داده کاوی

فصل هشتم

تهیه شده توسط
محمد پورزعفرانی
Pourzaferani.ir
دانشگاه اصفهان

دانشگاه اصفهان

تشکر

- رضا مفیدی
- عقیل استوار
- نرگس شجاعی
- محمد مهدی صالحی
- منصوره عسکری نژاد
- سمانه یوسفی
- هادی مویی
- فاطمه موسوی
- مینا شمسی زاده
- امین چابک رو

دسته‌بندی : مفاهیم اولیه

- مفاهیم اولیه
- درخت تصمیم (Decision Tree)
- روش‌های دسته‌بندی بیز (Bayes)
- روش‌های دسته‌بندی مبتنی بر قانون (Rule-Based)
- ارزیابی مدل‌های و انتخاب مدل مناسب
- روش‌های بهبود دقت دسته‌بندی
- خلاصه فصل

دسته‌بندی

- نام‌های دیگر : طبقه‌بندی، رده بندی، کلاسیفیکیشن
- هدف از این فصل (دسته‌بندی) استخراج مدل‌هایی است که کلاس‌های داده را توصیف می‌کنند
- این مدل‌ها دو دسته هستند
  1. دسته‌بندی (Classification)
  2. پیش‌بینی (Prediction)

کاربردهای این فصل

- تشخیص تقلب (Fraud detection) : معامله جعلی است یا خیر؟
- بازاریابی هدفمند (Target Marketing)
- تشخیصات پزشکی و .. (Medical deagnosis)
  - تشخیص اینکه تومور سرطانی است یا خوش‌خیم

What is Classification?

- مثال ۱: مدیر بانکی تصمیم می‌گیرد داده‌های خود را تحلیل نموده و مشخص کند اعطای وام به چه کسی «Safe» و به چه کسی «Risky» است
- مثال ۲: مدیر بازریابی یک شرکت می‌خواهد داده‌ها را تحلیل نماید و حدس بزند آیا مشتری قدیمی شرکت محصول جدید را خواهد خرید؟
- مثال ۳: یک پزشک محقق می‌خواهد از طریق تحلیل داده‌ها بفهمد برای درمان سرطان سینه از بین ۳ راه درمان، کدامیک موثرتر هستند؟
- در هر سه مثال کار ما در تحلیل داده‌ها، دسته‌بندی است
- در هر سه مثال نیاز به مدلی برای پیش‌بینی برچسب (لیبل) داده‌ها داریم
- مثال ۱: safe or risky
- مثال ۲: yes or no
- مثال ۳: treatment A, treatment B, treatment C

چگونه دسته‌بندی انجام می‌شود؟

- فرآیند دسته‌بندی در دو فاز انجام می‌گیرد

۱) فاز آموزش یا یادگیری (Learning step)

در این مرحله مدل ساخته می‌شود

۲) فاز دسته‌بندی (Classification step)

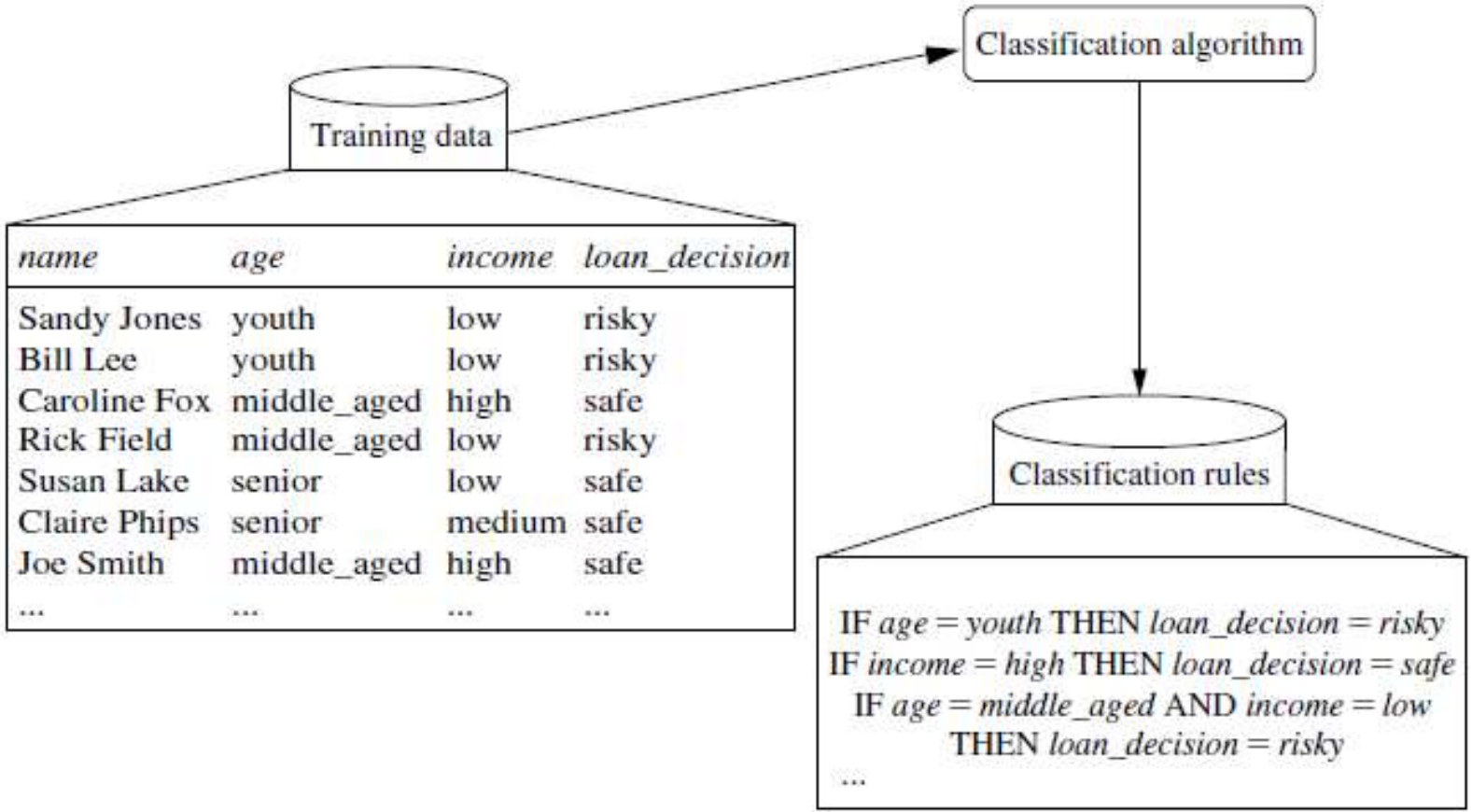
در این مرحله از مدل ساخته شده برای پیش‌بینی لیبل استفاده می‌شود

فاز اول

- در این فاز قسمتی از پایگاه داده برای تحلیل نمونه‌برداری می‌شود
- به این نمونه مجموعه آموزش (training set) گفته می‌شود
- هر تاپل از پایگاه داده با یک بردار X به صورت زیر نشان داده می‌شود
- $X = \{X1 , X2, X3, \dots ,Xn\}$
- هر کدام از خانه‌ها مشخص کننده یه ویژگی هستند
- هر تاپل بر اساس یک ویژگی به نام Class label attribute به یک کلاس خاص متعلق خواهد بود
- نکته : در موضوعیت دسته‌بندی تاپل را با نام‌های زیر نیز می‌شناسیم
- Samples, examples, instances, data points, objects

مثالی از فاز اول

- این فاز یادگیری با نظارت (Supervised learning) نیز گفته می‌شود چون برای هر داده مشخص است به چه کلاسی تعلق دارد



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
...	...	...	...

یادگیری بانظارت در مقابل بدون نظارت

- **یادگیری با نظارت (دسته‌بندی)**
  - داده‌های آموزشی (مشاهدات، اندازه‌گیری‌ها، و غیره) همراه با برچسب هستند که کلاس آن مشاهدات را نشان می‌دهند.
  - داده‌های جدید بر اساس مجموعه آموزشی دسته‌بندی می‌شوند.

- **یادگیری بدون نظارت (خوشه‌بندی)**
  - برای تاپل‌های داده، برچسب کلاس ناشناخته است
  - اغلب تعداد و نوع کلاس‌ها نیز از قبل مشخص نیست
  - به طور مثال در مثال بانک، وضعیت وام‌گیرنده مشخص نباشد. بنابراین از لحاظ شباهت افراد می‌توان ریسک داشتن دادن وام را بررسی نمود (فصل‌های ۱۰ و ۱۱)

ادامه فاز اول

- ارتباط بین داده‌ها و کلاس را گاهی اوقات با یکسری قانون، درخت تصمیم و یا یک فرمول ریاضی می‌توان نشان داد
- این قاعده‌ها به چه منظوری مورد استفاده قرارمی‌گیرند

۱) قاعده‌ها و قوانین پیدا شده برای دسته‌بندی داده‌های جدید می‌توانند به کار روند

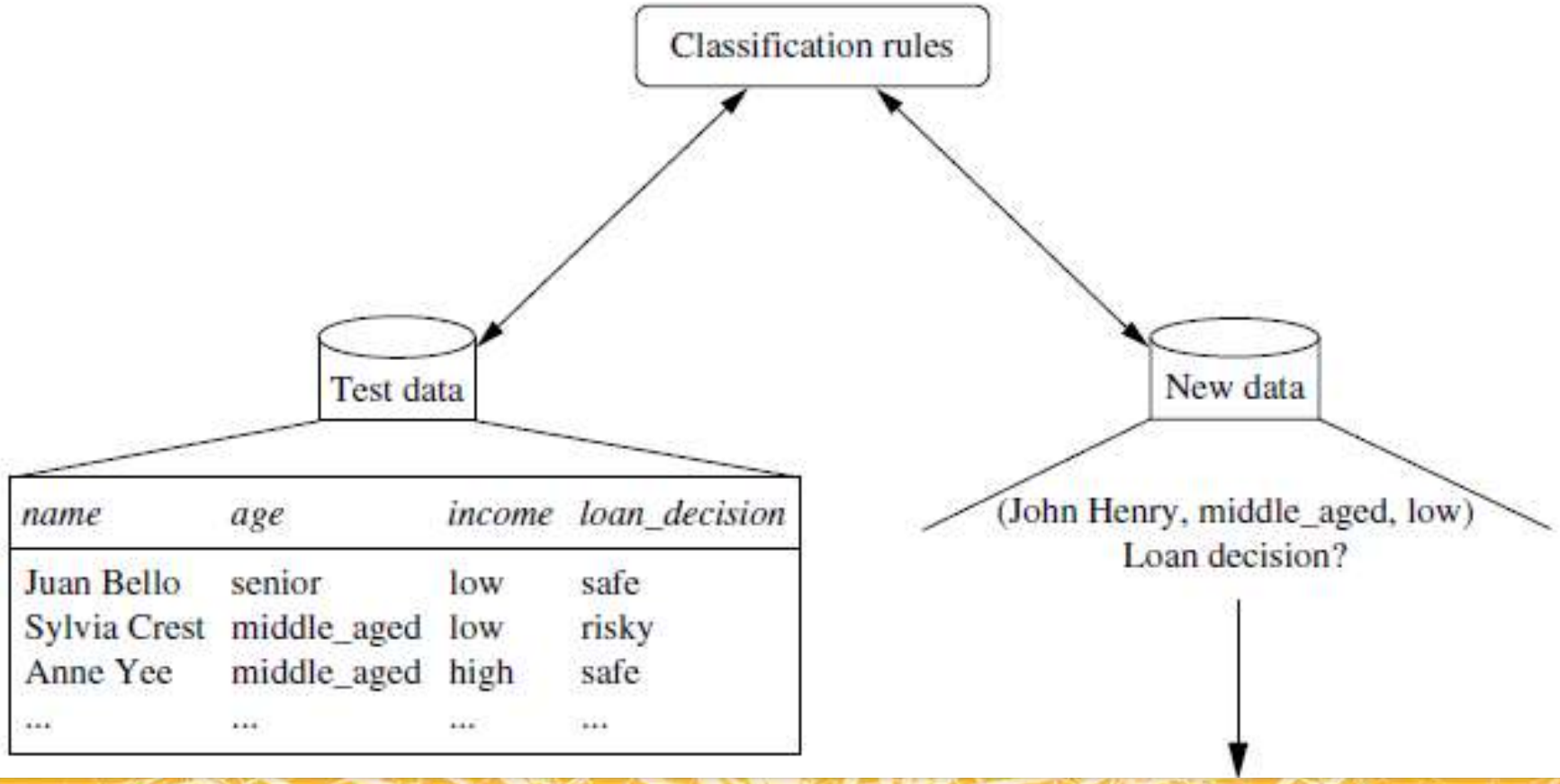
۲) دید واضح‌تری نسبت به داده‌های موجود می‌دهند

۳) نمایش داده‌ها را فشرده‌سازی می‌کنند

فاز دوم

- تا اینجا classifier ساخته شده است
- در این مرحله باید مدل ایجاد شده مورد ارزیابی قرار گیرد
- برای تست مدل از یک مجموعه تست (test set) استفاده می‌کنیم
- این مجموعه از مجموعه آموزش مستقل است
- تعریف دقت (accuracy) : درصدی از مجموعه تست را که classifier درست دسته‌بندی می‌کند
- زمانی که صحت کلاسیفایر تایید شد می‌تواند برای تشخیص برچسب داده‌های جدید (بدون لیبل) استفاده شود

فاز دوم



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

فرآیند (۱): ساخت‌وساز مدل

Training Data

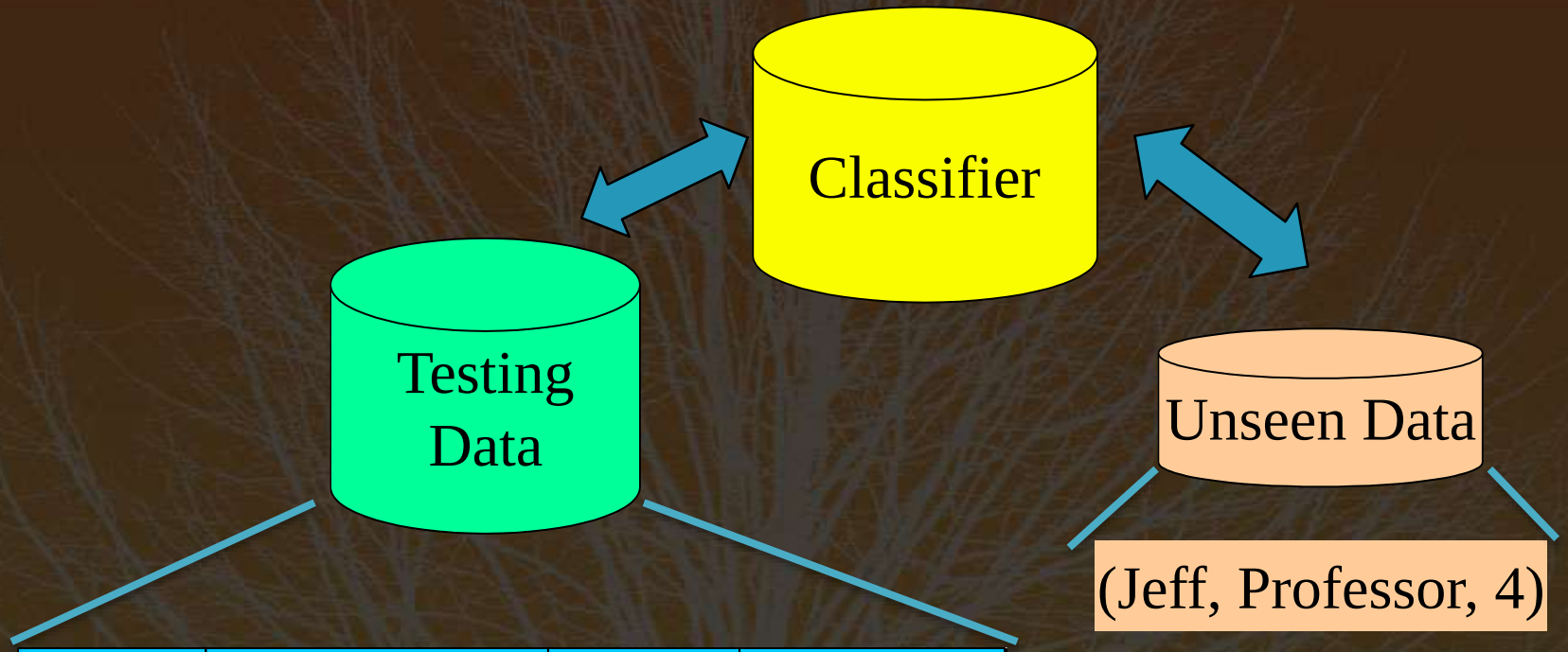
Classification Algorithms

Classifier (Model)

NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

IF rank = ‘professor’
OR years > 6
THEN tenured = ‘yes’

فرایند (۲): استفاده از مدل در پیش‌بینی



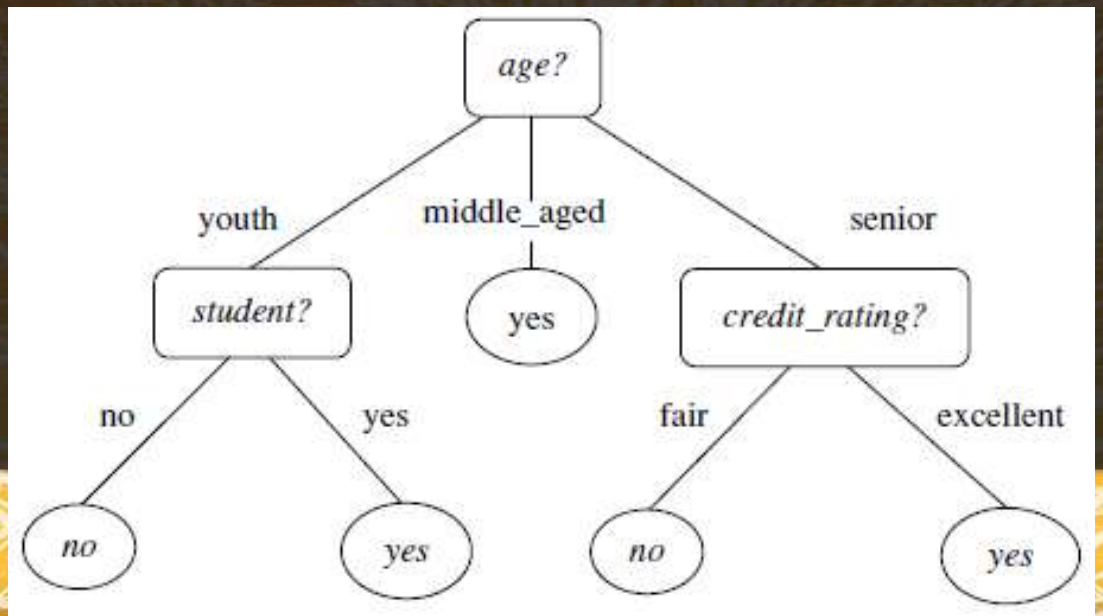
NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes

دسته‌بندی : مفاهیم اولیه

- مفاهیم اولیه
- درخت تصمیم (Decision Tree)
- روش‌های دسته‌بندی بیز (Bayes)
- روش‌های دسته‌بندی مبتنی بر قانون (Rule-Based)
- ارزیابی مدل‌های و انتخاب مدل مناسب
- روش‌های بهبود دقت دسته‌بندی
- خلاصه فصل

درخت تصمیم

- درخت تصمیم یک درخت با ساختار فلوچارت مانند است
- هر گره غیرپایانی، یک تست بر روی ویژگی انجام میدهد
- هر شاخه یا یال یک نتیجه از تست است
- هر گره برگ یک برچسب کلاس است
- مثال:



درخت تصمیم چگونه استفاده می‌شود؟

- نحوه‌ی استفاده از درخت تصمیم به صورت زیر است
- برای یک داده جدید که مقادیر ویژگی آن معلوم است ما برچسبی ندارد
- بر اساس درخت یک مسیر از ریشه به برگ طی می‌شود
- برچسب برگ نشان دهنده کلاس پیش‌بینی شده برای آن موجودیت است
- با این تفاسیر درخت تصمیم را به راحتی می‌توان به قوانین دسته‌بندی تبدیل نمود

چرا درخت تصمیم محبوب است؟

۱) ایجاد درخت تصمیم به هیچ دانش تخصصی یا تنظیم پارامتری احتیاج ندارد، به همین دلیل برای کشف دانش مناسب است.

۲) درخت‌های تصمیم‌گیری می‌تواند داده‌های با ابعادی زیاد را پوشش دهند

۳) نمایش دانش بدست آمده از ساختار درختی شهودی و به طور کلی آسان است و توسط انسان قابل‌درک است.

۴) آموزش و دسته‌بندی با استفاده از درخت تصمیم‌گیری ساده و سریع می‌باشد.

۵) به طور کلی، دسته‌بندی با درخت تصمیم‌گیری از دقت خوبی برخوردار است.

۶) الگوریتم درخت تصمیم‌گیری برای دسته‌بندی در بسیاری از زمینه‌های کاربردی قابل استفاده است.

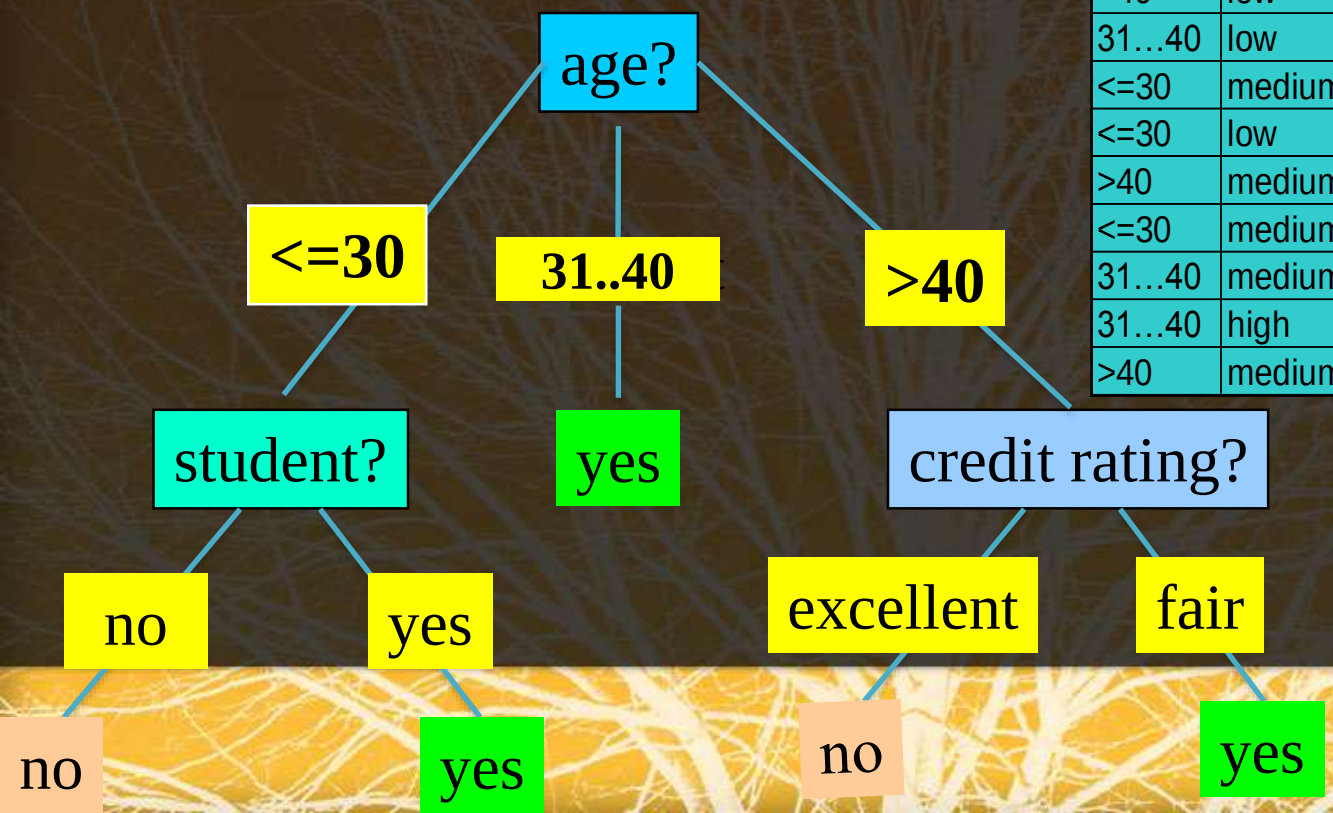
مشکلات ایجاد الگوریتم برای ساخت درخت

- سوال ۱ : کدام ویژگی‌ها در هر سطح از درخت استفاده شوند؟
- سوال ۲ : زمانی که درخت ساخته شد، برخی شاخه‌ها حاوی نویز و مقادیر پرت هستند
- سوال ۳ : چگونه این شاخه‌ها را تشخیص دهیم و هرس کنیم (tree pruning)

درخت تصمیم: یک مثال

- مجموعه داده آموزشی: Buys_computer
- مجموعه داده‌های زیر یک مثال از Quinlan's ID3 است.

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no



توضیحات مقدماتی

- الگوریتمهای مختلفی برای ساخت درخت تصمیم وجود دارد
- ID3، C4.5 و CART(Classification and regression tree)
- در این الگوریتم‌ها یک روش حریصانه (بدون بازگشت به عقب) استفاده می‌شود
- همچنین راهکار به صورت بالا به پایین وتقسیم و حل است

الگوریتم درخت تصمیم

- الگوریتم پایه (یک الگوریتم حریصانه)
  - درخت به صورت بازگشتی با روش تقسیم و حل، از بالا به پایین ساخته می‌شود.
  - در آغاز، همه نمونه‌های آموزشی در ریشه هستند.
  - ویژگی‌ها چند مقداری هستند (اگر مقادیر پیوسته باشد، گسسته می‌شوند)
  - داده‌ها به صورت بازگشتی و بر اساس ویژگی‌های انتخاب شده پارتیشن‌بندی می‌شوند.
  - ویژگی‌های آزمون بر اساس معیارهای اکتشافی و یا آماری انتخاب می‌شوند
  - (به عنوان مثال، information gain)

الگوریتم درخت تصمیم

- **شرایط برای توقف پارتیشن‌بندی**
  - همه نمونه‌ها برای یک گره داده شده متعلق به یک کلاس باشند.
  - هیچ ویژگی‌ای برای پارتیشن‌بندی بیشتر، باقی نمانده باشد- در این شرایط برچسب اکثریت نمونه‌های برگ برای دسته‌بندی به کار گرفته می‌شود.
  - هیچ نمونه بیشتری وجود نداشته باشد.

استنتاج بالا به پایین درخت‌های تصمیم ID3

۱) فرض کنید بهترین ویژگی تصمیم برای گره بعد A باشد

۲) A را به عنوان ویژگی تصمیم برای گره قرار بده

۳) برای هر مقدار از A یک فرزند جدید ایجاد کن

۴) مرتب‌سازی نمونه‌های آموزشی برای گره برگ با توجه به مقدار ویژگی شاخه

۵) اگر همه نمونه‌های آموزشی کاملا طبقه بندی شده باشند (همان مقدار ویژگی هدف)، متوقف شو، در غیر این صورت برای گره‌های جدید برگ تکرار کن

معیار انتخاب ویژگی: information gain

- این ویژگی در ID3 و C4.5 به کار رفته است
- ویژگی با بالاترین information gain انتخاب می‌شود.
- در صورتی که P_i احتمال این که تاپل دلخواه در D متعلق به کلاس C_i باشد
- توسط $\frac{|C_{i,D}|}{|D|}$ تخمین زده می‌شود.
- میزان اطلاعات مورد انتظار (آنتروپی) برای دسته‌بندی یک تاپل در D که مورد نیاز است:

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

معیار انتخاب ویژگی: information gain

- اطلاعات مورد نیاز (پس از استفاده از ویژگی A برای تقسیم D به V پارتیشن) برای دسته‌بندی D:

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

- Information gain بدست آمده بوسیله انشعاب روی ویژگی A

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

مثال

- Class P: buys_computer = “yes”
- Class N: buys_computer = “no”

$$Info(D) = I(9,5) = -\frac{9}{14}\log_2(\frac{9}{14}) - \frac{5}{14}\log_2(\frac{5}{14}) = 0.940$$

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

$$\begin{aligned} Info_{age}(D) = & \frac{5}{14} \times (-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}) \\ & + \frac{4}{14} \times (-\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4}) \\ & + \frac{5}{14} \times (-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}) \\ = & \ 0.694 \text{ bits.} \end{aligned}$$

$$Gain(age) = Info(D) - Info_{age}(D) = 0.246$$

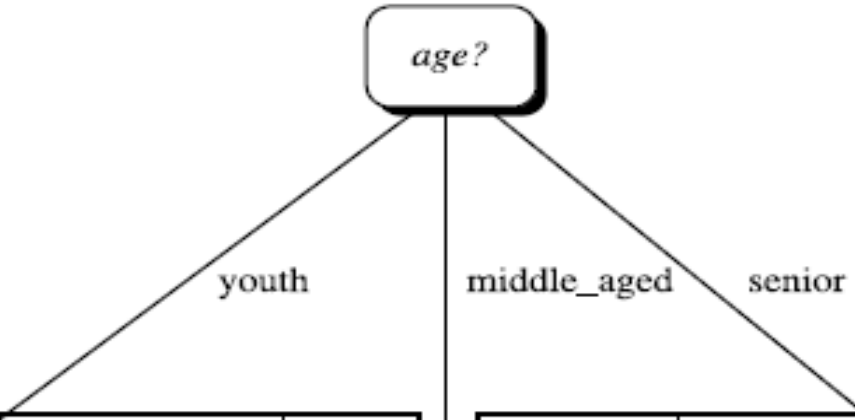
ادامه مثال

- این اعمال را برای تمام ویژگی‌ها انجام می‌دهیم

$$Gain(income) = 0.029$$
$$Gain(student) = 0.151$$
$$Gain(credit\_rating) = 0.048$$

- بالاترین information gain را ویژگی age بدست می‌آورد
- بنابراین برای تصمیم‌گیری در این سطح از ویژگی age استفاده می‌شود

خروجی: درخت تصمیم‌گیری

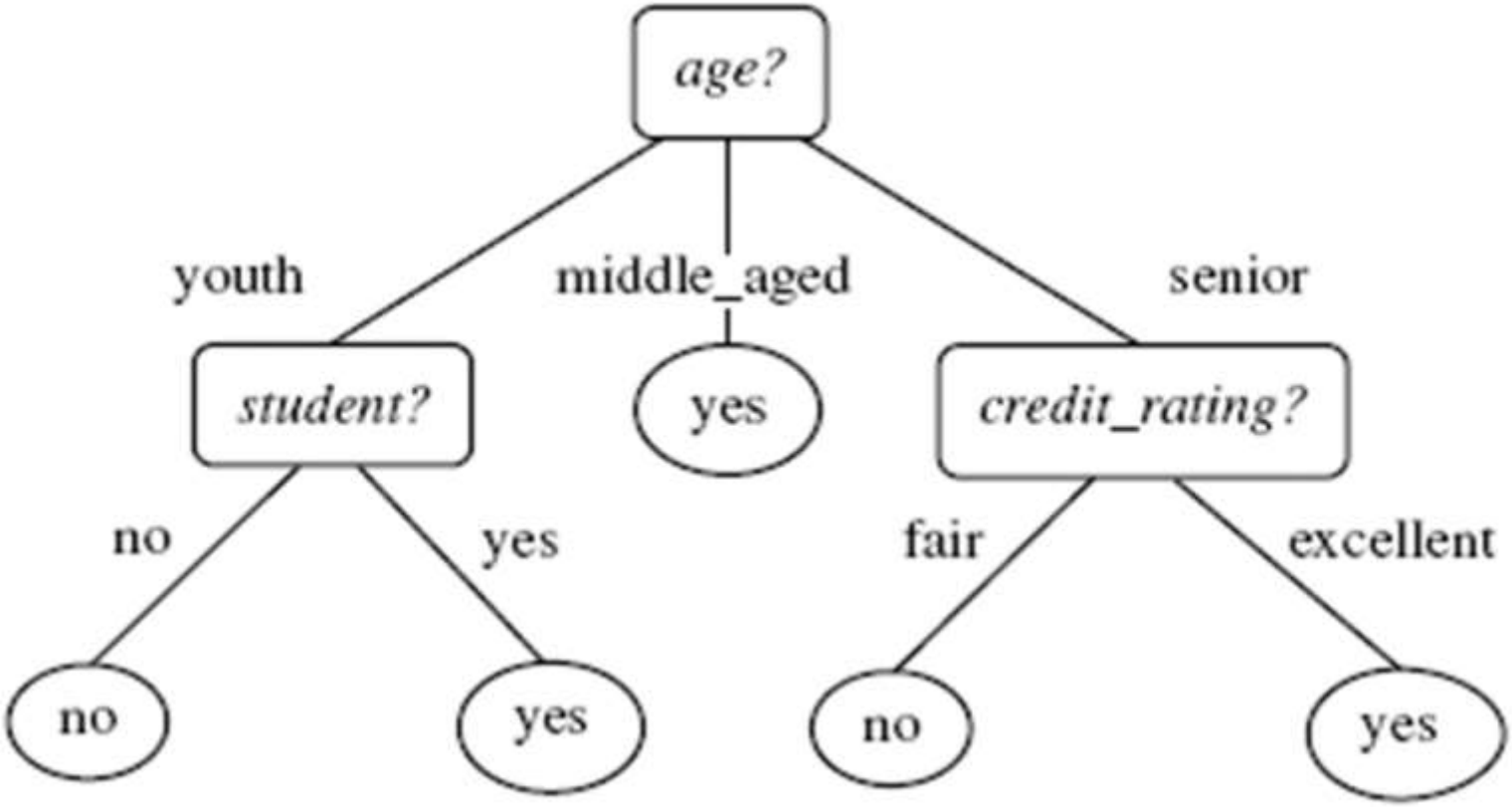


income	*student*	*credit_rating*	*class*
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

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

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

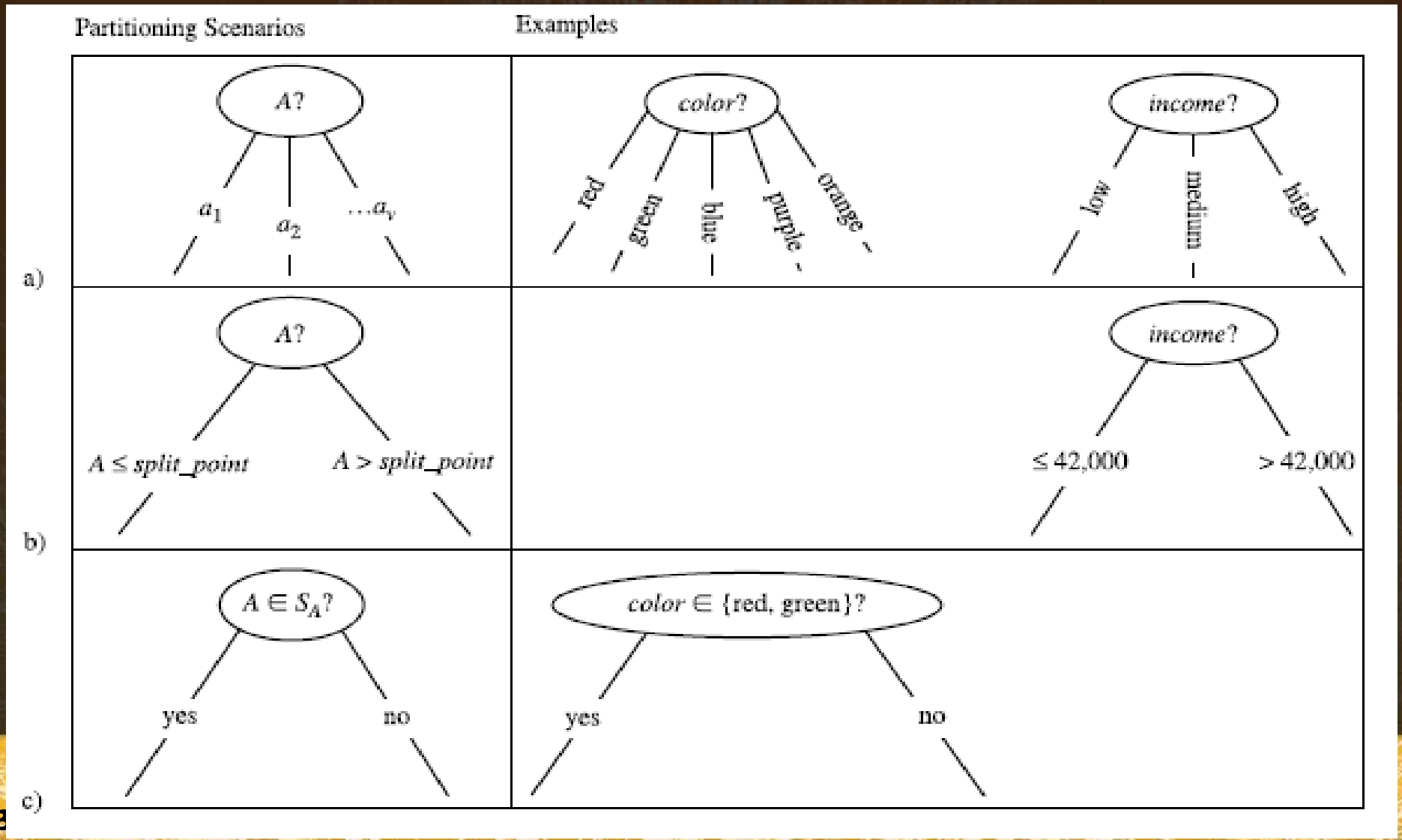
ادامه مراحل الگوریتم



محاسبه information gain برای مقدار پیوسته

- فرض کنید ویژگی A یک ویژگی با مقدار پیوسته باشد
- باید بهترین نقطه تقسیم (split point) را برای A تعیین کنیم.
  - مقدار A را به صورت صعودی مرتب می‌کنیم.
  - به طور معمول، نقطه وسط بین هر جفت از مقادیر مجاور به عنوان نقطه تقسیم در نظر گرفته می‌شود.
    - $(a_i+a_{i+1})/2$ نقطه وسط بین مقدار a_i و a_{i+1} است.
  - نقطه‌ای با کمترین expected information requirement را به عنوان نقطه تقسیم انتخاب می‌کنیم
- تقسیم:
  - D_1 مجموعه‌ای از تاپل در D است که مقدار A کمتر مساوی نقطه تقسیم
  - و D_2 مجموعه‌ای از تاپل در D است که مقدار A بیشتر از نقطه تقسیم است

محاسبه information-Gain برای ویژگی‌های پیوسته



ویژگی‌های با مقادیر پیوسته

- ایجاد یک ویژگی گسسته برای آزمایش داده‌های پیوسته
  - دما = ۲۴/۵ درجه سانتی‌گراد
  - دما بیشتر از ۲۰ درجه سانتی‌گراد = {true,false}

آستانه کجا تنظیم شود؟

دما	15⁰C	18⁰C	19⁰C	22⁰C	24⁰C	27⁰C
بازی تنیس	NO	NO	YES	YES	YES	NO

نقطه‌ای با حداقل information requirement مورد انتظار به عنوان نقطه تقسیم (Split-point) تعیین می‌شود

معیار (Gain Ratio)برای انتخاب ویژگی(C4.5)

- **معیار information gain نسبت به** ویژگی‌های با تعداد زیاد مقادیر بایاس است (مانند ویژگی ID)

- **C4.5 (جانشین ID3) از معیار gain ratio برای غلبه بر مشکل استفاده می‌کند(نرمال‌سازی بهره اطلاعات)**

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

GainRatio(A) = Gain(A)/SplitInfo(A)

معیار (Gain Ratio) برای انتخاب ویژگی (C4.5)

- مثال:

$$\text{Split}Info(D) = -\frac{4}{14}log_2\left(\frac{4}{14}\right) - \frac{6}{14}log_2\left(\frac{6}{14}\right) - \frac{4}{14}log_2\left(\frac{4}{14}\right) = 1.557$$

$$\text{gain\_ratio(income)} = 0.029/1.557 = 0.019$$

- ویژگی با حداکثر gain ratio به عنوان ویژگی تقسیم انتخاب شده است.

شاخص Gini (CART)

- این شاخص در درخت CART استفاده شده است
- شاخص gini تقسیم دودویی را برای هر ویژگی انجام می‌دهد
- ابتدا موردی را در نظر بگیرید که A یک ویژگی با مقدار گسسته است و دارای v مقدار مجزا $\{a_1,a_2,\ldots,a_v\}$ در D است
- برای تعیین بهترین تقسیم دوتایی در A، ما تمام زیر مجموعه‌های ممکن که می‌توان با استفاده از مقادیر شناخته شده از A تشکیل شود را بررسی می‌کنیم.
- هر زیر مجموعه مانند SA، می تواند به عنوان یک تست دودویی برای ویژگی‌ها A، در نظر گرفته شود و به صورت روبه رو خوانده شود «Aدر SA است؟»
- برای یک تاپل، اگر مقدار A در مقادیر ذکر شده در SA باشد، آنگاه جواب مثبت است.
- اگر A دارای v مقدار ممکن باشد ، پس 2^v مقدار ممکن زیر مجموعه وجود دارد.

شاخص Gini (CART)

- برای مثال، اگر درآمد سه مقدار ممکن داشته باشد، به صورت:

{low, medium, high}، آنگاه زیر مجموعه‌های زیر را در نظر میگیریم:

{*low, medium, high*}
{*low,medium*}
{*low, high*}
{*medium, high*}
{*low*}
{*medium*}
{*high*}
{}

حالت تهی و مجموعه کامل را در نظر نمیگیریم بنابراین $2^v - 2$ زیر مجموعه داریم

محاسبه شاخص Gini

- برای پیدا کردن معیار تقسیم برای تاپل در D، ما نیاز به محاسبه شاخص Gini برای هر ویژگی داریم.
- بیایید با ویژگی درآمد و در نظر گرفتن هر یک از زیر مجموعه تقسیم ممکن شروع کنیم.

زیر مجموعه {*low, medium*} را در نظر بگیرید. این امر منجر به ایجاد ۱۰ تاپل می‌شود و شرایط مورد قبول پارتیشن D_1 درآمد{*low, medium*} در نظر گرفته می‌شود.

۴ تاپل باقیمانده از D به پارتیشن D_2 اختصاص داده می شود. مقدار شاخص جینی بر این اساس محاسبه می‌شود:

شاخص Gini (CART)

- اگر یک مجموعه داده D شامل نمونه‌هایی از N کلاس باشد، شاخص gini(D) به صورت زیر تعریف می‌شود:

$$gini(D) = 1 - \sum_{j=1}^{n} {p_j}^2$$

که pj فراوانی نسبی کلاس j در D است

- اگر مجموعه داده D بر روی A به دو زیر مجموعه D_1 و D_2 تقسیم شده باشد شاخص gini(D) به صورت تعریف می‌شود:

$$gini_A(D) = \frac{|D_1|}{|D|} gini(D_1) + \frac{|D_2|}{|D|} \mathrm{gini}(D_2)$$

شاخص Gini (CART)

- کاهش ناخالصی (reduction in impurity):

$$\Delta gini(A) = gini(D) - gini_A(D)$$

- ویژگی که کوچکترین $gini_{split}$ (D) (یا بزرگترین کاهش ناخالصی)را فراهم می‌کند به عنوان گره تقسیم انتخاب می‌شود.(باید تمام نقاط تقسیم ممکن برای هر یک از ویژگی‌ها محاسبه شوند.)

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

$$gini(D) = 1 - \Big($$

$$gini_{income \in \{low,n} \quad \frac{4}{14}\Big) \operatorname{Gini}(D_1)$$

$$= \frac{10}{14}\left(1 - \left(\frac{7}{10}\right)^2 - \left(\frac{3}{10}\right)^2\right) + \frac{4}{14}\left(1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2\right) = 0.443$$

$$= gini_{income \in \{high\}}(D)$$

محاسبه شاخص Gini

- به طور مشابه، شاخص Gini برای سایر زیر مجموعه باقی‌مانده بصورت زیر است
- برای زیرمجموعه {*low, high*} and {*medium*} داریم ۰/۴۵۸
- و برای زیر مجموعه {*medium, high*} and {*low*} مقدار ۰/۴۵۰
- بنابراین، بهترین تقسیم دودویی برای ویژگی درآمد به این صورت است:

({*low, medium*} (or {*high*}) زیرا شاخص Gini آن کمترین مقدار را دارید

محاسبه شاخص Gini

- برای بررسی سن، داریم:
- ({*youth, senior*} (or {*middle aged*} بهترین تقسیم سن با شاخص ۰/۳۷۵ است
- ویژگی‌های student و credit rating هر دو باینری هستند و مقادیر شاخص ۰/۳۶۷ و ۰/۴۲۹ را دارند

محاسبه شاخص Gini

- از روی نتایج مشخص است که ویژگی income بهترین ویژگی برای تقسیم است و

  میزان کاهش ناخالصی به صورت: 0.459-0.357 =0.102

مقایسه معیارهای انتخاب ویژگی

- به طور کلی سه معیار زیر معرفی شد که هر کدام در مواردی اشکال دارند

Information Gain

- به سمت ویژگی‌های چند مقداره (multivalued) گرایش دارد.

Gain ratio

- به انشعابات نا متوازن تمایل دارد که در آن یک پارتیشن بسیار کوچکتر از دیگران است.

Gini Index

- گرایش به ویژگی‌های چند مقداره
- وقتی تعداد کلاس‌ها زیاد باشد محاسبات زمان‌گیر است

با این وجود درخت تصمیم نتایج خوبی دارد.

سایر معیارهای مربوط به گزینش ویژگی‌ها

- CHAID: یک الگوریتم درخت تصمیم که از معیار آماری تست کای-دو استفاده می‌کند
- C-SEP: در مقایسه با info. Gain و شاخص gini در برخی موارد عملکرد بهتری دارد
- سایر روش‌ها
- G-statistic
- Minimum description length (MDL)
- CART: برای ساخت درخت در هر گره غیرپایانی به جای یک ویژگی، چندویژگی را در مورد تست قرار می‌دهد.

هرس درخت تصمیم

در درخت تصمیم مشکلی با نام overfitting مطرح است

۱) ممکن است در ایجاد درخت تعداد زیادی شاخه به وجود آید

دلیل آن وجود آنومالی در داده‌ها است

آنومالی به دلیل وجود نویز و داده‌های پرت به وجود می‌آید

۲) از طرفی ممکن است درخت ایجاد شده برای داده‌های جدید ضعیف عمل کند

- **راه‌حل:**
- هرس نمودن شاخه‌های زائد است
- اینکار باعث کوچک شدن، ساده شدن و به الطبع فهم آسان درخت خواهد شد
- از جهتی برای داده‌های تست عملکرد بهتری خواهد داشت

هرس درخت تصمیم

- برای این منظور دو روش کلی وجود دارد
  - Prepruning –
  - Postpruning –

هرس درخت تصمیم

Prepruning

- این روش در زمان ساخت درخت اعمال می‌شود
- در واقع در صورتی که تشخیص داده شود که تقسیم بیشتر یک شاخه بهبودی در دقت نخواهد داشت
- آن شاخه هرس شده و دیگر تقسیم انجام نمی‌گیرد
- کلاس آن شاخه بر اساس بیشترین تعداد اعضای آن محاسبه میشود
- مثلا در صورتی که بیشتر داده‌های باقی مانده در آن شاخه yes باشند، شاخه را دیگر تقسیم نکرده و یک برگ با مقدار برچسب yes قرار می‌دهیم
- تقسیم شدن یا نشدن بر اساس تعیین یک آستانه مناسب است
- تعیین این آستانه ساده نیست

هرس درخت تصمیم

- Postpruning
- روش متداول‌تری است
- زمانی که درخت ایجاد شد، اعمال می‌شود
- در صورتی که زیر درخت یک گره، هرس شود. آن گره تبدیل به برگ می‌شود
- کلاس این برگ بیشترین درصد کلاس‌های آن زیر درخت خواهد شد
- این روش از روش قبلی هزینه‌برتر است

نمونه PostPruning

- به طور مثال در درخت CART از معیار cost complexity برای پس‌هرس استفاده می‌شود
- این معیار توسط دو عامل تعیین می‌شود
  - تعداد برگ‌های درخت (number of leaves)
  - نسبت خطای درخت (error rate)
- هرس از پایین درخت شروع می‌شود و بر اساس این معیار هرس شدن یا نشدن یک شاخه تعیین می‌شود
- برای تعیین خطای درخت از مجموعه pruning set برای تعیین درصد خطا استفاده میشود

نمونه Postpruning

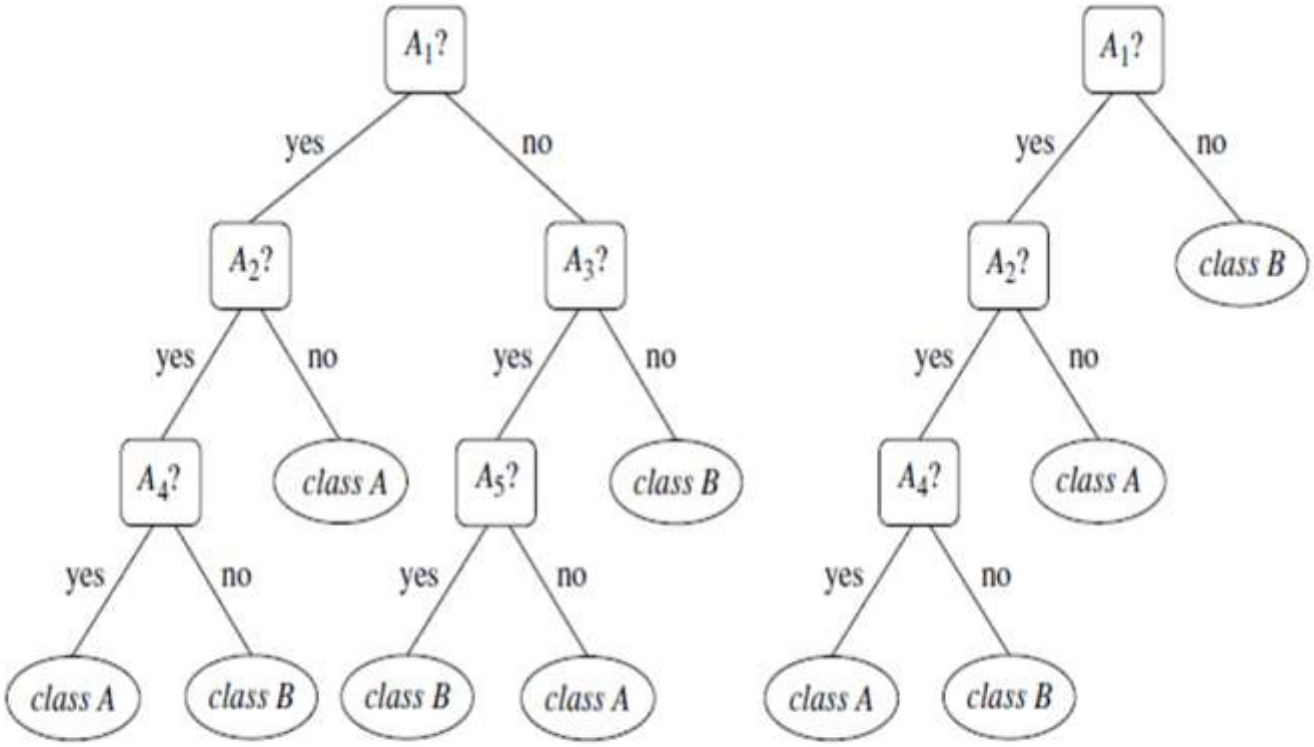


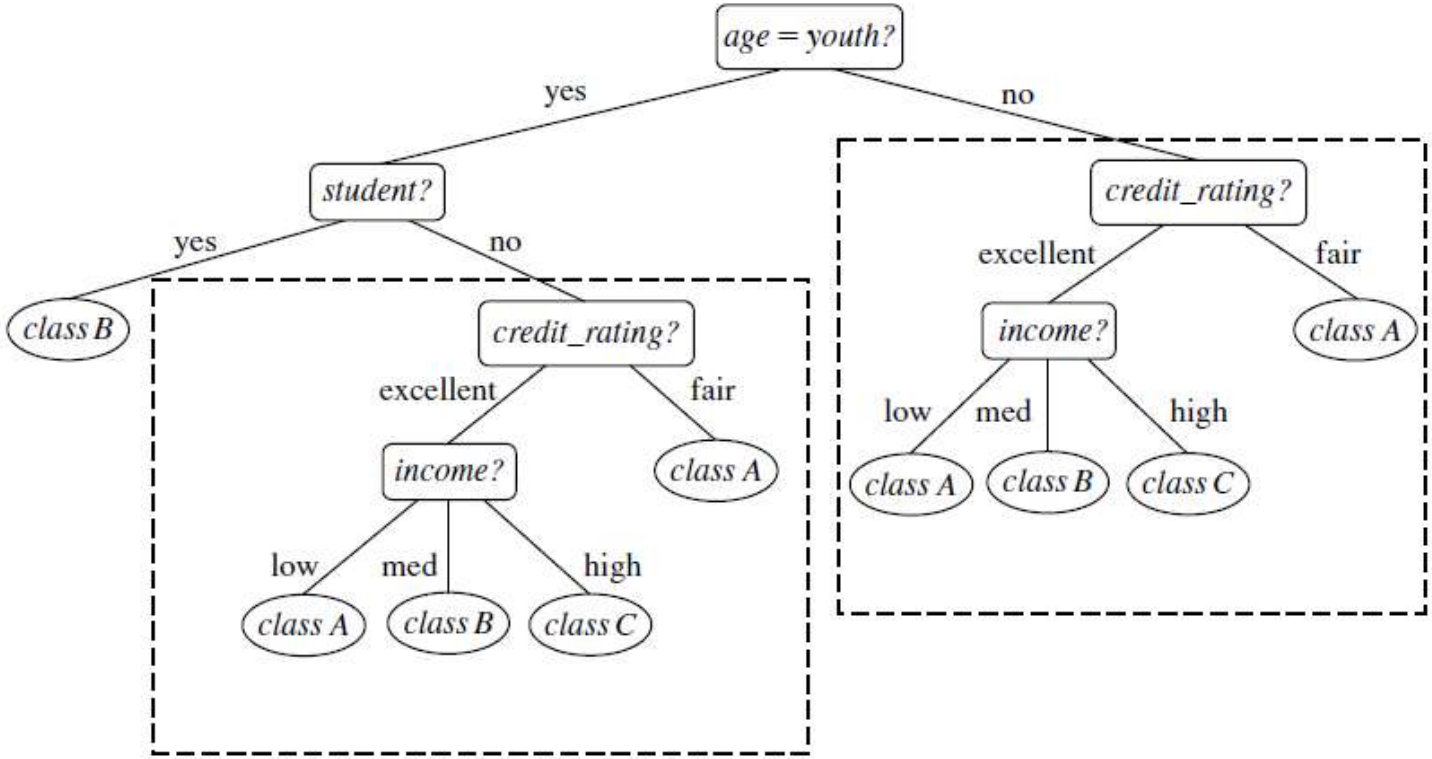
Figure 8.6 An unpruned decision tree and a pruned version of it.

سایر مشکلات درخت تصمیم

- به غیر از هرس نمودن، یک درخت ممکن است شامل مشکلات زیر نیز باشد

۱. **Replication**

– در این وضعیت یک زیردرخت در جای دیگری از درخت نیز تکرار شده است

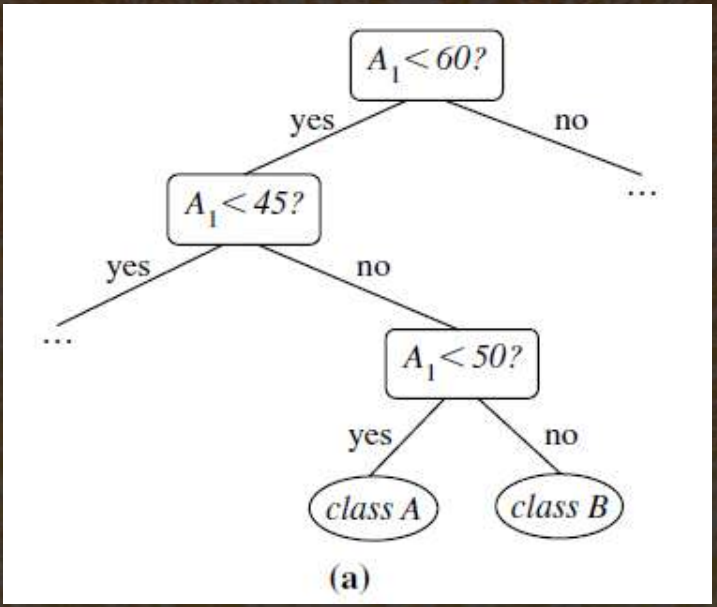


(b)

سایر مشکلات درخت تصمیم

۲. **repetition**

– در این حالت یک ویژگی در چند عمق تکرار می‌شود



یک راه‌حل برای این مشکلات استفاده از **multivariate split** است

یعنی تقسیم بر اساس ترکیبی از ویژگی‌ها انجام شود

مقیاس‌پذیری (Scalibility) درخت تصمیم

- روش‌های معرفی شده (ID3, C4.5, CART) برای مجموعه داده‌ها کوچک‌اند
- عیب این روش‌ها، این است که باید مجموعه داده آموزش در حافظه اصلی جا شود

- برای مقیاس‌پذیری درخت تصمیم برای مجموعه داده‌های بزرگ، روش‌هایی زیر وجود دارد

۱. Rainforest

- این روش از لیست‌هایی با نام AVC-List (Attribute-value,ClassLabel) استفاده می‌کند

Rainforest (AVC-List)

Training Examples

age	income	student	credit_rating	comp
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

AVC-set on *Age*

Age	Buy_Computer	
	yes	no
<=30	2	3
31..40	4	0
>40	3	2

AVC-set on *income*

income	Buy_Computer	
	yes	no
high	2	2
medium	4	2
low	3	1

AVC-set on *Student*

student	Buy_Computer	
	yes	no
yes	6	1
no	3	4

AVC-set on *credit_rating*

Credit rating	Buy_Computer	
	yes	no
fair	6	2
excellent	3	3

مقیاس‌پذیری (Scalibility) درخت تصمیم

۲) BOAT

- از یک تکنیک آماری به نام bootstrapping استفاده کرده و مجموعه داده را به نمونه‌های متعدد و کوچک می‌شکنند
- از هر کدام درخت ایجاد می‌کند
- درنهایت درخت‌ها با یکدیگر ترکیب می‌شوند

- مزیت‌ها
  - مجموعه داده تنها دوبار اسکن می‌شود
  - دو تا سه برابر سریعتر از RainForest است
  - می‌تواند با ورود داده‌های جدید، درخت ساخته شده را اصلاح نماید (بدون ساخت مجدد آن)

دسته‌بندی : مفاهیم اولیه

- مفاهیم اولیه
- درخت تصمیم (Decision Tree)
- روش‌های دسته‌بندی بیز (Bayes)
- روش‌های دسته‌بندی مبتنی بر قانون (Rule-Based)
- ارزیابی مدل‌های و انتخاب مدل مناسب
- روش‌های بهبود دقت دسته‌بندی
- خلاصه فصل

دسته‌بندی بیزین

- A statistical classifier: یک روش احتمالی است که در آن احتمال تعلق یک تاپل به یک کلاس سنجیده می‌شود
- Foundation: بر اساس قضیه بیز بنا نهاده شده است
- Performance: دسته‌بندی بیزی ساده، عملکردی قابل مقایسه با درخت تصمیم و برخی شبکه‌های عصبی دارد.
- از لحاظ سرعت نیز برای مجموعه‌های بزرگ مناسب است.
- ساده‌ترین کلاسیفایر بیزین با نام naïve bayesian classifier شناخته می‌شود
- در این کلاسیفایر فرض می‌شود که میزان تاثیر مقدار یک ویژگی، از مقادیر سایر ویژگی‌ها مستقل است.
- این فرض با نام class-conditional independence شناخته می‌شود
- این فرض باعث می‌شود محاسبات کم شده و نام این روش نیز از این خصوصیت گرفته شده

تئوری Bayes

- فرض کنید X یک مشاهده (evidence) باشد.
- در مجموعه داده‌ی مورد بررسی X با تعدادی ویژگی نشان داده می‌شود
- در نظر بگیرید H یک فرضیه است که تعلق X را به یک کلاس خاص مانند C نشان می‌دهد
- برای اعمال دسته‌بندی نیاز داریم P(H|X) را بسنجیم
- یعنی احتمال اینکه اگر X را داشته باشیم، فرضیه درست است
- در واقع می‌خواهیم احتمال تعلق تاپل X به کلاس C را در صورت دانستن ویژگی‌های X بدانیم

تئوری Bayes

- فرمول بیز :

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

- به P(H|X) احتمال پسین یا posterior probability گفته می‌شود
- در مقابل P(X) یا P(H) را از پیش می‌دانیم و احتمالات پیشین یا prior probability مینامیم
- P(X|H) (likelyhood) احتمال مشاهده نمونه X به شرط در نظر گرفتن فرضیه H
  - به عنوان مثال: در صورتی که بدانیم X کامپیوتر خواهد خرید، احتمال اینکه X دارای درآمد متوسط باشد چقدر است؟

مثالی از تئوری بیز

- دکتر می داند که مننژیت در ۵۰٪ مواقع باعث خشکی گردن می‌شود.

P(X | H) (likelyhood)

- احتمال اینکه یک بیمار مننژیت داشته باشد برابر ۱/۵۰۰۰۰ است.

P(H) (prior probability)

- احتمال اینکه یک بیمار خشکی گردن داشته باشد ۱/۲۰ است

P(X) (evidece)

سوال : اگر بیمار دارای خشکی گردن باشد، احتمال اینکه او مننژیت داشته باشد چقدر است؟

مثالی از تئوری بیز

- اگر بیمار دارای خشکی گردن باشد، احتمال اینکه او مننژیت داشته باشد چقدر است؟

P(H|X) (posteriori probability)

$$P(H \mid X) = \frac{P(X \mid H)P(H)}{P(X)} = \frac{0.5 \times 1/50000}{1/20} = 0.0002$$

- **posteriori = likelihood x prior/evidence**

Naïve Bayesian classification

- ویژگی و برچسب کلاس را به عنوان متغیرهای تصادفی در نظر بگیرید.
- یک داده با ویژگی‌های مختلف به صورت $(A_1, A_2,\ldots,A_n)$ نشان داده می‌شود
  - هدف این است که کلاس C را پیش‌بینی کنیم
  - به طور خاص، ما می‌خواهیم حداکثر مقدار C را پیدا کنیم.

$$P(C| A_1, A_2,\ldots,A_n)$$

- آیا می‌توانیم $P(C| A_1, A_2,\ldots,A_n)$ **را مستقیما با داده تخمین بزنیم؟**

Naïve Bayesian classification

محاسبه احتمال قبلی $P(C|\ A_1,\ A_2,\ldots,A_n)$ برای همه مقادیر C با استفاده از تئوری بیز

$$P(C \mid A_1 A_2 \ldots A_n) = \frac{P(A_1 A_2 \ldots A_n \mid C) P(C)}{P(A_1 A_2 \ldots A_n)}$$

- مقدار C را انتخاب میکنیم که $P(C|\ A1,\ A2,\ldots,An)$ ماکزیمم شود
- در واقع عبارت زیر باید ماکزیمم شود

$$P(A_1,\ A_2,\ \ldots,\ A_n|C)\ P(C)$$

- چگونه می‌توان $P(A_1,\ A_2,\ \ldots,\ A_n|C)\ P(C)$ را تخمین زد؟

Naïve Bayesian classification

- با فرض مستقل بودن ویژگی‌های A_i در زمانی که کلاس داده شده است:

$$P(A_1, A_2, \ldots, A_n \mid C_j) = P(A_1 \mid C_j)\, P(A_2 \mid C_j) \ldots P(A_n \mid C_j)$$

- می‌توان $P(A_i \mid C_j)$ برای همه A_i ها و C_j ها تخمین زده شود.
- اگر $P(C_j)\, \Pi\, P(A_i \mid C_j)$ ماکزیمال باشد. داده جدید در کلاس C_j دسته‌بندی می‌شود.

مثال : دسته‌بندی بیزین ساده

Class:

C1:buys_computer = ‘yes’

C2:buys_computer = ‘no’

Data to be classified:

X = (age <=30,

Income = medium,

Student = yes

Credit_rating = Fair)

age	income	student	credit_rating	comp
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

ادامه مثال

$P(C_i)$: P(buys_computer = "yes") = 9/14 = 0.643
P(buys_computer = "no") = 5/14= 0.357

Compute $P(X|C_i)$ for each class

P(age = "<=30" | buys_computer = "yes") = 2/9 = 0.222
P(age = "<= 30" | buys_computer = "no") = 3/5 = 0.6
P(income = "medium" | buys_computer = "yes") = 4/9 = 0.
P(income = "medium" | buys_computer = "no") = 2/5 = 0.4
P(student = "yes" | buys_computer = "yes) = 6/9 = 0.667
P(student = "yes" | buys_computer = "no") = 1/5 = 0.2
P(credit_rating = "fair" | buys_computer = "yes") = 6/9 =
P(credit_rating = "fair" | buys_computer = "no") = 2/5 =

age	income	student	credit_rating	comp
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

ادامه مثال

X = (age <= 30 , income = medium, student = yes, credit_rating = fair)

$P(X|C_i)$: P(X|buys_computer = "yes") = 0.222 x 0.444 x 0.667 x 0.667 = 0.044

P(X|buys_computer = "no") = 0.6 x 0.4 x 0.2 x 0.4 = 0.019

$P(X|C_i)*P(Ci)$: P(X|buys_computer = "yes") * P(buys_computer = "yes") = 0.028

P(X|buys_computer = "no") * P(buys_computer = "no") = 0.007

نتیجه : بنابراین X به کلاس **("buys_computer = yes")** تعلق دارد.

Naïve Bayesian classification

- اگر یکی از احتمالات شرطی صفر باشد، آنگاه کل عبارت صفر می‌شود.
- در این حالت احتمال تعلق صفر شده و سایر ویژگی‌ها نادیده گرفته می‌شود

- برای حل مشکل از تصحیح لاپلاس استفاده می‌کنیم (Laplacian correction)

- در این روش برای صفر نشدن احتمال یک تاپل به صورت تصنعی وارد می‌شود
- چون مجموعه داده بسیار بزرگ است اضافه نمودن یک تاپل اثری نخواهد داشت
- در این حالت احتمال به صورت زیر محاسبه می‌شود

$$\text{Original}: P(A_i \mid C) = \frac{N_{ic}}{N_c}$$

$$\text{Laplace}: P(A_i \mid C) = \frac{N_{ic} + 1}{N_c + c}$$

Naïve Bayesian classification

- مثال: یک مجموعه داده با ۱۰۰۰ رکورد را فرض کنید که در آن
- درآمد= پایین(۰)
- درآمد= متوسط(۹۹۰) و
- درآمد= بالا(۱۰)
- استفاده از **تصحیح لاپلاس** : اضافه کردن مقدار ۱ به هر حالت

Prob(income = low) = 1/1003
Prob(income = medium) = 991/1003
Prob(income = high) = 11/1003

توضیحات اضافی

- **مزایا**
- پیاده‌سازی آسان روش
- **نسبت به نقاط نویز مقاوم است**
- **برخورد با داده‌های miss با نادیده گرفتن آن نمونه‌ها در محاسبات احتمال**
- در اکثر موارد نتایج مناسبی در پی دارد

- **معایب**
- فرض مستقل بودن باعث کاهش دقت خواهد شد
- در عمل میان متغیرها وابستگی وجود دارد.
- وابستگی میان این ویژگی‌ها را به وسیله naïve bayes نمی‌توان مدل کرد.

- **چگونه با این وابستگی‌ها برخورد کنیم؟** Bayesian Belief Networks

دسته‌بندی : مفاهیم اولیه

- مفاهیم اولیه
- درخت تصمیم (Decision Tree)
- روش‌های دسته‌بندی بیز (Bayes)
- روش‌های دسته‌بندی مبتنی بر قانون (Rule-Based)
- ارزیابی مدل‌های و انتخاب مدل مناسب
- روش‌های بهبود دقت دسته‌بندی
- خلاصه فصل

دسته‌بندی مبتنی بر قانون

- این روش در دو بخش توضیح داده می‌شود
- در ابتدا روش استفاده از قوانین برای دسته‌بندی تشریح می‌شود
- در بخش دوم روش‌های تولید قوانین بررسی می‌شوند

استفاده از قوانین IF-THEN برای دسته‌بندی

- نمایش دانش به صورت قوانین IF-THEN

$$R:\ IF\ age = youth\ AND\ student = yes\ THEN\ buys\_computer = yes$$

$$R:\ (age = youth)\ \ (student = yes) \Rightarrow (buyes - computer = yes)$$

- به بخش اول پیش‌شرط یا مقدم قانون گفته می‌شود
- (Rule antecedent or precondition)
- به بخش دوم نتیجه یا تالی گفته می‌شود (Rule consequent)
- زمانی که برای یک تاپل مقدم‌ها برآورده شوند در اصطلاح گفته می‌شود که پیش‌شرط‌ها ارضا شده‌اند (The rule is satisfied)
- و قانون تاپل را پوشش داده است (the rule covers the tuple)

ارزیابی قوانین دسته‌بندی

- **برای ارزیابی از دو معیار پوشش و دقت استفاده می‌شود**
- ncovers= تعداد رکوردهایی که به وسیله R پوشش داده شده‌اند.
- ncorrect= از رکورهای پوشش داده شده چه تعداد رکورد به درستی دسته‌بندی شده‌اند.
- D= مجموعه داده آموزش است

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

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

چگونه با استفاده از قوانین دسته‌بندی کنیم؟

- اگر یک قانون برای تاپل X صدق کند، می گوییم که قانون ارضا شده است. به عنوان مثال، فرض کنید که داریم

X={age=youth, income=medium, student=yes, credit_rating=fair}

- می‌خواهیم بر اساس قوانین تشخیص دهیم که X متعلق به کدام دسته است
- اگر R1 تنها قانون برآورده شده باشد، قانون با برگرداندن پیش بینی کلاس برای X دسته آن را مشخص می‌کند.
- اگر بیش از یک قانون triggered شود، نیاز به برطرف کردن تصادم داریم
- چون هر قانون ممکن است یک دسته مختلف برای X مشخص کند

چگونه با استفاده از قوانین دسته‌بندی کنیم؟

- برای رفع اختلاف دو استراتژی وجود دارد

۱) Size ordering : **بالاترین اولویت را به قانونی می‌دهد که «دشوارترین» شروط را برآورد کرده است. یعنی بیشترین تعداد مقدم‌ها را داشته باشد**

۲) Rule ordering : به دو روش قابل انجام است

- Class-based ordering : قوانین بر اساس کلاس آن‌ها به ترتیب اهمیت مرتب می‌شوند. بنابراین قوانین مربوط به کلاس شایع‌تر (prevalent) اول می‌آیند
  - در این روش لازم نیست قوانین را مرتب نماییم
- Rule-based ordering : به قوانین اولویت می‌دهیم که این کار به روش‌های مختلفی انجام می‌شود
  - بر اساس دقت، پوشش، تعداد مقدم‌ها یا تشخیص فرد خبره
  - هر کدام زودتر fire شد، کلاس را تعیین می‌کند

Default Rule

روش‌های ایجاد rule-based classifier

۱. از طریق استخراج قوانین از درخت تصمیم

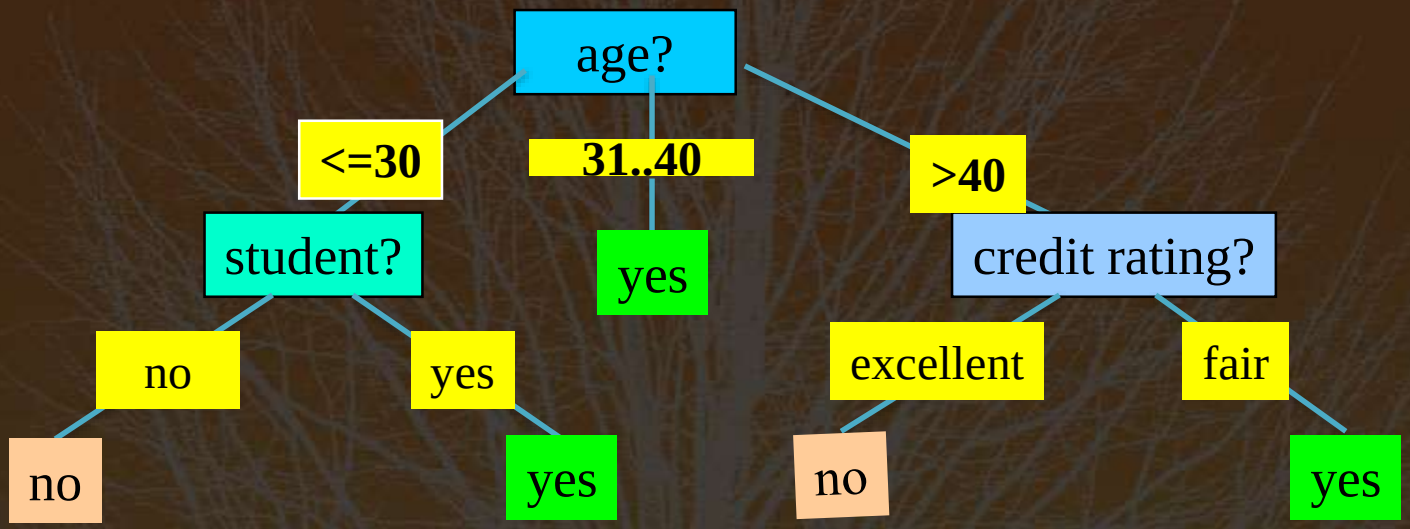
- دلیل اینکار : برخی اوقات درخت تصمیم بسیار بزرگ و غیرقابل فهم است

۲. با استفاده از الگوریتم Sequential Covering Algorithm

استخراج قوانین از درخت تصمیم

- برای هر مسیر از ریشه تا برگ یک قانون ایجاد می‌شود.
- هر زوج مقدار-خصوصیت در امتداد هر مسیر یک ترکیب عطفی می‌سازد: برگ دربردارنده برچسب کلاس است.
- قوانین دو به دو ناسازگار و جامع هستند
- **دو به دو ناسازگار (Mutually Exclusive):** به این معناست که تصادم قانون وجود ندارد، زیرا هیچ دو قانونی برای یک تاپل تریگر نمی‌شوند.
- **جامع (Exhaustive):** به این معناست که برای هر زوج ممکن مقدار-خصوصیت یک قانون وجود دارد
- بنابراین
  - نیازی به استفاده از قانون پیش‌فرض نداریم
  - نیازی به رتبه‌بندی قوانین نداریم

استخراج قوانین از درخت تصمیم



مثال : قوانین استخراج شده از درخت تصمیم مربوط به خرید کامپیوتر

IF *age* = young AND *student* = *no* THEN *buys_computer* = *no*
IF *age* = young AND *student* = *yes* THEN *buys_computer* = *yes*
IF *age* = mid-age THEN *buys_computer* = *yes*
IF *age* = old AND *credit_rating* = *excellent* THEN *buys_computer* = *no*
IF *age* = old AND *credit_rating* = *fair* THEN *buys_computer* = *yes*

الگوریتم پوشش ترتیبی (Sequential Covering Algorithm)

۱) برای هر کلاس با کلی‌ترین قانون شروع می‌کنیم

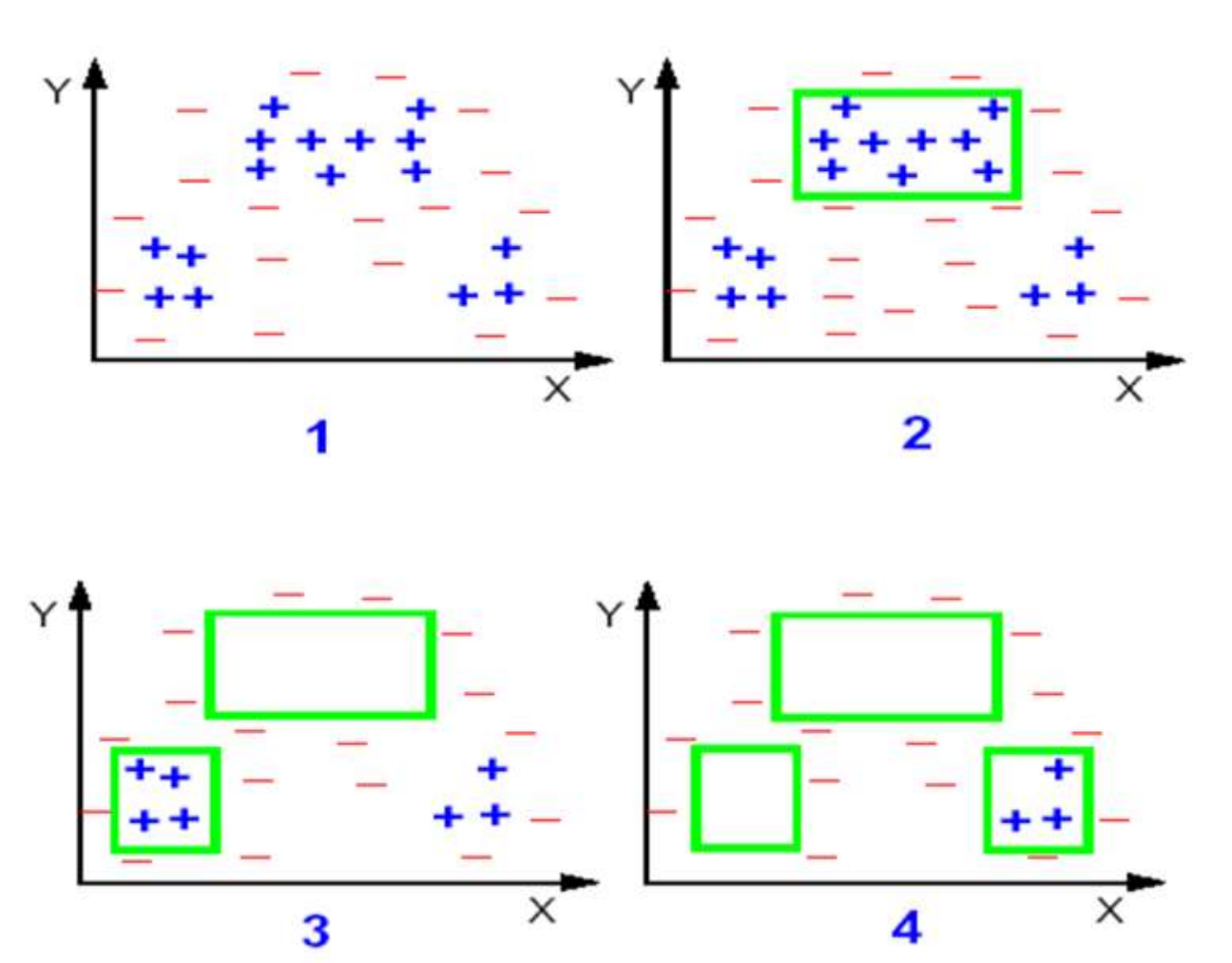
- IF {} THEN loan=accept

۲) در هر مرحله یک قانون را یاد بگیر که بیشترین مثال‌های آن کلاس را پوشش دهد

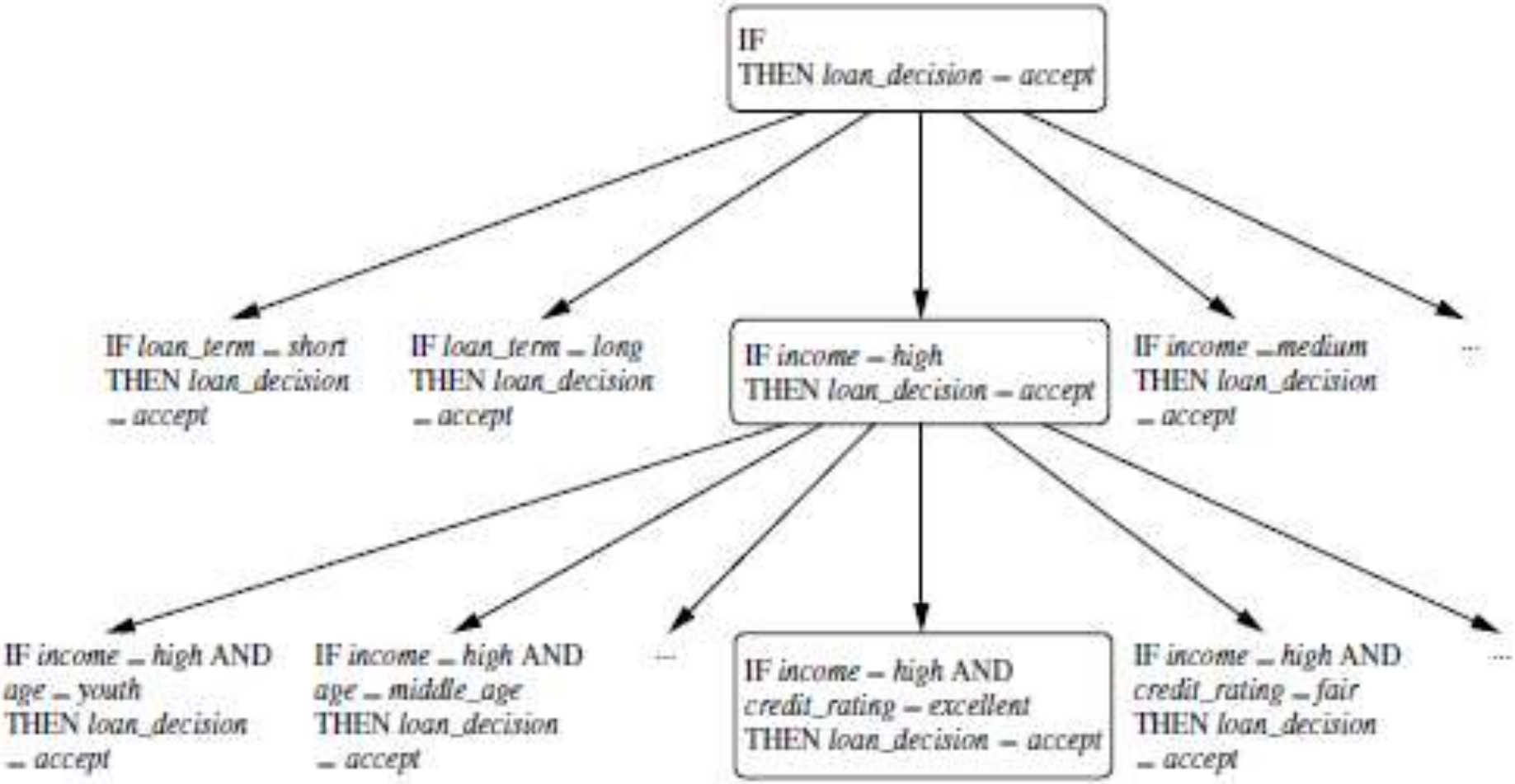
۳) داده‌هایی که پوشش داده شده است را از مجموعه آموزش پاک کن

۴) آنقدر این دو مرحله را تکرار کن که هیچ مثالی از آن کلاس باقی نماند یا کارایی پایین آید

الگوریتم پوشش ترتیبی



نمایی از الگوریتم



روش پوشش ترتیبی

الگوریتم پوشش ترتیبی: قوانین را به طور مستقیم از داده های آموزشی استخراج می کند.

نمونه الگوریتم های پوشش ترتیبی : FOIL، AQ، CN2، RIPPER

گام‌های الگوریتم:

- قوانین به صورت یک به یک آموخته می‌شوند.
- هر قانونی که آموخته می شود، تاپل‌هایی که توسط آن قانون پوشش داده می‌شوند حذف می‌شوند.

- تا زمانی که شرط پایان برآورده شود
  1) مثلا تا جایی که هیچ مثال آموزشی دیگری باقی نمانده باشد یا
  2) تا جایی که کیفیت قانون ایجاد شده کمتر از حد مشخص شده توسط کاربر باشد

  فرایند را روی چندتایی های باقیمانده تکرار کنید.

الگوریتم پوشش ترتیبی

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:

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

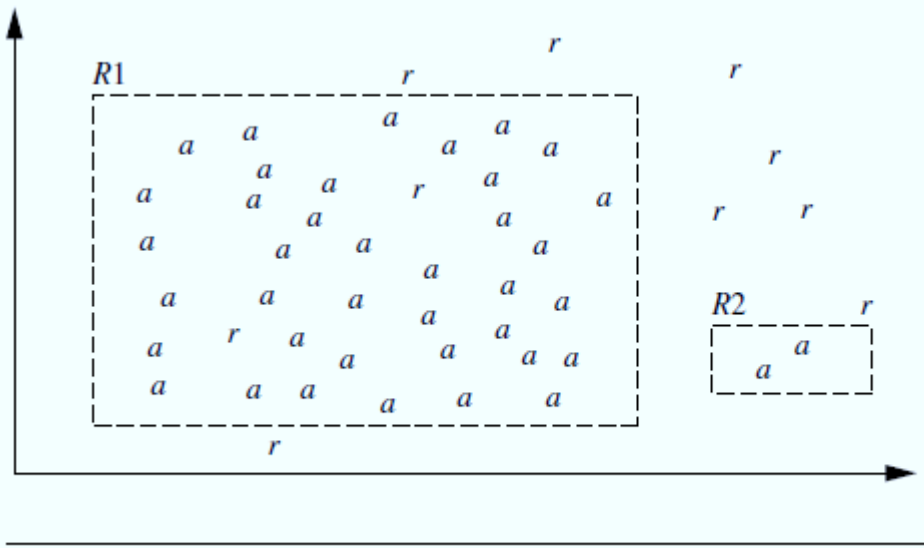
مشکلات روش پوشش ترتیبی

- در این روش در هر مرحله یکی از شاخه‌ها انتخاب می‌شود
- مشکل اول :
- روش انتخاب به صورت حریصانه است
- بنابراین ممکن است قانون خوبی تولید نکند
- راه‌حل :
  - در هر مرحله k بهترین انتخاب‌ها گزینش شوند

- مشکل دوم :
- در مراحل ساخت قوانین نیاز داریم که تعیین کنیم یک ویژگی جدید را به قانون اضافه کنیم یا خیر؟
- این کار مستلزم تعیین معیاری برای انجام دادن یا ندادن است
- معیار دقت یا accuracy را در نظر بگیریم ؟

مشکل معیار accuracy تنها

در صورتی که تنها دقت را در نظر بگیریم مشکل زیر پیش می‌آید



Rules for the class *loan_decision* = *accept*, showing *accept* (*a*) and *reject* (*r*) tuples.

قانون ۱
۴۰ مورد را پوشش داده است
۳۸ مورد درست

قانون ۲
۲ مورد را پوشش داده است
هر دو درست

نتیجه : آیا تنها معیار دقت کافی است؟

معیارهای تشخیص قانون مناسب

- فرض کنید می‌خواهیم قانونی مانند R را برای دسته‌بندی کلاس c ایجاد کنیم
- قانونی که تا به حال ایجاد شده دارای شرط‌هایی است که condition می‌نامیم
- می‌خواهیم تشخیص دهیم آیا ویژگی جدیدی به قانون ساخته شده اضافه کنیم یا خیر؟
- قانون جدید را با R′ و شرایط جدید آن را با condition′ نشان می‌دهیم
- می‌خواهیم معیارهایی تعیین کنیم که قانن جدید R′ مناسب است یا خیر؟

معیارهای تشخیص قانون مناسب

- **معیار اول**
  - از معیار information gain در ساخت درخت تصمیم استفاده میکنیم
  - در واقع نام دیگر آن entropy است

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

  - **سوال : آنترپی بیشتر باشد یا کمتر باشد؟**
- هرچه کمتر باشد یعنی
- اینکه شرایط جدید تعداد تاپل‌های بیشتری از یک کلاس را پوشش می‌دهند
- و از سایر کلاس‌ها تعداد کمتری را پوشش می‌دهند
- چون : مثلا

$$Entropy(D) = I(9,5) = -\frac{9}{14}\log_2(\frac{9}{14}) - \frac{5}{14}\log_2(\frac{5}{14}) = 0.940$$

معیارهای تشخیص قانون مناسب

- معیار دوم
- این معیار در روش FOIL به کار رفته است
- Pos و neg تعداد تاپل‌های درست پوشش داده شده توسط R
- Pos' و neg' در رابطه R' است

$$FOIL\_Gain = pos' \times \left(\log_2 \frac{pos'}{pos'+neg'} - \log_2 \frac{pos}{pos+neg}\right)$$

چطور یک قانون را یاد بگیریم؟

- هرس قانون بر اساس مجموعه ای مستقل از تاپل های آزمون Pos/neg کوتاه شده تاپل های مثبت/منفی تحت پوشش R هستند.

اگر FOIL_Prune برای نسخه هرس شده R بیشتر باشد، R را هرس کنید

$$FOIL\_Prune(R) = \frac{pos - neg}{pos + neg}$$

هرس کردن قوانین

- در زمان ساخت قوانین جهت ارزیابی تنها از معیارهای کیفیت (معیارهای قبلی) استفاده می‌شود
- اما این ارزیابی یک ارزیابی خوشبینانه است
- اما ممکن است در عمل (تست) دقت خوبی نداشته باشیم
- در واقع مدل ما overfit است و مانند درخت تصمیم از هرس کردن استفاده می‌کنیم
- برای هرس کردن روش‌های مختلفی ارائه شده است
- به طور مثال در FOIL از یک مجموعه pruning set استفاده می‌شود و بر اساس معیار زیر هرس انجام می‌گیرد

$$FOIL\_Prune(R) = \frac{pos - neg}{pos + neg}$$

- بنابراین اگر معیار بالا برای قانون هرس شده بیشتر باشد، هرس انجام می‌گیرد

دسته‌بندی : مفاهیم اولیه

- مفاهیم اولیه
- درخت تصمیم (Decision Tree)
- روش‌های دسته‌بندی بیز (Bayes)
- روش‌های دسته‌بندی مبتنی بر قانون (Rule-Based)
- ارزیابی مدل‌های و انتخاب مدل مناسب
- روش‌های بهبود دقت دسته‌بندی
- خلاصه فصل

مدل ارزیابی و انتخاب

معیارهای ارزیابی: چگونه میتوانیم درستی مدل را اندازه‌گیری کنیم؟ سایر معیارهای سنجش چیست؟

استفاده از validation test set : برای ارزیابی از یک مجموعه برای تست استفاده می‌شود

- روش‌های ارزیابی درستی دسته‌بندی:

۱) متدهای نگهدارنده (Holdout method)

زیر نمونه تصادفی (random subsampling)

۲) اعتبار سنجی (Cross-validation)

۳) تکرار سازی تصادفی (Bootstrap)

- مقایسه دسته‌بندی ها:

۱) فاصله اطمینان

۲) تجزیه و تحلیل با کمک منحنی ROC

معیارهای سنجش دسته‌بندی:(Confusion Matrix)

Confusion Matrix:

Actual class\Predicted class	C_1	$\neg C_1$
C_1	True Positives (TP)	False Negatives (FN)
$\neg C_1$	False Positives (FP)	True Negatives (TN)

مثال:

پیش بینی\واقعیت	خرید کامپیوتر=بله	خرید کامپیوتر=خیر	جمع
خرید کامپیوتر=بله	**6954**	**46**	7000
خرید کامپیوتر=خیر	**412**	**2588**	3000
جمع	7366	2634	10000

سایر معیارهای ارزیابی دسته‌بندی

A\P	C	¬C	
C	TP	FN	P
¬C	FP	TN	N
	P’	N’	All

- مشکل کلاس‌های غیرمتعادل:

یک نوع ممکن است نادر باشد.

به عنوان مثال نتیجه آزمایش HIVمثبت.

- در دسته‌بندی اکثریت منفی باشد و اقلیت مثبت
- میزان حساسیت(**Sensitivity**):نرخ تشخیص مثبت واقعی

Sensitivity = TP/P

- میزان خاص بودن(**Specificity**):نرخ تشخیص منفی واقعی

Specificity = TN/N

- درستی دسته‌بندی(**Classifier Accuracy**) یا نرخ شناخت:درصدی که به درستی دسته‌بندی شده اند.

Accuracy = (TP + TN)/All

- **Error rate:** *1 – accuracy*, or

Error rate = (FP + FN)/All

معیارهای ارزیابی دسته‌بندی:

- (Precision): درست بودن-چند درصداز تاپل‌هایی که به عنوان داده درست دسته‌بندی شده‌اند و واقعا درست هستند؟

$$\text{Precision}=\frac{TP}{TP+FP}$$

- (Recall): کامل بودن-چند درصد از تاپل های درست با برچسب درست طقه بندی شده اند؟

$$\text{recall}=\frac{TP}{TP+FN}$$

- F-measures: میانگین هارمونیک بین Precision وRecall

$$F=\frac{2*\text{Precision}*\text{Recall}}{\text{Precision}+\text{Recall}}$$

- F_β : معیار وزنی precision and recall .

βباراز وزن را به درست بودن اختصاص میدهد.

$$F_\beta=\frac{(1+\beta^2)*\text{Precision}*\text{Recall}}{\beta^2*\text{Precision}+\text{Recall}}$$

مثال:

Actual Class\Predicted class	cancer = yes	cancer = no	Total	Recognition(%)
cancer = yes	**90**	**210**	300	30.00 (*sensitivity*
cancer = no	**140**	**9560**	9700	98.56 (*specificity)*
Total	230	9770	10000	96.40 (*accuracy*)

- *Precision* = 90/230 = 39.13% *Recall* = 90/300 = 30.00%

معیارهای ارزیابی (خلاصه)

معیار	فرمول
Accuracy, recognition rate	$\frac{TP+TN}{P+N}$
Error rate, misclassification rate	$\frac{FP+FN}{P+N}$
Sensitivity	$\frac{TP}{P}$
Specificity	$\frac{TN}{N}$
Precision	$\frac{TP}{TP+FP}$
F, F_1 , F-score: میانگین بین precision و recall	$\frac{2*Precision*Recall}{Precision+Recall}$
F_β β یک عدد حقیقی مثبت است	$\frac{(1+\beta^2)*Precision*Recall}{\beta^2*Precision+Recall}$

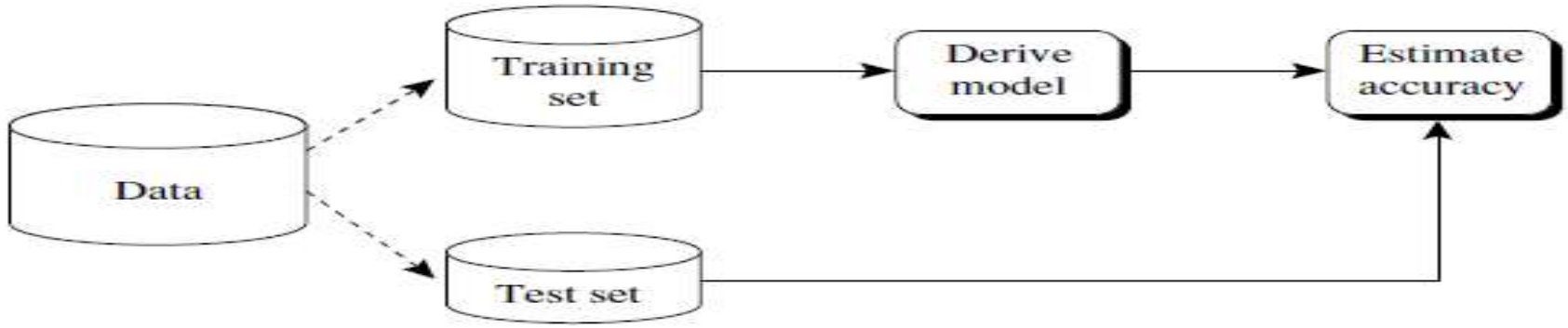
نحوه به دست آوردن یک دسته‌بندی قابل اطمینان

- روش‌های تشخیص دقت دسته‌بندی:
  - Holdout method, random subsampling
  - Cross-validation
  - Bootstrap

- مقایسه دسته‌بندی‌ها :
- فاصله اطمینان Confidence intervals
- تحلیل هزینه و سود و منحنی ROC

ارزیابی دقت به روش (hold out) (۱)

- روش (hold out)
- اطلاعات داده شده به طور تصادفی به دو مجموعه مستقل تقسیم می شوند.
- مجموعه آموزش (به عنوان مثال، ۲/۳) برای ایجاد مدل
- مجموعه آزمون (به عنوان مثال، ۱/۳) برای برآورد دقت
- نمونه گیری تصادفی(Random sampling) : تعمیم یافته‌ی روش Hold out
- روش holdout ، K بار تکرار می‌شود، دقت کلی می‌شود میانگین k بار.



Estimating accuracy with the holdout method.

ارزیابی مدل‌های ایجاد شده (۲)

- **Cross-validation** (K-fold، که در آن k=10 متداول‌ترین است)

داده ها به صورت تصادفی به K زیر مجموعه متقابلا منحصر به فرد تقسیم می‌شوند , با اندازه‌هایی تقریبا برابر.

در تکرار i ام، Di به عنوان مجموعه تست و دیگر مجموعه‌ها به عنوان مجموعه آموزش استفاده می‌شوند.

- **Leave-one-out**
- **مانند روش فوق با این تفاوت که k برابر تعداد تاپل‌های مجموعه است**

- Startified cross-validation
- ممکن است توزیع کلاس‌ها نامتوازن باشد بنابراین
- پارتیشن‌ها به شکلی تقسیم می‌شوند که نسبت توزیع کلاس‌ها حفظ شود (۶۰ درصد کارشناسی)

ارزیابی مدل‌های ایجاد شده (۳)

- روش Bootstrap
- در این روش آموزش به صورت sampling with replacement انجام می‌گیرد
- بنابراین هر داده‌ی آموزشی ممکن است مجدد انتخاب شود
- برای این منظور روش‌های مختلف bootstrap مطرح شده است
- روش bootstrap 632.
- در این روش اگر مجموعه داده d تاپل باشد، dبار نمونه‌گیری می‌کنیم
- ممکن است تاپل تکرار داشته باشیم
- از لحاظ آماری ثابت می‌شود که ۶۳,۲ درصد از داده‌ها در مجموعه bootstrap انتخاب خواهند شد
- بنابراین بقیه مجموعه یعنی ۳۶,۸ برای تست استفاده می‌شود

ارزیابی مدل‌های ایجاد شده (۳)

- برای ارزیابی به کمک این روش نمونه‌گیری k مرتبه انجام می‌گیرد
- در هر مرتبه مجموعه bootstrap set متفاوت خواهد بود
- برای محاسبه ارزیابی به کمک این روش از رابطه زیر استفاده می‌شود

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

مقایسه روش‌های دسته‌بندی

در این بخش می‌خواهیم نتایج روش‌های دسته‌بندی را با هم مقایسه کنیم

بطور مثال M1 و M2 دو روش دسته‌بندی هستند کدامیک بهترند؟

اگر معیار نرخ خطا (error rate) را برای مقایسه انتخاب کنیم درست است؟ (هر روشی میانگین خطای کمتری داشت بهتر است)

خیر چون

۱. برای ارزیابی به کمک روشی مانند K-fold ممکن است بین خطاهای هر مرحله تفاوت فاحشی باشد

۲. شاید خطای بین M1 و M2 تفاوت داشته باشد اما از لحاظ آماری خیلی متفاوت نباشند

۳. شاید اختلاف خطای پیش آمده به طور شانسی اتفاق افتاده باشد

مقایسه روش‌های دسته‌بندی:

فاصله اطمینان (Confidence intervals)

تجزیه و تحلیل هزینه-سود (cost-benefit) و منحنی ROC

برآورد فاصله اطمینان

- **فرض کنید که ما دو روش دسته‌بندی M1 و M2 داریم. کدام روش بهتر است؟**
- در این روش می‌خواهیم تشخیص دهیم آیا تفاوت بین خطای دو مدل، یک تفاوت واقعی است؟
- روش کار
- فرض کنید روش 10-fold cross validation ، را ۱۰ بار اجرا کنیم
- برای خطاهای بدست آمده در هر مرحله می‌توانیم میانگین بگیریم و «میانگین خطا» بنامیم $\overline{err}(M_1)$
- از لحاظ آماری نمونه‌های انتخاب شده از توزیع t-dirstribution با k-1 درجه آزادی تبعیت می‌کنند
- این نکته آماری را می‌توان در t-test به کار برد

T-test

فرضیه صفر : فرض می‌کنیم دو روش M1 و M2 یکسان هستند.

اگر ما بتوانیم فرض صفر را رد کنیم

نتیجه می گیریم که تفاوت بین M1 و M2 از نظر آماری معنی دار است.

در ابتدا مقدار t را از رابطه‌های زیر بدست می‌آوریم

$$t = \frac{\overline{err}(M_1) - \overline{err}(M_2)}{\sqrt{var(M_1 - M_2)/k}}$$

$$var(M_1 - M_2) = \frac{1}{k}\sum_{i=1}^{k}[err(M_1)_i - err(M_2)_i - (\overline{err}(M_1) - \overline{err}(M_2))]^2 .$$

T-test

- در مرحله بعدی یک آستانه significance level در نظر می‌گیریم
- به طور مثال اگر آستانه را ۵٪ در نظر بگیریم
- یعنی اینکه می‌خواهیم تشخیص دهیم آیا این دو مدل در ۹۵ درصد از داده‌ها تفاوت معنادار Significant دارند یا خیر

- برای این منظور z=sig/2 در نظر میگیریم
- جدول توزیع-t در کتاب‌های آماری آورده می‌شود
- درجه آزادی نیز برای k-1 در نظر گرفته می‌شود (k-fold)
- در صورتی عدد بدست آمده در جدول آماری را z بنامیم
- در صورتی که t>z یا t<-z باشد فرض صفر را رد می‌کنیم
- یعنی بین دو مدل تفاوت معناداری در دنیای واقع وجود دارد
- در صورتی که فرض رد نشود، می‌توان نتیجه گرفت که تفاوت شانسی بود است

جدول توزیع t

TABLE B: *t*-DISTRIBUTION CRITICAL VALUES

	Tail probability *p*											
df	.25	.20	.15	.10	.05	.025	.02	.01	.005	.0025	.001	.0005
1	1.000	1.376	1.963	3.078	6.314	12.71	15.89	31.82	63.66	127.3	318.3	636.6
2	.816	1.061	1.386	1.886	2.920	4.303	4.849	6.965	9.925	14.09	22.33	31.60
3	.765	.978	1.250	1.638	2.353	3.182	3.482	4.541	5.841	7.453	10.21	12.92
4	.741	.941	1.190	1.533	2.132	2.776	2.999	3.747	4.604	5.598	7.173	8.610
5	.727	.920	1.156	1.476	2.015	2.571	2.757	3.365	4.032	4.773	5.893	6.869
6	.718	.906	1.134	1.440	1.943	2.447	2.612	3.143	3.707	4.317	5.208	5.959
7	.711	.896	1.119	1.415	1.895	2.365	2.517	2.998	3.499	4.029	4.785	5.408
8	.706	.889	1.108	1.397	1.860	2.306	2.449	2.896	3.355	3.833	4.501	5.041
9	.703	.883	1.100	1.383	1.833	2.262	2.398	2.821	3.250	3.690	4.297	4.781
10	.700	.879	1.093	1.372	1.812	2.228	2.359	2.764	3.169	3.581	4.144	4.587
11	.697	.876	1.088	1.363	1.796	2.201	2.328	2.718	3.106	3.497	4.025	4.437
12	.695	.873	1.083	1.356	1.782	2.179	2.303	2.681	3.055	3.428	3.930	4.318
13	.694	.870	1.079	1.350	1.771	2.160	2.282	2.650	3.012	3.372	3.852	4.221
14	.692	.868	1.076	1.345	1.761	2.145	2.264	2.624	2.977	3.326	3.787	4.140
15	.691	.866	1.074	1.341	1.753	2.131	2.249	2.602	2.947	3.286	3.733	4.073
16	.690	.865	1.071	1.337	1.746	2.120	2.235	2.583	2.921	3.252	3.686	4.015
17	.689	.863	1.069	1.333	1.740	2.110	2.224	2.567	2.898	3.222	3.646	3.965
18	.688	.862	1.067	1.330	1.734	2.101	2.214	2.552	2.878	3.197	3.611	3.922
19	.688	.861	1.066	1.328	1.729	2.093	2.205	2.539	2.861	3.174	3.579	3.883
20	.687	.860	1.064	1.325	1.725	2.086	2.197	2.528	2.845	3.153	3.552	3.850
21	.686	.859	1.063	1.323	1.721	2.080	2.189	2.518	2.831	3.135	3.527	3.819
22	.686	.858	1.061	1.321	1.717	2.074	2.183	2.508	2.819	3.119	3.505	3.792
23	.685	.858	1.060	1.319	1.714	2.069	2.177	2.500	2.807	3.104	3.485	3.768
24	.685	.857	1.059	1.318	1.711	2.064	2.172	2.492	2.797	3.091	3.467	3.745
25	.684	.856	1.058	1.316	1.708	2.060	2.167	2.485	2.787	3.078	3.450	3.725
26	.684	.856	1.058	1.315	1.706	2.056	2.162	2.479	2.779	3.067	3.435	3.707
27	.684	.855	1.057	1.314	1.703	2.052	2.158	2.473	2.771	3.057	3.421	3.690
28	.683	.855	1.056	1.313	1.701	2.048	2.154	2.467	2.763	3.047	3.408	3.674
29	.683	.854	1.055	1.311	1.699	2.045	2.150	2.462	2.756	3.038	3.396	3.659
30	.683	.854	1.055	1.310	1.697	2.042	2.147	2.457	2.750	3.030	3.385	3.646
40	.681	.851	1.050	1.303	1.684	2.021	2.123	2.423	2.704	2.971	3.307	3.551
50	.679	.849	1.047	1.299	1.676	2.009	2.109	2.403	2.678	2.937	3.261	3.496
60	.679	.848	1.045	1.296	1.671	2.000	2.099	2.390	2.660	2.915	3.232	3.460
80	.678	.846	1.043	1.292	1.664	1.990	2.088	2.374	2.639	2.887	3.195	3.416
100	.677	.845	1.042	1.290	1.660	1.984	2.081	2.364	2.626	2.871	3.174	3.390
1000	.675	.842	1.037	1.282	1.646	1.962	2.056	2.330	2.581	2.813	3.098	3.300
∞	.674	.841	1.036	1.282	1.645	1.960	2.054	2.326	2.576	2.807	3.091	3.291

مقایسه روش‌های دسته‌بندی

مقایسه روش‌های دسته‌بندی:

فاصله اطمینان (Confidence intervals)

تجزیه و تحلیل هزینه-سود (cost-benefit)

منحنی ROC

Cost-benefit

- برای تحلیل هزینه و سود (risk & gain) میتوان از معیارهای زیر استفاده کرد
- TP, TN, FP, FN
- اما گاهی هزینه مرتبط با false negative (به طور مثال اینکه یک بیمار سرطانی را به اشتباه تشخیص دهیم سرطان ندارد)
- از یک اشتباه دیگر مانند false positive (یعنی به طور اشتباه کسی که بیمار نیست را بگوییم سرطان دارد) هزینه بیشتری دارد

- بنابراین در برخی نرم‌افزارهای داده‌کاوی (نرم‌افزارهای مالی، بیماری) این معیارهای وزن متفاوتی خواهند داشت

مقایسه روش‌های دسته‌بندی

مقایسه روش‌های دسته‌بندی:

فاصله اطمینان (Confidence intervals)

تجزیه و تحلیل هزینه-سود (cost-benefit)

منحنی ROC

ROC

- از این معیار برای مقایسه مدل‌های دسته‌بندی به طور گرافیکی استفاده می‌شود
- نتیجه حاصل از این تحلیل منحنی‌هایی هستند که با استفاده از آن‌ها به راحتی می‌توان تشخیص داد کدام مدل مناسب‌تر است

- منحنی ایجاد شده trade-off بین نرخ true positive و نرخ false positive است
- این دو نرخ به صورت زیر محاسبه می‌شوند
- TPR=TP/P و FPR=FP/N

- در منحنی ROC، بخش TP (در محور y) و FP (در محور x)

ROC

- برای ارزیابی از مجموعه تست استفاده می‌کنیم
- برای رسم منحنی از روش احتمال تعلق تاپل تست به یک کلاس استفاده می‌کنیم
- پس از مشخص شدن احتمال تعلق هر تاپل، تاپل‌ها را بر اساس احتمال آن‌ها مرتب می‌کنیم
- مرتب‌سازی بر این اساس است که آن‌هایی که احتمال yes شدن آن‌ها زیاد است در ابتدای لیست ظاهر می‌شوند
- به کمک روش بیزین می‌توان احتمال تعلق را بدست آورد، سایر روش‌ها مانند درخت تصمیم را نیز می‌توان تغییر داد تا احتمال را برگردانند
- برای رسم منحنی آستانه t در نظر گرفته می‌شود
- برای مسائل دو کلاسه تاپل‌هایی که f(X)>=t است جواب را yes در نظر می‌گیریم
- و برای تاپل‌هایی که f(X) < t باشد، جواب را false در نظر می‌گیریم

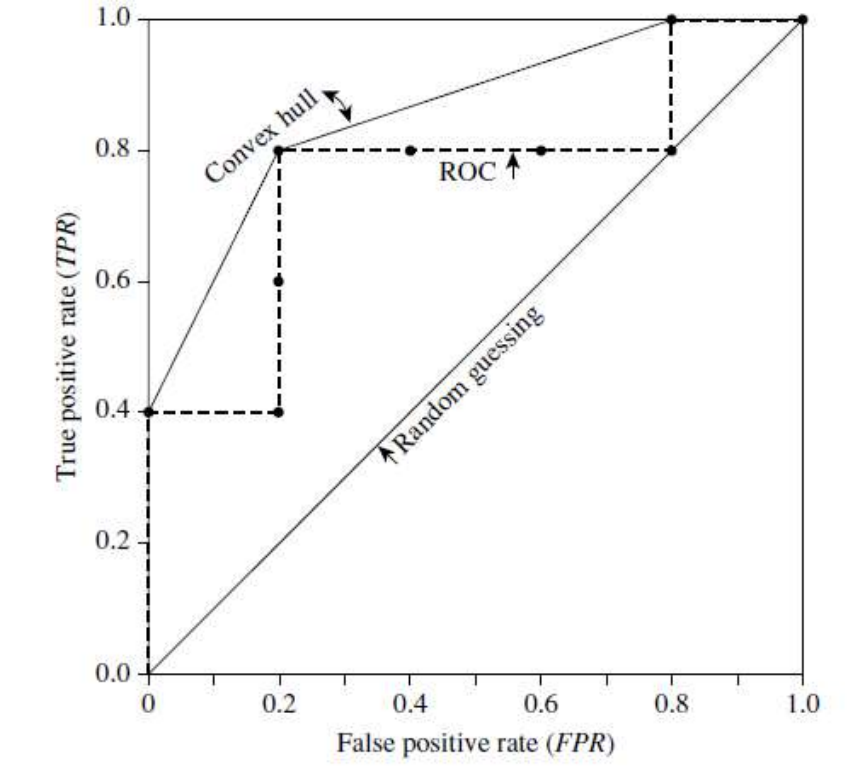
ROC

روش رسم منحنی

- نکته : اولین بار آستانه را برابر ردیف یک قرار می‌دهیم و در هر مرحله به مقدار ردیف بعدی تقلیل می‌دهیم

TPR=TP/P و FPR=FP/N

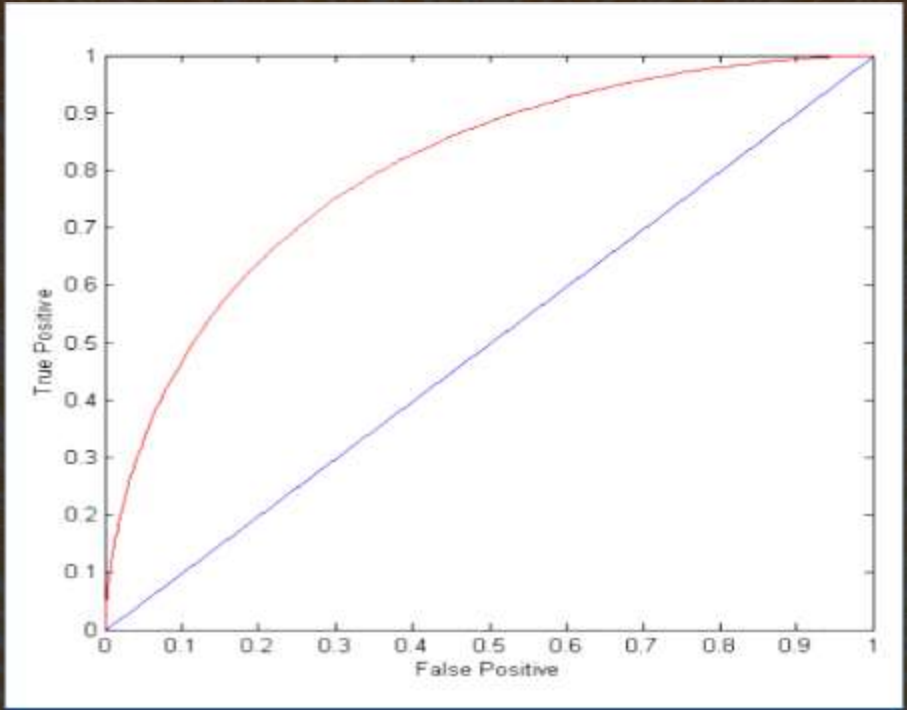
Tuple #	Class	Prob.	TP	FP	TN	FN	TPR	FPR
1	P	0.90	1	0	5	4	0.2	0
2	P	0.80	2	0	5	3	0.4	0
3	N	0.70	2	1	4	3	0.4	0.2
4	P	0.60	3	1	4	2	0.6	0.2
5	P	0.55	4	1	4	1	0.8	0.2
6	N	0.54	4	2	3	1	0.8	0.4
7	N	0.53	4	3	2	1	0.8	0.6
8	N	0.51	4	4	1	1	0.8	0.8
9	P	0.50	5	4	0	1	1.0	0.8
10	N	0.40	5	5	0	0	1.0	1.0



منحنی ROC

۵) تغییر آستانه در هر دو نرخ مثبت واقعی و مثبت کاذب تاثیر می‌گذارد.

نتیجه : روش classification که در منحنی آن سطح زیر نمودار بیشتر است، مناسب‌تر است



سایر مسائل مربوط به انتخاب مدل

۱. دقت مدل:

- دقت طبقه‌بندی : پیش‌بینی برچسب کلاس

۲. سرعت مدل:

- زمان برای ساخت مدل (زمان آموزش)
- زمان استفاده از مدل (طبقه‌بندی / زمان پیش‌بینی)

۳. مقاومت مدل: قابلیتی که در آن اگر به مدل داده نویزی یا missing بدهیم باز پیش‌بینی درستی انجام دهد

۴. مقیاس‌پذیری : مدل پیشنهادی برای مجموعه داده‌های بزرگ کارا باشد

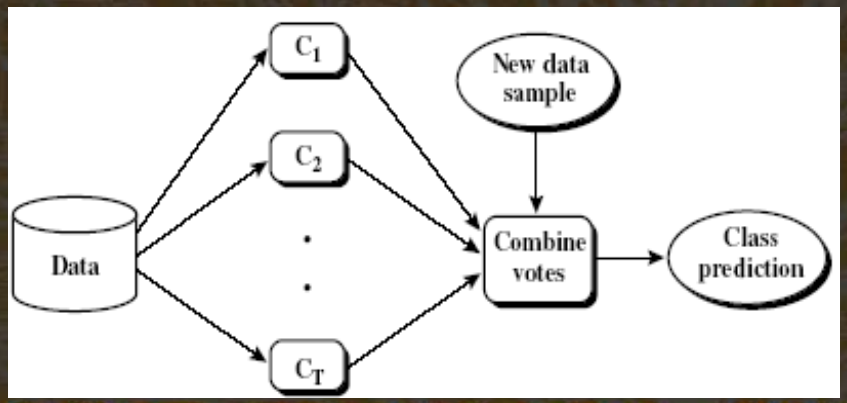
دسته‌بندی : مفاهیم اولیه

- مفاهیم اولیه
- درخت تصمیم (Decision Tree)
- روش‌های دسته‌بندی بیز (Bayes)
- روش‌های دسته‌بندی مبتنی بر قانون (Rule-Based)
- ارزیابی مدل‌های و انتخاب مدل مناسب
- روش‌های بهبود دقت دسته‌بندی
- خلاصه فصل

برای این منظور روش‌های متعددی وجود دارد
در این بخش تنها روش‌های گروهی مورد بحث قرار می‌گیرد

روش‌های گروهی (Ensemble Methods)

- روش‌های گروهی
  - استفاده از مد ل های ترکیبی به منظور افزایش دقت
  - یک سری از مدلهای آموخته شده را مانند M1,M2,.....Mk را با هدف ایجاد یک مدل بهبود یافته M ترکیب کنید.
  - تقویت کردن: آرا موافق با یک مجموعه از دسته‌بندی کننده‌ها
  - گروه: ترکیب یک سری از دسته‌بندی کننده‌های ناهمگن



bootstrap aggregation :(bagging)

- معیار: تشخیص بیماری بر اساس آرا چندین پزشک

Algorithm: Bagging. The bagging algorithm—create an ensemble of classification models for a learning scheme where each model gives an equally weighted prediction.

Input:

- D, a set of d training tuples;
- k, the number of models in the ensemble;
- a classification learning scheme (decision tree algorithm, naïve Bayesian, etc.).

Output: The ensemble—a composite model, $M*$.

Method:

(1) **for** $i = 1$ to k **do** // create k models:
(2) create bootstrap sample, D_i, by sampling D with replacement;
(3) use D_i and the learning scheme to derive a model, M_i;
(4) **endfor**

To use the ensemble to classify a tuple, X:

let each of the k models classify X and return the majority vote;

Boosting

- قیاس: مشاوره با چندین پزشکِ بر اساس ترکیب وزنی آن‌ها – وزن اختصاص داده شده بر اساس دقت تشخیص قبلی مشخص خواهد شد.

- مقایسه با bagging: روش Boosting گرایش به دقت بیشتری دارد ، اما این ریسک وجود دارد که مدل overfit شده و داده‌ها را به صورت نادرست دسته‌بندی نماید.

دسته‌بندی کلاس‌های غیر متعادل

- مشکل عدم تعادل دسته‌ها: مثال تشخیص طبی،کلاهبرداری،نشت نفت ،گسل و غیره به صورت فراوان وجود دارند.
- بکار بردن روش‌های سنتی توزیع متعادل طبقه‌ها و هزینه‌های خطای مساوی: برای داده‌های نا متعادل مناسب نمی‌باشد.
- روش‌های معمولی برای داده‌های آماری نا متعادل در ۲ کلاس دسته‌بندی:
  - Over sampling: انتخاب مجدد نمونه آمار و داده‌ها از کلاس مثبت
  - Under sampling: حذف تصادفی تاپل‌ها از کلاس منفی

- **هنوز برای مسئله عدم تعادل کلاس در مورد وظایف کلاسهای چند گانه مشکل وجود دارد.**

دانشگاه اصفهان

قول الرحمن

اِنّ اللّهَ لا یُغَیِّرُ ما بِقَومٍ حَتّیَ یُغَیِّروا ما بِاَنفُسِهِم

سوره رعد، آیه۱۱