دوره درسی ترم ۲ سال ۱۳۹۱

داده کاوی

میثم مدنی
دانشگاه صنعتی شریف
فصل هشتم،
http://www.madani.pro
Email: me@madani.pro

طبقه‌بندی

۱ طبقه‌بندی

داده کاوی

میثم مدنی

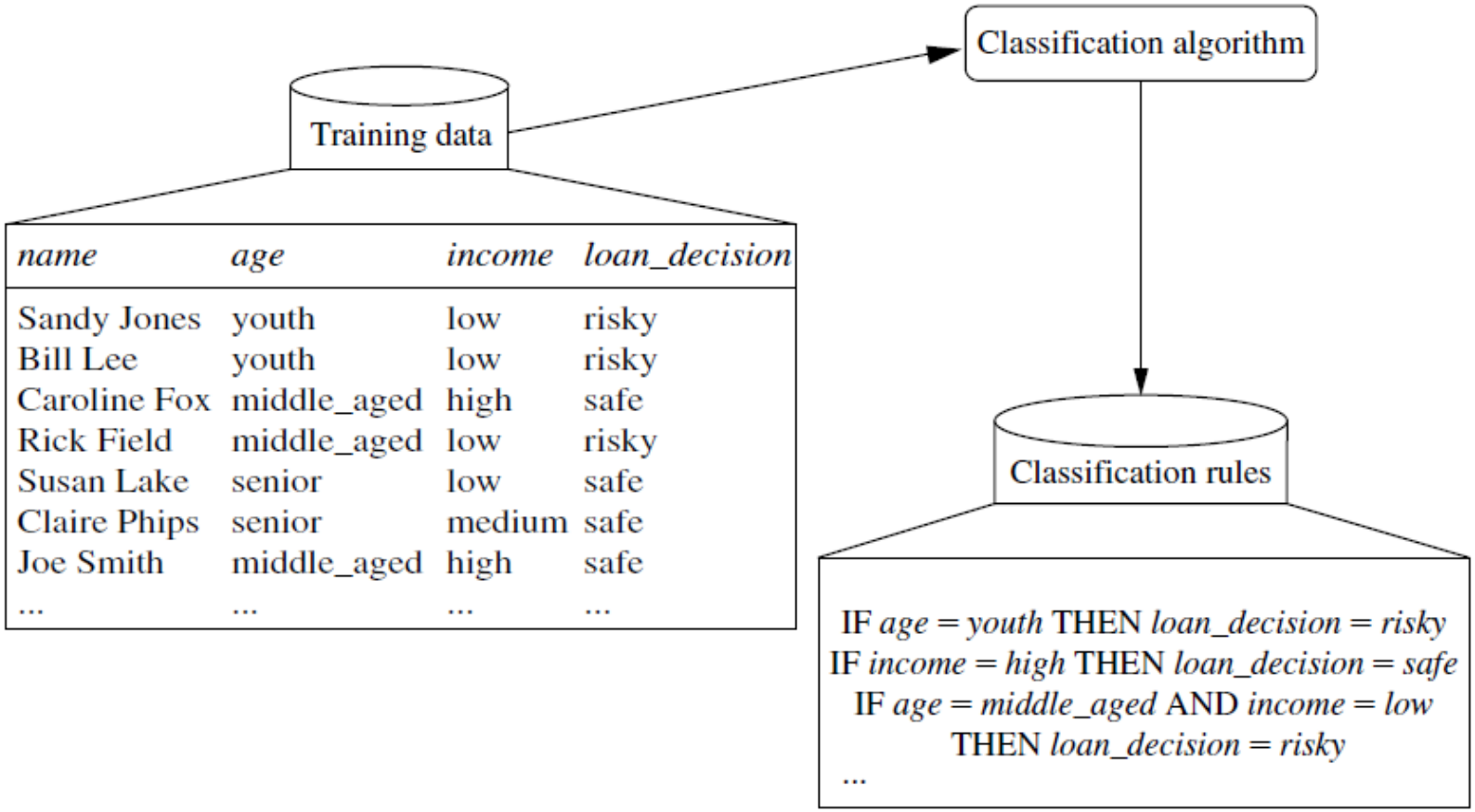
مفاهیم عمومی

طبقه‌بندی

- یک بانک می خواهد از روی داده هایش یاد بگیرد که چه قرض دادنی **امن** و چه **ریسکی** است.
- مدیر یک شرکت می خواهد از روی داده ها یاد بگیرد که یک مشتری کامپیوتر **می‌خرد** یا **نه**؟
- یک محقق پزشک می خواهد بداند که با توجه به سوابق و اطلاعات یک بیمار سرطان **دارد** یا **ندارد**

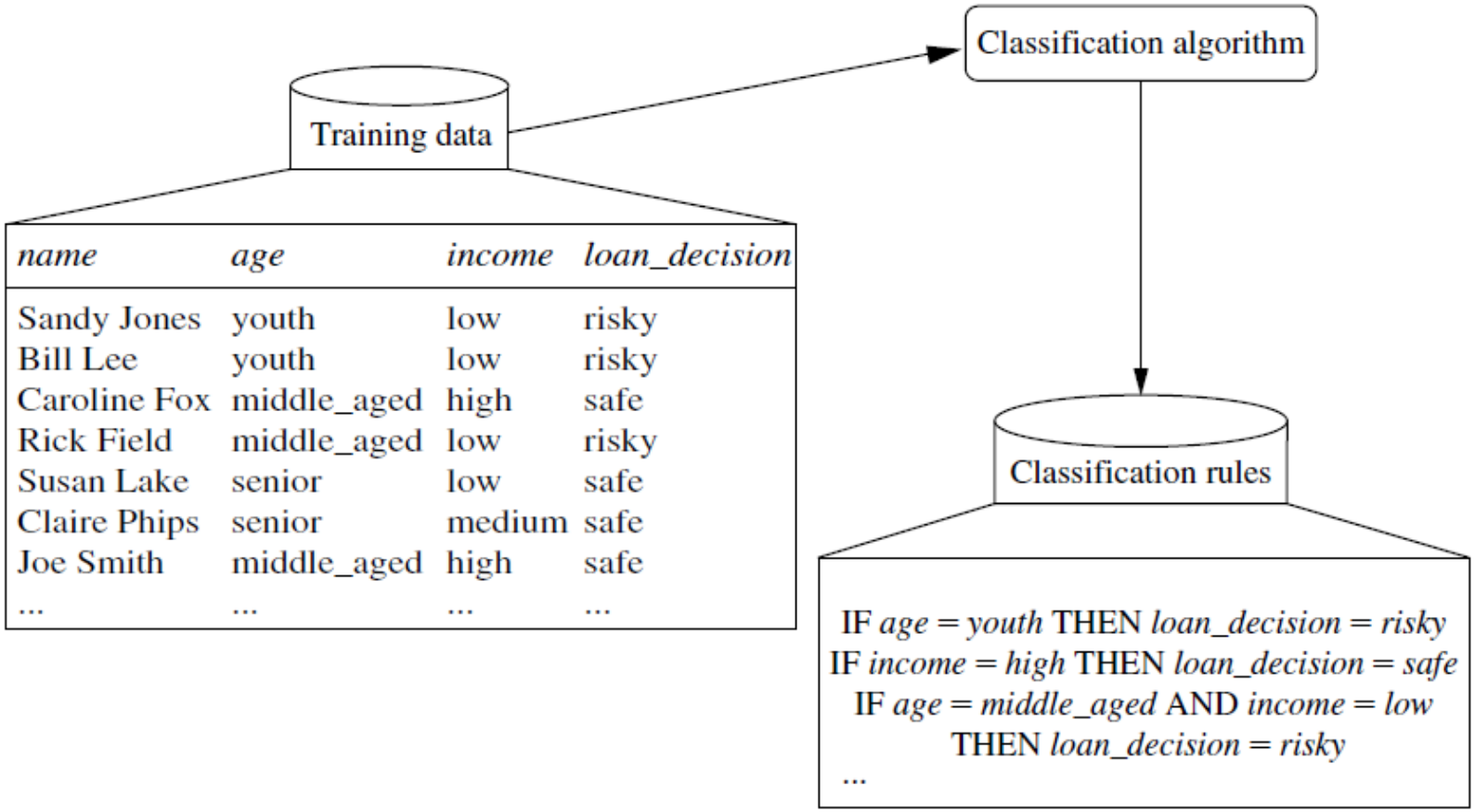
طبقه‌بندی تعیین برچسب یک داده است.

طبقه بندی



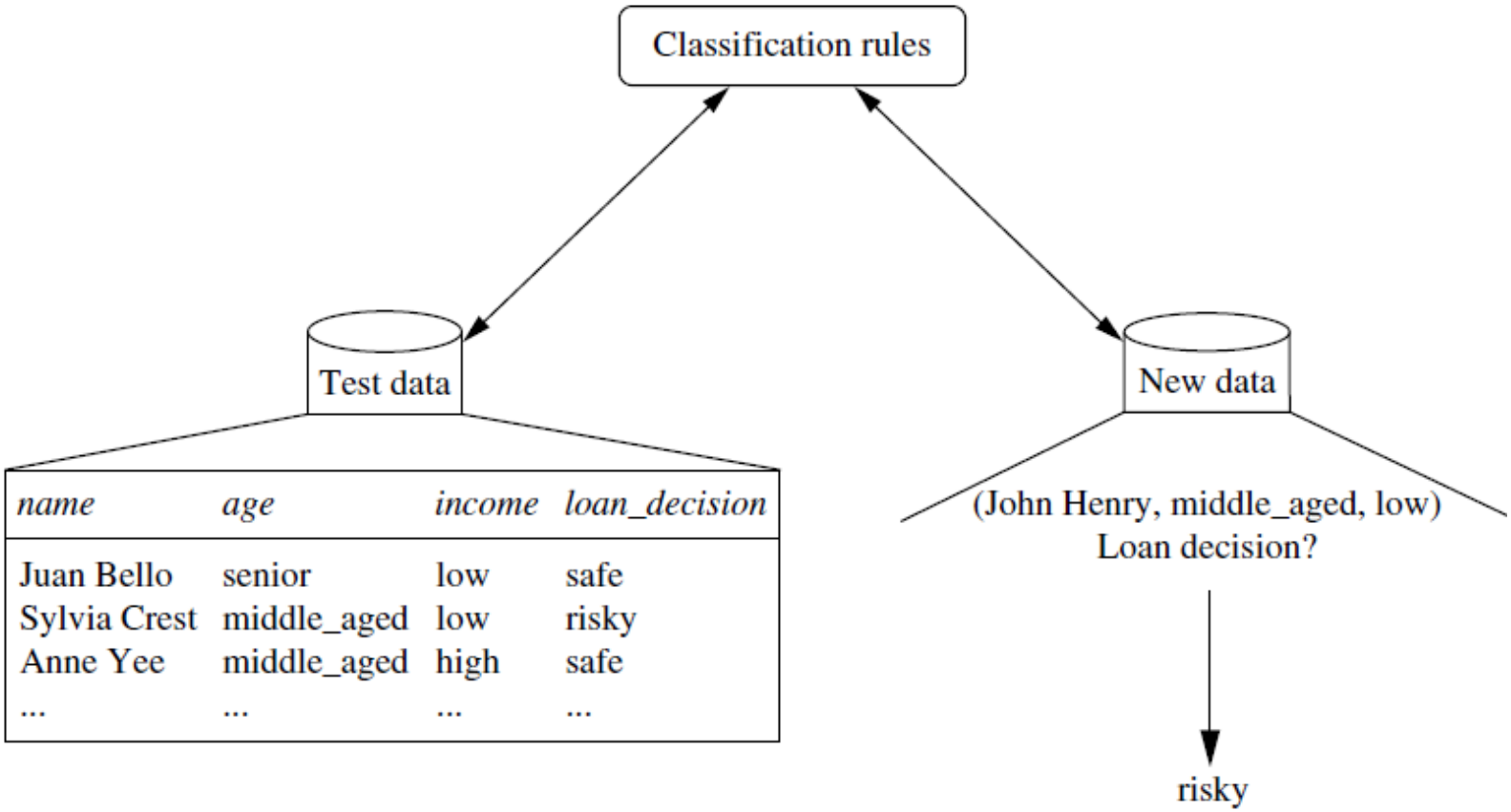
name	*age*	*income*	*loan_decision*
Sandy Jones	youth	low	risky
Bill Lee	youth	low	risky
Caroline Fox	middle_aged	high	safe
Rick Field	middle_aged	low	risky
Susan Lake	senior	low	safe
Claire Phips	senior	medium	safe
Joe Smith	middle_aged	high	safe
...	...	...	...

طبقه بندی



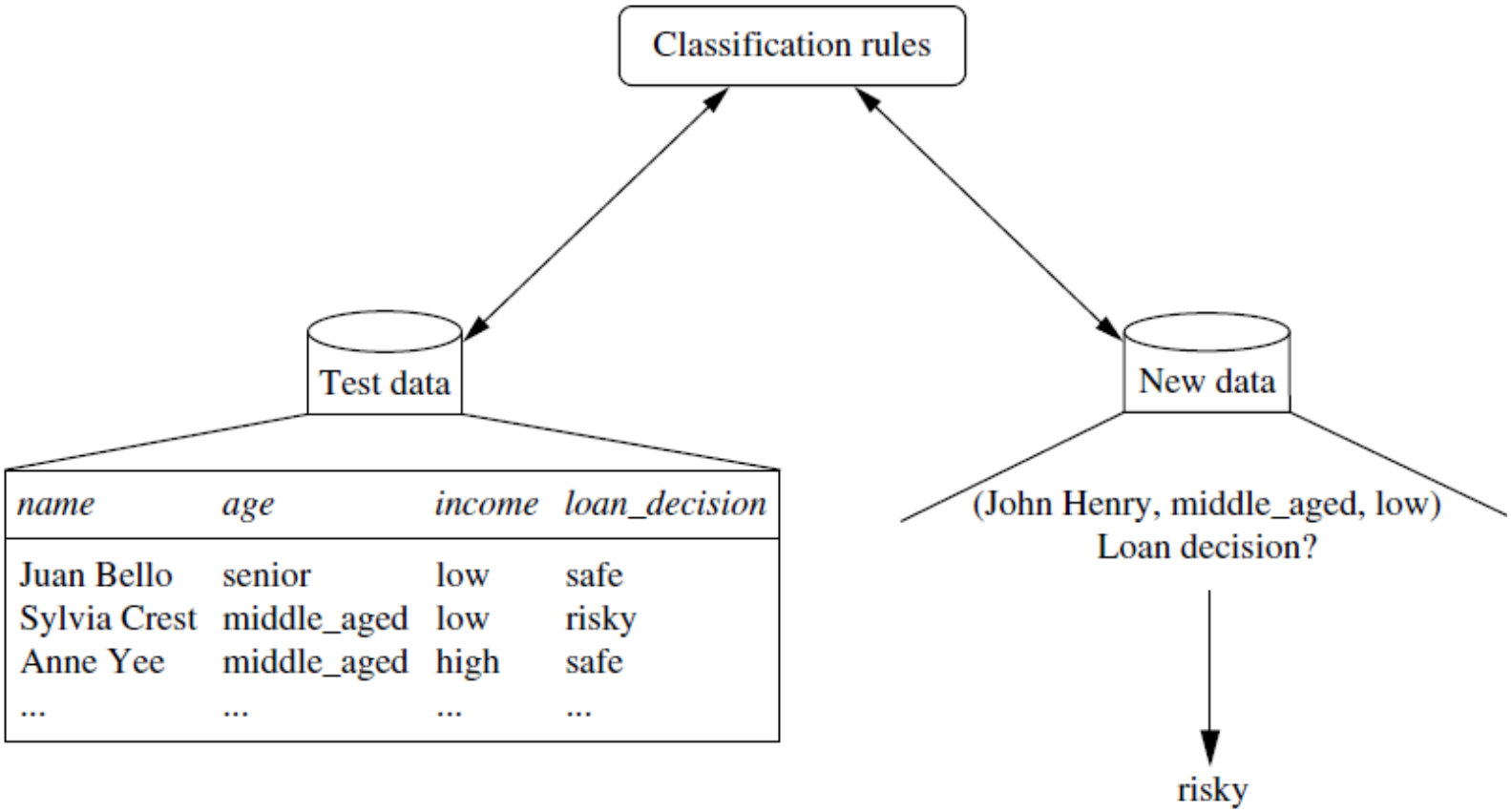
name	*age*	*income*	*loan_decision*
Sandy Jones	youth	low	risky
Bill Lee	youth	low	risky
Caroline Fox	middle_aged	high	safe
Rick Field	middle_aged	low	risky
Susan Lake	senior	low	safe
Claire Phips	senior	medium	safe
Joe Smith	middle_aged	high	safe
...	...	...	...

طبقه بندی



name	*age*	*income*	*loan_decision*
Juan Bello	senior	low	safe
Sylvia Crest	middle_aged	low	risky
Anne Yee	middle_aged	high	safe
...	...	...	...

طبقه بندی



name	*age*	*income*	*loan_decision*
Juan Bello	senior	low	safe
Sylvia Crest	middle_aged	low	risky
Anne Yee	middle_aged	high	safe
...	...	...	...

داده کاوی

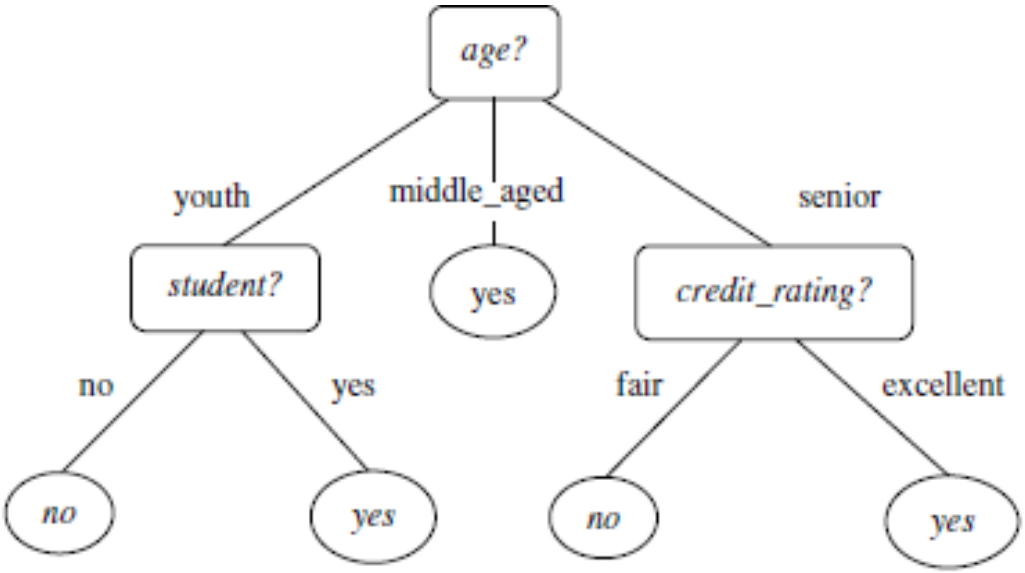
میثم مدنی

درخت تصمیم

استنتاج از روی درخت تصمیم

استنتاج از روی درخت تصمیم در واقع یادگیری درخت تصمیم از روی داده های با برچسب کلاس تمرینی هستند.
یک درخت تصمیم ساختاری است که

- هر راس داخلی یک آزمون را روی یک خصیصه انجام می دهد،
- هر انشعاب، نمایانگر خروجی آزمون،
- هر برگ، برچسب یک کلاس،



درخت تصمیم

- راس‌های داخلی با مستطیل مشخص می شوند.
- درخت تصمیم به سادگی می تواند به قواعد طبقه بندی تبدیل شود.
- فهم و درک درخت تصمیم ساده است و می تواند داده های چند بعدی را طبقه بندی کنند.

۱. اواخر دهه ۱۹۷۰ و ابتدای ۱۹۸۰ J.Ross Quinlan یک الگوریتم درخت تصمیم را که به عنوان ID3(Iterative Dichotomiser) شناخته می شود ارائه کرد
۲. بعدها B. Hunt, J. Marin, P. Stone این الگوریتم را توسعه دادند.
۳. خود J.Ross Quinlan الگوریتم قبلی اش را به C4.5 توسعه داد.
۴. بعدها یک گروه از آماردان ها کتابی با عنوان CART(Classification and Regression Trees) منتشر کردند که تولید درخت دودویی تصمیم را توصیف می کرد

الگوریتمی بلند اما ساده

۱ سه پارامتر داریم
D, Attribute_list, Attribute_Selection, Attribute_method

۲ با یک رأس تنها N شروع می کنیم که بردارهای تمرینی را در داده D مشخص می کند.

۳ اگر تمام بردارها در D یکسان بودند N یک برگ است و با آن کلاس مشخص می شود.

۴ Attribute_method را فرا می خوانیم. و شرط شکاف را می یابیم.
شرط شکاف انشعاب ها و خصیصه شکافتن در آن مرحله را به بهترین نحو ممکن تعیین می کند.

۵ اگر رأس N با یک شرط شکاف برچسب خورد، آن شاخه رشد پیدا می کند. سه حالت ممکن است اتفاق بیفتد. فرض کنید خصیصه شکاف A بتواند v مقدار به خود بگیرد $a_1, a_2, \ldots, a_v$

۱ A گسسته باشد. برای هر حالت یک انشعاب ایجاد می کنیم.

۲ A پیوسته باشد. در اینصورت از کوچکتر یا بزرگتری نقاط شکاف استفاده کرده

۳ برای حالت گسسته می توان از عضویت نیز استفاده نمود $A \in S_A$ که در آن S_A زیر مجموعه شکافنده است.

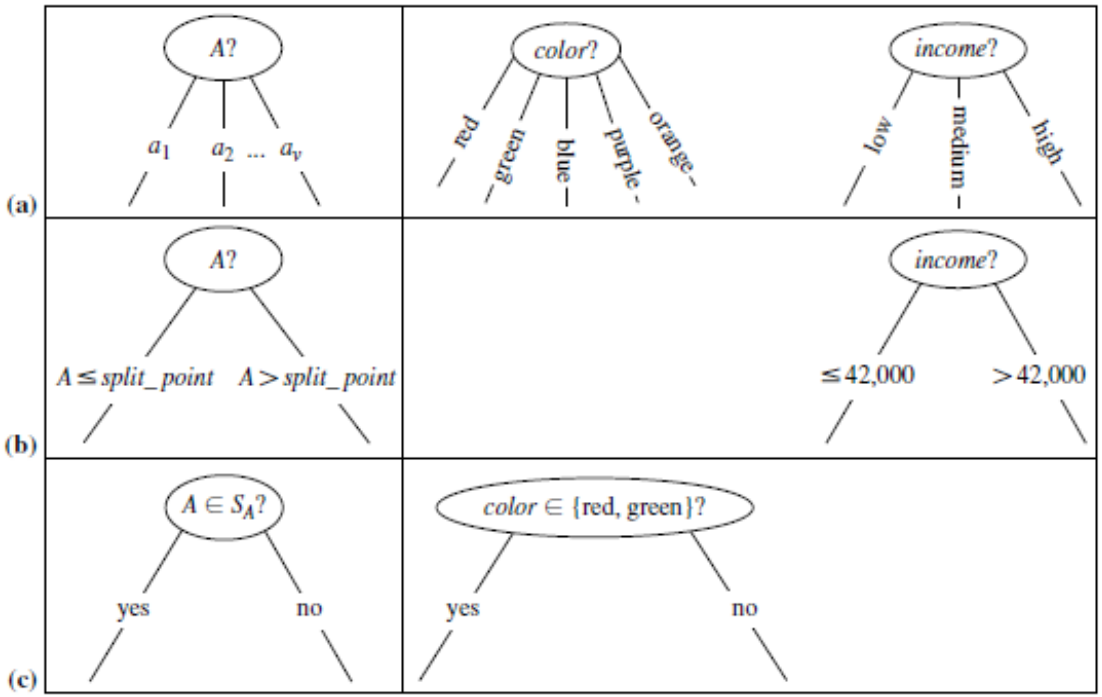
۶ الگوریتم به طور بازگشتی همین فرآیندها را ادامه می دهد تا یک درخت تصمیم از روی داده های کنونی D_j بسازد.

۷ محک‌های توقف الگوریتم

۱ تمامی بردارها به یک کلاس متعلق باشند

۲ هیچ خصیصه ای برای شکافته شدن نمانده باشد.

۳ برای یک شاخه هیچ برداری وجود نداشته باشد.



پیچیدگی محاسباتی این روش $O(n \times |D| \times \log(|D|))$ است که در آن n تعداد

الگوریتم

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 criterion consists of a *splitting_attribute* and, possibly, either a *split-point* or *splitting subset*.

Output: A decision tree.

الگوریتم

(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 labeled with the class C;
(4) **if** *attribute_list* is empty **then**
(5) return N as a leaf node labeled with the majority class in D; // majority voting
(6) apply **Attribute_selection_method**(D, *attribute_list*) to **find** the "best" *splitting_criterion*;
(7) label node N with *splitting_criterion*;
(8) **if** *splitting_attribute* is discrete-valued **and**
multiway 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 D_j be the set of data tuples in D satisfying outcome j; // a partition
(12) **if** D_j is empty **then**
(13) attach a leaf labeled with the majority class in D to node N;
(14) **else** attach the node returned by **Generate_decision_tree**(D_j, *attribute_list*) to node N;
endfor
(15) return N;

داده کاوی

میثم مدنی

اندازه‌هایی برای انتخاب ویژگی

اندازه‌هایی برای انتخاب ویژگی

۱ اندازه‌هایی برای انتخاب ویژگی روش‌هایی ابتکاری هستند محک شکافی را می‌یابند که به بهترین نحو داده ها موجود را افراز می‌کند.

۲ این اندازه، قاعده شکاف هم گفته می‌شود.

۳ قاعده شکاف یک رتبه بندی را نیز برای خصیصه ها در نظر می‌گیرد.

۴ قاعده ای که بهترین اندازه را داشته باشد به عنوان خصیصه شکاف در نظر گرفته می‌شود.

بهره اطلاعات Information Gain

۱ اطلاعات لازم برای طبقه‌بندی یک بردار

$$\mathrm{Info}(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

است که p_i احتمال ناصفر این است که یک داده به کلاس C_i مطعلق باشد و معمولا $\frac{|C_{i,D}|}{|D|}$ می باشد.

۲ این مقدار به نوعی همان آنتروپی هستند.

۳ فرض کنید یک خصیصه A داریم که دارای v مقدار متمایز هستند.

$$a_1, a_2, \ldots, a_v$$

۴ مقدلر اطلاعات لازم که برای یک طبقه بندی دقیق نیاز داریم

$$\mathrm{Info}_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \mathrm{Info}(D_j)$$

۵ در هر مرحله خصیصه ای با بیشترین بهره اطلاعات را انتخاب می کنیم.

$$\mathrm{Gain}(A) = \mathrm{Info}(D) - \mathrm{Info}_A(D)$$

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

$$Info(D) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940 \text{ bits}.$$

$$Info_{age}(D) = \frac{5}{14} \times \left(-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}\right) = 0.694 \text{ bits}.$$

$$Gain(age) = Info(D) - Info_{age}(D) = 0.940 - 0.694 = 0.246 \text{ bits}.$$

$$Gain(income) = 0.029 \text{ bits},$$
$$Gain(student) = 0.151 \text{ bits},$$
$$Gain(credit\_rating) = 0.048 \text{ bits}.$$

youth

income	*student*	*credit_rating*	*class*
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

middle_aged

income	*student*	*credit_rating*	*class*
high	no	fair	yes
low	yes	excellent	yes
medium	no	excellent	yes
high	yes	fair	yes

senior

income	*student*	*credit_rating*	*class*
medium	no	fair	yes
low	yes	fair	yes
low	yes	excellent	no
medium	yes	fair	yes
medium	no	excellent	no

بهره اطلاعاتی خصیصه های پیوسته

۱ بسیاری از اوقات خصیصه ما پیوسته است یا تعداد حالات بسیاری دارد. (هر پیوسته ای در کامپیوتر گسسته است!)

۲ بایستی بهترین نقطه شکاف را تعیین کنیم که نوعی آستانه برای آن خصیصه است.

۳ ابتدا خصیصه A را به صورت صعودی مرتب می‌کنیم.

۴ وسط هر دو مقدار A یک نقطه شکاف خواهد بود. پس $v - 1$ نقطه شکاف خواهیم داشت.

$$\frac{a_j + a_{j+1}}{2}$$

۵ برای هر نقطه شکاف D_1 داده هایی است با مشخصه $A \leq split - point$ و D_2 هم داده هایی که $A > split - point$.

۶ برای هر حالت ممکن مقدار $Info_A(D)$ را حساب می کنیم و نقطه با کمترین مقدار انتخاب می‌شود.

۷ روش ID3 از بهره اطلاعات استفاده می کند.

نرخ بهره Gain Ratio

۱ بهره اطلاعات بیشتر تمایل به تعداد زیاد قطعه بندی دارد. یعنی هرچه تعداد طبقاتی که یک خصیصه تقسیم باشد بهره اطلاعات ی بیشتری را برای آن در نظر می گیرد.

۲ مثلا $\text{product} - \text{id}$ تعداد زیادی مقدار دارد. هر هر افراز هم یک مقدار دارد پس در نهایت $\text{Info}_{\text{product}-\text{ID}}(D) = 0$!

۳ روش C4.5 از یک مقدار بهتر استفاده می کند به نام نرخ بهره.

۴

$$\text{Info}_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \text{Info}(D_j), \quad \text{Gain}(A) = \text{Info}(D) - \text{Info}_A(D)$$

$$\text{SplitInfo}_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

$$\text{GainRatio}(A) = \frac{\text{Gain}(A)}{\text{SplitInfo}_A(D)}$$

شاخص جینی Gini Index

۱ از شاخص جینی در الگوریتم CART استفاده می شود.
۲ شاخص جینی به صورت زیر تعریف می‌شود.

$$\mathrm{Gini}(D) = 1 - \sum_{j=1}^{m} p_i^2$$

که p_i احتمال اختصاص یک داده به کلاس C_i است.
۳ شاخص جینی یک شکاف دودویی برای هر خصیصه ایجاد می‌شود.
۴ حالت دودویی یعنی انتخاب یک مجموع S_A چنانکه دو کلاس داده هایی که خصیصه آن در این مجموعه باشد یا نه.
۵ نکته اینجاست که $2^v - 2$ مجموعه داریم.
۶ شاخص جینی هر خصیصه برابر است با

$$\mathrm{Gini}_A(D) = \frac{|D_1|}{|D|}\mathrm{Gini}(D_1) + \frac{|D_2|}{|D|}\mathrm{Gini}(D_2)$$

۷ مجموعه ای که کمترین مقدار شاخص را بگیرد انتخاب می‌شود.
۸ در حالت پیوسته نیز از نقاط شکاف مشابه نرخ بهره استفاده می‌کنیم

روش‌های طبقه‌بندی بیزی Bayes

قضیه بیزی

۱ روش بیزی یک روش آماری برای طبقه بندی است.

۲ احتمال عضویت در هر کلاس را می‌سنجد و طبقه بندی صورت می گیرد.

۳ فرض کنید X یک بردار داده باشد.این داده یک شاهد یا نمونه است.

۴ فرض کنید H یک فرض باشد که داده X به کلاس C تعلق دارد.

۵ برای طبقه بندی ما بایستی $P(H|X)$ را محاسبه کنیم.

۶ $P(H|X)$ یک احتمال استقرائی یا پیشین است.

۷ قضیه بیزی

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}$$

۸ مثال آماری قضیه

فرض کنید دو کیسه یکی شامل ۵ مهره قرمز و ۳ مهره آبی و دیگری ۸ مهره آبی و ۳ مهره قرمز داریم. یک مهره بیرون آوریم و آبی است. احتمال اینکه متعلق به کیسه دوم باشد چقدر است؟ $\frac{8/11 \times 1/2}{11/19}$

طبقه بندی ساده بیزی Naive Bayesian

۱ فرض D داده ها و هر داده یک بردار $X = (x_1, x_2, ..., x_n)$, نیز $A_1, ..., A_n$ اندازه هر خصیصه باشد.

۲ فرض m کلاس $C_1, ..., C_m$ داریم.داده X کلاس C_i تعلق دارد اگر و تنها اگر

$$P(C_i|X) > P(C_j|X)$$

از طرفی با قضیه بیزی

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}$$

۳ با توجه با ثابت بودن $P(X)$ کافیست $P(X|C_i)P(C_i)$ را ماکزیمم کنیم. و با توجه به اینکه معمولا فرض می شود کلاسها هم اندازه هستند $P(C_1) = P(C_2) = ... = P(C_m)$ کافیست $P(X|C_i)$ را ماکزیمم کنیم.

۴ در حجم عظیم داده ها از نظر محاسباتی حساب کردن $P(X|C_i)$ هم سخت است. از شرط **استقلال شرطی کلاس** استفاده می کنیم. یعنی مقادیر خصیصه ها از هم مستقل هستند.

$$P(X|C_i) = P(x_1|C_i) \times P(x_2|C_i) \times ... \times P(x_n|C_i)$$

طبقه بندی ساده بیزی Naive Bayesian

۴ می توانیم تخمینی از $P(x_1|C_i) \times P(x_2|C_i) \times \ldots \times P(x_n|C_i)$ به دست آوریم. x_i مقدار داده در خصیصه A_i است.

- اگر A_k گسسته باشد $P(x_k|C_i)$ تعداد بردارهای با مقدار x_k در خصیصه A_k تقسیم بر تعداد $|C_{i,D}|$ تعداد داده های کلاس C_i است.
- اگر A_k پیوسته باشد فرض می شود که توزیع گاوسی با میانگین μ و انحراف معیار σ است.

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma}}$$

- تعریف می کنیم $P(x_k|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i})$ که μ_{C_i}, σ_{C_i} به ترتیب انحراف معیار و میانگین کلاس C_i در خصیصه A_k است.

در ظاهر خطای این روش از سایر روشها کمتر است. با این حال در عمل همیشه اینچنین نیست.

طبقه‌بندی قاعده مبنا

قواعد اگر آنگاه

۱ یک روش استنتاج قواعد، تبدیل درخت تصمیم به قواعد است.

۲ یک روش دیگر هم استخراج مستقیم قاعد از داده هاست. یکی از چنین الگوریتمها الگوریتم پوشش ترتیبی است.

- نام ترتیبی یا دنباله ای به خاطر یادگیری مرحله به مرحله است.
- الگوریتم های پوششی ترتیبی متعددی وجود دارند مانند AQ,CN2, RIPPER .
- هر قاعده در یک زمان یاد گرفته می شود
- هر بار که یک قاعده یادگرفته می شود داده هایی که با آن قاعده پوشش داده می شوند حذف می شوند.
- از همان روش هایی که برای انشعاب و انتخاب ویژگی استفاده می کردیم استفاده می کنیم.

Algorithm: Sequential covering. Learn a set of IF-THEN rules for classification.

Input:

- D, a data set of class-labeled tuples;
- *Att_vals*, the set of all attributes and their possible values.

Output: A set of IF-THEN rules.

Method:

```
(1)  Rule_set = {}; // initial set of rules learned is empty
(2)  for each class c do
(3)      repeat
(4)          Rule = Learn_One_Rule(D, Att_vals, c);
(5)          remove tuples covered by Rule from D;
(6)          Rule_set = Rule_set + Rule; // add new rule to rule set
(7)      until terminating condition;
(8)  endfor
(9)  return Rule_Set;
```

کیفیت یک قاعده

فرض کنید یک داده X داریم و n_{covers} تعداد داده های پوشیده شده با قاعده R باشد و $n_{correct}$ تعداد داده هایی باشد که با آن قاعده درست حدس زده شده اند.

1. **پوشش**

$$coverage(R) = \frac{n_{covers}}{|D|}$$

2. **دقت**

$$accuracy(R) = \frac{n_{correct}}{n_{covers}}$$

انتخاب و برآورد مدل

ندازه هایی برای برآورد طبقه بندی

- **داده مثبت** داده هایی که به کلاس مورد نظر ما تعلق دارند.
- **داده منفی** داده هایی که به کلاس مورد نظر ما تعلق ندارند.

چهار اصطلاح وجود دارند که ما اغلب برای برآورد مدلها از آن استفاده می کنیم.

۱. **مثبت صحیح TP** داده مثبتی که درست طبقه بندی شده باشد.
۲. **منفی صحیحTN** داده منفی که درست طبقه بندی شده باشد
۳. **مثبت کاذب FP** داده مثبتی که اشتباه طبقه بندی شده باشد.
۴. **منفی کاذب FN** داده منفی که اشتباه طبقه بندی شده باشد.

		Predicted class		
		yes	*no*	Total
Actual class	*yes*	*TP*	*FN*	*P*
	no	*FP*	*TN*	*N*
	Total	P'	N'	$P + N$

Classes	*buys_computer* = *yes*	*buys_computer* = *no*	*Total*	*Recognition (%)*
buys_computer = *yes*	**6954**	**46**	7000	99.34
buys_computer = *no*	**412**	**2588**	3000	86.27
Total	7366	2634	10,000	95.42

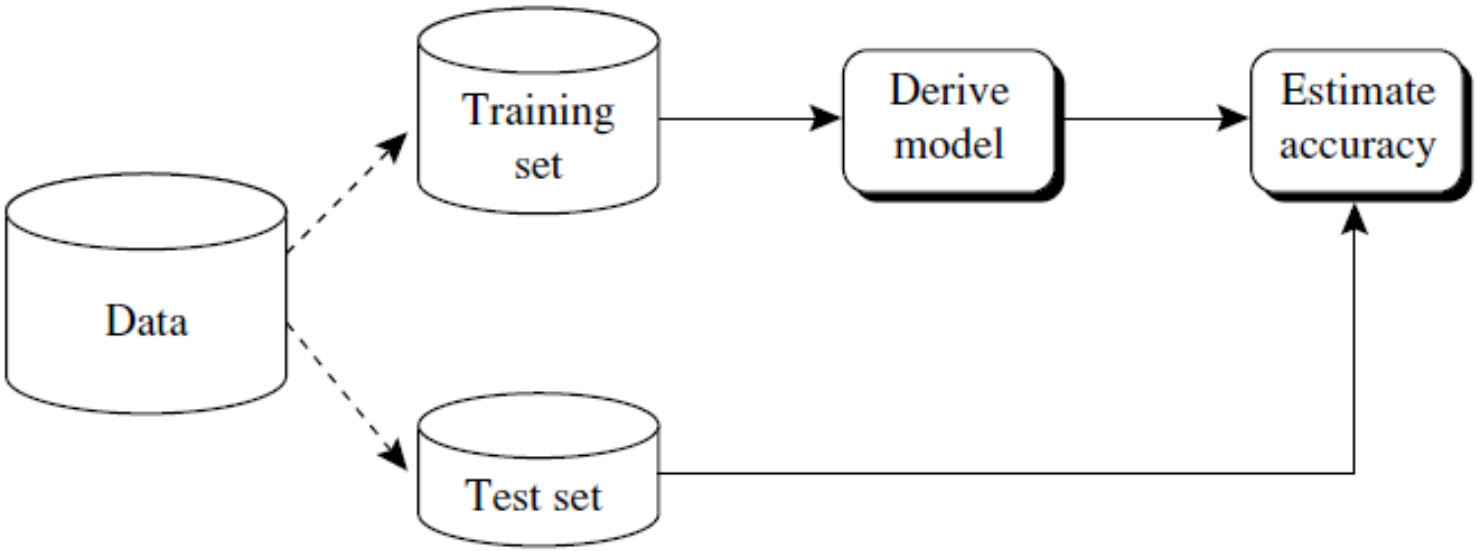
Measure	Formula
accuracy, recognition rate	$\frac{TP+TN}{P+N}$
error rate, misclassification rate	$\frac{FP+FN}{P+N}$
sensitivity, true positive rate, recall	$\frac{TP}{P}$
specificity, true negative rate	$\frac{TN}{N}$
precision	$\frac{TP}{TP+FP}$
F, F_1, F-score, harmonic mean of precision and recall	$\frac{2 \times precision \times recall}{precision + recall}$
F_β, where β is a non-negative real number	$\frac{(1+\beta^2) \times precision \times recall}{\beta^2 \times precision + recall}$

اندازه هایی برای برآورد طبقه بندی

- **سرعت** کم بودن هزینه های محاسباتی
- **قدرت** حدس صحیح با و وجود داده های نویزدار.
- **سازگاری** با افزایش تعداد داده ها توانایی خود را از دست ندهد
- **قابل تفسیر بودن** قابل فهم بودن (مثل درخت تصمیم)

روش Holdout و بازنمونه گیری تصادفی

- در روش Holdout داده ها به صورت تصادفی به دو بخش افراز می شوند
  1. **مجموعه یادگیری training set** معمولا دو سوم داده ها را در بر می گیرد و برای ایجاد مدل استفاده می شود
  2. **مجموعه تست test set** یک سوم داده ها و برای ارزیابی مدل به کار می رود
- **بازنمونه گیری تصادفی** یک حالت تغییر یافته از روش بالا است که k بار آن را تکرار می کند. دقت کلی میانگینی از دقت های به دست آمده است.



وارسی اعتبار cross-validation

- در وارسی اعتبار kتایی داده ها به صورت تصادفی به k دسته زیر محموعه $D_1, D_2, ..., D_n$ که fold هستند تقسیم می شود.
- فولدها تقریبا یک اندازه هستند
- k بار یادگیری و تست انجام می شود.
- در تکرار i ام داده D_i به عنوان تست محسوب می شود و بقیه به عنوان داده یادگیری
- مثلا در گام اول $D_2, ..., D_n$ برای یادگیری و D_1 برای تست.
- برخلاف روش های دیگر هر داده به تعداد مساوی هم برای یادگیری به کار می رود و هم برای آموزش
- دقت برابر مجموع حدس های درست تقسیم بر تعداد کل داده هاست.
- **level-one-up** یکی از حالتهای خاص روش وارسی اعتبار است که k برابر تعداد کل داده هاست.
- **stratified cross-validation** فولدها به گونه ای هستند که به تعداد تقریبا مساوی از هر کلاس را د رخود داشته باشد.
- $k = 10$ عدد شناخته شده و مناسبی برای اغلب داده هاست.

روش Bootstrapping

- برخلاف دیگر روش ها در این روش داده‌ها به صورت یکنواخت با جایگذاری نمونه‌گیری می‌شوند
- روش‌های متعددی برای Bootstrap وجود دارد، یکی از معروف‌ترین‌ها روش .632 است، فرض کنید d سطر داریم.
  1. d بار نمونه‌گیری)با جایگذاری(انجام می‌شود تا یک مجموعه آموزش داده تشکیل شود/
  2. داده‌هایی که به عنوان داده آموزشی ظاهر نشدند را به عنوان داده آزمون در نظر می‌گیریم.
  3. با توجه به محاسبات، حدود 63.2% از داده‌ها در داده آموزشی قرار می‌گیرند و 36.8% نیز به عنوان داده آزمون مطرح خواهند شد.
  4. k بار نمونه‌گیری فوق را انجام می‌دهیم و هر بار یک test_set و یک train_set به دست می‌آوریم. در نهایت دقت مدل مورد نظر ما برابر خواهد شد با

$$\mathrm{Acc(M)} = \sum_{i=1}^{k} (0.632 \times \mathrm{Acc(M_i)_{test\_set}} + 0.368 \times \mathrm{Acc(M_i)_{test\_set}}).$$

منظور از $\mathrm{Acc(M_i)_{test\_set}}$ دقت مدل روی داده‌های آزمون iام است.