عنوان: **مروری بر روش ها و رویکردهای مختلف در یادگیری مدل**
نویسنده: محمد رجبی
تاریخ: ۱۳۹۳/۰۹/۰۹ ۱۷:۳۵
آدرس: www.dotnettips.info
گروه‌ها: Analysis Services, data mining, Microsoft SQL Server

مقدمه

همان گونه که اشاره شد در روش‌های با ناظر (برای مثال الگوریتم‌های دسته بندی) کل مجموعه داده‌ها به دو بخش مجموعه داده‌های آموزشی و مجموعه داده‌های آزمایشی تقسیم می‌شود. در مرحله یادگیری (آموزش) مدل، الگوریتم براساس مجموعه داده‌های آموزشی یک مدل می‌سازد که شکل مدل ساخته شده به الگوریتم یادگیرنده مورد استفاده بستگی دارد. در مرحله ارزیابی براساس مجموعه داده‌های آزمایشی دقت و کارائی مدل ساخته شده بررسی می‌شود. توجه داشته باشید که مجموعه داده‌های آزمایشی برای مدل ساخته شده پیش از این ناشناخته هستند.

در مرحله یادگیری مدل؛ برای مقابله با مشکل به خاطرسپاری (Memorization) مجموعه داده‌های آموزشی، در برخی موارد بخشی از مجموعه داده‌های آموزشی را از آن مجموعه جدا می‌کنند که با عنوان مجموعه داده ارزیابی (Valid Dataset) شناسائی می‌شود. استفاده از مجموعه داده ارزیابی باعث می‌شود که مدل ساخته شده، مجموعه داده‌های آموزشی را حقیقتاً یاد بگیرد و در پی به خاطرسپاری و حفظ آن نباشد. به بیان دیگر در مرحله یادگیری مدل؛ تا قبل از رسیدن به لحظه ای، مدل در حال یادگیری و کلی سازی (Generalization) است و از آن لحظه به بعد در حال به خاطرسپاری (Over Fitting) مجموعه داده‌های آموزشی است. بدیهی است به خاطرسپاری باعث افزایش دقت مدل برای مجموعه داده‌های آموزشی و بطور مشابه باعث کاهش دقت مدل برای مجموعه داده‌های آزمایشی می‌شود. بدین منظور جهت جلوگیری از مشکل به خاطرسپاری از مجموعه داده ارزیابی استفاده می‌شود که به شکل غیر مستقیم در فرآیند یادگیری مدل، وارد عمل می‌شوند. بدین ترتیب مدلی که مفهومی را از داده‌های آموزشی فرا گرفته، نسبت به مدلی که صرفاً داده‌های آموزشی را به خوبی حفظ کرده است، برای مجموعه داده آزمایشی دقت به مراتب بالاتری دارد. این حقیقت در بیشتر فرآیندهای آموزشی که از مجموعه داده ارزیابی بهره می‌گیرند قابل مشاهده است.

در روش‌های بدون ناظر یا روش‌های توصیفی (برای مثال خوشه بندی) الگوریتم‌ها فاقد مراحل آموزشی و آزمایشی هستند و در پایان عملیات یادگیری مدل، مدل ساخته شده به همراه کارائی آن به عنوان خروجی ارائه می‌شود، برای مثال در الگوریتم‌های خوشه بندی خروجی همان خوشه‌های ایجاد شده هستند و یا خروجی در روش کشف قوانین انجمنی عبارت است از مجموعه ای از قوانین «اگر- آنگاه» که بیانگر ارتباط میان رخداد توامان مجموعه ای از اشیاء با یکدیگر می‌باشد.

در این قسمت عملیات ساخت مدل در فرآیند داده کاوی برای سه روش دسته بندی، خوشه بندی و کشف قوانین انجمنی ارائه می‌شود. بدیهی است برای هر کدام از این روش‌ها علاوه بر الگوریتم‌های معرفی شده، الگوریتم‌های متنوعی دیگری نیز وجود دارد. در ادامه سعی می‌شود به صورت کلان به فلسفه یادگیری مدل پرداخته شود. فهرست مطالب به شرح زیر است:

-1 دسته بندی:

-1-1 دسته بندی مبتنی بر درخت تصمیم (Decision Tree based methods) :

-1-2 دسته بندهای مبتنی بر قانون (Rule based methods) :

-1-3 دسته بندهای مبتنی بر نظریه بیز (Naïve Bayes and Bayesian belief networks) :

-2 خوشه بندی:

-2-1 خوشه بندی افرازی (Centroid Based Clustering) :

-2-1-1 الگوریتم خوشه بندی K-Means :

-2-1-2 الگوریتم خوشه بندی K-Medoids :

-2-1-3 الگوریتم خوشه بندی Bisecting K-Means :

-2-1-4 الگوریتم خوشه بندی Fuzzy C-Means :

-2-2 خوشه بندی سلسله مراتبی (Hierarchical Clustering) Connectivity Based Clustering :

-2-2-1 روش‌های خوشه بندی تجمیعی (Agglomerative Clustering) :

-2-2-2 روش‌های خوشه بندی تقسیمی (Divisive Clustering) :

-2-3 خوشه بندی مبتنی بر چگالی (Density Based Clustering) :

-3 کشف قوانین انجمنی :

-3-1 الگوریتم های Brute-Force ، Apriori و FP-Growth:

1- دسته بندی:

در الگوریتم‌های دسته بندی، برای هر یک از رکوردهای مجموعه داده مورد کاوش، یک برچسب که بیانگر حقیقتی از مساله است تعریف می‌شود و هدف الگوریتم یادگیری؛ یافتن نظم حاکم بر این برچسب هاست. به بیان دیگر در مرحله آموزش؛ مجموعه داده‌های آموزشی به یکی از الگوریتم‌های دسته بندی داده می‌شود تا بر اساس سایر ویژگی‌ها برای مقادیر ویژگی دسته، مدل ساخته شود. سپس در مرحله ارزیابی؛ دقت مدل ساخته شده به کمک مجموعه داده‌های آزمایشی ارزیابی خواهد شد. انواع گوناگون الگوریتم‌های دسته بندی را می‌توان بصورت ذیل برشمرد:

1-1- دسته بندی مبتنی بر درخت تصمیم (Decision Tree based methods):

از مشهورترین روش‌های ساخت مدل دسته بندی می‌باشد که دانش خروجی را به صورت یک درخت از حالات مختلف مقادیر ویژگی‌ها ارائه می‌کند. بدین ترتیب دسته بندی‌های مبتنی بر درخت تصمیم کاملاً قابل تفسیر می‌باشند. در حالت کلی درخت تصمیم بدست آمده برای یک مجموعه داده آموزشی؛ واحد و یکتا نیست. به بیان دیگر براساس یک مجموعه داده، درخت‌های تصمیم مختلفی می‌توان بدست آورد. عموماً به منظور فراهم نمودن اطلاعات بیشتری از داده ها، از میان ویژگی‌های موجود یک Case ابتدا آنهایی که دارای خاصیت جداکنندگی بیشتری هستند انتخاب می‌شوند. در واقع براساس مجموعه داده‌های آموزشی از میان ویژگی ها، یک ویژگی انتخاب می‌شود و در ادامه مجموعه رکوردها براساس مقدار این ویژگی شکسته می‌شود و این فرآیند ادامه می‌یابد تا درخت کلی ساخته شود. پس از ساخته شدن مدل، می‌توان آن را بر روی مجموعه داده‌های آزمایشی اعمال (Apply) نمود. منظور از اعمال کردن مدل، پیش بینی مقدار ویژگی یک دسته برای یک رکورد آزمایشی براساس مدل ساخته شده است. توجه شود هدف پیش بینی ویژگی دسته این رکورد، براساس درخت تصمیم موجود است.

بطور کلی الگوریتم‌های تولید درخت تصمیم مختلفی از جمله CART، ID3، C4.5، SLIQ، SPRINT و HUNT وجود دارد. این الگوریتم‌ها به لحاظ استفاده از روش‌های مختلف جهت انتخاب ویژگی و شرط توقف در ساخت درخت با یکدیگر تفاوت دارند. عموماً الگوریتم‌های درخت تصمیم برای شناسائی بهترین شکست، از یک مکانیزم حریصانه (Greedy) استفاده می‌کنند که براساس آن شکستی که توزیع دسته‌ها در گره‌های حاصل از آن همگن باشد، نسبت به سایر شکست‌ها بهتر خواهد بود. منظور از همگن بودن گره این است که همه رکوردهای موجود در آن متعلق به یک دسته خاص باشند، بدین ترتیب آن گره به برگ تبدیل خواهد شد. بنابراین گره همگن گره ای است که کمترین میزان ناخالصی (Impurity) را دارد. به بیان دیگر هر چه توزیع دسته‌ها در یک گره همگن‌تر باشد، آن گره ناخالصی کمتری خواهد داشت. سه روش مهم برای محاسبه ناخالصی گره وجود دارد که عبارتند از: ضریب GINI، روش Entropy و Classification Error.

از مزایای درخت تصمیم می‌توان به توانایی کار با داده‌های گسسته و پیوسته، سهولت در توصیف شرایط (با استفاده از منطق بولی) در درخت تصمیم، عدم نیاز به تابع تخمین توزیع، کشف روابط غیرمنتظره یا نامعلوم و ... اشاره نمود.

همچنین از معایب درخت تصمیم نسبت به دیگر روش‌های داده کاوی می‌توان این موارد را برشمرد: تولید درخت تصمیم گیری هزینه بالائی دارد، در صورت همپوشانی گره‌ها تعداد گره‌های پایانی زیاد می‌شود، طراحی درخت تصمیم گیری بهینه دشوار است، احتمال تولید روابط نادرست وجود دارد و ... .

می‌توان موارد استفاده از دسته بند درخت تصمیم نسبت به سایر دسته بندی کننده‌های تک مرحله ای رایج را؛ حذف محاسبات غیر ضروری و انعطاف پذیری در انتخاب زیر مجموعه‌های مختلفی از صفات برشمرد. در نهایت از جمله مسائل مناسب برای یادگیری درخت تصمیم، می‌توان به مسائلی که در آنها نمونه‌ها به شکل جفت‌های «صفت-مقدار» بازنمائی می‌شود و همچنین مسائلی که تابع هدف، مقادیر خروجی گسسته دارد اشاره نمود.

1-2- دسته بندهای مبتنی بر قانون (Rule based methods):

این دسته بندها دانش خروجی خود را به صورت یک مجموعه از قوانین «اگر-آنگاه» نشان می‌دهند. هر قانون یک بخش شرایط (LHS: Left Hand Side) و یک بخش نتیجه (RHS: Right Hand Side) دارد. بدیهی است اگر تمام شرایط مربوط به بخش مقدم یک قانون درباره یک رکورد خاص درست تعبیر شود، آن قانون آن رکورد را پوشش می‌دهد. دو معیار *Accuracy* و *Coverage* برای هر قانون قابل محاسبه است که هر چه میزان این دو معیار برای یک قانون بیشتر باشد، آن قانون؛ قانونی با ارزش‌تر محسوب می‌شود.

Coverage یک قانون، برابر با درصد رکوردهایی است که بخش شرایط قانون مورد نظر در مورد آنها صدق می‌کند و درست تعبیر می‌شود. بنابراین هر چه این مقدار بیشتر باشد آن قانون، قانونی کلی‌تر و عمومی‌تر می‌باشد.

Accuracy یک قانون بیان می‌کند که در میان رکوردهایی که بخش شرایط قانون در مورد آنها صدق می‌کند، چند درصد هر دو قسمت قانون مورد نظر در مورد آنها صحیح است.

چنانچه مجموعه همه رکوردها را در نظر بگیریم؛ مطلوب‌ترین حالت این است که همواره یک رکورد توسط یک و تنها یک قانون

پوشش داده شود، به بیان دیگر مجموعه قوانین نهایی به صورت جامع (Exhaustive Rules) و دو به دو ناسازگار (Mutually Exclusive Rules) باشند. جامع بودن به معنای این است که هر رکورد *حداقل* توسط یک قانون پوشش داده شود و معنای قوانین مستقل یا دو به دو ناسازگار بودن بدین معناست که هر رکورد *حداکثر* توسط یک قانون پوشش داده شود.
مجموعه قوانین و درخت تصمیم عیناً یک مجموعه دانش را نشان می‌دهند و تنها در شکل نمایش متفاوت از هم هستند. البته روش‌های مبتنی بر قانون انعطاف پذیری و تفسیرپذیری بالاتری نسبت به روش‌های مبتنی بر درخت دارند. همچنین اجباری در تعیین وضعیت هایی که در یک درخت تصمیم برای ترکیب مقادیر مختلف ویژگی‌ها رخ می‌دهد ندارند و از این رو دانش خلاصه‌تری ارائه می‌دهند.

-1-3 دسته بندهای مبتنی بر نظریه بیز (Naïve Bayes and Bayesian belief networks):
دسته بند مبتنی بر رابطه نظریه بیز (Naïve Bayes) از یک چهارچوب احتمالی برای حل مسائل دسته بندی استفاده می‌کند. براساس نظریه بیز رابطه **I** برقرار است:

$$P(C|A) = \frac{P(A|C)*P(C)}{P(A)} : \mathrm{I}$$

$$\mathrm{P}\ (\mathrm{C}|\ \mathrm{A}_1, \mathrm{A}_2, \mathrm{A}_3, \dots, \mathrm{A}_n) : \mathrm{II}$$

هدف محاسبه دسته یک رکورد مفروض با مجموعه ویژگی‌های (A1,A2,A3,...,An) می‌باشد. در واقع از بین دسته‌های موجود به دنبال پیدا کردن دسته ای هستیم که مقدار **II** را بیشینه کند. برای این منظور این احتمال را برای تمامی دسته‌های مذکور محاسبه نموده و دسته ای که مقدار این احتمال به ازای آن بیشینه شود را به عنوان دسته رکورد جدید در نظر می‌گیریم. ذکر این نکته ضروری است که بدانیم نحوه محاسبه برای ویژگی‌های گسسته و پیوسته متفاوت می‌باشد.

-2 خوشه بندی:
خوشه را مجموعه ای از داده‌ها که به هم شباهت دارند تعریف می‌کنند و هدف از انجام عملیات خوشه بندی فهم (Understanding) گروه رکوردهای مشابه در مجموعه داده‌ها و همچنین خلاصه سازی (Summarization) یا کاهش اندازه‌ی مجموعه داده‌های بزرگ می‌باشد. خوشه بندی از جمله روش هایی است که در آن هیچ گونه برچسبی برای رکوردها در نظر گرفته نمی‌شود و رکوردها تنها براساس معیار شباهتی که معرفی شده است، به مجموعه ای از خوشه‌ها گروه بندی می‌شوند. عدم استفاده از برچسب موجب می‌شود الگوریتم‌های خوشه بندی جزء روش‌های بدون ناظر محسوب شوند و همانگونه که پیشتر ذکر آن رفت در خوشه بندی تلاش می‌شود تا داده‌ها به خوشه هایی تقسیم شوند که شباهت بین داده ای درون هر خوشه بیشینه و بطور مشابه شباهت بین داده‌ها در خوشه‌های متفاوت کمینه شود.
چنانچه بخواهیم خوشه بندی و دسته بندی را مقایسه کنیم، می‌توان بیان نمود که در دسته بندی هر داده به یک دسته (طبقه) از پیش مشخص شده تخصیص می‌یابد ولی در خوشه بندی هیچ اطلاعی از خوشه‌ها وجود ندارد و به عبارتی خود خوشه‌ها نیز از داده‌ها استخراج می‌شوند. به بیان دیگر در دسته بندی مفهوم دسته در یک حقیقت خارجی نهفته است حال آنکه مفهوم خوشه در نهان فواصل میان رکورد هاست. مشهورترین تقسیم بندی الگوریتم‌های خوشه بندی به شرح زیر است:

-1-2 خوشه بندی افرازی (Centroid Based Clustering) :
تقسیم مجموعه داده‌ها به زیرمجموعه‌های بدون همپوشانی، به طریقی که هر داده دقیقاً در یک زیر مجموعه قرار داشته باشد. این الگوریتم‌ها بهترین عملکرد را برای مسائل با خوشه‌های به خوبی جدا شده از خود نشان می‌دهند. از الگوریتم‌های افرازی می‌توان به موارد زیر اشاره نمود:

2-1-1- الگوریتم خوشه بندی K-Means :

در این الگوریتم عملاً مجموعه داده‌ها به تعداد خوشه‌های از پیش تعیین شده تقسیم می‌شوند. در واقع فرض می‌شود که تعداد خوشه‌ها از ابتدا مشخص می‌باشند. ایده اصلی در این الگوریتم تعریف K مرکز برای هر یک از خوشه‌ها است. بهترین انتخاب برای مراکز خوشه‌ها قرار دادن آنها (مراکز) در فاصله هر چه بیشتر از یکدیگر می‌باشد. پس از آن هر رکورد در مجموعه داده به نزدیکترین مرکز خوشه تخصیص می‌یابد. معیار محاسبه فاصله در این مرحله هر معیاری می‌تواند باشد. این معیار با ماهیت مجموعه داده ارتباط تنگاتنگی دارد. مشهورترین معیارهای محاسبه فاصله رکوردها در روش خوشه بندی معیار فاصله اقلیدسی و فاصله همینگ می‌باشد. لازم به ذکر است در وضعیتی که انتخاب مراکز اولیه خوشه‌ها به درستی انجام نشود، خوشه‌های حاصل در پایان اجرای الگوریتم کیفیت مناسبی نخواهند داشت. بدین ترتیب در این الگوریتم جواب نهائی به انتخاب مراکز اولیه خوشه‌ها وابستگی زیادی دارد که این الگوریتم فاقد روالی مشخص برای محاسبه این مراکز می‌باشد. امکان تولید خوشه‌های خالی توسط این الگوریتم از دیگر معایب آن می‌باشد.

2-1-2- الگوریتم خوشه بندی K-Medoids :

این الگوریتم برای حل برخی مشکلات الگوریتم K-Means پیشنهاد شده است، که در آن بجای کمینه نمودن مجموع مجذور اقلیدسی فاصله بین نقاط (که معمولاً به عنوان تابع هدف در الگوریتم K-Means مورد استفاده قرار می‌گیرد)، مجموع تفاوت‌های فواصل جفت نقاط را کمینه می‌کنند. همچنین بجای میانگین گیری برای یافتن مراکز جدید در هر تکرار حلقه یادگیری مدل، از میانه مجموعه اعضای هر خوشه استفاده می‌کنند.

2-1-3- الگوریتم خوشه بندی Bisecting K-Means :

ایده اصلی در این الگوریتم بدین شرح است که برای بدست آوردن K خوشه، ابتدا کل نقاط را به شکل یک خوشه در نظر می‌گیریم و در ادامه مجموعه نقاط تنها خوشه موجود را به دو خوشه تقسیم می‌کنیم. پس از آن یکی از خوشه‌های بدست آمده را برای شکسته شدن انتخاب می‌کنیم و تا زمانی که K خوشه را بدست آوریم این روال را ادامه می‌دهیم. بدین ترتیب مشکل انتخاب نقاط ابتدایی را که در الگوریتم K-Means با آن مواجه بودیم نداشته و بسیار کاراتر از آن می‌باشد.

2-1-4- الگوریتم خوشه بندی Fuzzy C-Means:

کارائی این الگوریتم نسبت به الگوریتم K-Means کاملاً بالاتر می‌باشد و دلیل آن به نوع نگاهی است که این الگوریتم به مفهوم خوشه و اعضای آن دارد. در واقع نقطه قوت الگوریتم Fuzzy C-Means این است که الگوریتمی همواره همگراست. در این الگوریتم تعداد خوشه‌ها برابر با C بوده (مشابه الگوریتم K-Means) ولی برخلاف الگوریتم K-Means که در آن هر رکورد تنها به یکی از خوشه‌های موجود تعلق دارد، در این الگوریتم هر کدام از رکوردهای مجموعه داده به تمامی خوشه‌ها متعلق است. البته این میزان تعلق با توجه به عددی که درجه عضویت تعلق هر رکورد را نشان می‌دهد، مشخص می‌شود. بدین ترتیب عملاً تعلق فازی هر رکورد به تمامی خوشه‌ها سبب خواهد شد که امکان حرکت ملایم عضویت هر رکورد به خوشه‌های مختلف امکان پذیر شود. بنابراین در این الگوریتم امکان تصحیح خطای تخصیص ناصحیح رکوردها به خوشه‌ها ساده‌تر می‌باشد و مهم‌ترین نقطه ضعف این الگوریتم در قیاس با K-Means زمان محاسبات بیشتر آن می‌باشد. می‌توان پذیرفت که از سرعت در عملیات خوشه بندی در برابر رسیدن به دقت بالاتر می‌توان صرفه نظر نمود.

2-2- خوشه بندی سلسله مراتبی (Hierarchical Clustering) Connectivity Based Clustering:

در پایان این عملیات یک مجموعه از خوشه‌های تودرتو به شکل سلسله مراتبی و در قالب ساختار درختی خوشه بندی بدست می‌آید که با استفاده از نمودار Dendrogram چگونگی شکل گیری خوشه‌های تودرتو را می‌توان نمایش داد. این نمودار درخت مانند، ترتیبی از ادغام و تجزیه را برای خوشه‌های تشکیل شده ثبت می‌کند، یکی از نقاط قوت این روش عدم اجبار برای تعیین تعداد خوشه‌ها می‌باشد (بر خلاف خوشه بندی افرازی). الگوریتم‌های مبتنی بر خوشه بندی سلسله مراتبی به دو دسته مهم تقسیم بندی می‌شوند:

2-2-1- روش‌های خوشه بندی تجمیعی (Agglomerative Clustering) :

با نقاطی به عنوان خوشه‌های منحصر به فرد کار را آغاز نموده و در هر مرحله، به ادغام خوشه‌های نزدیک به یکدیگر می‌پردازیم، تا زمانی که تنها یک خوشه باقی بماند.

عملیات کلیدی در این روش، چگونگی محاسبه میزان مجاورت دو خوشه است و روش‌های متفاوت تعریف فاصله بین خوشه‌ها باعث تمایز الگوریتم‌های مختلف مبتنی بر ایده خوشه بندی تجمیعی است. برخی از این الگوریتم‌ها عبارتند از: خوشه بندی تجمیعی

- کمینه ای، خوشه بندی تجمیعی - بیشینه ای، خوشه بندی تجمیعی - میانگینی، خوشه بندی تجمیعی - مرکزی.

-2-2-2 روش های خوشه بندی تقسیمی (Divisive Clustering) :
با یک خوشه‌ی دربرگیرنده‌ی همه نقاط کار را آغاز نموده و در هر مرحله، خوشه را می‌شکنیم تا زمانی که K خوشه بدست آید و یا در هر خوشه یک نقطه باقی بماند.

-2-3 خوشه بندی مبتنی بر چگالی (Density Based Clustering):
تقسیم مجموعه داده به زیرمجموعه هایی که چگالی و چگونگی توزیع رکوردها در آنها لحاظ می‌شود. در این الگوریتم مهمترین فاکتور که جهت تشکیل خوشه‌ها در نظر گرفته می‌شود، تراکم و یا چگالی نقاط می‌باشد. بنابراین برخلاف دیگر روش‌های خوشه بندی که در آنها تراکم نقاط اهمیت نداشت، در این الگوریتم سعی می‌شود تنوع فاصله هایی که نقاط با یکدیگر دارند، در عملیات خوشه بندی مورد توجه قرار گیرد. الگوریتم **DBSCAN** مشهورترین الگوریتم خوشه بندی مبتنی بر چگالی است.

به طور کلی عملکرد یک الگوریتم خوشه بندی نسبت به الگوریتم‌های دیگر، بستگی کاملی به ماهیت مجموعه داده و معنای آن دارد.

-3 کشف قوانین انجمنی :
الگوریتم‌های کاشف قوانین انجمنی نیز همانند الگوریتم‌های خوشه بندی به صورت روش‌های توصیفی یا بدون ناظر طبقه بندی می‌شوند. در این الگوریتم‌ها بدنبال پیدا کردن یک مجموعه از قوانین وابستگی یا انجمنی در میان تراکنش‌ها (برای مثال تراکنشهای خرید در فروشگاه، تراکنشهای خرید و فروش سهام در بورس و ...) هستیم تا براساس قوانین کشف شده بتوان میزان اثرگذاری اشیایی را بر وجود مجموعه اشیاء دیگری بدست آورد. خروجی در این روش کاوش، به صورت مجموعه ای از قوانین «اگر-آنگاه» است، که بیانگر ارتباطات میان رخداد توامان مجموعه ای از اشیاء با یکدیگر می‌باشد. به بیان دیگر این قوانین می‌تواند به پیش بینی وقوع یک مجموعه اشیاء مشخص در یک تراکنش، براساس وقوع اشیاء دیگر موجود در آن تراکنش بپردازد. ذکر این نکته ضروری است که بدانیم قوانین استخراج شده تنها استلزام یک ارتباط میان وقوع توامان مجموعه ای از اشیاء را نشان می‌دهد و در مورد چرایی یا همان علیت این ارتباط سخنی به میان نمی‌آورد. در ادامه به معرفی مجموعه ای از تعاریف اولیه در این مبحث می‌پردازیم (در تمامی تعاریف تراکنش‌های سبد خرید مشتریان در یک فروشگاه را به عنوان مجموعه داده مورد کاوش در نظر بگیرید):
• **مجموعه اشیاء:** مجموعه ای از یک یا چند شیء. منظور از مجموعه اشیاء K عضوی، مجموعه ای است که شامل K شیء باشد.
برای مثال:{مسواک، نان، شیر}
• **تعداد پشتیبانی (Support Count) :** فراوانی وقوع مجموعه‌ی اشیاء در تراکنش‌های موجود که آنرا با حرف **σ** نشان می‌دهیم.
برای مثال: 2=({مسواک، نان، شیر})σ
• **مجموعه اشیاء مکرر (Frequent Item Set) :** مجموعه ای از اشیاء که تعداد پشتیبانی آنها بزرگتر یا مساوی یک مقدار آستانه (Min Support Threshold) باشد، مجموعه اشیاء مکرر نامیده می‌شود.
• **قوانین انجمنی:** بیان کننده ارتباط میان اشیاء در یک مجموعه از اشیاء مکرر. این قوانین معمولاً به شکل X=>Y هستند.
برای مثال:{نوشابه}=>{مسواک، شیر}

مهمترین معیارهای ارزیابی قوانین انجمنی عبارتند از:
• **Support:** کسری از تراکنش‌ها که حاوی همه اشیاء یک مجموعه اشیاء خاص هستند و آنرا با حرف S نشان می‌دهند.
برای مثال: 2.2=({نان، شیر})S
• **Confidence:** کسری از تراکنش‌های حاوی همه اشیاء بخش شرطی قانون انجمنی که صحت آن قانون را نشان می‌دهد که با آنرا حرف C نشان می‌دهند. برخلاف Support نمی‌توانیم مثالی برای اندازه گیری Confidence یک مجموعه اشیاء بیاوریم زیرا این معیار تنها برای قوانین انجمنی قابل محاسبه است.

با در نظر گرفتن قانون X=>Y می‌توان Support را کسری از تراکنش هایی دانست که شامل هر دو مورد X و Y هستند و Confidence برابر با اینکه چه کسری از تراکنش هایی که Y را شامل می‌شوند در تراکنش هایی که شامل X نیز هستند، ظاهر می‌شوند. هدف از کاوش قوانین انجمنی پیدا کردن تمام قوانین Rx است که از این دستورات تبعیت می‌کند:

$$\text{I} \qquad \text{Support}\,(R_x) \geq \text{Supp}_{MIN}$$

$$\text{II} \qquad \text{Confidence}\,(R_x) \geq \text{Conf}_{MIN}$$

$$\text{III} \qquad 3^d\text{-}2^{d+1}\text{+}1$$

در این دستورات منظور از SuppMIN و ConfMIN به ترتیب عبارت است از کمترین مقدار برای Support و Confidence که بایست جهت قبول هر پاسخ نهائی به عنوان یک قانون با ارزش مورد توجه قرار گیرد. کلیه قوانینی که از مجموعه اشیاء مکرر یکسان ایجاد می‌شوند دارای مقدار Support مشابه هستند که دقیقاً برابر با تعداد پشتیبانی یا همان σ شیء مکرری است که قوانین انجمنی با توجه به آن تولید شده اند. به همین دلیل فرآیند کشف قوانین انجمنی را می‌توان به دو مرحله مستقل «تولید مجموعه اشیاء مکرر» و «تولید قوانین انجمنی مطمئن» تقسیم نمائیم.

در مرحله نخست، تمام مجموعه اشیاء که دارای مقدار Support ≥ SuppMIN می‌باشند را تولید می‌کنیم. رابطه **I**

در مرحله دوم با توجه به مجموعه اشیاء مکرر تولید شده، قوانین انجمنی با اطمینان بالا بدست می‌آیند که همگی دارای شرط Confidence ≥ ConfMIN هستند. رابطه **II**

3-1- الگوریتم های Brute-Force ، Apriori و FP-Growth:

یک روش تولید اشیاء مکرر روش **Brute-Force** است که در آن ابتدا تمام قوانین انجمنی ممکن لیست شده، سپس مقادیر Support و Confidence برای هر قانون محاسبه می‌شود. در نهایت قوانینی که از مقادیر آستانه‌ی SuppMIN و ConfMIN تبعیت نکنند، حذف می‌شوند. تولید مجموعه اشیاء مکرر بدین طریق کاری بسیار پرهزینه و پیچیده ای می‌باشد، در واقع روش‌های هوشمندانه دیگری وجود دارد که پیچیدگی بالای روش Brute-Force را ندارند زیرا کل شبکه مجموعه اشیاء را به عنوان کاندید در نظر نمی‌گیرند. همانند تولید مجموعه اشیاء مکرر، تولید مجموعه قوانین انجمنی نیز بسیار پرهزینه و گران است.

چنانچه یک مجموعه اشیاء مکرر مشخص با d شیء را در نظر بگیریم، تعداد کل قوانین انجمنی قابل استخراج از رابطه **III** محاسبه می‌شود. (برای مثال تعداد قوانین انجمنی قابل استخراج از یک مجموعه شیء 6 عضوی برابر با 602 قانون می‌باشد، که با توجه به رشد d؛ سرعت رشد تعداد قوانین انجمنی بسیار بالا می‌باشد.)

الگوریتم‌های متعددی برای تولید مجموعه اشیاء مکرر وجود دارد برای نمونه الگوریتم‌های **Apriori** و **FP-Growth** که در هر دوی این الگوریتم ها، ورودی الگوریتم لیست تراکنش‌ها و پارامتر SuppMIN می‌باشد. الگوریتم Apriori روشی هوشمندانه برای یافتن مجموعه اشیاء تکرار شونده با استفاده از روش تولید کاندید است که از یک روش بازگشتی برای یافتن مجموعه اشیاء مکرر استفاده می‌کند. مهمترین هدف این الگوریتم تعیین مجموعه اشیاء مکرری است که تعداد تکرار آنها حداقل برابر با SuppMIN باشد. ایده اصلی در الگوریتم Apriori این است که اگر مجموعه اشیایی مکرر باشد، آنگاه تمام زیر مجموعه‌های آن مجموعه اشیاء نیز باید مکرر باشند. در واقع این اصل همواره برقرار است زیرا Support یک مجموعه شیء هرگز بیشتر از Support زیرمجموعه‌های آن مجموعه شیء نخواهد بود. مطابق با این ایده تمام ابرمجموعه‌های مربوط به مجموعه شیء نامکرر از شبکه مجموعه اشیاء حذف خواهند شد (هرس می‌شوند). هرس کردن مبتنی بر این ایده را هرس کردن بر پایه Support نیز عنوان می‌کنند که باعث کاهش قابل ملاحظه ای از تعداد مجموعه‌های کاندید جهت بررسی (تعیین مکرر بودن یا نبودن مجموعه اشیاء) می‌شود.

الگوریتم FP-Growth در مقایسه با Apriori روش کارآمدتری برای تولید مجموعه اشیاء مکرر ارائه می‌دهد. این الگوریتم با ساخت یک درخت با نام FP-Tree سرعت فرآیند تولید اشیاء مکرر را به طور چشمگیری افزایش می‌دهد، در واقع با یکبار مراجعه به مجموعه تراکنش‌های مساله این درخت ساخته می‌شود. پس از ساخته شدن درخت با توجه به ترتیب نزولی Support مجموعه اشیاء تک عضوی (یعنی مجموعه اشیاء) مساله تولید مجموعه اشیاء مکرر به چندین زیر مسئله تجزیه می‌شود، که هدف در هر کدام از این زیر مساله ها، یافتن مجموعه اشیاء مکرری است که به یکی از آن اشیاء ختم خواهند شد.

الگوریتم Aprior علاوه بر تولید مجموعه اشیاء مکرر، اقدام به تولید مجموعه قوانین انجمنی نیز می‌نماید. در واقع این الگوریتم با استفاده از مجموعه اشیاء مکرر بدست آمده از مرحله قبل و نیز پارامتر ConfMIN قوانین انجمنی مرتبط را که دارای درجه اطمینان

بالائی هستند نیز تولید می‌کند. به طور کلی Confidence دارای خصوصیت هماهنگی (Monotone) نیست ولیکن Confidence قوانینی که از مجموعه اشیاء یکسانی بوجود می‌آیند دارای خصوصیت ناهماهنگی هستند. بنابراین با هرس نمودن کلیه ابرقوانین انجمنی یک قانون انجمنی یا Confidence (Rx) ≥ ConfMIN در شبکه قوانین انجمنی (مشابه با شبکه مجموعه اشیاء) اقدام به تولید قوانین انجمنی می‌نمائیم. پس از آنکه الگوریتم با استفاده از روش ذکر شده، کلیه قوانین انجمنی با اطمینان بالا را در شبکه قوانین انجمنی یافت، اقدام به الحاق نمودن آن دسته از قوانین انجمنی می‌نماید که پیشوند یکسانی را در توالی قانون به اشتراک می‌گذارند و بدین ترتیب قوانین کاندید تولید می‌شوند.

جهت آشنائی بیشتر به List of machine learning concepts مراجعه نمائید.