# ۱۰ الگوریتم از برترین های داده کاوی

سید مجتبی سالاری
دانشکده برق و رایانه، دانشگاه آزاد اسلامی قزوین
Salari.sm@gmail.com

فضل ا... ادیب نیا
عضو هیئت علمی دانشگاه سراسری یزد
Adebniya.fazlollah@gmail.com

**چکیده - داده کاوی یکی از پیشرفتهای اخیر در حوزه کامپیوتر برای اکتشاف عمقی داده هاست. داده کاوی ، اطلاعات پنهانی که برای برنامه ریزیهای استراتژیک میتواند حیاتی باشد را آشکار می سازد. این مقاله به بررسی ۱۰ الگوریتم برتر داده کاوی منتخب کنفرانس بین المللی داده کاوی *(ICDM)* می پردازد. الگوریتم های *C4.5 , k-Means, SVM, Apriori, CART , PageRank, AdaBoost, KNN, Navie beys, EM* این ۱۰ الگوریتم برتر در زمره پرقدرت ترین الگوریتم هایی هستند که در تحقیقات مورد استفاده قرار میگیرند. این الگوریتم ها حوزه های *clustering, statistical learning, association analysis, link mining* و *classification* را پوشش میدهند که همگی ازمباحث بسیار مهم در تحقیقات داده کاوی محسوب میشوند. سعی بر آن است که این الگوریتم ها توضیح داده شده ، نقاط قوت و ضعف آن ها نیز مورد بررسی قرار گیرد.**

کلمات کلیدی - داده کاوی classification, clustering , k-Means, SVM, C4.5 , PageRank, AdaBoost KNN, Apriori, CART , Navie beys,EM,

**مقدمه:**

در دنیای امروزی، اطلاعات بعنوان یکی از فاکتورهای تولیدی مهم پدیدار شده است. در نتیجه تلاش برای استخراج اطلاعات از داده ها توجه بسیاری از افراد دخیل در صنعت اطلاعات را به خود جلب نموده است.

پیشرفتهای حاصله در علم اطلاع رسانی و تکنولوژی اطلاعات، فنون و ابزارهای جدیدی را برای غلبه بر رشد مستمر و تنوع بانکهای اطلاعاتی تامین می کنند. این پیشرفتها هم در بعد سخت افزاری و هم نرم افزاری حاصل شده اند.

داده کاوی یکی از پیشرفتهای اخیر در راستای فن آوریهای مدیریت داده هاست. داده کاوی مجموعه ای از فنون است که به شخص امکان میدهد تا ورای داده پردازی معمولی حرکت کند و به استخراج اطلاعاتی که در انبوه داده ها مخفی و یا پنهان است کمک می کند. برای داده کاوی الگوریتم های بسیاری معرفی شده است ولی موضوع مورد نظر انتخاب ده الگوریتم از این تعداد الگوریتم و توصیفی مختصری از انهاست.

در نحوه انتخاب این الگوریتم ها باید گفت که ابتدا در کنفرانس بین المللی داده کاوی تلاشی صورت گرفت تا الگوریتم های مطرح و پرقدرتی که در مبحث داده کاوی مورد استفاده قرار میگیرند شناسایی شوند و در مرحله بعد از یکسری کسانی که جوایز IEEE را در تحقیقات و نوآوری کسب کرده بودند دعوت شد تا حداکثر ۱۰ الگوریتم قوی ومطرح در داده کاوی را بعنوان کاندیدا معرفی کنند در نتیجه الگوریتم ها در ده سرفصل زیر دسته بندی شدند :

Association analysis, classification,clustering, statistical, rough sets, learning, bagging and boosting, Sequential patterns, Integrated mining, link mining, graph mining.

آنچه که در این مقاله به آن می پردازیم ده الگوریتم از برترین های انتخاب شده کنفرانس بین المللی داده کاوی است.

## ۱. الگوریتم CART

### ۱- مقدمه

درخت تصمیم گیری CART[1] در سال ۱۹۸۴ بطور مشترک توسط لئو بریمن[2]، ژئوم فریدمن[3] ، ریچارد اولشن[4] و چارلز استون[5] نوشته شد که این امر یک گام مهم در زمینه پیشرفت هوش مصنوعی ، سیستم یادگیری ، آمار غیر پارامتری و استخراج داده ، را به بار آورد . CART به دلیل

سادگی مطالعه درختان تصمیم گیری ، ابتکارات فنی ارائه شده بوسیله آن ، مباحث پیچیده و پیشرفته بررسی داده های درختی شکل ، و نگرش توانمندانه آن به تئوری نمونه های وسیع برای درختان حائز اهمیت است . با اینکه CART را می توان تقریبا در هر حوزه ای یافت ولی به طور وسیع در زمینه هایی مانند مهندسی الکترونیک ، زیست شناسی ، مطالعات پزشکی و مباحث اقتصادی یافت می شود . مثلا در تحقیقات بازاریابی یا اجتماعی.

## ۲- توصیف الگوریتم

درخت تصمیم گیری CART یک روش تقسیم بندی بازگشتی باینری است. داده ها در این الگوریتم بصورت خام استفاده می شوند و هیچگونه پاکسازی نه نیاز است نه پیشنهاد می شود . درختان بدون استفاده از یک قانون متوقف کننده به رشد حداکثری خود می رسند و سپس اصلاح می شوند (که این کار قسمت به قسمت انجام می شود .) اصلاح تا ریشه ادامه دارد و با اصلاح پیچیدگی کار بالا می رود. قسمت بعدی برای اصلاح بخشی است که کمترین کمک را به کارکرد کلی درخت در پردازش اطلاعات می کند (و ممکن است در یک زمان بیشتر از یک بخش حذف شود). هدف مکانیزم CART تولید تنها یک درخت نیست بلکه تولید یک سری درختان اصلاح شده تو در توست که همه آن درختان بهینه داوطلب هستند . درخت با اندازه مناسب یا "درخت درست " به وسیله ارزش گذاری عملکرد پیشگویانه هر درخت در توالی اصلاح ، شناخته می شود . CART هیچگونه اندازه عملکرد داخلی برای انتخاب درخت براساس پردازش اطلاعات پیشنهاد نمی کند زیرا این اندازه ها قابل اطمینان نیستند . به جای آن عملکرد درخت در آزمایش داده های جداگانه (یا از طریق تائید میانه ) اندازه گیری می شود و انتخاب درخت تنها پس از ارزشیابی آزمایش داده ها صورت می گیرد . اگر هیچ آزمایش داده ای وجود نداشته باشد و تائیدیه میانی انجام نشده باشد CART نمی تواند بهترین درخت توالی را مشخص کند . این با روش هایی مانند C4.5 که مدل های برتر را بر اساس اندازه های پردازش داده ایجاد می کنند کاملا در تضاد است .

مکانیزم CART شامل متعادل سازی ردیف اتوماتیک(اختیاری) و استفاده اتوماتیک از ارزش مفقود است و اجازه یادگیری حساس به هزینه ، ساخت خصوصیات دینامیک و تخمین احتمال درخت را می دهد. CART همچنین زمینه جدیدی برای نمایش چگونگی امکان استفاده از تائیدیه میانه برای تعیین عملکرد هر درخت در توالی اصلاح موجود، ارائه کرد که نشان میدهد درختان در دسته های CV مختلف ممکن است با تعداد گره های نهایی هم تراز نباشند.

# الگوریتمApriori

## ۱- مقدمه

بسیاری از الگوریتم های الگو یاب ازقبیل الگوریتم هایی که برای ساختن درخت تصمیم گیری ؛ استنتاج قوانین دسته بندی و جمع بندی داده ها که زیاد در استخراج اطلاعات استفاده می شوند ؛ در جامعه تحقیق یادگیری ماشین گسترش یافته اند . الگوی تکرار شونده و استخراج قوانین مرتبط یکی از معدود استثنائات این شیوه است و معرفی این روش در گذشته ؛ تحقیق برای استخراج را پیش برد.

الگوریتمهای پایه ای ساده اند و به آسانی قابل اجرا هستند . از آنجاییکه استقرا ء بسیار مهم بوده و شکل مجموعه داده به بازار معامله آن بستگی دارد فعالیتهای زیادی برای بهبود کیفیت مناسبات ؛ یافتن بسته های کوچکتر ارائه و بسط انواع داده که می تواند کنترل شود انجام گرفته است.

## ۲- توصیف الگوریتم

یکی از محبوب ترین دید گاههای استخراج اطلاعات ؛ یافتن یک مجموعه ارقام تکرار شونده از یک مجموعه داده اجرایی و استخراج قوانین مرتبط است . مشکل بطور رسمی در حالت زیر ارائه شده است . اگر I={i1,i2……im} یک مجموعه از ارقام باشد و D یک مجموعه از معادله باشد که در آن هر عضو t یک مجموعه از ارقام است که $t \subseteq \mathcal{I}$ هر عضو، یک شناساگر منحصر بفرد دارد که TID نام دارد . یک عضو t شامل X که مجموعه ای از تعدادی از ارقام از مجموعه I است . در صورتیکه $X \subseteq t$ یک قانون مرتبط بمعنی $X \Rightarrow Y$ است که $X \subset \mathcal{I}, Y \subset \mathcal{I}$ و $X \cap Y = \emptyset$ قانون $X \Rightarrow Y$ در مورد D با ضریب C نیز صدق می کند . $(0 \le c \le 1)$ در صورتیکه تقسیم اعضا

شامل Y نیز هستند . با وجود تعدادی از اعضای D مشکل استخراج قوانین مرتبط ؛ همه قوانین مرتبطی که حمایت و اطمینان آنها کمتر از حداقل حمایت مخصوص استفاده کننده بنام MINSUP نیست را تولید می کند و بدنبال آن حداقل اطمینان( MINCONF) را ایجاد می کند . یافتن مجموعه ارقام وسیع ( مجموعه ارقامی با فراوانی برابر یا بیشتر از MINSUP ) ساده نیست و این بدلیل پیچیدگی محاسبات حاصل از انفجارهای ترکیبی است . وقتی یک بار مجموعه ارقام وسیع کسب شوند ؛ ایجاد قوانین مرتبط با اطمینان برابر یا بیشتر از حداقل اطمینان (MINCONF) آسان خواهد بود. Apriori و AprioriTid که توسط R.SRIKANT , R.AGRAWAL پیشنهاد شده اند ؛ الگوریتمهای اصلی هستند که برای کار در مجموعه های داده وسیع طراحی شده اند .

## الگوریتم EM

### ۱- مقدمه

الگوریتم بیشینه سازی مورد انتظار[۶] در زمینه هایی چون میانگین داده ، ماشین یادگیری و شناسایی، بسیار کاربرد داشته است . برآورد درستنمایی ماکسیمم[۷] و استنباط درست نمایی در مرکز اهمیت قضییه آماری و تجزیه و تحلیل داده هستند . براورد درستنمایی ماکسیمم یک روش همه منظوره با خصوصیات قابل توجه است .

ترکیب توابع تعمیم یافته ی متناهی یک روش انعطاف پذیر ریاضی را برای مدل سازی و دسته بندی داده هایی که به عنوان پدیده تصادفی مشاهده شده اند را ارائه می دهد. در اینجا بر استفاده ی EM جهت پردازش ترکیب توابع تعمیم یافته ی متناهی از طریق روش حداکثر احتمال تمرکز می شود.

### ۲-توصیف الگوریتم :

الگوریتم EMیک الگوریتم تکراری است که در تکرار آن دو مرحله وجود دارد ، مرحله انتظار (مرحله E) و مرحله حداکثرسازی(مرحله M). در چارچوب داده های ناقص الگوریتم EM ، $y = (y_1^T, \ldots, y_n^T)^T$ نشان دهنده بردار داده های مشاهده شده و z نشان دهنده بردار داده های ناقص است، بردار های کامل به این صورت بیان می شود :

$$x = (y^T, z^T)^T$$

الگوریتم EMبه مشکل حل عملیات اطلاعات ناقص معادله ممکن نزدیک می شود. این کار را غیر مستقیم و با پیشرفت تکراری ، با توجه به عملیات ممکن" داده های کامل " یا عملیات $L_c(\Psi)$ انجام می دهد . از آنجایی که این کار به وضوح به داده های غیر قابل مشاهده z بستگی دارد، مرحله E طوری صورت می گیرد که در آن لگاریتم $L_c(\Psi)$ توسط عملیات Q جایگزین می شود و از اندازه اخیر $\Psi$ استفاده می کند . به طور ویژه در (k+1) امین تکرار الگوریتمEM ، مرحله E به این صورت عمل می کند :

$$Q(\Psi; \Psi^{(k)}) = E_{\Psi^{(k)}}\{\log L_c(\Psi)|y\}$$

که در آن $E_{\Psi}^{(k)}$ نتظار را با استفاده از بردار واحد $\Psi^{(k)}$ نشان می دهد . مرحله M تخمین $\Psi$ را بوسیله ارزش $\Psi^{(k+1)}$ که $\Psi$ در عملیات Q ،$Q(\Psi; \Psi^{(k)})$با توجه به $\Psi$ بر فضای واحد ، به حداکثر می رساند را به روز می کند . مراحل EوM به طور متوالی جابجا می شوند تا تغییرات ارزش های الگوریتم متحمل کمتر از برخی آستانه های مشخص شده باشند. همانطور که در بخش قبل اشاره شد الگوریتم EMبرای افزایش تکرار هر یک از EMهای ارزش محتمل ثابت و بصورت زیر است :

$$L(\Psi^{(k+1)}) \geq L(\Psi^{(k)})$$

می توان نشان داد که مراحل EوM شکل های ساده ای دارند در حالیکه عملیات چگالی محتمل داده های کامل از یک خانواده نمایی است . معمولا در عمل ، راه حل مرحله M به شکل محدود وجود دارد . در آن نمونه ها که این راه حل وجود ندارد ممکن است نتوان، برای یافتن ارزش $\Psi$ که در کل عملیات $Q(\Psi; \Psi^{(k)})$ را حداکثر می کند ، تلاش کرد . برای چنین شرایطی ممکن است یک الگوریتم EM کلی (GEM) انتخاب شود که در آن مرحله M نیازمند این است که $\Psi^{(k+1)}$ طوری انتخاب شود که $\Psi^{(k+1)}$ عملیات Q ،$Q(\Psi; \Psi^{(k)})$را بر ارزش موجود در$\Psi^{(k)}$ $\Psi=$ افزایش دهد . یعنی

$$Q(\Psi^{(k+1)}; \Psi^{(k)}) \geq Q(\Psi^{(k)}; \Psi^{(k)})$$

اتفاق بیفتد . برخی موانع الگوریتم EMهستند (a)که این خود بخود اندازه ماتریس واریانس اندازه های واحد را تولید نمی کند . با این حال این شکل را می توان به سادگی با استفاده از روش درست مربوط به الگوریتم EM از بین برد. (b) گاهی اوقات همگرایی بسیار آهسته صورت می

گیرد و (c) در برخی مشکلات مراحل MgE می توانند با بررسی جدا شوند.

## الگوریتم KNN

### ۱-مقدمه

یکی از ساده ترین و به نسبت طبقه بندی کننده های بدیهی , طبقه بندی کننده Rote است که کل داده های تعلیمی را به خاطر می سپارد و طبقه بندی را فقط اگر نسبتهای موضوع آزمایش بطور دقیق با صفات یکی از موضوعات آزمایش مطابقت کرد, طبقه بندی را اجرا میکند. یک مشکل بدیهی این راه این است که بسیاری از گزارشهای آزمایش، طبقه بندی نخواهند شد زیرا آنها بطور دقیق با هیچ کدام از گزارشات آموزشی مطابقت نمیکند. موضوع دیگر, موقعی که دو یا چند گزارش آموزش دارای نسبتهای مشابه اما برچسبهای متفاوت ردیف یا سری باشند, بوجود می آید.

در ساده ترین شکل, KNN میتواند یک موضوع از ردیفی که نزدیکترین همسایگان یا اکثریت آنها در حال واگذاری هستنددرگیر کند. بطور کلی KNN یک مورد ویژه علم است که وابسته به نمونه و مثل است. این شامل منطق وابسته به مورد و نمونه است که در مورد داده های نمادین بحث میکند. همچنین به علت سادگی آن KNN برای اصلاح و تغییر مسائل پیچیده طبقه بندی شده آسان است. برای مثال, KNN بصورت ویژه برای طبقات با کیفیت چندگانه مناسب است درست بخوبی تقاضاهایی که در یک موضوع میتواند در بسیاری از طبقه بندی های این کلاس داشته باشد. برخی از محققان دریافتند که KNN یک وسیله بردار حمایتی ابزاری را پشت سر گذاشته که یک طرح طبقه بندی بسیار پیچیده و ماهرانه است.

### ۲- توصیف الگوریتم

الگوریتم ۱ یک خلاصه با سطح بالایی از روش طبقه بندی نزدیکترین مجاور را فراهم میکند. یک مجموعه آموزشی D و یک موضوع آزمایش Z داده شده است که یک بردار ارزشهای صفات است و یک طبقه بندی ردیف ناشناخته دارد. این الگوریتم فاصله یا تشابه بین Z و همه موضوعات آموزشی را برای تعیین لیست نزدیکترین مجاور به آن محاسبه میکند. سپس یک طبقه را با بردن اکثریت موضوعات مجاور ردیف به Z منتقل میکند. روابط در یک روش نامشخص شکسته می شود.

**Input** : $D$, the set of training objects, the test object, $\mathbf{z}$, which is a vector of attribute values, and $L$, the set of classes used to label the objects
**Output** : $c_z \in L$, the class of $z$
**foreach** object $\mathbf{y} \in D$ **do**
  | Compute $d(\mathbf{z}, \mathbf{y})$, the distance between $\mathbf{z}$ and $\mathbf{y}$;
**end**
Select $N \subseteq D$, the set (neighborhood) of $k$ closest training objects for $z$;
$c_z = \underset{v \in L}{\operatorname{argmax}} \sum_{r \in N} I(v = class(c_r))$;
where $I(\cdot)$ is an indicator function that returns the value 1 if its argument is true and 0 otherwise.

الگوریتم۱: الگوریتم KNN

پیچیدگی ذخیره سازی الگوریتم O(n) است که در آن n تعداد موضوعات آموزش است. زمان پیچیدگی هم O(n) است. KNN از اکثر روشهای طبقه بندی دیگر متفاوت است.

## الگوریتمNaive Bayes

### ۱- مقدمه

هدف ما در این بخش ایجاد قانونیست که قادرمان سازد اعضای بعدی را در یک دسته قرار دهیم و اینکار را تنها با داشتن بردارهایی از متغیرهای توصیف کننده اجسام بعدی می توانیم انجام دهیم. مشکلاتی از این نوع که دسته بندی نظارت شده نام دارندهمه جا وجود دارند و روشهای بسیاری برای ایجاد چنین قوانینی بوجود آمده اند. یک روش بسیار مهم روش بیز ساده است که بیز سطحی,بیز ساده, و بیز مستقل نیز نامیده میشود. این روش به دلایل متعددی اهمیت دارد. ساخت آن بسیار ساده است و نیازی به برنامه های تخمین پارامتر تکرارشونده پیچیده ندارد. یعنی میتوان از آن برای مجموعه داده های بسیار وسیع استفاده کرد و در نهایت این روش معمولا فوق العاده عمل میکند. ممکن است بهترین دسته بندی کننده ممکن ،در یک کاربرد خاص نباشد اما اغلب میتوان به قوی بودن و عملکرد عالی آن اطمینان کرد.

### ۲-قوانین اصلی

برای راحتی تفسیر در اینجا دو دسته را در نظر میگیریم که بصورت i=0 and 1 رتبه بندی میشود. هدف ما استفاده از یک مجموعه اعضای اولیه با عضویتهای دسته شناخته شده(مجموعه آموزشی) برای ایجاد یک امتیاز است بطوریکه امتیازات بالاتر با اعضای دسته ۱ مرتبطند و امتیازات

کوچکتر با اعضای دسته ۰ در ارتباطند. بنابراین دسته بندی با مقایسه این امتیازات با یک آستانه بدست می آید. اگر ما $P(i|x)$ را این احتمال تعریف کنیم که یک عضو با بردار مکان $x = (x_1, \ldots, x_p)$ به دسته i تعلق داشته باشد, هر عملکرد یکنواخت $P(i|x)$ یک امتیاز مناسب ایجاد خواهد کرد. بطور ویژه نسبت P(1|x)/ P(0|x) مناسب خواهد بود. احتمال ابتدایی به ما میگوید که $P(i|x)$ را به نسبت p(i) / f(x|i) تجزیه کنیم که در آن f(x|i) توزیع شرطی x برای اجسام دسته i است و P(i) احتمال تعلق یک عضو به دسته i است درصورتیکه ما چیز بیشتری درباره آن ندانیم (احتمال اولیه دسته i ). یعنی نسبت به این صورت در می آید:

$$\frac{P(1|x)}{P(0|x)} = \frac{f(x|1)P(1)}{f(x|0)P(0)}.$$

برای استفاده از این نسبت در دسته بندی ها ما نیاز به تخمین f(x|i) و P(i) داریم. اگر مجموعه آموزشی یک نمونه تصادفی از جمعیت کل باشد P(i) را میتوان مستقیما از جمعیت اجسام دسته i محاسبه کرد. برای محاسبه f(x|i) روش بیز ساده اینطور در نظر میگیرد که اعضای x مستقلند, $\prod_{j=1}^{p} f(x_j|i)$ سپس هر یک از توزیعات یکنواخت $f(x_j|i), j = 1, \ldots, p;$ را جداگانه اندازه گیری میکند. بنابراین مشکل چند شکلی p بعدی به مشکلات اندازه گیری یک شکلی کاهش می یابد. تخمین یک شکلی آشنا و ساده بوده و به اندازه های مجموعه یادگیری کوچکتری برای دستیابی به اندازه های درست نیازمند است. این یکی از جذابیتهای خاص و نیز منحصر به فرد روش بیز ساده است: تخمین ساده ، و بسیار سریع بوده و به برنامه های اندازه گیری تکرارشونده پیچیده نیاز ندارد.

اگر توزیع حاشیه ای $f(x_j|i)$ مجزا باشد و هر $x_j$ تعداد اندکی از ارزشها را شامل شود, اندازه $f(x_j|i)$ یک اندازه گیرنده سابقه نمای چندجمله ایست که تنها نسبت اعضای دسته i را که در یک سلول قرار میگیرند محاسبه میکند. اگر $f(x_j|i)$ ادامه داشته باشند, یک روش رایج تقسیم آنها به فواصل کوچکتر و استفاده مجدد از اندازه گیرنده چندجمله ای است اما انواع پیشرفته تر آن که بر پایه اندازه های مداوم هستند(مانند اندازه های هسته ) نیز مورد استفاده قرار میگیرند نسبت داده شده به این صورت در می آید:

$$\frac{P(1|x)}{P(0|x)} = \frac{\prod_{j=1}^{p} f(x_j|1)P(1)}{\prod_{j=1}^{p} f(x_j|0)P(0)} = \frac{P(1)}{P(0)} \prod_{j=1}^{p} \frac{f(x_j|1)}{f(x_j|0)}.$$

اکنون با یادآوری اینکه هدف ما تنها معرفی یک امتیاز بود که بطور یکنواخت به P(i|x) مرتبط باشد, میتوانیم از آن لگاریتم بگیریم.

$$\ln\frac{P(1|x)}{P(0|x)} = \ln\frac{P(1)}{P(0)} + \sum_{j=1}^{p} \ln\frac{f(x_j|1)}{f(x_j|0)}.$$

اگر ما $w_j = \ln(f(x_j|1)/f(x_j|0))$ و یک $k = \ln(P(1)/P(0))$ مداوم را تعریف کنیم میبینیم که نسبت معادله بالا شکل ساده زیر را میگیرد:

$$\ln\frac{P(1|x)}{P(0|x)} = k + \sum_{j=1}^{p} w_j,$$

بطوریکه دسته بندی ساختار ساده ای دارد.

فرض استقلال $x_j$ هر دسته مشخص در مدل بیز ساده ممکن است زیادی سخت گیرانه به نظر برسد. با این وجود در واقع فاکتورهای مختلفی ممکن است وارد بازی شوند که به این معناست که فرض آنقدر که به نظر میرسد مضر نیست. اولا, یک مرحله انتخاب متغیر اولیه معمولا اتفاق می افتد که در آن متغیرهای بسیار مرتبط با یکدیگر در زمینه هایی که ممکن بود به شباهت جدایی بین دسته ها کمک کنند رفع شدند.ثانیا, در نظر گرفتن ارتباطات به عنوان صفر یک مرحله نظم دهی مشخص فراهم میکند که باعث کاهش واریانس مدل و دسته بندی های درست تر میشود. سوما, در برخی موارد وقتی متغیرها به هم مرتبطند سطح تصمیم گیری بهینه با سطح ایجاد شده تحت فرض مستقل تصادف میکند بطوریکه ایجاد فرض به هیچ عنوان زیانبار نیست. چهارما, مسلما سطح تصمیم ایجاد شده توسط مدل بیز ساده میتواند یک شکل غیر خطی پیچیده داشته باشد.

## ۳- نکات نهایی درباره بیز ساده

این مدل یکی از قدیمی ترین الگوریتمهای دسته بندی رسمی است و هنوز حتی در ساده ترین شکل بسیار موثر است. از این مدل در دسته بندی متون و جداسازی اسپمها بطور گسترده استفاده میشود. بسیاری از اصلاحات توسط جوامع آماری ,استخراج اطلاعات,یادگیری ماشینی, و الگوشناسی در تلاش برای انعطاف بیشتر آن ارائه شده اند اما باید دانست که چنین اصلاحاتی پیچیدگی هایی را ایجاب میکنند که باعث جدایی از سادگی اولیه میشود.

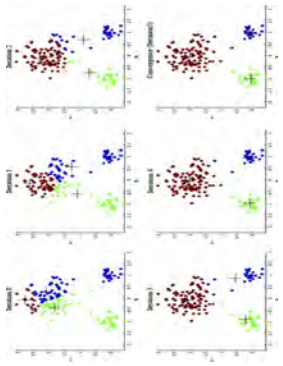
**الگوریتمk-means :**

این الگوریتم یک متد ساده تکرار شونده است، برای خوشه بندی مجموعه ای از داده های در اختیار (dataset) در تعداد مشخصی خوشه (k) که کاربر تعیین می کند. این الگوریتم توسط محققین مختلف و به صورت های مختلفی ارائه شده است. این الگوریتم بر روی مجموعه ای از بردارهای چند بعدی عمل می کند $D=\{xi|i=1,\ldots,n\}$ که xi نشان دهنده نام داده است . الگوریتم با انتخاب K نقطه از فضا به عنوان نماینده K خوشه یا مرکز ثقل شروع به کار می کند. تکنیک هایی که برای انتخاب این نقاط استفاده می شود شامل نمونه برداری تصادفی از مجموعه داده ها و تنظیم آنها به عنوان راه حلی برای خوشه بندی یک زیر مجموعه از داده ها و یا پخش کردن میانه سراسری به تعداد ۴ بار است. سپس الگوریتم بین دو مرحله تکرار می شود تا به همگرایی برسد:

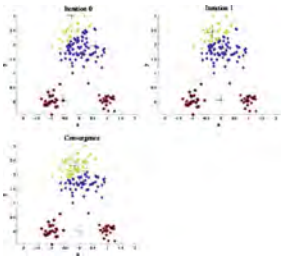
مرحله۱) انتخاب هر نمونه از داده به نزدیکترین میانه با یک رابطه قراردادی قابل شکستن .این به تقسیم بندی داده ها منتج می شود.

مرحله۲) جا به جایی میانه: اعضای هر خوشه با میانه جا به جا می شوند و با معیار سنجیده می شوند واگر وزن مورد نظر را کسب کرد این مکان یک جای قابل قبول است. الگوریتم زمانی همگرا می شود که این انتصاب ها تغییر نکند. اجرای الگوریتم به صورت نمایشی در شکل۱ آمده است. توجه کنید که هر تکرار الگوریتم N*K مقایسه احتیاج دارد که پیچیدگی زمانی هر دوره را معین می کند. تعداد تکرارها برای همگرایی مختلف و به مقدار N بستگی دارد ولی در اولین بار این الگوریتم به صورت خطی عمل می کند.

شکل ۱ جواب های نامطلوب را برای سه انتخاب مختلف میانه نشان میدهد. مساله نزدیکی محلی می تواند با اجرای چند باره الگوریتم با نقاط میانی مختلف و یا با اعمال محدودیت جستجو در مورد همگرایی، جواب داده شود.(شکل ۲)



شکل۱: جواب های نامطلوب را برای سه انتخاب مختلف



شکل۲: جواب های مطلوب را برای سه انتخاب مختلف

الگوریتمk-mean بیشترین استفاده در عمل تقسیم بندی خوشه ها رادارد. الگوریتم ساده, قابل فهم, و بطور منطقی قابل مقیاس بندی است و میتوان آنرا بسادگی اصلاح کرد تا با سناریوهای مختلف مانند یادگیری شبه مشاوره یا داده های جاری سروکار داشته باشد. پیشرفتها و کلیت های مداوم الگوریتم پایه، ارتباط مداوم آنرا تضمین میکند و به تدریج بر تاثیرگذاری آن افزوده است.

**الگوریتمAdaBoost :**

یادگیری دسته جمعی به روشهایی میگویند که از چندین یاد گیرنده برای حل یک مساله استفاده میکنند.قابلیت عمومیت بخشیدن به این روشها معمولا بطور قابل ملاحظه ای از تک یادگیرنده ها بهتر است پس آنها بسیار جذاب هستند.الگوریتم AdaBoost که توسط Robert Schapire و Yoav Freund پیشنهاد شده یکی از مهمترین روشهای یادگیری چندگانه است.چراکه از مبانی نظری قوی،محاسبات بسیار دقیق،سادگی خوبی بهره می برد . (Robert Schapire میگوید که تنها به ۱۰ خط کد نیاز دارد)

**توصیف الگوریتم:**

در توصیف الگوریتم بدین صورت فرض میکنیم که X نمونه ها را درفضا وY مجموعه کلاس ها را مشخص میکند . فرض

میکنیم که $Y = \{-1,+1\}$. یک الگوریتم پایه ای وضعیف داده شده است وهمچنین مجموعه یاد گیری بصورت $\{(x1, y1), (x2, y2), \ldots, (xm, ym)\}$ داده شده که درآن $xi \in X$ , $yi \in Y$ $(i = 1, \ldots, m)$ . الگوریتم AdaBoost بصورت زیر عمل می کند:

ابتدا وزن همه نمونه ها را برابر قرار میدهد.توزیع وزنی در T امین دوره یادگیری را با Dt مشخص میکنیم . با استفاده از مجموعه یادگیری و Dt ،الگوریتم یک یاد گیرنده ضعیف و پایه ای $ht : X \rightarrow Y$ را با فراخوانی الگوریتم یادگیری پایه ای تولید میکند.سپس با استفاده از مجموعه یادگیری به تست ht می پردازد و وزن نمونه هایی که اشتباها دسته بندی شده اند افزایش می یابد. در نتیجه وزن بروز رسانی شده (Dt+1) مشخص میشود.با استفاده از مجموعه یادگیری و Dt+1 ، AdaBoost یک یادگیرنده ضعیف دیگر را با فراخوانی الگوریتم یادگیری پایه ای تولید میکند.این رویه T بار تکرار میشود و در نهایت در طول این رویه مدل نهایی که از رای گیری اکثریت وزن داران ، T عدد یادگیر ضعیف بدست آمده، مشخص می شود.در عمل، الگوریتم پایه میتواند الگوریتمی باشد که مستقیما از نمونه های یادگیری وزن دار استفاده کند.در غیر اینصورت وزن ها میتواند بوسیله نمونه برداری از نمونه های یادگیری بر طبق توزیع وزنی آنها Dt مورد استفاده قرار بگیرد. شبه برنامه AdaBoost در الگوریتم ۲ امده است.

**Input:** Data set $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\}$;
Base learning algorithm $\mathcal{L}$;
Number of learning rounds $T$.

**Process:**
$D_1(i) = 1/m$. % Initialize the weight distribution
for $t = 1, \cdots, T$:
$h_t = \mathcal{L}(\mathcal{D}, D_t)$; % Train a weak learner $h_t$ from $\mathcal{D}$ using distribution $D_t$
$\epsilon_t = \Pr_{i \sim D_t}(h_t(x_i) \neq y_i)$; % Measure the error of $h_t$
$\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$; % Determine the weight of $h_t$
$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} \exp(-\alpha_t) & \text{if } h_t(x_i) = y_i \\ \exp(\alpha_t) & \text{if } h_t(x_i) \neq y_i \end{cases}$
$= \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$ % Update the distribution, where $Z_t$ is
% a normalization factor which enables $D_{t+1}$ be a distribution
end.
**Output:** $H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$

الگوریتم۲ : الگوریتمAdaBoost

برای اینکه بتوانیم در مسایل چند کلاسه هم از AdaBoost استفاده کنیم Freund وRobert Schapire الگوریتم AdaBoost.M1 را ارایه کردند که احتیاج داشت یادگیرنده های ضعیف بقدر کافی قدرت داشته باشند که از پس توزیع های مشکلی که در جریان پروسه AdaBoost تولید میشوند،، برآیند.یک نسخه عمومی دیگر از AdaBoost ، AdaBoost.MH است که با تجزیه کارهایی که روی چند کلاسی ها میباشد به دنباله ای از کارهای دودویی ،عمل میکند. AdaBoost برای رویارویی با مسایل Regression نیز تمهیداتی دیده است.بعد از اینکه در دهه اخیر انواع گوناگونی از AdaBoost ارایه وگسترش یافت ، روشهای تکاملی به خانواده مهمی از روشهای تجمعی تبدیل شد.

**مواجهه با الگوریتم:**

AdaBoost و دیگر مشتقات ان در حوزه های مختلف با موفقیت های زیادی بکار گرفته شده اند.بطور مثال Jones و Viola با ترکیب AdaBoost و فرایند های آبشاری از آن برای شناسایی چهره بکار برده اند.. آنها توانستند یک تشخیص دهنده صورت خیلی قوی ایجاد کنندکه برروی ماشین 466MHz وبا عکس هایی با ابعاد 288 × 384 تنها در زمان 0.067 ثانیه جوابگو باشد که ۱۵ بار سریعتر از جدیدترین تکنولوژی انروز بود.در دهه اخیر این تشخیص دهنده چهره توانست در بسیاری ازبخش های پیشرو در حوزه کامپیوتر مقبول واقع شود(مخصوصا در شناسایی چهره) .

**الگوریتم C4.5 :**

سیستم‌ها ی دسته بندی یکی از ابزارهای معمول به کاررفته در استخراج داده‌اند. این سیستم‌ها به عنوان ورودی یک مجموعه موضوعات را قبول می کنند، هر یک متعلق به یکی از تعداد دسته‌های کوچک کلاس وتوضیحاتی درباره ارزشهایش برای یک مجموعه ثابت است وخروجی یک دسته بندی کننده می تواند پیشگویی کند که یک کلاس جدید به کدام موضوع تعلق دارد.

الگوریتم C4.5 برگرفته از CLS و ID3 است و به بیان درخت تصمیم میپردازد .

**توصیف الگوریتم :**

C4.5 یک الگوریتم نیست بلکه مجموعه ای از الگوریتم هاست.

توصیف کلی چگونگی کارکرد C4.5 در الگوریتم۳ نشان داده شده است . همه روش های استنتاج درخت از گره ریشه آغاز می شوند که همه اطلاعات داده شده را ارائه می دهد و بطور بازگشتی اطلاعات را به مجموعه های کوچکتری تقسیم بندی می کند . و اینکار را بوسیله آزمایش هر نسبت در هر گروه انجام میدهد . درختان فرعی، دسته بندی های مجموعه اطلاعات اصلی را نشان می دهند که آزمایشات ارزش گذاری نسبت های مشخص شده را تکمیل می کنند . این فرآیند معمولا تا خالص سازی مجموعه ها ادامه می یابد ، یعنی همه نمونه ها در یک دسته قرار می گیرند و در این زمان رشد درخت متوقف می گردد .

**Algorithm 1.1** C4.5(D)
**Input:** an attribute-valued dataset $D$
1: Tree = {}
2: **if** $D$ is "pure" OR other stopping criteria met **then**
3: terminate
4: **end if**
5: **for all** attribute $a \in D$ **do**
6: Compute information-theoretic criteria if we split on $a$
7: **end for**
8: $a_{best}$ = Best attribute according to above computed criteria
9: Tree = Create a decision node that tests $a_{best}$ in the root
10: $D_v$ = Induced sub-datasets from $D$ based on $a_{best}$
11: **for all** $D_v$ **do**
12: $\text{Tree}_v$ = C4.5($D_v$)
13: Attach $\text{Tree}_v$ to the corresponding branch of Tree
14: **end for**
15: **return** Tree

الگوریتم۳ : الگوریتم C4.5

**تفاوت C4.5 با CART**

ساختار درخت C4.5 از چندین جهت از CART تفاوت دارد. برای مثال:

- تست در CART همیشه باینری است ولی در C4.5 دو یا چند خروجی دارد.
- CART برای تست rank از چندین شاخص استفاده می کند ولی در C4.5 از یک معیار استفاده می شود.
- هرس درختان در CART با استفاده از مدلهای پیچیده که از روش cross-validation است صورت گرفته ولی C4.5 از یک الگوریتم تک گذره استفاده می کند.

قوانین دسته بندی کننده ها:

C4.5 یک لیست از قوانین در قالب یک فرم معرفی می کند. قوانین برای هر کلاس با هم گروه بندی می شوند. یک نمونه با پیداکردن اولین قانون، به موقعیت امن میرود و اگر قانونی برای نمونه پیدا نشد به کلاس پیش فرض می رود. فایده مجموعه قوانین C4.5 در مقدار زمان CPU و حافظه مورد نیاز است.

**C 5.0**

در سال ۱۹۹۷ C4.5 بوسیله سیستم تجاری به C5.0 ارتقاء یافت. تغییرات جدید توانایی های جدید موثری را شامل شد. مقیاس پذیری و مجموعه قوانین درختان تصمیم بهبود قابل توجهی یافت. C5.0 می تواند با چند هسته ای بودن CPUیا استفاده از چند CPU باعث بهبودی سیستم کامپیوتری شود. C5.0 شامل انواع داده های جدید, مقادیر غیرکاربردی , ارزش داده هایی که به اشتباه دسته بندی شده اند و مکانیزمی برای پیش فیلترینگ ویژگی هاست. مجموعه قوانین طبقه بندی نشده, زمانی که یک نمونه دسته بندی می شود، تمام قوانین کاربردی آن یافت و اخذ می شود که این با تکرار مجموعه قوانین و پیشگویی میزان درستی آنها بهبود می یابد.

**الگوریتم Page Rank:**

Page Rank در سال ۱۹۹۸ درهفتمین کنفرانس بین المللی World Wide Web توسط Segey Brinو Larry Page ارائه شد . این الگوریتم یک الگوریتم Ranking است که از پیوند داده در وب استفاده می کند. گوگل به عنوان یک موفقیت بزرگ براساس این الگوریتم ساخته شده است. در حال حاضر تمام موتور های جستجو بر اساس این الگوریتم کار میکنند. این الگوریتم بر اساس طبیعت دموکراتیک وب با استفاده ساختار وسیع اتصالی اش به عنوان یک نماینده کیفیت یک صفحه شخصی کار می کند. Page Rank یک لینک از صفحه X به صفحه Y برقرار میکند . الگوریتم Page Rank به تعداد لینک دریافتی به یک صفحه خاص توجه دارد و همچنین صفحاتی که در نقش یک پیشنهاد هستند را تحلیل می کند.

**الگوریتم:**

بر طبق نفوذ Rank در شبکه اجتماعی اهمیت صفحه i به وسیله تعداد Page Rank همه صفحات که به صفحه i اشاره دارد تعیین می شود. ازیک صفحه ممکن است به تعداد زیادی صفحه اشاره شود که این صفحه میان تمام صفحات اشاره کننده به آن به اشتراک گذاشته می شود.

فرمول بالا در وب یک گراف (V,E) که V یک مجموعه بردار یا گره است برای مثال یک مجموعه از تمام صفحات است و

E یک مجموعه ازلبه های هدایت شده در گراف است .
PageRank صفحه i ( در تعریف زیر به صورت Pi ) به صورت زیر محاسبه می شود.

$$P(i) = \sum_{(j,i)\in E} \frac{P(j)}{O_j},$$

که Oj تعداد لینک خارجی صفحه j است . بصورت ریاضی یک سیستم با n خطا برابر با n مجهول است.از یک ماتریس می توان برای نشان دادن تمام تساویها استفاده کرد.P یک بردارستونی n بعدی است. از PageRank با مقادیری از قبیل :

$$P = (P(1), P(2), \ldots, P(n))T.$$

A ماتریس مجاورت گراف است :

$$A_{ij} = \begin{cases} \frac{1}{O_i} & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$

**PageRank-Iterate**($G$)
$P_0 \leftarrow e/n$
$k \leftarrow 1$
**repeat**
  $P_k \leftarrow (1-d)e + dA^T P_{k-1}$;
  $k \leftarrow k+1$;
**until** $\|P_k - P_{k-1}\|_1 < \varepsilon$
return $P_k$

که می توان یک سیستم n معادله ای را به صورت زیر نوشت : $P=A^TP$
که یک معادله مشخصه از eigensystem که راه حلی برای P است.
یک تکنیک مشهور ریاضی بیان می کند که از قدرت تکرار میتوان برای یافتن P استفاده کرد. با این وجود مشکل Eq است .به خاطر آن که گراف وب تمام وضعیت ها را در بر نمی گیرد.
در حقیقت Eq میتواند بر اساس زنجیره مارکوف مشتق شود. سپس بعضی از نتایج فردی از زنجیره مارکوف میتواند بکار گرفته شود. علاوه بر بحث گراف وب برای موقعیت امن معادله PageRank زیر حاصل می شود:

$$\mathbf{P} = (1-d)\mathbf{e} + d\mathbf{A}^{\mathrm{T}}\mathbf{P}.$$

که e یک بردار ستونی است که همه آن ها ۱ هستند .در زیر فرمول PageRank برای هر صفحه i نشان داده شده است:

$$P(i) = (1-d) + d\sum_{j=1}^{n} A_{ji}P(j),$$

که معادل فرمول داده شده در صفحات PageRank اصلی است.

$$P(i) = (1-d) + d\sum_{(j,i)\in E} \frac{P(j)}{O_j}.$$

پارامتر d به نام damping factor نامیده می شود که می تواند مجموعه ارزشی بین ۰ تا ۱ را دارا باشد . d= 0.85 در این جا استفاده شده است.
محاسبه مقادیر Page Rank در صفحات وب می تواند با استفاده از روش تکرار انجام گیرد که به تولید eigenvector با eigenvalue ۱ می انجامد.

## الگوریتمSVM

در کاربردهای امروزی یادگیری ماشین، ماشین بردارپشتیبان[۸] (SVM) به عنوان یکی از قویترین ودقیق ترین متدها در میان الگوریتم های معروف شناخته می شود. ماشین بردار پشتیبان، یکی از روش‌های یادگیری بانظر است که از آن برای طبقه‌بندی و رگرسیون استفاده می‌کنند.
این روش از جملهٔ روش‌های نسبتاً جدیدی است که در سال‌های اخیر کارایی خوبی نسبت به روش‌های قدیمی‌تر برای طبقه‌بندی از جمله شبکه‌های عصبی پرسپترون نشان داده است. مبنای کاری دسته‌بندی کنندهٔ SVM دسته‌بندی خطی داده‌ها است و در تقسیم خطی داده‌ها سعی بر ان است خطی انتخاب شود که حاشیه اطمینان بیشتری داشته باشد. حل معادلهٔ پیدا کردن خط بهینه برای داده‌ها به وسیله روش‌های QP که روش‌های شناخته شده‌ای در حل مسائل محدودیت‌دار هستند صورت می‌گیرد. قبل از تقسیمِ خطی برای اینکه ماشین بتواند داده‌های با پیچیدگی بالا را دسته‌بندی کند داده‌ها را باید به وسیله‌ی تابعِ phi به فضای با ابعاد خیلی بالاتر برد. برای اینکه بتوان مساله ابعاد خیلی بالا را با استفاده از این روش‌ها حل کرد از قضیه باینری لاگرانژ برای تبدیلِ مسالهٔ مینیمم‌سازی مورد نظر به فرم باینری آن که در آن به جای تابع پیچیده‌ی phi از تابعِ ساده‌تری به نامِ تابع هسته که ضرب برداری تابع phi است، استفاده می‌شود.
۲- این الگوریتم دارای مبانی نظری قوی و بی نقصی می باشد،فقط به یک دوجین نمونه احتیاج دارد وبه تعداد ابعاد مساله حساس نمی باشد. همچنین متد هایی کارآمد برای یاد گیری SVM به سرعت در حال رشد هستند.در یک فرایند یادگیری که شامل دوکلاس می باشد. هدف SVM پیدا کردن بهترین تابع برای طبقه بندی می باشد به نحوی که بتوان اعضای دو کلاس را در مجموعه داده ها از هم

تشخیص داد.معیار بهترین طبقه بندی بصورت هندسی مشخص می شود، برای مجموعه داده هایی که بصورت خطی قابل تجزیه هستند. بطورحسی آن مرزی که بصورت بخشی از فضا تعریف می شود یا همان تفکیک بین دو کلاس بوسیله hyperplane تعریف می شود. بطور هندسی مرز با کمترین فاصله بین نزدیکترین داده ونقطه ای روی hyperplane مطابقت می کند. همین تعریف هندسی به ما اجازه میدهد تا کشف کنیم که چگونه مرز ها را بیشینه کنیم ولو اینکه تعداد بیشماری hyperplane داشته باشیم وفقط تعداد کمی، شایستگی راه حل برای SVM دارند.دلیل اینکه SVM روی بزرگترین مرز برای hyperplane پافشاری می کند اینست که این قضیه قابلیت عمومیت بخشیدن به الگوریتم را بهتر تامین می کند. این نه تنها به کارایی طبقه بندی و دقت آن روی داده های آزمایشی کمک می کند ، فضا را نیز برای طبقه بندی بهتر داده های آتی مهیا می کند.

یکی از مشکلات SVM پیچیدگی محاسباتی آن است.با اینحال این مشکل بطور قابل قبولی حل شده است.یک راه حل اینست که یک مساله بهینه سازی بزرگ را به یک سری از مسایل کوچکتر تقسیم کرد که هر مساله شامل یک جفت با دقت انتخاب شده از متغیرها که مساله بطور موثر بتواند ازآنها بهره ببرد.این پروسه تا زمانی که همه این قسمتهای تجزیه شده حل شوند ادامه خواهد داشت.

**نتیجه :**

هدف در داده کاوی ، باید طراحی الگوریتم هایی باشد که دقت بالا و طیف وسیعی از داده ها را شامل شود.

در این بحث الگوریتم های زیادی مطرح هستند که ۱۰ تا از آن ها شامل k-Means, SVM, C4.5 , PageRank, AdaBoost, KNN, Apriori, CART ، EM ,Navie beys در این مقاله به صورت مختصر توضیح داده شد و می توان از آنها به صورت پایه برای ساخت و طراحی الگوریتم های دیگر استفاده کرد.

**مراجع :**

1. Agrawal R, Srikant R (1994) Fast algorithms for mining association rules. In: Proceedings of the 20th VLDB conference, pp 487–499
2. Ahmed S, Coenen F, Leng PH (2006) Tree-based partitioning of date for association rule mining. Knowl Inf Syst 10(3):315–331
3. Banerjee A, Merugu S, Dhillon I, Ghosh J (2005) Clustering with Bregman divergences. J Mach Learn
4. Bezdek JC, Chuah SK, Leep D (1986) Generalized k-nearest neighbor rules. Fuzzy Sets Syst 18(3)
5. Bloch DA, Olshen RA, Walker MG (2002) Risk estimation for classification trees. J Comput Graph Stat 11:263–288
6. Bonchi F, Lucchese C (2006) On condensed representations of constrained frequent patterns. Knowl Inf Syst 9(2):180–201
7. T. G. Dietterich. An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and 2000.
8. J. Gehrke, V. Ganti, R. Ramakrishnan, and W.-H. Loh. BOAT: Optimistic Decision Tree Construction. In Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD'99), pp. 169–180, 1999.
9. D. Kumar, N. Ramakrishnan, R. F. Helm, and M. Potts. Algorithms for Storytelling. In Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'06), pp. 604–610, Aug. 2006.
10.. H.Witten and E. Frank. Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, 2005.
11. L. Zhao, M. Zaki, and N. Ramakrishnan. BLOSOM: A Framework for Mining Arbitrary Boolean Expressions Attribute Sets. Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining , 2006.
12. A. Banerjee, S. Merugu, I. Dhillon, and J. Ghosh. "Clustering with Bregman ivergences," Journal of Machine Learning Research (JMLR), 2005.
13. S. Basu, A. Banerjee, and R. Mooney. "Semi-supervised clustering by seeding," International Conference on Machine Learning 2002.
14. C. M. Bishop. Pattern Recognition and Machine Learning (Information Science and Statistics). 2006.
15. P. S. Bradley, K. P. Bennett, and A. Demiriz. "Constrained k-means clustering," Technical Rep
[16] B. Goethals. Survey on frequent pattern mining, 2003
[17] J. Han, H. Cheng, D. Xin, and X. Yan. Frequent pattern mining: Current status and future direction. Data Mining and Knowledge Discovery, Vol. 15, No. 1, pages 55–86, 2007.

۱ Classification and Regression Trees
۲ Leo breiman
۳ Jeome friedman
۴ Richard olshen
۵ Charles stone
۶ expectation-maximization
۷ maximum likelihood (ML)
۸ Support vector machines