مروری بر کلاسه‌بندی و الگوریتم‌های آن

نیلوفر مظهری[۱]، مهدی ایمانی[۲]، مجید جودکی[۳]، احمد قلیچ‌پور[۴]

چکیده

کلاسه‌بندی یک تکنیک داده‌کاوی است که برای استخراج مدل از داده‌ها بر روی مقادیر گسسته به کار می‌رود. مدل‌ها به منظور گروه‌بندی داده‌ها میان برچسب‌های کلاس دسته‌ای مورد استفاده قرار می‌گیرند. لازم به ذکر است که ترتیب در میان این گروه‌های داده‌ای بی‌معناست. کلاسه‌بندی یک فرایند دو مرحله‌ای است. در مرحله اول (مرحله یادگیری)، الگوریتم کلاسه‌بندی با آنالیز کردن مجموعه داده‌ها، مدل را می‌سازد و به کشف روابط میان داده‌ها می‌پردازد و در مرحله دوم از مدل برای کلاسه‌بندی داده‌ها استفاده می‌کند. از درختان تصمیم‌گیری نیز برای کلاسه‌بندی داده‌ها استفاده می‌شود زیرا داده‌هایی با حجم انبوه را به راحتی مدیریت می‌کند و نمایشی ساده و قابل درک از کلاس‌های داده‌ای ارائه می‌کند و سرعت بالایی نیز در هر دو مرحله کلاسه‌بندی دارد.

در این مقاله پس از مروری بر کلاسه‌بندی، به بررسی کلاسه‌بندی با استنتاج درختان تصمیم‌گیری می‌پردازیم. در ادامه تکنیک‌های کلاسه‌بندی و موضوعات مربوط به آن را مورد بررسی قرار خواهیم داد. در پایان برخی الگوریتم‌های کلاسه‌بندی مطرح می‌شوند و به بررسی مزایا و معایب این الگوریتم‌ها نسبت به یکدیگر می‌پردازیم.

کلمات کلیدی

داده‌کاوی، درختان تصمیم‌گیری، کلاسه‌بندی، CART، C۴.۵، See۵/C۰.۵

An overview of classification and its algorithms

Niloofar Mazhari [۱], Mehdi Imani [۲], Majid Joudaki [۳], Ahmad Ghelich Pour [۴]

ABSTRACT

Classification is a data mining technique which is used to extract models from data on categorical variables. Models are used to classify the data into categorical class labels. It worth mentioning that order is meaningless among these classes. Classification is a two step process. In the first step (learning step or training phase) classification algorithm builds the model by analyzing data set and discovers relations between the data and in the second step uses the model to classify the data. We can use decision trees for data classification because they manage high volume data easily and give a simple and understanding represent of data classes. In addition, they have a high speed in both classification steps.

In this paper having a review of classification, we will discuss classification by decision tree induction. Further, we discuss classification techniques and its related issues. At the end, we will study classification algorithms and we will compare advantages and disadvantages of these algorithms.

KEYWORDS

Data mining, decision tree, classification, CART, C٤.٥, See٥/C٠.٥

[۱] نیلوفر مظهری، دانشجوی نرم‌افزار آموزشکده عالی ۱۷ شهریور کرج، n.mazhary@yahoo.com

[۲] مهدی ایمانی، مدرس دانشگاه آزاد اسلامی کرج (واحد سما) و آموزشکده عالی ۱۷ شهریور کرج m.imani@gmail.com

[۳] مجید جودکی، دانشجوی کارشناسی ارشد هوش مصنوعی دانشگاه صنعتی اصفهان، m.joudaki@cc.iut.ac.ir

[۴] احمد قلیچ‌پور، مدرس آموزشکده عالی ۱۷ شهریور کرج، fmamgh@yahoo.com

۱. مقدمه

پایگاههای داده غنی از اطلاعات پنهانی هستند که می‌توانند برای تصمیم‌گیری هوشمند به کار گرفته شوند. کلاسه‌بندی[۱] و پیش‌بینی[۲] دو تکنیک از تحلیل داده‌ها هستند که می‌توانند برای استخراج مدلهایی استفاده شوند که گروه‌های مهمی از داده‌ها را شرح داده و یا می‌توانند برای پیش‌بینی روند داده‌ها در آینده به کار آیند. چنین تحلیلاتی به ما کمک می‌کنند تا درک بهتری از داده‌های حجیم داشته باشیم[۱]. متدهای کلاسه‌بندی و پیش‌بینی زیادی توسط محققین در زمینه‌های یادگیری ماشین، شناسایی الگو و آمار ارائه شده است. اکثر الگوریتمها ماندگار در حافظه هستند و با فرض داده‌هایی با حجم کم عمل می‌کنند. پژوهش‌های اخیر داده‌کاوی بر اساس کارهای پیشین شکل گرفته است، اما این بار با ایده توسعه کلاسه‌بندها و پیش‌بینی‌کننده‌های توسعه‌پذیر که قادر به مدیریت داده‌های حجیم و ماندگار در حافظه باشند.

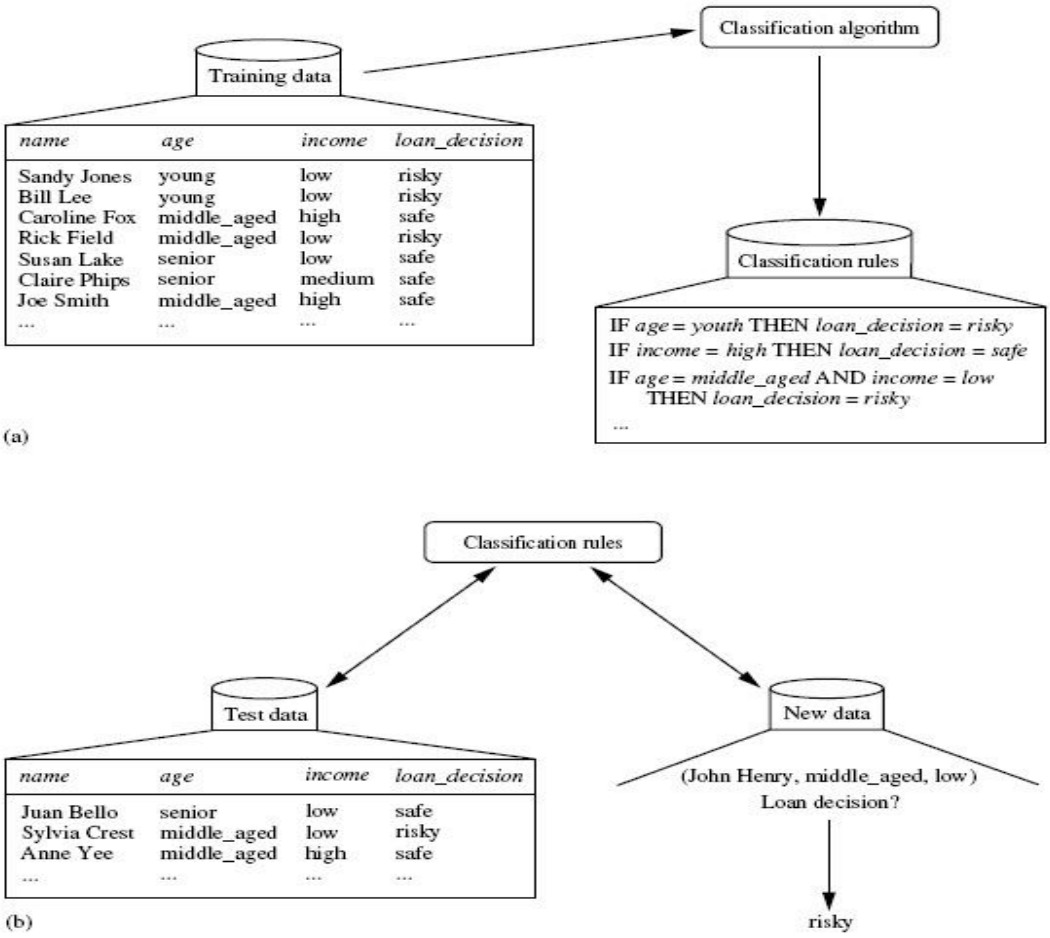
مدل کلاسه‌بند برچسبهای دسته‌ای را پیش‌بینی می‌کند. برای مثال، یک متصدی وام بانکی را در نظر بگیرید. وی برای تشخیص اینکه کدام یک از وامهای بانکی "امن" و کدام یک پرخطر هستند، به تحلیل داده‌های خود نیاز دارد. به عنوان مثالی دیگر، مدیریت یک شرکت بازاریابی کامپیوتری را در نظر بگیرید که نیاز به تحلیل داده‌ها دارد تا پیش‌بینی کند آیا خریداری با مشخصات داده شده، اقدام به خرید کامپیوتر می‌کند یا خیر. در این مورد، هدف از تحلیل داده‌ها کلاسه‌بندی است، تحلیلی که در آن یک مدل یا یک کلاسه‌بند برای پیش‌بینی برچسبهای دسته‌ای، مانند امن یا پرخطر ساخته می‌شود. این گروهها توسط مقادیر گسسته نمایش داده می‌شوند و ترتیب میان این مقادیر بی‌معناست.

۲.کلاسه‌بندي داده‌ها؛ یک پروسه دو مرحله‌اي

همانگونه که در شکل ۲.۱ نشان داده شده است، کلاسه‌بندی داده‌ها یک پروسه دو مرحله‌ای است. در مرحله اول، یک کلاسه‌بند با استفاده از مجموعه پیش‌تعریف شده‌ای از کلاسهای داده‌ای یا مفهومها ساخته می‌شود. این مرحله، مرحله یادگیری[۳] نامیده می‌شود. در این مرحله، الگوریتم کلاسه‌بندی توسط تحلیل –یا همان یادگیری از- مجموعه داده‌های آموزشی[۴] متشکل از سطرهای پایگاه داده‌ها (تاپلها) و برچسبهای کلاس[۵] مرتبط با آنها کلاسه‌بند را می‌سازد. یک تاپل، مانند X توسط بردارهای مشخصه[۶] n بعدی نشان داده می‌شود که n مقدار را بر روی تاپل از صفات خاصه n پایگاه داده‌ها بر می‌دارد. هر تاپل به یک کلاس از پیش تعریف شده تعلق دارد که توسط صفت خاصه‌ای از پایگاه داده‌های دیگر که خصوصیت برچسب کلاس (class label attribute) نام دارد، تعیین می‌شود. خصوصیت برچسب کلاس یک مقدار گسسته و بدون نظم است. از آنجایی آن را دسته‌ای می‌نامیم که هر مقدار آن یک گروه یا یک کلاس را ارائه می‌کند. تاپلهای مجزایی که مجموعه آموزشی را تشکیل می‌دهند، تاپلهای آموزشی نامیده می‌شوند و از پایگاه داده‌ای انتخاب می‌شوند که مورد تحلیل و بررسی قرار می‌گیرد. در زمینه داده‌کاوی تاپلهای داده‌ای با عنوانهایی مانند نمونه‌ها، مثالها، نقاط داده‌ای و یا اشیاء شناخته می‌شوند.

---

۱ Classification
۲ Prediction
۳ Learning step/Training phase
۴ Training set
۵ Class labels
۶ Attribute vector



شکل ۲.۱ پروسه کلاسه‌بندی داده‌ها. (a) مرحله یادگیری: داده‌های آموزشی توسط یک الگوریتم کلاسه‌بندی آزمایش می‌شوند. در اینجا، صفت خاصه برچسب کلاس تصمیم‌گیری برای وام بانکی است و مدل آموخته (learned model) یا همان کلاسه‌بند به صورت قوانین کلاسه‌بندی نمایش داده شده‌اند. (b) مرحله کلاسه‌بندی: از داده‌های تست شده برای تخمین میزان صحت قوانین کلاسه‌بندی استفاده می‌شود. اگر میزان صحت مورد تایید واقع شود، قوانین کلاسه‌بندی می‌توانند برای کلاسه‌بندی تاپلهای داده‌ای جدید بکار گرفته شوند.

از آنجایی که برچسب کلاس برای هر تاپل آموزشی از قبل فراهم شده است (به این معنا که با دانستن اینکه هر تاپل به چه کلاسی تعلق دارد، یادگیری کلاسه‌بند supervise است). این مرحله به عنوان مرحله Supervised Learning نیز شناخته می‌شود. این مفهوم در مقابل unsupervised learning قرار دارد که در آن برچسبهای کلاس برای هر تاپل آموزشی ناشناخته است و تعداد یا مجموعه کلاسهایی که در یادگیری استفاده می‌شوند، قبل از انجام این مرحله مشخص نیستند.

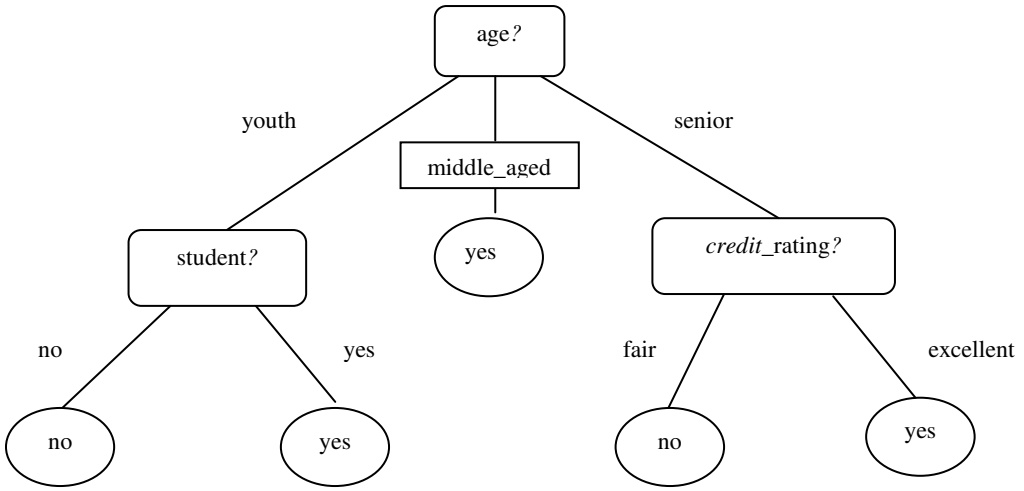
این مرحله ابتدایی از پردازش کلاسه‌بندی می‌تواند به عنوان مرحله کشف روابط و یا یادگیری تابع نیز بررسی شود، $y=f(X)$ ، که می‌تواند برچسب کلاس مرتبط y را با تاپل X پیش‌بینی کند. از این نظر، هدف ما در این مرحله کشف روابط و یا توابعی است که کلاسهای داده‌ای را از هم متمایز می‌کنند. معمولا این روابط در قالب قوانین کلاسه‌بندی، درختان تصمیم‌گیری و یا فرمولهای ریاضی نشان داده می‌شوند. در مورد مثال ما، این روابط در قالب قوانین کلاسه‌بندی نشان داده می‌شوند که مشخص می‌کنند که یک وام بانکی در کدام یک از گروههای امن یا پرخطر قرار می‌گیرد. قوانین علاوه بر فرهم آوردن دیدی عمیق‌تر از محتویات پایگاه‌داده‌ها، در آینده می‌توانند برای گروه‌بندی تاپلهای داده‌ای نیز مورد استفاده قرار گیرند. این قوانین همچنین نمایش خلاصه‌شده‌ای از داده‌ها فراهم می‌کنند.

در مرحله دوم، از مدل برای کلاسه‌بندی داده‌ها استفاده می‌شود. در ابتدا صحت پیش‌بینی کلاسه‌بند تخمین زده می‌شود. اگر از یک مجموعه داده‌های آموزشی برای تخمین استفاده کنیم، این تخمین خوش‌بینانه خواهد بود، زیرا اغلب کلاسه‌بند داده‌ها را over fit می‌کند (یعنی در طول مرحله یادگیری باعث ایجاد آنومالیهایی از داده‌های آموزشی می‌شود که در مجموعه داده‌های عمومی وجود ندارند). بنابراین از یک مجموعه تست[۷] استفاده می‌شود که از تاپلهای تست شده و برچسب کلاس مرتبط با آنها تشکیل شده است. این تاپلها به صورت تصادفی از مجموعه داده‌های عمومی انتخاب می‌شوند. این تاپلها مستقل از تاپلهای آموزشی هستند، به این معنا که در ساخت کلاسه‌بند استفاده نمی‌شوند.

صحت یک کلاسه‌بند بر روی مجموعه تست داده شده، درصد تاپلهایی از مجموعه تست است که به طور صحیح توسط کلاسه‌بند، کلاسه‌بندی شده‌اند. برچسب کلاس مرتبط هر تاپل تست شده با کلاس پیش‌بینی شده توسط کلاسه‌بند (در مرحله یادگیری) برای آن تاپل مقایسه می‌شود. اگر میزان صحت کلاسه‌بند قابل قبول واقع شود، می‌تواند در آینده برای کلاسه‌بندی تاپلهای داده‌ای استفاده شود که عنوان کلاس آنها مشخص نیست. برای مثال در قسمت a شکل ۲.۱، قوانین کلاسه‌بندی آموخته شده از تحلیل داده‌های مربوط به وامهای قبلی، می‌توانند برای تایید و یا رد کردن وام جدید بکار روند.

۳. کلاسه بندي با استنتاج درختان تصميم‌گيري

استنتاج درختان تصمیم‌گیری، به معنای یادگیری درختان تصمیم‌گیری از تاپلهای آموزشی با برچسبهای کلاس مشخص شده است. یک درخت تصمیم‌گیری دارای ساختاری شبیه به ساختار فلوچارت است، که در آن هر گره داخلی[۸] (گرهی که یک گره برگ نیست) تستی بر روی یک صفت خاصه را نشان می‌دهد و هر شاخه[۹] نشان‌دهنده خروجی تست است و هر گره برگ[۱۰] در برگیرنده یک برچسب کلاس است. بالاترین گره در یک درخت، گره ریشه[۱۱] نامیده می‌شود. گره‌های داخلی در فلوچارت با مستطیل و گره‌های برگ با بیضی نشان داده می‌شوند. بعضی الگوریتمهای درختان تصمیم‌گیری فقط درختهای دودویی تولید می‌کنند (یعنی هر گره داخلی دقیقا به دو گره دیگر تقسیم می‌شود) در حالی که الگوریتمهای دیگری نیز وجود دارند که درختهای غیر دودویی تولید می‌کنند. شکل ۳.۱ این موضوع را نشان می‌دهد.



شکل ۳.۱ درخت تصمیم‌گیری برای مثال پیش‌بینی خرید کامپیوتر که نشان می‌دهد یک خریدار اقدام به خرید کامپیوتر می‌کند یا خیر. هر گره داخلی (گره غیر برگ) نشان‌دهنده تستی بر روی صفت خاصه است. هر گره برگ نمایانگر یک کلاس است که به سوال "آیا یک خریدار اقدام به خرید می‌کند؟" با بله و خیر پاسخ می‌دهد.

---

[۷] Test set
[۸] Internal node
[۹] Branch
[۱۰] leaf node
[۱۱] Root node

۳.۱ چگونه درختان تصمیم‌گیري براي کلاسه‌بندي استفاده مي‌شوند؟

با داشتن تاپل X، که برچسب کلاس مرتبط با آن ناشناخته است، مقادیر صفات خاصه تاپل توسط درخت تصمیم‌گیری تست می‌شوند. مسیری از ریشه درخت تا گره برگ دنبال می‌شود تا کلاسی که تاپل متعلق به آن است، مشخص شود. درختان تصمیم‌گیری به راحتی قابل تبدیل به قوانین کلاسه‌بندی هستند.

۳.۲ چرا درختان تصمیم‌گیري بسیار محبوب هستند؟

ساختار کلاسه‌بندهای درختان تصمیم‌گیری به هیچ قلمرو دانش و یا تنظیم پارامتری نیاز ندارند. درختان تصمیم‌گیری می‌توانند داده‌هایی با حجم زیاد را مدیریت کنند. نمایش آنها از دانش کسب شده به صورت یک درخت، برای انسان قابل درک است و به راحتی شبیه‌سازی می‌شود. مراحل یادگیری و کلاسه‌بندی درختان تصمیم‌گیری سریع و ساده هستند. در حالت کلی، کلاسه‌بندهای درختان تصمیم‌گیری میزان صحت قابل قبولی دارند. با این حال، میزان موفقیت در استفاده از درختان بستگی به داده‌های مورد استفاده نیز دارد.

۳.۳ الگوریتمهاي درختان تصمیم‌گیري

الگوریتمهای مورد استفاده در درختان تصمیم‌گیری، درختان خود را توسط عملیات بازگشتی تقسیم و غلبه می‌سازند و ساختاری از بالا به پایین دارند. الگوریتم با یک مجموعه آموزشی از تاپلها و برچسب کلاس مرتبط با آنها شروع می‌کند. مجموعه آموزشی به صورت بازگشتی به زیرمجموعه‌های کوچکتری تجزیه شده و درخت را تشکیل می‌دهد. در زیر شبه کد یک الگوریتم پایه برای استنتاج درختان تصمیم‌گیری از تاپلهای آموزشی آورده شده 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 label;
- *Attribute_list*, the set of condidate attributes;
- *Attribute_selection_method*, a procedure to detemine the splitting criterion that “best” partitions the data tuples into indivisual classes. This criterion consists of a *splitting_attribute* and, possibly, either a *split point* or *splitting subset.*

Output: a decision tree.

Method:

(۱) create a node *N*;
(۲) **if** tuples in *D* are all of the same class, *C* **then**
(۳) return *N* as a leaf node labeled with class *C*;
(۴) **if** attribute_list is empty **then**
(۵) return *N* as a leaf node labeled with the majority class in *D*; //majority voting
(۶) apply **Attribute_selection_method** (*D*, attribute_list) to **find** the “best” *splitting_criterion*;
(۷) label node *N* with *splitting_criterion*;
(۸) **if** *splitting_attribute* is discrete-valued **and**
multiway splits allowed **then** //not restricted to binary trees
(۹) *attribute_list*← *attribute_list* - *splitting_attribute*; // remove *splitting_attribute*
(۱۰) **for each** outcome *j* of the *splitting_criterion*

//partition the tuples and grow subtrees for each partition

(۱۱) let D_j be the set of data tuples in *D* satisfying outcome *j*; //a partition

(۱۲) **if** *Dj* is empty **then**

(۱۳) attach a leaf labeled with the majority class in *D* to node *N*;

(۱۴) **else** attach the node returned by **Generate_decision_tree**(D_j, *attribute_list*) to node N;

endfor

(۱۵) return N;

استراتژی الگوریتم فوق به شرح ذیل است :

- الگوریتم با پارامترهای D، لیست صفات مشخصه[۱۲] و متد انتخاب صفت[۱۳] فراخوانی می‌شود. D را به عنوان یک بخش داده‌ای می‌شناسیم. در ابتدا D شامل تمامی تاپلهای مجموعه آموزشی و برچسب کلاسهای متناظر با آنها است. پارامتر صفات مشخصه، لیستی از صفات خاصه موجود در تاپلها است. متد انتخاب صفت، یک روال اکتشافی برای انتخاب صفتی است که به بهترین صورت به کلاسه‌بندی تاپلها بر اساس کلاسها می‌پردازد. این روال از یک متد انتخاب صفت مانند information gain یا gini index استفاده می‌کند. متد انتخاب صفت تعیین می‌کند که درخت دودویی محض است یا خیر؛ برخی از این متدها همانند gini index درخت را مستلزم به دودویی بودن می‌کنند و برخی دیگر مانند information gain تقسیمات چندگانه را ممکن می‌سازند.
- درخت در گام اول به عنوان یک گره تک، N، آغاز می‌شود که نشان دهنده تاپلهای آموزشی D است. (خط ۱)
- اگر تاپلهای D همگی متعلق به یک کلاس باشند، گره N یک برگ با برچسب آن کلاس خواهد بود. (خط ۲ و ۳). توجه داشته باشید که خطوط ۴ و ۵ شرایط خاتمه هستند. تمامی شرایط خاتمه در پایان الگوریتم شرح داده می‌شوند.
- در غیر این صورت، متد انتخاب صفت فراخوانی می‌شود تا ضابطه تقسیم[۱۴] را تعیین کند. ضابطه تقسیم با تعیین بهترین راه کلاسه‌بندی تاپلهای D در کلاسهای مجزا مشخص می‌کند که کدام صفت باید در گره N مورد آزمایش قرار گیرد (خط ۶). ضابطه تقسیم همچنین بیان می‌کند که با توجه به خروجیهای تست انتخاب شده، چه شاخه‌هایی باید از گره N حاصل شوند. به عبارت دیگر، ضابطه تقسیم نشان‌دهنده صفت خاصه تقسیم‌کننده است و ممکن است نقطه تقسیم[۱۵] و زیرمجموعه تقسیم[۱۶] را نیز تعیین کند. نقطه تقسیم، D را به یک سری بخش تبدیل می‌کند. در یک ضابطه تقسیم ایده‌آل، بخشهای نتیجه شده در هر شاخه باید تا حد ممکن خالص باشند؛ یک بخش در صورتی خالص است که تمامی تاپلهای موجود در آن متعلق به یک کلاس باشند.
- گره N توسط ضابطه تقسیم برچسب می‌خورد که تستی را بر روی گره مورد نظر ارائه می‌دهد (خط ۷). برای هر یک از خروجیهای ضابطه تقسیم، یک شاخه از گره N ایجاد می‌شود. تاپلهای D طبق این روال بخش‌بندی می‌شوند (خطوط ۱۰ و ۱۱). این سه حالت ممکن در شکل ۳.۳.۱ نشان داده شده‌اند. A را به عنوان صفت تقسیم‌کننده در نظر می‌گیریم که v مقدار متفاوت (با توجه به داده‌های آموزشی) دارد.

۱. A دارای مقادیر گسسته است. در این حالت، خروجیهای تست بر روی گره N به طور مستقیم به مقادیر شناخته شده A مرتبط هستند. برای هر مقدار شناخته شده A، یک شاخه ایجاد شده و با آن مقدار برچسب زده می‌شود. (شکل ۳.۳.۱، قسمت a) از آنجایی که تمامی تاپلها در یک بخش داده شده، دارای مقادیر یکسانی برای A هستند، A نیازی به اعمال بخش‌بندیهای بیشتر بر روی تاپلها نداشته و بنابراین از لیست صفات خاصه حذف می‌شود (خطوط ۸ و ۹).

۲. A دارای مقادیر پیوسته است. در این حالت، تست بر روی گره N دو خروجی ممکن دارد، به ترتیب در صورتی که نقطه تقسیم $A\leq$ باشد و یا اینکه نقطه تقسیم $A>$ باشد که نقطه تقسیم به عنوان قسمتی از ضابطه تقسیم توسط متد انتخاب صفت بازگردادنده می‌شود. دو شاخه از N ایجاد شده و بنابر خروجیها برچسب زده می‌شود (شکل ۳.۳.۱، قسمت b).

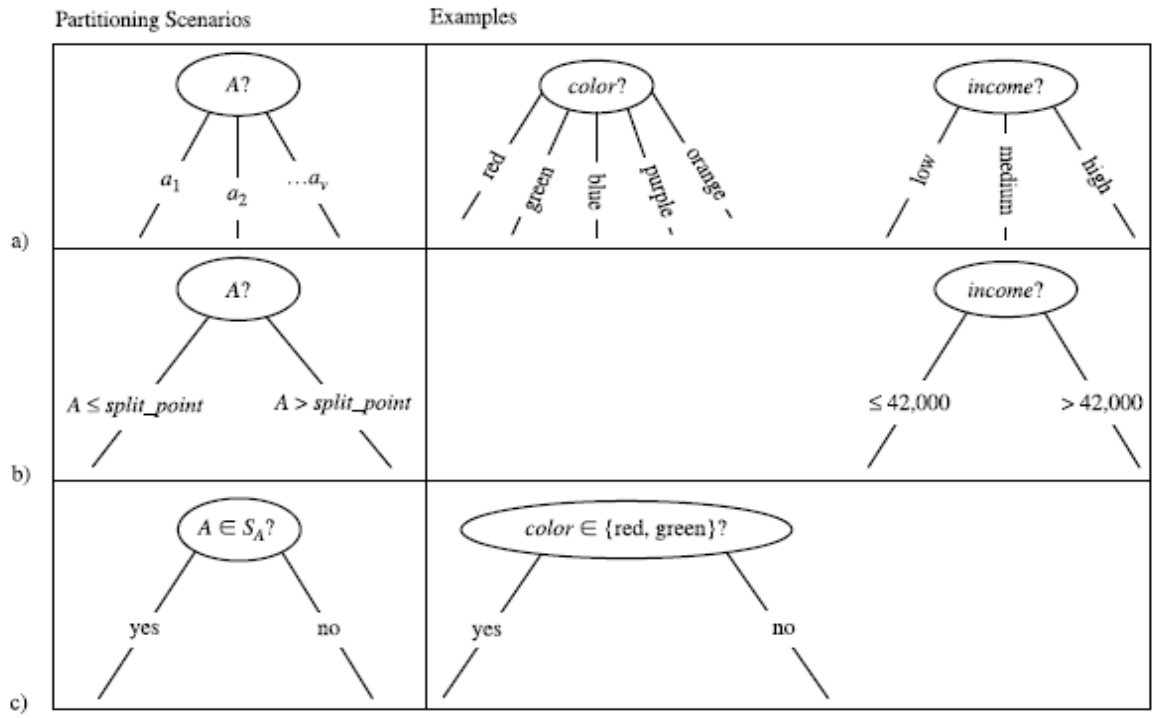
۳. A دارای مقادیر گسسته است و درخت دودویی است (که توسط متد انتخاب صفت یا الگوریتم مورد استفاده تعیین می‌شود). در این حالت، S_A زیرمجموعه تقسیم برای A است که به عنوان قسمتی از ضابطه تقسیم توسط متد انتخاب صفت بازگردادنده می‌شود. این زیرمجموعه متشکل از صفات شناخته شده‌ای از A است. نتیجه تست در صورتی مطلوب

---

[۱۲] Attribute list
[۱۳] Attribute selection method
[۱۴] Splitting criterion
[۱۵] Split-point
[۱۶] Splitting subset

است که تاپل مورد نظر از مجموعه A زیرمجموعه‌ای از S_A باشد. دو شاخه از N منشعب می‌شوند (شکل ۳.۳.۱، قسمت c).

- الگوریتم برای ساخت درخت تصمیم‌گیری، برای هر یک از تاپلهای نتیجه شده در هر بخش از فرآیندی مشابه به صورت بازگشتی استفاده می‌کند (خط ۱۴)
- بخش‌بندی بازگشتی فقط در صورت وقوع یکی از شرایط خاتمه زیر متوقف می‌شود:
  - تمامی تاپلها در بخش D (نمایش داده شده در گره N) متعلق به یک کلاس باشند (خطوط ۲ و ۳).
  - صفتی وجود ندارد که بر اساس آن بخش‌بندی بیشتری بر روی تاپلها صورت گیرد (خط ۴). در این حالت اکثر آراء به کار گرفته می‌شوند (خط ۵). یعنی گره N به یک برگ تبدیل شده و برچسب آن توسط کلاس متداول در D تعیین می‌شود.
  - تاپلی برای یک شاخه داده شده وجود ندارد، در واقع یکی از بخشهای D مانند D_j تهی است (خط ۱۲). در این موارد یک برگ با برچسب کلاس متداول در D ایجاد می‌شود (خط ۱۳).
- درخت تصمیم‌گیری نتیجه شده بازگردانده می‌شود (خط ۱۵).



شکل ۳.۳.۱ سه حالت ممکن برای تاپلهای بخش‌بندی شونده مبتنی بر ضابطه تقسیم، که با مثال نمایش داده شده‌اند. A به عنوان صفت تقسیم کننده مفروض است. (a) اگر A دارای مقادیر گسسته باشد، برای هر مقدار شناخته شده A یک شاخه در نظر گرفته می‌شود. (b) اگر A دارای مقادیر گسسته باشد، آنگاه دو شاخه از A برای حالات نقطه تقسیم $A \leq$ و نقطه تقسیم $A >$ ایجاد می‌شوند. (c) اگر A دارای مقادیر گسسته باشد و درخت ملزم به دودویی بودن باشد، تست به شکل $A \in S_A$ خواهد بود که S_A زیرمجموعه تقسیم برای A است.

CART ۳.٤

در سال ۱۹۸۴، CART: Classification and Regression Trees تالیف شده توسط Leo Breiman، Jerome Friedman، Richard Olshen و Charles Stone تحول عظیمی در علم هوش مصنوعی، یادگیری ماشین، علوم آماری غیرپارامتری و داده‌کاوی ایجاد کرد. این الگوریتم از لحاظ داشتن مطالعه‌ای جامع بر روی درختان تصمیم‌گیری، ارائه ابداعات فنی، بحث پیچیده‌ای بر روی تحلیل داده‌های با ساختار درختی و داشتن مدیریتی قدرتمند بر روی تئوری نمونه‌های حجیم برای درختان حائز اهمیت است[۴].

درخت تصمیم‌گیری CART یک روال بخش‌بندی بازگشتی و دودویی است که قادر به پردازش صفتهای خاصه با مقادیر پیوسته و گسسته است. داده‌ها در فرم سطری اداره می‌شوند و نیازی به عملیات binning نیست[۳]. درختان بدون استفاده از هیچ قانون توقفی تا بیشترین حد ممکن رشد داده شده و سپس توسط الگوریتم هرس cost-complexity تا به ریشه (تقسیم به تقسیم) هرس می‌شوند. تقسیم بعدی که مورد هرس قرار می‌گیرد تقسیمی است که در عملکرد کلی درخت بر روی داده‌های آموزشی کمترین نقش را ایفا می‌کند. در یک زمان ممکن است بیشتر از یک تقسیم توسط این عملیات حذف شوند. هدف مکانیزم CART تولید سلسله‌ای از درختهای تودرتو و هرس شده است که هر یک از آنها درختانی بهینه و کاندید هستند. اندازه مناسب درخت توسط محاسبه کارایی پیش‌بینی شده هر درخت در مراحل هرس تعیین می‌شود. کارایی درخت بر روی داده‌های تست مستقل (یا از طریق cross validation) سنجیده می‌شود و انتخاب درخت تنها پس از ارزش‌یابی مبتنی بر داده‌های تست ادامه می‌یابد. اگر داده تستی وجود نداشته باشد و cross validation اجرا نشده باشد، CART قادر به تشخیص بهترین درخت در یک مرحله نخواهد بود. در این مورد، این متد تفاوت عمده‌ای با متدهایی همانند C۴.۵ دارد که مدلهای ترجیح داده شده را بر اساس مقادیر داده‌های آموزشی تولید می‌کنند.

۳.۴.۱ قوانین تقسیم

قوانین تقسیم CART همواره به این صورت نمایش داده می‌شوند:

در صورت برقراری شرط نمونه در سمت چپ و در غیر این صورت در سمت راست درخت قرار داده می‌شود. برای صفات خاصه پیوسته، شرط به شکل "صفت خاصه $C >= X_i$" خواهد بود. در مورد صفات خاصه گسسته، شرط به صورت عضویت در لیست مشخصی از مقادیر است.

در این متد تقسیمات دودویی به چند دلیل ترجیح داده می‌شوند. تقسیمات دودویی بسیار آهسته‌تر از تقسیمات چندگانه یکپارچگی داده‌ها را از بین می‌برند. دیگر اینکه تقسیمات تکراری بر روی یک صفت خاصه مجاز است و در این صورت، قادر خواهیم بود برای یک صفت خاصه به هر تعداد مورد نیاز بخشهایی را ایجاد کنیم. دلیل سوم آن است که تئوری نمونه‌ای بزرگ که توسط مولفین CART توسعه یافت، محدود به تقسیمات دودویی بود.

CART قسمت عمده مباحث خود را بر روی ضابطه Gini متمرکز می‌کند که شبیه به ضابطه بهره اطلاعاتی است. برای یک مقصد دودویی (۱/۰)، میزان ناخالصی Gini برای گره t برابر است با:

که $p(t)$ فرکانس وابستگی کلاس ۱ در گره است و بهره تولید شده توسط تقسیم گره والد P به فرزند چپ، L، و فرزند راست، R، برابر است با:

در اینجا q کسری از نمونه‌هاست که به سمت چپ فرستاده می‌شوند. مولفین CART، ضابطه Gini را به بهره اطلاعاتی ترجیح داده‌اند زیرا Gini می‌تواند به راحتی برای شامل کردن هزینه‌های متقارن توسعه یابد و بسیار سریع‌تر از بهره اطلاعاتی محاسبه می‌شود (البته نسخه‌های جدید CART، بهره اطلاعاتی را به عنوان یک قانون تقسیم اختیاری شامل می‌شوند).

۳.۴.۲ قوانین توقف

در کارهای آغازین انجام شده بر روی درختان تصمیم‌گیری، عملیات هرس مجاز نبود [۳]. به جای آن درختان تا زمانی رشد می‌کردند که به یکی از شرایط توقف برخورد کنند و درخت حاصل، درختی نهایی بود. مولفین CART معتقدند هیچ قانون توقفی برای درختان این متد تضمین نمی‌کند که درخت ساختار داده‌ای مهمی را از دست ندهد (برای مثال، مسئله دو بعدی XOR را در نظر بگیرید). بنابراین درختان بدون قوانین توقف رشد می‌کنند. مدل بهینه و نهایی از درخت بزرگ نتیجه شده استخراج می‌گردد.

مکانیزم هرس به صورت کامل وابسته به داده‌های آموزشی است و با میزان cost complexity تعریف شده مطابق زیر شروع می‌شود:

که در آن، $R(T)$ هزینه نمونه‌ای آموزشی از درخت است، $|T|$ تعداد گرههای ترمینال درخت و a مجازات وضع شده برای هر گره است. اگر $a=0$ باشد آنگاه واضح است که درخت مینیمم cost-complexity، در بزرگترین حد ممکن خود خواهد بود. اگر a مجاز به افزایش یافتن به طور مستمر باشد، آنگاه درخت مینیمم cost-complexity کوچکتر و کوچکتر خواهد شد، زیرا تقسیمات پایینی درخت که باعث کمترین کاهش در $R(T)$ می‌شوند، هرس خواهند شد. پارامتر a به طور مستمر از صفر تا مقداری افزایش می‌یابد که برای هرس تمامی تقسیمات کافی باشد.

C4.5 ۳.۵

الگوریتم C4.5 [۴] ، مدل توسعه یافته‌ای از الگوریتمهای CLS و ID3 است. C4.5 همانند دو الگوریتم پیشین کلاسه‌بندهایی را تولید می‌کند که به فرم درختان تصمیم‌گیری نمایش داده می‌شوند، اما قادر به تولید کلاسه‌بندهایی قابل فهم‌تر و در فرم مجموعه قوانین نیز است. در ادامه به بررسی این الگوریتم پرداخته و به نکاتی اشاره می‌کنیم که این الگوریتم را از نسخه‌های قبلی خود متمایز ساخته است.

با داشتن مجموعه‌ای از حالات مختلف، S، C4.5 ابتدا درختی اولیه را با استفاده از الگوریتم تقسیم و غلبه همانند زیر می‌سازد:

- اگر تمامی موارد در S به یک کلاس تعلق داشته باشند یا S مجموعه‌ای کوچک باشد، درخت گره برگی خواهد بود که با کلاس متداول در S برچسب می‌خورد.
- در غیر این صورت، تستی بر اساس یک صفت خاصه با دو یا چند خروجی انتخاب کنید. این تست را ریشه درخت قرار داده و برای هر خروجی این تست یک شاخه در نظر بگیرید، به همان نسبت S را به زیرمجموعه‌های $S_1, S_2, \ldots$ بنابر خروجی هر مورد بخش‌بندی کنید و فرآیندی مشابه را به صورت بازگشتی برای هر زیرمجموعه تکرار کنید.

در مرحله آخر C4.5 از دو ضابطه اکتشافی برای درجه‌بندی تستهای ممکن استفاده می‌کند: بهره اطلاعاتی که آنتراپی کل زیرمجموعه‌ها را به حداقل خود می‌رساند (اما به شدت به تستهایی با خروجی عددی تمایل دارد) و gain ratio که بهره اطلاعاتی را بر اطلاعاتی تقسیم می‌کند که از خروجیهای تست حاصل شده است.

صفات خاصه می‌توانند عددی یا گسسته باشند و این قالب خروجی تست را تعیین می‌کند. برای یک صفت خاصه عددی به نام A داریم: $\{A \le h \ , \ A > h\}$ که نقطه h توسط مرتب‌سازی S بر روی مقادیر A و انتخاب تقسیمی در میان مجموعه مقادیر است که ضابطه بالا را به بیشترین میزان خود می‌رساند. صفت خاصه A با مقادیر گسسته، به صورت پیش‌فرض برای هر مقدار یک خروجی دارد، اما امکان گروه‌بندی مقادیر در دو یا چند زیرمجموعه با وجود یک خروجی برای هر زیرمجموعه نیز وجود دارد.

سپس برای جلوگیری از overfitting درخت اولیه هرس می‌شود. الگوریتم هرس بر اساس تخمینی بدبینانه بر روی میزان خطای مرتبط با N حالت است که E تعداد از آنها به کلاس بسیار متداول (the most frequent class) تعلق ندارند. به جای محاسبه E/N، C4.5 با استفاده از اطمینان تعریف شده توسط کاربر که دارای مقدار پیش‌فرض ۰.۲۵ است، حد بالای احتمالات دو جمله‌ای را در زمانی که E رویداد در N حالت مورد آزمایش قرار گرفته‌اند را تشخیص می‌دهد.

عملیات هرس از برگها تا به ریشه ادامه می‌یابد. میزان خطای تخمین زده شده در یک برگ با N حالت و E عدد خطا مساوی با N برابر میزان تخمین خطای بدبینانه در بالا است. برای یک زیردرخت، C4.5 میزان خطای تخمین زده شده برای شاخه‌ها را نیز به میزان قبلی اضافه کرده و این مقدار را با میزان خطای تخمین زده شده در حالتی مقایسه می‌کند که زیردرخت با یک برگ جایگزین شود. اگر میزان خطا در حالت دوم بیشتر از حالت اول نباشد، آنگاه زیردرخت هرس شده است. به صورت مشابه، C4.5، خطای تخمین زده شده را

در حالتی که زیردرخت با یکی از شاخه‌های خود جایگزین شود نیز بررسی می‌کند و در صورتی که این جایگزینی سودمند باشد، درخت را بر این اساس تغییر می‌دهد. عملیات هرس در یک گذر از درخت خاتمه می‌یابد.

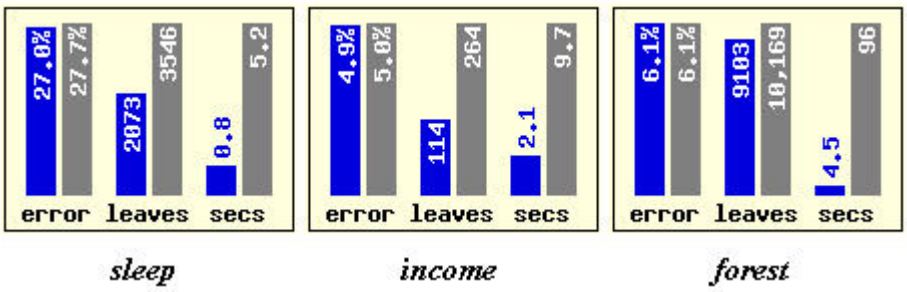
See۵/C۰.۵ ۳.٦

همانگونه که بررسی شد، C۴.۵ ابزاری با استفاده گسترده برای داده‌کاوی است که از متد ID۳ سرچشمه می‌گیرد و به همین ترتیب پیشینه متد See۵/C۵.۰ [۷] می‌باشد. در سال ۱۹۹۷ الگوریتم C۵.۰ جایگزین C۴.۵ شد. برای بررسی پیشرفتها و قابلیتهای این متد جدید، به مقایسه See۵/C۵.۰ نسخه ۲.۰۶ و C۴.۵ نسخه ۸ با استفاده از سه مجموعه داده با اندازه قابل تغییر می‌پردازیم:

- Sleep stage scoring data (Sleep با ۱۰۵۹۰۸ مورد) هر مورد در این کاربرد نمایشی توسط ۶ صفت خاصه با مقادیر عددی توصیف شده و به یکی از ۶ کلاس موجود تعلق دارد. C۵.۰ و C۴.۵ از ۵۲۹۵۴ مورد برای ساخت کلاسه‌بندهایی استفاده می‌کنند که بر روی ٥٢٩٥٤ مورد باقی مانده تست می‌شوند. برای دسترسی به داده‌ها به [۹] مراجعه کنید.
- Census income data (income، با ۱۹۹۵۲۳ مورد) هدف این برنامه کاربردی پیش‌بینی این است که آیا درآمد یک شخص بیشتر و یا کمتر از $۵۰،۰۰۰ است. با استفاده از ۷ صفت خاصه عددی و ۳۳ صفت خاصه گسسته است. داده‌ها به دو مجموعه آموزشی با ۹۹۷۶۲ مورد و مجموعه تست با ۹۹۷۶۱ مورد تقسیم می‌شوند. برای دسترسی به داده‌ها به [۱۰] مراجعه کنید.
- Forest cover type data (Forest، با ۵۸۱۰۱۲ مورد) این برنامه کاربردی شامل ۷ کلاس (انواع مختلف forest cover) است و موارد توسط ۱۲ صفت خاصه عددی و دو صفت خاصه چند مقداری و گسسته توصیف می‌شوند. همانند قبل، نیمی از داده‌ها یعنی ۲۹۰۵۰۶ مورد برای آموزش و بقیه موارد برای تست کلاسه‌بندهای آموخته استفاده می‌شوند. برای دسترسی به داده‌ها به [۱۱] مراجعه کنید.

از آنجایی که C۴.۵ سیستمی مبتنی بر Unix است، برای راحتی مقایسه، نتایج برای C۵.۰ نسخه Unix نمایش داده شده‌اند. هر دو توسط Intel C compiler ۱۰.۱ با تنظیمات بهینه یکسان کامپایل شده‌اند. زمان با واحد ثانیه برای یک unloaded Intel Q۹۵۵۰ (۲.۸GHz) با ۲GB حافظه RAM با ۶۴-bit Fedora Core در نظر گرفته شده است. حال اجازه دهید بررسی کنیم که چگونه C۵.۰ از C۴.۵ پیشی می‌گیرد.

۳.٦.۱ درختان تصمیم‌گیري سریع‌تر و کوچکتر



شکل ۳.٦.۱.۱ مقایسه میزان دقت، سرعت و حافظه مورد نیاز برای C٤.٥ و C٥.٠. نتایج C٥.٠ به رنگ آبی نمایش داده شده‌اند.

برای هر سه مجموعه داده، C۴.۵ و C۵.۰ درختانی با صحت پیش‌بینی یکسانی تولید می‌کنند (اگرچه درختان C۵.۰ برای مجموعه داده‌های sleep و income اندکی بهتر از درختان C۴.۵ هستند) تفاوت عمده در اندازه درختان و زمان مورد نیاز برای محاسبه است؛ درختان C۵.۰ به طرز قابل توجهی کوچکتر هستند و C۵.۰ برای سه مجموعه داده sleep، income، و forest به ترتیب ٦.٥، ٤.٦ و ۲۱ برابر سریع‌تر از C۴.۵ عمل می‌کند.

۳.٦.۲ قابلیتهاي جدید

C۵.۰ قابلیتهای جدیدی همچون هزینه‌های غیرکلاسه‌بندی متغیر (variable misclassification costs) ارائه می‌کند. در C۴.۵ رفتاری یکسان نسبت به تمامی خطاها وجود دارد، اما در کاربردهای عملی برخی خطاهای کلاسه‌بندی بسیار بحرانی‌تر از برخی خطاهای دیگر هستند. C۵.۰ برای هر جفت کلاس پیش‌بینی شده/واقعی (predicted/actual class pair) هزینه‌ای جداگانه تعریف می‌کند که در صورت استفاده از این قابلیت، C۵.۰ کلاسه‌بندهایی را برای به حداقل رساندن هزینه‌های مورد انتظار غیرکلاسه‌بندی (expected misclassification costs) به جای نرخهای خطا می‌سازد. حتی ممکن است خود موارد نیز از اهمیت یکسانی برخوردار نباشند. C۵.۰ امکاناتی را برای خصوصیت وزن موردی (case weight attribute) فراهم می‌آورد که به محاسبه میزان اهمیت هر مورد می‌پردازد. با داشتن این ویژگی، C۵.۰ تلاش به کاهش نرخ خطای پیش‌بینی‌شده وزین (weighted predictive error rate) دارد.

در برخی کاربردهای اخیر داده‌کاوی، حجم داده‌ها بسیار افزایش یافته است. در بعضی موارد، صدها و یا حتی هزاران صفت خاصه مشاهده می‌شود. C۵.۰ قبل از ساخته شدن کلاسه‌بند، قادر به غربال کردن خودکار صفات خاصه است و این عمل را با حذف صفاتی انجام می‌دهد که ربط و وابستگی کمتری نسبت به سایر صفات دارند. در کاربردهایی با حجم داده‌ای بالا، غربال سازی منجر به کلاسه‌بندهایی کوچکتر و صحت پیش‌بینی بالاتری می‌گردد.

۴. تفاوتهاي عمده الگوریتمهای مورد بحث

الگوریتم ساخت درخت C۴.۵ در برخی موارد با CART تفاوت دارد از جمله:

- در CART تستها همیشه دودویی هستند، در حالیکه C۴.۵ دو یا چند خروجی را مجاز می‌داند.
- CART از Gini diversity index برای رتبه‌بندی تستها استفاده می‌کند، درحالیکه C۴.۵ از ضابطه مبتنی بر اطلاعات استفاده می‌کند.
- CART درختان خود را با استفاده از یک مدل Cost-complexity هرس می‌کند، که پارامترهای آن توسط cross-validation تخمین زده می‌شوند. C۴.۵ از الگوریتمی تک گام استفاده می‌کند که از حد اطمینان دو جمله‌ای، مشتق شده است.
- در زمانی که صفت خاصه تست شده مقادیر ناشناخته‌ای دارد، CART از تستهای جایگزینی استفاده می‌کند که خروجیها را تقریب می‌زنند اما C۴.۵ احتمالات آن مورد را در میان خروجیها تقسیم می‌کند.

تغییرات ایجاد شده در C۵.۰ نسبت به C۴.۵ باعث بازدهی بیشتر شده و نیز قابلیتهای جدید زیر را شامل می‌شود:

- عملیات boosting اندکی متفاوت، که مجموعه‌ای از کلاسه‌بندها را تشکیل می‌دهد که بعد برای ساختن یک کلاسه‌بندی نهایی انتخاب خواهند شد. عملیات boosting اغلب منجر به بهبود چشمگیری در صحت پیش‌بینی می‌شود[۷].
- انواع داده‌ای جدید (مثل تاریخ)، مقادیر "not applicable"، هزینه‌های غیرکلاسه‌بندی متغیر و مکانیزمهایی برای پیش فیلترینگ صفات خاصه.
- مجموعه قوانین بدون ترتیب؛ زمانی که یک مورد کلاسه‌بندی می‌شود، تمامی قوانین مرتبط و تاثیرگذار پیدا شده و انتخاب می‌شوند. این عمل باعث بهبود یافتن صحت تخمین می‌گردد.
- داشتن قابلیت توسعه بهبود یافته در درختان تصمیم‌گیری و (به خصوص) مجموعه قوانین. قابلیت توسعه توسط چندنخی (multithreading) بهبود می‌یابد. C۵.۰ می‌تواند از سیستمهای چندپردازنده‌ای و یا چند هسته‌ای بهره گیرد

۵. نتیجه‌گیری

داده‌کاوی علمی بسیار گسترده است که شامل تکنیکهایی از زمینه‌های مختلف از جمله یادگیری ماشین، آمار، شناسایی الگو، هوش مصنوعی و سیستمهای پایگاه داده به منظور تحلیل داده‌ها می‌شود. الگوریتمهای بسیار زیادی در این زمینه برای انجام تحلیل‌های گوناگون بر روی داده‌ها وجود دارند. در این مقاله پس از بررسی مباحثی در مورد داده‌کاوی از جمله کلاسه‌بندی و کلاسه‌بندی با استنتاج درختان تصمیم‌گیری، ساختار الگوریتم درختان تصمیم‌گیری بررسی شده و تعدادی از این الگوریتمها به صورت دقیق مورد بحث قرار گرفتند و به بررسی تفاوتهای آنها با یکدیگر پرداختیم. با توجه به نقش و اهمیت کلاسه‌بندی صحیح و کارآمد در داده‌کاوی و کاربرد گسترده و روز افزون داده‌کاوی در امور صنعتی و تجاری، اهمیت کلاسه‌بندی و مقولات مربوط به آن هر روز بیشتر می‌شود و حوزه فعال و پویایی را برای محققان فراهم می‌کند.

٦. مراجع

١. Jiawei Han, Micheline Kamber, *Data Mining: Concepts and Techniques*, Second Edition, Morgan Kaufmann Publishers, Publisher's name:Diane Cerra,٢٠٠٦

٢. XindongWu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou, Michael Steinbach, David J. Hand, Dan Steinberg, "*Top ١٠ algorithms in data mining"*, ICDM, http://www.cs.uvm.edu/~icdm/), Springer-Verlag London Limited, ٢٠٠٧

٣. Breiman L, Friedman JH, Olshen RA, Stone CJ, "*Classification and regression trees*." Wadsworth, Belmont, ١٩٨٤

٤. Salvatore ruggieri, "Efficient C٤.٥". IEEE transactions on knowledge discovery and data engineering". Vol ١٤, no ٢, march/april ٢٠٠٢.

٥. Shihai Zhang, Shujun Liu, Shizhong Zhang, Jinping Ou, Guangyuan Wang, "*C٤.٥-based Classification Rules Mining of High-rise Building SFIO"*. Fifth International Conference on Fuzzy Systems and Knowledge Discovery, IEEE ٢٠٠٨.

٦. Quinlan JR (١٩٧٩), "*Discovering rules by induction from large collections of examples*. In: Michie D (ed), Expert systems in the micro electronic age. Edinburgh University Press, Edinburgh

٧. Freund Y, Schapire RE (١٩٩٧) "*A decision-theoretic generalization of on-line learning and an application to boosting"*. J Comput Syst Sci ٥٥(١):١١٩–١٣٩

٨. http://rulequest.com/see٥-comparison.html, RULEQUEST RESEARCH ٢٠٠٨

٩. http://www.sgi.com/tech/mlc/db/sleep.all, http://www.sgi.com/tech/mlc/db/sleep.data, http://www.sgi.com/tech/mlc/db/sleep.names, http://www.sgi.com/tech/mlc/db/sleep.test

١٠. http://kdd.ics.uci.edu/databases/census-income/census-income.html

١١. http://kdd.ics.uci.edu/databases/covertype/covertype.html