تحلیل داده‌های آماری تصادفات رانندگی
به وسیله درخت تصمیم

علیرضا پاک‌گوهر[1]
سرهنگ دوم عباس صادقی‌کیا[2]

تاریخ دریافت: ۱۳۸۶/۱۲/۱۱
تاریخ پذیرش: ۱۳۸۷/۳/۷

چکیده

امروزه محققان با انفجار اطلاعات مواجه شده‌اند. برای نمونه، چند منشا این اطلاعات، سرمایه‌گذاری‌های بسیار زیادی است که در تولید بانک‌های اطلاعاتی، انتقال مفاهیم (داده‌ها و اطلاعات) از طریق شبکه و کامپیوتری شدن فرآیندهای اجرایی انجام می‌گیرد. یکی از مجموعه داده‌هایی (Data bases) که حاوی اطلاعات ذی‌قیمتی درباره فاکتورهای موثر و احتمالا دارای همبستگی‌های خطی و غیرخطی (از دیدگاه تحلیل رگرسیونی) برای موضوع تصادفات است، نرم‌افزار سیستم جامع تصادفات جاده‌ای موسوم به نرم‌افزار تحلیل فرم‌های کام۱۱۳ و کام۱۱۴ است. این پژوهش با توجه به اهمیت دانش پنهان در انبوه اطلاعات موجود در مجموعه داده‌های یادشده و لزوم به کارگیری مدیریت دانش در این خصوص بالاخص به کارگیری الگوریتم‌های تحلیلی در حوزه داده‌کاوی هم اکنون طی موضوع تحقیقی‌ای با عنوان بررسی علل و عوامل موثر بر تصادفات بر اساس مدل‌های رگرسیونی LR و CART به تحلیل رگرسیونی درختی و لجستیک متغیرهای مستقل و وابسته پرداخته است. بر این عقیده‌ایم که این روش‌ها اساسا متکی به الگوریتم‌ها و ساختارهای داده برای آمار محاسباتی با کارایی بالا خواهند بود. همچنین معتقدیم برای اینکه یک سیستم اکتشاف واقعا برای جامعه اطلاعاتی محققان ترافیک مفید باشد باید بتواند تحلیل را به محض اینکه دانشمندان پرسش‌هایشان را فرمول‌بندی می‌کنند و فرضیه‌شان را توضیح می‌دهند، انجام دهد. این کار، نیاز به ساختارهای داده مقیاس‌پذیر و الگوریتم‌هایی دارد که قادر باشند میلیون‌ها نقطه داده را با ده‌ها یا ده‌ها هزار بعد روی سخت‌افزارهای محاسباتی مدرن در زمان چند ثانیه تحلیل کنند که نیازمند طراحی الگوریتم‌های مبتنی بر چنین نیازی بوده و تا حصول منظور نهایی در چنین سیستم اکتشافی که به محققان امکان می‌دهد به جای علم محاسبات روی موضوع تحقیقشان متمرکز شوند، گام‌های نپیموده بسیاری هست که در حوصله این مقاله نمی‌گنجد. این مقاله می‌کوشد با معرفی الگوریتم درخت تصمیم علاوه بر آموزش روش تحلیلی یادشده، محققان حوزه ترافیک را با یکی از ابزارهای داده‌کاوی آشنا کند.

کلید واژه‌ها: *تحلیل داده، تصادفات رانندگی، درخت تصمیم*

۱- کارشناس ارشد آمار و عضو هیئت علمی دانشگاه آزاد اسلامی واحد آیت الله آملی
۲- کارشناس ارشد مدیریت ترافیک عضو هیئت علمی دانشگاه علوم انتظامی ناجا

مقدمه

حیات اقتصاد نوین بر پایه داده‌ها بنا شده است. به علت گستردگی استفاده از بانک‌های اطلاعاتی و پایگاه‌های داده‌های بزرگ در تمامی زمینه‌های تجاری، علمی، صنعتی و خدماتی که در اثر توسعه فناوری اطلاعات (I.T) و انفورماتیک به وجود آمده است با داده‌های بسیار بسیارزیادی مواجهیم به طوری که فقط در سال ۲۰۰۰ میزان ظرفیت نصب شده جهت ذخیره‌سازی اطلاعات از کل ظرفیت موجود در دهه ۱۹۹۰ بیشتر بوده است [۱]. در حال حاضر تقریبا حجم کل اطلاعات در کامپیوترها هر پنج سال دو برابر می‌شود و با توجه به سرعت ایجاد برنامه‌های چندرسانه‌ای و بانک‌های اطلاعاتی پیش‌بینی می‌شود شتاب رشد اطلاعات به دو برابر در سال برسد [۲].

برای یک محقق و پردازشگر اطلاعات، تکنیک‌های تولید و ذخیره‌سازی پایگاه داده‌های کنونی و دستیابی به اطلاعات نهفته در این داده‌های حجیم از اهمیت بسیاری برخوردارند. از سوی دیگر نیاز به چگونگی بهره‌برداری از این داده‌ها معمولی است زیرا هرجایی که داده‌ای وجود دارد، اطلاعات نهفته‌ای نیز موجود است که تنها آگاهی از روش استخراج اطلاعات و پردازش داده‌ها را لازم دارد و این امر برای یک تحقیق آماری مفروض است [۳].

امروزه دیگر نمی‌توان آنچنان که باید و شاید تنها با به کارگیری سیستم‌ها و تکنیک‌های سنتی از داده‌های بانک‌های اطلاعاتی استفاده برد زیرا این داده‌ها معمولا جزو داده‌های دست دوم محسوب می‌شوند و براساس نیاز محقق برای دستیابی به اطلاعات خاص در مورد فرضیه، سوال یا هدف پژوهشی مورد نظر به دست نیامده‌اند تا به استخراج سریع اطلاعات مورد نظر و پردازش داده‌های موجود پرداخته شوند و به همین جهت نیاز به طراحی سیستم‌هایی که قادر به اکتشاف و دستیابی به اطلاعات مورد نظر کاربران با تاکید بر مداخله حداقل انسان و با همان سرعتی که داده‌ها در بانک‌های اطلاعاتی تولید می‌شوند، احساس شده است [۴].

در دنیای کنونی این کمبود اطلاعات نیست که مسئله است بلکه کمبود دانشی است که از این اطلاعات می‌توان حاصل کرد. میلیون‌ها صفحه وب، میلیون‌ها کلمه در کتابخانه‌های دیجیتال و هزاران صفحه اطلاعات در هر شرکت تنها چند دست از این منابع اطلاعاتی هستند اما نمی‌توان به طور مشخص منبعی از دانش را در این بین معرفی کرد. دانش خلاصه اطلاعات و نتیجه‌گیری و حاصل فکر و تحلیل روی اطلاعات [۵].

بر اساس این اصل که داده‌ها سرچشمه اطلاعاتند و اطلاعات سازمان‌یافته و غیرسازمان‌یافته در هر ساختار سازمانی به طور مشهودی در اختیار و در دسترس است، نیاز به مدیریت اطلاعات و در یک معنای غایی‌تر مدیریت دانش ملموس بوده و هست.

مدیریت دانش[1]

مدیریت دانش، حوزه نسبتا جدید و رو به توسعه‌ای است که به ارایه متدولوژی جهت جمع‌آوری و استفاده مجدد از دانش سازمانی می‌پردازد.

از جمله نتایج موفقیت‌آمیز مدیریت دانش، درک و استفاده سازنده از یادگیری سازمانی و جریان اطلاعاتی درون سازمانی است.

همانطور که گفته شد هر سازمانی متشکل از اطلاعات نهفته‌ای از افراد و دانش‌های در اختیار است که در این راستا چهار اصل باید رعایت شود:

❋ اصل یکم، دانش توانایی انسان است؛ به طور مثال توانایی انجام کار یا توانایی قضاوت در مورد مسئلهای چه در حال و چه در آینده. دانش، دگردیسی اطلاعات توسط شخص است.

❋ اصل دوم، کسب دانش یک فرآیند داینامیک و پویاست.

❋ اصل سوم، دانش زایشی و چند بعدی است. به آن معنا که می‌توان آنرا مورد کاوش، پژوهش و توسعه قرار داد یا اینکه می‌توان آن را خلاصه و چکیده کرد. به دیگر سخن، داشتن دانش، مالک آن را قادر می‌سازد تا صرفا به بازگویی اطلاعات کسب شده، نپردازد بلکه به تولید مطالب جدیدی در رابطه با موضوع نیز اقدام کند.

❋ اصل چهارم، دانش تودرتو و پیچیده است. در حقیقت طبیعت چند بعدی دانش به این دیدگاه گره خورده است.

دانش بدنه پیچیده اطلاعات سازمان یافته است که در بسته‌های بزرگ توزیع شده بنابراین اکتساب دانش توسط یک فرد بدون کمک ابزارهای فیلترکننده یا راهنما ممکن است به جهت افزونگی اطلاعات با مشکل مواجه شود.

در این بین، ابزارهای مدیریت دانش به جمع‌آوری، سازمان‌دهی، ذخیره و انتقال اطلاعاتی کمک می‌کنند که یک انسان توسط آنها به کسب دانش می‌پردازد [۶].

1- *Knowledge Management*

یادگیری دانش و سازمان‌های یادگیرنده

هدف از مطالعات در زمینه مدیریت اطلاعات غالبا تسهیل کار تیمی و در عین حال یادگیری دانش است. بخش‌های مهم مقوله یادگیری دانش عبارتند از:

۱- اکتساب دانش: یادگیری وقتی رخ می‌دهد که سازمان، دانش کسب کند که این دانش نه تنها از محیط خارج که از مرتب‌سازی و نوساماندهی دانش موجود به دست می‌آید.

۲- توزیع اطلاعات: توزیع و اشتراک اطلاعات بین واحدهای مختلف درون سازمان به افزایش یادگیری می‌شود.

۳- تفسیر اطلاعات: اهمیت تبدیل اطلاعات یا داده‌ها به دانش قبل از شروع یادگیری بسیار بالاست.

۴- حافظه سازمانی: دانش گروهی و فراگیری مطالب قبلی به یادگیری بهتر کمک می‌کند.

بسیاری از محققان، رابطه بین مدیریت دانش یادگیری سازمانی را به عنوان نمادی از پیشرفت سازمان در گذر از تمرکز بر منابع مادی به منابع و پتانسیل انسانی پذیرفته و بر آن تاکید دارند.

سازمان‌ها همواره نیازمند کسب اطلاع درباره محیط، شرایط و رخدادهای پیرامون خود هستند چراکه برای یک مدیریت مستمر و مفید باید اطلاعات مستمر و مفیدی در مورد مشتریان، فناوری‌ها و رقبای جدید و... کسب شود.

توسعه شیوه‌های صحیح گردآوری و تحلیل اطلاعات بیرونی به طور روزافزون، چالش عمده مدیریت دانش شد که یکی از رهاوردهای آن تولید و توسعه دانشی جدید با ریشه‌هایی استوار از علوم کهنسال از قبیل آمار، مدیریت، مهندسی سیستم و... به نام کشف دانش شد [۵].

کشف دانش[1]

امروزه اندازه داده‌های به دست آمده محاسبه شده در بانک‌های اطلاعاتی[2] فراتر از توانایی ما برای کاهش و تجزیه و تحلیل داده‌ها بدون استفاده از تکنیک‌های تحلیل مکانیزه[3] شده است. بسیاری از بانک‌های اطلاعاتی علمی و بازرگانی پرتراکنش[4] در حد خارق العاده‌ای در حال گسترش هستند. یک تشکیلات ساده مانند نقشه‌برداری نجومی اجرایی به نام

1- Knowledge Discovery
2- Databases
3- automated analysis techniques
4- transactional

SCICAT پیش‌بینی کرده است که متجاوز از سه تریلیون بایت داده تولید کند [۶]. کشف دانش در پایگاه‌های داده‌ها (K.D.D)[1] حوزه‌ای است که به استنتاج فراهم‌آوری پاسخ‌های تجزیه و تحلیل‌های مکانیزه شده می‌پردازد.

کشف دانش به عنوان استخراج غیربدیهی از اطلاعات ناشناخته، ضمنی و به طور بالقوه مفید در میان داده‌ها تعریف می‌شود [۷]. فیاض به طور کلی بین داده‌کاوی و کشف دانش یک تعمق کامل قایل است [۸]. طبق تعاریف، فرآیند کشف دانش از نتایج اولیه داده‌کاوی (فرآیند نظام‌یافته یا الگوهای تلخیص‌کننده داده‌ها) و با دقت و صحت تبدیلات آنها به اطلاعات قابل فهم و مورد استفاده به دست می‌آید. این اطلاعات نوعا با استفاده از تکنیک‌های استاندارد، بازیافتنی نیستند اما به واسطه استفاده از تکنیک‌های هوش مصنوعی A.I[2] آشکار شده‌اند. دانش K.D.D یک رشته در حال گسترش است. متدولوژی‌های[3] (روش‌شناسی) کشف دانش بسیاری در حال اجرا و گسترش وجود دارد. بعضی از این تکنیک‌ها کلی هستند در حالی که بعضی دیگر قلمرو خاص خود را دارند.

داده‌کاوی چیست؟

داده‌کاوی عبارت است از فرآیند به کارگیری تکنیک‌های یادگیری کامپیوتری برای تحلیل و تحلیل اتوماتیک و کشف دانش از داده‌های موجود در یک پایگاه داده است.

داده‌کاوی فرآیند کشف روابط ناشناخته و الگوها در داده استداده‌کاوی، یک روش بسیار کاراست برای کشف اطلاعات از داده‌های ساخت یافته‌ای که در جداول نگهداری می‌شوند. داده‌کاوی، الگوها را از تراکنش‌ها[4] ، استخراج، داده را گروه‌بندی و دسته‌بندی می‌کند. به وسیله داده‌کاوی می‌توان به وجود روابطی میان اقلام داده‌ای که پایگاه داده را پر کرده‌اند، پی برد اما در عین حال با داده‌کاوی مشکلی داریم و آن عدم وجود عامیت در کاربرد آن است.

تعداد منابع داده‌ای ساخت یافته‌ای که به حد کافی بزرگ نیز باشند که مفاهیم داده‌کاوی قابل اعمال بر آنها باشد، چندان زیاد نیستند. در واقع بیشتر دانش افراد اگر به صورت غیردیجیتال نباشند، کاملا غیرساخت یافته‌اند. کتابخانه‌های دیجیتال، اخبار، کتاب‌های الکترونیکی، بسیاری از مدارک مالی، مقالات علمی و تقریبا هر چیزی که می‌توان در داخل

1- Knowledge Discovery In Databases
2- Artificial Inteligence
3- methodologies
4- Transactions

وب یافت، ساخت یافته نیستند و در نتیجه نمی‌توان آموزه‌های داده‌کاوی را در مورد آنها به طور مستقیم استفاده کرد.

با این حال، سه روش اساسی در مواجهه با حجم وسیع از اطلاعات غیر ساخت یافته گسترده شده در جهان وجود دارد؛ بازیابی اطلاعات[۱]، استخراج اطلاعات[۲] و کشف دانش.

ضرورت داده‌کاوی

پس از توسعه فوق‌العاده پایگاه داده‌ها که در نتیجه پیشرفت فوق‌العاده فناوری اطلاعات در کسب و ذخیره‌سازی داده‌های عددی به وجود آمد در اواخردهه ۱۹۸۰ تلاش برای استخراج و استفاده از اطلاعات پایگاه‌های داده‌های بزرگ آغاز شد.

طی سال‌های ۱۹۸۹ و۱۹۹۱ کارگاهی کشف دانش از پایگاه داده‌ها[۳] توسط پیاتتسکی[۴] و همکارانش برگزار شد.

در فواصل سال‌های ۱۹۹۱ تا۱۹۹۴ کارگاه‌های کشف دانش از پایگاه داده‌ها توسط فیاض[۵] و پیاتتسکی و دیگران برگزار شد به طور رسمی اصطلاح داده‌کاوی برای اولین بار توسط فیاض در نخستین کنفرانس بین‌المللی کشف معرفت و داده‌کاوی در سال ۱۹۹۵ مطرح شد.

داده‌کاوی یک رشته نسبتا جدید علمی است که از انجام تحقیقات در رشته‌های آمار یا یادگیری ماشین[۶] علوم کامپیوتر[۷] به خصوص مدیریت پایگاه داده‌ها شکل گرفته است. البته مرزهای این رشته‌ها در داده‌کاوی کاری حجیم است [۹].

طبقه‌بندی با درخت تصمیم‌گیری در پایگاه‌داده‌ها

طبقه‌بندی داده‌ها یک فرآیند دو مرحله‌ای استکه در مرحله اول یک مدل ساخته می‌شود که مجموعه‌ای از طبقه‌های داده‌ای یا مفاهیم را مشخص می‌کند. این مرحله را مرحله یادگیری[۸] گوییم که در آن یک الگوریتم طبقه‌بندی یک مدل را با تحلیل مجموعه‌ای آموزشی[۹] که از

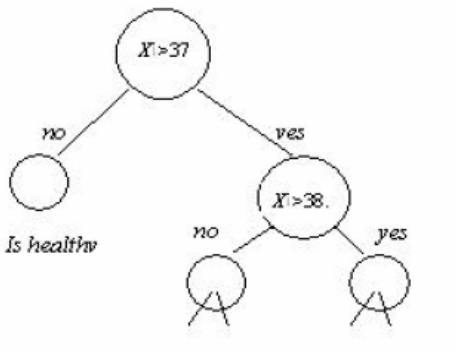
1- Information Retrieval
2- Information Extraction
3- Knowledge Discovery and Data Mining
4- Piatetsky
5- Fayyad
6- Machine learning
7- Computer Sience
8- Learning
9- training set

مولفه‌های پایگاه است می‌سازد و برچسب طبقه‌های مربوط به این مولفه‌ها را مشخص می‌کند. یک مولفه X با یک بردار صفت $X=(x1,x2,\ldots,xn)$ نمایش داده می‌شود.
فرض می‌شود که هر مولفه به یک طبقه از پیش تعریف شده متعلق است و طبقه با یک صفت که به آن صفت برچسب طبقه می‌گوییم، مشخص می‌شود. مجموعه آموزشی به صورت تصادفی از پایگاه انتخاب می‌شود.
در مرحله دوم، یادگیری از طریق یک تابع $y=f(X)$ انجام می‌گیرد که می‌تواند برچسب طبقه هر مولفه X از پایگاه را پیش‌بینی کند. این تابع به صورت قواعد طبقه‌بندی، درخت تصمیم‌گیری یا فرمول‌های ریاضی است [۱۰].
آنچه ما در اینجا برای طبقه‌بندی بررسی می‌کنیم طبقه‌بندی با درختان تصمیم‌گیری[1] است.

درخت تصمیم چیست؟

یک درخت معمولا تشکیل شده از ریشه[2]، شاخه‌ها[3]، گره‌ها[4] (جایی که شاخه‌ها منشعب می‌شوند) و برگ‌ها[5]. درخت تصمیم هم به صورت مشابه از گره‌ها که با دایره نشان داده می‌شوند و شاخه‌ها که با پاره‌خط‌های اتصال بین گره‌ها نشان داده می‌شوند، تشکیل شده‌اند. درخت تصمیم را به منظور سادگی در رسم معمولا از چپ به راست یا از بالا به پایین رسم می‌کنند به طوری که ریشه در بالا قرار بگیرد. گره اول را ریشه می‌گویند. انتهای یک زنجیره «ریشه، شاخه، گره و...» را یک «برگ» می‌نامند. از هر یک از گره‌های داخلی (یعنی هر گره‌ای که برگ نباشد) دو یا چند شاخه دیگر می‌تواند منشعب شود. هر گره مربوط به یک خصوصیت معین است و شاخه‌ها به معنای بازه‌ای از مقادیر هستند. این بازه‌های مقادیر باید بخش‌های مختلف مجموعه مقادیر معلوم برای خصوصیت‌ها را به دست دهند.
هنگامی که دقیقا دو شاخه از یک گره داخلی منشعب شود (چنین درختی را درخت دوحالته[6] می‌گویند) – همانطور که در شکل شماره یک نشان داده شده– هر یک از این دو شاخه می‌تواند نماینده یک عبارت درست یا غلط برحسب خصوصیات معلوم باشد.

1- Decision Trees
2- Root
3- beach
4- Node
5- Leaf
6- Binary



شکل یک- درخت دو حالته

هر مقدار Y به یکی از گره‌های پایانی درخت (برگ‌ها) منتسب می‌شود. در مورد مسئله تشخیص الگو، مقدار داده شده به صورت یک کلاس معین است و در مورد تحلیل رگرسیون این مقدار به صورت یک عدد حقیقی است.

با استفاده از یک درخت تصمیم می‌توانیم برای هر یک از مشاهدات X یک مقدار پیش‌بینی شده Y را پیدا کنیم. برای این منظور از ریشه درخت آغاز می کنیم، خصوصیات مربوط به ریشه را در نظر می‌گیریم و تعیین می‌کنیم که مقدار مشاهده شده برای خصوصیت معلوم به کدام شاخه تعلق دارد آنگاه گره‌ای را در نظر می‌گیریم که شاخه مورد نظر به آن می رسد. این کار را برای این گره نیز انجام می‌دهیم و به همین صورت ادامه می‌دهیم تا به یک برگ برسیم. مقدار Ys منتسب به برگ Sام مقدار پیش‌بینی شده برای X خواهد بود بنابراین درخت تصمیم مدل وابستگی T را برای Y از X به صورت Y=T(X) به دست می‌دهد.

درخت‌هایی تصمیمی که در یک مسئله تحلیل رگرسیون در نظر گرفته می‌شوند، درخت‌های رگرسیون نامیده می‌شوند.

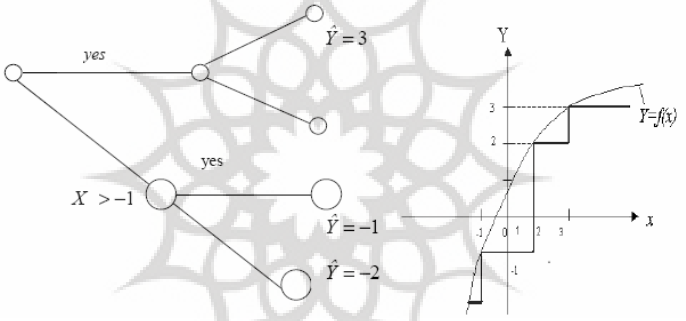
در این مقاله ساده‌ترین نوع درخت تصمیم را که در بالا توضیح داده شد در نظر می‌گیریم. اما انواع پیچیده‌تر درخت‌ها نیز وجود دارد که در آنها گره‌های داخلی متناظر با عبارات پیچیده‌تری از بیش از یک خصوصیت معلوم هستند. برای مثال، این عبارت‌ها می‌توانند ترکیبی خطی از خصوصیت‌های کمی مثلا عبارت $10X_1 + 5X_2 - 1 > 0$ که مربوط به چندین منطقه محلی از فضای چند متغیره است که توسط صفحات فرضی از هم جدا شده‌اند، باشند. از این دیدگاه، صفحات فرضی درخت تصمیم مورد نظر بر محورهای عددی عمود هستند.

درخت تصمیم باید یکپارچه باشد، یعنی در مسیر از ریشه تا یک برگ نباید بازه‌های تغییر در نظر گرفته نشده وجود داشته باشد. برای مثال «$X_1 < 30$» и «$X_1 > 37$».

توسط درخت تصمیم می‌توان خصوصیات کمی و خصوصیات کیفی را به طور همزمان پردازش کرد بنابراین درخت تصمیم نشان دهنده یک مدل منطقی از پدیده مورد تحقیق است.

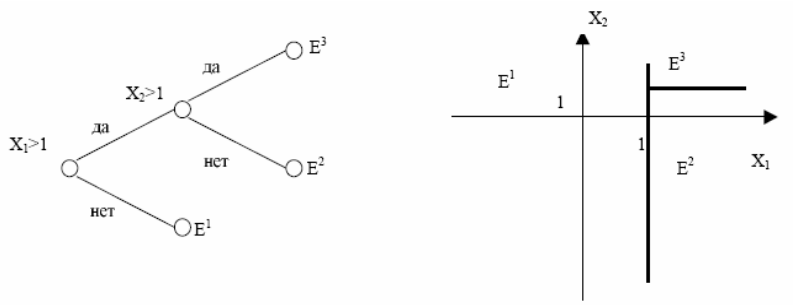
نقیصه درخت تصمیم در واقع این است که در موردی که تمام خصوصیت‌ها کمی باشند، درخت تصمیم تخمین‌های غیردقیقی را از جواب نهایی به دست می‌دهند. برای مثال یک درخت رگرسیون که در شکل زیر نشان داده شده، تقریبا تخمین ثابتی را از تابع رگرسیون ارایه می‌کند. از سوی دیگر، امکان جبران این کمبود توسط افزایش تعداد برگ‌ها، یعنی با کاهش طول «پاره‌خط‌ها» یا «مرحله‌ها» وجود دارد.

یک درخت تصمیم با M برگ را در نظر بگیرید. این درخت تصمیم نتیجه تجزیه فضای خصوصیات به M زیرمنطقه غیرهمپوشان E^1 تا E^M است به صورتی که زیرمنطقه E^S نشان دهنده برگ Sام (شکل شماره دو) است. هر زیر منطقه چگونه شکل گرفته است؟



شکل دو- درخت تصمیم با M برگ

E^s به صورت حاصلضرب دکارتی $Es = E_1^s \times E_2^s \times \ldots \times E_n^s$ تعریف شده که در آن E^s_j تصویر E^s روی jامین خصوصیت است. E^s_j در مسیر بعدی به دست می‌آید. اگر خصوصیت X_j در مسیر از ریشه به Sامین برگ اصلا اتفاق نیفتد، آنگاه E^s_j با بازه‌ای از تعریفات خصوصیت X_j تلاقی خواهد کرد. در غیر این صورت، E^s_j برابر با سطح مشترک بین تمام زیرمنطقه‌های خصوصیت X_j است که در مسیر ریشه تا Sامین برگ قرار دارند.



شکل سه- تشخیص الگوی P.R.P

فرض کنید که برخی مشاهدات به صورت $Data = (x^i, y^i), i = 1,..N$ را داشته باشیم که هر یک از این مشاهدات (بر حسب X) به یکی از زیرمجموعه‌های در نظر گرفته شده مربوط باشد، یعنی $x^i \in E^s$.

مجموعه داده‌های مربوط به E^s را $DATA^s$ می‌نامیم و تعداد مشاهدات را با N^s نشان می‌دهیم. فرض کنید N^s_i تعداد مشاهدات از $DATA^s$ باشد که به کلاس iام تعلق دارد (مسئله تشخیص الگو P.R.P) [۹].

چگونه درخت تصمیم را بسازیم

روش تشکیل دادن یک درخت تصمیم از داده‌های آماری را ساختن درخت نیز می گویند. در این قسمت با برخی روش‌های ساخت درخت و نیز روش‌های تعیین کیفیت درخت‌ها آشنا می‌شویم. برای هر منظور خاص در تحلیل آماری تعداد بسیار زیادی از انواع درخت تصمیم وجود دارد که بسیاری از آنها حتی شناخته نشده‌اند. سوالی که وجود دارد، این است که کدام درخت بهترین است و چگونه آن را پیدا کنیم.

برای پاسخ به قسمت اول، روش‌های مختلف تعریف پارامترهایی را در نظر می‌گیریم که کیفیت درخت را تعیین می‌کنند. از نظر تئوری، می‌توانیم خطای مورد انتظار در پیش‌بینی را به عنوان پارامتر پایه در نظر بگیریم. به هرحال در عمل این قانون هنوز به عنوان اصل شناخته نشده است بنابراین فقط می‌توانیم کیفیت را با توجه به مجموعه‌ای از مشاهدات که به ما داده شده، تخمین بزنیم [۹].

پارامترهای کیفیت یک درخت

فرض کنید یک درخت تصمیم و نمونه‌ای از N شیء داریم. امکان انتخاب دو نوع اصلی از پارامترهای توضیح‌دهنده کیفیت یک درخت وجود دارد؛ نوع اول پارامترهای دقت هستند و نوع دوم پارامترهای پیچیدگی درخت.

پارامترهای دقت یک درخت را می‌توان با کمک نمونه تعریف کرد و کیفیت تقسیم اشیا در کلاس‌های مختلف (در مورد یک مسئله تشخیص)، یا اندازه بزرگی خطا (در مورد یک مسئله تحلیل رگرسیون) را تعیین کرد.

عدد نسبی (فراوانی) خطاها به معنای کسری از اشیا است که توسط درخت به طور اشتباه به یک کلاس نسبت داده شده:

$$\hat{P}_{err} = \frac{N_{err}}{N}$$

که در آن

$$N_{err} = \sum_{S=1}^{M} \sum_{\substack{i=1 \\ i \neq \hat{Y}(S)}}^{K} N_i^S$$

و K تعداد کلاس‌هاست.

واریانس نسبی برای یک درخت تصمیم را می‌توان از فرمول زیر محاسبه کرد:

$$d_{om} = \frac{d_{oc}}{d_0}$$

که در آن $d_{oc} = \frac{1}{N}\sum_{S=1}^{M} \sum_{i \in Data^S} (\hat{Y}(S) - y^i)^2$ واریانس باقیمانده است، واریانس اولیه به صورت:

$$d_0 = \frac{1}{N}\sum_{i=1}^{N} (y^i - \overline{y})^2$$

تعریف می‌شود و داریم:

$$\overline{y} = \frac{1}{N}\sum_{i=1}^{N} y^i$$

پارامترهای پیچیدگی، خصوصیات شکل درخت را تعیین می‌کنند و به نمونه بستگی ندارند. برای مثال، پارامترهای پیچیدگی یک درخت به صورت تعداد برگ‌های درخت، تعداد گره‌های داخلی آن و بیشترین طول مسیر از ریشه تا یک برگ می باشند. همچنین می‌توان از طول

یک مسیر خارجی استفاده کرد که به صورت تعداد شاخه‌هایی تعریف می‌شود که یک درخت کامل را تشکیل می‌دهند.

پارامترهای پیچیدگی و دقت با هم دارای پیوستگی داخلی هستند و به عنوان یک قانون می‌توان گفت درختی که پیچیده‌تر باشد دارای دقت بیشتری است (در درختی که هر برگ آن کننده یک شیء باشد بیشترین میزان دقت وجود دارد).

اگر دیگر شرایط یکسان باشد، درختی که پیچیدگی کمتری داشته باشد، ترجیح داده می‌شود. چنین درختی مدل ساده‌تری از پدیده مورد تحقیق را به دست می‌دهد و تفسیرهای بعدی (توضیح مدل) را آسان می‌کند.

علاوه بر این، از تحقیقات تئوری چنین برمی‌آید که در صورت کوچک بودن اندازه نمونه (در مقایسه با تعداد خصوصیات) درخت‌هایی که بیش از حد پیچیده باشند، ناپایدار هستند، یعنی دارای تعداد خطاهای بیشتری برای مشاهدات جدید خواهند بود.

از طرف دیگر، روشن است که یک درخت خیلی ساده نیز امکان رسیدن به پیش‌بینی خوبی را فراهم نمی‌کند بنابراین در انتخاب بهترین درخت تصمیم باید به یک «توافق» معینی بین پارامترهای دقت و پیچیدگی برسیم. برای رسیدن به چنین توافقی مثلا می‌توانیم از این شرط برای کیفیت استفاده کنیم: $Q = p + \alpha M$ که در آن p یک پارامتر دقت و α یک پارامتر معلوم است. بهترین درخت با توجه به این شرط باید دارای کمترین مقدار Q باشد.

از روشی که در آن بیشترین پیچیدگی مجاز برای درخت تعیین می‌شود به طور همزمان با جست‌وجوی دقیق‌ترین درخت هم می‌توان استفاده کرد [۹].

تخمینی از کیفیت روی یک نمونه کنترل

نمونه کنترل (یا تست) به نمونه‌ای گفته می‌شود که برای ساختن یک درخت به کار برده نمی‌شود بلکه برای تخمین زدن کیفیت یک درخت ساخته شده به کار می‌رود و دو پارامتر محاسبه می‌شود که این دو پارامتر تعداد نسبی خطاها برای مسایل تشخیص و واریانس نمونه کنترل برای مسایل تحلیل رگرسیون هستند.

از آنجا که این نمونه در ساخت درخت تصمیم نقشی ندارد، این پارامترها خطای نامعلوم «واقعی» را بهتر نشان می‌دهد. هر چه اندازه نمونه کنترل بزرگ‌تر باشد، درجه تخمین هم بالاتر خواهد بود.

در یک مسئله تشخیص –تحت شرایط مستقل بودن مشاهدات– فراوانی خطاها از توزیع دو جمله‌ای به دست می‌آید بنابراین با دانستن تعداد خطاها در نمونه کنترل می‌توان بازه

اطمینانی را پیدا کرد که به احتمال معینی تعداد خطاهای کلاس‌بندی اشتباه به آن بازه تعلق دارد.

در مرجع [۵] نمودارهایی آمده که در آنها می‌توان بازه اطمینان را به ازای اندازه نمونه مشخص و تعداد خطاها در نمونه کنترل تعیین کرد [۹].

روش‌های ساخت درخت تصمیم

روش‌های موجود (چندین روش وجود دارد) را می‌توان به دو گروه اصلی تقسیم کرد؛ گروه اول شامل روش‌هایی برای ساختن درخت با میزان بهینگی صریح (با توجه به شرط کیفیت درخت) و گروه دوم شامل روش‌های ساخت درخت با میزان تقریبی بهینگی است.

مسئله جست‌وجو به دنبال درخت بهینه می‌تواند به مسئله برنامه‌ریزی گسسته[1] یا انتخاب از میان تعداد محدود (ولی بسیار بزرگ) منجر شود که این از این حقیقت ناشی می‌شود که برای یک نمونه آموزشی محدود، تعداد محدودی حالت شاخه‌ها برای هر خصوصیت وجود دارد.

انواع روش‌های پایه برنامه‌ریزی گسسته عبارتند از: جست‌وجوی کامل، روش برنامه‌ریزی دینامیک و روش شاخه‌ها و اتصالات. استفاده از این روش‌ها برای درخت تصمیم نیاز به کار بسیار زیادی دارد مخصوصا اگر تعداد مشاهدات یا تعداد خصوصیت‌ها زیاد باشد بنابراین روش‌های تقریبی را در نظر می‌گیریم که عبارتند از: روش شاخه‌بندی ساختمانی، روش هرس کردن و روش برگشتی. اجازه دهید تمام اعمال پایه روی درخت تصمیم را در نظر بگیریم. روش‌های ساخت درخت شامل توالی رشته‌ای این اعمال خواهد بود.

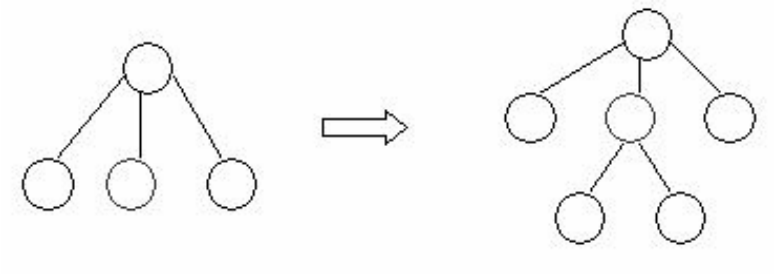
عمل شاخه‌بندی (تقسیم)

عمل شاخه‌بندی، پایه ساخت درخت است. یک گره از درخت و یک خصوصیت X_j را در نظر بگیرید. فرض کنید بازه تعریف این خصوصیت به تعداد L_j زیر مجموعه تقسیم شود (در ادامه روش‌های انتخاب چنین زیرمجموعه‌هایی خواهد آمد).

در مورد خصوصیت‌های کمی، این زیرمجموعه‌ها گروهی از زیربازه‌های مجزا هستند. در مورد خصوصیت‌های کیفی، زیر مجموعه‌ای از مقادیر و در مورد داده‌های خصوصیات ترتیبی زیر مجموعه‌هایی شامل مقادیر همسایه هستند. فرض کنید به هر یک از این زیر مجموعه‌ها

1- discrete programming

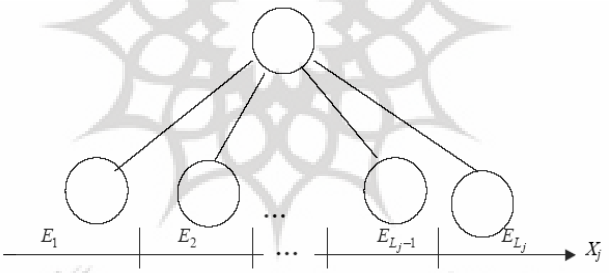
یک شاخه نسبت دهیم که از گره فعلی (مادر) به سمت یک گره جدید که (فرزند) نامیده می‌شود، امتداد می‌یابد بنابراین گره به تعداد L_j گره جدید «منشعب» یا «تقسیم» شده است (شکل شماره چهار).



شکل چهار- انشعاب گره به تعداد Lj گره جدید

توجه کنید برای درخت‌های باینری L_j همیشه برابر با دو است. اگر L_j برای یک درخت همیشه برابر با سه باشد آن درخت را «سه‌گانه» می‌نامیم. اگر L_j همیشه برابر با چهار باشد یک درخت «چهارگانه» خواهیم داشت.

چگونه بازه‌ای از تعریف را تقسیم کنیم؟ مجموعه‌ای از مشاهدات را برای گره مورد نظر به دست می‌آوریم و مقادیر X_j را برای این مشاهدات در نظر می‌گیریم.

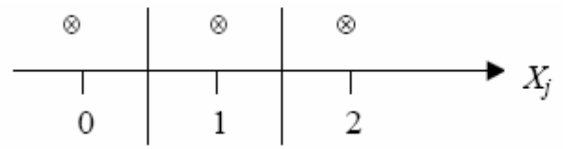


شکل پنج- نقشه بازه‌ها با توجه به مرزها

یک خصوصیت کمی را در نظر بگیرید. در این مورد، مرزها در وسط بازه‌های بین مقادیر همسایه قرار دارند و تقسیم با توجه به این مرزها صورت می گیرد (شکل شماره پنج).

برای مثال در شکل زیر (مقادیر مشاهده شده برای خصوصیات با ⊗ نشان داده شده‌اند) برای یک درخت باینری (دوحالته) می‌توان شرایط زیر را برای تقسیم در نظر گرفت: $X_j < 0.5$ یا $X_j >= 0.5$ و $X_j < 1.5$ یا $X_j >= 1.5$.

اگر خصوصیت کیفی باشد، آنگاه تقسیمات بر حسب مقادیر خصوصیت صورت می‌گیرد. برای مثال، اگر X_j به معنای یک کشور باشد، می‌توان شرایط تقسیم‌بندی زیر را در نظر گرفت: $X_j \in$ {کانادا، مکزیک، آمریکا} یا $X_j \in$ {برزیل، آرژانتین}



در موردی که تعداد مقادیر زیاد باشد تعداد شرایط تقسیم‌بندی بیش از حد زیاد می‌شود بنابراین برای تسریع فرآیند ساخت درخت، تمام شرایط را در نظر نمی‌گیریم بلکه از برخی از آنها استفاده می‌کنیم (برای مثال، "کانادا" = X_j یا "کانادا" ≠ X_j).

در مورد خصوصیات ترتیبی، شرایط تقسیم شامل مقادیر ترتیبی هستند. برای مثال، اگر X_j درجه نظامی باشد، تقسیم‌بندی می‌تواند به صورت [گروهبان یکم- سرباز] $\in X_j$ یا [سرگرد- ستوان] $\in X_j$ باشد. برای خصوصیت‌های کیفی یا ترتیبی مواردی می‌تواند وجود داشته باشد (هنگامی که اندازه نمونه مشاهدات کوچک است) که مجموعه مقادیر مشاهدات خصوصیت انجام شده برای یک گره تنها قسمتی از کل بازه تعریف مقادیر خصوصیت را در بر بگیرد. در چنین مواردی لازم است که بقیه مقادیر را به یک شاخه جدید نسبت دهیم تا در پیش‌بینی نمونه کنترل که دارای چنین مقداری باشد بتوانیم تعیین کنیم که این مقدار به کدام شاخه تعلق دارد. برای مثال، ممکن است مقادیر معلوم را با توجه به بیشترین تعداد مشاهدات به شاخه‌ها نسبت دهیم [۹].

عمل تعریف درجه توافق برای شاخه‌بندی گره (قانون توقف)

یک گره آزاد (گره‌ای که شاخه‌ای از آن منشعب نشده) را در درخت در نظر بگیرید که مشخص نیست آیا این گره یک برگ است یا اینکه باید شاخه‌بندی شود. زیرمجموعه مشاهدات مربوط به این گره را در نظر بگیرید.

گره‌ها را به دو دسته تقسیم می‌کنیم؛ اول آنکه این مقادیر همگن باشند، یعنی اساسا متعلق به یک کلاس باشند (مسئله تشخیص الگو R.P) یا اینکه واریانس Y آنها به‌اندازه کافی کوچک باشد (مسئله تحلیل رگرسیون R.A). موردی که در آن مقدار خصوصیت برای تمام مشاهدات یکسان باشد نیز به این مورد مربوط می‌شود. دوم آنکه تعداد مشاهدات کافی نباشد. گره‌ای که دارای شرایط شاخه‌بندی نباشد یک برگ نامیده می‌شود.

برای تعریف درجه توافق می‌توان پارامترهایی چون خطای مجاز برای گره (مسئله PR)، واریانس مجاز (مسئله R.A) و آستانه برای مشاهدات کیفی را تعریف کرد.

الگوریتم درخت تصمیم‌گیری

در زیر یک الگوریتم پایه برای درخت تصمیم‌گیری آمده است:

Algorithm ???: Generate_decision_tree. Generate a decision tree from the training tuples of data partition *D*.

Input:

- Data partition, *D*, which is a set of training tuples and their associated class labels;
- *attribute_list*, the set of candidate attributes;
- *Attribute_selection_method*, a procedure to determine the splitting criterion that "best" partitions the data tuples into individual classes. This criteria consists of a *splitting_attribute* and, possibly, either a *split point* or *splitting subset*.

Output: A decision tree.

Method:

```
(1)  create a node N;
(2)  if tuples in D are all of the same class, C then
(3)      return N as a leaf node labelled with the class C;
(4)  if attribute_list is empty then
(5)      return N as a leaf node labelled with the majority class in D; // majority voting
(6)  apply Attribute_selection_method(D, attribute_list) to find "best" splitting_criterion;
(7)  label node N with splitting_criterion;
(8)  if splitting_attribute is discrete-valued and
         multi-way splits allowed then // not restricted to binary trees
(9)      attribute_list ← attribute_list - splitting_attribute; // remove splitting_attribute
(10) for each outcome j of splitting_criterion
     // partition the tuples and grow subtrees for each partition
(11)     let Dj be the set of data tuples in D satisfying outcome j; // a partition
(12)     if Dj is empty then
(13)         attach a leaf labelled with the majority class in D to node N;
(14)     else attach the node returned by Generate_decision_tree(Dj, attribute_list) to node N;
     endfor
```

عملکرد این الگوریتم به شرح ذیل است:

* الگوریتم با پارامترهای D، attribute_list، Attribute_selection_method فراخوانی می‌شود. D در واقع یک بخش[1] داده‌ای است. در ابتدا D شامل مجموعه آموزشی و برچسب طبقه‌های متناظر با آنهاست. attribute_list فهرستی از صفات موجود در مولفه‌هاست. Attribute_selection_method یک روال ابتکاری[2] است که بهترین صفت را برای جداکردن مولفه‌ها براساس طبقه‌ها می‌دهد. این روش از یک معیار انتخاب صفت مانند information gain یا gini index استفاده می‌کند که در ادامه شرح داده می‌شود.

* درخت در گام اول با یک گره تنهای N که کل مجموعه آموزشی داده‌ها را نشان می‌دهد، ایجاد می‌شود.

* اگر مولفه‌های D همه از یک طبقه باشند، گره N یک برگ خواهد بود و با آن طبقه برچسب می‌خورد (گام ۲ و ۳). گام ۴ و ۵ شرایط خاتمه هستند که در ادامه شرح داده می‌شوند.

* در غیر این صورت Attribute_selection_method فراخوانی می‌شود تا معیار شکاف[3] را مشخص کند. معیار شکاف مشخص می‌کند که کدام صفت باید در گره N مورد آزمون قرار گیرد. معیار شکاف همچنین بیان می‌کند که چه شاخه‌هایی باید از گره N با توجه به آزمون مربوطه، خارج شوند. به عبارت دیگر معیار شکاف، صفت یا نقطه شکاف را تعیین می‌کند. نقطه شکاف، D را به یکسری بخش تبدیل می‌کند. این بخش‌ها باید تا حد ممکن خالص[4] باشند به این معنی که همه مولفه‌های موجود در یک بخش باید مربوط به یک طبقه باشند [۱۱].

* گره N با معیار شکاف برچسب می‌خورد (گام ۷). یک شاخه از گره N به هر یک از خروجی‌های معیار شکاف می رود. مولفه‌های D متناظرا بخش‌بندی می‌شوند (گام۱۰و۱۱).

* الگوریتم فرآیند مشابهی را به صورت بازگشتی در هر یک از بخش‌های حاصل شده دنبال می‌کند (گام ۱۴).

1- Partition
2- Heuristic
3- Splitting criterion
4- pure

* بخش‌بندی بازگشتی در صورتی که یکی از شرایط زیر به وجود آید، متوقف می‌شود:
۱- اگر تمام مولفه‌ها در بخش D متعلق به یک طبقه باشند (گام ۲ و ۳).
۲- صفتی برای بخش‌بندی بیشتر وجود نداشته باشد (گام ۴). در این حالت گره N بـه یـک برگ تبدیل می‌شود و برچسب طبقه آن طبقه متداول در D خواهد بود.
۳- مولفه‌ای برای یک شاخه وجود نداشته باشد در واقع اگر یکی از بخش‌های D ماننـد D_j تهی باشد (گام ۱۲). در این موارد یک برگ با برچسب طبقه متـداول در D ایجـاد مـی‌شـود (گام ۱۳) [۱۱].

معیارهای انتخاب صفت

معیارهای مختلفی برای تعیین صفتی که شکاف باید بر اسـاس آن انجـام شـود، وجـود دارد؛ مانند: بهره اطلاعاتی[۱]، نسبت بهره[۲] و شاخص جینی[۳].
در اینجا تنها به معرفی بهره اطلاعاتی می‌پردازیم.

نتیجه‌گیری

اطلاعات مورد نیاز برای طبقه‌بندی یک مولفه در D برابر: $Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$

است که در آن p_i احتمال آن است که یک مولفه دلخواه در D متعلق به طبقه C_i باشـد کـه این احتمال به صورت $|C_{i,D}|/|D|$ تخمین زده می‌شود ($|D|$ و $|C_{i,D}|$ تعداد مولفه‌ها در D و $C_{i,D}$ را نشان می‌دهد). تعداد طبقه‌های موجود m است. در واقع (InfoD) همان آنتروپی یـا بی‌نظمی[۴] است.
فرض می‌کنیم صفت A دارای v مقدار متمایز به صورت $\{a_1,a_2,\ldots,a_v\}$ باشد یا به عبـارت دیگر A یک صفت گسـسته باشـد. اگـر D را بـر حـسب صـفت A بـشکافیم v بخـش یـا زیرمجموعه مانند $\{D_1,D_2,\ldots,D_v\}$ حاصل می‌شود که در آن D_j شامل مولفـه‌هـایی از D است که مقدار صفت A در آنها برابر a_j است. اگر فرض کنیم که D در گره‌ای چون N قرار

1- Information Gain
2- Gain Ratio
3- Gini Index
4- entropy

داشته باشد آنگاه این بخش‌ها متناظر با شاخه‌هایی هستند که از N خارج می‌شوند. اطلاعات مورد نیاز برای طبقه‌بندی یک مولفه از D بر حسب صفت A برابر:

$$Info_A(D) = \sum_{j=1}^{v} | D_j | / | D | \times Info(D_j)$$

است. عبارت $|D_j|/|D|$ در واقع وزن بخش j را نشان می‌دهد.

اطلاعات حاصل از انشعاب بر حسب صفت A را به صورت زیر تعریف می‌کنیم:

$$Gain(A) = Info(D) - Info_A(D)$$

هر چه مقدار بهره صفت A (Gain(A)) بیشتر باشد یا به عبارت دیگر هر چه ($Info_A(D)$) کمتر باشد صفت A به عنوان صفت شکاف انتخاب می‌شود [۱۱].

منابع

[1]- Komarek, P. and Moore, A, Fast Robust Logistic Regression for Large Sparse Datasets wI.Th Binary Outputs. In Artificial Intelligence and Statistics; 2003

[۲]- جوادی، رضا، شتاب در رشد اطلاعاتی. (۱۳۸۶/۱۱/۱۱)

http://www.mehR.Avid.ir/mod.php

[۳]- ناظمی، عبدالرضا. رده‌بندی در داده‌کاوی، [پایان‌نامه] جهت اخذ مدرک (کارشناسی ارشد): دانشگاه فردوسی مشهد؛ ۱۳۸۴

[۴]- پاک‌گوهر، علیرضا. کاربرد داده‌کاوی در پلیس راهنمایی و رانندگی، دوماهنامه علمی- تخصصی پلیس راهور، ۱۳۸۴؛ سال اول (شماره ۸)

[۵]- وظیفه‌دوست، علیرضا. متن کاوی (۱۳۸۶/۱۱/۱۱)

http://mI.Tm3.blogfa.com/post-219.aspx

[۶]- پاشنه طلا - مدیریت دانش و سازمان‌های سومین هزاره (۱۳۸۶/۱۱/۱۱)

http://idm.persianblog.ir

[7]- Fayyad, U.M., Djorgovski, S.G., and Weir, N; 1996

[8]- FR.Awley, W.J., Piatetsky-Shapiro, G., and Matheus, C;1991

[9]- Fayyad, U.M., Piatetsky-Shapiro, G., and Smyth;1996

[۱۰]- پاک‌گوهر، علیرضا. کاربرد آمار در داده‌کاوی با نگرش موضوعی به رگرسیون لجستیک، [پایان‌نامه] جهت اخذ مدرک(کارشناسی ارشد): دانشگاه آزاد اسلامی واحد مشهد؛ ۱۳۸۵

[۱۱]- قدیمی، یوحنا و عباسی، علی و پاشایی، کاوه. داده‌کاوی جریان داده‌ها با درخت تصمیم‌گیری. چاپ اول، تهران: نشر سما؛ ۱۳۸۴

[12]- Jiawei Hand and Micheline Kamber; 2001, Data Mining: Concepts and Techniques, Morgan Kaufmann