داده‌کاوی و کشف دانش

داده‌کاوی و کشف دانش

گردآوری :

مهدی غضنفری، سمیه علیزاده، بابک تیمورپور

سرشناسه : غضنفری، مهدی، ۱۳۳۹-
عنوان ونام پدیدآور : داده کاوی و کشف دانش/ تالیف مهدی غضنفری ، سیمه علیزاده، بابک تیمورپور.
مشخصات نشر : تهران: دانشگاه علم و صنعت ایران، ۱۳۸۷.
مشخصات ظاهری : ح ، ۴۰۳ ص.: مصور، جدول، نمودار
شابک : ۵۰۰۰۰ ریال: ۲-۱۷۸-۴۵۴-۹۶۴
وضعیت فهرست نویسی : فیپا
یاداشت : واژه نامه.
یاداشت : کتابنامه.
موضوع : داده کاوی
شناسه افزوده : دانشگاه علم و صنعت ایران. مرکز انتشارات
رده بندی کنگره : ۱۳۸۷ ۶غ ۲ د/۷۶/۹ QA
رده بندی دیودی : ۰۰۶/۳
شماره کتابشناسی ملی : ۱۳۱۹۶۴۱

• مرکز انتشارات دانشگاه علم و صنعت ایران- تهران- نارمک صندوق پستی ۱۶۷۶۵-۱۶۳ تلفن: ۷۷۲۴۰۴۲۵- دورنویس: ۷۷۲۴۰۴۲۵
• فروشگاه شمارهٔ ۱: میدان انقلاب- خیابان شهید منیری جاوید (اردیبهشت)- پلاک ۱۸۲ تلفن: ۶۶۴۶۶۹۰۰
• وب سایت: Publication.iust.ac.ir

نام کتاب: داده کاوی و کشف دانش
گردآوری: مهدی غضنفری، سیمه علیزاده، بابک تیمورپور
چاپ اول: ۱۳۸۷
شمارگان: ۱۰۰۰ جلد
قیمت: ۵۰۰۰۰ ریال
شماره انتشارات: ۴۸۸
لیتوگرافی، چاپ و صحافی: مرکز انتشارات دانشگاه علم و صنعت ایران

شابک: ۲-۱۷۸-۴۵۴-۹۶۴ **ISBN: ۹۶۴-۴۵۴-۱۷۸-۲**

فهرست مطالب

فصل اول

مقدمه‌ای بر داده‌کاوی ۳

۱-۱-مروری بر کشف دانش و داده‌کاوی ۳
۱-۲- تعاریف کشف دانش/ داده‌کاوی ۶
۱-۲-۱- کشف دانش در پایگاه داده‌ها..... ۷
۱-۳- فرایند کشف دانش ۸
۱-۴- حوزه‌ها، وظایف و عملکردهای داده‌کاوی ۱۲
۱-۵-روشهای داده‌کاوی ۱۳
۱-۶- مثالهایی از روشهای داده‌کاوی ۱۶
۱-۶-۱- کاربردهای KDD ۲۱
۱-۶-۲- چالشهایی برای KDD..... ۲۱
منابع ۲۴

فصل دوم

پیش‌پردازش و آماده‌سازی داده‌ها..... ۲۶

۲-۱- انواع داده‌های مورد استفاده در داده‌کاوی..... ۲۶
۲-۱-۱- ویژگیهای کمی و کیفی ۲۷
۲-۱-۲- ویژگیهای گسسته و پیوسته ۲۹
۲-۱-۳- ویژگیهای نامتقارن ۲۹
۲-۲- آماده سازی داده‌ها ۳۱
۲-۲-۱- جایگاه آماده سازی داده‌ها در داده‌کاوی..... ۳۱
۲-۲-۲- چرا آماده‌سازی داده‌ها ۳۲
۲-۲-۳- تلخیص توصیفی داده‌ها ۳۴
۲-۲-۴- نمایش گرافیکی داده‌های توصیفی ۳۵
۲-۲-۵- اجزاء اصلی پیش پردازش داده‌ها..... ۴۰

۲-۳- پاکسازی داده‌ها ۴۲
۲-۳-۱- وظایف پاکسازی داده‌ها ۴۲
۲-۳-۲- پاکسازی داده به‌عنوان یک فرآیند ۵۰
۲-۴- یکپارچه‌سازی داده‌ها ۵۲
۲-۵- تبدیل داده‌ها ۵۵
۲-۵-۱- هموارسازی ۵۵
۲-۵-۲- تجمیع ۵۶
۲-۵-۳- تعمیم ۵۶
۲-۵-۴- ساخت ویژگی ۵۶
۲-۵-۵- نرمال‌سازی ۵۶
۲-۶- کاهش داده‌ها ۵۸
۲-۶-۱- تجمیع مکعبی داده ۶۳
۲-۶-۲- انتخاب زیرمجموعه مشخصه‌ها ۶۴
۲-۶-۳- کاهش تعدد نقاط ۶۹
۲-۷- تصویر کردن برای کاهش بُعد ۶۹
۲-۷-۱- تعاریف و مفاهیم ۷۰
۲-۷-۲- تحلیل مؤلفه‌های اصلی ۷۱
۲-۷-۳- تجزیه مقدار منفرد ۷۷
۲-۷-۴- تبدیلات گسسته فوریه ۷۷
۲-۷-۵- تبدیل موجک گسسته ۷۸
۲-۷-۶- تصویر کردن تصادفی ۸۲
۲-۷-۷- نگاشت سریع ۸۲
۲-۷-۸- مقیاس‌گذاری چند بعدی ۸۸
منابع ۹۵
ضمیمه ۱- مفاهیم پایه آماری ۹۷

فصل سوم

تحلیل خوشه‌ای ۱۰۳

۳-۱- تعاریف و مفاهیم تحلیل خوشه‌ای ۱۰۴
۳-۲- معیارهای شباهت و تمایز در انواع داده‌ها ۱۰۷

۳-۲-۱- انواع متغیرها و معیارهای شباهت و تمایز ۱۰۹
۳-۳- روشهای اصلی خوشه‌بندی ۱۱۶
۳-۳-۱- روش افرازی ۱۱۹
۳-۳-۲- روش خوشه‌بندی سلسله مراتبی ۱۲۸
۳-۳-۳- مقایسهٔ خوشه‌بندی سلسله مراتبی و غیر سلسله مراتبی ۱۳۳
۳-۳-۴- تعیین تعداد خوشه‌ها ۱۳۳
۳-۳-۵- روشهای مبتنی بر چگالی ۱۳۴
۳-۳-۶- روشهای مبتنی بر مشبک کردن فضا ۱۴۱
۳-۳-۷- نقشه‌های خودسازمانده ۱۴۳
منابع ۱۵۵

فصل چهارم

قواعد تلازمی ۱۵۷

۴-۱- تعاریف و مفاهیم اصلی در قواعد تلازمی ۱۵۸
۴-۱-۱- الگوریتم AIS ۱۶۵
۴-۱-۲- الگوریتم SETM ۱۶۶
۴-۱-۳- الگوریتم Apriori ۱۶۸
۴-۱-۴- الگوریتم AprioriTid ۱۷۳
۴-۱-۵- الگوریتم Apriori Hybrid ۱۷۷
منابع ۱۷۹

فصل پنجم

دسته‌بندی و پیش‌بینی ۱۸۱

۵-۱- مفاهیم دسته‌بندی ۱۸۱
۵-۱-۱- تفاوت دسته‌بندی و خوشه‌بندی ۱۸۲
۵-۱-۲- فرایند دو مرحله‌ای دسته‌بندی ۱۸۲
۵-۱-۳- روشهای مختلف دسته‌بندی ۱۸۵
۵-۲- روش دسته‌بندی بیزی ۱۸۷
۵-۲-۱- بیز ساده ۱۸۷
۵-۲-۲- شبکه‌های بیزی ۱۹۰

۵-۳-دسته‌بندی بر مبنای نزدیکترین همسایگی ۱۹۳
۵-۴- شبکه‌های عصبی در دسته‌بندی ۲۰۶
۵-۴-۱- تبدیلات ورودی و خروجی ۲۰۹
۵-۴-۲- توابع فعال سازی ۲۱۲
۵-۴-۳- الگوریتم پس انتشار خطا ۲۱۳
۵-۴-۴- روش کاهش گرادیان ۲۱۴
۵-۴-۵- برخی کاربردهای دسته‌بندی بر اساس شبکه‌های عصبی ۲۱۷
۵-۵- درخت تصمیم ۲۱۸
۵-۵-۱- خصوصیات درخت تصمیم ۲۱۸
۵-۵-۲- روش کار درخت تصمیم ۲۲۰
۵-۵-۳- مفاهیم اصلی در درختهای تصمیم ۲۲۲
۵-۵-۴- ساخت یک نمونه درخت تصمیم با استفاده از روش شاخص جینی ۲۲۵
۵-۵-۵- ارزیابی درخت ایجاد شده ۲۳۵
۵-۵-۶- استخراج قواعد دسته‌بندی از درختهای تصمیم ۲۳۷
۵-۶- پیش‌بینی ۲۳۹
۵-۶-۱-مدلهای رگرسیون برای دسته‌بندی ۲۴۱
۵-۷-روشهای ارزیابی دسته‌بندی ۲۴۴
۵-۷-۱- پیچیدگی در مدلسازی ۲۴۵
۵-۷-۲- اندازه‌گیری خطا و میزان صحت در اندازه‌گیریها ۲۴۸
۵-۷-۳- ارزیابی صحت روشهای دسته‌بندی ۲۴۸
۵-۷-۴-میزان خطای پیش‌بینی کننده‌ها ۲۵۱
منابع ۲۵۳

فصل ششم

انباره داده‌ها ۲۵۷

۶-۱-داده‌کاوی و انباره‌داده‌ها ۲۵۸
۶-۲- مفاهیم انباره‌داده‌ها ۲۵۹
۶-۲-۱- مدل مفهومی انباره‌داده‌ها ۲۶۲
۶-۲-۲- فرایند طراحی انباره‌داده ۲۶۶
۶-۲-۳- معماری انباره‌داده ۲۶۷
۶-۳- انواع انباره‌داده ۲۶۹

۶-۴- انباره‌داده و سیستم‌های عملیاتی ۲۷۱
۶-۴-۱- کاربران نهایی انباره‌داده‌ها ۲۷۳
منابع ۲۷۶

فصل هفتم

متدلوژی اجرا و پیاده‌سازی پروژه‌های داده‌کاوی ۲۷۷

۷-۱- گام شناخت سیستم ۲۷۸
۷-۲- گام شناخت داده ها ۲۷۹
۷-۳- گام آماده سازی داده ها ۲۸۰
۷-۴- گام مدلسازی ۲۸۱
۷-۵- گام ارزیابی ۲۸۲
۷-۶- گام توسعه ۲۸۳
منابع ۲۸۵

فصل هشتم

سریهای زمانی در داده‌کاوی ۲۸۹

۸-۱- داده‌کاوی سریهای زمانی ۲۹۰
۸-۱-۱- اجزاء سریهای زمانی و تحلیل آنها ۲۹۱
۸-۱-۲- شناسایی، تجزیه و حذف اجزاء سریهای زمانی ۲۹۴
۸-۱-۳- سریهای زمانی با روند خطی ۲۹۴
۸-۱-۴- جستجوی تشابه در تحلیل سریهای زمانی ۳۰۰
۸-۱-۵- مقیاسهای اندازه‌گیری تشابه در سریهای زمانی ۳۰۱
۸-۱-۶- تاباندن محور زمان به‌صورت پویا ۳۰۴
۸-۱-۷- الگوریتم کلاسیک DTW ۳۰۶
۸-۱-۸- شباهت بزرگترین زیردنباله مشترک(LCSS) ۳۱۰
۸-۱-۹- روشهای شاخص‌گذاری برای جستجوی تشابه در سریهای زمانی ۳۱۶
منابع ۳۲۸

فصل نهم

تحلیل شبکه‌های اجتماعی ۳۲۹

۹-۱-تعریف شبکه اجتماعی ۳۲۹
۹-۲- ویژگیهای شبکه‌های اجتماعی ۳۳۳
۹-۳- پیوندکاوی: وظایف و چالشها ۳۳۷
۹-۴- کاوش شبکه‌های اجتماعی ۳۴۴
۹-۴-۱- کاوش گروه‌های خبری با کمک شبکه‌ها ۳۴۹
۹-۴-۲- اجتماع کاوی شبکه‌های چندرابطه‌ای ۳۵۱
منابع ۳۵۵

فصل دهم

کاربرد داده‌کاوی در مدیریت ارتباط با مشتری ۳۵۷

۱۰-۱- معماری مدیریت ارتباط با مشتری ۳۵۹
۱۰-۱-۱- یافتن مشتریان احتمالی ۳۶۱
۱۰-۱-۲- داده‌کاوی برای انتخاب محل مناسب تبلیغ ۳۶۴
۱۰-۱-۳- داده‌های مشتریان ۳۷۲
۱۰-۲- برخی کاربردهای داده‌کاوی در مدیریت ارتباط با مشتری ۳۷۷
۱۰-۲-۱- لایه‌های کشف الگو ۳۸۱
منابع ۳۸۸

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 می‌شود 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 دوم آماده سازی داده‌ها در داده‌کاوی است که شامل روشهای متعددی در مورد آماده‌سازی داده‌ها و پیش‌پردازش آنهاست. موضوع کاهش بُعد مفاهیم پیشرفته‌ای در زمینه پیش‌پردازش داده‌ها را مطرح می‌کند که می‌توان مطالعه آن را به بعد از فصل 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 ملک‌محمدی، نگار رستگار و بنت‌الهدی‌علی‌احمدی، و آقایان عیسی چمبر، 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

تابستان۸۷

بخش اول

داده‌کاوی و آماده‌سازی داده‌ها

فصل اول: مقدمه‌ای بر داده‌کاوی

فصل دوم: پیش‌پردازش و آماده‌سازی 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[3] 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 داده خود ذخیره کرده‌اند. این سازمانها به فهم این داده‌ها و یا کشف دانش مفید از آنها نیاز دارند.

۱-۱- مروری بر کشف دانش و داده‌کاوی

کشف دانش و داده‌کاوی

همان‌طورکه الکترونها و امواج موضوع اصلی مهندسی برق شدند، داده‌ها[4]، اطلاعات[5] و دانش[1] نیز موضوع اصلی حوزه جدیدی از تحقیق و کاربرد به نام «کشف دانش و داده‌کاوی» یا به اختصار *KDD* هستند.

۱- Knowledge Discovery and Data Mining (KDD)

۲- Interdisciplinary

۳- Machine Learning

۴- Data

۵- Information

به‌طور کلی، *داده‌ها* رشته‌ای از بیتها (به صورت صفر و یک) یا اعداد و نشانه‌ها و یا اشیاء[۲] 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[۳] 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[۴] 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[۵] و مدلهای موجود در داده‌هاست. الگوها و مدلهایی که معتبر، بدیع[۶]، بالقوه مفید و کاملاً قابل فهم هستند. *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 جالبِ موجود در پایگاه داده‌ها است که در میان حجم عظیمی از داده‌ها مخفی هستند.

در ادامه ایده‌های[۷] گوناگون مرتبط با یک پایگاه داده واقعی مطرح خواهد شد. داده‌های این پایگاه در مورد التهاب مغزی[۸] است که در مؤسسه تحقیقات پزشکی دانشگاه پزشکی و دندانپزشکی توکیو از سال ۱۹۷۹ تا ۱۹۹۳ جمع‌آوری شده است. این پایگاه داده حاوی داده‌های بیمارانی است که دچار التهاب مغزی بوده و در بخش اورژانس و عصب شناسی بیمارستانهای

۱- Knowledge
۲- Objects
۳- Redundancy
۴- Facts
۵- Patterns
۶- Novel
۷- Notions
۸- Meningitis

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 مشاهده می‌شوند که ترکیبی از داده‌های عددی و طبقه‌ای[1] و نیز مقادیر مفقوده[2] می‌باشند (مقادیر مفقوده با علامت؟ مشخص شده‌اند)

,–, –,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*,?,

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 صحت آن با درجه *اطمینان* ۸۷/۵٪ اندازه گرفته شده است:

اگر تعداد سلولهای چند هسته‌ای با مشخصه CFS ≥ ۲۲۰ بوده
و عامل ریسک = n
و از دست دادن هوشیاری = مثبت
و شروع حالت تهوع به صورت Nausea> ۱۵
آنگاه پیش‌بینی = ویروس (اطمینان =۸۷/۵٪)

---

[1]- Categorical

[2]- Missing Value

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 یک «*یافته*» برحسب یک معیار آماری است. *افزونگی*، مقدار شباهت یک الگوی کشف شده نسبت به یافته‌های دیگر است و درجه تبعیت آنرا از دیگری اندازه می‌گیرد. *فایده*[3]، ارتباط یافته را با اهداف کاربران بیان می‌کند. *بدیع بودن*[4] بیانگر میزان تازگی نسبت به دانش قبلی کاربر یا سیستم است. *سادگی*[5] به پیچیدگی نحوی[6] و نمایش یک الگوی کشف شده و نحوه تعمیم آن اشاره دارد.

۱-۲- تعاریف کشف دانش/ داده‌کاوی

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

- تحلیل داده‌های توصیفی کامپیوتری، در مجموعه‌های بزرگ و پیچیده داده‌ها [۱۲].
- تحلیل ثانوی[7] 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 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].

---

[1]- Degree of Interest

[2]- Evidence

[3]- Usefulness

[4]- Novelty

[5]- Simplicity

[6]- Syntactical

[4]- ثانوی به این معنا است که منظور اصلی کسب و کار از جمع‌آوری پایگاه داده‌ها، کشف دانش نبوده است.

- داده‌کاوی یک حوزه میان‌رشته‌ای و با رشد سریع است که حوزه‌های مختلفی همچون پایگاه داده، آمار، یادگیری ماشین و سایر زمینه‌های مرتبط را با هم تلفیق 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 دانش از حجم عظیمی از داده‌ها می‌باشد داده‌کاوی الگوهای جالب را در میان حجم بزرگی از داده‌ها می‌یابد.

۱-۲-۱- کشف دانش در پایگاه داده‌ها

از دیدگاه منطق، «دانش» هر حقیقت صریحاً اظهارْ شده و موجه در یک زمینه است که با زبانهای رسمی بیان 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. ولیکن محققان پایگاه داده‌ها در ارتباط بیشتری برای گفتمان با اهل کسب و کار و رسانه‌ها بودند و واژه داده‌کاوی در اخبار کسب و کار متداول شد.» داده‌کاوی واژه‌ای قدیمی‌تر از KDD است که در جامعه تحلیل داده‌های آمارمحور، ابداع شده است.

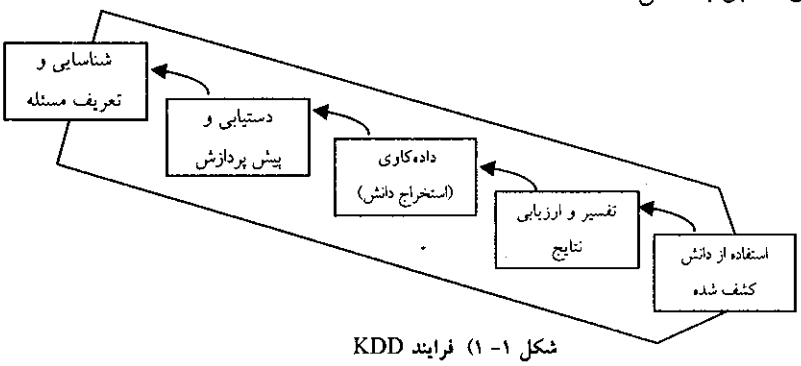
برخی محققان استفاده کننده از داده کاوی در زبان انگلیسی را واژه‌ای گمراه کننده می‌دانند. آنها معتقدند که در واقع آهن و طلا کاوش می‌شوند نه غبار یا سنگ و خاک. آنان می‌گویند دانش‌کاوی تمثیل بهتری است زیرا مانند آهن و طلا، خروجی مورد نظر، دانش است. برخی، داده‌کاوی را یک گام مرکزی در فرایند کشف دانش می‌دانند که الگوریتمهای استخراج و اثبات فرضیه را اعمال می‌کند [۵]. این تعبیر مورد پذیرش عموم در این حوزه نیست. بسیاری

داده‌کاوی را معادل با واژه متداول کشف دانش ممکن است در پایگاه داده‌ها می‌دانند. عناوین دیگری که در گذشته به‌جای داده‌کاوی استفاده می‌شدند عبارتند از: باستان‌شناسی داده[۱]، لایروبی داده[۲]، تحلیل وابستگی تابعی و درو کردن داده[۳].

باید توجه داشت که در زبان فارسی فعل «کاویدن» هم برای داده‌ها و هم برای دانشی که از داده‌ها استخراج می‌شود، قابل استفاده است. یعنی اصطلاح کاویدن بر روی داده‌ها و کاویدن بر روی دانش، هر دو درست است. ولی ما در این کتاب از منظور اول استفاده می‌کنیم. یعنی فعل «کاویدن» را برای داده‌ها و برای دانش از فعل «کشف» استفاده کرده و واژگان داده‌کاوی و کشف دانش را به کار می‌بریم.

۱-۳- فرایند کشف دانش

بر اساس دیدگاهی که داده‌کاوی را بخشی از فرایند کشف دانش می‌دانند، کشف دانش شامل مراحل متعددی مطابق با شکل (۱-۱) است.



شکل ۱- ۱) فرایند KDD

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.

۱- Data Archaeology

۲- Data Dredging

۳- Data Harvesting

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] و یا گسسته سازی[۶] 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 تصویر کلی[۸] از ساختاری است که روابط سیستماتیک میان داده‌ها را بیان می‌کند» در مقابل، «یک الگو، ساختاری محلی است که فقط به چند متغیرِ محدود و تعدادی مشاهده مرتبط است.»

روشهای اصلی داده‌کاوی به دو دسته توصیفی[۹] و پیشبینانه تقسیم می‌شوند. نمونه‌هایی از این روشها عبارتند از: مدلسازی برای پیش‌بینی[۱۰] (مثل دسته‌بندی[۱۱] و رگرسیون[۱۲])، بخش‌بندی یا تقطیع[۱۳] (خوشه‌بندی)[۱۴]، مدلسازی وابستگی[۱۵] (مانند مدلهای تصویری یا تخمین چگالی)،

۱- Preprocess
۲- Outliers
۳- Noise
۴- Missing Data
۵- Transformation
۶- Discrimination
۷- Reduction
۸- Global Representation
۹- Descriptive
۱۰- Predictive
۱۱- Classification
۱۲- Regression
۱۳- Segmentation
۱۴- Clustering
۱۵- Dependency

تلخیص(خلاصه‌سازی)[۱] پیدا کردن رابطه بین فیلدها، تلازم یا انجمنی[۲]، مصورسازی[۳] و مدلسازی یافتن تغییر و انحراف[۴] در داده و دانش.

قدم چهارم: تفسیرِ (یا پس پردازش[۵]) دانش کشف شده است. این تفسیر، عملاً توصیفی یا پیشبینانه 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[۶] می‌باشد. یک راه استاندارد ارزیابی قواعد استنتاج شده[۷] تقسیم داده‌ها به دو مجموعه برای آموزش و آزمون 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[۸] 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

۱- 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)

۳- Visualization

۴- Deviation

۵- Post-Process

۶- Iterative

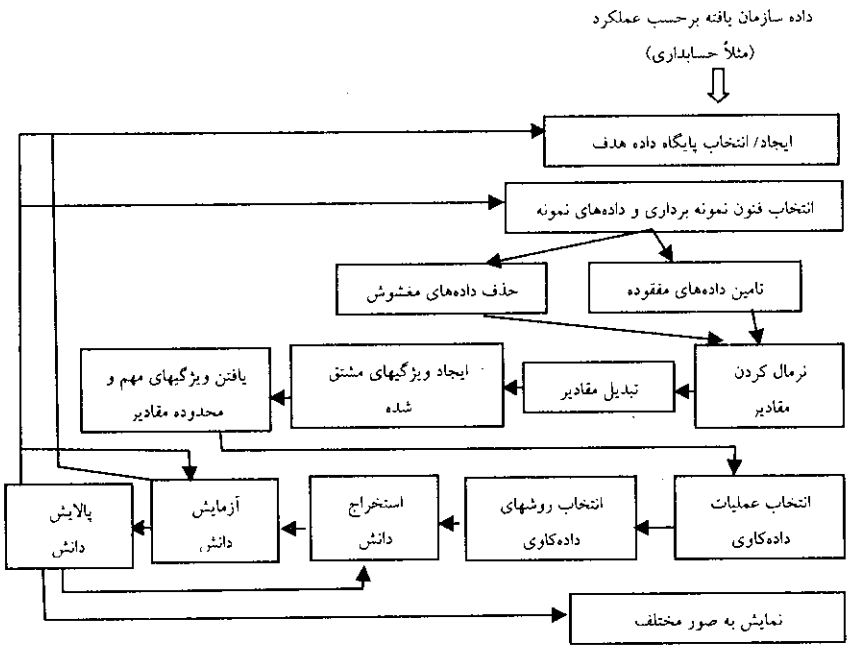
۷- Induced Rules

۸- Enumeration

طرحهای[۱] کدبندی، پیش‌پردازش، نمونه‌گیری[۲] و تصویر کردن[۳] داده قبل از مرحله داده‌کاوی است.

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

- درک کامل حوزه کاربرد: شامل درک دانش پیشین مرتبط، اهداف کاربرنهایی و غیره می‌باشد.



شکل ۱- ۲) وظایف فرایند KDD [۸،۱۵]

- 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.[۵]

۱- Schemes

۲- Sampling

۳- Projections

۵- Refine

- پیش- پردازش یا پاکسازی داده[۱]: عملیات مقدماتی مثل حذف اغتشاش یا نقاط پرت، جمع کردن اطلاعات لازم برای مدل کردن یا مقابله با اغتشاش، تصمیم‌گیری 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 مدل مورد استفاده برای داده‌های طبقه‌ای با مدل‌های مورد استفاده برای داده‌های عددی متفاوت می‌باشد. به علاوه 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 دسته‌بندی، 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.
- پایش یا تثبیت[۲] 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[۳] بالقوه این دانش با دانشهای مورد قبول (یا کشف شده) پیشین می‌باشد.

۱- Data Cleaning Preprocessing

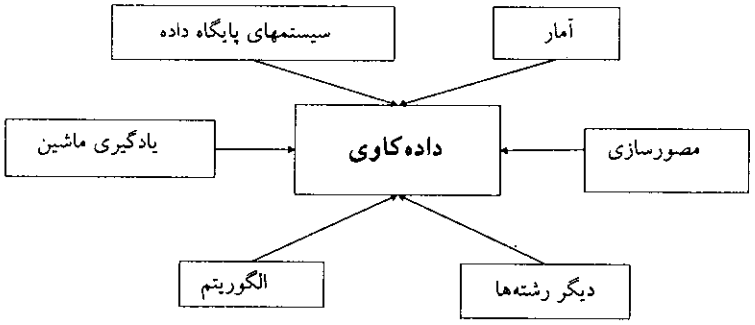
۲- Consolidation

۳- Conflicts

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

۱-۴- 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[۱] برای سیستمهای خبره. شکل (۱-۳) این ارتباطات را نشان می‌دهد.



شکل ۱- ۳) حوزه‌های مختلف داده‌کاوی [٦]

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[۲] و تشخیص الگو[۳] در مباحث مرتبط با نظریه‌ها و الگوریتمهای استخراج الگو از داده‌ها با حوزهٔ *KDD* به نوعی همپوشانی دارند. *KDD* بر روی توسعه این نظریه‌ها و الگوریتمها متمرکز شده است تا امکان یافتن الگوهای خاص را در مجموعه‌های بزرگ داده فراهم سازد.

KDD اشتراک زیادی نیز با آمار به خصوص تحلیل اکتشافی داده‌ها[٤] دارد. سیستمهای *KDD* معمولاً از رویه‌های آماری خاصی برای مدلسازی داده و بررسی اغتشاش استفاده می‌کنند. حوزهٔ

۱- Knowledge Acquisition

۲- Machine Learning

۳- Pattern Recognition

٤- Exploratory Data Analysis (EDA)

مرتبط دیگر، انباره‌داده‌ها[۱] است که به روندهای متداول سیستمهای اطلاعات مدیریت[۲] برای جمع‌آوری و پاکسازی داده‌های تراکنشی و قابل دسترسی کردن آن برای بازیافت فوری مربوط است. یک رویکرد برای تحلیل انباره داده‌ها، پردازش تحلیلی برخط، *OLAP*[۳] نام دارد. ابزارهای *OLAP* روی تحلیل چند بعدی داده متمرکز می‌شوند. این رویکرد در خلاصه‌سازی داده‌ها و ارائه آن بر حسب ابعاد مختلف نسبت به رویکرد *SQL* (زبان پرسش استاندارد)[٤] 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.

۱-۵- روشهای داده‌کاوی

روشهای اصلی داده‌کاوی دو دسته می‌باشند: توصیفی[۹] 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.

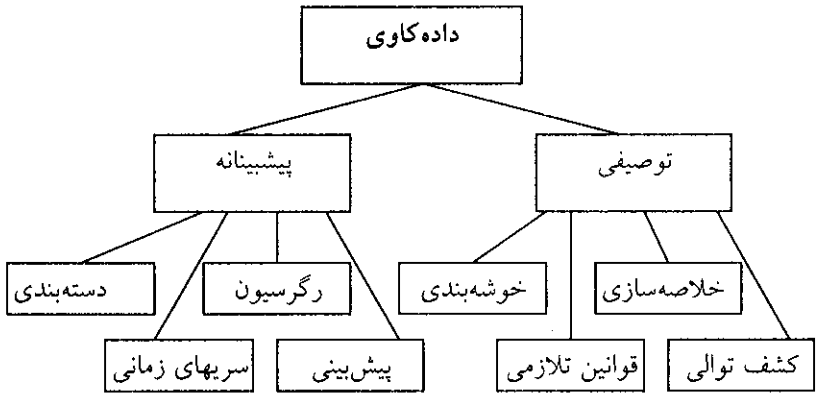
۱- Data Warehouse

۲- Management Information System (MIS)

۳ On_line Analytical Processing

٤- Standard Query Language

۹- Descriptive



شکل ۱- ٤) عملکردهای داده‌کاوی [٤]

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[۱] به کار می‌روند.

برخی روشهای متداول دسته‌بندی عبارتند از:

- درخت تصمیم. دو نمونه از الگوریتمهای درخت تصمیم C ٤,٥ و $CART$ هستند.
- دسته‌بندی بیزی: دارای دو نوع بیز ساده و شبکه‌های بیزی است.
- شبکه عصبی پس‌انتشار[۲].
- ماشینهای بردار پشتیبان[۳].
- دسته‌بندی تلازمی.
- یادگیرندگان کاهل: نزدیک‌ترین همسایگان، استدلال مبتنی بر مورد[٤].

---

[۱] Over Fit

[۲] Back Propagation

[۳] Support Vector Machine (SVM)

[٤] Case Base Reasoning

- روشهای دیگر: ژنتیک، مجموعه‌های نادقیق، مجموعه‌های فازی.

پیش‌بینی: درحالی‌که دسته‌بندی، برچسبهای[۱] طبقه‌ای (یعنی گسسته و بدون ترتیب) را پیش‌بینی می‌کنند، روشهای پیش‌بینی، توابع مقدار پیوسته را مدل می‌کنند.

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

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)[۲] و جامع[۳] بوده و یا دارای نمایشی غنی‌تر مانند نمایش سلسله مراتبی یا وضعیت هم‌پوشانی[۴] 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 خودسازمانده[۵] *(SOM)*.
- روشهای سلسله مراتبی[۶]: تجمعی، تقسیمی.
- روشهای مبتنی بر چگالی[۷].

تلخیص(خلاصه‌سازی): 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

۱- Label
۲- Exclusive
۳- Exhaustive
۴- Overlapping
۵- SOM
۶- Hierarchical
۷- Density based

و کشف رابطه تابعی بین متغیرها است. فنون تلخیص معمولاً در تحلیل اکتشافی داده‌ها 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 غالباً از طریق رسم 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[۴] از استقلال شرطی برای مشخص کردن جنبه ساختاری مدل و از احتمالات یا همبستگیها[۵] برای تعیین قدرت وابستگی استفاده می‌کنند. شبکه‌های وابستگی احتمالی به‌طور فزاینده‌ای در حوزه‌های کاملاً متفاوتی همانند توسعه سیستمهای خبره پزشکی، بازیافت اطلاعات[۶] *(IR)* و مدلسازی ژن انسانی استفاده شده‌اند.

۱-۶- مثالهایی از روشهای داده‌کاوی

شکل (۱-۵) مجموعه داده‌های دو بعدی شامل ۲۳ نمونه را نشان می‌دهد. هر نقطه روی شکل بیان کنندهٔ یک مشتری است که در گذشته از بانک 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. این دسته‌بندی‌ها می‌تواند به مدیر بانک در اخذ تصمیم اعطای وام کمک کند.

۱- Significant

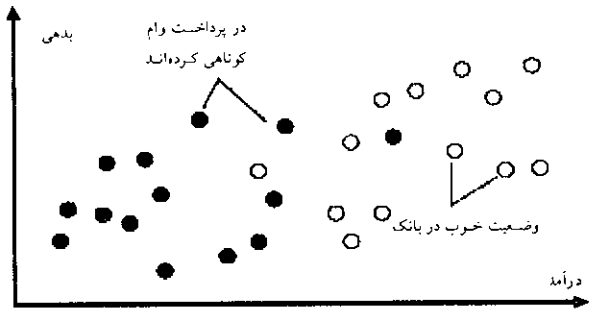
۲- Structural

۳- Quantitative

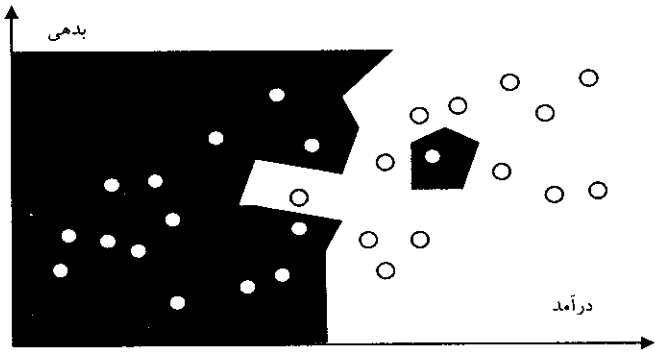
۴- Probabilistic Dependency Network

۵- Correlation

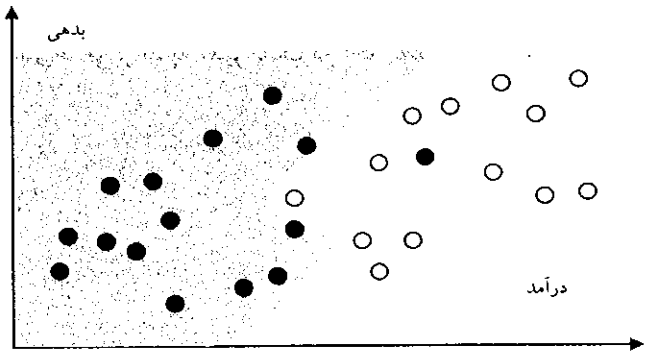
۶- Information Retrieval



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

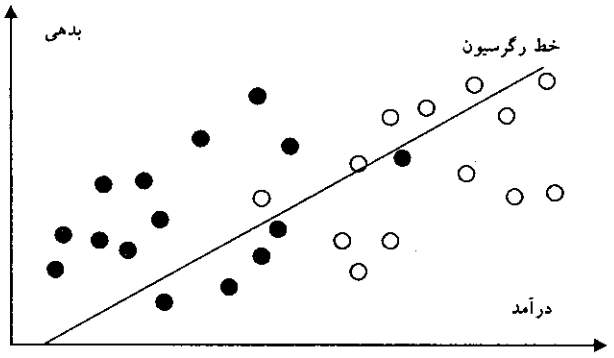


شکل ۱- ٦) مرزهای دسته‌بندی به روش نزدیکترین همسایه



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[۱] 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[۲] هستند.



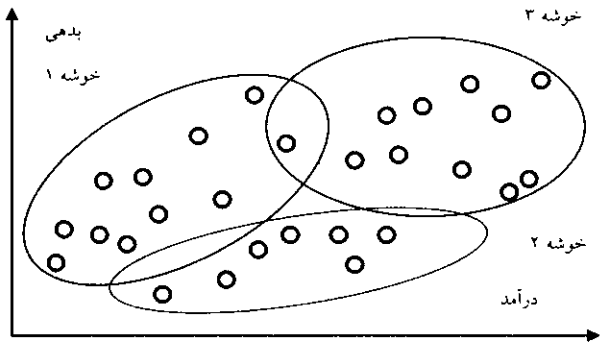
شکل ۱- ۸) یک رگرسیون خطی ساده برای مجموعه داده وام

شکل (۱-۸) نتایج یک رگرسیون خطی ساده را نشان می‌دهد. که در آن «بدهی» به‌عنوان تابعی خطی از «درآمد» برازش شده است: این برازش خوب نیست زیرا همبستگی ضعیفی بین دو متغیر وجود دارد.

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 با دایره سیاه و سفید نشان داده شده بود) با دوایر «توخالی» جایگزین شده‌اند تا نشان دهد که دیگر عضویت انحصاری به خوشه‌ها مطرح نیست.

۱- Biomass

۲- Time- Lagged



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

چرا *KDD* لازم است؟

برخی از دلایل نیاز به *KDD* به شرح زیر می‌باشند:

- 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 شده 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 تحلیلِ کلاسیک، بیش از اندازه بزرگ است. ممکن است نتوانیم آن را در حافظه کامپیوتر 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 روبه رشد 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.
- 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 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 فرموله 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)

۱-۶-۱- 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 و تولید

- کنترل و زمانبندی
- مدیریت شبکه
- تحلیل نتایج آزمایشات فنی
- اطلاعات علمی
- فهرست برداری تحقیقات مربوط به آسمان
- پایگاه داده‌های پزشکی
- زلزله یابی در زمین شناسی
- اطلاعات شخصی

۱-۶-۲- چالشهایی برای KDD

- **پایگاه داده بزرگتر:** پایگاه داده با صدها فیلد و جدول، میلیونها رکورد و اندازه‌های چند میلیارد بایتی کاملاً متداول هستند و استفاده از پایگاه داده ترابایتی ($۱۰^{۱۲}$ بایت) معمول می‌شود.
- **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 نیاز به تلاش برای استقراء مدل[۱] را به‌طور فزاینده‌ای بزرگ می‌کند. به‌علاوه این مشکل، یافتن شانسیِ الگوهای بدلی و جعلی[۲] را افزایش می‌دهد. چاره این مشکل استفاده از روشهای کاهش بعد مؤثر 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 به عملکرد ضعیف مدل روی داده‌های آزمون می‌شود.

۱- Model Induction

۲- Spurious

- **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 $N/1000$ 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 آماره آزمون را به‌عنوان تابعی از جستجو تنظیم می‌کنند.
- **داده‌ها و دانش در حال تغییر:** داده‌های سریعاً در حال تغییر و بی‌ثبات[1] 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 شده حذف و یا افزایش[2] 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[3] 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.
- **روابط پیچیده بین فیلدها:** ویژگیها یا مقادیر با ساختار سلسله مراتبی، روابط میان ویژگیها و نیز انواع روشهای پیچیده نمایش دانش، نیاز به الگوریتمهایی دارند 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.

---

[1]- Non Stationary

[2]- Augmented

[3]- Census

- **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 تعاملی[۱] 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 سیستمها تلفیق یا یکپارچه شود. نمونه‌های تلفیق عبارتند از تلفیق با *DBMS*[۲] (از طریق رابط پرس و جو)، تلفیق با صفحه گسترده‌ها و ابزارهای مصورسازی و همچنین می‌توان حسگرهایی را برای قرائت بلادرنگ داده‌ها با این سیستمها تلفیق نمود.

۱- Interactive

۲- 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, 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.

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 Minnig Techniques and Practice', Unesco Course (cited October 2004). Available from <URL:http://www.netnam.vn/unescocourse/knowlegde/know_frm.htm>.

9) Kaufman L, 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.

15) Ho T.B, Dam H.C. (2005) 'Introduction to Knowledge Discovery and Data Mining', Available from <URL: http://www.jaist.ac.jp/~bao/MOT-Ishikawa/MOT-Ishikawa.pdf>

فصل دوم

پیش‌پردازش و آماده‌سازی داده‌ها

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

۲-۱- انواع داده‌های مورد استفاده در داده‌کاوی

یک مجموعه داده از اشیاء داده تشکیل شده است. نامهای دیگر شیء داده عبارتند از رکورد، نقطه، بردار، الگو[3]، واقعه، مورد[4]، نمونه، مشاهده و یا موجودیت. هر شیء داده نیز با تعدادی

---

[1]- Garbage in Garbage Out

[2]- Estimated Predictive Performance

[3]- Pattern

[4]- Case

ویژگی توصیف می‌شود که خصوصیات اصلی آن شیء را بیان می‌کنند. نامهای دیگر ویژگی عبارتند از متغیر، خصیصه[۱]، فیلد، مشخصه[۲] و یا بُعد [۹].

مجموعه داده، اغلب یک فایل است که در آن اشیاء، رکوردهای (یا سطرهای) فایل بوده و هر فیلد (یا ستون) متناظر با یک ویژگی است.

۲-۱-۱- ویژگیهای کمی و کیفی

یک ویژگی خاصیتی از یک شیء داده می‌باشد که ممکن است از شیئی به شیئی دیگر یا از زمانی به زمان دیگر متفاوت باشد. برای مثال رنگ چشم برای افراد مختلف متفاوت است و دمای یک جسم در طول زمان تغییر می‌کند. رنگ چشم یک ویژگی نمادین[۳] بوده که دارای چند حالت محدود می‌باشد در حالی که دما یک ویژگی عددی با تعداد بالقوه نامحدودی مقدار است.

یک راه ساده و مفید برای تعیین نوع ویژگی، تشخیص خواص اعداد متناظر با آن ویژگی است. معمولاً خواص عددی ذیل برای توصیف ویژگیها استفاده می‌شود:

- تمایز = و ≠
- ترتیب <، >، =< و =>
- جمع‌پذیری + و −
- ضرب * و /

با توجه به این خواص می‌توان چهار نوع ویژگی تعریف کرد: اسمی[۴]، رتبه‌ای[۵] یا ترتیبی، فاصله‌ای[۶] یا بازه‌ای و نسبتی[۷] یا نسبی. جدول (۲-۱) شامل تعریف این انواع و عملیات آماری معتبر برای هر نوع است. در این جدول هر نوع ویژگی دارای خواص و عملیات متغیرهای

۱- Characteristic
۲- Feature
۳- Symbolic
۴- Nominal
۵- Ordinal
۶- Interval
۷- Ratio

متغیرهای بالایی خود است. مثلاً همه خواص و عملیات معتبر برای ویژگیهای اسمی، رتبه‌ای و فاصله‌ای برای ویژگیهای نسبتی نیز معتبر است.

جدول ۲- ۱) انواع ویژگی

نوع ویژگی		توصیف مقادیر ویژگی	مثال	عملیات
طبقه‌ای (کیفی)	اسمی	صرفاً نامهای متفاوتند و فقط اطلاعاتی برای تمایز اشیاء فراهم می‌کنند (=، ≠).	کد پستی، جنسیت شماره پرسنلی، رنگ چشم	مُد، آنتروپی، همبستگی توافقی، آزمون کای مربع
	رتبه‌ای	اطلاعات کافی برای مرتب کردن اشیاء فراهم می‌کند (<،>).	{خوب، بهتر، بهترین}، سطح تحصیلات	میانه، دهک، همبستگی رتبه‌ای، آزمونهای ردیف[1]، آزمونهای علامت
عددی (کمی)	فاصله‌ای	تفاوت بین مقادیر با معنی است یعنی واحد اندازه‌گیری وجود دارد (+،-).	تاریخ تقویم، درجه سانتیگراد	میانگین، انحراف معیار، همبستگی پیرسون، آزمون t و F
	نسبتی	هم تفاوت و هم نسبت با معنی است (*، /).	درجه کلوین، مقدار پول، سن، جرم، طول	میانگین هندسی، میانگین موزون، درصد تغییر

ویژگیهای اسمی و رتبه‌ای با هم به عنوان ویژگی طبقه‌ای یا اسمی شناخته می‌شوند. این ویژگیها واجد خواص عددی محدودی هستند. حتی اگر این ویژگیها با عدد مثلاً عدد صحیح بیان شوند با آنها باید به شکل نماد رفتار شود. دو نوع دیگر ویژگی شامل فاصله‌ای و نسبتی به عنوان ویژگی کمی یا عددی شناخته می‌شوند. ویژگیهای کمی با اعداد بیان شده و اکثر خواص اعداد را دارند. این ویژگیها می‌توانند مقدار صحیح یا پیوسته داشته باشند.

تفاوت این دو مقیاس فاصله‌ای با نسبتی در چگونگی قرارگیری نقطه صفر در مقیاس است. نقطه صفر در مقیاس فاصله‌ای به‌طور قراردادی و اختیاری تعریف شده است. بهترین مثال برای مقیاس فاصله‌ای مقیاس «حرارت» است. قرارگرفتن در نقطه صفر فارنهایت بیانگر این نیست که

---

[1]- Run tests

ابداً حرارت وجود ندارد. در مقیاس فاصله‌ای ، به دلیل 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.

بر عکس یک مقیاس نسبتی یک نقطه صفر مطلق دارد و در نتیجه نسبت مقادیر، واقعیت درستی از متغیر مورد اندازه‌گیری با این مقیاس را نشان می‌دهد. کمیتهایی همچون ارتفاع، طول و حقوق از این نوع مقیاس هستند.

۲-۱-۲- ویژگیهای گسسته و پیوسته

راه دیگر تفکیک ویژگیها، بر حسب تعداد مقادیری است که می‌توانند بگیرند.

- گسسته: ویژگی گسسته، مجموعه مقادیر محدود و یا نامحدود قابل شمارش دارد. این ویژگیها می‌توانند مانند کد پستی یا شماره پرسنلی از نوع طبقه‌ای باشند و یا مثل شمارش از نوع عددی باشند. ویژگیهای گسسته معمولاً با متغیرهای صحیح نمایش داده می‌شوند. ویژگیهای دودویی[۱] حالت خاصی از ویژگیهای گسسته‌اند که فقط دو مقدار مانند درست/غلط، بله/خیر، مرد/زن و یا ۱/۰ دارند. ویژگیهای دودویی اغلب به شکل متغیرهای بولی[۲] یا متغیرهای دارای دو مقدار ۰ و ۱ بیان می‌شوند.
- پیوسته: ویژگی پیوسته دارای مقادیری از نوع اعداد حقیقی است. برای مثال ویژگیهایی مانند دما، قد یا وزن پیوسته هستند. ویژگیهای پیوسته نوعاً با متغیرهای اعشاری با دقت محدود بیان می‌شوند.

به طور نظری، هر کدام از انواع مقیاسهای اندازه‌گیری (اسمی، رتبه‌ای، فاصله‌ای و نسبتی) می‌توانند با هر یک از انواع مقادیر عددی (دودویی، گسسته و پیوسته) ترکیب شوند. در عمل، برخی ترکیبات به ندرت اتفاق افتاده و چندان معنی ندارند. برای مثال ویژگی دودویی پیوسته چندان واقعی به نظر نمی‌رسد. ویژگیهای اسمی و رتبه‌ای نوعاً دودویی یا گسسته بوده و ویژگیهای فاصله‌ای و نسبتی نوعاً پیوسته هستند. البته ویژگیهای شمارشی که گسسته‌اند ویژگی نسبتی نیز می‌باشند.

---

۱ -Binary

۲ -Boolean

۲-۱-۳- ویژگیهای نامتقارن

در ویژگیهای نامتقارن فقط وجود (مقدار غیر صفر) مهم است. مجموعه داده‌ای را در نظر بگیرید که در آن هر شیء یک دانش‌جو و هر ویژگی نشان دهنده گذراندن یک درس می‌باشد. از آنجا که هر دانشجو فقط تعداد معدودی از دروس را می‌گذراند بیشتر مقادیر این مجموعه داده صفر می‌باشد. بنابراین همه دانشجویان شبیه هم به نظر می‌رسند. در این موارد بهتر است فقط روی مقادیر غیر صفر متمرکز شود. نتیجه مثبت آزمایش پزشکی مانند داشتن سرطان نیز از این قبیل موارد است. ویژگیهای دودویی نامتقارن به طور خاص در تحلیل قواعد تلازمی اهمیت دارند. مشخصه‌های نامتقارن گسسته یا پیوسته نیز وجود دارند. برای مثال اگر امتیاز عددی هر درس نیز ثبت شود، مجموعه حاصل حاوی ویژگیهای نامتقارن گسسته یا پیوسته است.

۲-۲- آماده سازی داده‌ها

آماده سازی داده‌ها برای داده‌کاوی هنر چلاندن[۱] و فشردن داده‌های موجود و بیرون کشیدن داده‌های با ارزش است. در حالیکه داده‌کاوی هنر کشف الگوهای معنی‌دار در 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.

۲-۲-۱- جایگاه آماده سازی داده‌ها در داده‌کاوی

در پروژه‌های داده‌کاوی، آماده‌سازی داده پس از مرحله فهم کسب و کار[۲] و فهم داده[۳] 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:

۱- Wringing

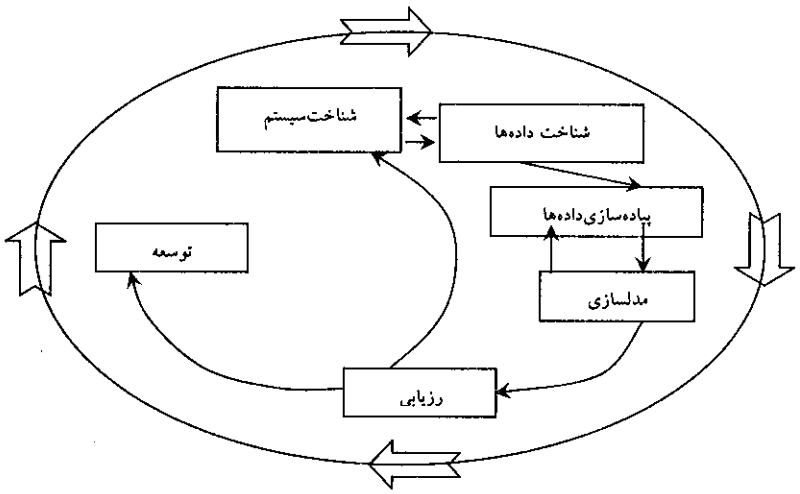
۲- Business Understanding

۳- Data Understanding

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؟

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، یک جدول داده یا صفحه گسترده‌ای با انواع متغیرهای ترتیبی، عددی و دودویی است.
- تهیه مشخصه‌هایی که منجر به بهترین کارایی مدل پیش‌بینی شود.



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

۲-۲-۲- چرا آماده‌سازی داده‌ها؟

آماده‌سازی داده‌ها، حدود ٦۰ 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 شماره‌های 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[۱] را دارند. می‌توان گفت داده‌ها در عالم واقع دارای آلودگیهای زیر هستند:

ناقص[۲]: مانند نمونه‌های ناکافی، کمبود برخی مقادیر مشخصه‌ها، داشتن نتایج به‌صورت تجمیع شده.

مغشوش[۳]: داده‌ها دارای خطا یا مقادیر پرت هستند. مثلاً سن = "۱۰-"

ناسازگار[۴]: 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، نرم‌افزاری و یا سخت‌افزاری 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.

۱- Dirty

۲- Incomplete

۳- Noisy

۴- 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 داده 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.

۲-۲-۳- تلخیص توصیفی داده‌ها [۱]

اگر پیش پردازش داده‌ها بخواهد موفق باشد باید تصویری جامع از داده‌های شما داشته باشد. فنون تلخیص توصیفی داده‌ها می‌توانند برای شناسایی مشخصه‌های داده 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[۲] و پراکندگی[۳] آنها بشناسیم.

معیار سنجش گرایش مرکزی شامل اندازه‌گیری میانگین[٤]، میانه[٥]، مد[٦] و میان دامنه[۷] است در حالی‌که سنجه‌های پراکندگی داده شامل چارکها[۸]، دامنه میان چارکی[۹] 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 که این سنجه‌ها در پایگاه داده‌های بزرگ چگونه به‌صورتی کارا محاسبه می‌شوند. برای مطالعه بیشتر این سنجه‌ها به [٤] مراجعه نمایید.

۱- Descriptive Data Summarization

۲- Central Tendency

۳- Dispersion

٤- Mean

٥- Median

٦- Mode

۷- Midrange

۸- Quartiles

۹- Interquartile range (IQR)

۲-۲-۴- نمایش گرافیکی داده‌های توصیفی

جدای ازگراف خطی[۱]، نمودار ستونی[۲] و نمودار کلوچه‌ای[۳] 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[٤]، و منحنی لورنس است. این گرافها برای بازرسی بصری داده‌ها بسیار مفید هستند. در مورد هر گراف توضیح مختصری داده می‌شود [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[۵]ی مجزا است.

جدول ۲-۲) داده‌های فروش

تعداد اقلام فروخته شده	قیمت واحد
۲۷۵	٤۰
۳۰۰	٤۳
۲۵۰	٤۷
--	--
۳٦۰	۷٤
۵۱۵	۷۵
۵٤۰	۷۸
--	--
۳۲۰	۱۱۵
۲۷۰	۱۱۷
۳۵۰	۱۲۰

۱- Line Graph

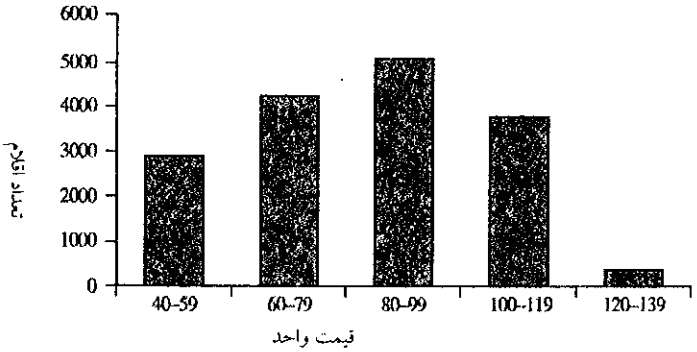
۲- Bar Chart

۳- Pie Chart

٤- Scatter Plot

۵- Bucket

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.



شکل ۲-۲) نمونه ای از یک هیستوگرام

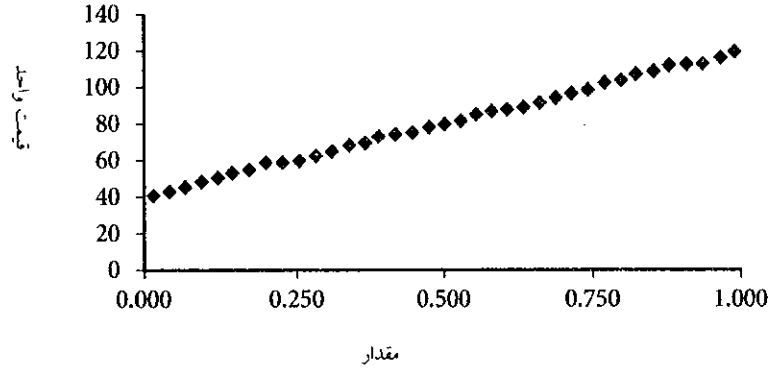
نمودار چندک[1]

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

---

[1]- Quantile Plot (Q-P)

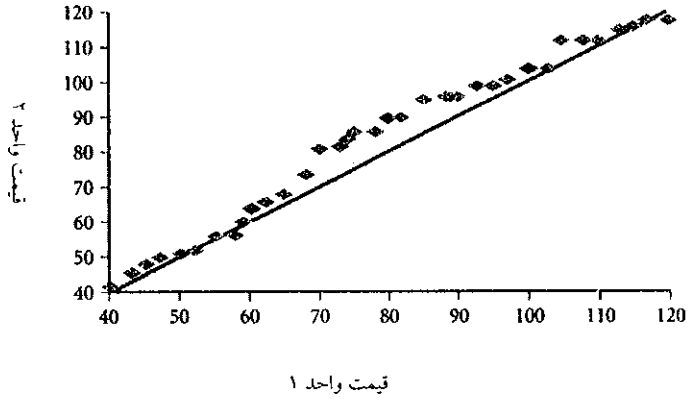
[2]- Quantile



شکل ۲-۳) نمونه ای از یک نمودار چندک

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.



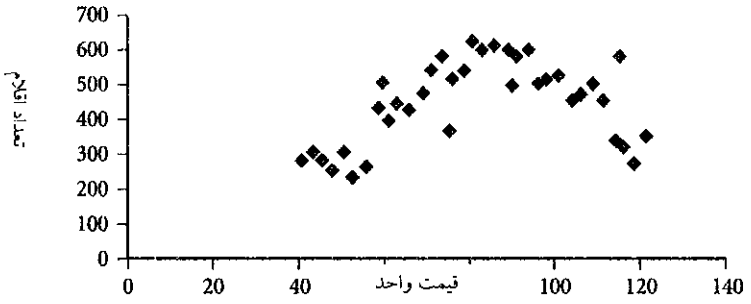
شکل ۲-٤) نمودار Q-Q

[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

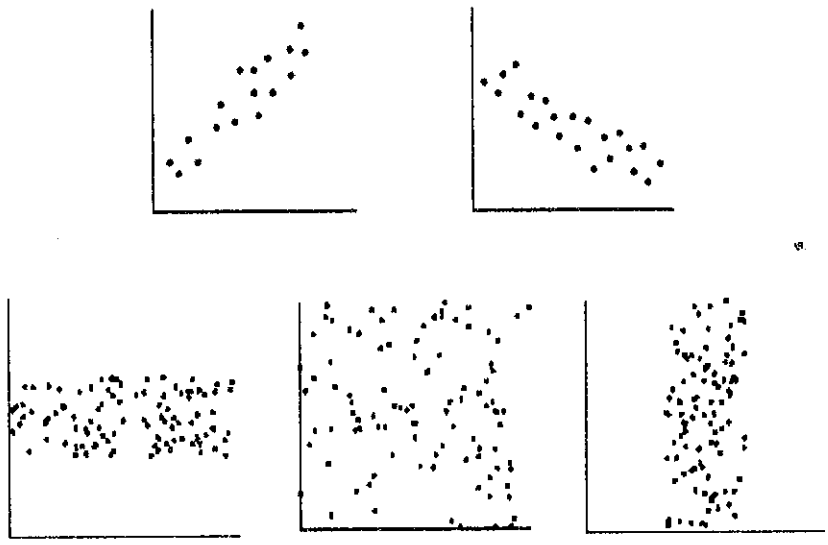
یکی از کارآمدترین روشهای گرافیکی برای تعیین وجود رابطه، الگو یا گرایش بین دو ویژگی عددی نمودار پراکنش است.



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.

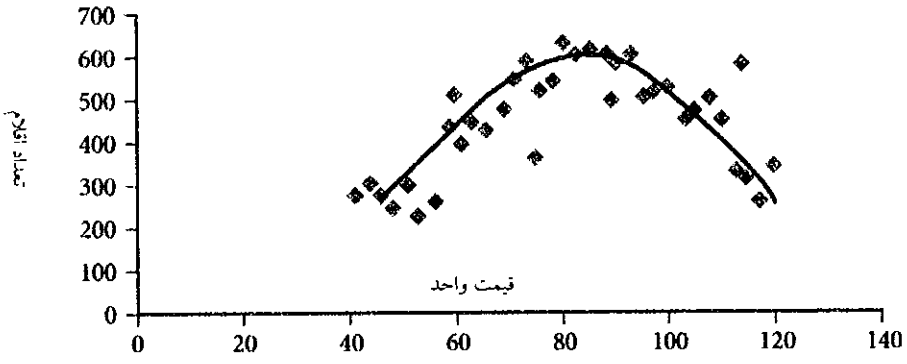
این همبستگی در صورت وجود می‌تواند به شکل مثبت یا منفی باشد. در شکل (۲-٦) نمودار پراکنش را برای دو ویژگی مشاهده می‌کنید. دو نمودار بالا به‌ترتیب بیانگر وابستگی منفی و مثبت و سه نمودار پایین بیانگر عدم وابستگی میان ویژگیها است.



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

۲-۲-۵- 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 بخشهای بعدی راجع به آنها به تفصیل صحبت خواهیم کرد.

الف. 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.

ب. یکپارچه‌سازی داده‌ها

یکپارچه‌سازی داده نقش مهمی‌در *KDD* بازی می‌کند. این عملیات شامل یکپارچه‌سازی چندین پایگاه داده ناهمگن بوده که قبلاً به‌وسیله چندین منبع ایجاد شده است.

ج. تبدیل داده

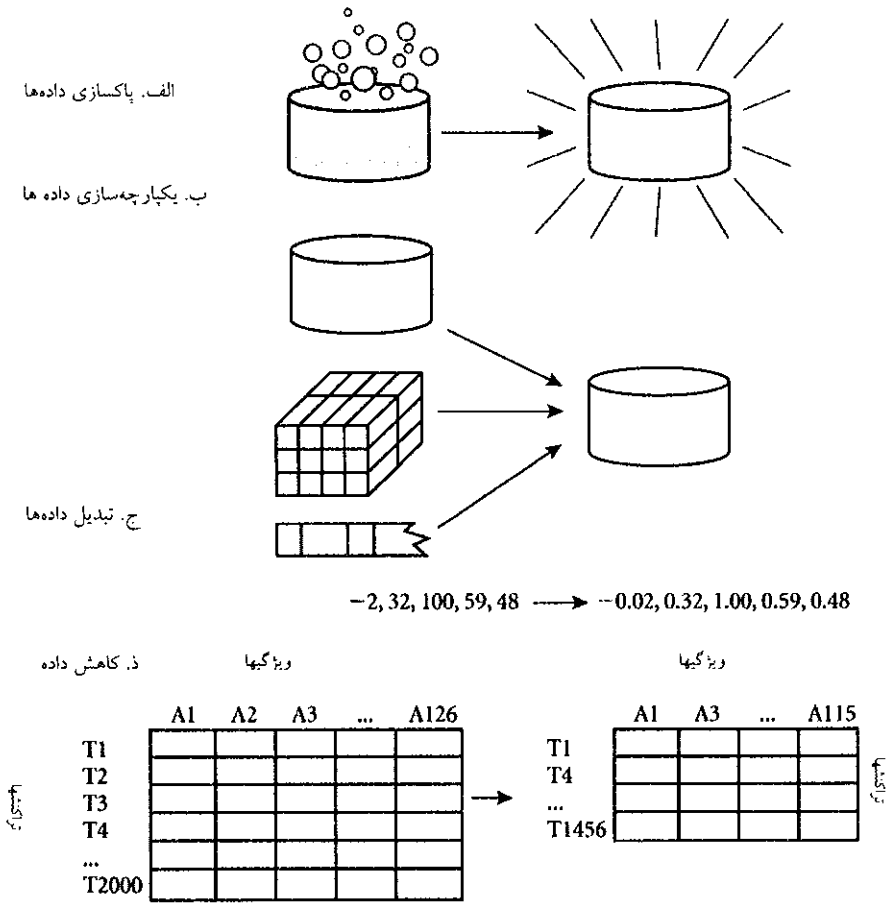
این کار شامل عملیاتی همچون هموارسازی، تجمیع و نرمال‌سازی است.

د. کاهش داده

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. کاهش داده می‌تواند سطری یا ستونی باشد.

ه. تصویر کردن[1] برای کاهش بُعد

تصویر کردن برای کاهش بُعد نوعی کاهش ستونی داده است با این فرق که در آن مشخصه‌های تغییر یافته جدیدی از روی مشخصه‌های اولیه ساخته می‌شوند. در کاهش بُعد از طریق تصویر کردن، تبدیلات و کدگذاریهایی روی داده انجام می‌شود که در نهایت بازنمایی کاهش یافته یا فشرده‌ای از داده‌های اصلی به‌دست می‌آید.



شکل ۲- ۸) عملیات مختلف در پاکسازی داده

[1]- Data Projection

برخی منابع مطالعاتی، فعالیت‌هایی همچون مستندسازی داده‌ها، طرح‌ریزی کدینگ و ورود داده را نیز از جمله کارهای مرتبط با آماده‌سازی داده دانسته‌اند. تمامی آنچه در مرحله پاکسازی انجام می‌شود در شکل (۲-۸) خلاصه شده است. در این شکل، تصویر کردن که موجب ایجاد ویژگیهای جدید می‌شود نشان داده نشده است. خروجی این مرحله، یک فایل آماده برای کار[۱] است.

۲-۳- پاکسازی داده‌ها

رالف کیمبال[۲] پاک‌سازی داده را یکی از سه مسئله بزرگ در انبارش داده دانسته است. گروه دی‌سی‌آی[۳]، پاکسازی داده را به‌عنوان مسئله اول در انبارش 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.

۲-۳-۱- 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.

۱- Work File

۲- Ralph Kimball

۳- DCI Group

برخی منابع مطالعاتی، به‌دست آوردن داده و ایجاد فراداده را نیز از جمله وظایف این مرحله دانسته‌اند. این داده‌ها می‌تواند در یک پایگاه داده یا در یک فایل تک‌جدولی[۱] قرارگرفته باشد. البته در چنین حالتی مقدار داده‌های یک ویژگی یا از روی تعداد ستون داده‌ها و یا 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، رتبه‌ای، عددی)، جداول ترجمه کدها برای فیلدهای اسمی و هم‌چنین نقش فیلد (ورودی، هدف و شناسه کمکی) بخشی از اطلاعاتی است که از فراداده قابل دستیابی است.

مقادیرمفقوده[۲]

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[۳]: معمولاً وقتی رکورد حذف می‌شود که برچسب دسته گم شده باشد (به فرض که کار داده‌کاوی نوعی دسته‌بندی باشد). یکی از ایرادات این شیوه کاهش اندازه

۱- Flat

۲- Missing Values

۳- List Wise

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 آماره‌هایی روی ستون حاوی مقادیر مفقوده محاسبه می‌شود. ایراد این شیوه این است که اندازه نمونه هر آماره‌ای (مانند میانگین، واریانس و کواریانس) که حساب می‌شود متفاوت 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: مقدار مفقوده با میانگین نمونه‌های دارای برچسب دسته مشابه رکورد فعلی جایگزین می‌شود.
- مقادیر با احتمال بیشتر: با استفاده از رابطه‌های بیزی، درخت تصمیم‌گیری و یا رگرسیون می‌توان مقدار مفقوده را پیش‌بینی 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 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

---

[1]- Biased

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 قواعدی 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، داده‌های آن‌را هموار کرد. از آنجا که این روش از همسایه‌های مقادیر استفاده می‌کند، بنابراین هموارسازی محلی است. گسسته‌سازی 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.

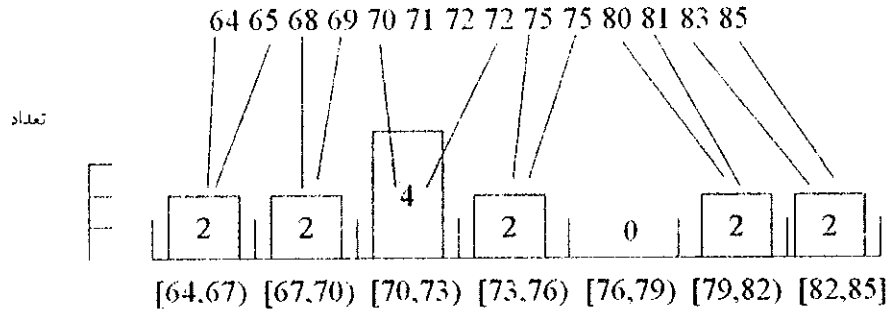
مثال: داده‌های زیر، درجه حرارت محیط در یک دوره است. می‌خواهیم آنها را به شیوه عرض ثابت گسسته کنیم. شکل (۲-۹) نمایش‌گر گسسته‌سازی به شیوه عرض ثابت این اعداد است. ٦٤,٧٠,٨١,٦٨,٨٥,٧١,٨٣,٦٥,٧٢,٧٢,٧٥,٦٩,٧٥,٨٠

---

[1]- Smooth

[2]- Binning

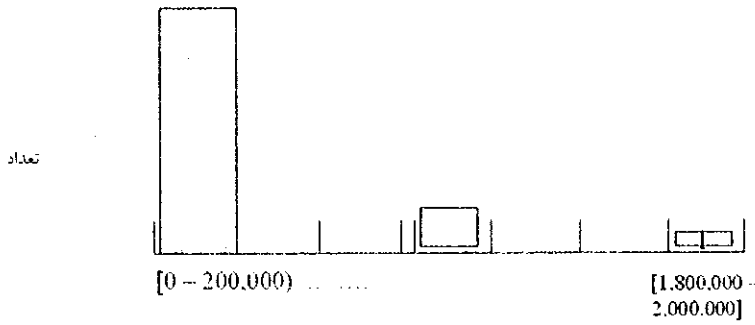
[3]- Equal-width



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.

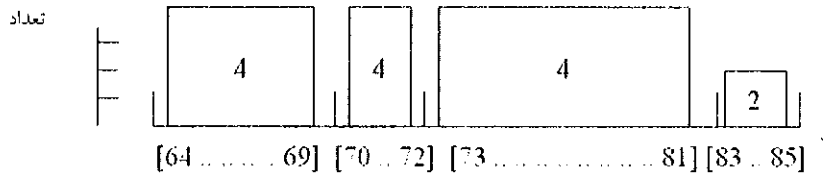
شکل (۲-۱۰) نمونه‌ای را نشان می‌دهد که گسسته‌سازی با روش عرض ثابت بر روی داده‌های حقوق یک شرکت باعث ایجاد دسته‌های جدا می‌شود.



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

عمق ثابت[1]

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



شکل ۲- ۱۱) گسسته‌سازی عمق ثابت

داده‌های مثال بالا در شکل (۲-۱۱) 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.

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 میانگین و مرزها 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 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

---

[1]- Equal-depth

بسته دوم: ۲۳,۲۳,۲۳,۲۳

بسته سوم: ۲۹,۲۹,۲۹,۲۹

هموارسازی به‌وسیله مرز بسته‌ها:

بسته اول: ٤,٤,٤,١٥

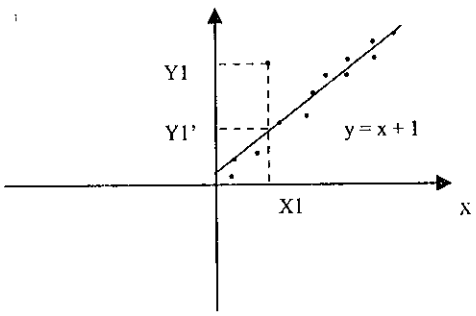
بسته : ۲۱,۲۱,۲۵,۲۵

بسته سوم: ۲٦,۲٦,۲٦,۳٤

در هموارسازی به‌وسیله میانگین، به جای تمام اعضای بسته، مقدار میانگین هر بسته قرار می‌گیرد. 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

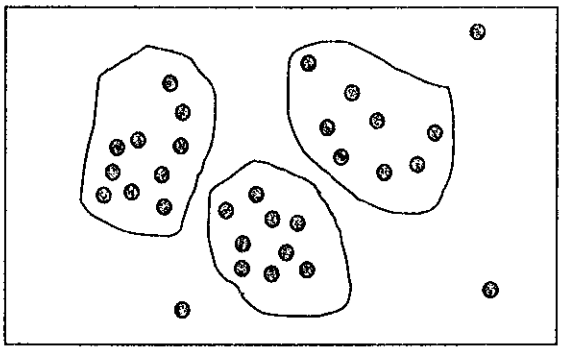
داده را می‌توان از راه تطبیق دادن داده با یک تابع مانند تابع رگرسیون که برای ویژگی به‌دست آمده، هموار کرد. رگرسیون خطی، بهترین خطی که بر دو ویژگی (یا متغیر) تطبیق کند را به‌گونه‌ای که مقدار یکی بتواند برای پیش‌بینی مقدار دیگری به‌کار رود، می‌یابد. رگرسیون چند متغیره خطی نیز توسعه یافته همین رگرسیون خطی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



شکل ۲- ۱۲) 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.



شکل ۲-۱۳) استفاده از خوشه‌بندی برای هموارسازی

این سه مقدار به‌عنوان اغتشاش شناسایی می‌شوند. بسیاری از روشهای هموارسازی، از جمله روشهایی که در گسسته‌سازی استفاده می‌شوند، روشهای کاهش داده نیز محسوب می‌گردند.

۲-۳-۲- 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[۱] است. این مغایرتها می‌تواند دلایل زیادی داشته باشد. از جمله می‌توان به طراحی ضعیف فرم ورود داده به‌دلیل داشتن تعداد فراوان فیلدهای اختیاری یا خطای انسانی در ورود داده، خطاهای عمدی[۲] (وقتی کسی نمی‌خواهد درباره خودش اطلاعات بدهد) و داده‌های تاریخ

۱- Discrepancy Detection

۲- Deliberate Errors

مصرف گذشته[۱] (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[۲] نیز خود مشکل دیگری است که می‌تواند ناشی از بیتهای بلااستفاده در تعریف فیلدها باشد. داده‌ها همچنین باید به‌گونه‌ای تعریف شوند که قانون یکتایی[۳] را رعایت کنند. یعنی مقادیر ویژگی (برای کلید اصلی) تکرار نشود. همچنین رعایت دیگر قواعد پایگاه داده از آن جمله قانون تهی[۴] (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[۵] به‌وسیله دانش ساده در مورد دامنه (به‌عنوان

۱- Data Decay
۲- Field Overloading
۳- Unique Rule
۴- Null Rule
۵- Data Scrubbing

مثال کد پستی، چک کردن املای کلمه) و ابزارهای ممیزی داده[۱] 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[۲] 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[۳] 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)[۴] دسترسی کاربر به تبدیلات را با استفاده از رابط گرافیکی کاربر میسر می‌سازند. البته این ابزارها اغلب تعداد محدودی از تبدیلات را پشتیبانی 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.

۲-۴- یکپارچه‌سازی داده‌ها

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].

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

۱- Data Auditing

۲- Data Transformations

۳- Data Migration Tools

۴- ETL (Extraction/Transformation/Loading)

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^2) استفاده می‌کنیم. پس از قرار دادن مقادیر در جدول تصادفی[1] مقدار آماره X^2 را به‌دست می‌آوریم. در صورتی‌که این مقدار از مقدار بحرانی که برای درجه آزادی $(r-۱)\times(c-۱)$ 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.

---

[1]- Contingency Table

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.

$$X^2 = \sum_{i=1}^{c} \sum_{j=1}^{r} \frac{(o_{ij} - e_{ij})^2}{e_{ij}} \qquad (۲-۳)$$

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:

$$e_{ij} = \frac{count(A = a_i) \times count(B = b_j)}{N} \qquad (۲-٤)$$

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 (۲-۳) 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_{11} به‌صورت زیر محاسبه می‌شود:

$$e_{11} = \frac{count(male) \times count(fiction)}{N} = \frac{۳۰۰ \times ۴۵۰}{۱۵۰۰} = ۹۰,$$

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:

$$X^2 = \frac{(250-90)^2}{90} + \frac{(50-210)^2}{210} + \frac{(200-360)^2}{360} + \frac{(1000-840)^2}{840}$$

$$= 284.44 + 121.90 + 71.11 + 30.48 = 507.93$$

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.

۲-۵- تبدیل داده‌ها

در این مرحله داده‌ها به شکل مناسب برای داده‌کاوی تبدیل می‌شوند.

۲-۵-۱- 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.

---

[1]- Smoothing

[2]- Sigmoid Scaling

۲-۵-۲- تجمیع[۱]

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 [۲] استفاده می‌شود.

۲-۵-۳- تعمیم[۳]

در تعمیم 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.

۲-۵-۴- ساخت ویژگی[۴]

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.

۲-۵-۵- نرمال‌سازی[۵]

نرمال‌سازی تغییر مقیاس داده‌ها به گونه‌ای است 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 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

---

[۱] Aggregation

[۲]- Data Cube

[۳] Generalization

[۴]- Attribute Construction

[۵]- Normalization

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[1] را بررسی می‌کنیم:

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:

$$v' = \frac{v - min_A}{max_A - min_A}(new - max_A - new - min_A) + new - min_A \qquad (۵-۲)$$

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 ۷۳۶۰۰ 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:

$$v' = \frac{v - \overline{A}}{\sigma_A} \qquad (۶-۲)$$

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.

---

[1]- Normalization by decimal scaling

مثال: فرض کنید که میانگین و انحراف استاندارد ویژگی درآمد ۵٤۰۰۰ و ۱٦۰۰۰ است. با نرمال‌سازی *Z-Score* مقدار ۷۳٦۰۰ 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:

$$v' = \frac{v}{10^j} \qquad (۷-۲)$$

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

مثال: فرض کنید مقادیر ثبت شده مرتبط با ویژگی A است که دامنه آن از ۹۸٦- تا ۹۱۷ است. حداکثر قدر مطلق مقادیر A مقدار ۹۸٦ است. برای نرمال کردن از طریق مقیاس‌بندی اعشاری، ما هر مقدار را بر ۱۰۰۰ $(j=۳)$ تقسیم می‌کنیم. بنابراین مقدار ۹۸٦- به ۰,۰۹۸٦- و ۹۱۷ به ۰,۹۱۷ تبدیل می‌شود.

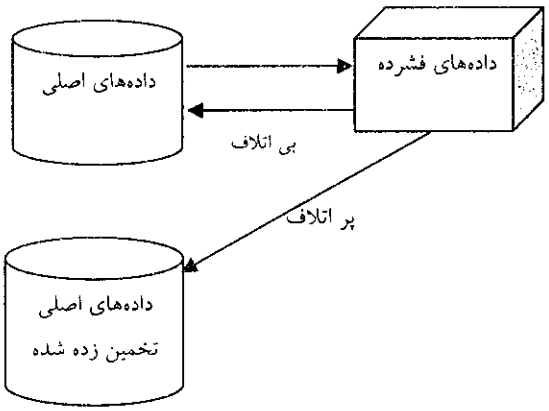
توجه کنید که در دو شیوه اول لازم است مقادیری از روی داده‌ها به‌دست آمده (مثل میانگین و انحراف استاندارد) و برای نرمال ساختن مقادیر بعدی استفاده شود.

۲-۶- کاهش داده‌ها

اگر بدون از دست دادن داده‌ها، داده‌های اصلی از داده‌های فشرده قابل بازسازی باشد این کاهش داده، بدون اتلاف[۱] 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[۲] می‌گویند.

۱- Lossless

۲- Lossy



شکل ۲- ۱٤) فشرده سازی بی اتلاف و پر اتلاف

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

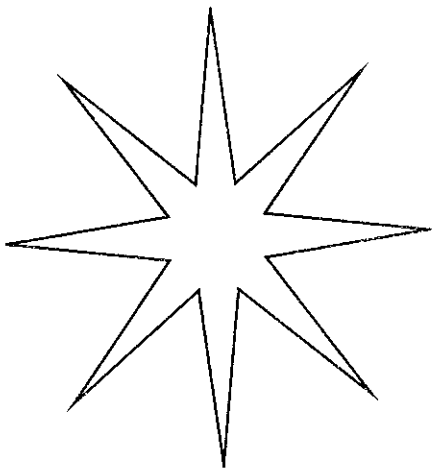
این ابعاد اضافی در مجموعه داده‌های بسیار بزرگ باعث ایجاد مشکلی می‌شوند که در ادبیات داده‌کاوی به آن «مصیبت بُعد[۲]» گفته می‌شود. این مسائل به علت حجم بالای داده‌ها در فضایی با ابعاد بالا ایجاد شده و مشکلاتی برای داده‌کاوی ایجاد می‌کرد. به خاطر ذهنیت و تجربه گذشته ما نسبت به فضایی با ابعاد دو یا سه بُعد، اغلب فضایی با ابعاد بالا برای ما غیر منتظره است. به‌طور مفهومی 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[۳] (مکعب چهار بُعدی) شبیه یک خارپشت شکل (۲-۱٤) است. هر چه تعداد ابعاد بیشتر شود، لبه‌ها بیشتر می‌شوند.

۱- Dimension

۲- The Curse Of Dimensionality

۳- Hypercube



شکل۲-۱۵) 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.

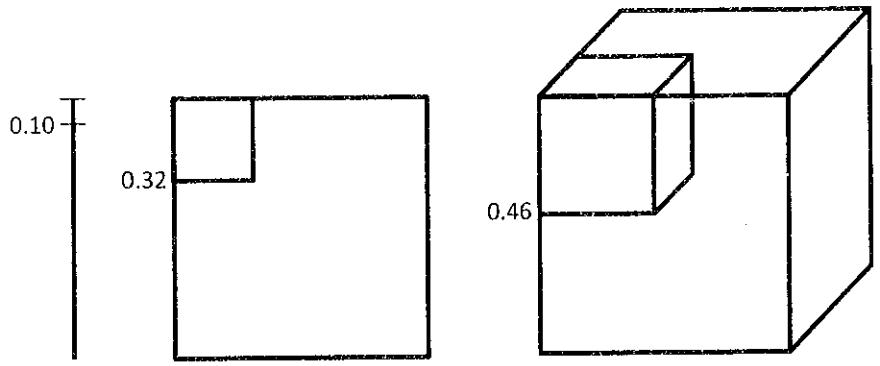
برای مثال اگر در یک نمونه یک بُعدی، N نقطهٔ داده در یک سطح تراکم وجود داشته باشد، برای رسیدن به همان تراکم در یک فضای k بُعدی، نیاز به N^k نقطه است. اگر مقادیر صحیح ۱ تا ۱۰۰ مقادیر نمونه یک بُعدی هستند برای به‌دست آوردن همان تراکم از نمونه‌ها در فضای ۵ بُعدی ما نیاز به $۱۰۰^۵ = ۱۰^{۱۰}$ نمونه متفاوت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

$$e(P) = P^{۱/d} \qquad (۸-۲)$$

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 ۲-۱٦) منطقه ای که ده درصد داده‌ها را پوشاند 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:

$$D(d,n)=۱/۲(۱/n)^{۱/d} \qquad (۹-۲)$$

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

انتخاب زیرمجموعه مشخصه‌ها[۲] **(کاهش ستونی)**: وقتی ابعاد با ویژگیهای نامربوط یا با ارتباط ضعیف یا افزونه شناسایی و حذف شوند.

کاهش تعدد نقاط[۳] **(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.

۱- Data Cube Aggregation

۲- Attribute Subset Selection

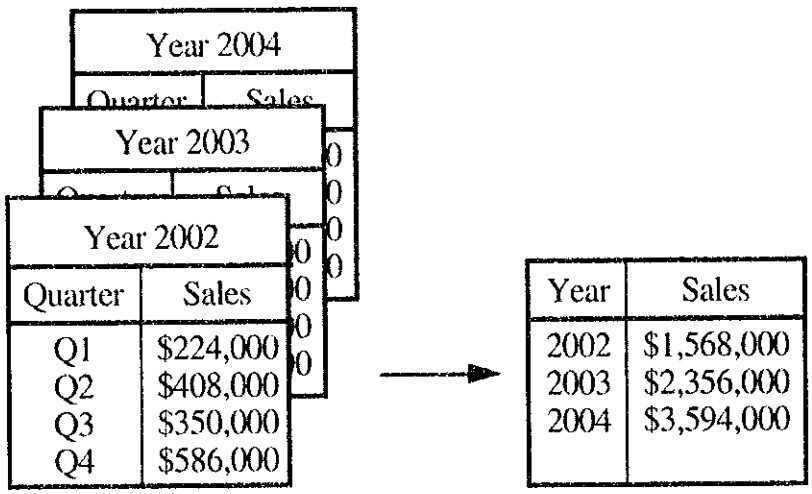
۳- Numerical Reduction

کاهش بُعد[1] **(کاهش ستونی)**: جایی‌که مکانیزم‌های کدکردن برای کاهش اندازه مجموعه داده استفاده می‌شود.

۲-۶-۱- تجمیع مکعبی 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.

به‌کارگیری 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.



شکل ۲-۱۷) تجمیع داده‌های مکعب داده

[1]- Dimensionality Reduction

۲-۶-۲- انتخاب زیرمجموعه مشخصه‌ها

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 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 سنود اطلاعاتی[3] 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.

---

[1]- Subsets Versus Supersets

[2]- Explore

[3]- Information Gain

لفاف[1]: 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.

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

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

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

---

[1]- Wrapper

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.

---

[1]- Redundant

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.

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

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.

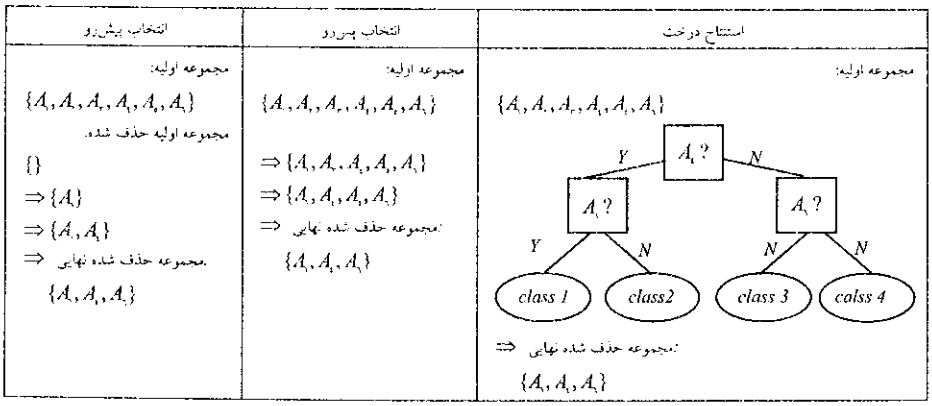
روش لفاف

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

۱- انتخاب گام‌به‌گام پیش‌رو[۱]: فرآیند با یک مجموعه تهی از ویژگیها به‌عنوان مجموعه کاهش یافته[۲] آغاز می‌شود. در هر گام تکرار، بهترین ویژگیهای اصلی انتخاب شده و به مجموعه قبلی اضافه می‌شوند.

۲- انتخاب گام‌به‌گام پس‌رو[۳] : فرآیند با مجموعه‌ای شامل تمام ویژگیها آغاز به‌کار می‌کند و در هر گام، بدترین ویژگیها از مجموعه حذف می‌شود.

۳- ترکیب دو روش انتخاب پیش‌رو و حذف پس‌رو[۴]: دو روش قبل به‌نحوی با هم ترکیب می‌شوند که در هرگام بهترین ویژگی اضافه شده و ویژگی بدتر حذف می‌شود.



شکل۲-۱۸) روشهای مختلف انتخاب زیرمجموعه ویژگیها

۴- استنتاج درخت تصمیم[۱]: الگوریتمهای درخت تصمیم در واقع در نقطه انشعاب درخت، بهترین ویژگی را نیز انتخاب می‌کنند.

۱- Stepwise Forward Selection

۲- Reduced Set

۳- Stepwise Backward Elimination

۴- Combination of Forward Selection and Backward Elimination

۲-۶-۳- کاهش تعدد نقاط

روشهای کاهش تعدد[۲] 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[۳] و نمونه مدلهای ناپارامتریک، هیستوگرام، خوشه‌بندی و نمونه‌برداری آماری است. بسیاری از این روشها در هموارسازی مطرح شدند.

۲-۷- تصویر کردن برای 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 فصل داده‌کاوی سری‌های زمانی مطالعه شوند. در کاهش بُعد از طریق تصویر کردن، تبدیلات و کدگذاریهایی روی داده انجام می‌شود که در نهایت بازنمایی کاهش یافته یا فشرده‌ای از داده‌های اصلی به‌دست می‌آید [٤]. تصویر کردن با انتخاب مشخصه متفاوت است. در انتخاب مشخصه، مشخصه‌های جدید زیرمجموعه‌ای از مشخصه‌های اصلی هستند در حالی که در تصویر کردن، مشخصه‌های جدید ترکیبی خطی یا غیرخطی از مشخصه‌های اولیه می‌باشند.

۱- Decision Tree Induction

۲- Numerisity

۳- Log Linear

۲-۷-۱- تعاریف و مفاهیم کاهش بعد

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.

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:

۱- Discrete Fourier Transform: DFT

۲- DWT

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[۱] حل نمود. نتیجه دقیقاً مانند نقشه ایران نخواهد شد، چرا که ممکن است نقاط یک جهت قرار دادی داشته باشند. سایر روشهای حفظ عمومی، شامل روشهای تجزیه مقدار منفرد، نگاشت سریع و روشهای 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.

۲-۷-۲- تحلیل مؤلفه‌های اصلی

تحلیل مولفه‌های اصلی[۲] 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 (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.

۱- Multidimensional Scaling: MDS

۲- Principal Component Analysis: PCA

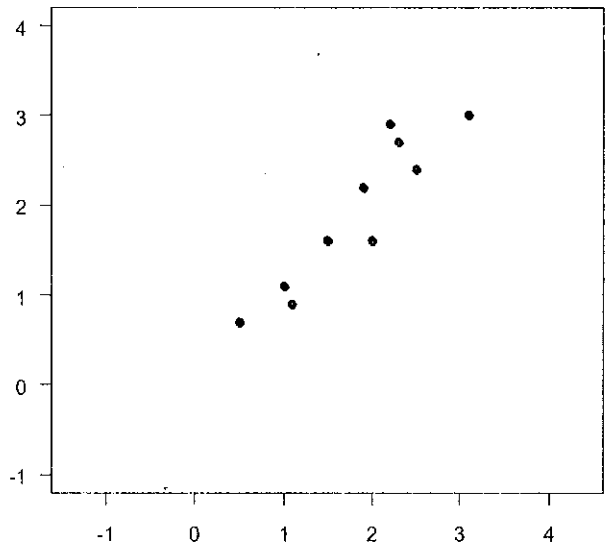
جدول ۲-٤) داده‌های اصلی در سمت چپ، داده‌های حاصل از تفاضل با میانگین در سمت راست،

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

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



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.

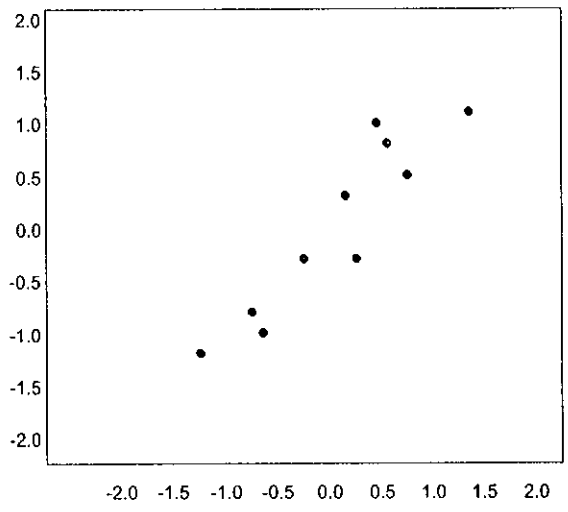
- **گام چهارم:** محاسبه بردارهای ویژه و مقادیر ویژه ماتریس کوواریانس

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 داده‌های 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

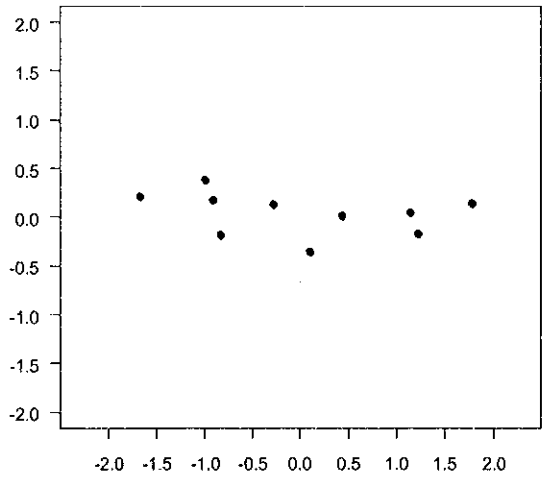
این مرحله، آخرین گام و در عین حال ساده‌ترین مرحله در *PCA* می‌باشد. در این مرحله ماتریس مشخصه را ترانهاده کرده و در مجموعه داده اصلی ضرب می‌نماییم:

Final Data = Row Feature Vector × 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.

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



شکل ۲-۲۱) یک نمودار از نقاط داده جدید

جدول ۲-۶) داده بعد از تبدیل بوسیله مهمترین بردار ویژه

X
-۰/۸۲۷۹۷۰۱۸۶
۱/۷۷۷۵۸۰۳۳
-۰/۹۹۲۱۹۷۴۹۴
-۰/۲۷۴۲۱۰۴۱۶
-۱/۶۷۵۸۰۱۴۲
-۰/۹۱۲۹۴۹۱۰۳
۰/۰۹۹۱۰۹۴۳۷۵
۱/۱۴۴۵۷۲۱۶
۰/۴۳۸۰۴۶۱۳۷
۱/۲۲۳۸۲۰۵۶

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.

۲-۷-۳- 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، 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.

۲-۷-۴- 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.

$$\sqrt{n} \sum_{i=0,...,n-1} S_i e - j 2\pi fi / n \qquad (۲-۱٤)$$

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.

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

---

[1]- Singular Value Decomposition: SVD

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.

۲-۷-۵- 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 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

---

[1]- Discrete Wavelet Transform (DWT)

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_۲*، *Daubechies_٤* و *Daubechies_٦* است. به‌کارگیری یک تبدیل موجک گسسته از یک الگوریتم هرمی سلسله مراتبی پیروی می‌کند:

در این تبدیل یک توالی به طول 2^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 2^n-1 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.

$$S=(1,3,5,11,12,13,0,1)$$

ابتدا این اعداد را به‌صورت جفت جفت با هم جمع می‌کنیم. $(4,16,25,1)$

همچنین اختلاف این جفتها را نیز محاسبه می‌کنیم. $(-2,-6,-1,-1)$

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 $\frac{4-2}{2}=1$ می‌شود که عنصر اول S است و $\frac{4-(-2)}{2}=3$ 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.

Resolution	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
۳	a_1+a_2	a_3+a_4	a_5+a_6	a_7+a_8	a_1-a_2	a_3-a_4	a_5-a_6	a_7-a_8
۲	$a_1+a_2+a_3+a_4$		$a_5+a_6+a_7+a_8$		$(a_1+a_2)-(a_3+a_4)$		$(a_5+a_6)-(a_7+a_8)$	
۱	$a_1+a_2+a_3+a_4+a_5+a_6+a_7+a_8$				$(a_1+a_2+a_3+a_4)-(a_5+a_6+a_7+a_8)$			

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

درخت تجزیه تبدیل موجک *Haar* برای یک رشته به طول ۸ در شکل (۲-۲۲) نشان داده شده است. مراحل اجرای این تبدیل برروی رشته S را می‌توانید در شکل (۲-۲۳) مشاهده کنید. حاصل‌جمع به‌دست آمده در آخرین مرحله، به‌همراه حاصل تفریق‌هایی که در تمام مراحل ذخیره شده، به‌عنوان خروجی این تبدیل در نظر گرفته می‌شود. بنابراین:

$$DWT(S)=(۴۶,-۶,-۱۲,۲۴,-۲,-۶,-۱,-۱)$$

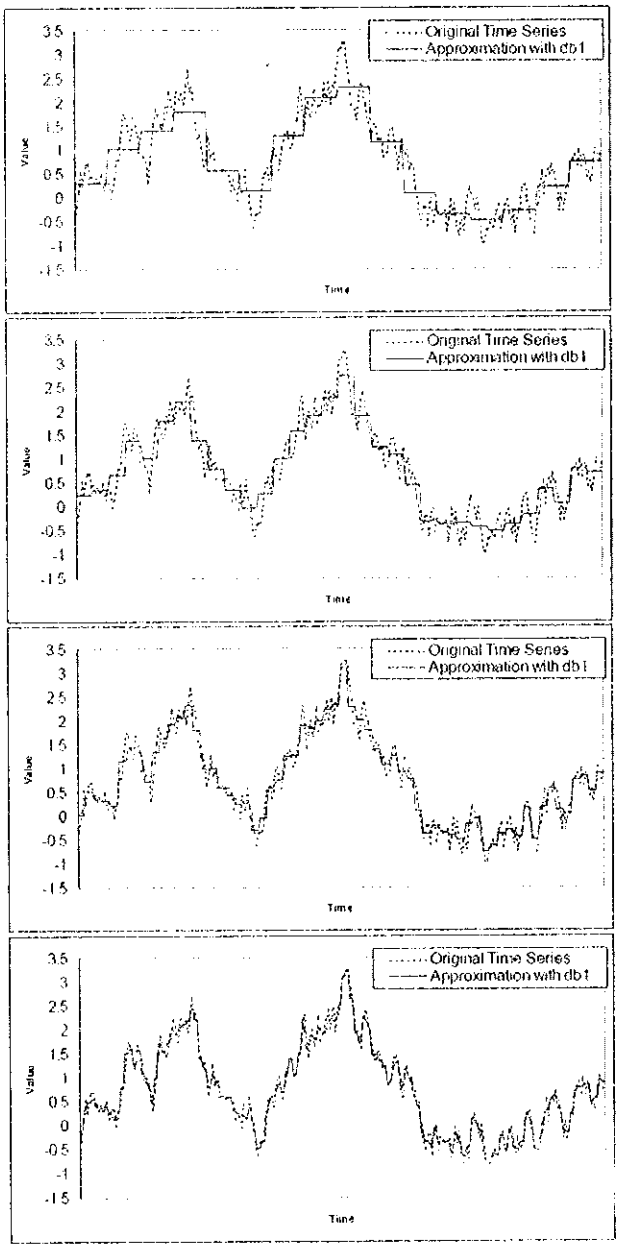
می‌توان دید که پیچیدگی زمانی این الگوریتم برای یک رشته به طول n برابر با $O(n)$ می‌باشد.

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

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*=۳ هستند، یعنی $(-۱,-۱,-۶,۲)$) 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

۲-۷-۶- تصویر کردن تصادفی[۱]

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.

۲-۷-۷- نگاشت سریع

یک تخمین و روش بسیار شبیه به روش مقیاس‌گذاری چند بعدی (*MDS*)، روش نگاشت سریع[۲] 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.
- یک روش کاهش بعد بهینه نیست.

روش کار الگوریتم نگاشت سریع:

- دو شیء که بیشترین فاصله را نسبت به هم دارند، پیدا می‌کند.

۱- Random Projection

۲- FastMap

- 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 $k-1$ مرتبه ادامه می‌دهد. [۶]

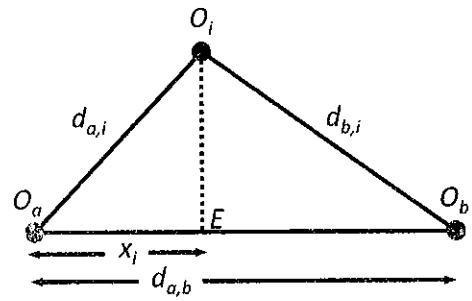
- مفاهیم و نمادهای مرتبط در زیر تعریف شده‌اند:

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_2 بردار X	$\|X\|_2$	=
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.

$$d_{b,i} = d_{a,i}^2 + d_{a,b}^2 - 2x_i d_{a,b} \qquad (۲-۱۶)$$

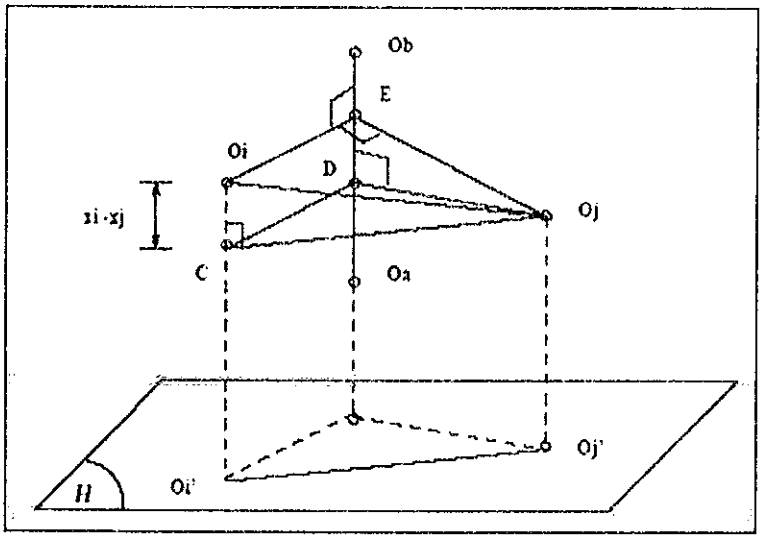
$$x_i = \frac{d_{a,i}^2 + d_{a,b}^2 - d_{b,i}^2}{2d_{a,b}} \qquad (۲-۱۷)$$



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 H یک فضای $(n-1)$ 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.

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



شکل ۲- ۲٦) 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(۱،۰),۰)* 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 *Ob*ها را در هر مرحله ذخیره می‌کند.

۰=# *Int col* که به ستونی از ارائه X که تازه به‌روز شده است اشاره می‌کند.

الگوریتم *Fastmap(K,D(۱،۰))*

- اگر ۰«*K* 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 ۰ =X[i,col#] 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×K

[PA] با بعد ۲×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

$S_1 = \{1,2,3\}$

$S_2 = \{1,1,4\}$

$S_3 = \{2,1,2\}$

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

$$\begin{matrix} S_1 \\ S_2 \\ S_3 \end{matrix}\begin{bmatrix} 0 & \sqrt{2} & \sqrt{3} \\ \sqrt{2} & 0 & \sqrt{5} \\ \sqrt{3} & \sqrt{5} & 0 \end{bmatrix}$$

$$S_1 \quad S_2 \quad S_3$$

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.

$$PA = \begin{bmatrix} 2 & PA_{12} \\ 3 & PA_{22} \end{bmatrix}$$

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

$$x_{11} = \frac{2+5-3}{2\sqrt{5}}$$

$$x_{21} = 0$$

$$x_{31} = \sqrt{5}$$

و در ستون اول این ماتریس جای می‌دهیم.

$$X = \begin{bmatrix} 2\sqrt{5} & x_{12} \\ 0 & x_{22} \\ \sqrt{5} & x_{32} \end{bmatrix}$$

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

$$(S'_1S'_2)^2 = (\sqrt{2})^2 - (\frac{2\sqrt{5}}{5})^2 = 2 - \frac{4}{5} = \frac{6}{5}$$

$$(S'_1S'_3)^2 = (\sqrt{3})^2 - \frac{9}{5} = \frac{6}{5}$$

$$(S'_2S'_3) = 5 - 5 = 0$$

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)

$$PA = \begin{bmatrix} 2 & 1 \\ 3 & 3 \end{bmatrix}$$

حال به محاسبه x_{i2}ها می‌پردازیم.

$$x_1 = 0$$

$$x_2 = \frac{\sqrt{30}}{5}$$

$$x_3 = \frac{\sqrt{6}}{\sqrt{5}} = \frac{\sqrt{30}}{5}$$

آنها را در ستون دوم ماتریس X جای می‌دهیم:

$$X = \begin{bmatrix} \frac{2\sqrt{5}}{5} & 0 \\ 0 & \frac{\sqrt{30}}{5} \\ \sqrt{5} & \frac{\sqrt{30}}{5} \end{bmatrix}$$

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

$$O_1 = (\frac{2\sqrt{5}}{5}, 0)$$

$$O_2 = (0, \frac{\sqrt{30}}{5})$$

$$O_3 = (\sqrt{5}, \frac{\sqrt{30}}{5})$$

۲-۷-۸- مقیاس‌گذاری چند بعدی

مقیاس‌گذاری چند بعدی نامی کلی برای گروهی از رویه‌ها و الگوریتمها می‌باشد که با یک ماتریس مجاورت[۱] ترتیبی شروع کرده و یک پیکره‌بندی از نقاط در یک، دو یا سه بعد ایجاد می‌کند. سمون[۲] و کراسکال[۳] 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[۴] کنیم کمک بزرگی خواهد بود. همچنین با اینکه داده‌ها معمولاً با بعد زیادی بیان می‌شوند ولی بعد ذاتی[۵] 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 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

۱- Proximity

۲- Sammon, 1969

۳- Kruskal, 1971

۴- Project

۵- Intrinsic

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:

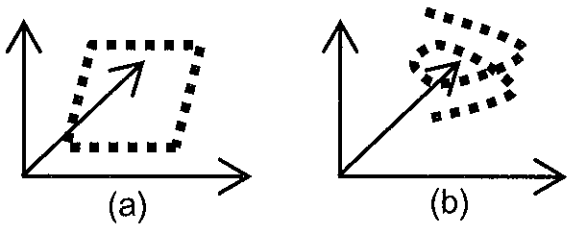
- داده‌های اصلی دارای بعد زیاد هستند.
- داده‌های ذاتی و اساسی دارای بعد بسیار کمتری هستند.
- نگاشت مناسب را طوری پیدا کنید که:

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[۱] یا منطبق باشد. معیار فاصله در فضای تصویر شده، فاصله مینکوفسکی است $D(i,j)=(\sum_{k=1}^{m}|x_{ik}-x_{jk}|^{r})^{1/r}, r\geq 1$. 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*[۲] کراسکال اندازه‌گرفته می‌شود. قبل از تعریف این معیار به اختصار مشکل قرار دادن نقاط در یک فضا را بررسی می‌کنیم.

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:

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

متناظراً عدم تشابه اشیاء اصلی عبارتند از:

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

تنش می‌تواند به شکل نمودار شپارد[۳] 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.

۱- Match

۲- Stress

۳- Shepard

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 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.

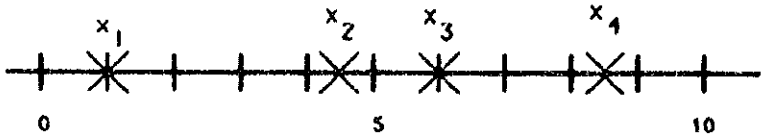
اگر ماتریس عدم تشابه اصلی که از طریق پرسشنامه گردآوری می‌شود، نامتقارن بود با میانگین‌گیری از عناصر متقارن نسبت به قطر اصلی، آن را تبدیل به ماتریس متقارن می‌کنیم.

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

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

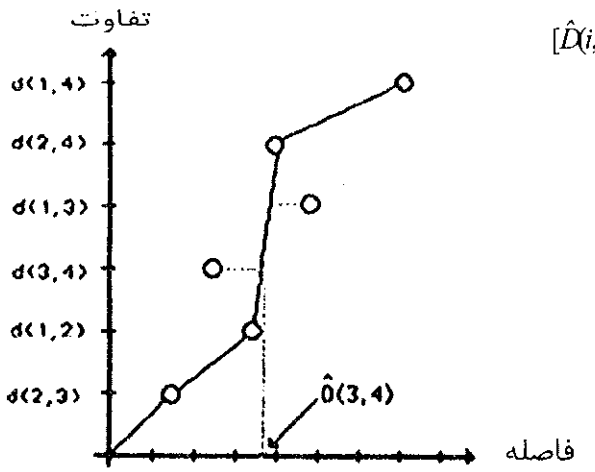
شکل (۲-۲۸) یک پیکره‌بندی از چهار نقطه در یک بعد و نمودار شپارد متناظر را نشان می‌دهد. دنباله خطوط مستقیم ترسیم شده در نمودار طوری رسم شده‌اند که تنش را حداقل کنند. نقاط روی نمودار که متناظر با عدم تشابه بین الگوهای $d(۱,٤), d(۲,۳), d(۱,۲), d(۲,٤)$ هستند روی دنباله خطوط مستقیم دارای شیب مثبت قرار دارند. 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.

$$[D(i,j)]=\begin{bmatrix} - & ۳/٥ & ٥/۰ & ۷/٥ \\ - & - & ۱/٥ & ۱/۰ \\ - & - & - & ۲/٥ \\ - & - & - & - \end{bmatrix}$$



(a)

$$[\hat{D}(i,j)]=\begin{bmatrix} - & 3/5 & 3.83 & 7/5 \\ - & - & 1/5 & 4 \\ - & - & - & 3.67 \\ - & - & - & - \end{bmatrix}$$



(b)

$$stress(curve)=\left[\frac{(1.17)^2+(1.17)^2}{(3.5)^2+(5.0)^2+(7.5)^2+(1.5)^2+(4.0)^2+(2.5)^2}\right]^{\frac{1}{2}}=0.152$$

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	٥۰	۷
٤	۲٤٦	۲٥۱	٥۱
٥	٥۹	۲۲٥	۲٤۳
٦	۱۲۳	٦۷	۲۳٥
۷	۲٤۸	٥٤	۸٦
۸	٥٤	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:

$$d_{۱,۲} = \sqrt{(۶۱-۱۳۹)^۲ + (۱۴۶-۱۶۳)^۲ + (۳۴-۱۷)^۲} = ۸۱.۶$$

جدول ۲-۸) فاصله رنگها

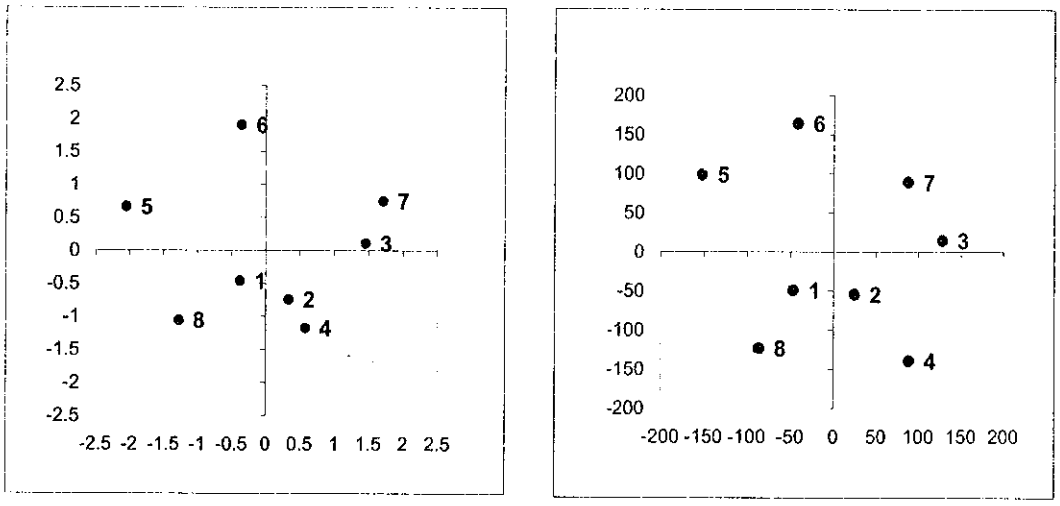
رنگ ۱	رنگ ۲	رنگ ۳	رنگ ٤	رنگ ٥	رنگ ٦	رنگ ۷	رنگ ۸
۰	۸۲	۱۵۰	۲۱۳	۲۲۳	۲۲۵	۲۱۵	۱۰۶
۸۲	۰	۱۱۸	۱٤۳	۲٤۸	۲۳۹	۱٦۹	۱۲۹
۱۵۰	۱۱۸	۰	۲۱۸	۳۱۵	۲۳٤	۱۰۹	۲۳۸
۲۱۳	۱٤۳	۲۱۸	۰	۲٦۹	۲۸۸	۲۰۰	۱۹۲
۲۲۳	۲٤۸	۳۱۵	۲٦۹	۰	۱۷۰	۱۹۰	۲۵۵
۲۲۵	۲۳۹	۲۳٤	۲۸۸	۱۷۰	۰	۱۹۵	۲۵۹
۲۱۵	۱٦۹	۱۰۹	۲۰۰	۱۹۰	۱۹۵	۰	۲۷۵
۱۰۶	۱۲۹	۲۳۸	۱۹۲	۲۵۵	۲۵۹	۲۷۵	۰

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) MDS (شکل سمت راست) با روش PCA (شکل سمت چپ) مقایسه

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"*

9) *Tan P.N , Steinbach M., Kumar V. (2005)"Chapter 2: Data", Introduction to Data Mining, Addison-Wesley.*

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:

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

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:

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

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

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

جدول ۲- ۱۰) محاسبه انحراف استاندارد

(۱)

X	$(X-\bar{X})$	$(X-\bar{X})^2$
۰	-۱۰	۱۰۰
۸	-۲	٤
۱۲	۲	٤
۲۰	۱۰	۱۰۰
Total		۲۰۸
Divided by (n-۱)		٦۹/۳۳۳
Square Root		۸/۳۲٦٦

(۲)

X	$(X-\bar{X})$	$(X-\bar{X})^2$
۸	-۲	٤
۹	-۱	۱
۱۱	۱	۱
۱۲	۲	٤
Total		۱۰
Divided by (n-۱)		۳/۳۳۳
Square Root		۱/۸۲۵۷

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:

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

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:

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

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

روشهای داده کاوی

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 *SAS*، *S-plus* (یا *R*) و *SPSS* 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 به‌وسیلهٔ مشاهدات است تا یادگیری با مثالها.

در این فصل مباحث زیرمطرح خواهند شد:

- تعریف تحلیل خوشه‌ای
- روش خوشه بندی افرازی[1]
- روش خوشه بندی سلسله مراتبی
- روش خوشه بندی مبتنی برچگالی[2]

---

[1]- Partitioning

[2]- Density Based methods

- روش خوشه بندی مبتنی بر مشبک کردن فضا[1]
- نقشه‌های خود سازمانده

۳-۱- تعاریف و مفاهیم تحلیل خوشه‌ای

خوشه‌بندی، گروه‌بندی نمونه‌های مشابه با هم در یک حجم داده می‌باشد. مسئله اساسی خوشه‌بندی عبارت است از: توزیع داده‌ها به K گروه مختلف که داده‌های 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 حجم، بعد: حجم و بعد داده‌ها را میتوان به وسیله خوشه بندی کاهش داد. این کاربرد بیشتر به‌عنوان پیش‌پردازش داده‌ها مورد استفاده قرار می‌گیرد.

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

- شیء: به هر مورد از اقلامی گفته میشود که قرار است خوشه بندی شود. (مثلا مشتریان یک فروشگاه)
- ویژگی: هر شیء دارای چند مشخصه است که آن شیء را از اشیاء دیگر متمایز میکند. (مثلا مقدار خرید، دفعات مراجعه و تعداد دفعات یک مشتری)
- داده: در برخی موارد، داده به یک شیء اشاره میکند و گاهی نیز به محتوی یکی از ویژگیهای یک شیء اشاره دارد. ما سعی میکنیم اولی را به صورت دادهٔ شیء و دومی را به صورت دادهٔ ویژگی بیان کنیم.
- خوشه: مجموعه‌ای از اشیاء داده‌ای است به شکلی که اشیاء درون یک خوشه به یکدیگر شبیه‌اند و با اشیاء خوشه‌های دیگر متفاوت هستند.

---

[1]- Grid Based Method

- تحلیل خوشه‌ای: گروه‌بندی مجموعه‌ای از اشیاء 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 اقلام یا مشاهدات با توجه به شباهت آنها است که از طریق آن اقلام یا مشاهدات به گروه‌های 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 به دو طبقهٔ مستقل و وابسته تقسیم نمی‌شوند. در اینجا تمرکز روی گروه‌هایی از اشیاء 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 روشهای خوشه‌بندی تصویر با معنا و جامعی از انبوه داده‌هایی که مرتبا جمع میشوند را 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 یعنی داده‌های پرت، استفاده نمود. مثلاً به‌جز یکی از مشتریان، دیگران خریدی بالای یک میلیون 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 مختلف مشتریانش را کشف کرده و این دانش را برای توسعه برنامه‌های بازاریابی مورد نظرش استفاده کند. این روش در بخش‌بندی مصرف‌کنندگان محصولات و تمرکز بر روی گروه هدف در بازاریابی بسیار کاربرد دارد.
- خاک برداری: شناسایی مناطقی که دارای خاک مشابه در زمین هستند.
- بیمه: شناسایی مشتریان بیمهٔ موتوریای 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 نقطه 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.

۳-۲- معیارهای شباهت و تمایز در انواع داده‌ها

در فصل دوم انواع متغیرها شامل اسمی، رتبه‌ای، فاصله‌ای و نسبتی شرح داده شدند. دراین فصل شباهت و تمایز بی مشاهدات(رکوردها) بر حسب نوع این متغیرها توضیح داده میشود.

فرض کنید مجموعه‌ای از داده‌های اشیاء که باید خوشه‌بندی شوند، شامل n شیء داده باشند. 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 (ماتریس شیء – ویژگی): این نوع ساختمان داده n شیء را با P ویژگی همانند سن، وزن، ارتفاع و غیره نمایش می‌دهد. این ماتریس در شکل (۳-۱) نشان داده شده است.

ویژگی →

شیء ↓

$$\begin{bmatrix} x_{11} & \dots & x_{1f} & \dots & x_{1p} \\ \dots & \dots & \dots & \dots & \dots \\ x_{i1} & \dots & x_{if} & \dots & x_{ip} \\ \dots & \dots & \dots & \dots & \dots \\ x_{n1} & \dots & x_{nf} & \dots & x_{np} \end{bmatrix}$$

شکل ۳-۱) ماتریس داده

این ماتریس داده‌ها کاملاً شبیه یک جدول در پایگاه داده است در شکل(۳-۱) ماتریسی که شامل n داده مختلف (رکورد پایگاه داده‌ها) که هر کدام P بعد دارند را نشان میدهد.

ماتریس تمایز[۱] (ماتریس شیء – شیء): این ماتریس فاصله یا عدم تشابه بین هر دو شیء را مشخص می‌کند و معمولاً $n \times n$ می‌باشد. $d(i,j)$ مقداری 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[۲]:

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

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

۱) $d(i,j) \geq ۰$

۲) $d(i,i) = ۰$

۳) $d(i,j) = d(j,i)$

که البته در بعضی موارد شرط سوم قابل تعدیل است.

٤) $d(i,j) \leq d(i,k) + d(k,j)$

توجه داشته باشید که فاصله مینکوفسکی، در اصل حالت کلی‌تری برای فاصله مانهاتان و فاصله‌اقلیدسی است. برای q های بزرگتر از دو این رابطه معنی خاصی ندارد اما می‌تواند در

۱- Dissimilarity - Distance

۲- Minkowski

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.

از آنجاکه پراکندگی و مقیاس داده-ویژگیهای مختلف با یکدیگر متفاوت هستند لذا ابتدا باید همه داده‌ها به یک مقیاس تبدیل شوند.[۱] روشهای مختلف نرمال سازی ویژگیها در فصل دوم مطرح شدند.

۳-۲-۱- انواع متغیرها و معیارهای شباهت و تمایز

متغیرهای عددی (فاصله‌ای و نسبتی)

همانطور که در فصل دوم مطرح شد، متغیرهایی که تفاوت بین مقادیرشان بامعنی است، متغیرهای عددی نام دارند مانند وزن و ارتفاع. برای نرمال کردن این متغیرها برای هر ویژگی مانند f نسبت به اشیاء i $(i=1,2,...,n)$ به‌صورت زیر عمل می‌کنیم:

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

$$m_f=\frac{1}{n}(x_{1f}+x_{2f}+...+x_{nf}) \qquad (۴-۳)$$

$$s_f=\frac{1}{n}(|x_{1f}-m_f|+|x_{2f}-m_f|+...+|x_{nf}-m_f|) \qquad (۵-۳)$$

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

حال میتوان با یکی از فاصله‌های «اقلیدسی»، «مانهاتان» یا «مینکوفسکی» تمایز بین اشیاء را اندازه‌گیری کنیم. 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، ۱۶۰ 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.

۲- این عمل را می‌توان معادل نرمال‌سازی داده‌ها دانست.

متغیرهای دودویی متقارن و نامتقارن

همانطور که بیان شد، 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.[1] برای اندازه‌گیری تمایز بین اشیاء با این ویژگیها در صورتی که همهٔ آنها از درجهٔ اهمیت یکسانی برخوردار باشند، ماتریس تمایز(ماتریس عدم تشابه) شکل (۳-۳) را تشکیل می‌دهیم.

شیء

شـ		۱	۰	*Sum*
	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 متغیرها و یا ویژگیهایی هستند که مقادیرشان در هر دو شی i , j برابر ۱ می‌باشد و به همین ترتیب b , c , d طبق شکل (۳-۳) تعریف می‌شوند. برای محاسبه عدم تشابه بین شیء i , j در صورتیکه همهٔ متغیرهای دودویی متقارن باشند از رابطه (۳-۶) که به جاکارد[2] معروف است، استفاده میکنیم.

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

۱- اگر جواب آزمایش دو نفر برای تشخیص یک بیماری نادر، مثبت باشد، آن دو نفر بسیار شبیه به یکدیگر هستند. اما منفی بودن جواب دلیلی برای شباهت آن دو نمی‌باشد.

[2] Jacard Coefficient

به منظور محاسبه عدم تشابه بین دو شیء i , j برای متغیرهای غیر متقارن از رابطه زیر استفاده میشود.

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

(توجه کنید که d حذف شده است زیرا بنابر قرارداد، متغیرهای منفی یا با مقدار صفر برای هر دو شیi , j اهمیت کمی دارند)

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

جدول ۳- ۱) نتایج آزمایشها

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

جدول ۳-۲) ماتریس تمایز 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:

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

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

با توجه به نتایج محاسبات میتوان گفت که بیماری مریم و علی زیاد شبیه نیست درحالیکه مریم و محمد احتمالا بیماری مشابهی دارند. همانطور که گفته شد متغیرهای این مثال غیر متقارن هستند.

متغیرهای اسمی

همانطور که در فصل دوم بیان شد، این متغیرها صرفا نامهای متفاوت دارند و فقط اطلاعاتی برای تمایز اشیاء فراهم میکند. 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 (۳-۸) استفاده می‌کنیم.

$$d(i,j) = \frac{p-m}{p} \qquad (۳-۸)$$

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

P: تعداد کل متغیرها

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)
$d(i,j)=(۳-۲)/۳=۰/۳۴$		$p=۳, m=۲$	

همانطورکه مشاهده شد تمایز دو شیء اول و دوم برابر ۰/۳۴ است.

متغیرهای رتبه‌ای

متغیرهای رتبه‌ای 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 رتبه‌ای f به صورت 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:

$r_{if} \in \{1,...,M_f\}$

دراینجا x_{if} مقدار یا محتوی متغیر مدال بوده و r_{if} رتبه‌ای است که به آن نسبت میدهیم.

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

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

M_f حداکثر حالات ممکن متغیر رتبه‌ای f است.

قدم ۳) حال هر یک از روشهای سنجش فاصله را می‌توان برای z_{if} به‌کار برد.

ترکیب متغیرهایی از انواع مختلف

در بخش قبل به محاسبه عدم شباهت (فاصله) مقادیر متغیرها پرداختیم. جدول (۳-٤) روشهای محاسبه شباهت و عدم شباهت را به طور خلاصه نشان می‌دهد. در این جدول x_{if} و

x_{jf} اشیاء (رکوردهای) i, j ام مشخصه f ام هستند. همچنین s_{ij}^f شباهت بین دو شیء و d_{ij}^f عدم شباهت بین دو شیء را نشان می‌دهد. پارامتر M_f نیز حداکثر تعداد حالتهای متغیر رتبه‌ای می‌باشد.

جدول ۳-٤) محاسبه شباهت و عدم شباهت با توجه به نوع متغیر

نوع ویژگی	شباهت	عدم شباهت
متغیر اسمی (و دودویی)	$s_{ij}^f = \begin{cases} 1 \text{ if } x_{if} = x_{jf} \\ 0 \text{ if } x_{if} \neq x_{jf} \end{cases}$	$d_{ij}^f = \begin{cases} 0 \text{ if } x_{if} = x_{jf} \\ 1 \text{ if } x_{if} \neq x_{jf} \end{cases}$
متغیر رتبه‌ای	$s_{ij}^f = 1 - d_{ij}^f$	$d_{ij}^f = \frac{\lvert r_{if} - r_{jf} \rvert}{(M_f - 1)} = \lvert z_{if} - z_{jf} \rvert$
متغیر فاصله‌ای یا نسبتی	$s_{ij}^f = \frac{1}{1 + d_{ij}^f}$ $s_{ij}^f = -d_{ij}^f$ $s_{ij}^f = e^{-d_{ij}^f}$ $s_{ij}^f = 1 - \frac{d_{ij}^f - \min d_{ij}^f}{\max d_{ij}^f - \min d_{ij}^f}$	$d_{ij}^f = \frac{\lvert x_{if} - x_{jf} \rvert}{\max_h x_{hf} - \min_h x_{hf}}$ h اندیس همه اشیاء (مشاهدات) مشخصه fام است

اگراشیاء دارای ویژگیهایی با انواع گوناگون باشند، برای سنجش عدم شباهت (یا شباهت) آنها از روش زیر استفاده می‌شود. فرض کنید مجموعهٔ داده‌ها شامل p متغیر و از انواع مختلف باشد. در اینجا تمایز مجموع متغیرها، متوسط تمایزهای فردی آنهاست. عدم تشابه بین اشیاء i, j با رابطه (۳-۱۰) تعریف می‌شود.

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

که در آن:

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

اگر اهمیت متغیرها با هم متفاوت باشد، به هر متغیر f وزن w^f داده می‌شود.

$$d(i,j)=\frac{\sum_{f=1}^{p} w^f \delta_{ij}{}^f d_{ij}{}^f}{\sum_{f=1}^{p} \delta_{ij}{}^f} \qquad (۱۲-۳)$$

توجه کنید که همه متغیرها در ابتدا نرمال شده‌اند.

مثال: در یک تیم فوتبال ویژگیهای رنگ لباس، نوع مدال کسب شده، جواب آزمایش دوپینگ و تعداد گلها در ۱۰ شوت برای تعیین مشابهت بازیکنان در نظر گرفته شده است. جدول (۳-۵) ویژگیهای مذکور را برای سه فوتبالیست مختلف نشان می‌دهد. هدف ما در اینجا به‌دست آوردن فاصله سه فوتبالیست از یکدیگر است.

جدول ۳-۵) ویژگیهای متفاوت سه فوتبالیست

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

برای متغیر رتبه‌ای مدال، $z_{i۲}^f$ فوتبالیستها به ترتیب برابر با ۰، ۰.۵، ۰.۵ است. از آنجاکه سه نوع مدال داریم ارزش طلا برابر ۰، نقره $\frac{۱}{۲}$ و برنز برابر $\frac{۲}{۲}$ فرض می‌شوند و مجموعه حالات ممکن برای متغیر نسبتی تعداد گلها در ۱۰ شوت یعنی z_{it}^f به صورت $\{۰,۱,۲,۳,۴,۵,۶,۷,۸,۹,۱۰\}$ است. در ایجا بازیکنی با ۱۰ گل زده دارای رتبه اول، با نه گل زده دارای رتبه دوم و در نهایت با صفر گل زده دارای رتبه یازدهم خواهد بود. در عمل Z_{if} به صورت زیر محاسبه می‌شود.

$$Z_{if}=\frac{\text{۱- رتبه تعداد گل زده}}{\text{۱- کل تعداد رتبه‌ها}}$$

(به عنوان مثال برای بازیکنی که ۲ گل زده و دارای رتبه نهم است مقدار Z_{if} معادل ۰/۸=(۱−۱۱)/(۹−۱) محاسبه میشود.) 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:

$$d(1,2) = \frac{0+1\times\frac{|0-0.5|}{0.5-0}+0+1\times\frac{|0.8-0.9|}{0.9-0.5}}{0+1+0+1} = 0.625$$

در هر مورد پس از به‌دست آوردن قدر مطلق فاصله برای نگاشت آن به بازهٔ $[0,1]$ آن را بر حداکثر اختلاف موجود بین عناصر مختلف در آن بعد (متغیر) تقسیم می‌کنیم.

۳-۳- روشهای اصلی خوشه‌بندی

رویکردهای اصلی خوشه‌بندی عبارتند از:

- روشهای افرازی
- روشهای سلسله مراتبی
- روشهای مبتنی بر چگالی
- روشهای مبتنی بر مشبک کردن فضا
- نقشه‌های خود سازمانده

روشهای افرازی

فرض کنید یک پایگاه داده با n شیء داریم. یک روش افرازی، k افراز از این داده‌های اشیاء درست می‌کند به‌طوریکه هر افراز یک خوشه را نشان می‌دهد و $k < n$. پس داده‌های اشیاء 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 جابجایی تکراری[۱] 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 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 خوشهیا مرکز خوشه ، نمایش داده می‌شود.
- 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

این روش ساختاری 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[۲] 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

۱- Itterative Relocation Technique

۲- Bottom - Up

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 خوشهٔ کوچکتر تقسیم می‌شود.

روش مبتنی بر چگالی

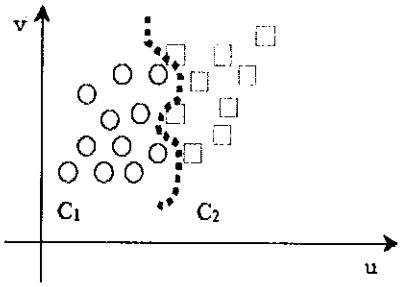
بسیاری از روشهای افرازی، اشیاء را بر اساس فاصلهٔ آنها نسبت به یکدیگر خوشه‌بندی می‌کنند. برخی روشها تنها خوشه‌های کروی شکل را پیدا می‌کنند و در برابر خوشه‌هایی به شکلهای دلخواه با مشکل مواجه می‌شوند. در مقابل برخی روشهای دیگر خوشه‌بندی برپایهٔ چگالی توسعه یافته‌اند. ایدهٔ عمومی این روشها رشد دادن خوشه‌ها 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 می‌رود. 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، تقسیم‌بندی می‌کنند. مهم‌ترین مزیت آن افزایش سرعت پردازش می‌باشد که براساس تعداد سلولها و تعداد نقاط داده متفاوت است. مانند الگوریتمهای *STING*، *CLIQUE*، *WAVE*.

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.

۳-۳-۱- روش افرازی



شکل ۳-٤) خوشه بندی افرازی

فرض کنیم یک پایگاه داده با n شی داریم. علاوه بر آن تعداد خوشه‌هایی که باید تشکیل شوند، نیز معلوم است. 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 نیز نام دارد[۲]، 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*

الگوریتم *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 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:

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

- قدم ۱) به‌صورت تصادفی k نقطهٔ دلخواه را به‌عنوان مراکز خوشه‌های ابتدایی انتخاب کن.(بهتر است k نقطه از n نقطهٔ موجود انتخاب شود.)
- قدم ۲) هر شی را با توجه به بیشترین شباهت آن به مراکز خوشه‌ها، به خوشه‌ها تخصیص بده.
- قدم ۳) مراکز خوشه‌ها را به‌روز کن به این معنی که برای هر خوشه میانگین اشیاء آن خوشه را محاسبه کن.
- قدم ٤) با توجه به مراکز جدید خوشه‌ها به قدم دوم برگرد تا هنگامی که هیچ تغییری در خوشه‌ها رخ ندهد. (در این حالت الگوریتم پایان یافته است.)
- در عمل این الگوریتم یک روش هیوریستیکی برای کاهش معیار مربع خطا است که در رابطهٔ (۳-۱۳) آمده است.

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

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 پایگاه دادههای بزرگ، کارآ نیست و باید توسعه داده شود. 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[۱] ختم می‌شود نه یک بهینهٔ سراسری[۲].

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 k های مختلف و بررسی معیار مربع خطا برای هر k می‌باشد. همچنین این روش برای

۱- Local Optimum

۲- Global Optimum

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 ۱: 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 کرده و بقیهٔ اعضاء مجموعه را با توجه به فاصلهٔ آنها از این دو مرکز تخصیص می‌دهیم. یعنی هر عضو را به خوشهای تخصیص میدهیمکه به مرکز آن نزدیکتر باشند. 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:

$m_۱=۳$, $m_۲=۱۸$

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

$m_۱=٤.۷۵$, $m_۲=۱۹.٦$

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

$m_۱=۷$, $m_۲=۲۵$

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

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

مثال۲: ۷ نوع غذا (۷ شیء) با توجه به دو صفت میزان پروتین (P) و میزان چربی (F) در جدول (۳-٦) آورده شده اند:

جدول ۳ - ٦) ویژگیهای متفاوت غذاها

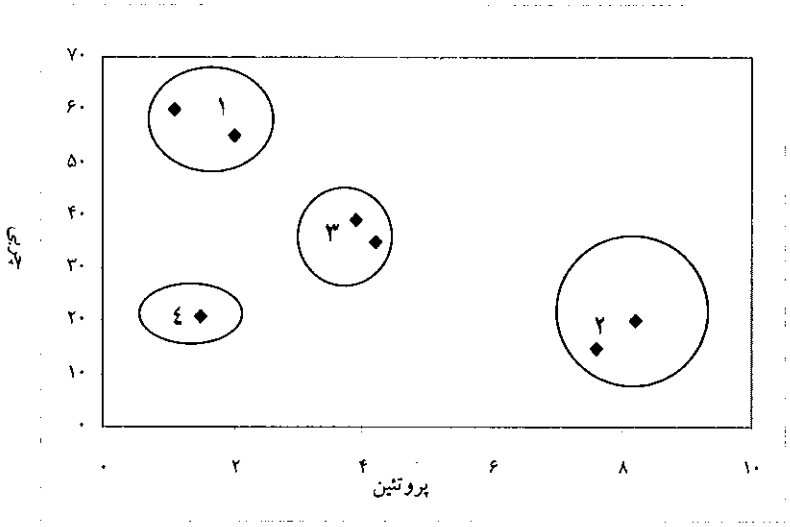
شماره غذا	میزان پروتئین	میزان چربی
۱	۱/۱	٦۰
۲	۸/۲	۲۰
۳	٤/۲	۳۵
٤	۱/۵	۲۱
۵	۷/٦	۱۵
٦	۲/۰	۵۵
۷	۳/۹	۳۹

اگر روش *k-means* را با $k=٤$ شروع کنیم به‌طوری‌که $m_٤=۴$ و $m_۳=۳$ و $m_۲=۲$ و $m_۱=۱$ باشند ، آنگاه:

$K_۱=\{۱,٦\}$, $K_۲=\{۲,۵\}$

$K_۳=\{۳,۷\}$, $K_٤=\{٤\}$

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 الگوریتم *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 الگوریتم *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 اسمی یا 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.

الگوریتم *k-medoids*

در این الگوریتم به جای استفاده از مرکز یک خوشه به‌عنوان مرجع، می‌توان از *medoid*ها (اشیایی که در مرکزی‌ترین محل یک خوشه می‌باشند) استفاده کرد. این روش بر اساس اصل حداقل‌سازی مجموع عدم شباهتها میان هر شیء و شیء مرجع عمل میکند. 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 تابع هزینه بیان می‌شود.

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:

- قدم ۱) k شیء تصادفی به‌عنوان *medoid*های اولیه اختیار کن.
- قدم ۲) هر کدام از اشیاء باقیمانده را به خوشه‌ای با نزدیک‌ترین *medoid* تخصیص بده.
- قدم ۳) به‌طور تصادفی یک شی غیر *medoid* را انتخاب کن، O_{random}
- قدم ٤) 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. در غیر اینصورت مراکز را عوض نکرده و به قدم ۳ برو.
- قدم ٥) این الگوریتم را تا زمانیکه همه نقاط به عنوان *medoid* انتخاب شده و تغییری در خوشه‌ها ایجاد نشود، ادامه بده.

برای اندازه‌گیری اینکه شیء O' بهتر از O به‌عنوان یک *medoid* هست یا خیر، کافیست حاصل رابطه (۳-۱٤) را به‌دست آوریم. اگر $E(o')-E(o)<0$ آنگاه جابه‌جایی O' با O مفید است.

$$E=\sum_{i=1}^{k}\sum_{p\in c_i} d(p,o_i)^2 \qquad (۱٤-۳)$$

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 = ۶−۱(cos۰) , ۲(cos۰) , ۷(۱−۰=۱)

خوشه۲ = ۸−۱۰ *(cos۰)*

خوشه جدید = ۱۵−۱۷*(cos۱۲−۹=−۷)* , ۲۰*(cos۱۵−۱۲=−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 ۶ جایگزین کنیم، هزینه‌ها به‌صورت زیر می‌باشد:

خوشه ۱ = ۸−۶*(cos۲−۰=۲)* , ۷*(cos۱−۱=۰)* , ۱۰*(cos۰)*

خوشه ۲ = ۱۵−۱۷*(cos۰)* , ۲۰*(cos۰)*

خوشه جدید = ۱−۲*(cos۱−۴=−۳)* $T_c = -۱$

پس ۱ به جای ۶ 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(cos۰)

$T_c = ۲$

خوشه ۲ = ۱۵−۱۷*(cos۰)* , ۲۰*(cos)*

خوشه جدید = ۱۰−۶*(cos۰)* , ۷*(cos۰)* , ۸*(cos۲−۰=۲)*

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

خوشه ۱ = ۱−۲*(cos۰)*

خوشه ۲ = ۸−۶*(cos۰)* , ۷*(cos۰)* , ۱۰*(cos۰)*

خوشه جدید = ۱۷−۱۵*(cos۲−۰=۲)* , ۲۰*(cos۳−۵=−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.

الگوریتم *CLARA*[2]

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

۱- Partitioning Around Medoids

۲- Clustering Large Application

۳- Rousseeuw

۴- Kaufmann

$O(ks^2+k(n-k))$ می‌باشد که k تعداد خوشه‌ها، s تعداد اشیاء نمونه و n کل اشیاء می‌باشد. لذا پیچیدگی الگوریتم کاهش می‌یابد و از مرتبه تعداد اشیاء نمونه است.

۳-۳-۲- روش خوشه‌بندی سلسله مراتبی

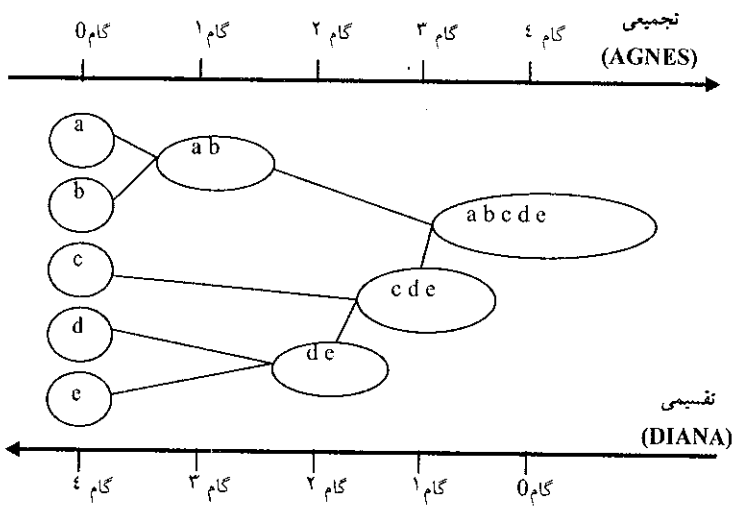
این روش با گروه‌بندی اشیاء به‌صورت یک درخت کار می‌کند و معمولاً به دو صورت پایین به بالا (تجمیعی) یا با بالا به پایین (تقسیمی) انجام می‌شود. این دو روش را می‌توان به‌صورتهای زیر بیان کرد:

- تجمیعی[۱]: در این روش خوشه‌ها مکرراً با هم ترکیب می‌شوند. به این صورت که ابتدا هر یک از اشیاء را به عنوان یک خوشه در نظر می‌گیرد و سپس با ترکیب کردن این خوشه‌ها، آنها را به خوشه‌های بزرگ و بزرگتر تبدیل می‌کند تا اینکه همهٔ اشیاء در یک خوشه قرارگیرند و یا به شرط پایان برسد.
- تجزیه‌ای یا تقسیمی[۲]: در این روش خوشه‌ها مکرراً تقسیم می‌شوند. این روش دقیقاً بر عکس روش تجمیعی عمل می‌کند به این صورت که ابتدا یک خوشه شامل همهٔ اشیاء ایجاد می‌شود و سپس الگوریتم این خوشه‌ها 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 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 (۳-۶) دقت کنید.

۱- Agglomerative

۲- Divisive



شکل ۳- ٦) نمودار دو روش تجمیعی و تقسیمی

درالگوریتم *AGNES*[1]، ابتدا هر شی در داخل یک خوشه قرار می‌گیرد. مثلاً در شکل (۳-٦) پنج خوشه برای مجموعهٔ $\{a,b,c,d,e\}$ به‌وجود می‌آید. 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 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 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

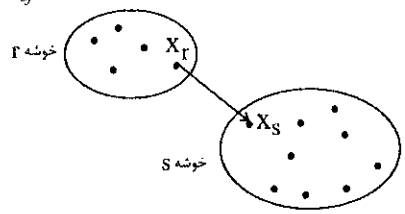
---

[1]- Agglomerative Nesting

[2]- Single Link

حداقل آنها، فاصله بین دو خوشه را معین کرد. مثلاً فاصله بین دو خوشه r, s به‌صورت زیر حساب می‌شود:

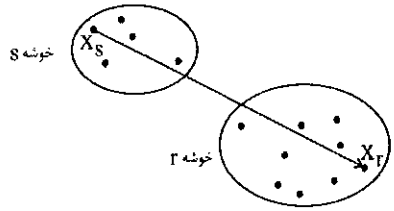
$$d(r,s) = \min(dist(x_{ri}, x_{sj})) \qquad (۱۹-۳)$$



شکل ۳- ۷) روش حداقل فاصله

- پیوند کامل[۱]: در این حالت فاصلهٔ بین خوشه‌ها بر حسب دورترین فاصلهٔ ممکنه بین عناصر آنها محاسبه می‌شود:

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



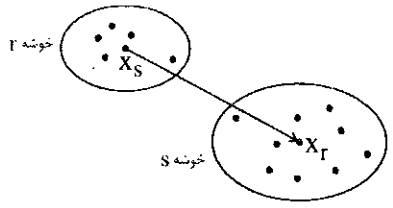
شکل ۳- ۸) روش حداکثر فاصله

- پیوند متوسط[۲]: فاصله دو خوشه مساوی مقادیر متوسط کلیهٔ فاصله‌های ممکنه بین عناصر دو خوشه است:

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

۱- Complete Link

۲- Average Link



شکل ۳- ۹) روش مرکز ثقل

- پیوند مرکزی[1]: فاصلهٔ بین دو خوشه بر اساس فاصلهٔ بین مراکز آن دو خوشه محاسبه می‌شود. برای محاسبه مراکز خوشه‌ها می‌توان از روشهای مختلفی ازجمله روش میانگین استفاده نمود

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 پیوند مرکزی استفاده می‌شود. شرط پایان الگوریتم رسیدن به ۵ خوشه می‌باشد. مجموعهٔ داده‌هایی که باید خوشه‌بندی شوند، عبارتست از:

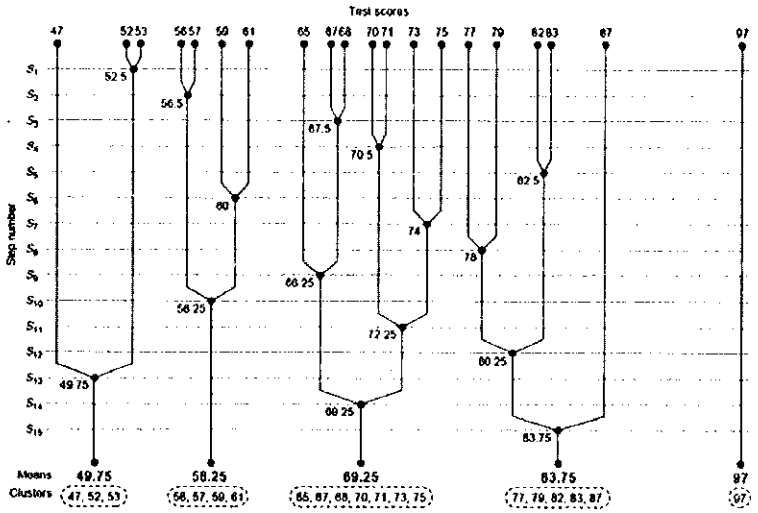
{۴۷, ۵۲,۵۳,۵۶,۵۷,۵۹, ۶۱, ۶۵, ۶۷, ۶۸, ۷۰, ۷۱, ۷۳, ۷۵, ۷۷, ۷۹, ۸۲, ۸۳, ۸۷, ۹۷}

این خوشه‌بندی در شکل (۳-۱۰) آمده است. به نمودار درختی تشکیل شده در شکل (۳-۱۰) نمودار دندانه‌ای[2] 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.

---

[1]- Centroid

[2]- Dendogram



شکل ۳-۱۰) روش AGNES

الگوریتم *DIANA*[۲]

این الگوریتم عکس الگوریتم *AGNES* عمل می‌کند. به این صورت که ابتدا همه اشیاء را درون یک خوشه قرار می‌دهد و این خوشه‌ها را تقسیم می‌کند تا اینکه نهایتاً هر شیء در یک خوشه قرار گیرد. برای بیان چگونگی تجزیهٔ خوشه‌ها فرض کنیم الگوریتم به k خوشه رسیده است. ابتدا در هر خوشه بزرگترین فاصلهٔ ممکنه بین اشیاء آن خوشه را پیدا کرده و 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 پیدا کرده 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 از خوشه انجام می‌دهیم. (n تعداد اعضای خوشه) فاصله‌های به‌دست آمده را در مجموعه‌های

۲- Divisive Analysis

$M_1, M_2, \ldots, M_{n-1}$ قرار می‌دهیم. بدیهی است تعداد اعضای $M_1 = \binom{n}{1}$ و 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. اگر این عضو از مجموعه M_k باشد، در این صورت خوشهٔ n تایی مورد نظر به یک خوشهٔ k تایی (اعضای مربوط به بزرگترین مقدار) و یک خوشهٔ $n\text{-}k$ تایی تجزیه می‌شود. پیچیدگی الگوریتم *DIANA* خیلی زیاد است، یعنی $O(2^{n-1}-1)$ و معمولاً مقرون به صرفه نیست، لذا بیشتر اوقات از *AGNES* استفاده می‌شود.

۳-۳-۳- 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 تعداد خوشه‌ها 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.

۳-۳-۴- تعیین تعداد خوشه‌ها

اگر تمام متغیرها کاملاً مستقل باشند، هیچ خوشه‌ای ایجاد نمی‌شود. (تمام فضا به صورت 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 هر تکرار، 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 کاربردهای خاصی مشخص شوند. 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]

۳-۳-۵- 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.

الگوریتمی که در اینجا بیان می‌شود *DBSCAN*[۱] (یا خوشه‌بندی فضایی بر پایه چگالی برای داده‌های مغشوش) نام دارد که از متداول ترین روشهای مبتنی بر چگالی است.

در این الگوریتم 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[۲] حداقل را برای شروع کار الگوریتم تعریف کند. چگالی توزیع داده‌ها در اطراف این نقاط زیاد

۱- Density- Based Spatial Clustering of Applications with Noise

۲- Minpts

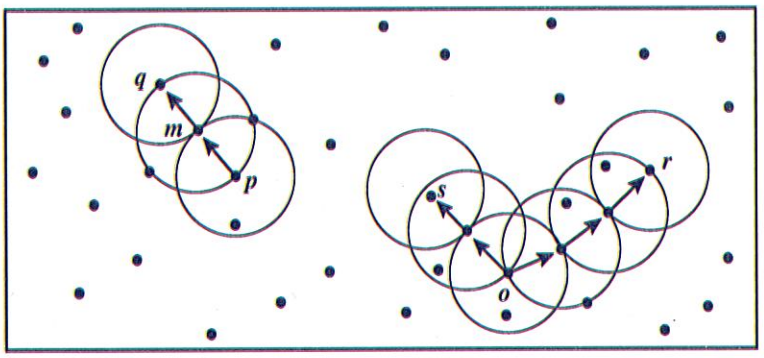
است. حال اجازه دهید اصطلاحات زیر را برای ادامه کار تعریف کنیم. نقطه p را از نقطه q مستقیماً قابل دسترس چگال[۱] می‌نامیم اگر p در شعاع ε از q قرارگرفته باشد و در شعاع ε از q حداقل نقاط مورد نظر ما نیز وجود داشته باشد.

نقطه p را از نقطه q قابل دسترس چگال[۲] 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[۳] است به شرطی که با حفظ شرایط ε و حداقل نقاط، یک شیء مانند o وجود داشته باشد که هر دوی q, p از 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

در شکل (۳-۱۱) با فرض حداقل نقاط برابر با ۳ و شعاع ٤، خوشه‌هایی مشخص شده است. نقطه M از نقطه P مستقیماً قابل دسترس چگال است و Q از نقطه p قابل دسترس چگال

۱- Directly Density Reachable
۲- Density Reachable
۳- Density Connected

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*[۱] (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[۲] و فاصله دسترسی[۳] نیاز دارد.

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

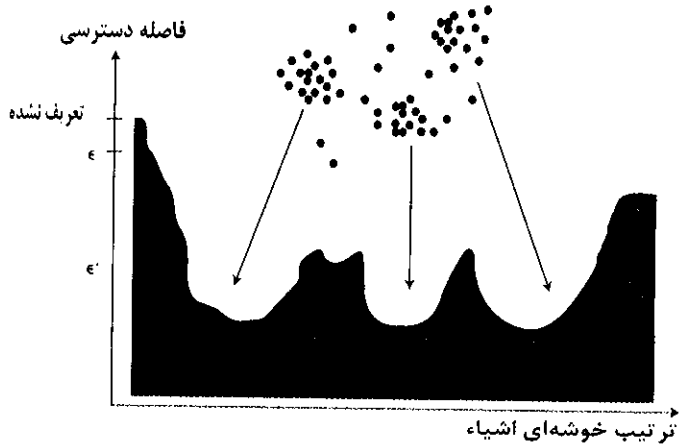
۱- Ordering Points To Identify the Clustering Structure

۲- Core- Distance

۳- Reachability- Distance

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.

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.



شکل ۳- ۱۲) ترتیب خوشه‌بندی در OPTICS

روش *DENCLUE* بر اساس سه ایده اصلی زیر استوار است:

---

[1]- Density Based Clustering

[2]- Influence Function

- تأثیر هر داده‌ای بر فضا را می‌توان به‌طور رسمی با یک تابع ریاضی به نام تابع تأثیر[۳] 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[۱] 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[۲] 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.

$$f_B^y = f_B(x,y). \qquad (۲۳-۳)$$

توابع معروفی که در اینجا استفاده می‌شوند عبارتند از:

- تابع اثر موج مربع که به‌صورت زیر تعریف می‌شود:

$$f_{Square}(x,y) = \begin{cases} ۰ & if\ d(x,y) > \sigma \\ ۱ & otherwise \end{cases} \qquad (۲۴-۳)$$

- تابع تأثیر فاصله گوسی که به‌صورت زیر تعریف می‌شود:

$$f_{Gaues}(x,y) = e^{\frac{-d(x,y)^۲}{۲\sigma^۲}} \qquad (۲۵-۳)$$

- تابع اقلیدسی:

۱- Density Attractors

۲- Basic Function

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[۱] 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.

$$F_B^D=\sum_{i=1}^{N} f_B^{x_i}(x) \qquad (۲۶-۳)$$

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:

$$f_{Gauesion}^D(x)=\sum_{i=1}^{N} e^{\frac{-d(x,y)^2}{2\sigma^2}} \qquad (۲۷-۳)$$

حال از روی این تابع می‌توان جاذب چگالی را محاسبه نمود که حداکثر محلی تابع مورد بحث است. در چنین حالتی با الگوریتمهای تپه‌نوردی[۲] می‌توان این نقاط و مجموعه نقاط اطراف آنها را مشخص کرد. حال خوشه‌ها را این‌گونه تعریف می‌کنیم:

یک خوشه مرکزمحور[۳]: 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:

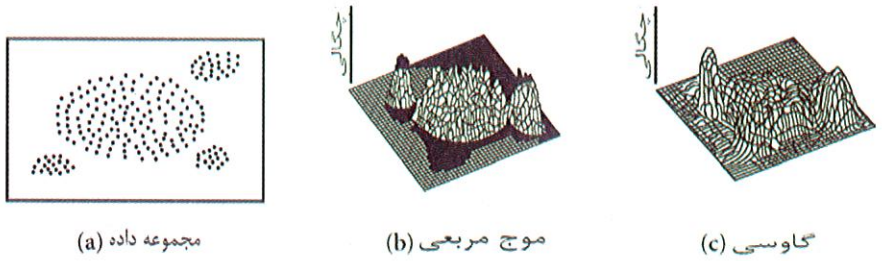
- از پشتوانه‌ای ریاضی برخوردار بوده و روشهای دیگر مانند افرازی و روشهای سلسله مراتبی را نیز در بر می‌گیرد.
- برای مجموعه داده‌هایی با اغتشاش بالا بسیار مناسب است.

۱- Density Function

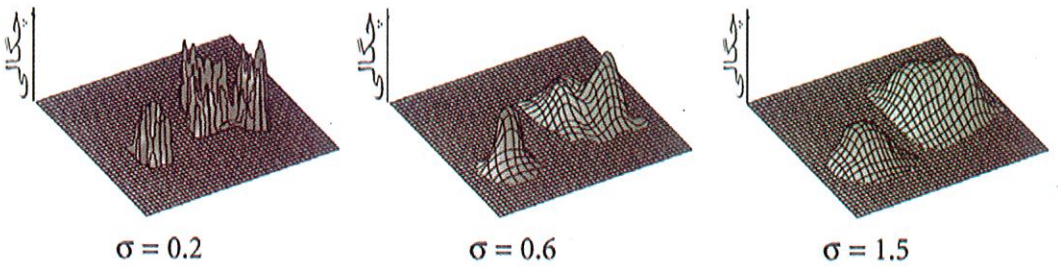
۲- Hill Climbing

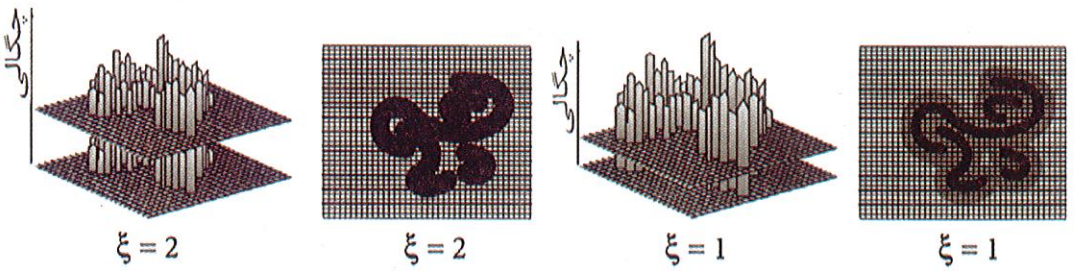
۳- Center Defined Cluster

- 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.



شکل ۳- ۱۳) تابع چگالی مربوط به فضای آنها





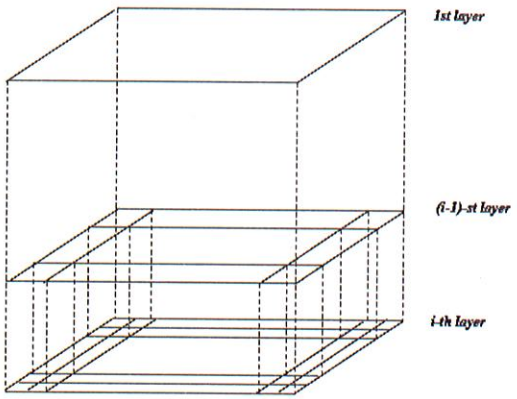
شکل ۳-۱٤) تابع چگالی و میزان حساسیت به پارامترهای اولیه

شکل (۳-۱٤) 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.

۳-۳-۶- روشهای مبتنی بر مشبک کردن فضا[۱]

روش مشبک‌سازی فضا به سلولهای مختلف، امکان کار بر روی اطلاعات با درجه تفکیک شفافیت‌های متفاوت[۲] 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*[۳] 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.



شکل ۳ - ۱۵) ساختار سلسله مراتبی

۱- Grid- Based Methods

۲- Multi - Resolution

۳- A Statistical Information Grid Approach

سپس برای هر سلول اطلاعات آماری مانند میانگین، میانه، بیشینه، کمینه، انحراف معیار استاندارد و... محاسبه می‌شود. شکل (۳-۱۵) یک ساختار سلسله مراتبی را نشان می‌دهد. 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[۱] در پایگاههای داده بسیار مناسب است.

مزایای این روش عبارتند از:

- این روش پردازش موازی را تسهیل می‌کند.
- 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.

۱- Query

۳-۳-۷- 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. تفاوت SOM با شبکه رقابتی[۷] عبارت است از:

- در *SOM* هیچ سوگیری[۸] وجود ندارد. سوگیری، مقدار وزن نرون ورودی ثابت است.
- علاوه بر نرون برنده، نرونهای همسایه نیز تطبیق یافته و اوزان آنها اصلاح می‌شود.

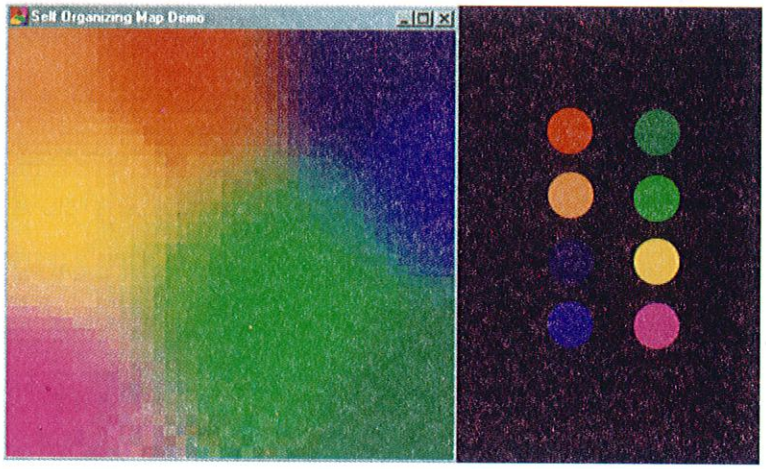
مثالی متداول برای کمک به آموزش مبانی *SOM*، نگاشت رنگها در صفحه دو بعدی است. فرض کنید هزاران مشاهده داریم و هر مشاهد یکی از ۸ رنگ سمت راست شکل (۳-۱۶) باشد. هر رنگ از سه جزء قرمز، سبز و آبی تشکیل شده است که می‌توانند دارای مقداری بین ۰ تا ۲۵۵ باشند، بنابراین هر مشاهده دارای سه ویژگی می‌باشد.

اگر بخواهیم رنگها را در فضای واقعی خود ترسیم کنیم نیاز به سه بعد داریم. 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،

۱- Self-Organizing Maps: SOM
۲- SOFM: Self-Organizing Feature Maps
۳- Kohonen Maps
۴- Teuvo Kohonen
۵- Vector Quantization
۶- Topologic
۷- Competitive Network
۸- Bias

معمولاً نواحی مشابه در کنار یکدیگر 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[۱] 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[۲]، 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.



شکل ۳- ۱٦) نمایش خروجی شبکه (چپ) و رنگهای خوشه‌بندی شده توسط آن (راست)

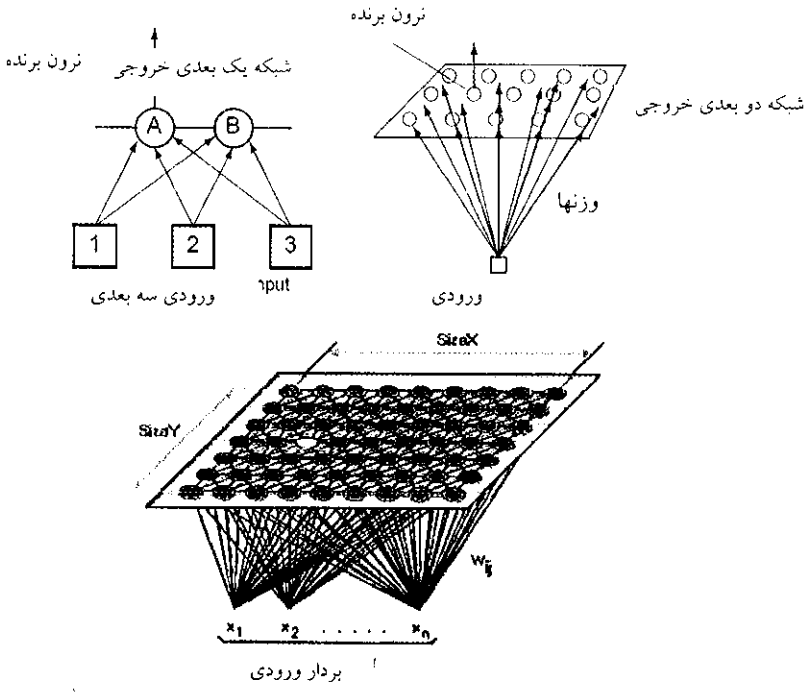
۱- Back propagation

۲- Feed Forward

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[۳] 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.



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)

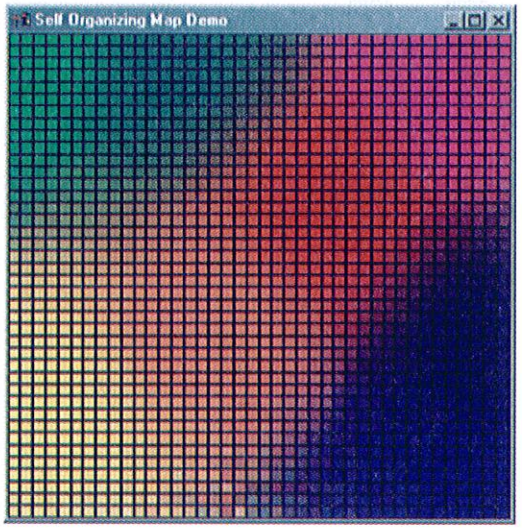
۳- Lattice

۴- Lateral

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[۵] 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 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[۱]: برای تعیین نرون برنده
- همکاری[۲]: کمک به همسایگان در جدول شبکه
- تطبیق[۳]: 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*[۴]) شناخته می‌شود.

۱- Competition

۲- Cooperation

۳- Adaptation

۴- BMU: Best Matching Unit

- 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

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: $0 < w < 1$. 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:

$$Dist = \sqrt{\sum_{i=0}^{n} (X_i - W_i)^2} \qquad (۲۸-۳)$$

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.

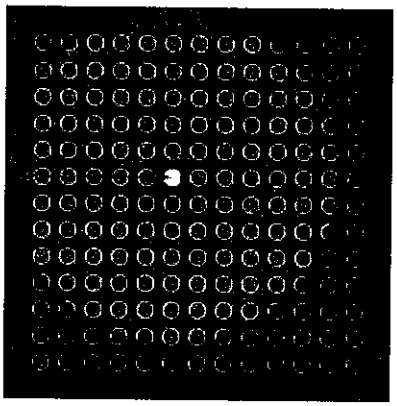
به‌عنوان مثال برای محاسبه فاصله بین بردار رنگ قرمز (۰، ۰، ۱) با بردار دلخواه اوزان (۰/۵، ۰/٤، ۰/۱) داریم:

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

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.



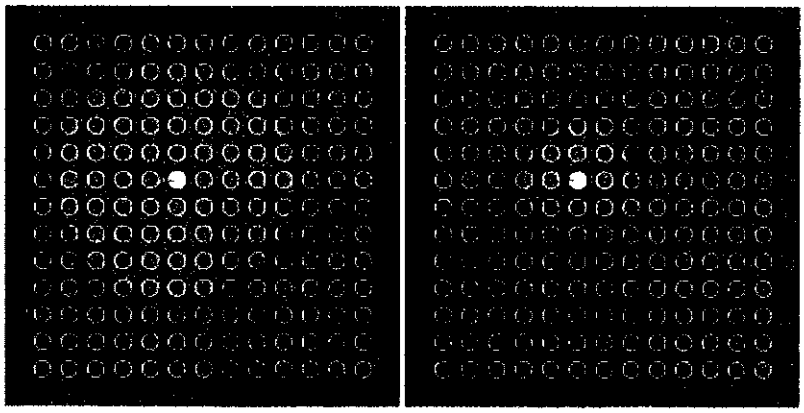
شکل ۳- ۱۹) همسایگی BMU

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:

$$\sigma(t)=\sigma_{\cdot}\, e^{\frac{-t}{\lambda}} \qquad t=1,2,3,... \qquad (۲۹-۳)$$

که در آن $\sigma_{\cdot}$ نشان دهنده عرض جدول در زمان $t_{\cdot}$ 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 (۳-۲۰) طی زمان کاهش می‌یابد. 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.



شکل ۳- ۲۰) شعاع همواره در حال کاهش

اصلاح وزنها

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

$$W(t+1)=W(t)+L(t)(X(t)-W(t)) \qquad (۳۰-۳)$$

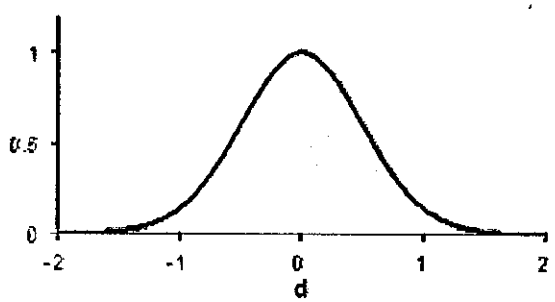
که در آن t نشانگر گام زمانی و L متغیر کوچکی به نام *نرخ یادگیری* است که در طول زمان کم می‌شود. در واقع این رابطه بیان می‌کند که وزن اصلاح شده جدید برابر وزن قدیمی *(W)* به اضافه بخشی *(L)* از تفاوت بین وزن قدیمی و بردار ورودی *(X)* است.

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

$$L(t) = L_0 e^{-t/\lambda} \quad t = ۱,۲,۳,... \qquad (۳۱-۳)$$

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

در رابطه (۳-۳۰) 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 (۳-۳۰) باید کمی تغییر یابد:

$$W(t+۱) = W(t) + \Theta(t)L(t)(X(t) - W(t)) \qquad (۳۲-۳)$$

Θ نمایانگر مقدار تأثیر فاصله یک گره از *BMU* روی یادگیری آن است و با رابطه (۳- ۳۳) بیان می‌شود. که در آن *dist* فاصله گره از *BMU* و σ عرض تابع همسایگی محاسبه شده در رابطه (۳-۲۸) است. همچنین توجه کنید Θ نیز در طول زمان کاهش می‌یابد.

$$\Theta(t) = e^{-dist^2/2\sigma^2(t)} \quad t = ۱,۲,۳,... \qquad (۳۳-۳)$$

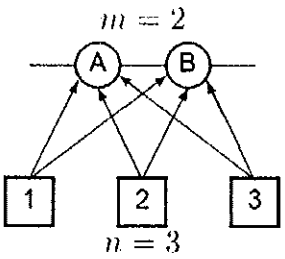
مثال عددی

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: $w_A = (۲، -۱، ۳)$ $w_B = (-۲، ۰، ۱)$

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:

$$\| x - w_A \| = \sqrt{(۱-۲)^۲ + (-۲+۱)^۲ + (۲-۳)^۲} = \sqrt{۳}$$

$$\| x - w_B \| = \sqrt{(۱+۲)^۲ + (-۲-۰)^۲ + (۲-۱)^۲} = \sqrt{۱۴}$$

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[۱] 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[۲] 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[۳]، 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]. می‌توان این اطلاعات رنگی را روی نقشه زمین شکل (۳-۲٤) رسم کرد.

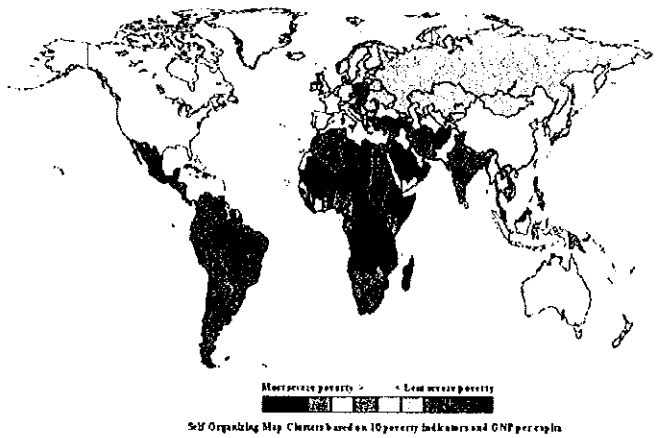
۱- Poverty Map

۲- Quality-of-Life

۳- U-Matrix

شکل ۳-۲۳) کشورها روی یک SOM (نقشه خودسازمانده) بر مبنای شاخصهای فقر سازمان یافته‌اند

نقشه فقر جهانی



شکل ۳-۲٤) نقشه جهان که در آن کشورها بر مبنای SOM (نقشه خودسازمانده) در شکل قبل رنگ‌آمیزی شده‌اند

---

[۲]- Association Rules

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[۲] 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[۱] 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[۲] 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 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.

۱- Market Basket -Basket Data

۲- Confidence

در اینجا به مثالهایی از کاربرد قواعد تلازمی اشاره می‌شود:

- بررسی ارتباط بین توانایی خواندن کودکان با خواندن داستان توسط والدین برای آنها.
- 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 تلازمی ماهیتاً قواعد احتمالی هستند. به‌عبارت دیگر قاعده X⇒A لزوماً قاعده X+Y⇒A را نتیجه نمی‌دهد، زیرا این قاعده ممکن است از شرط حداقل پشتیبان برخوردار نباشد. به‌طور مشابه قواعد X⇒Y و Y⇒Z لزوماً قاعده X⇒Z را نتیجه نمی‌دهند زیرا قاعده اخیر ممکن است از شرط حداقل اطمینان برخوردار نباشد. [۱]

۴-۱- 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 تلازمی به شکل زیر می‌باشد:

$X \Rightarrow Y$ [اطمینان , پشتیبان]

به‌طوری‌که داریم: $X \subset I,\ Y \subset I$ و $X \cap Y = \varnothing$

- پشتیبان [۱] (X,Y): نشان‌دهنده درصد یا تعداد مجموعه تراکنش‌های D است که شامل هر دوی X و Y باشند.
- اطمینان[۲]: میزان وابستگی یک کالای خاص را به دیگری بیان می‌کند و مطابق فرمول زیر محاسبه می‌شود:

۱- Support

۲ Confidence

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. غالباً قاعدی انتخاب می‌شوند که عدد اطمینان بزرگی داشته باشند.

فرض کنید اطلاعات مشتریانی که مجصول X را خریده‌اند همچنین علاقه دارند در همان زمان از محصول Y نیز بخرند در قاعده تلازمی زیر نشان داده شده است.

X=>Y (٪۶۰=اطمینان و ٪۲۰=پشتیبان)

شاخصهای اطمینان و پشتیبان قواعد بیانگر جذابیت آنها هستند. این دو شاخص به ترتیب مفید بودن و اطمینان از قواعد مکشوفه را نشان می‌دهند. پشتیبان ٪۲۰ برای قاعده تلازمی فوق به این معنی است که ٪۲۰ همه تراکنشهای موجود نشان می‌دهند که کالای X و Y با هم خریداری شده‌اند. اطمینان ٪۶۰ به این معنی است که ٪۶۰ مشتریانی که کالای X را خریده‌اند کالای Y را نیز خریداری کرده‌اند.

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:

$D = \{T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8\}$

T_1 = {جعفری، پیاز، زیتون، خیار، گوجه‌فرنگی}

T_2 = {جعفری، خیار، گوجه‌فرنگی}

T_3 = {نان، نمک، گوجه‌فرنگی، پیاز، جعفری، خیار}

T_4 = {نان، پیاز، خیار، گوجه فرنگی}

T_5 = {پیاز، نمک، گوجه‌فرنگی}

T_6 = {پنیر، نان}

T_7 = {خیار، پنیر، گوجه‌فرنگی}

T_8 = {کره، نان}

فرض کنیم یک قاعده تلازمی به شکل زیر داریم:

$X \Rightarrow Y$ [اطمینان , پشتیبان]

{پیاز, جعفری} ⇒ {خیار, گوجه‌فرنگی}

$X =$ { خیار , گوجه‌فرنگی }

$Y =$ { جعفری, پیاز }

$X \cup Y =$ {گوجه‌فرنگی، خیار، جعفری، پیاز} $= \{T_1, T_2\}$

پشتیبان $(X \cup Y) = \frac{۲}{۸} = ۰/۲۵$

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.

$T = \{T_3, T_4, T_7, T_2, T_1\}$

یعنی {خیار, گوجه‌فرنگی} در T_1 و T_2 و T_7 و T_4 و T_3 خریداری شده‌اند.

بنابراین داریم:

پشتیبان $(x) = \frac{۵}{۸} =$٪۶۲

پشتیبان = اطمینان $(X \cup Y)$ / پشتیبان $(X) = (\frac{۲}{۸})/(\frac{۵}{۸}) =$۲/۵=٪۴۰

یعنی هنگامی که افراد «خیار و گوجه‌فرنگی» می‌خرند، در ۴۰٪ 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 مقدار پشتیبان و اطمینان بیش از مقدار آستانه باشند. با استفاده از این مفاهیم پیدا کردن قواعد تلازمی در دو گام خلاصه می‌شود، یعنی پیدا کردن مجموعه‌های مکرر و استخراج قواعد قوی.

---

[1]- Frequent

[2]- Strong

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 ⇒ Financial management software [sup=۲%, confidence=۶۰%]
age (X,"۳۰..۳۰") and income (X,"۴۲k..۴۸k") ⇒ 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) ⇒ buys (X, "financial management software")

اما قاعده زیر سه بعدی است و ابعاد سن، درآمد و خرید را شامل می‌شود.

Age (X,"۳۰..۳۹") and income (X,"۴۲k..۴۸k") ⇒ buys (X, high resolution TV)

از آنجایی که داده‌ها می‌توانند در سطوح[۱] و یا مقیاسهای[۲] 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:

age (X ,"۳۰..۳۹") ⇒ buys (X , "Laptop ")
age (X ,"۳۰..۳۹") ⇒ 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.

۱- Level
۲- Scale

استخراج قواعد تک‌سطحی تک‌بعدی دودویی

قبل از ارائه 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 $L_k = \{a[1], a[2], ..., a[k]\}$ باشد، مطابق این قاعده باید رابطه «$a[1] < a[2] < ... < a[k]$» 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_k ها 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:

برای تمامی اقلام مکرر L همه زیرمجموعه‌های غیرتهی آن را (s) درنظر می‌گیریم. برای تمامی این زیرمجموعه‌ها، 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.

(۲-٤) حداقل اطمینان = <((s) پشتیبان / (L) پشتیبان)

پروسه استخراج قواعد تلازمی عبارت است از:

- ابتدا همه اقلام مکرر را که بیشتر یا مساوی با آستانهٔ پشتیبان هستند بیابید.
- برای تمامی اقلام مکرر همه زیرمجموعه‌های آنها را استخراج کنید.

---

[1]- Lexicographic Order

- همه قواعد ممکن را استخراج کنید.
- قواعدی را بپذیرید که از بیشتر و یا آستانه اطمینان برخوردارباشند.

در اینجا برای پیدا کردن این قواعد از الگوریتم ساده *Apriori* یا الگوریتم «پیشینار» 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 زیرمجموعهٔ یک مجموعه مکرر، خود نیز مکرر است. مثلاً اگر مجموعه $A =$ {سیگار، نان، شیر} مکرر باشد آنگاه مجموعه‌های زیر نیز مکرر هستند.

{سیگار} ,{ نان} ,{شیر} {سیگار ,نان} ,{سیگار ,شیر} ,{ نان ,شیر}

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 $k-1$ عضوی یا همان L_{k-1} 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-1} با آنها، L_k به دست آید. این مجموعه از عناصر را C_k نامیده و مرحله به‌دست آوردن آنها را پیوست[۲] می‌نامیم. مرحله بعد اضافه کردن این عناصر به مجموعه‌های قبلی است که آن را مرحله هرس[۳] می‌نامیم. در زیر این دو مرحله شرح داده می‌شوند.

مرحله پیوست

ابتدا باید مطمئن شویم که عناصر بر مبنای ترتیب حروف الفبا مرتب شده‌اند. دو مجموعه از L_{k-1} با یکدیگر قابل پیوست هستند اگر $k-2$ عنصر اول آنها با یکدیگر برابر باشند. یعنی:

۱- مشابه این اصل در خوشه‌بندی اطلاعات بر اساس چگالی در تعیین مقادیر همسایگی استفاده می‌شود.

۲- Join

۳- Prune

$(l_1[1]) = l_2[1]) \wedge (l_1[2] = l_2[2]) \wedge (l_1[k-2] = l_2[k-2]) \wedge (l_1[k-1] < l_2[k-1])$ توجه کنید که دو عنصر آخر 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: ۴ > ۳ > ۲ > ۱ پس می‌توان مجموعهٔ ترکیب شدهٔ زیر را به‌دست آورد:

L_{k-1} = مجموعهٔ اول = (۱ ۲ ۳)

|| ||

C_k = مجموعهٔ ترکیب شده = (۱ ۲ ۳ ٤)

|| ||

L_{k-1} = مجموعهٔ دوم = (۱ ۲ ٤)

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:

$$\text{اطمینان } (A \Rightarrow B) = P(B \mid A) = \frac{\text{پشتیبان } (A \cup B)}{\text{پشتیبان } (A)} \qquad (۳-٤)$$

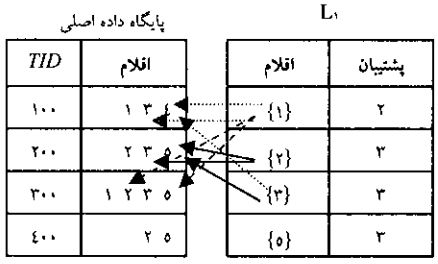
برای هر مجموعه مکرر L تمام زیر مجموعه‌های غیر تهی را در نظر می‌گیریم. برای هر زیرمجموعه s قواعد را به‌صورت زیر شکل می‌دهیم. "$s \Rightarrow (L-s)$" سپس اطمینان را حساب کرده و اگر بیشتر از حداقل قابل قبول بود آن را می‌پذیریم.

۴-۱-۱- 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_1 ثبت می‌شوند.



شکل۴-۱) قدم اول الگوریتم AIS

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.

۱- R. Agrawal, T. Imielinski, and A. Swami

	اقلام	پشتیبان
TI_1	{۱۳}*	۲
	{۱۴}	۱
	{۳۴}	۱
TI_2	{۲۳}*	۲
	{۲۵}*	۳
	{۳۵}*	۲
TI_3	{۱۲}	۱
	{۱۵}	۱

شکل ٤-۲) قدم دوم الگوریتم AIS

اقلام	پشتیبان
{۱۳۴}	۱
{۲۳۵}*	۲
{۱۳۵}	۱

شکل ٤-۳) قدم سوم الگوریتم AIS

در قدم سوم همانند قدم دوم به محاسبهٔ C_2 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 {۱,۵} , {۱,۲} , {۳,٤} , {۱,٤}

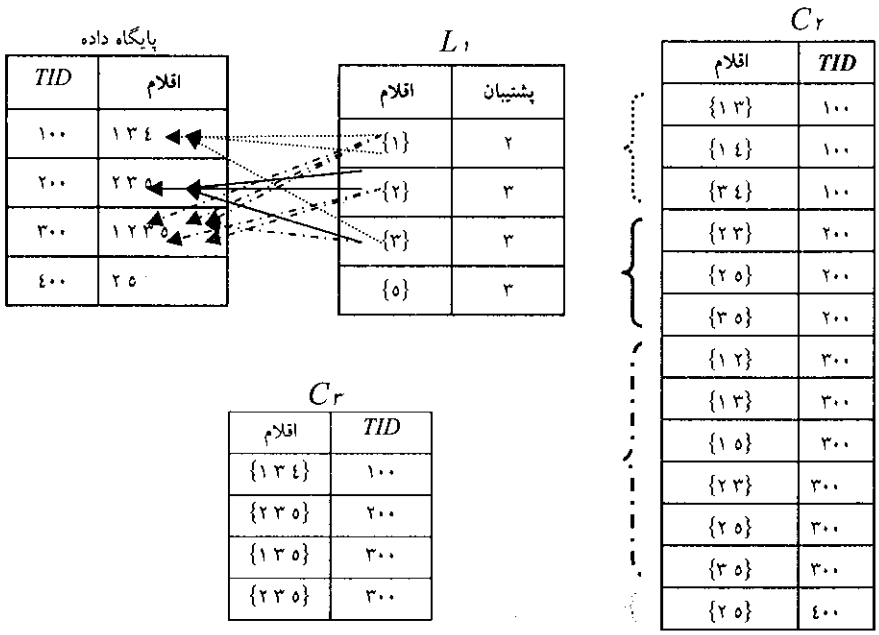
۴-۱-۲- الگوریتم SETM

این الگوریتم توسط هوتسما[1] در سال ۱۹۹۵ 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]

مشابه الگوریتم *AIS*، این الگوریتم نیز چندین گذر برروی پایگاه داده انجام می‌دهد. این الگوریتم به شکل زیر می‌باشد.

۱- Houtsma

۲- Srikant



شکل٤-٤) الگوریتم SETM

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_k نگهداری کرده و سپس پشتیبان هر یک از C_K ها با جمع‌کردن تعداد تکرار آنها در مرحلهٔ قبل محاسبه شده و C_k 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.

۴-۱-۳- الگوریتم Apriori یا پیشینار

این الگوریتم در سال ۱۹۹٦ توسط چیونگ[1] ابداع شد و یکی از مهم‌ترین یافته‌ها در تاریخ استخراج قواعد تلازمی 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.

```
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_4به‌صورت زیر باشد:

---

[1]- Cheung

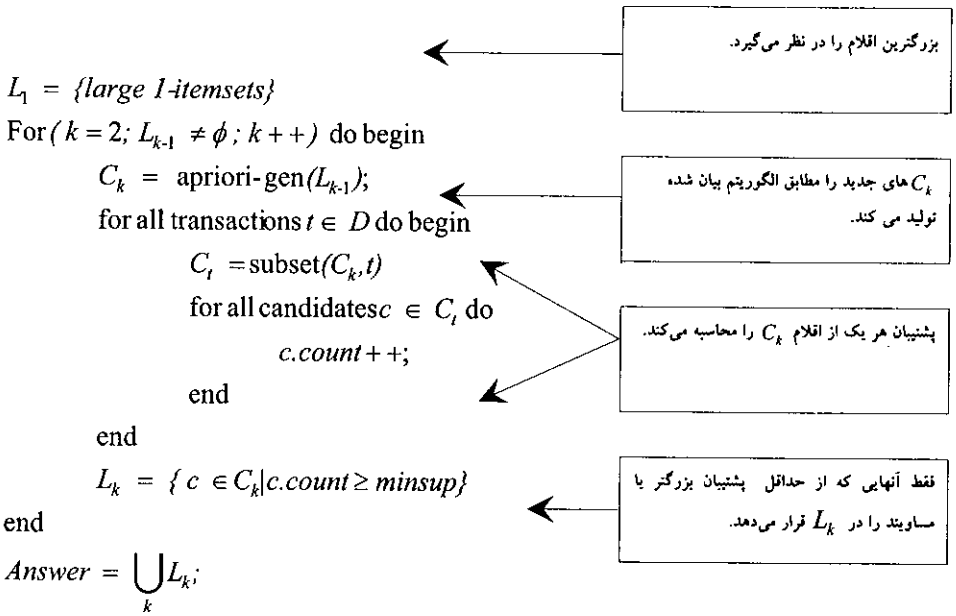
$L_۳ = \{ \{۱ ۲ ۳\}, \{۱ ۲ ٤\}, \{۱ ۳ ٤\}, \{۱ ۳ ٥\}, \{۲ ۳ ٤\} \}$

پس از مرحله اتصال خواهیم داشت:

$\{\{۱ ۲ ۳ ٤\}, \{۱ ۳ ٤ ٥\}\}$

و پس از مرحله هرس خواهیم داشت:

$C_٤ = \{۱ ۲ ۳ ٤\}$

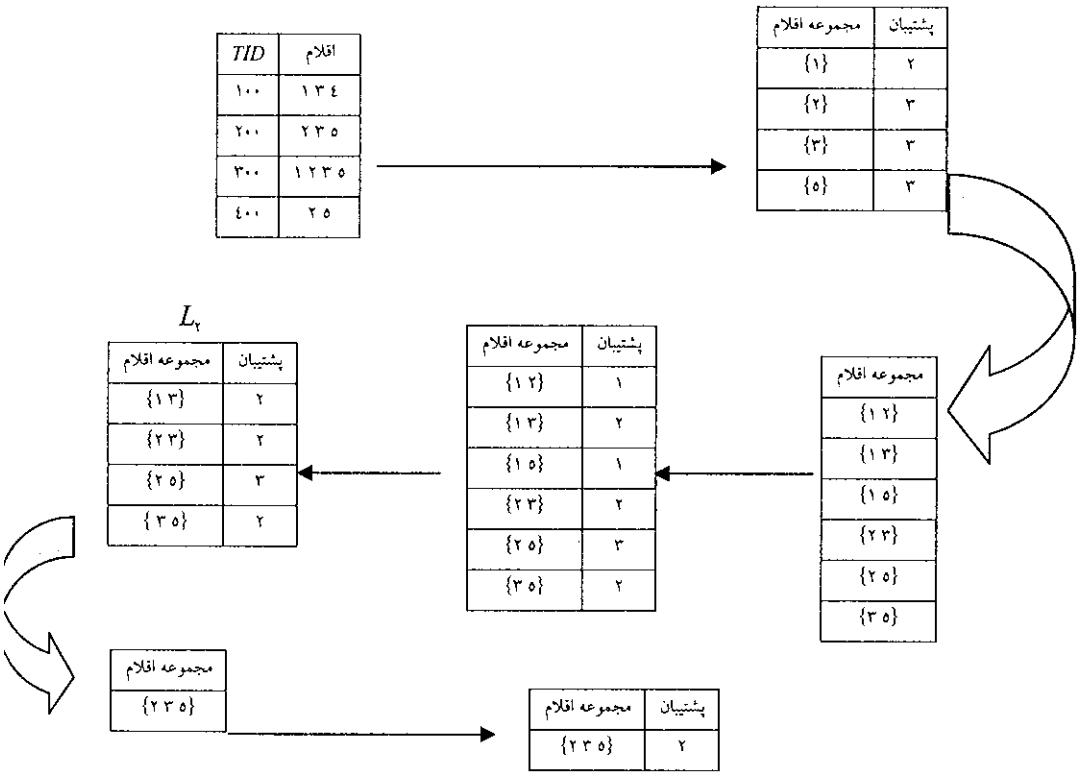


شکل ٤-٦) توضیح الگوریتم Apriori

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. 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 { {۲,۵} , {۲,۳}} چون ۲ در هر دو مشترک است می‌توان سه تایی جدیدی بر اساس ترکیب آنها ساخت، به‌طوری‌که بر اساس قاعده ترتیب الفبا مجموعهٔ {۲,۳,۵} ساخته شود.



شکل ٤-۷) توضیح الگوریتم Apriori با ارائه یک مثال

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= ٪٦۰

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}
٥	{ چیپس ,لیموناد }

در ابتدا پشتیبان تک تک اقلام را محاسبه می‌کنیم:

جدول ٤-٢) پشتیبان اقلام خریداری شده

(C_1) اقلام خریداری شده	
اقلام	پشتیبان
آب پرتقال	٪٦٠
لیموناد	٪٨٠
شیر	٪٢٠
شیشه‌پاک‌کن	٪٤٠
پاک‌کننده	٪٢٠
چیپس	٪٢٠

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

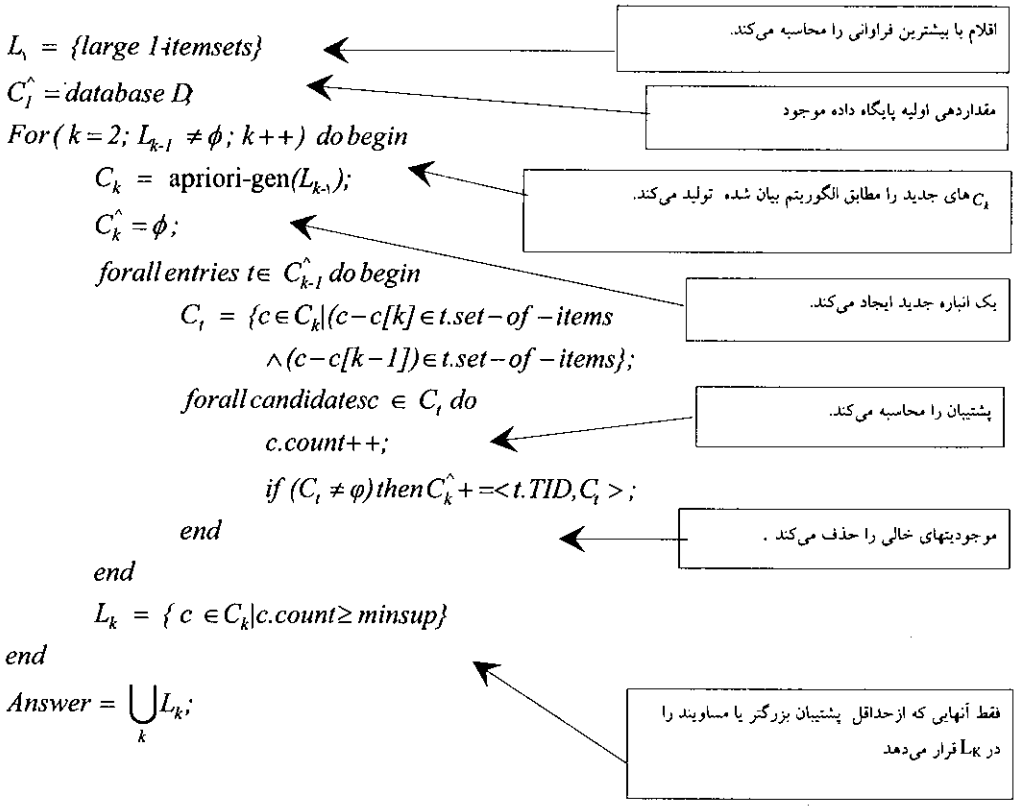
همان‌طورکه مشاهده شد، تفاوت عمدهٔ این الگوریتم با الگوریتمهای دیگر در حجم محاسبات کمتر آن است. در این الگوریتم اقلام زاید کمتری در هر مرحله ایجاد شده و با آزمایشهای مختلفی که برای کشف اقلام مکرر توسط *IBM RS/*٦٠٠٠ انجام شد مشخص شد که این الگوریتم عملکرد بسیار بهتری نسبت به دیگر الگوریتم قبلی دارد.

معایب الگوریتم

برای محاسبه پشتیبان اقلام کاندیدا، الگوریتم همه تراکنشها را بررسی می‌کند و بنابراین نیازمند زمان زیادی است.

۴-۱-۴- 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 $\hat{C_k}$ برای محاسبه پشتیبان استفاده می‌کند. مشابه الگوریتم *SETM* اعضای این الگوریتم نیز به فرم $<TID,X_k>$ ذخیره می‌شوند.

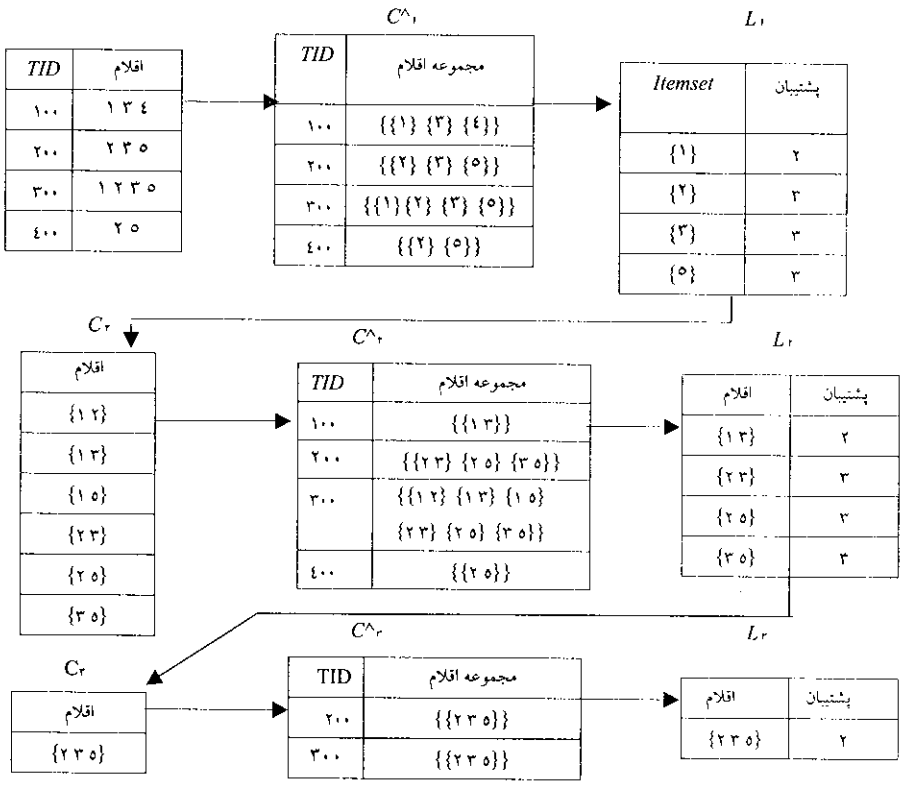


شکل ٤- ٨) الگوریتم AprioriTid

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.

در شکل (٤-۹) 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 $C_k^{\wedge}$ تمامی اقلام تکی ساخته شده بر اساس پایگاه دادهٔ اصلی با ذکر شمارهٔ تراکنش *TID* بیان می‌شوند.



شکل ٤- ۹) توضیح الگوریتم AprioriTid با یک مثال

در جدول $C_۲$ مجموعه‌های دو تایی از اقلام موجود در جدول $L_۱$ ساخته می‌شوند و در جدول $C_۲^*$ شمارهٔ TID این مجموعه اقلام نیز ذکر می‌شوند. در مجموعهٔ $L_۲$، پشتیبان این مجموعه اقلام محاسبه شده و آنهایی که از حداقل کمتر هستند حذف می‌شوند. جدول $C_۳$ بر اساس قاعده ترتیب الفبا و بر مبنای داده‌های جدول $L_۲$، مجموعه اقلام سه‌تایی‌ها را می‌سازد به‌عنوان مثال از ترکیب دو مجموعه {۲,۳} و {۲,۵} مجموعهٔ سه تایی {۲,۳,۵} ساخته می‌شوند.

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

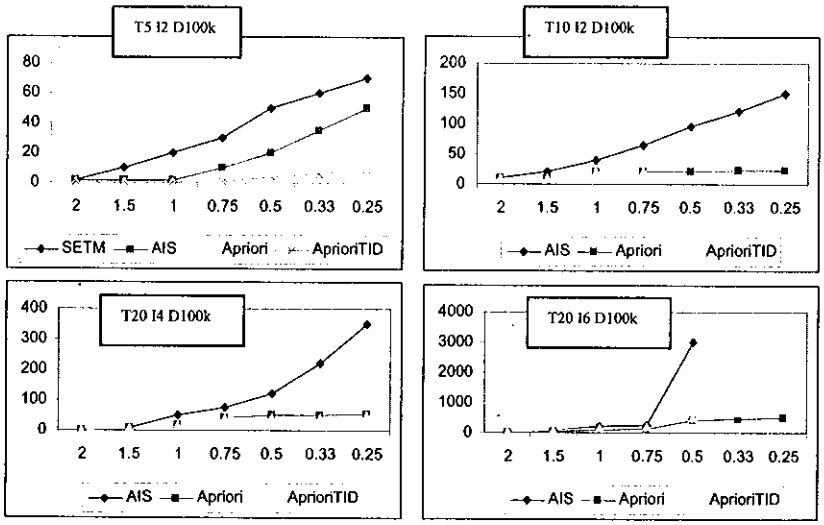
تعداد اقلام = ۱۰۰۰

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

نمادی بالای هر کدام از نمودارها نوشته شده است که معرف تراکنشها، میانگین اندازهٔ اقلام مکرر و میانگین اندازهٔ تراکنشها می‌باشد به‌عنوان نمونه $T۵I۲D۱۰۰K$ 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.

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 $\hat{C_k}$ بجای پایگاه داده در نظر گرفته می‌شوند. اگر $\hat{C_k}$ بتواند در حافظه جای گیرد، این الگوریتم سریعتر از *Apriori* عمل خواهد کرد. زمانیکه $\hat{C_k}$ 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.

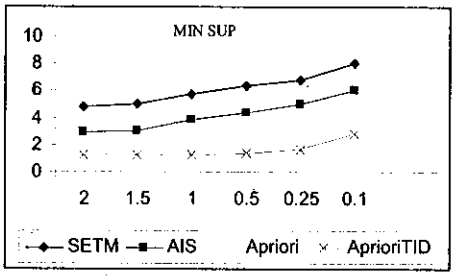


شکل ٤-١٠) تغییرات رفتار الگوریتمهای مختلف

داده‌های واقعی

فروشگاه خرده‌فروشی شامل:

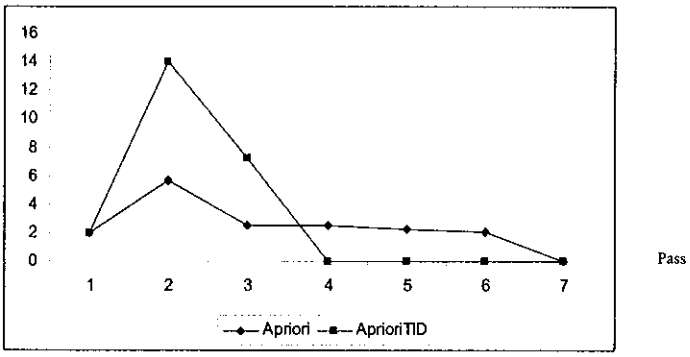
- ٦٣ بخش
- ٤٦٨٧٣ تراکنش(با میانگین اندازه ٤٧/٢)



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

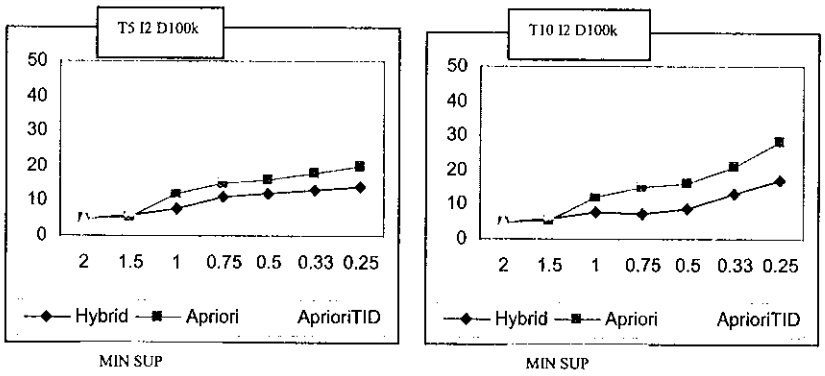
درمراحل انتهایی $C_k^{\wedge}$ به اندازه کافی کوچک شده و حافظه مصرفی کم می‌شود. بنابراین از فاز ٤ به بعد زمان اجرای الگوریتم *AprioriTid* بسیار کم شده و تقریباً این زمان برابر صفر شده است. به منظوراستفاده بهینه از این‌دو الگوریتم، الگوریتم جدیدی بنام *AprioriHybrid* شکل گرفت.

۴-۱-۵- 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(1998),Fast algorithm for mining association rules,In Proc. of the VLDB Conference, Santiago, Chile, September 1994. Expanded version available as IBM Research Report, RJ9839, June 1994.*

فصل پنجم

دسته‌بندی و پیش‌بینی

دسته‌بندی و پیش‌بینی دو نوع عملیات برای تحلیل داده‌ها و استخراج مدل به‌منظور توصیف دسته‌های مهم داده‌ها، فهم و پیش‌بینی رفتار آیندهٔ آنها می‌باشند. مدلهای دسته‌بندی در تحلیل داده‌های گسسته و طبقه‌ای بکار رفته 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.

۵-۱- مفاهیم دسته‌بندی

بسیاری از روشهای دسته‌بندی و پیش‌بینی در علومی مانند یادگیری ماشین، بازشناسی الگو و آمار کاربرد دارند. در این فصل به روشهای سادهٔ دسته‌بندی از قبیل درختهای تصمیم، شبکه‌های عصبی، نزدیکترین همسایگی و دیگر روشها اشاره شده است. دسته‌بندی برای تخصیص یک برچسب به مجموعه‌ای ازداده‌ها که هنوز دسته‌بندی نشده‌اند، استفاده می‌شود. پس از آن داده‌ها بر اساس ویژگیهایشان به دسته‌هایی که نام آنها از قبل مشخص می‌باشد، تخصیص داده می‌شوند.

معمولا هنگامی‌که یک افته[1] بر اساس ویژگیهایش به دسته‌ای تخصیص یافت آن افته بر اساس بر چسب دسته توصیف می‌شود. به بیان دیگر 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.

۵-۱-۱- 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 درون خوشه و تفاوت بیرون خوشه) تقسیم می‌کند. لذا اگر بخواهیم با استفاده از مفهوم یادگیری، دسته‌بندی و خوشه‌بندی را متمایز کنیم، باید بگوییم دسته بندی یادگیری با نظارت و خوشه‌بندی یادگیری بدون نظارت است.

یادگیری با نظارت یا دسته‌بندی عبارتست از یادگیری به‌وسیله نمونه‌ها. به‌عبارت دیگر در این روش دسته‌ها از قبل مشخص هستند. ولی در یادگیری بدون نظارت یا خوشه‌بندی، خوشه‌ها مشخص نیستند و هدف خوشه‌بندی، تعیین خوشه‌های داده‌ها است.

۵-۱-۲- فرایند دو مرحله‌ای دسته‌بندی

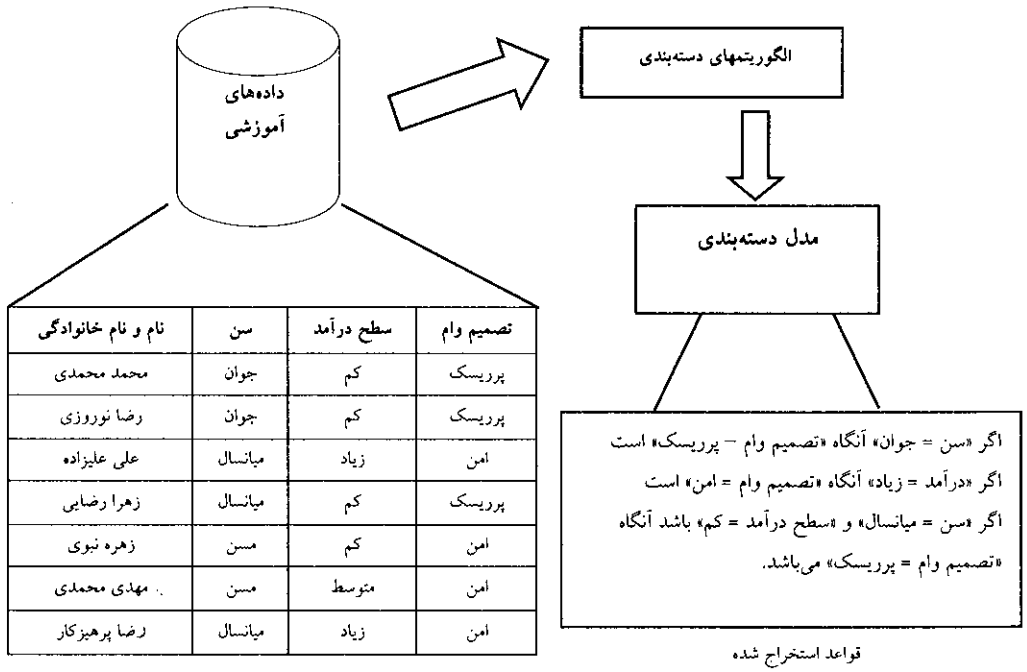
دسته‌بندی داده‌ها، فرآیندی دو مرحله‌ای است. اولین مرحله ساخت مدل و دومین مرحله استفاده از مدل و پیش‌بینی از طریق داده‌های قبلی می‌باشد.[۱]

مرحله اول یا ساخت مدل: این مرحله عبارتست 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 برچسب 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.

---

[1] Case

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 هر دادهٔ x به یک کلاس y اختصاص دارد. هدف دسته‌بندی، یادگیری این تابع می‌باشد تا بتوان به‌راحتی کلاس هر داده را پیدا کرد. در شکل (۵-۱) این نگاشت به‌صورت قواعد دسته‌بندی بیان شده که تعیین‌کنندهٔ پرخطر و یا بی‌خطر بودن اعطای وام به مشتریان می‌باشد.



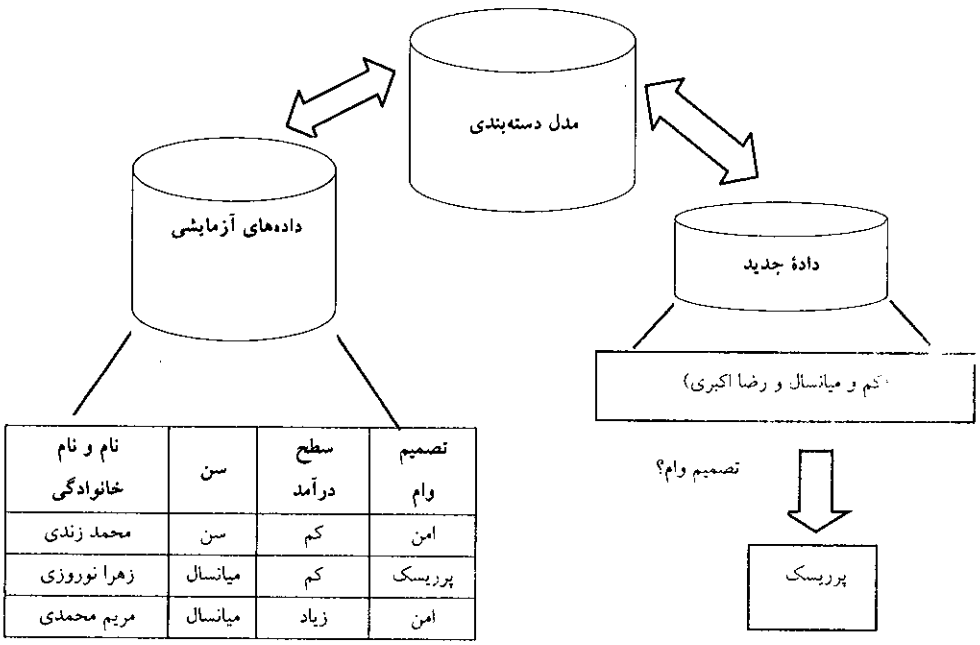
نام و نام خانوادگی	سن	سطح درآمد	تصمیم وام
محمد محمدی	جوان	کم	پرریسک
رضا نوروزی	جوان	کم	پرریسک
علی علیزاده	میانسال	زیاد	امن
زهرا رضایی	میانسال	کم	پرریسک
زهره نبوی	مسن	کم	امن
مهدی محمدی	مسن	متوسط	امن
رضا پرهیزکار	میانسال	زیاد	امن

شکل ۵-۱) یک نمونه از ساخت مدل بر اساس داده‌های قدیمی

مرحلهٔ دوم استفاده از مدل: این مرحله دارای دو بخش است. در بخش نخست مدل ساخته شد، مورد آزمون واقع می‌شود تا دقت پیش‌بینی آن بررسی شود. در بخش دوم نیز مدلی که دارای دقت مناسبی است، برای دسته بندی داده ها به کار گرفته می‌شود. به منظور تخمین دقت

پیش‌بینی مدل مجموعه‌ای از داده‌های آزمایشی به‌طور اتفاقی از میان داده‌ها انتخاب شده و مدل روی آنها اجرا می‌شود. هدف اصلی در اینجا بالاتر بردن تخمین دقت مدل می‌باشد تا به هنگام استفاده هر داده را به دسته مناسب آن تخصیص دهد.

برای اینکار داده‌های آموزشی را می‌توان به دو قسمت تقسیم کرد: اول آن دسته که مدل بر اساس آنها ساخته می‌شود و دوم گروهی که برای ارزیابی مدل استفاده می‌شود. داده‌های گروه دوم که دسته آنها مشخص است را به مدل داده و خروجی مدل را با دسته‌های مشخص توسط مدل مقایسه کرده و دقت کل مدل استخراج می‌شود. در واقع، برچسب شناخته شده از نمونه آزمون با نتایج دسته‌بندی مقایسه می‌شود. 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.



نام و نام خانوادگی	سن	سطح درآمد	تصمیم وام
محمد زندی	سن	کم	امن
زهرا نوروزی	میانسال	کم	پرریسک
مریم محمدی	میانسال	زیاد	امن

شکل ۵-۲) دسته‌بندی داده‌های وام

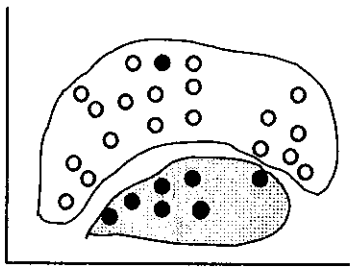
۵-۱-۳- روشهای مختلف دسته‌بندی

روشهای زیادی برای دسته‌بندی وجود دارد که از آن جمله می‌توان به موارد ذیل اشاره کرد:

- بیز ساده و شبکه‌های بیزی
- نزدیک‌ترین همسایگی
- شبکه‌های عصبی
- درخت تصمیم
- رگرسیون (خطی، غیرخطی، لجستیک)
- در اینجا به اختصار سه روش شبکه‌های عصبی، درخت تصمیم و رگرسیون خطی بیان شده ودر ادامه هر یک از روشها به تفصیل بیان می‌شوند

شبکه‌های عصبی

با این روش می‌توان نواحی با اشکال پیچیده را پوشش داد. این روش دقیق‌تر از سایر روشهاست و کاملاً می‌تواند برازش شود.

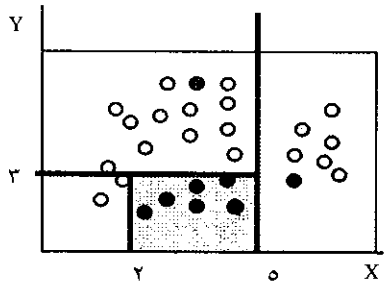


شکل ۵-۳) شبکه‌های عصبی

درخت تصمیم

درخت تصمیم فضا را به نواحی مستطیلی تقسیم می‌کند به طوریکه در هر مستطیل داده‌ها بر اساس برچسب دسته همگن باشند.

$$if\ X > ۵\ then\ \ classA\ \ else\ if\ Y > ۳\ then\ classA$$
$$else\ if\ X > ۲\ then\ classB\ \ else\ blue$$



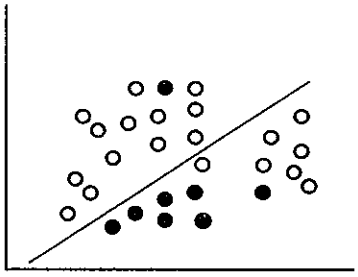
شکل ۵-٤) درخت تصمیم

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 ۵-۵) رگرسیون خطی

۵-۲- 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؟

۵-۲-۱- 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 نظربگیرید. هدف ما پیش‌بینی و انتخاب دسته‌ای است که $P(C=c|A_1=a_1 \cup A_2=a_2 ... \cup A_n=a_n)$ ماکزیمم شود. فرمول ساده بیز عبارت است از:

$$P(A/B)=\frac{P(B/A)P(A)}{P(B)} \qquad (۲-۵)$$

در فرمول: $P(C=c|A_1=a_1 \cup A_2=a_2 ... \cup A_n=a_n)$ با استفاده از قاعده بیزین داریم:

$$P(C=c|\ A_1=a_1 \cup A_2=a_2\ ... \cup A_n=a_n) \qquad (۳-۵)$$

$$=\frac{P(A_1=a_1 \cup A_2=a_2 \cup ... A_n=a_n|\ C=c)\cdot P(C=c)}{P(A_1=a_1 \cup A_2=a_2 \cup A_n=a_n)} \qquad (۴-۵)$$

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 $P(A_1=a_1 \cup A_2=a_2 \cup ... A_n=a_n|\ C=c)$ محاسبه شود. از طرفی با فرض استقلال داریم:

$$P(X\ |\ C_i)=\prod_{k=1}^{n} P(X_k\ |C_i)=P(X_1\ |C_i)*P(X_2\ |C_i)...*P(X_n\ |C_i) \qquad (۵-۵)$$

بنابراین رابطه $P(A_1=a_1 \cup A_2=a_2 \cup ... A_n=a_n|\ C=c)$ به‌همین ترتیب و با همین منطق قابل گسترش است و داریم:

$$P(A_1=a_1 \cup ... \cup A_k=a_k|C=c)=P(A_1=a_1|C=c)\times ... \times P(A_k=a_k\ |C=c)$$

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:

$$P(A_1=a_1 \cup ... \cup A_k=a_k | C=c) = P(A_1=a_1|C=c) \times ... \times P(A_k=a_k|C=c)$$

و به‌همین ترتیب برای A_1 تا A_k نیز همین فرض برقرار است. حال می‌خواهیم $P(A_1=a_1|C=c)$ را تخمین بزنیم. مثال زیر به این مسئله کمک می‌کند. در این مثال، ویژگی داده‌ها عبارتند از: سن، سطح درآمد، عضویت دانشجو و میزان اعتبار. داده‌های آموزشی در جدول (۵-۱) آمده است.

جدول ۵-۱) داده‌های خریداران کامپیوتر

خریدار کامپیوتر	اعتبار	دانشجو	درآمد	سن
خیر	متوسط	خیر	بالا	جوان
خیر	عالی	خیر	بالا	جوان
بلی	متوسط	خیر	بالا	میانسال
بلی	متوسط	خیر	متوسط	بالای ۴۰ سال
بلی	متوسط	بلی	کم	بالای ۴۰ سال
خیر	عالی	بلی	کم	بالای ۴۰ سال
بلی	عالی	بلی	کم	میانسال
خیر	متوسط	خیر	متوسط	جوان
بلی	متوسط	بلی	کم	جوان
بلی	متوسط	بلی	متوسط	بالای ۴۰ سال
بلی	عالی	بلی	متوسط	جوان
بلی	عالی	خیر	متوسط	میانسال
بلی	متوسط	بلی	بالا	میانسال
خیر	عالی	خیر	متوسط	بالای ۴۰ سال

برچسب مورد نظر عبارتست از: «خرید کامپیوتر» که دو مقدار مجزای {خیر و بلی} دارد و بنابراین داریم:

دسته اول: «بلی = خرید کامپیوتر» = C_1

دسته دوم: «خرید کامپیوتر = خیر» $= C_2$

داده‌ای که قرار است دسته‌اش تشخیص داده شود، عبارتست از:

$X =$ (سن = جوان , درآمد = متوسط , دانشجو = بله , میزان اعتبار = متوسط)

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:

$P($خرید کامپیوتر = بله$) = \frac{۹}{۱۴} = ۰/۶۴۳$

$P($خرید کامپیوتر = خیر$) = \frac{۵}{۱۴} = ۰/۳۵۷$

برای محاسبهٔ $P(X|C_i)$ برای $i = ۱,۲,...$ داریم:

$P($سن = جوان $|$ خرید کامپیوتر = بله$) = \frac{۲}{۹} = ۰/۲۲۲$

$P($سن = جوان $|$ خرید کامپیوتر = خیر$) = \frac{۳}{۵} = ۰/۶۰۰$

$P($درآمد = سطح متوسط $|$ خرید کامپیوتر = بله$) = \frac{۴}{۹} = ۰/۴۴۴$

$P($درآمد = سطح متوسط $|$ خرید کامپیوتر = خیر$) = \frac{۲}{۵} = ۰/۴۰۰$

$P($دانشجو = بله $|$ خرید کامپیوتر = بله$) = \frac{۶}{۹} = ۰/۶۶۹$

$P($دانشجو = خیر $|$ خرید کامپیوتر = خیر$) = \frac{۱}{۵} = ۰/۲۰۰$

$P($میزان اعتبار = متوسط $|$ خرید کامپیوتر = بله$) = \frac{۶}{۹} = ۰/۶۶۷$

$P($میزان اعتبار = متوسط $|$ خرید کامپیوتر = خیر$) = \frac{۲}{۵} = ۰/۴۰۰$

برای همه ویژگیهای «سن»، «سطح درآمد»، «عضویت دانشجویی» و «میزان اعتبار» احتمال‌ها را محاسبه کرده و سپس $P(X|$ خرید کامپیوتر = بلی$)$ را محاسبه می‌کنیم. برای این‌کار در موارد بالا آنهایی که برچسب دسته آنها بلی است را انتخاب کرده و محاسبات را ادامه می‌دهیم، یعنی:

$P(X|$ خرید کامپیوتر = بلی$) =$

$P($سن = جوان $|$ خرید کامپیوتر = بلی$) \times P($درآمد = سطح متوسط $|$ خرید کامپیوتر = بلی$) \times$

$P($دانشجو = بلی $|$ خرید کامپیوتر = بلی$) \times P($میزان اعتبار = متوسط $|$ خرید کامپیوتر = بلی$)$

در نتیجه داریم:

$P(X \mid$ خریدار کامپیوتر = بلی) = ۰/۲۲۲×۰/۴۴۳×۰/۶۶۷×۰/۶۶۷ = 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.

۵-۲-۲- 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[۱] 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[۲] 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 احتمال شرطی» یا CPT[۳] 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. به‌عنوان مثال برای متغیر «سرطان ریه» داریم:

P (سرطان ریه = بله | سابقه خانوادگی دارد , سیگار می‌کشد = بله) = ۰/۸

P (سرطان ریه = خیر | سابقه خانوادگی دارد , سیگار می‌کشد = خیر) = 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.

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

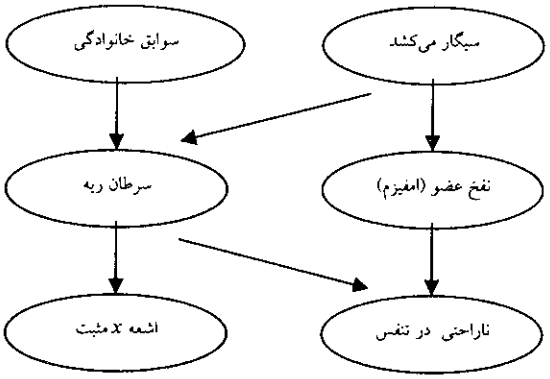
۱- Mutually Exclusive

۲ Acyclic

۳- Conditional Probability Table

$$P(x_1, x_2, ..., x_n) = \prod_{i=1}^{n} P(x_i \mid parents\ (Y_i)) \qquad (۶-۵)$$

Parent (Y) والدین Y هستند.



شکل ۵- ۶) اطلاعات مربوط به ارتباط سرطان ریه و سوابق خانوادگی و کشیدن سیگار

در رابطه (۵-۵) مقدار $P(x_i, x_2, ...x_n)$ 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 ترسیم تخمینی گراف نیز وجود دارد [۱].

۵-۳- دسته‌بندی بر مبنای نزدیکترین همسایگی

در یک نگاه کلی می‌توان دسته‌بندها[۱] را به دو گروه مشتاق[۲] و کاهل[۳] تقسیم کرد. در نوع مشتاق، مدلی از داده‌ها در مرحله آموزش ساخته می‌شود. درختهای تصمیم، نمونه‌ای از دسته‌بندهای مشتاق 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[۴] هم می‌گویند.

تفاوت دو روش در این است که نوع مشتاق زمان زیادی را در مرحلهٔ آموزش، صرف ساخت مدل کرده و در زمان دسته‌بندی بسیار سریع عمل می‌کند، در نقطه مقابل، نوع کاهل آن، در هنگام ورود داده‌ها در مرحله آموزش، فقط آنها را ذخیره کرده و زمان بیشتری را صرف دسته‌بندی می‌کند. هر یک از این روشها کاربرد خود را دارند که در ادامه به آنها اشاره خواهد شد. نزدیک‌ترین همسایگی[۵]، 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.

---

[۱] در این کتاب روشهای دسته‌بندی معادل Classifying Method در نظر گرفته شده و برای Classifier از واژه دسته بند استفاده شده است ومجازا به این معنی است که کار دسته بندی توسط دسته بند انجام شده است

[۲]- Eager

[۳]- Lazy

[۴]- Instance Based Learner

[۵]- K Nearest Neighborhood

- 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 (به‌عنوان نمونه‌های آموزشی) را نشان می‌دهد.

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

X_1 = مقاومت در برابر اسید (*seconds*)	X_2 = 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_1 و X_2 را به‌صورت زیر تعریف کرده‌ایم:

$$X_1 = (x_{1i}, x_{2i}, \ldots, x_{1n}) \quad , \quad X_2 = (x_{2i}, x_{2i}, \ldots x_{2n}) \qquad (۷-۵)$$

یعنی X_1 و X_2 به ترتیب دارای n ویژگی با مقادیر $x_{11} \ldots x_{1n}$ و $x_{21} \ldots x_{2n}$ هستند. برای محاسبه فاصله دو نمونه می‌توان از رابطه اقلیدسی استفاده کرد، تابع فاصلهٔ زیر این‌کار را انجام می‌دهد:

$$dist(X_1, X_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2} \qquad (۸-۵)$$

با محاسبه فاصله نمونه جدید (یعنی (۳,۷) که قرار است دسته‌بندی روی آن انجام شود) با نمونه‌های آموزشی، نتایج جدول (۵-٤) به‌دست می‌آید.

جدول ۵-٤) فاصله نمونه جدید (۳,۷) با تمام نمونه‌های آموزشی

x_1 = مقاومت (seconds)	x_2 = دوام (kg/square meter)	فاصله نمونه جدید از نمونه‌ها
۷	۷	٤
۷	٤	۵
۳	٤	۳
۱	٤	$\sqrt{۱۳}$

در گام دوم الگوریتم باید k همسایه نزدیک‌تر را انتخاب کند. با فرض $k = ۳$ جدول (۵-۵) محاسبه می‌کنیم.

جدول ۵- ۵) پیدا کردن همسایه نزدیک‌تر به نمونه جدید، در نمونه‌های آموزشی

جزء ۳ همسایه نزدیک هست؟	رتبه (فاصله‌اقلیدسی)	فاصله از نمونه جدید.	x_2 =دوام (kg/square meter)	x_1 = مقاومت (seconds)
بله	۳	٤	۷	۷
خیر	٤	۵	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 همسایه‌هاست به عنوان دستهٔ نمونهٔ جدید در نظر بگیرد.

با مراجعه به جدول (۵-٦) می‌بینیم که از بین نزدیک‌ترین همسایه‌ها (مشابه‌ترها به نمونه جدید) دو تا خوب و یکی بد است. بنابراین حدس می‌زنیم که نمونه جدید نیز در دسته خوب، که حائز اکثریت است، قرار بگیرد.

جدول۵- ٦) بررسی کلاس نزدیک‌ترین همسایه‌ها برای تخمین کلاس نمونه جدید

کلاس نزدیک‌ترین همسایه	جزء ۳ همسایه نزدیک هست؟	رتبه (فاصله اقلیدسی)	فاصله‌اقلیدسی با نمونه جدید	دوام= X_2 (kg/square meter)	مقاومت X_1 (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

بررسی دقیق‌تر روش نزدیک‌ترین همسایگی

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 عمومیِ 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 آنرا یک در نظر می‌گیریم. 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 v به مقدار v' در فاصله [۰،۱] است که با فرمول زیر انجام می‌شود:

$$v' = \frac{v - min_A}{max_A - min_A} \qquad (۹-۵)$$

قابل ذکر است در رابطه (۵-۸) مقادیر min_A و max_A (حداقل و حداکثر) روی مجموعه آموزشی محاسبه می‌شود.

- **ویژگی در یک (یا چند) 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 زیر 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 در فرمول(۵-۷)، این توابع به‌صورت جدول (۵-۸) تعریف می‌شوند.

جدول ۵- ۸) (a) تابع مانهاتان، (b) تابع اقلیدسی

(a)	(b)
$dist(x_1, x_2) = \sum_{i=1}^{n} \lvert x_{1i} - x_{2i} \rvert$	$dist(x_1, x_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2}$

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 فاصله

اقلیدسی 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 جدول (۵-۸) 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 هستند. برای ایجاد تمایز بین ویژگیها، تابع اقلیدسی را به‌صورت زیر دستکاری کرده و برای ویژگی i، وزن Wi را تعریف می‌کنیم.

$$Euclidean\ (X_1, X_2) = \sqrt{\sum_{i=1}^{n} w_i (x_{1i} - x_{2i})^2} \qquad (۱۰-۵)$$

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 گونه‌ای تغییر می‌دهیم 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 ایها[2] ارائه شده، با یک مقدار اولیه شروع کرده و پس از هر بار دسته‌بندی، اوزان را عوض می‌کند. جزئیات این روش در زیر توضیح داده شده است.

روش ایها (*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 (۵-۹) عوض می‌کنیم.

۱- Fitness

۲- 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 مشکلی یا گره‌ای 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، $k=C+1$ تا از نزدیک‌ترین همسایه‌ها را انتخاب کرد تا این مشکل پیش نیاید[۲]. 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 (۵-۱۱) محاسبه می‌شود. 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.

$$weight(X)=\frac{1}{dist(A,X)^2} \qquad (۵-۱۱)$$

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

۱- Combination

۲- طبق اصل لانه کبوتری

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[۱] مناسبی برخوردار باشد. برای رسیدن به این پوشش، روشهای زیادی وجود دارد. یکی از این روشها استفاده از دسته‌های مختلف های مختلف به تعداد برابر در مجموعه آموزشی است. روش دیگر انتخاب تصادفی است. در روش انتخاب تصادفی ممکن است وجود دسته‌های با فراوانی بالا موجب پوشش نامناسب شود. مثلاً در داده‌های مربوط به اعطا و بازپرداخت وام مطمئناً تعداد وامهای پرداخت نشده عدد اندکی است. حال اگر قصد ما تعیین وضعیت یک وام از نظر پرداخت یا عدم پرداخت باشد، با انتخاب تصادفی احتمال دارد هیچ داده مربوط به عدم پرداخت در مجموعه آموزشی قرار نگیرید

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 $IB3$[۲] است. این الگوریتم نسخه سوم از الگوریتمهای پنج‌گانه، $IB1$ تا $IB5$ است که روش نزدیک‌ترین

۱- Coverage

۲- Instance Based Learner Version 3

همسایگی را تکمیل کرده‌اند. ایده اصلی این الگوریتمها این است که: «فقط نمونه‌هایی را که کارآیی خوبی برای دسته‌بندی داشته‌اند در مجموعه آموزشی نگه دارند.»

الگوریتم *IB۳*

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 که کارآیی خوبی داشته 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 IB۳

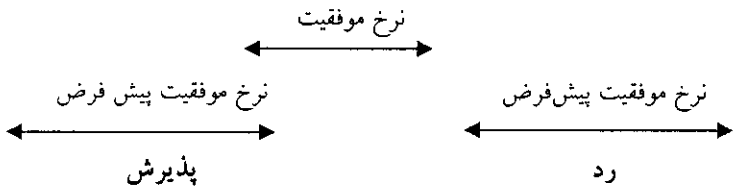
افزودن و حذف عناصر از S با توجه به مفاهیم نرخ موفقیت نمونه و نرخ موفقیت پیش فرض آن صورت می‌گیرد. نرخ موفقیت نمونه این‌گونه تعریف می‌شود:

فرض کنید که نمونه‌ای N بار(از زمان ورود به S) برای دسته‌بندی انتخاب شده و متغیر تصادفی f بیانگر دقت دسته بندی باشد. با قرار دادن این مقادیر در فرمول زیر و داشتن مقادیر اطمینان می‌توان نرخ موفقیت p را حساب کرد.

$$p=\left(f+\frac{z^2}{2N}\pm z\sqrt{\frac{f}{N}-\frac{f^2}{N}+\frac{z^2}{4N^2}}\right)\Big/\left(1+\frac{z^2}{N}\right) \qquad (۵-۱۲)$$

در رابطه (۵-۱۲) مقدار z از جدول مربوط به توزیع نرمال به‌دست می‌آید. در واقع اگر متغیر تصادفی f را 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 نمونه، f را 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 $K = ۱$ باشد، دسته‌بندی نمونه جدید از مرتبه زمانی $O(D)$ یعنی 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 نزدیک‌ترین 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 ساختْ 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;
   }
}
```

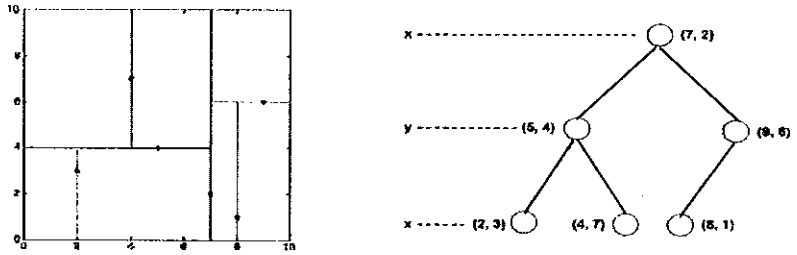
شکل ۵-۹) الگوریتم ساخت k-Dtree

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.

۵-۴- 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 کارآ 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 نسبی 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 می‌شود، کاربرد شبکه‌های عصبی در دسته‌بندی تشریح گردد. تاکنون تحقیقاتی بسیار در زمینه ساختار مغز انسان صورت پذیرفته ولی هنوز سؤالات فراوانی 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 دیگر سیناپس نام دارد. همچنین هر نرون بر اساس یک آستانه تحریک در یکی از دو وضعیت تحریک شده[۴] و ساکن[۵] قرار می‌گیرند.

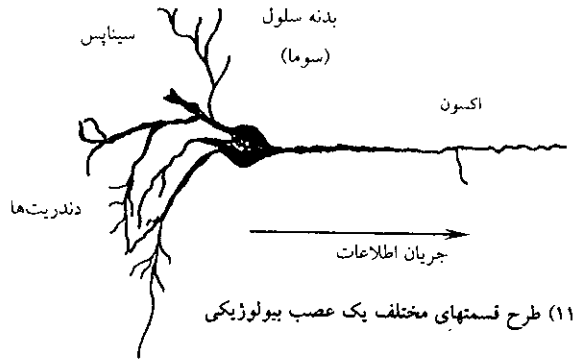
۱- Artificial Neural Networks

۲- Neural Network Data Mining

۳- Neuron

۴- Firing

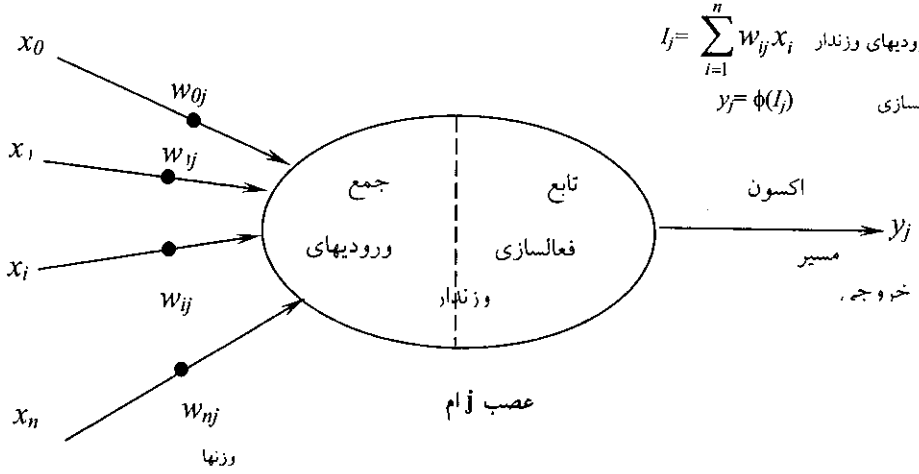
۵- Rest



شکل ۵-۱۱) طرح قسمتهای مختلف یک عصب بیولوژیکی

این ساختار نرون در مغز انسان به تعداد $۱۰^{۱۱}$ تکرار می‌شود و از آنجا که هر نرون حداقل به ۱۰۰۰۰ نرون دیگر متصل می‌باشد، در مغز انسان $۱۰^{۱۵}$ 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.



شکل ۵-۱۲) مدل ریاضی پیشنهادی برای شبکه‌های عصبی مصنوعی

با به استعاره گرفتن ساختار شبکه‌های عصبی زنده، مدل ریاضی شبکه‌های عصبی مصنوعی ارائه شد. در شکل (۵-۱۲) هر شبکه عصبی مصنوعی، مجموعه‌ای از ورودی‌هاست و براساس یک تابع فعال‌سازی مقدار خروجی آن محاسبه می‌شود. مشابهت‌هایی میان مفاهیم شبکه‌های عصبی زنده و مصنوعی وجود دارد که در شکل (۵-۱۳) آمده است.

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

درعین‌حال 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 دارد.
- به دلیل کارکرد موازی نسبت به سایر روشها سرعت بالاتری دارد.

معایب شبکه‌های عصبی

- آموزش این شبکه‌ها بسیار حساس است.

- مانند یک جعبه سیاه عمل می‌کنند[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 کاربردهای فراوانی دارند.

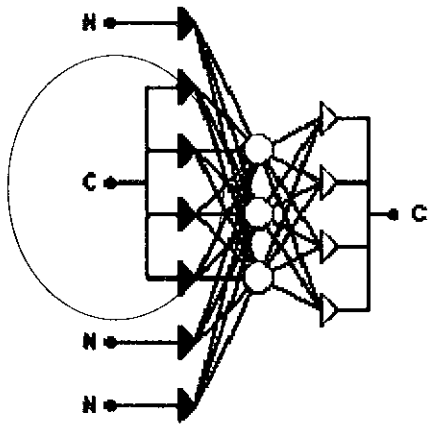
۵-۴-۱- تبدیلات ورودی و خروجی

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:

$$X^* = \frac{X - min(X)}{range(X)} = \frac{X - min(X)}{max(X) - min(X)} \qquad (۱۳-۵)$$

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 متغیرهای 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

[1]- Black Box

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 کمّی 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:

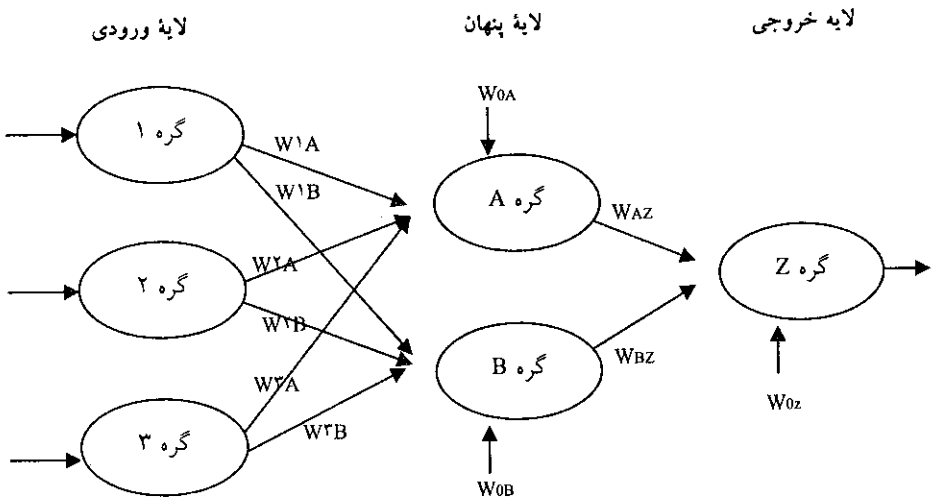
Prediction = output (data range) + minimum

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،

---

[1]- Fully Connected

پنهان و خروجی تشکیل شده لذا 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.



$W_{OZ} = ۰/۵$	$W_{OB} = ۰/۷$	$W_{OA} = ۰/۵$	$x_۰ = ۱/۰$
$W_{AZ} = ۰/۹$	$W_{۱B} = ۰/۹$	$W_{۱A} = ۰/۶$	$x_۱ = ۰/۴$
$W_{BZ} = ۰/۹$	$W_{۲B} = ۰/۸$	$W_{۲A} = ۰/۸$	$x_۲ = ۰/۲$
$W_{۳B} = ۰/۴$	$W_{۳A} = ۰/۶$	$x_۳ = ۰/۷$	

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.

$$net_j = \sum_i W_{ij}\, x_{ij} = W_{oj}\, x_{oj} + W_{1j} x_{1j} + \ldots + W_{nj} x_{nj} \qquad (۱۴-۵)$$

مقدار x_{oj} 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.

$$net_A = \sum_i W_{iA} x_{iA} = W_{OA}(1) + W_{1A} x_{1A} + W_{2A} x_{2A} + W_{3A}\, x_{3A}$$

$$= ۰.۵ + ۰.۶(۰.۴) + ۰.۸(۰.۲) + ۰.۶(۰.۷) = ۱.۳۲$$

$$net_B = \sum_i W_{iB} x_{iB} = W_{OB}(1) + W_{1B} x_{1B} + W_{2B} x_{2B} + W_{3B} x_{3B}$$

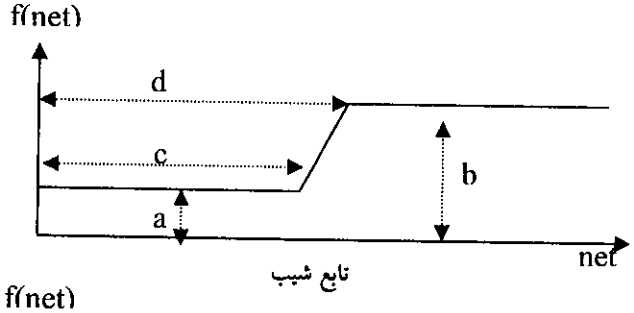
$$= ۰.۷ + ۰.۹\,(۰.۴) + ۰.۸(۰.۲) + ۰.۴(۰.۷) = ۱.۵$$

۵-۴-۲- توابع فعال سازی[1]

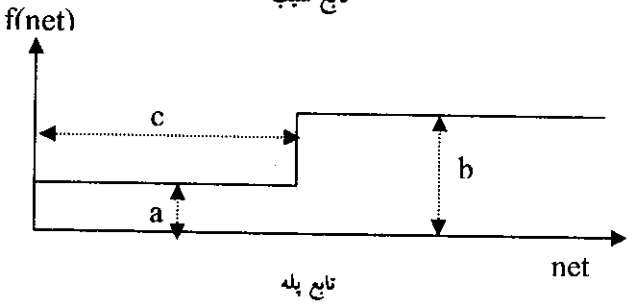
همان‌طورکه در مقدمه بیان شد، یک نرون دو حالت ساکن و فعال دارد که معمولاً بر اساس یک آستانه تحریک ایجاد می‌شود. در واقع می‌توان چنین بیان نمود که ورودیهای یک نرون در داخل یک تابع قرار می‌گیرند و بر اساس مقدار ترکیب ورودیها یا نرون تحریک می‌شود و یا تحریک نمی‌شود. این مفهوم در شبکه‌های عصبی مصنوعی به نام تابع فعال‌سازی شناخته می‌شود که انواع متفاوت ذیل را می‌توان برای آن در نظر گرفت.

- تابع مشخصات
- تابع ثابت
- تابع پله (آستانه)

$$f(net) = \begin{cases} a & if\ net < c \\ b & if\ net > c \end{cases}$$



تابع شیب



تابع پله

$$f(net) = \begin{cases} a & if\ net \le c \\ b & if\ net \ge d \\ a + \dfrac{(net - c)(b - a)}{(d - c)} & otherwise \end{cases}$$

شکل ۵-۱۶) توابع شیب و پله

[1]- Activation Function

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 متغیر x تابع $y=\frac{۱}{۱+e^{-x}}$ قرار می‌گیرد و خروجی گره را ایجاد می‌کند. مقدار net_A را به جای x قرار داده و مقدار خروجی گره A محاسبه شده 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:

$$f(net_B)=\frac{۱}{۱+e^{-۱.۵}}=۰.۸۱۷۶$$

$$net_z=\sum_i W_{iz}x_{iz}=W_{oz}(۱)+W_{AZ}x_{AZ}+W_{BZ}x_{BZ}$$

$$=۰.۵+۰.۹(۰.۷۸۹۲)+۰.۹(۰.۸۱۷۶)=۱.۹۴۶۱$$

$$f(net_z)=\frac{۱}{۱+e^{-۱.۹۴۶۱}}=۰.۸۷۵۰$$

۵-۴-۳- الگوریتم پس انتشار خطا

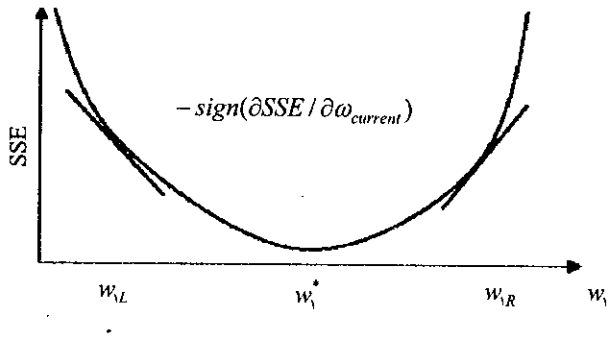
چنانچه مثال قبل را دنبال کرده باشید، تاکنون هیچ‌گونه فعالیت یادگیری توسط شبکه صورت نپذیرفته است و این درحالی است که فلسفه وجودی شبکه‌های عصبی مصنوعی، یادگیری یک وظفیه مانند تشخیص الگو، دسته‌بندی و یا پیش‌بینی می‌باشد. به‌همین دلیل الگوریتمها و روشهای بسیاری ابداع گردیده است تا شبکه‌ها بتوانند یاد بگیرند. 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:

$$SSE = \sum_{records} \sum_{output\ nodes} \text{(مقدار تخمینی - مقدار واقعی)}^{\wedge}۲ \qquad (۵-۱۵)$$

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.

۵-۴-۴- 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.



شکل ۵-۱۷) استفاده از شیب تابع خطا برای جهت تصحیح اوزان

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 افزایشW باشد، پس جهت تغییر اوزان همواره مخالف گرادیان می‌باشد.

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 و تصحیح بیشتری با استفاده از

---

[1]- Sum Square Error

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:

$$\Delta w_{current} = -\eta(\partial SSE / \partial w_{current}) \qquad (۱۶-۵)$$

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:

$$w_{ij.new} = w_{ij.current} + \Delta w_{ij} \qquad where \qquad \Delta w_{ij} = \eta \delta_j \qquad (۱۷-۵)$$

$$\delta_j = \begin{cases} output_j(1-output_j)(actual_j - output_j) & for\ output\ layer\ nodes \\ output_j(1-output_j)\sum_{downstream} W_{jk}\delta_j & for\ hidden\ layer\ nodes \end{cases} \qquad (۱۸-۵)$$

در این قواعد δ نشان دهنده مسئولیت هر گره در تولید خطای شبکه می‌باشد. حال برای فهم قواعد، مثال را ادامه می‌دهیم. همان‌طورکه 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:

$$\delta_z = output_z(1-output_z)(actual_z - output_z)$$
$$= ۰/۸۷۵(۱-۰/۸۷۵)(۰/۸-۰/۸۷۵) = -۰/۰۰۸۲$$
$$\Delta W_{oz} = \eta\delta_z(۱) = ۰/۱(-۰/۰۰۸۲)(۱) = -۰/۰۰۰۸۲$$
$$w_{oz,new} = w_{oz,curent} + \Delta w_{oz} = ۰/۵ - ۰/۰۰۰۸۲ = ۰/۴۹۹۱۸$$

همان‌طورکه به خاطر دارید، $W_{.z}$ دارای مقدار ۰/۵ بود و 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:

$$\delta_A = output_A(1-output_A)\sum_{downstream} W_{jk}\,\delta_j$$

$$\delta_A = ۰/۷۸۹۲(۱-۰/۷۸۹۲)(۰/۹)(-۰/۰۰۸۲) = -۰/۰۰۱۲۳$$

$$\Delta W_{AZ} = \eta\delta_z.output_A = ۰/۱(-۰/۰۰۸۲)(۰/۷۸۹۲) = -۰/۰۰۰٦٤۷$$

$$w_{AZ,new} = w_{AZ,current} + \Delta w_{AZ} = ۰/۹ - ۰/۰۰۰٦٤۷ = ۰/۸۹۹۳۵۳$$

$$\delta_B = output_B(1-output_B)\sum_{downstream} W_{jk}\delta_j$$

$$\Delta W_{BZ} = \eta\delta_z.output_B = ۰/۱(-۰/۰۰۸۲)(۰/۸۱۷٦) = -۰/۰۰۰٦۷$$

$$w_{BZ,new} = W_{BZ,current} + \Delta w_{BZ} = ۰/۹ - ۰/۰۰۰٦۷ = ۰/۸۹۹۳۳$$

$$\Delta W_{۱A} = \eta\delta_A x_۱ = ۰/۱(-۰/۰۰۱۲۳)(۰/٤) = -۰/۰۰۰۰٤۹۲$$

$$w_{۱A,new} = w_{۱A,current} + \Delta w_{۱A} = ۰/٦ - ۰/۰۰۰۰٤۹۲ = ۰/۵۹۹۹۵۰۸$$

$$\Delta W_{۲A} = \eta\delta_A x_۲ = ۰/۱(-۰/۰۰۱۲۳)(۰/۲) = -۰/۰۰۰۰۲٤٦$$

$$w_{۲A,new} = w_{۲A.current} + \Delta_{۲A} = ۰/۸ - ۰/۰۰۰۰۲٤٦ = ۰/۷۹۹۹۷۵٤$$

$$\Delta W_{۳A} = \eta\delta_A x_۳ = ۰/۱\ (-۰/۰۰۱۲۳)(۰/۷) = -۰/۰۰۰۰۸٦۱$$

$$w_{۳A,new} = w_{۳A,current} + \Delta w_{۳A} = ۰/٦ - ۰/۰۰۰۰۸٦۱ = ۰/۵۹۹۹۱۳۹$$

این فرایند را برای تمام اوزان انجام داده تا تمام اوزان به‌روز 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 تعداد تکرارهای از پیش تعیین شده
- 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. الگوریتم یادگیری پس‌انتشارخطا که در فوق توضیح داده شد، تنها یکی از چندین الگوریتم یادگیری شبکه‌های عصبی است. برای آشنایی با سایر روشها به منابع مرتبط مراجعه نمایید.

۵-۴-۵- برخی کاربردهای دسته‌بندی بر اساس شبکه‌های عصبی

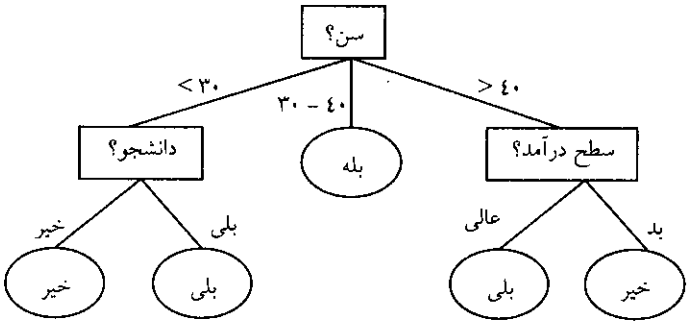
الگوریتمها و ساختارهای مختلف شبکه‌های عصبی هر کدام می‌توانند کاربردهای زیادی را در عملیات‌های مختلف داشته باشند. به منظور درک صحیح‌تر، این حوزه‌ها مثال‌های زیر ارائه می‌شوند:

- 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 عملیات تخمین و پیش‌بینی داده‌کاوی است. 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.

۵-۵- درخت تصمیم

ساختار درخت تصمیم یک ساختار درختی، شبیه فلوچارت است. بالاترین گره در درخت، گرهٔ ریشه است و گره‌های برگ، دسته‌ها یا توزیع دسته‌ها را نشان می‌دهند. 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؟ 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 یعنی در این مثال هر گره داخلی (غیر برگ) آزمایش ویژگی سن دانشجو و سطح درآمد را نمایش می‌دهد و هر گره برگ دسته بلی یا خیر را نشان می‌دهد، که نشانگر خریدن (*yes*) یا نخریدن (*no*) کامپیوتر است.

۵-۵-۱- 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. ممکن است شرکت دیگری که قصد بازاریابی و کسب مشتریان جدید را دارد، علاقه‌مند باشد تا بداند ویژگیهای مشتریانی که احتمالاً به محصول جدید 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 ۲ 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.

۵-۵-۲- 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 پرسش 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 راه ایجاد شده از ریشه به یک برگ، معادل یک قاعده است و البته بعضی از این قواعد از دیگر قواعد قویتر می‌باشند. گاهی 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

در مسائل مرتبط با درختهای تصمیم با دو نوع از متغیرها مواجه هستیم [۱]:

- متغیرهای عددی مثل مشخصه «سن» که مقادیر آن عددی است.
- متغیرهای طبقه‌ای 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» که متغیرهایی مستقل هستند، متغیر پیش‌بینی کننده گویند و به متغیرهای وابسته، برچسب دسته[۱] می‌گویند. در مثال بالا متغیر «ریسک تصادف» 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 طبقه‌ای 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 طبقه‌ای به ازای هر مقدار می‌توان یک شاخه در نظر گرفت اما تمرکز ما در این بخش بر روی درختان دودویی می‌باشد.

۵-۵-۳- مفاهیم اصلی در درختهای تصمیم

گره: به متغیر مستقلی که آزمون روی آن صورت می‌گیرد، گره گفته می‌شود.

گره ریشه: گره ای که در بالاترین نقطه درخت وجود دارد.

برگ: به متغیر وابسته‌ای 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 طبقه‌ای به‌صورت $q_n : X_n \subseteq x_n$ انجام می‌گیرد.

نکات قابل توجه برای استفاده از الگوریتمهای درخت تصمیم

برای استفاده از روشهای مربوط به درختهای تصمیم اطلاعات و شرایط زیر باید فراهم باشند.

۱- Class Label

- توضیحات ویژگی – ارزش[۱]: 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[۲]: از آنجاییکه در روشهای دسته‌بندی، برچسب دسته‌ها از قبل مشخص می‌شوند و درخت تصمیم نیز نوعی روش دسته‌بندی است لذا نام دسته‌ها از قبل مشخص می‌باشد.
- دسته‌های گسسته[۳]: 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.
- دادهٔ کافی[۴]: 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. یکی از روشهای معمول برای ایجاد درخت انتخاب معیاری برای انشعاب گره‌های بالایی به تعدادی زیر گره می‌باشد.

۱- Attribute Value Description

۲- Predefined Classes

۳- Discrete Classes

۴- Sufficient Data

در این کتاب فرض میشود هر گره تنها به دو گره پایین تر شکسته میشود که اصطلاحا به آن درخت دودویی گویند. الگوریتم مراحل ایجاد یک درخت دودویی در شکل(۵-۱۹) نشان داده شده است.

```
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

همانگونه که ذکر شد انتخاب نقطه شکست و ایجاد انشعاب در درخت از اهمیت خاصی برخوردار است که در ادامه به آن می‌پردازیم.

روشهای انتخاب نقطه انشعاب[۱]

در این قسمت روشهای مبتنی بر ناخالصی[۲] 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 یکی از دسته‌ها تبدیل شود اولویت دارد. (برای راحتی از (I) استفاده می‌کنیم)

انواع روشهای انتخاب نقطه شکست عبارتند از:

- شاخص جینی[۳] : $gini(T) = ۱ - \sum P_i^۲$
- آنتروپی[٤] : $Entropy(T) = -\sum P_i.\log_۲^{P_i}$
- کارت[۱]

۱- Split Selection

۲- Impurity - Based

۳- Gini Index

٤- Entropy

- $2P_i$ (P_i فراوانی نسبی از کلاس j در درخت T می‌باشد.)
- $Min(P_i)$
- $C_{4.5}$

در روش شاخص جینی همه متغیرهای گره‌ها را امتحان کرده و آن متغیری که از همه بهتر باشد را برگی‌گزینیم. حال بهترین انتخاب برای تقسیم مجموعه S به دو مجموعه S_1 و S_2 از معیار زیر تبعیت می‌کند یعنی حداقل کردن تابع زیر:

$$I(S) = \frac{|S_1|}{|S|}.I(S_1) + \frac{|S_2|}{|S|}.I(S_2) \qquad (۱۹-۵)$$

۵-۵-۴- ساخت یک نمونه درخت تصمیم با استفاده از روش شاخص جینی

مثال: درخت تصمیم را برای مجموعه داده‌های جدول (۵-۱۱) رسم کنید.

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 در جدول(۵-۱۲) مرتب می‌کنیم. حال از روش شاخص جینی برای انتخاب نقطهٔ انشعاب استفاده می‌کنیم. هر دو متغیر «سن» و «نوع ماشین» را بررسی می‌کنیم. توجه کنید که هر دو متغیر «سن» و «نوع ماشین» را به موازات هم مورد بررسی قرار می‌دهیم

۱- CART

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

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.

اجرای اول:

در مرحله نخست تمام داده‌ها برای پیدا کردن نقطه انشعاب اول از گره ریشه و ترسیم سطح نخست درخت مورد ارزیابی قرار میگیرند.

۱- حالت ۱۷ = > «سن»

پس از اینکه جدول داده‌ها را بر اساس متغیر«سن» به صورت صعودی مرتب می‌کنیم، برای هر مقدار از متغیرها جدولی را مطابق زیر تشکیل می‌دهیم. به‌عنوان مثال خانه چپ در سطر اول 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 P_j^L نشان دهنده مقدار فراوانی در خانه‌های چپ و P_j^R نشان دهنده مقدار فراوانی در خانه‌های راست باشد. به علاوه 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 طبق شاخص جینی فرمولهای زیر را خواهیم داشت.

حال براساس فرمول $I(S_1): 1-\sum(P_j^L)^2$ و $I(S_2): 1-\sum(P_j^R)^2$ و سپس بر اساس رابطه(۵-۱۸) مقدار را محاسبه می‌کنیم.

پس از محاسبه $I(S)$ برای تمام نقاط انشعاب آن نقطه ای را بر می‌گزینیم که دارای کمترین مقدار $I(S)$ می‌باشد. فراموش نکنید که باید این مقدار هم برای متغیر سن و هم برای متغیر نوع خودرو مقایسه شود. یعنی حداقل یابی پس از محاسبه $I(S)$ های مربوط به هر دو متغیر صورت می‌گیرد.(چرا؟)

با توجه به مطالب فوق داریم:

	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- حالت ۲۰ = > «سن». در این حالت داریم:

	H	L
L	۲	۰
R	۲	۲

$I(S_1): ۱-(۲/۲)^۲-(۰/۲)^۲ =۱-۱-۰=۰$

$I(S_2): ۱-(۲/۴)^۲-(۲/۴)^۲ =۱-۴/۱۶-۴/۱۶=۰/۵$

$|s_1|=۲ , |s_2|=٤ , |s|=٦ \Rightarrow I(s)=|۲|/|٦|*۰+|٤|/|٦|*۰/٥$
$=٤/٦*۰/٥=۰/۳۳$

۳- حالت ۲۳ = > «سن».

در این حالت داریم:

	H	L
L	۳	۰
R	۱	۲

$I(S_1): ۱-(۳/۳)^۲-(۰/۳)^۲=۱-۱-۰=۰$
$I(S_2): ۱-(۱/۳)^۲-(۲/۳)^۲=۱-۱/۹-٤/۹=۰/٤٤٤$
$|s_1|=۳ , |s_2|=۳ , |s|=٦ \Rightarrow I(s):|۳|/|٦|*۰+|۳|/|٦|*۰/٤٤٤$
$=۳/٦*۰/٤٤٤=۰/۲۲۲$

٤- ۳۲ = > «سن»

	H	L
L	۳	۱
R	۱	۱

$I(S_1): ۱-(۳/٤)^۲-(۱/٤)^۲=۱-۹/۱٦-۱/۱٦=۰/۳۷٥$
$I(S_2): ۱-(۱/۲)^۲-(۱/۲)^۲=۱-۱/٤-۱/٤=۰/٥$
$|s_1|=٤ , |s_2|=۲ , |s|=٦ \Rightarrow I(s):|٤|/|٦|*۰/۳۷٥+|۲|/|٦|*۰/٥$
$=٤/٦*۰/۳۷٥+۲/٦*۰/٥=۰/٤۱٦٦$

٥- ٤۳ = > «سن»

	H	L
L	٤	۱
R	۰	۱

$I(S_1): ۱-(٤/٥)^۲-(۱/٥)^۲=۱-۱٦/۲٥-۱/۲٥=۰.۳۲$
$I(S_2): ۱-(۰/۱)^۲-(۱/۱)^۲=۱-۰-۱=۰$

$|s_1|=۵, |s_2|=۱, |s|=٦ \Rightarrow I(s):|۵|/|٦|*۰/۳۲+|۱|/|٦|*۰$
$=۵/٦*۰/۳۲+۰=۰/۲٦٦$

«سن» = < ٦٨ -٦

	H	L
L	٤	۲
R	۰	۰

$I(S_1):۱-(٤/٦)^۲-(۲/٦)^۲=۱-۱٦/۳٦-٤/۳٦=۰.٤٤٤$
$I(S_2):۱-۰-۰=۱$
$|s_1|=٦, |s_2|=۰, |s|=٦ \Rightarrow I(s);|٦|/|٦|*۰/۳۲+|۰|/|٦|*۱$
$=٦/٦*۰/٤٤٤+۰=۰/٤٤٤$

حال محاسبات مشابهی را برای متغیر «نوع خودرو» انجام می‌دهیم. توجه کنید که این متغیر غیر عددی است. 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 خودرو» 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 (۵-۱۳) به‌عنوان نتیجه حاصل می‌شود.

جدول ۵-۱۳) جدول نتیجه درخت تصمیم

نوع خودرو	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 خودرو» 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- حالت اسپورت = «نوع خودرو»

	H	L
اسپورت = نوع ماشین	۲	۰
اسپورت ≠ نوع ماشین	۲	۲

$$I(S_1): ۱-(۲/۲)^۲-(۰/۲)^۲=۱-۱-۰=۰$$
$$I(S_2): \ ۱-(۲/٤)^۲-(۲/٤)^۲=۱-۱٦/٤-۱٦/٤=۰.۵$$
$$|s_1|=۲\ ,\ |s_2|=۴\ ,\ |s|=۶\Rightarrow I(s)=|۲|/|۶|*۰+۴/۶*۰/۵=۰/۳۳۳$$

۸- حالت 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- حالت باری = «نوع ماشین»

	H	L
باری = نوع ماشین	۰	۱
باری ≠ نوع ماشین	٤	۱

$$I(S_1):۱-(۰/۱)^۲-(۱/۱)^۲=۱-۰-۱=۰$$
$$I(S_2):\ ۱-۱٦/۲۵-۱/۲۵=۰/۳۲$$
$$|s_1|=۱\ ,|s_2|=۵\ ,\ |s|=٦\Rightarrow I(s);|۱|/|٦|*۰+|۵|/|٦|*۰/۳۲$$
$$=۵/٦*۰/۳۲+۰=۰/۲٦٦$$

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.

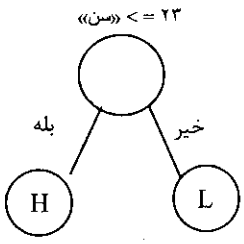
$Age <= ۲۳ = \{۱۷ , ۲۰ , ۲۳\}$

داده‌های مطابق با این شرط در جدول (۵-۱٤) نمایش داده شده‌اند.

جدول ۵- ۱٤) داده‌های مورد استفاده در شرط Age <=۲۳

ریسک	نوع ماشین	سن
زیاد	اسپورت	۱۷
زیاد	اسپورت	۲۰
زیاد	خانوادگی	۲۳

چون برچسب دستهٔ این مجموعه همه «زیاد» می‌باشد درختی به شکل (۵-۲۰) زیر ایجاد می‌شود.



شکل ۵-۲۰) دسته‌بندی ایجاد شده در مرحله اول

همانگونه که می‌بینید سمت چپ درخت ما به گره انتهایی خود رسیده است و همگی دارای برچسب ریسک زیاد هستند. اما سمت راست این درخت وضعیت متفاوتی دارد که در جدول (۵-۱۵) نشان داده شده است.

در واقع این جدول را بر اساس سمت راست درخت یعنی « ۲۳ < سن » تشکیل می‌دهیم تا معیار انشعاب بعدی با استفاده از همان روش فوق مجددا برای این بخش از داده‌ها نیز انتخاب شود.

جدول ۵- ۱۵) داده‌های مورد استفاده در شرط Age> =۲۳

ریسک	نوع ماشین	سن
کم	باری	۳۲
زیاد	اسپورت	۴۳
کم	خانوادگی	۶۸

اجرای دوم:

در این مرحله داده‌های جدول (۵-۱۵) برای پیدا کردن نقطه انشعاب دوم و ترسیم سطح بعدی درخت مورد ارزیابی قرار می‌گیرند.

۱- حالت (اسپورت = «نوع ماشین») . در این حالت داریم:

	H	L
اسپورت = نوع ماشین	۱	۰
اسپورت ≠ نوع ماشین	۰	۲

$$I(S_1): ۱-(۱/۱)^۲-(۰/۱)^۲=۱-۰-۱=۰$$
$$I(S_2): ۱-(۰/۲)^۲-(۲/۲)^۲=۱-۰-۱=۰$$
$$|s_1|=۱, |s_2|=۲, |s|=۳ \Rightarrow I(s):|۱|/|۳|*۰+|۲|/|۳|*۰=۰$$

۲- حالت (باری = «نوع ماشین») در این حالت داریم:

	H	L
باری = نوع ماشین	۰	۱
باری ≠ نوع ماشین	۱	۱

$$I(S_1): ۱-(۰/۱)^۲-(۱/۱)^۲=۱-۰-۱=۰$$
$$I(S_2): ۱-(۱/۲)^۲-(۱/۲)^۲=۱-۱/۴-۱/۴=۰.۵$$
$$|s_1|=۱, |s_2|=۲, |s|=۳ \Rightarrow I(s):|۱|/|۳|*۰+|۴|/|۳|*۰/۵=۰/۳۳۳$$

۳- حالت (خانوادگی = «نوع ماشین») در این حالت داریم:

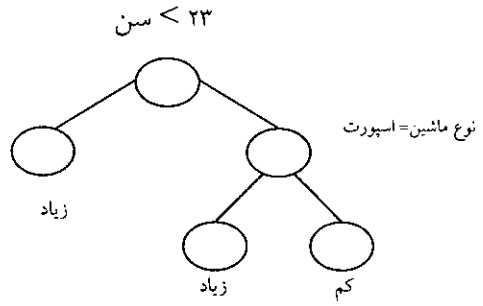
	H	L
خانوادگی = نوع ماشین	۰	۱
خانوادگی ≠ نوع ماشین	۱	۱

$I(S_1): \ ۱-(۰/۱)^۲-(۱/۱)^۲ = ۱-۰-۱=۰$

$I(S_2): \ ۱-(۱/۲)^۲-(۱/۲)^۲ = ۱-۱/٤-۱/٤=۰/۵$

$|s_1|=۱, |s_2|=۲, |s|=۳ \Rightarrow I(s):|۱|/|۳|*۰+|۲|/|۳|*۰/۵=۰/۳۳۳$

پس از بررسی تمام حالات در اجرای دوم درمی‌یابیم که حداقل *I(S)* برابر صفر می‌باشد پس درخت به شکل زیر تکمیل می شود. 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[۱] 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 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.

۱- Diversity

۵-۵-۵- 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 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

---

[1]- Pruning

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 $CHAID$[1] 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 از بین رفتن بخشهایی از درخت (زیردرختها) که در دقت و صحت دسته‌بندیِ 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 خلاصه‌سازی در داده‌های آموزشی، نابهنجاری‌هایی را در مدل منعکس می‌کنند. روشهای هرس‌کردن درختها به مشکل بیش‌برازش[2] 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.

---

[1] Chi-square Automatic Intraction Detection

[2]- Overfitting

۵-۵-۶- استخراج قواعد دسته‌بندی از درختهای تصمیم

آیا امکان به‌دست آوردن قواعد از درخت تصمیم وجود دارد؟ دانش نمایش داده شده در درختهای تصمیم 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. درک قواعد «اگر- آنگاه» ساده‌تر است به‌خصوص اگر درخت مفروض بسیار بزرگ باشد. در اینجا به بررسی استخراج قواعد به‌دست آمده از شکل (۵-۱۸) بررسی 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

IF "سن" = "< ۳۰" ، دانشجو = بله THEN کامپیوتر می‌خرد

IF "سن" = "۳۰ - ۴۰" THEN کامپیوتر می‌خرد

IF سن = "> ۴۰"، درآمد = عالی THEN کامپیوتر می‌خرد

IF "سن" = "> ۴۰"، درآمد = بد THEN کامپیوتر نمی‌خرد

نقاط قوت درخت تصمیم

درخت تصمیم به ما این توانایی را می‌دهد که پیش‌بینیهای خود را در قالب یک سری قواعد قابل فهم ارائه کنیم. این روش نیاز به محاسبات پیچیده‌ای برای دسته‌بندی داده‌ها ندارد. درخت تصمیم برای انواع مختلف داده‌ها از قبیل داده‌های عددی و طبقه‌ای قابل استفاده می‌باشد. دقت این روش با سایر روشهای دسته‌بندی قابل رقابت است.

این روش نشان می‌دهد که کدام فیلد یا متغیرها تأثیرات مهمی در پیش‌بینی و دسته‌بندی دارند. هر چه متغیر به ریشه نزدیکتر باشد اهمیت آن بیشتر است. از این خاصیت می‌توان در

انتخاب مشخصه استفاده کرد (رجوع شود به فصل آماده سازی داده‌ها بخش انتخاب زیرمجموعه مشخصه‌ها)

نقاط ضعف درخت تصمیم

بعضی از روشهای درخت تصمیم تنها می‌توانند روی متغیرهای هدف دودویی (بله یا خیر- پذیرش یا عدم پذیرش) دسته‌بندی و پیش‌بینی انجام دهند. در برخی روشها هنگامی که تعداد مثالها یا رکوردهای هر دسته کم باشد، نرخ خطا بالا می‌رود. برخی الگوریتمهای ایجاد درخت به حافظه زیادی نیاز دارند زیرا برای پیدا کردن بهترین ویژگی، وضعیت هر ویژگی نگهداری می‌شود که این عملیات نیاز به حافظه زیادی دارد. همچنین در قسمت هرس شاخه‌ها نیز، برای انتخاب بهترین زیر درختی برای برش، وضعیت هر زیرشاخه را باید به‌خاطر سپرد. اکثر الگوریتمهای درخت تصمیم در هر گره تنها یک فیلد را برای شاخه زدن در نظر می‌گیرند.

۵-۶- پیش‌بینی

پیش‌بینی[۱]، عبارت است از تعیین مقدار یک متغیر پاسخ پیوسته (متغیر وابسته) برحسب مقادیر متغیرهای مستقل. پیش‌بینی مشابه دسته‌بندی است با این تفاوت که متغیر وابسته در دسته‌بندی، گسسته می‌باشد. برآورد حقوق فارغ التحصیلان با ۱۰ سال تجربه کاری یا فروش بالقوه یک محصول جدید بر حسب قیمت آن، مواردی از پیش‌بینی می‌باشند. مهم‌ترین روش مورد استفاده در پیش‌بینیِ عددی، رگرسیون است.

البته برخی دیگر از روشهای دسته‌بندی نظیر الگوریتم پس‌انتشار و ماشینهای بردار پشتیبان نیز می‌توانند به‌عنوان روشهای پیش‌بینی مورد استفاده قرار گیرند. در داده‌کاوی، متغیرهای مستقل و متغیر وابسته همان ویژگیهای تشریح شده برای هر نمونه یا مشاهده می‌باشند. معمولاً مقادیر متغیرهای مستقل معلوم است. هر چند با استفاده از روشهای خاصی، می‌توان مواردی‌که در آنها بعضی از مقادیرمفقوده را نیز پیش‌بینی کرد. 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 به شکل $y = w_0 + w_1 x$ 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. پس از تعیین مقادیر w_0 و w_1 در مشاهدات جدید با جایگذاری مقدار متغیر مستقل x در رابطه $y = w_0 + w_1 x$ می‌توان، مقدار متناظر متغیر وابسته y را پیش‌بینی نمود.

$$w_1 = \frac{\sum_{i=1}^{|D|}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{|D|}(x_i - \bar{x})^2} \qquad (۲۰-۵)$$

۱- Prediction

$$w_0 = \bar{y} - w_1 \bar{x} \qquad (۲۱-۵)$$

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 آنها y_i است. در این صورت معادله رگرسیونی به شکل رابطه (۵-۲۲) خواهد بود. این معادله با استفاده از روش حداقل مربعات قابل حل است.

$$y = w_0 + w_1 x_1 + w_2 x_2 \qquad (۲۲-۵)$$

رگرسیون غیرخطی

اگر داده‌ها، دارای وابستگی خطی نباشند (مثلاً اگر وابستگیها به‌صورت یک تابع چندجمله‌ای باشد)، چگونه می‌توان از مدل رگرسیون خطی استفاده کرد؟ در برخی از این حالات با استفاده از روشهای تبدیل وتغییر متغیر می‌توان مدل غیرخطی را به یک مدل رگرسیون خطی تبدیل و براساس روش حداقل مربعات مسئله را حل کرد.

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.

$$y = w_0 = w_1 x + w_2 x^2 + w_3 x^3 \qquad (۲۳-۵)$$

$$x_1 = x \qquad x_2 = x^2 \qquad x_3 = x^3$$

$$y = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 \qquad (۲٤-۵)$$

البته برخی از مدلهای غیرخطی با استفاده از تغییر متغیر، به راحتی قابل تبدیل به شکل خطی نیستند. در چنین مواردی نیز ممکن است تخمینهای حداقل مربعات را بر اساس محاسبات پیچیده‌ای به‌دست آوریم. قبل از به‌کار بردن تحلیل رگرسیون، بهتر است ویژگیهایی که پیش‌بینی کننده خوبی برای Y نیستند را حذف کرده و از بقیه ویژگیها استفاده کنیم (این مطلب با تفصیل بیشتر در بحث آماده‌سازی داده‌ها عنوان شده است). تحلیل رگرسیون یک روش دقیق و مناسب برای پیش‌بینی است. البته بهتر است داده‌های پرت و مغشوش قبل از تحلیل حذف شوند. لزوم نرمال بودن داده‌ها نیز از مشکلات دیگر رگرسیون می‌باشد.

۵-۶-۱- مدلهای رگرسیون برای دسته بندی

آیا رگرسیون خطی می‌تواند متغیرطبقه‌ای را پیش‌بینی کند؟ برای اینکار لازم است رگرسیون خطی تعمیم یابد. در این مدلهای تعمیم‌یافته، واریانس متغیر پاسخ (Y) تابعی از مقدار میانگین Y است، درحالی‌که در رگرسیون خطی، واریانس Y ثابت بود. انواع متعارف مدلهای خطی تعمیم‌یافته، رگرسیون لجستیک و رگرسیون پواسون هستند. مدلهای رگرسیون لجستیک، احتمال وقوع پدیده‌هایی که تابعی خطی از یک مجموعه متغیرهای مستقل هستند را پیش‌بینی می‌کند. 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 زیراین 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 یک شود که اصولا چنین مقادیری قابل قبول نمی‌باشند. رگرسیون لجستیک برای پیش‌بینی احتمال وقوع مقدار یک متغیر دودویی به‌عنوان تابعی از مجموعه متغیرهای مستقل استفاده می‌شود. این مدل مبتنی بر نسبت برد به باخت یا همان توفیر یا شانس[۱] است.

$$odds \text{ توفیر} = \frac{\text{(برد) احتمال موفقیت}}{\text{(باخت) احتمال شکست}} = \frac{p}{۱-p} \quad , P = \frac{odds}{۱ + odds}$$

مثلاً اگر احتمال موفقیت یک پیشامد ۰/۷۵ باشد توفیر آن برابر با ۳ است.

(احتمال موفقیت -۱ = احتمال شکست) و داریم:

$$odds = \frac{۰/۷۵}{۱-۰/۷۵} = ۳$$

در این مدل از روش [illegible] برد حداکثر درستنمایی برای تخمین لگاریتم طبیعی توفیر استفاده می‌شود (رابطه (۵-۲۵)).

۱- نسبتی است [illegible] شانس بردن به شانس باختن را نشان می‌دهد یعنی برتری، توفیر و مزیت را می‌رساند.

$$\text{Ln (توفیر برآورد شده)} = b_0 + b_1 x_1 + \ldots + b_k x_k \qquad (۲۵-۵)$$

پس از برازش روی مجموعه‌ای از داده‌ها، برآورد توفیر از رابطه (۵-۲۶) به‌دست می‌آید.

$$\text{برآورد توفیر} = \exp(\text{Ln (توفیر برآورد شده)}) \qquad (۲۶-۵)$$

با محاسبه توفیر برآورد شده، احتمال موفقیت از رابطه (۵-۲۷) محاسبه می‌شود.

$$\text{احتمال موفقیت} = \frac{\text{توفیر}}{\text{توفیر} + ۱} = \frac{odds}{Odds + ۱} \qquad (۲۷-۵)$$

پس از پیش‌بینی با مدل رگرسیون لجستیک، باید انطباق مدل و قابل توجه بودن سهم هر کدام از متغیرهای مورد استفاده در مدل را 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 اظهار نظر 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 یا خیر. این مطالب درمراجع اقتصادسنجی و آمار به تفصیل عنوان شده است.

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 تماس بگیرد. از نمونه ۳۰ تایی مشتریانی که در بازاریابی سال گذشته، با آنها تماس حاصل شده است، اطلاعات ذیل موجود است:

- آیا مشتریانی که کارت معمولی داشته‌اند مبادرت به خرید کارت ویژه نیز کرده‌اند یا خیر؟ $(y = ۰/۱)$
- کل مبلغ سالیانه خرید (بر حسب هزار دلار) با کارت معمولی شرکت (x_1)
- آیا دارنده کارت اعتباری برای دیگر اعضای خانواده خود نیز کارت اعتباری خریده است یا خیر (x_2)؟

---

[1]- Deviance Statistic

مدل رگرسیون لجستیک به‌صورت رابطه (۵-۲۸) خواهد بود.

$$\text{Ln} = Y \text{ (برآورد توفیر خرید کارت ویژه)} = b_0 + b_1 x_1 + b_2 x_2 \qquad (۲۸-۵)$$

$$x_1 = ۳۶ \ , \ x_2 = ۱$$

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

بر اساس داده‌های نمونه، 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 ۳٦۰۰۰ دلار خرید کرده

است. درصورتی‌که بدانیم برای اعضای دیگرخانواده خود نیز کارت داشته باشد، احتمال اینکه کارت ویژه شرکت را بخرد چقدر است؟

بوسیله معادله رگرسیون لجستیک مقدار لگاریتم (برآورد نسبت توفیرخرید کارت ویژه) به‌دست می‌آید و از روی آن برآورد توفیر محاسبه شده و در نهایت احتمال خرید کارت اعتباری ویژه به دست می‌آید. جدول (۵-۱٦) داده‌های مربوط به نمونه ۳۰ تایی ازمشتریان سال گذشته شرکت را نشان می‌دهد.

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.

$b_0 = -٦/٩٤ \qquad b_1 = ٠/١٣٩٤٧ \qquad b_2 = ٢/٧٧٤$

Ln (برآورد توفیر خرید کارت ویژه) $= -٦/٩٤ + ٠/١٣٩٤٧(٣٦) + ٢/٧٧٤(١) = ٠/٨٥٤٩٢$

توفیر برآورد شده $= Exp(٠/٨٥٤٨٢) = ٢/٣٥١٢$

احتمال خرید کارت اعتباری ویژه $= (٢/٣٥١٢/(١ + ٢/٣٥١٢)) = ٠/٧٠١٦$

۵-۷-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]: صحت یک روش دسته‌بندی، بستگی به تعداد پیش‌بینی‌های درستی است که آن مدل انجام داده است.

سرعت[2]: زمان لازم برای ساخت و استفاده از مدل در دسته‌بندی است.

پایداری[3]: چنین شاخصی توانایی برخورد مدل در مواجهه با داده‌های غیرمعمول و یا مقادیر مفقوده را نشان می‌دهد.

1- Accuracy

2- Speed

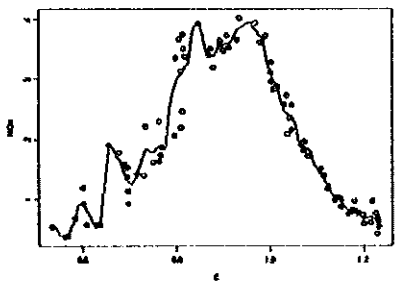
3- Robustness

تفسیر پذیری[۱]: این شاخص نشان‌دهندهٔ میزان قابل فهم بودن مدل توسط دیگران ارائه دیدگاهی روشن نسبت به نحوهٔ دسته‌بندی و نوع دسته‌ها می‌باشد.

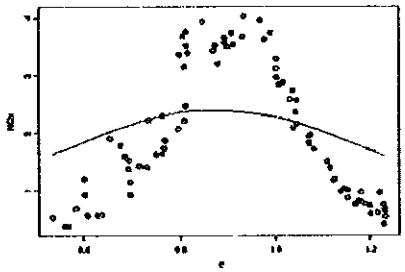
جمع و جور بودن مدل[۲]: اندازهٔ مدل در ایجاد انگیزه جهت استفاده از آن بسیار مهم است. اندازه مدل می‌تواند اندازه درخت و یا تعداد قواعد ایجاد شده توسط آن مدل باشد.

۵-۷-۱- پیچیدگی در مدلسازی

مدلهای پیچیده دقت بالا و درنتیجه انحراف پایینی دارند. البته این مدلها باعث به وجود آمدن پدیده‌ای به‌نام بیش‌برازش می‌شوند. بیش‌برازش یعنی مدل روی داده‌های آموزشی با دقت بالا جواب می‌دهد ولی در مورد داده‌های جدید دقت پایینی دارد. به عبارت دیگر، مدل تعمیم‌پذیری کمی دارد. در این مدلها بیش‌برازش باعث سوگیری بالا خواهد شد. از طرف دیگر، مدلهایی با پیچیدگی کمتر نمی‌توانند مدلسازی را خیلی دقیق انجام دهند، اما پایدارتر هستند، در این مدلها پراکندگی کم شده اما در مقابل واریانس[۳] 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.



شکل ۵-۲۲-الف) انحراف کم -پراکندگی بالا



شکل ۵-۲۲-ب) انحراف بالا -پراکندگی کم

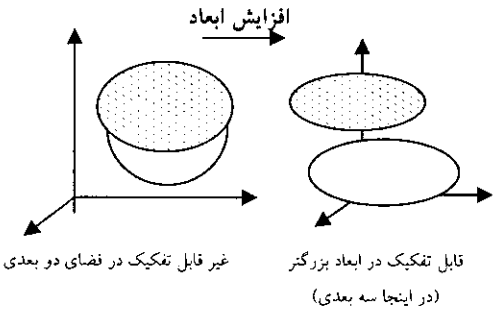
۱- Interpretability

۲- Compactness

۳- Variance

تعادل بین انحراف و سوگیری

در بسیاری موارد ترجیح داده می‌شود که ابعاد بالایی انتخاب شوند تا دسته‌بندی به نحو مطلوبتری انجام پذیرد شکل (۵-۲۳). اما بالا رفتن ابعاد سبب مشکلات خاص خود در زمینه بیش‌برازش خواهد شد. بالا رفتن ابعاد باعث تُنُک شدن فضای ویژگیها خواهد شد، به‌نحوی که حجم محاسبات بسیار بالا خواهد رفت و اغلب آنها غیرضروری هستند. پس آنچه که مهم است به‌دست آوردن سطح مطلوبی از ابعاد یا پیچیدگی در مسئله می‌باشد. روشهایی برای اجتناب از بیش‌برازش مطرح شده‌اند. البته تضمینی وجود ندارد که این روشها برای موارد و وضعیتهای مختلف جواب خوبی ارائه دهند.



شکل ۵-۲۳) مثالی از افزایش ابعاد اجتناب از بیش برازش

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 به وجود آمده، ممکن است درخت حتی داده‌های مغشوش را هم دسته‌بندی کرده باشد که موجب شاخه‌های زائد در درخت و اشکال در دسته‌بندی داده‌های جدید می‌شود. دو روش برای اجتناب از بیش برازش در درخت تصمیم وجود دارد.

- هرس اولیه[1]: توقف ساخت درخت در مراحل اولیه.

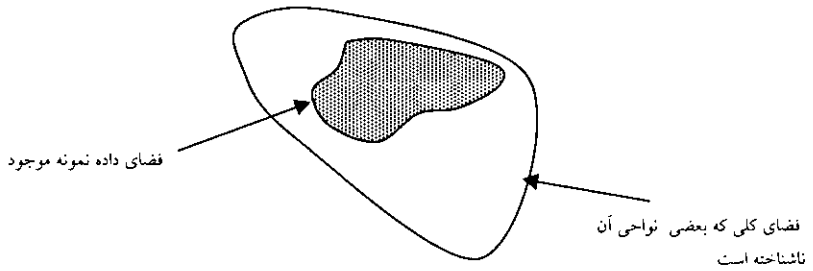
---

[1]- Prepruning

- هرس ثانویه[1]: حذف بعضی شاخه‌ها از درخت ساخته شده (که به‌صورت معمول این روش استفاده می‌شود).

مسئله تعمیم[2]

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 مشکلی است که از آن به‌عنوان مسئله تعمیم یاد می‌شود. تعمیم مشخص می‌کند که تا چه میزان مدل نسبت به ورودی‌های ناشناس، که با مقادیر داده‌های آموزش مدل متفاوتند، پایدار است.



شکل ۵-۲٤) نمایی از ریسک در دسته‌بندی

مدل ساخته شده در روش دسته‌بندی برای داده‌های استفاده شده در ساخت آن و یا داده‌های شبیه به آنها درست جواب می‌دهد، اما همهٔ داده‌ها شبیه به داده‌های آموزش نیستند و حتی در برخی موارد فضای ناشناخته‌ای وجود دارد که در مورد داده‌های آن فضا، هیچ‌گونه اطلاعاتی در دسترس نیست. در هر صورت ناچار هستیم مدل را بر اساس داده‌های موجود بسازیم ولی باید سعی شود تا خطا و یا ریسک مدل را کم کرد.

---

[1]- Postpruning

[2]- Generalization Problem

۵-۷-۲- اندازه‌گیری خطا و میزان صحت در اندازه‌گیریها

فرض کنید با استفاده از داده‌های گذشته، یک مدل دسته‌بندی یا پیش‌بینی را آموزش داده و می‌خواهیم رفتار آینده متغیر هدف را بررسی کنیم. سؤال اساسی این است که صحت روش دسته‌بندی یا پیش‌بینی مورد استفاده چه اندازه است و اینکه چگونه می‌توان صحت دو یا چند روش دسته‌بندی یا پیش‌بینی را با هم مقایسه کرد؟ در ادامه، چگونگی محاسبه صحت روشهای دسته‌بندی یا میزان خطای روشهای پیش‌بینی و همچنین روشهای مورد استفاده در تعیین صحت و چگونگی انتخاب مدل دسته‌بندی یا پیش‌بینی مناسب، به اختصار بیان می‌شود[۱].

۵-۷-۳- ارزیابی صحت روشهای دسته‌بندی

میزان صحت 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 که نشان دهنده کیفیت تشخیص نمونه‌های دسته‌های متفاوت است.

برای محاسبهٔ این شاخص داده‌های آزمون استفاده می‌شود. در اینجا می‌توان نرخ خطا یا دسته‌بندی نادرست را بر اساس شاخص صحت محاسبه کرد. اگر میزان صحت یک روش دسته‌بندی را با $ACC(m)$ نشان دهیم، میزان خطای آن برابر با $ACC(m)$ -۱خواهد بود. از طرف دیگر خطایی که بر اساس داده‌های آموزشی (به جای داده‌های آزمون) محاسبه می‌شود خطای «بازجانشانی»[۲] نامیده می‌شود. این خطا تخمین خوشبینانه‌ای از خطای حقیقی می‌باشد.

ماتریس اغتشاش[۳] ابزاری 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_{ij} در i امین سطر و j امین ستون، نشان دهنده تعداد مشاهداتی از دسته i است که توسط روش دسته‌بندی به‌عنوان دسته j تشخیص داده شده است. برای اینکه یک روش دسته‌بندی، صحت بالایی داشته باشد، حالت ایده‌آل آن است

۲- Resubstitution

۳- Confusion Matrix

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 درصد شناخت 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. از آنجا که در این مثال دو دسته وجود دارد، 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 دسته آنها *"Yes"* بوده ولی به نادرستی در کلاس *"No"*ها دسته‌بندی شده‌اند را نشان می‌دهد و همین‌طور عنصر (۲,۱) نیز تعداد عناصری که برچسب دسته آنها *"No"* است ولی به نادرستی در دستهٔ *"Yes"*ها دسته‌بندی شده را نشان می‌دهد.

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	٧٣٦٦	٢٦٣٤	١٠٠٠٠	٩٥/٥٢

در این مثال از مفاهیمی استفاده شده که به توضیح آنها می‌پردازیم. عنصر «مثبت درست»[۵] به مشاهداتی از دسته c_1 دلالت دارد که توسط روش دسته‌بندی به درستی تشخیص داده شده است. عنصر «منفی درست»[۶] به مشاهداتی از دسته c_2 دلالت دارد که توسط روش دسته‌بندی به درستی تشخیص داده شده است. به‌طور مشابه «منفی غلط»[۷] مشاهداتی از دسته c_1 است که

۵- True Positive

۶- True Negative

۷- False Negative

توسط روش دسته‌بندی به نادرستی در دسته $c_۲$ قرارگرفته و «مثبت غلط»[۱] مشاهداتی از دسته $c_۲$ است که به نادرستی در دسته $c_۱$ قرارگرفته‌اند.

جدول ۵-۱۸) ماتریس دسته‌بندی

	$C_۱$	$C_۲$
$C_۱$	*TP*	*FN*
$C_۲$	*FP*	*TN*

	$C_۱$	$C_۲$
$C_۱$	درست دسته‌بندی شده‌اند	غلط دسته‌بندی شده‌اند
$C_۲$	غلط دسته‌بندی شده‌اند	درست دسته‌بندی شده‌اند

مدلهای مختلف با درجه صحتهای مختلفی قابل پذیرش هستند. به‌عنوان مثال در یک مدل تشخیص سرطان، مدلی با ۹۰٪ صحت قابل‌قبول نیست. بدین منظور شاخصهای دیگری نیز نیاز است که در اینجا به آنها اشاره می‌شود.

$$\text{حساسیت}^{۲} = \frac{\text{تعداد داده‌های برچسب مثبتی که درست دسته‌بندی شده‌اند}}{\text{کل تعداد داده‌های مثبت}} = \frac{TP}{pos} \qquad (۲۹-۵)$$

$$\text{شفافیت}^{۳} = \frac{\text{تعداد داده‌های برچسب منفی که درست دسته‌بندی شده‌اند}}{\text{کل تعداد داده‌های منفی}} = \frac{TN}{neg} \qquad (۳۰-۵)$$

در این فرمول *FP* تعداد داده‌هایی هستند که جزء دسته *No*ها هستند ولی به نادرست در دستهٔ *Yes* ها واقع شده‌اند.

۱- True Negative

۲- Sensitivity

۳- Specificity

$$\text{دقت}^{1} = \frac{TP}{TP + FP} \qquad (۳۱-۵)$$

شاخص آخر یا همان دقت، ترکیبی از دو شاخص قبل است و به‌صورت زیر محاسبه می‌شود:

$$\text{صحت}^{2} = \text{حساسیت} \frac{pos}{(pos + neg)} + \text{شفافیت} \frac{neg}{pos + neg} \qquad (۳۲-۵)$$

توجه به این نکته ضروری است که در روابط فوق، وزن یا اهمیت عناصر ماتریس یکسان در نظر گرفته شده است. درحالی‌که در مسائل مختلف، اهمیت این عناصر می‌تواند متفاوت باشد. مثلا عدم تشخیص سرطان با تشخیص سرطان به اشتباه هزینه‌های کاملا متفاوتی دارند. 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 فوق، چندان مناسب به‌نظر نمی رسد.

۵-۷-۴- میزان خطای پیش‌بینی کننده‌ها

بحث بعدی آن است که چگونه می‌توان صحت روشهای پیش‌بینی را اندازه‌گیری کرد. برای ارزیابی صحت روشهای پیش‌بینی (اختلاف بین مقدار واقعی و مقدار پیش‌بینی شده متغیر وابسته) از مفهوم تابع زیان استفاده کرده و دو شاخص زیر را برای سنجش خطای پیش‌بینی مورد استفاده قرار می‌دهیم. در این روابط d تعداد مثالها یا مشاهدات است.

$$\text{Mean absolute error: } MAE = \frac{\sum_{i=1}^{d} |y_i - y_i'|}{d}$$
$$\text{Mean squared error: } MSE = \frac{\sum_{i=1}^{d} (y_i - y_i')^2}{d} \qquad (۳۳-۵)$$

واضح است که برای بالابردن صحت یک روش پیش‌بینی، لازم است که مقادیر دو شاخص یاد شده تا حد مقدور کوچک باشند. همچنین از دو شاخص زیر نیز برای محاسبه خطای نسبی

۱- Precision

۲ Accuracy

۴- W. H Inmon

پیش‌بینی تک تک مقادیر نمونه، در مقایسه با خطای پیش‌بینی مقادیر نسبت به میانگین استفاده می‌شود.

$$\textit{Relative absolute error}: \quad \frac{\sum_{i=1}^{d} |y_i - y_i'|}{\sum_{i=1}^{d} |y_i - \bar{y}|} \qquad (۳۴-۵)$$

$$\textit{Relative squared error}: \quad \frac{\sum_{i=1}^{d} (y_i - y_i')^2}{\sum_{i=1}^{d} (y_i - \bar{y})^2}$$

منابع

1) Jiawei Han, Micheline Kamber, 2006, Data Mining Concepts & Techniques, Elsevier Inc.

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.

Hand, D. J. , Mannila, H. and Smyth, P. (2001): Principles of Data Mining. Bradford Book.

4) Wilson, D. R. and Martinez, T. R. (2000): Reduction Techniques for Instance-Based Learning Algorithms. In: Machine Learning Vol. 38 (3) pp. 257–286.

5) 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.

6) 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.

7) Wang, J. (2005): Encyclopedia Of Data Warehousing And Mining. Idea Group Publishing.

8) Berry, M. J. A. and Linoff, G. (1997): Data Mining Techniques: For Marketing, Sales, and Customer Support. Wiley.