, *Bacteria*,۱۰, *F*,−,٤۹,۹۷,۷۱۲,
۲۱۸٤,۲۸۵۲, *Abnormal*, *Abnormal*,−,۰,۲,٦۰۰۰, *Negative*, *N*, *N*, *N*,۲۱۳۷, *Multiple*,?,

,−,−,۱۵,۰,۱,۲,۳۸/۵, *Acute*,۰,۰,۰,۵,۵,۰, *M*, *Bacteria*, *Virus*,۱۲, *F*,−, *ABPC* + ۵۹,
۷۱,٤۰۰,٦۸۰,۱۰۸۰, *Normal*, *Abnormal*,+,۰,٤,۱۰۷۰۰, *Negative*, *N*, *N*, *N*۷۰*CZX*,?,

---

[1]- Novel
[2]- Notions
[3]- Meningitis
[4]- Categorical
[5]- Missing Value

جدول ۱- ۱) ویژگیها در پایگاه داده التهاب مغزی

تعداد ویژگیها	نوع ویژگی	طبقه
۰۷	عددی و طبقه‌ای	سابقه بیماری
۰۸	عددی و طبقه‌ای	معاینه فیزیکی
۱۱	عددی	معاینه آزمایشگاهی
۰۲	طبقه‌ای	تشخیص پزشکی
۰۲	طبقه‌ای	معالجه
۰٤	طبقه‌ای	دوره بستری
۰۲	طبقه‌ای	وضعیت نهایی
۰۲	طبقه‌ای	عامل ریسک
۳۸	جمع	

یک الگوی کشف شده از این پایگاه داده‌ها به زبان قواعد اگر- آنگاه به شکل زیر داده شده که کیفیت آن با درجه *اطمینان* ۸۷/۵٪ اندازه گرفته شده است:

اگر تعداد سلولهای چند هسته‌ای در $CFS \geq ۲۲۰$

و عامل ریسک = n

و از دست دادن هوشیاری = مثبت

و شروع حالت تهوع > ۱۵

آنگاه پیش‌بینی = ویروس (اطمینان =۸۷/۵٪)

در اینجا دانش یافته شده به زبان منطق ارائه شده است.

با توجه به تعریف ارائه شده از کشف دانش، «درجه جذابیت[1]» با معیارهای متعددی بیان می‌شود که به شرح زیرند:

تصدیق یا گواهی[2]، نشانگر معنی‌دار بودن «الگوی کشف شده» برحسب یک معیار آماری است. *افزونگی* مقدار شباهت یک الگوی کشف شده نسبت به الگوهای دیگر

[1]- Degree of Interest
[2]- Evidence

است و درجه تبعیت آن از دیگری را اندازه می‌گیرد. *فایده*[1]، ارتباط الگوی کشف شده با اهداف کاربران را بیان می‌کند. *بدیع بودن*[2] بیانگر میزان تازگی نسبت به دانش قبلی کاربر یا سیستم است. *سادگی*[3] به پیچیدگی نحوی[4] و نمایش یک الگوی کشف شده و نحوه تعمیم آن اشاره دارد.

تعاریف کشف دانش/ داده‌کاوی

برخی از تعاریف متداول از کشف دانش و داده‌کاوی به شرح زیر می‌باشد:

- تحلیل داده‌های توصیفی کامپیوتری، در مجموعه‌های بزرگ و پیچیده داده‌ها[۱۲].
- تحلیل ثانوی[5] مجموعه‌های بزرگ داده[۷].
- پرس و جوی الگو در پایگاه داده‌ها[۱۳]. این دیدگاه بر مشابهت جستجوی الگوها با پرس‌وجوهای انجام شده توسط سیستمهای مدیریت پایگاه داده‌ها تأکید می‌کند.
- کشف دانش، فرایند غیربدیهی[6] تشخیص الگوهای متعبر، نو، مفید و نهایتاً قابل درک در داده‌ها است. [۵]
- ویرایشی از یادگیری ماشین که به مجموعه‌های بزرگ داده اعمال شده و علاوه بر یادگیری با ناظر، طیف وسیعتری از وظایف و روشهای بدون ناظر را نیز در بر می‌گیرد.
- داده‌کاوی، آمار در مقیاس و سرعت است[۱٤].
- داده‌کاوی یک حوزه میان‌رشته‌ای و با رشد سریع است که حوزه‌های مختلفی همچون پایگاه داده، آمار، یادگیری ماشینی و سایر زمینه‌های مرتبط را با هم تلفیق

---

[1]- Usefulness
[2]- Novelty
[3]- Simplicity
[4]- Syntactical
[4]– ثانوی به این معنا است که منظور اصلی کسب و کار از جمع‌آوری پایگاه داده‌ها، کشف دانش نبوده است.
[6]- Nontrivial

کرده است تا اطلاعات و دانش ارزشمند نهفته در حجم بزرگی از داده‌ها را استخراج نماید [۲].

- داده‌کاوی، اکتشاف و تحلیل حجم زیادی از داده‌ها برای کشف الگوها و قواعد معنادار است. فرایند داده‌کاوی گاهی کشف دانش نیز نامیده می‌شود. ترجیح ارائه‌کنندگان [٦] این تعریف بر استفاده از اصطلاح خلق دانش است. [۳]
- داده‌کاوی به معنای استخراج یا کاوش[1] دانش از حجم عظیمی از داده‌ها می‌باشد داده‌کاوی الگوهای جالب را در میان حجم بزرگی از داده‌ها می‌یابد.

کشف دانش در پایگاه داده‌ها

از دیدگاه منطق، دانش هر حقیقت صریحاً اظهار شده و موجه در یک زمینه است که با زبانهای رسمی ارائه شده است. کشف دانش، گزاره‌هایی را تولید می‌کند که اشیاء جهان حقیقی، مفاهیم و نظمها را توصیف می‌کنند. پایگاه‌های داده، مخازنی ساخت‌یافته از داده‌ها درباره زمینه‌های مختلف دنیای واقعی می‌باشند. *KDD* بیش از تحلیل داده‌ها و فراتر از کشف الگو در آنها است. بسیاری از الگوهای موجود در داده‌ها، دانشی در زمینه بیان شده توسط داده‌ها ارائه نمی‌کنند.

شاپیرو [۱۱] که در سال ۱۹۸۹ واژه *KDD* را ابداع کرده است می‌گوید: «واژه *KDD* در جامعه هوش مصنوعی و یادگیری ماشین متداول شد. البته محققان پایگاه داده‌ها در موضع بهتری برای گفتمان با اهل کسب و کار و رسانه‌ها بودند و واژه داده‌کاوی در اخبار کسب و کار بسیار متداول‌تر شد.» داده‌کاوی واژه‌ای قدیمی‌تر از *KDD* است که در جامعه تحلیل داده‌های آمارمحور، ابداع شده است.

داده‌کاوی، واژه‌ای گمراه‌کننده است. در واقع طلا کاوش می‌شوند نه غبار یا سنگ و خاک. دانش‌کاوی تمثیل بهتری است زیرا مانند طلا، خروجی مورد نظر، دانش است. برخی، داده‌کاوی را گامی مرکزی در فرایند کشف دانش می‌دانند که الگوریتمهای

---

[1]- Mining

استخراج و اثبات فرضیه را اعمال می‌کند [۵]. این تعبیر مورد پذیرش عموم در جامعه این حوزه نیست. بسیاری داده‌کاوی را معادل با واژه متداول کشف دانش ممکن است در پایگاه داده‌ها می‌دانند. با اینکه به نظر می‌رسد واژه کشف دانش مناسبتر باشد، از این به بعد کشف دانش و داده‌کاوی را معادل فرض می‌کنیم.

فرایند کشف دانش

فرایند کشف دانش ذاتاً شامل مراحل متعددی مطابق با شکل (۱-۱) است.

اولین قدم: درک حوزه کاربرد مورد نظر و نحوه رابطه بندی مسئله است. این قدم به وضوح پیش نیاز استخراج دانش مفید و *انتخاب* روشهای داده‌کاوی مناسب در قدم سوم، با توجه به هدف کاربرد و طبیعت داده‌ها است.

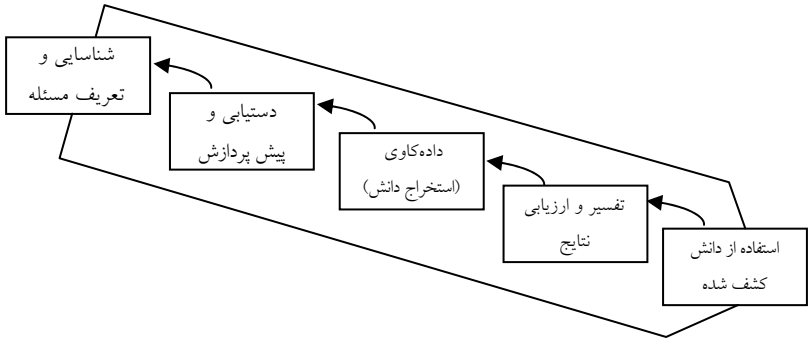
قدم دوم: جمع‌آوری و پیش پردازش داده‌ها[۱] شامل انتخاب منابع داده، حذف نقاط پرت[۲] یا مغشوش[۳]، طرز برخورد با داده‌های مفقوده[۴] و تبدیل[۵] (گسسته سازی[۶] در صورت نیاز) و کاهش[۷] داده‌ها است. این مرحله معمولاً در کل فرایند *KDD* بیشترین زمان را می‌برد.

قدم سوم: داده‌کاوی است که هدف آن استخراج الگوها و یا مدلهای مخفی در داده‌ها است. مدل را می‌توان به شکل زیر بیان نمود: «مدل یک بازنمایی سراسری[۸] از ساختاری است که یا اجزایِ سیستم در برگیرنده داده‌ها را خلاصه کرده و یا چگونگی رخداد داده را توصیف می‌کند» در مقابل، «یک الگو، ساختاری محلی است که شاید فقط به چند متغیرِ محدود و تعدادی مشاهده مرتبط است.»

---

[1]- Preprocess
[2]- Outliers
[3]- Noise
[4]- Missing Data
[5]- Transformation
[6]- Discrimination
[7]- Reduction
[8]- Global Representation

روشهای اصلی داده‌کاوی عبارتند از: مدلسازی برای پیش‌بینی[۱] (مثل دسته‌بندی[۲] و رگرسیون[۳])، بخش‌بندی یا تقطیع[٤] (خوشه‌بندی)[٥]، مدلسازی وابستگی[٦] (مانند مدلهای تصویری یا تخمین چگالی)، تلخیص[۷] (مانند پیدا کردن رابطه بین فیلدها، تلازم یا انجمنی[۸]، مصورسازی[۹]) و مدلسازی یافتن تغییر و انحراف[۱۰] در داده و دانش.



شکل ۱- ۱) فرایند KDD

قدم چهارم: تفسیرِ (یا پس پردازش[۱۱]) دانش کشف شده است. این تفسیر، به خصوص شامل توصیف[۱۲] و پیش‌بینی است که دو هدف اصلی سیستمهای اکتشافی می‌باشند. تجربه نشان داده است که همیشه الگوها یا مدلهای کشف شده از داده‌ها مفید و جالب نیستند بنابراین فرایند *KDD* فرایندی تکراری[۱۳] می‌باشد. یک راه استاندارد ارزیابی

---

[1]- Predictive
[2]- Classification
[3]- Regression
[4]- Segmentation
[5]- Clustering
[6]- Dependency
[7]- Summarization
[۸]- معادل کلمه Association از واژه تلازم استفاده شده است که در منطق و فلسفه اسلامی به‌کار می‌رود. تلازم و ملازمه هر دو به معنای لزوم طرفین می‌باشند در حالی که استلزام تنها رابطه یک طرفه را می‌رساند. (کتاب درآمدی بر آموزش فلسفه، استاد محمدتقی مصباح)
[9]- Visualization
[10]- Deviation
[11]- Post-Process
[12]- Description
[13]- Iterative

قواعد استنتاج شده[1] تقسیم داده‌ها به دو مجموعه آموزشی و آزمایشی است. می‌توان این فرایند را بارها با تقسیمات مختلف تکرار کرد و میانگین نتایج را برای تخمین عملکرد قواعد در نظر گرفت.

قدم پنجم: استفاده عملی از دانش کشف شده است. برخی اوقات می‌توان از دانش کشف شده بدون کامپیوتری کردن آن استفاده کرد. در مواقع دیگر کاربر انتظار دارد دانش کشف شده از طریق یک برنامه کامپیوتری به‌کار گرفته شود. بی‌شک به‌کارگیری عملی نتایج فرایند کشف دانش هدف نهایی این فرایند است.

توجه کنید که فضای الگوها اغلب نامحدود است و شمارش[2] الگوها دربر گیرنده نوعی جستجو در این فضا است. کارایی محاسباتی محدودیت خاصی روی زیرفضای قابل بررسی توسط الگوریتم اعمال می‌کند. بخش داده‌کاوی در فرایند *KDD* به‌طور عمده دربردارنده ابزارهایی است که به کمک آنها الگوها از داده‌ها استخراج و شمارش می‌شوند.

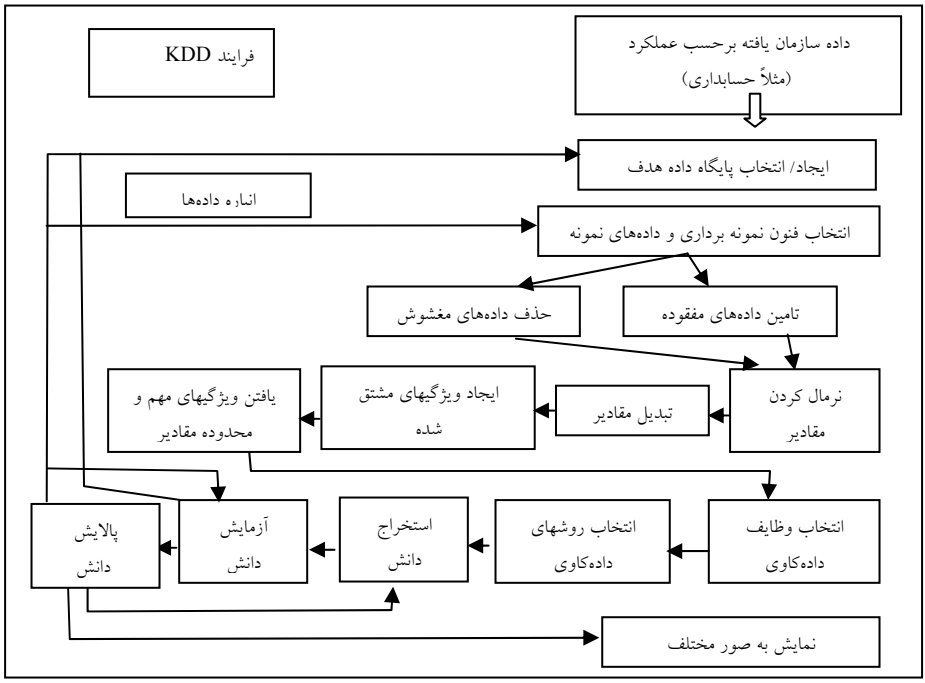
کشف دانش شامل ارزیابی و احتمالاً تفسیر الگوها برای تفکیک دانش از غیر دانش است. *KDD* همچنین شامل انتخاب طرحهای[3] کدبندی، پیش‌پردازش، نمونه‌گیری[4] و تصویر کردن[5] داده قبل از مرحله داده‌کاوی است.

عناوین دیگری که در گذشته به‌جای داده‌کاوی استفاده می‌شدند عبارتند از: باستان‌شناسی داده[6]، لایروبی داده[7]، تحلیل وابستگی تابعی و درو کردن داده[8].

جزئیات وظایف مربوط به فرایند *KDD* که در شکل (۱-۲) آمده، در زیر تشریح شده است:

1- Induced Rules
2- Enumeration
3- Schemes
4- Sampling
5- Projections
6- Data Archaeology
7- Data Dredging
8- Data Harvesting

- درک کامل حوزه کاربرد: شامل درک دانش پیشین مرتبط، اهداف کاربرنهایی و غیره می‌باشد.
- ایجاد مجموعه داده‌های هدف: انتخاب مجموعه داده‌ها یا تمرکز روی زیرمجموعه‌ای از متغیرها یا نمونه‌های داده که قرار است روی آنها اکتشاف انجام شود، ایجاد مجموعه داده‌های هدف نامیده می‌شود.[1]



شکل ۱- ۲) وظایف فرایند KDD

- پیش- پردازش یا پاکسازی داده[2]: عملیات مقدماتی مثل حذف اغتشاش یا نقاط پرت، جمع کردن اطلاعات لازم برای مدل کردن یا مقابله با اغتشاش، تصمیم‌گیری

[1]- Refine
[2]- Data Cleaning Preprocessing

در مورد چگونگی رفتار با داده‌های مفقوده، در نظر گرفتن توالی زمانی و تغییرات شناخته شده در اطلاعات، پاکسازی داده‌ها نامیده می‌شود.

- کاهش داده‌ها و تصویر کردن آنها: یافتن مشخصه‌های مفید برای نمایش داده بسته به هدف وظیفه و استفاده از روشهای کاهش بُعد یا تبدیل برای کاهش تعداد مؤثر متغیرهای مورد نظر یا پیدا کردن نمود مناسب و معادل داده‌ها، کاهش داده‌ها نامیده می‌شود.
- انتخاب عملیات داده‌کاوی: تصمیم‌گیری در مورد هدف فرایند *KDD* که می‌تواند دسته‌بندی، رگرسیون، خوشه‌بندی یا غیره باشد. عملیات مختلف الگوریتم داده‌کاوی به‌طور مفصل در فصل‌های بعدی تشریح می‌شوند.
- انتخاب روشهای داده‌کاوی: این گام شامل انتخاب روشهای جستجوی الگوها در داده‌ها بوده و شامل انتخاب مدلها و پارامترهای مناسب تطابق یک روش داده‌کاوی خاص با معیارهای کلی فرایند *KDD* است. برای مثال در مورد اول مدل مورد استفاده در داده‌های طبقه‌ای با مدل‌های بردارهای اعداد حقیقی متفاوت بوده و در مورد دوم ممکن است کاربر نهایی علاقه‌مند به درک مدل بوده و به قابلیتهای پیش‌بینی آن علاقه‌ای نداشته باشد.
- داده‌کاوی برای استخراج الگوها/ مدلها: در این گام به جستجوی الگوهای مورد نظر به یک یا چند شکل خاص (قواعد یا درختان طبقه‌بندی، رگرسیون، خوشه‌بندی و مانند آن) پرداخته می‌شود. کاربر با انجام درست مراحل قبل می‌تواند کمک بسیاری به روش داده‌کاوی کند.
- تفسیر و ارزیابی الگوها/ مدلها: لازم است الگوها و مدلهای مختلف به منظور استفاده بعدی مورد ارزیابی و تفسیر قرار گیرند.

- تثبیت[1] دانش کشف شده: ترکیب این دانش با سیستم اجرایی یا حداقل مستندسازی و گزارش آن به گروه‌های علاقه‌مند، تثبیت دانش نامیده می‌شود این کار شامل بررسی و حل تضادهای[2] بالقوه این دانش با دانشهای مورد قبول (یا کشف شده) پیشین می‌شود.

ممکن است میان هر قدم و قدم قبلی آن عملاً نوعی تکرار رخ دهد.

نوع داده‌ها

می‌توان داده‌ها را از نگاه پایگاه داده‌ها به دسته‌های تراکنشی[3]، شیء- رابطه‌ای[4]، زمانی[5]، فضایی یا مکانی[6]، متنی، چندرسانه‌ای، جریانی[7] و پیوندی[8] تقسیم نمود [٦]

حوزه‌ها، وظایف و عملکردهای داده‌کاوی

KDD یک حوزه میان رشته‌ای است که با موضوعات زیر مرتبط است: آمار، یادگیری ماشین، پایگاه داده، الگوریتمها، مصورسازی، محاسبات موازی و کسب دانش[9] برای سیستمهای خبره. سیستمهای *KDD*، مبتنی بر روشها و الگوریتمهای این حوزه‌ها می‌باشند. هدف مشترک همهٔ آنها استخراج دانش از داده‌ها در محیط پایگاه‌های بزرگ داده است.

حوزه‌های یادگیری ماشین[10] و تشخیص الگو[11] در مباحث مربوط به نظریه‌ها و الگوریتمهای استخراج الگو از داده‌ها (عمدتاً روشهای داده‌کاوی) با *KDD* اشتراک

---

[1]- Consolidation
[2]- Conflicts
[3]- Transactional
[4]- Object-Relational
[5]- Temporal
[6]- Spatial
[7]- Streams
[8]- Link
[9]- Knowledge Acquisition
[10]- Machine Learning
[11]- Pattern Recognition

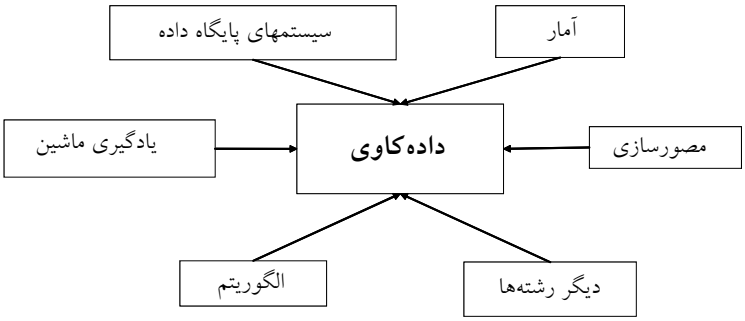
دارند. *KDD* روی توسعه این نظریه‌ها و الگوریتمها متمرکز شده است تا امکان یافتن الگوهای خاص (آنهایی که به‌عنوان دانش مفید یا جالب مدنظرند) را در مجموعه‌های بزرگ داده فراهم سازد.

KDD اشتراک زیادی نیز با آمار به خصوص تحلیل پوششی داده‌ها[1] دارد. سیستمهای *KDD* معمولاً از رویه‌های آماری خاصی برای مدلسازی داده و بررسی اغتشاش استفاده می‌کنند.

حوزهٔ مرتبطِ دیگر، انباره‌داده‌ها[2] است که به روندهای متداول سیستمهای اطلاعات مدیریت[3] برای جمع‌آوری و پاکسازی داده‌های تراکنشی و قابل دسترسی کردن آن برای بازیافت فوری مربوط است. یک رویکرد متداول برای تحلیل انباره‌های داده، پردازش تحلیلی بلادرنگ (*OLAP*) نام دارد. ابزارهای *OLAP* روی تحلیل چند بعدی داده متمرکز می‌شوند. این رویکرد در تدارک خلاصه‌ها و حرکت در طول ابعاد متعدد نسبت به رویکرد *SQL* (زبان پرسش استاندارد)[4] ارجح است. کشف دانش و *OLAP* دو جنبه مرتبط نسل جدید ابزارهای هوشمند استخراج و مدیریت دانش هستند. همان‌طورکه در تعریف داده‌کاوی گفته شد، داده‌کاوی یک حوزه میان‌رشته‌ای است که حوزه‌های مختلفی همچون پایگاه داده، آمار، یادگیری ماشینی و سایر زمینه‌های مرتبط را با هم تلفیق می‌کند.

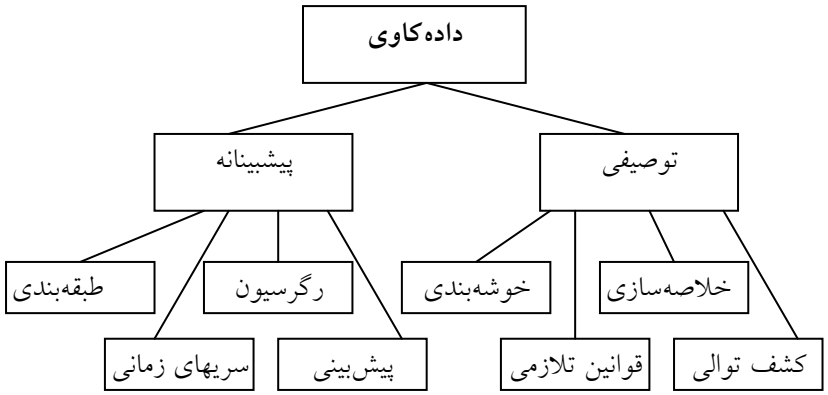
---

[1]- Exploratory Data Analysis (EDA)
[2]- Data Warehouse
[3]- Management Information System (MIS)
[4]- Standard Query Language



شکل ۱- ۳) حوزه‌های مختلف داده‌کاوی [٦]

وظایف اصلی داده‌کاوی دو دسته می‌باشد: توصیفی[1] و پیشبینانه. وظایف توصیفی خواص عمومی داده‌ها را مشخص می‌کنند. هدف از توصیف، یافتن الگوهایی در مورد داده‌هاست که برای انسان قابل تفسیر باشد. وظایف پیشبینانه به منظور پیش‌بینی رفتارهای آیندهٔ آنها استفاده می‌شوند. منظور از پیش‌بینی به‌کارگیری چند متغیر یا فیلد در پایگاه داده برای پیش‌بینی مقادیر آینده یا ناشناخته دیگر متغیرهای مورد علاقه است. عملکردهای داده‌کاوی در شکل (۱-٤) نشان داده شده‌اند.



شکل ۱- ٤) عملکردهای داده‌کاوی [٤]

[1]- Descriptive

از نظر هن و کمبر [٦] عملکردهای اصلی داده‌کاوی عبارتند از:

توصیف مفهوم/دسته: مشخص کردن و تمایز.

کاوش الگوهای مکرر، تلازم (انجمنی، تداعی‌گر) و همبستگی.

دسته‌بندی: دسته‌بندی، فرایند یافتن مدلی است که با تشخیص دسته‌ها یا مفاهیم داده می‌تواند دسته ناشناخته اشیاء دیگر را پیش‌بینی کند. دسته‌بندی یک تابع یادگیری است که یک قلم داده را به یکی از دسته‌های از قبل تعریف شده نگاشت می‌کند. داده‌های موجود به دو قسمت آموزش و آزمون تقسیم می‌شوند. داده‌های آموزش برای یادگیری قواعد توسط سیستم استفاده می‌شوند و داده‌های آزمون برای بررسی دقت دسته‌بندی و جلوگیری از بیش‌برازش به کار می‌روند. برخی روشهای متداول دسته‌بندی عبارتند از:

- درخت تصمیم‌گیری: *C 4.5*، *CART*.
- دسته‌بندی بیزی: دارای دو نوع بیز ساده و شبکه‌های بیزی است.
- شبکه عصبی پس‌انتشار.
- ماشینهای بردار پشتیبان[1].
- دسته‌بندی تلازمی.
- یادگیرندگان ;کاهل: نزدیک‌ترین همسایگان، استدلال مبتنی بر مورد.
- روشهای دیگر: ژنتیک، مجموعه‌های نادقیق، مجموعه‌های فازی.

پیش‌بینی: دسته‌بندی، برچسبهای[2] طبقه‌ای (گسسته، بدون ترتیب) را پیش‌بینی می‌کند. در حالی که پیش‌بینی، توابع مقدار پیوسته را مدل می‌کند.

- رگرسیون: خطی، غیر خطی
- شبکه عصبی، ماشینهای بردار پشتیبان

---

[1] Support Vector Machine (SVM)
[2]- Label

خوشه‌بندی: خوشه‌بندی به معنای تقسیم داده‌ها به گروه‌های مشابه است. داده‌ها بر اساس اصل حداکثر کردن شباهت داخل گروه‌ها و حداقل کردن شباهت بین گروه‌ها، خوشه‌بندی می‌شوند. خوشه‌بندی یک روش متداول توصیفی است که در جستجوی تشخیص تعداد محدودی خوشه برای توصیف داده‌ها است [۹]. خوشه‌ها ممکن است مانع (متقابلاً ناسازگار)[1] و جامع[2] بوده و یا دارای نمایشی غنی‌تر مانند نمایش سلسله مراتبی یا وضعیت هم‌پوشانی[3] باشند. مثالهای خوشه‌بندی در یک موضوع کشف دانش عبارتند از کشف زیرگروه‌های همگنی از مصرف‌کنندگان در یک پایگاه داده بازاریابی و یا تشخیص زیرگروه‌های طیف در وسایل اندازه‌گیری فضایی مادون قرمز. خوشه‌بندی نه تنها داده‌های بدون برچسب را تحلیل می‌کند بلکه این برچسبها را نیز تولید می‌کند. روش‌های مختلف خوشه‌بندی عبارتند از:

- روشهای افرازی: *K*- میانگین، *K*- میانه، نقشه‌های خود سازمان[4] *(SOM)*، روشهای مبتنی بر مدل مانند *EM*.
- روشهای سلسله مراتبی[5]: تجمعی، تقسیمی.
- روشهای مبتنی بر چگالی[6].

تحلیل نقاط پرت

تحلیل تکامل: تحلیل تکامل با داده‌های متغیر در طول زمان کار می‌کند و روی آنها همه عملکردهای گفته شده قبلی را انجام می‌دهد. شامل تحلیل سریهای زمانی، تحلیل توالی و تحلیل مبتنی بر تشابه می‌باشد.

---

[1]- Exclusive
[2]- Exhaustive
[3]- Overlapping
[4]- SOM
[5]- Hierarchical
[6]- Density based

برخی مباحث دیگر نیز در داده‌کاوی مطرح هستند که با عملکردهای مطرح شده قبلی تداخل دارند[۸]:

تلخیص دربرگیرنده روشهایی برای یافتن یک توصیف فشرده از زیر مجموعه‌ای از داده‌هاست. مثال ساده‌ای از آن می‌تواند تهیه جدول میانگین و انحراف معیار برای تمام فیلدها باشد. روشهای پیچیده‌تر شامل استنتاج قواعد خلاصه، فنون مصورسازی چند متغیره و کشف رابطه تابعی بین متغیرها است. فنون تلخیص معمولاً در تحلیل دادهٔ اکتشافی و تولید گزارش خودکار به‌کار برده می‌شوند.

مدلسازی وابستگی شامل یافتن مدلی برای توصیف وابستگیهای معنی‌دار[۱] بین متغیرهاست. مدلهای وابستگی در دو سطح وجود دارند: سطح ساختاری[۲] مدل (اغلب با شکل) مشخص می‌کند که کدام متغیرها به‌طور محلی به دیگری وابسته‌اند در حالی‌که سطح کمّی[۳] مدل قدرت وابستگیها را با مقیاس عددی مشخص می‌کند. برای مثال شبکه‌های وابستگی احتمالی[٤] از استقلال شرطی برای مشخص کردن جنبه ساختاری مدل و از احتمالات یا همبستگیها[٥] برای تعیین قدرت وابستگی استفاده می‌کنند. شبکه‌های وابستگی احتمالی به‌طور فزاینده‌ای در کاربردهای کاملاً متفاوتی همانند توسعه سیستمهای خبره پزشکی احتمالی از پایگاه داده‌ها، بازیافت اطلاعات[٦] *(IR)* و مدلسازی ژن انسانی استفاده شده‌اند.

بازنمایی مدل از دیدگاه منطق به معنای زبانی همچون L است که الگوهای قابل کشف را تشریح می‌کند. اگر توانایی نمایش مدل خیلی محدود باشد آنگاه نه زمان طولانی آموزش و نه تعداد انبوه نمونه نمی‌توانند مدل دقیقی برای داده‌ها تولید کند. برای مثال نمایش درخت تصمیم با استفاده از گره‌های تقسیم تک متغیره (یک فیلد)، فضای

---

[1]- Significant
[2]- Structural
[3]- Quantitative
[4]- Probabilistic Dependency Network
[5]- Correlation
[6]- Information Retrieval

ورودی را به ابرصفحه‌های موازی با محورهای ویژگیها تقسیم می‌کند. این درخت تصمیم، علیرغم اینکه همه داده‌های آموزشی برای آن فراهم می‌شود، نمی‌تواند از داده‌ها رابطه $x = y$ را کشف کند. بنابراین مهم است که یک تحلیلگر داده به‌طور کامل مفروضات بازنمایی[1] که ممکن است مختص یا ذاتی[2] یک روش خاص باشد، را درک کند. علاوه بر آن مهم است که یک طراح الگوریتم به وضوح مفروضات نمایشی یک الگوریتم خاص را مشخص کند.

ارزیابی مدل تعیین می‌کند که یک الگوی خاص (یک مدل و پارامترهایش) چقدر با معیارهای فرایند *KDD* تطابق دارد. ارزیابی دقت پیش‌بینی (یا اعتبار مدل) مبنی‌بر اعتبار سنجی متقاطع[3] است. ارزیابی «کیفیت توصیف» شامل دقت پیش‌بینی، بدیع بودن، مطلوبیت و قابل فهم بودن مدل برازش شده می‌شود. هر دو معیار منطقی و آماری را می‌توان برای ارزیابی مدل به‌کار برد. برای مثال اصل حداکثر درستنمایی[4]، پارامترهایی را برای مدل انتخاب می‌کند که بهترین برازش را روی داده‌های آموزشی می‌دهند.

روش جستجو شامل دو بخش است: جستجوی پارامتر و جستجوی مدل. در جستجوی پارامتر، الگوریتم باید پارامترهایی را جستجو کند که معیارهای ارزیابی مدل را با توجه به داده‌های مشاهده شده و نمایش مدل بهینه می‌کنند. جستجوی مدل به شکل حلقه تکراری روی روش جستجوی پارامتر عمل می‌کند: نمایش مدل عوض می‌شود تا یک گروه از مدلها در نظر گرفته شوند. برای نمایش هر مدل مشخص، روش جستجوی پارامتر اجرا می‌شود تا کیفیت آن مدل ارزیابی شود. پیاده‌سازی روشهای جستجوی مدل معمولاً ابتکاری هستند زیرا اندازه فضای مدلهای ممکن عملاً

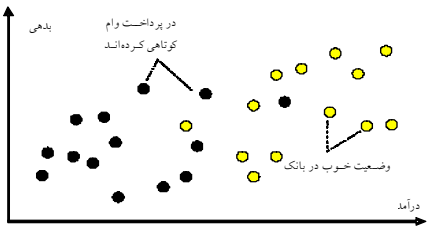
---

[1]- Representational Assumptions
[2]- Inherent
[3]- Cross Validation
[4]- Likelihood

جستجوی جامع را ناممکن می‌کند و دستیابی به راه‌حلهای شکل بسته[1] نیز به آسانی مقدور نیست.

مثالهایی از روشهای داده‌کاوی

شکل (۱–۵) مجموعه داده مصنوعی دو بعدی شامل ۲۳ مورد را نشان می‌دهد.

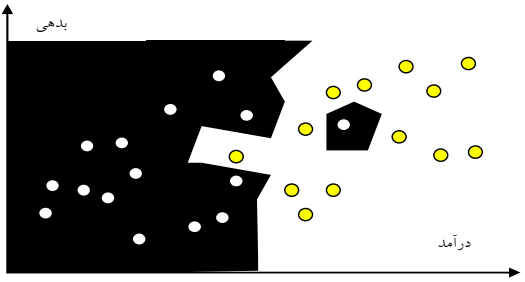


شکل ۱– ۵) یک مجموعه داده ساده با دو کلاس به منظور نمایش مسئله

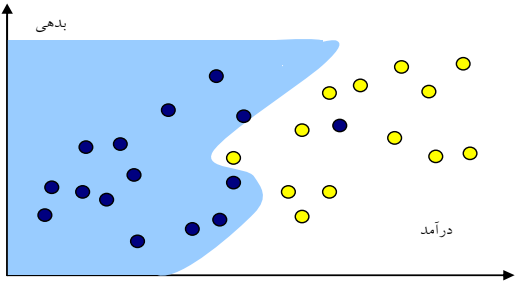
هر نقطه روی شکل نشانگر یک مشتری است که در گذشته از بانک مشخصی وام گرفته است. داده‌ها به دو دسته تقسیم شده است. افرادی که در پرداخت وام کوتاهی کرده‌اند و افرادی که وضعیت پرداخت وامشان خوب است.

دسته‌بندی: اشکال (۱–٦) و (۱–۷) دسته‌بندی داده‌های مربوط به مسئله وام را در دو دسته نشان می‌دهد. توجه کنید که جداسازی کامل دسته‌ها به کمک یک مرز تصمیم‌گیری خطی ممکن نیست. بانک ممکن است بخواهد با استفاده از نواحی دسته‌بندی شده برای تصمیم‌گیریِ خودکار، در مورد دادن یا ندادن وام به متقاضیان استفاده کند.

[1]- Closed Form



شکل ۱– ٦) مرزهای دسته‌بندی به روش نزدیکترین همسایه

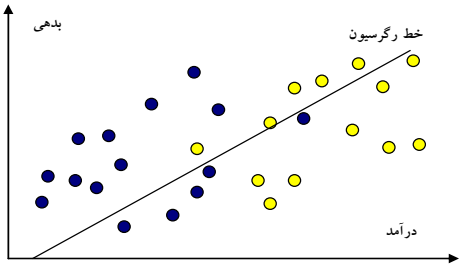


شکل ۱– ۷) مثالی از مرزهای دسته‌بندی از یک دسته‌بندی کننده غیرخطی

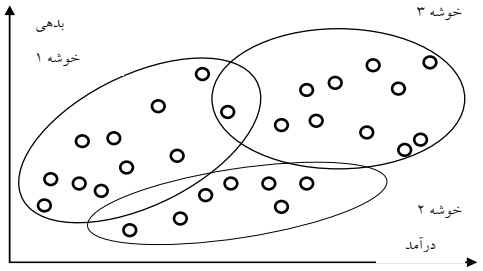
پیش‌بینی یا رگرسیون: رگرسیون یک تابع یادگیری است که یک قلم داده را به یک متغیر پیش‌بینی با مقدار حقیقی نگاشت می‌کند. رگرسیون کاربردهای بسیاری دارد. مثلاً پیش‌بینی مقدار جرم حیاتی[1] موجود در یک جنگل از روی اندازه‌گیری راه دور میکرو موج، تخمین احتمال مرگ یک بیمار از روی نتایج آزمایشهای تشخیص بیماری، پیش‌بینی تقاضای مشتری برای محصول جدید به‌عنوان تابعی از هزینه تبلیغات و بالاخره پیش‌بینی سریهای زمانی وقتی که متغیرهای ورودی همان متغیرهای پیش‌بینی هستند که در زمان قبل واقع شده یا اصطلاحاً تأخیری[2] هستند. شکل (۱–۸) نتایج یک رگرسیون خطی ساده را نشان می‌دهد. که در آن «جمع بدهی» به‌عنوان تابعی خطی از

[1]- Biomass
[2]- Time- Lagged

«درآمد» برازش شده است: این برازش خوب نیست زیرا همبستگی ضعیفی بین دو متغیر وجود دارد.



شکل ۱– ۸) یک رگرسیون خطی ساده برای مجموعه داده وام



شکل ۱– ۹) یک خوشه بندی ساده از داده وام به سه خوشه

خوشه‌بندی: شکل (۱–۹)، سه خوشه از داده‌های مربوط به وام مشتریان را نشان می‌دهد. توجه کنید که خوشه‌ها همپوشان هستند و اجازه تعلق نقاط داده به بیش از یک خوشه را می‌دهند. برچسبهای دسته‌های اولیه (که دایره سیاه و سفید نشان داده شده بود) با دوایر «توخالی» جایگزین شده‌اند تا نشان دهد که دیگر عضویت انحصاری به خوشه‌ها مطرح نیست.

چرا *KDD* لازم است؟

دلایل بسیاری نیاز به *KDD* را توضیح می‌دهند:

- بسیاری از سازمانها داده‌های زیادی جمع کرده‌اند، با آن چه می‌کنند؟

- مردم داده‌ها را ذخیره می‌کنند زیرا فکر می‌کنند آنها بطور ضمنی حاوی دارایی با ارزشی هستند. در تحقیقات علمی، داده‌ها بیانگر مشاهداتی هستند که درباره پدیده‌های تحت مطالعه به دقت جمع‌آوری شده‌اند.
- داده‌ها در تجارت، اطلاعات مربوط به بازارهای حیاتی، رقبا و مشتریان را دربرمی‌گیرد. در ساخت، داده‌ها فرصتهای بهینه‌سازی و عملکرد بهتر را به موازات کلیدهایی برای بهبود فرایند و رفع مشکلات فراهم می‌آورند.
- تاکنون فقط بخش کوچکی (حدود ۵٪ تا ۱۰٪) از داده‌های جمع‌آوری شده تحلیل شده است.
- داده‌هایی که ممکن است هرگز تحلیل نشوند، با هزینه زیاد و به‌طور پیوسته جمع‌آوری می‌شوند تا اطمینان حاصل شود چیز بالقوه مهمی برای آینده از دست نمی‌رود.
- بدیهی است با توجه به نرخ داده‌ها، به‌کارگیری روشهای سنتی (که دستی و زمان هستند) برای تحلیل آنها کارساز نخواهد بود.
- حجم داده‌ها برای روشهای تحلیل کلاسیک بیش از اندازه بزرگ است. ممکن است نتوانیم آن را در حافظه نگه داریم و یا به‌طور جامع آن را تحلیل کنیم.
- تعداد بسیار زیاد رکوردها ($۱۰^{۱۲}$– $۱۰^{۸}$ بایت) و داده با بعد زیاد (تعداد زیادی فیلد: $۱۰^{۴}$– $۱۰^{۲}$ عوامل مهم دیگری هستند.
- چگونه می‌توان میلیونها رکورد و دهها یا صدها فیلد را کاوش کرد و الگوها را یافت؟
- شبکه‌سازی، فرصتی مناسب و رشد یابنده برای دسترسی بیشتر فراهم کرده است.
- به‌طور روزافزونی، کاوش بلادرنگ مشخصات کالاها، اطلاعات سفر و سایر خدمات بر روی اینترنت مورد نیاز است.
- کاربر نهایی، آماردان نیست.
- نیاز به تشخیص و پاسخ سریع به فرصتهای در حال ظهور، قبل از رقبا وجود دارد.

- ابزارهای ویژه مالی، عملیات بازاریابی هدف و غیره از لوازم کسب و کار است.
- با رشد پایگاه داده‌ها، توانایی انجام تحلیل و تصمیم‌گیری به کمک پرس و جوی سنتی *(SQL)* غیر ممکن می‌شود.
- بیان بسیاری از پرس و جوهای جالب (برای انسانها) با یک زبان پرس و جوی معمولی دشوار است مثلا: «تمام رکوردهای نشان‌دهنده تقلب[1] را برایم پیدا کن» یا «افرادی که احتمالاً محصول x را می‌خرند را پیدا کن» و یا «تمام رکوردهای شبیه به رکوردهای جدول x را پیدا کن».
- مشکل رابطه‌ای کردن پرس و جو وجود دارد. این مشکل با بهینه‌سازی پرس و جو قابل حل نیست و در حوزه پایگاه داده‌ها یا در روشهای کلاسیک آماری به آن توجه کافی نشده است.
- راه طبیعی، روش آموزش از طریق مثال است (مثلاً در یادگیری ماشین و تشخیص الگو)

کاربردهای *KDD*

در بسیاری از حوزه‌ها فنون *KDD* قابل به‌کار گرفتن هستند، برای مثال:

- اطلاعات کسب و کار
  - تحلیل داده‌های بازاریابی و فروش
  - تحلیل سرمایه‌گذاری
  - تأیید وام
  - تشخیص تقلب
- اطلاعات ساخت
  - کنترل و زمانبندی

[1]- Fraud

- مدیریت شبکه
- تحلیل نتایج آزمون

- اطلاعات علمی
  - فهرست برداری تحقیقات مربوط به آسمان
  - پایگاه داده توالی حیات[1]
  - زلزله یابی در زمین شناسی
- اطلاعات شخصی

چالشهایی برای *KDD*

- **پایگاه داده بزرگتر**: پایگاه داده با صدها فیلد و جدول، میلیونها رکورد و اندازه‌های چند میلیارد بایتی کاملاً متداول هستند و پایگاه داده ترابایتی ($۱۰^{۱۲}$ بایت) در حال پدیدار شدن هستند.
- **بُعد زیاد**: نه تنها اغلب تعداد زیادی رکورد در پایگاه داده وجود دارد بلکه تعداد زیادی فیلد (ویژگی، متغیر) ممکن است موجود باشند بنابراین مسئله دارای ابعاد زیادی است. یک مجموعه داده با بعد بالا مشکل‌زا است زیرا اندازه فضای جستجو برای استقراء مدل[2] را به‌طور ترکیبی[3] و انفجاری بزرگ می‌کند. به‌علاوه این مشکل یافتن شانسی الگوهای بدلی و جعلی[4] را که به‌طور کلی معتبر نیستند افزایش می‌دهد. چاره این مشکل استفاده از روشهای کاهش بعد مؤثر مسئله و استفاده از دانش پیشین برای تشخیص متغیرهای نامربوط است.
- **بیش- برازش**: وقتی الگوریتم به دنبال بهترین پارامترهای یک مدل خاص با استفاده از مجموعه محدودی داده می‌گردد، ممکن است داده‌ها را بیش‌برازش کند که منجر

---

[1]- Biosequence
[2]- Model Induction
[3]- Combinatorial
[4]- Spurious

به عملکرد ضعیف مدل روی داده‌های آزمون می‌شود. راههای ممکن شامل اعتبارسنجی متقاطع، تنظیم[1] و دیگر استراتژی‌های آماری پیچیده است.

- **تشخیص معنادار بودن آماری:** وقتی سیستم در جستجوی مدلهای متعددی است این مشکل (که مرتبط به بیش برازش است) رخ می‌دهد. برای مثال اگر یک سیستم N مدل را در سطح معنادار بودن ۰/۰۰۱ آزمون کند، آنگاه با داده‌های کاملاً تصادفی به‌طور متوسط $N/۱۰۰۰$ این مدلها به‌طور معناداری قبول می‌شوند. این نکته بسیاری اوقات در تلاشهای اولیه *KDD* نادیده گرفته می‌شود. یک راه غلبه بر این مشکل استفاده از روشهایی است که آمار آزمون را به‌عنوان تابعی از جستجو تنظیم می‌کنند.
- **داده‌ها و دانش در حال تغییر:** داده‌های سریعاً در حال تغییر (و بی‌ثبات[2]) ممکن است الگوهای کشف شده قبلی را بی‌اعتبار کنند. به‌علاوه متغیرهای اندازه‌گیری شده در یک پایگاه داده ممکن است با اندازه‌گیریهای جدید در طول زمان اصلاح، حذف و یا افزایش[3] یابند. راه‌حلهای ممکن عبارتند از: روشهای تدریجی برای به‌هنگام کردن الگوها، و برخورد با تغییر به‌عنوان یک فرصت کشف با به‌کاربردن آن به‌عنوان راهنمایی برای جستجوی خود الگوهای تغییر.
- **داده مفقوده و مغشوش:** این مشکل به خصوص در پایگاه داده‌های تجاری حاد است. داده‌های سرشماری[4] آمریکا نرخ خطایی تا ۲۰٪ دارند. اگر پایگاه داده از ابتدا با هدف کشف دانش طراحی نشده باشد ممکن است فاقد برخی ویژگیهای مهم باشد. راه حل ممکن به‌کار بردن استراتژیهای آماری پیچیده‌تر برای تشخیص متغیرها و وابستگی‌های مخفی است.

---

[1]- Regularization
[2]- Non Stationary
[3]- Augmented
[4]- Census

- **روابط پیچیده بین فیلدها:** ویژگیها یا مقادیر دارای ساختار سلسله مراتبی، روابط میان ویژگیها، و دیگر انواع روشهای پیچیده نمایش دانش نیاز به الگوریتمهایی دارد که بتوانند به‌طور مؤثر از این اطلاعات استفاده کند. الگوریتمهای داده‌کاوی به‌طور تاریخی برای رکوردهای «ویژگی- مقدار» ساده توسعه یافته‌اند. البته روشهای جدیدی برای عمل روی رابطه بین متغیرها در حال توسعه‌اند.
- **قابل درک بودن الگوها:** در بسیاری از کاربردهای داده‌کاوی، اینکه کشفیات برای انسان قابل فهم‌تر شوند، بسیار مهم است. راههای ممکن عبارتند از نمایش گرافیکی، ساختاربندی قواعد با گرافهای جهت‌دار غیردوری، به‌کارگیری زبان طبیعی و فنون مصورسازی داده و دانش.
- **تعامل با کاربر و دانش پیشین:** بسیاری از روشها و ابزارهای فعلی *KDD* واقعاً محاوره‌ای[1] نیستند و نمی‌توانند به آسانی دانش پیشین درباره یک مسئله (به‌جز در موارد ساده) در نظر بگیرند. استفاده از دانش حوزه مورد مطالعه در همه مراحل فرایند *KDD* مهم است.
- **تلفیق با سیستمهای دیگر:** یک سیستم اکتشاف دانش ممکن است به تنهایی مفید نبوده و بهتر باشد با سایر سیستم تلفیق یا یکپارچه شود. نمونه‌های تلفیق عبارتند از تلفیق با *DBMS*[2] (از طریق رابط پرس و جو)، تلفیق با صفحه گسترده‌ها و ابزارهای مصورساز و در برگیری حسگرهای زمان حقیقی.

---

[1]- Interactive
[2]- Data Base Management System

منابع

۱) یکی از مراجع اصلی اطلاعات درباره کشف دانش و داده‌کاوی سایت http://www.kdnuggets.com است که دارای خبرنامه ماهانه نیز می‌باشد.

2) ACM SIGKDD Curriculum Committee (2006) 'Data Mining Curriculum: A Proposal (Version 1.0)' (Online) Available from <URL:http://www-sal.cs.uiuc.edu/~hanj/kdd_curriculum.pdf>.

3) Berry M. J. A. and Linoff G. S. (2004) Data Mining Thechniques For Marketing, Sales, and Customer Relationship Management (2nd edn), Wiley.

4) Dunham M.H. (2002) Data Mining, Introductory and Advanced Topics, Prentice Hall.

5) Fayyad U., Piatetsky-Shapiro G., Smyth P., Uthurusamy R. (1996) Advances in Knowledge Discovery and Data Mining, MIT Press.

6) Han. J, Kamber. M. (2006) "Chapter 1:Introduction", Data mining concepts and techniques, 2nd edition, , Morgan Kaufmann Publishers.Han J. and Kamber M. (2006) Data Mining: Concepts and Techniques (2nd edn), Morgan Kaufmann.

7) Hand D. (1998) 'Data mining – reaching beyond statistics', Research in Official Stat. 1(2): 5-17.

8) Ho, T.B (nd) 'KNOWLEDGE DISCOVERY AND DATA MINING TECHNIQUES AND PRACTICE', Unesco Course (cited October 2004). Available from <URL:http://www.netnam.vn/unescocourse/knowlegde/know_frm.htm>.

9) Kaufman L. and Rousseeuw P. (1990) Finding Groups in Data: An Introduction to Cluster Analysis, John Wiley and Sons.

10) Klösgen W. and Żytkow J. M. (2002) Handbook of Data Mining and Knowledge Discovery, Oxford university press.

11) Shapiro G. P. (2000) 'Knowledge Discovery in Databases: 10 years after', ACM SIGKDD Explorations, Feb 2000, Volume 1, No 2

12) .Friedman, J. H. (1997) "Data Mining and Statistics. What's the Connection?", Proc. of the 29th Symposium on the Interface: Computing Science and Statistics, May 1997, Houston, Texas.

13) Imielinski T., Virmani A. (1999) "MSQL – query language for data mining applications" ,Data Mining and Knowledge Discovery Journal, December 1999.

14) Pregibon D. (1999) "2001: a statistical odyssey", Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining.

فصل دوم

آماده‌سازی داده‌ها در داده‌کاوی

مرحله آماده‌سازی داده‌ها مهم‌ترین و زمانبرترین مرحله در پروژه‌های داده‌کاوی است. از آنجا که داده‌ها در این پروژه‌ها ورودی پروژه هستند هر قدر این ورودی دقیق‌تر باشد، خروجی کار دقیق‌تر خواهد بود. یعنی ما از پدیده «ورودی نامناسب، خروجی نامناسب[1]» دور می‌شویم. هر چند به هر حال می‌توان یک روش داده‌کاوی را بر روی داده‌ها اعمال کرد و سپس بر اساس عملکرد پیش‌بینی تخمینی[2] آن نتایج را ارزیابی نمود، ولیکن این کار به هیچ وجه موجب کاهش اهمیت وظیفه اولیه ما یعنی توجه دقیق به *آماده‌سازی داده‌ها* نمی‌شود. با اینکه روشهای پیش‌بینی ممکن است توانایی‌های

[1]- Garbage in Garbage Out
[2]- Estimated Predictive Performance

نظری قوی داشته باشند ولی توان همه آنها در عمل با توجه به وضعیت داده‌ها در مقایسه با فضای نامحدود جستجو، محدود می‌شود.

انواع داده‌های مورد استفاده در داده‌کاوی

نمایش داده خام

نمونه‌های داده[1] که در جدول (۲-۱) با سطرها مشخص شده‌اند، اجزاء اصلی فرآیند داده‌کاوی هستند. هر نمونه با چندین مشخصه[2] که برای هر ویژگی مقادیر مختلفی وجود دارد، توصیف شده است [۱].

دو نوع معمول‌تر، داده‌های عددی[3] و طبقه‌ای هستند.

جدول ۲- ۱) نمونه ای از داده‌های خام

Feature value	*Code*
Black	۱۰۰۰
Blue	۰۱۰۰
Green	۰۰۱۰
Brown	۰۰۰۱

مقادیر عددی شامل متغیرهای حقیقی یا عدد صحیح مانند سن، سرعت و طول است.

متغیرهای عددی دو خاصیت زیر را دارند:

مقادیر آن یک رابطهٔ ترتیبی[4] دارند: $(۷ > ۵ , ۵ > ۲)$

رابطه فاصله‌ای دارند: $(D(۲.۳ , ٤.۲)=۱.٤)$

---

[1]- Data Samples
[2]- Feature
[3]- Numerical
[4]- Ordinal

بر عکس، متغیرهای طبقه‌ای هیچ‌کدام از دو رابطه بالا را ندارند. دو مقدار یک متغیر طبقه‌ای می‌توانند با هم برابر یا نابرابر باشند. بنابراین تنها رابطهٔ برابری برای آنها تعریف می‌شود. (آبی = آبی یا قرمز≠ سیاه) مثالهایی از این نوع، رنگ چشم (اگر تدریجی بودن طیف رنگ را در نظر نگیریم) و جنسیت است. یک متغیر طبقه‌ای با دو مقدار می‌تواند به یک متغیر عددی دودویی[1] با دو مقدار ۰ یا ۱ تبدیل شود. یک متغیر طبقه‌ای با n مقدار می‌تواند به n متغیر دودویی یعنی یک متغیر دودویی برای هر مقدار طبقه‌ای تبدیل شود.

برای مثال اگر متغیر رنگ چشم، چهار مقدار سیاه، آبی، سبز و قهوه‌ای داشته باشد آنها را می‌توان با چهار رقم دودویی به‌صورت عددی نوشت. روش دیگر دسته‌بندی متغیرها بر اساس مقادیر آنهاست که می‌تواند مقدار پیوسته[2] یا گسسته[3] باشد. متغیرهای پیوسته را به‌عنوان متغیرهای کمی یا متریک[4] نیز می‌شناسیم. این متغیرها با مقیاسهای بازه‌ای (فاصله‌ای)[5] یا نسبی[6] اندازه‌گیری می‌شوند.

هر دوی این مقیاسها، اندازه‌گیری با دقت نامحدود را ممکن می‌سازند. تفاوت این دو مقیاس در چگونگی قرارگیری نقطه صفر در مقیاس است. نقطه صفر در مقیاس بازه‌ای به‌طور قراردادی و اختیاری تعریف شده است و بنابراین نبودن متغیری را که اندازه‌گیری می‌کند بیان نمی‌کند. بهترین مثال برای مقیاس بازه‌ای مقیاس «حرارت» است. قرارگرفتن در نقطه صفر فارنهایت بیانگر این نیست که ابداً حرارت وجود ندارد. در مقیاس بازه‌ای، به خاطر جایگاه قراردادی، نقطه صفر واقعیت درستی از متغیری که اندازه‌گیری شده را نشان نمی‌دهد. برای مثال ۸۰ درجه فارنهایت به معنای دو برابر ٤٠ درجه نیست.

---

[1]- Binary
[2]- Continuous Variable
[3]- Discrete
[4]- Metric
[5]- Interval Scale
[6]- Ratio Scale

بر عکس یک مقیاس نسبی یک نقطه صفر مطلق دارد و در نتیجه ارتباط نسبی، واقعیت درستی از متغیر مورد اندازه‌گیری با این مقیاس را نشان می‌دهد. کمیتهایی همچون ارتفاع، طول و حقوق این نوع مقیاسها را استفاده می‌کنند. متغیرهای پیوسته در مجموعه‌های داده بزرگ با متغیرهای عددی از نوع صحیح و اعشاری ارائه شده‌اند. متغیرهای گسسته، متغیرهای کیفی[1] نیز نامیده می‌شوند. چنین متغیرهایی با استفاده از یک یا دو نوع مقیاس غیرمتریک ترتیبی و یا اسمی[2] اندازه‌گیری و تعریف می‌شوند.

یک مقیاس اسمی، یک مقیاس بی‌ترتیب است که سمبلها، کاراکترها و اعداد مختلف را برای نمایش حالات و مقادیر مختلف متغیر اندازه‌گیری شده، به‌کار می‌برد. یک مثال از متغیرهای اسمی، یک شناسه کاربری نوع مشتری با مقادیر ممکن تجاری، مسکونی و صنعتی است که مقادیر آنها می‌تواند به‌صورت الفبایی با A,B,C و یا به‌صورت عددی با ۱و۲و۳ نمایش داده شود. البته در بسیاری از داده‌ها چنین متغیرهایی وجود ندارند. در هر دو مثال، اعداد استفاده شده برای معین کردن مقادیر ویژگیهای متفاوت ترتیب مشخص ندارند و الزاماً ارتباطی با یکدیگر ندارند.

یک مقیاس ترتیبی شامل درجه‌بندی گسسته مرتب شده است. به‌عنوان مثال رتبه‌بندیها از چنین مقیاسی استفاده می‌کنند. یک متغیر ترتیبی یک متغیر طبقه‌ای است که برای رابطه ترتیبی و نه فاصله‌ای مورد استفاده قرار می‌گیرد. برخی از مثالهای متغیرهای ترتیبی، رتبه یک دانش‌آموز در یک کلاس و مدال طلا، نقره و برنز و موقعیت آن در مسابقات ورزشی است. مقیاس مرتب‌شده لزوما نباید خطی باشد. به‌عنوان مثال تفاوت بین دانش‌آموزان رتبه چهارم و پنجم لزوماً با تفاوت بین دانش‌آموزان رتبه‌های ۱۵ و ١٦ برابر نیست.

---

[1]- Qualitative Variables
[2]- Nominal

همه اینها می‌توانند از مقیاسهای ترتیبی برای صفات ترتیبی «بزرگتر از، کوچکتر از و یا مساوی با» به‌دست آیند. به‌عنوان نمونه، متغیرهای ترتیبی یک متغیر عددی را به درون مجموعه کوچکی از فاصله‌های منطبق با مقادیر متغیرهای ترتیبی نگاشت می‌کنند. این متغیرهای ترتیبی ارتباط بسیار نزدیکی با متغیرهای فازی مانند سن (با مقادیر جوان، میانسال یا مسن) و درآمد (با مقادیر کم، متوسط و ثروتمند) دارند.

یک نوع خاص از متغیرهای گسسته، متغیرهای تناوبی[1] است. یک متغیر تناوبی (برای مثال روزهای هفته، روزهای ماه یا سال) مشخصات متغیر بازه‌ای را دارد، اما ترتیبی نیست. چرا که دوشنبه و سه‌شنبه به‌عنوان مقادیر یک ویژگی نزدیک‌تر از دوشنبه و پنج‌شنبه هستند (فاصله‌ای است). البته دوشنبه می‌تواند قبل یا بعد از جمعه باشد. (ترتیبی نیست)

بالاخره نوع دیگری از تقسیم‌بندی داده‌ها بر اساس رفتار آنها با توجه به زمان است. بعضی از داده‌ها با زمان تغییر نمی‌کنند که ما به آنها داده‌های ایستا[2] می‌گوییم. از دیگر سو ویژگیهایی وجود دارند که با زمان تغییر می‌کنند که به این‌گونه داده‌ها، داده‌های پویا یا زمانی[3] می‌گوییم.

اکثر روشهای داده‌کاوی برای داده‌های ایستا مناسبند و برای داده‌های پویا، پیش پردازشهای ویژه مورد نیاز است.

آماده سازی داده‌ها

آماده‌سازی داده‌ها برای داده‌کاوی هنر چلاندن[4] و فشردن داده‌های موجود و بیرون کشیدن داده‌های با ارزش است. در حالیکه داده‌کاوی هنر کشف الگوهای معنی‌دار در

---

[1]- Periodic Variables
[2]- Static Data
[3]- Dynamic or Temporal Data
[4]- Wringing

داده‌ها است. معناداری الگو بستگی به مسئله دارد. آماده‌سازی نیز به‌عنوان جزئی از داده‌کاوی بستگی به نوع مسئله و نیز روشها و ابزارهایی دارد که می‌خواهیم بر روی داده به‌کار ببندیم. مثلاً شبکه‌های عصبی نیازمند ارائه داده‌هایی است که حداقل عددی یا ترتیبی باشند و به مقادیر مفقوده بسیار حساس است. درخت‌های تصمیم‌گیری اغلب بر روی داده‌های طبقه‌ای کار می‌کنند.

جایگاه آماده سازی داده‌ها در داده‌کاوی

در پروژه‌های داده‌کاوی، آماده‌سازی داده پس از مرحله فهم کسب و کار[1] و فهم داده[2] قرارگرفته است.[۳] در شکل(۲-۱) این ترتیب مشخص شده است.

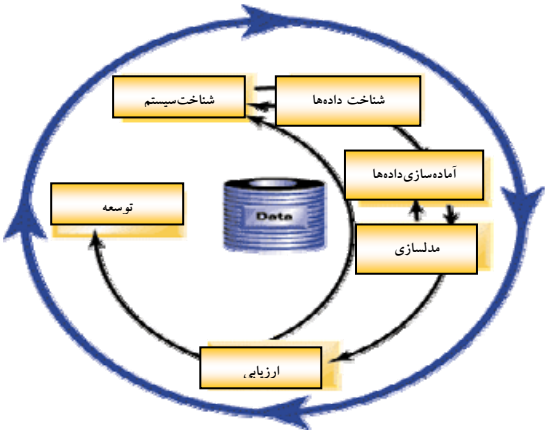
می‌دانیم که در مرحله فهم داده در جستجوی پاسخی برای پرسشهای زیر هستیم:

چه داده‌هایی برای این کار وجود دارد؟ آیا داده‌ها مربوطند؟ آیا داده‌های اضافه وجود دارد؟ چه مقدار داده‌های تاریخی وجود دارند؟ چه کسی خبره این داده‌ها است؟ اما می‌توان گفت که در مرحله آماده‌سازی داده‌ها به دنبال موارد زیر هستیم[۳].

- سازماندهی داده‌ها به شکلی استاندارد که آماده پردازش توسط برنامه‌های داده‌کاوی باشند. این شکل استاندارد صفحه گسترده‌ای (یعنی یک جدول داده) با انواع متغیرهای ترتیبی، عددی و دودویی است.
- تهیه مشخصه‌هایی که منجر به بهترین کارایی مدل پیش‌بینی شود.

---

[1]- Business Understanding
[2]- Data Understanding



شکل ۲– ۱) جایگاه آماده‌سازی داده‌ها در گام‌های انجام پروژه داده‌کاوی

چرا آماده‌سازی داده‌ها؟

آماده‌سازی داده‌ها، حدود ٦٠ تا ۹۰ درصد زمان مورد نیاز برای کاوش داده را صرف کرده و ۷۵ تا ۹۰ درصد موفقیت پروژه‌های داده‌کاوی به آن مربوط می‌شود [٤]. عدم آماده‌سازی داده یا آماده‌سازی ضعیف آن سبب شکست کامل پروژه می‌شود. نتیجه داده‌های بی‌کیفیت، داده‌کاوی بی‌کیفیت و در نتیجه تصمیمات بی‌کیفیت است[۲]. ممکن است داده مفقوده یا تکراری باعث آماره‌های نادرست یا حتی گمراه کننده شود. پیش‌پردازش داده‌ها جهت بهبود کیفیت داده‌های واقعی برای داده‌کاوی لازم است. داده‌ای با کیفیت خوانده می‌شوند که صحیح، کامل، سازگار، به روز، قابل قبول، با ارزش، قابل تفسیر و در دسترس باشد.

اما اغلب مجموعه‌های داده خام که برای داده‌کاوی آماده‌سازی اولیه می‌شوند، بزرگ بوده و بسیاری از آنان به علایق و تعلقات افراد بستگی داشته و پتانسیل آشفتگی و آلودگی[1] را دارند. می‌توان گفت داده‌ها در عالم واقع دارای آلودگیهای زیر هستند:

[1]- Dirty

ناقص[1]: مانند نمونه‌های ناکافی، کمبود برخی مقادیر مشخصه‌ها، داشتن نتایج به‌صورت تجمیع شده.

مغشوش[2]: داده‌ها دارای خطا یا مقادیر پرت هستند. مثلاً حقوق = "۱۰-"

ناسازگار[3]: دارای تناقض در کدها یا نامها هستند. مانند:

- سن = "۳۰" و تاریخ تولد ="۱۳۵۲/۰۵/۲۸"
- رتبه قبلی "۱و۲و۳" رتبه فعلی"*C,B,A*"
- تضاد در رکوردهایی که دوبار ثبت شده‌اند.

اما نکته‌ای که نباید از آن غفلت کرد این است که این داده‌ها چگونه تولید می‌شوند و یا از کجا می‌آیند؟ در این بخش به مبدا این آلودگیها می‌پردازیم.

داده‌های ناقص می‌تواند در نتیجه موارد زیر باشند:

- مقدار داده هنگام جمع‌آوری قابل قبول نبوده است.
- بین زمان جمع‌آوری داده و تحلیل آن تفاوت قابل ملاحظه‌ای وجود داشته است.
- مشکلات انسانی / نرم افزاری / سخت افزاری وجود داشته است.

داده‌های مغشوش می‌تواند ناشی از ایراد ابزارهای جمع‌آوری داده یا خطای انسان یا کامپیوتر هنگام ورود داده و یا خطا در انتقال داده‌ها باشد. اما داده‌های ناسازگار اغلب نتیجه منابع مختلف داده و یا مسائل و اختلافات بخشهای وظیفه‌ای است.

پردازش اولیه‌ای مورد نیاز است تا مقادیر مفقوده، انحرافات، مقادیر ثبت نشده، نمونه‌های ناکافی و مسائلی از این دست را در داده‌های اولیه بیابد. داده‌های خامی که هیچ‌یک از این مشکلات را ندارند باید سوء ظن شما را برانگیزند.

---

[1]- Incomplete
[2]- Noisy
[3]- Inconsistent

تنها دلیل درستی که می‌تواند باعث کیفیت بالای داده‌های ارائه شده باشد این است که داده‌ها پیش از رسیدن به تحلیل‌گر، پیش پردازش شده و به‌صورت انبارهٔ دادهٔ طراحی شده در آمده باشند.

این کار همچنین کارایی کاوش را از طریق کاهش زمان لازم برای کاوش داده‌های پیش‌پردازش شده افزایش می‌دهد. پیش‌پردازش داده شامل پاکسازی داده، تبدیل داده، یکپارچه‌سازی و کاهش داده یا فشرده سازی داده است.

تلخیص توصیفی داده‌ها [1]

اگر پیش پردازش داده‌ها بخواهد موفق باشد باید تصویری جامع از داده‌های شما داشته باشد. فنون تلخیص توصیفی داده‌ها می‌توانند برای شناسایی مشخصه‌های داده شما و برجسته ساختن داده‌هایی که باید به‌عنوان داده مغشوش یا داده‌های پرت با آنها رفتار شود، مورد استفاده قرار گیرد[٤]. بنابراین در ابتدا مفاهیم اصلی تلخیص توصیفی داده‌ها را پیش از ورود به بحث روشهای پیش پردازش داده معرفی می‌کنیم.

برای بسیاری از کارهایی که هنگام پیش پردازش داده‌ها انجام می‌دهیم لازم است تا ویژگیهای داده‌ها را با توجه به گرایش مرکزی[2] و پراکندگی[3] آنها بشناسیم.

معیار سنجش گرایش مرکزی شامل اندازه‌گیری میانگین[4]، میانه[5]، مد[6] و میان دامنه[7] است در حالی‌که سنجه‌های پراکندگی داده شامل چارکها[8]، دامنه میان چارکی[9] و واریانس است. این آماره‌های توصیفی کمک شایانی به فهم توزیع داده‌ها می‌کنند. این سنجه‌ها در علم آمار بررسی و مطالعه می‌شوند و ما از نقطه نظر داده‌کاوی باید بدانیم

1- Descriptive Data Summarization
2- Central Tendency
3- Dispersion
4- Mean
5- Median
6- Mode
7- Midrange
8- Quartiles
9- Interquartile range (IQR)

که این سنجه‌ها در پایگاه داده‌های بزرگ چگونه به‌صورتی کارا محاسبه می‌شوند؟ برای مطالعه بیشتر این سنجه‌ها به [٤] مراجعه نمایید.

نمایش گرافیکی داده‌های توصیفی

جدای ازگراف خطی[1]، نمودار ستونی[2] و نمودار کلوچه‌ای[3] که در بیشتر نرم افزارهای آماری برای نمایش گرافیکی داده‌ها استفاده می‌شود، چندین نوع گراف دیگر برای نمایش خلاصه داده‌ها و توزیعها وجود دارد که شامل هیستوگرامها، نمودار چارک، نمودارهای *Q-P*، نمودار پراکنش[4]، و منحنی لوئِس است. این گرافها برای بازرسی بصری داده‌ها بسیار مفید هستند. ما در مورد هر یک توضیح مختصری می‌آوریم[٤].

هیستوگرام

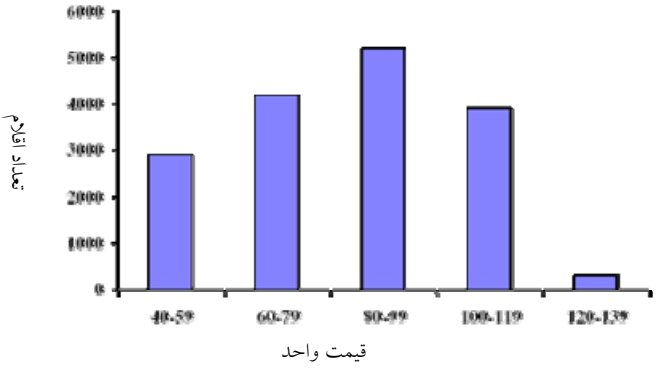
هیستوگرام یک طرح گرافیکی برای خلاصه کردن توزیع یک ویژگی معین است. یک هیستوگرام برای یک ویژگی در واقع نوعی بخش‌بندی توزیع داده درون زیرمجموعه‌های گسسته یا سطلها[5]ی مجزا است. معمولاً عرض تمام این سطلها برابر است. هر سطل با یک مستطیل نمایش داده می‌شود که طول آن برابر تعداد فراوانی داده‌هایی است که در دامنه آن قرار داشته‌اند. اگر ویژگی از نوع طبقه‌ای باشد آن‌گاه می‌توان این هیستوگرام را نوعی نمودار ستونی تعریف کرد. مثلاً برای داده‌های جدول (۲-۲) هیستوگرام شکل (۲-۲) رسم می‌شود که در محور افقی قیمت واحد و در محور عمودی فراوانی قرار می‌گیرد.

---

[1]- Line Graph
[2]- Bar Chart
[3]- Pie Chart
[4]- Scatter Plot
[5]- Bucket

جدول ۲-۲) داده‌های فروش

تعداد اقلام فروخته شده	قیمت واحد
۲۷۵	٤٠
۳۰۰	٤۳
۲۵۰	٤۷
--	--
۳٦۰	۷٤
۵۱۵	۷۵
۵٤۰	۷۸
--	--
۳۲۰	۱۱۵
۲۷۰	۱۱۷
۳۵۰	۱۲۰



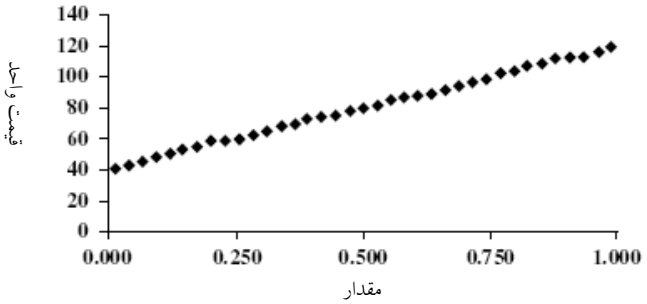
شکل ۲-۲) نمونه ای از یک هیستوگرام

نمودار چندک[1]

این نقشه، راه ساده و کارایی برای نگاهی اجمالی به یک توزیع تک متغیره است. در ابتدا این پلات، تمام داده‌ها را برای یک متغیر معین نمایش می‌دهد و بعد اطلاعات چندک[2] را ارائه می‌دهد.

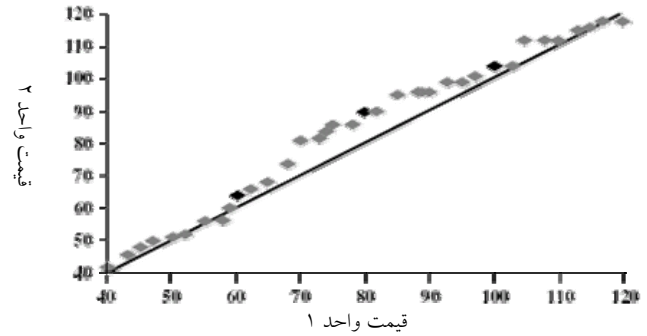
---

[1]- Quantile Plot (Q-P)
[2]- Quantile



شکل ۲-۳) نمونه ای از یک نمودار چندک

به فرض اگر متغیر x_i را با $i=1$ تا $i=n$ انتخاب کنیم، مقدار f_i که بر روی منحنی با آن مطابقت می‌کند، بیانگر این است که تقریباً f درصد داده‌ها از x_i کوچک‌تر یا با آن مساویند. نمونه این نمودار برای داده‌های جدول (۲-۲) را در شکل (۲-۳) می‌بینیم.



شکل ۲-٤) نمودار Q-Q

نمودار چندک – چندک[1]

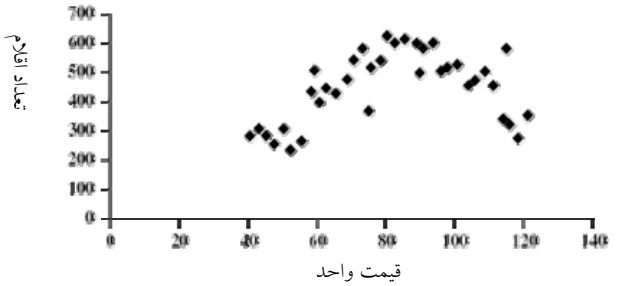
این نمودار چندک یک توزیع یک متغیره را در برابر چندک متناظر از یک توزیع دیگر رسم کرده و ابزار قدرتمندی برای مشاهده تغییر در یک متغیر به ازای حرکت در متغیر دیگر است.

[1]- Quantile –Quantile (Q-Q)

فرض کنید که ما دو دسته داده از متغیر قیمت واحد از دو شعبه متفاوت داریم. که x_1 تا x_N مربوط به شعبه اول و y_1 تا y_M مربوط به شعبه دوم باشد. شکل (۲-٤) نمودار Q-Q را برای این داده‌ها نشان می‌دهد.

نمودار پراکنش

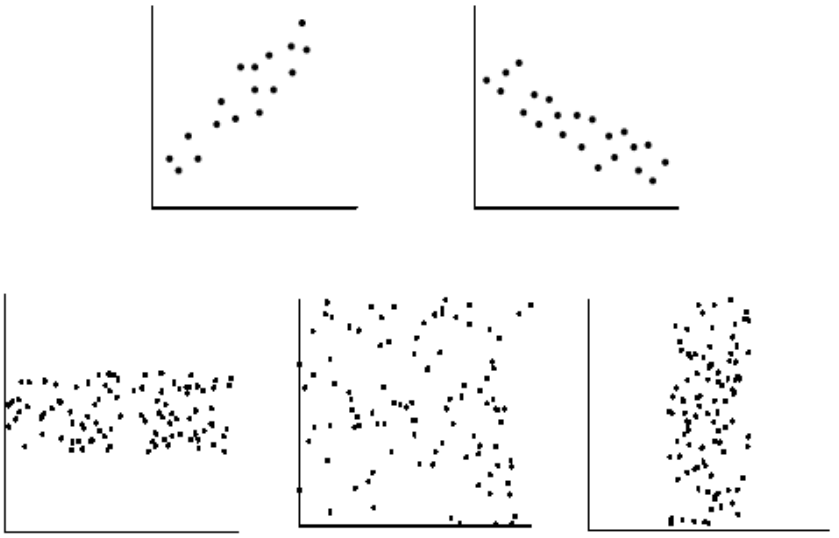
یکی از کارآمدترین روشهای گرافیکی برای تعیین وجود رابطه، الگو یا گرایش بین دو ویژگی عددی نمودار پراکنش است.



شکل ۲-۵) نمودار پراکنش

برای ساختن یک نمودار پراکنش مقادیری (زوج داده‌هایی) که برای دو ویژگی داریم در یک نمودار رسم می‌کنیم. نمونه یک نمودار پراکنش برای دو ویژگی قیمت واحد و اقلام فروخته شده را در شکل (۲-۵) می‌بینیم. نمودار پراکنش روش سودمندی برای ایجاد یک نگاه اجمالی به داده‌های دومتغیره و بخش‌بندی آن و یا تعیین مقادیر پرت و نیز برای بررسی احتمال وجود همبستگی میان دو ویژگی است.

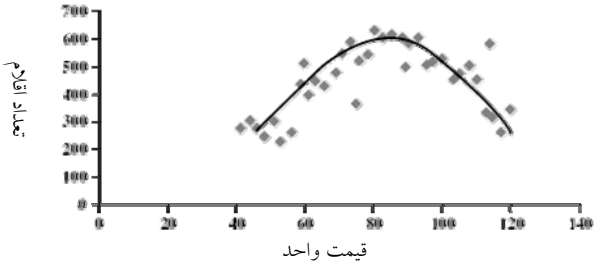
این همبستگی در صورت وجود می‌تواند به شکل مثبت یا منفی باشد. در شکل (۲-٦) نمودار پراکنش را برای دو ویژگی مشاهده می‌کنید. دو نمودار بالا به‌ترتیب بیانگر وابستگی منفی و مثبت و سه نمودار پایین بیانگر عدم وابستگی میان ویژگیها است.



شکل ۲- ٦) نمودارهای همبستگی میان دو ویژگی

نمودار لوئس[1]

این نمودار در واقع یک منحنی هموار از نمودار پراکنش می‌گذراند تا درک بهتری از الگوی وابستگی آنها ارائه دهد. شکل (۲-۷) این نمودار را برای نمودار پراکنش مثال قبل نمایش می‌دهد.



شکل۲-۷) نمودار لوئس

[1]- Loess Curve

اجزاء اصلی پیش پردازش داده‌ها

از دیدگاه آمار در بررسی مسائل مرتبط با پیش‌پردازش داده‌ها می‌توان گفت مشکلات به دو دسته تقسیم می‌شوند:

- مسائل مربوط به نمونه مانند نمونه‌های مفقوده و داده‌های پرت
- مسائل مربوط به توزیع مانند نرمالیتی و خطی بودن

در اینجا ما دسته نخست مسائل را بررسی می‌کنیم و توضیح مختصری درباره هر یک می‌آوریم و در بخش بعدی مفصل راجع به آنها صحبت خواهیم کرد.

پاکسازی داده

اغلب به جهت خطاهای عملیاتی و پیاده‌سازی سیستمها، داده‌های برآمده از منابع دنیای واقعی پر غلط، ناقص و ناسازگار هستند. لازم است در ابتدا چنین داده‌های کم کیفیتی تمیز شوند. این کار شامل برخی عملیات پایه مانند نرمال‌سازی، حذف نویز یا اغتشاش، مواجهه با داده‌های مفقوده، کاهش افزونگی، برطرف کردن ناسازگاری و از این گونه کارها است.

یکپارچه‌سازی داده‌ها

یکپارچه‌سازی داده نقش مهمی‌در *KDD* بازی می‌کند. این عملیات شامل یکپارچه‌سازی چندین پایگاه داده ناهمگن که قبلا به‌وسیله چندین منبع ایجادشده، است.

تبدیل داده

این کار شامل عملیاتی همچون نرمال‌سازی و تجمیع است.

کاهش داده و تصویر کردن داده[1]

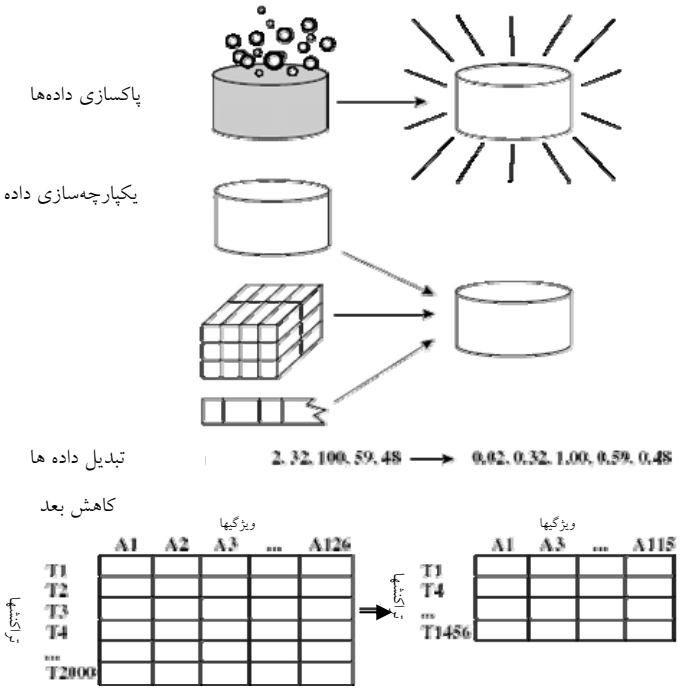
این کار شامل یافتن ویژگیهای مفید برای بازنمایی داده (بسته به هدف کار) و استفاده از روشهای کاهش بُعد، گسسته‌سازی و استخراج (تبدیل) ویژگیها است. به‌کارگیری

[1]- Data Projection

اصول فشرده‌سازی داده می‌تواند نقش مهمی‌در کاهش داده بازی کند. البته داده کاهش یافته باید ما را به نتایج تحلیلی مشابه داده‌های اصلی برساند.

فشرده‌سازی داده‌ها

روشی است برای کاهش افزونگی در بازنمایی داده‌ها به منظور کاهش حافظه مورد نیاز و در نتیجه کاهش هزینه‌های ارتباطی و انتقال در یک شبکه ارتباطی. برخی منابع فعالیت‌هایی همچون مستندسازی داده‌ها، طرح‌ریزی کدینگ و ورود داده را نیز از جمله کارهای مرتبط با آماده‌سازی داده دانسته‌اند. تمامی آنچه در این مرحله انجام می‌شود در شکل (۲–۸) خلاصه شده است. به هر حال در نتیجه آن‌چه در این مرحله انجام می‌شود یک فایل آماده برای کار[1] ایجاد می‌گردد.



شکل ۲– ۸) عملیات مختلف در پاکسازی داده

[1]- Work File

پاکسازی داده‌ها

رالف کیمبال[1] پاکسازی داده را یکی از سه مسئله بزرگ در انباره سازی داده دانسته است. گروه دی سی آی[2]، پاکسازی داده را به‌عنوان مسئله اول در انباره سازی مطرح کرده است.

پاکسازی داده در واقع مرحله کنترل کیفی قبل از تحلیل داده است. به‌طور کلی می‌توان گفت در این مرحله بررسی‌های زیر انجام می‌شود[٤]:

- اطمینان از وجود تعداد مناسبی نمونه در فایل و اینکه شناسه هیچ‌کدام تکرار نشده باشد.
- بررسی کدهای آشفته
- کنترلها و بررسی‌های سازگاری
- یک بررسی تکمیلی برای اینکه تمام نمونه‌های جمع‌آوری شده و در فایل آمده‌اند.

وظایف پاکسازی داده‌ها

وظایف اصلی فاز پاکسازی داده‌ها عبارتند از:

- پرکردن داده‌های مفقوده
- شناخت داده‌های پرت و هموار کردن داده‌های مغشوش
- درست کردن داده‌های ناسازگار
- حل کردن مشکل افزونگی که بر اثر یکپارچه ساختن داده‌ها ایجاد شده است.

برخی، به‌دست آوردن داده و ایجاد فراداده را نیز از جمله وظایف این مرحله دانسته‌اند. این داده‌ها می‌تواند در یک *DBMS* یا در یک فایل دارای یک جدول[3] قرارگرفته باشد. البته در چنین حالتی مقدار داده‌های یک ویژگی یا از روی تعداد ستون داده‌ها و یا

---

[1]- Ralph Kimball
[2]- DCI Group
[3]- Flat

توسط کاراکترهای جدا کننده از داده‌های ویژگی دیگر متمایز می‌شود. آنچه ما به‌عنوان فراداده گردآوری می‌کنیم در واقع اطلاعاتی راجع به ماهیت داده‌هایی است که می‌خواهیم بر روی آنها داده‌کاوی انجام دهیم. به‌عنوان مثال نوع فیلد (دودویی، طبقه‌ای، ترتیبی، عددی) برای فیلدهای اسمی جداول ترجمه کدها و هم‌چنین نقش فیلد (ورودی، هدف و شناسه کمکی) بخشی از اطلاعاتی است که از فراداده قابل دستیابی است.

مقادیر مفقوده[1]

در داده‌های اولیه که برای داده‌کاوی در اختیار داریم ممکن است برخی نمونه‌ها برای برخی ویژگیها مقدار نداشته باشند. مثلاً در داده‌های فروش ممکن است برای چند مشتری مقدار درآمد مشتری درج نشده باشد، ما به این مقادیر، مقادیر مفقوده می‌گوییم. داده مفقوده ممکن است به دلایل زیر ایجاد شده باشد:

- تجهیزات ایراد داشته است.
- با داده دیگر ناسازگار بوده و به ناچار حذف شده است.
- به خاطر دشواری فهم داده وارد نشده است.
- ممکن است هنگام ورود داده‌ها حایز اهمیت نبوده است.
- تاریخچه یا تغییرات داده ثبت نشده است.

ما برای شروع کار داده‌کاوی نیاز داریم که این مقادیر را حذف و یا جای خالی آنها را پرکنیم. در مواجهه با چنین داده‌هایی می‌توانیم راهکارهای گوناگونی در پیش گیریم.

- رکورد را حذف کنیم[2]: معمولاً وقتی رکورد حذف می‌شود که برچسب دسته گم شده باشد(به فرض که کار داده‌کاوی نوعی دسته‌بندی باشد). یکی از ایرادات این

[1]- Missing Values
[2]- List Wise

شیوه کاهش اندازه نمونه است. این روش، کارا نیست مگر اینکه تعداد ویژگیهای فاقد مقدار در یک نمونه زیاد نباشد.

- مشاهده را حذف کنیم، البته این روش تنها وقتی استفاده می‌شود که مقادیر مفقوده در محاسبات به‌کار می‌رود. ایراد این شیوه این است که هر میانگین، واریانس و کواریانسی که می‌گیریم متعلق به اندازه‌ها متفاوت نمونه است.
- مقادیر مفقوده را به‌صورت دستی پر کنیم. ایراد این شیوه این است که خسته کننده و در دنیای واقعی و با ابعاد داده‌های واقعی نشدنی است.
- به‌صورت خودکار با مقادیر زیر پر کنیم:
- یک مقدار ثابت سراسری (مثل "*Unknown*"). ممکن است برخی برنامه‌های داده‌کاوی این مقدار را با مقدار ویژگی اشتباه بگیرند.
- میانگین ویژگی: مثلاً میانگین حقوق را برای حقوق کسانی که دارای مقدار نیستند، وارد کنیم.
- میانگین ویژگی برای کلاسهای مشابه: برای ویژگی در نمونه‌هایی که دارای برچسب کلاس مشابه می‌باشند، مقدار یکسان قرار داده می‌شود.
- مقادیر با احتمال بیشتر: با استفاده از رابطه‌های بیزی، درخت تصمیم‌گیری و یا رگرسیون می‌توان مقدار مفقوده را پیش‌بینی کرده و پر کرد.

روشهایی که از پر کردن خودکار استفاده می‌کنند، دارای سوگیری[1] هستند. برای مثال اگر داده‌های مفقوده یک مشخصه با میانگین مشخصه همان دسته جایگزین شوند، ممکن است یک برچسب معادل به‌طور ضمنی جانشین برچسب یک دسته متفاوت ولی مخفی شود. واضح است که استفاده از این برچسب، درست نیست. از طرفی جایگزینی مقادیر مفقوده با یک مقدار ثابت، رکوردهای مربوط به آنها را به شکل یک زیر مجموعه همگن درمی‌آورد که متمایل به برچسب دسته بزرگترین گروه رکوردهای

[1]- Biased

دارای مقادیر مفقوده است. اگر مقادیر مفقوده همه مشخصه‌ها با یک ثابت عمومی جایگزین شوند، ممکن است بدون اینکه قصد داشته باشیم مقدار نامعلومی به‌طور ضمنی در عامل دیگری اثرگذار شود. برای مثال در پزشکی ممکن است به دلیل اینکه تشخیص یک بیماری قبلاً تأیید شده، از انجام یک آزمایش پرهزینه بپرهیزیم. البته این رکورد باعث نمی‌شود تا ما همواره در غیاب نتایج مربوط به این آزمایش پرهزینه، همان بیماری قبلی را نتیجه بگیریم.

به‌طورکلی جایگزینی مقادیر مفقوده به کمک یک طرح ساده آماده‌سازی داده‌ها، خطرناک و اغلب گمراه‌کننده است. بهترین کار این است که با و بدون مشخصه‌های دارای مقادیر مفقوده جواب‌های متعدد ایجاد کرده یا اینکه متکی بر روشهای پیش‌بینی مثل برخی روشهای منطقی بود، که دارای طرحهای جانشینی باشند. پر کردن مقدار با استفاده از روشهای پیش‌بینی بیشتر رایج است. چرا که در مقایسه با دیگر روشها، از داده‌های موجود برای پرکردن داده مفقوده بیشترین بهره را می‌برد.

باید توجه داشت که فقدان مقدار برای یک ویژگی همیشه دلیل وجود خطا نیست. مثلاً وقتی از کاربران یک سیستم کلمه عبور خواسته می‌شود، کاربری که این کلمه را نداند مقداری وارد نمی‌کند. البته می‌توان به گونه‌ای برنامه‌ریزی کرد که در این موارد سیستم به‌صورت خودکار عبارت "*I don't know*" یا "*Null*" را در آن محل قرار دهد. قاعدتاً باید در سیستم قوانینی برای برخورد با این موارد پیش‌بینی کرد.

داده مغشوش

اغتشاش یا نویز، خطای تصادفی یا مغایرت در متغیر اندازه‌گیری شده است. مقادیر ویژگی ممکن است به دلایل زیر نادرست باشد:

- ابزارهای معیوب جمع‌آوری داده
- مسائل و مشکلات حین ورود داده
- محدودیت فناوری

این خطاها پس از انجام روشهای ترکیبی بازرسی انسان و کامپیوتری و یا تشخیص داده‌های مشکوک و بررسی آنها به‌وسیله انسان، مشخص می‌شوند. حال به فرض آنکه متغیر عددی مانند پرداخت یا حقوق دارای اغتشاش باشد، این مقدار را چگونه می‌توان هموار[1] کرد؟ ما استفاده از سه روش بسته‌بندی[2]، رگرسیون و خوشه‌بندی را برای این کار پیشنهاد می‌کنیم.

بسته‌بندی

در این روش مقدار داده بر اساس مقدار همسایگانش در همان حوالی، هموار می‌شود. برای این کار ابتدا داده‌ها را مرتب کرده و در تعدادی جعبه یا بسته قرار می‌دهیم. تا این جای کار در واقع روشی است که برای گسسته‌سازی مقادیر پیوسته هم می‌توان به‌کار بست. سپس می‌توان به وسیله میانگین میانه یا مرزهای هر بسته، داده‌ها آن‌را هموار کرد. از آنجا که این روش از همسایه‌های مقادیر استفاده می‌کند، بنابراین هموارسازی محلی است. اما این گسسته‌سازی هم به دو شیوه قابل انجام است:

- عرض ثابت[3]

دامنه را به N دوره با اندازه و عرض مساوی تقسیم کنید. اگر B,A به ترتیب کمترین و بیشترین مقدار ویژگی باشند، عرض هر دوره از رابطه زیر محاسبه می‌شود:

$$W=(B-A)/N \qquad (۱-۲)$$

سپس داده‌ها را در بسته‌ای که در دامنه یا عرض آن قرارگرفته اند، تقسیم کنید.

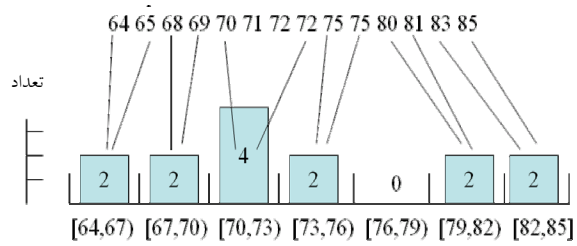
مثال: داده‌های زیر درجه حرارت محیط در یک دوره است، می‌خواهیم آنها را به شیوه عرض ثابت گسسته کنیم:

٦٤,٧٠,٨١,٦٨,٨٥,٧١,٨٣,٦٥,٧٢,٧٢,٧٥,٦٩,٧٥,٨٠

---

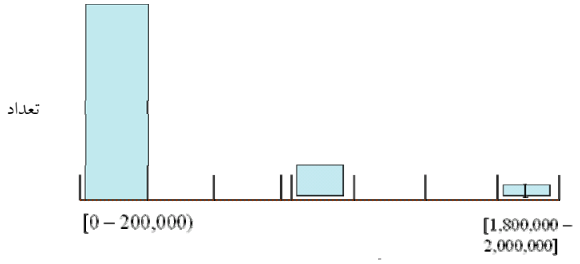
[1]- Smooth
[2]- Binning
[3]- Equal-width

شکل (۲-۹) نمایش‌گر شیوه گسسته‌سازی عرض ثابت این اعداد است.



شکل ۲- ۹) گسسته‌سازی به شیوه عرض ثابت

همان‌گونه که می‌بینیم کاملاً مشابه رسم نمودار فراوانی آنها است. این شیوه اگرچه بسیار ساده است اما داده‌های پرت آن را تحت تاثیر قرار می‌دهند و در مورد داده‌های دارای چولگی نیز مناسب نمی‌باشد. شکل (۲-۱۰) نمونه‌ای را نشان می‌دهد که گسسته‌سازی با روش عرض ثابت بر روی داده‌های حقوق یک شرکت باعث ایجاد دسته‌های جدا می‌شود.



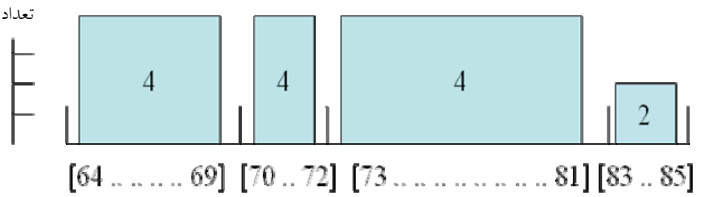
شکل ۲- ۱۰) ایجاد دسته‌های جدا در گسسته‌سازی عرض ثابت

عمق ثابت[1]

در این شیوه داده‌ها را به N دسته تقسیم می‌کنیم به‌گونه‌ای که در هر بسته تعداد تقریباً برابری از داده‌ها قرارگیرد. این روش مقیاس بندی بهتری دارد و به ویژه داده‌های

[1]- Equal-depth

طبقه‌ای را به‌خوبی تقسیم می‌کند. داده‌های مثال بالا در شکل (۲-۱۰) با استفاده از روش عمق ثابت گسسته شده است. همان‌گونه که می‌بینید تمام بسته‌ها دارای ٤ مقدار هستند. به‌جز بسته آخری که دارای ۲ عضو است و البته دلیل این امر نیز آن است که تعداد کل داده‌ها یعنی ١٤ مضربی از ٤ نیست. می‌بینید که در این روش چولگی ایجاد نمی‌شود.



شکل ۲- ۱۱) گسسته‌سازی عمق ثابت

اکنون که گسسته‌سازی انجام شده است، می‌توان مقادیر هر بسته را با مقدار میانگین بسته یا با مقادیر مرز یا لبه آن هموار کرد. در مثال زیر پس از گسسته ساختن مقادیر ویژگی قیمت با استفاده از روش عمق ثابت، آنها را با دو روش گفته شده هموار می‌کنند:

داده‌های ذخیره شده برای قیمت (برحسب دلار): ٤,٨,٩,١٥,٢١,٢١,٢٤,٢٥,٢٦,٢٨,٢٩,٣٤

تقسیم‌بندی آنها به سه بسته یا بسته با روش عمق ثابت:

بسته اول ٤,٨,٩,١٥

بسته دوم ٢١,٢١,٢٤,٢٥

بسته سوم ٢٦,٢٨,٢٩,٣٤

هموارسازی به‌وسیله میانگین بسته‌ها :

بسته اول: ٩,٩,٩,٩

بسته دوم: ٢٣,٢٣,٢٣,٢٣

بسته سوم: ٢٩,٢٩,٢٩,٢٩

هموارسازی به‌وسیله مرز بسته‌ها:

بسته اول: ٤,٤,٤,١٥

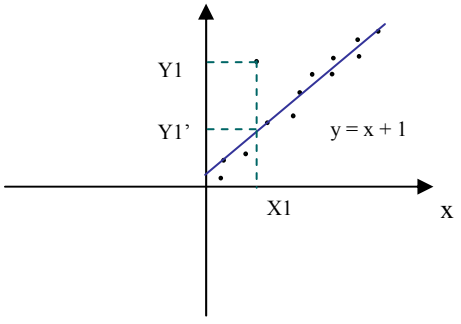
بسته : ٢١,٢١,٢٥,٢٥

بسته سوم: ٢٦,٢٦,٢٦,٣٤

در هموارسازی به‌وسیله میانگین، به جای تمام اعضای بسته، مقدار میانگین هر بسته قرار می‌گیرد و در هموارسازی به‌وسیله مرز بسته، ابتدا مقدار حداقل و حداکثر هر بسته به‌عنوان مرزهای آن تعریف شده و سپس به جای هر کدام از مقادیر درون بسته مقدار مرزی (مرز نزدیک‌تر به مقدار مورد بحث) به جای آن قرار می‌گیرد. برخی اوقات برای هموارسازی از میانه هم استفاده می‌شود. یعنی به جای هر کدام از مقادیر بسته، میانه مقادیر بسته قرار می‌گیرد.

رگرسیون

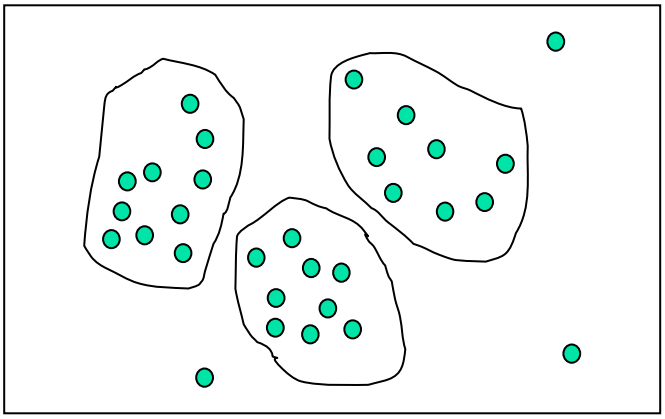
داده را می‌توان از راه تطبیق دادن داده با یک تابع مانند تابع رگرسیون که برای ویژگی به‌دست آمده، هموار کرد. رگرسیون خطی، بهترین خطی که بر دو ویژگی (یا متغیر) تطبیق کند را به‌گونه‌ای که مقدار یکی بتواند برای پیش‌بینی مقدار دیگری به‌کار رود، می‌یابد. رگرسیون چند متغیره خطی نیز توسعه یافته همین رگرسیون خطی است جایی که بیشتر از دو ویژگی در میان باشد و داده‌ها با یک سطح چند بُعدی تطبیق داده شوند.



شکل ٢- ١٢) استفاده از رگرسیون برای هموارسازی

خوشه‌بندی

وقتی داده‌ها در چند خوشه تقسیم‌بندی می‌شوند، داده‌هایی که در هیچ‌کدام از خوشه‌ها نیستند را می‌توان داده‌های پرت فرض کرد. استفاده از این روش را در شکل (۲-۱۳) می‌توان دید که داده‌ها به سه خوشه تقسیم شده‌اند و سه مقدار از داده‌های موجود در هیچ‌کدام از خوشه‌ها عضو نبوده‌اند. این سه مقدار به‌عنوان اغتشاش شناسایی می‌شوند. بسیاری از روشهای هموارسازی، در ضمن روشهای کاهش داده نیز محسوب می‌شوند از جمله روشهایی که در گسسته‌سازی استفاده می‌شوند.



شکل ۲-۱۳) استفاده از خوشه‌بندی برای هموارسازی

پاکسازی داده به‌عنوان یک فرآیند

- تا به‌حال درباره مواجهه با داده‌های مفقوده و مغشوش بحث کردیم، اما واقعیت این است که پاکسازی داده‌ها یک کار حجیم و بزرگ است و ما بایستی آن‌را به‌صورت یک فرآیند کامل دیده و اجزا و توالی آنها را بررسی کنیم. اولین گام در پاکسازی داده‌ها تشخیص مغایرت[1] است. این مغایرتها می‌تواند دلایل زیادی داشته باشد. از

[1]- Discrepancy Detection

جمله می‌توان به طراحی ضعیف فرم ورود داده به‌دلیل داشتن تعداد فراوان فیلدهای اختیاری یا خطای انسانی در ورود داده، خطاهای عمدی[1] (وقتی کسی نمی‌خواهد درباره خودش اطلاعات بدهد) و داده‌های تاریخ مصرف گذشته[2] (برای مثال آدرسهایی که عوض شده‌اند) اشاره کرد. مغایرتها می‌تواند ناشی از بازنمایی داده‌های ناسازگار و استفاده از کدهای ناسازگار باشد. یا به جهت تجمیع پایگاه داده‌ها از منابع گوناگون رخ داده باشد. اما برای شناخت داده‌های مغایر، از کجا باید آغاز کنیم؟

- نخست باید هر گونه دانشی که هم اکنون پیرامون خاصیت و ویژگیهای داده وجود دارد، مورد ملاحظه قرارگیرد. این دانش را «داده درباره داده» یا فراداده می‌گوییم. برای مثال دامنه داده و نوع هر کدام از ویژگیها یا اینکه برای هر کدام چه مقادیری قابل قبول است؟ طول مقدار چقدر می‌تواند باشد؟ بین ویژگیها چه وابستگی وجود دارد؟ تلخیص توصیفی داده (که پیش از این اشاره کردیم) در این مرحله برای فهم گرایش داده و شناخت مقادیر غیر متعارف در داده‌ها بسیار راه‌گشاست. مثلاً درک اینکه برای یک ویژگی مفروض چه داده‌هایی در فاصله بیش از دو انحراف استاندارد از میانگین هستند، به ما کمک می‌کند تا مقادیری که می‌توانند پرت باشند را شناسایی کنیم.
- به‌عنوان تحلیلگر داده‌ها باید مراقب استفاده ناسازگار از کدها و هر گونه بازنمایی ناسازگار داده‌ها باشید. (برای مثال نشان دادن تاریخ به‌صورت "۱۳۸۵/۱۰/۰۵" و همزمان "۰۵/۱۰/۱۳۸۵").
- سربار شدن فیلد[3] نیز خود مشکل دیگری است که می‌تواند ناشی از بیتهای بلااستفاده در تعریف فیلدها باشد. داده‌ها همچنین باید به‌گونه‌ای تعریف شوند که

---

[1]- Deliberate Errors
[2]- Data Decay
[3]- Field Overloading

قانون یکتایی[۱] را رعایت کنند. یعنی مقادیر ویژگی (برای کلید اصلی) تکرار نشود. همچنین رعایت دیگر قواعد پایگاه داده از آن جمله قانون تهی[۲] (تهی نبودن مقدار کلید اصلی) باید مد نظر باشد.

برخی ابزارهای تجاری وجود دارند که در این مرحله برای تشخیص مغایرتها به ما کمک می‌کنند. از آن جمله می‌توان به ابزارهای داده‌روبی[۳] به‌وسیله دانش ساده در مورد دامنه (به‌عنوان مثال کد پستی، چک کردن املای کلمه) و ابزارهای ممیزی داده[٤] برای تحلیل داده و کشف قوانین و روابط برای تشخیص انحرافات اشاره کرد. نمونه‌ای از ابزارهای ممیزی داده، استفاده از خوشه‌بندی و رگرسیون برای شناخت داده‌های پرت است. همچنین می‌توان از ابزارهای تلخیص توصیفی داده‌ها در این گام استفاده کرد.

برخی از ناسازگاریها ممکن است به‌صورت دستی تصحیح شوند مثلاً خطاهایی که هنگام ورود داده رخ داده است از طریق ردگیری گزارشات کاغذی شناسایی شده و برطرف شوند.

اما خطاهای بیشتر به تبدیلات داده[۵] نیاز دارند که دومین گام در پاکسازی داده است. زیرا پس از اینکه مغایرتها شناسایی شدند، یک‌سری اقدامات و تبدیلات تعریف و اجرا می‌شوند تا تصحیح اتفاق افتد. ابزارهای تجاری می‌توانند در گام تبدیل داده نیز کمک کنند. ابزارهای مهاجرت داده[٦] به ما اجازه تبدیلات ساده در یک موضوع مشخص را می‌دهند. مثلاً می‌توان مقدار یک رشته را در کل داده‌ها عوض کرد. ابزارهای ا ت ب (استخراج، تبدیل، بارگذاری)[۷] دسترسی کاربر به تبدیلات را با استفاده از رابط گرافیکی کاربر میسر می‌سازند. البته این ابزارها اغلب تعداد محدودی از تبدیلات را پشتیبانی

---

[1]- Unique Rule
[2]- Null Rule
[3]- Data Scrubbing
[4]- Data Auditing
[5]- Data Transformations
[6]- Data Migration Tools
[7]- ETL (Extraction/Transformation/Loading)

می‌کنند و ما کماکان ناچاریم برخی تبدیلات را با استفاده از برنامه‌نویسی انجام دهیم. مرحله بعدی در واقع این است که این دوگام با هم یکپارچه و هماهنگ شوند. رویکردهای نوین در پاکسازی داده‌ها به افزایش تعامل با کاربر تأکید می‌کنند.

یکپارچه‌سازی و تبدیلات

داده‌کاوی اغلب به یکپارچه‌سازی داده (ادغام داده‌ها از چندین منبع داده) نیاز دارد. همچنین ممکن است لازم باشد که داده‌ها به شکل مناسب داده‌کاوی تبدیل شوند[٤].

یکپارچه‌سازی داده

در این مرحله، داده‌های چندین منبع را در یک مخزن منسجم ترکیب می‌کنیم. مسئله‌ای که وجود دارد شناخت موجودیتهای مشابه درون چندین منبع است. مثلاً اگر در پایگاه داده A برای نام مشتری فیلد $A.Cust\_id$ و در پایگاه داده B از فیلد $B.Cust\#$ به‌همان منظور استفاده شده باشد، در صورت عدم حذف یکی از این دو، آنگاه مشکل افزونگی داده ایجاد می‌شود. البته این مشکل می‌تواند درون یک پایگاه داده هم رخ دهد و آن وقتی است که یک فیلد که از روی فیلد دیگری درون همان پایگاه داده قابل استنتاج بوده، در آن نگهداری شود. مثلاً نگهداری تاریخ تولد و سن به‌صورت همزمان ایجاد افزونگی می‌کند.

بنابراین برای رفع مشکل افزونگی داده‌ها بایستی فیلدهای تکراری شناسایی شوند. اما همان‌گونه که در مثال بالا مشخص است ممکن است این فیلدها در پایگاه داده‌های متفاوت، دارای نامهای مختلف باشند بنابراین استفاده از فراداده و اطلاعاتی که در هنگام طراحی پایگاههای داده مستند شده است، می‌تواند به ما کمک کند. علاوه بر این استفاده از روشهای آماری برای شناخت ویژگیهایی که دارای وابستگی هستند نیز به ما کمک می‌کند. در واقع برای این کار نیاز به استفاده از تحلیلهای همبستگی داریم. وقتی

همبستگی بین دو ویژگی عددی A و B را می‌آزماییم لازم است تا ضریب همبستگی را مطابق رابطه (۲-۲) به‌دست آوریم:

$$r_{A,B} = \frac{\sum_{i=1}^{N}(a_i - \overline{A})(b_i - \overline{B})}{N\sigma_A\sigma_B} = \frac{\sum_{i=1}^{N}(a_i b_i) - N\overline{A}\,\overline{B}}{N\sigma_A\sigma_B} \qquad (۲-۲)$$

در رابطه (۲-۲) N تعداد نمونه‌ها، a_i و b_i مقادیر دو ویژگی در نمونه‌ها و $\overline{A}$ و $\overline{B}$ ترتیب میانگین دو ویژگی و σ_a و σ_b به ترتیب انحراف استاندارد آنها هستند. مقدار $-۱ \le r_{A,B} \le +۱$ است. اگر r بزرگتر از صفر باشد همبستگی مثبت و اگر کمتر از صفر باشد همبستگی منفی است. البته وقتی این مقدار بیانگر همبستگی بالاست که نزدیک به ۱ یا ۱- باشد. در چنین حالتی (که قدر مطلق آن بزرگتر از ۰,٦ باشد.) لازم است بررسی موردی روی آن دو ویژگی انجام شود تا اگر تکراری هستند، یکی از آنها در هنگام یکپارچه‌سازی حذف شود.

هنگامی‌که ویژگیهای مورد نظر عددی نباشند از ضریب همبستگی نمی‌توان استفاده کرد. در این حالت از آزمون مربع کای ($X^۲$) استفاده می‌کنیم. پس از قرار دادن مقادیر در جدول تصادفی[1] مقدار آماره $X^۲$ را به‌دست می‌آوریم. در صورتی‌که این مقدار از مقدار بحرانی که برای درجه آزادی $(c-۱)\times(r-۱)$ که از جدول توزیع مربوطه به‌دست می‌آید بیشتر بود، فرض صفر آزمون یعنی استقلال دو ویژگی رد می‌شود و بنابراین دو ویژگی احتمالاً دارای همبستگی هستند.

در رابطه (۲-۳)، (o_{ij}) تعداد مشاهده شده و (e_{ij}) تعداد مورد انتظار را وقتی تعداد سطرها *(c)* تعداد مقادیر مجزای ویژگی A و تعداد ستونها *(r)* تعداد مقادیر مجزای B را نشان می‌دهد.

$$X^۲ = \sum_{i=1}^{c}\sum_{j=1}^{r}\frac{(o_{ij} - e_{ij})^۲}{e_{ij}} \qquad (۳-۲)$$

---

[1]- Contingency Table

مقدار e_{ij} تعداد مورد انتظار برای ویژگیهای B و A در خانه متناظر جدول است و o_{ij} تعداد مشاهده شده در همان خانه است که از مسئله داده شده به‌دست می‌آید. اما برای محاسبه e_{ij} از رابطه (۲-٤) داریم:

$$e_{ij} = \frac{count(A = a_i) \times count(B = b_j)}{N} \qquad (٤-٢)$$

که N تعداد کل نمونه‌ها است و در صورت کسر نیز تعداد مشاهده‌ها وجود دارد.

جدول ۲-۳) جدول تصادفی برای محاسبه آماره آزمون کای دو

	مونث	مذکر	جمع
خواندن	۲۰۰ (۳٦۰)	۲۵۰ (۹۰)	٤۵۰
نخواندن	۱۰۰۰ (۸٤۰)	۵۰ (۲۱۰)	۱۰۵۰
جمع	۱۲۰۰	۳۰۰	۱۵۰۰

مثال: یک گروه ۱۵۰۰ نفره از مردم مورد مطالعه قرارگرفته‌اند. که جنسیت هر کدام نیز ثبت شده است. هر یک به این سؤال پاسخ داده‌اند که آیا کتابهای داستانی می‌خوانند یا خیر؟ پاسخها در جدول زیر خلاصه شده است. سطرها بیانگر خواندن یا نخواندن داستان و ستونها بیانگر مرد یا زن بودن می‌باشند. داده‌های هر خانه (خارج از پرانتز) تعداد مشاهدات است. یعنی مثلاً ۲۵۰ مرد که داستان می‌خوانند در نمونه‌ها بوده‌اند. سپس از رابطه بالا تعداد مورد انتظار را برای هر خانه حساب می‌کنیم. مثلاً برای خانه اول این مقدار یا $e_{۱۱}$ به‌صورت زیر محاسبه می‌شود:

$$e_{۱۱} = \frac{count(male) \times count(fiction)}{N} = \frac{۳۰۰ \times ۴۵۰}{۱۵۰۰} = ۹۰, \qquad (۵-۲)$$

یعنی در صورت کسر، حاصل‌ضرب دو سطر و ستون آخر متناظر و در مخرج کسر تعداد کل داده‌ها. این مقدار را برای تمام خانه‌ها به‌دست آورده و در رابطه محاسبه آماره مربع کای می‌گذاریم:

$$X^۲ = \frac{(۲۵۰-۹۰)^۲}{۹۰} + \frac{(۵۰-۲۱۰)^۲}{۲۱۰} + \frac{(۲۰۰-۳٦۰)^۲}{۳٦۰} + \frac{(۱۰۰۰-۸٤۰)^۲}{۸٤۰}$$

$$= ۲۸۴.۴۴ + ۱۲۱.۹۰ + ۷۱.۱۱ + ۳۰.۴۸ = ۵۰۷.۹۳$$

مقدار آماره به‌دست آمده یعنی ۵۰۷,۹۳ را با مقدار آماره برای درجه آزادی(۱-۲)(۱-۲) یا درجه آزادی ۱ که برابر ۱۰,۸۲۸ است مقایسه می‌کنیم. مقدار آماره به‌دست آمده بزرگتر است، بنابراین شرط صفر آزمون یا مستقل بودن جنسیت از داستان خواندن رد می‌شود. بنابراین نتیجه می‌گیریم در داده‌های ما این دو ویژگی با یکدیگر همبستگی دارند.

تبدیل داده‌ها

در این مرحله داده‌ها به شکل مناسب برای داده‌کاوی تبدیل می‌شوند. تبدیل داده‌ها به اشکال زیر انجام می‌شود:

هموارسازی[1]: با حذف کردن مقادیر مغشوش داده سر و کار دارد. برخی روشهای مورد استفاده برای هموارسازی، بسته‌بندی، رگرسیون و خوشه‌بندی است. هموارسازی در داده‌های مغشوش بررسی شده است. حتی مشخصه‌هایی که انتظار می‌رود خطای کمی در مقادیرشان داشته باشند، می‌توانند از هموارسازی مقادیرشان برای کاهش تغییرات تصادفی استفاده کنند. برخی روشها مثل شبکه‌های عصبی با توابع سیگموئید[2] یا درختان رگرسیونی که از مقدار میانگین یک قسمت استفاده می‌کنند، در بازنمایی خود به‌طور ضمنی هموارساز دارند.

تجمیع[3]: گاه عملیات تلخیص و تجمیع بر روی داده‌ها انجام می‌شود. برای مثال فروش روزانه ممکن است تجمیع شده و به شکل فروش هفتگی یا ماهانه نمایش داده شود. این کار عموما در ایجاد مکعب داده [4] استفاده می‌شود.

---

[1]- Smoothing
[2]- Sigmoid Scaling
[3]- Aggregation
[4]- Data Cube

تعمیم[1]**:** جایی که با استفاده از سلسله مراتب مفهومی، داده سطح پایین یا اولیه با مفاهیم سطح بالاتر جایگزین می‌شود. برای مثال ویژگی طبقه‌ای مانند خیابان با مفهومی بالاتر مانند شهر یا کشور عمومیت داده می‌شود. همان‌طور در داده‌ای عددی مانند سن می‌توان آن‌را با یک مفهوم سطح بالاتر مثل جوان، میانسال یا مسن نگاشت کرد.

ساخت ویژگی[2]: جایی که از ویژگیهای موجود ویژگی جدیدی ساخته شده و برای کمک به فرآیند داده‌کاوی به آن اضافه می‌شود. برای مثال، ممکن است ویژگی مساحت را از ضرب دو ویژگی طول و عرض که موجودند، بسازیم.

نرمال‌سازی[3]: جایی که مقیاس داده‌ها چنان تغییر کند که آنها را به یک دامنه کوچک و معین مانند فاصله بین ۱- تا ۱ نگاشت کند. نرمال‌سازی به روشهای گوناگون انجام می‌شود که در ادامه توضیح داده شده است.

نرمال‌سازی

نرمال‌سازی به‌ویژه برای الگوریتمهای دسته‌بندی همچون شبکه‌های عصبی یا اندازه‌گیری فاصله همچون دسته‌بندی از طریق نزدیک‌ترین همسایه و خوشه‌بندی مفید است. در این الگوریتمها نرمال‌سازی باعث می‌شود که وقتی داده‌ها برای اندازه‌گیری فاصله به‌کار می‌روند، داده‌های با مقیاس بزرگ نتیجه را به سمت خویش منحرف نکنند. چندین شیوه برای نرمال‌سازی وجود دارد که ما نرمال‌سازی *Min-Max*، *Z-Score* و نرمال‌سازی با استفاده از مقیاس بندی اعشاری[4] را بررسی می‌کنیم:

---

[1]- Generalization
[2]- Attribute Construction
[3]- Normalization
[4]- Normalization by decimal scaling

نرمال‌سازی *Min-Max*

این روش یک تبدیل خطی بر روی داده‌های اصلی انجام می‌دهد. فرض کنید که Min_A و Max_A به‌ترتیب حداقل و حداکثر مقادیر یک ویژگی باشند. یک نرمال‌سازی *Min-Max* یک مقدار v از A را به مقدار v' در فاصله $[new\ min_A\ ,\ new\ max_A]$ نگاشت می‌کند که:

$$v' = \frac{v - min_A}{max_A - min_A}(new - max_A - new - min_A) + new - min_A \qquad (۶-۲)$$

نرمال‌سازی *Min-Max* رابطه بین مقادیر داده‌های اصلی را حفظ می‌کند.

مثال نرمال‌سازی *Min-Max*: فرض کنید که حداقل و حداکثر مقادیر برای ویژگی درآمد ۱۲۰۰۰ و ۹۸۰۰۰ دلار است. ما می‌خواهیم درآمد را به دامنه نگاشت کنیم. با استفاده از نرمال‌سازی *Min-Max* مقدار ۷۳۰۰۰ دلار برای درآمد تبدیل می‌شود به:

$$\frac{۷۳.۶۰۰ - ۱۲.۰۰۰}{۹۸.۰۰۰ - ۱۲.۰۰۰}(۱.۰ - ۰) + ۰ = ۰/۷۱۶$$

نرمال‌سازی *Z-Score*

در این شیوه مقدار ویژگی با استفاده از میانگین و انحراف استاندارد ویژگی، نرمال می‌شود. مقدار v از ویژگی A به مقدار v' نگاشت می‌شود:

$$v' = \frac{v - \overline{A}}{\sigma_A} \qquad (۷-۲)$$

در اینجا $\overline{A}$ میانگین و σ_A انحراف استاندارد ویژگی A هستند. این شیوه وقتی که حداقل و حداکثر واقعی ویژگی A نامعلوم بوده و یا مقادیر پرت، نرمال‌سازی *Min-Max* را تحت تاثیر قرار می‌دهند، مناسب است.

مثال: فرض کنید که میانگین و انحراف استاندارد ویژگی درآمد ۵۴۰۰۰ و ۱۶۰۰۰ است. با نرمال‌سازی *Z-Score* مقدار ۷۳۶۰۰ برای درآمد به مقدار $\frac{۷۳.۶۰۰ - ۵۴.۰۰۰}{۱۶.۰۰۰} = ۱.۲۵۵$ تبدیل می‌شود.

نرمال‌سازی به‌وسیله مقیاس بندی اعشاری

در این روش نرمال‌سازی به‌وسیله حرکت نقطه اعشار مقدار ویژگی انجام می‌شود. میزان حرکت نقطه اعشار بستگی به حداکثر قدر مطلق مقادیر ویژگی A دارد. یک مقدار v از A با استفاده از رابطه زیر نرمال و به v' تبدیل می‌شود:

$$v' = \frac{v}{10^j} \qquad (۸-۲)$$

جاییکه j کوچک‌ترین عدد صحیح باشد که $Max(|v'|)<1$.

مثال: فرض کنید مقادیر ثبت شده برای ویژگی A است که دامنه آن از ۹۸۶- تا ۹۱۷ است. حداکثر قدر مطلق مقادیر A مقدار ۹۸۶ است. برای نرمال کردن از طریق مقیاس-بندی اعشاری، ما هر مقدار را بر $1000 (j=3)$ تقسیم می‌کنیم. بنابراین مقدار ۹۸۶- به ۰,۰۹۸۶- و ۹۱۷ به ۰,۹۱۷ تبدیل می‌شود.

توجه کنید که در دو شیوه اول لازم است مقادیری از روی داده‌ها به‌دست آمده (مثل میانگین و انحراف استاندارد) و برای نرمال ساختن مقادیر بعدی استفاده شود.

کاهش داده‌ها

اغلب مشکلات داده‌کاوی به علت وجود مقادیر زیادی از نمونه‌ها با ویژگیهای مختلف بوجود می‌آید. به‌علاوه این نمونه‌ها اغلب ابعاد[1] بالایی دارند[۱].

این ابعاد اضافی در مجموعه داده‌های بسیار بزرگ باعث ایجاد مشکلی می‌شوند که در ادبیات داده‌کاوی به آن «مصیبت بُعد[2]» گفته می‌شود.

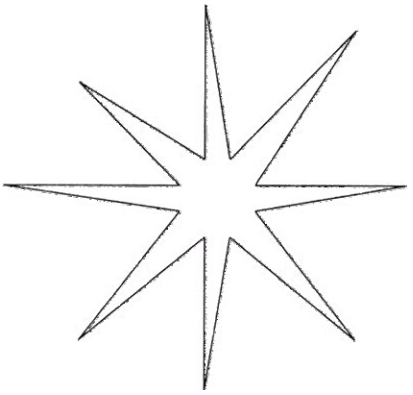
این مسائل به علت حجم بالای داده‌ها در فضایی با ابعاد بالا ایجاد شده و مشکلاتی برای داده‌کاوی ایجاد می‌کرد. به خاطر ذهنیت و تجربه گذشته ما نسبت به فضایی با ابعاد دو یا سه بُعد، اغلب فضایی با ابعاد بالا برای ما غیر منتظره است. به‌طور مفهومی

---

[1]- Dimension
[2]- The Curse Of Dimensionality

اشیاء با حجم معین در یک فضا با ابعاد بالاتر دارای سطح بیشتری نسبت به فضای با ابعاد کمتر هستند.

برای مثال تصویر یک اَبَر مکعب[1] (مکعب چهار بُعدی) شبیه یک خارپشت شکل (۲-۱٤) است. هر چه تعداد ابعاد بیشتر شود، لبه‌ها بیشتر می‌شوند.



شکل۲-۱٤) تصویر یک اَبَر مکعب

چهار ویژگی مهم داده‌های با ابعاد بالا، که کمک زیادی در تفسیر داده‌های ورودی و خروجی می‌کنند، عبارتند از:

۱- با افزایش تعداد ابعاد برای حفظ چگالی نقاط، اندازه مجموعه داده باید به‌صورت نمایی افزایش یابد.

برای مثال اگر در یک نمونه یک بُعدی، N نقطهٔ داده در یک سطح تراکم وجود داشته باشد، برای رسیدن به همان تراکم در یک فضای k بُعدی، نیاز به N^k نقطه است. اگر مقادیر صحیح ۱ تا ۱۰۰ مقادیر نمونه یک بُعدی هستند برای به‌دست آوردن همان تراکم از نمونه‌ها در فضای ۵ بُعدی ما نیاز به $۱۰۰^۵ = ۱۰^{۱۰}$ نمونه متفاوت داریم. این امر

[1]- Hypercube

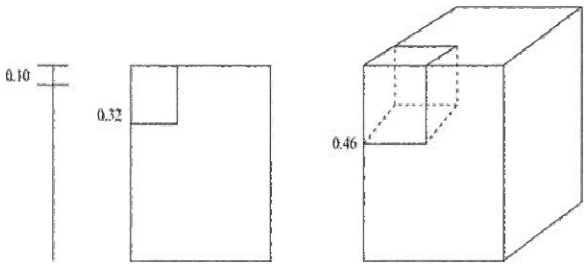
برای مجموعه داده‌های بزرگتر دنیای واقعی نیز درست است. به جهت بُعد بالای آنها اغلب تراکم نمونه‌ها بسیار پایین است که برای داده‌کاوی اصلاً رضایت‌بخش نیست.

۲- در فضایی با ابعاد بالاتر، برای اینکه نسبتی فرضی از نقاط را داشته باشیم باید شعاع بزرگتری داشته باشیم. برای یک نسبت معین از نمونه‌ها طول لبه‌های اَبَرمکعب که با e نمایش داده می‌شود، از رابطه زیر به‌دست می‌آید:

$$e(P) = P^{1/d} \qquad (۹-۲)$$

وقتی که P کسر دلخواه از نمونه‌ها و d تعداد ابعاد است.

برای مثال اگر یکی خواسته باشد تا ۱۰٪ نمونه‌ها را داشته باشد، $(p = ۰/۱)$ لبه اَبَرمکعب برای فضای دوبُعدی برابر $e_۲(۰/۱) = ۰/۳۲$ و برای فضای سه بُعدی $e_۳(۰/۱) = ۰/٤٦$ و برای فضای ده بُعدی $e_{۱۰}(۰/۱) = ۰/۸۰$ است. تفسیر گرافیکی این مسئله در شکل (۲-۱۵) آمده است.



شکل ۲-۱۵) منطقه ای که ده درصد داده‌ها را بپوشاند در یک بعد، دو بعد و سه بعد

این شکل نشان می‌دهد که برای به‌دست آوردن حتی بخش کوچکی از داده‌ها در فضایی با ابعاد بالا نیاز به یک همسایگی بزرگ است.

۳- در فضایی با ابعاد بالا هر نقطه به لبه نزدیک‌تر است و از نقطه‌ای که بیانگر نمونه‌ای دیگر است، دور می‌باشد. برای اندازه n نمونه فاصله مورد انتظار D بین نقاط داده در فضای d بُعدی برابر مقدار $D(d,n)$ است:

$$D(d,n) = ۱/۲(۱/n)^{1/d}$$

برای مثال در یک فضای دوبُعدی با ۱۰۰۰۰ نقطه، فاصله مورد انتظار $D(۲,۱۰۰۰۰)=۰/۰۰۰۵$ و برای فضای ۱۰ بُعدی با همان تعداد نقطه، این فاصله $D(۱۰,۱۰۰۰۰)=۰/٤$ است. به خاطر داشته باشید که حداکثر فاصله هر نقطه با لبه در مرکز توزیع رخ می‌دهد و برای مقادیر نرمال شده تمام ابعاد برابر ۰,۵ است.

٤- اغلب داده‌ها پرت هستند. هر چه ابعاد فضای ورودی افزایش یابد، فاصله بین نقطه پیش‌بینی و مرکز نقاط دسته‌بندی‌شده افزایش خواهد یافت. برای مثال وقتی $d=۱۰$ است، مقدار مورد انتظار نقطه پیش‌بینی ۳,۱ برابر انحراف استاندارد از مرکز داده متعلق به یک دسته دور است. وقتی $d=۲۰$ فاصله برابر ٤,٤ انحراف استاندارد است. از این منظر، پیش‌بینی هر نقطه جدید شبیه یک داده پرت برای داده‌های دسته‌بندی‌شده ابتدایی است. نقاط پیش‌بینی شده در شکل اغلب در لبه‌های خارپشت هستند و از بخش مرکزی دورند.

این قواعد «مصیبت بُعد» وقتی که با تعداد محدود نمونه‌ها در فضای بالا همراه شوند، اغلب نتایج حادی در پی دارند. از ویژگیهای ۱ و۲ ما دشواری تخمین زدن محلی برای نمونه‌های با ابعاد بالا را در می‌یابیم. بنابراین برای فعالیتهای داده‌کاوی در ابعاد بالا نیاز به داشتن نمونه‌های بیشتری است. ویژگیهای ۳ و ٤ دشواری پیش‌بینی پاسخ در یک نقطه فرضی را بیان می‌کنند، چرا که هر نقطه جدید به لبه‌ها نزدیک‌تر خواهد بود تا به نمونه‌هایی در بخش مرکزی.

روشهای کاهش داده می‌تواند برای به‌دست آوردن یک بازنمایی کوچک‌تر و کاهش یافته از داده که بسیار کم‌حجم‌تر از داده‌های اصلی بوده و البته یکپارچگی داده‌های اصلی را حفظ کند، به‌کار رود. بنابراین کاوش روی مجموعه داده‌های کاهش یافته بسیار کارآتر است و البته سبب ایجاد نتایج تحلیلی مشابه می‌شود [٤].

استراتژیهای کاهش داده شامل موارد زیر است:

تجمیع مکعبی داده[1] **(کاهش سطری)**: وقتی تجمیع بر روی داده‌هایی که به شکل مکعب گرد آمده‌اند، انجام شود.

انتخاب زیرمجموعه ویژگی‌ها[2] **(کاهش ستونی)**: وقتی ابعاد با ویژگیهای نامربوط یا با ارتباط ضعیف یا افزونه شناسایی و حذف شوند.

کاهش تعدد نقاط[3] **(کاهش سطری)**: جایی‌که داده به‌وسیله جایگزینهای کوچکتر از داده قبلی با استفاده از مدلهای پارامتریک (که تنها نیاز به ذخیره پارامترهای مدل دارند) یا مدلهای ناپارامتریک مانند خوشه‌بندی، نمونه برداری و استفاده از هیستوگرام کاهش یابد.

گسسته‌سازی و تولید سلسله مراتب مفهومی: جایی‌که مقادیر داده‌های خام با دامنه یا سطوح مفهومی بالاتر جایگزین می‌شود. گسسته‌سازی یک روش کاهش تعدد نقاط است که راه مفیدی برای تولید خودکار سلسله مراتب مفهومی است.

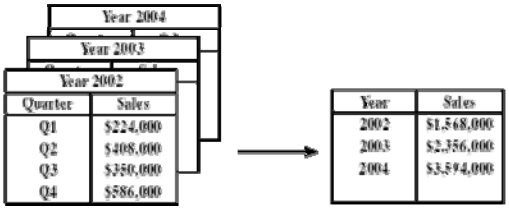
کاهش بُعد[4] **(کاهش ستونی)**: جایی‌که مکانیزم‌های کدکردن برای کاهش اندازه مجموعه داده استفاده می‌شود.

تجمیع مکعب داده

در مکعب‌های داده می‌توان داده‌ها را در ابعاد مختلف تجمیع کرد، بدون اینکه اطلاعات لازم برای وظایف تحلیلی از میان برود. مثلاً در شکل (۲-۱۶) فروش فصلهای مختلف جمع‌آوری شده و سرجمع سالانه آنها نیز محاسبه و نگهداری می‌شود.

---

[1]- Data Cube Aggregation
[2]- Attribute Subset Selection
[3]- Numerical Reduction
[4]- Dimensionality Reduction



شکل ۲-۱۶) تجمیع داده‌های مکعب داده

انتخاب زیرمجموعه ویژگیها

مجموعه داده‌های تحلیلی ممکن است شامل هزاران ویژگی باشد که بسیاری از آنها ممکن است به وظایف کاوش داده ارتباطی نداشته و یا افزونه باشند. برای مثال اگر کار ما دسته‌بندی مشتریان به‌منظور دانستن وجود یا عدم وجود علاقه آنها به خرید محصول جدیدی باشد، ویژگیهایی از قبیل شماره تلفن مشتری نسبتا بی‌ارتباطند، اما به عکس، سن مقوله مرتبطی است. اگر چه این انتخاب می‌تواند توسط فرد خبره انجام شود، اما این کار برای مجموعه‌هایی با ابعاد واقعی دشوار و زمان بر است.

در عمل، نرخ خطای زیرمجموعه‌ها در مقایسه با خطای فوق مجموعه‌ها[1] ممکن است حتی گاهی بهتر باشد. این موضوع به دلیل محدودیت عملی روشهای پیش‌بینی و عدم توانایی آنها برای پویش و یا کاوش[2] در یک فضای جواب پیچیده است. حذف ویژگیهای نامرتبط معمولاً منجر به ساخت مدلی می‌شود که روی داده آزمون بهتر جواب می‌دهد، یعنی تعمیم بهتری دارد. البته در هنگام انتخاب مشخصه تقریباً فقط از خطای آموزشی استفاده می‌شود.

برای n ویژگی 2^n زیرمجموعه وجود دارد، اما چگونه می‌توانیم یک زیرمجموعه خوب از ویژگیهای اصلی را بیابیم؟ وقتی n بزرگ باشد، که در موارد واقعی بزرگ است، آزمودن تمام این زیرمجموعه‌ها، تقریباً ناممکن است. بنابراین روشهای هیوریستیک

---

[1]- Subsets Versus Supersets
[2]- Explore

برای این کار استفاده می‌شوند، که جوابهای بهینه محلی به ما می‌دهند. اما به هرحال عملاً این جوابها در بسیاری موارد پاسخگوی نیازهای ما می‌باشند.

ویژگیهای بهتر یا بدتر عموما به وسیله آزمون‌های معنادار آماری به‌دست می‌آیند، که فرض می‌کنند که ویژگیها مستقل از هم هستند. بسیاری از سنجه‌های ارزیابی دیگر ممکن است به‌کار آیند، همچون سنجه سود اطلاعاتی[1] که در ساختن درختهای تصمیم جهت دسته‌بندی استفاده می‌شود. دو شکل متداول انتخاب مشخصه عبارتند از:

فیلتر: این روش بر اساس معیار حساب شده روی مشخصه‌ها عمل می‌کند.

لفاف[2]: از خطای یک مدل پیش‌بینی برای انتخاب استفاده می‌کند. در هر مرحله از انتخاب مشخصه، مدل پیش‌بینی اجرا می‌شود.

روش فیلتر

در ادامه متداول‌ترین روشهای فیلتر انتخاب مشخصه مبتنی بر میانگین و واریانس مرور می‌شود [۵].

مشخصه‌های مستقل: در این حالت میانگین مشخصه‌های دسته مربوط به یک مسئله دسته‌بندی داده شده، مقایسه می‌شوند. معادلات (۲-۱۰) و (۲-۱۱) آزمون مورد نظر را خلاصه می‌کنند. در آنها se انحراف معیار بوده و مقدار ۲ برای معنادار بودن sig انتخاب شده است. A و B مشخصه یکسانی هستند که برای دسته ۱ و دسته ۲ اندازه‌گیری شده‌اند و n_1 و n_2 تعداد افته‌های متناظر هستند. اگر رابطه (۲-۱۲) برقرار باشد، تفاوت میانگینهای مشخصه معنادار است.

$$se(A-B)=\sqrt{\frac{var(A)}{n_1}+\frac{var(B)}{n_2}} \qquad (۲-۱۰)$$

$$\frac{|mean(A)-mean(B)|}{se(A-B)} > sig \qquad (۲-۱۱)$$

---

[1]- Information Gain
[2]- Wrapper

میانگین یک مشخصه در هر دو دسته بدون توجه به ارتباط آن با مشخصه‌های دیگر مقایسه می‌شود. شاید با داشتن داده‌های زیاد و سطح معنادار بودن دو انحراف معیار، دیگر لازم نباشد یک آزمون آماری انجام شده تا نشان دهد که تفاوت موجود ابداً تصادفی نیست. اگر به هنگام مقایسه، این آزمون رد شود می‌توان ویژگی را حذف کرد. در ۵٪ مواقعی که تفاوتی وجود دارد ولی مشخص نمی‌شود چه باید کرد؟ این تفاوتهای جزئی میانگینها، معمولاً به اندازه‌ای نیستند که به یک مسئله پیش‌بینی با داده‌های زیاد خدشه‌ای وارد کنند. می‌توان گفت که در یک فضای بزرگ حتی سطح اطمینان بزرگتری نیز توجیه‌پذیر است. جالب است که بدانیم بسیاری از مشخصه‌ها از این آزمون ساده شکست خورده و رد می‌شوند.

برای k دسته می‌توان k مقایسه زوجی انجام داد که در آن هر دسته با مکملش مقایسه می‌شود. برای هر یک از مقایسات زوجی اگر مقایسه معنادار باشد، آن مشخصه نگه‌داشته می‌شود. مقایسه میانگینها به‌طور طبیعی برای مسائل دسته‌بندی مناسب است. اگرچه در مسائل رگرسیون این کار پرزحمت‌تر بوده ولی از همان روش می‌توان استفاده کرد. به‌منظور انتخاب مشخصه می‌توان مسئله رگرسیون را یک مسئله شبه دسته‌بندی در نظر گرفت که در آن هدف ما جداسازی خوشه‌های مقادیر از یکدیگر است. برای این‌کار می‌توان به سادگی نیمی از مقادیر بزرگ هدف را در یک دسته و نیمه کوچک‌تر را در دسته دیگر قرار داد.

انتخاب بهینه مشخصه بر مبنای فاصله: اگر به جای بررسی جداگانه مشخصه‌ها، آنها را به‌طور جمعی بررسی نماییم، می‌توانیم اطلاعات بیشتری در مورد آنها کسب کنیم. معمولاً هنگامی‌که مشخصه‌ها را جداگانه بررسی می‌کنیم، ممکن است برخی از ستونهای جدول داده به اشتباه حذف شوند، زیرا این روش قاعدتاً به این نتیجه می‌رسد که برخی از مشخصه‌ها افزونه هستند.

برخی از مشخصه‌ها ممکن است وقتی جداگانه ملاحظه می‌شوند مفید به نظر آیند ولی از نظر قدرت پیش‌بینی افزونه[1] یا زاید باشند. برای مثال ممکن است یک مشخصه چندین بار در جدول داده تکرار شود. اگر این مشخصه‌های تکراری به‌طور جداگانه بررسی شوند همه آنها باقی خواهند ماند، حال آنکه لازم است تنها یکی از آنها برای پیش‌بینی باقی مانده و بقیه حذف شوند.

وقتی ارتباطات ضمنی پیچیده‌ای در فضای جستجو و جواب حاصله وجود دارند، با فرض نرمال یا خطی بودن، راه ظریفی برای انتخاب زیرمجموعهٔ مشخصه وجود دارد. در بسیاری از حالتهای دنیای واقعی فرض نرمال بودن نقض می‌شود و مدل نرمال مدل ایده‌آلی است که نمی‌توان آن را مدل آماری دقیقی برای انتخاب زیرمجموعهٔ مشخصه دانست. توزیع‌های نرمال، دنیای ایده‌آلی هستند که می‌توان در آنها از میانگینها برای انتخاب مشخصه‌ها بهره جست. به هرحال حتی در حالت غیر نرمال، مفهوم فاصله بین میانگینها که با واریانس، نرمال شده باشد برای انتخاب مشخصه‌ها بسیار مفید است. «تحلیل زیرمجموعه» نوعی فیلتر است ولی فیلتری که تحلیل استقلال را برای بررسی مشخصه‌های افزونه به‌نوعی توسعه می‌دهد.

یک توزیع نرمال چندمتغیره با دو توصیف‌گر مشخص می‌شود: M یعنی بردار m میانگین مشخصه و C یعنی ماتریس $m \times m$ کوواریانس میانگینها. هر عنصری از C رابطه یک جفت از مشخصه‌ها است که در رابطه (۲-۱۲) بیان شده است. در این رابطه $m(i)$ میانگین مشخصه i ام، $v(k,i)$ مقدار مشخصه i برای رکورد k ام و n تعداد رکورد است. $C_{i,i}$ یعنی عناصر قطری C، واریانس هر مشخصه هستند و عناصر غیر قطری، همبستگی هر جفت مشخصه می‌باشند.

$$C_{i,j} = \frac{1}{n} \sum_{k=1}^{n} [(v(k,i) - m(i)) \times (v(k,j) - m(j))] \qquad (۲-۱۲)$$

[1]- Redundant

در این جا علاوه بر میانگین و واریانس که برای مشخصه‌های مستقل استفاده شده‌اند، همبستگی بین مشخصه‌ها نیز در نظر گرفته می‌شوند. این کار پایه‌ای برای کشف افزونگی در مجموعه‌ای از مشخصه‌ها است. در عمل روشهای انتخاب مشخصه‌ای که مبتنی بر این اطلاعات باشد نسبت به تحلیل مستقل مشخصه، مجموعه کوچکتری از مشخصه‌ها را انتخاب می‌کنند.

معیار فاصله رابطه (۲-۱۲) را برای تفاوت میانگینهای مشخصه دو دسته در نظر بگیرید. M_1 بردار میانگینهای مشخصه دسته ۱ و C_1^{-1} ماتریس معکوس همبستگی دسته ۱ است. این معیار فاصله یک معیار چندمتغیره مشابه آزمون معنادار بودن استقلال است. به‌عنوان یک روش هیوریستیکی که (به هنگام فقدان اطلاعات در مورد توزیع احتمالی) به‌طور کامل بر داده‌های نمونه استوار است، D_M معیار خوبی برای فیلتر کردن مشخصه‌هایی است که دو دسته را از هم جدا می‌کند.

$$D_M = (M_1 - M_2)(C_1 + C_2)^{-1}(M_1 - M_2)^T \qquad (۲-۱۳)$$

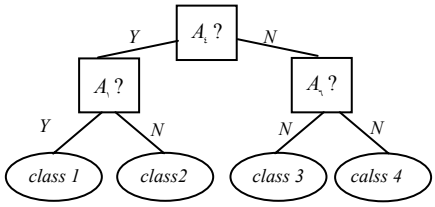
حال ما یک معیار عمومی فاصله بر پایه میانگین و واریانس داریم. لذا مسئله یافتن زیرمجموعه مشخصه‌ها می‌تواند به شکل جستجوی بهترین k مشخصه بر حسب معیار D_M بیان شود. اگر مشخصه‌ها مستقل باشند آنگاه همه عناصر غیر قطری ماتریس معکوس همبستگی صفر بوده و عناصر قطری C^{-1} برابر $\frac{1}{Var(i)}$ برای مشخصه i می‌باشند. در این حالت بهترین مجموعه k مشخصه مستقل، k مشخصه دارای بزرگترین مقدار $(m_1(i) - m_2(i))^2 / (var_1(i) + var_2(i))$ است که در آن $M_1(i)$ میانگین مشخصه i در دسته ۱ و $Var_1(i)$ واریانس آن است. این معیار فیلتر کردن مشخصه، با روش آزمون معنادار بودن مشخصه‌های مستقل کمی تفاوت دارد.

روش لفاف

روشهای هیورستیک لفاف که در انتخاب زیرمجموعه ویژگیها استفاده می‌شوند شامل روشهای زیر می‌باشند:

۱- انتخاب گام‌به‌گام پیش‌رو[1]: فرآیند با یک مجموعه تهی از ویژگیها به‌عنوان مجموعه کاهش یافته[2] آغاز می‌شود. در هر گام تکرار، بهترین ویژگیهای اصلی انتخاب شده و به مجموعه قبلی اضافه می‌شوند.

۲- انتخاب گام‌به‌گام پس‌رو[3] : فرآیند با مجموعه‌ای شامل تمام ویژگیها آغاز به‌کار می‌کند و در هر گام، بدترین ویژگیها از مجموعه حذف می‌شود.

انتخاب پیش‌رو	انتخاب پس‌رو	استنتاج درخت
مجموعه اولیه: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ مجموعه اولیه حذف شده: $\{\}$ $\Rightarrow \{A_4\}$ $\Rightarrow \{A_1, A_4\}$ $\Rightarrow$ مجموعه حذف شده نهایی: $\{A_1, A_4, A_6\}$	مجموعه اولیه: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ $\Rightarrow \{A_1, A_2, A_4, A_5, A_6\}$ $\Rightarrow \{A_1, A_4, A_5, A_6\}$ $\Rightarrow$ مجموعه حذف شده نهایی: $\{A_1, A_4, A_6\}$	مجموعه اولیه: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$  $\Rightarrow$ مجموعه حذف شده نهایی: $\{A_1, A_4, A_6\}$

شکل۲-۱۷) روشهای مختلف انتخاب زیرمجموعه ویژگیها

۳- ترکیب دو روش انتخاب پیش‌رو و حذف پس‌رو[4]: دو روش قبل به‌نحوی با هم ترکیب می‌شوند که در هرگام بهترین ویژگی اضافه شده و ویژگی بدتر حذف می‌شود.

٤- استنتاج درخت تصمیم[5]: الگوریتمهای درخت تصمیم در واقع در نقطه انشعاب درخت، بهترین ویژگی را نیز انتخاب می‌کنند.

---

[1]- Stepwise Forward Selection
[2]- Reduced Set
[3]- Stepwise Backward Elimination
[4]- Combination of Forward Selection and Backward Elimination
[5]- Decision Tree Induction

کاهش تعدد

روشهای کاهش تعدد[1] در حقیقت به منظور انتخاب جایگزینی کوچکتر در بازنمایی داده به‌کار می‌روند[٤].

ممکن است حجم داده‌ها برای برخی از برنامه‌های داده‌کاوی بیش از حد بزرگ باشند. در عصری که صحبت از داده‌های ترابایتی آن هم فقط برای یک کاربرد تنها می‌شود، به‌سادگی امکان تجاوز از ظرفیت یک برنامه داده‌کاوی وجود دارد. روشهای کاهش تعدد روی داده‌های شکل استاندارد اعمال می‌شوند. سپس روشهای پیش‌بینی روی داده‌های کاهش‌یافته اعمال می‌شوند.

این روشها می‌تواند پارامتریک یا ناپارامتریک باشد. برای روشهای پارامتریک، یک مدل برای تخمین داده به‌کار می‌رود و بنابراین برای داشتن تخمینی از داده‌ها نیاز داریم تا تنها پارامترهای مدل را (نه همه داده‌های واقعی) نگه داریم. نمونه روشهای پارامتریک، رگرسیون و مدلهای خطی- لگاریتمی[2] و نمونه مدلهای ناپارامتریک، هیستوگرام، خوشه‌بندی و نمونه‌برداری آماری است. بسیاری از این روشها در هموارسازی مطرح شدند.

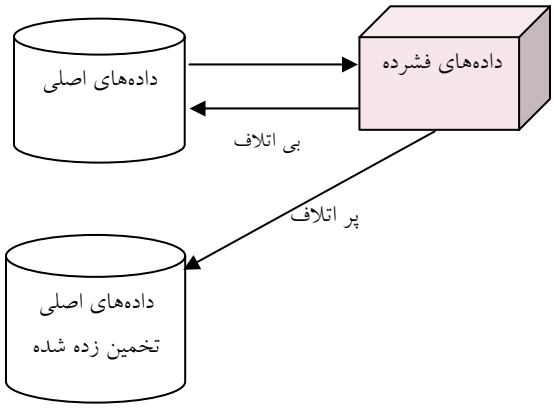
کاهش بُعد

این بخش جزء مباحث پیشرفته داده‌کاوی و پیش‌پردازش داده‌ها محسوب می‌شود. توصیه می‌شود مفاهیم و نیز تحلیل مؤلفه‌های اصلی مطالعه‌شده و بقیه مطالب قبل از داده‌کاوی سری‌ها زمانی مطالعه شوند.

در کاهش بُعد، تبدیلات و کدگذاریهایی روی داده انجام می‌شود که در نهایت بازنمایی کاهش یافته یا فشرده‌ای از داده‌های اصلی به‌دست می‌آید [٤].

---

[1]- Numerisity
[2]- Log Linear



شکل ۲- ۱۸) فشرده سازی بی اتلاف و پر اتلاف

اگر بدون از دست دادن داده‌ها، داده‌های اصلی از داده‌های فشرده قابل بازسازی باشد این کاهش داده، بدون اتلاف[1] نامیده می‌شود و اگر این بازسازی امکان پذیر نباشد و به عبارت دیگر در این تبدیل برخی از داده‌ها از میان بروند، این کاهش داده را با اتلاف[2] می‌گویند.

تعاریف و مفاهیم

برخی از داده‌ها مانند داده‌های متنی، سری‌های زمانی و داده‌های تصویری، دارای صدها و هزاران بُعد می‌باشند. بسیاری از الگوریتمهای داده‌کاوی نمی‌توانند با داده‌ای با ابعاد زیاد کار کنند. علاوه براین در داده‌های معمولی نیز بسیاری از ابعاد به دلیل همبستگی با ابعاد دیگر تا حد زیادی افزونه هستند. بنابراین لازم است قبل از تحلیل، ابعاد داده‌های پر بُعد کاهش داده شوند. برای مصورسازی و تحلیل اکتشافی نیز نیاز به کاهش ابعاد به ۲ یا ۳ بعد می‌باشد.

---

[1]- Lossless
[2]- Lossy

روشهای کاهش بعد، نمایش کوتاهتری از مجموعه داده‌های اولیه را محاسبه می‌کند. این نمایش معمولاً یک نمایش تغییر یافته است، زیرا هنگام انتخاب نمایش کوتاهتر، بعضی از اطلاعات از بین رفته‌اند. روشهای کاهش بعد برای نگهداری ساختار اصلی تا حد امکان تلاش می‌کنند. دو گروه عمومی برای تشخیص این روشها مطرح است: [۷]

۱- حفظ شکلی یا محلی (تغییر ندادن)

۲-حفظ توپولوژی یا عمومی

اولین گروه شامل روشهایی است که اجزاء عمومی مجموعه داده را تغییر نداده و بیشتر تلاش می‌کنند تا نمایش هر دنباله را بدون توجه به بقیه مجموعه داده‌ها، ساده کنند. انتخاب k مشخصه باید به‌گونه‌ای باشد که مشخصه‌های انتخاب شده بیشترین اطلاعات سیگنال اصلی را نگه دارند. برای مثال این مشخصه‌ها می‌توانند اولین ضرایب تجزیه فوریه[۱] یا تجزیه موجک[۲] باشند. دومین گروه از روشها بیشتر برای مقاصد تصویرکردن، استفاده می‌شوند (البته به‌کاربردهای تصویری محدود نمی‌شوند) و هدف اصلی آن، کشف نمایش فضای کاهش بعد یافته اشیاء است. این روش با روش قبلی متفاوت است، زیرا هدف آن یافتن k مشخصه به گونه‌ای است که تابع هدف عمومی را کمینه کند. یک مسئله رایج در این گروه به شرح زیراست:

فرض کنید یک جدول داریم که فواصل بین شهرهای مهم ایران را نشان می‌دهد. آیا می‌توان تنها با استفاده از این اطلاعات شهرها را به شکل نقطه‌هایی روی یک نقشه دو بعدی به گونه‌ای که فاصله‌ها تا جای ممکن به همان اندازه داده شده باشد، نشان داد؟ این مسئله را می‌توان با استفاده از مقیاس‌بندی چندبعدی[۳] حل نمود. نتیجه دقیقاً مانند نقشه ایران نخواهد شد، چرا که ممکن است نقاط یک جهت قرار دادی داشته باشند.

---

[1]- Discrete Fourier Transform: DFT
[2]- DWT
[3]- Multidimensional Scaling: MDS

سایر تکنیکهای حفظ عمومی، شامل روشهای تجزیه مقدار منفرد، نگاشت سریع و تکنیکهای غیر خطی مانند هم‌نگاشت و تصویر کردن تصادفی می‌باشند [٦]. از میان این روشها *PCA* هم برای پیش‌پردازش و هم برای مصورسازی استفاده شده و *MDS* فقط برای مصورسازی استفاده می‌شود. از *PCA* برای ایجاد متغیرهای جدید ناهمبسته برای استفاده در رگرسیون نیز استفاده می‌شود.

تحلیل مؤلفه‌های اصلی

تحلیل مولفه‌های اصلی[1] روشی برای تشیخص الگو در داده‌ها و فشرده‌سازی داده‌ها به شیوه‌ای می‌باشد که تشابهات و تفاوت‌های آنها را واضح‌تر نماید. روش *PCA* یک روش آماری مفید می‌باشد که در زمینه‌هایی مانند تشخیص چهره، فشرده‌سازی تصویر و یافتن الگو در داده‌هایی با ابعاد زیاد، کاربرد دارد. درحالی‌که یافتن الگوها در داده‌هایی با ابعاد بزرگ، مشکل است، *PCA* ابزاری قدرتمند برای تحلیل داده می‌باشد. این روش برای مصور کردن داده‌های پُربُعد در ابعاد ۲ یا ۳ نیز استفاده می‌شود. در این بخش گامهای لازم برای اجرای تحلیل مولفه‌های اصلی روی یک مجموعه از داده بیان می‌شود[۸].

- **گام اول**: جمع‌آوری یک مجموعه از داده‌ها

اولین گام تهیه داده‌هایی است که باید تحلیل شوند، مجموعه داده‌هایی که برای توضیح این اصل در این قسمت ارائه می‌شود، دو بعدی می‌باشد که در شکل (۲-۱۹) نمایش داده شده است.

- **گام دوم**: تفاضل از مقدار میانگین

در این قسمت، مقدار متوسط را از هر یک از داده‌های دو بعدی، کم می‌کنیم.

---

[1]- Principal Component Analysis: PCA

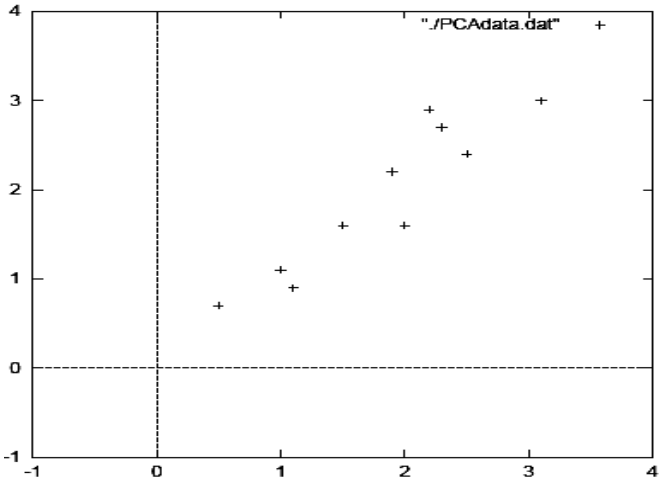
جدول ۲-٤) داده‌های اصلی در سمت چپ، داده‌های حاصل از تفاضل با میانگین در سمت راست،

x	y
۲/٥	۲/٤
۰/٥	۰/۷
۲/۲	۲/۹
۱/۹	۲/۲
۳/۱	۳/۰
۲/۳	۲/۷
۲	۱/٦
۱	۱/۱
۱/٥	۱/٦
۱/۱	۰/۹

داده‌های اصلی

x	y
۰/٦۹	۰/٤۹
−۱/۳۱	−۱/۲۱
۰/۳۹	۰/۹۹
۰/۰۹	۰/۲۹
۱/۲۹	۱/۰۹
۰/٤۹	۰/۷۹
۰/۱۹	−۰/۳۱
−۰/۸۱	−۰/۸۱
−۰/۳۱	−۰/۳۱
−۰/۷۱	−۱/۰۱

داده‌های ساخته شده



شکل ۲-۱۹) داده‌های نمونه PCA و یک نمودار از داده‌ها

- **گام سوم:** محاسبه ماتریس کوواریانس

در این قسمت ماتریس کوواریانس، محاسبه می‌شود. وقتی داده‌ها دو بعدی باشند، ماتریس کوواریانس ۲×۲ خواهد شد. اگر این ماتریس را برای داده‌های ارائه شده محاسبه کنیم، نتیجه به‌صورت ذیل خواهد بود:

$$\text{cov} = \begin{pmatrix} .٦١٦٥٥٥٥٥٦ & .٦١٥٤٤٤٤٤٤ \\ .٦١٥٤٤٤٤٤٤ & .٧١٦٥٥٥٥٥٦ \end{pmatrix}$$

چون همه مؤلفه‌های غیرقطری در این ماتریس کوواریانس مثبت می‌باشند، انتظار داریم که هر دو متغیر x , y توأماً افزایش یابند.

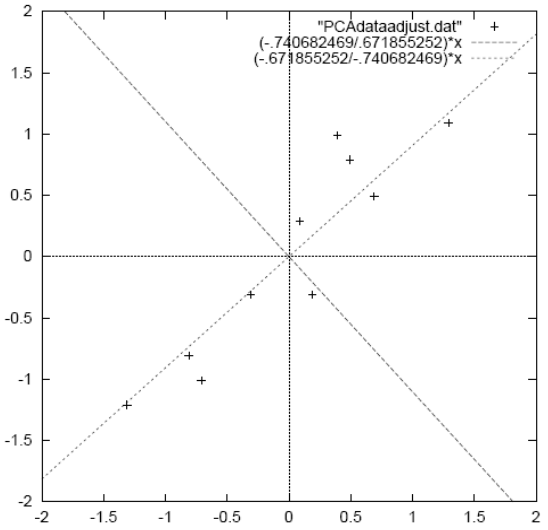
- **گام چهارم:** محاسبه بردارهای ویژه و مقادیر ویژه ماتریس کوواریانس

ماتریس کووارایانس مربعی است و می‌توان بردارهای ویژه و مقادیر ویژه را برای این ماتریس محاسبه نمود. در ادامه مقادیر ویژه و بردارهای ویژه محاسبه شده است:

مقادیر ویژه: $\begin{pmatrix} .٠٤٩٠٨٣٣٩٨٩ \\ ١.٢٨٤٠٢٧٧١ \end{pmatrix}$

بردارهای ویژه: $\begin{pmatrix} -.٧٣٥١٧٨٦٥٦ & -.٦٧٧٨٧٣٣٩٩ \\ .٦٧٧٨٧٣٣٩٩ & -.٧٣٥١٧٨٦٥٦ \end{pmatrix}$

دقت شود که بردارهای ویژه نرمال شده‌اند، یعنی هر یک از آنها با طول واحد می‌باشند. همان‌طور که در شکل (۲-۲۰) مشاهده می‌شود، دو متغیر به همراه یکدیگر افزایش می‌یابند. دو بردار ویژه در بالای داده‌ها، رسم شده است. آنها به‌صورت خطوط نقطه‌چین قطری عمود بر هم می‌باشند.



شکل ۲-۲۰) یک نمودار از داده‌های نرمال شده، به همراه بردارهای ویژه ماتریس کوواریانس

همان‌طور که مشاهده می‌شود، یکی از بردارهای ویژه از میانه نقاط می‌گذرد. این بردار ویژه نشان می‌دهد که چگونه این دو مجموعه داده در طول آن خط، به‌هم مرتبط می‌شوند. بردار ویژه دوم، الگویی با اهمیت کمتر در مجموعه داده‌ها فراهم می‌آورد. بنابراین به‌وسیله این پردازش و محاسبه بردارهای ویژهٔ ماتریس کوواریانس، استخراج خطوطی که داده‌ها را مشخص می‌کنند، ممکن می‌شود. گامهای باقیمانده شامل تبدیل داده است، به‌گونه‌ای که حول و حوش خطوط مذکور فشرده می‌شود.

- **گام پنجم**: انتخاب مولفه‌ها و تشکیل یک بردار مشخصه

اگر به مقادیر ویژه و بردارهای ویژه حاصله در بخش قبل توجه نمایید، متوجه خواهید شد که مقادیر ویژه، تفاوت زیادی با یکدیگر دارند. در واقع، اثبات می‌شود که بردار ویژه با بیشترین مقدار ویژه، مؤلفه اصلی از مجموعه داده می‌باشد. در مثال ارائه شده، بردار ویژه با مقدار ویژه بزرگتر، برداری بود که به پایین مرکز داده اشاره دارد.

این امر مهم‌ترین رابطه بین ابعاد می‌باشد. وقتی که بردارهای ویژه مشخص گردید، گام بعدی مرتب‌کردن آنها برحسب اندازه مقادیر ویژه آنها از بالا به پایین می‌باشد. با این کار مؤلفه‌های با اهمیت کمتر به دست می‌آیند. اگر برخی مؤلفه‌ها حذف شوند، مجموعه داده باقیمانده، نسبت به مجموعه اصلی، ابعاد کوچک‌تری خواهد داشت. به بیان دقیق‌تر، اگر یک مجموعه داده n- بعدی موجود باشد، n بردار ویژه و n مقدار ویژه محاسبه می‌شود، آنگاه تنها P بردار ویژه نخست انتخاب می‌شوند. مجموعه داده-های باقیمانده تنها P بعد خواهند داشت. حال باید بردار مشخصه را تشکیل داد. این بردار به‌وسیله ماتریسی که ستونهایش همان بردارهای ویژه می‌باشند، ساخته می‌شود:

$$FeatureVector = (eig_1\ eig_2\ eig_3\ ....eig_n)$$

در مثال داده شده، با توجه به این که دو بردار ویژه وجود دارد، دو انتخاب نیز وجود دارد. همچنین می‌توان یک بردار مشخصه با هر دو بردار ویژه تشکیل داد:

$$\begin{pmatrix} -.٦٧٧٨٧٣٣٩٩ & -.٧٣٥١٧٨٦٥٦ \\ -.٧٣٥١٧٨٦٥٦ & .٦٧٧٨٧٣٣٩٩ \end{pmatrix}$$

یا می‌توان بردار ویژه مربوط به مقدار ویژه کوچک‌تر را حذف نمود، که در این صورت تنها یک بردار مشخصه با یک ستون به‌دست می‌آید:

$$\begin{pmatrix} -.٦٧٧٨٧٣٣٩٩ \\ -.٧٣٥١٧٨٦٥٦ \end{pmatrix}$$

- **گام ششم:** استنتاج مجموعه داده‌های جدید

این مرحله، آخرین گام و در عین حال ساده‌ترین مرحله در *PCA* می‌باشد. در این مرحله ماتریس مشخصه را ترانهاده کرده و در مجموعه داده اصلی ضرب می‌نماییم:

$$Final\ Data = Row\ Feature\ Vector \times Row\ Data\ Adjust$$

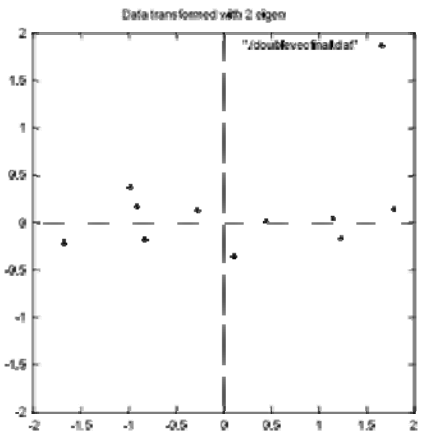
که *Row Feature Vector* ماتریسی با بردارهای ویژه در ستونهایش می‌باشد که ترانهاده شده، به گونه‌ای که بردارهای ویژه در ستونهایش قرارگرفته، بردارهای ویژه مهم‌تر، در ابتدای ماتریس قرار داشته و *Row Data Adjust* ترانهاده داده‌های نرمال شده می‌باشد. ماتریس نهایی داده اصلی را منحصراً برحسب بردارهایی که انتخاب می‌شوند، می‌دهد. در حالتی که هر دو بردار ویژه از تبدیل حفظ شود، داده‌های نهایی حاصله در شکل (۲-۲۱) مشاهده می‌شود. این نمودار اساساً داده اصلی است، به گونه‌ای که حول بردارهای ویژه چرخیده‌اند. در تبدیل دیگر می‌توان تنها یک بردار ویژه با بزرگترین مقدار ویژه را انتخاب نمود. داده‌های حاصله در جدول (۲-۵) مشاهده می‌شوند. طبق انتظار، این داده‌ها تنها یک بعد دارند. اگر این مجموعه داده با مجموعه داده حاصله از حالت قبل مقایسه شود، مشاهده می‌شود که این مجموعه داده، دقیقا ستون اول دیگری می‌باشد. در واقع با استفاده از این تبدیلات، داده‌ها برحسب الگوهای بین آنها بیان می‌شوند. این الگوها خطوطی است که دقیقاً رابطه بین داده‌ها را توصیف می‌کنند. این امر مفید است، زیرا با انجام این کار، نقاط مجموعه به‌عنوان ترکیبی از سهم‌های هر یک از آن خطوط، دسته‌بندی می‌شوند. در روش *PCA*، پس از چرخش، ابعادی انتخاب شده‌اند که دارای بیشترین پراکندگی داده هستند. واریانس هر بعد جدید برابر مقدار ویژه متناظر با آن بُعد است.

جدول ۲-۵) تبدیل داده‌ها با استفاده از دو بردار ویژه

x	y
-۰/۸۲۷۹۷۰۱۸۶	-۰/۱۷۵۱۱۵۳۰۷
۱/۷۷۷۵۸۰۳۳	۰/۱٤۲۸۵۷۲۲۷
-۰/۹۹۲۱۹۷٤۹٤	۰/۳۸٤۳۷٤۹۸۹
-۰/۲۷٤۲۱۰٤۱٦	۰/۱۳۰٤۱۷۲۰۷
-۱/٦۷۵۸۰۱٤۲	۰/۲۰۹٤۹۸٤٦۱
-۰/۹۱۲۹٤۹۱۰۳	۰/۱۷۵۲۸۲٤٤٤
۰/۰۹۹۱۰۹٤۳۷۵	-۰/۳٤۸۲٤٦۹۸
۱/۱٤٤۵۷۲۱٦	۰/۰٤٦٤۱۷۲۵۸۲
۰/٤۳۸۰٤٦۱۳۷	۰/۰۱۷۷٦٤٦۲۹۷
۱/۲۲۳۸۲۰۵٦	-۰/۱٦۲٦۷۵۲۸۷

جدول ۲-٦) داده بعد از تبدیل بوسیله مهمترین بردار ویژه

X
-۰/۸۲۷۹۷۰۱۸٦
۱/۷۷۷۵۸۰۳۳
-۰/۹۹۲۱۹۷٤۹٤
-۰/۲۷٤۲۱۰٤۱٦
-۱/٦۷۵۸۰۱٤۲
-۰/۹۱۲۹٤۹۱۰۳
۰/۰۹۹۱۰۹٤۳۷۵
۱/۱٤٤۵۷۲۱٦
۰/٤۳۸۰٤٦۱۳۷
۱/۲۲۳۸۲۰۵٦



شکل ۲-۲۱) یک نمودار از نقاط داده جدید

تجزیه مقدار منفرد

تجزیه مقدار منفرد[1] از پرکاربردترین روشهای کاهش بعد در تبدیلات *Karhunen Lo`eve* است. این تبدیلات یک روش بهینه برای تصویر نقاط n بعدی به فضای K بعدی است،

[1]- Singular Value Decomposition: SVD

به‌گونه‌ای که خطای تصویر (مجموع فواصل مربع شده) حداقل شود. تبدیلات *KL* مجموعه‌ای از محورهای متعامد است که هرکدام ترکیبی خطی از محورهای اصلی می‌باشد. این محورها با توجه به میزان توانایی آنها برای نگهداری فواصل نقاط در فضای اصلی مرتب شده‌اند.

تبدیلات *SVD* دارای مزیت کاهش بعد بهینه تصاویر خطی می‌باشند. یعنی بهترین حفظ را از میانگین مربع خطا بین تصاویر اصلی و تصاویر تقریبی انجام می‌دهد. البته محاسبه آن در مقایسه با روشهای دیگر دشوار است، مخصوصا اگر تعداد بُعد زیاد باشد (مثلاً در سریهای زمانی خیلی طولانی). علاوه بر این، این روش برای شاخص‌گذاری زیر دنباله‌ها کاربرد ندارد.

روش *SVD* ارتباط نزدیکی با روش تحلیل مؤلفه‌های اصلی دارد. فرق آنها در این است که در تحلیل مؤلفه‌های اصلی باید ابتدا مشخصه‌ها تصحیح به میانگین شوند (میانگین هر متغیر ویژگی از مقادیر آن ویژگی کم شود). هر دو روش، از بردارهای ویژه برای کاهش بُعد استفاده می‌کنند.

تبدیلات گسسته فوریه

این روش طیف فرکانس یک سیگنال یک بعدی را توصیف می‌کند. روش *DFT* به عنوان یک روش کاهش بعد برای سریهای زمانی ارائه شده است. برای سیگنال داده شده $S = (S_0, ..., S_{n-1})$ تبدیل گسسته فوریه به‌صورت رابطه (۲-۱٤) تعریف می‌شود.

$$\sqrt{n} \sum_{i=0,...,n-1} S_i e - j2\pi fi/n \qquad (۲-۱٤)$$

که در آن $f = 0,1,...,n-1, and\ j^2 = -1$ می‌باشد. برای تخمین سریهای زمانی، k ضریب اولیه تبدیل فوریه در نظر گرفته می‌شود. برمبنای تئوری پارسوال رابطه (۲-۱۵) برقرار است.

$$\sum_{i=0,...,n-1} S_i^2 = \sum_{f=0,...,n-1} s_f^2 \qquad (۲-۱۵)$$

این رابطه به این معنی است که محاسبه فاصله‌ها با در نظر گرفتن k ضریب فوریه، یک حد پایین برای فاصله‌اقلیدسی دنباله‌های اصلی فراهم می‌کند. مهمترین مزیت این روش، آن‌است که یک الگوریتم مؤثر برای محاسبات آن وجود داشته و به‌عنوان یک روش کاهش بعد در بسیاری از کاربردها مطرح می‌شود. این به دلیل تمرکز بیشترین انرژی برروی فرکانسهای پائین در این روش است. این روش یک الگوریتم کارآمد با پیچیدگی محاسباتی $n \log n$ است. برای تصویر سریهای زمانی n- بعدی بر روی فضای k بعدی، k ضریب فوریه یکسان باید برای همه سریها، ذخیره شود و ممکن است برای تمام دنباله‌ها، بهینه نباشد. برای یافتن k ضریب بهینه برای M سری زمانی، باید میانگین انرژی را برای هر ضریب محاسبه کنیم.

تبدیل موجک گسسته[1]

تبدیل موجک گسسته یک روش پردازش سیگنال خطی است که به‌کار گرفته می‌شود تا یک بردار از داده‌ها مثل x را به یک بردار x' از ضرایب موجک تبدیل کند. هر دو بردار هم‌اندازه هستند. با به‌کارگیری این روش برای کاهش داده، ما به هر نمونه یا رکوردی به‌عنوان یک بردار داده n بعدی می‌نگریم. به‌عنوان مثال $X(X_1, X_2, ..., X_n)$ بیانگر n مقدار اندازه‌گیری‌شده مبتنی بر رکوردی از n ویژگی پایگاه داده است.

ممکن است این پرسش مطرح شود که «اگر این روش بردار داده ورودی را به برداری هم‌طول تبدیل می‌کند، تأثیرش بر کاهش داده‌ها چیست؟»

پاسخ این است که فایده این کار در حقیقت این است که داده تبدیل شده می‌تواند هرس شود. با نگهداری بخش کوچکی از ضرایب قوی موجک یک تخمین فشرده از داده‌های واقعی به‌دست می‌آید. برای مثال، می‌توان آستانه قبولی را تعیین کرده و تنها ضرایب بزرگتر از آن را نگهداری کرد. البته در این حالت تمام ضرایب دیگر برابر

---

[1]- Discrete Wavelet Transform (DWT)

صفردر نظر گرفته می‌شود. این کار می‌تواند بسیار سریع انجام شده و یک بازنمایی از داده‌ها به‌صورت پراکنده‌تر ایجاد کند.

این روش همچنین می‌تواند برای حذف اغتشاشات بدون هموارسازی ویژگیهای اصلی داده‌ها به کار رود. در نتیجه این روش را می‌توان به خوبی در پاکسازی داده‌ها به‌کار بست.

حال با مجموعه ضرایبی که در دست داریم، می‌توانیم تخمینی از داده‌های اصلی را با استفاده از معکوس تبدیل موجک به‌کار گرفته شده، به‌دست آوریم.

DWT ارتباط نزدیکی با تبدیل فوریه گسسته یا *DFT* دارد. می‌دانیم که *DFT* یک روش پردازش سیگنال با استفاده از توابع سینوسی و کسینوسی است. البته معمولاً *DWT* به فشرده سازی بهتری دست می‌یابد، بنابراین اگر تعداد ضرایب باقیمانده در *DWT* و *DFT* برای یک بردار معین داده برابر باشد *DWT* تخمین بهتری از داده‌های واقعی می‌دهد. از این رو برای یک تقریب برابر، *DWT* نسبت به *DFT* فضای کوچکتری نیاز دارد. برای *DFT* تنها یک گونه تعریف شده در حالی‌که چندین گونه *DWT* وجود دارد. رایج‌ترین تبدیلات موجک شامل *Haar_2*، *Daubechies_4* و *Daubechies_6* است. به‌کارگیری یک تبدیل موجک گسسته از یک الگوریتم هرمی سلسله مراتبی پیروی می‌کند:

در این تبدیل یک توالی به طول ۲^n در ورودی داریم. در صورت لزوم مقادیر اضافه صفر در نظر می‌گیریم تا تعداد، توانی از دو شود. این اعداد به‌صورت جفت جفت با هم جمع شده و این حاصل‌جمع‌ها به مرحله بعد فرستاده می‌شوند. همچنین اختلاف هر جفت نیز محاسبه و ذخیره می‌شود. دوباره این مرحله تکرار می‌شود با این تفاوت که در ورودی، حاصل‌جمع جفتهای مرحله قبل قرار می‌گیرد. این فرایند به‌طور بازگشتی تکرار می‌شود تا در نهایت یک عدد که حاصل‌جمع کل اعداد است، به‌دست آید. این عدد به همراه $۲^n - ۱$ اختلاف جفتها که در مراحل مختلف الگوریتم محاسبه شده به‌عنوان خروجی این تبدیل بازگردانده می‌شود.

به‌عنوان مثال فرض کنید می‌خواهیم تبدیل موجک *Haar* را برروی رشته S بطول ۸ اعمال نماییم. $S=(۱,۳,۵,۱۱,۱۲,۱۳,۰,۱)$

ابتدا این اعداد را به‌صورت جفت جفت با هم جمع می‌کنیم. $(٤,١٦,٢٥,١)$

همچنین اختلاف این جفتها را نیز محاسبه می‌کنیم. $(-٢,-٦,-١,-١)$

واضح است که با استفاده از حاصل‌جمع جفتها و نیز اختلاف جفتها می‌توان رشته S را بدون از دست دادن هیچ اطلاعاتی دوباره بازسازی کرد. مثلاً $\frac{٤-٢}{٢}=١$ می‌شود که عنصر اول S است و $\frac{٤-(-٢)}{٢}=٣$ می‌شود که عنصر دوم S می‌باشد. اکنون با اختلاف جفتها کاری نداریم و فقط آنها را ذخیره می‌کنیم. سپس همین مراحل را برروی این چهار حاصل‌جمع تکرار می‌کنیم. درخت تجزیه تبدیل موجک *Haar* برای یک رشته به طول ۸ در شکل (۲-۲۲) نشان داده شده است.

Resolution	$a_١$	$a_٢$	$a_٣$	$a_٤$	$a_٥$	$a_٦$	$a_٧$	$a_٨$
٣	$a_١+a_٢$	$a_٣+a_٤$	$a_٥+a_٦$	$a_٧+a_٨$	$a_١-a_٢$	$a_٣-a_٤$	$a_٥-a_٦$	$a_٧-a_٨$
٢	$a_١+a_٢+a_٣+a_٤$		$a_٥+a_٦+a_٧+a_٨$		$(a_١+a_٢)-(a_٣+a_٤)$		$(a_٥+a_٦)-(a_٧+a_٨)$	
١	$a_١+a_٢+a_٣+a_٤+a_٥+a_٦+a_٧+a_٨$				$(a_١+a_٢+a_٣+a_٤)-(a_٥+a_٦+a_٧+a_٨)$			

شکل ۲-۲۲) مراحل اجرای تبدیل Haar برروی یک رشته به طول ۸

مراحل اجرای این تبدیل برروی رشته S را می‌توانید در شکل (۲-۲۳) مشاهده کنید.

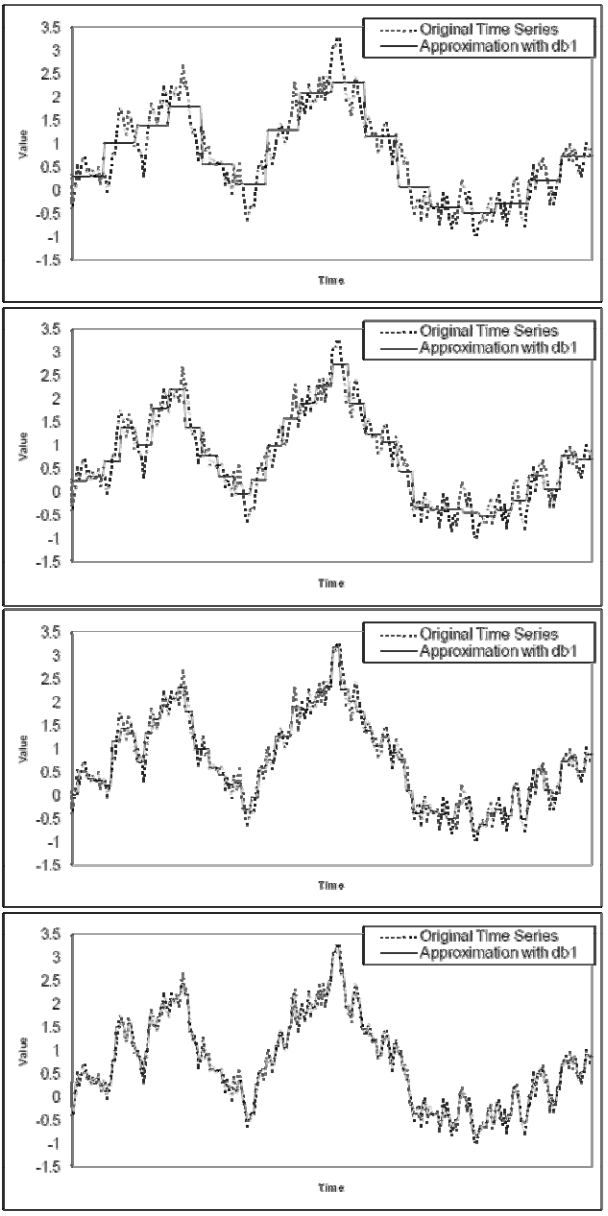
Resolution	*Sum*				*Detail*			
٤	١	٣	٥	١١	١٢	١٣	٠	١
٣	٤	١٦	٢٥	١	-٢	-٦	-١	-١
٢	٢٠		٢٦		-١٢		٢٤	
١	٤٦				-٦			

شکل ۲-۲۳) مراحل اجرای تبدیل Haar برروی رشته S

حاصل‌جمع به‌دست آمده در آخرین مرحله، به‌همراه حاصل تفریق‌هایی که در تمام مراحل ذخیره شده، به‌عنوان خروجی این تبدیل در نظر گرفته می‌شود. بنابراین:

$$DWT(S) = (۴۶, -۶, -۱۲, ۲۴, -۲, -۶, -۱, -۱)$$

می‌توان دید که پیچیدگی زمانی این الگوریتم برای یک رشته به طول n برابر با $O(n)$ می‌باشد. اما چگونه می‌توان با استفاده از تبدیل DWT ابعاد داده را کاهش داد؟ در اینجا نیز همانند تبدیل فوریه، ضرایب به‌دست آمده به ترتیب پراهمیت تا کم‌اهمیت مرتب شده‌اند. در واقع ضرایب کم‌اهمیت همانهایی هستند که در مراحل اولیه الگوریتم به‌دست می‌آیند. (مثلاً کم‌اهمیت‌ترین ضرایب مربوط به $Resolution=3$ هستند، یعنی $(۶,۲, -۱, -۱, -۱)$) با حذف ضرایب کم‌اهمیت می‌توان حجم داده‌ها را کاهش داد. البته مقدار کمی از اطلاعات نیز از بین می‌رود. برای اینکه درک شهودی بهتری نسبت به حذف ضرایب کم اهمیت و تأثیر آن در از دست رفتن اطلاعات داشته باشید به شکل (۲-۲٤) توجه کنید. در این شکل یک سری زمانی که با نقطه چین نشان داده شده، به همراه تبدیل $Haar$ با حذف ضرایب کم اهمیت را مشاهده می‌کنید.



شکل ۲-۲٤) کاهش ابعاد یک سری زمانی توسط تبدیل Haar Wavelet

از بالا به پایین، سطح resolution به ترتیب برابر است با ۳، ٤، ٥ ٦

تصویر کردن تصادفی[1]

تصویر کردن تصادفی، یک روش کاهش بعد عمومی است که درسال ۱۹۹۸ ارائه شد. این روش در سال ۱۹۹۹ برای متن‌کاوی و در سال ۲۰۰۱ برای حوزه سریهای زمانی به‌کار گرفته شد. این روش در عمل بسیار سریع و مفید است، خصوصاً هنگامی‌که همراه با یک روش دیگر به‌کار گرفته شود. برای مثال ما می‌توانیم ازتصویر کردن تصادفی برای کاهش بعد از چندهزار به چندصد استفاده کنیم وسپس روش *SVD* برای کاهش بعد بیشتر به‌کار گرفته شود.

نگاشت سریع

یک تخمین و روش بسیار شبیه به روش مقیاس‌گذاری چند بعدی (*MDS*)، روش نگاشت سریع[2] است. این روش اشیاء را به نقاط k بعدی طوری نگاشت می‌کند که فواصل به خوبی نگهداری شود. یکی از مزایای نگاشت سریع این است که فقط به فواصل بین اشیاء احتیاج داشته و به رابطه اشیاء کاری ندارد. به‌علاوه به‌کاربر اجازه می‌دهد که جستجو را برروی فضای جدید در زمان *O(k)* نگاشت کند.

در این روش N شیء و تابع فاصله *D()* آنها داده شده است. لازم است N نقطه را در فضای kبعدی پیدا کنید به‌طوری که فاصله‌ها تا حد امکان، ثابت نگه داشته شوند.

خصوصیات کاهش بعد به روش نگاشت سریع:

- اشیاء را به نقاط k بعدی به گونه‌ای که فواصل به خوبی نگه داشته شوند نگاشت می‌کند.
- زمانی که تنها فواصل شناخته شده هستند نیز کار می‌کند.
- مؤثر است و از تبدیلات جستجوی مؤثر نیز استفاده می‌کند.

---

[1]- Random Projection
[2]- FastMap

- یک روش کاهش بعد بهینه نیست.

روش کار الگوریتم نگاشت سریع:

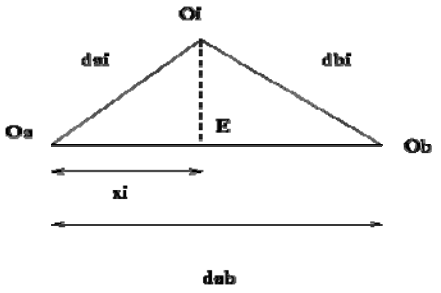
- دو شیء که بیشترین فاصله را نسبت به هم دارند، پیدا می‌کند.
- همه نقاط را روی خطی که از اتصال دو نقطه به وجود می‌آید، تصویر می‌کند و فاصله هرجفت از نقاط تصویر را می‌یابد.
- این کار را $k-۱$ مرتبه ادامه می‌دهد. [٦]
- مفاهیم و نمادهای مرتبط در زیر تعریف شده‌اند:

مفهوم		نماد
تعداد اشیاء موجود در پایگاه داده	=	N
بعد فضای اصلی	=	N
بعد فضای هدف	=	K
تابع فاصلهء بین دو شیء	=	D(*,*)
نرم $L_۲$ بردار X	=	$\|\| X \|\|_۲$
طول بخش AB	=	(AB)

هر شیء به‌عنوان یک نقطه $n-$ بعدی رفتار می‌کند. دو شیء محوری Oa و Ob برای فرآیند نگاشت کردن به‌کار می‌روند. اساس نگاشت براساس قانون کسینوس‌ها است.

$$d_{b,i} = d_{a,i}^{۲} + d_{a,b}^{۲} - ۲x_i d_{a,b} \qquad (۱٦-۲)$$

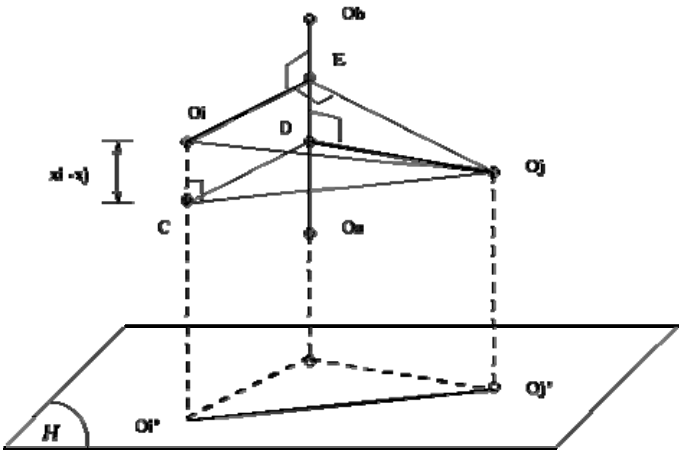
$$x_i = \frac{d_{a,i}^{۲} + d_{a,b}^{۲} - d_{b,i}^{۲}}{۲d_{a,b}} \qquad (۱۷-۲)$$



شکل ۲-۲۵) نمایش قانون نگاشت cosها روی خط Oa و Ob

نگاشت فضای n- بعدی: فرض کنید اشیاء در فضای n- بعدی قرار دارند. فضای Hیک فضای $(n\text{-}1)$ بعدی است که توسط Oa و Ob ساخته می‌شود. سعی می‌کنیم Oi و Oj را در ابرصفحه H تصویر کنیم. تابع فاصله جدید را با استفاده از رابطه فیثاغورث به‌دست می‌آوریم.

$$(D'(O'_i,O'_j))^2 = (D(O_i,O_j))^2 + (x_i - x_j)^2) \quad i,j = 1,\dots,N \qquad (۱۸\text{-}۲)$$



شکل ۲- ۲۶) نگاشت بر روی ابرصفحه H

به نگاشت اشیاء روی ابر صفحه ساخته شده توسط Ob و Oa توجه کنید. تابع فاصله D' بین دو تصویر در رابطه (۲-۱۸) آورده شده است. این بار الگوریتم $Fastmap(K,D(1,0),0)$ را بخوانید.

الگوریتم نگاشت سریع

الگوریتم اولیه (انتخاب تابع فاصله، اشیاء *((o,dist()*)

- به دلخواه یک شیء را انتخاب کرده نام آن را Ob بگذارید.
- Oa را طوری انتخاب کنید که با استفاده ازتابع فاصله بیشترین فاصله را تا Ob داشته باشد.

- Ob را طوری انتخاب کنید که با استفاده از تابع فاصله بیشترین فاصله را تا Oa داشته باشد.
- Oa و Ob را به‌عنوان جفت شیء دلخواه معرفی کنید.

الگوریتم ثانویه

متغیرهای عمومی

ماتریس $N\times K$ به نام X که در پایان الگوریتم i امین سطر آن نمایشگر تصویر i امین شیء است.

ماتریس $K\times$۲ به نام PA که اشیاء محوری یعنی Oa و Obها را در هر مرحله ذخیره می‌کند.

$Int\ col\ \#=0$ که به ستونی از ارائه X که تازه به‌روز شده است اشاره می‌کند.

الگوریتم $Fastmap(K,D(1,0))$

- اگر $K«0$ الگوریتم پایان یافته است.
- در غیر این‌صورت به $\#\ Int\ col$ یک عدد اضافه کن.
- اشیاء محوری را انتخاب کن (Oa و Ob نتیجه الگوریتم اول هستند)
- در ماتریس PA نتیجه سطر ۲ را ثبت کنید:

$PA[۱,COL\#]=a$

$PA[۲,COL\#]=b$

- اگر (Ob و Oa) D آنگاه برای هر i و $X[i,col\#]=$۰ و الگوریتم متوقف می‌شود.
- همهء اشیاء را روی خط Ob و Oa نگاشت کنید. برای هر شیء مثل Oi با استفاده از رابطه (۲-۱۷) xi, را محاسبه کرده و ماتریس عمومی X را کامل کنید.

$X[i,col\#]=xi$

ورودیهای الگوریتم نگاشت سریع

- مجموعه ای از N شیء
- تابع فاصله D
- عدد بُعد دلخواه

خروجی‌های الگوریتم نگاشت سریع

- $X[\,]$ با بعد $N\times K$
- $PA[\,]$ با بعد $۲\times K$

پیچیدگی الگوریتم نگاشت سریع:

- $O(NK)$
- $O(N)$ برای گامهای ۲ تا ۵

نتیجه‌گیری:

- الگوریتمی سریع برای نگاشت اشیاء در فضای k بعدی.
- فاصله (عدم تشابه) میان اشیاء تاجای ممکن ثابت نگه داشته می‌شود.
- شاخص‌گذاری سریع و نگاشت سریع اشیاء جدید
- مفید برای داده‌کاوی، تحلیل خوشه‌بندی و مصورسازی

حل یک مثال عددی با استفاده از الگوریتم نگاشت سریع:

فرض کنید در یک فضای ۳ بعدی، ۳ شیء داریم که می‌خواهیم آنها را به فضای ۲ بعدی ببریم پس:

N=۳ بعد فضای اصلی

K=۲ بعد فضای دلخواه

N=۳ تعداد اشیاء

$S_۱ = \{۱,۲,۳\}$

$S_۲ = \{۱,۱,٤\}$

$S_۳ = \{۲,۱,۲\}$

از ورودیها در می‌یابیم ماتریس PA، ۲×۲ و ماتریس X ,۲×۳ می‌باشد. تابع فاصله $D()$ را فاصله اقلیدسی در نظر می‌گیریم وماتریس فاصله زیر را به‌دست می‌آوریم:

$$\begin{matrix} S_1 \\ S_2 \\ S_3 \end{matrix}\begin{bmatrix} ۰ & \sqrt{۲} & \sqrt{۳} \\ \sqrt{۲} & ۰ & \sqrt{۵} \\ \sqrt{۳} & \sqrt{۵} & ۰ \end{bmatrix}$$
$$S_1 \quad S_2 \quad S_3$$

با توجه به ماتریس فوق به سادگی در می‌یابیم که S_2 و S_3 بیشترین فاصله را از یکدیگر دارند. پس آنها را به‌عنوان Ob و Oa معرفی کرده و در ستون اول ماتریس PA جای می‌دهیم.

$$PA = \begin{bmatrix} ۲ & PA_{12} \\ ۳ & PA_{22} \end{bmatrix}$$

با استفاده از رابطه (۲-۱۷) ستون اول ماتریس X را محاسبه می‌کنیم.

$$x_{11} = \frac{۲+۵-۳}{۲\sqrt{۵}}$$
$$x_{21} = ۰$$
$$x_{31} = \sqrt{۵}$$

و در ستون اول این ماتریس جای می‌دهیم.

$$X = \begin{bmatrix} ۲\sqrt{۵} & x_{12} \\ ۰ & x_{22} \\ \sqrt{۵} & x_{32} \end{bmatrix}$$

تابع فاصله (D')را مطابق رابطه (۲-۱۸) برای محاسبه فواصل بین اشیاء استفاده کرده و دورترین‌ها را به‌عنوان جفت شیء محور انتخاب می‌کنیم.

$$(S_1'S_2')^2 = (\sqrt{۲})^2 - (\frac{۲\sqrt{۵}}{۵})^2 = ۲ - \frac{۴}{۵} = \frac{۶}{۵}$$
$$(S_1'S_3') = (\sqrt{۳})^2 - \frac{۹}{۵} = \frac{۶}{۵}$$
$$(S_2'S_3') = ۵ - ۵ = ۰$$

از نتایج بالا جفت شیء ۱و۲ و یا جفت شیء ۱و۳ را انتخاب کرده و در ستون دوم ماتریس PA قرار می‌دهیم (ما ۱ و۳ را انتخاب کرده ایم)

$$PA = \begin{bmatrix} ۲ & ۱ \\ ۳ & ۳ \end{bmatrix}$$

حال به محاسبه $x_{i۲}$ها می‌پردازیم.

$$x_۱ = ۰$$

$$x_۲ = \frac{\sqrt{۳۰}}{۵}$$

$$x_۳ = \frac{\sqrt{۶}}{\sqrt{۵}} = \frac{\sqrt{۳۰}}{۵}$$

آنها را در ستون دوم ماتریس X جای می‌دهیم:

$$X = \begin{bmatrix} \frac{۲\sqrt{۵}}{۵} & ۰ \\ ۰ & \frac{\sqrt{۳۰}}{۵} \\ \sqrt{۵} & \frac{۵}{\sqrt{۳۰}} \\ & \frac{\sqrt{۳۰}}{۵} \end{bmatrix}$$

ماتریس X نشان می‌دهد که مختصات اشیاء ۱و۲و۳ و در فضای ۲ بعدی به شرح زیر است.

$$O_۱ = (\frac{۲\sqrt{۵}}{۵}, ۰)$$

$$O_۲ = (۰, \frac{\sqrt{۳۰}}{۵})$$

$$O_۳ = (\sqrt{۵}, \frac{\sqrt{۳۰}}{۵})$$

مقیاس‌گذاری چند بعدی

مقیاس‌گذاری چند بعدی نامی کلی برای گروهی از رویه‌ها و الگوریتمها می‌باشد که با یک ماتریس مجاورت[1] ترتیبی شروع کرده و یک پیکره‌بندی از نقاط در یک، دو یا سه بعد ایجاد می‌کند. سمون[2] و کراسکال[3] هر یک سعی می‌کردند یک تابع درجه دو از انحراف فاصله را حداقل کنند (تابع آنها متفاوت است). الگوریتم وقتی خاتمه می‌یابد که خطا از حد مقبول کمتر شده یا اینکه تفاوت مقادیر آن در دو تکرار متوالی الگوریتم ناچیز باشد. در *MDS* داده‌های مقیاس ترتیبی را به مجموعه‌ای از مقیاس نسبی تبدیل می‌کنند. بیشترین توسعه نظری *MDS* در علوم رفتاری و اجتماعی انجام شده است. اکثر کاربردهای مهندسی با یافتن خواص عددی از طریق تصویر کردن الگوها به فضای پایین‌تر شروع شد. در عمل*MDS* برای مصور کردن داده‌ها به‌کار می‌رود نه برای پیش‌پردازش آنها.

نمایش در ابعاد پایین

انسانها معمولاً داده‌ها را در ۲ یا ۳ بعد خوب تحلیل می‌کنند ولی اغلب داده‌هایی که با آنها سر و کار دارند چند بعدی است. یعنی دارای چند ویژگی آشکار یا پنهان می‌باشند. اگر بتوانیم ساختار داده‌ها را در ۲ یا ۳ بعد تصویر[4] کنیم کمک بزرگی خواهد بود. همچنین با اینکه داده‌ها معمولاً با بعد زیادی بیان می‌شوند ولی بعد ذاتی[5] آنها به مراتب کمتر است.[۷]

تصویر کردن خطی توانایی حفظ ساختارهای پیچیده داده را ندارد. مثلاً تحلیل مؤلفه‌های اصلی *(PCA)* نمی‌تواند نمایش دو بعدی مناسبی از داده‌های یک الگوی مارپیچ سه بعدی به‌دست دهد. این موضوع به بعد ذاتی مرتبط است. این موضوع

---

[1]- Proximity
[2]- Sammon, 1969
[3]- Kruskal, 1971
[4]- Project
[5]- Intrinsic

تصویر کردن غیر خطی را در سالیان اخیر متداول‌تر کرده است. اغلب تصویرهای غیر خطی مبنی بر حداکثر یا حداقل کردن یک تابع از تعداد زیادی متغیر هستند. این نوع مسئله بهینه‌سازی وابسته به داده‌ها بوده و تابع نگاشت صریحی ندارد. بنابراین تغییر تعداد الگوها نیاز به محاسبه مجدد کل الگوریتم تصویر را دارد. محاسبات تصویر غیرخطی سنگین بوده و برای کاهش زمان از روشهای ابتکاری استفاده می‌شود. برای مثال اگر ویژگیهای داده صریحاً معلوم باشند، می‌توان بهترین تصویر مؤلفه‌های اصلی را نقطه شروع الگوریتم تصویر غیر خطی در نظر گرفت.

مدلسازی غیرخطی مسئله دارای خواص زیر است:

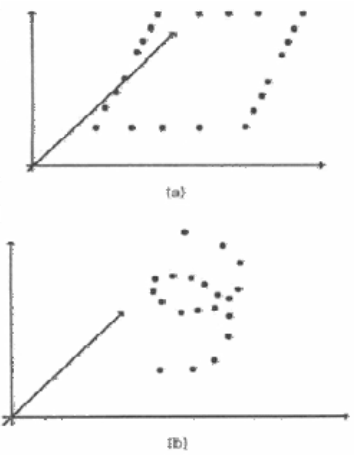
- داده‌های اصلی دارای بعد زیاد هستند.
- داده‌های ذاتی و اساسی دارای بعد بسیار کمتری هستند.
- نگاشت مناسب را طوری پیدا کنید که:

  مشخصه‌های مهم را به بهترین وجه در نظر بگیرد.

  بعد مناسب برای بهترین توصیف داده‌ها در بعد پایین را بیابید.

بُعد ذاتی

بعد ذاتی یا توپولوژی در اصل تعیین می‌کند که آیا می‌توان الگوهای d بعدی را با کفایت در زیرفضای کوچکتر از d تعریف کرد یا خیر. برای مثال الگوهای d بعدی که روی یک سطح صاف قرارگرفته باشند دارای بعد ذاتی ۲ هستند (با ۲ پارامتر قابل تعریف هستند). مفهوم بعد ذاتی با بعد خطی که تعداد مقادیر ویژه مهم ماتریس کوواریانس (در PCA) می‌باشد کاملاً متفاوت است.



شکل ۲-۲۷) بعد ذاتی ۱: (a) بیست و دو نقطه در صفحه با بعد ذاتی یک، (b) بیست نقطه روی یک منحنی با بعد ذاتی یک

الگوریتم *MDSCAL*

ماتریس مجاورت $n \times n$ از عدم تشابه $[d(i,j)]$ داریم. دنبال پیکره‌بندی از نقاط m بعدی $(x_1, x_2, ..., x_n)$ هستیم که در آنها $m.x_i = [x_{i1}, x_{i2}, ..., x_{im}]T$ با توجه به بعد تصویر ۱ یا ۲ یا ۳ است (اغلب ۲ بعدی). مختصات این نقاط باید طوری محاسبه شود که فاصله آنها از هم $[D(i,j)]$ با فواصل مجاورت متناظر جور[1] یا منطبق باشد. معیار فاصله در فضای تصویر شده، فاصله مینکوفسکی است $D(i,j) = (\sum_{k=1}^{m} \left| x_{ik} - x_{jk} \right|^{r})^{1/r}, r \geq 1$. اگر فضای تصویر ۲ بعدی انتخاب شود، فاصله‌اقلیدسی استفاده می‌شود. $(r = 2)$

انطباق کامل وقتی رخ می‌دهد که ترتیب رتبه عناصر ماتریس $[D(i,j)]$ با ماتریس $[d(i,j)]$جور باشد. درجه توافق ترتیب رتبه دو مجموعه با *تنش*[2] کراسکال اندازه‌گرفته می‌شود. قبل از تعریف این معیار به اختصار مشکل قرار دادن نقاط در یک فضا را بررسی می‌کنیم.

---

[1]- Match
[2]- Stress

بدیهی است که می‌توان دو نقطه را طوری روی یک خط قرار داد که فاصله آنها متناسب با عدم تشابه بین دو شیء باشند. سه نقطه در فضای متریک یک صفحه را تعریف می‌کنند بنابراین همیشه می‌توان یک پیکربندی از سه نقطه در فضای دو بعدی طوری تعریف کرد که فواصل بینابین نقاط دقیقاً نظیر عدم تشابه بین سه شیء باشد. در واقع n نقطه در فضای متریک می‌توانند در یک فضای $(n-1)$ بعدی طوری قرار داده شوند که دقیقاً مجاورت بین اشیاء را بازسازی کرده و ترتیب رتبه فواصل نظیر مجاورتهای مرتب داده شده باشند.

برای تعریف تنش، با داشتن n شیء $M = n(n-1)/2$ فاصله داریم که فواصل مرتب شده آنها عبارتند از:

$$D(i_1, j_1) \le D(i_2, j_2) \le \cdots \le D(i_M, j_M) \qquad (۱۹-۲)$$

متناظراً عدم تشابه اشیاء اصلی عبارتند از:

$$[d(i_1, j_1) \le d(i_2, j_2) \le \cdots \le d(i_M, j_M)] \qquad (۲۰-۲)$$

تنش می‌تواند به شکل نمودار شپارد[1] دیده شود که ترسیمی از M نقطه است که هر یک مقادیر (عدم‌تشابه، فاصله) را برای یک زوج از الگوها نمایش می‌دهند. فاصله روی محور افقی نشان داده می‌شود. اگر بتوان همه M نقطه را با دنباله‌ای از خطوط مستقیم دارای شیب غیر منفی به هم وصل کرد، انطباق کامل است. ابتدا منحنی دلخواه از دنباله خطوط متصل با شیب غیرمنفی را در نظر بگیرید. $\hat{D}(i,j)$ را طول افقی تلاقی خط افقی از مختصات $d(i,j)$ در نظر بگیرید. دراین صورت $\left|D(i,j) - \hat{D}(i,j)\right|$ مقدار خارج از منحنی بودن $D(i,j)$ را بر حسب واحد فاصله اندازه می‌گیرد. تنش این منحنی در رابطه زیر تعریف شده و نشان می‌دهد که ترتیب رتبه عدم تشابهات میان اشیاء تا

[1]- Shepard

چه اندازه نظیر فاصله میان نقاط تصویر شده است. رابطه تنش فقط شامل مقادیر محور x که دارای مقیاس نسبی می‌شود زیرا محور y دارای مقیاس ترتیبی است:

$$Stress(curve) = \left[\frac{\sum\sum_{i<j} \left| D(i,j) - \hat{D}(i,j) \right|^2}{\sum\sum_{i<j} D^2(i,j)} \right]^{1/2} \qquad (۲۱-۲)$$

از آنجا که این مسئله یک بهینه‌سازی درجه دو می‌باشد، از یک جواب اولیه (مثلاً مقادیر *PCA*) شروع کرده و با روشهای برنامه‌ریزی غیرخطی مانند حداکثر شیب به‌طور تکراری به سمت جواب بهینه محلی حرکت می‌کند.

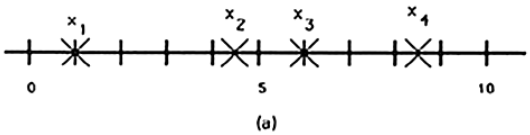
اگر ماتریس عدم تشابه اصلی که از طریق پرسشنامه گردآوری می‌شود، نامتقارن بود با میانگین‌گیری از عناصر متقارن نسبت به قطر اصلی، آن را تبدیل به ماتریس متقارن می‌کنیم.

مثال: ماتریس ترتیب رتبه ۴×۴عدم تشابه $[d(i,j)]$ داده شده است:

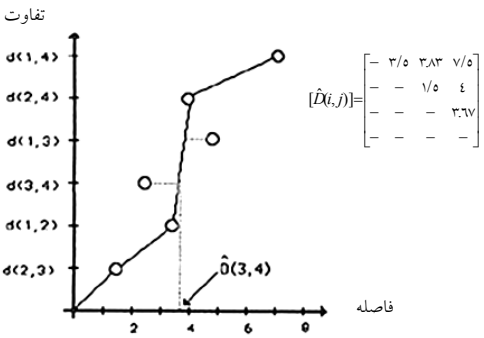
$$[d(i,j)] = \begin{array}{c} \\ ۱ \\ ۲ \\ ۳ \\ ٤ \end{array} \begin{array}{c} \begin{array}{cccc} ۱ & ۲ & ۳ & ٤ \end{array} \\ \begin{bmatrix} - & ۲ & ٤ & ٦ \\ - & - & ۱ & ٥ \\ - & - & - & ۳ \\ - & - & - & - \end{bmatrix} \end{array}$$

شکل (۲-۲۸) یک پیکره‌بندی از چهار نقطه در یک بعد و نمودار شپارد متناظر را نشان می‌دهد. دنباله خطوط مستقیم ترسیم شده در نمودار طوری رسم شده‌اند که تنش را حداقل کنند. نقاط روی نمودار که متناظر با عدم تشابه بین الگوهای $d(۱,٤), d(۲,۳), d(۱,۲), d(۲,٤)$ هستند روی دنباله خطوط مستقیم دارای شیب مثبت قرار دارند. مقادیر ماتریس $[\hat{D}(i,j)]$ از طریق تقاطع بین خطوط تشابه ثابت و قسمتهای پاره خطوط مستقیم به دست می‌آیند. مقدار تنش حداقل شده برابر ۰,۱۵۲ می‌باشد.

$$[D(i,j)]=\begin{bmatrix} - & ۳/۵ & ۵/۰ & ۷/۵ \\ - & - & ۱/۵ & ۱/۰ \\ - & - & - & ۲/۵ \\ - & - & - & - \end{bmatrix}$$



(a)



$$[\hat{D}(i,j)]=\begin{bmatrix} - & ۳/۵ & ۳.۸۳ & ۷/۵ \\ - & - & ۱/۵ & ٤ \\ - & - & - & ۳.٦۷ \\ - & - & - & - \end{bmatrix}$$

$$stress(curve)=\left[\frac{(۱.۱۷)^۲+(۱.۱۷)^۲}{(۳.۵)^۲+(۵.۰)^۲*(۷.۵)^۲*(۱.۵)^۲*(٤.۰)^۲*(۲.۵)^۲}\right]^{\frac{۱}{۲}}=۰.۱۵۲$$

شکل ۲–۲۸) نمودار شپارد

مثال *MDS*

در جدول(۲–۷) هر رنگ دارای ۳ ویژگی (۳ بعد) می‌باشد.

جدول ۲–۷) داده‌های رنگها

رنگ	R	G	B
۱	٦۱	۱٤٦	۳٤
۲	۱۳۹	۱٦۳	۱۷
۳	۱۷۳	۵۰	۷
٤	۲٤٦	۲۵۱	۵۱
۵	۵۹	۲۲۵	۲٤۳
٦	۱۲۳	٦۷	۲۳۵
۷	۲٤۸	۵٤	۸٦
۸	۵٤	۲٤۸	٦۳

اولین کار این است که ماتریس مجاورت (در این جا فاصله) هر دو رنگ (الگو) را محاسبه کنیم. برای مثال عدم تشابه رنگ ۱ با ۲ چنین می‌شود:

$$d_{۱۲} = \sqrt{(۶۱-۱۳۹)^۲ + (۱۴۶-۱۶۳)^۲ + (۳۴-۱۷)^۲} = ۸۱.۶$$

جدول ۲-۸) فاصله رنگها

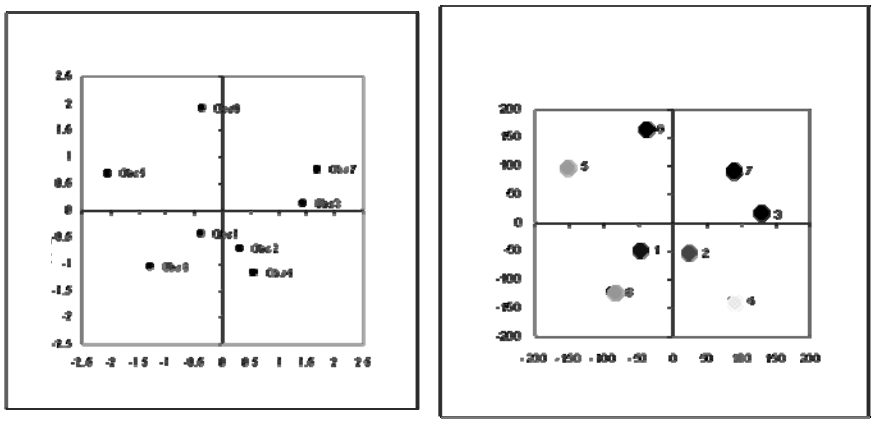
رنگ ۱	رنگ ۲	رنگ ۳	رنگ ٤	رنگ ٥	رنگ ٦	رنگ ۷	رنگ ۸
۰	۸۲	۱۵۰	۲۱۳	۲۲۳	۲۲۵	۲۱۵	۱۰۶
۸۲	۰	۱۱۸	۱٤۳	۲٤۸	۲۳۹	۱۶۹	۱۲۹
۱۵۰	۱۱۸	۰	۲۱۸	۳۱۵	۲۳٤	۱۰۹	۲۳۸
۲۱۳	۱٤۳	۲۱۸	۰	۲٦۹	۲۸۸	۲۰۰	۱۹۲
۲۲۳	۲٤۸	۳۱۵	۲٦۹	۰	۱۷۰	۱۹۰	۲۵۵
۲۲۵	۲۳۹	۲۳٤	۲۸۸	۱۷۰	۰	۱۹۵	۲۵۹
۲۱۵	۱۶۹	۱۰۹	۲۰۰	۱۹۰	۱۹۵	۰	۲۷۵
۱۰۶	۱۲۹	۲۳۸	۱۹۲	۲۵۵	۲۵۹	۲۷۵	۰

سپس از طریق حداقل کردن تابع تنش در *MDS*، در فضای دو بعدی ماتریس فواصل *D(i,j)* چنین به دست می‌آید:

جدول ۲-۹) ماتریس فواصل بعد از MDS

رنگ ۱	رنگ ۲	رنگ ۳	رنگ ٤	رنگ ٥	رنگ ٦	رنگ ۷	رنگ ۸
۰	۲۵	۷۳	۷۰	۳۹	۵۱	۸۷	۱۲
۲۵	۰	۳۰	۳۵	۳۹	۵۲	۹۷	٦۰
۷۳	۳۰	۰	۱۵	۳۵	۲٦	۷۰	۹۱
۷۰	۳۵	۲۹	۰	٤۱	۳۲	۹۵	۹۹
۳۹	۳۹	۳۵	٤۱	۰	٦	۱۹	۸۰
۵۱	۵۲	۲٦	۳۲	٦	۰	۱۷	۷۵
۸۷	۸۷	۷۰	۹۵	۱۹	۱۷	۰	۳۰
۱۲	٦۰	۹۱	۹۹	۸۰	۷۵	۳۰	۰

حالا با داشتن فواصل *D(i,j)* نقاط رنگ را در فضای دو بعدی رسم کرده و با روش *PCA* مقایسه می‌کنیم.



شکل ۲-۲۹) مقایسه MDS (شکل سمت راست) با روش PCA (شکل سمت چپ)

توجه کنید که فقط رعایت ترتیب فواصل مهم است و مقیاس و چرخش مهم نیستند. به‌طور عمومی روش *MDS* سعی می‌کند نقاط را بیش از روش *PCA* پراکنده کند.

منابع

1) *Kantardzic M. (2003) 'Chapter 2: Preparing the Data', Data Mining: Concepts, Models, Methods, and Algorithms, John Wiley & Sons.*
2) *Pyle D. (2003) 'chapter 14: Data Collection, Preparation, Quality, and Visualization', The Handbook Of Data Mining , Edited by Ye N. ,LawrenceErlbaum Associates, Inc.*
3) *http://www.crisp-dm.org/Process/index.htm*
4) *Han. J, Kamber. M. (2006) "Chapter 2: Data Preprocessing", Data mining concepts and techniques, 2nd edition, , Morgan Kaufmann Publishers.*
5) *Ho, T.B (nd) 'KNOWLEDGE DISCOVERY AND DATA MINING TECHNIQUES AND PRACTICE', Unesco Course (cited October 2004). Available from <URL:http://www.netnam.vn/unescocourse/knowlegde/know_frm.htm>*
6) *Ye N. (2003) "The hand book of data mining"*
7) *Jain A. k., Dubes R.C. (1988) "Algorithms for clustering data" Prentice Hall, Available from.*
8) *Smith L.I. (2002) "A tutorial on principal components analysis"*

ضمیمه ۱- مفاهیم پایه آماری

این مفاهیم شامل کوواریانس، انحراف معیار، بردارهای ویژه و مقادیر ویژه می‌باشند.

انحراف معیار

به منظور فهم انحراف معیار، به یک مجموعه داده نیازمندیم. متخصصین علم آمار معمولاً یک نمونه از یک جامعه را انتخاب می‌کنند. جامعه شامل تمام مردم یک کشور می‌شود، درحالی‌که یک نمونه، زیرمجموعه‌ای از جمعیت می‌باشد که به تصادف انتخاب می‌شود. نکته مهم درباره نمونه گیری این است که تنها با اندازه‌گیری یک نمونه از جامعه، می‌توان اطلاعاتی را استخراج کرد که بسیار مشابه اطلاعاتی است که از ارزیابی کل جامعه به دست می‌آید.

مجموعه داده‌های زیر را در نظر بگیرید:

$X = \{۱ ۲ ٤ ٦ ۱۲ ۱۵ ۲۵ ٤۵ ٦۸ ٦۷ ٦۵ ۹۸ \}$

X نشان دهنده مجموعه اعداد می‌باشد. اگر بخواهیم عددی در این مجموعه را نمایش دهیم از زیر نویس برای X استفاده می‌کنیم، مثلاً X_3 نشان دهنده سومین عدد از مجموعه X می‌باشد. با این توصیف می‌توان مفهوم انحراف معیار را توضیح داد.

انحراف معیار عبارتست از فاصله متوسط نقاط مجموعه از میانگین مجموعه، که با نماد S نشان داده شده و توسط رابطه ذیل تعریف می‌شود:

$$S = \sqrt{\frac{\sum_{i=۱}^{n} (X_i - \overline{X})^۲}{(n-۱)}}$$

به‌عنوان مثال انحراف معیار برای دو مجموعه داده محاسبه شده، که در جدول (۲-۱۰) آورده شده است.

جدول ۲- ۱۰) محاسبه انحراف استاندارد

(۱)

X	$(X-\overline{X})$	$(X-\overline{X})^2$
۰	-۱۰	۱۰۰
۸	-۲	۴
۱۲	۲	۴
۲۰	۱۰	۱۰۰
Total		۲۰۸
Divided by (n-۱)		۶۹/۳۳۳
Square Root		۸/۳۲۶۶

(۲)

X	$(X-\overline{X})$	$(X-\overline{X})^2$
۸	-۲	۴
۹	-۱	۱
۱۱	۱	۱
۱۲	۲	۴
Total		۱۰
Divided by (n-۱)		۳/۳۳۳
Square Root		۱/۸۲۵۷

همان‌طورکه انتظار می‌رود، مجموعه نخست انحراف معیار خیلی بزرگتری نسبت به مجموعه دوم دارد، زیرا داده‌ها از مقدار میانگین، پراکندگی بیشتری دارد.

واریانس

واریانس، وسیله دیگری برای اندازه‌گیری انحراف داده‌ها در یک مجموعه می‌باشد. این معیار با نماد S^2 نشان داده شده و توسط رابطه ذیل تعریف می‌شود:

$$S^2 = \frac{\sum_{i=1}^{n}(X_i - \overline{X})^2}{(n-1)}$$

کوواریانس

دو معیاری که در قسمتهای قبل ارائه گردید، صرفا برای داده‌های یک بعدی قابل استفاده می‌باشند. به‌عنوان نمونه اگر با یک هیستوگرام دو بعدی از داده‌ها سروکار داشته

باشیم، تنها می‌توانیم واریانس و انحراف استاندارد را برای یک بعد به‌طور مستقل از بعد دیگر به‌دست آوریم. درحالی‌که، دانستن اینکه بعدهای مختلف چگونه نسبت به یکدیگر از مقدار متوسط فاصله می‌گیرند، مفید است. کوواریانس معیاری برای به‌دست آوردن این دانش می‌باشد. کوواریانس همواره بین دو بعد اندازه‌گیری می‌شود. بنابراین، اگر یک مجموعه سه بعدی $(x\ y\ z)$ از داده‌ها داشته باشیم، می‌توان کوواریانس را بین x و y، y و z، x و z اندازه‌گیری نماییم.

رابطه محاسبه کوواریانس به‌صورت ذیل می‌باشد:

$$cov(X,Y)=\frac{\sum_{i=1}^{n}(X_i-\overline{X})(Y_i-\overline{Y})}{(n-1)}$$

کووارایانس میان دو X و Y را توسط $\mathrm{cov}(X,Y)$ نمایش می‌دهیم. اگر این مقدار مثبت باشد، بدین معناست که هر دو بعد به همراه یکدیگر افزایش می‌یابند. اگر مقدار منفی باشد، بدین معناست که با افزایش در یک بعد، بعد دیگر کاهش می‌یابد. اگر کوواریانس صفر باشد، بدین معناست که دو بعد مستقل از یکدیگر می‌باشند. محاسبه کووارایانس را می‌توان بین هر دو بعد در یک مجموعه داده انجام داد، به‌گونه‌ای که این روش اغلب برای یافتن رابطه بین ابعاد در مجموعه‌های با ابعاد بزرگ استفاده می‌گردد.

ماتریس کوواریانس

اگر یک مجموعه داده با ابعادی بیشتر از دو داشته باشیم، بیشتر از یک اندازه‌گیری کوواریانس را می‌توان محاسبه نمود. تعریف ماتریس کوواریانس برای یک مجموعه داده با n بعد عبارتست از:

$$C^{n\times n}=\left(c_{i,j}\ ;\ c_{i,j}=cov\left(Dim_i,Dim_j\right)\right)$$

که $C^{n\times n}$ یک ماتریس با n سطر و n ستون می‌باشد، و Dim_x، x امین بعد می‌باشد. به عنوان مثال، اگر کوواریانس برای یک مجموعه سه بعدی از داده‌ها محاسبه شود، آن گاه ماتریس کواریانس سه سطر و سه ستون دارد و به فرم ذیل می‌باشد:

$$\begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix}$$

بردارهای ویژه

ضریب یک ماتریس در دو بردار مختلف را در نظر بگیرید:

$$\begin{pmatrix} ۲ & ۳ \\ ۲ & ۱ \end{pmatrix} \times \begin{pmatrix} ۱ \\ ۳ \end{pmatrix} = \begin{pmatrix} ۱۱ \\ ۵ \end{pmatrix}$$

$$\begin{pmatrix} ۲ & ۳ \\ ۲ & ۱ \end{pmatrix} \times \begin{pmatrix} ۳ \\ ۲ \end{pmatrix} = \begin{pmatrix} ۱۲ \\ ۸ \end{pmatrix} = ۴ \times \begin{pmatrix} ۳ \\ ۲ \end{pmatrix}$$

در مثال نخست، بردار حاصل را نمی‌توان به‌صورت حاصل‌ضرب یک عدد صحیح در بردار اصلی نوشت، در حالی که در مثال دوم، ماتریس حاصله را می‌توان به‌صورت حاصل ضرب عدد چهار در بردار اصلی نوشت. ماتریس دیگر، یعنی ماتریس ۲×۲، را می‌توان به‌عنوان یک ماتریس تبدیل تصور کرد. اگر این ماتریس در یک بردار ضرب شود، حاصل ضرب بردار دیگری است که از مکان اصلی‌اش انتقال یافته است. برداری که خاصیت اول را داشته باشد بردار ویژه ماتریس تبدیل، نامیده می‌شود. اگر A ماتریس $n \times n$ و v یک بردار $n \times 1$ باشد و λ عددی صحیح باشد و رابطه ذیل را داشته باشیم:

$$A.v = \lambda.v$$

بردار v یک بردار ویژه برای ماتریس A می‌باشد.

بخش دوم

فصل سوم: تحلیل خوشه‌ای

فصل چهارم: قواعد تلازمی

فصل پنجم: دسته‌بندی و پیش‌بینی

فصل سوم

تحلیل خوشه‌ای

بچه‌ها خیلی زود یاد می‌گیرند گربه را از سگ تشخیص دهند یا بین حیوانات و گیاهان تفاوت قائل شوند. این تشخیص‌ها براساس حس نیمه هوشیار خوشه‌بندی آنها است که به‌طور پیوسته بهبود می‌یابد. تحلیل خوشه‌ای کاربردهای گسترده‌ای مانند: شناسایی متن، تحلیل داده‌ها، پردازش تصویر، تحقیقات بازار و غیره دارد. تحلیل خوشه‌ای به عنوان شاخه‌ای از آمار، نیز مورد مطالعه قرارگرفته و برروی تحلیل فاصله تمرکز دارد. ابزارهای تحلیل خوشه‌ای که مبتنی بر *K-means* و *K-medoids* و روشهایی مانند آنها هستند، در اغلب بسته‌های آماری مانند *SPSS* , *S-plus* , *SAS* وجود دارند. برخلاف دسته‌بندی، خوشه‌بندی و یا یادگیری بدون نظارت، روی دسته‌های از قبل تعریف شده و یا ویژگی خاصی به‌عنوان هدف تکیه ندارد. به همین دلیل خوشه‌بندی بیشتر شکلی از یادگیری بوسیلهٔ مشاهدات است تا یادگیری با مثالها.

در این فصل مباحث زیرمطرح خواهند شد:

- تعریف تحلیل خوشه‌ای
- روشهای خوشه‌بندی افرازی
- روشهای خوشه‌بندی سلسله مراتبی
- روشهای خوشه‌بندی مبتنی بر چگالی
- روشهای خوشه‌بندی مبتنی بر شبکه‌های مشبک
- نقشه‌های خود سازمان

تعاریف و مفاهیم تحلیل خوشه‌ای

خوشه‌بندی، گروه‌بندی نمونه‌های مشابه باهم در یک نمونهٔ داده‌ای می‌باشد. مسئله اساسی خوشه‌بندی عبارت است از: توزیع داده‌ها به K گروه مختلف که نقاط هر گروه با یکدیگر مشابه بوده و داده‌های گروه‌های مختلف با یکدیگر نامتشابه باشند. این تشابه یا عدم تشابه بر اساس معیارهای اندازه‌گیری فاصله تعریف می‌شود. خوشه‌بندی را می‌توان در موارد زیر استفاده نمود:

- تجزیه و تحلیل شباهت یا عدم شباهت: تجزیه و تحلیل اینکه کدام نقاط داده در یک نمونه به یکدیگر نزدیک‌تر می‌باشند.
- کاهش بعد: داده‌های با ابعاد بالا با یک خوشه جایگزین می‌شوند، که این کاربرد بیشتر به‌عنوان پیش‌پردازش داده‌ها مورد استفاده قرار می‌گیرد.

پیش از ادامه مطالب بهتر است اصطلاحات مورد استفاده تعریف شوند:

- خوشه: مجموعه‌ای از اشیاء داده‌ای است به شکلی که اشیاء درون یک خوشه به یکدیگر شبیه‌اند و با اشیاء خوشه‌های دیگر متفاوت هستند.
- تحلیل خوشه‌ای: گروه‌بندی مجموعه‌ای از اشیاء داده‌ای در خوشه‌ها را تحلیل خوشه‌ای گویند. در مقایسه با دسته‌بندی می‌توان گفت: خوشه‌بندی یک دسته‌بندی بدون نظارت است که دسته‌ها از قبل تعریف نشده‌اند.

تجزیه و تحلیل خوشه‌ای روشی برای گروه‌بندی داده‌ها یا مشاهدات با توجه به شباهت آنها است که از طریق آن داده‌ها یا مشاهدات به دسته‌های همگن اما متمایز از یکدیگر تقسیم می‌شوند.

برای درک بهتر تفاوت خوشه‌بندی و دسته‌بندی می‌توان از مثال زیر استفاده کرد. در یک پایگاه دادهٔ مربوط به بازاریابی ممکن است افراد جامعه را به وسیلهٔ متغیرهایی که از قبل به‌عنوان معیارهای مناسبی می‌شناختیم، دسته‌بندی کنیم. در حالی‌که ممکن است به دلیل پیچیدگی پایگاه داده‌ها هیچ نظری در مورد متغیرهای دسته‌بندی کننده و یا چگونگی تعیین آنها نداشته باشیم. در چنین شرایطی بهره‌گیری از روشهای خوشه‌بندی مفید است.

خوشه‌بندی نوعی عملیات داده‌کاوی غیر مستقیم است. در اکثر روشهای داده‌کاوی مثل درخت تصمیم و شبکه‌های عصبی، با یک مجموعهٔ آموزشی شروع کرده و به کمک این مجموعه سعی می‌کنیم یک مدل ایجاد نماییم که داده‌ها را بخش‌بندی کرده و سپس برای یک دادهٔ جدید دسته مناسب را پیش‌بینی کنیم. اما در روش خوشه‌بندی هیچ دسته‌ای از قبل وجود ندارد و در واقع متغیرها به‌صورت مستقل و وابسته تقسیم نمی‌شوند. در اینجا تمرکز روی گروه‌هایی از داده‌ها است که به هم شبیه هستند، تا با کشف این شباهتها بتوان رفتارها را بهتر شناسایی کرده و بر مبنای این شناخت بهتر تصمیم گیری نمود.

در واقع تکنیکهای خوشه‌بندی منظر مناسبی از اتفاقاتی را که در پایگاه داده‌ها در حال رخ دادن است به استفاده‌کنندگان ارائه می‌دهند. البته در برخی موارد از خوشه‌بندی استفاده‌های دیگری نیز می‌شود. به‌عنوان مثال می‌توان از خوشه‌بندی برای تشخیص داده‌هایی که با سایر داده‌ها تفاوت چشمگیر دارند، استفاده نمود (داده‌های پرت). مثلاً به‌جز یکی از مشتریان، دیگران خریدی بالای ۱۰۰ هزار تومان در ماه دارند.

برخی کاربردهای خوشه‌بندی

- شناسایی متن
- تجزیه و تحلیل داده‌های فضایی: که شامل ایجاد نگاشتهای شماتیک در سیستم اطلاعات جغرافیایی توسط خوشه‌بندی شکل فضاها و سپس شناسایی خوشه‌های فضایی و شرح آنها در داده‌کاوی فضایی می‌باشند.
- پردازش تصویر
- علوم اقتصادی
- بازاریابی: به بازاریاب کمک می‌کند تا بتواند گروه‌های مجزایی مبتنی بر مشتریانش را کشف کرده و این دانش خود را برای توسعه برنامه‌های بازاریابی مورد نظرش استفاده کند. این روش در بخش‌بندی گروه‌های مصرف‌کننده محصول و تمرکز روی گروه هدف در بازاریابی بسیار کاربرد دارد.
- خاک برداری: شناسایی مناطقی که دارای خاک مشابه در زمین هستند.
- بیمه: شناسایی دارندگان سیاست بیمهٔ موتوری که میانگین هزینهٔ بالایی را ادعا می‌کنند.
- برنامه‌ریزی شهری: شناسایی گروه‌هایی از خانه‌ها که بر اساس نوع، ارزش و مکان جغرافیایی تقسیم‌بندی شده‌اند.
- مطالعات زمین‌لرزه: مراکز زمین‌لرزه‌های مشاهده شده بر اساس ویژگیها خوشه‌بندی شوند.

خوشه‌بندی خوب چه خوشه‌بندی است؟

یک روش خوشه‌بندی خوب، خوشه‌هایی با کیفیت بالا براساس دو معیار زیر تولید می‌کند: شباهت بالای نقاط داخلی‌هر خوشه و شباهت کم بین نقاط خوشه‌های مختلف. کیفیت نتایج خوشه‌بندی بستگی به روش اندازه‌گیری شباهت به‌کار رفته و همچنین پیاده‌سازی آن روش دارد.

اندازه‌گیری کیفیت خوشه‌بندی

ابتدا باید یک «تابع سنجش تشابه» تعریف شود که شباهت دو نقطه به یکدیگر را نشان دهد. عکس این تابع، تابع فاصله‌است که میزان عدم تشابه دو نقطه از یکدیگر و در نتیجه فاصله (فرضی) بین آن دو داده را نشان می‌دهد. گاه نیز یک تابع جداگانهٔ کیفیت وجود دارد که کیفیت یک خوشه را اندازه‌گیری می‌کند. اما چنین توابعی نیز اغلب بر اساس همان معیار تشابه عمل می‌کنند. تعریف تابع فاصله معمولاً برای انواع داده‌های فاصله‌ای، دودویی، دسته‌ای، ترتیبی و نسبی متفاوت است، به این صورت که میزان اهمیت ابعاد مختلف یک فضا را مشخص می‌کنند.

انواع داده‌ها در تحلیل خوشه‌ای

در این قسمت انواع داده‌ها را بیان می‌کنیم و چگونگی پردازش روی آنها را شرح می‌دهیم. فرض کنید مجموعه‌ای از داده‌ها که باید خوشه‌بندی شوند، شامل n شیء باشد. این داده‌ها ممکن است داده‌های مرتبط با اشخاص، خانه‌ها، مدارک، کشورها و غیره باشند. در الگوریتمهای خوشه‌بندی دو نوع ساختمان دادهٔ خاص که به شکل ماتریس می‌باشد اهمیت به‌سزایی دارند. این ساختمان داده‌ها عبارتند از: ماتریس داده و ماتریس تمایز، که در ادامه معرفی می‌شوند:

ماتریس داده (شیء – ویژگی): این نوع ساختمان داده n شیء را با P ویژگی مانند سن، وزن، ارتفاع و غیره نمایش می‌دهد. یعنی:

$$\begin{array}{c} \text{ویژگی} \rightarrow \\ \text{شیء} \downarrow \end{array} \begin{bmatrix} x_{11} & \dots & x_{1f} & \dots & x_{1p} \\ \dots & \dots & \dots & \dots & \dots \\ x_{i1} & \dots & x_{if} & \dots & x_{ip} \\ \dots & \dots & \dots & \dots & \dots \\ x_{n1} & \dots & x_{nf} & \dots & x_{np} \end{bmatrix}$$

شکل ۳-۱) ماتریس داده

این ماتریس داده‌ها کاملاً شبیه یک جدول در پایگاه داده است در شکل فوق ماتریسی که شامل n داده مختلف (رکورد پایگاه داده‌ها) که هر کدام P بعد دارند، مشاهده می‌کنید.

ماتریس تمایز[1] (شیء – شیء): این ماتریس فاصله یا عدم تشابه بین هر دو عنصر را مشخص می‌کند و معمولاً $n \times n$ می‌باشد. $d(i,j)$ یک اندازه برای نمایش تمایز وعدم شباهت بین اشیاء j,i می‌باشد.

شی →

شی ↓

$$\begin{bmatrix} ۰ & & & & \\ d(۲,۱) & ۰ & & & \\ d(۳,۱) & d(۳,۲) & ۰ & & \\ \vdots & \vdots & \vdots & & \\ d(n,۱) & d(n,۲) & \dots & \dots & ۰ \end{bmatrix}$$

شکل ۳-۲) ماتریس تمایز

برای شباهت و یا عدم شباهت بین اشیاء معمولاً فواصل معیارهای خوبی هستند. برخی از فواصل معروف عبارتند از:

فاصله مانهاتان:

$$d(i,j) = |x_{i۱} - x_{j۱}| + |x_{i۲} - x_{j۲}| + \dots + |x_{ip} - x_{jp}| \qquad (۱-۳)$$

فاصله مینکوفسکی[2]:

$$d(i,j) = (|x_{i۱} - x_{j۱}|^q + \dots + |x_{ip} - x_{jp}|^q)^{\frac{۱}{q}} \qquad (۲-۳)$$

به ازای $q = ۲$ فاصله‌اقلیدسی به دست می‌آید. این فواصل دارای خواص زیر هستند:

۱) $d(i,j) \geq ۰$

۲) $d(i,i) = ۰$

---

[1]- Dissimilarity - Distance
[2]- Minkowski

$$d(i,j) = d(j,i) \quad (۳$$

که البته در بعضی موارد شرط سوم قابل تعدیل است.

$$d(i,j) \le d(i,k) + d(k,j) \quad (٤$$

توجه داشته باشید که فاصله مینکوفسکی، در اصل حالت کلی‌تری برای فاصله مانهاتان و فاصله‌اقلیدسی است. برای qهای بزرگتر از ۲ این رابطه معنی خاصی ندارد اما می‌تواند در بعضی شرایط جوابهای بهتری بدهد (مثلاً در شرایطی که می‌خواهیم به فواصل دور وزن بیشتری بدهیم، می‌توان از اعداد بزرگتر و یا حتی اعشاری نیز استفاده کرد). q در اغلب موارد عددی طبیعی انتخاب می‌شود.

مقیاس‌دهی و وزن‌دهی

چون پراکندگی و مقیاس داده‌های مختلف با یکدیگر متفاوت هستند لذا ابتدا باید همه داده‌ها به یک مقیاس تبدیل شوند.[۱]

معمولاً برای مقیاس‌دهی سه روش متداول است تا بتوان تمامی متغیرها را به یک محدودهٔ قابل قیاس تبدیل نمود:

- تقسیم کردن یک متغیر به متوسط مقادیری که می‌تواند بگیرد.
- از یک متغیر، کمترین مقدارش را کم کرده و بر محدوده تقسیم می‌کنیم.
- از متغیر مورد نظر، میانگین را کسر نموده و بر انحراف معیار تقسیم می‌کنیم.

اگر ما معتقد باشیم که خانواده‌هایی با درآمد یکسان شبیه‌تر از خانواده‌های با تعداد افراد یکسان هستند، این مطلب را به چه صورت نشان دهیم؟ اینجاست که از وزن‌دهی ویژگیها استفاده می‌کنیم. باید به این نکته توجه داشت که به‌دست آوردن وزنها دارای اهمیت بسیاری است و در عمل می‌توان وزنهای متفاوتی داد و جوابها را در حالت

۲- این عمل را می‌توان معادل نرمال‌سازی داده‌ها دانست.

متفاوت بررسی نمود و سپس برای اندازه‌گیری تمایز بین اشیاء با توجه به متغیرهای مختلف به بحث پرداخت.

انواع متغیرها

متغیرهای فاصله‌ای

متغیرهایی که مجموعه مقادیرشان را یک فاصله تشکیل می‌دهد، متغیرهای فاصله‌ای نام دارند مانند وزن و ارتفاع. برای استاندارد کردن این متغیرها برای هر ویژگی مانند f نسبت به اشیاء i $(i = ۱,۲,...,n)$ به‌صورت زیر عمل می‌کنیم:

$$z_{if} = \frac{x_{if} - m_f}{s_f} \qquad (۳-۳)$$

$$m_f = \frac{۱}{n}(x_{۱f} + x_{۲f} + ... + x_{nf}) \qquad (٤-۳)$$

$$s_f = \frac{۱}{n}(|x_{۱f} - m_f| + |x_{۲f} - m_f| + ... + |x_{nf} - m_f|) \qquad (۵-۳)$$

که در آن x_{if} مقدار ویژگی f درشیء i ام می‌باشد.

پس با یکی از فاصله‌های «اقلیدسی»، «مانهاتان» یا «مینکووسکی» تمایز بین اشیاء را اندازه‌گیری می‌کنیم. توجه داشته باشید که این فاصله‌ها در اصل عدم تشابه بین نقاط را نشان می‌دهند. در متغیرهایی با مقیاس فاصله‌ای علاوه بر اینکه ترتیب متغیرها مشخص می‌شود، میزان فاصله بین آنها نیز معین می‌شود. به‌عنوان مثال اگر قد سه نفر به ترتیب ۱۵۰، ۱٦۰ و ۱۷۰ سانتی‌متر باشد علاوه بر اینکه درمی‌یابیم کدام بلند قدتر است، متوجه می‌شویم که این بلندتر بودن به همان میزانی است که فرد دوم از کوتاه قدترین آنها، بلندتر است.

متغیرهای دودویی

متغیرهایی که تنها دو مقدار ۰ یا ۱ دارند، دودویی نامیده می‌شوند. این متغیرها دو نوع متقارن و نامتقارن دارند. متغیر دودویی متقارن متغیری است که دو حالت اخذ شده توسط آن هر دو دارای ارزش یکسانی از نظر تشابه باشند، مانند متغیر جنسیت که فقط

حالتهای مرد و زن را می‌گیرد و مرد و زن بودن دارای یک ارزش هستند. در متغیر دودویی نامتقارن حالتهای مختلف ۰ و ۱ ارزش یکسانی نداشته و هر یک اهمیت خاص خود را دارند. مانند مثبت و منفی شدن جواب آزمایش یک مریض به‌طوریکه مثبت بودن اهمیت زیادتری داشته باشد.[۱] برای اندازه‌گیری تمایز بین اشیاء با این ویژگیها در صورتی که همهٔ آنها از درجهٔ اهمیت یکسانی برخوردار باشند، ماتریس تمایز شکل (۳-۳) را تشکیل می‌دهیم:

شیء		۱	۰	*Sum*
شیء	۱	A	B	$a+b$
	۰	C	D	$c+d$
	Sum	$a+c$	$b+d$	P

شکل۳-۳) ماتریس عدم تشابه

که در آن $p = a+b+c+d$ و a تعداد متغیرهایی هستند که مقادیرشان در هر دو شی i , j برابر ۱ می‌باشد و به همین ترتیب b , c , d طبق جدول تعریف می‌شوند. سپس اگر همهٔ متغیرهای دودویی متقارن بودند، برای عدم تشابه بین شی i , j مقدار زیر را حساب می‌کنیم:

$$d(i,j) = \frac{b+c}{a+b+c+d} \qquad (۶-۳)$$

و برای متغیرهای غیر متقارن عدم تشابه بین دو شی i و jاین گونه محاسبه می‌شود:

$$d(i,j) = \frac{b+c}{a+b+c} \qquad (۷-۳)$$

(d حذف شده است زیرا بنابر قرارداد، متغیرهای منفی یا با مقدار صفر برای هر دو شیj , i اهمیت کمی دارند)

۱- توضیح آنکه اگر دو فرد در مقابل آزمایش یک بیمار نادر جواب مثبت باشند بسیار شبیه یکدیگر هستند اما منفی بودن هر دوی آنها، آن دو نفر را به یکدیگر مشابه نمی کند.

مثال: جدول (۳-۱) افراد مختلفی را که پیش پزشک رفته‌اند با متغیرهای دودویی جنسیت، تب داشتن، سرفه کردن و نتایج انجام چهار آزمایش مختلف نشان می‌دهد:

جدول ۳- ۱) نتایج چهار آزمایش

نام	جنسیت	تب داشتن	سرفه کردن	آزمایش۱	آزمایش۲	آزمایش۳	آزمایش٤
محمد	M	Y	N	P	N	N	N
مریم	F	Y	N	P	N	P	N
علی	M	Y	P	N	N	N	N

ویژگی جنسیت متقارن و بقیه نامتقارنند، جدول (۳-۲) را بدون توجه به متغیر جنسیت شکل می‌دهیم، چون این بیماری به جنسیت وابستگی ندارد.

جدول ۳-۲) نتایج ٤ آزمایش برای دو فرد خاص

		مریم	
		۱	۰
محمد	۱	$a = ۲$	$b = ۰$
	۰	$c = ۱$	$d = ۳$

$$d(\text{مریم، محمد}) = \frac{۰+۱}{۲+۰+۱} = ۰.۳۳$$

مانند جدول فوق می‌توان برای «علی» و «محمد» و «مریم» و «علی» نیز جداول مشابهی ساخت و شباهت مریضی آنها را اندازه‌گرفت در این صورت داریم:

$$d(\text{علی، محمد}) = \frac{۱+۱}{۱+۱+۱} = ۰.۶۷$$

$$d(\text{مریم، علی}) = \frac{۱+۲}{۱+۱+۲} = ۰.۷۵$$

این جدول پیشنهاد می‌کند که (فقط برای متغیرهای غیرمتقارن) مریم و علی مریضی‌شان خیلی کم به هم شبیه است. در حالی‌که مریم و محمد مریضی‌شان بیشتر به هم شبیه است.

متغیرهای اسمی

این متغیرها شبیه متغیرهای دودویی هستند ولی می‌تـــوانند بیش از دو مقدار بگـیرند مانند مجموعه رنگ‌ها یا روزهـــای هفته.

{آبی و صورتی و سبز و زرد و قرمز}= مجموعه رنگها

اگر m تعدادحالات متغیر اسمی باشد، آنگاه موقعیت این حالات را می‌توان با اعداد ۱ و ۲ و... و m نشان داد. برای اندازه‌گیری عدم تشابه اشیاء با توجه به متغیرهای اسمی از رابطه زیر استفاده می‌کنیم:

m تعداد متغیرهایی که اشیاء i و j حالات یکسانی از آن متغیر را دارا می‌باشند.

تعداد کل متغیرها یعنی P نیز تعریف شده است.

$$d(i,j)=\frac{p-m}{p} \qquad (۸-۳)$$

می‌توان این متغیرها را تبدیل به متغیرهای دودویی نمود به این ترتیب که برای هر یک از m حالت متغیر اسمی، یک متغیر دودویی تعریف می‌کنیم که به ازای مکان آن حالت یک و به ازای بقیهٔ حالات، صفر می‌باشد. در اینجا نیز به خوبی مشاهده می‌شود که همرنگ بودن دو شی آنها را به یکدیگر نزدیک می‌کند.

مثال: فرض می‌کنیم i , j دو شیء باشند که ماشین و رنگ مو و رنگ لباس آنها در جدول (۳-۳) آمده است.

جدول ۳-۳) ویژگیهای متفاوت دو فرد خاص

رنگ مو	رنگ لباس	ماشین	
مشکی	خاکستری	سمند	شیء ۱ (فرد اول)
زرد	خاکستری	سمند	شیء ۲ (فرد دوم)
$d(i,j)=(۳-۲)/۳=۰/۳۴$		$p=۳, m=۲$	

متغیرهای ترتیبی

متغیرهای ترتیبی متغیرهای گسسته‌ای هستند که با توجه به ارزش حالتهایشان مرتب شده‌اند. در این متغیرها ارزش ترتیبی هر جایگاه مشخص شده اما فاصله بین این جایگاه‌ها بی‌معنی است. مانند متغیر {برنز، نقره، طلا} = مدال. در این متغیر مشخص است که مدال طلا جایگاهی بهتر از مدال نقره دارد اما مشخص نیست که این برتری به چه میزان است. فرض کنید شمارهٔ حالتهای مختلف متغیر ترتیبی f به‌صورت ۱ , ۲,...,M_f باشد. محاسبهٔ عدم تشابه اشیاء بر پایهٔ این متغیرها در سه قدم انجام می‌گیرد:

قدم ۱) x_{if} را با شمارهٔ مکان مرتب شده‌اش در f جایگزین کنید یعنی:

$r_{if} \in \{۱,...,M_f\}$

قدم ۲) برای اینکه متغیرهای ترتیبی دامنه‌های متفاوتی دارند، لذا آنها را از طریق رابطه زیر به فاصلهٔ [۰,۱] نگاشت می‌کنیم:

$$z_{if} = \frac{r_{if} - ۱}{M_f - ۱} \qquad (۹-۳)$$

قدم۳) حال هر کدام از اندازه‌های فاصله‌ای را می‌توان برای z_{if} به‌کار برد.

متغیرهای نسبی

این نوع متغیرها اندازه‌گیری مثبتی روی مقیاس غیرخطی دارند مانند مقیاس نمایی بر پایهٔ e که Ae^{Bt}, Ae^{-Bt} نمونه‌هایی از متغیرهای نسبی هستند و A و B اعداد ثابتی هستند. به‌عنوان مثال متغیر رشد جمعیت باکتریها نسبت به زمان نمونه‌ای از این نوع متغیرها است در چنین مثالی اعدادی چون ۱۰ و ۱۰۰ و ۱۰۰۰ هر کدام ۱۰ برابر دیگری نیستند بلکه ممکن است رابطه دیگری با یکدیگر داشته باشند. مثلاً اگر باکتریها در هر ثانیه ۱۰ برابر شوند، فاصله این سه متغیر برابر یک واحد زمان است.

برای محاسبه عدم تشابه بین این نوع متغیرها چند روش وجود دارد. یکی از این روشها درنظرگرفتن متغیرها به‌عنوان متغیرهای ترتیبی است.

ترکیب متغیرهایی از انواع مختلف

اگر متغیرهای یک شیء از انواع متغیرها باشند، برای اندازه‌گیری عدم تشابه دو راه موجود است. یکی اینکه متغیرها را براساس انواع مشابه گروه‌بندی کرده و تحلیل خوشه‌ای را روی هر گروه جداگانه انجام دهیم. زمانی این کار مؤثر و ممکن است که بتوانیم نتایج را با هم مقایسه کنیم. اما راه دیگر پردازش همهٔ متغیرها با یکدیگر است که در این صورت فقط یک تحلیل خوشه‌ای داریم. در اینجا می‌بایست همهٔ متغیرها در یک مقیاس مشترک به فاصلهٔ [۰,۱] نگاشت شوند. فرض کنید مجموعهٔ داده‌ها شامل p متغیر و از انواع مختلف باشد. عدم تشابه بین اشیاء i, j را به‌صورت زیر تعریف می‌کنیم:

$$d(i,j) = \frac{\sum_{f=1}^{p} \delta_{ij}(f) d_{ij}(f)}{\sum_{f=1}^{p} \delta_{ij}(f)} \qquad (۱۰-۳)$$

که در آن:

$$\delta_{ij}^{f} = \begin{cases} ۰ & \rightarrow \text{اگر } x_{if} \text{ یا } x_{jf} \text{ یا هر دو باینری و یا بدون مقدار باشند:} \\ ۱ & \rightarrow \text{در غیر اینصورت:} \end{cases} \qquad (۱۱-۳)$$

d_{ij}^{f} بسته به نوع متغیر f به‌صورتهای زیر تعریف می‌شود:

- اگر f دودویی یا اسمی باشد و $x_{if} = x_{jf}$ آنگاه $d_{ij}^{f} = ۰$ و در غیر این‌صورت برابر ۱ است.
- اگر f جزء متغیرهای فاصله‌ای باشد، فاصله به‌صورت زیر محاسبه می‌شود:

$$d_{ij}^{f} = \frac{\left| x_{if} - x_{jf} \right|}{max_h\, x_{hf} - min_h\, x_{hf}} \qquad (۱۲-۳)$$

- در این رابطه h شمارهٔ همهٔ اشیایی است که متغیر f در آنها دارای مقدار است.
- اگر f ترتیبی یا نسبی باشد، ترتیبهای r_{if} و را محاسبه کرده و از z_{if} به‌عنوان مقیاس‌بندیهای فاصله‌ای استفاده می‌کنیم.

در واقع به‌نوعی ابتدا تمام متغیرها را نرمال به بازه صفر و یک کرده و سپس با هم در نظر می‌گیریم.

مثال: در یک تیم فوتبال ویژگیهای رنگ لباس، نوع مدال کسب شده، جواب آزمایش دوپینگ و تعداد گلها در ۱۰ شوت در نظر گرفته شده است. جدول (۳-٤) ویژگیهای مذکور را برای سه تیم مختلف فوتبال نشان می‌دهد. هدف ما در اینجا به‌دست آوردن فاصله سه تیم فوتبال از یکدیگر است.

جدول ۳-٤) ویژگیهای متفاوت سه فوتبالیست

	رنگ	مدال	تست دوپینگ	تعداد گلها در ۱۰ شوت
	(اسمی)	(ترتیبی)	(باینری)	(نسبی)
۱	زرد	*gold* (طلا)	N	۲
۲	-	*silver* (نقره)	N	۱
۳	سبز	*silver* (نقره)	P	۵

برای متغیر ترتیبی مدال، $z_{i۲}^{f}$ فوتبالیستها به ترتیب برابر است با ۰، ۰.۵، ۰.۵

چون سه مدال داریم طلا برابر ۰، نقره $\frac{۱}{۲}$ و برنز برابر $\frac{۲}{۲}$ می‌شوند و برای متغیر نسبی تعداد گلها در ۱۰ شوت، $z_{i٤}^{f}$ در مجموعه حالات ممکن {۱۰,۹,۸,۷,٦,۵,٤,۳,۲,۱,۰} با توجه به اینکه ۱۰ گل رتبه اول ۹ گل رتبه دوم و... تا در نهایت ۰ گل رتبه یازدهم را دارند به‌صورت زیر حساب می‌شود.

$$Z_{if} = \frac{\text{رتبه تعداد گل زده}}{\text{کل تعداد رتبه‌ها}}$$

۰/۵,۰/۹,۰/۸ (مثلاً برای ۲ داریم: $(۹-۱)/(۱۱-۱)=۰/۸$) برای فوتبالیست ۱ و ۲ صفت رنگ ماشین و تست دوپینگ، در محاسبهٔ فاصله در نظر گرفته نمی‌شوند؛ چون اولی اسمی و دومی دودویی نامتقارن است و در هر دو مورد نیز تشابهی وجود ندارد. پس ضرایب آنها صفر بوده و در نتیجه داریم:

$$d(۱,۲) = \frac{۰+۱\times\frac{|۰-۰.۵|}{۰.۵-۰}+۰+۱\times\frac{|۰.۸-۰.۹|}{۰.۹-۰.۵}}{۰+۱+۰+۱} = ۰.۶۲۵$$

در هر مورد پس از به‌دست آوردن قدر مطلق فاصله برای نگاشت آن به فاصلهٔ $[۰,۱]$ آن را بر حداکثر اختلاف موجود بین عناصر مختلف در آن بعد (متغیر) تقسیم می‌کنیم.

روشهای اصلی خوشه‌بندی

رویکردهای اصلی خوشه‌بندی عبارتند از:

- روش افرازی[1]
- روش سلسله مراتبی
- روش مبتنی برچگالی
- روش مبتنی بر مشبک کردن فضا
- روش مبتنی بر مدل

روشهای افرازی

فرض کنید یک پایگاه داده با n شیء داریم. یک روش افرازی، K افراز از این داده‌ها درست می‌کند به‌طوریکه هر افراز یک خوشه را نشان می‌دهد و $k < n$. پس داده‌ها در k گروه خوشه‌بندی شده و دارای دو شرط زیر می‌باشند:

- هر گروه حداقل یک شیء دارد.
- هر شی تنها به یک گروه تعلق دارد. (این شرط در روشهای افرازی فازی می‌تواند قابل انعطاف باشد.)

در روش افرازی برای k معلوم، یک افراز ابتدایی ایجاد می‌شود. سپس یک روش جانشانی تکراری را به‌کار برده که تلاش به بهبود افرازبندی دارد. به این صورت‌که اشیاء را از یک گروه به دیگرگروه‌ها می‌برد. یک معیار عمومی برای یک افرازبندی

[1]- Partitioning

خوب این است که اشیاء در یک خوشه به هم نزدیک یا به یکدیگر وابسته باشند و در مقابل اشیاء در خوشه‌های مختلف، از یکدیگر دور یا تا حد امکان متفاوت باشند.

برای دستیابی به خوشه‌بندی بهینه در روش افرازی، به شمارش کامل همهٔ افرازهای ممکن نیاز خواهد بود یعنی تمام حالات ممکن باید بررسی شوند که این روش برای پایگاه داده‌های بزرگ ناممکن است، لذا الگوریتمهای هیوریستیک زیر برای بررسی این‌گونه موارد استفاده می‌شوند.

- الگوریتم *K-means* که هر خوشه با میانگین اشیاء آن خوشه، نمایش داده می‌شود. (با مرکز خوشه).
- الگوریتم *K-medoids* که هر خوشه با یکی از اشیاء که در نزدیکی مرکز خوشه جای گرفته است، نمایش داده می‌شود.

این روشها برای یافتن خوشه‌هایی به شکل کره در پایگاه داده‌های کوچک تا متوسط به خوبی کار می‌کنند، اما برای یافتن خوشه‌هایی با اشکال پیچیده و یا دارای مجموعه داده‌های بزرگ، باید توسعه داده شوند.

روشهای سلسله مراتبی

روش سلسله مراتبی یک ساختار سلسله مراتبی از اشیاء یک مجموعهٔ معلوم ایجاد می‌کند. روش سلسله مراتبی می‌تواند خوشه‌بندی را به‌صورت تجمیعی و یا به‌صورت تقسیمی انجام دهد. به رویکرد تجمیعی، رویکرد پایین به بالا[1] نیز گفته می‌شود. این روش با شکل‌دهی گروه‌های مجزا که هر یک شامل حداقل یک شیء می‌باشند شروع می‌شود. سپس اشیاء یا گروه‌های نزدیک به هم را یکی می‌کند تا این‌که در نهایت یک گروه کلی در بالاترین سطح ایجاد شود. در روش تقسیمی کل اشیاء در یک خوشه در نظر گرفته شده و در هر تکرار یک خوشه به دو خوشه کوچکتر تقسیم می‌شود.

---

[1]- Bottom - Up

روش مبتنی بر چگالی

بسیاری از روشهای افرازی، اشیاء را بر اساس فاصلهٔ آنها نسبت به یکدیگر خوشه‌بندی می‌کنند. برخی روشها تنها خوشه‌های کروی شکل را پیدا می‌کنند و در برابر خوشه‌هایی به شکلهای دلخواه با مشکل مواجه می‌شوند. در مقابل برخی روشهای دیگر خوشه‌بندی برپایهٔ چگالی توسعه یافته‌اند. ایدهٔ عمومی این روشها رشد دادن خوشه‌ها بر پایهٔ چگالی در همسایگی آنها است. به این معنی که برای هر نقطه داده در یک خوشهٔ معلوم، همسایه‌ای با شعاع مشخص در نظر گرفته می‌شود. این نوع خوشه‌بندی برای هموارسازی اغتشاشات و کشف خوشه‌هایی با اشکال دلخواه به‌کار می‌روند. برخی الگوریتمهای مبتنی بر چگالی عبارتند از *DBSCAN* و *OPTICS*.

روشهای مبتنی بر مشبک کردن فضا

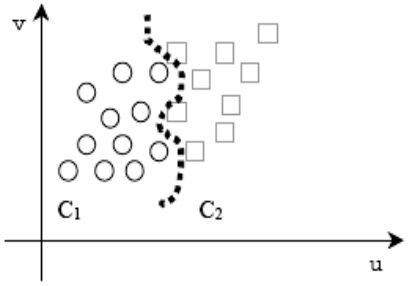
این روش فضای اشیاء را در تعدادی سلول که ساختمانی مشبک شکل دارند، تقسیم-بندی می‌کنند. مهم‌ترین مزیت آن افزایش سرعت پردازش می‌باشد که براساس تعداد سلولها و تعداد نقاط داده متفاوت است. مانند الگوریتمهای *STING*، *CLIQUE*، *WAVE*.

روشهای مبتنی بر مدل

در این روشها برای هر خوشه یک مدل فرض شده و سعی می‌کنند داده‌ها را به بهترین نحو ممکن در آنها جای دهند. یک الگوریتم مبتنی بر مدل، ممکن است خوشه‌ها را با یک تابع چگالی تعیین محل کرده و یا راهی برای تعیین تعداد خوشه‌ها با استفاده از استانداردهای آماری ارائه کند. البته با توجه به پیش‌فرضی که در این روشها در مورد خوشه‌ها در نظر گرفته می‌شود گاه این روشها را خارج از دامنه خوشه‌بندی و مثلاً زیر مجموعه‌ای از دسته‌بندی فرض می‌کنند.

در این بخش، ابتدا به توضیح الگوریتمهای مهم خوشه‌بندی بر مبنای افراز به نام‌های *K-means* و *K-medoids* می‌پردازیم و سپس در ادامه به بررسی دو روش *AGNES* و *CLARA* که دو نمونه از خوشه‌بندیهای سلسله مراتبی هستند می‌پردازیم.

روش افرازی[1]



شکل ۳-٤) افرازی

فرض کنیم یک پایگاه داده با n شی داریم. علاوه بر آن تعداد خوشه‌هایی که باید تشکیل شوند، نیز معلوم باشند. یک الگوریتم افرازی، اشیاء را در K افراز سازماندهی کرده به‌طوریکه هر افراز یک خوشه را نمایش می‌دهد. خوشه‌ها معمولاً با معیاری که تابع شباهت نام دارد[2]، شکل می‌گیرند. بنابراین اشیاء داخل یک خوشه به هم شبیهند و در مقابل اشیاء درخوشه‌های مختلف به هم شبیه نیستند. این شباهت و عدم شباهت اشیاء بر مبنای داده‌های پایگاه داده تعیین می‌شود. دو الگوریتم مهم این روش عبارتند از *K-means ,K-medoids*.

روش *K-means*

این الگوریتم پارامتر k را به‌عنوان ورودی گرفته و مجموعهٔ n شیء را به k خوشه افراز می‌کند. به‌طوریکه سطح شباهت داخلی خوشه‌ها بالا بوده و سطح شباهت اشیاء بیرون

---

[1]- Partitional Clustering

۲- توجه کنید این تابع اغلب عدم تشابه یا فاصله را نشان می‌دهد اما آن را معیار شباهت می‌نامیم.

خوشه‌ها پایین باشد. شباهت هر خوشه نسبت به متوسط اشیاء آن خوشه سنجیده شده که این متوسط، مرکز خوشه نیز نامیده می‌شود. این الگوریتم به‌صورت زیر کار می‌کند:

ورودی: k، تعداد خوشه‌ها و یک پایگاه داده شامل n شیء

خروجی: یک مجموعه از k خوشه که معیار مربع خطا را حداقل می‌کند.

الگوریتم:

- به‌صورت دلخواه (تصادفی) k شیء را به‌عنوان مراکز خوشه‌های ابتدایی انتخاب کن.
- تکرار کن.
- هر شی را با توجه به بیشترین شباهت آن به مراکز خوشه‌ها، به خوشه‌ها تخصیص بده.
- مراکز خوشه‌ها را به‌روز کن به این معنی که برای هر خوشه میانگین اشیاء آن خوشه را محاسبه کن.
- تا هنگامی که هیچ تغییری در خوشه‌ها رخ ندهد.

معیار مربع خطا که در فوق ذکر شد، عبارتست از:

$$E = \sum\sum |p - m_i|^2 \qquad (۱۳-۳)$$

در این رابطه E مجموع مربع خطا برای تمام اشیاء پایگاه داده می‌باشد. p نقطه‌ای در فضاست که نمایانگر یک شیء می‌باشد، و m_i میانگین خوشهٔ C_i می‌باشد که نقطه p به آن متعلق است. (هم p و هم m_i چند بعدی هستند).

این الگوریتم هنگامی‌که خوشه‌ها به‌صورت ابرهای فشرده هستند و این ابرها نیز خودشان از یکدیگر مجزا هستند، به خوبی کار می‌کنند. این روش برای پایگاههای دادهٔ بزرگ کارا و مقیاس‌پذیر می‌باشد، زیرا پیچیدگی محاسباتی آن عبارتست از $O(tkn)$ که: n تعداد کل اشیاء، K تعداد خوشه‌ها و t تعداد تکرارهای الگوریتم است. این روش اغلب به یک بهینهٔ محلی[1] ختم می‌شود نه یک بهینهٔ سراسری[1].

---

[1]- Local Optimum

روش *K-means* تنها هنگامی کاربرد دارد که بتوان مراکز خوشه‌ها را تعریف نمود. مثلاً برای داده‌هایی با ویژگیهای طبقه‌ای این روش کارا نیست. از معایب این روش تعیین K است که می‌بایست کاربر ابتدا آنرا معین کند و راه خاصی برای تعیین آن مشخص نشده است. یک راه امتحان kهای مختلف و بررسی معیار مربع خطا برای هر k می‌باشد. همچنین این روش برای کشف خوشه‌هایی با شکلهای پیچیده مناسب نیست. یکی از مهمترین نقاط ضعف این روش این است که در برابر اغتشاشات و نقاط پرت حساس است زیرا این داده‌ها به راحتی مراکز را تغییر می‌دهند و ممکن است نتایج مطلوبی حاصل نشود.

مثال: به فرض مجموعهٔ {۲,٤,۱۰,۱۲,۳,۲۰,۳۰,۱۱,۲۵}را می‌خواهیم به $k=۲$ خوشه افراز کنیم. با استفاده از روش *K-means* مراحل زیر را طی می‌کنیم:

به‌طور تصادفی دو مرکز $m_۲=٤$ و $m_۱=۲$ را انتخاب می‌کنیم و بقیهٔ اعضاء مجموعه را نسبت به این دو مرکز تخصیص می‌دهیم به این صورت که به هر یک از دو مرکز که نزدیک‌تر بودند، به همان خوشه تعلق می‌گیرند. خوشه‌های حاصل عبارتند از:

$K_۱=\{۲,۳\}$ $K_۲=\{٤,۱۰,۱۲,۲۰,۳۰,۱۱,۲۵\}$

حال مراکز جدید را محاسبه می‌کنیم و تخصیص را نسبت به مراکز جدید انجام می‌دهیم. (مراکز در این مثال میانگین اعداد هر دسته می‌باشد):

$m_۱=۲/۵$, $m_۲=۱٦$

خوشه‌های جدید عبارتند از:

$K_۱=\{۲,۳,٤\}$, $K_۲=\{۱۰,۱۲,۲۰,۳۰,۱۱,۲۵\}$

روند فوق را آنقدر تکرار می‌کنیم تا اینکه دیگر تغییری در خوشه‌ها رخ ندهد:

$m_۱=۳$, $m_۲=۱۸$

$K_۱=\{۲,۳,٤,۱۰\}$, $K_۲=\{۱۲,۲۰,۳۰,۱۱,۲۵\}$

---

[1]- Global Optimum

$m_۱ = ٤.٧٥$, $m_۲ = ١٩.٦$

$K_۱ = \{۲,۳,٤,۱۰,۱۱,۱۲\}$, $K_۲ = \{۲۰,۳۰,۲٥\}$

$m_۱ = ۷$, $m_۲ = ۲٥$

$K_۱ = (۲,۳,٤,۱۰,۱۱,۱۲\}$, $K_۲ = \{۲۰,۳۰,۲٥\}$

در این مرحله دیگر تغییری در خوشه‌ها رخ نمی‌دهد. لذا دو خوشهٔ فوق به‌دست آمده است والگوریتم خاتمه می‌یابد.

مثال: ۷ نوع غذا (۷ شیء) با توجه به دو صفت محتوی پروتین (P) ومحتوی چربی (F) در جدول زیر آورده شده اند:

جدول ۳ – ٥) ویژگیهای متفاوت غذاها

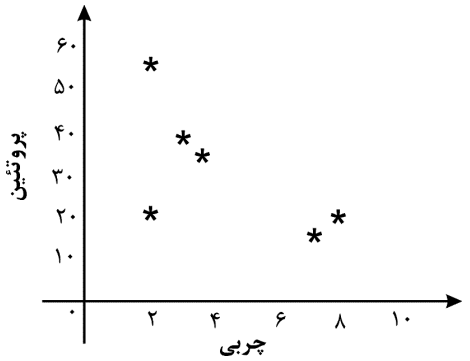
شماره غذا	میزان پروتئین	میزان چربی
۱	۱/۱	٦۰
۲	۸/۲	۲۰
۳	٤/۲	۳٥
٤	۱/٥	۲۱
٥	۷/٦	۱٥
٦	۲/۰	٥٥
۷	۳/۹	۳۹

اگر روش*K-means* را با $k = ٤$ شروع کنیم به‌طوری‌که $m_۳ = ۳$ و $m_۲ = ۲$ و $m_۱ = ۱$ باشند (در اینجا m و k شماره رکورد هستند)، آنگاه:

$K_۱ = \{۱,٦\}$, $K_۲ = \{۲,٥\}$

$K_۳ = \{۳,۷\}$, $K_٤ = \{٤\}$

به‌دست می‌آید که در صورت ادامه دادن روش تغییری در خوشه‌ها حاصل نمی‌شود و لذا خوشه‌های فوق بهینه هستند. شکل زیر گویای این مطلب است.



شکل ۳-۵) نمودار غذاها بر اساس چربی و پروتئین

برای رفع اشکالات روش *K-means* تغییراتی روی آن ایجاد شده است. این روشهای توسعه‌یافته در انتخاب k مرکز اولیه، محاسبهٔ عدم شباهت و استراتژیهای محاسبهٔ مراکز خوشه‌ها بایکدیگر متفاوتند. یکی از این تغییرات این است که ابتدا روی پایگاه داده، الگوریتم تجمیع سلسله مراتبی (که بعداً توضیح داده خواهد شد) اجرا می‌شود تا تعداد خوشه‌های مطلوب را پیدا کرده و سپس از خوشه‌های به‌دست آمده، به‌عنوان مرحلهٔ اول روش *K-means* استفاده می‌شود.

یکی دیگر از روشهای مشابه *K-means* روش *K-modes* می‌باشد. در این جا روش *K-means* را به منظور استفاده از داده‌های طبقه‌ای توسعه می‌دهد و به جای استفاده از مراکز خوشه‌ها از مُدهای خوشه‌ها استفاده می‌کند. لذا از یک رابطه اندازه‌گیری عدم شباهت جدید برای داده‌های طبقه‌ای استفاده می‌کند. برای محاسبه مدها نیز از یک روش مبتنی بر فراوانی استفاده می‌شود و می‌تواند برای داده‌های طبقه‌ای نیز به‌کار رود، و گاه از ترکیب دو روش*K-means* و*K-modes* برای داده‌های ترکیبی طبقه‌ای و عددی استفاده می‌شود. اگر به جای مرکز یا وسط یک خوشه، از میانهٔ آن خوشه استفاده کنیم، آنگاه روش نسبت به داده‌های دور از مرکزحساس نمی‌شود زیرا میانه از مقادیر بزرگ تأثیر نمی‌پذیرد. مثلاً:

- متوسط ۱ و ۳ و ۵ و ۷ و ۹ می‌شود ۵.
- متوسط ۱ و ۳ و ۵ و ۷ و ۱۰۰۹ می‌شود ۲۰۵.
- میانه ۱ و ۳ و ۵ و ۷ و ۱۰۰۹ می‌شود ۵.

برای بهبود بعضی از این ایرادات روشی دیگرکه مبتنی بر خود اشیاء می‌باشد و نمایندهٔ خوشه‌ها را از میان اشیاء پایگاه داده‌ها انتخاب می‌کند نه مرکز خوشه‌ها، عنوان می‌شود.

روش *K-medoids*

در این روش به جای استفاده از مرکز یک خوشه به‌عنوان مرجع، می‌توان از *medoid*ها استفاده کرد. یعنی اشیایی که در مرکزی‌ترین محل یک خوشه می‌باشد، بنابراین روش افراز هنوز می‌تواند مبتنی بر اصل حداقل‌سازی مجموع عدم شباهتها میان هر شیء و شیء مرجع متناظرش شکل بگیرد. این روش *K-medoids* نام دارد. استراتژی اساسی الگوریتم خوشه‌بندی *K-medoids* پیدا کردن k شیء نمایندهٔ آغازین (*medoid*) به‌طور دلخواه از n شیء پایگاه داده می‌باشد. هر شیء باقیمانده با *medoid*ای هم خوشه می‌شود که بیشترین شباهت را به آن داشته باشد. سپس این استراتژی مکرراً یکی از اشیاء *medoid* را با یکی از اشیاء غیر *medoid* جایگزین می‌کند به‌طوری‌که کیفیت نتیجهٔ خوشه‌بندی بهبود یابد. این کیفیت با به‌کارگیری تابع هزینه تخمین زده می‌شود که میانگین عدم تشابه بین یک شیء و *medoid* آن خوشه را اندازه‌گیری می‌کند. در اینجا ابتدا الگوریتم و سپس چگونگی تشکیل این تابع هزینه را بیان می‌کنیم:

ورودی: k تعداد خوشه‌ها و پایگاه داده‌ها شامل n شیء

خروجی: یک مجموعه از خوشه‌ها که مجموع عدم تشابه بین تمام اشیاء و نزدیک‌ترین *medoid* آنها را حداقل می‌کند.

الگوریتم:

- K شیء به‌عنوان *medoid*های اولیه به‌صورت دلخواه اختیار کن.
- تکرار کن تا اینکه هیچ تغییری رخ ندهد.

- هر کدام از اشیاء باقیمانده را به خوشه‌ای با نزدیک‌ترین *medoid* تخصیص بده.
- به‌طور تصادفی یک شی غیر *medoid* را انتخاب کن، O_{random}
- هزینهٔ نهایی s را از عوض کردن O_j (*medoid* آن خوشه) و O_{random} محاسبه کن.
- اگر $s<0$ آنگاه جای O_j و O_{random} را عوض کن تا مجموعه K تا *medoid* جدید شکل بگیرد.

برای اندازه‌گیری اینکه شیء O' بهتر از O به‌عنوان یک *medoid* هست یا خیر، کافیست حاصل رابطه زیر را به‌دست آوریم. اگر $E(o')-E(o)<0$ آنگاه جابه‌جایی O' با O مفید است.

$$E=\sum_{i=1}^{k}\sum_{p\in c_i} d(p,o_i)^2 \qquad (۱۴-۳)$$

در این رابطه E در اصل میزان کل فاصله‌ها از هر نقطه را نشان داده و s میزان هزینه تعویض می‌باشد که منفی بودن آن بهتر است و برابر سود در نظر گرفته می‌شود.

می‌توان این روش را به تنهایی در هر خوشه به‌کار برد و یا در ادامه با بررسی هزینه نهایی این نقل و انتقال، به سوی نقطه بهینه حرکت کرد. در این روش نقاط مختلف به عنوان جایگزینهایی برای مراکز انتخاب شده و هزینه‌ها محاسبه می‌شوند. هدف در این روش کاهش E است.

تابع هزینهٔ نهایی که برابر مجموع توابع هزینه همهٔ اشیاء می‌باشد، در هر تکرار از قواعد زیر پیروی می‌کند:

به فرض O_{random} یک جایگزین خوب برای O_j که یک *medoid* است، باشد. چهار حالت برای هر شیء غیر *medoid* مانند P رخ می‌دهد:

حالت۱: در این حالت P به O_j تعلق دارد. اگر O_j با O_{random} به‌عنوان *medoid* عوض شود و P به یکی از *medoid*های O_i که $i\neq j$ نزدیک‌تر باشد، آنگاه P به O_i تعلق می‌گیرد و از تفاضل فاصله فعلی و فاصله قبلی داریم:

$$C_p = d(P,O_i) - d(P,O_j) \qquad (۱۵-۳)$$

حالت۲: در این حالت P به O_j تعلق دارد. اگر O_j با O_{random} به‌عنوان *medoid* عوض شود و P به O_{random} نزدیک‌تر باشد، آنگاه P به O_{random} تعلق می‌گیرد و داریم:

$$C_p = d(P,O_{random}) - d(P,O_j) \qquad (۱۶-۳)$$

حالت۳: در این حالت P به O_i و $i \neq j$ تعلق دارد. اگر O_j با O_{random} به‌عنوان *medoid* عوض شود و P هنوز به O_i نزدیک‌تر باشد، آنگاه در تخصیص تغییری صورت نمی‌گیرد و داریم:

$$C_p = d(P,O_i) - d(P,O_i) = ۰ \qquad (۱۷-۳)$$

حالت٤: در این حالت P به O_i و $i \neq j$ تعلق دارد. اگر O_j با O_{random} به‌عنوان *medoid* عوض شود و P به O_{random} نزدیک‌تر باشد، آنگاه P به O_{random} تعلق می‌گیرد، یعنی $C_p = d(P,O_{random}) - d(P,O_i)$ در نهایت هزینه کل T_c از مجموع C_pها به‌دست می‌آید.

$$T_C = \sum C_p \qquad (۱۸-۳)$$

مثال: فرض کنید مجموعه نقاط {۲۰,۱۷,۱۵,۱۰,۸,۷,٦,۲,۱}را می‌خواهیم به ۳ خوشه تقسیم کنیم. اگر در مرحلهٔ اول تصادفاً ٦ و ۷ و ۸ به‌عنوان *medoid* انتخاب شوند و تخصیص را انجام دهیم، آنگاه:

خوشهٔ ۱ = ۷

خوشهٔ ۲ = ۸–۱۰ ،۱۵ ،۱۷ ،۲۰

خوشهٔ ۳ = ٦–۱ ،۲

نقطهٔ غیر*medoid* ۱۵ راجایگزین ۷ کرده و هزینه‌ها و هزینهٔ کل (T_c) را محاسبه می‌کنیم:

خوشهٔ ۱ = ٦−۱(cos۰) , ۲(cos۰) , ۷(۱−۰=۱)

خوشه ۲ = ۸−۱۰ *(cos۰)*

خوشه جدید $= ۱۵-۱۷(cost۲-۹=-۷)\ ,\ ۲۰(cost۵-۱۲=-۷)$

پس در نهایت: $T_c = -۷-۷+۱=-۱۳$

بنابراین جایگزینی ۱۵ به جای ۷ خوشه‌بندی را بهبود می‌بخشد و خطا را کم می‌کند. لذا:

خوشهٔ ۱ $= ۶-۱, ۲, ۷$

خوشه ۲ $= ۸-۱۰$

خوشهٔ ۳ $= ۱۵-۱۷, ۲۰$

اگر به‌طور تصادفی ۱ را به جای ٦ جایگزین کنیم، هزینه‌ها به‌صورت زیر می‌باشد:

خوشه ۱ $= ۸-۶(cost۲-۰=۲)\ ,\ ۷(cost۱-۱=۰)\ ,\ ۱۰(cost۰)$

خوشه ۲ $= ۱۵-۱۷(cost۰)\ ,\ ۲۰(cost۰)$

خوشه جدید $= ۱-۲(cost۱-۴=-۳)$ $\qquad T_c = -۱$

پس ۱ به جای ٦ جایگزین می‌شود و داریم:

خوشه ۱ $= ۱-۲$

خوشه ۲ $= \ ۸-۶, ۷, ۱۰$

خوشه ۳ $= ۱۵-۱۷, ۲۰$

اگر به‌طور تصادفی ۱۰ به جای ۸ جایگزین شود آنگاه:

خوشه ۱ $= ۱-۲(\cos t۰)$

$T_c = ۲$

خوشه ۲ $= ۱۵-۱۷(cost۰)\ ,\ ۲۰(cost)$

خوشه جدید $= ۱۰-۶(cost۰)\ ,\ ۷(cost۰)\ ,\ ۸(cost۲-۰=۲)$

پس ۱۰ را به جای ۸ جایگزین نمی‌کنیم. حال اگر به‌طور تصادفی ۱۷ و ۱۵ عوض شوند:

خوشه ۱ $= ۱-۲(cost۰)$

خوشه ۲ $= ۸-۶(cost۰)\ ,\ ۷(cost۰)\ ,\ ۱۰(cost۰)$

خوشه جدید $= ۱۷-۱۵(cost۲-۰=۲)\ ,\ ۲۰(cost۳-۵=-۲)$

لذا جابه جایی صورت نمی‌پذیرد. اگر به‌صورت تصادفی ۲۰ با ۱۵ و ۱ با ۱۵ و... جابجا شوند هیچ تغییری در خوشه‌ها حاصل نمی‌شود. لذا خوشه‌های نهایی عبارتند از:

خوشه ۱ $= ۱-۲$

خوشه ۲ $= ۸-۶,۷,۱۰$

خوشه ۳ $= ۱۵-۱۷,۲۰$

در روش *K-medoids* هر بارکه یک جابه‌جایی رخ می‌دهد، تغییری در خطای مربع E حاصل می‌شود که ناشی از همان تابع هزینه می‌باشد. بنابراین در صورتی که *medoid* فعلی با شیء غیر *medoid* جابه‌جا شود، تابع هزینه، تفاوت در خطای مربع را محاسبه می‌کند.

روش ذکر شده *PAM*[1] نام دارد که یکی از اولین الگوریتمهای *K-medoids* است و تلاش می‌کند k افراز برای n شیء تعیین کند. بعد از انتخاب تصادفی k تا *medoid* الگوریتم مکرراً سعی می‌کند انتخاب *medoid*ها را بهتر کند. همهٔ زوجهای ممکن از اشیاء که یکی *medoid* و دیگری غیر *medoid* است را تحلیل می‌کند. یک شیء O_j با شیءای جابه جا می‌شود که بیشترین کاهش را در خطای مربع داشته باشد. لذا این روش برای پایگاه داده‌های بزرگ مشکل است. برای رفع این اشکال از الگوریتمهای *CLARA , CLARANS* استفاده می‌شود.

روش *CLARA*[2]

این روش توسط روسیو[3] و کافمن[4] در سال ۱۹۹۰ ارائه شده و برای پایگاه داده‌های بزرگ به‌کار می‌رود. به این ترتیب‌که چندین نمونهٔ تصادفی از این پایگاه داده بر می‌دارد

---

[1]- Partitioning Around Medoids
[2]- Clustering Large Application
[3]- Rousseeuw
[4]- Kaufmann

و الگوریتم PAM را روی هر نمونه اجرا کرده و آن نمونه را خوشه‌بندی می‌کند. سپس عناصر باقیماندهٔ پایگاه داده را به نزدیک‌ترین خوشه تخصیص می‌دهد. تعداد اعضای هر نمونه نسبت به پایگاه داده خیلی کوچکتر است. جواب آخر این روش گاه قابل ارزیابی نیست زیرا نمونه‌ها به‌طور تصادفی انتخاب می‌شوند. پیچیدگی محاسبات این روش درهرتکرار متناظر با $O(ks^2 + k(n-k))$ می‌باشد که k تعداد خوشه‌ها، s تعداد شی‌های نمونه و n کل اشیاء می‌باشد. لذا پیچیدگی الگوریتم کاهش می‌یابد و از مرتبه اشیاء نمونه است.

روش خوشه‌بندی سلسله مراتبی

این روش با گروه‌بندی اشیاء به‌صورت یک درخت کار می‌کند و معمولاً به دو صورت پایین به بالا (تجمیعی) یا با بالا به پایین (تقسیمی) پیاده‌سازی می‌شود. این دو روش را می‌توان به‌صورت‌های زیر بیان کرد:

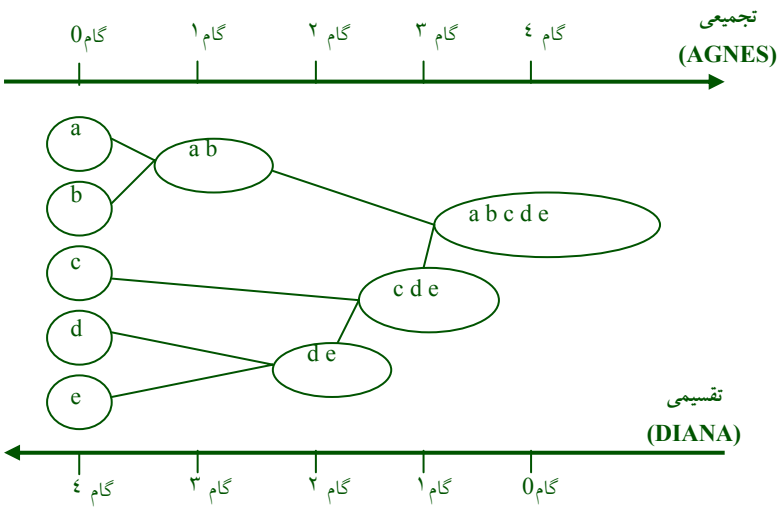
- تجمیعی[1]: در این روش خوشه‌ها مکرراً با هم ترکیب می‌شوند. به این صورت که ابتدا هر یک از اشیاء یک خوشه در نظر می‌گیرد و سپس با ترکیب کردن این خوشه‌ها به خوشه‌های بزرگ و بزرگتر تبدیل می‌کند تا اینکه همهٔ اشیاء در یک خوشه قرارگیرند و یا به شرط پایان برسد.
- تجزیه‌ای یا تقسیمی[2]: در این روش خوشه‌ها مکرراً تقسیم می‌شوند. این روش دقیقاً بر عکس روش تجمیعی عمل می‌کند به این صورت‌که ابتدا همهٔ اشیاء در یک خوشه قرار دارند و الگوریتم این خوشه را به خوشه‌های کوچک و کوچکتر تجزیه می‌کند تا اینکه هرشیء در یک خوشه قرارگیرد. این روش معمولاً مناسب نیست و خیلی کم مورد استفاده قرار می‌گیرد زیرا پیچیدگی محاسباتش بالاست. توجه کنید

---

[1]- Agglomerative
[2]- Divisive

که هر خوشه را به چندین حالت متفاوت می‌توان به خوشه‌های کوچکتر تقسیم کرد که باید بهترین حالت آن انتخاب شود.

حال به تفسیر دو الگوریتم *AGNES* در روش ترکیبی و *DIANA* در روش تقسیمی می‌پردازیم. به شکل (۳-٦) دقت کنید:



شکل ۳- ٦) نمودار دو روش تجمیعی و تقسیمی

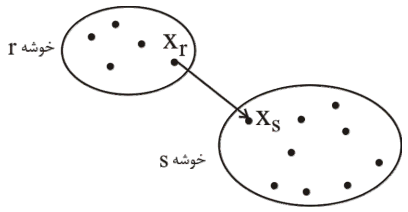
درالگوریتم *AGNES*[1]، ابتدا هر شی در داخل یک خوشه قرار می‌گیرد. مثلاً در شکل فوق ۵ خوشه برای مجموعهٔ $\{a,b,c,d,e\}$ به‌وجود می‌آیند. سپس خوشه‌ها گام به گام بر اساس برخی معیارها ترکیب می‌شوند. اگر فاصله بین اشیاء هر خوشه با اشیاء خوشه دیگر را حساب کنیم و دو شیء متعلق به دو خوشه، کمترین فاصله را داشته باشند، آن دو خوشه با هم ترکیب می‌شوند. این روش پیوند تکی[2] نام دارد که شباهت بین دو خوشه را با شباهت نزدیک‌ترین نقاط متعلق به خوشه‌های مختلف نمایش می‌دهد. لذا می‌بایست در هر تکرار از الگوریتم تمام فاصله‌های بین زوجهای موجود در خوشه‌های

[1]- Agglomerative Nesting
[2]- Single Link

مختلف محاسبه شود تا حداقل فاصله یک زوج به‌دست آید. این فاصله‌ها را می‌توان در یک ماتریس به نام ماتریس عدم شباهت قرار دارد. ترکیب خوشه‌ها آنقدر ادامه می‌یابد تا نهایتاً همهٔ اشیاء درون یک خوشه قرارگیرند. معیارهای گوناگونی که در روشهای سلسله مراتبی برای فاصلهٔ بین خوشه‌ها به‌کار می‌روند، عبارتند از:

- پیوند تنها (تکی): فاصلهٔ بین خوشه‌ها بر حسب حداقل فاصلهٔ ممکنه بین عناصر آنها محاسبه می‌شود. در این حالت باید کلیه فاصله‌ها بین زوج عناصر دو خوشه را محاسبه و از طریق حداقل آنها، فاصله بین دو خوشه را معین کرد. مثلاً فاصله بین دو خوشه r,s به‌صورت زیر حساب می‌شود:

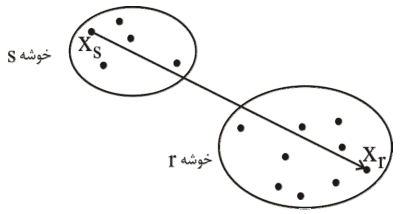
$$d(r,s) = \min(dist(x_{ri}, x_{sj})) \qquad (۱۹-۳)$$



شکل ۳- ۷) روش حداقل فاصله

- پیوند کامل[1]: در این حالت فاصلهٔ بین خوشه‌ها بر حسب دورترین فاصلهٔ ممکنه بین عناصر آنها محاسبه می‌شود:

$$d(r,s) = max(dist(x_{ri}, x_{sj})) \qquad (۲۰-۳)$$

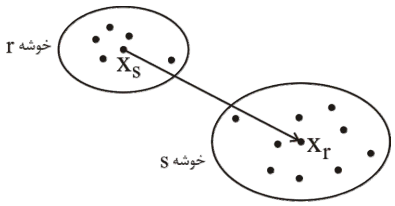


شکل ۳- ۸) روش حداکثر فاصله

[1]- Complete Link

- پیوند متوسط[1]: فاصله دو خوشه مساوی مقادیر متوسط کلیهٔ فاصله‌های ممکنه بین عناصر دو خوشه است:

$$d(r,s) = \frac{1}{n \times n} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} dist(x_{ri}, x_{sj}) \qquad (۲۱-۳)$$



شکل ۳- ۹) روش مرکز ثقل

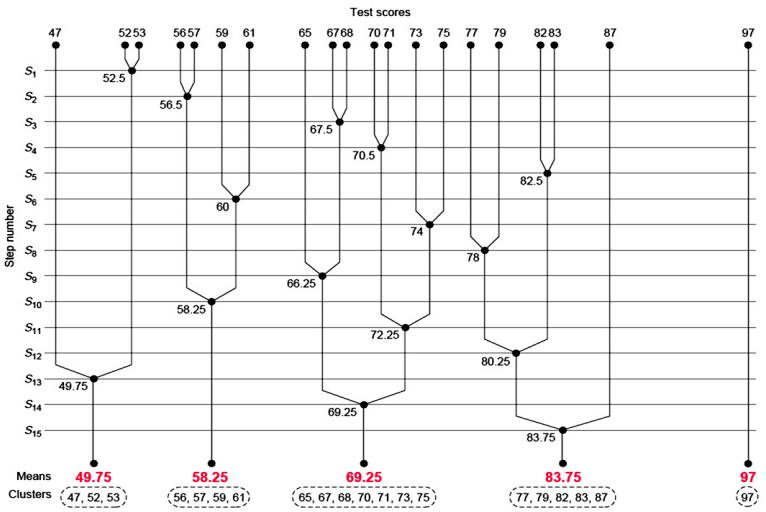
- مرکز ثقل[2]: فاصلهٔ بین دو خوشه بر اساس فاصلهٔ بین مراکز دو خوشه محاسبه می‌شود.

در الگوریتمهای سلسله مراتبی خوشه‌بندی، کاربر می‌تواند تعداد خوشه‌ها را انتخاب کند و از آن برای شرط پایان الگوریتم استفاده کند. اکنون یک مثال را به روش *AGNES* حل می‌کنیم که در آن برای اندازه‌گیری فاصلهٔ بین خوشه‌ها از معیار مرکز ثقل استفاده می‌شود. شرط پایان الگوریتم رسیدن به ۵ خوشه می‌باشد. مجموعهٔ داده‌هایی که باید خوشه‌بندی شوند، عبارتست از:

{۴۷, ۵۲,۵۳,۵۶,۵۷,۵۹,۶۱,۶۵,۶۷,۶۸,۷۰,۷۱,۷۳,۷۵,۷۷,۷۹,۸۲,۸۳,۸۷,۹۷}

این خوشه‌بندی در شکل (۳-۱۰) آمده است.

[1]- Average Link
[2]- Centroid



شکل ۳-۱۰) روش AGNES

به نمودار درختی تشکیل شده در فوق نمودار دندانه‌ای[1] گفته می‌شود. در قدم اول نزدیک‌ترین نقاط موجود در خوشه‌ها، ۵۳، ۵۲ یا ۵۷، ۵۶ یا ۶۸، ۶۷ یا ۷۱، ۷۰ یا ۸۳، ۸۲ می‌باشند که می‌بایست در هرگام فقط یک کاندید انتخاب شود. به فرض ۵۳، ۵۲ با هم ترکیب شوند وخوشهٔ {۵۲,۵۳} با مرکز ۵۲/۵ تشکیل گردد. لذا به جای دو نقطهٔ ۵۳، ۵۲ نقطهٔ ۵۲/۵ جایگزین می‌شود. در قدم بعدی باز این فرآیند تکرار می‌شود (یافتن نزدیک‌ترین دو نقطه، محاسبهٔ میانگین و یکی‌کردن نقاط) تا اینکه تمام نقاط در یک خوشه قرارگیرند. برای مثال ۱۵ تکرار لازم است تا ۲۰ نقطهٔ پایگاه داده در ۵ خوشه جای گیرند.

الگوریتم *DIANA*[2]

این الگوریتم عکس الگوریتم *AGNES* عمل‌می‌کند. به این صورت که ابتدا همه اشیاء را درون یک خوشه قرار می‌دهد و این خوشه‌ها را تقسیم می‌کند تا اینکه نهایتاً هر

[1]- Dendogram
[2]- Divisive Analysis

شیء در یک خوشه قرار گیرد. برای بیان چگونگی تجزیهٔ خوشه‌ها فرض کنیم الگوریتم به k خوشه در حال حاضر رسیده است. ابتدا در هر خوشه بزرگترین فاصلهٔ ممکنه بین اشیاء آن خوشه را پیدا کرده سپس از این K خوشه، خوشه‌ای برای تقسیم انتخاب می‌شود که بزرگترین فاصله‌اش، از همهٔ بزرگترین فاصلهٔ خوشه‌های دیگر، بزرگتر باشد. بعد از انتخاب خوشهٔ مناسب برای تجزیه، مرکز این خوشه را می‌یابیم و آنرا M می‌نامیم. سپس فاصلهٔ تک‌تک اعضای این خوشه را نسبت به M به‌دست می‌آوریم و آنها را در مجموعهٔ M_1 قرار می‌دهیم. بعد برای هر دو عضو خوشه، مرکز آن دو را به‌دست آورده و فاصله مرکز آنها را از M به‌دست می‌آوریم و آنها را در مجموعه M_2 قرار می‌دهیم. همین کار را برای هر سه عضو، چهار عضو و بالاخره $n-1$ عضو (n تعداد اعضای خوشه) از خوشه انجام می‌دهیم و فاصله‌های به‌دست آمده را در مجموعه‌های قرار می‌دهیم. بدیهی است تعداد اعضای $M_1 = \binom{n}{1}, \ldots,$ تعداد اعضای $M_{n-1} = \binom{n}{n-1}$ است. با استفاده از مجموعه‌ای شامل همه اعضای مجموعه‌های $M_1, M_2, \ldots, M_{n-1}$ بزرگترین عضو موجود را محاسبه می‌کنیم. به فرض این عضو از M_k باشد، در این صورت خوشهٔ n تایی مورد نظر به یک خوشهٔ k تایی (اعضای مربوط به بزرگترین مقدار) و یک خوشهٔ n-k تایی تجزیه می‌شود. پیچیدگی الگوریتم *DIANA* خیلی زیاد است، یعنی $O(2^{n-1}-1)$ و معمولاًمقرون به صرفه نیست، لذا بیشتر اوقات از *AGNES* استفاده می‌شود.

مقایسهٔ خوشه‌بندی سلسله مراتبی و غیر سلسله مراتبی

روشهای خوشه‌بندی غیرسلسله‌مراتبی معمولاً سریعتر عمل می‌کنند ولی نیاز به یکسری تصمیم‌گیری از طرف تحلیل‌گر و استفاده کننده دارند. از جملهٔ این تصمیمها انتخاب تعدادخوشه‌ها یا انتخاب حداقل نزدیکی برای قرارگرفتن دو عنصر در یک خوشه می‌باشد. در این گونه روشها معمولاً یک سری خوشه‌های اولیه ایجاد شده و سپس در مراحل بعدی بهبود داده می‌شوند. از آنجایی‌که این روشها به تعداد خوشه‌های اولیه و

انتخاب مناسب آنها حساسیت دارند. برخی اوقات به کمک روش سلسله مراتبی، تعداد مناسب خوشه را تخمین زده و سپس از افرازبندی استفاده می‌کنند. ایراد روش سلسله‌مراتبی این است که تخصیص انجام‌شده در یک مرحله، قابل تغییر در مراحل بعد نیست که این امر ممکن است به تصمیمات برگشت‌ناپذیر و نامناسب منجر شود.

تعیین تعداد خوشه‌ها

اگر تمام متغیرها کاملاً مستقل باشند، هیچ خوشه‌ای ایجاد نمی‌شود. (تمام فضا به‌صورت تصادفی با نقاط داده پر می‌شود) بر عکس اگر تمام متغیرها وابسته باشند، آنگاه تمام داده‌ها تشکیل یک خوشه می‌دهند. در شرایط بین استقلال و وابستگی کامل ما نمی‌دانیم که واقعاً چند خوشه وجود دارد. معمولاً در انتخاب مقدار k، نقش تحلیل‌گر بسیار بیشتر از رایانه می‌باشد. برای همین با توجه به کاربردهای متفاوت روشهای خوشه‌بندی، ممکن است به تعداد بیشتر یا کمتری از خوشه‌ها نیاز باشد. در بسیاری از موارد با یک مقدار K خوشه‌بندی را انجام داده و نتایج را بررسی می‌کند و دوباره به سراغ یک K دیگر می‌روند. بعد از انجام هربار این کار، قدرت و ارزش نتایج را به‌وسیلهٔ اندازه‌گیری میزان متوسط فواصل در داخل خوشه‌ها و میزان متوسط فواصل بین مراکز خوشه‌ها و یا روشهای دیگر بررسی می‌کنند. باید به این نکته توجه داشت که گاه خوشه‌ها به وسیلهٔ قضاوتهای ذهنی تحلیل‌گر هم مورد ارزیابی قرار می‌گیرند تا ارزش آنها در موارد خاصی مورد بررسی قرارگیرند.

مزیت خوشه‌بندی سلسله‌مراتبی این است که به تحلیل‌گر اجازه می‌دهد که از بین حالات مختلف، یک عدد برای تعداد خوشه‌ها انتخاب نماید. معیارهایی برای ارزیابی دسته‌های تشکیل شده و همچنین تعیین k مناسب، وجود دارد. [۱]

روشهای مبتنی بر چگالی

همان‌طورکه در روشهای قبل به خصوص روشهای افرازی مشاهده شد، خوشه‌های حاصل از این روشها اغلب دارای شکلهایی متقارن در فضای مسئله بودند. بدین صورت که اغلب حول یک مرکزیت (مثلاً میانگین متغیرهای درون خوشه و یا عنصری که به عنوان مرکزیت آن خوشه انتخاب شده بود یعنی ***medoid***) شکل دایره‌ای، کروی و... را تشکیل می‌دادند. گاه ممکن است بنا به ماهیت مسئله به دنبال خوشه‌هایی با الگوهایی پیچیده‌تر باشیم و یا اینکه رابطه‌ای خاص بین ابعاد مختلف داده‌ها و متغیرها وجود داشته باشد و به دنبال یافتن عناصری باشیم که چنین خصوصیتی را دارند. در این حالت از روشهای مبتنی بر چگالی استفاده می‌کنیم. ایده اصلی این روشها بر این اساس است که ابتدا به دنبال نقاطی می‌گردیم که چگالی حول آنها زیاد باشد سپس سعی می‌کنیم به گونه‌ای نقاطی را که با این مراکز تجمع در ارتباط هستند، پیدا کنیم. گاه پس از طی چند مرحله دو یا چند مرکز تجمع به یکدیگر متصل شده و یک خوشه را شکل می‌دهند. این روشها همچنین در حذف داده‌های پرت و مغشوش بسیار مفید هستند.

روشی که در اینجا بیان می‌شود ***DBSCAN***[1] (یا خوشه‌بندی فضایی بر پایه چگالی داده‌های مغشوش) نام دارد که از متداول ترین روشهای مبتنی بر چگالی است.

در این روش ابتدا برای تمامی نقاط یک شعاع فرضی در نظر می‌گیریم و تعداد نقاطی که اطراف این شعاع فرضی (مثلاً ε) قرار دارند را مشخص می‌کنیم. سپس کاربر باید تعداد نقاط[2] حداقل را برای شروع کار الگوریتم تعریف کند. چگالی توزیع داده‌ها در اطراف این نقاط زیاد است. حال اجازه دهید اصطلاحات زیر را برای ادامه کار تعریف

---

[1]- Density- Based Spatial Clustering of Applications with Noise
[2]- Minpts

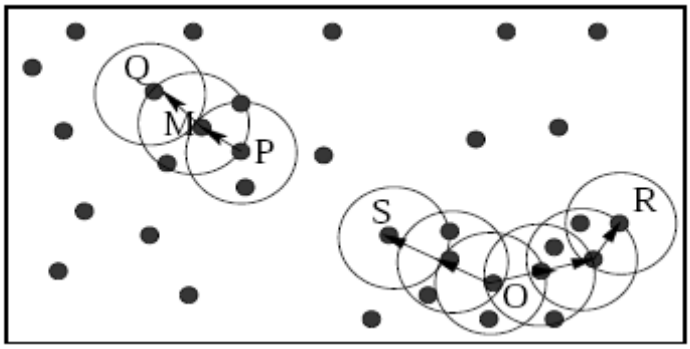
کنیم. نقطه p را از نقطه q مستقیماً قابل دسترس چگال[1] می‌نامیم اگر p در شعاع ε از q قرارگرفته باشد و در شعاع ε از q حداقل نقاط مورد نظر ما نیز وجود داشته باشد.

نقطه p را از نقطه q قابل دسترس چگال[2] می‌نامیم به‌طوری که با در نظر گرفتن حداقل نقاط، زنجیره‌ای از P_iها وجود داشته باشد که اولاً P_{i+1} از P_i مستقیماً دسترس پذیر بوده و ثانیاً $p = p_n$, $q = p_1$ باشند.

نقطه p به نقطه q متصل چگال[3] است به شرطی که با حفظ شرایط ε و حداقل نقاط، یک شیء مانند o وجود داشته باشد که هر دوی q, pاز نقطه o قابل دسترس چگال باشند.

حال با توجه به تعاریف بالا یک خوشه مبتنی بر چگالی را به‌صورت زیر تعریف می‌کنیم: یک خوشه مبتنی بر چگالی مجموعه‌ای از اشیاء (نقاط) متصل به یکدیگر از نظر چگالی است.

با توجه به این تعریف هر داده‌ای را که خارج از این خوشه‌ها باشد به‌عنوان داده پرت و اغتشاش در نظر گرفته می‌شود.



شکل ۳- ۱۱) روش مبتنی بر چگالی

[1]- Directly Density Reachable
[2]- Density Reachable
[3]- Density Connected

در شکل (۳-۱۱) با فرض حداقل نقاط برابر با ۳ و شعاع ٤، خوشه‌هایی مشخص شده است. نقطه M از نقطهP مستقیماً قابل دسترس چگال است و Q از نقطه p قابل دسترس چگال به‌طور غیرمستقیم است.P از Q قابل دسترسی نبوده اما R و S هر دو از O قابل دسترسی بوده و از نظر چگالی به یکدیگر متصل هستند. توجه داشته باشید که درست است که Q از نقطه M دسترسی پذیر مستقیم است اما عکس آن درست نیست.

در پیاده سازی، *DBSCAN* ابتدا نقاط مرکزی را مشخص کرده و هر کدام به‌عنوان یک خوشه در نظر گرفته می‌شوند. سپس نقاط قابل دسترس به آن اضافه می‌شوند و گاه خوشه‌ها را نیز با یکدیگر ادغام می‌کنند. این کار آنقدر تکرار می‌شود تا دیگر تغییری در خوشه‌ها ایجاد نشود یعنی هیچ عنصری به خوشه‌ها اضافه نشود.

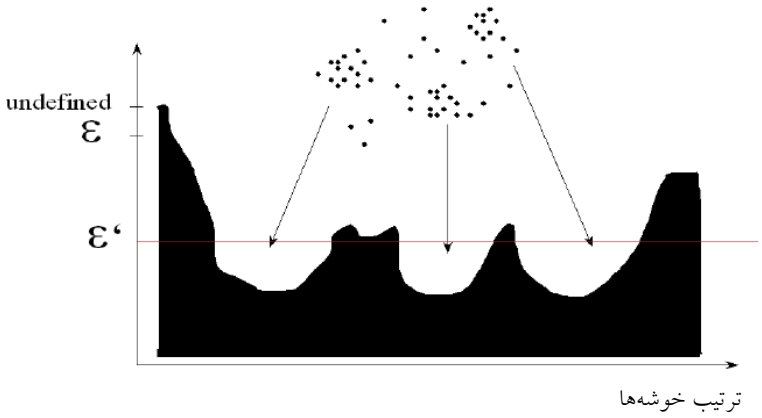
اصلی‌ترین مشکلی که در روش *DBSCAN* مشاهده می‌شود معین نبودن مقدار ε و همچنین حداقل نقاط است که کاربر باید آنها را تعیین کند. ممکن است در ابتدا این امر ساده به نظر بیاید اما پس از کمی دقت مشاهده می‌شود که تعیین این مقادیر مخصوصاً در پایگاه داده‌های بزرگ و زمانی که ابعاد مختلف پایگاه دارای ضرایب و مقیاسهای مختلفی هستند بسیار مشکل است. برای اصلاح این مشکلات روش دیگری به نام *OPTICS*[1] (یا مرتب‌سازی نقاط برای شناسایی ساختار خوشه‌بندی) ابداع شد.

با مطالعه روش *DBSCAN* مشخص می‌شود که برای یک مقدار ثابت حداقل نقاط، خوشه‌های مبتنی بر چگالی بالاتر (یعنی ε کوچکتر) کاملاً در داخل خوشه‌هایی مبتنی بر چگالی کمتر (یعنی ε بیشتری) قرارگرفته است. پس ترتیب انتخاب اشیاء باید به‌صورتی باشد که آن عنصری که برای عضویت خوشه به کمترین میزان ε نیاز دارد اول از همه مورد بررسی قرارگیرد. *OPTICS* روشی است که این ترتیب را مشخص می‌کند و برای این کار به محاسبه دو متغیر فاصله مرکزی[2] و فاصله دسترسی[3] نیاز دارد.

---

[1]- Ordering Points To Identify the Clustering Structure
[2]- Core- Distance
[3]- Reachability- Distance

فاصله مرکزی شیء p در واقع کوچکترین مقدار فاصله ε است بین p و یک شی در داخل همسایگی $\varepsilon(P_C)$ به‌طوری که p با این مقدار ε یک شی مرکزی شود. این فاصله حتما بزرگتر یا مساوی فاصله این دو نقطه است و بزرگی آن به میزانی است که حداقل تعداد نقاط مورد نظر ما را برای ایجاد یک نقطه مرکزی شامل شود. متغیر دیگری که تعریف می‌شود فاصله دسترسی است. فاصله دسترسی p با توجه به شی o کمترین فاصله‌ای است به‌طوری که p از o مستقیماً قابل دسترس چگال باشد. *OPTICS* این دو متغیر را برای همه عناصر پایگاه داده محاسبه می‌کند. چنانچه در شکل (۳-۱۲) مشاهده می‌کنید *OPTICS* یک ترتیب نیز برای خوشه‌ها ارائه می‌کند که با εها مختلف متفاوت است، اما با توجه به رعایت این نکته که خوشه‌های ساخته شده با چگالی کمتر خوشه‌های ساخته شده با چگالی بیشتر را شامل می‌شوند، می‌توان به سادگی مشاهده کرد که برای εها مختلف تعداد خوشه‌های مختلفی ایجاد می‌شود.



شکل ۳- ۱۲) ترتیب خوشه‌بندی در OPTICS

با شروع از کوچکترین ε و افزایش تدریجی آن می‌توان تعداد خوشه‌های مختلفی را ایجاد کرد. به این صورت که در ابتدا هیچ خوشه‌ای وجود ندارد و در نهایت همه به یک خوشه تبدیل می‌شوند. با این روش می‌توان ابزاری برای کمک به کاربر در انتخاب تعداد خوشه‌ها ایجاد نمود.

در ادامه به روش دیگری اشاره می‌شود که بر اساس تابع توزیع چگالی در فضا عمل می‌کند این روش خوشه‌بندی بر پایه چگالی یا به اختصار *DENCLUE*[1] نام دارد.
روش *DENCLUE* بر اساس سه ایده اصلی زیر استوار است:

- تأثیر هر داده‌ای بر فضا را می‌توان به‌طور رسمی با یک تابع ریاضی به نام تابع تأثیر[2] مدل کرد. این تابع می‌تواند توصیفی از اثر دادهٔ مورد بحث بر همسایگی خودش باشد.
- تأثیر کل داده‌ها بر فضا را می‌توان به‌صورت مدلی متأثر از تمام داده‌های آن فضا بیان نمود.
- خوشه‌ها را می‌توان به‌طور خودکار با شناسایی عوامل جاذب چگالی[3] در جاهایی که افزایش چگالی وجود دارد مشخص نمود.

اجازه دهید با یک مثال این ایده‌های اصلی را مشخص کنیم. فرض کنید هر داده یک لامپ نورانی در فضا باشد که اطراف خود را روشن می‌کند. روشنایی هر نقطه از فضا از مجموع روشنایی لامپهای اطراف مشخص می‌شود. حال در چنین فضایی نقاط و مناطق نورانی‌تر را به‌عنوان خوشه‌ها درنظر می‌گیریم.
تابع تأثیر، هر نقطه از فضای d بعدی (f^d) را به عددی حقیقی و مثبت نگاشت می‌کند این تابع را تابع اصلی یا پایه[4] تأثیر می‌نامند که این‌گونه تعریف می‌شود:

$$f_B^y : f^d \to R_+^+ \qquad (۲۲-۳)$$

این تابع می‌تواند هر شکل دلخواهی داشته باشد اما باید خصوصیات انعکاسی و تقارنی را دارا باشد. این تابع دو متغیر x, y دارد و خروجی آن تأثیر این دو نقطه را بر یکدیگر نشان می‌دهد.

$$f_B^y = f_B(x, y).$$

---

[1]- Density Based Clustering
[2]- Influence Function
[3]- Density Attractors
[4]- Basic Function

توابع معروفی که در اینجا استفاده می‌شوند عبارتند از:

- تابع اثر موج مربع که به‌صورت زیر تعریف می‌شود:

$$f_{Square}(x,y)=\begin{cases}0 & if\ \ d(x,y)>\sigma \\ 1 & otherwise\end{cases} \qquad (۲۳-۳)$$

- تابع تأثیر فاصله گوسی که به‌صورت زیر تعریف می‌شود:

$$f_{Gaues}(x,y)=e^{\frac{-d(x,y)^2}{2\sigma^2}} \qquad (۲۴-۳)$$

- تابع اقلیدسی:

در رابطه‌های بالا σ عددی ثابت است که برای فضای مورد نظر تعریف می‌شود و $d(x,y)$ همان فاصله بین دو نقطه یا عدم شباهت بین آنها را نشان می‌دهد. با توجه به تعاریف بالا تابع چگالی[1] به‌صورت مجموع توابع تأثیر همه نقاط تعریف می‌شود با فرض اینکه N داده به‌صورت $D=\{x_1,...,x_N\}\subset F^d$ داشته باشیم تابع چگالی به‌صورت زیر تعریف می‌شود.

$$F_B^D=\sum_{i=1}^{N} f_B^{x_i}(x) \qquad (۲۵-۳)$$

B نشان دهنده تابع اصلی بوده و D به مجموعهٔ بالا اشاره می‌کند. به‌عنوان مثال تابع حاصل شده از تابع تأثیر گوسی به‌صورت زیر خواهد بود:

$$f_{Gauesion}^D(x)=\sum_{i=1}^{N} e^{\frac{-d(x,y)^2}{2\sigma^2}} \qquad (۲۶-۳)$$

حال از روی این تابع می‌توان جاذب چگالی را محاسبه نمود که حداکثر محلی تابع مورد بحث است. در چنین حالتی با الگوریتمهای تپه‌نوردی[2] می‌توان این نقاط و مجموعه نقاط اطراف آنها را مشخص کرد.

حال خوشه‌ها را این‌گونه تعریف می‌کنیم:

---

[1]- Density Function
[2]- Hill Climbing

یک خوشه مرکزمحور[1]: یعنی خوشه‌ای که به‌طور منظم حول یک یا چند محور شناسایی شده است و زیر مجموعه‌ای از نقاط فضا است که به‌صورت چگالشی استخراج شده، و در هیچ نقطه‌ای چگالی‌ای کمتر از حداقل ε ندارند و در نقاطی که تابع چگالی کمتر از ε است دادهٔ پرت یا اغتشاش وجود دارد.
یک خوشه با شکل غیر منظم: مجموعه‌ای از خوشه‌های کروی است که با یک مسیر مانند p که در طول آن چگالی کمتر از ε نشده باشد به یکدیگر متصل شده باشند.

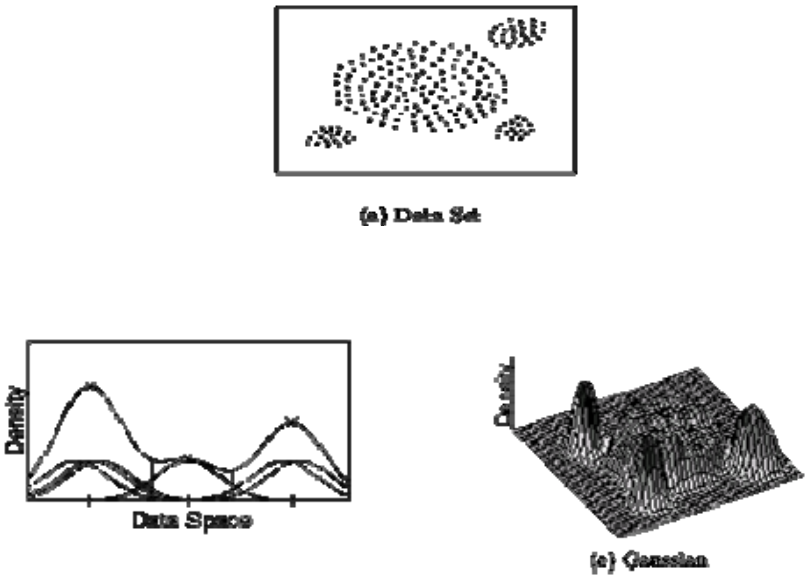
مزایای روش *DENCLUE*

این روش نسبت به دیگر روشها دارای مزایای زیر است:

- از پشتوانه‌ای ریاضی برخوردار بوده و روشهای دیگر مانند افرازی و روشهای سلسله مراتبی را نیز در بر می‌گیرد.
- برای مجموعه داده‌هایی با اغتشاش بالا بسیار مناسب است.
- امکان استفاده از توابع پیچیده را برای تشخیص شکل خوشه‌ها و چگالی فراهم می‌کند.
- با ترکیب با دیگر روشها از جمله روشهای مبتنی بر مشبک‌کردن فضا بسیار سریع‌تر عمل می‌کند.

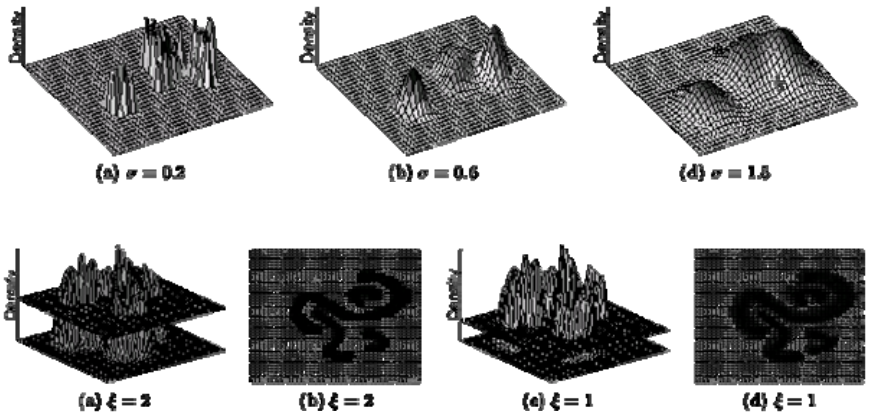
این روش حدود ٤٥ برابر از *DBSCAN* سریع‌تر بوده اما به پارامترهای اولیه مانند σ و آستانه اغتشاش یعنی ε شدیداً حساس است. شکلهای زیر مجموعه‌ای از داده‌ها و تابع چگالی مربوط به فضای آنها را نشان می‌دهد.

[1]- Center Defined Cluster



شکل ۳- ۱۳) تابع چگالی مربوط به فضای آنها

شکل بعد نشان می‌دهد که این روش تا چه میزان به پارامترهای اولیه حساس است. در بخش‌های b و d می‌توان تفاوت حاصل شده از هر برش را مشاهده کرد.



شکل ۳-۱٤) تابع چگالی و میزان حساسیت به پارامترهای اولیه

روشهای مبتنی بر مشبک کردن فضا[1]

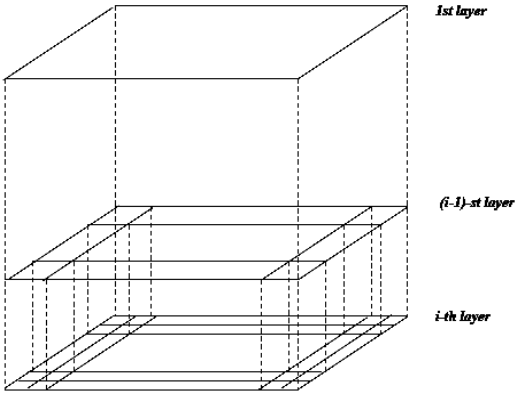
روش مشبک‌سازی فضا به سلولهای مختلف، امکان کار بر روی اطلاعات با درجه تفکیک شفافیت‌های متفاوت[2] را فراهم می‌کند.

در این روش ابتدا فضا به سلولهایی تقسیم شده و سپس عملیات خوشه‌بندی روی این سلولها انجام می‌گیرد. مهمترین مزیت این روش افزایش سرعت است زیرا پیچیدگی محاسباتی را کاهش می‌دهد چرا که پیچیدگی وابسته به تعداد سلولهاست نه تعداد داده‌ها.

ابتدایی‌ترین و ساده‌ترین روش در این دسته روش شبکه اطلاعات آماری یا *STING*[3] است. در این روش فضا به سلولهایی ابتدایی تقسیم می‌شود. اغلب اوقات از روی این سلولها سلولهایی دیگر در لایه‌ای بالاتر تشکیل می‌شوند یعنی مثلاً از ترکیب هر ٤ سلول، یک سلول در لایه‌ای بالاتر با درجه تفکیک کمتر شکل می‌گیرد و این کار به‌صورت سلسله مراتبی برای چندین لایه تکرار می‌شود. سپس برای هر سلول اطلاعات آماری مانند میانگین، میانه، بیشینه، کمینه، انحراف معیار استاندارد و... محاسبه می‌شود. شکل (۱۵-۳) یک ساختار سلسله مراتبی را نشان می‌دهد.

---

[1]- Grid- Based Methods
[2]- Multi - Resolution
[3]- A Statistical Information Grid Approach



شکل ۳ – ۱۵) ساختار سلسله مراتبی

این پارامترهای آماری و حتی نوع توزیع آماری داده‌های پایگاه‌های داده محاسبه شده و به هر سلول تخصیص داده می‌شوند. چنین توزیعی می‌تواند توسط کاربر مشخص شده و یا توسط امتحان فرضیه‌هایی مانند تست χ^2 معین شوند. کاملاً مشخص است که اطلاعات مرتبه‌های بالاتر از مرتبه‌های پایین تر به سادگی قابل محاسبه خواهند بود.

نوع توزیع مرتبه‌های بالاتر می‌تواند از نوع اکثریت توزیع سلولهای پایین به‌دست آید. برای تحلیل خوشه‌ها، ابتدا یک لایه که دارای تعداد کمی خوشه است انتخاب می‌شود، سپس برای تحلیل بیشتر در سلولهایی که خوشه‌ها را تشکیل می‌دهند به لایه پایین‌تر رفته و تحلیل را ادامه می‌دهیم. توجه کنید که در هر گام عملاً با حذف بسیاری از سلولها در اصل داده‌های پرت را کنار می‌گذاریم. این روش برای جستجو[1] در پایگاههای داده بسیار مناسب است.

مزایای این روش عبارتند از:

- این روش پردازش موازی را تسهیل می‌کند.

[1]- Query

- چون این روش فقط یک بار روی کل داده‌ها اجرا می‌شود پیچیدگی آن از مرتبهٔ N است $O(N)$.
- از آنجا که محدوده خوشه‌ها به‌صورت مربعی، یعنی خط‌های طولی و عرضی سلولها مشخص می‌شوند، لذا در اینجا نیز محاسبات با سهولت بیشتری انجام خواهد شد. حتی بعضاً تفسیر آنها نیز ساده تر خواهد بود.

الگوریتمهای خوشه‌بندی قطعی، داده‌ها را به گونه‌ای افراز می‌کنند که هر داده دقیقاً به یک خوشه تخصیص داده می‌شود. در هر حال، اغلب نمی‌توان هر داده را دقیقاً به یک خوشه تخصیص داد چرا که برخی داده‌ها بین خوشه‌ها قرار می‌گیرند. در این موارد، روشهای خوشه‌بندی فازی ابزارهایی بسیار مناسب‌تر برای نمایش ساختار واقعی این نوع داده‌ها هستند برای اطلاعات بیشتر به مرجع [۲] مراجعه کنید.

نقشه‌های خودسازمانده

نقشه‌های خودسازمان یا خودسازمانده[۱] *(SOM)* ابزار قدرتمند و جذابی برای نمایش داده‌های چند بعدی در فضاهای با ابعاد پایین، (معمولاً یک یا دو بعد) فراهم می‌کند.[۳] همچنین *SOM* روشی برای خوشه‌بندی و پیش‌پردازش اطلاعات می‌باشد. نقشه‌های خودسازمانده که گاهی نقشه‌های مشخصه خودسازمان[۲] و یا نقشه‌های کوهونن[۳] نامیده می‌شود، توسط پروفسور تیوو کوهونن[۴] از دانشگاه فنلاند ابداع شده است. این فرآیند کاهش بعد بردارها، روشی برای فشرده‌سازی داده‌ها به نام کمی‌سازی برداری[۵] می‌باشد. علاوه بر این، *SOM* شبکه‌ای برای ذخیره اطلاعات ایجاد می‌کند به نحوی که ارتباط مکانی[۶] بین مجموعه آموزشی حفظ می‌شود.

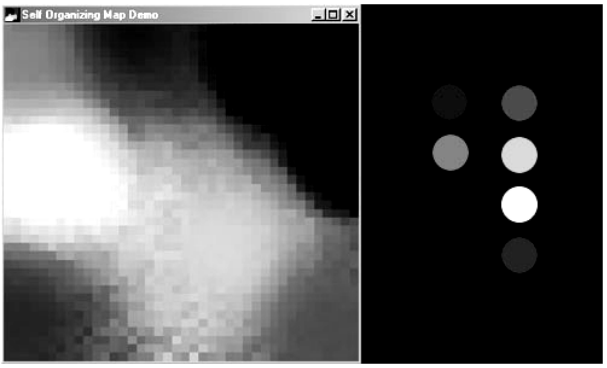
---

[1]- Self-Organizing Maps: SOM
[2]- SOFM: Self-Organizing Feature Maps
[3]- Kohonen Maps
[4]- Teuvo Kohonen
[5]- Vector Quantization
[6]- Topologic

تفاوت *SOM* با شبکه رقابتی[1] عبارت است از:

- در *SOM* هیچ سوگیری[2] وجود ندارد. سوگیری، مقدار وزن نرون ورودی ثابت است.
- علاوه بر نرون برنده، نرونهای همسایه نیز تطبیق یافته و اوزان آنها اصلاح می‌شود.

مثالی متداول برای کمک به آموزش مبانی *SOM*، نگاشت رنگها در صفحه دو بعدی است. فرض کنید هزاران مشاهده داریم و هر مشاهده یکی از ۸ رنگ سمت راست شکل (۳-۱۶) باشد. هر رنگ از سه جزء قرمز، سبز و آبی تشکیل شده است که می‌توانند دارای مقداری بین ۰ تا ۲۵۵ باشند، بنابراین هر مشاهده دارای سه ویژگی می‌باشد. اگر بخواهیم رنگها را در فضای واقعی خود ترسیم کنیم نیاز به سه بعد داریم.



شکل ۳- ۱۶) نمایش خروجی شبکه (چپ) و رنگهای خوشه‌بندی شده توسط آن (راست)

رنگها به‌صورت بردارهای سه بعدی (یک بعد برای هر جزء رنگ) به شبکه *SOM* معرفی شده و شبکه بعد از آموزش، هر مشاهده (رنگ) را به یکی از نقاط نقشه دو بعدی در شکل سمت چپ، نگاشت می‌کند. برای درک تصویری بهتر، هر نقطه از نقشه را با متوسط رنگ مشاهدات نگاشت شده به آن، رنگ‌آمیزی می‌کنیم. توجه کنید که علاوه بر خوشه‌بندی رنگها به نواحی مجزا، معمولاً نواحی مشابه در کنار یکدیگر

---

[1]- Competitive Network
[2]- Bias

قرار می‌گیرند. همان‌طورکه بعداً خواهید دید، اغلب می‌توان از این ویژگی نقشه‌های کوهونن استفاده خوبی کرد.

چنانچه گفته شد یکی از جالبترین جنبه‌های *SOM* یادگیری آنها برای خوشه‌بندی است، ممکن است قبلاً با فنون آموزش با ناظر مثل پس‌انتشار[1] آشنا باشید. در این روش داده‌های آموزشی شامل زوج بردار ورودی و بردار هدف هستند. در روش پس‌انتشار یک بردار ورودی به شبکه‌ای مثل شبکه چندلایه پیشخور[2]، داده شده و خروجی با بردار هدف مقایسه می‌شود. اگر تفاوتی وجود داشته باشد، اوزان شبکه طوری اصلاح می‌شوند تا خطای خروجی را کاهش دهند. این عمل بارها با مجموعه‌های متعددی از زوج بردارها تکرار می‌شود تا زمانی که خروجی موردنظر ارائه شود. در مقابل آموزش *SOM* به بردار هدف نیازی ندارد. یک *SOM* یاد می‌گیرد که داده‌های آموزشی را بدون ناظر بیرونی خوشه‌بندی کند.

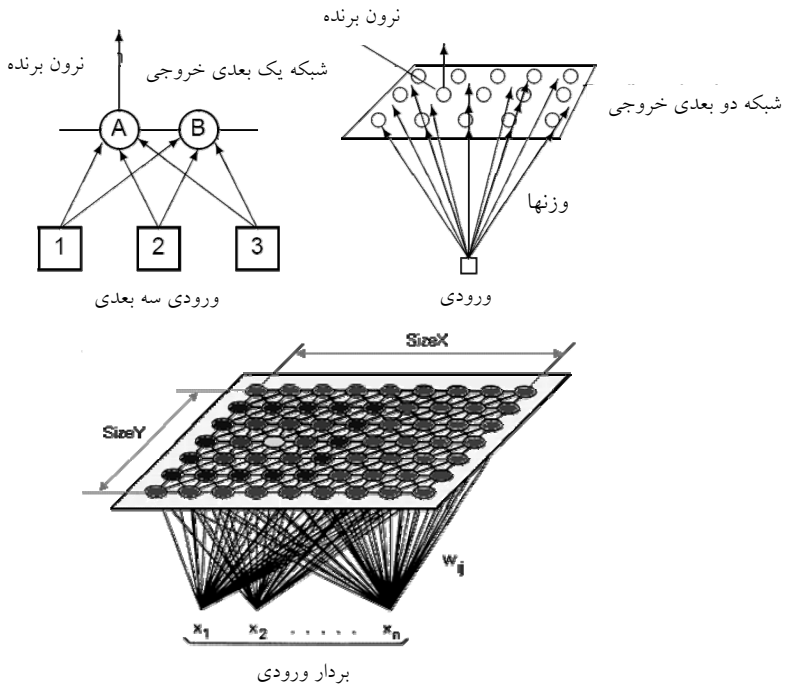
بهتر است قبل از ادامه، هر چیزی را که از قبل در مورد شبکه عصبی می‌دانید فراموش کنید. اگر به شبکه *SOM* به دید نرونها، توابع فعال‌سازی و اتصالات پیشخور/بازگشتی نگاه کنید سریعاً سردرگم می‌شوید. پس قبل از مطالعه بیشتر، همه دانش قبلی را موقتاً کنار بگذارید.

ساختار شبکه

در ابتدا یک *SOM* دو بعدی بررسی می‌شود. شبکه از گره‌های شبکه نردبان[3] دو بعدی ایجاد می‌شود که هر یک به‌طور کامل به لایه ورودی وصل شده‌اند. شکل (۳-۱۷) سمت راست یک شبکه کوهونن بسیار کوچک ۳×۵ را نشان می‌دهد که به لایه ورودی وصل شده ونشانگر یک بردار دو بعدی است.

---

[1]- Back propagation
[2]- Feed Forward
[3]- Lattice



شکل ۳-۱۷) شبکه کوهونن با یک بعد و سه ورودی (چپ بالا)، دو بعد و یک ورودی (راست بالا) و دو بعد و n ورودی (پایین)

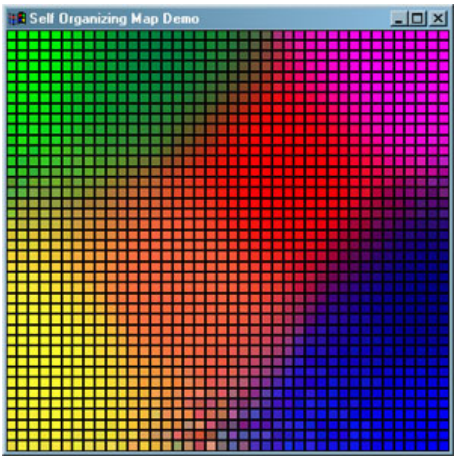
هر گره دارای موقعیت مکانی مشخصی بوده (یک جدول مختصات x,y) و دارای برداری از اوزان با همان ابعاد بردارهای ورودی می‌باشد. اگر داده‌های آموزشی دارای بردارهای X با n بعد باشند: $X_1, X_2, X_3, ..., X_n$ آنگاه هر گره دارای بردارهای اوزان W با n بعد خواهد بود: $W_1, W_2, W_3, ..., W_n$.

خطوط اتصال گره‌ها که برخی اوقات ترسیم می‌شوند فقط برای نمایش مجاورت بوده و بر خلاف شبکه‌های عصبی معمولی اتصالی را معین نمی‌کنند. هیچ اتصال جانبی[1] بین گره‌های شبکه نیست.

در شکل SOM دارای اندازه پیش فرض ٤٠×٤٠ نقطه (خوشه) می‌باشد. هر گره در جدول سه وزن دارد، یکی برای هر جزء بردار ورودی: قرمز، سبز و آبی. هر گره در

[1]- Lateral

هنگام ترسیم در صفحه با یک سلول مستطیلی نشان داده می‌شود. شکل (۳–۱۸) خروجی شبکه را نشان می‌دهد. در این شکل، هر سلول با کادر سیاه نشان داده شده تا بتوان به وضوح گره‌ها را دید.



شکل ۳– ۱۸) هر سلول نشانگر یک گره جدول است

مروری بر الگوریتم یادگیری

بر خلاف بسیاری از شبکه‌ها، یک *SOM* احتیاجی به مشخص کردن خروجی هدف ندارد. در عوض وقتی اوزان یک گره با بردار ورودی منطبق هستند (یعنی فاصله بردار اوزان تا بردار الگوی ورودی کم است)، ناحیه‌ای از جدول به‌طور انتخابی بهینه می‌شود تا بیشتر داده‌های خوشه‌ای را که بردار ورودی به آن تعلق دارد، تقلید کند. با شروع از یک توزیع اولیه اوزان تصادفی و طی دورهای مکرر، *SOM* سرانجام به نقشه‌ای از نواحی باثبات میل می‌کند. می‌توان خروجی را تصویری (نقشه‌ای) از مشخصه‌های ورودی در نظر گرفت. اگر دوباره نگاهی به شبکه آموزش دیده شکل (۳–۱۸) بکنید، بلوکهای رنگ مشابه، نمایانگر نواحی انفرادی هستند. با ورود هر بردار ورودی جدید، شبکهٔ دارای بردار اوزان مشابه تحریک می‌شود، در اینجا نرون تحریک شده اوزان خود و همسایگانش را طوری اصلاح می‌کند که به اوزان الگوی ورودی نزدیک شود.

فرآیندهای اصلی *SOM* عبارتند از:

- رقابت[1]: برای تعیین نرون برنده
- همکاری[2]: کمک به همسایگان در جدول شبکه
- تطبیق[3]: اصلاح وزنها برای نزدیکی بیشتر به بردار ورودی

آموزش در چند قدم و طی دورهای مکرر انجام می‌شود الگوریتم آموزش عبارت است از:

- قدم اول: اوزان هر گره مقداردهی اولیه می‌شوند.
- قدم دوم: برداری از داده‌های آموزشی به تصادف انتخاب شده و به جدول داده می‌شود.
- قدم سوم: هر گره بررسی می‌شود تا گرهی که دارای مشابه‌ترین اوزان به بردار ورودی است پیدا شود. گره برنده معمولاً به‌عنوان بهترین واحد انطباق (یا BMU[4]) شناخته می‌شود.
- قدم چهارم: شعاع همسایگی BMU محاسبه می‌شود. مقدار این شعاع در ابتدا بزرگ و معمولاً برابر شعاع جدول است ولی با هر گام زمانی کوچک می‌شود. هر گره داخل این شعاع به‌عنوان همسایه BMU در نظرگرفته می‌شود.
- قدم پنجم: اوزان هر گره همسایه (که در قدم چهارم پیدا شده است) برای تشابه بیشتر به بردار ورودی تصحیح می‌شوند. هر چه یک گره به BMU نزدیک‌تر باشد، اوزانش بیشتر تغییر می‌یابد.
- قدم ششم: قدم دوم برای N دور تکرار می‌شود.

اکنون قدم‌های الگوریتم یادگیری به‌طور مفصل بررسی می‌شود.

وزن‌دهی اولیه

---

[1]- Competition
[2]- Cooperation
[3]- Adaptation
[4]- BMU: Best Matching Unit

پیش از آموزش، اوزان هر گره باید وزن‌دهی اولیه شوند. معولاً مقادیر تصادفی کوچکی به این اوزان تخصیص داده می‌شود. اوزان اولیه در *SOM* معمولاً بین ۰ و ۱ مقدار دهی می‌شوند: $۰ < w < ۱$. برخی اوقات از کلیه بردارهای ورودی میانگین گرفته شده و به آنها یک عدد کوچک تصادفی اضافه می‌شود تا اوزان اولیه ایجاد شود.

محاسبه *BMU*

یک راه برای تعیین *BMU*، جستجوی همه گره‌ها و محاسبه فاصله‌اقلیدسی بین بردار اوزان هر گره و بردار ورودی فعلی است. گرهٔ دارای نزدیک‌ترین بردار اوزان به بردار ورودی به‌عنوان *BMU* برچسب‌گذاری می‌شود.

فاصله‌اقلیدسی با این رابطه داده می‌شود:

$$Dist = \sqrt{\sum_{i=۰}^{n} (X_i - W_i)^۲} \qquad (۲۷-۳)$$

که در آن X بردار ورودی فعلی و W بردار اوزان گره است.

با اینکه هر جزء رنگ (قرمز، سبز و آبی) در کامپیوتر با عددی از ۰ تا ۲۵۵ تعیین می‌شود، بردارهای ورودی طوری نرمال می‌شوند که هر جزء مقداری بین ۰ و ۱ داشته باشد. گاهی اوقات همه بردارها در فاصله ۰ و ۱ نرمال می‌شوند. این کار برای هماهنگی با محدوده مقادیر وزنها انجام می‌شود. اشکال نرمال کردن طول بردار این است که اطلاعات اندازه بردار از بین می‌رود. می‌توان برای جلوگیری از این اثر جانبی ابتدا یک جزء مصنوعی با مقدار یک به همه بردارهای ورودی اضافه کرد و سپس نرمال کردن را انجام داد. این آخرین جزء مصنوعی از نرمال شدن حاوی اطلاعات اندازه بردار اصلی به‌صورت معکوس اندازه خواهد بود.

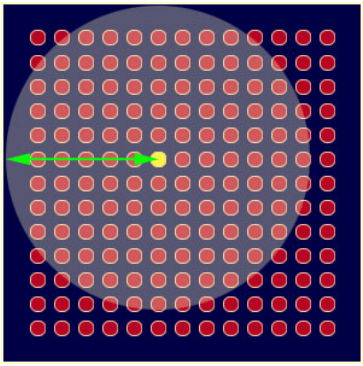
به‌عنوان مثال برای محاسبه فاصله بین بردار رنگ قرمز (۰، ۰، ۱) با بردار دلخواه اوزان (۰/۵، ۰/٤، ۰/۱) داریم:

$$Distance = \sqrt{(۱-۰)^۲ + (۰-۰/۴)^۲ + (۰-۰/۶)^۲} = \sqrt{(۱/۴۲)} = ۱/۱۹$$

گاهی اوقات در نرم‌افزار *Matlab* به جای فاصله دو بردار از زاویه بین دو بردار برای اندازه‌گیری شباهت استفاده می‌شود. در صورت نرمال شدن هر بردار به طول یک، زاویه بین دو بردار برابر ضرب داخلی دو بردار (مشابه شبکه‌های رقابتی) خواهد بود.

تعیین همسایگی محلی بهترین واحد منطبق (جور)

قدم بعدی پس از تعیین *BMU*، یافتن همسایگان *BMU* است. اوزان همه این گره‌ها در قدم بعدی تغییر می‌یابد. برای این کار باید ابتدا شعاع همسایگی محاسبه و سپس با روش ساده فیثاغورث، وجود هر گره در داخل شعاع تعیین شود. شکل (۳-۱۹) مثالی از شروع آموزش همسایگی است.



شکل ۳- ۱۹) همسایگی BMU

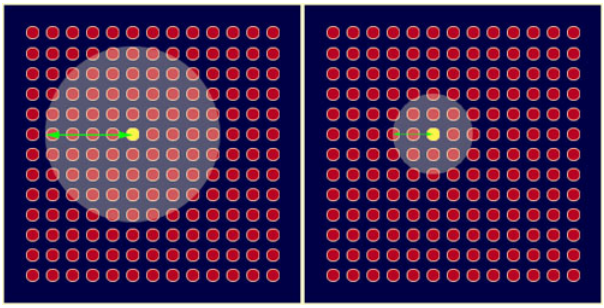
می‌بینید که همسایگی نشان داده شده در این شکل حول مرکز *BMU* بوده و اکثر نقاط دیگر را در بر می‌گیرد. فلش نشان دهنده شعاع است. در برخی موارد همسایگی را به جای دایره به شکل مستطیل در نظر می‌گیرند.

ویژگی منحصر به فرد الگوریتم یادگیری کوهونن، کوچک شدن همسایگی در طی زمان است. این کار با کم کردن شعاع در طول زمان انجام می‌شود. برای این کار از تابع کاهش نمایی استفاده می‌شود:

$$\sigma(t) = \sigma_{۰} e^{\frac{-t}{\lambda}} \qquad t = ۱,۲,۳,... \qquad (۲۸-۳)$$

که در آن σ_0 نشان دهنده عرض جدول در زمان t_0 و λ یک ثابت زمانی است. t قدم زمانی فعلی (دور حلقه) می‌باشد. مقدار λ وابسته به σ و تعداد دور انتخاب شده برای اجرای الگوریتم است.

شکل (۳-۲۰) نشان می‌دهد که چگونه همسایگی در شکل طی زمان کاهش می‌یابد.



شکل ۳- ۲۰) شعاع همواره در حال کاهش

در طی زمان همسایگی به کوچکی یک گره یعنی همان *BMU* می‌شود. وقتی شعاع را بدانیم، به آسانی می‌توان همه گره‌های جدول را بررسی کرد که آیا داخل شعاع هستند یا خیر وقتی گره‌ای در همسایگی پیدا می‌شود آنگاه بردار اوزان آن اصلاح می‌شود.

اصلاح وزنها

بردار اوزان هر گره همسایه *BMU* از طریق این رابطه اصلاح می‌شود:

$$W(t+1) = W(t) + L(t)(X(t) - W(t)) \qquad (۲۹-۳)$$

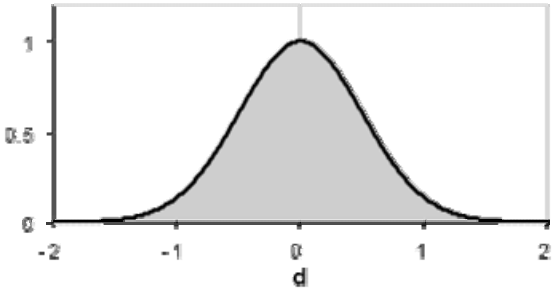
که در آن t نشانگر گام زمانی و L متغیر کوچکی به نام *نرخ یادگیری* است که در طول زمان کم می‌شود. در واقع این رابطه بیان می‌کند که وزن اصلاح شده جدید برابر وزن قدیمی *(W)* به اضافه بخشی *(L)* از تفاوت بین وزن قدیمی و بردار ورودی *(X)* است.

کاهش نرخ یادگیری در هر دور از طریق این رابطه انجام می‌شود:

$$L(t) = L_0 e^{-t/\lambda} \quad t = 1,2,3,... \qquad (۳۰-۳)$$

در ابتدا نرخ یادگیری مقدار ثابتی مثل ۰٫۱ است و به تدریج در طول زمان به صفر میل می‌کند.

رابطه (۳-۲۷) نه تنها باید نرخ یادگیری در طول زمان کاهش یابد بلکه باید اثر یادگیری متناسب با فاصله یک گره از *BMU* باشد. در واقع در لبه‌های بیرونی همسایگی *BMU*، مقدار یادگیری بسیار ناچیز است. به‌طور ایده‌آل مقدار یادگیری باید طبق نزول گاوسی شکل (۳-۲۱) در امتداد فاصله کاهش یابد.



شکل ۳- ۲۱) کاهش یادگیری بر حسب فاصله طبق منحنی گاوسی

برای دستیابی به این هدف، رابطه (۳-۲۸) باید کمی تغییر یابد:

$$W(t+1) = W(t) + \Theta(t)L(t)(X(t) - W(t)) \qquad (۳۱-۳)$$

Θ نمایانگر مقدار تأثیر فاصله یک گره از *BMU* روی یادگیری آن است و با رابطه (۳- ۳۱) بیان می‌شود.

$$\Theta(t) = e^{-dist^2 / 2\sigma^2(t)} \quad t = 1,2,3,... \qquad (۳۲-۳)$$

که در آن *dist* فاصله گره از *BMU* و σ عرض تابع همسایگی محاسبه شده در رابطه (۳-۲۷) است. همچنین توجه کنید Θ نیز در طول زمان کاهش می‌یابد.

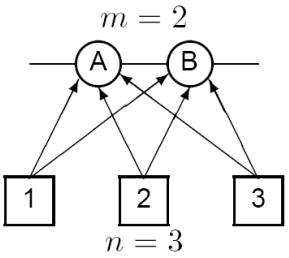
مثال عددی

یک *SOM* با ۳ گره (مشخصه) ورودی و دو گره خروجی A و B در شکل را در نظر بگیرید.

اوزان اولیه A و B از این قرار هستند: $w_A = (۳، -۱، ۲)$ $w_B = (-۲، ۰، ۱)$

مقدار ورودی برابر است با: $x = (۱ ،-۲ ،۲)$

توجه کنید که در این مثال برای سادگی عمل نرمال‌کردن روی بردارها انجام نشده است.



شکل ۳- ۲۲) شبکه ساده با خروجی یک بعدی

فاصله‌ها را محاسبه می‌کنیم:

$$\| x - w_A \| = \sqrt{(۱-۲)^۲ + (-۲+۱)^۲ + (۲-۳)^۲} = \sqrt{۳}$$

$$\| x - w_B \| = \sqrt{(۱+۲)^۲ + (-۲-۰)^۲ + (۲-۱)^۲} = \sqrt{۱٤}$$

پس نرون A برنده می‌شود چون فاصله کمتری دارد. حال اوزان نرون برنده را اصلاح می‌کنیم:

$$w_A = \begin{pmatrix} ۲ \\ -۱ \\ ۳ \end{pmatrix} + ۰.۵ \times ۱ \times \left[\begin{pmatrix} ۱ \\ -۲ \\ ۲ \end{pmatrix} - \begin{pmatrix} ۲ \\ -۱ \\ ۳ \end{pmatrix} \right] = \begin{pmatrix} ۲ \\ -۱ \\ ۳ \end{pmatrix} + ۰.۵ \begin{pmatrix} -۱ \\ -۱ \\ -۱ \end{pmatrix} = \begin{pmatrix} ۱.۵ \\ -۱.۵ \\ ۲.۵ \end{pmatrix}$$

کاربردهای *SOM*

معمولاً *SOM* مانند بقیه روشهای خوشه‌بندی به دو منظور استفاده می‌شود:

- پیش‌پردازش (در شبکه‌های عصبی): معمولاً در ابتدای شبکه‌های عصبی دیگر مثل پس‌انتشار خطا یک لایه *SOM* نیز قرار می‌دهند تا با خوشه‌بندی اطلاعات ورودی و استخراج مشخصه‌ها به حذف اغتشاش، بهبود صحت نتایج و افزایش سرعت آموزش کمک شود. برای مثال اگر می‌خواهیم مصرف برق را در روز بعد پیش‌بینی کنیم بهتر است ابتدا داده‌های آموزشی سوابق شامل مشخصه‌های دمای هوا و

مصرف روز قبل را در داخل یک شبکه *SOM* نگاشت کنیم تا به‌جای داده‌های اصلی، نگاشتهای برآیندی آنها را داشته باشیم و سپس این برآیندها را به‌عنوان ورودی شبکه عصبی پس‌انتشار خطا برای پیش‌بینی استفاده کنیم.

- ابزار مصورسازی: برای تحلیل اکتشافی داده‌ها به کار می‌رود. نقشه‌های خودسازمان، مشاهده روابط بین حجم بزرگی از داده‌ها را برای انسان آسان می‌کنند. این مورد در مثال زیر بهتر شرح داده شده است.

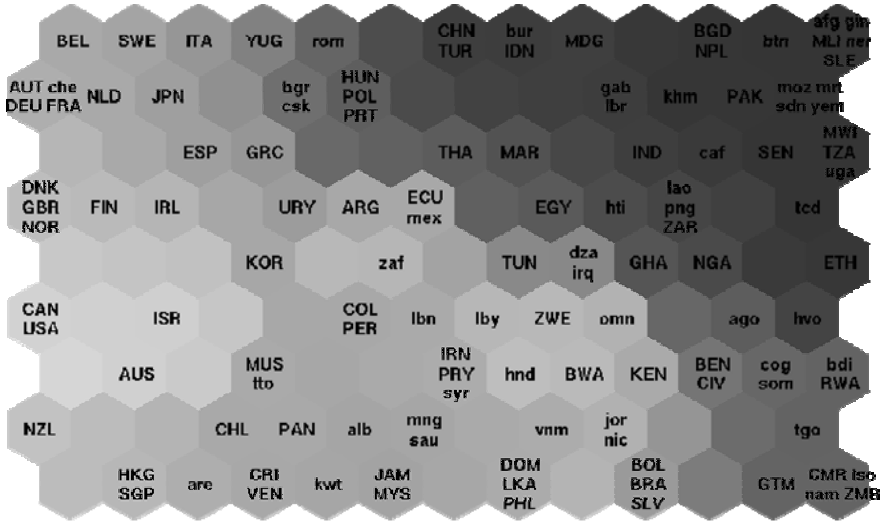
مثال: نقشه فقر[1] جهانی

شبکه *SOM* می‌تواند برای نشان دادن همبستگی‌های پیچیده در داده‌های آماری استفاده شود. در اینجا داده‌ها شامل آمار بانک جهانی از کشورها در سال ۱۹۹۲ می‌باشد. [٤] ۳۹ شاخص برای طبقه‌بندی داده‌های آماری عوامل کیفیت زندگی[2] مانند سطح سلامت، تغذیه، خدمات تحصیلی و غیره استفاده شده است. کشورهای دارای عوامل کیفیت زندگی مشابه در خوشه‌های یکسانی قرار می‌گیرند. در شکل (۳-۲۳) مشاهده می‌کنید که کشورهایی با کیفیت زندگی بهتر در سمت گوشه چپ بالا و اکثر کشورهای فقیر در گوشه راست پایین قرارگرفته‌اند. هر چند ضلعی نمایانگر یک گره در *SOM* است.

به‌طور عمومی برای رنگ‌آمیزی *SOM* دو راه نیز وجود دارد. در مصورسازی به روش ماتریس u[3]، رنگ تیره متناظر با تفاوت قابل ملاحظه بین بردارهای مدل واحدهای مجاور در نقشه بوده (وجود مرز)، در حالی که رنگ روشن شباهت بین همسایگان را نشان می‌دهد. در روش دوم به نام نمودار چگالی، رنگ روشن نشان‌دهنده تعداد زیاد الگوهای مشابه و رنگ تیره نشانه نقاط خالی‌تر است. در اینجا ساختار خوشه‌ها با یک روش تصویر کردن غیر خطی به فضای رنگ *CIELAB* نگاشت شده‌اند [۳].

---

[1]- Poverty Map
[2]- Quality-of-Life
[3]- U-Matrix

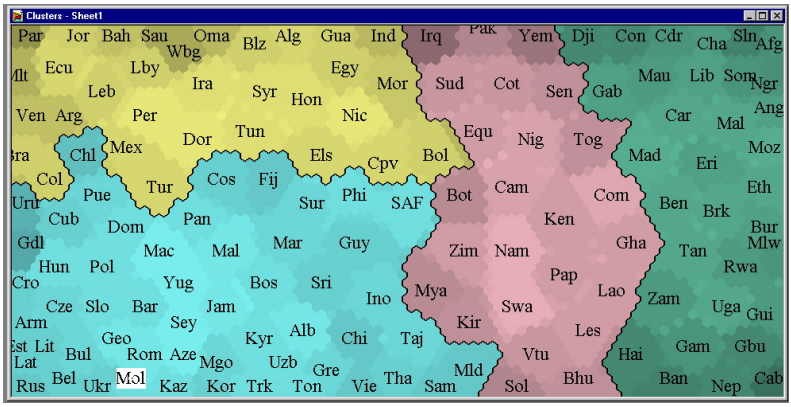


شکل ۳-۲۳) کشورها روی یک SOM (نقشه خودسازمانده) بر مبنای شاخصهای فقر سازمان یافته‌اند

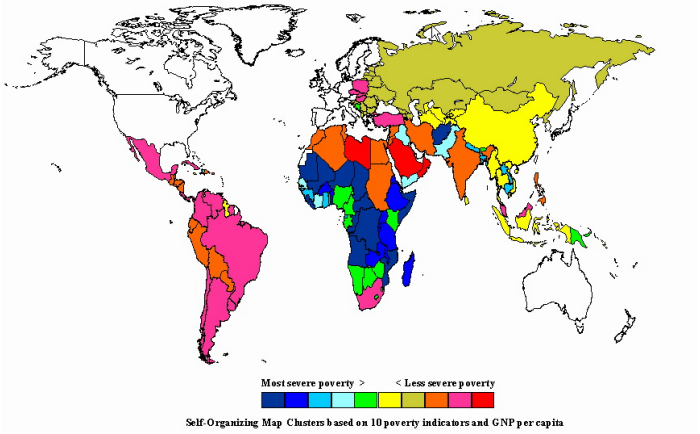
می‌توان این اطلاعات رنگی را روی نقشه زمین شکل (۳-۲٤) رسم کرد.

شکل ۳-۲٤) نقشه جهان که در آن کشورها بر مبنای SOM (نقشه خودسازمانده) در شکل قبل رنگ‌آمیزی شده‌اند

در مطالعه‌ای دیگر نقشه‌های زیر به‌دست آمده است.[۵] داده‌های این مطالعه تا سال ۱۹۹۷ بوده ۱۰ شاخص مربوط به کیفیت زندگی در نظر گرفته شده است.



نقشه فقر جهانی



منابع

1) *Han. J, Kamber. M. (2006) "Chapter7: Cluster Analysis", Data mining concepts and techniques, 2nd edition, , Morgan Kaufmann Publishers .*

۲) *کتاب نظریه مجموعه‌های فازی، غضنفری، رضایی، انتشارات علم وصنعت، زمستان ۸۵*

3) *Kohonen T. (2001) Self-Organizing maps, 3rd Edition.*

4) *World Bank Group - Data and Statistics (Sited 2005/2/20) http://www.worldbank.org/data/*

5) *World Poverty Map (Sited 2005/2/20) http://www.cis.hut.fi/research/som-research/worldmap.html*

فصل چهارم

قواعد تلازمی

استخراج قواعد تلازمی[1] یا انجمنی نوعی عملیات داده‌کاوی است که به جستجو برای یافتن ارتباط بین ویژگیها در مجموعه داده‌ها می‌پردازد. نام دیگر روش تحلیل تلازمی، تحلیل سبد بازار[2] می‌باشد. به‌عبارت دیگر، تحلیل تلازمی، مطالعه ویژگیها یا خصوصیاتی می‌باشد که با یکدیگر همراه بوده و به دنبال استخراج قواعد از میان این خصوصیات می‌باشد. این روش به دنبال استخراج قواعد به‌منظور کمی‌کردن ارتباط میان دو یا چند خصوصیت است. قواعد تلازمی به شکل اگر و آنگاه به همراه دو معیار پشتیبان و اطمینان[3] تعریف می‌شوند. همان‌طورکه اشاره شد، یکی از کاربردی‌ترین حالت‌های تحلیل قواعد تلازمی، تجزیه و تحلیل سبد بازار است. پیشرفت فنآوری، فروشگاه‌های خرده‌فروشی را قادر ساخته است تا حجم زیادی از داده‌های خرید

[1]- Association Rules
[2]- Market Basket -Basket Data
[3]- Confidence

مشتریان (که از آن به‌عنوان سبد بازار یاد می‌شود) را جمع‌آوری و ذخیره نمایند. هر مشتری خرید مجزایی را در مقادیر مختلف و زمانهای متفاوت انجام می‌دهد و داده‌های موجود در سبد بازار نشان‌دهنده خرید مشتری در یک زمان خاص است. با تجزیه و تحلیل سبد بازار خرده‌فروشان می‌توانند رفتار خرید مشتریان را پیش‌بینی کنند. این کار به آنها کمک می‌کند تا بتوانند کالاهای خود را بهتر ساماندهی کرده و چیدمان بهتری از محصولات خود داشته باشند و از این طریق سودآوری خود را افزایش دهند.

در اینجا به مثالهایی از کاربرد قواعد تلازمی اشاره می‌شود:

- بررسی ارتباط بین توانایی خواندن کودکان با خواندن داستان توسط والدین برای آنها
- پیش‌بینی تنزل رتبه در شبکه‌های مخابراتی.
- بررسی اینکه چه اقلامی در یک فروشگاه با یکدیگر خریداری می‌شوند و اینکه چه اقلامی هیچ‌گاه با یکدیگر خریداری نمی‌شوند.
- تعیین سهم نمونه‌ها در بررسی تأثیرات خطرناک یک داروی جدید.

قواعد التزامی ماهیتاً قواعد احتمالی هستند. به‌عبارت دیگر قاعده $X \Rightarrow A$ لزوماً قاعده $X+Y \Rightarrow A$ را نتیجه نمی‌دهد، زیرا این قاعده ممکن است از شرط حداقل پشتیبان برخوردار نباشد. به‌طرزمشابه قواعد $X \Rightarrow Y$ و $Y \Rightarrow Z$ لزوماً قاعده $X \Rightarrow Z$ را نتیجه نمی‌دهند زیرا قاعده اخیر ممکن است از شرط حداقل اطمینان برخوردار نباشد. [1]

تعاریف و مفاهیم اصلی در قواعد تلازمی

$I = \{I_1, I_2, ..., I_m\}$: مجموعه اقلام خریداری شده است.

T: هر زیرمجموعه‌ای از I می‌باشد که از آن به‌عنوان تراکنش یاد می‌شود.

D: مجموعه تراکنشهای موجود در T است

TID: شناسه منحصر به فرد و یکتایی است که به هر یک از تراکنشها اختصاص می‌یابد.

نمای کلی یک قاعده التزامی به شکل زیر می‌باشد:

$X \Rightarrow Y$ [*پشتیبان* , *اطمینان*]

$X \subset I,\ Y \subset I$ و $X \cap Y = \varnothing$ به‌طوری‌که داریم:

- پشتیبان[1]: نشان‌دهنده درصد یا تعداد مجموعه تراکنش‌های D است که شامل هر دوی X و Y (X,Y) باشند.
- اطمینان: میزان وابستگی یک قلم کالای خاص را به دیگری بیان می‌کند و مطابق فرمول زیر محاسبه می‌شود:

اطمینان (X,Y) = پشتیبان $(X \cup Y)$ / پشتیبان (X) (۱-٤)

این شاخص درجه وابستگی بین دو مجموعه X و Y را محاسبه می‌کند و به‌عنوان شاخصی برای اندازه‌گیری توان یک قاعده درنظرگرفته می‌شود. اغلب اطمینان بزرگی برای قواعد در نظر گرفته می‌شود.

عبارت $X=>Y$ را در نظر بگیرید. فرض کنید که عبارت X آنگاه Y بدین معنی است که هرگاه خریداری محصول X را بخرد، محصول Y را نیز خواهد خرید. اگر ۹۸٪ خریداران هنگامی که X را می‌خرند Y را نیز بخرند ما ۹۸٪ به این قاعده اطمینان خواهیم داشت. اما این کافی نیست چون برای اطمینان بیشتر باید این قاعده زیاد تکرار شود. آنچه این قاعده را پشتیبانی می‌کند درصد خریدهایی است که هر دوی این محصولها با یکدیگر در آن وجود دارند. حال اگر $X=>Y$ را یک قاعده التزامی بنامیم، آنگاه پیدا کردن قواعدی که از یک حداقل پشتیبانی برخوردار بوده و ما به اندازه کافی به آنها اطمینان داشته باشیم، مهم است. در مثال زیر با استفاده از سبد خرید روزانه افراد، به تحلیل خریدهای آنان می‌پردازیم: مجموعه اقلام خریداری شده را به‌صورت زیر فرض کنید. این اقلام در I آمده است.

I={خیار، جعفری، پیاز، گوجه‌فرنگی، نمک، نان، زیتون، پنیر، کره}

مجموعه D شامل تک تک تراکنش‌ها و خریدها است و به فرم زیر تعریف شده است:

---

[1]- Support

$D = \{T_۱, T_۲, T_۳, T_٤, T_٥, T_٦, T_۷, T_۸\}$

$T_۱ = \{$جعفری، پیاز، زیتون، خیار، گوجه‌فرنگی$\}$

$T_۲ = \{$جعفری، خیار، گوجه‌فرنگی$\}$

$T_۳ = \{$نان، نمک، گوجه‌فرنگی، پیاز، جعفری، خیار$\}$

$T_٤ = \{$نان، پیاز، خیار، گوجه فرنگی$\}$

$T_٥ = \{$پیاز، نمک، گوجه‌فرنگی$\}$

$T_٦ = \{$پنیر، نان$\}$

$T_۷ = \{$خیار، پنیر، گوجه‌فرنگی$\}$

$T_۸ = \{$کره، نان$\}$

فرض کنیم یک قاعده تلازمی به شکل زیر داریم:

$X \Rightarrow Y$ [*اطمینان* , *پشتیبان*]

$\{$پیاز, جعفری$\} \Rightarrow \{$خیار, گوجه‌فرنگی$\}$

$X = \{$ گوجه‌فرنگی, خیار $\}$

$Y = \{$ جعفری, پیاز $\}$

$X \cup Y = \{$گوجه‌فرنگی، خیار، جعفری، پیاز$\} = \{T_۱, T_۳\}$

پشتیبان $(X \cup Y) = ۲/۸ = ۰/۲۵$

ازآنجایی‌که مجموعه $X \cup Y$، ۲ عضو و مجموعه D، ۸ عضو دارد، بنابراین الگوی خرید «گوجه‌فرنگی، خیار، جعفری، پیاز» در ۲۵٪ سبد خرید ما رخ می‌دهد.

$T = \{T_۳, T_٤, T_۷, T_۲, T_۱\}$

یعنی $\{$خیار, گوجه‌فرنگی$\}$ در $T_۱$ و $T_۲$ و $T_۷$ و $T_٤$ و $T_۳$ خریداری شده‌اند.

بنابراین داریم:

پشتیبان *(x)*=۵/۸=٦۲٪

پشتیبان *(X ∪ Y)* / پشتیبان *(X)* = *(۲/۸)/(۵/۸)* =۲/۵=٤۰٪ = اطمینان

یعنی هنگامی که افراد «خیار و گوجه‌فرنگی» می‌خرند، در ٤٠٪ اوقات، «جعفری و پیاز» را نیز می‌خرند. هدف اصلی داده‌کاوی در پیدا کردن تلازم و یافتن چنین قواعد محکم و قابل توجهی است.

اگر مجموعه‌ای از عناصر حداقل پشتیبانی لازم را داشته باشند مکرر[1] خوانده می‌شوند. قواعد قوی[2] قواعدی هستند که به‌طور توأمان دارای حداقل پشتیبانی و حداقل اطمینان باشند. با استفاده از این مفاهیم پیدا کردن قواعد التزامی در دو گام خلاصه می‌شود، یعنی پیدا کردن مجموعه‌های مکرر و استخراج قواعد قوی.

تقسیم‌بندی قواعد تلازمی

بر اساس **ارزش عناصر** درون قواعد می‌توان قواعد را به انواع دودویی و کمی تقسیم کرد، در مثال زیر قاعده اولی دودویی و دومی کمی است.

$$computer \Rightarrow Financial\ management\ software\ [sup=٢\%, confidence=٦٠\%]$$

$$age\ (X, "٣٠..٣٠")\ and\ income\ (X, "٤٢k..٤٨k")$$
$$\Rightarrow buys\ (X, high\ resolution\ TV)$$

براساس ابعاد یک قاعده می‌توان آن را تک بعدی یا چند بعدی نامید. قاعده زیر فقط بعد خرید را شامل می‌شود.

$$Buys\ (X, computer) \Rightarrow buys\ (X, "financial\ management\ software")$$

اما قاعده زیر سه بعدی است و ابعاد سن، درآمد و خرید را شامل می‌شود.

$$Age\ (X, "٣٠..٣٩")\ and\ income\ (X, "٤٢k..٤٨k") \Rightarrow buys\ (X, high\ resolution\ TV)$$

از آنجایی که داده‌ها می‌توانند در سطوح[3] و یا مقیاسهای[4] مختلف تعریف شوند، قواعد را می‌توان براساس این سطوح خلاصه نمود. مراتب خلاصه‌سازی و اینکه آیا قواعد در یک سطح هستند یا در چند سطح، می‌تواند مبنای تقسیم‌بندی باشد.

مثال زیر را در نظر بگیرید:

---

[1]- Frequent
[2]- Strong
[3]- Level
[4]- Scale

$$age\ (X\ ,"۳۰..۳۹") \Rightarrow buys\ (X\ ,\ "Laptop\ ")$$

$$age\ (X\ ,"۳۰..۳۹") \Rightarrow buys\ (X\ ,\ "computer")$$

از آنجایی‌که رایانه همراه، زیرمجموعه‌ای از رایانه است این قواعد در دو سطح قرار دارند و این یک مجموعه چند سطحی است. ما در این کتاب بیشتر روی مجموعه‌های تک سطحی تأکید داریم.

استخراج قواعد تک‌سطحی تک‌بعدی دودویی

قبل ازارائه الگوریتمهای استخراج قواعد، نمادها و قراردادهایی را به منظور درک بهتر این الگوریتمها مطرح می‌کنیم.

اقلام مطابق با قاعده ترتیب حروف الفبا[1] چیده می‌شوند به‌عنوان مثال اگر $L_k = \{a[۱], a[۲], ..., a[k]\}$ باشد، مطابق این قاعده باید رابطه« $a[۱] < a[۲] < ... < a[k]$ » برقرار باشد.

در تمامی این الگوریتمها مراحلی که طی می‌شوند به قرار زیر می‌باشند:

- در اولین گذر، پشتیبان هر یک از اجزاء محاسبه شده و اقلام مکرر (با بیشترین میزان فراوانی) با در نظر گرفتن آستانه حداقل پشتیبان انتخاب می‌شوند. (L_K)
- در هر گذر، اقلام مکرر که از فاز قبلی محاسبه شده‌اند برای ایجاد اقلام کاندیدا به‌کار می‌روند. (C_K)
- پشتیبان هر یک از C_Kها محاسبه شده و بزرگ‌ترین آنها انتخاب می‌شوند. این کار تا زمانی‌که هیچ قلم بزرگتری یافت نشود ادامه می‌یابد.

درهرفاز، پس از یافتن اقلام بزرگ (L_K) می‌توان قواعد مطلوب را به‌صورت زیر استخراج کرد:

---

[1]- Lexicographic Order

برای تمامی اقلام مکرر L همه زیرمجموعه‌های غیرتهی آن را درنظر می‌گیریم. برای تمامی این زیرمجموعه‌ها (a)، یک قاعده به‌صورت زیر استخراج می‌کنیم:

"$s \Rightarrow (L-s)$" این قاعده در صورتی برقرار می‌شود که اطمینان حاصل از آن بزرگ‌تر یا مساوی حداقل اطمینان درنظرگرفته شده توسط کاربر باشد به بیان دیگر اگر رابطه زیر برقرار باشد، قاعده فوق پذیرفته می‌شود ودرغیراینصورت این قاعده لغو می‌شود.

(حداقل اطمینان = < (a) پشتیبان / (L) پشتیبان) (۲-٤)

پروسه استخراج قواعد التزامی عبارت است از:

- ابتدا همه اقلام مکرر را که دارای حداقل پشتیبان هستند بیابید.
- برای تمامی اقلام مکرر همه زیرمجموعه‌های آنها را استخراج کنید.
- همه قواعد ممکن را استخراج کنید.
- قواعدی را بپذیرید که از حداقل اطمینان برخوردارهستند.

در اینجا برای پیدا کردن این قواعد از الگوریتم ساده *Apriori* استفاده می‌کنیم. فرض کنید که ابتدا باید تمام مجموعه‌های تک عضوی مکرر را پیدا کنید، سپس بر اساس آن مجموعه‌های دو عضوی مکرر را پیدا کنید و الی آخر. در هر مرحله باید کل فضا جستجو شود اما این الگوریتم از خصوصیت *Apriori* استفاده می‌کند به این صورت که «اگر مجموعه‌ای از عناصر مکرر باشد، تمام زیر مجموعه‌های غیر تهی آن نیز مکرر خواهند بود»[۱].

هر زیرمجموعه یک مجموعه مکرر، خود نیز مکرر است. مثلاً اگر مجموعه {سیگار، نان، شیر} = A مکرر است آنگاه مجموعه‌های زیر نیز مکرر هستند.

{سیگار} ,{ نان} ,{شیر} {سیگار ,نان} ,{سیگار ,شیر} ,{ نان ,شیر}

۱- مشابه این اصل در خوشه‌بندی اطلاعات بر اساس چگالی در تعیین مقادیر همسایگی استفاده می‌شود.

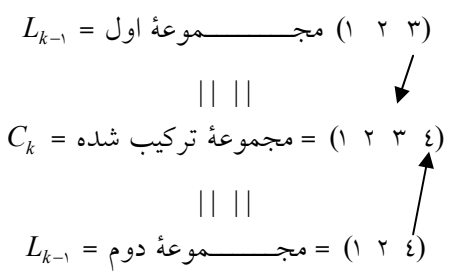
این خصوصیت را این‌گونه نیز می‌توان توصیف کرد: اگر مجموعه I به تعداد مشخصی تکرار شده باشد و اگر ما A را به آن اضافه کنیم تعداد تکرار این مجموعه از مجموعه قبلی بیشتر نخواهد بود. پس اگر اولی مکرر نباشد دومی نیز مکرر نخواهد بود. این الگوریتم از این خصوصیت استفاده می‌کند و در اینجا عملکرد آن را شرح می‌دهیم: می‌دانیم که از یک زیرمجموعه $k-۱$ عضوی یا همان $L_{k-۱}$ برای به‌دست آوردن L_k یعنی مجموعه‌های k عضوی استفاده می‌شود. این کار در دو مرحله صورت می‌گیرد، ابتدا باید مجموعه‌ای از اعضا پیدا شود که با ترکیب $L_{k-۱}$ با آنها، L_k به دست آید. این مجموعه از عناصر را C_k نامیده و مرحله به‌دست آوردن آنها را پیوست[1] می‌نامیم. مرحله بعد اضافه کردن این عناصر به مجموعه‌های قبلی است که آن را مرحله هرس[2] می‌نامیم. در زیر این دو مرحله شرح داده می‌شود.

مرحله پیوست

ابتدا باید مطمئن شویم که عناصر بر مبنای ترتیب حروف الفبا مرتب شده‌اند. دو مجموعه از $L_{k-۱}$ با یکدیگر قابل پیوست هستند اگر $k-۲$ عنصر اول آنها با یکدیگر برابر باشند. یعنی: $(l_۱[۱]=l_۲[۲])\wedge(l_۱[۲]=l_۲[۲])\wedge(l_۱[k-۲]=l_۲[k-۲])\wedge(l_۱[k-۱]<l_۲[k-۱])$ توجه کنید که دو عنصر آخر به ترتیب مرتب شده‌اند و از وجود عناصر تکراری جلوگیری می‌کنند. با اجتماع دومجموعه قابل پیوست، آن دومجموعه ترکیب می‌شوند. با این روش، مجموعه ترکیب شده حاصل k عضو خواهد داشت که البته عنصر آخر (از نظر ترتیبی) از مجموعه دوم خواهد بود. در مثال زیر دو مجموعهٔ (٤، ۲، ۱) و (۳، ۲، ۱) را در نظر بگیرید: مجموعه اول و دوم مرتب هستند و داریم: $۱<۲<۳<۴$ پس می‌توان مجموعهٔ ترکیب شدهٔ زیر را به‌دست آورد:

---

[1]- Join
[2]- Prune



مرحله هرس

C_k مجموعه‌ای از L_kها است که هر عنصر آن یا مکرر است یا خیر، اما تمام عناصر مکرر در آن قرار دارند. حال تمام عناصر این مجموعه باید بررسی شوند تا مکرر بودن آنها مشخص شود اما چون ممکن است تعداد آنها زیاد باشد لذا برای کاهش حجم محاسبات از اصل *Apriori* استفاده می‌شود. به این صورت اگر یکی از زیرمجموعه‌های این مجموعه مکرر نباشد آن مجموعه نیز مکرر نخواهد بود. بنابراین برای پیدا کردن مجموعه‌های مکرر کافی است مجموعه‌های غیر مکرر را از آنها جدا کنیم به این صورت که اگر عضوی از C_k در L_{k-1} نباشد مکرر نیز نخواهد بود.

محاسبهٔ اطمینان و استخراج قواعد نهایی

پس از آنکه مجموعه‌های قوی استخراج شدند حال نوبت استخراج قواعد است:

$$\text{اطمینان } (A \Rightarrow B) = P(B \mid A) = \frac{\text{پشتیبان } (A \cup B)}{\text{پشتیبان } (A)} \qquad (۳-٤)$$

برای هر مجموعه مکرر L تمام زیر مجموعه‌های غیر تهی را در نظر می‌گیریم. برای هر زیرمجموعه s قواعد را به‌صورت زیر شکل می‌دهیم. "$s \Rightarrow (L-s)$"

سپس اطمینان را حساب کرده و اگر بیشتر از حداقل قابل قبول بود آن را می‌پذیریم. در زیر این محاسبات انجام گرفته‌اند.

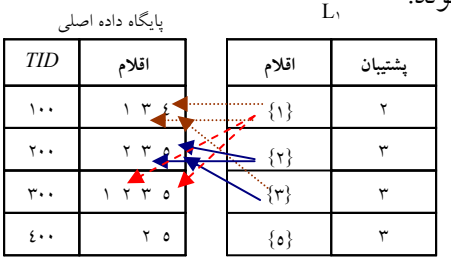
$$L = \{I_۱, I_۲, I_۵\}$$
$$I_۱ \wedge I_۲ \Rightarrow I_۵,\ \ confidence = ۲/٤ = ۵۰\% \rightarrow (S = \{I_۱, I_۲\})$$
$$I_۱ \wedge I_۵ \Rightarrow I_۲,\ \ confidence = ۲/۲ = ۱۰۰\% \rightarrow (S = \{I_۱, I_۵\})$$
$$I_۲ \wedge I_۵ \Rightarrow I_۱,\ \ confidence = ۲/۲ = ۱۰۰\% \rightarrow (S = \{I_۲, I_۵\})$$
$$I_۱ \Rightarrow I_۲ \wedge I_۵,\ \ confidence = ۲/٦ = ۳۳\% \rightarrow (S = \{I_۱\})$$
$$I_۲ \Rightarrow I_۱ \wedge I_۵,\ \ confidence = ۲/۷ = ۲۹\% \rightarrow (S = \{I_۲\})$$
$$I_۵ \Rightarrow I_۱ \wedge I_۲,\ \ confidence = ۲/۲ = ۱۰۰\% \rightarrow (S = \{I_۵\})$$

الگوریتم *AIS*

این الگوریتم ازاولین الگوریتمهایی بود که برای استخراج همه اقلام مکرر از پایگاه داده درسال ۱۹۹۳توسط اگریوال، ایمیلنسکی و سوامی[۱] ابداع گردید.[۲] نام این الگوریتم برگرفته حروف اول نام ابداع کنندگان آن می‌باشد. این الگوریتم چندین گذر بر روی پایگاه داده انجام داده و درهرگذر همه تراکنشها را می‌پیماید. گامهای این الگوریتم به‌صورت زیر می‌باشند:

- برای هر یک ازتراکنشها بزرگ‌ترین قلم انتخاب می‌شود.
- اقلام کاندید (C_k) با گسترش هر یک از این اقلام مکرر به سایر اقلام در هر تراکنش ساخته می‌شوند.

مثال: در قدم اول پشتیبان هر قلم محاسبه شده و آنهایی‌که بیشتر از حداقل پشتیبان هستند در $L_۱$ ثبت می‌شوند.



شکل٤-۱) قدم اول

[1]- R. Agrawal, T. Imielinski, and A. Swami

در قدم دوم به ازای تک تک اقلام مرحلهٔ L_1 به پایگاه دادهٔ اصلی برگشته و تمامی مجموعه‌های دوتایی را ساخته و پشتیبان آنها را محاسبه می‌کنیم. خروجی این مراحل در C_2 ذخیره می‌شود.

پشتیبان	اقلام
۲	{۱ ۳} *
۱	{۱ ٤}
۱	{۳ ٤}
۲	{۲ ۳} *
۳	{۲ ٥} *
۲	{۳ ٥} *
۱	{۱ ۲}
۱	{۱ ٥}

شکل ٤-۲) قدم دوم

پشتیبان	اقلام
۱	{۱ ۳ ٤}
۲	{۲ ۳ ٥} *
۱	{۱ ۳ ٥}

شکل ٤-۳) قدم سوم

در قدم سوم همانند قدم دوم به محاسبهٔ C_3 می‌پردازیم این اعمال را تا جایی ادامه می‌دهیم که دیگر مجموعهٔ مکرر جدیدی اضافه نشود.

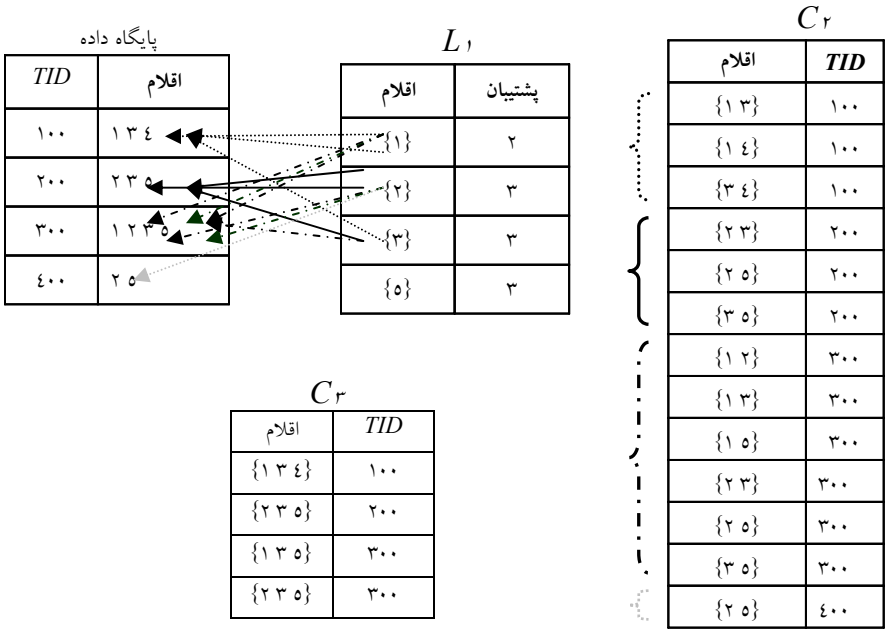
از معایب این روش این است که در هر گذر تعدادی از اقلام انتخاب شده که حداقل مقدار پشتیبان (در اینجا ۲) را نداشته و باید کنار گذاشته شوند. به‌عنوان مثال در C_2 مجموعه اقلام اضافی عبارتند از {۱,٥} , {۱,۲} , {۳,٤} , {۱,٤}

الگوریتم *SETM*

این الگوریتم توسط هوتسما[1] در سال ۱۹۹٥ ابداع شد و در سال ۱۹۹٦ نسخه دوم آن به منظور محاسبه اقلام مکرر در *SQL* توسط اسریکنت[2] مطرح شد در این الگوریتم هر یک از اعضای مجموعه به فرم <*TID,Itemset*> هستند. [۲]

[1]- Houtsma
[2]- Srikant

مشابه الگوریتم *AIS*، این الگوریتم نیز چندین گذر برروی پایگاه داده انجام می‌دهد. این الگوریتم به شکل زیر می‌باشد.



شکل٤-٤) الگوریتم AIS

گامهای این الگوریتم به قرار زیر می‌باشند:

پشتیبان هر یک از اقلام به‌طور مجزا محاسبه و بزرگ‌ترین آنها انتخاب می‌شوند. اقلام کاندید (C_K) با گسترش هر یک از این قلم‌های مکرر به سایر اقلام در هر تراکنش ساخته می‌شوند. علاوه برآن در این مرحله *TID*های مربوط به هر یک از C_K را در یک ساختار ترتیبی به نام $C_۲$ نگهداری کرده و سپس پشتیبان هر یک از C_K ها با جمع‌کردن تعداد تکرار آنها در مرحلهٔ قبل محاسبه شده و $C_۳$ ساخته می‌شود. این مراحل ادامه پیدا کرده تا جایی‌که دیگر مجموعهٔ مکرر جدیدی اضافه نشود. عمده‌ترین معایب این الگوریتم ناشی از تعداد C_K ها است و از آنجایی‌که مقدار *TID* هر C_K نگهداری می‌شود، فضای بیشتری اشغال می‌شود.

معایب الگوریتمهای *SETM* و*AIS*

- این الگوریتمها خیلی کند هستند.
- اقلام زیادی با «پشتیبانی» پایین‌تر از حداقل پشتیبان در نظرگرفته شده توسط کاربر، تولید می‌کنند.

الگوریتم *Apriori*

این الگوریتم در سال ۱۹۹۶ توسط چیونگ[1] ابداع شد و یکی از مهم‌ترین یافته‌ها در تاریخ استخراج قواعد التزامی است. در این الگوریتم از این حقیقت که همه زیرمجموعه‌های اقلام مکرر، خود نیز مکرر هستند و اقلام باید بر مبنای قاعده ترتیب الفبا مرتب باشند، استفاده شده است. تفاوت اساسی این الگوریتم با الگوریتمهای دیگر در روش محاسبه اقلام C_k و گزینش آنها برای مراحل بعدی است. در الگوریتمهای دیگر اقلام مکرر با گسترش به هر یک از اقلام مجزا (که ممکن است خودشان مکرر نباشند) در هر یک از تراکنشها ایجاد می‌شدند تا C_kها را تولید کنند و به این ترتیب C_kهای زیادی تولید شده که باید در مراحل بعدی کوچک می‌شدند و پایگاه داده چندین بار پیموده می‌شد، در حالی‌که این الگوریتم پایگاه داده را فقط یک‌بار می‌پیماید واقلام مکرر را پیدا می‌کند.

الگوریتم*Apriori* این موضوع مهم را مدنظر قرار می‌دهد و C_k ها را با اتصال اقلام مکرر حاصل از فاز قبلی و حذف آنهایی که در فاز قبلی بوده‌اند، بدون توجه به هر یک از تراکنشها به‌طور مجزا تولید می‌کند. بدین ترتیب تعداد C_kهای اضافی به‌طورچشمگیری کاهش می‌یابند.

تذکر: C_kها در این الگوریتم مطابق الگوریتم زیر محاسبه می‌شود.

---

[1]- Cheung

```
Apriori-gen(Lk-1)
Join step
insert into Ck
select p. item1, p. item2,. . . , p. itemk-1, q. itemk-1
  from Lk-1 p, Lk-1 q
    where p. item1= q. item1,. . . , p. itemk-2= q. itemk-2,
    Prune step
    p. itemk-1< q. itemk-1
for all item sets c ∈ Ck do
for all (k-1)-subsets s of c do
        if ( s ∉ Lk-1 ) then
        delete c from Ck;
```

شکل ٤-٥) الگوریتم Apriori

مثال ۱: فرض کنید مجموعه $L_۳$ به‌صورت زیر باشد:

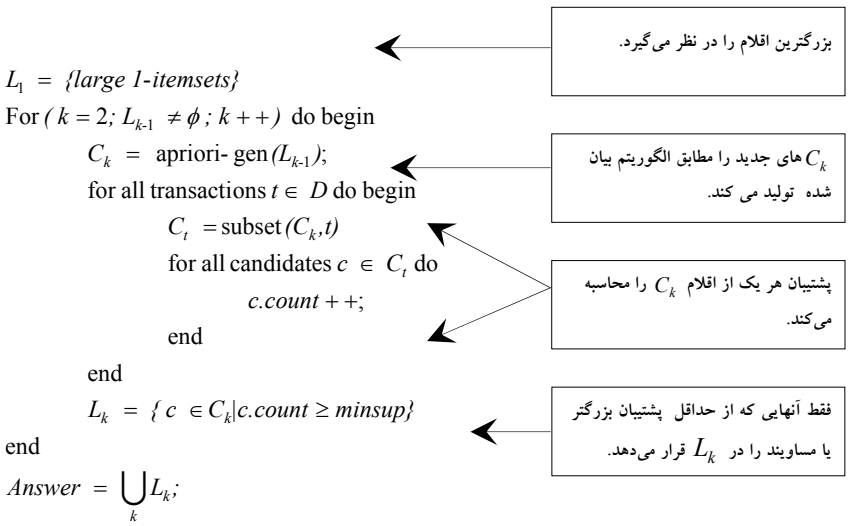
$L_۳ = \{ \{۱ ۲ ۳\}, \{۱ ۲ ٤\}, \{۱ ۳ ٤\}, \{۱ ۳ ٥\}, \{۲ ۳ ٤\} \}$

پس از مرحله اتصال خواهیم داشت:

$\{\{۱ ۲ ۳ ٤\}, \{۱ ۳ ٤ ٥\}\}$

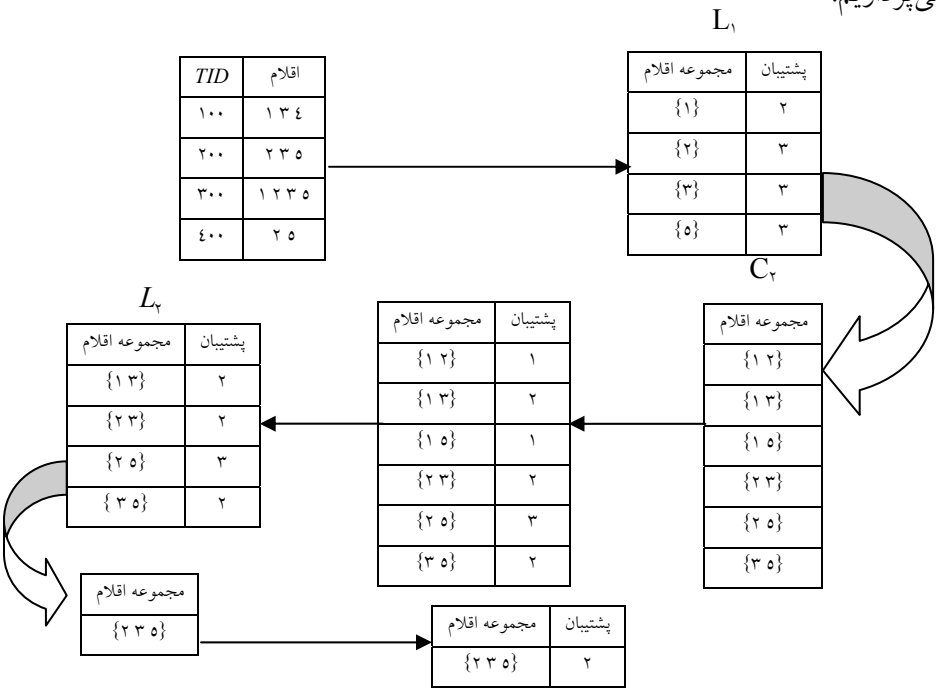
و پس از مرحله هرس خواهیم داشت:

$C_٤ = \{۱ ۲ ۳ ٤\}$



شکل ٤-٦) توضیح الگوریتم Apriori

برای ساختن $L_۱$، پشتیبان اقلام تکی محاسبه می‌شود. در قدم بعد $C_۲$ بر اساس اقلام دوتایی ترکیب شده از $L_۱$ ساخته می‌شود. در زیر با مثالی به‌بررسی این الگوریتم می‌پردازیم:



شکل ٤-۷) توضیح الگوریتم Apriori با ارائه یک مثال

پشتیبان هر کدام از اقلام موجود در $L_۲$ محاسبه شده و اقلامی که پشتیبان آنها کمتر از حداقل پشتیبان است، حذف می‌شوند و سپس $L_۲$ محاسبه می‌شود. در قدم بعد $C_۳$ بر اساس اقلام ۳ تایی از جدول $L_۲$ مطابق قدم‌های زیر محاسبه می‌شود:

در ابتدا داریم: $L_۲ = \{\{۱,۳\},\{۲,۳\},\{۲,٥\},\{۳,٤\}\}$

برای محاسبه $C_۳$ فقط مجموعه‌هایی که مؤلفه اول برابر دارند انتخاب می‌شوند. به‌عنوان مثال در مجموعه $\{\{۲,٥\},\{۳\}\}$ چون ۲ در هر دو مشترک است می‌توان سه تایی جدیدی بر اساس ترکیب آنها ساخت، به‌طوری‌که بر اساس قاعده ترتیب الفبا مجموعهٔ $\{۲,۳,٥\}$ ساخته شود.

مثال ۲: فرض کنیم که اقلام خریداری شده از یک فروشگاه به‌صورت زیر ثبت شده‌اند برای پیدا کردن اقلامی که بیشتر مواقع با هم خریداری می‌شوند به‌صورت زیر عمل می‌کنیم:

حداقل پشتیبان, s=٪۳۰

حداقل اطمینان, c= ٪۶۰

جدول ٤-١) لیست اقلام خریداری شده

شماره تراکنش‌ها	اقلام خریداری شده
۱	{آب پرتقال ,لیموناد}
۲	{ آب پرتقال ,شیر ,شیشه پاک کن}
۳	{آب پرتقال ,پاک کننده ,لیموناد}
٤	{شیشه پاک کن ,لیموناد}
۵	{ چیپس ,لیموناد }

در ابتدا پشتیبان تک تک اقلام را محاسبه می‌کنیم:

جدول ٤-٢) پشتیبان اقلام خریداری شده

اقلام خریداری شده (C_1)	
اقلام	پشتیبان
آب پرتقال	٪۶۰
لیموناد	٪۸۰
شیر	٪۲۰
شیشه‌پاک‌کن	٪٤۰
پاک‌کننده	٪۲۰
چیپس	٪۲۰

با توجه به حداقل پشتیبان یکسری از اقلام حذف می‌شوند:

جدول ٤-٣) اقلام خریداری شده با حداقل پشتیبان

(L_1)	
اقلام	پشتیبان
آب پرتقال	٪٦٠
لیموناد	٪٨٠
شیشه‌پاک‌کن	٪٤٠

مجموعه دوبعدی اقلام را در نظر گرفته و پشتیبان آنها را محاسبه می‌کنیم:

جدول ٤-٤) محاسبه پشتیبان مجموعه دوبعدی اقلام

(C_2)	
اقلام	پشتیبان
{ آب پرتقال ، لیموناد}	٪٤٠
{ آب پرتقال، شیشه پاک‌کن }	٪٢٠
{ شیشه پاک‌کن، لیموناد }	٪٢٠

با توجه به حداقل پشتیبان یکسری از اقلام حذف می‌شوند:

جدول ٤-٥) مجموعه دوبعدی اقلام با حداقل پشتیبان

(L_2)	
اقلام	پشتیبان
{ آب پرتقال ،لیموناد }	٪٤٠

قواعدی که می‌توان استخراج کرد به قرار زیر می‌باشند:

(٦٦. ٦٧ ٪= اطمینان) لیموناد⇒ آب پرتقال

(٥٠٪ = اطمینان) آب پرتقال⇒ لیموناد

اما با توجه به اینکه طبق فرضیات مسئله ٦٠٪ = حداقل اطمینان در نظر گرفته شده است، بنابراین تنها قاعده اول یعنی قاعده زیر پذیرفته می‌شود.

(٦٦. ٦٧٪ = اطمینان) لیموناد ⇒ آب پرتقال

همان‌طورکه مشاهده شد، تفاوت عمدهٔ این الگوریتم با الگوریتمهای دیگر در حجم محاسبات کمتر آن است. در این الگوریتم اقلام زاید کمتری در هر مرحله ایجاد شده و با آزمایشهای مختلفی که برای کشف اقلام مکرر توسط *IBM RS/٦٠٠٠* انجام شد مشخص شد که این الگوریتم عملکرد بسیار بهتری نسبت به دیگر الگوریتم قبلی دارد.

معایب الگوریتم

برای محاسبه پشتیبان اقلام کاندیدا، الگوریتم همه تراکنشها را بررسی می‌کند و بنابراین نیازمند زمان زیادی است.

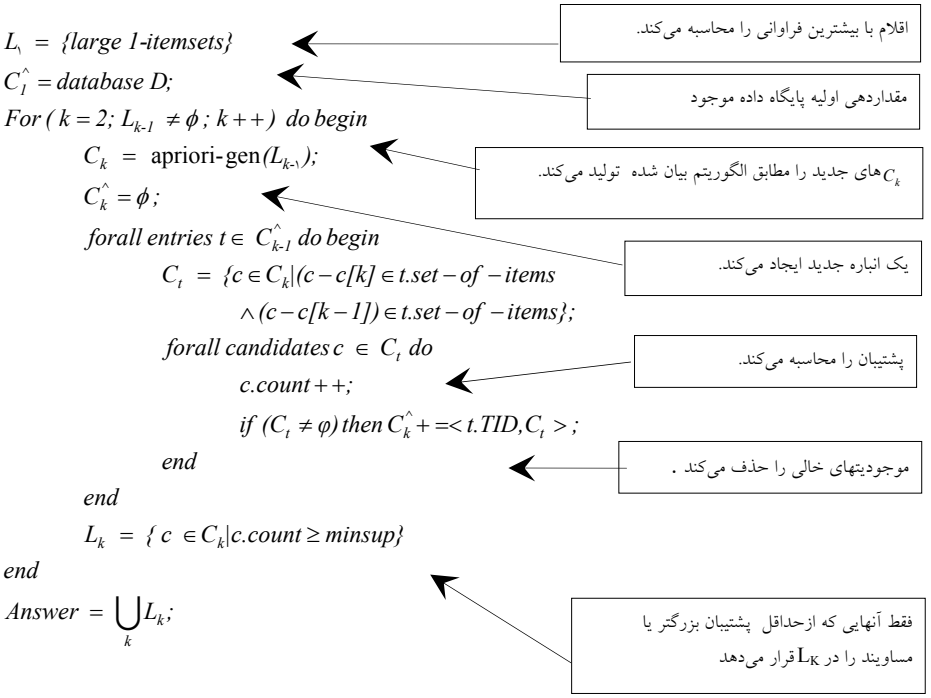
الگوریتم *AprioriTid*

همان‌گونه که قبلاً نیزذکرشد الگوریتم *Apriori* در هر گذر همه پایگاه‌داده را می‌پیماید تا پشتیبانها را محاسبه کند و پیمودن همه پایگاه داده ممکن است در همه فازها مورد نیاز نباشد. بر مبنای این مشکل، الگوریتم دیگری بنام *AprioriTid* ابداع شد. این الگوریتم نیز روشی مشابه با الگوریتم *Apriori*، برای محاسبه C_kها درهر فاز به کار می‌برد. تفاوت عمده‌ای که این الگوریتم با الگوریتم *Apriori* دارد در این است که این الگوریتم کل پایگاه داده را برای محاسبه پشتیبان بعد از مرحله اول نمی‌پیماید و از مجموعه $C_k^{\wedge}$ برای محاسبه پشتیبان استفاده می‌کند. مشابه الگوریتم *SETM* اعضای این الگوریتم نیز به فرم $<TID,X_k>$ ذخیره می‌شوند.

مزایای الگوریتم: از مزایای عمده این روش این است که در فازهای آخر اندازه $C_k^{\wedge}$ بسیار کوچک‌تر از کل اندازه پایگاه داده شده و باعث صرفه‌جویی در زمان می‌شود. این الگوریتم از نظر عملکرد نیز بر الگوریتمهای *SETM* و *AIS* برتری دارد. مشکلی که ممکن است وجود داشته باشد مدیریت حافظه است و دیده می‌شود که این الگوریتم در فازهای انتهایی (اندازه $C_k^{\wedge}$ کوچک‌تر می‌شود) عملکرد بهتری نسبت به الگوریتم *Apriori* دارد.

معایب الگوریتم: در فازهای اولیه $C_k^{\wedge}$های تولید شده بزرگ بوده و فضای زیادی اشغال می‌کنند. بنابراین مدت زمانی معادل زمان الگوریتم *Apriori* را نیازمند است. اگر

فضای اشغال شده بیشتر از حافظه در دسترس باشد، هزینهٔ اضافه‌ای را نیز در برخواهد داشت.



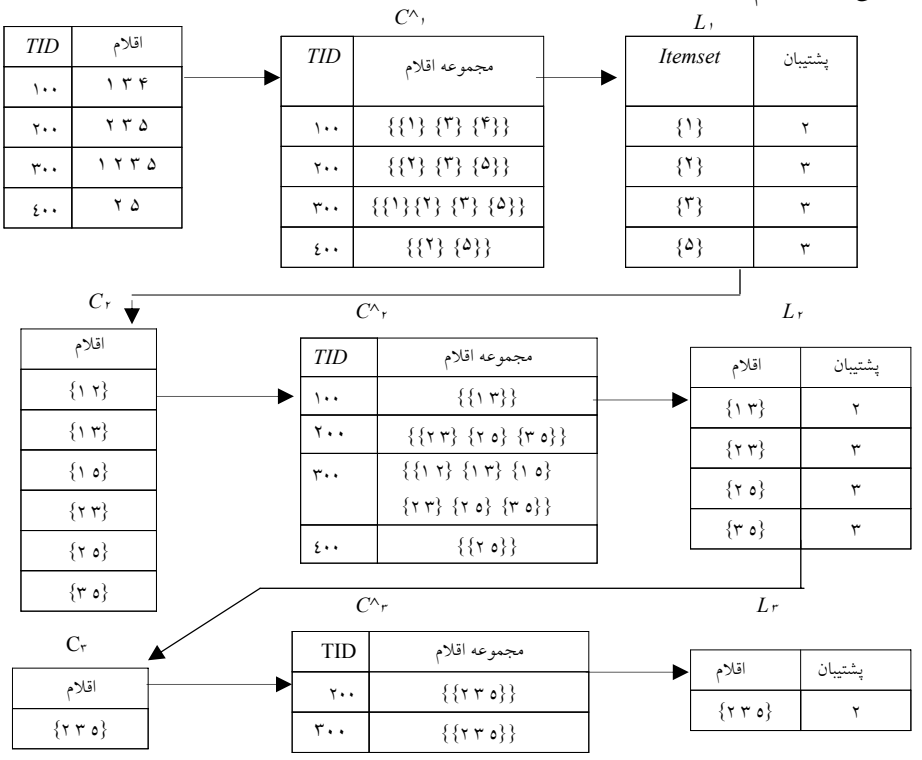
شکل ٤- ٨) الگوریتم AprioriTid

در شکل (٤-٩) L_1 بر اساس پایگاه دادهٔ اصلی و قلم کالاهای تکی به‌دست می‌آید. البته اقلامی که پشتیبان آنها کمتر از حداقل است، حذف می‌شوند. در $\hat{C_k}$ تمامی اقلام تکی ساخته شده بر اساس پایگاه دادهٔ اصلی با ذکر شمارهٔ تراکنش TID بیان می‌شوند.

در جدول C_2 مجموعه‌های دو تایی از اقلام موجود در جدول L_1 ساخته می‌شوند و در جدول $\hat{C_2}$ شمارهٔ TID این مجموعه اقلام نیز ذکر می‌شوند. در مجموعهٔ L_2، پشتیبان این مجموعه اقلام محاسبه شده و آنهایی که از حداقل کمتر هستند حذف می‌شوند. جدول C_3 بر اساس قاعده ترتیب الفبا و بر مبنای داده‌های جدول L_2،

مجموعه اقلام سه‌تایی‌ها را می‌سازد به‌عنوان مثال از ترکیب دو مجموعه {۲,۳} و {۲,۵} مجموعهٔ سه تایی {۲,۳,۵} ساخته می‌شوند.

در جدول $C_3^{\wedge}$ شماره *TID*های این مجموعه اقلام بیان شده و در جدول L_3، پشتیبان این مجموعه اقلام محاسبه می‌شود و این الگوریتم در اینجا خاتمه می‌یابد چرا که دیگر نمی‌توان اقلام جدیدی به‌دست آورد.



شکل ٤- ۹) توضیح الگوریتم AprioriTid با یک مثال

تحلیل عملکرد الگوریتمها

تفاوت عمده الگوریتمهایی که در فوق آمده‌اند، در روش تولید اقلام مکرر (L) می‌باشد. عملکرد الگوریتمها در دو نوع داده شامل داده‌های آزمایشی و داده‌های واقعی با

یکدیگر مقایسه شده‌اند. [۲] پارامترهای به‌کاررفته به‌منظور مقایسه این الگوریتمها به قرار زیر می‌باشند:

D: تعداد تراکنشها

T: میانگین اندازه تراکنشها

I: میانگین اندازه اقلام مکرر

L: تعداد اقلام مکرر

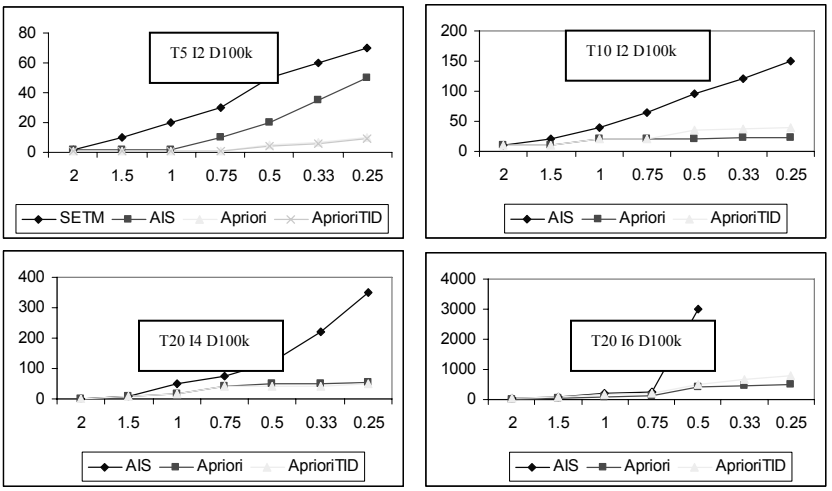
N: تعداد اقلام

K:۱۰۰۰

T5. I2. D100k =>*T=5, I=2, D=100,000*
T10. I2. D100k
T10. I4. D100k
T20. I2. D100k
T20. I4. D100k
T20. I6. D100k

تعداد اقلام = ۱۰۰۰

نمادی بالای هر کدام از نمودارها نوشته شده است که معرف تراکنشها، میانگین اندازهٔ اقلام مکرر و میانگین اندازهٔ تراکنشها می‌باشد به‌عنوان نمونه *T5I2D100K* عبارت است از D=۱۰۰۰۰۰ و I=۲ و T=۵ و به این معنی است که آزمایش برای تعداد تراکنشهای ۱۰۰۰۰۰ و میانگین اندازهٔ اقلام مکرر ۲ و میانگین اندازهٔ تراکنشهای ۵ انجام شده است. محور افقی نیز حداقل پشتیبان است. آزمایشهای مختلفی برای نمونه‌های متفاوت انجام شده است و نتایج حاصله در نمودارهای زیر آمده‌اند. البته زمانهای ناشی از اجرای الگوریتم *SETM* آنقدر زیاد بوده‌اند که نتوانسته‌اند در نمودارهای زیر بگنجند.



شکل ٤-١٠) تغییرات رفتار الگوریتمهای مختلف

با دقت در این نمودارها درمی‌یابیم که: الگوریتم *Apriori* همواره بر الگوریتم *AIS* غالب است و *Apriori* در اندازه‌های بزرگ بهتر از *AprioriTid* عمل می کند.

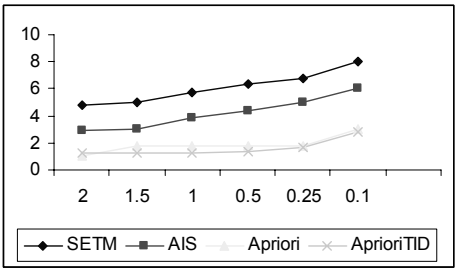
درالگوریتم *AprioriTid* مقادیر $C_k^{\wedge}$ بجای پایگاه داده در نظر گرفته می‌شوند. اگر $C_k^{\wedge}$ بتواند در حافظه جای گیرد، این الگوریتم سریعتر از *Apriori* عمل خواهد کرد.

زمانیکه $C_k^{\wedge}$ خیلی بزرگ باشد، نمی‌تواند در حافظه جای بگیرد و در نتیجه زمان محاسبه بسیار بالا می‌رود، بنابراین الگوریتم *Apriori* سریعتر از الگوریتم *AprioriTid* عمل خواهد کرد.

داده‌های واقعی

فروشگاه خرده‌فروشی شامل:

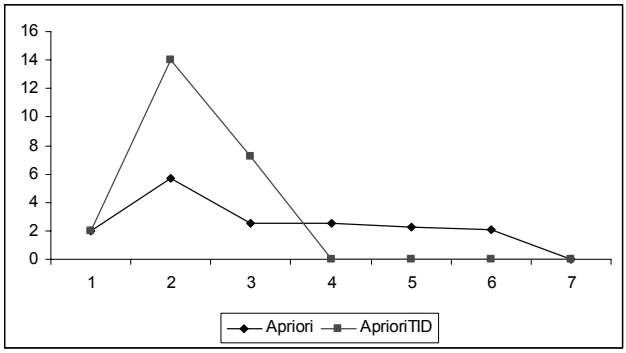
- ٦٣ بخش
- ٤٦٨٧٣ تراکنش(با میانگین اندازه ٤٧/٢)



شکل ٤-١١) تغییرات رفتار الگوریتمهای مختلف در یک فروشگاه خرده فروشی

همان‌گونه که مشاهده می‌شود در اینجا اندازه پایگاه داده کوچک است و بنابراین $C_k^{\wedge}$ مشکلی با حافظه نخواهند داشت و در نتیجه الگوریتم *AprioriTid* در زمان کمتری نسبت به الگوریتم *Apriori* اجرا می‌شود. بنابراین کدامیک بهتراست؟ *AprioriTid* یا *Apriori*

به منظور پاسخ به این سؤال مقایسه‌ای بین این دو الگوریتم در طی فازهای مختلف صورت گرفته است که نتایج آن در شکل زیرآمده است.



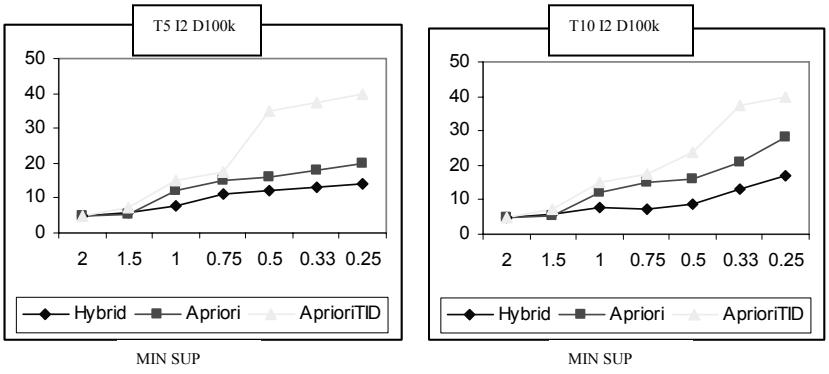
شکل ٤-١٢) مقایسه رفتار الگوریتمهای AprioriTid و Apriori

درمراحل انتهایی $C_k^{\wedge}$ به اندازه کافی کوچک شده و حافظه مصرفی کم می‌شود. بنابراین از فاز ٤ به بعد زمان اجرای الگوریتم *AprioriTid* بسیار کم شده و تقریباً این زمان برابر صفر شده است. به منظوراستفاده بهینه از این‌دو الگوریتم، الگوریتم جدیدی بنام *AprioriHybrid* شکل گرفت.

الگوریتم *Apriori Hybrid*

خصوصیات این الگوریتم به ترتیب زیر است:

- این الگوریتم درفازهای اولیهٔ اجرا مطابق الگوریتم *Apriori* عمل می‌کند.
- اندازه تخمینی $C_k^{\wedge}$ به‌صورت زیر محاسبه می‌شود:
- تعداد تراکنشها + حاصل‌جمع پشتیبان همهٔ اقلام = اندازه تخمینی $C_k^{\wedge}$
- وقتی که $C_k^{\wedge}$ها به اندازهٔ کافی کوچک شده و حافظهٔ مصرفی کم می‌شود به الگوریتم *AprioriTid* سوئیچ کرده و مطابق این الگوریتم پیش می‌رود.
- اگرچه تغییر از *Apriori* به *AprioriTid* زمان‌براست، اما در بسیاری از موارد نتایج مثبتی دارد. در نمودارهای زیر عملکرد سه الگوریتم اخیر با یکدیگر مقایسه شده است. در تمامی این نمودارها نشان داده شده است که الگوریتم ترکیبی زمان اجرای کمتری نسبت به *Apriori* و *AprioriTid* دارد.



شکل ٤-١٣) مقایسه عملکرد الگوریتمهای AprioriTid و Apriori در آزمایشهای مختلف

منابع

1) *Han. J, Kamber. M. (2006) "Chapter 5: Mining Frequent Patterns, Associations, and Correlations", Data mining concepts and techniques, 2nd edition, , Morgan Kaufmann Publishers.*

2) *R.Agrawal R.Srikant, Fast algorithm for mining association rules,1998*

فصل پنجم

دسته‌بندی و پیش‌بینی

دسته‌بندی و پیش‌بینی دو نوع عملیات برای تحلیل داده‌ها و استخراج مدل به‌منظور توصیف دسته‌های مهم داده‌ها و پیش‌بینی رفتار آیندهٔ آنها می‌باشند. هدف این‌گونه تحلیلها کمک به فهم بهتر رفتار آینده داده‌ها می‌باشد. دسته‌بندی در پیش‌بینی داده‌های گسسته و طبقه‌ای، نقش داشته و مدلهای پیش‌بینی یا رگرسیون بیشتر برروی داده‌های پیوسته کار می‌کنند به‌عنوان مثال یک مدل دسته‌بندی ممکن است برای دسته‌بندی کردن وامهای بانک به دو طبقهٔ وامهای بی‌خطر و پرخطر، به‌کار رود درحالی‌که مدلهای پیش‌بینی به کار گرفته شده در این کسب و کار خاص، سعی در پیش‌بینی میزان مخارج و هزینه‌های مشتریان براساس ویژگیهای درآمدی و شغلی آنها دارند.

مفاهیم دسته‌بندی

بسیاری از روشهای دسته‌بندی و پیش‌بینی در علومی مانند یادگیری ماشینی، بازشناسی الگو و آمار کاربرد دارند. در این فصل به روشهای سادهٔ دسته‌بندی از قبیل ساخت درختهای تصمیم، شبکه‌های عصبی، نزدیکترین همسایگی و دیگر روشها اشاره شده است. دسته‌بندی یعنی تخصیص یک برچسب به مجموعه‌ای ازداده‌ها که هنوز دسته‌بندی نشده‌اند. دسته‌بندی عبارت است از تخصیص داده‌ها بر اساس ویژگیهایشان به دسته‌هایی که نام آنها از قبل مشخص می‌باشد. داده‌ها دارای k ویژگی[1] هستند که به‌صورت $A_1,...,A_k$ نشان داده می‌شود. هرمورد یا مثال به وسیله مقادیر ویژگیها و یک برچسب دسته، توصیف می‌شود. دسته‌بندی برای یادگیری قواعد و یا ساختن مدلی به‌منظور پیش‌بینی دستهٔ داده‌های جدید به‌کار می‌رود. داده‌های مورد استفاده برای ساختن مدل، داده‌های آموزش یا داده‌های تربیت مدل نامیده می‌شوند.

تفاوت دسته‌بندی و خوشه‌بندی

دسته‌بندی، هر جزء از داده‌ها را بر مبنای اختلاف بین داده‌ها به مجموعه‌های از پیش تعریف شدهٔ دسته‌ها تصویر می‌کند. درحالی‌که خوشه‌بندی، داده‌ها را به گروه‌های مختلف (خوشه‌ها) که از قبل معین نیستند، (براساس مشابهت آنها در یک خوشه و تفاوت بین اعضای دو خوشه) تقسیم می‌کند. لذا اگر بخواهیم با استفاده از مفهوم یادگیری، دسته‌بندی و خوشه‌بندی را متمایز کنیم، یادگیری با نظارت در مقابل یادگیری بدون نظارت مطرح می‌شود.

یادگیری با نظارت یا دسته‌بندی عبارتست از یادگیری به‌وسیله نمونه‌ها. به‌عبارت دیگر دسته‌بندی یک‌نوع یادگیری با نظارت از نمونه‌ها می‌باشد، و دسته‌ها از قبل مشخص

---

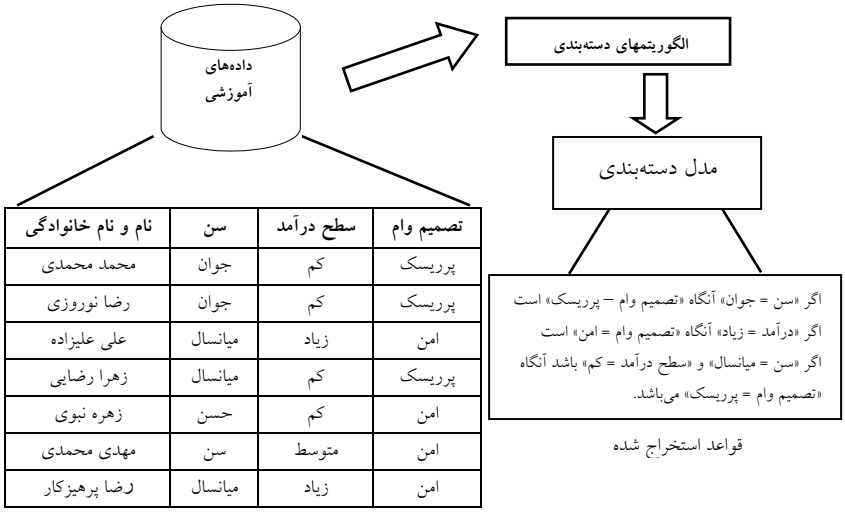
[1]- Attribute

هستند. ولی در یادگیری بدون نظارت یا خوشه‌بندی، خوشه‌ها مشخص نیستند و هدف خوشه‌بندی، تعیین خوشه‌های داده‌ها است.

فرایند دو مرحله‌ای دسته‌بندی

دسته‌بندی داده‌ها، فرآیندی دو مرحله‌ای است. اولین مرحله ساخت مدل و دومین مرحله استفاده از مدل و پیش‌بینی از طریق داده‌های قبلی می‌باشد.[۱]

مرحله اول یا ساخت مدل عبارت است از: توصیف یک سری از دسته‌های از پیش تعیین شده بر مبنای مجموعه داده‌های آموزش مدل که البته این فرایند، یادگیری نیز نامیده می‌شود. در این فرایند سعی می‌شود با توجه به نمونه‌های موجود، مدلی ساخته شود که براساس آن بتوان داده‌های فاقد مشخصهٔ دسته را در دسته‌های مربوط به خودشان قرار داد. البته فرض می‌شود که هر نمونه به یکی از دسته‌های از پیش تعریف شده تعلق دارد و در نهایت مدل به‌صورت قواعد دسته‌بندی، قابل ارائه است. البته مدل به شکلهای غیر از قواعد نیز قابل بازنمایی است.



نام و نام خانوادگی	سن	سطح درآمد	تصمیم وام
محمد محمدی	جوان	کم	پرریسک
رضا نوروزی	جوان	کم	پرریسک
علی علیزاده	میانسال	زیاد	امن
زهرا رضایی	میانسال	کم	پرریسک
زهره نبوی	حسن	کم	امن
مهدی محمدی	سن	متوسط	امن
رضا پرهیزکار	میانسال	زیاد	امن

شکل ۵-۱) یک نمونه از ساخت مدل بر اساس داده‌های قدیمی

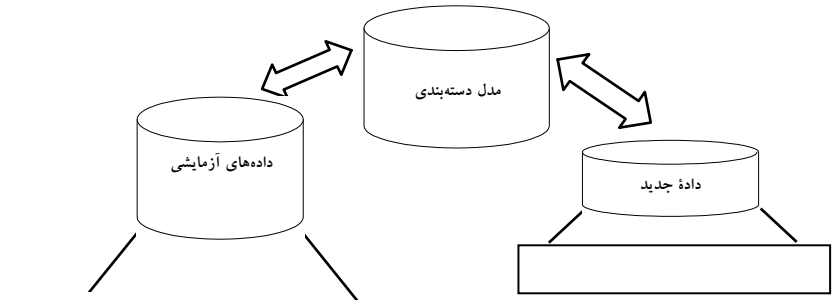
در شکل (۵-۱) مدل جدیدی بر اساس داده‌های قدیمی ساخته شده و در آن بیان می‌شود که آیا وام دادن به مشتریان بی‌خطر است یا خیر؟ که البته بی‌خطر یا پرخطر بودن وام‌دهی به مشتریان بر اساس ویژگیهای دیگر آنها فرموله شده و نهایتاً در مدل به صورت یک‌سری قواعد اگر - آنگاه ارائه می‌شود. اولین مرحله از فرآیند تصمیم‌گیری می‌تواند به‌عنوان یادگیری یک تابع نگاشت $y = f(x)$ درنظرگرفته شود که در این تابع نگاشت هر دادهٔ x به یک کلاس y اختصاص دارد. هدف دسته‌بندی، یادگیری این تابع نگاشت می‌باشد تا بتوان به‌راحتی کلاس هر داده را پیدا کرد. در مثال بالا این نگاشت به‌صورت قواعد دسته‌بندی بیان می‌شود که تعیین‌کنندهٔ پرخطر و یا بی‌خطر بودن وام دادن به مشتریان می‌باشد.

مرحلهٔ دوم استفاده از مدل: در این مرحله، مدل ساخته شده برای دسته‌بندی استفاده شده ودرابتدا دقت پیش‌بینیِ مدل تخمین زده می‌شود. به‌همین منظور مجموعه‌ای از داده‌های آزمایشی به‌طور اتفاقی از داده‌ها انتخاب شده و مدلسازی با کمک آنها انجام می‌شود. در واقع پس از ساخته شدن مدل، باید از آن برای دسته‌بندی داده‌های جدید استفاده کرد. استفاده در اینجا به معنی تعیین دسته‌بندی داده‌های آینده است. تابع هدف در اینجا بالاتر بردن تخمین دقت مدل تا حد امکان می‌باشد. در مرحله استفاده از مدل موارد ذیل قابل توجه می‌باشند.

برچسب شناخته شده از نمونه آزمون با نتایج دسته‌بندی مقایسه می‌شود. [۱] دقت مدل، درصد تعداد دفعاتی است که نمونه‌های آزمایشی با موفقیت دسته‌بندی می‌شوند. اگر دقت مدل قابل قبول باشد می‌توان مدل را برای دسته‌بندی داده‌هایی که دسته آنها مشخص نیستند، به‌کار برد. شکل (۵-۲) مراحل استفاده از مدل را نشان می‌دهد.

---

[۱] - داده‌های آموزشی را می‌توان به دو قسمت تقسیم کرد: اول آن دسته که مدل بر اساس آنها ساخته می‌شود و دوم گروهی برای ارزیابی مدل. این دسته دوم با دسته‌های مشخص توسط مدل دسته‌بندی شده و نتایج آن با دسته‌های داده‌ها مقایسه شده و دقت کل مدل استخراج می‌شود.



نام و نام خانوادگی	سن	سطح درآمد	تصمیم وام
لیلا زندی	سن	کم	امن
زهرا نوروزی	میانسال	کم	پرریسک
مریم محمدی	میانسال	زیاد	امن

شکل ۵-۲) دسته‌بندی داده‌های وام

روشهای مختلف دسته‌بندی

روشهای زیادی برای دسته‌بندی وجود دارد که از آن جمله می‌توان به موارد ذیل اشاره کرد:

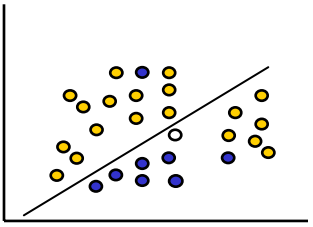
- درخت تصمیم
- نزدیک‌ترین همسایگی
- بیز ساده و شبکه‌های بیزی
- شبکه‌های عصبی
- رگرسیون (خطی، غیرخطی، لجستیک)

رگرسیون خطی

معادله زیر را در نظر بگیرید:

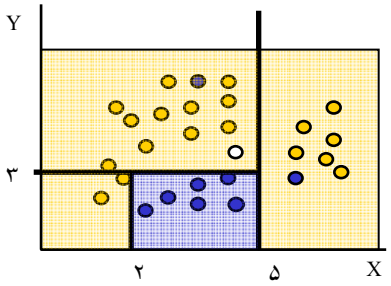
$$w_0 + w_1 x + w_2 y \geq 0 \qquad (۱-۵)$$

رگرسیون w_i را از داده‌ها به نحوی محاسبه می‌کند که مجموع مربعات خطا حداقل شود. این روش به اندازه کافی منعطف نیست.



شکل ۵-۳) رگرسیون خطی

درخت تصمیم



if $X >$ ۵ *then blue*

else if $Y >$ ۳ *then blue*

else if $X >$ ۲ *then green*

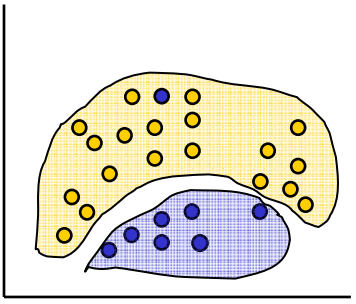
else blue

شکل ۵-٤) درخت تصمیم

درخت تصمیم فضا را به نواحی مستطیلی تقسیم می‌کند. به‌طوری‌که در هر مستطیل داده‌ها از نظر برچسب دسته همگن باشند.

شبکه‌های عصبی

با این روش می‌توان نواحی با اشکال پیچیده را پوشش داد. این روش دقیق‌تر از سایر روشهاست و کاملاً می‌تواند برازش شود.



شکل ۵-۵) شبکه‌های عصبی

روش دسته‌بندی بیزی

در اینجا برای بررسی چگونگی انجام دسته‌بندی بیزی، از تئوری اولیه بیز شروع می‌کنیم. یادگیری احتمالی: یادگیری احتمالی می‌تواند معادل محاسبه $P(C=c\,|\,d)$ باشد، برای مثال احتمال اینکه یک داده نمونه d در کلاس c قرارگیرد، چیست؟

بیز ساده

فرض کنید A_1 تا A_k ویژگیهایی با مقادیر گسسته باشند، این مقادیر برای پیش‌بینی یک کلاس گسسته C به‌کار می‌روند. نمونه‌ای با مقادیر ویژگی مشاهده شده a_1 تا a_k را در نظر بگیرید. هدف پیش‌بینی و انتخاب دسته‌ای است که $P(C=c|A_1=a_1\cup A_2=a_2\ldots\cup A_n=a_n)$ ماکزیمم شود.

فرمول ساده بیز عبارت است از:

$$P(A/B)=\frac{P(B/A)P(A)}{P(B)} \qquad (۲-۵)$$

در فرمول: $P(C=c|A_1=a_1\cup A_2=a_2\ldots\cup A_n=a_n)$ با استفاده از قاعده بیزین داریم:

$$P(C=c|\ A_1=a_1\cup A_2=a_2\ \ldots\cup A_n=a_n)$$
$$=\frac{P(A_1=a_1\cup A_2=a_2\cup\ldots A_n=a_n\,|\,C=c)\cdot P(C=c)}{P(A_1=a_1\cup A_2=a_2\cup A_n=a_n)} \qquad (۳-۵)$$

در این فرمول $P(C=c)$ به سادگی از داده‌های آموزش مدل قابل استخراج است. $P(A_1=a_1\cup A_2=a_2\cup A_n=a_n)$ برای تصمیم‌گیری بی‌تأثیر است زیرا که برای همه مقادیر c یکسان است.

پس فقط لازم است که مقدار $P(A_1=a_1\cup A_2=a_2\cup\ldots A_n=a_n\,|\,C=c)$ محاسبه شود. از طرفی با فرض استقلال داریم:

$$P(X\mid C_i)=\prod_{k=1}^{n}P(X_k\,|\,C_i)=P(X_1\,|\,C_i)*P(X_2\,|\,C_i)\ldots*P(X_n\,|\,C_i) \qquad (٤-۵)$$

بنابراین رابطه $P(A_1 = a_1 \cup A_2 = a_2 \cup ... A_n = a_n \mid C = c)$ به‌همین ترتیب و با همین منطق قابل گسترش است و داریم:

$$P(A_1 = a_1 \cup ... \cup A_k = a_k \mid C = c) = P(A_1 = a_1 \mid C = c) \times ... \times P(A_k = a_k \mid C = c)$$

فرضی که در بیز ساده وجود دارد این است که ویژگیها به‌طور شرطی از هم مستقل هستند. فرض می‌کنیم که برای یک دسته C همه ویژگیها به‌طور شرطی از هم مستقل هستند و در نهایت به‌طور کلی فرض می‌کنیم که:

$$P(A_1 = a_1 \cup ... \cup A_k = a_k \mid C = c) = P(A_1 = a_1 \mid C = c) \times ... \times P(A_k = a_k \mid C = c)$$

و به‌همین ترتیب برای A_1 تا A_k نیز همین فرض برقرار است. حال می‌خواهیم $P(A_1 = a_1 \mid C = c)$ را تخمین بزنیم. مثال زیر به این مسئله کمک می‌کند.

در این مثال، ویژگی داده‌ها عبارتند از: سن، سطح درآمد، دانشجویی و میزان اعتبار. داده‌های آموزشی در جدول زیر آمده است:

جدول ۵-۱) داده‌های خریداران کامپیوتر

خریدار کامپیوتر	اعتبار	دانشجو	درآمد	سن
خیر	بد	خیر	بالا	جوان
خیر	عالی	خیر	بالا	جوان
بلی	بد	خیر	بالا	میانسال
بلی	بد	خیر	متوسط	بالای ۴۰ سال
بلی	بد	بلی	کم	بالای ۴۰ سال
خیر	عالی	بلی	کم	بالای ۴۰ سال
بلی	عالی	بلی	کم	میانسال
خیر	بد	خیر	متوسط	جوان
بلی	بد	بلی	کم	جوان
بلی	بد	بلی	متوسط	بالای ۴۰ سال
بلی	عالی	بلی	متوسط	جوان
بلی	عالی	خیر	متوسط	میانسال
بلی	بد	بلی	بالا	میانسال
خیر	عالی	خیر	متوسط	بالای ۴۰ سال

برچسب دسته عبارتست از: خرید کامپیوتر که دو مقدار مجزای {خیر و بلی} دارد و بنابراین داریم:

دسته اول: C_1 = «بلی = خرید کامپیوتر»

دسته دوم: C_2 = «خیر = خرید کامپیوتر»

داده‌ای که قرار است دسته‌اش تشخیص داده شود، عبارتست از:

X = (بد= میزان اعتبار , بله = دانشجو , متوسط = سطح درآمد , جوان = سن)

بدین منظور نیاز است که $P(X|C_i)P(C_i)$ حداکثر شود. $P(C_i)$ احتمال قبلی هر دسته است که براساس داده‌های آموزشی قابل محاسبه است و داریم:

P(خریدار کامپیوتر = بله) $= \frac{۹}{۱۴} = ۰/۶۴۳$

P(خریدار کامپیوتر = خیر) $= \frac{۵}{۱۴} = ۰/۳۵۷$

برای محاسبهٔ $P(X|C_i)$ برای $i = ۱,۲,...$ داریم:

P (جوان = سن | خریدار کامپیوتر = بلی) $= \frac{۲}{۹} = ۰/۲۲۲$

P (جوان = سن | خریدار کامپیوتر = خیر) $= \frac{۳}{۵} = ۰/۶۰۰$

P (سطح درآمد = متوسط | خریدار کامپیوتر = بلی) $= \frac{۴}{۹} = ۰/۴۴۴$

P (سطح درآمد = متوسط | خرید کامپیوتر = خیر) $= \frac{۲}{۵} = ۰/۴۰۰$

P (دانشجو = بلی | خریدار کامپیوتر = بلی) $= \frac{۶}{۹} = ۰/۶۶۹$

P(دانشجو = بلی | خریدار کامپیوتر = خیر) $= \frac{۱}{۵} = ۰/۲۰۰$

P (میزان اعتبار = بلی | خریدار کامپیوتر = بلی) $= \frac{۶}{۹} = ۰/۶۶۷$

P (میزان اعتبار = بلی | خریدار کامپیوتر = خیر) $= \frac{۲}{۵} = ۰/۴۰۰$

برای همه ویژگیهای «سن»، «سطح درآمد»، «دانشجویی» و «میزان اعتبار» احتمال‌ها را محاسبه کرده و سپس (بلی = خریدار کامپیوتر $P(X \mid$ را محاسبه می‌کنیم. برای این‌کار در موارد بالا آنهایی که برچسب دسته خریدار کامپیوتر آنها بلی است را انتخاب کرده و محاسبات را ادامه می‌دهیم، یعنی:

$P(X \mid$ خریدار کامپیوتر = بلی) =

P (سن= جوان | خریدار کامپیوتر = بلی) × P (سطح درآمد = متوسط | خریدار کامپیوتر = بلی) ×

P (دانشجو = بلی | خریدار کامپیوتر = بلی) × P (میزان اعتبار = بد | خریدار کامپیوتر = بلی) ×

در نتیجه داریم:

$P(X \mid$ خریدار کامپیوتر = بلی) = ۰/۲۲۲ × ۰/۶۶۷ × ۰/۴۴۳ × ۰/۶۶۷ = ۰/۰۴۴

در ادامه احتمال $P(x \mid$ خریدار کامپیوتر = خیر) را نیز از روش فوق محاسبه می‌کنیم. یعنی داریم:

$P(X \mid$ خریدار کامپیوتر = خیر) = ۰/۶۰۰ × ۰/۴۰۰ × ۰/۲۰۰ × ۰/۴۰۰ = ۰/۰۱۹

همان‌طور که ذکر شد، هدف حداکثر کردن $P(X \mid Ci) \cdot P(Ci)$ می‌باشد و برای این کار دو مقدار زیر را محاسبه می‌کنیم:

$P(X \mid$ خریدار کامپیوتر = بلی) P. (خریدار کامپیوتر = بلی)

= ۰/۰۴۴ × ۰/۶۴۳ = ۰/۰۲۸

$P(X \mid$ خریدار کامپیوتر = خیر) P. (خریدار کامپیوتر = خیر)

= ۰/۰۱۹ × ۰/۳۵۷ = ۰/۰۰۷

بنابراین پیش‌بینی می‌کنیم که دادهٔ جدید X در کلاس «خریدار کامپیوتر = بلی» می‌باشد.

مزایای بیز ساده

- اجرای راحت
- نتایج خوب برای بسیاری از کاربردها

معایب بیز ساده

- استقلال شرطی دسته‌ها فرضی است که در اینجا مطرح شده است اما در مواردی که این فرض برقرار نیست دقت مدل پایین است.
- در عمل وابستگی وجود دارد و فرض استقلال همواره برقرار نیست. نحوه برخورد با این وابستگیها شبکه‌های بیزی می‌باشد.

شبکه‌های بیزی

شبکه‌های بیزی وابستگیهای شرطی بین متغیرها (ویژگیها) را شرح می‌دهد. با استفاده از این شبکه‌ها دانش قبلی در زمینه وابستگی بین متغیرها با داده‌های آموزش مدل دسته‌بندی، ترکیب می‌شوند. در زیر با مفاهیم اساسی شبکه بیزی آشنا می‌شویم.

گره: گره‌ها، متغیرهایی هستند که هرکدام مجموعه مشخصی از وضعیتهای دوبه‌دو ناسازگار[1] دارند.

کمان: نشان‌دهنده وابستگیهای متغیرها به یکدیگر می‌باشند.

فرض مهم در روش بیز ساده استقلال شرطی دسته‌ها از یکدیگر می‌باشد اما در عمل این وابستگی بین متغیرها وجود دارد. شبکه‌های احتمالی بیزی این نوع احتمالها را بررسی می‌کند. یک شبکه بیزی از دو بخش گراف غیردوری و احتمالهای شرطی تشکیل شده است. اگر کمانی از گره Y به Z وصل شود، مبین این است که Y پدر Z می‌باشد. هر کمان دانش علل و معلولی بین متغیرهای مرتبط را نشان می‌دهد. به هر متغیر A با والدین $B_1, \ldots, B_n$ یک «جدول احتمالی شرطی» یا CPT[2] متصل می‌شود. در این جدول برای هر متغیر Y بر اساس ارتباط با والدینش می‌توانیم عناصر ماتریس مربوطه را محاسبه کنیم. جدول (۵-۲) بر اساس شکل (۵-۶) و احتمالهای مرتبط محاسبه شده است.

---

[1]- Mutually Exclusive
[2]- Conditional Probability Table

جدول ۵-۲) اطلاعات مربوط به ارتباط سرطان ریه و سوابق خانوادگی و کشیدن سیگار

	سوابق خانوادگی دارد سیگار می‌کشد	سوابق خانوادگی دارد سیگار نمی‌کشد	سوابق خانوادگی ندارد سیگار می‌کشد	سوابق خانوادگی ندارد سیگار نمی کشد
سرطان ریه دارد	۰/۸	۰/۵	۰/۷	۰/۱
سرطان ریه ندارد	۰/۲	۰/۵	۰/۳	۰/۹

به‌عنوان مثال برای متغیر «سرطان ریه» داریم:

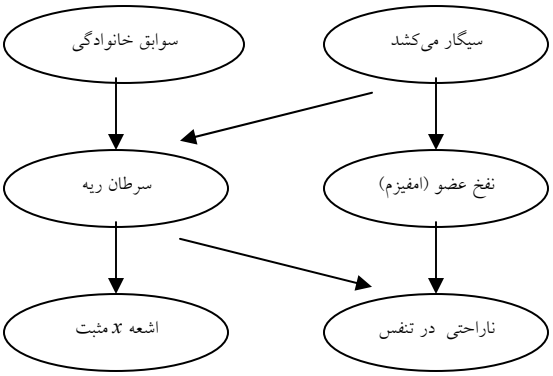
P (سرطان ریه = بله | سابقه خانوادگی دارد , بله = سیگار می‌کشد) = ۰/۸

P (سرطان ریه = خیر | سابقه خانوادگی دارد , خیر = سیگار می‌کشد) = ۰/۹

فرض کنید که $X = (x_1, \ldots x_n)$ دادهٔ جدیدی با ویژگیهای $x_1, x_2, \ldots, x_n$ باشد در این صورت معادلهٔ زیر بیانگر توزیع احتمال توأم می‌باشد.

$$P(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid parents\ (Y_i)) \qquad (۵-۵)$$

Parent (Y) والدین Y هستند.



شکل ۵- ٦) اطلاعات مربوط به ارتباط سرطان ریه و سوابق خانوادگی و کشیدن سیگار

در رابطه (۵-۵) $P(x_i, x_۲, ... x_n)$ احتمال ترکیب خاصی از مقادیر X و مقادیر مرتبط با آنها در ماتریس CPT متناظرش می‌باشد. یک گره در این گراف می‌تواند به‌عنوان گرهٔ خروجی انتخاب شده و بیانگر برچسب دسته باشد. البته دربیشتر موارد یک خروجی داریم.

چگونگی یادگیری در شبکه‌های بیزی

برای یادگیری این نوع شبکه‌ها چند سناریو وجود دارد: یکی از روشها استفاده از دانش افراد خبره در ترسیم گراف مربوطه و ماتریس CPT آن می‌باشد. افراد خبره باید احتمالات شرطی مربوط به گره‌هایی که در وابستگی مستقیم شرکت دارند را بیان کرده سپس این احتمالات در محاسبهٔ احتمالات متغیرهای دیگر استفاده شوند.
روش دیگر حدس زدن مقادیر ماتریس CPT از طریق روشهای هیوریستیک می‌باشد. با روشهای پیشرفته شبیه‌سازی و با داشتن داده کافی حتی امکان تخمین ترسیم گراف نیز وجود دارد.[۱]

دسته‌بندی بر مبنای نزدیکترین همسایه‌ها

در یک نگاه کلی می‌توان دسته‌بندیها را به دو دسته مشتاق[۱] و کاهل[۲] تقسیم کرد. در نوع مشتاق، در مرحله آموزش، مدلی از داده‌ها ساخته می‌شود. درختهای تصمیم، نمونه‌ای از دسته‌بندها هستند، که با دریافت نمونه‌های آموزشی مدلی به شکل درخت می‌سازند. نوع دیگر دسته‌بندها به کاهل معروفند. در این نوع روشها نمونه‌های آموزشی دریافت و ذخیره شده وتنها در هنگام دسته‌بندی از آنها استفاده می‌شود، در واقع تا اینجا مدلی از داده‌ها ساخته نشده و یادگیری تا زمان دسته‌بندی به تعویق می‌افتد. به این نوع دسته‌بندها، یادگیر مبتنی بر نمونه[۳] هم می‌گویند. تفاوت دو روش در این است که انواع

---

[1]- Eager
[2]- Lazy
[3]- Instance Based Learner

مشتاق زمان زیادی را در مرحلهٔ آموزش، صرف ساخت مدل کرده و در زمان دسته‌بندی بسیار سریع عمل می‌کنند، در نقطه مقابل، انواع کاهل آن، در هنگام ورود داده‌ها در مرحله آموزش، فقط آنها را ذخیره کرده و زمان بیشتری را صرف دسته‌بندی می‌کنند. هر یک از این روشها کاربرد خود را دارند که در ادامه به آنها اشاره خواهد شد. نزدیک‌ترین همسایگی[1]، روشی که در این فصل درباره آن صحبت خواهیم کرد، نمونه‌ای از دسته‌بندهای کاهل است.[۲] و[۵]

روش نزدیک‌ترین همسایگی

الگوریتم نزدیک‌ترین همسایگی از سه گام زیر تشکیل شده است:

- محاسبه فاصله نمونه ورودی با تمام نمونه‌های آموزشی.
- مرتب کردن نمونه‌های آموزشی براساس فاصله و انتخاب K همسایه نزدیکتر.
- استفاده از دسته‌ای که اکثریت را در همسایه‌های نزدیک، به‌عنوان تخمینی برای دستهٔ نمونه ورودی دارد.

قبل از ورود به جزئیات بیشتر روش نزدیک‌ترین همسایگی، برای فهم بهتر به‌بررسی یک مثال کوچک می‌پردازیم.

مثال: یک شرکت کاغذ سازی برای دریافت بازخور از مشتریانش، در یک بررسی پرسشنامه‌ای، از آنها خواست کاغذها را به دو دسته خوب و بد تقسیم کنند. این کاغذها دارای دو ویژگی مقاومت در برابر اسید و دوام هستند. جدول (۵-۳) اطلاعات به‌دست آمده از تحقیق را (به‌عنوان نمونه‌های آموزشی)ا نشان می‌دهد. سؤال این است: کارخانه، کاغذ جدیدی تولید می‌کند که تست آزمایشگاه $x_1 = ۳$ و $x_2 = ۷$ را برای آن تعیین کرده است. می‌خواهیم بدون تحقیق پرهزینه، دسته‌بندی این کاغذ را بدانیم.

جدول ۵-۳) نمونه‌های آموزشی، به‌دست آمده از تحقیق پرسشنامه‌ای از مشتریان

[1]- K Nearest Neighborhood

$x_۱$ = مقاوت در برابر اسید (*seconds*)	$x_۲$ = دوام (*kg/square meter*)	Y = *دسته‌ها*
۷	۷	بد
۷	٤	بد
۳	٤	خوب
۱	٤	خوب

در گام اولِ روش نزدیک‌ترین همسایگی، باید فاصله نمونه ورودی با تمام نمونه‌های آموزشی محاسبه شود. برای انجام این کار باید فاصله بین دو نمونه تعریف شود. فرض کنید دو نمونه $x_۱$ و $x_۲$ را به‌صورت زیر تعریف کرده‌ایم:

$$X_۱ = (x_{۱i}, x_{۲i}, \ldots, x_{۱n}) \quad , \quad X_۲ = (x_{۲i}, x_{۲i}, \ldots x_{۲n}) \qquad (۶-۵)$$

یعنی $x_۱$ و $x_۲$ به ترتیب دارای n ویژگی با مقادیر $x_{۱۱} \ldots x_{۱n}$ و $x_{۲۱} \ldots x_{۲n}$ هستند. برای محاسبه فاصله دو نمونه می‌توان از رابطه اقلیدسی استفاده کرد، تابع فاصلهٔ زیر این‌کار را انجام می‌دهد:

$$dist(X_۱ , X_۲) = \sqrt{\sum_{i=۱}^{n} (x_{۱i} - x_{۲i})^۲} \qquad (۷-۵)$$

با محاسبه فاصله نمونه ورودی (یعنی (۳,۷) که قرار است دسته‌بندی روی آن انجام شود) با نمونه‌های آموزشی، نتایج زیر به‌دست می‌آید.

جدول ٥-٤) فاصله نمونه ورودی (۳,۷) با تمام نمونه‌های آموزشی

مقاوت در برابر اسید $x_۱$ = (seconds)	دوام $x_۲$ = (kg/square meter)	(۳,۷) فاصله‌اقلیدسی با نمونه
۷	۷	٤
۷	٤	٥
۳	٤	۳
۱	٤	$\sqrt{۱۳}$

در گام دوم الگوریتم باید K همسایه نزدیک‌تر را انتخاب کند. با فرض k = ۳ داریم.

جدول ۵- ۵) پیدا کردن همسایه نزدیک‌تر به نمونه ورودی، در نمونه‌های آموزشی

جزء ۳ همسایه نزدیک هست؟	رتبه (فاصله‌اقلیدسی)	فاصله‌اقلیدسی با نمونه (۳,۷)	دوام= $x_۲$ (kg/square meter)	مقاومت در برابر اسید = $x_۱$ (seconds)
بله	۳	٤	۷	۷
خیر	٤	٥	٤	۷
بله	۱	۳	٤	۳
بله	۲	$\sqrt{۱۳}$	٤	۱

نهایتاً در گام سوم الگوریتم باید دسته‌ای را که حائز اکثریت در بین همسایه‌ها است به عنوان دستهٔ نمونهٔ ورودی در نظر بگیرد.

جدول۵- ٦) بررسی کلاس نزدیک‌ترین همسایه‌ها برای تخمین کلاس نمونه ورودی

کلاس نزدیک‌ترین همسایه	جزء ۳ همسایه نزدیک هست؟	رتبه (فاصله‌اقلیدسی)	فاصله‌اقلیدسی با نمونه (۳,۷)	دوام= $x_۲$ (kg/square meter)	مقاوت در برابر اسید= $x_۱$ (seconds)
Bad	بله	۳	٤	۷	۷
-	خیر	٤	٥	٤	۷
Good	بله	۱	۳	٤	۳
Good	بله	۲	$\sqrt{۱۳}$	٤	۱

از بین نزدیک‌ترین همسایه‌ها (مشابه‌ترها به نمونه ورودی) دو تا خوب و یکی بد است. بنابراین حدس می‌زنیم که نمونه ورودی نیز در دسته خوب، که حائز اکثریت است، قرار بگیرد.

بررسی دقیق‌تر روش نزدیک‌ترین همسایگی

مسائل مربوط به تابع فاصله: در گام اولِ روش نزدیک‌ترین همسایگی، فاصله نمونه ورودی با تمام نمونه‌های آموزشی محاسبه می‌شود. دقت در تعریف درست این تابع و همچنین در تعریف محدوده و دامنه متغیرهای ورودی آن (فیلدهای نمونه‌ها) از اهمیت

به‌سزایی برخوردار است. توجه کنید که اکثر مسائل مطرح شده در ذیل با به کار گرفتن تابع فاصله عمومی خوشه‌بندی که ویژگیهای مختلف را در تابع فاصله با هم ترکیب می‌کرد، رفع می‌شود. در مورد این تابع باید به مسائل زیر توجه کنیم:

- **مقایسه ویژگیهای غیرعددی:** این مسئله از اینجا ناشی می‌شود که همیشه ویژگیها عددی نیستند، مثلا در مورد ویژگی رنگ، چه باید کرد؟ ساده‌ترین روش مقایسه اینست که اگر مقدار ویژگی در دو نمونه برابر است تفاوت را صفر و در غیر این‌صورت یک در نظر گرفته شود. البته روشهای دیگری نیز وجود دارد که در فصل خوشه بندی به برخی از آنها اشاره شده است.
- **تفاوت در مقیاس اندازه‌گیری ویژگیها:** نکته دیگر این است که مقیاس اندازه‌گیری ویژگیها متفاوت است. مشکل اینجاست که ویژگی‌ای مانند قد محدوده بسیار بیشتری از نمره یک امتحان دارد. با توجه به جمع شدن مقدار تفاوت در ویژگیهای متناظر در تابع فاصله، ویژگیهای با مقیاس بالا اثر ویژگیهای با مقیاس پایین را محو می‌کنند. راه حل این است که مقادیر قبل از مقایسه نرمال شوند، ساده‌ترین راه نرمالسازی ویژگی A با مقدار vبه مقدار v' در فاصله [۰،۱] است که با فرمول زیر انجام می‌شود:

$$v' = \frac{v - min_A}{max_A - min_A} \qquad (۸-۵)$$

قابل ذکر است در رابطه (۵-۸)، min_A و max_A (حداقل و حداکثر) روی مجموعه آموزشی محاسبه می‌شود.

- **ویژگی در یک (یا دو) نمونه مقدار ندارد:** در این موارد حداکثر مقدار ممکن به عنوان تفاوت مقدار ویژگی در دو نمونه درنظر گرفته می‌شود. در حالت کلی با توجه به عددی یا غیرعددی بودن ویژگیها از جدول (۵-۷) استفاده می‌کنیم.

جدول ۵-۷) محاسبه تفاوت مقدار ویژگیها در حالت نبود مقدار برای ویژگی در نمونه‌ها

ویژگی	تفاوت
غیر عددی	۱
هیچکدام مقدار عددی ندارند	۱
در یکی مقدار v و در دیگری مقدار ندارد	مقدار بزرگتر v و $1-v$ در نظر گرفته می‌شود.

- **انتخاب تابع فاصله:** برای محاسبه تابع فاصله، روشهای بسیار زیادی وجود دارد، ولی استفاده از دو تابع برای محاسبه فاصله مرسوم است: یکی تابع اقلیدسی که قبلاً به آن اشاره شد و دیگری تابع مانهاتان. برای محاسبه فاصله دو نمونهٔ x_1 و x_2 فرمول(۵-۸)، این توابع به‌صورت زیر تعریف می‌شوند:

جدول ۵- ۸) (a) تابع مانهاتان، (b) تابع اقلیدسی

$dist(x_1, x_2) = \sum_{i-1}^{n} \lvert x_{1i} - x_{2i} \rvert$	$dist(x_1, x_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2}$
(a)	(b)

تابع اقلیدسی، به تفاوت‌ها حساس‌تر است یعنی تفاوت (یا شباهت) مقدار ویژگیها در آن، مهم‌تر از تابع مانهاتان است. نکته دیگر اینکه، با توجه به زمان‌بر بودن عمل جذر در تابع اقلیدسی و اینکه نهایتاً فاصله‌ها با هم مقایسه می‌شوند، می‌توان از جذر در محاسبه فاصله‌اقلیدسی صرف‌نظر کرد. تابع اقلیدسی به‌دلیل سادگی در محاسبه و کارایی، مرسوم‌ترین تابع استفاده شده در روش نزدیک‌ترین همسایگی برای محاسبه فاصله‌است.

- **یکسان گرفتن اهمیت ویژگیها در تابع فاصله:** تابع اقلیدسی اشاره شده در رابطه برمبنای این فرض است که تمام ویژگیها برای محاسبه فاصله مرتبط بوده و به یک اندازه اهمیت دارند. ولی در دنیای واقعی این‌گونه نیست. بعضی از ویژگیها

نامرتبط‌اند و بعضی دیگر بسیار مهم. برای ایجاد تمایز بین ویژگیها، تابع اقلیدسی را به‌صورت زیر دستکاری کرده و برای ویژگی i، وزن Wi را تعریف می‌کنیم.

$$Euclidean\ (X_1, X_2) = \sqrt{\sum_{i=1}^{n} w_i (x_{1i} - x_{2i})^2} \qquad (۹-۵)$$

ولی این وزنها چطورتعیین می‌شوند؟ برای این کار می‌توان از نظر خبره‌ای که کسب و کار را می‌شناسد، استفاده کرد تا او اوزان را تعیین کند. راه دیگر استفاده از روشهای الگوریتمی برای این کار است. اساس این روشها بر مبنای اعتبارسنجی تقاطعی یا چندمرحله‌ای استوار است. یعنی با یک مقدار تصادفی اولیه برای وزن ویژگیها شروع کرده، دسته‌بندی را روی نمونه‌های تست انجام داده، خطای دسته‌بندی را محاسبه کرده و وزنها را به تدریج طوری تغییر داده تا خطا حداقل شود. یکی از روشهای ممکن، استفاده از الگوریتم ژنتیک است که در آن مجموعه اوزان به‌عنوان یک کرموزوم و برازندگی[1] آنها از روی خطای دسته‌بندی محاسبه می‌شود. روش جالب دیگر که توسط اِیها[2] ارائه شده، با یک مقدار اولیه شروع کرده و بعد از هر بار دسته‌بندی، اوزان را عوض می‌کند. جزئیات این روش در بخش بعد توضیح داده شده است.

روش اِیها (*Aha*)

فرض کنید نمونه تست X برای تعیین دسته، وارد شده و Y به‌عنوان نزدیک‌ترین همسایه آن انتخاب شده است. برای تعیین وزن ویژگی i، ابتدا تفاوت مقدار این ویژگی در دو نمونه را پیدا کرده با توجه به درستی / نادرستی دسته‌بندی وزن را طبق جدول (۹-۵) عوض می‌کنیم.

---

[1]- Fitness
[2]- Aha

جدول ۵-۹) تغییر وزن ویژگی بر اساس صحت دسته‌بندی و تفاوت مقدار آن در نمونه ورودی و آموزشی

تفاوت / دسته‌بندی	درست	غلط
کم	افزایش زیاد	کاهش زیاد
زیاد	افزایش کم	کاهش کم

مسائل مربوط به تابع ترکیب

همان‌طورکه اشاره شد، بعد از مشخص شدن نزدیک‌ترین همسایه‌ها، در گام آخر الگوریتم باید از روی دسته آنها، دسته نمونه ورودی را تعیین کند. به این عمل ترکیب[۱] می‌گویند. ساده‌ترین روش ترکیب، روش بدون وزن است که بر مبنای رای اکثریت است، یعنی کلاسی که حائز اکثریت در بین نزدیک‌ترین همسایه‌ها باشد انتخاب می‌شود. در این روش فاصله در اهمیت رأی تأثیر ندارد، به‌علاوه ممکن است با مشکل بند[۲] مواجه شویم، یعنی دو (یا چند) گروه حائز اکثریت باشند. برای مشکل‌گشایی می‌توان به‌طور تصادفی یکی از گروه‌های حداکثر را انتخاب کرده یا اینکه در صورت وجود C دسته مختلف، $C+1$ تا از نزدیک‌ترین همسایه‌ها را انتخاب کرد تا این مشکل پیش نیاید[۳]. روش بهتر برای ترکیب، روش وزن‌دار است. در این روش هر رأی دارای وزنی است که با توجه به فاصله تعیین می‌شود. قاعدتاً رأی همسایه‌های نزدیک‌تر باید وزن بیشتری داشته باشند. اگر A نمونه ورودی و X نمونه آزمایشی باشد، وزن رأی آن ازرابطه (۵-۱۰) محاسبه می‌شود:

$$weight(X) = \frac{1}{dist(A,X)^2} \qquad (۱۰-۵)$$

[1]- Combination
[2]- Tie

[۳]- طبق اصل لانه کبوتری

که در آن $dist(A,X)$ فاصله بین A و X است. معمولاً برای رفع مشکل تقسیم بر صفر، مخرج را با یک جمع می‌کنند. این روش ترکیب، علاوه بر عادلانه بودن، احتمال وقوع **بند** را حداقل می‌کند.

انتخاب مقدار K

یکی از پارامترهای مهم در روش نزدیک‌ترین همسایگی، مقدار K می‌باشد. واقعیت این است که مقدار دقیقی برای K وجود نداشته و مقدار مناسب آن بستگی به توزیع داده‌ها و فضای مسئله دارد. ولی مقدار کوچک K، روش را متأثر از داده‌های مغشوش کرده و مقدار بزرگ آن، رفتارهای محلی را در نظر نمی‌گیرد. نهایتاً مقدار K با سعی و خطا تعیین می‌شود. مثلاً در روش اعتبارسنجی تقاطعی، با مقدار اولیه شروع کرده K را تغییر می‌دهیم تا به حداقل خطای دسته‌بندی برسیم.

انتخاب مجموعه آموزشی مناسب

عملکرد هر دسته‌بند اساساً وابسته به مجموعه آموزشی آن است. مجموعه آموزشی زیرمجموعه‌ای از فضای نمونه‌هاست که باید تنوع کافی از دسته‌های مختلف را در خود داشته باشد. در غیر این‌صورت نتایج به یک سمت خاص (دسته‌های با فرکانس بالا) سوگیری خواهند داشت. در واقع مجموعه آموزشی باید از پوشش[1] مناسبی برخوردار باشد. برای رسیدن به پوشش، روشهای زیادی وجود دارد از جمله اینکه از دسته‌های مختلف به تعداد برابر در مجموعه آموزشی قرار داد. روش دیگر انتخاب تصادفی است. در این روش انتخاب دسته‌های با فراوانی بالا در انتخاب تصادفی، پوشش را کم می‌کنند، مثلاً در داده‌های مربوط به تراکنش‌های مالی، مطمئناً تراکنشهای کلاه‌بردارانه تعداد بسیار اندکی را تشکیل می‌دهند. حال اگر قصد ما تعیین وضعیت یک تراکنش از نظر کلاه‌برداری بودن یا نبودن باشد، با انتخاب تصادفی، احتمال دارد هیچ تراکنش کلاه‌برداری در مجموعه آموزشی قرار نگیرد.

---

[1]- Coverage

داده‌های مغشوش

یکی دیگر از مشکلات موجود در مجموعه آموزشی (و در حالت کلی یکی از چالشهای داده‌کاوی) وجود داده‌های با اغتشاش یا نویزدار است. همان‌طورکه قبلاً اشاره شد با افزایش مقدار K، اثر داده‌های مغشوش محو می‌شود. اصولاً فلسفه وجودی K همین رفع اثر اغتشاشهاست و در صورت اطمینان از عدم وجود اغتشاش می‌توان از آن صرف‌نظر کرد (یعنی K را یک در نظر گرفت). روش دیگر مقابله با اغتشاش، الگوریتمی به نام یادگیری بر اساس نمونه‌ها یا *IB3*[1] است. این الگوریتم نسخه سوم از الگوریتمهای پنج‌گانه، *IB1* تا *IB5* است که روش نزدیک‌ترین همسایگی را تکمیل کرده‌اند. ایده اصلی این است که: «فقط نمونه‌هایی را که کارآیی خوبی برای دسته‌بندی داشته‌اند در مجموعه آموزش نگه دارند.»

الگوریتم *IB3*

این الگوریتم در واقع یک مرحله پیش‌پردازش روی داده‌های آموزشی است. فرض کنید مجموعه آموزشی اولیه T باشد. نهایتاً زیر مجموعه S را نگه می‌داریم، در انتها، مجموعه S به‌عنوان مجموعه آموزشی در نظر گرفته می‌شود. این روش «فقط نمونه‌هایی را که کارآیی خوبی داشته‌اند و درست دسته‌بندی شده باشند را در مجموعه آموزش نگه می‌دارد». الگوریتم را در شکل (۵-۷) مشاهده می‌کنید.

```
1. For each instance t in T
2.       Let a be the nearest acceptable instance in S to t.
3.       (if there are no acceptable instances in S, let a be a random instance in S)
4.       If class(a) ≠ class(t) then add t to S.
5.       For each instance s in S
6.              If s is at least as close to t as a is
7.                     Then update the classification record of s
8.                     and remove s from S if its classification record is
                       significantly poor.
9. Remove all non-acceptable instances from S.
```

شکل ۵-۷) الگوریتم IB3

[1]- Instance Based Learner Version 3

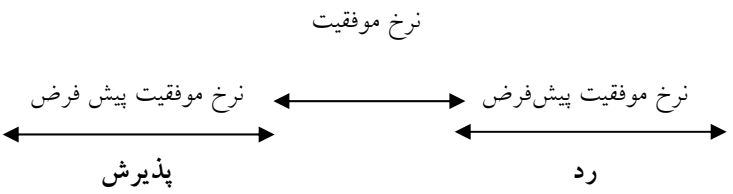
افزودن و حذف عناصر از S با توجه به مفاهیم نرخ موفقیت نمونه و نرخ موفقیت پیش فرض آن صورت می‌گیرد. نرخ موفقیت نمونه این‌گونه تعریف می‌شود:

فرض کنید که نمونه‌ای N بار(از زمان ورود به S) برای دسته‌بندی انتخاب شده و fدقت این موارد است. با قرار دادن این مقادیر در فرمول زیر و داشتن مقادیر اطمینان می‌توان نرخ موفقیت pرا حساب کرد.

$$p=\left(f+\frac{z^{۲}}{۲N}\pm z\sqrt{\frac{f}{N}-\frac{f^{۲}}{N}+\frac{z^{۲}}{٤N^{۲}}}\right)/\left(۱+\frac{z^{۲}}{N}\right) \qquad (۱۱-۵)$$

در رابطه (۵-۱۱) z از جداول مربوط به توزیع نرمال به‌دست می‌آید، در واقع اگر متغیر تصادفی fرا دقت دسته‌بند در N بار امتحان بدانیم با در نظر گرفتن دسته‌بندی به‌عنوان یک فرایند برنولی با دو پیشامد درست یا غلط، می‌توان در مقادیر بالای N، آن را با توزیع نرمال تقریب زد.

برای محاسبه نرخ موفقیت پیش فرض نمونه، fرا برابر نسبتی از نمونه‌ها که تا به حال از این کلاس دیده شده‌اند و N را تعداد نمونه‌هایی که تا به حال پردازش شده‌اند در نظر گرفته و نرخ موفقیت را حساب می‌کنیم. شرط پذیرش یک نمونه این است که حد پایین نرخ موفقیت آن از حد بالای نرخ پیش‌فرض موفقیت تجاوز کند و شرط رد آن این است که حد بالای نرخ موفقیت، کمتر از حد پایین نرخ موفقیت پیش‌فرض باشد.



شکل ۵-۸) شکل شماتیک فاصله‌های اطمینان رد یا قبول یک نمونه

مقادیر پیشنهادی درجه اطمینان، برای قبول، ۵ درصد و برای رد ۱۲/۵ درصد است. هرچه درصد اطمینان کمتر باشد فاصله‌اطمینان بزرگتر و سختگیرانه‌تر است، زیرا همان‌طور که در شکل (۵–۸) مشخص است این کار احتمال تلاقی فاصله‌ها را زیادتر می‌کند. در کل شرایط قبول سخت‌گیرانه‌تر است، زیرا با رد نمونه‌های با کیفیت متوسط چیزی از دست نمی‌دهیم، و این نمونه‌ها براحتی جایگزین می‌شوند.

مشکل سرعت روش نزدیک‌ترین همسایگی

اگر چه روش نزدیک‌ترین همسایگی، روش ساده و مؤثری است ولی سرعت کمی دارد. اگر اندازه مجموعه آموزشی D و $K=1$ باشد، دسته‌بندی نمونه جدید از مرتبه زمانی D یعنی $O(D)$ خواهد بود. تلاشهای زیادی برای افزایش سرعت صورت گرفته است مثل: خوشه‌بندی، فاصله‌جزئی، رأی‌گیری بر مبنای فاصله‌های ویژگیها و *kd-tree*. در روش خوشه‌بندی، ابتدا مجموعه آموزشی خوشه‌بندی شده ولی در هنگام دسته‌بندی، نمونه ورودی ابتدا با مرکز خوشه‌ها مقایسه شده و بعد جستجو در نزدیک-ترین خوشه ادامه پیدا می‌کند. در روش فاصله جزئی، فاصله روی زیر مجموعه‌ای از n ویژگی اندازه‌گیری شده اگر مقدار آن از آستانهٔ فاصلهٔ تعریف شده بیشتر بود، محاسبات بیشتری برای این نمونه انجام نمی‌شود. در روش رأی‌گیری بر مبنای فاصله‌های ویژگیها، ابتدا ویژگیهای نمونه‌های آموزشی را به فاصله‌هایی تقسیم کرده و فراوانی هر دسته را محاسبه می‌کنیم. سپس نمونه ورودی را با این فاصله‌ها مقایسه کرده و دسته‌ای که بیشترین تطابق را دارد انتخاب می‌کنیم.

روش *k-Dtree*

یکی از روشهای بسیار مفید برای بالابردن سرعت روش *k-Dtree* است. این روش از روی نمونه‌های آموزشی درختی می‌سازد که گره‌های آن نمونه‌ها هستند. k، تعداد ویژگیهاست. در واقع نمونه‌ها را به‌عنوان نقاطی در فضای k بعدی در نظر می‌گیرد. این درخت **دودویی** فضای ورودی را به بخش‌هایی افراز می‌کند. روال کلی به این صورت

است که در هر مرحله یک ویژگی انتخاب شده و بر اساس آن تقسیم‌بندی مجدد انجام می‌شود، تمام تقسیمات موازی یکی از محورها بوده و در نهایت هر ناحیه دارای حداکثر یک نقطه است. شکل (۵-۹) الگوریتم ساخت را نشان می‌دهد.

```
function kdtree (list of points pointList, int depth)
{
  if pointList is empty
    return nil;
  else
  {
    // Select axis based on depth so that axis cycles through all  valid values
    var int axis:= depth mod k;
    // Sort point list and choose median as pivot element
    select median from pointList;
    // Create node and construct subtrees
    var tree_node node;
    node. location:= median;
    node. leftChild:= kdtree(points in pointList before median, depth+1);
    node. rightChild:= kdtree(points in pointList after median, depth+1);
    return node;
  }
}
```

شکل ۵-۹) الگوریتم ساخت k-Dtree

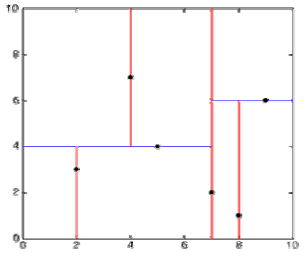
در این الگوریتم بازگشتی، در هر مرحله یک ویژگی به تناوب و با توجه به عمق انتخاب می‌شود. میانه حول آن محاسبه شده و نهایتاً روال به‌صورت بازگشتی برای نقاط سمت چپ و راست میانه و با افزایش عمق فراخوانی می‌شود. شکل (۵-۱۰) نمونه‌ای از ساخت *k-Dtree* را نشان می‌دهد.

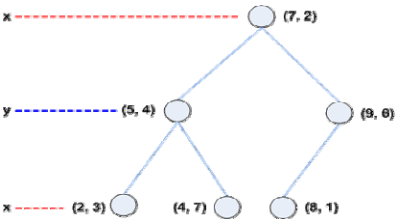
مزیت اصلی روشهای مبتنی برنمونه امکان اضافه شدن راحت نمونه‌هاست. برای اضافه کردن ورودی‌های جدید به *k-Dtree* الگوریتم زیر را داریم.

- ناحیه نقطه ورودی را پیدا کن.
- اگر خالی بود نقطه را در آن قرار بده.

- در غیر این‌صورت، تقسیم بندی را انجام داده و نقطه را به‌عنوان برگ سمت چپ یا راست قرار بده.

$$point\ List = [(۲,۳),(۵,٤),(۹,٦),(٤,۷),(۸,۱),(۷,۲)]$$
$$tree = kdtree(point\ List)$$





شکل ۵- ۱۰) فراخوانی روال ساخت، k-Dtree و افراز فضای نقاط

جستجو در درخت: مجزا از ساخت و به‌روزرسانی درخت، عملیات اصلی روی درخت (در واقع هدف اصلی ایجاد آن) کاهش زمان جستجو برای پیدا کردن نزدیک‌ترین نقطه به نقطه ورودی است. الگوریتم زیر این کار را انجام می‌دهد:

- درخت را از ریشه پیمایش کن تا به ناحیه‌ای که نقطه ورودی در آن قرار می‌گیرد، برسی.
- برگ ناحیه، لزوماً نزدیکترین همسایه نیست ولی تخمین خوبی است (نزدیکترین همسایه اولیه)
- بررسی امکان وجود همسایه نزدیکتر.

آیا می‌تواند در ناحیه هم‌ردیف قرار بگیرد؟ در صورتی‌که دایره به مرکز نقطه ورودی و شعاع فاصله ورودی با نزدیک‌ترین همسایه فعلی، ناحیه‌ای را قطع می‌کرد باید دوباره بررسی شود. با این الگوریتم مرتبه زمانی جستجو از D به $Log_۲(D)$ تقلیل می‌یابد که در مقادیر بالای داده‌ای (اصل بحث داده‌کاوی) بسیار مؤثر است.

شبکه‌های عصبی در دسته‌بندی

شبکه‌های عصبی روشی است که قصد دارد با استفاده از مدلهای ریاضی و توان کامپیوتر، برخی از جنبه‌های ساده مغز انسان را شبیه‌سازی کند. شبکه‌های عصبی به صورت یکی از بخشهای پیچیده مغز انسان، به‌عنوان یک ساختار یادگیری غیر قابل درک، مشهور شده است. این ساختار پیچیده از مجموعه‌ای از نرونها بوجود آمده است که خود نرونها ساختار ساده‌ای داشته، ولی شبکه اتصال این نرونها وظایف یادگیری بسیار پیچیده‌ای را به انجام می‌رساند. لذا شناخت و درک ساختار بیولوژی مغز انسان می‌تواند ما را در ایجاد شبکه‌های عصبی مصنوعی[۱] به‌عنوان یک ابزار کارا در حل مسائل و کاربردهای علمی و فنی یاری رساند.

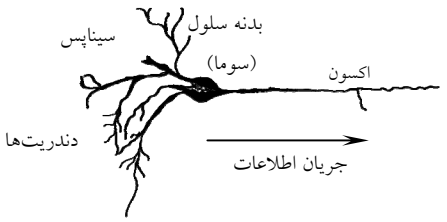
یکی از کاربردهای بارز شبکه‌های عصبی مصنوعی در داده‌کاوی می‌باشد. تا آنجایی که حوزه‌ای، تحت عنوان داده‌کاوی بر مبنای شبکه‌های عصبی[۲] بوجود آمده است. شبکه‌های عصبی مصنوعی در برخی از عملیات مانند پیش‌بینی و دسته‌بندی در مقایسه با سایر روشها دارای مزایای مناسبی بوده و معمولاً در کارهای اجرایی ترجیح داده می‌شوند. در این بخش ضمن آشنایی با مفاهیم و اصول مورد نیاز شبکه‌های عصبی برای به‌کارگیری در مسائل داده‌کاوی، سعی می‌گردد، حداقل کاربرد شبکه‌های عصبی در دسته‌بندی تشریح گردد.

تاکنون تحقیقاتی بسیار در زمینه ساختار مغز انسان صورت پذیرفته ولی هنوز سؤالات بسیاری وجود دارد. سلولهای مغز انسان دارای ساختار متفاوتی از سایر سلولهای بدن انسان می‌باشند به این سلولهای مغزی نرون[۳] گفته می‌شود. هر نرون یک بدنه، یک آکسون و چندین دندریت داشته و واسط بین آکسون یک نرون و دندریتهای نرونهای

---

[1]- Artificial Neural Networks
[2]- Neural Network Data Mining
[3]- Neuron

دیگر سیناپس نام دارد. همچنین هر نرون بر اساس یک آستانه تحریک در یکی از دو وضعیت تحریک شده[1] و ساکن[2] قرار می‌گیرند.



شکل ۵-۱۱) طرح قسمتهای مختلف یک عصب بیولوژیکی

این ساختار نرون در مغز انسان به تعداد 10^{11} تکرار می‌گردد و از آنجا که هر نرون حداقل به ۱۰۰۰۰ نرون دیگر متصل می‌باشد، در مغز انسان 10^{15} اتصال سیناپسی وجود دارد که تمامی فعالیتهای ذهنی را به انجام می‌رسانند.

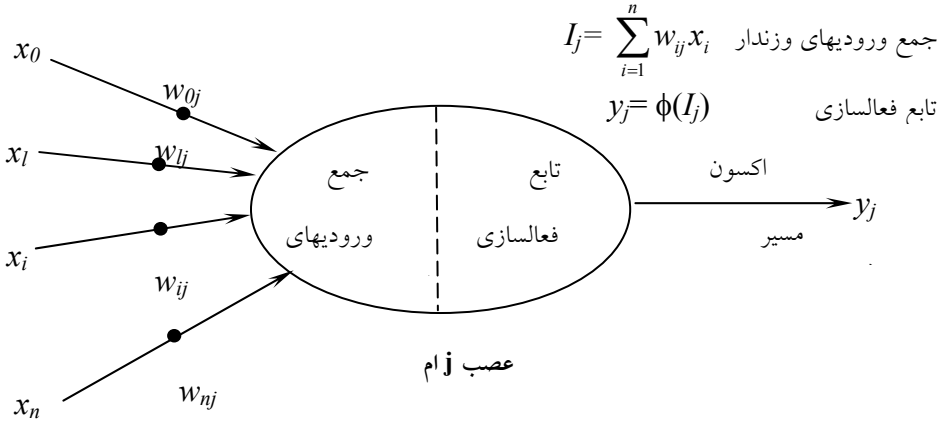
با توجه به ساختار فوق می‌توان ایده شبکه عصبی مصنوعی را به‌صورت ذیل تشریح نمود:

- مجموعه‌ای از گره‌ها (واحدها، نرونها، عناصر محاسباتی)
- هر گره ورودی و خروجی دارد.
- هر گره بر اساس تابعی خاص محاسبه ساده‌ای انجام می‌دهد.
- بین گره‌ها، اتصالات موزون وجود دارد.
- اتصالات بر اساس معماری شبکه مشخص می‌شوند.
- نتیجه یک شبکه تابعی بسیار پیچیده از ارتباطات موزون می‌باشد.

---

[1]- Firing
[2]- Rest

با به استعاره گرفتن ساختار شبکه‌های عصبی زنده، مدل ریاضی شبکه‌های عصبی مصنوعی ارائه شد. شکل (۵-۱۳) در واقع در هر شبکه عصبی مصنوعی، مجموعه‌ای از ورودی‌ها، مجتمع گشته و براساس یک تابع فعال‌سازی یک خروجی محاسبه می‌شود.



شکل ۵-۱۲) مدل ریاضی پیشنهادی برای شبکه‌های عصبی مصنوعی

شبکه عصبی زنده	شبکه عصبی مصنوعی
بدنه سلول	**گره**
- سیگنال نرون دیگر	- ورودی
-	- خروجی
- مکانیزم تحریک	- تابع تحریک گره
سیناپسها	**اتصالات**
- قدرت سیناپسها	- وزنها

شکل ۵-۱۳) مقایسه مفاهیم شبکه‌های عصبی زنده و مصنوعی

ولی همواره این سؤال مطرح است، که شبکه‌های عصبی مصنوعی برای حل چه نوع مسائلی مناسب می‌باشند. با توجه به مزایای و مشکلات ذیل می‌توان تصمیم‌گیری نمود که در چه نوع مسائلی می‌توان از این رویکرد استفاده نمود.

مزایای شبکه‌های عصبی

- قابلیت مواجه با داده‌های مغشوش
- قابلیت استفاده در زمانی که دانش بسیار کمی در مورد مسئله وجود دارد.
- برای هر دو نوع داده کمی و کیفی مناسب است.
- در مسائل متفاوتی از پردازش تصویر گرفته تا تشخیص درمان کاربرد دارند.
- به دلیل کارکرد موازی نسبت به سایر روشها سرعت بالاتری دارد.

معایب شبکه‌های عصبی

- آموزش این شبکه‌ها بسیار حساس است.
- غیر قابل تفسیرند[1].

از مزایا و معایب این شبکه‌ها می‌توان نتیجه گرفت که این شبکه‌ها می‌توانند به‌عنوان روش مناسب در ایجاد مدل‌های تحلیلی و تخمینی و برخورد با داده‌های متفاوت سازمانی در حوزه‌ها و پروژه‌های متفاوت داده‌کاوی به کار گرفته شوند. به‌طور مثال در عملیات پیش‌بینی و سریهای زمانی، داده‌های پیچیده مالی و داده‌های بورس شبکه‌های عصبی کاربرد فراوانی دارند.

تبدیلات ورودی و خروجی

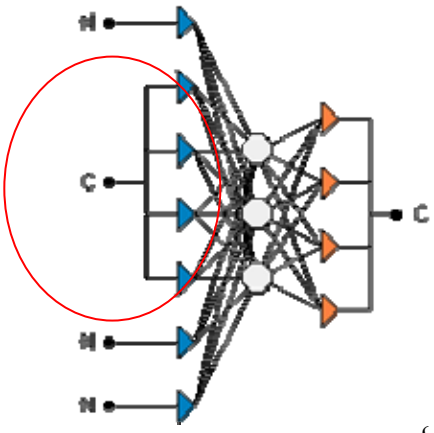
به دلیل ساختار و معماری خاص و الگوریتمهای شبکه‌های مصنوعی، کلیه ویژگیهای ارزشی در مدل این شبکه‌ها می‌بایست به‌صورت استاندارد تبدیل شوند. برای متغیرهای کمی پیوسته از روشهای مناسب نرمال‌سازی داده‌ها مانند روش ذیل استفاده می‌شود:

$$X^* = \frac{X - min(X)}{range(X)} = \frac{X - min(X)}{max(X) - min(X)} \qquad (۱۲-۵)$$

[1]- Black Box

برای متغییرهای کیفی و دسته‌ای معمولاً از متغییرهای شاخصی استفاده می‌شود. مثلاً برای جنسیت دو ویژگی زن و مرد تعریف شده و براساس داده‌ها، مقادیر صفر یا یک به آنها تخصیص می‌یابد. در واقع نوع متغییرهای ورودی معماری شبکه را تحت تأثیر قرار می‌دهد. به‌طور مثال وجود ورودی ملیت با چهار حالت ممکن:

{ایرانی، آمریکایی، چینی، فرانسوی}



شکل۵-۱٤) متغیرهای ورودی

چهار گره ورودی را به خود اختصاص می‌دهد. که در هر رکورد به منظور مشخص نمودن ملیت، برای یکی از چهار فیلد، مقدار یک و برای بقیه صفر در نظر گرفته می‌شود. خروجی شبکه عصبی همواره اعداد کمی می‌باشند. از آنجا که در دسته‌بندی ما به دنبال تخصیص برچسب به داده‌ها می‌باشیم، می‌بایست با توجه به نوع دسته‌های مورد انتظار گره‌های خروجی مربوط به شبکه را تعریف نموده و قواعد تفسیر اعداد کمی خروجی‌ها را نیز مشخص کنیم. به‌عنوان مثال ما از یک گره خروجی زمانی که دسته‌ها کاملاً روشن و دارای ترتیب باشند استفاده می‌کنیم:

- اگر مقدار خروجی بیش از ۰/۷۵ باشد، فرد در ارزیابی در سطح الف قرار می‌گیرد.
- اگر مقدار خروجی بین ۰/۵ و ۰/۷۵ باشد، فرد در ارزیابی در سطح ب قرار می‌گیرد.
- اگر مقدار خروجی بین ۰/۵ و ۰/۲۵ باشد، فرد در ارزیابی در سطح ج قرار می‌گیرد.
- اگر مقدار خروجی کمتر از ۰/۲۵ باشد، فرد در ارزیابی در سطح د قرار می‌گیرد.

لیکن در برخی از شرایط نمی‌توان دسته‌ها را به‌صورت ترتیبی مشخص نمود و می‌بایست به تعداد دسته‌های مورد انتظار گره، خروجی تعریف نمود و در صورت

تخصیص مقدار یک به گره، دسته مورد نظر مشخص می‌شود. وضعیت تأهل (مجرد، متاهل، مطلقه، بیوه و نامشخص) از این نوع دسته‌بندی می‌باشد.

به دلیل اینکه شبکه‌های عصبی خروجیهای کمی پیوسته تولید می‌کنند، در تخمین و پیش‌بینی بسیار کاربرد دارند. مثلاً در تخمین قیمت سهام در ماه بعد با استفاده از شبکه‌های عصبی می‌بایست مقدار گره خروجی به مقدار واقعی خود تبدیل شود، به همین دلیل از فرمول زیر استفاده می‌شود که عکس عمل استاندارد کردن داده است:

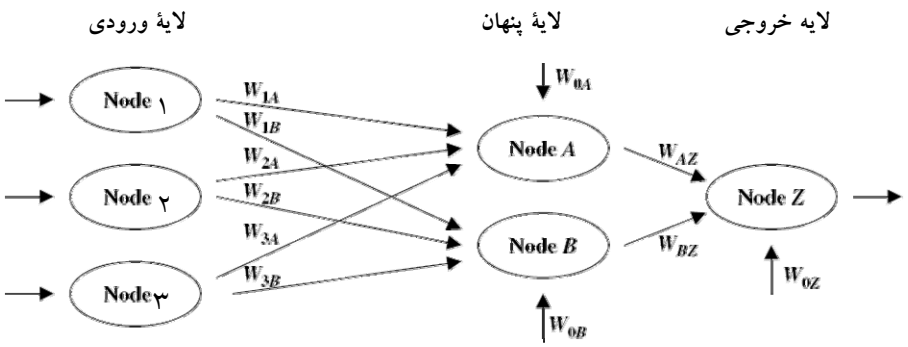
Prediction = output (data range) + minimum

مثال: به منظور تشریح ساختار محاسباتی شبکه‌های عصبی و فهم بسته سیاه این شبکه‌ها در این بخش یک مثال از شبکه عصبی چند لایه، پیش‌خور و کاملا متصل[1] بیان می‌شود. این شبکه، در شکل (۵-۱۵) نشان داده شده است. ویژگی پیش‌خور این شبکه باعث می‌گردد که در این شبکه، حلقه و یا برگشت به عقب وجود نداشته باشد. همچنین این شبکه از سه لایه ورودی، پنهان و خروجی تشکیل شده پس یک شبکه چند لایه بوده و از آنجا که هر گره به تمام گره‌های لایه بعد متصل است به آن شبکه کاملاً متصل نیز می‌گویند. هر اتصال (سیناپس) دارای یک وزن (قدرت سیناپس) می‌باشد، که در ابتدا تصادفی و بین صفر و یک در نظر گرفته می‌شود.

همان‌طورکه در بالا تشریح شد، تعداد گره‌های ورودی به تعداد و نوع ویژگیهای مجموعه داده‌ها و تعداد گره‌های خروجی به نوع عملیات دسته‌بندی بستگی دارد. لیکن تعداد گره‌ها (نرونهای) لایه پنهان یک مفهوم ابتکاری است و با سعی و خطا حاصل می‌شود.

---

[1]- Fully Connected



$x_{\cdot} = ۱/۰$ $W_{OA} = ۰/۵$ $W_{OB} = ۰/۷$ $W_{OZ} = ۰/۵$

$x_{۱} = ۰/٤$ $W_{۱A} = ۰/٦$ $W_{۱B} = ۰/۹$ $W_{AZ} = ۰/۹$

$x_{۲} = ۰/۲$ $W_{۲A} = ۰/۸$ $W_{۲B} = ۰/۸$ $W_{BZ} = ۰/۹$

$x_{۳} = ۰/۷$ $W_{۳A} = ۰/٦$ $W_{۳B} = ۰/٤$

شکل ۵-۱۵) مثال شبکه عصبی ساده

در ابتدا یک ترکیب خطی (مقدار اسکالر) از ورودی‌های یک گره ایجاد کرده که به آن *net* گره می‌گویند.

$$net_j = \sum_i Wij\, x_{ij} = W_{oj}\, x_{oj} + W_{۱j} x_{۱j} + \ldots + W_{۱j} x_{۱j} \qquad (۱۳-۵)$$

مقدار $x_{\cdot j}$ برابر یک بوده و اوزان متناظر با آن مانند ثابت رگرسیون عمل می‌نمایند. بر اساس همین فرمول مقادیر *net* را برای گره‌های A و B محاسبه می‌کنیم.

$$net_A = \sum_i W_{iA} x_{iA} = W_{OA}(۱) + W_{۱A} x_{۱A} + W_{۲A} x_{۲A} + W_{۳A}\, x_{۳A}$$

$$= ۰.۵ + ۰.٦(۰.٤) + ۰.۸(۰.۲) + ۰.٦(۰.۷) = ۱.۳۲$$

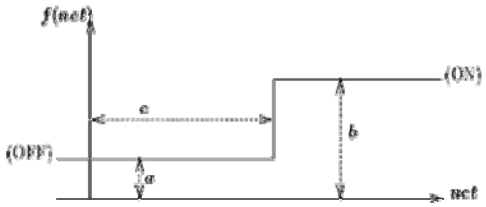
$$net_B = \sum_i W_{iB} x_{iB} = W_{OB}(۱) + W_{۱B} x_{۱B} + W_{۲B} x_{۲B} + W_{۳B} x_{۳B}$$

$$= ۰.۷ + ۰.۹\,(۰.٤) + ۰.۸(۰.۲) + ۰.٤(۰.۷) = ۱.۵$$

توابع فعال سازی[1]

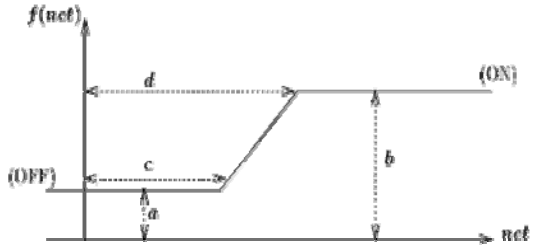
همان‌طورکه در مقدمه بیان شد، یک نرون دو حالت ساکن و فعال شده دارد که معمولاً بر اساس یک آستانه تحریک ایجاد می‌شود. در واقع می‌توان چنین بیان نمود که ورودیهای یک نرون در داخل یک تابع قرار می‌گیرند و بر اساس مقدار ترکیب ورودیها یا نرون تحریک می‌شود و یا تحریک نمی‌شود. این مفهوم در شبکه‌های عصبی مصنوعی به نام تابع فعال‌سازی شناخته می‌شود که انواع متفاوت ذیل را می‌توان برای آن در نظر گرفت.

- **تابع مشخصات**
- **تابع ثابت**
- **تابع پله (آستانه)**



تابع پله

$$f(net)=\begin{cases} a & if\ net < c \\ b & if\ net > c \end{cases}$$



تابع شیب

$$f(net)=\begin{cases} a & if\ net \le c \\ b & if\ net \ge d \\ a+\dfrac{(net-c)(b-a)}{(d-c)} & otherwise \end{cases}$$

شکل ۵-۱٦) توابع شیب و پله

یکی از معروف‌ترین و پرکاربردترین توابع فعال‌سازی، تابع سیگموئید است که با توجه به فیزیولوژی بدن انسان شبیه‌ترین تابع به نحوه تحریک واقعی نرونها می‌باشد. در ادامه مثال قبل، مقدار net محاسبه شده برای هر گره در این تابع $y=\frac{1}{1+e^{-x}}$ قرار می‌گیرد

[1]- Activation Function

و خروجی گره شکل می‌گیرد. مقدار net_A را به جای x قرار داده و مقدار خروجی گره A محاسبه می‌شود به لایه بعد منتقل می‌شود. این تابع x را به به فاصله ۰ تا ۱ می‌برد.

$$y = ۱/(۱ + e^{-۱.۳۲}) = ۰.۷۸۹۲$$

همین روال برای گره‌های دیگر ادامه می‌یابد و در نهایت مقدار خروجی برای گره آخر Z محاسبه می‌گردد:

$$f(net_B) = \frac{۱}{۱ + e^{-۱.۵}} = ۰.۸۱۷۶$$

$$net_z = \sum_i W_{iz} x_{iz} = W_{oz}(۱) + W_{AZ} x_{AZ} + W_{BZ} x_{BZ}$$

$$= ۰.۵ + ۰.۹(۰.۷۸۹۲) + ۰.۹(۰.۸۱۷۶) = ۱.۹۴۶۱$$

$$f(net_z) = \frac{۱}{۱ + e^{-۱.۹۴۶۱}} = ۰.۸۷۵۰$$

الگوریتم پس انتشار خطا

چنانچه مثال قبل را دنبال کرده باشید، تاکنون هیچ‌گونه فعالیت یادگیری توسط شبکه صورت نپذیرفته است و این درحالی است که فلسفه وجودی شبکه‌های عصبی مصنوعی، یادگیری یک وظفیه مانند تشخیص الگو، دسته‌بندی و یا پیش‌بینی می‌باشد. به‌همین دلیل الگوریتمها و روشهای بسیاری ابداع گردیده است تا شبکه‌ها بتوانند یاد بگیرند. یکی از مشهورترین الگوریتمهای یادگیری که بر اساس کاهش خطا و به صورت نظارتی شکل گرفته است، الگوریتم پس انتشار خطا (BP) نام دارد. در واقع بر اساس وزنهای تصادفی یک پاسخ توسط شبکه تولید می‌شود و در یک فرایند تکراری میزان خطای میان خروجی شبکه با مقادیر واقعی بر اساس تغییر وزنها کاهش می‌یابد. در زمانی که حداقل خطای ممکن حاصل شود، در حقیقت شبکه توسط داده‌ها آموزش داده شده و می‌تواند برای داده‌های جدید، همان الگوی قبلی را ارائه دهد و به‌طور مثال شبکه می‌تواند برای داده‌ها دسته‌بندی ارائه نماید. عمل مقایسه خروجی با مقدار واقعی با ایجاد یک شاخص خطا با نام مجموع مربع خطاها به‌صورت ذیل صورت می‌گیرد:

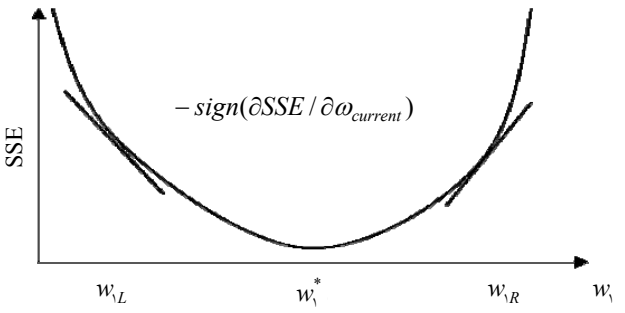
$$SSE = \sum_{records} \sum_{output\ nodes} (\text{مقدار تخمینی} - \text{مقدار واقعی}) \qquad (۱۴-۵)$$

حال باید با استفاده از یک روش بهینه‌سازی این مقدار خطا در هر بار تکرار کاهش یابد، که بدین منظور از روش کاهش گرادیان استفاده می‌شود.

روش کاهش گرادیان

در الگوریتم پس انتشار خطا، کاهش گرادیان به‌عنوان روش بهینه‌سازی و تنظیم اوزان به کار می‌رود. فرض نمایید در شبکه عصبی مصنوعی مورد بحث، یک بردار وزن وجود دارد که ما می‌خواهیم مقادیر این بردار را به‌گونه‌ای پیدا کنیم که مجموع مربع خطاها[1] به حداقل مقدار ممکن برسد.

با توجه به شکل (۵-۱۷) و فرض وجود تنها یک وزن برای درک مسئله، اگر نزدیک W_{1L} باشیم می بایست برای رسیدن به حالت بهینه W را افزایش دهیم و چون مشتق جزئی در این نقطه (همان شیب) منفی است در حالی که جهت حرکت می بایست افزایش W باشد، پس جهت تغییر اوزان همواره مخالف گرادیان می‌باشد.



شکل ۵-۱۷) استفاده از شیب تابع خطا برای جهت تصحیح اوزان

حال سؤال بعدی این است که اوزان چقدر باید تصحیح شوند؟ پاسخ به این سؤال دوباره به شیب منحنی برمی‌گردد. مطمئنا میزان تغییر و تصحیح اوزان بستگی به گرادیان یا همان شیب دارد. چنانچه شیب زیادی داشته باشیم از نقطه بهینه بسیار دوریم

[1]- Sum Square Error

و تصحیح بیشتری با استفاده از یک ضریب باید صورت گیرد و چنانچه میزان گرادیان یا شیب کم باشد، نزدیک مقدار بهینه هستیم و می‌بایست مقدار تصحیح اوزان کاهش یابد.

برای محاسبه میزان تصحیح اوزان، می‌بایست هر کدام از وزنها به میزان تأثیر در تولید خطا تغییر کنند. خطای نهایی به‌صورت بر عکس در شبکه منتشر می‌شود و هر یک از اوزان به میزان سهم خود تنبیه و یا تشویق می‌شوند (پس انتشار خطا).

جمع مربع خطاها شاخص یادگیری است. در هر نقطه (هر مرحله الگوریتم) خلاف علامت گرادیان (مشتق منحنی) یا تابع مجموع مربع خطاها، نشان‌دهنده جهت تصحیح اوزان و مقدار گرادیان (مشتق جزئی) نشان‌دهنده مقدار خطاها است که با توجه به یک نرخ یادگیری(η) قابل اعمال در وزنها به‌صورت ذیل می‌باشد:

$$\Delta w_{current} = -\eta(\partial SSE / \partial w_{current}) \qquad (۱۵-۵)$$

ولی با توجه به مشکلات محاسبه مشتق جزئی در چنین شبکه پیچیده‌ای، میشل در سال ۱۹۹۷ توانست قواعد جایگزین ذیل را برای الگوریتم پس انتشار خطا ارائه دهد:

$$w_{ij.new} = w_{ij.current} + \Delta w_{ij} \qquad where \qquad \Delta w_{ij} = \eta \delta_j \qquad (۱۶-۵)$$

$$\delta_j = \begin{cases} output_j(۱ - output_j)\,(actual_j - output_j) & for\ output\ layer\ nodes \\ output_j(۱ - output_j) \sum_{dawnstream} W_{jk}\delta_j & for\ hidden\ layer\ nodes \end{cases} \qquad (۱۷-۵)$$

در این قواعد δ نشان دهنده مسئولیت هر گره در تولید خطای شبکه می‌باشد. حال برای فهم قواعد مثال را ادامه می‌دهیم. همان‌طور که به خاطر دارید، مقدار خروجی شبکه ۰/۸۷۵ محاسبه شد، حال فرض کنید که مقدار هدف یا جواب که در مجموعه داده‌ها برای این ورودی‌ها وجود داشته ۰/۸ باشد، در نتیجه خطای شبکه برابر ۰/۰۷۵- است. حال قواعد پس انتشار خطا را برای شبکهٔ مربوط به مثال به‌کار می‌بریم، چون گره نهایی شبکه، گره Z یک گره خروجی است محاسبات به شرح ذیل انجام می‌شوند:

$$\delta_z = output_z(۱ - output_z)(actual_z - output_z)$$
$$= ۰/۸۷۵(۱ - ۰/۸۷۵)(۰/۸ - ۰/۸۷۵) = -۰/۰۰۸۲$$
$$\Delta W_{oz} = \eta\delta_z(۱) = ۰/۱(-۰/۰۰۸۲)(۱) = -۰/۰۰۰۸۲$$
$$w_{oz,new} = w_{oz,curent} + \Delta w_{oz} = ۰/۵ - ۰/۰۰۰۸۲ = ۰/٤۹۹۱۸$$

همان‌طور که به خاطر دارید، W_{0z} دارای مقدار ۰/۵ بود که به‌عنوان مقدار ثابت در گره Z وارد می‌شد، که با استفاده از محاسبات فوق به مقدار ۰/٤۹۹۱۸ تصحیح گردید. از گره Z به سمت گره A می‌رویم و محاسبات را به همین صورت ادامه می‌دهیم و تمام اوزان را تصحیح می‌کنیم:

$$\delta_A = output_A(۱ - output_A) \sum_{downstream} W_{jk}\,\delta_j$$
$$\delta_A = ۰/۷۸۹۲(۱ - ۰/۷۸۹۲)(۰/۹)(-۰/۰۰۸۲) = -۰/۰۰۱۲۳$$

$$\Delta W_{AZ} = \eta\delta_z.output_A = ۰/۱(-۰/۰۰۸۲)(۰/۷۸۹۲) = -۰/۰۰۰٦٤۷$$

$$w_{AZ,new} = w_{AZ,current} + \Delta w_{AZ} = ۰/۹ - ۰/۰۰۰٦٤۷ = ۰/۸۹۹۳۵۳$$

$$\delta_B = output_B(۱ - output_B) \sum_{downstream} W_{jk}\delta_j$$

$$\Delta W_{BZ} = \eta\delta_z.output_B = ۰/۱(-۰/۰۰۸۲)(۰/۸۱۷٦) = -۰/۰۰۰٦۷$$
$$w_{BZ,new} = W_{BZ,current} + \Delta w_{BZ} = ۰/۹ - ۰/۰۰۰٦۷ = ۰/۸۹۹۳۳$$

$$\Delta W_{۱A} = \eta\delta_A x_۱ = ۰/۱(-۰/۰۰۱۲۳)(۰/٤) = -۰/۰۰۰۰٤۹۲$$
$$w_{۱A,new} = w_{۱A,current} + \Delta w_{۱A} = ۰/٦ - ۰/۰۰۰۰٤۹۲ = ۰/۵۹۹۹۵۰۸$$

$$\Delta W_{۲A} = \eta\delta_A x_۲ = ۰/۱(-۰/۰۰۱۲۳)(۰/۲) = -۰/۰۰۰۰۲٤٦$$
$$w_{۲A,new} = w_{۲A.current} + \Delta_{۲A} = ۰/۸ - ۰/۰۰۰۰۲٤٦ = ۰/۷۹۹۹۷۵٤$$

$$\Delta W_{۳A} = \eta\delta_A x_۳ = ۰/۱\,(-۰/۰۰۱۲۳)(۰/۷) = -۰/۰۰۰۰۸٦۱$$
$$w_{۳A,new} = w_{۳A,current} + \Delta w_{۳A} = ۰/٦ - ۰/۰۰۰۰۸٦۱ = ۰/۵۹۹۹۱۳۹$$

این فرایند را برای تمام اوزان انجام داده تا تمام اوزان به روز شوند. بر اساس اوزان تصحیح شده دوباره ورودیها را به شبکه داده و خطا را اندازه‌گیری می‌کنیم و دوباره اوزان را تصحیح کرده و این عمل را تا شرط توقف دنبال می‌کنیم.

شروط توقف

از آنجا که فرایند الگوریتم پس از انتشار خطا یک الگوریتم مبتنی بر تکرار است، می‌بایست، شروطی را برای توقف الگوریتم در نظر گرفت که برخی از آنها به شرح ذیل می‌باشند:

- بر اساس عدم تغییر در *SSE*
- بر اساس یک سری تکرارهای خاص
- بر اساس نسبت بهینه *SSE* آزمون نسبت به *SSE* آموزش
- براساس زمان اجرای الگوریتم

با توجه به موضوع شرط توقف نمی‌توان اثبات نمود که جواب الگوریتم یک بهینه کلی است بلکه یک بهینهٔ محلی است که البته می‌تواند برای مسائلی مانند دسته‌بندی در داده‌کاوی مناسب باشد.

برخی کاربردهای دسته‌بندی بر اساس شبکه‌های عصبی

الگوریتم یادگیری پس انتشار خطا که در فوق توضیح داده شد، یکی از چندین الگوریتم یادگیری شبکه‌های عصبی تنها در توپولوژی شبکه‌های عصبی چندلایه پیش‌فرض بود. الگوریتمها و ساختارهای مختلف شبکه‌های عصبی هر کدام می‌توانند کاربردهای زیادی را در عملیات‌های مختلف داشته باشند. به منظور درک صحیح‌تر، این حوزه‌ها مثال‌های زیر ارائه می‌شوند:

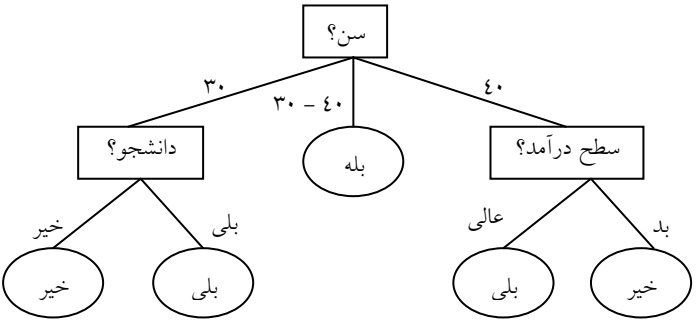
- دسته‌بندی مسافرین خطوط هواپیمایی بر مبنای اطلاعات سفرها در دسته‌بندی مسافرین کثیرالسفر و ارائه خدمات ویژه به آنها با استفاده از ایجاد مدلهای شبکه عصبی در خطوط هواپیمایی آمریکا.
- دسته‌بندی خانواده محصولات تولیدی بر اساس زمانهای تولید در سیستمهای تولید انعطاف‌پذیر کارخانجات چند منظوره ژاپن با استفاده از داده‌کاوی بر اساس شبکه‌های عصبی.
- ایجاد سیستم‌های خبره تحت وب در زمینه شناسایی بازارهای هدف محصولات متفاوت، با استفاده از آموزش شبکه‌های عصبی مصنوعی از طریق داده‌های خرید مشتریان تحت وب.
- دسته‌بندی فعالیتها در دسته‌های بحرانی و غیربحرانی در بحث برنامه‌ریزی و کنترل پروژه بر اساس بانک اطلاعات زمان تخصیص یافته به هر پروژه در شرکت‌های عمرانی اروپایی با به‌کارگیری مدل دسته‌بندی بر مبنای شبکه‌های عصبی مصنوعی.

بخش عمده‌ای از کاربردهای شبکه‌های عصبی در عملیاتهای تخمین و پیش‌بینی داده‌کاوی است و همچنین دسته‌بندی یکی از حوزه‌های مناسب برای کاربرد این شبکه‌ها می‌باشد. شبکه‌های خودسازمانده (SOM) و سایر شبکه‌های غیرنظارتی نیز می‌توانند در خوشه‌بندی کاربرد داشته باشند. لیکن در بحث داده‌کاوی آشنایی با شبکه‌های عصبی می‌تواند دریچه‌های جدیدی برای نوآوری و ایجاد کاربرد برای داده‌کاوی در حوزه‌های مختلف صنعت و کسب و کار به دنبال آورد.

درخت تصمیم

ساختار درخت تصمیم یک ساختار درختی، شبیه فلوچارت است. در این ساختار هر گره داخلی آزمونی را بر روی یک ویژگی مشخص می‌کند و هر شاخه خارج شده از این گره، دستاورد این آزمون را نشان می‌دهد و گره‌های برگ، دسته‌ها یا توزیع دسته‌ها را نشان می‌دهند. بالاترین گره در درخت، گرهٔ ریشه است. تصویر یک درخت تصمیم

نمونه در شکل (۵–۱۸) نشان داده شده است. این شکل مفهوم امکان خرید کامپیوتر توسط مشتری را نشان می‌دهد که پیش‌بینی می‌کند آیا مشتری در شعب فروشگاه علاقه‌مند به خرید کامپیوتر می‌باشد یا خیر؟ گره‌های داخلی با مستطیل و گره‌های برگ با بیضی مشخص شده‌اند.



شکل ۵–۱۸) نمونه‌ای از یک درخت تصمیم برای خرید کامپیوتر در شعب فروش کامپیوتر

شکل (۵–۱۸) نشان می‌دهد که آیا مشتری علاقه‌مند به خرید کامپیوتر است یا خیر؟ هر گره داخلی (غیر برگ) آزمایش یک ویژگی را نمایش می‌دهد و هر گره برگ یک دسته را نشان می‌دهد. که نشانگر خریدن (*yes*) یا نخریدن (*no*) کامپیوتر است.

خصوصیات درخت تصمیم

درخت تصمیم یکی از ابزارهای قوی و متداول برای دسته‌بندی و پیش‌بینی می‌باشد. درخت تصمیم برخلاف شبکه‌های عصبی به تولید قاعده می‌پردازد. در ساختار درخت تصمیم، پیش‌بینیِ به‌دست آمده از درخت در قالب یکسری قواعد توضیح داده می‌شود درحالی‌که در شبکه‌های عصبی تنها نتیجهٔ پیش‌بینی بیان می‌شود و چگونگی به‌دست آمدن آنها در خود شبکه پنهان می‌ماند. همچنین در درخت تصمیم بر خلاف شبکه‌های عصبی ضرورتی وجود ندارد که داده‌ها لزوماً به‌صورت عددی باشند.

در برخی موارد تنها صحت دسته‌بندی و پیش‌بینی مهم است و لزوماً ارائه توضیحی برای پیش‌بینیِ انجام شده، نیاز نیست. به‌عنوان مثال یک شرکت مخابراتی را در نظر بگیرید که می‌خواهد ببیند کدام‌یک از مشتریانش به خدمت جدیدی که ارائه می‌شود پاسخ مثبت خواهند داد. برای این شرکت صحت نتیجه پیش‌بینی مهم است و به‌علت و چگونگی پیش‌بینی نیازی نیست، در حالی‌که شرکت دیگری که قصد بازاریابی و کسب مشتریان جدید را دارد، علاقه‌مند است تا بداند ویژگیهای مشتریانی که احتمالاً به محصول جدیدِ این شرکت پاسخ مثبت می‌دهند، چیست؟ در واقع با اطلاع از این ویژگیها، شرکت می‌تواند به سراغ افرادی برود که با احتمال بیشتری به محصولات جدید این شرکت پاسخ مثبت می‌دهند. به‌عبارت دیگر این شرکت به یکسری قواعد برای بهبود فعالیت بازاریابی خود نیاز دارد. به‌طور مثال یکی از این قواعد می‌تواند به‌صورت زیر باشد:

«افراد متاهلی که از خود خانه دارند و درآمدی بالای ۱,۵ میلیون تومان درماه دارند به این محصول جدید پاسخ مثبت می‌دهند.»

درچنین مواقعی استفاده از درخت تصمیم نسبت به شبکه‌های عصبی ترجیح داده می‌شود. در مورد خصوصیات درخت تصمیم به موارد زیر می‌توان اشاره نمود:

- روش درخت تصمیم در تقسیم بندی داده‌ها به گروه‌های مختلف، به‌گونه‌ای است که هیچ داده‌ای حذف نمی‌شود (تعداد داده‌ها در گروه مادر با مجموع داده‌ها در شاخه‌های درختِ ایجاد شده، برابر هستند).
- استفاده از درخت تصمیم آسان می‌باشد.
- درک مدل ایجاد شده توسط درخت تصمیم آسان می‌باشد. به‌عبارت دیگر با وجود اینکه فهمیدن روش کار الگوریتمهای سازندهٔ درخت، چندان ساده نیست ولی فهمیدن نتایج به‌دست آمده از آنها آسان است.
- دسته‌بندیهایی که توسط درخت تصمیم ایجاد می‌شوند، از روی شباهت داده‌های ذخیره شده در پارامترهای پیش‌بینی کننده، قابل انجام می‌باشد.

روش کار درخت تصمیم

افرادی که بازی بیست سؤالی را انجام داده‌اند راحت‌تر روش کار درخت تصمیم را درک می‌کنند. در این بازی یک نفر مفهوم یا شیء خاصی را در ذهن خود در نظر می‌گیرد و شخص دیگری سعی می‌کند تا با انجام یک سری سؤالات که جواب آنها بلی یا خیر است، مفهوم یا شیء مورد نظر شخص اول را شناسایی نماید.

در ایجاد درخت تصمیم نیز یکسری سؤال وجود دارد و با مشخص شدن پاسخ هر سؤال یک سؤال دیگر پرسیده می‌شود. اگر سؤالها درست و مناسب با ویژگیها پرسیده شوند یک مجموعه کوتاه از سؤالات برای پیش‌بینی کردن دستهٔ مربوط به هر شیء جدید کافی می‌باشد.

ساختار کلی درخت تصمیم به این صورت است که یک گره ریشه در بالای آن و برگها در پایین آن می‌باشند. یک رکورد جدید در گره ریشه وارد می‌شود و در این گره یک آزمون صورت می‌گیرد تا معلوم شود که این رکورد به کدام‌یک از گره‌های فرزند (شاخه پایین‌تر) تعلق دارد. معمولاً روشهای مختلفی برای انتخاب این آزمون اولیه وجود دارد ولی هدف همه آنها یکی است یعنی «انتخاب روشی که بهترین جداسازی را در دسته‌های هدف انجام دهد. » این فرآیند آنقدر ادامه پیدا می‌کند تا رکورد جدید به گره برگ برسد. تمام رکوردهایی که به یک برگ از درخت می‌رسند در یک دسته قرار می‌گیرند. همچنین برای رسیدن از ریشه به یک برگ تنها یک راه وجود دارد و آن راه در واقع بیان قاعده‌ای است که برای دسته‌بندی رکوردها استفاده شده است. ممکن است تعداد زیادی برگ وجود داشته باشد که همگی یک دسته داشته باشند ولی هر برگ برای قرارگرفتن در دسته مورد نظر علت متفاوتی دارد. برای مثال در درختی که برای دسته‌بندی میوه‌ها بر اساس رنگ به‌کار رفته است سیب، گوجه فرنگی و توت فرنگی همگی دارای پیش‌بینی رنگ قرمز می‌باشند و در دسته مربوط به این رنگ قرار

می‌گیرند ولی درجه اطمینان هر یک از آنها متفاوت است زیرا سیبهای سبز، گوجه‌های زرد و توتهای سیاه نیز وجود دارند.

اثربخشی یک درخت تصمیم پس از ایجاد، باید اندازه‌گیری شود. برای این کار از داده‌های آزمون استفاده می‌شود که از داده‌های اولیهٔ ایجاد کنندهٔ درخت متفاوت می‌باشند. معیاری که در این قسمت اندازه‌گیری می‌شود عبارت است از: «درصد داده‌هایی که درست دسته‌بندی می‌شوند و دسته پیش‌بینی شده با دسته واقعی آنها یکسان است. » کیفیت شاخه‌های ایجاد شده نیز باید در نظر گرفته شوند. هر راهِ ایجاد شده از ریشه به یک برگ معادل یک قاعده است و البته بعضی از این قواعد از دیگر قواعد قویتر می‌باشند. در بعضی مواقع بریدن برخی شاخه‌های ضعیف‌تر درخت، باعث بهبود قدرت پیش‌بینی در شاخه‌های دیگر درخت می‌شود.

الگوریتم درخت تصمیم با انتخاب آزمونی شروع می‌شود که بهترین جداسازی را برای دسته‌ها انجام دهد. در مراحل بعدی، همین کار برای گره‌های بعدی با داده‌های کمتر صورت می‌گیرد تا بهترین قواعد ایجاد شوند و درخت باید آنقدر بزرگ شود که دیگر نتوان جداسازی بهتری را برای داده‌های گره انجام داد.

مهم‌ترین هدف از دسته‌بندی، به‌دست آوردن مدلی برای پیش‌بینی می‌باشد. بدین منظور از مجموعه‌ای به نام داده‌های آموزشی که مجموعه‌ای از متغیرها و رکوردها است، استفاده می‌کنیم. در جدول (۵-۱۰) مثالی از داده‌های آموزشی خرید خودرو آمده است.

جدول ۵- ۱۰) داده‌های آموزشی خرید خودرو

ریسک	نوع ماشین	سن
زیاد	خانوادگی	۲۳
زیاد	اسپورت	۱۷
زیاد	اسپورت	٤۳
کم	خانوادگی	٦۸
کم	باری	۳۲
زیاد	خانوادگی	۲۰

انواع متغیرهای موجود در داده‌های درخت تصمیم [۱]

در مسائل مرتبط با درختهای تصمیم با دو نوع از متغیرها مواجه هستیم:

- متغیرهای عددی مثل مشخصه «سن» که مقادیر آن عددی است.
- متغیرهای رده‌ای مثل مشخصه «نوع ماشین» که مقادیر آن متنی و گروهی است.

از این متغیرها برای پیش‌بینی متغیر هدف یا متغیر وابسته استفاده می‌کنیم. در مثال فوق، به متغیرهای «سن» و «نوع ماشین» در مثال بالا که متغیرهایی مستقل هستند، متغیر پیش‌بینی کننده گویند و به متغیرهای وابسته برچسب دسته[1] می‌گویند. در مثال بالا متغیر «ریسک» از نوع برچسب دسته می‌باشد.

نکته ۱: اگر متغیر وابسته از نوع عددی باشد مسئله به یک مسئله رگرسیون یا پیش‌بینی تبدیل خواهد شد و اگر این متغیر از نوع رده‌ای باشد با یک مسئله دسته‌بندی مواجه هستیم.

نکته ۲: درخت تصمیم می‌تواند یک درخت دودویی بوده و یا اینکه تعداد شاخه‌هایش بیشتر از دو نیز باشد. مثلاً برای یک متغیر رده‌ای به ازای هر مقدار می‌توان یک شاخه در نظر گرفت اما تمرکز ما در این بخش بر روی درختان دودویی می‌باشد.

مفاهیم اصلی در درختهای تصمیم

گره: به متغیر وابسته‌ای که آزمون روی آن صورت می‌گیرد، گره گفته می‌شود.

برگ: به متغیر مستقل یا برچسب دسته، برگ گفته می‌شود.

شاخه: به مقیاسی که خروجی ازآن تعیین می‌شود، شاخه گویند. نکته قابل توجه این است که برای متغیرهای عددی، تست به‌صورت $q_n : X_n < x_n$ و برای متغیرهای رده‌ای به‌صورت $q_n : X_n \subseteq x_n$ انجام می‌گیرد.

---

[1]- Class Label

چرا درخت تصمیم؟

- نتیجه آن نسبت به سایر مدلهای دسته‌بندی زودتر محاسبه می‌گردد.
- دقت آن نسبت با سایر مدلها قابل رقابت است.
- یادگیری آن ساده و آسان است.
- قواعد به‌دست آمده در آن آسان‌تر فهمیده می‌شود.

نکات قابل توجه برای استفاده از الگوریتمهای درخت تصمیم

برای استفاده از روشهای مربوط به درختهای تصمیم موارد زیر باید در نظر گرفته شوند:

- توضیحات ویژگی – ارزش[1]: داده‌های مورد نظر باید در یک فایل بوده و شکلی یکنواخت از همهٔ ویژگیها وجود داشته باشد. هر ویژگی می‌تواند مقادیر عددی یا گسسته داشته باشد، ولی ویژگیهایی که برای شرح نمونه‌ها استفاده می‌شوند نباید از یک حالت به حالت دیگر متفاوت باشند.
- دسته‌های ازپیش‌تعیین‌شده[2]: دسته‌بندی‌هایی که نمونه‌ها به آنها نسبت داده می‌شوند، قبلاً تخمین زده شده‌اند. در اصطلاحات علمی یادگیری ماشین به این موضوع یادگیری تحت کنترل گویند.
- دسته‌های گسسته[3]: دسته‌ها باید به صراحت شرح داده شوند. یک شیء می‌تواند به یک دستهٔ خاص تعلق داشته باشد یا خیر و می‌توان انتظار داشت که تعداد نمونه‌ها بیشتر از دسته‌ها باشد.

---

[1]- Attribute Value Description
[2]- Predefined Classes
[3]- Discrete Classes

- دادهٔ کافی[1]: تعداد داده‌های مورد نیاز از عواملی مانند تعداد ویژگیها، دسته‌ها و پیچیدگی مدل دسته‌بندی تأثیر می‌گیرد. همین‌طورکه این عوامل افزایش می‌یابد، داده‌های بیشتری برای ساخت یک مدل قابل اطمینان مورد نیاز خواهد بود.

مراحل ایجاد درخت تصمیم

پیدایش درخت تصمیم از دو مرحله تشکیل شده است:

- مرحله رشد و ایجاد درخت
- مرحله هرس درخت با هدف حداقل کردن خطای پیش‌بینی

تمام الگوریتمهای ایجاد درخت، با نگرش بالا به پایین اجرا می‌شوند. روشهای متفاوتی برای ایجاد درخت وجود دارند:

- روش انتخاب معیار برای شاخه زدن
- روشهای مدیریت پایگاه داده‌های بزرگ[2]

الگوریتم ایجاد درخت دودویی

مراحل ایجاد یک درخت دودویی بر اساس الگوریتم ذیل می‌باشد.

```
Apply (split selection) to D to find the splitting criterion
If n split
        Use best split to partition D to D1 and D2
        Build tree(n1,D1,ss)
        Build tree(n2,D2,ss)
End if
```

شکل ۵-۱۹) مراحل ایجاد درخت

[1]- Sufficient Data
[2]- Data Access

روشهای انتخاب نقطه شکست و انشعاب[1]

در این قسمت روشهای مبتنی بر ناخالصی[2] را برای انتخاب معیار استفاده می‌کنیم و آن را با $Imp\,\theta$ نمایش می‌دهیم. هدف، کاهش این تابع یا کاهش گوناگونی در هر سطح می‌باشد تا جایی که به گره برگ برسیم. در انتخاب نقطه شکست، متغیری که زیرگروهش به یک دسته از کلاس تبدیل شود اولویت دارد. (برای راحتی از (I) استفاده می‌کنیم)

انواع روشهای انتخاب نقطه شکست عبارتند از:

- شاخص جینی[3] : $gini(T) = ۱ - \sum P_i^۲$
- آنتروپی[4] : $Entropy(T) = -\sum P_i . \log_۲^{P_i}$
- کارت[5]
- $۲P_i$ (P_i فراوانی نسبی از کلاس j در درخت T می‌باشد.)
- $Min(P_i)$
- $C_{٤/٥}$

در روش شاخص جینی همه متغیرهای گره‌ها را امتحان کرده و آن متغیری که از همه بهتر باشد را انتخاب می‌کنیم. حال بهترین انتخاب برای تقسیم مجموعه S به دو مجموعه $S_۱$ و $S_۲$ از معیار زیر تبعیت می‌کند یعنی ماکزیمم کردن تابع زیر:

$$S = \frac{|S_۱|}{|S|} . I(S_۱) + \frac{|S_۲|}{|S|} . I(S_۲) \qquad (۱۸-۵)$$

ساخت یک نمونه درخت تصمیم با استفاده از روش شاخص جینی

مثال: درخت تصمیم را برای مجموعه زیر رسم کنید.

---

[1]- Split Selection
[2]- Impurity - Based
[3]- Gini Index
[4]- Entropy
[5]- CART

جدول ۵-۱۱) داده‌های مورد استفاده در درخت تصمیم

ریسک	نوع ماشین	سن
زیاد	خانوادگی	۲۳
زیاد	اسپورت	۱۷
زیاد	اسپورت	٤۳
کم	خانوادگی	٦۸
کم	باری	۳۲
زیاد	خانوادگی	۲۰

ابتدا جدول را بر اساس متغیر «سن» به‌صورت صعودی مرتب می‌کنیم.

جدول ۵-۱۲) داده‌های مرتب شده

ریسک	نوع ماشین	سن
زیاد	اسپورت	۱۷
زیاد	خانوادگی	۲۰
زیاد	خانوادگی	۲۳
کم	باری	۳۲
زیاد	اسپورت	٤۳
کم	خانوادگی	٦۸

حال از متد شاخص جینی برای انتخاب نقطهٔ انشعاب استفاده می‌کنیم. هر دو متغیر «سن» و «نوع ماشین» را بررسی می‌کنیم.

اختصارات استفاده شده در روش ایجاد درخت عبارتند از:

ریسک کم $L:$ ریسک زیاد $H:$

بچه چپ $L:$ بچه راست $R:$

و همچنین داریم: $(gini(T)=۱-\sum P_j^۲)$

برای هر کدام از مقادیر عددی و هر دسته متغیر طبقه‌ای، محاسبات را گام‌به‌گام با استفاده از مفروضات فوق و داده‌های موجود انجام می‌دهیم. از آنجا که نمی‌دانیم آستانه تصمیم متغیر پیوسته سن چند می‌باشد، تمام مقادیر ممکن متغیر سن را بررسی می‌کنیم.

۱- ۱۷ = > «سن»

پس از مرتب کردن صعودی جدول بر اساس متغیر «سن»، برای هر مقدار از متغیرها جدولی را مطابق زیر تشکیل می‌دهیم. به‌عنوان مثال خانه چپ در سطر دوم بیانگر تعداد رکوردهایی است که سن آنها کمتر و یا مساوی ۱۷ بوده و ریسک در آنها از نوع «زیاد» بوده است. در اینجا فقط همان رکورد اول یعنی ۱۷ = «سن» با شرط مسئله مطابقت دارد، پس مقدار ۱ را در خانه قرار می‌دهیم و سایر خانه‌های جدول نیز با همین ترتیب مقدار دهی می‌شوند.

فرض می‌کنیم که S_1 معادل با مجموع اعداد در سطر اول جدول و S_2 معادل با مجموع اعداد درسطر دوم جدول و S معادل با مجموع اعداد در کل جدول است.

	H	L
L	۱	۰
R	۳	۲

$I(S_1): ۱-(۱/۱)^۲-(۰/۱)^۲=۱-۱-۰=۰$

$I(S_2): ۱-(۳/۵)^۲-(۲/۵)^۲=۱-۹/۲۵-۴/۲۵=۰/۴۸$

$|s_1|=۱, |s_2|=۵, |s|=۶ \Rightarrow I(S):|۱|/|۶|*۰+|۵|/|۶|*۰/۴۸$

$=۰+۵/۶*۰/۴۸=۰/۴$

۲- ۲۰ = > «سن»

	H	L
L	۲	۰
R	۲	۲

$I(S_1): ۱-(۲/۲)^۲-(۰/۲)^۲=۱-۱-۰=۰$

$I(S_۲): ۱-(۲/٤)^۲-(۲/٤)^۲ = ۱-٤/۱٦-٤/۱٦ = ۰/٥$

$|s_۱|=۲ , |s_۲|=٤ , |s|=٦ \Rightarrow I(s) = |۲|/|٦|*۰+|٤|/|٦|*۰/٥$

$= ٤/٦*۰/٥ = ۰/۳۳$

۳- ۲۳ = > «سن»

	H	L
L	۳	۰
R	۱	۲

$I(S_۱): ۱-(۳/۳)^۲-(۰/۳)^۲ = ۱-۱-۰ = ۰$

$I(S_۲): \ ۱-(۱/۳)^۲-(۲/۳)^۲ = ۱-۱/۹-٤/۹ = ۰/٤٤٤٤$

$|s_۱|=۳ , |s_۲|=۳ , |s|=٦ \Rightarrow I(s): |۳|/|٦|*۰+|۳|/|٦|*۰/٤٤٤$

$= ۳/٦*۰/٤٤٤ = ۰/۲۲۲$

٤- ۳۲ = > «سن»

	H	L
L	۳	۱
R	۱	۱

$I(S_۱): ۱-(۳/٤)^۲-(۱/٤)^۲ = ۱-۹/۱٦-۱/۱٦ = ۰/۳۷٥$

$I(S_۲): \ ۱-(۱/۲)^۲-(۱/۲)^۲ = ۱-۱/٤-۱/٤ = ۰/٥$

$|s_۱|=٤ , |s_۲|=۲ , |s|=٦ \Rightarrow I(s): |٤|/|٦|*۰/۳۷٥+|۲|/|٦|*۰/٥$

$= ٤/٦*۰/۳۷٥+۲/٦*۰/٥ = ۰/٤۱٦٦$

٥- ٤۳ = > «سن»

	H	L
L	٤	۱
R	۰	۱

$I(S_۱): ۱-(٤/٥)^۲-(۱/٥)^۲ = ۱-۱٦/۲٥-۱/۲٥ = ۰.۳۲$

$I(S_۲): \ ۱-(۰/۱)^۲-(۱/۱)^۲ = ۱-۰-۱ = ۰$

$|s_1|=۵, |s_2|=۱, |s|=٦ \Rightarrow I(s):|۵|/|٦|*۰/۳۲+|۱|/|٦|*۰$
$=۵/٦*۰/۳۲+۰=۰/۲٦٦$

«سن» = < ٦۸ -٦

	H	L
L	٤	۲
R	۰	۰

$I(S_1): ۱-(٤/٦)^۲-(۲/٦)^۲=۱-۱٦/۳٦-٤/۳٦=۰.٤٤٤$
$I(S_2): ۱-۰-۰=۱$
$|s_1|=٦, |s_2|=۰, |s|=٦ \Rightarrow I(s);|٦|/|٦|*۰/۳۲+|۰|/|٦|*۱$
$=٦/٦*۰/٤٤٤+۰=۰/٤٤٤$

برای بررسی متغیرهای غیر عددی- طبقه‌ای و به‌منظور سهولت در انجام کار، جدول فراوانی هر دسته را از روی همان جدول اولیه برای متغیرهای غیرعددی تشکیل داده و سپس محاسبات را مشابه قبل انجام می‌دهیم. برای این‌کار جدولی را در نظر گرفته و رکوردهای آن را با مقادیر متغیرهای غیرعددی پرمی‌کنیم. البته هر مقدار فقط باید یکبار در نظر گرفته شود. مثلاً فرض کنید که در یک رکورد متغیر «نوع ماشین» برابر با مقدار «اسپورت» است. اولین رکورد جدول جدید را با توجه به تعداد دسته‌های «ریسک زیاد» و «ریسک کم» برای این مقدار مشخص نموده و در جدول قرار می‌دهیم و به همین ترتیب برای سایر متغیرهای موجود در جدول نیز رکورد ایجاد می‌کنیم. در نظر داشته باشید که رکورد تکراری به ازای مقادیر مختلف متغیرها وجود نداشته باشد. پس از بررسی کلیه رکوردها جدول ذیل به‌عنوان نتیجه حاصل می‌شود.

جدول ۵-۱۳) جدول نتیجه درخت تصمیم

نوع ماشین	کم	زیاد
اسپورت	۰	۲
خانوادگی	۱	۲
باری	۱	۰

برای پرکردن جدول محاسباتی در نظر داشته باشید که سطر اول خانه سمت چپ نمایانگر تعداد رکوردهایی است که متغیر «نوع ماشین» آن برابر با «اسپورت» بوده و دسته ریسک در آن از نوع «زیاد» می‌باشد. خانه سمت چپ در سطر دوم بیانگر تعداد رکوردهایی از جدول فوق است که متغیر «نوع ماشین» آن برابر با اسپورت نبوده و دسته ریسک در آن از نوع زیاد می‌باشد.

۷- اسپورت = «نوع ماشین»

	H	L
اسپورت = نوع ماشین	۲	۰
اسپورت ≠ نوع ماشین	۲	۲

$I(S_1): ۱-(۲/۲)^۲-(۰/۲)^۲=۱-۱-۰=۰$

$I(S_2): \ ۱-(۲/٤)^۲-(۲/٤)^۲=۱-۱٦/٤-۱٦/٤=۰.۵$

$|s_1|=۲\ ,|s_2|=٤\ ,|s|=٦\Rightarrow I(s);|۲|/|٦|*۰/۵$

$=٤/٦*۰/۵+۰=۰/۳۳۳$

۸- خانوادگی = «نوع ماشین»

	H	L
خانوادگی = نوع ماشین	۲	۱
خانوادگی ≠ نوع ماشین	۲	۱

$I(S_1):۱-(۲/۳)^۲-(۱/۳)^۲=۱-٤/۹-۱/۹=۰/٤٤٤$

$I(S_2):۱-(۲/۳)^۲-(۱/۳)^۲=۱-٤/۹-۱/۹=۰/٤٤٤$

$|s_1|=۳\ ,|s_2|=۳\ ,|s|=٦\Rightarrow I(s):|۳|/|٦|*۰.٤٤٤+|۳|/|٦|*۰.٤٤٤=۰.٤٤٤$

۹- باری = نوع ماشین

	H	L
باری = نوع ماشین	۰	۱
باری ≠ نوع ماشین	٤	۱

$I(S_1): ۱-(۰/۱)^۲-(۱/۱)^۲=۱-۰-۱=۰$

$I(S_2): \ ۱-۱٦/۲٥-۱/۲٥=۰/۳۲$

$|s_1|=۱ , |s_2|=٥ , |s|=٦ \Rightarrow I(s);|۱|/|٦|*۰+|٥|/|٦|*۰/۳۲$

$=٥/٦*۰/۳۲+۰=۰/۲٦٦$

پس از بررسی کلیه حالتها، حداقل $I(S)$ ها را به‌دست می آوریم:

$Min\{۰/٤ , ۰/۳۳ , ۰/۲۲۲ , ۰/٤۱٦٦ , ۰/۲٦٦ , ۰/٤٤٤ , ۰/۳۳۳ , ۰/۲٦٦ , ۰/٤٤٤\}=۰/۲۲۲$

پس معیار ۲۳ => سن، را به‌عنوان نقطه انشعاب انتخاب می‌کنیم.

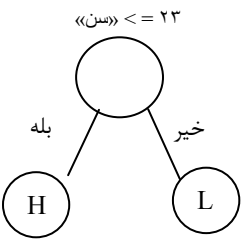
$Age<=۲۳=\{۱۷ , ۲۰ , ۲۳\}$

داده‌های مطابق با این شرط در جدول ذیل نمایش داده شده‌اند:

جدول ٥- ١٤) داده‌های مورد استفاده در شرط Age <=۲۳

سن	نوع ماشین	ریسک
۱۷	اسپورت	زیاد
۲۰	اسپورت	زیاد
۲۳	خانوادگی	زیاد

چون برچسب دستهٔ این مجموعه همه «زیاد» می‌باشد درخت به شکل زیر ایجاد می‌شود:



شکل ٥-۲۰) دسته‌بندی ایجاد شده در مرحله اول

حال جدول زیر را بر اساس « ۲۳ > سن » را تشکیل می‌دهیم تا معیار انشعاب بعدی با استفاده از همان روش فوق مجددا برای این بخش از داده‌ها نیزانتخاب شود:

جدول ۵- ۱۵) داده‌های مورد استفاده در شرط Age> =۲۳

ریسک	نوع ماشین	سن
کم	باری	۳۲
زیاد	اسپورت	٤۳
کم	خانوادگی	٦۸

۱۰- اسپورت = «نوع ماشین»

	H	L
اسپورت = نوع ماشین	۱	۰
اسپورت ≠ نوع ماشین	۰	۲

$I(S_1): ۱-(۱/۱)^۲-(۰/۱)^۲=۱-۰-۱=۰$

$I(S_2): \ ۱-(۰/۲)^۲-(۲/۲)^۲=۱-۰-۱=۰$

$|s_1|=۱, |s_2|=۲, |s|=۳ \Rightarrow I(s):|۱|/|۳|*۰+|۲|/|۳|*۰=۰$

۱۱- باری = «نوع ماشین»

	H	L
باری = نوع ماشین	۰	۱
باری ≠ نوع ماشین	۱	۱

$I(S_1): ۱-(۰/۱)^۲-(۱/۱)^۲=۱-۰-۱=۰$

$I(S_2): \ ۱-(۱/۲)^۲-(۱/۲)^۲=۱-۱/٤-۱/٤=۰.۵$

$|s_1|=۱, |s_2|=۲, |s|=۳ \Rightarrow I(s):|۱|/|۳|*۰+|٤|/|۳|*۰/۵=۰/۳۳۳$

۱۲- خانوادگی = «نوع ماشین»

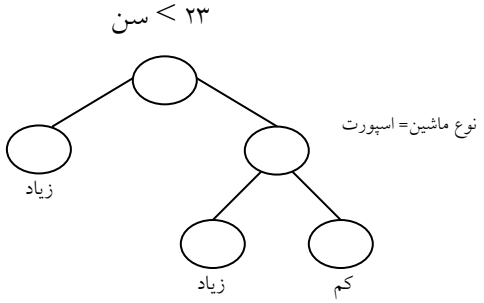
	H	L
خانوادگی = نوع ماشین	۰	۱
خانوادگی ≠ نوع ماشین	۱	۱

$I(S_1): \ ۱-(۰/۱)^۲-(۱/۱)^۲ = ۱-۰-۱=۰$

$I(S_2): \ ۱-(۱/۲)^۲-(۱/۲)^۲ = ۱-۱/٤-۱/٤=۰/۵$

$|s_1|=۱, |s_2|=۲, |s|=۳ \Rightarrow I(s):|۱|/|۳|*۰+|۲|/|۳|*۰/۵=۰/۳۳۳$

پس از بررسی تمام حالات حداقل *I(S)* بین آنها ۰ می‌باشد $(I(S)=0)$ پس درخت به شکل زیر تکمیل می گردد. این درخت، درخت نهایی است چرا که تمام برگهای آن به برچسب دسته ختم شده‌اند.



شکل ۵-۲۱) درخت نهایی

الگوریتم کارت

این الگوریتم یکی دیگر از روشهای ایجاد درخت تصمیم است و به وسیلهٔ بریمن و همکارانش در سال ۱۹۸٤ ایجاد شد. [*۱*] بسیاری از بسته‌های نرم افرازی موجود، این الگوریتم را دارا بوده و یا اینکه با تغییرات کوچکی قابلیت ارائه این الگوریتم را دارند. در ابتدا تعدادی رکورد داریم که دستهٔ آنها از قبل معلوم می‌باشد. (به‌عبارتی متغیر وابسته در آنها معلوم است) هدف، ایجاد درختی است که بتوان به‌وسیله آن متغیر وابسته یا همان برچسب دسته را برای یک رکورد جدید پیش‌بینی نمود.

روش کارت شاخه‌های خود را به‌صورت دوتایی و تنها بر اساس یک فیلد (متغیر مستقل) انشعاب می‌زند یعنی هر گروه غیر برگ آن، به دو گروه دیگر تفکیک می‌شود. حال اولین کار این است که کدامیک از فیلدها بهترین شاخه را ایجاد می‌کنند. بهترین شاخه زدن، هنگامی رخ می‌دهد که شاخه‌های حاصل به‌گونه‌ای ایجاد شوند که در هر شاخه یک دسته بر سایر دسته‌ها غلبه کند. یکی از مفاهیم کاربردی در این خصوص واژه «گوناگونی» است. گوناگونی[1] معیاری است که برای ارزیابی شاخه‌ها به‌کار می‌رود. برای محاسبه گوناگونیِ یک مجموعه از رکوردها، روشهای بسیاری وجود دارد که در تمامی آنها «گوناگونی زیاد» عبارت است از وجود دسته‌های گوناگون در درون یک مجموعه و «گوناگونی کم» عبارت است از وجود دسته‌های غیر گوناگون در درون آن مجموعه. بهترین شاخه زدن آن است که «گوناگونی» مجموعه‌ها را تا حد امکان کم کند. برخی از معیارهای محاسبه گوناگونی عبارتند از:

- $\min(P(C_1), P(C_2))$
- $2P(C_1)P(C_2)$
- $[P(C_2)\log P(C_1)]+[P(C_1)\log P(C_2)]$

در واقع ما می‌خواهیم مقدار زیر را حداکثر کنیم:

[(بچه‌های راست) گوناگونی) + ((بچه‌های چپ) گوناگونی)] – (قبل از انشعاب) گوناگونی

برای هر کدام از فیلدها سعی می‌کنیم تا با کمک یکی از فرمولهای محاسبه گوناگونی، حداقل مقدار گوناگونی ایجاد شده را به‌دست آوریم. سپس با مقایسه فرمول فوق قبل و بعد از شاخه زدن بوسیلهٔ همهٔ فیلدها، بهترین فیلدی که کمترین گوناگونی را ایجاد می‌کند انتخاب کرده و بر اساس آن دو شاخه می‌زنیم.

در مرحله بعد دو شاخه داریم که هر کدام دارای یکسری رکورد می‌باشند (هریک از رکوردهای گره بالاتر در یکی از شاخه‌ها قرارگرفته است). حال برای هر شاخه مثل قبل عمل می‌کنیم. یعنی برای هر یک از آنها دوباره یک فیلد را طوری انتخاب می‌کنیم

---

[1]- Diversity

که بتوان بهترین شاخه‌های جدید را با حداقل گوناگونی ایجاد نمود. این مراحل را آنقدر ادامه می‌دهیم تا در هر زیر شاخه به گره‌ای برسیم که ایجاد شاخه جدید، گوناگونی را تغییر نمی‌دهد. به این گره نهایی برگ گفته می‌شود.

ارزیابی درخت ایجاد شده

برای ارزیابی درخت ایجادشده توسط روشهای مختلف، معیارهای متفاوتی وجود دارند. یکی از مهم‌ترین و اصلی‌ترین این معیارها محاسبه نرخ خطا در درخت می‌باشد. برای محاسبه نرخ خطا در درخت ابتدا باید نرخ خطا در هر برگ را به‌دست آوریم. نرخ خطا در هر برگ عبارت است از نسبت تعداد رکوردهایی که دسته آنها درست پیش‌بینی نشده است. مثلاً اگر در یک برگ ۱۰ رکورد وجود داشته باشد و برای این رکوردها کلاس A پیش‌بینی شده باشد و حال آنکه تنها ۸ عدد از این رکوردها واقعا دارای کلاس A باشند و دوتای دیگر متعلق به کلاس دیگری باشند آنگاه نرخ خطا ۰/۲۰ می‌باشد. پس از محاسبه نرخ خطا در هر شاخه، برای محاسبه نرخ خطای کل درخت مجموع وزنی نرخ خطاهای برگها را به‌دست می‌آوریم (وزن هر برگ در واقع نسبت جمعیت آن برگ به کل جمعیت رکوردهای موجود در آن برگ می‌باشد).

کیفیت درخت حاصله نیز مهم می‌باشد. فرض کنید هدف، پیش‌بینی قد افراد است و دو دستهٔ کوتاه و بلند برای افراد در نظر گرفته شده است. یک مجموعه ۱۱ نفری از افراد وجود دارند که همگی بجز محمد که کمتر از ۲۸ سال دارد، قدشان کوتاه بوده و بالای ۲۸ سال سن دارند. اگر این گره را به دو شاخه تقسیم کنیم ممکن است قاعده‌ای مانند زیر ایجاد شود:

«افراد کمتر از ۲۸ سال که نام آنها محمد است، بلند قد هستند. »

این شاخه زدن با آنکه نرخ خطای درخت را برای مجموعه آموزشی کاهش می‌دهد ولی باعث ایجاد یک قاعده بدون کیفیت می‌شود. برای جلوگیری از ایجاد چنین

قواعدی در بعضی از شاخه‌ها که شرایط خاصی در آنجا وجود دارد، عملیات هرس[1] صورت می‌گیرد. این کار با آنکه نرخ خطا را افزایش می‌دهد ولی از ایجاد بعضی قواعد ناکارآمد جلوگیری می‌کند. یعنی با افزایش نرخ خطا در آموزش، نرخ خطا را در آزمون کاهش می‌دهد و در واقع مدلی با تعمیم بهتر ایجاد می‌کند. برای انجام عمل هرس، روش خاصی وجود دارد که در بخش بعد به آن خواهیم پرداخت. همچنین باید به این نکته توجه داشت که عملیات هرس به‌گونه‌ای صورت گیرد که خطا از مقدار معینی بیشتر نشود. بعد از هرس‌کردنِ شاخه‌های زائد، عملکرد درخت جدید را مورد بررسی قرار داده تا اطمینان حاصل شود که نرخ خطای محاسبه شده بر اساس این مجموعهٔ آموزشی با نرخ خطای به‌دست آمده از مجموعهٔ آزمایشیِ دیگر، تفاوت زیادی نداشته باشد. البته درصورت وجود تفاوت زیاد باید درخت ایجاد شده را مورد بازنگری قرار داده و با تغییراتی سعی در بهبود روش پیش‌بینی درخت شود.

پس از توضیح چگونگی روش دسته‌بندی در الگوریتم کارت باید به این نکته اشاره نمود که الگوریتمهای دیگر ایجاد درخت تصمیم مانند *C4. 5* و *CHAID* نیز برای دسته‌بندی، ساختار تقریباً مشابهی دارند و هدف همه آنها به‌دست آوردن درختی با کیفیت بالا و نرخ خطای کم در دسته‌بندی داده‌ها می‌باشد و بیشتر تفاوتها در شیوه شاخه زدن و هرس شاخه‌ها است.

هرس‌کردن درخت تصمیم

دور انداختن یک یا چند زیردرخت و جایگزینی آنها با برگها، ساختار درخت تصمیم را ساده می‌سازد. در جایگزینی زیر درخت با یک برگ، انتظار می‌رود نرخ خطای پیش‌بینی شده کاهش یافته و کیفیت مدل دسته‌بندی افزایش یابد. ولی محاسبه نمودن نرخ خطا ساده نیست. از طرفی محاسبه نرخ خطا فقط بر اساس اطلاعات یک مجموعه دادهٔ آموزشی، تخمین مناسبی را ارائه نمی‌کند. ایدهٔ هرس کردنِ درخت تصمیم، باعث

---

[1]- Pruning

از بین رفتن بخشهایی از درخت (زیر درختها) که در دقت و صحت دسته‌بندی نمونه‌های آزمایشی، مشارکت نمی‌کنند، می‌شود و همچنین درختی با پیچیدگی کمتر و بنابراین قابلیت درک بیشتر ایجاد می‌کند.

زمانی‌که درخت تصمیم ساخته شد، بسیاری از شاخه‌ها به علت اختلال و یا خلاصه-سازی در داده‌های آموزشی، نابهنجاری‌هایی را در مدل منعکس می‌کنند. روشهای هرس‌کردن درختها به مشکل بیش‌برازش[1] اشاره می‌کنند. چنین روشهایی عموماً از ابزارهای آماری برای از بین بردن شاخه‌هائی که کمترین قابلیت اطمینان را دارند، استفاده می‌کنند که عموماً منجر به دسته‌بندی سریعتر و بهبود در میزان توانایی درخت در جهت دسته‌بندی صحیح داده‌های مستقل آزمون، می‌شود.

استخراج قواعد دسته‌بندی از درختهای تصمیم

آیا امکان به‌دست آوردن قواعد از درخت تصمیم وجود دارد؟ دانش نمایش داده شده در درختهای تصمیم‌گیری را می‌توان استخراج نمود و در قالب قواعد دسته‌بندی «اگر–آنگاه» نمایش داد. برای هر مسیری که از ریشه تا یک برگ وجود دارد، یک قاعده ایجاد می‌شود. هر جفت ویژگی – ارزش که در طولِ مسیرِ مورد نظر برای ایجاد قاعده وجود دارند، یک ترکیب عطفی (و) در بخش مقدم قاعده (بخش اگر) ایجاد می‌کند. گره برگ، دستهٔ پیش‌بینی شده را نگه داشته و بخش تالی قاعده (بخش آنگاه) را شکل می‌دهد. قواعد «اگر– آنگاه» برای درک ساده‌تر هستند به‌خصوص اگر درخت مفروض بسیار بزرگ باشد. در اینجا به بررسی استخراج قواعد به‌دست آمده از شکل (۵–۱۸) پرداخته می‌شود. در درخت تصمیم شکل (۵–۱۸) با ردیابی مسیرها از گره ریشه تا هر برگ موجود در درخت به قواعد دسته‌بندی «اگر–آنگاه» زیر می‌رسیم:

IF "سن" = "< ۳۰" ، دانشجو = خیر *THEN* کامپیوتر نمی‌خرد

---

[1]- Overfitting

کامپیوتر می‌خرد *THEN* بله = دانشجو ، "۳۰ >" = "سن" *IF*

کامپیوتر می‌خرد *THEN* "۳۰ – ۴۰" = "سن" *IF*

کامپیوتر می‌خرد *THEN* عالی = سطح درآمد، "۴۰ >" = سن *IF*

کامپیوتر نمی‌خرد *THEN* بد = سطح درآمد، "۴۰ >" = "سن" *IF*

نقاط قوت درخت تصمیم

درخت تصمیم به ما این توانایی را می‌دهد که پیش‌بینیهای خود را در قالب یکسری قواعد ارائه دهیم. درخت تصمیم نیاز به محاسبات خیلی پیچیده برای دسته‌بندی داده‌ها ندارد. درخت تصمیم برای انواع مختلف داده‌ها از قبیل داده‌های عددی، پیوسته یا طبقه‌ای قابل استفاده می‌باشد. درخت تصمیم نشان می‌دهد که کدام فیلد یا متغیرها تأثیرات مهمی در پیش‌بینی و دسته‌بندی دارند.

نقاط ضعف درخت تصمیم

بعضی از روشهای درخت تصمیم تنها می‌توانند در مورد متغیرهای هدف دودویی (بله یا خیر- پذیرش یا عدم پذیرش) دسته‌بندی و پیش‌بینی انجام دهند و در بعضی از آنها هنگامی که تعداد مثالهای هر دسته کم باشد نرخ خطا بالا می‌رود. برخی الگوریتمهای ایجاد درخت به حافظه زیادی نیاز دارند زیرا برای پیدا کردن بهترین فیلد، وضعیت هر فیلد نگهداری می‌شود که این عملیات نیاز به حافظه زیادی دارد. همچنین در قسمت هرس شاخه‌ها نیز برای انتخاب بهترین زیر درختی که می‌توان آن را برش داد، وضعیت هر زیرشاخه را باید بخاطر سپرد. اکثر الگوریتمهای درخت تصمیم در هر گره تنها یک فیلد را برای شاخه زدن در نظر می‌گیرند.

پیش‌بینی[1]

پیش‌بینی، عبارت است از تعیین یک مقدار حقیقی پیوسته برای یک متغیر وابسته، بر حسب مقادیر متغیر یا متغیرهای مستقل: مانند پیش‌بینی حقوق فارغ التحصیلان با ۱۰ سال تجربه کاری یا فروش بالقوه یک محصول جدید بر حسب قیمت آن. مهم‌ترین روش مورد استفاده در پیش‌بینی عددی، رگرسیون است. البته برخی دیگر از روشهای دسته‌بندی نظیر الگوریتم پس‌انتشار و *SVM* نیز می‌توانند به‌عنوان روشهای پیش‌بینی مورد استفاده قرار گیرند. تحلیل رگرسیون برای مدلسازی روابط بین یک یا چند متغیر مستقل (پیش‌بینی کننده) و یک متغیر پاسخ (وابسته)، با مقدار پیوسته به کار می‌رود. در داده‌کاوی، متغیرهای مستقل همان ویژگیهای تشریح شده برای هر نمونه یا مشاهده می‌باشند. معمولاً مقادیر متغیرهای مستقل معلوم است. هر چند با استفاده از تکنیکهای خاصی، می‌توان مواردی‌که در آنها بعضی از مقادیرمتغیرها از بین رفته‌اند را نیز پیش‌بینی کرد.

برخی از مسائل پیش‌بینی با استفاده از رگرسیون خطی قابل حل می‌باشد، هر چند در بسیاری از موارد با به کار بردن روشهای تبدیل و تغییر متغیر، می‌توان یک مسئله غیرخطی را با استفاده از رگرسیون خطی حل کرد.

رگرسیون خطی (تک متغیره)

اگر X متغیر مستقل و Y متغیر وابسته باشد، آن گاه معادله رگرسیون خطی تک متغیره به‌صورت رابطه $y = w_0 + w_1 x$ خواهد بود.

فرض کنید D مجموعه داده‌های آموزشی یک جامعه به‌صورت $(X_1, Y_1)....(X_D, Y_D)$ باشد. ضرایب رگرسیون خطی، بر اساس روش کمترین مربعات خطا به‌دست می‌آید.

---

[1]- Prediction

پس از تعیین مقادیر w_0 و w_1 ، با جایگذاری مقدار متغیر مستقل (x) در رابطه $y = w_0 + w_1 x$ می‌توان، مقدار متناظر متغیر وابسته y را پیش‌بینی نمود.

$$w_1 = \frac{\sum_{i=1}^{|D|}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{|D|}(x_i - \bar{x})^2} \qquad (۱۹-۵)$$

$$w_0 = \bar{y} - w_1 \bar{x} \qquad (۲۰-۵)$$

رگرسیون خطی (چند متغیره)

در این روش تعداد متغیرهای مستقل در معادله رگرسیونی بیش از یکی است. مثلاً فرض کنید مقادیر x_i، نمونه‌های آزمایشی n بعدی (ویژگی) باشد که برچسب کلاس آنها y_iاست. در این صورت معادله رگرسیونی به شکل رابطه (۵-۲۱) خواهد بود. این معادله با استفاده از روش حداقل مربعات و نوشتن معادلات هم زمان یا توسط نرم‌افراز قابل حل است.

$$y = w_0 + w_1 x_1 + w_2 x_2 \qquad (۲۱-۵)$$

رگرسیون غیرخطی

اگر داده‌ها، یک وابستگی خطی را نشان ندهند، چگونه می‌توان از مدل رگرسیونی استفاده کرد؟ (مثلاً اگر وابستگیها به‌صورت یک تابع چندجمله‌ای باشد) در برخی از این حالات با استفاده از تکنیکهای تبدیل وتغییرمتغیر می‌توان مدل غیرخطی را به یک مدل رگرسیون خطی تبدیل کرده و براساس روش حداقل مربعات مسئله را حل کنیم. رابطه (۵-۲۲) نمونه‌ای از یک مدل رگرسیون غیرخطی، که یک معادله درجه ۳ است را نشان می‌دهد. این معادله با تغییر متغیرهای رابطه (۵-۲۳) به معادله (۵-۲٤) که یک رگرسیون چند جمله‌ای است تبدیل می‌شود.

$$y = w_0 = w_1 x + w_2 x^2 + w_3 x^3 \qquad (۲۲-۵)$$

$$x_1 = x \qquad x_2 = x^2 \qquad x_3 = x^3 \qquad (۲۳-۵)$$

$$y = w_0 + w_1x_1 + w_2x_2 + w_3x_3 \qquad (۲٤-۵)$$

اما برخی از مدلهای غیرخطی با استفاده از تغییر متغیر، به راحتی قابل تبدیل به شکل خطی نیستند. برای چنین مواردی نیز ممکن است تخمینهای حداقل مربعات را بر اساس محاسبات پیچیده‌ای به‌دست آوریم. در این‌گونه موارد روشهای آماری‌ای موجود است که بر اساس میزان دقت برازش مدل برای پیش‌بینی، مقادیر y را تعیین می‌کند. قبل از به‌کار بردن تحلیل رگرسیونی، معمولاً بهتر است زیرمجموعه‌ای از ویژگیهایی را که به نظر می‌رسد پیش‌بینی کننده خوبی برای Y نیستند، حذف کرده و از بقیه ویژگیها استفاده کنیم (این مطلب با تفصیل بیشتر در بحث آماده‌سازی داده‌ها عنوان شده است.) تحلیل رگرسیونی یک روش دقیق و مناسب برای پیش‌بینی است، به‌جز در مواردی که داده‌ها شامل داده‌های پرت و مغشوش باشند. این داده‌ها با داده‌های دیگر بسیار ناسازگار هستند. در این حالت بهتر است چنین داده‌هایی در نظر گرفته نشوند.

سایر روشهای مبتنی بر رگرسیون

آیا رگرسیون خطی می‌تواند داده‌های طبقه‌ای را پیش‌بینی کند؟ مدلهای خطی تعمیم یافته، مبنای استفاده از رگرسیون خطی برای پیش‌بینی متغیرهای طبقه‌ای را فراهم می‌کنند. در این مدلها، واریانس متغیر پاسخ (Y) تابعی از مقدار میانگین Y است، در حالی‌که در رگرسیون خطی، واریانس Y ثابت بود. انواع متعارف مدلهای خطی تعمیم یافته، شامل رگرسیون لجستیک و رگرسیون پواسون هستند. مدلهای رگرسیون لجستیک، احتمال وقوع پدیده‌هایی با تابع خطی از یک مجموعه متغیرهای مستقل می‌باشد، داده‌های شمارشی نیز از توزیع پواسون پیروی کرده و به‌طور معمول با رگرسیون پواسون مدلسازی می‌شوند. با توجه به اهمیت رگرسیون لجستیک در پیش‌بینی متغیرهای طبقه‌ای، در بخش بعد، این روش مورد بررسی قرار می‌گیرد. [۱]

رگرسیون لجستیک

در بسیاری از موارد، متغیر وابسته (پاسخ) تنها دو مقدار ۰ و ۱ را می‌پذیرد. استفاده از رگرسیون معمولی، برای این نوع متغیرهای وابسته ممکن است منجر به تعیین مقادیر کمتر از صفر یا بیشتر از ۱ شود که اصولا چنین مقادیری نمی‌تواند اتفاق بیفتد. یک روش جایگزین، استفاده از رگرسیون لجستیک است که ما را قادر می سازد از مدل رگرسیون، برای پیش‌بینی احتمال وقوع مقدار یک متغیر مجازی به‌عنوان تابعی از مجموعه متغیرهای مستقل استفاده کنیم. این مدل مبتنی بر نسبت برد به باخت یا همان توفیر یا شانس[1] است.

$$\text{توفیر } odds = \frac{\text{(برد) احتمال موفقیت}}{\text{(باخت) احتمال شکست}} = \frac{p}{1-p} \quad , P = \frac{odds}{1 + odds}$$

مثلاً اگر احتمال موفقیت یک پیشامد ۰/۷۵ باشد توفیر آن برابراست با ۳ است.

(احتمال موفقیت −۱ = احتمال شکست) و داریم:

$$odds = \frac{0/75}{1-0/75} = 3$$

این مدل مبتنی بر لگاریتمِ طبیعیِ نسبت توفیرها می‌باشد و با استفاده از روش برآورد درست نمایی حداکثر مدل رگرسیون، جهت پیش‌بینی (توفیر) Ln از رابطه (۵–۲۵) محاسبه می‌شود.

$$Ln \text{ (توفیر برآورد شده)} = b_0 + b_1 x_1 + \ldots + b_k x_k \qquad (۵–۲۵)$$

پس از برازش دادن رگرسیون لجستیک به یک مجموعه از داده‌ها، برآورد توفیر رابطه (۵–۲۵) به‌دست می آید.

$$\text{برآورد توفیر} = exp\ (Ln \text{ (توفیر برآورد شده)}) \qquad (۵–۲۶)$$

و با محاسبه توفیر برآورد شده، احتمال موفقیت از رابطه (۵–۲۷) محاسبه می‌شود.

---

[1] – نسبتی است که شانس بردن به شانس باختن را نشان می‌دهد یعنی برتری، توفیر و مزیت را می‌رساند.

$$\text{احتمال موفقیت} = \frac{\text{توفیر}}{\text{۱ + توفیر}} = \frac{odds}{\text{۱+ } odds} \qquad (۲۷-۵)$$

مثال: بخش بازاریابی یک شرکت کارت اعتباری، می‌خواهد مانند سالهای گذشته، مشتریان کارتهای معمولی خود را متقاعد به خرید کارتهای ویژه نماید. مهم‌ترین تصمیمی که شرکت باید بگیرد، این است که با کدام‌یک از مشتریان کارتهای اعتباری خود تماس بگیرد. از نمونه ۳۰ تایی مشتریانی که در بازاریابی سال گذشته، با آنها تماس حاصل شده است، اطلاعات زیر موجود است. اینکه آیا مشتریانی که کارت معمولی داشته اند، آیا مبادرت به خرید کارت ویژه نیز کرده اند یا خیر؟ (y = ۰/۱)

کل مبلغ سالیانه خرید (بر حسب هزار دلار) با کارت معمولی شرکت (x_1) و اینکه دارنده کارت اعتباری برای دیگر اعضای خانواده خود کارت اعتباری داشته است، یا خیر (x_2) در این صورت مدل رگرسیون لجستیک به‌صورت رابطه (۵-۲۸) خواهد بود.

$$Y = Ln \text{ (برآورد توفیر خرید کارت ویژه)} = b_0 + b_1x_1 + b_2x_2 \qquad (۲۸-۵)$$

بر اساس داده‌های نمونه‌ای مقادیر b_2، b_1، b_0 به‌دست آمده و سپس به پیش‌بینی متغیر پاسخ می‌پردازیم. فرض کنید، یک دارنده کارت اعتباری معمولی، سال گذشته ۳۶۰۰۰ دلار خرید کرده است، درصورتی‌که بدانیم برای اعضای دیگرخانواده خود نیز کارت داشته باشد، احتمال اینکه کارت ویژه شرکت را بخرد چقدر است؟

$x_1 = ۳۶$, $x_2 = ۱$

از روی معادله رگرسیون لجستیک مقدار (برآورد نسبت توفیرهای خرید کارت ویژه) lnبه‌دست می‌آید و از روی آن توفیر برآوردشده محاسبه شده و درنتیجه احتمال خرید کارت اعتباری ویژه به دست می‌آید. جدول (۵-۱۶) داده‌های مربوط به نمونه ۳۰تایی ازمشتریان سال گذشته شرکت را نشان می‌دهد.

جدول ۵- ۱۶) داده‌های مشتریان

کارت اضافی	خرید	Y	نمونه
۰	۱۲۰۷ .۳۲	۰	۱
۱	۳۷۰۶ .۳٤	۱	۲
۰	۸۷٤۹ .٤	۰	۳
۰	۱۲٦۳ .۸	۰	٤
۰	۹۷۸۳ .۱۲	۰	٥
۰	۰٤۷۱ .۱٦	۰	٦
۰	٦٦٤۸ .۲۰	۰	۷
۱	۰٤۸۳ .٤۲	۱	۸
۱	۲۲٦٤ .٤۲	۰	۹
۱	۹۹۳ .۳۷	۱	۱۰
۱	٦۰٦۳ .٥۳	۱	۱۱
۰	۷۹۳۸ .۳۸	۰	۱۲
۰	۹۹۹۹ .۲۷	۰	۱۳
۰	۱٦۹٤ .٤۲	۱	۱٤
۱	۱۹۹۷ .٥٦	۱	۱٥

کارت اضافی	خرید	Y	نمونه
۰	۷٦۰۹ .۲۳	۰	۱٦
۱	۰۳۸۸ .۳٥	۰	۱۷
۱	۷۳۸۸ .٤۹	۱	۱۸
۰	۷۳۷۲ .۲٤	۰	۱۹
۱	۱۳۱٥ .۲٦	۱	۲۰
۱	۳۲۲۰ .۳۱	۰	۲۱
۱	۱۹٦۷ .٤۰	۱	۲۲
۰	۳۸۹۹ .۳٥	۰	۲۳
۰	۲۲۸۰ .۳۰	۰	۲٤
۰	۳۷۷۸ .٥۰	۱	۲٥
۰	۷۷۱۳ .٥۲	۰	۲٦
۰	۳۷۲۸ .۲۷	۰	۲۷
۱	۲۱٤٦ .٥۹	۱	۲۸
۱	۰٦۸٦ .٥۰	۱	۲۹
۱	٤۲۳٤ .۳٥	۱	۳۰

بااستفاده از نرم‌افزار *MINITAB*، مقادیر پارامترهای مدل به شرح زیر به‌دست آمده و در نهایت احتمال خرید کارت اعتباری ویژه برای مشتری جدید با ویژگیهای ذکرشده، محاسبه می‌شود.

$b_0 = -٦/٩٤ \qquad b_1 = ۰/۱۳۹٤۷ \qquad b_2 = ۲/۷۷٤$

Ln (برآورد توفیر خرید کارت ویژه) $= -٦/٩٤ + ۰/۱۳۹٤۷(۳٦) + ۲/۷۷٤(۱) = ۰/۸٥٤۹۲$

توفیر برآورد شده $= Exp(۰/۸٥٤۸۲) = ۲/۳٥۱۲$

احتمال خرید کارت اعتباری ویژه $= (۲/۳٥۱۲/(۱+۲/۳٥۱۲)) = ۰/۷۰۱٦$

روشهای ارزیابی دسته‌بندی

همان‌طورکه بیان شد روشهای مختلفی برای دسته‌بندی استفاده می‌شوند و این روشها در شرایط مختلف، رفتارهای متفاوتی از خود نشان می‌دهند. شاخصهای زیر این روشها را با یکدیگر مقایسه می‌کنند.

صحت مدل[1]: دقت یک روش دسته‌بندی بستگی به تعداد پیش‌بینی‌های درستی است که آن مدل انجام داده است.

سرعت[2]: زمان لازم برای ساخت مدل واستفادهٔ مدل جهت دسته‌بندی عامل مهم دیگری است.

پایداری[3]: چنین شاخصی توانایی برخورد مدل در مواجهه با داده‌های غیرمعمول و یا مقادیر جا افتاده را نشان می‌دهد.

تفسیر پذیری[4]: این شاخص نشان‌دهندهٔ میزان قابل فهم بودن آن توسط دیگران است و اینکه این مدل دسته‌بندی بتواند دیدگاهی روشن نسبت به نحوهٔ دسته‌بندی و نوع دسته‌ها ارائه دهد.

جمع و جور بودن مدل[5]: اندازهٔ مدل در ایجاد انگیزه جهت استفادهٔ آن بسیار مهم است اندازه مدل همان اندازه درخت و یا تعداد قواعد ایجاد شده توسط آن مدل می‌باشد.

پیچیدگی در مدلسازی

مدلهای با پیچیدگی بالا می‌توانند مدلسازی را با دقت بالایی انجام دهند، از این‌رو انحراف پایینی دارند، ولی موجب به‌وجود آمدن پدیده‌ای به‌نام بیش‌برازش می‌شوند. در این مدلها بیش‌برازش باعث سوگیری بالا خواهد شد. مدلهای با پیچیدگی کمتر نمی‌توانند مدلسازی را خیلی دقیق انجام دهند، اما پایدارتر هستند، در این مدلها

---

[1]- Accuracy
[2]- Speed
[3]- Robustness
[4]- Interpretability
[5]- Compactness

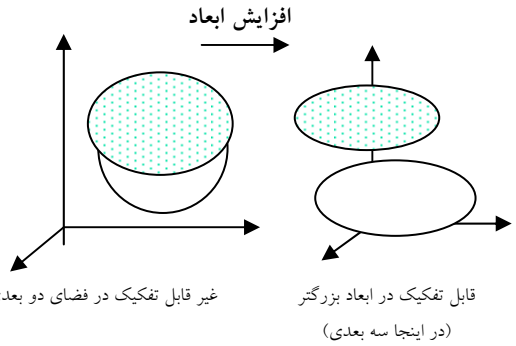
پراکندگی کم شده اما در مقابل واریانس[1] بیشتر می‌شود. مسئله‌ای که در اینجا با آن روبرو هستیم این است که به سطح بهینه‌ای از پیچیدگی یا ابعاد برسیم تا تعادل مطلوبی میان انحراف و پراکندگی به‌دست آید.

نمایشی از تعادل بین انحراف و سوگیری

در بسیاری موارد ترجیح داده می‌شود که ابعاد بالایی انتخاب شوند تا دسته‌بندی به نحو مطلوب و ساده‌تری انجام پذیرد. اما بالا رفتن ابعاد سبب مشکلات خاص خود در زمینه بیش‌برازش خواهد شد. مسئله بالا رفتن ابعاد باعث تنگ شدن فضای ویژگیها خواهد شد، به‌نحوی که حجم محاسبات بسیار بالا خواهد رفت و لذا باید محاسباتی انجام شود، که اغلب غیرضروری هستند. پس آنچه که مهم است به‌دست آوردن سطح مطلوبی از ابعاد یا پیچیدگی در مسئله است.

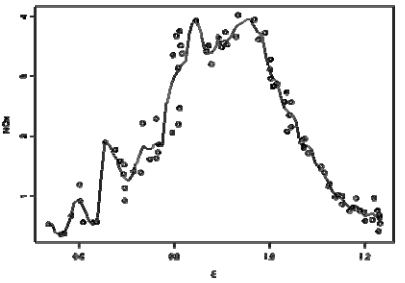
تعادل سوگیری و انحراف

به‌طورکلی این‌گونه روشها برای اجتناب از بیش برازش مطرح شده‌اند و در کل می‌توان گفت که بعضی از این روشها خوب جواب می‌دهند اما هیچ تضمین آماری وجود ندارد که آیا این روشها برای موارد و وضعیتهای مختلف جواب خوبی خواهند داشت یا خیر.

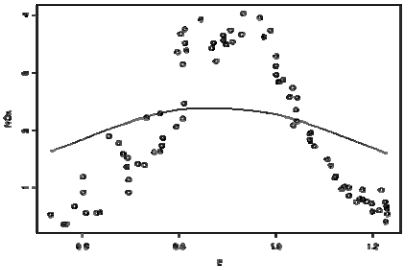


شکل ۵-۲۲) مثالی از افزایش ابعاد

[1]- Variance



شکل ۵-۲۳-الف) انحراف کم -پراکندگی بالا



شکل ۵-۲۳-ب) انحراف بالا -پراکندگی کم

اجتناب از بیش برازش در دسته‌بندی

در روشهای دسته‌بندی ممکن است مسئله بیش‌برازش اتفاق بیفتد. مثلاً یک درخت تصمیم باعث بیش‌برازش داده‌های آموزش مدل شود. در این حالت دقت روی داده‌های آموزش مدل بالاست اما دقت در مورد داده‌های بعدی آزمون پایین می‌آید. در این مورد به‌علت اینکه شاخه‌های بسیاری در درخت وجود دارد ممکن است درخت حتی داده‌های مغشوش را هم تحلیل کرده باشد که به این ترتیب کاری غیر ضروری انجام شده و مشکلات بعدی به همراه خواهد داشت.

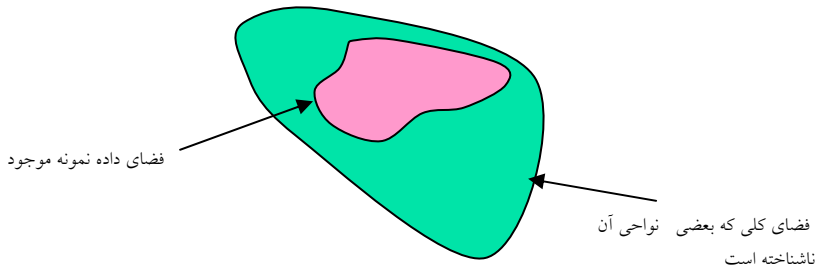
دو روش برای اجتناب از بیش برازش وجود دارد:

- هرس اولیه[1]: توقف ساخت درخت در مراحل اولیه.
- هرس ثانویه[2]: حذف بعضی شاخه‌ها از درخت ساخته شده (که به‌صورت معمول این روش استفاده می‌شود).

[1]- Prepruning
[2]- Postpruning

مسئله تعمیم[1]

در مسائل دسته‌بندی از مجموعهٔ محدودی از نمونه‌ها برای به‌دست آوردن مدل دسته‌بندی استفاده می‌شود. اگر داده‌های آزمون شبیه داده‌هایی باشند که مدل با آنها به‌دست آمده است، مشکلی پیش نمی‌آید ولی در عالم واقع با داده‌های آموزش مدل نمی‌توان همه سناریوهای ممکن را مشخص نمود. این همان مسئله‌ای است که از آن به‌عنوان مسئله تعمیم یاد می‌شود. تعمیم مشخص می‌نماید که تا چه میزان سیستم نسبت به ورودی‌های ناشناس، که با مقادیر داده‌های آموزش مدل متفاوتند، پایدار است. مدل ساخته شده در روش دسته‌بندی برای داده‌های استفاده شده در ساخت مدل و یا داده‌های شبیه به آنها درست جواب می‌دهد، اما همهٔ داده‌ها شبیه به داده‌های استفاده شده نیستند و حتی در برخی موارد فضای ناشناخته‌ای وجود دارد که در مورد داده‌های آن فضا هیچ‌گونه اطلاعاتی در دسترس نیست. در چنین مواردی گریزی از ساخت مدل بر اساس اطلاعات قبلی نیست ولی باید سعی شود تا خطا و یا ریسک مدل را کم کرد.



شکل ۵-۲٤) نمایی از ریسک در دسته‌بندی

[1]- Generalization Problem

اندازه‌گیری خطا و میزان دقت در اندازه‌گیریها

فرض کنید با استفاده از داده‌های گذشته، یک روش دسته‌بندی یا پیش‌بینی را آموزش داده و می‌خواهیم رفتار آینده متغیر مورد مطالعه را با این روشها بررسی کنیم. یک سؤال اساسی که در این این زمینه پیش می‌آید، این است که دقت روش دسته‌بندی یا پیش‌بینی مورد استفاده چه اندازه است و یا اینکه چگونه می‌توان دقت دو یا چند روش دسته‌بندی یا پیش‌بینی را با هم مقایسه کرد؟ در بخشهای بعد، چگونگی محاسبه دقت روشهای دسته‌بندی یا میزان خطای روشهای پیش‌بینی و همچنین روشهای مورد استفاده در تعیین دقت و افزایش دقت و چگونگی انتخاب مدل دسته‌بندی یا پیش‌بینی مناسب، به اختصار بیان می‌شود.[۱]

ارزیابی دقت روشهای دسته‌بندی

میزان دقت یک روش دسته‌بندی بر روی مجموعه داده‌های آموزشی، در صد مشاهداتی از مجموعه آموزشی است که به درستی توسط روش مورد استفاده، دسته‌بندی شده‌اند. در ادبیات تشخیص الگو، به این شاخص خاص «نرخ تشخیص» گفته می‌شود و آن به این معنی است که روش دسته‌بندی با چه کیفیتی نمونه‌های مربوط به کلاسهای متفاوت را تشخیص می‌دهد. برای محاسبهٔ این شاخص داده‌های آزمون استفاده می‌شود. میزان این شاخص به‌عنوان خطای «جایگزینی»[1] نام برده می‌شود ولی با دید خوشبینانه می‌توان از این شاخص به‌عنوان شاخص کل خطا نیز استفاده کرد. در اینجا می‌توان نرخ خطا یا دسته‌بندی نادرست را بر اساس شاخص دقت محاسبه کرد. اگر میزان دقت یک روش دسته‌بندی را با $ACC(m)$ نشان دهیم، میزان خطای آن برابر با $ACC(m)$ – ۱خواهد بود.

[1]- Resubstitution

ماتریس دسته‌بندی[1] یک ابزار مفید برای تحلیل چگونگی عملکرد روش دسته‌بندی در تشخیص داده‌ها یا مشاهدات دسته‌های مختلف است. اگر داده‌ها در m دسته قرارگرفته باشند، یک ماتریس دسته‌بندی، جدولی با حداقل اندازه $m*m$ است. عنصر $C_{i,j}$ در iامین سطر و jامین ستون، نشان دهنده تعداد مشاهداتی از دسته i است که توسط روش دسته‌بندی به‌عنوان کلاس j تشخیص داده شده است. برای اینکه یک روش دسته‌بندی، دقت بالایی داشته باشد، حالت ایده‌آل آن است که اکثر داده‌های مرتبط به مشاهدات بر روی قطر اصلی ماتریس قرارگرفته باشند و بقیه مقادیر ماتریس صفر یا نزدیک صفر باشند. ماتریس ممکن است سطر یا ستون اضافی داشته باشد که نشان دهنده مجموع عناصر یا در صد دقت می‌باشد.

در مثال زیر مشتریان به دو دسته تقسیم شده‌اند: مشتریانی که کامپیوتر می‌خرند و آنهایی که نمی‌خرند، در اینجا از ماتریس دسته‌بندی استفاده شده است. در این مثال دو دسته وجود دارد، بنابراین ماتریس ۲ × ۲ تعریف می‌شود. البته ردیف‌ها و ستون‌های دیگری نیز برای محاسبات درصدها به این ماتریس اضافه می‌شوند. عنصر (۱,۲) این ماتریس تعداد عناصری که برچسب کلاس آنها *"Yes"* بوده ولی به نادرست در کلاس *"No"*ها دسته‌بندی شده‌اند را نشان می‌دهد و همین‌طور عنصر (۲,۱) نیز تعداد عناصری که برچسب کلاس آنها *"No"* است ولی در دستهٔ *"Yes"*ها دسته‌بندی شده است را نشان می‌دهد.

در این مثال از مفاهیمی استفاده شده است که در اینجا به آنها می‌پردازیم. عنصر «مثبت درست»[2] به مشاهداتی از دسته c_1 دلالت دارد که توسط روش دسته‌بندی به درستی تشخیص داده شده است. عنصر «منفی درست»[3] به مشاهداتی از دسته c_2 دلالت دارد که توسط روش دسته‌بندی به درستی تشخیص داده شده است. به‌طور مشابه «منفی

---

[1]- Confusion Matrix
[2]- True Positive
[3]- True Negative

غلط»[1] مشاهداتی از دسته c_1 است که توسط روش دسته‌بندی به غلط در دسته c_2 قرارگرفته و «مثبت غلط»[2] مشاهداتی از دسته c_2 است که به غلط در دسته c_1 قرارگرفته‌اند. بر مبنای تعریف عناصر ماتریس، شاخصهای زیر برای سنجش حساسیت، تشخیص و تمایز روش دسته‌بندی تعریف می‌شوند.

جدول ۵-۱۷) داده‌های مشتریان در ماتریس دسته‌بندی

دسته‌ها	*Yes*	*No*	*Total*	درصد شناخت
Yes (دستهٔ حقیقی)	٦٩٥٤	٤٦	٧٠٠٠	٩٩/٣٤
No (دستهٔ حقیقی)	٤١٢	٢٥٨٨	٣٠٠٠	٨٦/٢٧
Total	٧/٣٦٦	٢/٦٣٤	١٠٠٠٠	٩٥/٥٢

جدول ۵-۱۸) ماتریس دسته‌بندی

	C_1	C_2
C_1	*True-positive*	*False-negative*
C_2	*False-positive*	*True-negetive*

	C_1	C_2
C_1	درست دسته‌بندی شده‌اند	غلط دسته‌بندی شده‌اند
C_2	غلط دسته‌بندی شده‌اند	درست دسته‌بندی شده‌اند

مدلهای مختلف با درجه دقتهای مختلفی قابل پذیرش هستند. به‌عنوان مثال در یک مدل تشخیص سرطان، مدلی با ۹۰٪ دقت قابل‌قبول نیست. بدین منظور شاخصهای دیگری نیز نیاز است که در اینجا به آنها اشاره می‌شود.

[1]- False Negative
[2]- True Negative

$$\text{حساسیت}^{1} = \frac{\text{تعداد داده‌هایی که درست دسته‌بندی شده‌اند و جواب نیز مثبت است}}{\text{کل تعداد داده‌های مثبت}} = \frac{t\text{-}pos}{pos} \qquad (۲۹-۵)$$

$$\text{شفافیت}^{2} = \frac{\text{تعداد داده‌هایی که درست دسته‌بندی شده‌اند و جواب منفی است}}{\text{کل تعداد داده‌های منفی}} = \frac{t\text{-}neg}{neg} \qquad (۳۰-۵)$$

در این فرمول $f\text{-}pos$ تعداد داده‌هایی است که جزء دسته Noها هستند ولی به نادرست در دستهٔ Yes ها واقع شده‌اند، می‌باشد.

$$\text{دقت}^{3} = \frac{t-pos}{t-pos+f-pos} \qquad (۳۱-۵)$$

شاخص آخر یا همان دقت ترکیبی از دو شاخص قبل است و به‌صورت زیر محاسبه می‌شود:

$$\text{صحت} = \text{حساسیت} \frac{pos}{(pos+neg)} + \text{شفافیت} \frac{neg}{pos+neg} \qquad (۳۲-۵)$$

توجه به این نکته ضروری است که در روابط فوق، وزن یا اهمیت عناصر ماتریس یکسان در نظر گرفته شده است. درحالی‌که در مسائل و زمینه‌های علوم مختلف، اهمیت این عناصر می‌تواند متفاوت باشد. همچنین گاهی اوقات که حجم داده‌ها به اندازه کافی زیاد نبوده و یا یک مشاهده به بیش از یک دسته تعلق داشته باشد، استفاده از شاخصهای دقت فوق، چندان مناسب به‌نظر نمی رسد.

پس از پیش‌بینی با مدل رگرسیون لجستیک، باید خوب بودن انطباق مدل و اینکه آیا هر کدام از متغیرهای مورد استفاده در مدل سهم قابل توجهی در مدل داشته‌اند یا خیر را

---

[1]- Sensitivity
[2]- Specificity
[3]- Precision

بررسی کنیم. برای این‌کار از آماره انحراف[1] استفاده می‌شود. این آماره مبتنی بر توزیع کای‌دو بوده و با مقایسه مقدار آن با ناحیه بحرانی متناظر، در مورد انطباق مدل اظهارنظر می‌شود و سپس با استفاده ازشاخص والد، که از توزیع نرمال پیروی می‌کند، بررسی می‌کنیم که آیا هر کدام از متغیرها در حضور متغیر دیگر، سهم قابل توجهی را در مدل دارا هستند؟ این مطالب درمراجع اقتصادسنجی و آمار با تفصیل بیشتر عنوان شده است.

میزان خطای پیش‌بینی کننده‌ها

بحث بعدی آن است که چگونه می‌توان دقت روشهای پیش‌بینی را اندازه‌گیری کرد. برای ارزیابی دقت روشهای پیش‌بینی (اختلاف بین مقدار واقعی و مقدار پیش‌بینی متغیر وابسته) از مفهوم تابع زیان استفاده کرده و دو شاخص زیر را برای سنجش خطای پیش‌بینی مورد استفاده قرار می‌دهیم.

$$\textit{Mean absolute error}: \quad \frac{\sum_{i=1}^{d} | y_i - y_i' |}{d} \qquad (۳۳-۵)$$

$$\textit{Mean squared error}: \quad \frac{\sum_{i=1}^{d} (y_i - y_i')^2}{d}$$

واضح است که برای بالابردن دقت یک روش پیش‌بینی، لازم است که مقادیردوشاخص یادشده تاحدمقدور کوچک باشند. همچنین از دو شاخص زیر نیز برای محاسبه خطای نسبی پیش‌بینی تک تک مقادیر نمونه، در مقایسه با خطای پیش‌بینی مقادیر، نسبت به میانگین استفاده می‌شود.

[1]- Deviance Statistic

$$\textit{Relative absolute error}: \quad \frac{\sum_{i=1}^{d} |y_i - y_i'|}{\sum_{i=1}^{d} |y_i - \bar{y}|} \qquad (۳٤-۵)$$

$$\textit{Relative squared error}: \quad \frac{\sum_{i=1}^{d} (y_i - y_i')^2}{\sum_{i=1}^{d} |(y_i - \bar{y})^2}$$

منابع

1) *Jiawei Han, Micheline Kamber, Data Mining Concepts & Techniques, Elsevier Inc. 2006.*
2) *Martin, B. (1995): Instance-Based Learning: Nearest Neighbour with Generalisation. Univercity of Waikato.*
3) *Pyle, D. (2003): Business Modeling and Data Mining. Morgan Kaufmann.*
4) *Hand, D. J. , Mannila, H. and Smyth, P. (2001): Principles of Data Mining. Bradford Book.*
5) *Wilson, D. R. and Martinez, T. R. (2000): Reduction Techniques for Instance-Based Learning Algorithms. In: Machine Learning Vol. 38 (3) pp. 257–286.*
6) *Larose, D. T. (2005): Discovering knowledge in data: an introduction to data mining. Wiley-Interscience. Daniel Larose, Data Mining Methods and Models, Wiley-Interscience, Hoboken, NJ, 2005.*
7) *Witten, I. H. and Frank, E. (2000): Data mining: practical machine learning tools and techniques with Java implementations. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.*
8) *Wang, J. (2005): Encyclopedia Of Data Warehousing And Mining. Idea Group Publishing.*
9) *Berry, M. J. A. and Linoff, G. (1997): Data Mining Techniques: For Marketing, Sales, and Customer Support. Wiley.*

بخش سوم

فصل ششم: انباره داده‌ها

فصل هفتم: متدلوژی اجرا و پیاده‌سازی پروژه‌های داده‌کاوی

فصل ششم

انباره داده‌ها

سازمانها درک کرده‌اند که سیستم‌های انباره‌داده ابزارهای ارزشمندی در رقابتهای امروزه هستند. بنگاههای بسیاری، میلیونها دلار برای ساخت انباره‌داده صرف کرده‌اند. طبق تعریف اینمون[1] یکی از پیشتازان معماری در ساخت سیستمهای انباره‌داده، انباره‌داده مجموعه‌ای موضوع‌گرا، یکپارچه، از زمانهای مختلف و غیرفرار به منظور پشتیبانی از فرآیند تصمیم‌سازی است. داده‌کاوی چیزی فراتر از پردازش بر روی یک پایگاه داده معمولی می‌باشد. مثالهای زیر این تفاوت را نشان می‌دهند. یک پرس و جوی ساده و پیدا کردن تمامی افراد با نام «علی» در یک پایگاه داده بسیار ساده است ولی در مقابل پیدا کردن افرادی که کارت اعتباری آنها وضعیت مناسبی ندارد و در مرز ورشکستگی می‌باشند، خیلی ساده نیست. پیدا کردن افرادی که بیش از ۱۰۰/۰۰۰ تومان خرید داشته‌اند ساده است ولی در مقابل پیدا کردن افرادی که عادتهای خرید مشابهی دارند و یکسری اقلام خاصی را با هم خرید می‌کنند، کار ساده‌ای نیست. پیدا کردن افرادی که

[1]- W. H Inmon

در یک تاریخ خاص از یک فروشگاه خاص شیر خریده‌اند، بسیار ساده است ولی در مقابل پیدا کردن افرادی که غالباً شیر خریداری می‌کنند خیلی ساده نیست.
با توجه به مثالهای مطرح شده، کاملاً مشخص است که داده‌هایی که در داده‌کاوی و الگوریتم‌های آن استفاده می‌شوند، تفاوت عمده‌ای با داده‌های عادی در پایگاه داده‌ها دارند. جهت فراهم کردن این نوع داده‌ها که در انباره‌داده‌ها قرار می‌گیرند، باید یکسری پردازش‌های خاص روی آنها صورت پذیرد. این نوع پردازش‌ها به نام‌های پاکسازی داده‌ها و یکپارچه‌سازی داده‌ها معروفند، که در فصل دوم کاملاً توضیح داده شده‌اند.

داده‌کاوی و انباره‌داده‌ها

در دهه ۹۰ میلادی پدیده انباره‌داده‌ها ظهور یافت. قبل از انبارسازی داده‌ها سیستمهای کامپیوتری جهت ذخیره، جمع‌آوری، تغییر و تصحیح داده‌ها طراحی شده بودند. این سیستمهای اولیه به سیستمهای عملیاتی یا میراثی[1] موسوم هستند. گرچه جمع‌آوری و ذخیره داده‌ها کارهای مفیدی به حساب می‌آیند، اما دسترسی به آنها و تحلیل داده‌های عملیاتی به راحتی امکان پذیر نیست و علت نیز عدم یکپارچگی داده‌ها می‌باشد. هر برنامه کاربردی داده‌ها را بنا بر نیاز خود تفسیر می‌کند. مدیران برای تصمیم‌گیری نیازی به کوهی از اطلاعات جزئی روزانه ندارند، آنها نیازمند چکیده اطلاعات برای دوره‌های زمانی متفاوت می‌باشند و به همین علت است که داده‌های تاریخی دارای مفهوم و ارزش بیشتری می‌باشد. انباره‌داده‌ها برای شرکتهایی که مصمم به استفاده از داده‌کاوی هستند یک ضرورت می‌باشد، یکی از ماهیتهای وجودی انباره‌داده‌ها، یکپارچگی داده‌ها هنگام قرارگیری در انباره‌داده‌هاست. این بدین معنی است که دقت بسیاری به‌کار گرفته می‌شود تا یکنواختی و پیوستگی در درک اهداف عام سازمانی بوجود آید. اگر انباره‌داده‌ها وجود نداشته باشد، داده‌کاو باید زمان بسیار زیادی را صرف جمع‌آوری،

[1]- Legacy

پاکسازی و یکپارچه‌سازی داده‌ها کند. بدین ترتیب وقت بسیاری باید صرف شود تا تحلیل داده‌ها آغاز شود.

در انباره‌داده‌ها، داده‌های تاریخی جمع‌آوری و سازماندهی می‌شوند. وجود داده‌های تاریخی برای یافتن الگوها و روابطی که سازمان به دنبال آنهاست، برای داده‌کاوی یک ضرورت است. اگر چنانچه این داده‌های تاریخی وجود نداشته باشند، داده‌کاو باید به دنبال جمع‌آوری آنها باشد.

علت دیگر اهمیت انباره‌داده‌ها این است که انباره‌داده‌ها شامل داده‌های جزئی و داده‌های کلی، در کنار یکدیگر می‌باشد. بدون تردید، داده‌کاو به اطلاعات جزئی برای تحلیل نیازمند است، اما داده‌های خلاصه شده نیز به‌کار می‌آیند. از آنجا که در انباره‌داده انواع داده‌های خلاصه شده وجود دارد، داده‌کاو می‌تواند به سرعت داده‌های انباره‌داده را بررسی کند و این باعث کاهش تکرار تحلیلها توسط داده‌کاو می‌شود.

انباره‌داده‌ها

ویژگیهای مهم یک انباره‌داده عبارتند از:

- موضوع محوری[۱]
- جامعیت[۲]
- پویاپذیری[۳] (مهم بودن عامل زمان)
- پایایی[٤] (غیرفرار و دایمی بودن)

موضوع محوری: داده‌ها طبق یک موضوع خاص سازماندهی می‌شوند، به‌عنوان مثال داده‌های مربوط به مشتریان، محصولات و یا داده‌های مرتبط با فروش، هر کدام جداگانه در نظر گرفته می‌شوند. اما در پایگاه داده‌های معمولی، داده‌ها بر اساس

---

[1]- Subject Oriented
[2]- Integrated
[3]- Time Variant
[4]- NON Volatile

عملیات و پردازشهای روزانه ایجاد می‌شوند و موضوع آنها مرتبط با کل پردازش می‌باشد.

جامعیت: داده‌های انباره، از تجمع دیگر داده‌ها ساخته می‌شوند. این داده‌ها ممکن است مربوط به پایگاه داده‌های رابطه‌ای، فایلهای بدون ساختار و یا رکوردهای مرتبط با پردازش‌های برخط باشند. جهت یکسان‌کردن داده‌ها، روشهای پاکسازی به‌کار برده می‌شود. نوعی هماهنگی کلی در مورد داده‌های مختلفی که از سیستمهای متفاوت آمده‌اند، لازم است. به‌عنوان نمونه لازم است مقادیر عددی مربوط به «قیمت هتل»، «هزینه صبحانه»، «مالیات» و دیگر موارد مشابه که ممکن است از مکانهای متفاوت آمده باشند، یکسان شده و یک انباره‌داده با نام «مسافر» ایجاد شود.

پویاپذیری: افق زمانی برای انباره‌داده‌ها بسیار مهم‌تر ازداده‌های مرتبط با سیستمهای عملیاتی می‌باشد. در ساختار انباره‌داده‌ها عاملی به نام زمان در نظر گرفته می‌شود. این عامل می‌تواند به‌طور ضمنی و یا به وضوح بیان شود. اما در سیستمهای عملیاتی عامل زمان، عاملی کلیدی نیست.

پایائی: پایگاه داده‌ها شامل داده‌هایی است که روزانه با آنها کار می‌شود و بخشهایی به آن اضافه و یا از آن حذف می‌شود، در مقابل انباره‌داده این ویژگی را ندارد. با توجه به همین امر واضح است که به‌روز شدن داده‌ها در انباره‌داده‌ها مقدور نمی‌باشد انباره‌داده‌ها نیازی به پردازشهایی از قبیل: تراکنشهای داده‌ای، بازیافت و مکانیزم‌های کنترل هم‌زمان ندارد. تنها اعمالی که در انباره‌داده‌ها صورت می‌پذیرد عبارتند از: بارگذاری (مقدار دهی) اولیه داده‌ها و دسترسی به داده‌ها.

پردازشی که بر روی انباره‌داده‌ها انجام می‌گیرد *OLAP* نامیده می‌شود (*OLAP* در مقابل *OLTP* آمده است که پردازش‌هایی است که بر روی داده‌های پایگاه داده‌ها انجام

می‌گیرد). جدول (٦-١) *OLAP*[1] و *OLTP*[2] را با توجه به برخی پارامترهای مهم مقایسه کرده است، این پارامترها عبارتند از:
«کاربران»، «کارکرد»، «طراحی پایگاه داده»، «داده»، «کاربرد»، «نحوه دسترسی»، «واحد کاری»، «تعداد رکوردهای در دسترس»، «تعداد کاربران»، «اندازه پایگاه داده» و «شاخص».

جدول ٦-١) مقایسه پردازشها در انباره‌داده‌ها و پایگاه داده‌ها

ویژگیها	*OLTP*	*OLAP*
کاربران	کارشناسان خبره	اپراتورها
کارکرد	تصمیم گیری	کارهای روزمره
طراحی پایگاه داده	بر مبنای موضوع	بر مبنای کاربرد
داده	تاریخی، چندبعدی مجتمع	جاری، روزبه‌روز و با جزییات
کاربرد	در موارد خاص	تکراری
نحوه دسترسی	کاوش و کشف	خواندن ونوشتن
واحد کاری	جستجوهای پیچیده	پردازش ساده
تعداد رکوردها	میلیونها	دهها
تعداد کاربران	صدها	هزاران
اندازه پایگاه داده	۱۰۰ *TB_GB*	۱۰۰ *MB_GB*
شاخص	جستجو و پاسخ	پردازش

ساختار انباره‌داده

انباره‌داده‌ها بر مبنای ساختاری چند بعدی، که مدل داده‌ها را به شکل مکعب اطلاعاتی نشان می‌دهد، ساخته شده است. یک مکعب اطلاعاتی داده‌ها را در چندین بعد مختلف

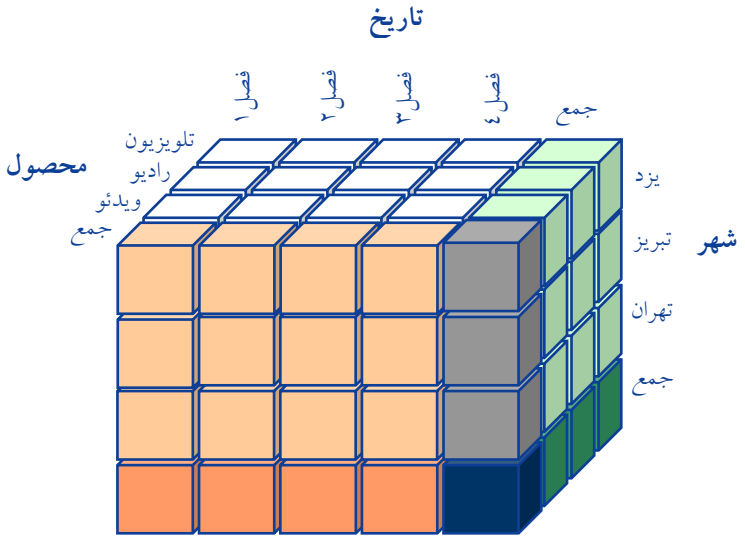
[1]- Online Analytical Process
[2]- Online Transaction Process

مدلسازی می‌کند. در زیر مثالی از یک مکعب اطلاعاتی به نام «فروش» بررسی شده است که شامل این ابعاد می‌باشد:

Item (نوع، نام تجاری، نام کالا)

Time (سال، فصل، ماه، هفته، روز)

یک مکعب اطلاعاتی اجازه می‌دهد که داده‌ها در ابعاد مختلفی مدلسازی شده و استفاده شوند. عموماً سازمانها با توجه به نیازهای آینده‌شان ابعاد مختلفی از داده‌ها را نگهداری می‌کنند. به‌عنوان مثال داده‌های فروش در سازمانها نیاز است با دیدگاههای زیر ذخیره شوند: در فرایند فروش مهم است که دقیقاً چه اقلامی فروخته شده‌اند. نام آنها چه بوده است نام تجاری آنها چیست و دیگر اطلاعات مرتبط. زمانهای فروش نیز مهم است. اینکه فروش روزانه، ماهانه، فصلی و سالانه هر کدام چه میزان بوده است. اگر فروش در مکانهای مختلف جغرافیایی صورت گرفته است، هر کدام چگونه بوده‌اند. شکل (٦-١) این ابعاد را نشان می‌دهد.

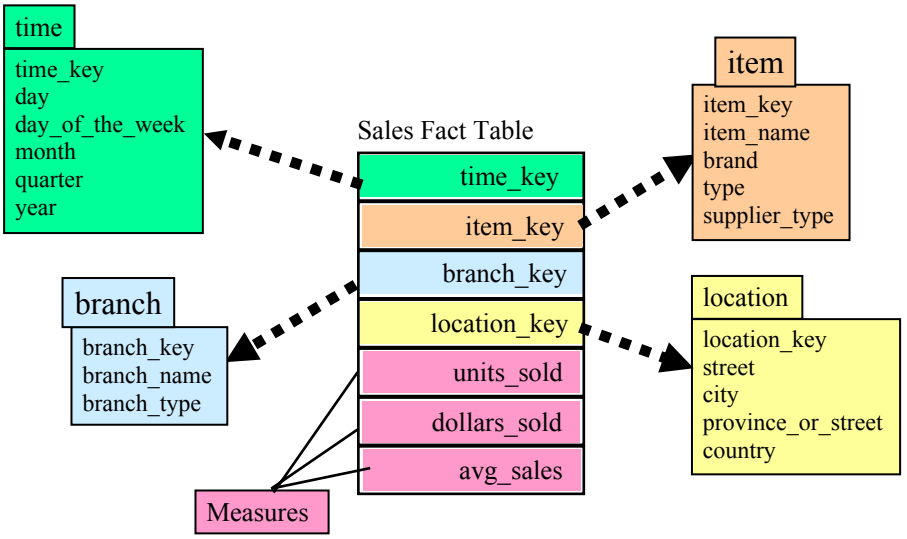


شکل ٦-١) ابعاد داده فروش

مدل مفهومی انباره‌داده‌ها

در ادامه به چند مفهوم اساسی در انباره‌داده‌ها اشاره می‌کنیم:

مدل ستاره‌ای[1]: یک جدول اصلی که متصل به مجموعه‌ای از جداول دیگر باشد مدل ستاره‌ای نامیده می‌شود. شکل (٦-٢) یک مدل ستاره‌ای می‌باشد که جدول اصلی فروش[2] را به جداول دیگر از جمله اقلام فروش[3]، مکان[4] و زمان[5] و شعب فروش[6] متصل می‌کند.



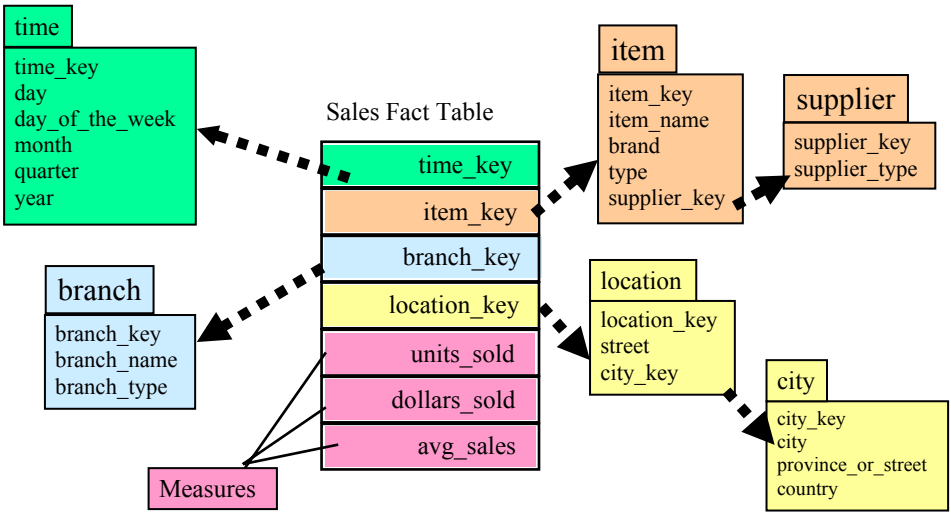
شکل ٦-٢) مدل ستاره‌ای

اگر در مدل ستاره‌ای جداول جانبی با شکستن به چند جدول، نرمال شوند. این مدل تبدیل به مدل برف‌دانه[7] می‌شود. به‌عنوان مثال در شکل (٦-٣) جدول مکان به دو

---

[1]- Star Schema
[2]- Sales
[3]- Item
[4]- Location
[5]- Time
[6]- Branch
[7]- Snow-Flake Schema

جدول شکسته شده است و بخشی از اطلاعات آن در جدول دیگری به نام شهر[۱] وارد شده است.

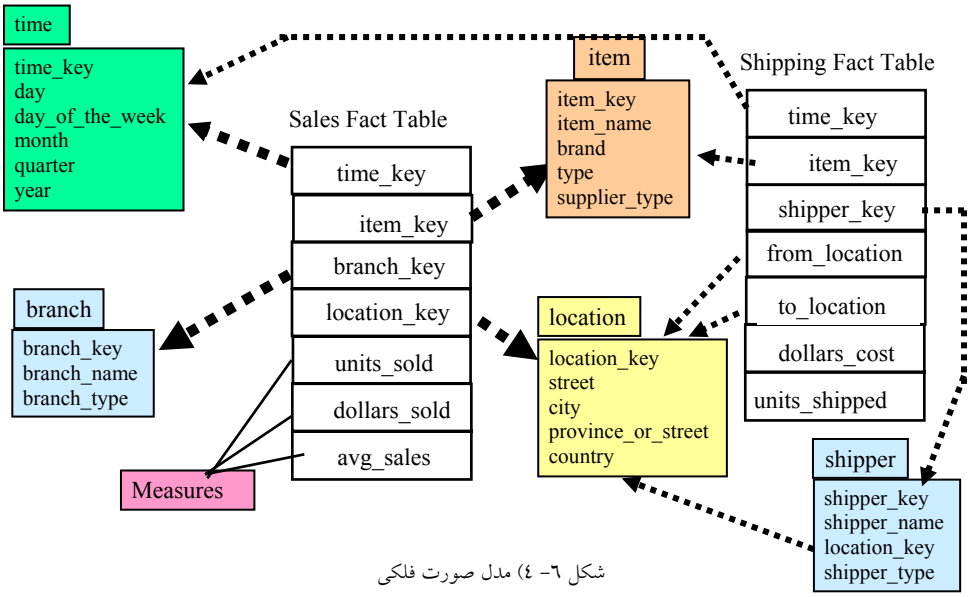


شکل ٦-۳) مدل برف‌دانه‌ای

حال اگر این اطلاعات اصلی به همراه جداولش به اطلاعات دیگری نیز مرتبط باشند و بخواهیم این ارتباطات را نیز نمایش دهیم از مدل صورت فلکی[۲] استفاده می‌کنیم. شکل (٦-٤) ارتباطات جدول اصلی فروش را با جدول اصلی دیگری به نام حمل و نقل[۳] نشان می‌دهد.

---

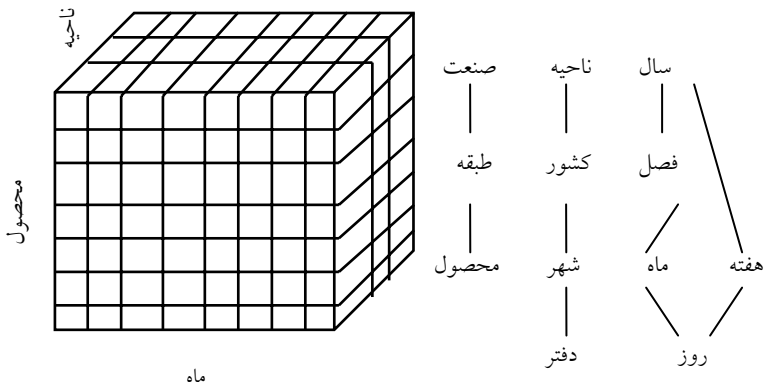
[1]- City
[2]- Fact Constellation
[3]- Shipping



شکل ٦– ٤) مدل صورت فلکی

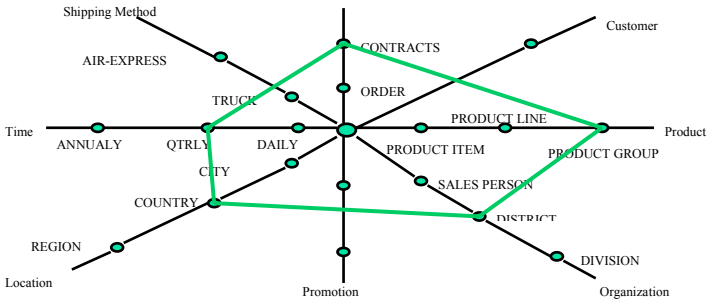
داده‌های چند بعدی

«فروش» یک مفهوم چندبعدی است که از بخشهای محصول، زمان فروش و محل فروش تشکیل شده است و می‌توان دادهٔ چند بعدی فروش را به شکل زیر نمایش داده در شکل (٦–٥)، سلسله مراتب ابعاد داده‌ها نمایش داده شده‌اند.



شکل ٦–٥) ابعاد مختلف داده‌های فروش

اگر ابعاد داده‌ها بیشتر از سه بعد باشد، می‌توان با مدل شبکه ستاره‌ای[1] آن را نمایش داد. شکل (٦-٦) داده‌های مرتبط با ابعاد فروش را نشان می‌دهد. روی هرمحور در این مدل یک بعد نمایش داده می‌شود. به‌عنوان مثال بعد زمان روی یک محورنشان داده شده و علاوه بر آن سلسله مراتب زمان نیز که عبارتند از روزانه، فصلی و سالانه روی این محور نشان داده می‌شوند به‌عنوان مثال اگر یک محصول خاص، فروش فصلی داشته و در یک مکان جغرافیایی خاص فروخته شود، در این صورت نقاط روی این نمودار که حاوی این اطلاعات می‌باشند با یک خط شکسته به هم متصل می‌شوند و بدین ترتیب داده‌های این محصول خاص با ابعاد مورد نظر نمایش داده می‌شود.



شکل ٦-٦) مدل شبکه ستاره‌ای

زبان *MQL* جهت پیاده‌سازی انباره‌داده‌ها

زبان *MQL*[2] شبیه زبان *SQL*[3] می‌باشد و برای تعریف هر کدام از مفاهیم ارائه شده در بخش قبلی روشهای خاصی وجود دارد. جهت تعریف یک جدول اصلی از دستور *Define Cube* و برای تعریف جداول جانبی از دستور *Define Dimension* استفاده می‌شود. مثال زیر دستورات مربوط به شکل مدل ستاره‌ای را به زبان *MQL* نشان می‌دهد.

---

[1]- Star Net
[2]- Mining Query Language
[3]- Structured Query Language

define cube sales_star [time, item, branch, location[:
dollars_sold = sum(sales_in_dollars), avg_sales = avg(sales_in_dollars) , units_sold = count(*)
define dimension time as (time_key, day, day_of_week, month, quarter, year)
define dimension item as (item_key, item_name, brand, type, supplier_type)
define dimension branch as (branch_key, branch_name, branch_type)
define dimension location as (location_key, street, city, province_or_state, country)

دستورات زیر نیز تعریفی از مدل برف دانه‌ای را با استفاده از زبان *MQL* بیان می‌کنند.

define cube sales_snowflake [time, item, branch, location[:
dollars_sold = sum(sales_in_dollars), avg_sales = avg(sales_in_dollars), units_sold = count(*)
define dimension time as (time_key, day, day_of_week, month, quarter, year)
define dimension item as (item_key, item_name, brand, type, supplier(supplier_key, supplier_type))
define dimension branch as (branch_key, branch_name, branch_type)
define dimension location as (location_key, street, city(city_key, province_or_state, country))

فرایند طراحی انباره‌داده

برای طراحی یک انباره‌داده مؤثر، اولین مرحله، درک و تحلیل نیازهای کسب و کار و ساخت چارچوب تحلیل کسب و کار می‌باشد. ساخت یک سیستم اطلاعاتی پیچیده و بزرگ می‌تواند مانند ایجاد یک ساختمان بزرگ و پیچیده در نظر گرفته شود که مالک، معمار و سازنده آن دیدهای مختلفی داشته واین نظرات برای تشکیل یک چارچوب پیچیده ترکیب می‌شوند. دیدگاه‌های مختلفی در طراحی انباره‌داده وجود دارد که عبارتند از:

- **دید بالا به پایین**[1]: روش بالا به پایین عبارت است از روشی که در آن ابتدا یک طرح کلی ایجاد شده و سپس به جزئیات پرداخته می‌شود.

[1]- Top-Down

- **دید پایین به بالا**: ابتدا نمونه‌های کوچک ساخته شده و سپس نمونه گسترش داده می‌شود.
- **دید پرس و جوی کسب وکار**[1]: یک شکل کلی از داده‌های انباره‌داده بر مبنای نگرش کاربر نهایی ایجاد می‌شود.

از دیدگاه مهندسی نرم افراز دو روش عمده جهت طراحی انباره‌داده‌ها وجود دارد که عبارتند از:

- **مدل آبشاری**[2]: این مدل یک روش ساخت‌یافته و نظام‌مند بوده که در ایجاد انباره‌داده، گام به گام جلو می‌رود.
- **مدل مارپیچی**[3]: این روش یک نوع مدلسازی سریع است که در ابتدا یک مدل کوچک ساخته شده و سپس با بررسی مجدد آن را بهبود می‌دهند.

معماری انباره‌داده

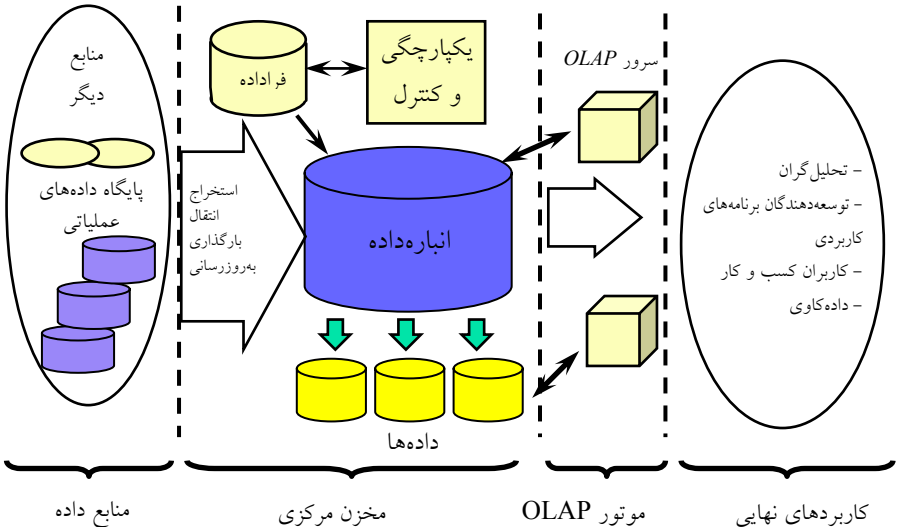
رویکرد چند لایه در انباره‌داده‌ها نیازمند این است که داده‌ها به شکلهای مختلف درآیند. این رویکرد باعث به‌وجود آمدن یک سیستم جامع برای مدیریت داده به منظور تصمیم‌گیری می‌شود. مهم‌ترین اجزاء این سیستم همان‌طور که در شکل (٦-٧) نشان داده شده است عبارتند از:

- سیستمهای منبع[4]، جایی که داده‌ها از آنجا می‌آیند و همان سیستمهای عملیاتی می‌باشند.
- استخراج، انتقال و بارگذاری داده میان منابع مختلف داده.
- مخزن مرکزی[1]، محل اصلی ذخیره‌سازی داده در انباره‌داده‌ها است و یک پایگاه دادهٔ رابطه‌ای با مدل منطقی می‌باشد.

---

[1]- Business Query
[2]- Water Fall
[3]- Spiral
[4]- Source System

- مخزن فراداده[2]، توضیح می‌دهد که چه چیزهایی وجود داشته و در کجا موجود هستند.
- فراداده، دسترسی سریع و اختصاصی را برای کاربران نهایی و برنامه‌های کاربردی فراهم می‌کند.
- بازخور عملیاتی، سیستمهای پشتیبان تصمیم را با سیستمهای عملیاتی یکپارچه می‌کند.
- کابران نهایی، مهم‌ترین دلیل اصلی توسعه انباره‌های داده در مرحله اول می‌باشند. آنها از داده‌ها و دانش استخراج شده از آنها استفاده می‌کنند.



شکل ٦-٧) معماری انباره‌داده‌ها

تقریباً همهٔ مؤلفه‌هایی که ذکر شد در تمامی انباره‌های داده وجود دارند. داده همانند آب می‌باشد که از منابع سیستم سرچشمه گرفته و در انباره‌داده جاری شده تا به کاربران

---

[1]- Central Repository
[2]- Metadata Repository

نهایی ارائه شود. این مؤلفه‌ها در بسترهای سخت‌افراز، نرم‌افراز و شبکه سوار شده‌اند، این زیرساختارها، باید به اندازه کافی قوی باشند تا نیازمندی‌های کاربران نهایی و همچنین نیازمندیهای پردازش و رشد داده را پوشش دهند. در ابتدا با چهار عمل اصلی استخراج[۱]، به‌روزرسانی[۲]، بارگذاری[۳] و انتقال[۴]، داده‌ها از پایگاه داده‌های معمولی جمع‌آوری شده و به انباره‌داده فرستاده می‌شوند. داده‌های موجود در انباره‌داده‌ها مستقیما با دیگر سیستمها در ارتباط نیستند و اگر یک سیستم عملیاتی بخواهد از داده‌های انباره‌داده استفاده کند از بازارچه داده‌ها[۵] استفاده می‌کند. بازارچه داده‌ها بخشی است که داده‌های مربوط به یک برنامهٔ کاربردی خاص را به‌طور موقت از انباره‌داده‌ها دریافت کرده و در اختیار کاربر می‌گذارد. در واقع به‌جز ذخیره و بازیابی اطلاعات هیچ عملیات دیگری بر روی انباره‌داده‌ها امکان‌پذیر نیست. همان‌طور که اشاره شد بازارچه داده‌ها یک سیستم تخصصی است که کلیه داده‌های مورد نیاز یک بخش یا یک برنامه کاربردی را فراهم می‌کند. بازارچه داده‌ها معمولاً در سیستمهای گزارش‌دهی مورد استفاده قرار می‌گیرند. این قبیل بازارچه داده‌ها معمولاً از فنآوری *OLAP* استفاده می‌کنند. نیازی نیست که تمامی اطلاعات بازارچه داده مستقیماً از مخزن مرکزی آمده باشند، درواقع مخزن مرکزی یکی از منابع داده‌ای بازارچه داده‌ها می‌باشد.

فراداده‌ها، داده‌های مربوط به داده‌ها هستند. فراداده‌ها وضعیت‌های مختلف داده‌ها را توصیف می‌کنند. مثالهایی ساده از توصیف فراداده‌ها عبارتند از:

- اطلاعات ساختاری داده‌ها چگونه ذخیره و سازماندهی شده‌اند.
- اطلاعات متریک: مقدار داده‌ها و نحوه توزیع آنها چگونه است.
- اطلاعات تجاری: داده‌ها چگونه استفاده می‌شوند.

---

[1]- Extract
[2]- Refresh
[3]- Load
[4]- Transform
[5]- Data Mart

این مخزن می‌تواند به‌عنوان یکی از مؤلفه‌های بانک اطلاعاتی تلقی شود. در واقع فراداده، ابزاری را در اختیار کاربران نهایی قرار می‌دهد تا به راحتی در انباره‌داده به جستجو بپردازند.

انواع انباره‌داده

از نقطه نظر معماری سه مدل انباره‌داده وجود دارد: انبارهٔ بنگاه[1]، بازارچه داده و انباره مجازی[2].

- **انبارهٔ بنگاه اقتصادی**: این مدل کلیه اطلاعات درباره موضوعات معین داخل سازمان را گردآوری می‌کند. معمولاً از یک یا چند سیستم عملیاتی و یا ارائه کنندگان اطلاعات، داده‌ها فراهم می‌شوند. این مدل شامل داده‌های کلی و داده‌های جزء می‌باشد و می‌تواند در اندازه‌های کوچکی از گیگابایت تا هزاران گیگابایت، ترابایت و یا بیشتر مرتب شود. یک انباره‌داده بنگاه می‌تواند تحت سیستمهای مین-فریم[3] سنتی، ابرسرورهای *unix* یا بسترهای معماری موازی پیاده‌سازی شود. این انباره نیازمند مدلسازی کسب و کار گسترده می‌باشد که طراحی و ساخت آن ممکن است سالهای زیادی به طول انجامد.
- **بازارچه داده**: شامل زیرمجموعه‌ای از داده‌های سازمانهای گسترده است که شامل داده‌های مرتبط با گروه ویژه‌ای از کاربران می‌باشد. به‌عنوان مثال یک بازارچه داده بازاریابی می‌تواند به موضوعاتی مانند مشتری، اقلام جنس و فروش محدود شود. داده‌های موجود در بازارچه داده تمایل به خلاصه شدن دارند. بازارچه‌های داده اغلب تحت سرورهای اداری ارزان قیمت که بر مبنای *unix*، *windows/NT* یا *OS/2* هستند، پیاده‌سازی می‌شوند. چرخه پیاده‌سازی بازارچه‌های داده بیشتر در

---

[1]- Enterprise Warehouse
[2]- Virtual Warehouse
[3]- Main Frame

مقیاس هفته برآورد می‌شود. بازارچه‌های داده بر اساس منبع داده به دو دسته زیر تقسیم می‌شوند:

- **بازارچه‌های داده مستقل**: این نوع بازارچه‌های داده مرتبط با بیش از یک سیستم عملیاتی بوده و یا مرتبط با داده‌هایی هستند که به‌طور محلی درون یک بخش خاص یا محدوده جغرافیایی خاص تولید شده‌اند.
- **بازارچه‌های داده وابسته (غیر مستقل)**: به‌طور مستقیم از انباره بنگاه استخراج می‌شوند.

- **انباره مجازی**: این انباره مجموعه‌ای از برشها بر اساس دیدگاهها[1]ی مختلف بر روی پایگاههای داده عملیاتی می‌باشد. برای پردازش یک پرس‌وجوی مؤثر فقط برخی از دیدگاههای خلاصه به‌کار می‌رود. ساخت یک انباره مجازی آسان است اما نیازمند ظرفیت اضافه در سرورهای مربوطه می‌باشد.

انباره‌داده و سیستم‌های عملیاتی

انباره‌داده مخزنی از داده‌های یک بنگاه است که اغلب برای تحقیق و پشتیبانی تصمیمات از آن استفاده می‌شود. این انباره در مقابل سیستم عملیاتی سازمان[2] که با تراکنشهای روزمره سازمان سروکار دارد (مثلاً *OLTP*) قرار می‌گیرد. از آنجایی‌که سیستمهای عملیاتی اغلب به‌صورت مستقل طراحی می‌شوند، برای پردازش یکپارچه داده‌ها به مخزنی خاص نیاز دارند که کلیه داده‌های مرتبط را یکجا دربرگیرد. مزیت دیگر انباره‌داده این است که فعالیتهای تحلیلی مانند *OLAP* را از سیستم عملیاتی جدا می‌کند. از انباره‌داده‌ها می‌توان برای داده‌کاوی، مصورسازی داده‌ها[3]، ارائه گزارشات

---

[1]- View
[2]- Operational System
[3]- Data Visualization

پیشرفته و به‌کارگیری انواع ابزارهای *OLAP* استفاده کرد. همچنین اطلاعات از منبع اصلی مستقل می‌شوند که این مسئله در زمانی که اطلاعات اجرایی در حال تغییر هستند بسیار اهمیت دارد. علاوه بر این اگر ساختار ذخیره اطلاعات جهت یک نوع عملیات خاص، طراحی شده باشد (مثلاً ساختار ستاره‌ای برای عملیات فروش) جستجو در ابعاد مختلف نیز بسیار ساده‌تر خواهد شد.

با توجه به آنچه گفته شد می‌توان در موارد زیر انباره‌داده‌ها را از سیستمهای عملیاتی متمایز نمود:

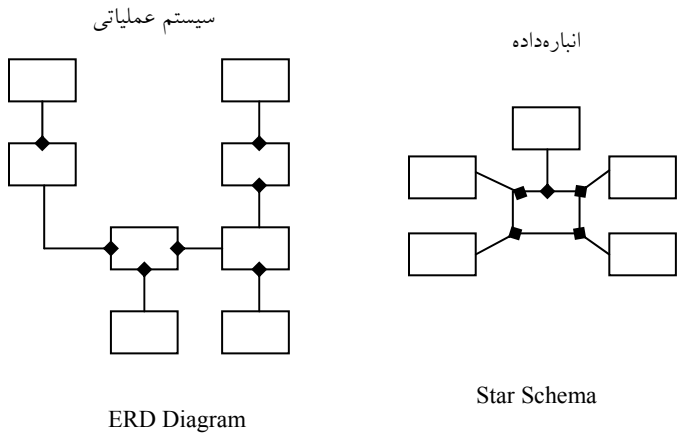
- اهداف
- ساختار
- اندازه
- بهینه بودن عملکرد
- فنآوری‌های استفاده شده

جدول (٦-٢) این تفاوت‌ها را به تفصیل نشان می‌دهد:

جدول ٦-٢) مقایسه سیستمهای عملیاتی و انباره‌داده‌ها

سیستمهای عملیاتی	انباره‌داده‌ها
بر مبنای کاربرد	بر مبنای موضوع
کوچک *(several MB up to GB)*	بزرگ *(hundreds of GB up to TB)*
داده‌های جاری	داده‌های تاریخی
جداول نرمال	جداول غیرنرمال
به روز شدن همیشگی	به روز شدن دسته‌ای
جستجوهای ساده	جستجوهای پیچیده

ساختار اغلب انباره‌ها به‌صورت ستاره‌ای طراحی می‌شود زیرا اولاً موضوع‌گرا هستند و ثانیاً از آنجا که هر نوع ارتباطی بین موجودیتها ممکن است مورد مطالعه قرارگیرد، تا حد امکان لازم است موجودیتها بدون واسطه مرتبط شده باشند.



شکل ٦-٨) تفاوت ساختارها

شکل (٦-٨) تفاوت در ساختار دو نوع سیستم را نشان می‌دهد. عامل مهم دیگری که در انباره‌داده‌ها باید به آن توجه شود نوع و ماهیت داده‌ها است چرا که داده‌ها باید دقیق[1]، سازگار[2]، به‌موقع[3]، یکپارچه، کامل[4]، معتبر[5] بوده و در راستای قواعد کسب و کار باشند و به‌علاوه باید به خوبی قابل درک[6] باشند.

کاربران نهایی انباره‌داده‌ها

کاربران نهایی در واقع آخرین و مهم‌ترین مؤلفه در هر انباره‌داده می‌باشند. این کاربران نهایی تحلیل‌گران، توسعه دهندگان برنامه‌های کاربردی و کاربران کسب و کار می‌باشند.

تحلیل‌گران

[1]- Accurate
[2]- Consistent
[3]- Timely
[4]- Complete
[5]- Valid
[6]- Well Understood

تحلیل‌گران نیاز دارند که به غالب داده‌ها به منظور استخراج مدلهای مختلف و تهیه گزارشها دسترسی داشته باشند. آنها از یکسری ابزارهای خاص از جمله بسته‌های آماری، ابزارهای داده‌کاوی و صفحات گسترده استفاده می‌کنند. معمولاً تحلیل‌گران به عنوان نخستین ذینفعان انباره‌های داده محسوب می‌شوند. تعداد افراد خبره‌ای که در این دسته قرار می‌گیرند بسیار کم است. کاری که آنها انجام می‌دهند از درجه اهمیت بسیار بالایی برخوردار بوده و بسیار پیچیده است. یک انباره‌داده، داده‌های پاکسازی شده را به‌طور یکجا جمع‌آوری می‌کند. این داده‌ها باید ویژگیهای زیر را دارا باشند تا بتوانند به راحتی مشکلات تحلیل‌گران را حل کنند:

- داده‌های سراسر بانک اطلاعاتی باید سازگار باشند.
- داده‌ها باید با زمان سازگار باشند.
- یک سیستم باید بتواند به پایین‌ترین سطح اطلاعات و تراکنشها دسترسی داشته باشد.

توسعه دهندگان برنامه‌های کاربردی

انباره‌های داده معمولاً طیف گسترده‌ای از برنامه‌های کاربردی را پشتیبانی می‌کنند. به‌منظور توسعه یک برنامه کاربردی پایدار انباره‌های داده نقش به‌سزایی دارند. اول اینکه برنامه‌ای را که آنها توسعه می‌دهند باید در برابر تغییرات در ساختار انباره‌داده ایمن باشد. ایجاد جداول جدید، فیلدهای جدید و تغییرات ساختاری جداول باید حداقل تأثیر را بر روی برنامه‌های کاربردی موجود داشته باشد. وجود یکسری مشخصه ویژه بر روی داده‌ها به تحقق این امر کمک می‌کند.

علاوه بر آن داشتن دانشی درباره اینکه هر برنامه کاربردی از چه فیلدهایی استفاده می‌کند، می‌تواند مانع بن بست[1] شود. توسعه دهندگان سیستم نیاز دارند بدانند ارزش معتبر فیلدها چیست و علاوه بر آن ارزش هر فیلد به چه معناست. پاسخ این سؤالات

---

[1]- Gridlock

هدف فراداده می‌باشد. فراداده مستنداتی را در ارتباط با ساختار داده ارائه می‌کند. از آنجا که کسب و کار واقعی نیازمند توسعه برنامه‌های کاربردی می‌باشد، درک نیاز توسعه‌دهندگان و دسترسی آنها به انباره‌داده‌ها از درجه اهمیت بالایی برخوردار است. انباره‌های داده به مرور دچار تغییر می‌شوند و برنامه‌های کاربردی نیز همچنان از آنها استفاده می‌کنند. این امر مهم‌ترین عامل موفقیت کنترل و مدیریت تغییرات می‌باشد.

کاربران کسب و کار

کاربران کسب و کار نیز از جمله استفاده‌کنندگان انباره‌داده می‌باشند. نیازمندیهای آنها موجب توسعه برنامه‌های کاربردی و معماری انباره‌داده می‌شود. در برخی از کسب و کارها کاربران نهایی تنها با گزارشهای تهیه شده از انباره‌های داده‌ها، یا صفحات گسترده سروکار دارند. اینکه افراد بر روی میز خود کامپیوتر داشته و قادر باشند به‌طور مستقیم به انباره‌داده دسترسی داشته باشند، از درجه اهمیت بالایی برخوردار است. کاربران با استفاده از ابزارهای موجود می‌توانند گزارشهای ترسیمی و تحلیلی بسیار جالب و با ارزشی متناسب با نیاز خود از انباره‌داده استخراج کنند. از طرف دیگر آنها همچنین قادر خواهند بود تا به درون مخزن انباره‌داده وارد شده و تا پایین‌ترین سطح، داده‌های موجود را بررسی کنند.

کاربردهای انباره‌داده

سه کاربرد مهم برای انباره‌داده شناخته شده است: یکی از این کاربردها، داده‌کاوی می‌باشد که این ارتباط قبلاً به تفصیل توضیح داده شد. کاربردهای دیگر انباره‌داده در پردازش اطلاعات و پردازشهای تحلیلی می‌باشد. به‌عنوان مثال پرس‌وجوها و تحلیلهای آماری و ایجاد جداول، نمودارها و گزارشهای گرافیکی با استفاده از پردازش اطلاعات در انباره‌داده‌ها قابل ارائه می‌باشد.

انباره‌داده‌ها به تحلیل‌گران کسب و کار کمکهای شایانی می‌کند:

- داشتن انباره‌داده، یک مزیت رقابتی است چرا که با دسترسی سریع و به‌موقع به داده‌ها به غلبه بر رقبا کمک می‌کند.
- یک انباره‌داده می‌تواند بهره‌وری کسب و کار را توسعه دهد. چرا که قادر است اطلاعات سازمان را خیلی دقیق، با سرعت و به‌طور کارا تشریح کند.
- یک انباره‌داده بازاریابی، ارتباط با مشتریان را تسهیل می‌کند، چرا که داده‌های مرتبط با مشتریان و اقلام مختلف در کلیه بخش‌ها و کلیه فروشگاهها را در اختیار قرار می‌دهد. همچنین انباره‌داده‌ها می‌توانند توسط ردیابی روندها، الگوها و استثنائات در دوره‌های زمانی طولانی با یک روش سازگار و قابل اطمینان به کاهش هزینه‌ها کمک کنند.

منابع

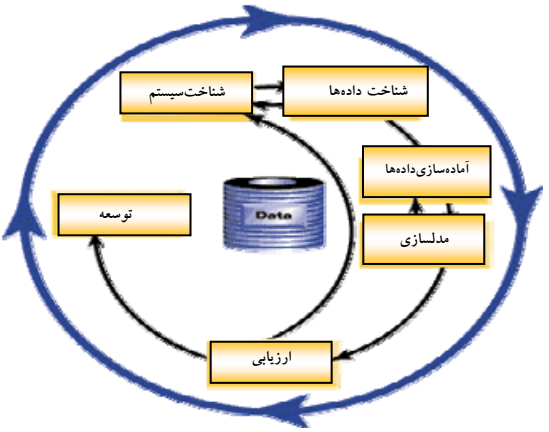
1) *Han ،J. ،and Kamber ،M. (2006) Data Mining: Concepts and Techniques ،San Francisco ،CA: Morgan Kaufmann.*

فصل هفتم

متدلوژی اجرا و پیاده‌سازی پروژه‌های داده‌کاوی

روشهای مختلفی برای پیاده‌سازی و اجرای پروژه‌های داده‌کاوی وجود دارد. یکی از روشهای بسیار قوی، متدلوژی *CRISP*[1] می‌باشد. این متدلوژی از گامهای شناخت سیستم، شناخت داده‌ها، آماده‌سازی داده‌ها، مدلسازی، ارزیابی و توسعه سیستم تشکیل شده است. هر کدام از این گامها به زیر بخشهایی تقسیم می‌شوند.

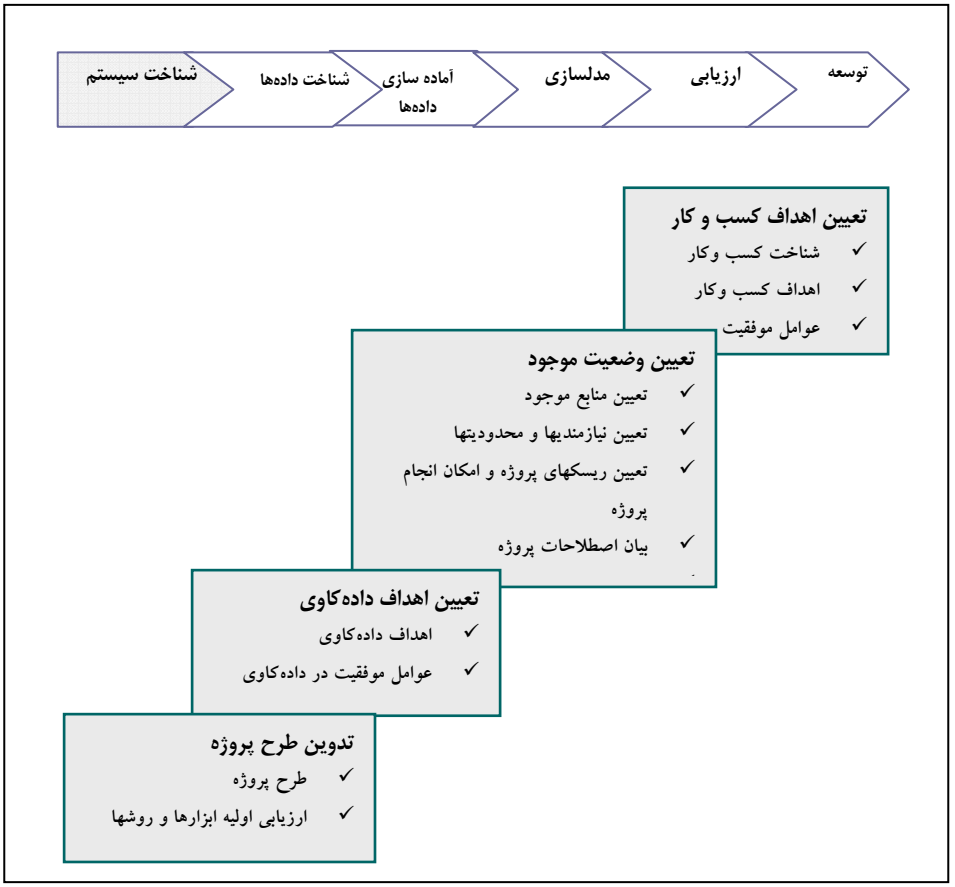
[1]- Cross Industry Standard Process for Data Mining



شکل ۷-۱) گامهای متدولوژی CRISP

درگام شناخت سیستم ابتدا به شناخت کسب وکار مورد نظر پرداخته می‌شود. سپس اهداف مورد نظر و عوامل موفقیت کلیدی آن تعیین شده و دوباره اهداف کسب و کار بازنگری می‌شود. شناسایی فرصتها و عوامل موفقیت کلیدی یک کسب وکار قدم بسیار مهمی می‌باشد، چرا که باعث افزایش اطلاعات و انجام بهتر کارها می‌شود. در واقع در این گام به تعیین زمینه‌ها و عواملی می‌پردازیم که داده‌ها باعث افزایش ارزش در آنها می‌شوند. پس از تدوین اهداف مورد نظر کسب وکار، می‌بایست به شناخت وضعیت موجود پرداخت. به‌منظور تعیین دقیق وضعیت موجود، به تعیین منابع موجود پرداخته و نیازمندیها و محدودیتهای موجود تعیین می‌شوند. شناخت ریسکهای بر سر راه پروژه و نیازمندیهای پروژه کمک می‌کنند تا طرح امکان‌سنجی پروژه تدوین شود. در این قدم به‌منظور تدوین طرح امکان‌سنجی، تعیین منابع پروژه ضروری می‌باشد. منابع پروژه عبارتند از: منابع انسانی، منابع مالی، تجهیزات و دیگر منابع. به همین دلیل در این گام تعیین سودها و هزینه‌های پروژه امری اجتناب‌ناپذیر می‌باشد. تعیین اهداف داده‌کاوی و تدوین طرح پروژه نیز از دیگر بخشهایی است که در این گام باید به آنها پرداخت. به عنوان مثال در یک کسب و کار خاص اهداف زیر برای پروژه داده‌کاوی تعیین می‌شود:

- برنامه‌ریزی بازاریابی محصولات و خدمات جدید
- قیمت‌گذاری محصولات و خدمات جدید
- شناخت مشتریان ناراضی و جلوگیری از رویگردانی آنها
- تعیین بازار هدف

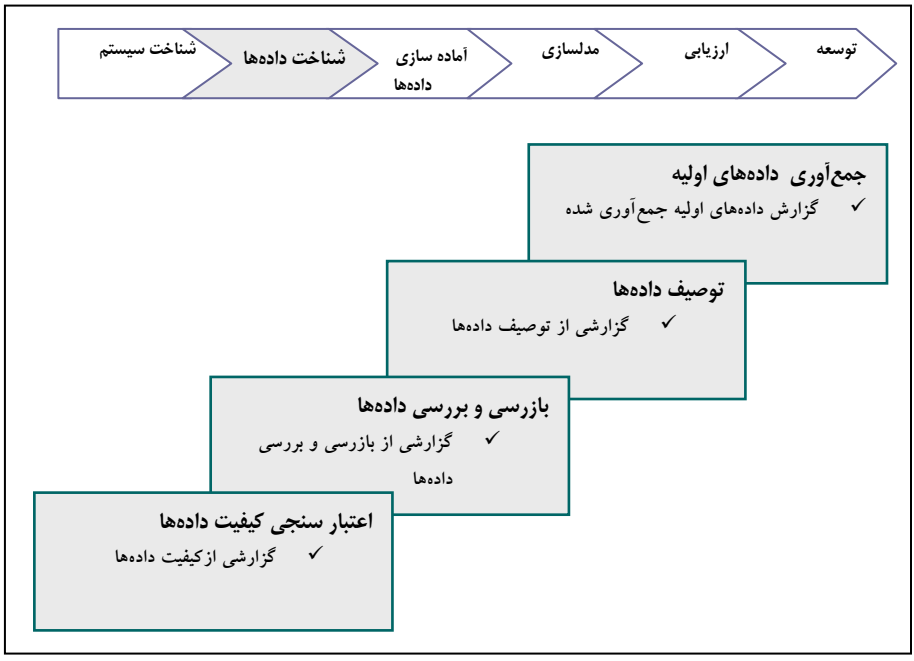


شکل ۷-۲) جزئیات مربوط به گام شناخت سیستم

در این صورت پروژه داده‌کاوی باید در راستای این اهداف صورت گیرد. مثلاً داده‌های مرتبط با این اهداف را جمع‌آوری و یکپارچه کرده و در برآورده کردن این اهداف و

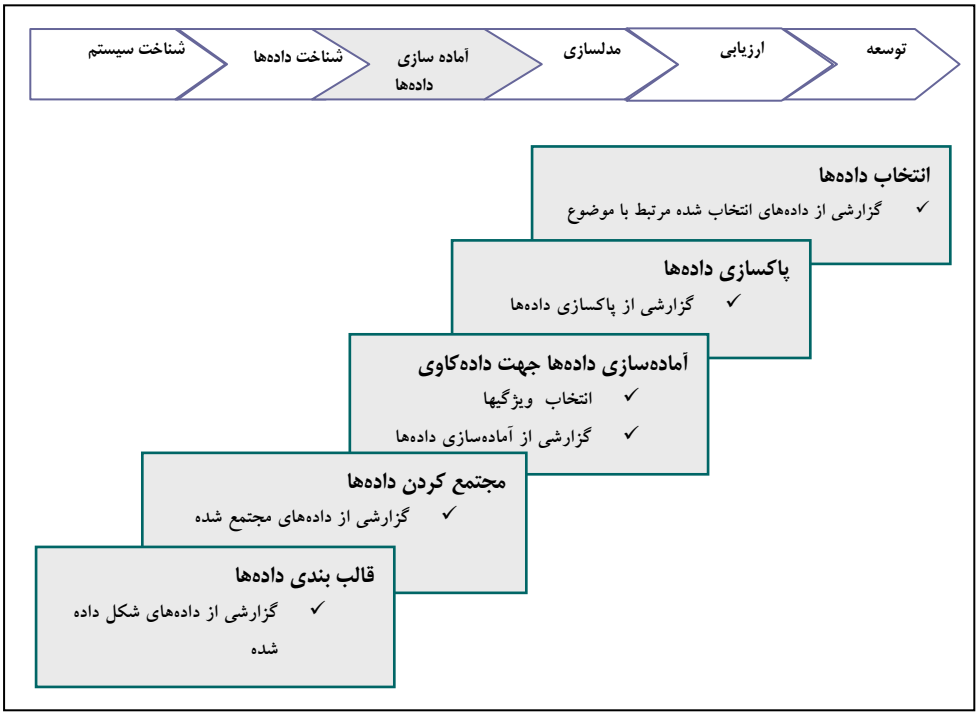
شاخصهای مرتبط با آن بکوشد. در شکل (۷-۲) بخشها و زیربخشهای این گام نشان داده شده است.

پس از شناخت کسب و کار به سراغ شناخت داده‌ها می‌رویم. شناخت داده‌ها عبارت است از جمع‌آوری داده‌های اولیه، توصیف داده‌ها، بازرسی و بررسی داده‌ها و اعتبارسنجی کیفیت داده‌ها. کارآیی داده‌کاوی مستقیماً مرتبط با داده‌های مورد استفاده دارد. هر اندازه داده‌ها دقیق‌تر جامع‌تر و با کیفیت بهتری باشند خروجی داده‌کاوی کارآتر خواهد بود. بنابراین انتخاب وجمع‌آوری داده‌های درست، توصیف آنها، یکپارچه‌سازی قالب آنها به‌منظور استفاده در داده‌کاوی، از اهمیت بسیار بالایی برخوردار می‌باشد. علاوه بر این بازرسی و بررسی داده‌ها به‌منظور تعیین میزان کیفیت آنها بسیار مهم می‌باشد. در شکل (۷-۳) به این گامها و خروجیهای هر مرحله اشاره شده است.



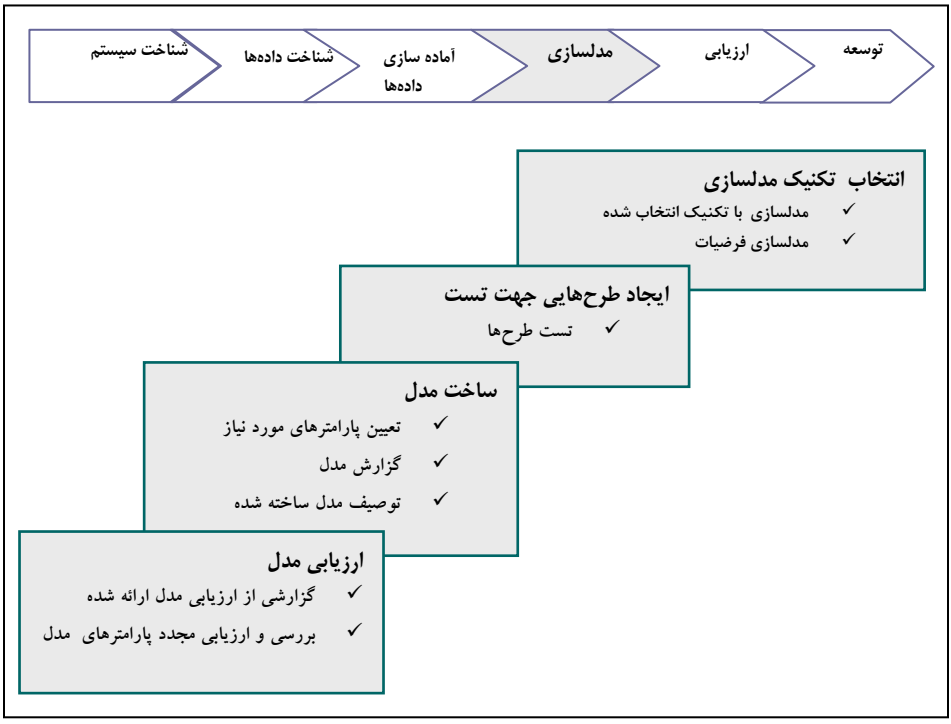
شکل ۷-۳) جزئیات مربوط به گام شناخت داده‌ها

گام آماده‌سازی داده‌ها عبارت است از: انتخاب داده‌ها، پاکسازی داده‌ها، آماده‌کردن داده‌ها جهت داده‌کاوی، مجتمع کردن آنها و قالب بندی داده‌ها. برای اجرای هرکدام از این زیر بخشها، فعالیتهای دیگری نیز ضروری است که در شکل (۷-٤) آمده است. جمع‌آوری و محافظت از داده‌ها گام بسیار مهمی می‌باشد. اصولاً چون قالب و نوع داده‌ها در طول زمان تغییر می‌کند، ممکن است قالب بسیاری از داده‌های موجود متفاوت باشد. همچنین به‌علت اینکه داده‌ها از منابع مختلف داخلی و خارجی جمع‌آوری شده و یکپارچه می‌شوند، باز هم ممکن است قالب داده‌ها با هم یکسان نبوده و یا حتی برخی از داده‌های قبلی از بین رفته و دور ریخته شده باشند و بخشهایی از داده‌ها موجود باشد. در داده‌کاوی اهمیت داده‌های قدیمی به هیچ وجه کمتر از داده‌های جدید نمی‌باشد.



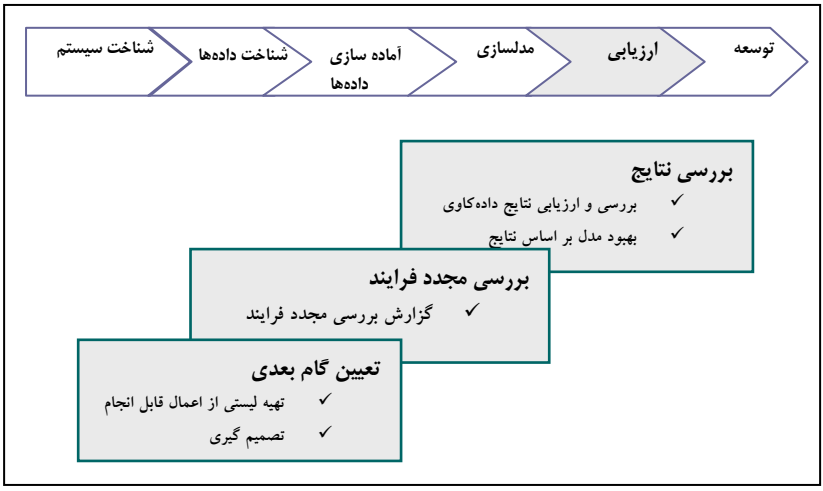
شکل ۷-٤) جزئیات مربوط به گام آماده‌سازی داده‌ها

پس از شناخت داده‌ها و آماده‌سازی آنها، حال می‌توان به مدلسازی پرداخت. در اولین قدم از مدلسازی می‌بایست تکنیک و روش مناسب را انتخاب کرد. انتخاب تکنیک مناسب بسیار تعیین کننده می‌باشد. پارامترهای مورد نیاز مدل نیز پس از تعیین تکنیک و روش مورد استفاده، مشخص می‌شوند. پس از انتخاب مدل و تعیین پارامترها، بخشهای کوچکی از پروژه تعریف شده و پس از اجرا شدن، در هر مرحله به دقت تست می‌شوند تا کیفیت مدل ایجاد شده تضمین شود. در این مرحله اگرمدل مورد نظر دقت لازم را نداشت و یا کیفیت مطلوب را حاصل نکرد، ابتدا به تغییر پارامترهای مدل می‌پردازیم و مجدداً مدل را تست می‌کنیم. اگر هنوز کیفیت لازم را کسب نکرده بود، مدل را تغییر داده و مدل جدیدی می‌سازیم. برای انجام هر کدام از زیر بخشهای مدلسازی، فعالیتهای دیگری نیز ضروری است که در شکل (۷-۵) آمده است.



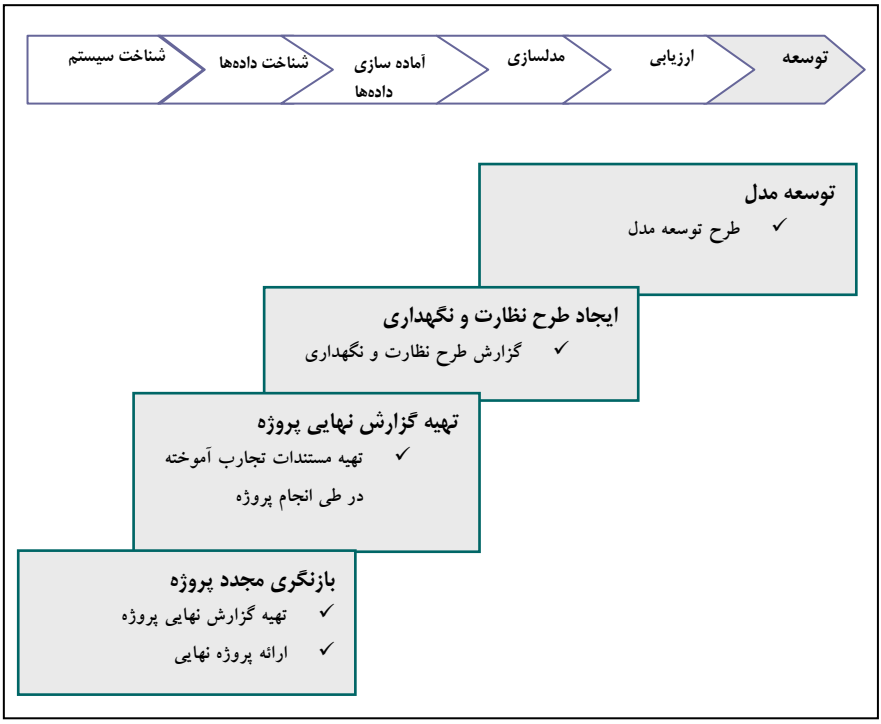
شکل ۷-۵) جزئیات مربوط به گام مدلسازی

پس از مدلسازی، حال می‌بایست به ارزیابی نتایج حاصل از مدل پرداخت. نتایج ارزیابی باعث بهبود مدل شده و مدل را قابل استفاده می‌کند. در این گام اعتبار مدل بررسی شده وگزارشی از کل فرایند تهیه می‌شود. در انتها نیز لیستی از اقدامات اصلاحی قابل انجام تهیه شده و به‌عنوان راهکار ارائه شده و تصمیم‌گیریها بر این اساس انجام می‌شود.



شکل ۷- ٦) جزئیات مربوط به گام ارزیابی

پس از استخراج لیست اقدامات قابل انجام، دورنمایی از طرح توسعه ایجاد می‌شود. در این گام این طرح مجددا بررسی شده و مدون می‌شود. علاوه بر آن طرح نظارت و نگهداری پس از اتمام پروژه نیز در این گام تهیه می‌شود. در پروژه‌های داده‌کاوی تأکید زیادی بر روی مستندسازی تجارب آموخته در طی انجام پروژه وجود دارد و در این گام گزارش مرتبط با آن نهایی می‌شود. در این مرحله پروژه قابل ارائه نهایی بوده و گزارش نهایی آن نیز استخراج شده است.



شکل ۷–۷) جزئیات مربوط به گام توسعه

منابع

۱) راهنمای نرم افزار SPSS Clementine

بخش چهارم

فصل هشتم: سریهای زمانی در داده‌کاوی

فصل نهم: شبکه‌های اجتماعی

فصل دهم: کاربرد داده‌کاوی در مدیریت ارتباط با مشتری

فصل هشتم

سریهای زمانی در داده‌کاوی

یک سری زمانی دنباله‌ای از مشاهدات بر روی یک متغیر مورد توجه است که در نقاط گسسته‌ای از زمان که معمولاً فاصله‌های مساوی دارند (روزانه – هفتگی – ماهانه – فصلی – سالانه)، رخ می‌دهد. تجزیه و تحلیل سریهای زمانی، متضمن توصیف فرآیند یا پدیده‌ای است که تولید دنباله می‌کند. جهت پیش‌بینی سریهای زمانی، لازم است که رفتار فرآیند را با یک مدل ریاضی که قابل تعمیم به آینده باشد، توصیف کرد. معمولاً لازم نیست مدل نماینده مشاهدات خیلی قدیمی یا فراتر از زمان مورد انتظار پیش‌بینی باشد[۱]. مهم‌ترین نکته در داده‌های سری زمانی، آن است که این داده‌ها دارای

همبستگی هستند. از ضریب همبستگی برای تعیین همبستگی بین مقادیر X و Y استفاده می‌شود، اما وقتی خود متغیرهای مستقل، مقادیرشان به هم مرتبط باشد، به آن خودهمبستگی[1] گویند. با توجه به تعریف داده‌های سری زمانی، روشهای آماری مبتنی بر فرض مستقل بودن مشاهدات، مناسب نبوده و به جای آن می‌توان از معادلات خودهمبستگی در تحلیل سریهای زمانی استفاده کرد. [۲]

با توجه به اهمیت و نقش سریهای زمانی، در حوزه‌های مختلفی از آنها استفاده می‌شود که نمونه‌هایی از کاربردهای آن به شرح زیر است. [۲]

- بازرگانی و اقتصاد: مقادیر فروش و قیمتهای ماهیانه، قیمت سهام در روزهای مختلف
- علوم مهندسی: ثبت علائم الکتریکی، ولتاژ و سیگنالها.
- پزشکی: داده‌های الکتروکاردیوگرام و دیگر کاربردها.
- هواشناسی: درجه حرارت روزانه و میزان بارندگی سالیانه.
- کنترل کیفیت: ثبت مشاهدات مربوط به فرایند در نمودارهای کنترل.
- علوم اجتماعی: نرخهای زاد و ولد و مرگ و میر سالیانه.

داده‌کاوی سریهای زمانی[2]

یک سری زمانی ساده‌ترین شکل داده‌های زمانی است. سری زمانی دنباله‌ای از اعداد حقیقی است که به‌صورت منظم در طول زمان گردآوری شده است. هر عدد، نشان‌دهنده مقدار یک متغیر مشاهده شده می‌باشد. همان‌طور که اشاره شد، داده‌های سری زمانی در حوزه‌های مختلفی مثل تحلیل بازار سهام، علوم ارتباطات، پزشکی، داده‌های مالی و غیره مطرح می‌شوند. همچنین داده‌های وب که میزان استفاده از وب سایتهای مختلف را ثبت می‌کنند (برای مثال تعداد کلیکها) را می‌توان با سریهای زمانی

---

[1]- Auto Correlation
[2]- Time Series Data Mining

مدل کرد. در حقیقت، سریهای زمانی برای نمایش بخش بزرگی از داده‌های ذخیره شده در بانکهای اطلاعاتی تجاری به‌کار می‌رود که به تدریج به‌عنوان یک نوع داده متفاوت، اهمیت بیشتری یافته است. اهمیت داده‌های سریهای زمانی موجب تحقیقات زیادی در زمینه تحلیل این نوع داده‌ها شده است. ادبیات آماری در مورد سریهای زمانی بسیار وسیع است و به‌طور عمده به مسائلی مانند شناسایی الگوها و تحلیل روند (مانند رشد خطی فروش شرکت در طول یک سال)، تحلیلهای فصلی (مثلاً فروش زمستانی یک محصول تقریباً دو برابر فروش تابستانی است) و پیش‌بینی (مانند پیش‌بینی فروش فصل آینده) می‌پردازند. این موضوعات کلاسیک در تعدادی از مراجع آماری بررسی شده است. [۵]

هدف اصلی داده‌کاوی سریهای زمانی کشف الگوهای موجود در وقایع و داده‌های یک سری زمانی است. کشف این الگوهای ناشناخته، همگام با استفاده از دیگر روشهای مختلف داده‌کاوی مانند سیستمهای پایگاه داده، آمار، یادگیری ماشینی، شبکه‌های عصبی، تئوری مجموعه‌ها، منطق فازی و غیره در داده‌کاوی اتفاق می‌افتد. از مهم‌ترین کاربردهای داده‌کاوی سریهای زمانی، می‌توان به دسته‌بندی، خوشه‌بندی و کشف قواعد از داده‌ها اشاره کرد. در داده‌کاوی سریهای زمانی دو سؤال اساسی زیر مطرح می‌شود:

- چگونه می‌توان روابط همبستگی در درون سریهای زمانی را پیدا کرد؟
- چگونه می‌توان سریهای زمانی با حجم انبوهی از داده‌ها را تحلیل کرده و الگوهای منظم، روند، تغییرات تصادفی، داده‌های مغشوش و غیره را از آنها استخراج کرد؟[٦]

در ادامه به بررسی جنبه‌های مختلفی از داده‌کاوی سریهای زمانی، با تمرکز بر شناسایی اجزاء سریهای زمانی و روشهای جستجوی تشابه در سریهای زمانی پرداخته می‌شود.

اجزاء سریهای زمانی و تحلیل آنها

یک سری زمانی شامل یک متغیر وابسته (y) می‌باشد که تابعی از زمان است. چنین تابعی به شکل یک نمودار سری زمانی نمایش داده می‌شود. در مطالعه داده‌های سریهای زمانی، همواره دو هدف اساسی زیر دنبال می‌شود:[٦]

- مدلسازی سریهای زمانی با تأکید بر فرآیند ایجاد سریهای زمانی
- پیش‌بینی سریهای زمانی با تأکید بر پیش‌بینی مقادیر متغیرهای سری زمانی

تجزیه و تحلیل روند شامل شناسایی چهار جزء یا مشخصه اساسی هر سری زمانی می‌باشد. [٢]

- **روند خطی و غیرخطی**: تغییر دراز مدت در میانگین یا به‌عبارت دیگر حرکت دراز مدت تدریجی افزایشی یا کاهشی داده‌ها در طول زمان است.
- **تغییر سیکلی**: تغییرات دوره‌ای موجود در داده‌های یک سری زمانی است. معمولاً این نوع افزایشها و کاهشهای لحظه‌ای در داده‌ها، در دوره‌های بیشتر از یکسال اتفاق می‌افتد.
- **تغییرات فصلی**: تغییراتی است که به‌صورت فصلی در داده‌های سری زمانی اتفاق می افتد. این نوع الگوی تغییر در طول یکسال مشاهده می‌شود.
- **تغییرات باقیمانده‌ها یا تصادفی**: اگر سه جزء قبلی از یک سری زمانی حذف شود، سری باقیمانده حاصل می‌شود. که ممکن است تصادفی باشد.

با توجه به توضیحات ارائه شده، می‌توان معادله یک سری زمانی را به‌صورت یکی از دو حالت (٨–١) و (٨–٢) نشان داد. [٦]

$$y = T_t + C_t + S_t + R_t \qquad (١–٨)$$

$$y = T_t * C_t * S_t * R_t \qquad (٢–٨)$$

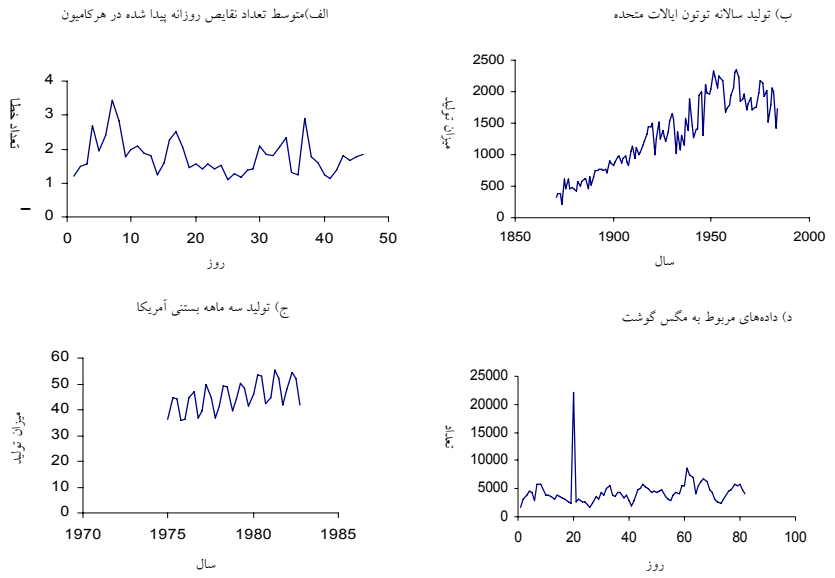
یکی از مهم‌ترین جنبه‌های کاربرد سریهای زمانی، پیش‌بینی می‌باشد. پیش‌بینی سریهای زمانی با استفاده از یک معادله ریاضی، الگویی تاریخی در داده‌های سری زمانی ایجاد

می‌کند. این روش برای پیش‌بینی کوتاه مدت یا بلند مدت مقادیر آینده مورد استفاده قرار می‌گیرد. روشهای مختلفی برای پیش‌بینی سریهای زمانی، مورد استفاده است که از بین آنها روش میانگین متحرک تلفیق‌شده با اتو رگرسیون[1] که به مدل «باکس– جنکینز» نیز موسوم است از اهمیت ویژه‌ای برخوردار است[٦]. برای آشنایی بیشتر با روشهای پیش‌بینی سریهای زمانی، می‌توان به مراجع آمار و اقتصادسنجی مراجعه کرد.

معمولاً مطلوب است که سیستم پیش‌بینی بتواند تغییرات پایدار را مشخص و با تعدیل مدل پیش‌بینی، فرآیند جدید را تعقیب کند. در عین حال سیستم پیش‌بینی، تغییرات تصادفی و موقتی را تشخیص داده و در مقابل آنها واکنش نشان ندهد. در هنگام کار با مدلهای پیش‌بینی با توجه به مراحل مختلف سیکل پیش‌بینی (مثلاً عمر محصول)، لازم است که مدلهای پیش‌بینی مختلفی به‌کارگرفته شوند. مثلاً ممکن است گاهی اوقات فقط روند را حفظ کرده و بقیه علل تغییرات سری زمانی را حذف کنیم. [٤] به‌طور کلی یک سری زمانی را ایستا[2] گویند، هرگاه تغییر منظمی در میانگین و واریانس آن وجود نداشته و تغییرات دوره‌ای اکید حذف شده باشد. نظریه احتمال سریهای زمانی بیشتر با سریهای زمانی ایستا سر و کار دارد و به این دلیل است که در تجزیه و تحلیل سریهای زمانی، برای استفاده از نظریه ایستایی لازم است که سری ناایستا را به ایستا تبدیل کنیم. مثلاً می‌توانیم روند و تغییرات فصلی را از مجموعه داده‌ها حذف کرده و سپس به وسیله یک فرآیند تصادفی ایستا، تغییر در باقیمانده‌ها را الگوسازی کنیم. [٢]

---

[1]- Arima
[2]- Stationary



شکل ۸–۱) چهار سری زمانی

شکل (۸–۱) که چهار سری زمانی را نشان می‌دهد، خصوصیات اجزاء سری‌های زمانی را آشکار می‌کند. به نظر می‌رسد که متوسط تعداد نقایص روزانه پیدا شده در هرکامیون، در انتهای خط تولید کارخانه تولید کامیون که در شکل (۸–۱–الف) نشان داده شده است، در حول سطح ثابتی، نوسان می‌کند. همان‌طورکه گفته شد سریهای زمانی که این پدیده را نشان می‌دهند، ایستا در میانگین نامیده شده و حالتهای ویژه سریهای زمانی ایستا هستند. تولید سالانه توتون ایالات متحده که در شکل (۸–۱–ب) نشان داده شده در حول سطح ثابتی تغییر نمی‌کند، بلکه در کل یک روند رو به بالا را نشان می‌دهد. علاوه بر این، واریانس این سری توتون، با اضافه شدن سطح سری، افزایش می‌یابد. سریهای زمانی که این پدیده را نشان می‌دهند، ناایستا در میانگین و واریانس گفته شده و مثالهایی از سریهای زمانی ناایستا هستند. تولید سه ماهه بستنی آمریکا در شکل (۸–۱–ج) طرح خاص دیگری را نشان می‌دهد که به‌واسطه تغییرات فصلی، طبیعتی تکراری دارد. سریهای زمانی که تغییرات فصلی را دربرمی‌گیرند، سریهای زمانی فصلی نامیده می‌شوند. سریهای زمانی ناایستا را مانند آنهایی که در

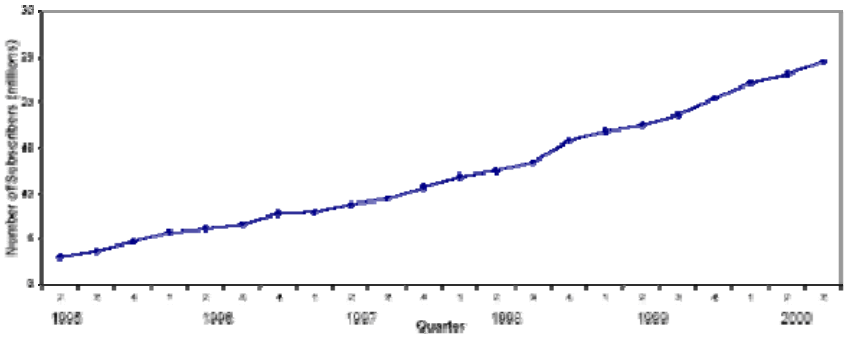
شکل‌های (۸– ۱– ب) و(۸–۱–ج) نشان داده شده‌اند، می‌توان با تبدیلات مناسبی به سری ایستا تبدیل نمود. سری زمانی چهارم که در شکل (۸–۱– د) نشان داده شده است، داده‌های مربوط به مگس گوشت است. این سری پدیده دیگری از ناایستایی به‌دلیل تغییر ساختاری ناشی از یک یا چند اغتشاش خارجی را منعکس می‌کند. این نوع ناایستایی را نمی‌توان با یک تبدیل استاندارد حذف نمود.

شناسایی، تجزیه و حذف اجزاء سریهای زمانی

برای تجزیه و تحلیل مجموعه‌ای از داده‌ها، در اولین مرحله لازم است که نمودار مشاهدات را نسبت به زمان رسم کنیم. این کار غالباً مهم‌ترین خواص یک سری زمانی مانند روند، فصلی بودن و مشاهدات دورافتاده را آشکار می‌کند. روشهای متعددی برای شناسایی، تعیین و یا حذف برخی از اجزاء سری‌های زمانی وجود دارد. با توجه به اینکه مهم‌ترین اجزاء یک سری زمانی، جزء روند و فصلی می‌باشد، در ادامه به بررسی روشهای شناسایی، هموارسازی، تجزیه و یا حذف این اجزاء پرداخته می‌شود. [۲]

سریهای زمانی با روند خطی

اگر جزء روند در یک سری زمانی یک حالت افزایشی یا کاهشی مستقیم داشته باشد، می‌توان معادله روند را به شکل $y_t = a + bt + e_t$ نشان داد که e_t جزء تصادفی در این رابطه است. شکل (۸–۲) معادله یک روند خطی را نشان می‌دهد.



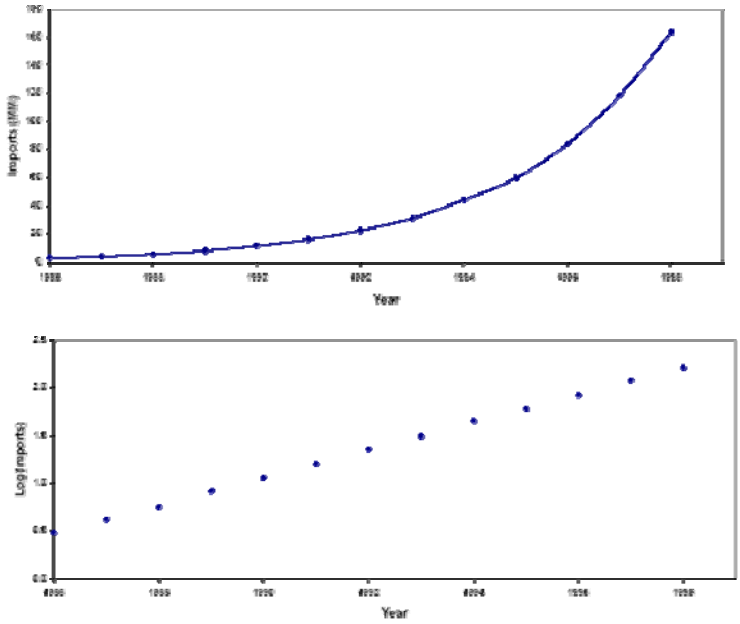
شکل ۸–۲) سری زمانی با روند خطی

سریهای زمانی با روند غیرخطی

گاهی اوقات می‌توان با در نظر گرفتن تبدیلات خاصی، مانند لگاریتم یا ریشه دوم داده‌ها، روند غیرخطی سری‌ها را به یک روند خطی تبدیل کرد. درصورتی‌که انحراف معیار با میانگین نسبت مستقیم داشته باشد، یک تبدیل لگاریتمی مطابق روابط (۸–۳) و (۸–٤) مناسب است.

$$log(y_t) = a + b + e_t \qquad (۸–۳)$$

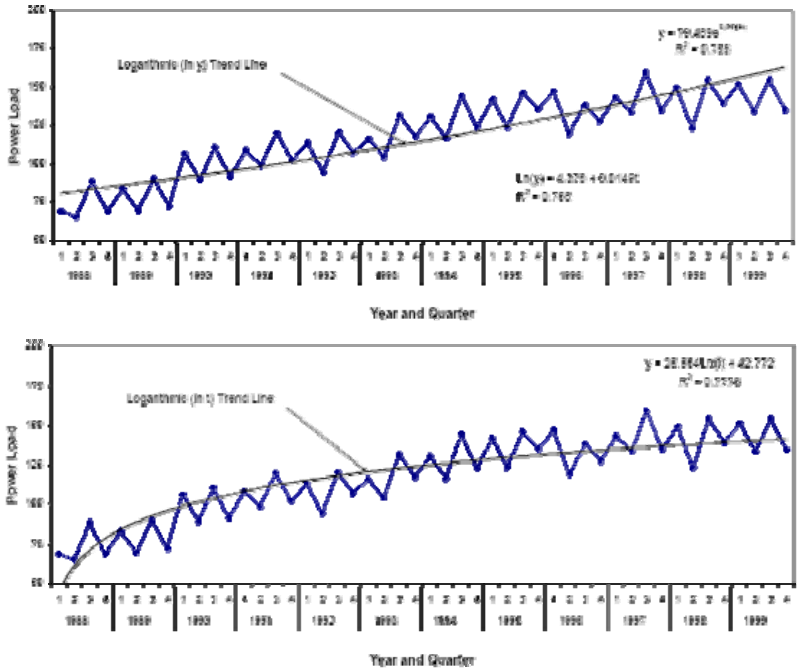
$$y_t = \exp(a + bt + e_t) \qquad (۸–٤)$$



شکل ۸–۳) سری زمانی به ترتیب با روند غیرخطی و تبدیل یافته آن

اگر یک سری زمانی با یک نرخ کاهشی بر حسب زمان ظاهرشود، ممکن است رابطه (۸–۵) مناسب باشد.

$$y_t = a + bln(t) + e_t \qquad (۸–۵)$$



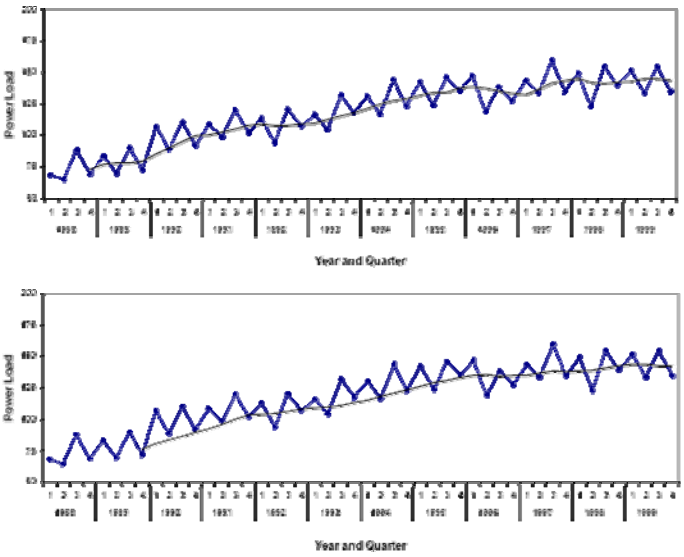
شکل ۸–٤) سری زمانی به ترتیب با روند غیرخطی و تبدیل لگاریتمی

میانگین متحرک

یک روش دیگر برای ارزیابی روند در سری‌های زمانی محاسبه میانگین m مشاهده اخیر می‌باشد. این روش به میانگین متحرک موسوم است و مطابق رابطه (۸–٦) محاسبه می‌گردد.

$$\bar{y}_{ma(٤)} = \frac{(y_t + y_{t-١} + y_{t-٢} + y_{t-٣})}{٤} \qquad (٦–٨)$$

میانگین متحرک عمدتاً نوسانات موجود در داده‌ها را همواره می‌کند. این روش معمولاً زمانی به‌خوبی عمل می‌کند که داده‌ها یک روند خطی و الگوی منظمی از نوسانات داشته باشند.



شکل ۸-۵) روش میانگین متحرک درسری زمانی به ترتیب چهار نقطه و هشت نقطه

هموارسازی نمایی

مهم‌ترین عیب روش میانگین متحرک این است که وزن یا اهمیت داده‌های گذشته، به صورت یکسان در نظر گرفته می‌شود. برای برطرف کردن این اشکال از یک روش جامع‌تر که یک روش میانگین متحرک موزون است و در آن تخصیص وزن به دوره‌های گذشته به‌صورت یک تصاعد هندسی می‌باشد، استفاده می‌کنیم. این روش هموارسازی نمایی نام دارد و به مقادیر گذشته سری زمانی تا به حال، وزن داده می‌شود و البته وزن بیشتری برای داده‌های جدید در نظر گرفته شده و هرچه به سمت داده‌های قدیمی پیش می‌رویم، وزن آنها طی یک فرآیند نمایی، کاهش می‌یابد.

$$S_1 = Y_1$$

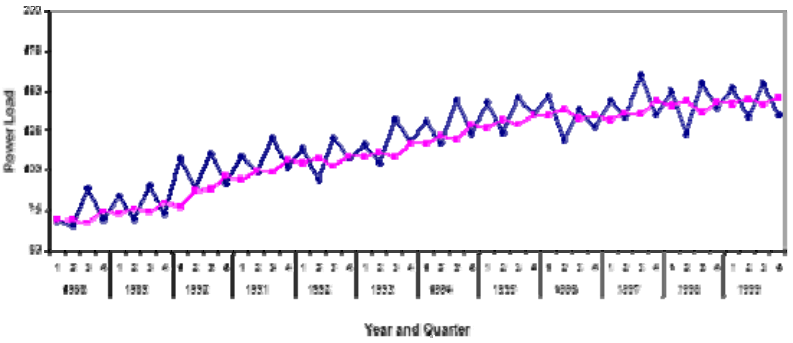
$$S_t = wY_t + (1-w)S_{t-1}$$
$$= wY_t + w(1-w)Y_{t-1} + w(1-w)^2 Y_{t-2} + \cdots$$

Let w=0.5

$$S_1 = Y_1$$
$$S_2 = 0.5Y_2 + (1-0.5)S_1 = 0.5Y_2 + 0.5Y_1$$
$$S_3 = 0.5Y_3 + (1-0.5)S_2 = 0.5Y_3 + 0.25Y_2 + 0.25Y_1$$
$$S_4 = 0.5Y_4 + (1-0.5)S_3 = 0.5Y_4 + 0.25Y_3 + 0.125Y_2 + 0.125Y_1$$

شکل ۸-۶) هموارسازی

هرچه مقدار ثابت هموارسازی (w) بیشتر باشد نشان دهنده آن است که داده‌های قدیمی اثر کمتری بر روی هموارسازی پیش‌بینی دارند و هر چه مقدار w کمتر باشد داده‌های سری زمانی هموارتر خواهند بود.

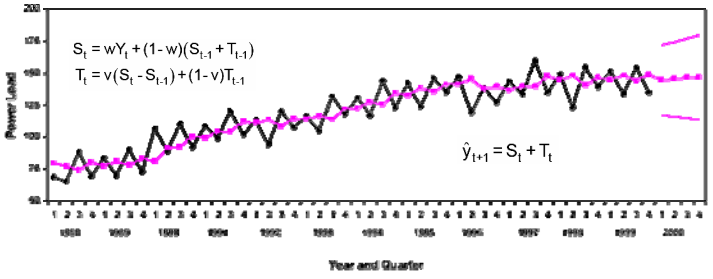


شکل ۸-۷) تحلیل هموارسازی

انتخاب مقدار w را می‌توان براساس حداقل کردن شاخصهای میانگین قدر مطلق خطا و میانگین خطای مربع شده که دربحث پیش‌بینی به آنها اشاره شد، انجام داد. در مثال بالا $W = ۰/۳٤$، براساس حداقل کردن مقدارمیانگین خطای مربع شده به‌دست آمده است.

هموارسازی نمایی با تنظیم روند

مانند روش میانگین متحرک، هموارسازی نمایی ساده، نسبت به روند واکنش نشان نمی‌دهد. در این حالت استفاده از روشهای دیگر نظیر هموارسازی نمایی با تنظیم روند مناسب است.



شکل ۸-۸) هموارسازی نمایی

هموارسازی نمایی با تنظیم اثر روند و فصلی

همان‌طورکه گفته شد، در تجزیه و تحلیل سریهای زمانی گاهی اوقات لازم است اثرات روند فصلی یا سیکلی را به‌طور مجزا مطالعه کنیم. برای حذف کردن و یا اندازه‌گیری هریک از این مؤلفه‌ها، در تحلیل سریهای زمانی از روشهای مربوطه استفاده می‌شود. روشهای میانگین متحرک و غیره که قبلاً به آنها اشاره شد، برخی از این روشها می‌باشند. یکی دیگر از روشهای مورد استفاده برای حذف جزء روند یا فصلی، استفاده از روش تفاضلی است. مثلاً برای داده‌های غیر فصلی، تفاضل مرتبه اول طبق رابطه (۸-۷) برای رسیدن به ایستایی کافی است.

$$y_t = x_{t+1} - x_t \qquad (۸-۷)$$

همچنین برای برآورد اثر فصلی برحسب اینکه اثر فصلی جمعی یا ضربی باشد، می‌توان با استفاده از روابط (۸-۸) تا (۸-۹) آن را برآورد کرد.

$$x_t - S_m(x_t) \qquad (۸-۸)$$

$$\frac{x_t}{S_m(x_t)} \qquad (۸-۹)$$

$$S_m(x_t) = \frac{\frac{1}{2}x_{t-6} + x_{t-5} + \ldots + x_{t+5} + \frac{1}{2}x_{t+6}}{12} \qquad (۸-۱۰)$$

شاخص داربین

همچنین یکی از راههای حذف کردن اثر فصلی، استفاده از شیوه تفاضلی است. [۲]

جستجوی تشابه در تحلیل سریهای زمانی

مطابق آنچه که در ابتدای فصل اشاره شد، مباحث مدلسازی و پیش‌بینی سریهای زمانی در بسیاری از مراجع آماری مورد بررسی قرارگرفته است. لیکن آماردانها روشهای مناسبی را برای تشابه و شاخص‌گذاری سریهای زمانی بررسی نکرده‌اند. تعداد زیادی از این مسائل توسط جامعه علمی کامپیوتر حل شده‌اند. یکی از مسائل جالب در داده‌های سری زمانی، یافتن سریهای زمانی متفاوتی است که رفتار مشابه داشته باشند. مسئله را

می‌توان این‌گونه نیز مطرح کرد که آیا دو سری زمانی X و Y داده شده مشابه هستند یا خیر؟ به‌عبارت دیگر، تابع $Sim(X,Y)$ را تعریف کرده و میزان تشابه دو سری یا به‌طور مشابه، تابع فاصله $Dis(X,Y)$ را محاسبه می‌کنیم. برای نمونه، هر سری زمانی مقدار و سیر تدریجی یک شیء را به‌صورت تابعی از زمان در مجموعه‌ای از داده‌های جمع‌آوری شده توصیف می‌کند (مثلاً قیمت سهام). هدف می‌تواند خوشه‌بندی اشیاء مختلف در گروه‌های مشابه (مانند گروهی از سهامها که تغییر قیمت یکسان داشته‌اند) یا دسته‌بندی اشیاء براساس مجموعه‌ای از ویژگیهای شناخته شده باشد. این مورد مشکل است، چون مدل تشابه باید اجازه تطابق غیردقیق را بدهد.

یکی از مسائل جالب مطرح شده در تشابه دنباله‌ها[1]، تشابه زیر دنباله‌ها می‌باشد که در آن برای یک سری زمانی داده شده X و الگوی سری زمانی کوتاهتر Y، می‌خواهیم زیر دنباله‌ای از X را که مشابه الگوی Y عمل می‌کند پیدا کنیم. برای پاسخ به این پرسش، نظریه‌های متفاوتی از تشابه سریهای زمانی در پژوهش‌های داده‌کاوی مطرح شده است. در این بخش مدلهای مختلف اندازه‌گیری تشابه سریهای زمانی مطرح می‌شود که براساس شاخصهای کارآیی و دقت، می‌توان آنها را ارزیابی کرد. نمونه‌هایی از اندازه‌گیری تشابه را که بر اساس نرم اقلیدسی، تخمینهای خطی قطعه‌ای[2]، تاباندن زمانی پویا[3] *(DTW)* و بزرگترین زیردنباله‌های مشترک[4] *(LCSS)* هستند را بررسی خواهیم کرد. [۵]

مقیاسهای اندازه‌گیری تشابه در سریهای زمانی

فاصله‌اقلیدسی و نرم[5] L_p

[1]- Sequences
[2]- Piecewise Linear Approximations: PLA
[3]- Dynamic Time Warping: DTW
[4]- Longest Common Subsequences Similarity: LCSS
[5]- Euclidean Distances and Lp Norms

یکی از ساده‌ترین راههای اندازه‌گیری تشابه در سریهای زمانی اندازه‌گیری فاصله اقلیدسی است. دو دنباله زمانی با طول یکسان n را فرض کنید. ما هر دنباله را در فضای n بعدی اقلیدسی، به‌عنوان یک نقطه می‌بینیم. عدم شباهت یا فاصله بین دو دنباله X و Y را با $L_p(X,Y)$ تعریف می‌کنیم (وقتی $p=۲$ است، این فاصله، همان فاصله اقلیدسی معروف است). این اندازه‌گیری مزایای مختلفی دارد. فهم آن آسان، محاسبه آن ساده و برای حل مشکلات دیگر مثل شاخص‌گذاری وخوشه‌بندی سریهای زمانی قابل استفاده است. هرچند معایب زیادی نیز دارد که آن را برای کاربردهای متعددی نامناسب می‌کند. یکی از اصلی‌ترین معایب آن این است که اجازه نمی‌دهد که دنباله-های زمانی خط مبنای متفاوتی داشته باشند. برای مثال سهام X با نوسان حدود ۱۰۰$ و Y با نوسان حدود ۳۰$ را در نظر بگیرید. حتی اگر شکل هر دو دنباله زمانی به هم خیلی شبیه باشد، ممکن است فاصله اقلیدسی بین آنها خیلی زیاد شود. همچنین نمی‌توان به این روش در مقیاسهای مختلف اندازه‌گیری نمود. برای مثال ممکن است سهام X در دامنه کوچکی نوسان کند (بین ۹۵$ تا ۱۰۵$) درحالی که سهام Y در دامنه بزرگتری نوسان کند (بین ۲۰$ تا ٤۰$).[۵]

تبدیلات نرمال[1]

با استفاده از نرمال‌کردن دنباله‌ها می‌توان معایب نرم L_p را در اندازه‌گیری تشابه برطرف کرد. در رابطه (۸-۱۰) اگر $\mu(X)$ میانگین و $\sigma(X)$ واریانس دنباله باشد، دنباله $X=\{x_1,...,x_n\}$ را با دنباله نرمال X' جایگزین می‌کنیم:

$$x'_i = (x_i - \mu(X))/\sigma(X) \qquad (۸-۱۱)$$

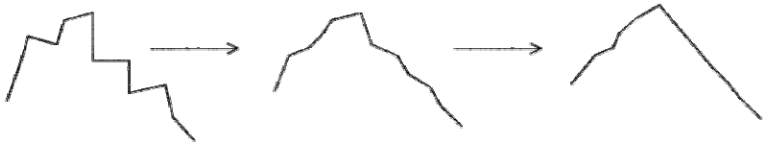
[1]- Normalization Transformations

همچنین دنباله Y را با دنباله نرمال Y' جایگزین می‌کنیم. در نهایت عدم تشابه بین X و Y را با $L_p(X',Y')$ تعریف می‌کنیم. این تعریف تشابه، معایب استفاده مستقیم از نرم L_p را که دنباله‌ها نرمال نیستند، حل می‌کند. برای مثال دو سهام X و Y که قبلاً در مورد آن صحبت شد، را در نظر بگیرید. بعد از نرمال شدن هر دو دارای یک خط مبنا شده (چون میانگین هر دو با نرمال کردن یکسان می‌شود) و دامنه یکسانی خواهند داشت (چون با نرمال کردن واریانس داده‌ها یکسان شده است).

فرآیند نرمال کردن هم معایب خاص خود را دارد. مثلاً به تأخیر و تقدم فاز در زمان خیلی حساس است. به‌عنوان نمونه دو دنباله X و Y را در نظر بگیرید که X شبیه به موج سینوسی است، در صورتی‌که Y شبیه به موج کسینوسی است. هر دو اصولاً دارای شکل یکسانی هستند به‌جز اینکه یک تأخیر فاز دارند. ولی فاصله بین این دو دنباله قابل ملاحظه است (در هردو حالت نرمال شده یا نرمال نشده). همچنین فاصله‌اقلیدسی، شتاب وکاهش شتاب در طول محور اصلی را نشان نمی‌دهد. برای مثال دو دنباله X و Y که شبیه موج سینوسی هستند را در نظر بگیرید که فقط پریود X دو برابر دوره Y است. حتی اگر این دو دنباله نرمال شوند، فاصله‌اقلیدسی نمی‌تواند شباهت بین این دو سیگنال را نشان دهد [۵].

تبدیلات عمومی[1]

تشخیص اهمیت نظریه حالت، در محاسبات تشابه در سال ۱۹۹۵ مطرح شد. یک چارچوب تشابه عمومی شامل زبان قواعد تبدیلات توصیف شده است. هر روش در زبان تبدیلات، یک دنباله ورودی را گرفته و با هزینه مربوط به آن روش، دنباله خروجی متناظرش را تولید می‌کند. تشابه بین دنباله X و Y، حداقل هزینه برای تغییر X به Y با استفاده از چنین روشهایی می‌باشد. برای مثال در شکل زیر دنباله‌هایی که به شکل منحنی‌های خطی-قطعه‌وار هستند، نشان داده شده است. به‌عنوان نمونه، یک قاعده تبدیل می‌تواند یکی کردن بخش‌های مجاور روی یک بخش باشد. هزینه این قاعده می‌تواند تابعی از طول و شیب بخش جدید و بخش ابتدایی باشد. قاعده دیگر می‌تواند یک بخش منفرد را با یک جفت بخش مجاور جایگزین کند.



شکل ۸-۹) دنباله‌هایی به شکل منحنی‌های خطی

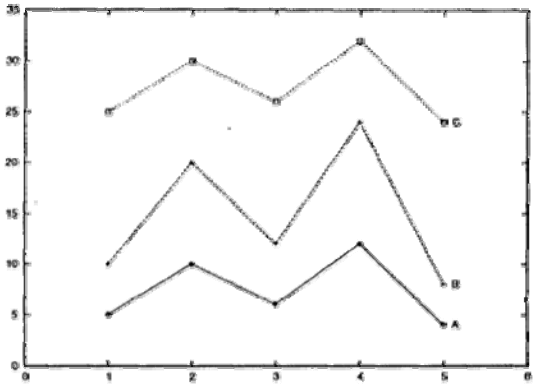
یکی از قواعد تبدیلات استفاده از میانگین متحرک برای هموارکردن سریهای زمانی است. برای مثال سری زمانی X که قیمت روزانه سهام در طول یک فاصله زمانی چند روزه است را در نظر می‌گیریم. برای هر ۳ روز یک میانگین متحرک حساب کرده و در دنباله X تغییر می‌دهیم و نتیجه دنباله X' می‌شود که $x' = (x_{i-1} + x_i + x_{i+1})/3$.

یکی دیگر از روشهای تبدیل، تبدیل مقیاس و انتقال است. در تبدیل مقیاس، هر جزء با مقیاس ثابتی افزایش می‌یابد. برای مثال هر x_i با cx_i جایگزین می‌شود که c مقداری ثابت است. هر تبدیل انتقال هر جزء را با یک عدد ثابت از موقعیت فعلی به سمت

[1]- General Transformations

راست یا چپ منتقل می‌کند (هر x_i با x_i+c جایگزین می‌شود که c یک عدد صحیح ثابت است).

مثال: سه دنباله شکل زیر را در نظر بگیرید.



شکل ۸-۱۰) سه دنباله نمونه

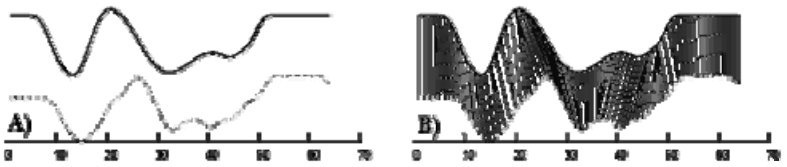
$A = (۵.۱۰, ٦, ۱۲, ٤)$
$B = (۱۰, ۲۰, ۱۲, ۲٤, ۸)$
$C = (۲۵, ۳۰, ۲٦, ۳۲, ۲٤)$

این سه سری زمانی متفاوتند، اما باهم رابطه نزدیکی دارند. سری A می‌تواند با دو برابر کردن جملات به B تبدیل شود و C با ۲۰ واحد انتقال می‌تواند به A تبدیل شود. به علاوه B می‌تواند با نصف کردن جملاتش و سپس ۲۰ واحد انتقال تبدیل به C شود. این بدین معنی است که این سریها با تبدیل مقیاس‌بندی و انتقال مناسب، در واقع یکی هستند. اگر سه دنباله بالا را به‌عنوان روند قیمت سه سهام در نظر بگیریم، با وجود اینکه قیمت سهام شرکت C بیشتر از شرکت A است، اما چون نوسان یکسانی دارند دقیقاً از روند قیمتی یکسانی پیروی می‌کنند. یا اگر چه قیمت سهام شرکت B همیشه دو برابر قیمت سهام شرکت A ولی نوسان آنها متناسب با قیمتشان است در نتیجه روند قیمتی آنها باید یکسان در نظرگرفته شود. سری A با سری B مشابه است، اگر A بتواند با یکی از تبدیلات گفته شده به B تبدیل شود [۷].

قواعد تبدیل، یک روش عمومی را برای تعریف تشابه که مناسب کاربردهای خاصی است، پیشنهاد می‌دهد. هرچند بعضی از معایب آنها، مشکلاتی نیز ایجاد می‌کنند. مثلاً محاسبات زیر دنباله‌ها (مانند شاخص‌گذاری) مشکل می‌شود، چون استخراج مشخصه‌ها از دنباله X، مخصوصاً اگر قواعد استفاده شده به هر دو دنباله X و Y بستگی داشته باشد، کار پیچیده‌ای است. همچنین فاصله‌اقلیدسی در فضای مشخصه ممکن است تقریب خوبی برای عدم تشابه دنباله‌های ابتدایی نباشد [۵].

تاباندن محور زمان به‌صورت پویا[1]

یکی از معمول‌ترین کارهایی که با داده‌های سریهای زمانی انجام می‌دهند، مقایسه یک دنباله با دنباله دیگر است. در بعضی از کاربردها یک مقیاس اندازه‌گیری ساده مانند اندازه‌گیری فاصله‌اقلیدسی، کافی است. اگرچه اغلب این حالت پیش می‌آید که دو دنباله با اجزای تقریباً یکسان در محور xها، در قسمتی نسبت به هم کشیده‌تر هستند. شکل زیر این موضوع را با یک مثال ساده نشان می‌دهد. برای فهمیدن شباهت چنین دنباله‌هایی، قبل از میانگین گرفتن از آنها، به‌عنوان یک قدم پیش‌پردازش، یک یا هر دو دنباله را روی محور زمان می‌پیچانیم. روش DTW برای این نوع تاباندن روی محور زمان، کارآمد می‌باشد. علاوه بر داده‌کاوی، DTW در تشخیص حرکات علم روباتیک، تحلیل سخنرانی و پزشکی کاربرد دارد.



شکل ۸-۱۱) دو دنباله که وضعیت دستخط یک شخص را هنگام نوشتن کلمه Pen در زبان علامت روی محورها نشان می‌دهد.

[1]- Dynamic Time Warping

شکل (۸-۱۱) یک مثال از کاربرد *DTW* می‌باشد. نمودار *A* شامل دو دنباله است که وضعیت دستخط یک شخص را هنگام نوشتن کلمه *Pen* در زبان علامت روی محورها نشان می‌دهد. دنباله‌ها در دو روز مختلف ضبط شده‌اند. توجه کنید با وجود اینکه دنباله‌ها شکل عمومی یکسانی دارند، ولی در محور زمان بر هم منطبق نیستند. یک مقیاس فاصله که فرض می‌کند *i*امین نقطه روی یک دنباله منطبق بر *i*امین نقطه روی دنباله دیگر است، باعث یک عدم تشابه ناامید کننده می‌شود. نمودار *B* می‌تواند به صورت کارآمد یک تطابق بین دو دنباله ایجاد کند که محاسبه فاصله پیچیده‌تری را ایجاب می‌کند.

اگر چه استفاده از *DTW* در بسیاری از زمینه‌ها موفق بوده است، می‌تواند نتایج غیر قابل کنترل و نامطلوبی داشته باشد. مشاهدات حاکی از آن است که الگوریتم می‌تواند تغییرات در محور *y*ها را با تاباندن روی محور *x* ها توضیح دهد. این می‌تواند باعث تطابق‌های غیرشهودی هنگام تصویر یک نقطه منفرد از یک سری زمانی، روی یک زیربخش بزرگ از سری زمانی دیگر شود. ما چنین رفتار غیردلخواهی را مقادیر منفرد یا تکینها[1] می‌نامیم. بخش وسیعی از روشها برای مقابله با این رفتارها ارائه شده‌اند. دستاورد این روشها، چگونگی تاباندن مجاز را مشخص می‌کنند. اگر چه استفاده از این روشها در بعضی مواقع از یافتن روش تاباندن صحیح جلوگیری می‌کند.

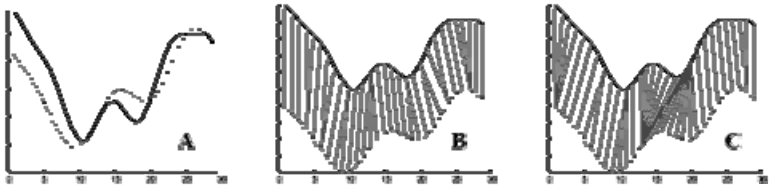
در موارد شبیه‌سازی شده، تاباندن وقتی صحیح تشخیص داده می‌شود که ما ابتدا یک سری زمانی را بتابانیم و سپس سعی کنیم سری اصلی را از روی سری تابانده شده به‌دست آوریم. در رویدادهای طبیعی، منظور ما از روش تاباندن صحیح آن است که مانند شکل (۸-۱۲- *B*) به‌طور شهودی تطابق یک به یک مشخصه‌ها واضح باشد.

یک مشکل دیگر با *DTW* آن است که الگوریتم ممکن است در پیدا کردن تطابق‌های طبیعی و واضح در دو دنباله، تنها به این دلیل که یک مشخصه در یک دنباله کمی بالاتر

---

[1]- Singularities

یا پایین‌تر از مشخصه مربوطه در دنباله دیگر است، اشتباه کند. برای مثال نقاط اوج یا حضیض، نقطه خمیدگی، قسمت مسطح و غیره شکل (۸-۱۲) این مسئله را نشان می‌دهد.



شکل ۸-۱۲- A) دو سیگنال ترکیبی (با میانگین وواریانس یکسان). B) تطابق نظیر به نظیر مشخصه‌ها. C) تطابق ایجاد شده به‌وسیله DTW

توجه کنید که *DTW*، دو نقطه اوج مرکزی را به‌علت اینکه آنها در محور Yها کمی با هم فاصله دارند، منطبق در نظر گرفته است [۸].

الگوریتم کلاسیک *DTW*

فرض کنید دو سری زمانی Q و C را با طول‌های m و n داریم[۸] :

$$Q = q_1, q_2, \ldots, q_i, \ldots, q_n$$
$$C = c_1, c_2, \ldots, c_j, \ldots, c_m \qquad (۸-۱۲)$$

برای تطابق دو دنباله با استفاده از *DTW* یک ماتریس $m*n$ می‌سازیم به‌طوری‌که عنصر *(ith, jth)* ماتریس، شامل فاصله $d(q_i, c_j)$ بین دو نقطه c_j و q_i می‌باشد (معمولاً از فاصله اقلیدسی استفاده می‌کنیم)، بنابراین $d(q_i, c_j) = (q_i - c_j)^2$. هر عنصر *(i,j)* ماتریس به تطابق بین نقاط q_i و c_j مربوط است که در شکل (۸-۱۳) نشان داده شده است. یک مسیر تاباندن W، یک دنباله از عناصر پیوسته ماتریس است که یک نگاشت را بین Q و C مشخص می‌کند. عنصر k ام W به‌صورت $w_k = (i,j)$ تعریف می‌شود و در نتیجه خواهیم داشت.

$$W = w_1, w_2, \ldots, w_k, \ldots, w_k \qquad \max(m,n) \le K < m + n - 1 \qquad (۸-۱۳)$$

مسیر تاباندن معمولاً محدودیتهایی دارد.

شرایط حدی: $w_1 = (1,1)$ و $w_k = (m,n)$ به‌راحتی نشان می‌دهند که مسیر تاباندن از اولین نقطه روی قطر اصلی شروع و به نقطه مقابل آن در انتهای قطر اصلی خاتمه می‌یابد.

پیوستگی: اگر $w_k = (a,b)$ و $w_{k-1} = (a',b')$ باشند، لازم است که $b - b' \le 1$ و $a - a' \le 1$. این شروط گامهای مجاز را در مسیر تاباندن برای سلولهای مجاور از جمله سلولهای مجاور قطری مشخص می‌کند (هیچ جزئی نمی‌تواند در دنباله حذف شود).

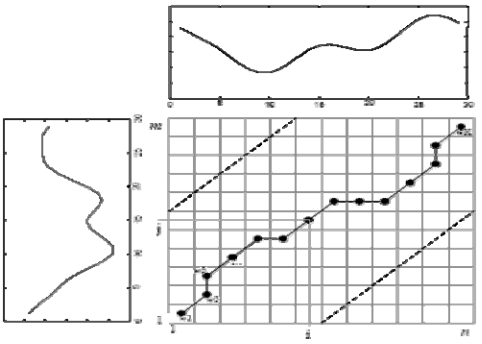
یکنواختی: اگر $w_k = (a,b)$ و $w_{k-1} = (a',b')$ لازم است که $a - a' \ge 0$ و $b - b' \ge 0$. این باعث می‌شود نقاط در W به‌صورت یکنواخت و بر حسب زمان قرار بگیرند.

باتوجه به محدودیتهای گفته شده تعداد زیادی مسیر تاباندن وجود دارد ولی تنها مسیرهایی که هزینه تاباندن را حداقل می‌کنند مورد نظر ما هستند.

$$DTW(Q,C) = \min\left\{\sqrt{\sum_{k=1}^{K} w_k} / K \right. \qquad (۸-۱۴)$$

برای نشان دادن اینکه مسیرهای مختلف می‌توانند طول‌های مختلف داشته باشند، K در رابطه (۸-۱۲) در مخرج ظاهر می‌شود. این مسیر را می‌توان به‌صورت کارآمد با استفاده از برنامه‌ریزی پویا به‌دست آورد. برای این کار $\gamma(i,j)$ را که فاصله تجمعی است به‌دست می‌آوریم. $d(i,j)$ در فرمول (۸-۱۳) همان فاصله i و jای است که در ماتریس به‌دست آوردیم و قسمت بعد حداقل مقدار فاصله سلولهای مجاور است.

$$\gamma(i,j) = d(q_i,c_j) + \min\{\gamma(i-1,j-1), \gamma(i-1,j), \gamma(i.j-1)\} \qquad (۸-۱۵)$$



شکل ۸-۱۳) مثالی از یک مسیر تاباندن

محدودیت‌های الگوریتم کلاسیک *DTW*

مشکل مقادیر تکین از اوایل سال ۱۹۷۸ توسط ساکوئی و چیبا[1] مورد توجه قرارگرفت. روشهای مختلفی برای کم‌رنگ‌تر کردن این مشکل مطرح شده است که ما اجمالاً به بررسی آنها می‌پردازیم [۸].

۱) پنجره بندی[2]: عناصر مجاز ماتریس به آنهایی که در پنجره می‌افتند، محدود می‌شوند $|i-(n/(m/j))|<R$ که R یک عدد صحیح مثبت است و نمایانگر عرض پنجره است. این بدین معنی است که گوشه‌های ماتریس هرس می‌شوند. همان‌طورکه در شکل (۸-۱۳) با خط‌چین نشان داده شده است. دیگران پنجره‌بندی را با اشکال مختلف دیگری تجربه کردند این دستاورد تا جای ممکن مشکل تکینها را محدود می‌کند اما از رخداد آن جلوگیری نمی‌کند.

۲)وزن‌دهی به شیب[3]: اگر در معادله (۸-۱۵) فاصله را به‌صورت زیر در نظر بگیریم:

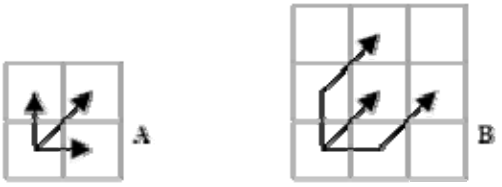
$$\gamma(i,j)=d(i,j)+min\{\gamma(i-۱,j-۱),X\gamma(i-۱,j),X\gamma(i,j-۱)\} \qquad (۱٦-۸)$$

---

[1]- Sakoe & Chiba
[2]- Windowing
[3]- Slope Weighting

به‌طوری‌که X یک عدد حقیقی مثبت است و می‌توانیم شکل تاب دادن را با عوض‌کردن مقدار X محدود کنیم. هرچه مقدار X بزرگ‌تر شود مسیرتاب دادن به سمت قطری شدن تمایل پیدا می‌کند.

۳) محدودیت‌های شیب[1]: ما می‌توانیم معادله فاصله را به‌صورت یک نمودار الگوی گامهای قابل قبول به‌تصویر درآوریم. مثلاً در شکل (۸-۱٤-A) پیکانها نشان‌دهنده گامهای مجازی هستند که مسیر تاباندن در هر مرحله می‌تواند بردار را نشان دهد. می‌توانستیم معادله (۸-۱۵) را با معادله‌های زیر که به الگوی گام نشان داده شده در شکل (۸-۱٤-B) مربوط است جایگزین کنیم با استفاده از این معادله، مسیر تاباندن یک قدم قطری، گام برمی‌دارد که این قدم می‌تواند بعد از طی یک قدم موازی با یکی از محورها صورت گیرد.



شکل ۸-۱٤) یک نمایش تصویری از رو نوع الگوی مسیر متفاوت

$$\gamma(i,j) = d(i,j) + \min(\gamma(\bar{i}-1,\bar{j}),\gamma(i-1,j),\gamma(i.j-1)) \quad \text{الگوی } (A)$$

$$\gamma(i,j) = d(i,j) + \min[\gamma(i-1,j-1),\gamma(i-1,j-2),\gamma(i-2,j-1) \quad \text{الگوی } (B)$$

همه موارد گفته شده می‌توانند در محدود کردن مشکل تکینها کمک کنند ولی ریسک از دست دادن روش تاب‌دادن صحیح، هنوز وجود دارد. مشکل دیگر آن است که چگونگی انتخاب پارامترهای موجود در الگوها هنوز برای ما واضح نیست. برای مثال نمی‌دانیم عدد R را در پنجره‌بندی و عدد صحیح X را در الگوی وزن‌دهی شیب چگونه به‌دست آوریم.

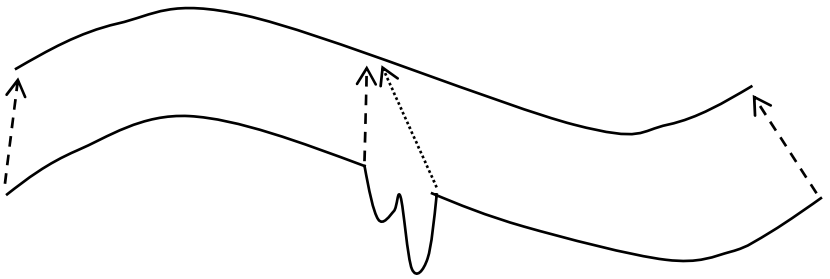
---

[1]- Slope Constraints

شباهت بزرگترین زیردنباله مشترک (LCSS)

شباهت بزرگترین زیردنباله مشترک برای اندازه‌گیری اختلاف سریهای زمانی، در مواردی مانند تشخیص صدا و تطابق الگوی متن به‌کار می‌رود. ایده اصلی، تطبیق دو دنباله بر یکدیگر است، حتی اگر چند جزء آنها منطبق نباشد. روش *LCSS* دو مزیت دارد: (الف) بعضی از اجزا می‌توانند منطبق نباشد (مثل نقاط پرت)، ولی در فاصله اقلیدسی و *DTW* همه اجزاء حتی نقاط پرت باید منطبق باشد. (ب) همان‌طورکه در ادامه دیده خواهد شد اندازه‌گیری *LCSS*، کارآیی محاسبات تقریبی را بیشتر می‌کند. حال به بررسی یک مثال شهودی‌تر می‌پردازیم. دو دنباله X و Y را در نظر بگیرید:

$$X = \{۳,۲,۵,۷,٤,۸,۱۰,۷\} \quad , \quad Y = \{۲,۵,٤,۷,۳,۱۰,۸\}$$



شکل ۸-۱۵) اندازه‌گیری LCSS

شکل (۸-۱۵) ایده اصلی اندازه‌گیری *LCSS* را مشخص می‌کند. فرض کنید قسمتهای مشخصی از هر دنباله در فرآیند تطابق حذف شده باشد (مثل نقاط پرت یا داده‌های مغشوش). مقدار *LCSS* بین X و Y برابر {۲,۵,۷,۱۰} است.

دو دنباله X و Y، به ترتیب با طول m و n را در نظر می‌گیریم. مشابه همان کاری که در *DTW* انجام شد، یک رابطه بازگشتی برای طول *LCSS* دنباله X و Y به‌دست می‌آید. مقدار $L(i,j)$ نشان دهنده *LCSS* زیر دنباله $\{x_۱,...,x_i\}$ و $\{y_۱,...,y_j\}$ می‌باشد. $L(i,j)$ می‌تواند یک رابطه بازگشتی به‌صورت زیر باشد.

$$If\ x_i = y_i\ then\ L(i,j) = ۱ + L(i-۱,j-۱), else\ \ l(i,j) = max\{D(i-۱,j), D(i,j-۱)\}$$

ما عدم تشابه بین X و Y را با $LCSS(X,Y)=(m+n-۲l)/(m+n)$ تعریف می‌کنیم، که l طول $LCSS$ است. این کمیت، حداقل مقدارنرمال شده تعداد عناصری است که باید از X حذف و یا به X اضافه شوند تا X به Y تبدیل شود. مشابه DTW، اندازه‌گیری $LCSS$ به‌وسیله برنامه‌ریزی پویا درزمان $O(mn)$، می‌تواند محاسبه شود. اگر پنجره تطابق با طول w مشخص شده باشد که $|i-j|$ می‌تواند حداکثر w باشد، می‌توان پیچیدگی آن را به مقدار $O(wn)$ کاهش داد. الگوریتم یافتن طولانی‌ترین مسیر مشترک بین دو دنباله به‌صورت زیر است:

```
LCSS - LENGTH(X, Y)
 m = length [X], n = length [Y]
 for i = ۱ to m do c[i,۰] ← ۰
 for j = ۱ to n do c[۰, j] ← ۰
 for i = ۱ to m
   do for j = ۱ to n
     do if x_i = y_j
       then c[i, j] ← c[i −۱, j −۱] +۱ b[i, j] ="↖"
       else if c[i −۱, j] ≥ c[i, j −۱]
         then c[i, j] ← c[i −۱, j]  b[i, j]="↑"
         else c[i, j] ← c[i, j −۱]  b[i, j]="←"
```

در $LCSS$ نیاز به اینکه داده‌های متناظر در زیر دنباله‌های مشترک دقیقاً منطبق باشند، کار را کمی سخت می‌کند. این مشکل با استفاده از تلرانسهای (ε>۰) هنگام مقایسه عناصرقابل حل است. بنابراین، مطابق گفته بالا، دو جزء a وb (به ترتیب از دنباله‌های X و Y) مطابقند اگر: $a(۱-\varepsilon)<b<a(۱+\varepsilon)$

به جای تلرانس نسبی، ممکن است از تلرانس مطلق استفاده شود، یعنی b باید بین $a-\varepsilon$ و $a+\varepsilon$ باشد[۵]. برای آشنایی بیشتر با این الگوریتم، مثال زیر توضیح داده می‌شود.

مثال: فرض کنید دنباله‌های X وY به‌صورت زیر داده شده باشند. برای به‌دست آوردن $LCSS$ این دو دنباله، قدمهای زیر را دنبال می‌کنیم.

$X = ABCB$
$Y = BDCAB$

همه عناصر رشته‌ها را در ستونها و ردیفها می‌چینیم

	Y_j	B	D	C	A	B
X_i						
A						
B						
C						
B						

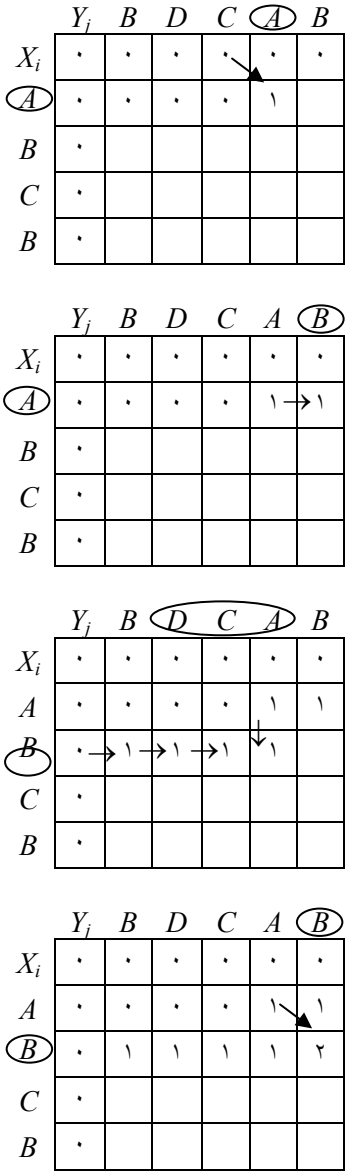
همه عناصر ستون و ردیف اول را صفر می‌کنیم

	Y_j	B	D	C	A	B
X_i	۰	۰	۰	۰	۰	۰
A	۰					
B	۰					
C	۰					
B	۰					

طبق قدمهای الگوریتم همه عناصر ستونها و ردیفها را با هم مقایسه کرده و عناصر متناظر در جدول را مقداردهی می‌کنیم اول را صفر می‌کنیم

j	۰	۱	۲	۳	٤	۵
	↓					
	Y_j	B	D	C	A	B
X_i →	۰	۰	۰	۰	۰	۰
A	۰					
B	۰					
C	۰					
B	۰					

	Y_j	B	D	C	A	B
X_i	۰	۰	۰	۰	۰	۰
A	۰	۰	۰	۰		
B	۰					
C	۰					
B	۰					

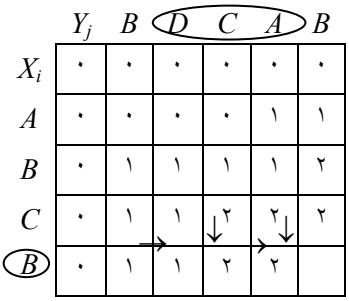

Yj B D C A B
Xi ۰ ۰ ۰ ۰ ۰ ۰
A ۰ ۰ ۰ ۰ ۱
B ۰
C ۰
B ۰
Yj B D C A B
Xi ۰ ۰ ۰ ۰ ۰ ۰
A ۰ ۰ ۰ ۰ ۱ ۱
B ۰
C ۰
B ۰
Yj B D C A B
Xi ۰ ۰ ۰ ۰ ۰ ۰
A ۰ ۰ ۰ ۰ ۱ ۱
B ۰ ۱ ۱ ۱ ۱
C ۰
B ۰
Yj B D C A B
Xi ۰ ۰ ۰ ۰ ۰ ۰
A ۰ ۰ ۰ ۰ ۱ ۱
B ۰ ۱ ۱ ۱ ۱ ۲
C ۰
B ۰

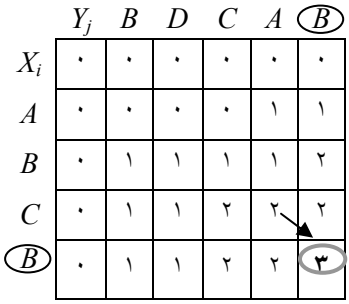
	Y_j	B	D	C	A	B
X_i	۰	۰	۰	۰	۰	۰
A	۰	۰	۰	۰	۱	۱
B	۰	۱	۱	۱	۱	۲
C	۰	۱	۱			
B	۰					

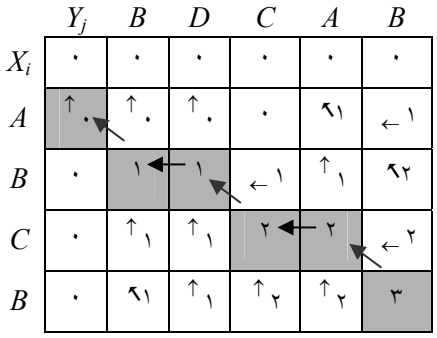
	Y_j	B	D	C	A	B
X_i	۰	۰	۰	۰	۰	۰
A	۰	۰	۰	۰	۱	۱
B	۰	۱	۱	۱	۱	۲
C	۰	۱	۱	۲		
B	۰					

	Y_j	B	D	C	A	B
X_i	۰	۰	۰	۰	۰	۰
A	۰	۰	۰	۰	۱	۱
B	۰	۱	۱	۱	۱	۲
C	۰	۱	۱	۲	۲	۲
B	۰					

	Y_j	B	D	C	A	B
X_i	۰	۰	۰	۰	۰	۰
A	۰	۰	۰	۰	۱	۱
B	۰	۱	۱	۱	۱	۲
C	۰	۱		۲	۲	۲
B	۰	۱				


Y_j B D C A B
X_i
A
B
C
B


Y_j B D C A B
X_i
A
B
C
B


Y_j B D C A B
X_i
A
B
C
B

i		Y_j	(B)	D	(C)	A	(B)
۰	X_i	۰	۰	۰	۰	۰	۰
۱	A	۰	۰	۰	۰	۱	۱
۲	(B)	۰	۱	۱	۱	۱	۲
۳	(C)	۰	۱	۱	۲	۲	۲
٤	(B)	۰	۱	۱	۲	۲	۳

LCS (reversed order): ***B C B***

LCS (straight order): ***B C B***

شکل ۸-۱٦) قدم‌های الگوریتم

خوشه‌بندی جریان کلیکها[1] با کمک بزرگترین دنباله‌های مشترک وزنی

گروه‌بندی بازدیدکنندگان براساس تعامل آنها با یک وب سایت، یک مشکل کلیدی در کاوشهای کاربردی وب است. جریان کلیکهای تولید شده به‌وسیله کاربرهای مختلف اغلب الگوهای مشخصی را دنبال می‌کنند. برای خوشه‌بندی کاربران وب، براساس جریان کلیک در یک وب سایت و زمان صرف شده روی هر صفحه، از یک الگوریتم *LCSS* استفاده شده است [۱۱].

با افزایش سریع کاربردهای تجارت الکترونیک، آگاهی از رفتار کاربر بر اساس تعاملاتش با یک وب سایت برای صاحبان وب سایت اهمیت بیشتری پیدا کرده است. تشخیص رفتار هر کاربر در بازدید از یک وب سایت می‌تواند مدیران سایت را برای تهیه محتوای سفارش‌شده برای دیگرکاربران قادر سازد. این موضوع کاربردهای زیادی در کسب وکار دارد. جریان کلیکهای یک کاربر، دنباله‌ای از صفحه‌های بازدید شده توسط او در یک وب سایت خاص در یک جلسه[2] می‌باشد. هدف، خوشه‌بندی کاربران

[1]- Clickstream
[2]- Session

بر اساس جریان کلیکها در یک وب سایت خاص و یافتن گروه‌هایی از کاربران است که با علایق و انگیزه‌های مشابه از یک سایت بازدید می‌کنند. در نتیجه همبستگی قوی بین جریان کلیکهای کاربران وجود دارد که نشان دهنده تشابه علایق کاربران است.

روشهای شاخص‌گذاری برای جستجوی تشابه در سریهای زمانی

مسئله دیگری که در بحثهای کاربردی سریهای زمانی مطرح است مسئله شاخص‌گذاری/بازیابی[1] است [۵]. مجموعه سریهای زمانی $S = \{Y_1,...,Y_N\}$ را در نظر گرفته وسری مورد نظر X را داریم. سریهای زمانی موجود در S را که بیشترین تشابه با سری X دارند، پیدا می‌کنیم. برای مثال، روزهایی از سال که یک سهام مشخص تغییرات مشابهی مانند امروز داشته باشد را جستجو می‌کنیم.

مسئله دیگر، شاخص‌گذاری زیر دنباله است. مجموعه دنباله‌های S و دنباله یا الگوی مورد نظر X داده شده است. دنباله‌هایی را در S می‌یابیم که شامل زیر دنباله‌های مشابه X باشد. برای حل کارآی این نوع مسائل، باید از روشهای مناسب شاخص‌گذاری استفاده کنیم.

مسئله تشابه، مرتبط با مسئله شاخص‌گذاری می‌باشد. معمولاً تعیین شاخص برای روشهای اندازه‌گیری ساده شباهت، آسان و احتمالاً کم‌دقت است. ولی اندازه‌گیری شباهتهای پیچیده، تعیین شاخص را دشوار و جالب می‌کند.

یک سری زمانی با طول n می‌تواند به‌عنوان یک مشاهده در فضای n بعدی درنظر گرفته شود. شاخص‌گذاری مستقیم در این فضا به‌علت ابعاد خیلی زیاد آن ناکارآ می‌باشد. راه حل، استفاده از روشهای کاهش بُعد است که در آن سری زمانی X با n مشاهده درنظرگرفته شده، چند مشخصه کلیدی آن استخراج و به نقطه $f(X)$ در فضای مشخصه با ابعاد کمتر k نگاشت می‌شود (امید داریم که $k<<n$ باشد). این نگاشت باید

---

[1]- Indexing/Retrieval

طوری انجام شود که تشابه یا فاصله بین X و Y تقریباً با فاصله اقلیدسی دو نقطه $f(X)$ و $f(Y)$ برابر باشد. می‌توان از روشهای شناخته‌شده دسترسی فاصله‌ای برای شاخص‌گذاری فضای مشخصه‌های دارای ابعاد کمتر استفاده کرد، مانند *R-trees*، *kd-trees* یا *VP-trees*.

در بسیاری از موارد سرعت دستیابی به سریهای زمانی مشابه با سری زمانی مورد جستجو، حائز اهمیت است. این مسئله در بسیاری از کاربردها دیده می‌شود. مثلاً یافتن سهامهایی که شبیه یک سهام خاص رفتار می‌کنند، پیدا کردن محصولاتی که چرخه تقاضای یکسانی دارند و پیدا کردن ژنهایی که الگوی مبین آنها شبیه ژن خاصی است. این کاربردها نیازمند مکانیزم بازیابی برای دسته‌بندی سریهای زمانی با استفاده از روش دسته‌بندی نزدیک‌ترین همسایگی جهت بهبود زمان اجرا در الگوریتمهای خوشه‌بندی یا برای تحلیل اکتساب در داده‌های سریهای زمانی هستند.

شاخص‌گذاری در سریهای زمانی، توجه بسیاری از محققان را در سالهای اخیر جلب کرده است. با توجه به گسترش روزافزون اندازه بانکهای اطلاعاتی، شاخص‌گذاری می‌تواند در داده‌کاوی بسیار مؤثر باشد. اگر کاربر بخواهد در بانکهای اطلاعاتی گسترده شروع به اکتشاف کند، باید داده‌ها به شکلی سازمان‌دهی شوند که او بتواند به شکلی مؤثر و کارا داده‌های مورد نظر خود را بازیابی کند. به‌طور عمومی می‌توان مسئله بازیابی سریهای زمانی را به‌صورت زیر تعریف کرد. بانک اطلاعاتی D شامل مجموعه‌ای از سریهای زمانی داده شده است. یک روش پیش‌پردازشی تعریف می‌کنیم که هدف آن پیدا کردن کارآی سری X عضو D، نزدیک به سری داده شده Q است، (سری Q لزوماً در بانک اطلاعاتی وجود ندارد). برای حل این مسئله باید موارد زیر را در نظر گرفت:

- یک تابع فاصله که با درک کاربر از آنچه شباهت نامیده می‌شود، مطابقت دارد.
- یک رویه مؤثر شاخص‌گذاری که سرعت جستجوی کاربر را بالا می‌برد.

درزیربخش قبلی روشهای مختلف برای تعریف شباهت (یا فاصله) بین دوسری زمانی بررسی شد. آسان‌ترین روش، تعریف فاصله بین دو سری با نگاشت هریک برروی یک بردار وسپس استفاده از نرم L_p برای محاسبه بود. فاصله نرم L_p بین دوبردار n بعدی $\bar{x}$ و $\bar{y}$ به‌صورت زیر تعریف می‌شود:

$$L_p(\bar{x},\bar{y}) = (\sum_{i=1}^{n}(x_i - y_j)^p)^{\frac{1}{p}} \qquad (۱۷-۸)$$

برای p=۲ این فرمول همان فاصله‌اقلیدسی مشهور و برای p=۱ فاصله مانهاتان است. اگر چه محاسبه چنین مقیاس فاصله‌ای آسان است، اما به کوچک‌ترین تغییرات در محور زمان بسیار حساس بوده و برای داده‌های مغشوش به‌خوبی عمل نمی‌کند. همان‌طور که گفته شد، انعطاف پذیرترین روشها برای تعریف تشابه در سریهای زمانی، روشهای DTW و $LCSS$ هستند.

صورت مسئله برای کاربران مختلف می‌تواند متفاوت باشد. مثلاً کاربران می‌توانند به دنبال تطابق کل دنباله و یا تنها به دنبال تطابق یک زیردنباله باشند. مقیاس‌های اندازه‌گیری تشابه یا فاصله، کاربردهای مختلفی دارند. همچنین کاربر ممکن است علاقه‌مند باشد که k تا از شبیه‌ترین سریهای زمانی که با فاصله ε ازسری مورد جستجو قرار دارند را بیابد. شاخص‌گذاری سریهای زمانی در دو حالت مورد بررسی قرار می‌گیرد که در حالت اول تابع فاصله متریک و در حالت بعدی تابع فاصله غیر متریک است. در ادامه به‌طور خلاصه به‌بررسی هر یک از این حالات پرداخته می‌شود.

شاخص‌گذاری سریهای زمانی با تابع فاصله متریک

شاخص‌گذاری علاوه بر اینکه موارد مشابه را درسازمان داده‌ها گردآوری می‌کند، امکان هرس داده‌های غیر مرتبط را نیز فراهم می‌نماید. هرس کردن کاملاً به متریک بودن تابع فاصله بستگی دارد.

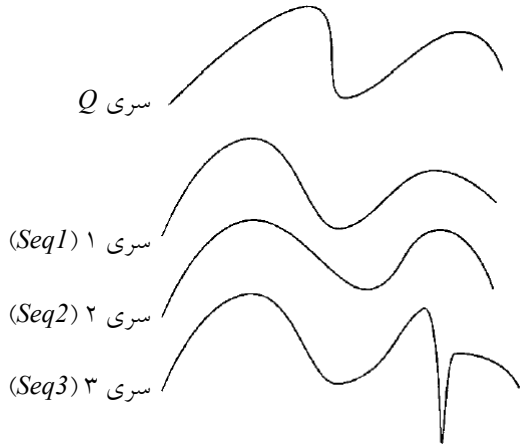
یک تابع فاصله $d(X,Y)$ بین دو شیء X و Y متریک است اگر دارای شرایط زیر باشد:

مثبت بودن $d(X,Y) \geq 0$ و $d(X,Y) = 0$ اگر $X=Y$

تقارن $d(X,Y) = d(Y, X)$

نامساوی مثلثی $d(X,Y) + d(Y,Z) \geq d(X,Z)$

فاصله‌اقلیدسی یک تابع فاصله متریک است. قدرت هرس کردن چنین تابعی در مثال زیر نشان داده شده است.



شکل ۸–۱۷) قدرت هرس نامساوی مثلثی

مثال: فرض کنید مجموعه دنباله‌های $S = \{Seq۱, Seq۲, Seq۳\}$ در شکل (۸–۱۷) داده شده است. همچنین فرض کنید فاصله سه دنباله را با روش فواصل جفتی محاسبه و در جدول (۸–۱) مرتب کرده‌ایم.

جدول ۸–۱) فواصل جفتی

	سری ۱	سری ۲	سری ۳
سری ۱	۰	۲۰	۱۱۰
سری ۲	۲۰	۰	۹۰
سری ۳	۱۱۰	۹۰	۰

برای پیدا کردن شبیه‌ترین دنباله به دنباله Q از بین سه دنباله فوق بهترین روش، تصویر دنباله‌هاست. در این روش فاصله همه آنها با Q محاسبه شده و یکی از آنها که

کوتاهترین فاصله را با Q دارد، انتخاب می‌شود. اگر تابع فاصله، نامساوی مثلثی را برآورده کند و داشته باشیم $D(Q,Seq۱)=۲۰$ و $D(Q,Seq۲)=۱۳۰$ آنگاه به این علت که:

$$D(Q,Seq۳)\geq D(Q,Seq۲)-D(Seq۲,Seq۳)\rightarrow D(Q,Seq۳)\geq ۱۳۰-۹۰=۴۰$$

ما می‌توانیم به‌راحتی Seq ۳ را حذف کنیم زیرا راه حل بهتری را پیشنهاد نمی‌کند. برای بهبود بیشتر باید از یک شاخص چند بعدی استفاده کنیم.

شاخصهای چند بعدی و پیدا کردن نزدیک‌ترین همسایگی‌ها به شکلی مؤثر در فضایی با ابعاد بالا، توجه بسیاری از متخصصین را در علوم کامپیوتری و تحقیقات بانکهای اطلاعاتی به خود جلب کرده است. یک روش ساده شاخص‌گذاری، درنظرگرفتن سری زمانی با طول n به‌عنوان یک نقطه n بعدی است. ما می‌توانیم هر سری زمانی را به‌عنوان یک نقطه در ساختار n بعدی *R-tree* ذخیره کنیم. برای پیدا کردن نزدیک‌ترین همسایگی، با تبدیل هر سری زمانی به یک نقطه n بعدی و استفاده از ساختار شاخص‌گذاری مانند استفاده از *R-tree* به جستجوی نزدیک‌ترین همسایگی می‌پردازیم. متأسفانه این ایده در عمل به‌خوبی کار نمی‌کند زیرا طول بلند سریهای زمانی معمولاً نقاطی با ابعاد بسیار بالا می‌سازد. هنگامی‌که ابعاد زیاد شود، کارآیی ساختارهای شاخص‌گذاری مختلف به تدریج کاهش می‌یابد.

شاخص گذاری تشابه سریهای زمانی بازگشتی با تابع فاصله غیر متریک

توابع فاصله که نسبت به داده‌های بسیارمغشوش، مقاوم هستند، معمولاً نامساوی مثلثی را نقض می‌کنند. چنین توابعی، همه بخشها در سری زمانی را یکسان در نظر نمی‌گیرند. اگر چه این رفتار مفید است، زیرا مدل صحیح‌تری از ادراک بشری را نشان می‌دهد. زمانی‌که مردم هر نوع داده‌ای (تصویر، سری زمانی و. . .) را مقایسه می‌کنند، بیشتر روی قسمتهایی که شبیه هستند، تمرکز کرده و توجه کمتری به قسمتهایی که شبیه نیستند، می‌کنند. فواصل غیرمتریک امروزه در بسیاری از دامنه‌ها به‌کار گرفته می‌شوند، مانند تطابق رشته (*DNA*)، فیلتر کردن مشارکتی (هنگامی‌که مشتری با الگوی

ازپیش ذخیره شده مشتریان منطبق شود) و بازیابی تصاویر مشابه از بانکهای اطلاعاتی. علاوه بر آن تحقیقات علم روانشناسی مطرح می‌کند که قضاوت‌های مشابه بشری نیز غیرمتریک هستند. به‌علاوه برای توابع فاصله غیرمتریک، حالت‌های زیر مطرح می‌شود:

- سریهای زمانی در نرخ‌های نمونه‌گیری متفاوت یا سرعت‌های مختلف جمع‌آوری می‌شود. سریهای زمانی به‌دست آمده نتایج نمونه‌گیری در فواصل زمانی ثابت را تضمین نمی‌کنند. گیرنده‌های جمع‌آوری‌کننده داده، ممکن است برای یک دوره زمانی خاص، یکسان عمل کنند و به نرخ‌های نمونه گیری متناقض منتهی شوند. علاوه برآن وقتی دوسری زمانی دقیقاً در یک جهت حرکت می‌کنند، ولی یکی با سرعتی دو برابر نسبت به دیگری در حرکت است، نتیجه به احتمال زیاد، فاصله‌اقلیدسی بسیار بزرگی است.
- سریهای زمانی شامل نقاط پرت هستند. نقاط پرت به دوصورت تعریف می‌شوند. یکی رخداد غیر عادی در برگیرنده جمع‌آوری‌کننده داده و دیگری واکنش به خطای بشری می‌باشد.
- سریهای زمانی طول‌های مختلفی دارند. فاصله‌اقلیدسی، مربوط به سریهای زمانی با طول یکسان است. هنگامی‌که طول‌ها متفاوت هستند، ما باید تصمیم بگیریم که آیا می‌خواهیم سری طولانی‌تر را کوتاه کنیم یاسری کوتاه‌تر را باجملات صفر لایه‌گذاری کنیم.

مثالهایی از چنین مترهای فاصله، توابع فاصله $LCSS$ و DTW می‌باشند. مشاهده اینکه هیچ‌کدام از آن دو متریک نیستند ساده است.

مثال: سه دنباله زیر را داریم:

$Seq۱ = (۰,۰,۰,۰,۰,۰,۰)$
$Seq۲ = (۱,۱,۱,۱,۱,۱,۱)$
$Seq۳ = (۲,۲,۲,۲,۲,۲,۲)$

به‌طور مشابه $LCSS$ برای این دنباله‌ها زیر اگر ε=۱ برابر است با:

$LCSS(Seq۱, Seq۲) = ۰$
$LCSS(Seq۲, Seq۳) = ۰$
$LCSS(Seq۱, Seq۳) = ۱$

مثال: سه دنباله زیر را داریم:

$Seq۱ = (۰,۱,۱,۰,۰,۱,۰,۰)$
$Seq۲ = (۰,۰,۰,۱,۱,۰,۰,۰)$
$Seq۳ = (۰,۱,۰,۰,۱,۰)$
$DTW(Q,C) = min\left\{\sqrt{\sum_{k=۱}^{K} w_k} \Big/ K\right\}$
$LCSS(X,Y) = (m+n-۲l)/(m+n)$

	DTW	*LCSS*
Dis (*seq*۱ , *seq*۲)	۱/۰	۰/۲۵۰
Dis (*seq*۱ , *seq*۳)	۰	۰/۱۴۳

	۰	۱	۱	۰	۰	۱	۰	۰
	۰	۰	۰	۰	۰	۰	۰	۰
۰	۰	۰	۱	۱	۰	۰	۱	۰	۰
۰	۰	۰	۱	۲	۰	۰	۱	۰	۰
۰	۰	۰	۱	۲	۰	۰	۱	۰	۰
۱	۰	۱	۰	۰	۱	۱	۰	۱	۱
۱	۰	۱	۰	۰	۱	۲	۰	۱	۲
۰	۰	۰	۱	۱	۰	۰	۱	۰	۰
۰	۰	۰	۱	۲	۰	۰	۱	۰	۰
۰	۰	۰	۱	۲	۰	۰	۱	۰	۰

$Dis(seq۲, seq۳)$ ۰/۱۱۱ ۰/۲۸۶

$DTW(Seq۱, Seq۲) = ۱/۱۰$

		٠	١	١	٠	٠	١	٠	٠
	٠	٠	٠	٠	٠	٠	٠	٠	٠
٠	٠	٠	١	١	٠	٠	١	٠	٠
١	٠	١	٠	٠	١	١	٠	١	١
٠	٠	٠	١	١	٠	٠	١	٠	٠
٠	٠	٠	١	٢	٠	٠	١	٠	٠
١	٠	١	٠	٠	١	١	٠	١	١
٠	٠	٠	١	١	٠	٠	١	٠	٠

$$DTW(Seq١, Seq٣) = ٠$$

		٠	٠	٠	١	١	٠	٠	٠
	٠	٠	٠	٠	٠	٠	٠	٠	٠
٠	٠	١	١	١	١	١	١	١	١
١	٠	١	١	١	٢	٢	٢	٢	٢
١	٠	١	١	١	٢	٣	٣	٣	٣
٠	٠	١	٢	٣	٣	٣	٤	٤	٤
٠	٠	١	٢	٣	٣	٣	٤	٥	٥
١	٠	١	٢	٣	٤	٤	٤	٥	٥
٠	٠	١	٢	٣	٤	٤	٥	٥	٦
٠	٠	١	٢	٣	٤	٤	٥	٦	٦

$$DTW(Seq٣, Seq٢) = ١/٩$$

		٠	٠	٠	١	١	١	٠	٠
	٠	٠	٠	٠	٠	٠	٠	٠	٠
٠	٠	٠	٠	٠	١	١	٠	٠	٠
١	٠	١	١	١	٠	٠	١	١	١
٠	٠	٠	٠	٠	١	١	٠	٠	٠
٠	٠	٠	٠	٠	١	٢	٠	٠	٠
١	٠	١	١	١	٠	٠	١	١	١
٠	٠	٠	٠	٠	١	١	٠	٠	٠

$$LCSS(Seq١, Seq٢) = (٨ + ٨ - ٢ \times ٦)/(٨ + ٨) = ١/٤$$

		۰	۱	۰	۰	۱	۰
	۰	۰	۰	۰	۰	۰	۰
۰	۰	۱	۱	۱	۱	۱	۱
۰	۰	۱	۱	۲	۲	۲	۲
۱	۰	۱	۱	۲	۳	۳	۳
۱	۰	۱	۲	۲	۳	٤	٤
۰	۰	۱	۲	۳	۳	٤	٤
۰	۰	۱	۲	۳	٤	٤	۵
۰	۰	۱	۲	۳	٤	٤	۵
	۰	۱	۲	۳	٤	٤	۵

$$LCSS(Seq,۳Seq۲) = (۸+۶-۲\times۵)/(۸+۶) = ٤/۱٤$$

		۰	۱	۰	۰	۱	۰
	۰	۰	۰	۰	۰	۰	۰
۰	۰	۱	۱	۱	۱	۱	۱
۱	۰	۱	۲	۲	۲	۲	۲
۱	۰	۱	۲	۲	۲	۳	۳
۰	۰	۱	۲	۳	۳	۳	٤
۰	۰	۱	۲	۳	٤	٤	٤
۱	۰	۱	۲	۳	٤	۵	۵
۰	۰	۱	۲	۳	٤	۵	۶
۰	۰	۱	۲	۳	٤	۵	۶

$$LCSS(Seq۱,Seq۳) = (۸+۶-۲\times۶)/(۸+۶) = ۱/۷$$

شکل ۸-۱۸) قدم‌های الگوریتم

$$DTW(Seq۳,Seq۲) = ۱/۹$$
$$LCSS(Seq۱,Seq۲) = (۸+۸-۲\times۶)/(۸+۸) = ۱/٤$$
$$LCSS(Seq۳,Seq۲) = (۸+۶-۲\times۵)/(۸+۶) = ٤/۱٤$$
$$LCSS(Seq۱,Seq۳) = (۸+۶-۲\times۶)/(۸+۶) = ۱/۷$$

برای شاخص گذاری مناسب سریهای زمانی براساس مفهوم تشابه، نیاز به استفاده از تکنیکهای کاهش بعد می‌باشد. در ادامه به بررسی ضرورت استفاده از تکنیکهای کاهش بعد و تبدیل داده‌ها پرداخته می‌شود.

تکنیکهای تبدیل و کاهش داده

به سبب اندازه بسیار زیاد و ابعاد بالای داده‌هایی به شکل سری‌های زمانی، تکنیکهای کاهش داده به‌عنوان اولین قدم در تحلیل سریهای زمانی به‌کار می‌رود. تکنیکهای کاهش داده نه تنها منجر به اشغال فضای کمتر می‌شود، بلکه سرعت پردازش داده‌ها را نیز افزایش می‌دهد. همان‌طورکه در فصل دوم اشاره شد، مهم‌ترین استراتژی برای کاهش داده، انتخاب زیرمجموعه‌ای از ویژگیها می‌باشد که در آن ویژگیهای نامرتبط و اضافی این داده‌ها در نظر گرفته نمی‌شود. علاوه بر این روشهای کاهش بعد و کاهش حجم داده‌ها (نظیر نمونه گیری، خوشه‌بندی و. . .) نیز از دیگر ویژگیهای مؤثر و کارآمد برای کاهش مقادیر عظیم داده‌ها می‌باشد. به‌دلیل اینکه سریهای زمانی به‌عنوان یک نوع از داده‌های با حجم بالا مد نظر قرار می‌گیرند و هر نقطه بر حسب زمان به‌عنوان یک بعد در نظر گرفته می‌شود، کاهش بعد یکی از مهم‌ترین مباحث مرتبط با تحلیل سریهای زمانی است. یکی از دلایل توجه روزافزون به کاهش بعد، در تحلیل و داده‌کاوی سریهای زمانی، کاهش پیچیدگی محاسبات در اثر کاهش حجم و بعد داده‌ها می‌باشد [٦].

ایده کلیدی شاخص‌گذاری مؤثر سریهای زمانی، کاهش بُعد فضاست. برای کاهش ابعاد فضا، نمونه n بعدی را که نشان‌دهنده سری زمانی است، در یک فضای k بعدی ($k<<n$) تصویر می‌کنیم، به‌طوری‌که فاصله‌ها تاجای ممکن بدون تغییر باقی بمانند. سپس می‌توانیم از یک روش شاخص‌گذاری روی فضای جدید که ابعاد کمتری دارد، استفاده کنیم. چارچوب عمومی‌روش *GEMINI*، برای شاخص‌گذاری سریهای زمانی، با تکنیکهای کاهش بعد از گامهای زیر تبعیت می‌کند.

- مجموعه سریهای زمانی را بر روی یک فضای کاهش بعد یافته، نگاشت می‌کنیم.
- از یک روش شاخص‌گذاری برای شاخص‌گذاری فضای جدید، استفاده می‌کنیم.
- برای سری زمانی مورد جستجوی Q، دنباله Q را بر روی فضای جدید نگاشت کرده و نزدیک‌ترین همسایگی‌ها به Q را در فضای جدید، با استفاده از شاخص‌گذاری پیدا می‌کنیم. سپس فواصل واقعی را محاسبه کرده و نزدیک‌ترین را انتخاب می‌کنیم.

تکنیکهای کاهش بعد متعددی در تحلیل سریهای زمانی مورد استفاده قرار می‌گیرند که از آن جمله می‌توان به تبدیل فوریه گسسته، تبدیل موجک گسسته، تجزیه مقدار منفرد بر مبنای تحلیل مؤلفه‌های اصلی و تصویرکردن تصادفی اشاره کرد. این فنون در فصل دوم، قسمت کاهش بعد توضیح داده شده‌اند.

تبدیلات *SVD* دارای مزیت کاهش بعد بهینه تصاویر خطی می‌باشد. یعنی بهترین حفظ را از میانگین مربع خطا بین تصاویر اصلی و تصاویر تقریبی انجام می‌دهد. اگر چه محاسبه آن در مقایسه با روشهای دیگر دشوار است، مخصوصا اگر سریهای زمانی خیلی طولانی باشند. علاوه بر این، این روش برای شاخص‌گذاری زیر دنباله‌ها کاربرد ندارد.

تبدیلات گسسته فوریه، طیف فرکانس یک سیگنال یک بعدی را توصیف می‌کند. روش *DFT* به‌عنوان یک روش کاهش بعد برای سریهای زمانی ارائه شده است.

تجزیه موجک[1] که سریهای زمانی را به شکل مجموع توابع اولیه نشان می‌دهد، مشابه روش *DFT* می‌باشد. ساختار روش *WD* با *DFT* فرق دارد، زیرا تأثیر ضرایب مختلف در آن بیشتر در زمان متمرکز شده‌اند تا در فرکانس. فواید *WD* آن است که سریهای زمانی تبدیل شده در همان دامنه (دامنه موقت) باقی می‌ماند و الگوریتم کارآیی با

[1]- WD

پیچیدگی $O(n)$ برای محاسبه تبدیلات وجود دارد. معایب آن، مشابه روش *DFT* است.

تصویر کردن تصادفی یک روش کاهش بعد عمومی است که درسال ۱۹۹۸ ارائه ودرسال ۲۰۰۱ برای حوزه سریهای زمانی به کار گرفته شد.

استفاده از مقیاس‌بندی چند بعدی برای شاخص‌گذاری سریهای زمانی، دشوار است. روش *MDS*، نگاشت یک مجموعه از سریهای زمانی به نقاط k بعدی است (که k یک مقدار کوچک است). برای پاسخ‌گویی به جستجوی تشابه، باید قادر باشیم سری زمانی قابل جستجو را روی فضای k بعدی نگاشت کنیم. به علت نوع روشی که الگوریتم *MDS* برای یافتن جستجوهای مورد نظر استفاده می‌کند، باید فواصل همه سریهای زمانی داخل مجموعه آن را بیابیم و این عملیاتی خطی است.

نگاشت سریع، تخمینی بوده و روشی بسیار شبیه به روش مقیاس‌دهی چندبعدی است. برای بررسی مبسوط روشهای کاهش بعد به فصل پیش‌پردازش، بخش کاهش بعد رجوع کنید.

تخمین قطعه‌ای خط[1]

به جای استفاده از *DFT* برای تقریب یک سری زمانی می‌توانیم از تقریب چندجمله‌ای استفاده کنیم. اگر چه با استفاده از یک روش خطی، یک روش عمومی برای تعریف مقیاس اندازه‌گیری غیراقلیدسی به دست می‌آید، ولی هیچ روش شاخص‌گذاری عمومی قابل قبولی شناخته نشده است. تحقیقات اخیر نشان می‌دهند که شاخص‌گذاری این سریهای زمانی درصورتی‌که سریهای زمانی با تابع ثابت قطعه‌ای تقریب زده شود، امکان‌پذیر است. ایده اصلی این روش، کاهش بعد از طریق تقسیم سریهای زمانی به k قطعه هم‌اندازه می‌باشد. مقدار میانگین داده‌هایی که در هر قطعه می‌افتند محاسبه شده و

[1]- Line Segment Approximation

برداری از این مقادیر، نمایش داده‌های کاهش بعد یافته می‌باشد. این نوع نمایش داده‌ها، تخمینهای خطی- قطعه‌وار نامیده می‌شود. این نوع ارائه، نسبت به روش DFT مزایایی دارد. مثلاً تبدیلات می‌تواند در زمانهای خطی اجرا شود. مهم‌تر از آن، این روش گستره‌ای از مقیاس‌های اندازه‌گیری، مانند نرم L_p گسسته، DTW و جستجوی اقلیدسی وزندار را پشتیبانی می‌کنند. این روش مانند WD است، هنگامی‌که K، ضریب ۲ داشته و میانگین برای تعریف زیردنباله‌ها استفاده شود.

منابع

۱) فاطمی قمی م. ت. (۱۳۷۵) «پیش‌بینی وتجزیه وتحلیل سریهای زمانی"، نشردانش امروز .

۲) نیرومند ح. ، بزرگ‌نیا ا. (۱۳۷۲) "مقدمه‌ای برتحلیل سریهای زمانی"، نشردانشگاه فردوسی مشهد .

۳) نیرومند، حسینعلی، ۱۳۷۱، تجزیه وتحلیل سریهای زمانی، نشردانشگاه فردوسی مشهد

٤) ابریشمی، حمید، ۱۳۷۳، اقتصادسنجی کاربردی

5) Ye N. (2003) "The hand book of data mining"

6) Han. J, Kamber. M,(2006) "data mining concepts and techniques"

7) Chu K. W. ,Wong M. H. ,(1998) "fast time series searching with scaling and shifting"

8) Keogh. E, Pazzani M. (2000) "Scaling up dynamic time warping for datamining application",proceeding of the sixth acm sigkdd conference on knowledge discovery and data mining

9) Keogh E. , Pazzani. M,(2001) "derivative dynamic time warping"

10) Keogh E. et al (2004) "Exact indexing of dynamic time warping"

11) Banerjee A. , Ghosh J. (2000) "clickstream clustering using weighted longest common subsequences"

12) Faloutsos C. , Lin K. (1995) "Fast map"

13) Smith L. I. (2002) "A tutorial on principal components analysis"

فصل نهم

تحلیل شبکه‌های اجتماعی

در دهه‌های اخیر، نظریه شبکه‌های اجتماعی (که در آن رابطه میان موجودیت‌ها [1] به صورت پیوندهای درون یک گراف نشان داده می‌شوند) توجهات روز افزونی را به خود جلب کرده است. بدین ترتیب تحلیل شبکه‌های اجتماعی از دید داده‌کاوی، تحلیل پیوندها یا پیوندکاوی نیز نامیده می‌شود. در این بخش، ما ابتدا مفهوم شبکه‌های اجتماعی را مطرح نموده و به مطالعه ویژگی‌های شبکه‌های اجتماعی می‌پردازیم. سپس نگاهی به وظایف و چالشهای موجود در پیوندکاوی خواهیم داشت و در نهایت چند نمونه از شبکه‌های اجتماعی را مورد کاوش قرار می‌دهیم.

[1]- Entity

تعریف شبکه اجتماعی

از دید داده‌کاوی، شبکه اجتماعی مجموعه داده‌های ناهمگن[1] (نامتجانس) و چندرابطه‌ای[2] است که توسط یک گراف نمایش داده می‌شود. نوعاً گرافها بسیار بزرگ و متشکل از گره‌هایی معادل اشیاء و یالهایی[3] معادل پیوندها (که معرف رابطه یا تعامل بین اشیا می‌باشند) هستند. هم گره‌ها و هم پیوندها ویژگیهایی دارند. اشیا ممکن است برچسب دسته داشته باشند. پیوندها می‌توانند یک‌طرفه باشند و لزومی ندارد که دوحالته باشند.

لازم نیست شبکه‌های اجتماعی زمینه‌ای اجتماعی داشته باشند. مثالهای واقعی بسیاری از شبکه‌های اجتماعی تکنولوژیکی، تجاری، اقتصادی و زیستی وجود دارد: شبکه‌های توزیع نیروی الکتریسیته، گرافهای تماسهای تلفنی، انتشار ویروسهای کامپیوتری، شبکه گسترده جهانی، شبکه‌های استناد و هم‌نویسندگی دانشمندان. مثال دیگر عبارت است از شبکه‌های مشتریان در مواقعی که توصیه محصول بر اساس اولویت سایر مشتریان صورت می‌گیرد. در بیولوژی، نمونه‌ها طیف وسیعی را از شبکه‌های شیوع بیماری، شبکه‌های متابولیک و سلولی، شبکه غذایی تا شبکه عصبی کرم (تنها موجودی که شبکه عصبی‌اش کاملاً نگاشت شده است) دربرمی‌گیرند. تبادل پیامهای پست الکترونیک بین شرکتها، گروه‌های خبری، اتاقهای گپ، شبکه دوستی، هم‌پوشانی هیئت مدیره‌های شرکتهای بزرگ آمریکایی و غیره نمونه‌هایی از حوزه جامعه‌شناسی هستند.

«شبکه‌های (اجتماعی) دنیای کوچک[4]» اخیراً نظرات بسیاری را به خود معطوف داشته است. آنها مفهوم «دنیای کوچک» را منعکس می‌نمایند که در اصل بر شبکه‌های بین

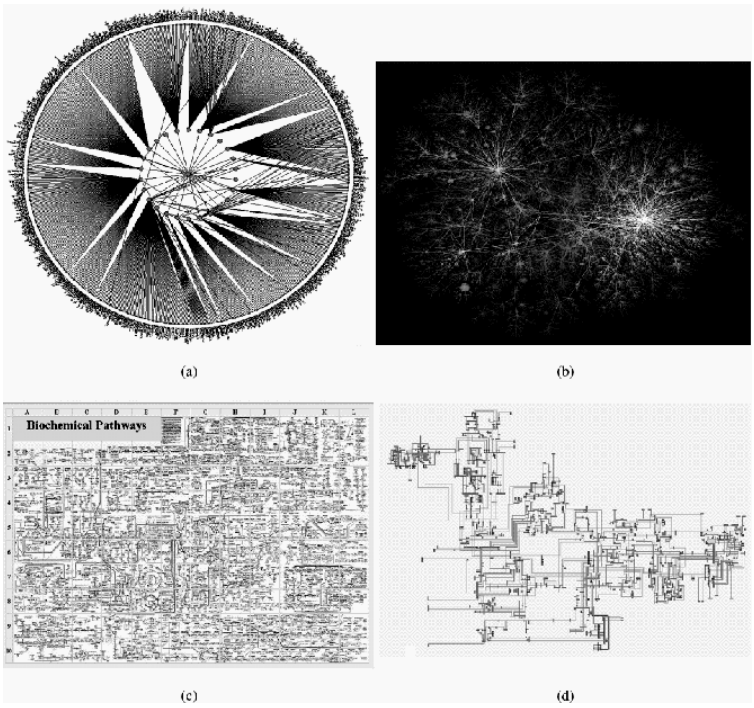
---

[1]- Heterogeneous
[2]- Multirelational
[3]- Edges
[4]- Small World

افراد تمرکز دارد. همان عبارتی‌که نشان دهنده تعجب زیاد اولیه میان دو غریبه است، وقتی در می‌یابند که به‌صورت غیر مستقیم از طریق نفر سومی که هر دو با وی سابقه آشنایی دارند، با هم رابطه دارند: «چه دنیای کوچکی!». در ۱۹٦۷ استنلی میلگرام[1] جامعه‌شناس هاروارد و همکارانش آزمایشی را ترتیب دادند که در آن از مردم کانزاس و نبراسکا درخواست شده بود تا نامه‌هایی را به دست افرادی غریبه در بوستون برسانند، بدین صورت‌که این نامه‌ها را به دوستانی ارسال کنند که گمان می‌کنند ممکن است آن افراد غریبه را بشناسند. نیمی از نامه‌ها به‌طور موفقیت‌آمیزی از طریق کمتر از ۵ واسطه به مقصد رسیدند. مطالعات دیگری که میلگرام و سایرین در شهرهای دیگر صورت دادند، نشان داد که ظاهراً به‌طور عمومی «شش سطح جدایی[2]» بین هر دو نفر در جهان وجود دارد. نمونه‌هایی از شبکه‌های دنیای کوچک در شکل (۹-۱) نشان داده شده است. شبکه‌های دنیای کوچک با ویژگی دارا بودن درجه بالایی از خوشه‌پذیری محلی برای نسبت کوچکی از گره‌ها شناخته می‌شوند (به‌عنوان مثال این گره‌ها با دیگری به هم وصل می‌شوند)، که در عین حال بیش از چند سطح از سایر گره‌ها باقیمانده جدا نیستند. این باور وجود دارد که بسیاری از شبکه‌های زیستی، اجتماعی، و فیزیکی ساخته دست بشر این ویژگی‌های دنیای کوچک را به معرض نمایش می‌گذارند. این ویژگیها بعداً توضیح داده شده و مدلسازی می‌شوند.

---

[1]- Stanley Milgram
[2]- Six Degrees of Separation



شکل ۹-۱) مثالهای شبکه اجتماعی در دنیای حقیقی: a) هم‌نویسندگی علمی، b) صفحات مرتبط در بخشی از اینترنت، c) مسیر بیوشیمی، d) شبکه قدرت الکتریسیته نیویورک

«به‌طور کلی چرا این همه علاقه به شبکه‌های دنیای کوچک و شبکه‌های اجتماعی وجود دارد؟ چه فایده‌ای در مشخص نمودن ویژگی‌های شبکه‌ها و کاوش آنها به منظور بیشتر آموختن از ساختارشان وجود دارد؟» علت آن است که ساختار همواره بر عملکرد مؤثر است. به‌عنوان مثال توپولوژی شبکه‌های اجتماعی بر شیوع بیماریهای مسری در یک جمعیت ساختار یافته تأثیر دارد. توپولوژی شبکه نیرو بر ثبات و انسجام انتقال نیرو مؤثر است. به‌طور نمونه، یک نارسایی مرتبط در تاریخ ١٤ آگوست ۲۰۰۳ در کلیولند واقع در اوهایو در سیستم شبکه ایجاد شده. این نارسایی منجر به از کار افتادن کارخانجات نیروی هسته‌ای در ایالت نیویورک و اوهایو گردید و باعث خاموشی گسترده در بخشهای زیادی از شمال شرقی ایالات متحده و جنوب شرقی کانادا شد که حدود ۵۰ میلیون نفر را در بر می‌گرفت. توجه به شبکه‌ها بخشی از مطالعات وسیع‌تری

است که در توصیف کامل و دقیق «سیستمهای پیچیده[1]» انجام می‌شود. قبلا شبکه‌هایی که برای مطالعات تجربی در دسترس بودند، محدود و کوچک بودند و اطلاعات ناچیزی از گره‌های آنها وجود داشت. باید از اینترنت ممنون باشیم که در حال حاضر مقادیر عظیمی از داده‌های مرتبط با شبکه‌های اجتماعی بسیار بزرگ را در دسترس ما قرار داده است. این شبکه‌ها نوعاً دهها هزار تا میلیونها گره را دربرمی‌گیرد و اغلب اطلاعات وسیعی در سطح هر گره موجود می‌باشد. دسترسی به کامپیوترهای قدرتمند نیز بررسی ساختار شبکه‌ها را ممکن ساخته است. مطالعه شبکه‌های اجتماعی می‌تواند به ما در دسترسی بهتر و آسانتر به سایر مردم دنیا کمک کند. به‌علاوه، مطالعه دنیای کوچک، با توجه به جدایی نسبتا کم بین گره‌هایشان، می‌تواند به ما در طراحی شبکه‌هایی که انتقال کارای اطلاعات یا سایر منابع را تسهیل می‌کنند، یاری رساند، بدون آنکه مجبور باشیم از شبکه‌ای با تعداد زیادی رابطه زائد استفاده کنیم. به‌عنوان مثال می‌تواند به ما در طراحی موتورهای جستجو هوشمندتری در وب کمک کند، به‌طوریکه در پاسخ به یک پرس‌وجو، وب سایتهای مرتبطی که کمترین میزان جدایی از وب سایت اولیه را دارند، بیابند.

ویژگیهای شبکه‌های اجتماعی

همان‌طورکه در قسمت قبل توضیح داده شد، دانستن ویژگیهای شبکه‌های دنیای کوچک در موقعیتهای بسیاری مفید است. می‌توان مدلهای مولد گراف را که دارای همین ویژگیها باشند، ایجاد نمود. تا قادر به پاسخگویی به سؤالات «اگر – آنگاه» و پیش‌بینی چگونگی یک شبکه در آینده باشند. به‌عنوان مثال در مورد اینترنت، می‌پرسیم «اگر تعداد گره‌ها در اینترنت دو برابر شود، آنگاه اینترنت چگونه به نظر می‌رسد؟» و «تعداد یالها چندتا خواهد بود؟». چنانچه فرضیه‌ای با ویژگیهای عموماً پذیرفته شده،

[1]- Complex Systems

تناقض داشته باشد، پرچمی در برابر مشکوک بودن فرضیه برمی‌افرازد. این موارد به کشف ناهنجاریها در گرافهای موجود کمک می‌نماید که ممکن است تقلب، هرزنگاری[۱] یا حمله‌های *Ddos* را مشخص کند. به‌علاوه مدلهای مولد گراف می‌تواند برای شبیه سازی مواردی که گرافهای واقعی بیش از اندازه بزرگ هستند و بدین لحاظ جمع‌آوریشان غیرممکن است (مثل شبکه بسیار بزرگی از رابطه دوستی) کمک کند. در این قسمت، ویژگیهای اساسی شبکه‌های اجتماعی به همراه مدلی برای تولید گراف بررسی می‌شود.

چه ویژگیهایی شبکه‌های اجتماعی را مشخص می‌کند؟ اکثر مطالعات، امتیاز گره‌ها[۲] را بررسی کرده‌اند که همان تعداد یالهای منتهی به هر گره است و یا فاصله بین هر دو گره را که با محاسبه طول کوتاهترین مسیر تعیین می‌شود. سایر محاسبات فاصله میان دو گره شامل فاصله متوسط[۳] و قطر مؤثر[٤] (به‌عنوان مثال، حداقل فاصله d به‌طوری‌که لااقل برای ۹۰٪ گره‌ها در دسترس، طول مسیر حداکثر d باشد) می‌باشد.

شبکه‌های اجتماعی به ندرت ایستا هستند. نمایش گراف آنها با افزایش یا حذف گره‌ها و یالها در طول زمان، تکامل می‌یابد. به‌طورکلی، شبکه‌های اجتماعی پدیده‌های زیر را نشان می‌دهند:

قانون تراکم توانی[۵]: در گذشته، باور بر این بود که با تکامل یک شبکه، امتیاز گره‌ها به صورت خطی نسبت به تعداد گره‌ها افزایش می‌یابد. این باور با نام فرض میانگین امتیاز ثابت[٦] شناخته می‌شد. با این وجود، آزمایشات گسترده‌ای نشان داده‌اند که برعکس، با افزایش متوسط امتیاز در طول زمان، شبکه‌ها به‌طور فزاینده‌ای چگال می‌شوند (بدین

---

[1]- Spam
[2]- Nodes Degrees
[3]- Average Distance
[4]- Effective Diameter
[5]- Densification Power Law
[6]- Constant Average Degree Assumption

ترتیب، تعداد یالها به نسبت تعداد گره‌ها به‌صورت فوق خطی[1] افزایش می‌یابند). این متراکم شدن از قانون تراکم توانی (یا قانون رشد توانی[2]) پیروی می‌کند و عبارتست از:

$$e(t) \propto n(t)^a$$

که در آن $e(t)$ و $n(t)$ به ترتیب نمایانگر تعداد یالها و گره‌های گراف در زمان t هستند و توان a عموما بین ۱و۲ قرار می‌گیرد. توجه داشته باشید که چنانچه $a=1$ باشد، معادل میانگین امتیاز ثابت در طول زمان است، درحالی‌که $a=2$ بیانگر گرافی به شدت متراکم است که در آن هر گره توسط یالهایش به نسبت ثابتی از کل گره‌ها، متصل است.

جمع شدن قطر[3]: به‌طور تجربی اثبات شده که قطر مؤثر با رشد شبکه کاهش می‌یابد. این با باور اولیه‌ای که معتقد بود قطر به آهستگی به‌صورت تابعی از اندازه شبکه افزایش می‌یابد، درتناقض است.

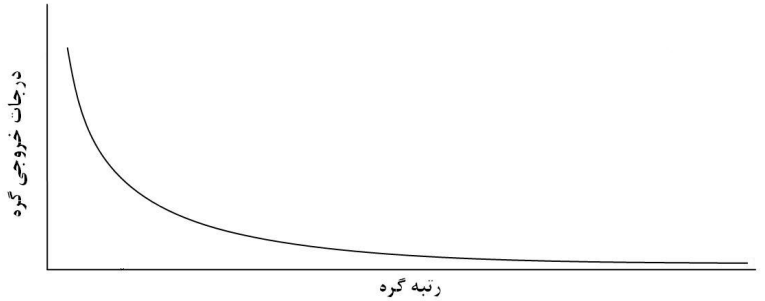
برای درک شهودی این مطلب یک شبکه استناد را درنظر بگیرید که در آن گره‌ها مقالات هستند و استناد (ارجاع) از یک مقاله به مقاله‌ای دیگر با یک یال جهت‌دار نمایش داده می‌شود. یالهای خروجی از گره v (نمایانگر مقالاتی که v به آنها ارجاع داده است) در لحظه پیوستن به گراف «منجمد[4]» هستند. متعاقبا به نظر می‌رسد فاصلهٔ در حال کاهش بین دو گره، مقالات بعدی باشد که به‌عنوان «پل[5]» عمل کرده و مقالات اولیه را مورد ارجاع قرار داده‌اند.

توزیع‌های دم‌پهن[6] درجات ورودی و خروجی: با توجه به «قانون توان» تعداد امتیازات خروجی یک گره به پیروی از توزیعی دم‌پهن گرایش دارد: $1/n^a$ که در آن n رتبه گره در ترتیب کاهش امتیاز خروجی است و نوعاً $0 < a < 2$ است (شکل (۹-۲)).

---

[1]- Superlinearly
[2]- Growth Power Law
[3]- Shrinking Diameter
[4]- Frozen
[5]- Bridge
[6]- Heavy-Tailed

هرچه مقدار a کوچک‌تر باشد، دنباله پهن‌تر خواهد بود. این حالت بیانگر الحاق ترجیحی[1] است که در آن هر گره جدید با تعداد ثابتی یالهای خروجی به شبکهٔ موجود وصل می‌شود و از قاعده «ثروتمند، ثروتمندتر می‌شود[2]» پیروی می‌کند. امتیازات ورودی هم از توزیع دم‌پهن پیروی می‌کند که در ضمن نسبت به توزیع امتیازت خروجی چولگی بیشتری دارد.



شکل ۹ – ۲) تعداد درجات خروجی (محور y) یک گره تمایل به پیروی از توزیعی دم‌پهن دارد. رتبه گره (محور x) ترتیب نزولی درجات خروجی گره است.

برای تولید گراف مدل آتش جنگل[3] پیشنهاد شد که این خصوصیات تکامل گراف در طول زمان را دارا است. این مدل بر پایه این نظریه استوار است که گره‌های جدیدی از طریق سوزاندن[4] یالهای موجود به طریق مسری به شبکه متصل شده و از دو پارامتر بهره می‌گیرد: احتمال سوزاندن پیشرو (p) و نسبت سوزاندن پسرو (r) که در ادامه توضیح داده می‌شوند. فرض کنید گره جدید v در زمان t اضافه شود، این گره در طی گامهای زیر به G_t، گرافی که تاکنون ایجاد شده، متصل می‌شود:

[1]- Preferential Attachment Model
[2]- Rich-Get-Richer
[3]- Forest fire Model
[4]- Burning

گام اول: به‌طور تصادفی یک گره سفیر[1]، w انتخاب می‌کند و یک یال به w متصل می‌کند.

گام دوم: x یال متصل به w را انتخاب می‌کند. x مقداری تصادفی است که دارای توزیع برنولی با میانگین $(1-p)^{-1}$ است. هم یالهای ورودی به w و هم یالهای خروجی از w در نظر گرفته می‌شوند ولی یالهای ورودی را با احتمال r بار کمتر از یالهای خروجی برمی‌گزینند. گره‌هایی را که در سر دیگر یالهای انتخاب شده قرار دارند، $w_1, w_2, ..., w_x$ بنامید.

گام سوم: گره جدید (w) یالهای خروجی را به $w_1, w_2, ..., w_x$ متصل نمود، حال گام ۲ را به‌طور بازگشتی برای هر کدام از گره‌های $w_1, w_2, ..., w_x$ انجام دهید. به‌منظور جلوگیری از افتادن در حلقه، هیچ گره‌ای برای بار دوم نباید انتخاب شود. این فرآیند تا زمانی که خاموش شود ادامه می‌یابد.

برای درک شهودی از مدل، به مثال خودمان از شبکهٔ استناد، باز می‌گردیم. نویسنده مقاله جدید، v، ابتدا به w مراجعه می‌کند. سپس یک زیر مجموعه از منابع w را دنبال می‌کند (که ممکن است پیشرو یا پسرو باشند) و به مقالات $w_1, w_2, ..., w_x$ دست می‌یابد. با ارجاع دادن این مقالات، تجمیع مراجع به‌صورت بازگشتی ادامه می‌یابد.

بسیاری از مدلهای تکامل شبکه بر گرافهای ایستا مبتنی است که ویژگیهای شبکه را بر اساس یک یا تعداد معدودی تصویر آنی از آن با کمی تأکید بر یافتن روندها در طول زمان تعیین می‌کنند. مدل آتش جنگل درحالی‌که به تکامل شبکه در طول زمان توجه دارد، ماهیت بسیاری از مدلهای قبلی را درهم می‌آمیزد. مثلاً خاصیت امتیازهای خروجی دم‌پهن که به طبیعت بازگشتی تشکیل یالها تعلق دارد را نیز در نظر می‌گیرد؛ بدین ترتیب، گره‌های جدید شانس خوبی در سوزاندن بسیاری یالها و لذا ایجاد امتیازات خروجی بزرگ دارند. خاصیت امتیازات ورودی دم‌پهن نیز حفظ می‌شود، زیرا

---

[1]- Ambassador

آتش جنگل از قاعده «ثروتمند، ثروتمندتر می‌شود» پیروی می‌کند. گره‌هایی که خیلی زیاد به هم متصل هستند، بدون در نظر گرفتن اینکه گره جدید از کدام گره سفیر آغاز می‌شود، به آسانی به گره‌ای جدید متصل می‌شوند. گونه‌ای از مدل کپی‌کننده[1] نیز مورد نظر قرار گرفته است بدین صورت که یک گره جدید بسیاری از همسایگان گره سفیر خود را کپی می‌کند. قانون تراکم توانی نیز تأیید می‌شود: یک گره جدید یالهای بسیاری در نزدیکی اجتماع گره سفیر خود خواهد داشت. مطالعات تجربی خاصیت جمع شدن قطر را تأیید می‌کنند. گره‌هایی که امتیازات خروجی دم‌پهن دارند، ممکن است به‌عنوان «پلهایی» به‌کار روند که قسمتهایی از شبکه را که پیش‌تر ناهمگن بودند، متصل کنند و قطر شبکه را کاهش دهند.

پیوندکاوی[2]: وظایف و چالشها

چگونه می‌توان شبکه‌های اجتماعی را مورد کاوش قرار داد؟ روشهای سنتی یادگیری ماشینی و داده‌کاوی که به‌عنوان ورودی، نمونه‌های تصادفی اشیاء همگنی را از یک رابطهٔ واحد اخذ می‌کنند، ممکن است برای این حالت مناسب نباشند. داده‌های تشکیل‌دهنده شبکه‌های اجتماعی بیشتر ناهمگن، چندرابطه‌ای و نیمه‌ساخت‌یافته هستند. در نتیجه، حوزه جدیدی از پژوهش با نام پیوندکاوی پدید آمده است. پیوندکاوی محل تلاقی پژوهش در شبکه‌های اجتماعی، تحلیل پیوندها، وب‌کاوی و ابرمتنها[3]، گراف-کاوی، یادگیری رابطه‌ای[4] و برنامه‌ریزی منطقی قیاسی[5] است. پیوندکاوی متضمن مدلهای توصیفی و پیشبینانه است. با درنظر گرفتن پیوندها (رابطه میان اشیاء) اطلاعات

---

[1]- Copying Model
[2]- Link Mining
[3]- Hypertext
[4]- Relational Learning
[5]- Inductive Logic Programming

بیشتری برای فرآیند کاوش حاصل می‌شود. این امر وظایف جدید بسیاری را به همراه می‌آورد، لیست این وظایف با مثالهایی از حوزه‌های مختلف در زیر فهرست شده‌اند.

دسته‌بندی اشیاء مبتنی بر پیوندها: در روشهای سنتی دسته‌بندی، اشیاء بر اساس ویژگیهایی که آنها را توصیف می‌کردند، دسته‌بندی می‌شدند. در دسته‌بندی مبتنی بر پیوندها، دستهٔ یک شیء نه تنها بر اساس ویژگیهایش، بلکه بر اساس پیوندهایش و ویژگیهای اشیایی که با آنها پیوند دارد، پیش‌بینی می‌شود.

دسته‌بندی صفحات وب مثال بسیار مناسبی از دسته‌بندی مبتنی بر پیوندهاست. در دسته‌بندی صفحات وب، دسته یک صفحه هم بر اساس رخداد واژه‌ها (کلماتی که بر روی آن صفحه واقع شده‌اند) و هم بر اساس متن لنگر[۱] (کلمات ابَرپیوندی[۲] یعنی کلماتی که شما هنگام کلیک بر روی یک پیوند، بر روی آنها کلیک می‌کنید) انجام می‌شود و هر دوی آنها به‌عنوان ویژگیهای صفحه به‌کار می‌روند. به‌علاوه، دسته‌بندی بر پایه پیوندهای بین صفحات و سایر ویژگیهای صفحات و پیوندها نیز می‌باشد. در حوزه کتاب‌شناسی[۳]، اشیاء ما مقالات، نویسندگان، مؤسسات، مجلات و کنفرانسها هستند. در این حالت وظیفه دسته‌بندی، پیش‌بینی مبحث یک مقاله است که بر اساس رخداد واژه‌ها، مورد ارجاع بودن (سایر مقالاتی که به این مقاله ارجاع داده‌اند)، و استناد کردن (سایر مقالاتی که در این مقاله به آنها استناد شده است) انجام می‌شود. استنادها به‌عنوان پیوند عمل می‌کنند. در مبحث شناخت بیماریهای همه‌گیر، وظیفه دسته‌بندی، پیش‌بینی نوع بیماری یک فرد است که بر اساس ویژگیهای (علائم بیماری) فرد بیمار و ویژگیهای سایر افرادی که فرد بیمار با آنها تماس داشته است (از این افراد با عنوان تماسهای بیمار یاد می‌شود) صورت می‌گیرد.

---

[1]- Anchor Text
[2]- Hyperlink
[3]- Bibliography

پیش‌بینی نوع شیء: این پیش‌بینی، نوع یک شیء را بر اساس ویژگیهای خودش و پیوندهایش و ویژگیهای اشیایی که به آن متصلند تعیین می‌کند. در حوزه کتاب‌شناسی ممکن است بخواهیم محل یک مطلب منتشر شده را مشخص کنیم مثلاً کنفرانس یا مجله یا کارگاه. در حوزه ارتباطات، وظیفه مشابهی وجود دارد که پیش‌بینی آن است که آیا یک رابطه از طریق پست الکترونیک است، یا تلفن یا پست؟

پیش‌بینی نوع پیوند: این پیش‌بینی، بر اساس مشخصه‌های اشیاء درگیر در یک پیوند، نوع یا هدف آن پیوند را معلوم می‌سازد. به‌عنوان مثال با در دست داشتن داده‌ها شیوع بیماریهای همه‌گیر، ممکن است بخواهیم پیش‌بینی کنیم آیا دو نفر که یکدیگر را می‌شناسند از اعضای یک خانواده‌اند، یا همکارند، یا صرفاً با یکدیگر آشنا هستند. در نمونه‌ای دیگر ممکن است بخواهیم بدانیم آیا رابطه بین دو مؤلف به‌صورت مشاور–مشورت کننده است؟ با دستیابی به داده‌های صفحات وب می‌توانیم پیش‌بینی کنیم آیا یک پیوند روی یک صفحه، پیوند تبلیغاتی است یا پیوندی مربوط به پیمایش وب؟

پیش‌بینی وجود پیوند: در پیش‌بینی نوع پیوند ما می‌دانستیم پیوندی بین دو شیء موجود است و می‌خواستیم نوع این پیوند را معلوم کنیم اما در این حالت می‌خواهیم پیش‌بینی کنیم آیا اصلاً بین دو شیء پیوندی وجود دارد؟ مثال: پیش‌بینی اینکه آیا پیوندی بین دو صفحه وب وجود خواهد داشت؟ آیا مقاله‌ای به مقاله دیگری استناد خواهد کرد؟ و یا در علم شناخت بیماریهای همه‌گیر می‌توانیم پیش‌بینی کنیم یک بیمار با چه کسی در تماس بوده است.

تخمین عدد اصلی[1] **پیوند**: دو شکل تخمین عدد اصلی پیوند وجود دارد. اول، می‌توان تعداد پیوندهای متصل به یک شیء را پیش‌بینی نمود. به‌طور مثال تعیین مأخذیت[2] یک صفحه وب می‌تواند بر اساس تعداد پیوندهایی که به آن منتهی شده‌اند (پیوندهای

---

[1]- Cardinality
[2]- Authoritativeness

ورودی) مشخص شود. به‌طور مشابه تعیین تعداد پیوندهای خروجی[۱] می‌تواند مبین این باشد که آیا آن صفحه وب به‌عنوان قطب[۲] عمل می‌کند (منظور از قطب، یک یا مجموعه‌ای از صفحات وب است که به تعدادی صفحات مأخذ در همان مبحث استناد می‌کنند)، در حوزه کتاب‌شناسی، تعداد استنادات به یک مقاله می‌تواند نشانگر تأثیر آن مقاله باشد (هرچه استنادات به مقاله بیشتر باشد، انتظار می‌رود تأثیرگذاری بیشتری داشته باشد)، در علم بیماریهای همه‌گیر، پیش‌بینی تعداد پیوندهای بین یک بیمار و تماسهایش، نشانگر انتقالهای بالقوه بیماری است.

حالت مشکل‌تر تخمین عدد اصلی پیوند، پیش‌بینی تعداد اشیایی است که در خلال یک مسیر از یک شیء قرار دارند. این مسئله در تخمین تعداد اشیائی که در پاسخ به یک پرس‌وجو باز می‌گردد، حائز اهمیت است. در حوزه صفحات وب، می‌توان تعداد صفحاتی را که از طریق پویش یک سایت بازیابی می‌شود، پیش‌بینی نمود (منظور از پویش، جستجویی خودکار و روشمند در وب است که اساساً به‌منظور تهیه یک کپی از تمام صفحات مشاهده شده است تا برای پردازشهای بعدی یک موتور جستجو استفاده شود). با در نظر گرفتن مسئله استناددهی، می‌توان از تخمین کاردینالیتی پیوندی برای پیش‌بینی تعداد استنادات یک نویسنده خاص در یک مجله خاص بهره برد.

مصالحه[۳] اشیاء: در مصالحه اشیاء، وظیفه پیش‌بینی این است که آیا در واقع دو شیء بر اساس ویژگیها و پیوندهایشان یکسان هستند؟ این وظیفه در استخراج اطلاعات[۴]، حذف دوباره‌کاری[۵]، یکی‌سازی اشیاء و انطباق[۶] استنادات امری معمول است و با عنوان عدم‌قطعیت موجودیتها[۷] یا ارتباط رکورد[۸] نیز شناخته می‌شود. به‌عنوان نمونه: پیش‌بینی

---

[1]- Out-Links
[2]- Hub
[3]- Reconciliation
[4]- Information Extraction
[5]- Duplication
[6]- Matching
[7]- Identity Uncertainty
[8]- Record Linkage

اینکه آیا دو وب سایت آینه یکدیگرند، آیا دو استناد واقعاً به یک مقاله استناد می‌کنند، و آیا دو درد آشکار حاصل از بیماری، در واقع یکی هستند؟

کشف گروه‌ها: کشف گروه نوعی خوشه‌بندی است و به پیش‌بینی این امر می‌پردازد که چه هنگام مجموعه‌ای از اشیاء به یک گروه یا خوشه تعلق دارند و این کار را بر اساس ویژگیهای آن اشیاء و ساختار پیوندهایشان انجام می‌دهد. یک حوزه کاربرد آن تعیین اجتماعات وبی[۱] است یعنی مجموعه‌ای از صفحات وب که بر یک عنوان یا زمینه[۲] خاص متمرکز هستند. کاربرد مشابه، تعیین اجتماعات پژوهشی در حوزه کتاب‌شناسی می‌باشد.

کشف زیر گرافها[۳]: تعیین زیر گرافها، زیر گرافهای ویژه‌ای را در درون شبکه می‌یابد و نوعی از جستجوی گرافی است. مثالی از بیولوژی کشف زیر گرافهای متناظر با ساختار پروتئینهاست. در شیمی نیز می‌توان زیر گرافهایی را جستجو کرد که زیر ساختارهای شیمیایی را نمایش می‌دهند.

فراداده‌کاوی: فراداده‌ها، داده‌هایی در مورد داده‌ها هستند. فراداده‌ها، داده‌هایی نیمه‌ساخت‌یافته درباره داده‌های ساخت‌نیافته فراهم می‌کنند که طیف وسیعی از داده‌های وب و متن گرفته تا پایگاههای داده‌های چند رسانه‌ای را در برمی‌گیرد. این وظیفه برای یکپارچه‌سازی داده‌ها در بسیاری حوزه‌ها مفید است. فراداده‌کاوی را می‌توان برای نگاشت شماتیک[٤] (مثلاً ویژگی شماره مشتری از یک پایگاه داده به شماره مشتری از پایگاه داده‌ای دیگر نگاشت می‌شود چون هر دو آنها به یک موجودیت اشاره دارد)؛ کشف شماتیک[٥] (از داده‌های نیمه‌ساخت‌یافته طرحهایی ایجاد می‌کند)؛ و تشکیل مجدد شماتیک[٦] (بر اساس فراداده‌های داده‌کاوی شده، طرح را اصلاح می‌کند) به‌کار گرفت.

---

[1]- Web Communities
[2]- Theme
[3]- Subgraph
[4]- Schema Mapping
[5]- Schema Discovery
[6]- Reformulation Schema

نمونه‌ها شامل این موارد است: انطباق دو منبع کتاب‌شناختی، کشف طرح از داده‌های نیمه‌ساخت‌یافته یا غیرساخت‌یافته روی وب، و نگاشت بین دو هستان‌شناسی[۱] دارویی. به‌طور خلاصه، استفاده از اطلاعات پیوند بین اشیاء، وظیفه‌ای اضافی برای پیوندکاوی در مقایسه با رویکردهای سنتی کاوش به همراه می‌آورد. به‌کارگیری این وظایف در هر حال چالشهای بسیاری را دربرخواهد داشت. در اینجا چند نمونه از این چالشها را بررسی می‌کنیم.

وابستگیهای منطقی در برابر وابستگیهای آماری: دو نوع وابستگی در ساختارهای گراف قرار دارند: ساختارهای پیوند (مبین رابطه منطقی بین اشیاء) و وابستگیهای احتمالی[۲] (مبین رابطه آماری مثل همبستگی بین ویژگیهای اشیاء درحالتی‌که نوعاً چنین اشیایی منطقاً به هم مرتبطند). کار هم‌زمان با این وابستگیها نیز خود چالشی برای کاوش داده‌های چندرابطه‌ای است که در آن داده‌هایی که مورد کاوش قرار می‌گیرند، در جداول چند لایه‌ای قرار دارند. علاوه بر جستجوهای استاندارد بر روی وابستگیهای احتمالی بین ویژگیها، می‌بایست بر روی رابطه‌های منطقی مختلف ممکن بین اشیاء نیز جستجو صورت گیرد. این امر فضای جستجوی وسیعی را می‌طلبد که یافتن یک مدل ریاضی موجه را مشکل می‌سازد. روشهای توسعه‌یافته در برنامه‌ریزی منطق قیاسی که بر روی ارتباطات منطقی جستجو می‌کند، اینجا می‌تواند مفید واقع شود.

ساخت مشخصه‌ها[۳]: در دسته‌بندی مبتنی بر پیوندها، ما هم به ویژگیهای یک شیء توجه می‌کنیم و هم به ویژگیهای اشیای متصل به آن. به‌علاوه، ممکن است پیوندها هم ویژگیهایی داشته باشند. هدف از ساخت مشخصه‌ها، ایجاد یک صفت منفرد مبین این ویژگیهاست. این امر می‌تواند شامل انتخاب مشخصه‌ها[۴] و تجمیع مشخصه‌ها[۵] باشد. در

---

[1]- Ontologies
[2]- Probabilistic Dependencies
[3]- Feature Construction
[4]- Feature Selection
[5]- Feature Aggregation

انتخاب مشخصه‌ها، فقط مشخصه‌های مهم و متمایزکننده درنظر گرفته می‌شوند. تجمع مشخصه‌ها، چند مجموعه از مقادیر بر روی مجموعه اشیاء مرتبط را می‌گیرد و خلاصه‌ای از آن را باز می‌گرداند. این خلاصه می‌تواند به‌عنوان مثال مُد مجموعه (مقداری که بیشترین تعداد رخداد را دارد)؛ میانگین مجموعه (اگر مقادیر عددی باشد)؛ یا میانه (اگر مقادیر به ترتیب مرتب شده باشند) باشد. برخی اوقات این روش مناسب نیست.

نمونه‌ها[1] **در برابر دسته‌ها**: این چالش مربوط است به اینکه آیا این مدل صریحاً به تک‌تک نمونه‌ها اشاره دارد یا به دسته‌هایی (طبقه‌های عمومی) از نمونه‌ها. یک مزیت مدل پیشین در این است که می‌تواند برای اتصال تک تک نمونه‌های خاص با احتمال بالا به‌کار رود. یک مزیت مدل آخر این است که می‌تواند برای عمومیت بخشیدن به موقعیتهای جدید با نمونه‌های منفرد مختلف استفاده شود.

دسته‌بندی جمعی و یکی‌سازی جمعی[2]: آموزش دادن[3] یک مدل برای دسته‌بندی را در نظر بگیرید که بر اساس مجموعه‌ای از اشیاء که برچسب یا نام دسته‌شان مشخص است، انجام شود. روشهای دسته‌بندی سنتی تنها به ویژگیهای شیء توجه می‌نمود. فرض کنید پس از آموزش مدل، مجموعه جدیدی از اشیایی که برچسب دسته‌شان معلوم نیست در اختیار داریم. به‌کارگیری مدل برای تعیین برچسب دسته اشیاء جدید، با توجه به همبستگی‌های احتمالی بین اشیاء پیچیده است (برچسب دسته اشیاء به هم مرتبط ممکن است همبسته باشد). بنابراین دسته‌بندی می‌بایست گام تکرار شونده دیگری را هم در بربگیرد که برچسب دسته هر شیء را بر اساس برچسب دسته اشیاء مرتبط با آن تغییر دهد (یا تثبیت کند). در این معنا، دسته‌بندی بیشتر جمعی انجام می‌شود تا مستقلاً.

---

[1]- Instances
[2]- Collective Consolidation
[3]- Training

استفاده مؤثر از داده‌های برچسب خورده و برچسب نخورده: یک استراتژی جدید در یادگیری این است که مخلوطی از داده‌های برچسب خورده و برچسب نخورده را شرکت دهید. داده‌های برچسب نخورده می‌تواند به استنتاج توزیع ویژگیهای اشیاء کمک کند. پیوند بین داده‌های برچسب خورده (داده‌های آموزشی) و برچسب نخورده (داده‌های آزمون) پی به وابستگیهایی می‌برد که می‌تواند به استدلالهای دقیق‌تر کمک کند.

پیش‌بینی پیوندها: یک چالش موجود در پیش‌بینی پیوندها این است که احتمال پیشین یک پیوند خاص بین اشیاء بسیار کم است. رویکردهای مختلف در باب پیش‌بینی پیوندها بر اساس تعدادی معیار برای تحلیل مجاورت گره‌ها در یک شبکه، مطرح شده‌اند. مدلهای احتمالی هم مطرح شده‌اند. برای مجموعه‌های بزرگ داده ممکن است مدل کردن پیوندها در سطحی بالاتر مؤثر باشد

فرض دنیای باز در مقابل دنیای بسته[1]: سنتی‌ترین رویکردها فرض می‌کنند که ما تمام موجودیتهای بالقوه را در این حوزه می‌شناسیم. این فرض دنیای بسته در کاربردهای دنیای واقع، غیرواقعی است. کار در این زمینه، معرفی یک زبان برای تعیین توزیعهای احتمال ساختارهای رابطه‌ای را دربرمی‌گیرد که متضمن مجموعه‌ای متغیر از اشیاء است.

اجتماع‌کاوی بر روی شبکه‌های چندرابطه‌ای: کار بر روی تحلیل شبکه‌های اجتماعی نوعاً کشف گروه‌هایی از اشیاء را که در ویژگیهای مشابهی سهیم هستند، دربرمی‌گیرد. به این‌کار اجتماع[2]کاوی گفته می‌شود. پیوند[3] صفحات وب یک مثال است که در آن اجتماع کشف شده می‌تواند مجموعه‌ای از صفحات وب مربوط به یک مبحث خاص باشد. اغلب الگوریتمهای اجتماع‌کاوی فرض می‌کنند که تنها یک شبکه اجتماعی وجود دارد که نشانگر رابطه نسبتاً همگنی است. در واقعیت، چندین شبکه اجتماعی ناهمگن

---

[1]- Closed Versus Open World Assumption
[2]- Community
[3]- Linkage

وجود دارد که بیانگر روابط گوناگونی هستند. چالش جدید کاوش اجتماعات پنهان در چنین شبکه‌های اجتماعی ناهمگن است که به آن اجتماع‌کاوی بر روی شبکه‌های اجتماعی چندرابطه‌ای گفته می‌شود. این چالشها ادامه دارد تا انگیزه‌ای برای تحقیقات بیشتر در پیوند کاوی باشد.

کاوش شبکه‌های اجتماعی

در این قسمت ما چند مثال در حوزه‌های مختلف کاوش بر روی شبکه‌های اجتماعی را بررسی می‌کنیم. این نمونه‌ها شامل پیش‌بینی پیوندها، کاوش شبکه‌های مشتریان برای بازاریابی ویروسی[1]، کاوش گروه‌های خبری با استفاده از شبکه‌ها و اجتماع‌کاوی از شبکه‌های چندرابطه‌ای است. سایر نمونه‌ها شامل کشف زیرگرافهای مشخصه در گراف‌کاوی و کاوش ساختارهای پیوند در وب‌کاوی است. کاربردهای دیگری مانند خوشه‌بندی و دسته‌بندی مبتنی بر پیوند نیز وجود دارند.

پیش‌بینی پیوندها: چه یالهایی به شبکه افزوده خواهد شد؟

شبکه‌های اجتماعی پویا هستند. پیوندهای جدیدی ظاهر می‌شده که نشان دهنده تعاملی جدید بین اشیاء هستند. در مسئله پیش‌بینی پیوندها، یک تصویر آنی[2] از شبکه اجتماعی در زمان t در اختیار ما قرار داده می‌شود و پیش‌بینی اینکه در بازه زمانی t تا $t+1$ چه یالهایی به این شبکه افزوده خواهد شد، از ما خواسته می‌شود. در این حالت ما به دنبال این هستیم که با استفاده از صفات حقیقی خود مدل، از توسعه‌ای که می‌تواند تکامل یک شبکه اجتماعی را مدل کند، پرده برداریم. به‌عنوان مثال یک شبکه اجتماعی هم‌نویسندگی یا تألیف مشترک، بین دانشمندان را درنظر بگیرند. به‌طور شهودی ممکن است پیش‌بینی کنیم که دو دانشمند که در شبکه نزدیک یکدیگر قرار دارند، احتمال

[1]- Viral Marketing
[2]- Snapshot

دارد در آینده با هم همکاری داشته باشند. بر این اساس پیش‌بینی پیوندها را می‌توان به عنوان بخشی از مطالعات مدلهای تکامل شبکه‌های اجتماعی در نظرگرفت.

رویکردهایی که در باب پیش‌بینی پیوندها مطرح شده‌اند، مبتنی بر معیارهای مختلفی از تحلیل مجاورت گره‌های یک شبکه می‌باشند. بسیاری از معیارها از روشهای تحلیل شبکه‌های اجتماعی و نظریه گراف سرچشمه می‌گیرند. روش کلی بدین صورت است: در تمام روشها به هر جفت گره X و Y، بر اساس گراف ورودی G و اندازه مجاورت موجود، یک وزن اتصال[1] $Scroe(X,Y)$ تخصیص می‌دهند. سپس یک ترتیب نزولی از $Scroe(X,Y)$ تهیه می‌شود که پیوندهای جدید پیش‌بینی شده را به ترتیب نزولی اطمینان به ما می‌دهد. این پیش‌بینیها را می‌توان بر اساس مشاهدات واقعی از مجموعه داده‌های تجربی ارزیابی نمود. ساده‌ترین رویکرد، جفتهای (X,Y) را براساس طول کوتاهترین مسیرشان در G مرتب می‌کند. این رویکرد نظریه دنیاهای کوچک را مجسم می‌کند که در آن تک تک اجزا از طریق کوتاهترین زنجیره به یکدیگر متصل هستند. از آنجا که هدف مشترک همه روشها، مرتب کردن تمام جفتها به ترتیب کاهش امتیاز می‌باشد، در اینجا $Scroe(X,Y)$ به‌صورت منفی طول کوتاهترین مسیر تعریف می‌شود.

بسیاری از معیارها از اطلاعات همسایگی استفاده می‌کنند. ساده‌ترین نوع چنین معیارهایی همسایگان مشترک[2] است (هر قدر تعداد همسایگان مشترک X,Y بیشتر باشد، احتمال اینکه در آینده بین X,Y پیوندی ایجاد شود بیشتر است). از لحاظ شهودی، اگر نویسنده X,Y هرگز مقاله مشترکی تالیف نکرده باشند ولی همکاران مشترک بسیاری داشته باشند، با احتمال بیشتری آنها در آینده با یکدیگر همکاری خواهند داشت. سایر معیارها بر اساس مجموع تمام مسیرهای[3] بین دو گره استوار است. به‌عنوان مثال معیار کتز[4] تمام مسیرهای وزن دهی شده بین X و Y را محاسبه

---

[1]- Connection Weight
[2]- Common Neighbors
[3]- Ensemble of All Paths
[4]- Katz

می‌کند به‌طوری‌که به مسیرهای کوتاهتر وزن بیشتری اختصاص می‌دهد. تمام این معیارها را می‌توان با ترکیبی از رویکردهای سطح بالاتر مانند خوشه‌بندی به‌کار برد. به‌طور نمونه، روش پیش‌بینی پیوند را می‌توان برای نسخه اصلاح شده یک گراف به‌کار برد که در آن یالهای قلابی حذف شده‌اند.

در آزمایشات انجام شده برروی مجموعه داده‌ها استناد یا نقل قول، هیچ‌کدام از روشها بر دیگر روشها مقدم نیست. بسیاری روشها به‌طور عمده یک پیشگویی تصادفی[1] را به‌دست می‌آورند که معتقد است توپولوژی شبکه‌ها می‌تواند اطلاعات مفیدی برای پیش‌بینی پیوندها فراهم آورد. معیار کتز و نسخه‌ها گوناگون مبتنی بر خوشه‌بندی آن، همواره خوب عمل کرده‌اند هرچند دقت پیش‌بینی همچنان بسیار پایین است. کارهای آینده در زمینه پیش‌بینی پیوندها ممکن است هم بر یافتن راههای بهتر استفاده از اطلاعات توپولوژی شبکه متمرکز شود و هم کارآیی محاسبات فاصله گره‌ها را مثلاً از طریق تخمین زدن بهبود بخشد.

کاوش شبکه‌های مشتریان به منظور بازاریابی ویروسی

بازاریابی ویروسی کاربردی از کاوش شبکه‌های اجتماعی است که مشخص می‌کند چگونه افراد می‌توانند بر رفتار خرید سایرین تأثیر بگذارند. به‌طور سنتی شرکتها، بازاریابی مستقیم[2] (که در آن تصمیم فروش از طریق یک فرد خاص و صرفاً بر اساس خصوصیات آن فرد گرفته می‌شد) یا بازاریابی انبوه[3] (که در آن افراد بر اساس اینکه به کدام بخش[4] از جمعیت متعلق باشند، صورت می‌گیرد) را به‌کار برده‌اند. در هر حال این رویکردها تأثیری را که مشتریان می‌توانند بر تصمیم خرید سایرین داشته باشند نادیده می‌گیرند. به‌عنوان مثال فردی را در نظر بگیرید که تصمیم می‌گیرد یک فیلم خاص را

---

[1]- Random Predictor
[2]- Direct Marketing
[3]- Mass Marketing
[4]- Segment

ببیند و گروهی از دوستان را نیز برای تماشای آن فیلم تشویق می‌کند. هدف بازاریابی ویروسی بهینه‌کردن تأثیر مثبت گفتار شفاهی[۱] در میان مشتریان است. ممکن است بخواهیم هزینه بیشتری برای جذب یک نفر که تماسهای اجتماعی زیادی دارد، اختصاص دهیم. بنابراین با در نظر گرفتن این تعاملات بین مشتریان، بازاریابی ویروسی می‌تواند سود بیشتری نسبت به بازاریابی سنتی که چنین تعاملاتی را نادیده می‌گرفت، کسب نماید. رشد اینترنت در دهه‌های اخیر موجب در دسترس قرار گرفتن شبکه‌های اجتماعی بسیاری شده است که می‌توان با هدف بازاریابی ویروسی به کاوش آنها پرداخت، مثل لیستهای پست الکترونیک، گروه‌های خبری[۲]، اجتماعات برخط[۳] گپهای *IRC*[۴]، پیام آنی[۵]، سیستمهای فیلتر مشارکتی و سایتهای شراکت دانش[۶]. سایتهای شرکت دانش (مانند *Epinions* در *www.epinions. com*) به‌کاربران خود اجازه می‌دهند (نوعاً به‌صورت مجانی) محصولات را به سایرین توصیه کنند یا برای محصولات برآورد قیمت نمایند تا به سایرین کمک کرده باشند. کاربران می‌توانند مفید بودن یا قابلیت اعتماد[۷] یک مقاله مروری[۸] را ارزش‌گذاری کنند، و حتی درصورت امکان سایر منتقدان را نیز ارزیابی نمایند. در این صورت یک شبکه روابط با عنوان شبکه اعتماد[۹] بر اساس اعتماد بین کاربران شکل می‌گیرد که بیانگر یک شبکه اجتماعی است که می‌تواند مورد کاوش قرار بگیرد.

ارزش شبکه‌ای[۱۰] یک مشتری، افزایش مورد انتظار در فروش به سایرین است که از جذب آن مشتری حاصل می‌شود. در مثال فوق، اگر مشتری مورد نظر ما سایرین را

1- Word of Mouth
2- Use Net Groups
3- On Line Forums
4- Instant Relay Chat
5- Instant Messaging
6- Knowledge- Sharing Sites
7- Trust Worthiness
8- Review
9- Web of Trust
10- Network Values

متقاعد کند که یک فیلم خاص را ببینند، شرکت فیلم‌سازی به صرف هزینه بیشتر برای ترغیب وی به مشاهده فیلم مشتاق می‌شود. در عوض اگر این مشتری نوعاً هنگام تصمیم‌گیری درباره اینکه چه فیلمی را تماشا کند، به سایرین گوش می‌کند، هزینه بازاریابی صرف شده برای وی را می‌توان اتلاف منابع محسوب نمود. بازاریابی ویروسی ارزش شبکه‌ای یک مشتری را درنظر می‌گیرد. در حالت ایده‌آل، ترجیح می‌دهیم شبکه یک مشتری را مورد کاوش قرار دهیم (به‌عنوان مثال شبکه دوستان و اقوام وی) تا نه تنها بر اساس ویژگیهای مشتری، بلکه بر اساس تأثیر همسایگان مشتری در شبکه، پیش‌بینی کنیم، چقدر احتمال دارد وی یک کالای خاص را بخرد. اگر مجموعه مشخصی از مشتریان را مورد بازاریابی قرار دهیم، از طریق بازاریابی ویروسی می‌توانیم میزان سود مورد انتظار از کل شبکه را پس از آنکه تأثیر آن مشتریان در شبکه تکثیر شد، در تحقیقاتمان دنبال کنیم. این کار می‌تواند به ما در یافتن مجموعه بهینه مشتریانی که مورد بازاریابی قرار می‌گیرند، یاری رساند. ارزش شبکه‌ای مشتریان (که در بازاریابی مستقیم سنتی نادیده گرفته می‌شوند) می‌تواند به طرح بازاریابی بهبود یافته‌ای بیانجامد.

مجموعه n مشتری بالقوه را در نظر بگیرید، X_i را یک متغیر دوحالته فرض کنید که اگر مشتری i محصول بازاریابی شده را بخرد مقدار یک می‌گیرد و در غیر این صورت مقدارش صفر می‌شود. همسایگان X_i مشتریانی هستند که مستقیماً بر X_i تأثیر می‌گذارند. M_i اقدام بازاریابی[1] است که در مورد مشتری iام صورت می‌گیرد. M_i می‌تواند دوحالته (اگر کوپنی[2] برای این مشتری ارسال شود، یک و در غیر این صورت صفر) یا طبقه‌ای (نشانگر اقدامات متفاوت ممکن اتخاذ شده در قبال مشتری) باشد. در حالتی دیگر M_i می‌تواند مقادیری پیوسته (مثلاً نشان دهنده میزان تخفیفی که برای

[1]- Marketing Action
[2]- Coupon

مشتری قائل شده‌ایم) باشد. یافتن طرح بازاریابی که سود را بیشینه کند، مطلوب ماست. یک مدل احتمالی مطرح شده است که M_i را به‌عنوان مقداری پیوسته، بهینه می‌کند. در این مدل به جای اینکه تنها تصمیمی دوحالته اتخاذ شود که آیا این مشتری را مورد بازاریابی قرار بدهد یا خیر، مقدار هزینه صرف شده برای بازاریابی هر مشتری، بهینه می‌شود.

این مدل عواملی را که بر ارزش شبکه مؤثرند، در نظرمی‌گیرد. اول این مشتری می‌بایست اتصالات زیادی در شبکه داشته باشد و به‌علاوه محصول در نظر وی خوب قلمداد شود. درصورتی‌که یک مشتری دارای اتصالات زیاد، نگرش منفی نسبت به محصول مورد نظر داشته باشد، ارزش شبکه‌ای او می‌تواند منفی باشد که در این حالت، بازاریابی (جذب) وی توصیه نمی‌شود. دوم، این مشتری می‌باید ترجیحاً بر دیگران تأثیرگذار باشد تا تأثیرپذیر. سوم طبیعت بازگشتی[1] اثر گذاری تبلیغ شفاهی را می‌بایست مد نظر قرار داد. یک مشتری ممکن است بر آشنایان خود تأثیر بگذارد. آنها هم به نوبه خود ممکن است محصول را بپسندند و بر سایر افراد تأثیر بگذارند وضع به همین ترتیب ادامه می‌باید تا جایی که کل شبکه پوشش داده می‌شود. در ضمن این مدل ملاحظات مهم دیگری را نیز در نظر می‌گیرد: ممکن است تولید کننده بپذیرد که مقداری پول را برای مشتریانی که به اندازه کافی بر دیگران تأثیر مثبت دارند از دست بدهد. به‌عنوان مثال، فروش مجانی به یک مشتری برگزیده مناسب می‌تواند چندین برابر در فروش به سایر مشتریان جبران شود. این رویکرد چرخشی بزرگ نسبت به بازاریابی سنتی مستقیم است که در آن درصورتی‌که سود حاصل از فروش به یک مشتری به تنهایی از هزینه توصیه محصول به وی تجاوز کند، تحفیفی به آن مشتری تعلق می‌گیرد. این حقیقت را که، تنها دانش جزئی نسبت به شبکه داریم و جمع‌آوری

[1]- Recursive

چنین دانشی نیز می‌تواند هزینه‌های مربوط به خود را داشته باشد، در این مدل مورد توجه قرار می‌گیرد.

یافتن مجموعه مشتریان بهینه به‌صورت یک مسئله بهینه‌سازی که به خوبی تعریف شده، فرموله شده است: یافتن مجموعه مشتریانی که سود خالص را بیشینه می‌کنند. این مسئله از نوع دشوار *NP* شناخته شده است. با به‌کارگیری رویه جستجوی تپه‌نوردی ساده می‌توان با تقریب ٦۳٪ جواب بهینه را به‌دست آورد. مادامی‌که افزایش مشتریان سود سرانه را بهبود بخشد، مشتریان جدیدی به مجموعه مشتریان افزوده خواهند شد. این روش با وجود دانش ناقص از شبکه باثبات[1] شناخته شده است.

روشهای بازاریابی ویروسی را می‌توان برای حوزه‌های دیگر نیز به‌کار برد. کاهش انتشار ویروس *HIV*، مبارزه با استعمال دخانیات در نوجوانان و ابتکار سیاسی اجتماعی محلی[2] مثالهایی از این نوع هستند. به‌کارگیری روشهای بازاریابی ویروسی برای حوزه وب و عکس آن، زمینه‌های جالبی برای تحقیقات آتی هستند.

کاوش گروه‌های خبری با کمک شبکه‌ها

تحلیل شبکه‌های اجتماعی مبتنی بر وب با وب‌کاوی رابطه نزدیکی دارد. در وب‌کاوی دو الگوریتم رتبه‌بندی متداول به نام صفحه‌رتبه و *HITS* به‌کار می‌رود. این الگوریتمها بر این پایه استوارند که پیوندی از صفحه وبی *A* به *B* معمولاً نشانه تأیید *B* توسط *A* است.

وضعیت در گروه‌های خبری روی مباحث موضوعی متفاوت است. یک پست نوعی در گروه خبری شامل یک یا چند خط نقل قول[3] از پستی دیگر و به دنبال آن نظر نویسنده پست فعلی است. این پاسخ‌های نقل قول‌دار پیوندهای نقل قولی را شکل داده و

1- Robust
2- Grass- Root Political Initiative
3- Quote

شبکه‌ای را ایجاد می‌کنند که در آن رئوس، بیانگر افراد و پیوندها رابطه پاسخ هستند. پدیده جالب این است که مردم بیشتر به پیامی که با آن مخالفند پاسخ می‌دهند تا پیامی که موافقند. این رفتار که در بسیاری از گروه‌های خبری وجود دارد کاملاً با گراف پیوند صفحات وب که در آن پیوند نشانه توافق یا علاقه مشترک می‌باشد، متضاد است. بر پایه این رفتار می‌توان با تحلیل ساختار گراف پاسخ‌ها به‌طور مؤثر، نویسندگان داخل گروه خبری را به دسته‌های مخالف دسته‌بندی و افراز کرد.

این فرایند دسته‌بندی گروه خبری، با نظریه گراف قابل انجام است. اگر فرد i از پست قبلی فرد j نقل قول کند، پیوند نقل قول بین i و j ساخته شده و از روی این پیوندها شبکه یا گراف نقل قول ایجاد می‌شود. حال دوبخشی کردن رئوس به دو مجموعه را در نظر می‌گیریم:

مجموعه F بیانگر موافقین یک مطلب و مجموعه A بیانگر مخالفین آن مطلب است. اگر یالهای گراف گروه خبری بیانگر مخالفت باشند آن‌گاه انتخاب بهینه، حداکثر کردن تعداد یالها بین این دو مجموعه است. از آنجا که مسئله حداکثر برش (حداکثر کردن تعداد یالهای برش خورده برای دو بخش کردن گراف) به‌طور نظری مسئله‌ای از نوع NP است، نیاز به راه حل عملی دیگری داریم. به‌خصوص می‌توان از دو حقیقت دیگر در وضعیت فعلی استفاده کرد: ۱) نمونه ما بیشتر از آن‌که گراف عامی باشد، گرافی دوبخشی است که برخی گره‌های مغشوش به آن اضافه شده‌اند، و ۲) هیچ‌کدام از دو بخش چندان از دیگری کوچکتر نیستند. در چنین وضعیتهایی می‌توان مسئله را به مسئله برش تقریباً متوازن حداقل وزن تبدیل کرد، که به نوبه خود با روشهای طیفی ساده تقریب زده می‌شود. برای بهبود دقت دسته‌بندی می‌توان ابتدا به‌طور دستی تعداد کمی از پست‌کنندگان فعال را دسته‌بندی کرد و رئوس متناظر در گراف را علامت زد. سپس از این اطلاعات برای دستیابی به افرازی بهتر استفاده کرد به این نحو که در حین اجرای الگوریتم افراز، قرار داشتن موارد دستی در دو سو حفظ شود.

بر پایه این ایده‌ها، الگوریتم مؤثری ارائه شده است. آزمایشاتی با چند مجموعه داده گروه خبری در مباحث اجتماعی بحث‌انگیز مانند سقط جنین، کنترل اسلحه و مهاجرت نشان می‌دهد که پیوندها حاوی اطلاعاتی کم‌اغتشاش‌تر از متون هستند. روشهای مبتنی بر تحلیل زبانی و آماری متن، صحت کمتری از تحلیل پیوند در این نوع گروه‌های خبری داشتند، زیرا الفاظ مورد استفادهٔ طرفهای مقابل تا حد زیادی مشابه بوده و بسیاری از پستها حاوی متنی بسیار مختصر هستند که امکان تحلیل زبانی قابل اعتمادی نمی‌دهد.

اجتماع کاوی شبکه‌های چندرابطه‌ای

با رشد وب، اجتماع کاوی توجهات روز افزونی را به خود جلب نموده است. قسمت اعظم چنین کارهایی بر کاوش اجتماعات ضمنی صفحات وب، اجتماعات ضمنی ادبیات علمی برگرفته از وب و اجتماعات ضمنی استنادات متمرکز شده است. در اصل یک اجتماع را می‌توان به‌صورت گروهی از اشیاء که در چندین خصوصیات مشترک سهیم هستند، تعریف نمود. اجتماع کاوی را می‌توان به‌صورت تعیین زیر گرافها در نظر گرفت. به‌عنوان مثال، در صفحات وب مربوط، دو صفحه وب (اشیاء) در صورتی به هم مرتبط هستند که ابرپیوندی بین آنها وجود داشته باشد. گرافی از روابط صفحات وب را می‌توان به منظور تعیین اجتماع یا تعیین مجموعه‌ای از صفحات وب در مورد یک موضوع خاص، مورد کاوش قرار داد.

اغلب روشهای گراف‌کاوی و اجتماع کاوی مبتنی بر گرافهای همگن است. یعنی آنها فرض می‌کنند تنها یک نوع رابطه بین اشیاء وجود دارد. در حالی‌که در شبکه‌های اجتماعی واقعی، همواره انواع مختلف روابط بین اشیاء برقرار است. به هر نوع رابطه می‌توان در قالب یک شبکه روابط[1] نگریست. (که به آن شبکه‌های اجتماعی همگن نیز

[1]- Relation Network

گفته می‌شود). در این‌صورت چند نوع رابطه یک شبکه اجتماعی چندرابطه‌ای[1] را تشکیل می‌دهند (که به آن شبکه‌های اجتماعی غیر همگن نیز گفته می‌شود). هر نوع رابطه ممکن است نقش معینی در یک وظیفه خاص ایفا نمایند. در اینجا گرافهایی با روابط مختلف می‌توانند برای ما اجتماعات متفاوتی را ایجاد نمایند.

به‌منظور یافتن اجتماعی با خصوصیات معلوم ابتدا نیاز است تعیین شود، کدام رابطه نقشی مهم در چنین اجتماعی ایفا می‌کند. چنین رابطه‌ای ممکن است به‌طور صریح وجود نداشته باشد، یعنی ما نیاز داشته باشیم قبل از آنکه آن اجتماع را در چنان شبکه رابطه‌ای بیابیم ابتدا چنین رابطه پنهانی را کشف کنیم. کاربران مختلف ممکن است به روابط متفاوتی در درون شبکه علاقه‌مند باشند. بدین لحاظ اگر ما شبکه‌ها را تنها با در نظر گرفتن تنها یک نوع رابطه مورد کاوش قرار دهیم، ممکن است اطلاعات ارزشمند بسیاری از اجتماع پنهان را از دست بدهیم. چنین کاوشی نمی‌تواند نیازهای متنوع اطلاعاتی کاربران مختلف را برآورده کند. این امر ما را به مسئله کاوش اجتماعات چندرابطه‌ای می‌رساند که کاوش اجتماعات پنهان در شبکه‌های اجتماعی ناهمگن را دربرمی‌گیرد.

مثالی ساده را بررسی می‌کنیم. در یک اجتماع انسانی، ممکن است چندین نوع رابطه وجود داشته باشند: برخی از مردم در یک محل کار می‌کنند؛ بعضی علایق مشترکی دارند؛ برخی به یک درمانگاه می‌روند، والی آخر. از لحاظ ریاضی این اجتماع را می‌توان با یک گراف بزرگ نمایش داد که در آن گره‌ها نماد افرادهستند و یالها قوت رابطه بین آنها را مشخص می‌کنند. از آنجا که انواع مختلف رابطه وجود دارد، یالهای این گراف می‌بایست غیر همگن باشند. در برخی موارد، می‌توان این اجتماع را با به‌کارگیری چندین گراف همگن نیز مدلسازی نمود. هر گراف یک نوع رابطه را منعکس می‌کند. فرض کنید یک بیماری مسری شیوع پیدا کند، و دولت تلاش می‌کند

---

[1]- Multi Relational Social Network

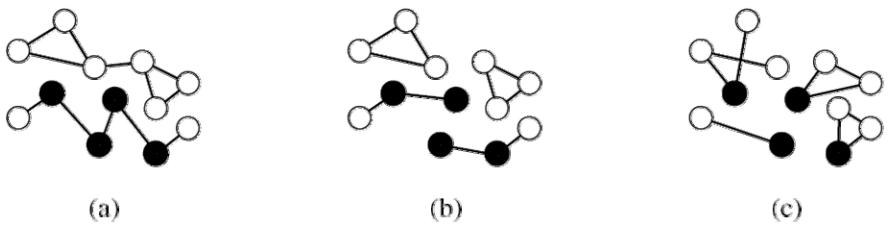
افرادی را که بیشترین احتمال بیمار شدن را دارند، شناسایی نماید. به وضوح پیداست که روابط موجود بین مردم نمی‌توانند نقش یکسانی داشته باشند. منطقی است که فرض کنیم در چنین شرایطی رابطه «کارکردن در یک محل» یا «زندگی کردن با یکدیگر» می‌بایست نقشی حیاتی ایفا نمایند. سؤالی که پیش می‌آید این است: «چگونه می‌توانیم رابطه‌ای که بیشترین نقش را در شیوع بیماری دارد، انتخاب کنیم؟ آیا رابطه پنهانی (بر اساس روابط صریح و آشکار) وجود دارد که به بهترین نحو مسیر شیوع بیماری را فاش نماید؟»

سؤال از لحاظ ریاضی می‌تواند به‌صورت انتخاب و استخراج رابطه[1] در تحلیل شبکه‌های اجتماعی چندرابطه‌ای، مدلسازی شود. مسئله استخراج رابطه را می‌توان به صورت ساده آن‌گونه که در ادامه خواهد آمد، بیان نمود: در یک شبکه اجتماعی چندرابطه‌ای، بر اساس چند نمونه برچسب گذاری شده است (به‌عنوان مثال به‌صورت پرس‌وجوهایی که کاربر فراهم می‌کند). چگونه می‌توان اهمیت روابط مختلف را ارزیابی نمود؟ درضمن چگونه می‌توان به ترکیبی از روابط موجود دست یافت که بیشترین تطابق را با رابطه نمونه‌های برچسب خورده داشته باشد؟

به‌عنوان مثال شبکه نشان داده شده در شکل (۹– ۳) را ببینید که سه نوع رابطه مختلف دارد؛ به ترتیب با (a)، (b) و (c) مشخص شده‌اند. فرض کنید کاربری نیاز دارد که ٤ شیء رنگی به یک اجتماع تعلق داشته باشند و این امر را با یک پرس‌وجو مشخص می‌کند. پیداست که اهمیت نسبی هر کدام از سه نوع رابطه با توجه به نیاز اطلاعاتی کاربر فرق می‌کند. در بین این سه نوع رابطه، (a) بیشترین تطابق را با نیاز کاربر دارد و لذا مهم‌ترین است، در حالی که (b) در رتبه دوم قرار می‌گیرد. رابطه (c) را می‌توان با توجه به نیاز اطلاعاتی کاربر، یک اغتشاش در نظر گرفت. تحلیل سنتی شبکه‌های اجتماعی بین این روابط تمایزی قائل نمی‌شود و با روابط مختلف یکسان برخورد

[1]- Relation Selection & Extraction

می‌کند. آنها به سادگی برای توصیف ساختار بین اشیا با یکدیگر ترکیب می‌شوند. متاسفانه، در این مثال، رابطه (c) تأثیری منفی بر این هدف می‌گذارد. با این وجود، اگر ما این روابط را بر اساس اهمیتشان ترکیب کنیم، رابطه (c) به آسانی حذف می‌شود و روابط (a) و (b) را برای کشف ساختار اجتماع باقی می‌گذارد که با نیاز کاربر مطابق است.



شکل ۹-۳) شبکه‌هایی با روابط مختلف

بعضی مواقع یک کاربر ممکن است پرس‌وجوی پیچیده‌تری را ارائه دهد. مثلاً ممکن است مشخص کند که دو شیء رنگی پایینی می‌بایست به دو اجتماع متفاوت متعلق باشند. در این صورت، اهمیت سه نوع رابطه شکل (۹-۳) تغییر می‌کند. رابطه (b) مهمترین رابطه می‌شود، درحالی که رابطه (a) بی فایده قلمداد می‌شود (حتی با توجه به پرس‌وجو اثری منفی دارد). درنتیجه، در شبکه‌های اجتماعی چندرابطه‌ای، اجتماع‌کاوی می‌بایست بر اساس پرس‌وجوی کاربر (یا اطلاعاتی که لازم دارد) صورت بگیرد. پرس‌وجوی یک کاربر می‌تواند بسیار منعطف باشد. روشهای اولیه تنها بر یک شبکه رابطه متمرکز بودند و بر اساس پرس‌وجوی کابر انجام نمی شدند و لذا نمی‌توانستند پاسخگوی چنین موقعیتهای پیچیده‌ای باشند.

الگوریتمی برای استخراج و انتخاب رابطه مطرح شده که مسئله را به‌صورت یک مسئله بهینه‌سازی مدل می‌کند. این مسئله از لحاظ ریاضی می‌تواند بدین صورت تعریف شود: مجموعه‌ای از اشیاء و مجموعه‌ای از روابط را در اختیار داریم که به‌صورت مجموعه‌ای از گرافهای $G_i(V,E_i)$، $i=1,...,n$ در آن n تعداد روابط است، V مجموعه گره‌هاست

(اشیاء) و E_i مجموعه یالهای مربوط به iامین رابطه است. وزن یالها را می‌توان به صورت طبیعی بر اساس قوت رابطه بین دو شیء تعیین نمود. این الگوریتم هر رابطه را با یک گراف و یک ماتریس اوزان مشخص می‌کند.

M_i را نماد ماتریس اوزان مربوط به Gi در نظر بگیرید. هر مؤلفه در ماتریس قوت رابطه بین یک جفت شیء مربوط را مشخص می‌کند. فرض کنید یک رابطه پنهانی با گراف $G^{\wedge}(V,E^{\wedge})$ مشخص شود و $M^{\wedge}$ هم ماتریس اوزان مرتبط با $G^{\wedge}$ است.

یک کاربر نیاز اطلاعاتی خود را به‌صورت یک پرس‌وجو مشخص می‌کند که در آن مجموعه‌ای از اشیاء برچسب‌گذاری‌شده $X=[x_1,...,x_m]$ را اعلان می‌نماید.

$Y=[y_1,...,y_m]$ نیز به گونه‌ای است که در آن y_j برچسب x_j است (چنین اشیاء برچسب گذاری شده‌ای اطلاعات جزئی از روابط پنهان $G^{\wedge}$ را نشان می‌دهند). هدف این الگوریتم یافتن ترکیب خطی این ماتریسهای اوزان است، به نحوی‌که به بهترین صورت $G^{\wedge}$ (ماتریس اوزان مربوط به اشیاء دارای برچسب) را تخمین بزنند. ترکیب حاصله با احتمال بیشتری نیازمندی اطلاعاتی کاربر را برآورده ساخته و لذا موجب عملکرد بهتری در اجتماع کاوی خواهد شد.

این الگوریتم روی داده‌های کتاب‌شناسی آزمایش شده است. به‌طور طبیعی روابط چندگانه‌ای بین نویسندگان وجود دارد. نویسندگان می‌توانند مقالات خود را در هزاران کنفرانس مختلف به چاپ برسانند، و هر کنفرانس را می‌توان به‌صورت یک رابطه در نظر گرفت که یک شبکه اجتماعی چندرابطه‌ای را ایجاد خواهد کرد. با در دست داشتن چند مثال به‌دست آمده از کاربران (مانند گروه‌های نویسندگان)، این الگوریتم می‌تواند یک رابطه جدید را با استفاده از مثالها استخراج کند و تمام گروه‌های مرتبط دیگر را بیابد. رابطه استخراجی می‌تواند به‌صورت گروهی از نویسندگان باشد که علایق مشترکی را دنبال می‌کنند.

منابع

1) *Han. J, Kamber. M. (2006) "Chapter 9: Graph Mining, Social Network Analysis, and Multirelational Data mining", Data mining concepts and techniques, 2nd edition, , Morgan Kaufmann Publishers*

فصل دهم

کاربرد داده‌کاوی در مدیریت ارتباط با مشتری

توجه: مطالب این فصل به‌طور مستقل از فصول قبل قابل مطالعه بوده و برای دانشجویان رشته‌های تجارت الکترونیک و مدیریت بازرگانی مناسب می‌باشد.

در طی چند سال گذشته تعامل شرکتها با مشتریانشان به‌طور قابل توجهی تغییر کرده است به‌طوری‌که تداوم کسب وکار با مشتری تضمین بلند مدت ندارد. به همین دلیل برای موفقیت یک سازمان لازم است سازمان‌ها مشتریانشان را به‌درستی درک کرده، نیازها و خواسته‌های آنها را پیش‌بینی کنند و با مجهز شدن به این اطلاعات، سلامت کاری خود را بهبود بخشند. بسیاری از سازمانها داده‌های بسیار زیادی را درباره مشتریان، تامین کنندگان و شرکای تجاریشان جمع‌آوری و ذخیره می‌کنند ولی ناتوانی این سازمانها برای کشف دانش پنهان باارزش در این داده‌ها سبب می‌شود که این داده‌ها به دانش تبدیل نشوند و این کار عملاً بیهوده باشد. صاحبان کسب‌وکار میل به استخراج

اطلاعاتی ناشناخته، معتبر و قابل درک از بانک‌های اطلاعاتی عظیم خود و استفاده از این اطلاعات برای کسب سود بیشتر دارند.

برای برخی، داده‌کاوی از نظر فنی جالب است ولی برای بیشتر مردم، این علم وسیله‌ای برای رسیدن به نتایج جالب می‌باشد. داده‌کاوی به تنهایی مفید نیست، بلکه زمانی که به صورت کاربردی در یک مورد خاص استفاده می‌شود، معنا پیدا می‌کند. برای محقق شدن این اهداف سازمانها باید مراحل زیر را طی نمایند:

- جمع‌آوری و یکپارچه‌سازی داده‌های داخلی و خارجی (خرید) درکل سازمان به شکلی قابل درک.
- کاوش داده‌های یکپارچه برای تولید دانش.
- سازماندهی و ارائه اطلاعات و دانش به شیوه‌ای که فرآیندهای تصمیم‌گیری پیچیده را تسریع نماید.

برای محقق شدن همه این اهداف، سازمانها نیاز به یکپارچه‌سازی مؤلفه‌های مختلف برنامه‌های کاربردی خود دارند. یکی از حوزه‌هایی که به سرعت درحال رشد است فنآوری تصمیم‌گیری علمی[1] (معمولاً به آنها موتورهای تصمیم‌گیری گفته می‌شود) و داده‌کاوی در مدیریت ارتباط با مشتری است.

برای حفظ رقابت، سازمانها نیاز به تدوین استراتژیهای تمرکز بر مشتری[2]، مشتری محوری[3]، مشتری مداری[4] دارند. همه این موارد خواسته‌های سازمانها را در راستای ارتباط با مشتریان تعریف می‌کند. مدیریت ارتباط با مشتری[5] راه حلی است که این تلاشها را برای سازمانها و همچنین مشتریان محقق می‌سازد.

---

[1]- Decision Science
[2]- Customer Focused
[3]- Customer Driven
[4]- Customer Centric
[5]- Customer Relationship Management (CRM)

فرض اولیه این است که همه مشتریان با یکدیگر برابر نیستند. هدف اصلی *CRM*، بهینه‌سازی نسبت وقایع سودآور به وقایع زیانبار برای گروه معلومی از مشتریان است. برخی وقایع خاص مانند فروش محصول، ایجاد درآمد و بعضاً تولید سود می‌کنند درحالی‌که برخی دیگر مانند تماسهای تلفنی و دیگر موارد موارد مشابه به منظور پشتیبانی این‌گونه نیستند. در این‌گونه موارد هدف داده‌کاوی افزایش درآمد و کاهش هزینه می‌باشد. چیزی که باید سازمانها بدانند این است که مشتریان به چه شیوه‌ای و چگونه تمایل به تعامل دارند تا بتوانند وفاداری مشتری را کسب نموده و به طبع سودآوری خود را نیز بهبود بخشند.

مدیریت ارتباط با مشتری به سازمانها این اجازه را می‌دهد که مشتریان خود را بهتر بشناسند و تفاوت بین آنها را بهتر درک نمایند. در نتیجه در تخصیص منابع به مشتریان مطلوب‌تر، کارآمدی بیشتری داشته باشند. از طریق تلاشهای *CRM*، سازمانها می‌توانند هماهنگی بهتری در ارتباط با مشتری ایجاد نمایند، بنابراین سازمان می‌تواند مدیریت مؤثرتری برروی منابع بازاریابی و ارتباط بامعناتری با مشتریانش داشته باشد. ارتباط کارا با مشتری نیاز به درک الزامات این رابطه دارد. توانایی ارائه خدمات شخصی شده، ایجاد ارزش و پذیرش دوطرفه، تعهد به ارتباط متقابل و موارد مشابه همه در ایجاد ارتباط قوی تأثیر به‌سزایی دارند

معماری مدیریت ارتباط با مشتری

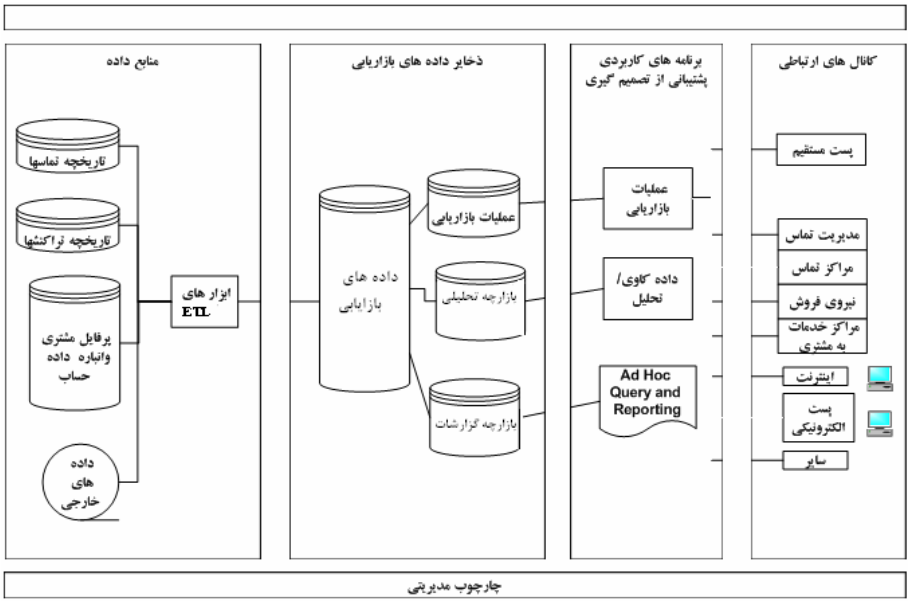
ازدید معماری، کل چارچوب کاری *CRM* به سه مؤلفه اصلی تقسیم می‌شود:

- *CRM* عملیاتی[1]: عبارتست ازخودکارسازی فرآیندهای کسب‌وکاری افقی، شامل نقاط تماس مشتری، کانالها و یکپارچه کردن موضوعات پشت صحنه و با موارد قابل مشاهده در روی صحنه.

[1]- Operational CRM

- *CRM* تحلیلی[1]: تحلیل داده‌های ایجاد شده توسط *CRM* عملیاتی می‌باشد.
- *CRM* مشارکتی[2]: برنامه‌های کاربردی خدمات مشارکتی شامل موارد زیر می‌باشد: پست الکترونیکی، انتشارات شخصی شده، ارتباطات الکترونیکی، رسانه‌های مشابهی که برای تسهیل تعامل بین مشتری و سازمان طراحی شده است.

همان‌طور که در شکل (۱۰-۱) مشاهده می‌کنید معماری *CRM* شامل نقاط تماس با مشتری و کانالهای ارسال می‌باشد که اطلاعات را تولید و مصرف می‌نمایند. این اطلاعات نیاز به یکپارچگی و تحلیل به‌منظور تعیین تصویر کامل و دقیق از مشتریان، عملکرد، نیازها، شکایات و مشخصه‌های آنها دارد [۳].



شکل ۱۰-۱) معماری CRM

[1]- Analytical CRM
[2]- Collaborative CRM

یافتن مشتریان احتمالی

هر یک از اهداف کسب و کار به فنون داده‌کاوی مناسب با آن مسئله مربوط می‌شوند. مباحث کسب و کار مورد بررسی متناسب با به ترتیب پیچیدگی رابطه مشتری می‌باشد. این بحث در ارتباط با مشتریان احتمالی[1] که کمتر شناخته شده‌اند شروع شده و تا فرصتهای متنوع بیان شده توسط روابط مستمر مشتری ادامه می‌یابد. موضوعات مورد نظر می‌تواند شامل محصولات، کانالهای ارتباطی متعدد و تعاملات فردی فزاینده باشد[٤].

به‌نظر می‌رسد جستجوی مشتریان احتمالی، شروع مناسبی برای بحث در مورد کاربردهای داده‌کاوی در این مبحث باشد. در بازاریابی، یک مشتری احتمالی کسی است که اگر به روش درستی با او برخورد شود، انتظار می‌رود به یک مشتری خوب تبدیل شود.

برای بیشتر کسب و کارها، تعداد کمی از شش میلیارد مردم زمین، مشتری احتمالی هستند. می‌توان بر اساس محل، سن، توانایی مالی و نیاز به محصول یا خدمت، اکثر مردم را کنار گذاشت. به‌عنوان مثال شرکتی که صندلی تاب حیاط می‌فروشد، قاعدتاً کاتالوگ خود را به خانوارهای دارای فرزند در مناطقی که حیاط خلوت دارند، ارسال می‌کند. یک مجله، گروهی را هدف‌گیری می‌کند که آن را مطالعه کرده و به آگهی‌هایش علاقه‌مند باشند.

داده‌کاوی می‌تواند نقش مهمی در یافتن مشتریان احتمالی داشته باشد. مهم‌ترین نقشهای آن عبارتند از:

- تشخیص مشتریان احتمالی خوب
- انتخاب کانال ارتباطی مناسب به منظور دسترسی به مشتریان احتمالی
- انتخاب پیام مناسب برای گروه‌های مختلف مشتریان احتمالی

---

[1]- Prospects

داده‌کاوی تا کنون بیشتر در تشخیص مشتریان احتمالی خوب استفاده شده است.

تشخیص مشتریان احتمالی خوب

بنابر ساده‌ترین تعریف، « مشتری احتمالی خوب » کسی است که ممکن است علاقه به مشتری شدن نشان دهد. مشتریان احتمالی خوب نه فقط علاقه‌مند هستند تا مشتری شوند، بلکه استطاعت مالی مشتری شدن را نیز داشته و به‌عنوان مشتری سودآور خواهند بود. این مشتریان به احتمال زیاد صورتحساب‌هایشان را به‌موقع پرداخته، احتمال کمی دارد که سر شرکت کلاه بگذارند، اگر با آنها درست رفتار شود مشتریان وفاداری[1] خواهند بود و دیگران را نیز همراه خواهند کرد. جدا از سادگی و پیچیدگی تعریف مشتری احتمالی خوب، اولین وظیفه هدف‌گیری آنها است.

اگر قرار است پیام از طریق تبلیغ یا کانالهای مستقیم‌تر مانند پست، تلفن یا پست الکترونیک انتقال یابد، هدف‌گیری[2] بسیار مهم است. پیامهای تابلوهای تبلیغاتی تا حدودی هدف‌گیری شده‌اند. مثلاً تابلوهای تبلیغاتی خطوط هوایی و شرکتهای اجاره ماشین معمولاً در کنار بزرگراههای منتهی به فرودگاهها دیده می‌شوند، جایی که احتمالاً در بین رانندگان این راهها، استفاده‌کنندگان این خدمات وجود خواهند داشت.

برای به‌کاربردن داده‌کاوی در این مسئله، باید اول مشتری احتمالیِ خوب را تعریف کرده و سپس قواعد را یافت که اجازه هدف‌گیری مشتریان دارای خصوصیات مورد نظر را می‌دهد. برای اکثر شرکتها، اولین قدم به منظور استفاده از داده‌کاوی برای تشخیص مشتریان احتمالی خوب، ساختن یک مدل پاسخ[3] است.

---

[1]- Loyal
[2]- Targeting
[3]- Response Model

انتخاب کانال ارتباطی

یافتن مشتریان احتمالی نیاز به ارتباط دارد. شرکتها به عمد از راههای مختلفی با مشتریان احتمالی تماس می‌گیرند. یکی از راه‌ها، روابط عمومی است که هدف آن تشویق رسانه‌ها برای پوشش داستانی درباره شرکت و نیز انتشار پیامهای مثبت افواهی می‌باشد. این کار برای برخی شرکتها بسیار مؤثر است ولی پیامهای بازاریابی مستقیم با روابط عمومی متفاوتند.

در اینجا تبلیغات و بازاریابی مستقیم مورد نظر است. تبلیغات می‌توانند روی جلد مجله، پنجره‌های باز شده مزاحم در برخی سایتهای تجاری، زیرنویسهای تلویزیونی در حین وقایع ورزشی مهم یا تبلیغات محصول در فیلمها باشد. این نوع تبلیغات، گروه-های مردم را بر اساس صفات مشترک هدف‌گیری می‌کند ولی پیام را برای هر فرد مشخص نمی‌کند. در بخشهای آتی، از طریق تطابق پروفایل[1] مشتریان احتمالی با پروفایل یک ناحیه جغرافیایی، به‌منظور انتخاب محل مناسب تبلیغ بحث می‌شود.

پروفایل مشتری

اگر مشخصات عمومی مشتری مانند سن، جنسیت و آدرس با مشخصات مشتریان دیگر مقایسه شود، شرکت را قادر به شناسایی نوع افرادی می‌کند که محصولاتش را می‌خرند. این مشخصات به شرکت کمک می‌کند که محصولات دیگری برای همان گروه تولید کند یا استراتژیهای متفاوتی برای فروختن همان محصولات به بازارهای هدف دیگر توسعه دهد.

بازاریابی مستقیم، اجازه می‌دهد پیام برای هر فرد، شخصی شود. این کار می‌تواند از راه تماس تلفنی (مثل *SMS*)، پست الکترونیکی، کارت پستال و یا کاتالوگ رنگی گلاسه

[1]- Profile

باشد. داده‌کاوی می‌تواند به تعیین کانالهای مؤثر برای هر گروه از مشتریان احتمالی کمک کند.

انتخاب پیامهای مناسب

حتی برای فروختن محصول یا خدمتی یکسان، پیامهایی متفاوت برای افراد مختلف مناسب است. مثلاً، ممکن است یک روزنامه برای یکی به دلیل پوشش اخبار ورزشی و برای دیگری به خاطر پوشش اخبار سیاسی یا هنری جذاب باشد. وقتی محصول دارای تنوع بوده یا چند محصول پیشنهاد می‌شود، انتخاب پیام مناسب مهم‌تر هم می‌شود.

حتی در یک محصول نیز پیام مهم است. یک مثال کلاسیک، تعامل میان معیار هزینه و معیار راحتی است. برخی مردم به قیمت خیلی حساس بوده و تمایل دارند از تعاونی خرید کرده، نصفه شب تلفن کنند، بین مسیر هواپیما عوض کنند و سفرهایشان شامل شب آخر هفته باشد. دیگران حاضرند برای خدمات راحت‌تر، مبالغ اضافه بپردازند. پیامی بر مبنای قیمت کمتر، نه فقط در انگیزش راحت‌طلبان شکست می‌خورد بلکه ممکن است خطر راندن آنان به سمت محصولات کم‌سودتر را داشته باشد، درحالی‌که آنان تمایل دارند پول بیشتری خرج کنند.

مدلهای پاسخ که شامل یک فعالیت تبلیغی[1] هستند، می‌توانند با هم ترکیب شده تا بهترین پیشنهاد را به مشتری بدهند. برای دسته‌بندی مشتریان به بخشهای دارای تشابه فکری در پاسخگویی به پیشنهادات، می‌توان از فیلترکردن مشارکتی استفاده کرد.

داده‌کاوی برای انتخاب محل مناسب تبلیغ

یک راه هدف‌گیری مشتریان احتمالی، نگاه کردن به مشتریان فعلی است. برای مثال یک نشریه کشوری از طریق پرسشنامه، مشخصات زیر را برای خوانندگانش به‌دست آورده است:

- ۵۸٪ تحصیلات عالی داشتند.

[1]- Campaign

- ٤٦٪ مشاغل تخصصی یا مدیریتی داشتند.
- ۲۱٪ درآمد خانواری بیش از ۸ میلیون تومان در سال داشتند.
- ۷٪ درآمد خانواری بیش از ۱۲ میلیون تومان در سال داشتند.

درک این پروفایل از دو طریق می‌تواند به نشریه کمک کند. اول اینکه با هدف‌گیری مشتریان احتمالی مطابق با این پروفایل، می‌توان نرخ پاسخ به فعالیتهای ترویجی[1] خود را افزایش داد. دوم، می‌توان فضای تبلیغاتی نشریه را به شرکتهای علاقه‌مند به این نوع خوانندگانِ تحصیل‌کرده و پردرآمد فروخت. از آنجا که موضوع این بخش، هدف‌گیری مشتریان احتمالی است، بیایید ببینیم چگونه این نشریه از این پروفایل برای متمرکز کردن فعالیتهای مشتری‌یابی خود استفاده کرده است. ایده اصلی ساده است. وقتی نشریه می‌خواهد در روزنامه آگهی بدهد، باید به دنبال روزنامه‌هایی باشد که شنوندگانشان مطابق این پروفایل باشند. باید در جاهایی کارتهای معرفی خود را روی پیشخوان مغازه‌ها بگذارد که منطبق بر این پروفایل باشند. وقتی می‌خواهد بازاریابی با پیامک (*SMS*) انجام دهد باید با مردمی مطابق این پروفایل تماس بگیرد. چالش داده‌کاوی، ارائه تعریف مناسبی از معنای انطباق با این پروفایل است.

چه کسی با این پروفایل مطابقت دارد؟

یک راه تعیین تطابق مشتری با پروفایل مورد نظر، اندازه‌گیری شباهت یا فاصله آان دو است. بسیاری از فنون داده‌کاوی از ایده اندازه‌گیری شباهت به‌عنوان فاصله استفاده می‌کنند. برای مثال، مدل استدلال بر مبنای حافظه[2]، روشی برای دسته‌بندی مشاهدات بر پایه دسته‌های مشاهده شده‌ای می‌باشد که در همان همسایگی وجود دارند. کشف خوشه خودکار، فن داده‌کاوی دیگری است که بر اساس امکان محاسبه فاصله بین دو مشاهده برای یافتن گروه مشاهدات شبیه به‌هم کار می‌کند. در این مثال، به دنبال

---

[1]- Promotional
[2]- Memory Based Reasoning

تعریف معیار فاصله‌ای برای درجه تطابق مشتری احتمالی با پروفایل موجود هستیم. داده‌ها شامل نتایج پرسشنامه بوده و نشان‌دهنده مشترکین در یک مقطع زمانی می‌باشند. در اینجا با این پرسشها روبرو هستیم: چه نوع معیارهایی مناسب این داده‌ها هستند؟ پروفایل‌ها چه نیازهایی را برآورده می‌کنند؟

دو فرد شرکت‌کننده در تحقیق را در نظر بگیرید. مریم، تحصیل‌کرده بوده، ۹ میلیون تومان در سال درآمد داشته و متخصص است. داوود دیپلمه بوده و ٤ میلیون تومان در سال درآمد دارد. کدام یک بیشتر با پروفایل خوانندگان تطابق دارند؟ جواب، بسته به نحوه مقایسه متفاوت است. جدول (۱۰-۱) راه مناسبی را برای محاسبه امتیاز از روی پروفایل و معیار فاصله نشان می‌دهد.

این جدول امتیازی بر پایه نسبت توافق مخاطب با هر خصوصیت را محاسبه می‌کند. برای مثال چون ۵۸٪ خوانندگان تحصیل‌کرده‌اند، مریم امتیاز ۰,۵۸ برای این خصوصیت می‌گیرد. داوود که دیپلمه است امتیاز ۰,٤۲ می‌گیرد، زیرا ٤۲٪ خوانندگان نیز تحصیلات عالی ندارند. این امتیاز برای هر خصوصیت محاسبه شده و امتیازها جمع می‌شوند. امتیاز مریم ۲,۱۸ و امتیاز داوود ۲,٦۸ می‌شود. امتیاز بالاتر داوود نشان می‌دهد او از مریم به خوانندگان فعلی شبیه‌تر است.

جدول ۱۰-۱) محاسبه امتیازهای تطابق برای افراد با مقایسه معیارهای دموگرافیک

	خوانندگان نشریه	امتیاز بله	امتیاز خیر	مریم	داوود	امتیاز مریم	امتیاز داوود
تحصیل‌کرده	٪۵۸	۰,۵۸	۰,٤۲	بله	خیر	۰,۵۸	۰,٤۲
متخصص یا مدیر	٪٤٦	۰,٤٦	۰,۵٤	بله	خیر	۰,٤٦	۰,۵٤
درآمد بیشتر از ۸	٪۲۱	۰,۲۱	۰,۷۹	بله	خیر	۰,۲۱	۰,۷۹
درآمد بیشتر از ۱۲	٪۷	۰,۰۷	۰,۹۳	خیر	خیر	۰,۹۳	۰,۹۳
جمع						۲,۱۸	۲,٦۸

مشکل این روش این است که با اینکه طبق این امتیازدهی داوود مناسب‌تر از مریم به نظر می‌رسد ولی در واقع مریم به مخاطبین هدف نشریه یعنی افراد تحصیل‌کرده و پردرآمد شبیه‌تر است. موفقیت این هدف‌گیری از مقایسه پروفایل خوانندگان با خصوصیات جمعیت‌شناسی[1] معلوم می‌شود. این موضوع لزوم برخوردی پخته‌تر با اندازه‌گیری تطابق فرد با مخاطبین نشریه از طریق در نظر گرفتن خصوصیات جمعیت عمومی علاوه بر خصوصیات خوانندگان را ایجاب می‌کند. این روش، درجه تفاوت یک مشتری احتمالی از جمعیت عمومی و تفاوت خواننده از جمعیت عمومی را محاسبه کرده و سپس شباهت این دو تفاوت را اندازه می‌گیرد.

دموگرافی

مطالعه آماری جمعیت (جمعیت شناسی) شامل خصوصیاتی مانند توزیع جغرافیایی، محیط فیزیکی، بیماری، ترکیب جنسیتی و سنی، و نرخ تولد و مرگ.

خواننده نشریه در مقایسه با جمعیت عمومی تحصیل‌کرده‌تر، متخصص‌تر و پردرآمدتر است. درجدول (۱۰-۱) ستونهای شاخص از تقسیم درصد خوانندگانی که ویژگی خاصی دارند بر درصد جمعیتی که دارای این ویژگی است به‌دست آمده‌اند. می‌توان خصوصیات خواننده را با جمعیت عمومی از طریق این شاخصها مقایسه کرد. دیده می‌شود که خواننده نشریه تقریباً سه برابر جمعیت عمومی تحصیل‌کرده است. به‌طور مشابه، خواننده تقریباً نصف جمعیت عمومی تحصیل‌نکرده است. با استفاده از شاخصها به‌عنوان امتیاز هر خصوصیت، مریم امتیاز ۸/۴۲(=۰/۹۵+۲/۲۱+۲/۴۰+۲/۸۶) می‌گیرد در حالی که امتیاز داوود فقط ۳/۰۲(=۰/۹۵+۰/۸۷+۰/۶۷+۰/۵۳) می‌شود. امتیازهای بر پایه شاخصها با توجه به جمعیت مخاطب هدف، متناسب‌تر می‌باشند. این امتیازها با

[1]- Demographic

معناتر هستند زیرا شامل اطلاعات اضافه تفاوت مخاطبین هدف با جمعیت عمومی می‌باشند.

شاخصها به جای مقادیر خام

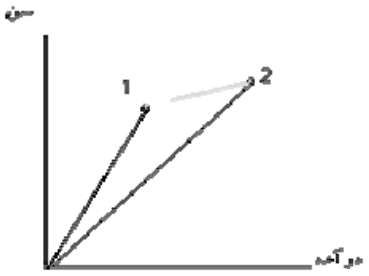
در مقایسه پروفایل مشتری، در نظر گرفتن پروفایل جمعیت عمومی مهم است. برای همین، استفاده از شاخصها اغلب از مقادیر خام بهتر است.

جدول ۱۰-۲) محاسبه امتیازها با در نظر گرفتن نسبتها در جمعیت

	بله			خیر		
	خوانندگان نشریه	جمعیت عمومی	شاخص	خوانندگان نشریه	جمعیت عمومی	شاخص
تحصیل‌کرده	٪۵۸	٪۲۰,۳	۲,۸٦	٪٤۲	٪۷۹,۹	۰,۵۳
متخصص یا مدیر	٪٤٦	٪۱۹,۲	۲,٤۰	٪۵٤	٪۸۰,۸	۰,٦۷
درآمدبیشتر از ۸	٪۲۱	٪۹,۵	۲,۲۱	٪۷۹	٪۹۰,۵	۰,۸۷
درآمدبیشتر از ۱۲	٪۷	٪۲,٤	۲,۹۲	٪۹۳	٪۹۷,٦	۰,۹۵

مفهوم تشابه بر پایه زاویه و تفاوت بر پایه فاصله

می‌توان مفهوم تشابه را بر پایه تفاوت دو زاویه توضیح داد. هر ویژگی اندازه‌گیری شده یک بُعد جدا در نظر گرفته می‌شود. با در نظر گرفتن مقدار متوسط هر ویژگی به‌عنوان مبدأ، پروفایل خوانندگان فعلی، برداری است که نشان می‌دهد خواننده نوعی، چقدر و در چه جهتی از جمعیت عمومی دور است. داده‌های یک مشتری احتمالی نیز یک بردار است. اگر زاویه بین این دو بردار کوچک باشد، آن گاه مشتری احتمالی نیز در همان جهت پروفایل خوانندگان با جمعیت عمومی تفاوت دارد.



شکل ۱۰-۲) شباهت (زاویه) و تفاوت (فاصله)

در شکل ۱۰-۲)، دو نفر (۱ و ۲) از نظر سن و درآمد مقایسه شده‌اند. می‌توان شباهت بین این دو را از طریق زاویه (کسینوس) بین دو بردار سنجید. ولی اگر یکی از آنها دقیقاً دو برابر دیگری سن و درآمد داشته باشد چه طور؟ در این صورت بردار یکی روی دیگری قرار می‌گیرد و شباهت ۱۰۰٪ می‌شود! پس معیار شباهت همیشه مناسب نیست و در بسیاری از موارد بهتر است از تفاوت دو نقطه ۱ و ۲ بر حسب معیار فاصله استفاده شود.

نرمال کردن

به نظر شما آیا اندازه‌گرفتن درآمد بر حسب تومان یا ریال باید تأثیری در شباهت یا تفاوت دو نفر بگذارد؟ برای جلوگیری از این مشکلات معمولاً ابتدا داده‌های هر مشخصه مانند درآمد طوری نرمال می‌شوند که در یک مقیاس قرارگیرد. برای مثال درآمد کلیه افراد را تقسیم بر حداکثر درآمد ممکن می‌کنیم تا کلیه درآمدها بین صفر تا یک قرارگیرند.

مشخصه‌های غیرعددی

چگونه مشخصه‌ای مانند جنسیت را در شباهت در نظر بگیریم؟ کافی است بنا بر قرارداد بگوییم اگر دو نفر هم‌جنس باشند، تفاوت آنها صفر و اگر دارای جنسیت مخالف باشند تفاوت آنها یک است. در این صورت با توجه به اینکه مشخصه‌های عددی را نیز نرمال کرده‌ایم (در فاصله صفر–یک) می‌توان همه مشخصه‌ها را در یک مقیاس در تفاوت یا شباهت در نظر گرفت.

اندازه‌گیری مطابقت برای گروه‌های مشتریان

محاسبه امتیازهای بر پایه شاخص، قابل توسعه به گروه‌های بزرگ‌تر مردم است. اهمیت این موضوع در آن می‌باشد که ممکن است ویژگی خاص مورد استفاده برای هر مشتری یا مشتری احتمالی در دسترس نباشد. خوشبختانه و نه بر حسب تصادف، خصوصیات مطرح شده قبلی همگی خصوصیات جمعیت‌شناسی موجود در سرشماری آماری بوده و قابل اندازه‌گیری بر حسب نواحی جغرافیایی مانند منطقه سرشماری می‌باشند.

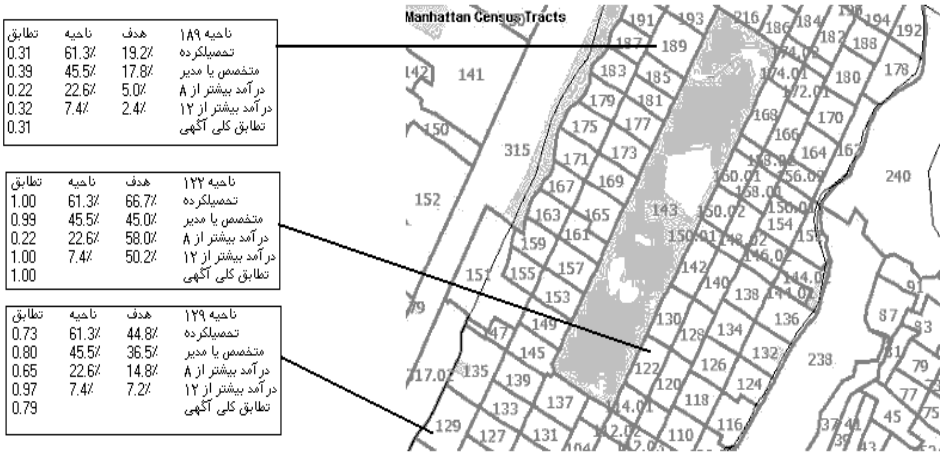
داده‌های هر منطقه سرشماری

یکی از فروض بازاریابی بر پایه ضرب‌المثل قدیمی "کبوتر با کبوتر، باز با باز، کند هم‌جنس با هم‌جنس پرواز. " می‌باشد. یعنی مردم دارای علایق و سلیقه‌های مشابه در نواحی مشابه زندگی می‌کنند (داوطلبانه یا به دلیل الگوهای تاریخی تمایز). با توجه به این فرض، معامله با مردمی که قبلاً در میانشان و یا در نواحی مشابه مشتری داشته‌اید، ایده خوبی است. آمار سرشماری، هم برای یافتن مکانهای تمرکز مشتری و هم برای تعیین پروفایل نواحی مشابه، باارزش است.

فرایند مورد نظر، نرخ‌گذاری هر منطقه سرشماری با توجه به تطابق آن با نشریه می‌باشد. ایده اصلی، تخمین نسبت هر منطقه سرشماری مطابق با پروفایل خوانندگان

نشریه است. برای مثال اگر یک منطقه سرشماری دارای جمعیت بزرگسالی با ۵۸٪ تحصیل‌کرده باشد باز هم امتیاز مریم، ۱ یعنی تطابق کامل است. اگر فقط ۵,۸٪ تحصیل‌کرده باشند، آنگاه امتیاز تطابق این خصوصیت ۰,۱ است. امتیاز تطابق کل، متوسط امتیازهای هر خصوصیت است.

شکل (۱۰-۳) مثالی از سه منطقه سرشماری را نشان می‌دهد. هر منطقه دارای نسبت متفاوتی از چهار خصوصیت مورد نظر است. این داده‌ها را می‌توان برای محاسبه امتیاز تطابق کلی هر منطقه ترکیب نمود. توجه کنید که همه ساکنان آن منطقه، امتیاز یکسانی می‌گیرند. این امتیاز بیانگر نسبتی از جمعیت آن منطقه است که مطابق پروفایل مورد نظر می‌باشد.



شکل ۱۰-۳) مثالی از محاسبه تطابق با خوانندگان در سه منطقه سرشماری مانهاتان

استفاده از مشتریان فعلی برای یادگیری در مورد مشتریان احتمالی

یک راه مناسب برای یافتن مشتریان احتمالی، نگاه به همان مکانهایی است که بهترین مشتریان فعلی از آن جا می‌آیند. بنابراین باید راهی برای تشخیص بهترین مشتریان فعلی داشت. لازمه این کار، نگهداری سابقه نحوه دستیابی به مشتریان فعلی و وضعیت آنان در زمان دستیابی می‌باشد.

البته تکیه به مشتریان فعلی برای یادگیری در مورد مشتریان احتمالی، خطر انعکاس تصمیمات بازاریابی گذشته را دارد. مطالعه مشتریان فعلی، پیشنهادی برای جستجوی مشتریان احتمالی در مکانهای جدید نمی‌دهد. با این حال، عملکرد فعلی راه مناسبی برای ارزیابی کانالهای مشتری‌یابی موجود است. برای یافتن مشتریان احتمالی، مهم است بدانیم وقتی مشتریان فعلی، خودشان زمانی مشتری احتمالی بوده‌اند، چگونه به نظر می‌رسیدند. به‌طور ایده‌آل باید:

- ردگیری مشتریان را قبل از مشتری شدن شروع کرد.
- اطلاعات مشتریان جدید را در حین مشتری شدن آنان جمع کرد.
- رابطه بین داده‌های زمان مشتری شدن و نتایج مورد نظر آتی را مدل کرد.
- بخشهای بعدی، این بحث را تکمیل می‌کنند.

شروع ردگیری مشتریان قبل از مشتری شدن

خوب است اطلاعات مشتریان احتمالی قبل از مشتری شدن، ثبت شود. سایتهای وب می‌توانند با ایجاد یک کوکی[1] در اولین بازدید مشتری از سایت و سپس پروفایل کردن این مشتری بی‌نام با ثبت کارهایی که انجام می‌دهد، به این منظور برسند. وقتی بازدیدکننده از طریق همان برنامه پویشگر وب مانند *IE* و روی همان کامپیوتر قبلی، دوباره به سایت رجوع می‌کند، این کوکی شناخته شده و پروفایل به‌روز می‌شود. وقتی این بازدیدکننده تبدیل به مشتری یا کاربر ثبت‌نام کرده می‌شود، فعالیت منجر به انتقال بخشی از سابقه مشتری می‌گردد.

---

[1]- Cookie

ردگیری پاسخها و پاسخ‌دهندگان در دنیای غیر وِبی نیز کار خوبی است. اولین اطلاعات مهم برای ثبت، پاسخ یا عدم پاسخ مشتری می‌باشد. مشخصات کسانی که پاسخ داده‌اند و آنهایی که پاسخ نداده‌اند، جزء لاینفک مدلهای پاسخ است. در صورت امکان، داده‌های اقدام بازاریابی محرک پاسخ، کانال دریافت پاسخ و زمان پاسخ را نیز باید ثبت کرد.

تشخیص پیام بازاریابی محرک پاسخ از میان پیامهای متعدد، می‌تواند دشوار و یا غیرممکن باشد. برای آسان کردن این تشخیص، فرمهای پاسخ و کاتالوگها دارای کُدهای تشخیص مناسب هستند. وب سایتها آدرس سایتی را که از آن مراجعه شده، ثبت می‌کنند. حتی می‌توان عملیات تبلیغات را از طریق تلفنهای جدا، بسته‌های پستی و یا آدرس سایتهای مختلف تفکیک نمود. بسته به طبیعت محصول یا خدمت، ممکن است پاسخ‌دهندگان ملزم به ارائه اطلاعات اضافه در فرم محصول یا ثبت نام باشند. اگر انجام خدمت مستلزم داشتن اعتبار باشد، اطلاعات اعتبار درخواست می‌شود. اطلاعات جمع‌آوری شده در ابتدای ارتباط با مشتری از هیچ گرفته تا معاینه کامل پزشکی برای بیمه عمر متغیر است. اکثر شرکتها جایی در وسط هستند.

جمع‌آوری اطلاعات از مشتریان جدید

وقتی یک مشتری احتمالی برای اولین بار مشتری می‌شود، فرصتی طلایی برای جمع‌آوری اطلاعات بیشتری پیش می‌آید. داده‌های مربوط به مشتری احتمالی قبل از تبدیل شدن به مشتری، از نوع جغرافیایی و یا دموگرافیک هستند. بعید است لیستهای خریداری شده شامل چیزی غیر از نام، آدرس تماس و منبع لیست باشند. با داشتن آدرس می‌توان اطلاعات دیگری در مورد مشتریان احتمالی بر پایه خصوصیات همسایگان به دست آورد. با داشتن نام و آدرس با هم می‌توان اطلاعات سطح خانوار را از فراهم‌کنندگان داده‌های بازاریابی خرید. این نوع از داده‌ها برای هدف‌گیری بخشهای عمومی مانند «مادران جوان» یا « نوجوانان حومه » مفید است ولی برای ایجاد رابطه فردی با مشتری به اندازه کافی دارای جزئیات نیست.

مفیدترین مشخصات قابل جمع‌آوری برای داده‌کاوی آتی، تاریخ خرید اولیه، کانال خرید اولیه، پیشنهاد پاسخ، محصول اولیه، امتیاز اعتبار اولیه، مدت تا پاسخ و محل جغرافیایی است. این مشخصات در عمل پیش‌بینی کننده خوبی برای مواردی مانند مدت مورد انتظار رابطه، بدحسابی[1] و خریدهای اضافی هستند.

پیش‌بینی نتایج آتی بر اساس متغیر زمان مشتری شدن

با ثبت همه اطلاعات ممکن در زمان مشتری شدن و سپس ردگیری مشتریان در طول زمان، می‌توان از داده‌کاوی برای مرتبط کردن متغیرهای زمان خرید به نتایج آتی مانند طول عمر مشتری، ارزش مشتری و ریسک پیش‌فرض استفاده کرد. این اطلاعات را می‌توان برای هدایت تلاشهای بازاریابی و تمرکز روی کانالها و پیامهای مؤثر (دارای بهترین نتایج) به کار برد. کشف اینکه برخی کانالها، مشتریانی با طول عمر دو برابر کانالهای دیگر می‌یابند، غیر معمول نیست. با فرض امکان تخمین ارزش ماهانه مشتری و داشتن طول عمر، می‌توان ارزش ریالی مشتری هر کانال را محاسبه کرد. برای نرخ‌گذاری کانالها، ارزش ریالی مشتری به اندازه هزینه تماس، مهم است.

داده‌های مشتریان

اگر بخواهید از مشتری تصویر درستی داشته باشید باید از جهات مختلف به مشتری بنگرید و همه اطلاعات درباره مشتری را از بخش‌های مختلف سازمان در کنار هم بگذارید، دراین صورت است که این تصویر می‌تواند خصوصیات و رفتار صحیح مشتری را نمایان سازد. برای مثال چون بخش فروش با مشتری ارتباط برقرار می‌کند، تصویری از نیازهای مشتری را دریافت می‌کند. زمانی که مشتری برای حل مشکلی با بخش خدمات تماس می‌گیرد در واقع تصویر دیگری از خود را به سازمان نمایان می‌سازد (رضایت یا عدم رضایت). هر کدام از این نقاط تماس[2] فرصتی برای تعامل هستند و سازمان می‌تواند از طریق آنها با مشتریانش در تماس باشد یا مشتریان با

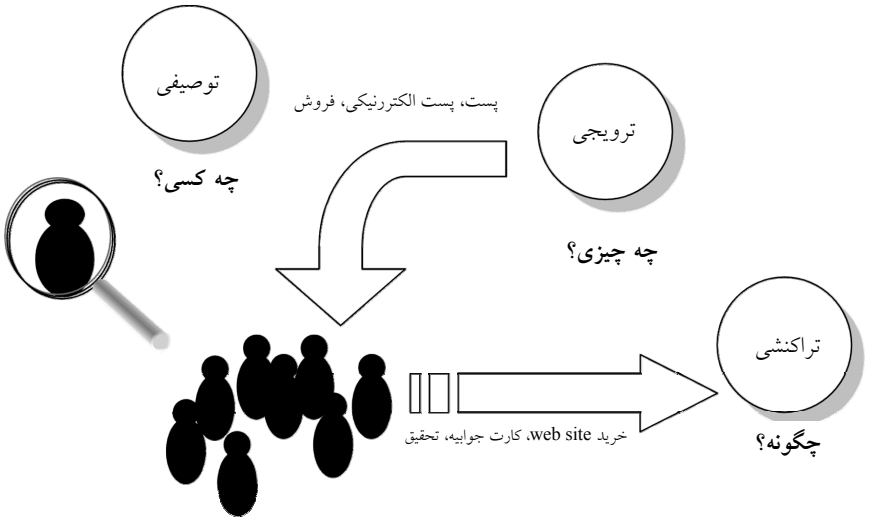
---

[1]- Bad Dept
[2]-Touch Point

سازمان در تماس باشند، به همین دلیل به آنها نقاط تماس گفته می‌شود. در واقع نقاط تماس بهترین اطلاعات را درباره مشتریان تامین می‌کنند. ولی چون این اطلاعات به‌صورت پراکنده و مجزا وجود دارد، هر قسمت به تنهایی برای ارائه یک تصویرکامل از مشتری کافی نیست. درنتیجه یکپارچگی این داده‌ها که از منابع مختلفی می‌آیند در ارائه تصویر کامل از مشتری بسیار حائز اهمیت است [۱].

داده‌های مرتبط با مشتریان را می‌توان از منابع مختلفی به‌دست آورد شکل (٤-۱۰). در سیستم‌های داده‌کاوی مدیریت ارتباط با مشتری سه نوع اصلی داده وجود دارد که عبارتند از:

- توضیح اینکه مشتری کیست.
- توضیح اینکه چه بازاریابی یا تبلیغ فروشی برای مشتری انجام شده است.
- توضیح اینکه مشتری در مقابل این تبلیغات چه واکنشی نشان داده است.



شکل ۱۰-٤) برای فراهم کردن اطلاعات کافی برای داده‌کاوی باید به دنبال چه کسی، چه چیزی و چگونه باشید.

اگر شما این سه چیز درباره مشتریان یا حتی درباره افرادی که هنوز مشتری شما نیستند بدانید، در واقع داده کافی برای شروع پیش‌بینی را دارید. شما می‌توانید با کمک داده‌کاوی الگوهایی را از میان داده‌ها کشف کنید یا حتی با بهره‌گیری از این تجربیات،

تراکنش‌های بازاریابی و فروش با این مشتریان را بهینه کنید. بدون دانستن اینکه مشتری کیست، چه کاری انجام می‌دهد و واکنش آن چیست، بهینه‌سازی یا بهبود سیستم امکان‌پذیر نمی‌باشد.

برای داشتن سودآورترین تراکنش با مشتری تا جای ممکن و بهینه کردن عملکرد سیستم *CRM*، باید توانایی تفکیک مشتریان خوب و بد و مشتریان سودآور و غیرسودآور را داشته باشید. باید بدانید آنها چه کسانی هستند و چه تفاوتهایی با هم دارند.

به‌طور مشابه برای اینکه بدانید در تبلیغات و بازاریابی چگونه سرمایه‌گذاری کنید، نیاز به دانستن کارهایی که برای هر مشتری انجام می‌دهید و نتایج پیگیری این کارها دارید. شما باید تعداد زیادی تجربه‌های کوچک در امور تبلیغاتی و بازاریابی با مشتریان اصلی‌تان داشته باشید، و توجه کنید که تفاوت در هرتجربه بهترین راه برای پی بردن به کارهایی که باید انجام دهید و نباید انجام دهید، می‌باشد. برای یافتن ارزش واقعی سیستم، باید بتوانید نتایج را اندازه‌گیری کنید. اگر مشخص نشود که نتیجه تجربه خوب یا بد بوده است، پس واقعا چیز جدیدی که بتواند برای بهبود سیستم در دفعات بعد استفاده شود، حاصل نشده است. گروه‌بندی دسته‌های مختلف تجربه‌ها نیز مفید است چون باعث شکسته شدن داده‌های ذخیره شده در بانک اطلاعاتی به انواع مختلف داده می‌شود. همچنین این دسته‌بندیها توصیف خوبی از منابع داده ارائه می‌دهند. قراردادن داده در این سه دسته به شما برای داده‌کاوی موفق و تولید یک سیستم بهینه *CRM* کمک شایانی خواهد کرد.

داده توصیفی

داده توصیفی شامل توضیحاتی درباره مشتری یا مصرف‌کننده است. این نوع داده‌ها معمولاً به‌طور خلاصه در ستونهای مختلف جدول اطلاعات مشتریان در بانک اطلاعاتی ذخیره می‌شود. این‌گونه داده‌های توصیف‌کننده مشتری شامل اطلاعاتی مثل سن،

جنسیت، موقعیت منزل، تعداد فرزندان، درآمد خانگی و درآمد فردی هستند. این اطلاعات قابل تغییر هستند ولی معمولاً زودتر از یکسال عوض نمی‌شود. البته اطلاعاتی مانند آدرس و تلفن نیاز به بروز رسانی فصلی یا حداقل شش ماه یکبار در بانک اطلاعاتی دارند. این داده‌ها شامل موارد کلی ذیل هستند: دموگرافی، مالی و پروفایل.

رفتار مشتری است که مهم می‌باشد

هدف داده‌کاوی در *CRM*، اصلاح نسبت رفتارهای مثبت به رفتارهای منفی است. اطلاعات ایستا یا همان توصیفی مانند سن یا کُد پستی صرفاً جانشینی برای اطلاعات واقعی مهم یعنی رفتار مشتری هستند. پس چرا فقط از داده‌های رفتاری مانند سابقه خرید مشتری استفاده نمی‌کنیم؟ زیرا در بسیاری از اوقات اطلاعات کافی از سابقه رفتاری مشتریان موجود نیست در حالی که یک مشخصه ایستا مانند کد پستی مشتری، می‌تواند پیش‌بینی کننده خوبی برای بسیاری از رفتارهای مشتری باشد.

داده تبلیغاتی

داده تبلیغاتی شامل اطلاعات کارها و فعالیتهایی است که برای مشتری صورت گرفته است. قدرت این نوع داده معمولاً به پیچیدگی سیستم *CRM* بستگی دارد. این داده‌ها شامل موارد زیر است:

- لیستی از کارهایی است که برای تبلیغات صورت گرفته است مثل پست، کاتالوگ، نمونه‌های تبلیغاتی یا کارتهای تخفیف
- تبلیغات تعاملی مثل تبلیغ در تلویزیون، رادیو، روزنامه، و مجله‌های تبلیغاتی و غیره
- اطلاعات دقیقی مثل ارسال پست الکترونیکی و تعداد کلیکهای کاربران قابل شناسایی در سایتهای وب

انواع اطلاعاتی که می‌تواند جمع‌آوری شود عبارت است از:

- نوع تعامل[1]: فروش، بازاریابی از راه دور، تبلیغات چاپی، تبلیغات رسانه‌ای، تبلیغات وبی.
- توصیف تعامل: مانند رنگ کارت پستال.
- رسانه: بازاری که تبلیغات در آن صورت می‌گیرد، وب سایتهایی که آگهی تبلیغاتی در آنها قرار دارد.
- زمان‌بندی: زمان تعامل.
- توصیف قصد و نیت: یک توصیف کامل از اینکه برای چه کسی تعامل معنی دارد و چرا؟ (مثلاً چرا این رنگ یا موسیقی زمینه باید انتخاب شود.)
- مالی: هزینه‌های ثابت و متغیر تعامل.

داده تراکنشی

به‌طورکلی داده‌های تراکنشی، داده‌هایی است که مربوط به تعامل با مشتریان می‌باشد. این داده‌ها می‌توانند هر چیزی از یک تماس تلفنی برای درخواست خدمات گرفته تا توضیح و توصیف محصولاتی که مشتری خریده‌است باشد. این داده‌ها هم مثل داده‌های تبلیغاتی، می‌تواند خیلی سریع در طول زمان تغییر کند. بنابراین طبیعی است که باید داده‌ها در ساختاری ذخیره شود که به سادگی قابل به روز رسانی و تغییر باشد. اطلاعات تراکنشی با اطلاعات توصیفی مشتریان که در طول زمان اساساً تغییر نمی‌کند متفاوت است.

تغییر آرایش داده‌های تراکنشی در فاصله زمانی کوتاهی می‌تواند خیلی چشمگیر باشد. برای مثال معرفی محصولات جدید یا مورد توجه قرارگرفتن محصولات قدیمی و فروش بیشتر آنها، الگوی محصولات فروخته شده را تغییر می‌دهد. این داده‌ها شامل موارد ذیل است: خرید، کلیک صفحه وب، تماس تلفنی، پست الکترونیک، بازدید از مغازه و پست فیزیکی.

---

[1]- Intervention

لزوم تجمیع داده‌های تراکنشی

بسیاری از روشهای داده‌کاوی نیاز به یک رکورد اطلاعاتی برای هر نمونه آموزشی (در *CRM*، هر مشتری) دارند. داده‌های ایستایی مانند سن و جنسیت برای هر مشتری فقط یک عدد در هر رکورد هستند. درحالی‌که داده‌های واقعه‌ای یا همان تراکنشی مانند سابقه خرید برای هر مشتری ممکن یک یا چند رکورد می‌باشند. برای همین لازم است این داده‌ها از نظر زمانی یا مقداری تجمیع شوند. برای مثال محاسبه شود که هر مشتری از زمان شروع خریدش تا کنون، هر سه ماه چه مبلغی خرید داشته است یا از هر محصول چه تعداد خریداری کرده است. هر مشخصه تراکنشی غیر عددی (مانند نوع محصول) یا ترتیبی، نامزد خوبی برای تجمیع است. بسته به کاربرد، راههای مختلفی برای تعریف تجمیع وجود دارد (مثلاً ماهانه یا فصلی، متوسط خرید یا حداکثر مقدار خرید) بنابراین می‌توان از روی داده‌های واقعه‌ای تعداد زیادی مشخصه برای هر مشتری ساخت. این موضوع، مسئله بُعد زیاد داده و لزوم استفاده از فنون کاهش بعد را ایجاب می‌کند که قبلاً بحث شده است.

برخی کاربردهای داده‌کاوی در مدیریت ارتباط با مشتری

داده‌کاوی ابزاری بنیادی است که برای آشکارسازی خصوصیات جمعیت‌شناختی مشتریان الزامی می‌باشد. از فنون داده‌کاوی می‌توان برای دستیابی به دامنه وسیعی از اهداف مختلف استفاده کرد. چند مثال از کاربردهای آن عبارتند از:

- شناسایی مشتریان سودآور و پروفایل آنان
- پیش‌بینی رفتار خرید مشتری
- رتبه‌بندی رویگردانی مشتری به منظور ارائه برنامه‌های مؤثر حفظ مشتری
- تمرکز تلاشهای بازاریابی بر مشتریان بالقوه‌ای که احتمال خرید کردن بیشتری دارند
- تخمین کارآمدی تبلیغات
- تخمین و اولویت‌بندی ریسک اعتباری مشتری
- برآورد میزان جدی بودن احتمالی مشتری

- فروش کناری[1] و بالاسری[2] به مشتریان بر اساس خرید محصولات قبلی
- هدفگیری مستقیم بازاریابی به سمت افرادی که بیشترین احتمال پاسخ را دارند
- پیش‌بینی کلاه‌برداری و تقلبات
- بهینه‌سازی سهم سبد خرید مشتری

به‌طور خلاصه با به‌کارگیری انواع روشهای داده‌کاوی، سازمان از مفهوم فروش صِرف به سوی خدمت‌رسانی به مشتریان حرکت می‌کند. برخی از این کاربردها به اختصار توضیح داده می‌شوند.

مدلسازی حفظ و رویگردانی

حفظ باارزش‌ترین مشتریان و دانستن اینکه کدام‌یک در خطر رویگردانی هستند می‌تواند به‌طور چشمگیری سودآوری سازمان را تحت تأثیر قرار دهد. سازمان در این مدل باید از موارد زیر آگاه باشد:

- اینکه کدام مشتریان در حال رویگردانی به سوی رقیب هستند و دلایل آن.
- اینکه کدامیک از ارزشمندترین مشتریانِ سازمان در خطر هستند.
- اینکه آیا سازمان بودجه حفظ مشتریان را برای با ارزش‌ترین مشتریان خرج می‌کند یا خیر.
- اینکه آیا سازمان راههای متناوب معنی‌دار و مؤثری به‌منظور تماس با مشتریان دارد یا خیر.

از دست دادن مشتریان برای شرکتها تا حدی قابل اجتناب است. از طریق تجزیه و تحلیل داده‌ها می‌توان مدلهایی را به منظور پیش‌بینی احتمال خروج مشتریان و احتمال جذب آنان به دیگران از طریق تبلیغات فروش و عملیات تبلیغاتی ساخت.

---

[1]- Cross-Selling
[2]- Up-Selling

با این مدلها سازمان می‌تواند با تهیه مشخصه‌های افرادی که در گذشته رویگردان شده‌اند، مشتریانی را تعیین کند که دارای بیشترین گرایش به کاهش یا قطع ارتباطند. مدل حفظ مشتری، پتانسیل یک مشتری را در ماندن با سازمان پس از رخ دادن اتفاقات احتمالی، بررسی می‌کند. مدل رویگردانی، احتمال توقف خریدهای مشتریان فعال را بررسی می‌نماید. با این اطلاعات می‌توان سیاستهای پیشگیرانه‌ای در نظر گرفت و مشتریان را به منظور توجه ویژه به آنان، فعالانه تعقیب یا علامت‌گذاری نمود.

مدلسازی پرهیز از ریسک

در این مدل سعی در پرهیز از کسب مشتریان غیرسودآور است. داده‌کاوی می‌تواند به‌طور تقریبی پیش‌بینی کند که کدام مشتریان بالقوه تبدیل به مشتریان بالفعل می‌شوند. ولی به‌طور خاص مفید است که تعیین شود کدام مشتریان سودآور خواهند بود. این موضوع مهم باید در نظر گرفته شود که به هر حال برخی مشتریان جدید بدهی خود را نخواهند پرداخت و شرکت مجبور به احتساب زیانهای وارده می‌باشد. این مدل، رفتار خرید، رفتار پرداخت، تاریخچه اعتباری و دیگر عوامل را بررسی می‌کند.

مدلسازی فروش جانبی

فروش جانبی و فروش بالاسری در *CRM* اموری محوری محسوب می‌شوند. تعامل سازمان با مشتریان فرصتی اساسی برای بازاریابی محصولات یا خدمات اضافی ایجاد می‌کند. در این زمینه می‌توان داده‌های متفاوتی را در نظر گرفت مانند اینکه مشتریان چه چیزی خریداری می‌کنند، علایق آنها چیست، چه کالاها یا خدماتی مد نظرشان است یا درباره کدام محصولات یا خدمات استعلام می‌کنند. در ابتدا باید در نظر گرفت که کدام محصولات و یا خدمات توان بالقوه‌ای در فروش جانبی دارند. این کار با قضاوت بر مبنای داده‌های فروش گذشته انجام می‌گیرد. به‌عبارت دیگر، تعیین می‌شود که یک مشتری منفرد کدام محصولات را بیشتر خریداری کرده است. سپس تعیین می‌گردد که پروفایل چه مشتریانی بیشترین تناسب را با گروه‌های متنوع محصول دارد. با این کار مشتریانی شناسایی می‌شوند که بیشترین تمایل را به خرید محصولات مشابه دارند ولی

هنوز شرکت به این‌گونه مشتریان توجهی نشان نداده است. در این حالت سازمان فرصتی عالی برای تبلیغ آن محصولات به مشتریان فعلی خود پیدا می‌کند. تمام این فرایند در محدوده *CRM* می‌باشد.

مدلسازی سودآوری

در این روش، ارزش طول عمر *(LTV)*[1] مشتری اندازه‌گیری و تنظیم می‌شود تا معلوم گردد مشتریان در دوره طول عمر خریدشان چه چیزی را خریداری می‌کنند یا پتانسیل خرید چه چیزهایی را دارند. این روش یکی از روشهای تحلیلی داده‌کاوی است که از شیوه‌های محاسبه *LTV* و استفاده از آن برای دسته‌بندی کردن مشتریان استفاده می‌کند. فهم عناصر اصلی سودآوری کمک می‌کند تا سازمان درک نماید چه زمانی مشتریان سودآور خواهند شد و آیا اصلاً این اتفاق (سودآور شدن مشتریان) روی خواهد داد یا خیر.

مدلسازی تجزیه و تحلیل اینترنتی

رفتار اینترنتی، یعنی چگونگی گشت و گذار مشتری درون صفحات اینترنتی مربوط به شرکت را می‌توان به منظور فهم رفتار و ترجیحات مشتری ضبط و تجزیه و تحلیل نمود. ترجیحات مشتری شامل موارد زیر است: زمانی که مشتری صرف مشاهده یک صفحه می‌کند، کدام پیوندها را انتخاب می‌کند، به کدام آگهی‌ها توجه بیشتری دارد، از طریق کدام صفحه وارد سایت شرکت شده و از کدام صفحه خارج می‌گردد و نظایر آن.

همهٔ این موارد به اطلاعات آماری پایگاه داده‌های شرکت تبدیل می‌شوند. این داده‌ها از این نظر مفیدند که به کمک آنها شرکتها می‌توانند مکانیزم فروششان را بشناسند و بدانند که چگونه محصولات و دیگر اقلام جانبی را که ممکن است مورد علاقه مشتریان باشد

---

[1]- Life Time Value

مکان‌یابی نمایند. این روش همچنین درصد ترک صفحه را نیز نشان می‌دهد. یعنی معلوم می‌کند که بازدیدکنندگان از روی کدام صفحات می‌پَرند، از کدام صفحات اجتناب می‌کنند یا کدام صفحات موجب می‌شود که بازدیدکننده سایت وب شرکت را ترک کند. تأکید این اطلاعات بر صفحاتی است که نیازمند ارزیابی بیشتر از نظر محتوا و سادگیِ استفاده می‌باشند. البته با اینکه نرم‌افزارهای قوی تجزیه و تحلیل اینترنتی فراوانند ولی هیچ‌کدام از آنها قادر نیستند به تنهایی تصویر کاملی از رفتارهای بازدیدکنندگان سایت ارائه دهند. مثلاً نمی‌توان به راحتی فهمید که آیا بازدیدکننده‌ها محصولات را برای خودشان می‌خرند یا به‌عنوان هدیه و برای دیگران می‌خرند، اینکه مشتریان سودآور هستند یا فقط در سایت به گشت و گذار می‌پردازند و غیره. بنابراین دموگرافی بازار هدف را نمی‌توان به تنهایی با استفاده از ابزار تجزیه و تحلیل اینترنتی انجام داد.

بازاریابی مستقیم

هدف تبلیغات آن دسته از مشتریان احتمالی است که در مورد تک‌تک آنان اطلاعات کافی نداریم. بازاریابی مستقیم حداقل نیاز به مقداری اطلاعات اضافه مانند نام و آدرس یا تلفن یا پست الکترونیک دارد. در بسیاری از کشورها، داده‌های قابل توجهی درباره بخش بزرگی از جمعیت در دسترس است. قبل از برنامه‌ریزی برای استفاده در بازاریابی، لازم است دسترسی به داده‌ها در بازار مورد نظر و محدودیتهای قاعدهی استفاده، بررسی شود. مشکل این است که حتی از طریق غربالهای بدیهی، نسبت مشتریان مورد بررسی به مشتریان احتمالی پاسخ‌دهنده، بسیار زیاد باشند. بنابراین یکی از کاربردهای اصلی داده‌کاوی برای یافتن مشتریان احتمالی، «هدف‌گیری» است. یعنی به دنبال یافتن آن مشتریان احتمالی هستیم که احتمال بیشتری دارد به یک پیشنهاد خاص پاسخ دهند.

فعالیتهای بازاریابی مستقیم عموماً نرخ پاسخی کمتر از ۱۰٪ دارند. این مدلها با تشخیص آن دسته از مشتریان احتمالی که احتمال پاسخشان به مشتری‌یابی مستقیم

بیشتر است، نرخهای پاسخ را بهبود می‌بخشند. مفیدترین مدلهای پاسخ، تخمینی واقعی از احتمال پاسخ می‌دهند. البته الزامی برای محاسبه احتمال واقعی پاسخ نیست بلکه داشتن مدلی که مشتریان احتمالی را به ترتیب بیشترین امتیاز پاسخ، رتبه‌بندی کند کافی است. با داشتن یک لیست مرتب شده می‌توان درصد پاسخگویان در یک فعالیت بازاریابی مستقیم را با پست به افراد بالای لیست یا تماس با آنها، زیاد کرد.

لایه‌های کشف الگو

وقتی داده از انباره‌داده استخراج شد و مرحله آماده‌سازی را طی کرد، هرکدام از روشهای داده‌کاوی می‌تواند برای پاسخ به سؤالات کسب وکاری که در ذهن است استفاده شود. دسته‌بندی، خوشه‌بندی، رگرسیون، الگوهای مکرر[1]، قواعد تلازمی و سریهای زمانی مرسوم‌ترین روشهای داده‌کاوی هستند. همان‌طور که در جدول (۱۰-۳) نشان داده شده کشف الگو شامل چهار لایه مهم است [۲].

جدول ۱۰-۳) لایه‌های کشف الگو

لایه اول	**سؤالهای تجاری** مانند توصیف مشتری
لایه دوم	**کاربردها** مانند امتیازدهی، پیش‌بینی برای حل یک نوع خاص سؤال کسب وکار
لایه سوم	**روشها** مانند سریهای زمانی، طبقه‌بندی باتوجه به نوع داده خروجی که نتیجه فرآیند بر روی داده ورودی است
لایه چهارم	**الگوریتمها**

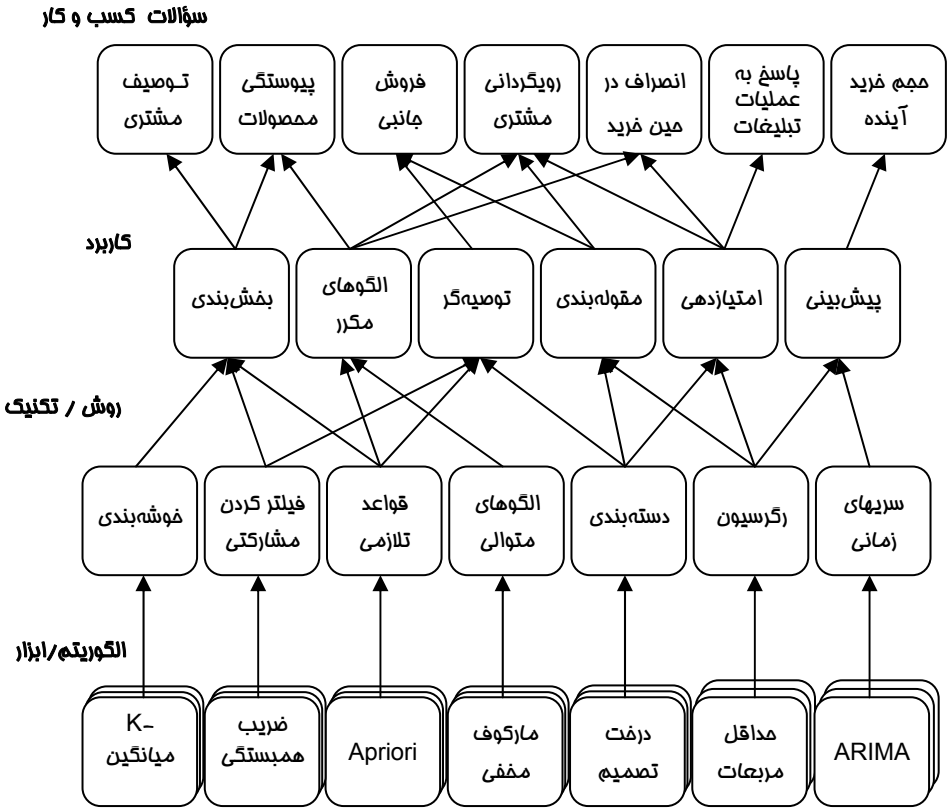
سؤال استراتژیک و سؤال عملیاتی

سؤال استراتژیک، سؤالی است که جواب آن راهنما یا تأثیرگذار بر یک تصمیم است.

[1]- Frequent Pattern

یک مثال می‌تواند تحلیل برای تعیین محرکهای کلیدی رضایت مشتری باشد. در مثال دیگر می‌توان با خوشه‌بندی بر حسب کل دفعات خرید هر مشتری، مشتری با ارزش را تعریف کرد.
سؤال عملیاتی، سؤالی است که نتایج آن مستقیماً برای افزایش وقایع سودآور استفاده می‌شود. برای مثال، در مدلسازی پاسخ بازاریابی مستقیم، خود امتیازهای مدل مهم‌ترین نتیجه هستند. اگر چه داشتن الگو یا مدل قابل توضیح، مطلوب است، ولی خود مدل هدف تحلیل می‌باشد.

درشکل (۱۰-۵) این لایه‌ها به‌طور جزئی‌تر نشان داده شده‌اند. البته امکان نمایش تمام حالات ممکن نمی‌باشد.



شکل ۱۰-۵) لایه‌های کشف دانش درمدیریت ارتباط با مشتری

دسته‌بندی (در مدیریت ارتباط با مشتری)

آیا می‌توان از روی خصوصیات ظاهری، تشخیص داد که یک فرد زن است یا مرد؟ این سؤالی است که دسته‌بندی به دنبال جواب آن است. چگونه می‌توان این کار را کرد؟ ابتدا باید تعدادی زن و مرد داشته باشید که خصوصیات فیزیکی و الگوهای رفتاری آنها ثبت شده باشد. حالا اگر فرد جدیدی را ببینید، می‌توانید از روی شباهت خصوصیات او را به یکی از دو دسته زن یا مرد، تخصیص دهید. برخی اوقات به دلیل کامل نبودن اطلاعات و یا غیرعادی بودن فرد، امکان اشتباه نیز وجود دارد.

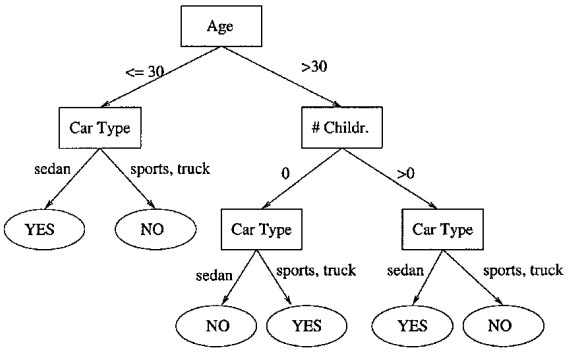
اگر دسته‌ها از قبل مشخص باشند (مثلاً اینکه خرید کرده یا نکرده) می‌توان به کمک روش دسته‌بندی ابتدا به‌وسیله داده‌های موجود (داده‌های آموزشی) مدلی ساخته و از آن برای پیش‌بینی موارد جدید استفاده کرد.

درخت تصمیم (در مدیریت ارتباط با مشتری)

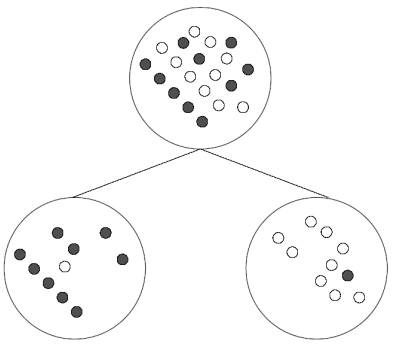
درخت تصمیم به دلیل سادگی بیان دانش یافته شده و قدرت پیش‌بینی مناسب، یکی از پرکاربردترین الگوریتمهای دسته‌بندی می‌باشد. دانش یافته شده به شکل قواعد اگر–آنگاه بیان می‌شود.

مسابقه ۲۰ سؤالی، نمونه‌ای از درخت تصمیم می‌باشد. در هر مرحله از مسابقه با پرسیدن یک سؤال، فضای جواب را محدودتر می‌کنیم تا در نهایت به جواب مشخصی که در ذهن طرف مقابل است برسیم. چه سؤالی را ابتدا انتخاب می‌کنید؟ سؤالی را ابتدا انتخاب می‌کنیم که فضای جواب را به خوبی به دو قسمت تقسیم کند.

مانند روشهای دیگر دسته‌بندی، هدف در درخت تصمیم، تعیین دسته (مثلاً خرید یا عدم خرید مشتری) از روی مشخصات می‌باشد. برای اینکار ابتدا مشخصه‌ای (متغیری) انتخاب می‌شود که با گذاشتن شرط بیشتر یا کمتر از یک مقدار روی آن، بهتر از همه متغیرهای دیگر قدرت تفکیک کلیه رکوردهای موجود را داشته باشد.



برای مثال هدف در شکل بالا، پیش‌بینی اشتراک افراد در یک مجله است. با بررسی روی تک‌تک مشخصات مشتری معلوم شده است که بهترین مشخصه تفکیک کننده مشتری، سن مشتری و آستانه تصمیم آن سن ۳۰ سال است. طبق شکل ذیل با بررسی شرط بیشتر بودن سن از ۳۰ سال، ۲۰ فرد موجود را که ۱۰ نفر از آنها مشتری هستند، به دو دسته تقسیم می‌شوند که در یکی از ۱۰ نفر، ۹ نفر مشترک هستند و در دیگر از ۱۰ نفر، ۹ نفر مشترک نیستند.



این فرایند برای شاخه‌های زیرین درخت دنبال شده و هر بار مشخصه‌های دیگری انتخاب می‌شود که بهتر از بقیه، داده‌ها را تفکیک می‌کند. این روند تا رسیدن به حد قابل قبولی از خطا یا رسیدن به دسته‌های ۱۰۰٪ خالص ادامه می‌یابد.

یکی از موارد مهم در دسته‌بندی، استحکام می‌باشد. همان‌طورکه متوجه شده‌اید امکان اشتباه در دسته‌بندی وجود دارد. بنابراین پس از ساختن مدل باید خطاهای دسته‌بندی را بررسی کرد تا در حد مقبولی باشند. ممکن است مدل برای داده‌های آموزشی خیلی خوب و کم‌خطا دسته‌بندی کند ولی در مورد داده‌های جدید (مثلاً تصمیم‌گیری در مورد مشتریان جدید) دارای خطای زیادی باشد. معیار ارتباط خطای آموزش به خطای استفاده در عمل، استحکام نام دارد. مدلی خوب است که دارای استحکام بالایی باشد.

خوشه‌بندی (در مدیریت ارتباط با مشتری)

انسان به‌طور فطری تمایل به گروه‌بندی اشیاء و مفاهیم بر اساس شباهت یا تفاوت (فاصله) دارد. برای مثال کودک می‌آموزد که موجودات زنده دو دسته هستند: آنهایی که سبزند و راه نمی‌روند (گیاهان) و آنهایی که معمولاً قهوه‌ای‌اند، حرکت می‌کنند و احتمالاً خطرناکند (حیوانات)!

گروه‌بندی در ادبیات داده‌کاوی، خوشه‌بندی[1] یا خوشه‌یابی نامیده می‌شود. روش خوشه‌بندی برای توصیف گروه‌های مختلف در مجموعه داده به‌کار می‌رود و بر خلاف دسته‌بندی خوشه‌ها از قبل مشخص نیستند. معمولاً ابتدا خوشه‌بندی انجام شده و بعد خوشه‌ها به‌عنوان نام دسته‌ها برای دسته‌بندی به‌کار می‌روند.

پس از خوشه‌بندی، برای تعبیر خوشه‌ها، نماینده هر خوشه را در نظر می‌گیرند. نماینده می‌تواند یکی از مشاهدات میانه خوشه و یا میانگین همه مشاهدات یک خوشه در نظر گرفته شود. با توجه به تفاوت و شباهت نماینده هر خوشه به نماینده کل داده‌ها از نظر مشخصات (مانند سن، جنسیت، . . .) می‌توان هر خوشه را تعبیر کرد. مثلاً به یک خوشه نام«مادران جوان » و به خوشه دیگر نام « نوجوانان حومه » را داد.

[1]- Clustering

در *CRM* گروه‌بندی از آن جهت اهمیت دارد که ایجاد پروفایلی برای مشتریان مختلف را فراهم می‌کند و امکان برنامه‌ریزی استراتژیک روی گروه‌های مشتریان را می‌دهد. از طرف دیگر امکان انواع گروه‌بندیهای دیگر مانند محصولات مشابه را ایجاد می‌کند.

خوشه‌بندی هدف‌گذاری شده (در مدیریت ارتباط با مشتری)

هر چند گروه‌های تشکیل شده بر اساس مشخصه‌هایی مانند سن، جنسیت یا رفتار قابل تمایز از هم هستند، ولی این ممکن است این گروه‌ها در بافت کسب و کار بامعنا نباشند. اگر همه گروه‌ها دارای عمر مشتری متوسطی بوده یا همه مقادیر یکسانی خرید کنند، این گروه‌بندی مفید نیست. خوشه‌بندی هدف به دنبال یافتن خوشه‌هایی است که با توجه به مقدار یک متغیر هدف خاص متفاوت هستند. این کار با افزایش وزن (اهمیت) متغیر هدف در معیار فاصله یا کُد کردن متغیرها با توجه به هدف انجام می‌شود. در حالت حدی که فقط متغیر هدف در معیار فاصله در نظر گرفته می‌شود، خوشه‌بندی شبیه دسته‌بندی می‌شود. فرق مهم خوشه‌بندی هدف و دسته‌بندی، این است که مقوله از پیش تعریف شده‌ای برای خوشه‌بندی هدف وجود ندارد.

ممکن است بسیاری از خوشه‌های یافته شده، بدیهی و فاقد دانش جدید و مفیدی برای ما باشند. بنابراین لازم است خوشه‌های یافته شده از نظر جالب بودن[1] طبق نظر خبرگان بررسی شوند. گاهی اوقات، نشان دادن شباهت و تفاوت خوشه‌های مختلف با هم مفید می‌باشد. راه متداول برای اینکار شبکه‌های کوهونن یا همان *SOM* است.

رگرسیون و سریهای زمانی (در مدیریت ارتباط با مشتری)

روش رگرسیون سعی در پیش‌بینی یک خروجی پیوسته با استفاده از یک تابع دارد که میزان خطا از الگو را در داده‌ها نشان می‌دهد. روشهای رگرسیون را می‌توان برای کاربرد دسته‌بندی دوتایی و همچنین برای امتیازدهی و پیش‌بینی استفاده کرد. روشهای

[1]- Interestingness

سری‌زمانی مثل رگرسیون مقادیر پیوسته را پیش‌بینی می‌کنند، ولی در طول زمان، گرایشها و چرخه رفتار را نیز مدل می‌کند.

قواعد تلازمی (در مدیریت ارتباط با مشتری)

قوعد تلازمی و الگوهای مکرر، وقایعی که در مجموعه داده اتفاق می‌افتد را توصیف می‌کنند. قواعد تلازمی، ارتباط بین موارد موجود در یک مجموعه داده است بدون اینکه ترتیب زمانی یا ترتیب خاصی داشته باشند.

فیلتر کردن مشارکتی[1]

فیلتر کردن مشارکتی وقایع را برای یافتن مجموعه‌هایی که شبیه یکدیگرند تحلیل می‌کنند. در مفهوم *CRM*، وقایع معمولاً بر اساس مشتریان گروه‌بندی می‌شود. ایده اصلی این است که اگر مشتریان با سابقه خرید مشابه کالای خاصی را بخرند، ممکن است مشتری جدید دارای همان مشخصات نیز همان کالا را بخرد. برای هر مشتری در مجموعه داده، با روش فیلتر کردن مشارکتی یک گروه از مشتریان پیدا می‌شوند که گزارشات وقایعشان به یکدیگر شبیه هستند. وقتی گروه شکل می‌گیرد، وقایع می‌تواند با چیزی که در کل گروه عمومیت دارد، امتیازدهی شود. خروجی فیلتر کردن مشارکتی می‌تواند هم مشتریان شبیه به هم و هم وقایع امتیاز داده شده باشد. برای مثال، اگر واقعه، خرید محصولات باشد، امتیازدهی وقایع می‌تواند برای توصیه محصولات به کار برده شود. هر محصولی که بیشترین امتیاز را بیاورد توصیه می‌شود. فیلتر کردن مشارکتی مشابه خوشه‌بندی برای کشف گروه‌های طبیعی در مجموعه داده است. تفاوت مهم فیلتر کردن مشارکتی این است که گروه‌های متفاوتی گرداگرد یکدیگر در مجموعه داده شکل می‌گیرند. در واقع هر مشتری مرکز یک گروه واحد است و مشتری A ممکن است در گروه مشتری B ظاهر شود ولی عکس آن درست نباشد. در نتیجه روش فیلتر کردن مشارکتی برای شخصی‌سازی سیستم به کار برده می‌شود.

---

[1]- Collaborative Filtering

منابع

1) *Berson A. , Smith S. ,Thearling K. (2001) "Bulding Data Mining Application for CRM" , McGraw-Hill.*

2) *Ye N. (2003) "THE HANDBOOK OF DATA MINING", LAWRENCE ERLBAUM ASSOCIATES , PUBLISHERS Mahwah, New Jersey London.*

3) *Rygielski C. , Wang J. C. , Yen D. C. (2002), "Data mining techniques for customer relationship management", Technology in Society 24 (2002) 483–502.*

4) *Berry M. J. A. , Linoff G. S. (2004) Data Mining Techniques for Marketing, Sales, and Cutomer Relationship Management, Second edition, Wiley.*

۵)سمیرا ملک محمدی، سمینار کارشناسی ارشد، کاربرد داده‌کاوی در مدیریت ارتباط با مشتری دانشگاه علم وصنعت، استاد راهنما دکتر مهدی غضنفری.