Data Mining: Classification

Jef Wijsen

Université de Mons (UMONS)

Outline

Un exemple simple : les "Weather Data"

outlook	temperature	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

Classification et régression

Deux types de prédiction :

Classification Prédire un attribut non-numérique.

outlook	temperature	humidity	windy	play
sunny	cool	high	true	?

Régression Prédire un attribut numérique.

Méthodologie en grandes lignes

1. Créer un seul ensemble S d'exemples (instances, tuples, vecteurs...), avec les classes connues.
2. Diviser l'ensemble S en deux parties :
   - l'ensemble d'apprentissage A,
   - l'ensemble de test T.
3. S'appuyer sur A pour construire un modèle prédictif (appelé classificateur dans le cas d'une classification).
4. Vérifier que le modèle colle bien à T.
5. Utiliser le modèle pour prédire de nouveaux cas.

Différents types de modèle

- Les arbres de décision.
- Le voisin le plus proche.
- Les règles de classification.
- Modèle bayésien.
- Les réseaux de neurones.
- Les algorithmes génétiques.

Matrice de confusion

```
If windy=true then play=no
If windy=false then play=yes
```

Matrice de confusion :

		classe prédite	
		play=yes	play=no
classe observée	play=yes	6 (VP)	3 (FN)
	play=no	2 (FP)	3 (VN)

V=Vrai, F=Faux, P=Positif, N=Négatif

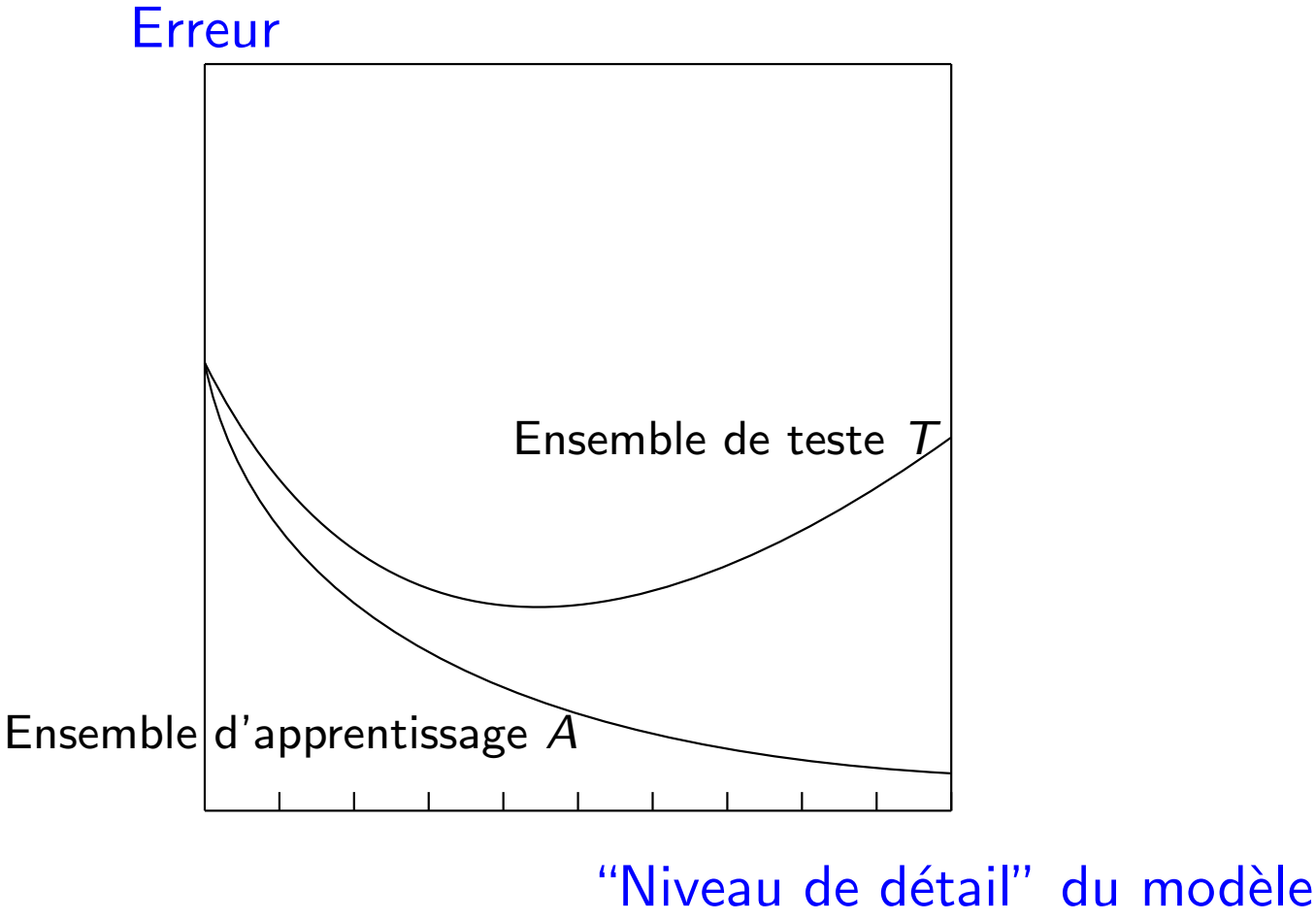
Le taux d'erreur est de 5/14.

Test Options in Weka

- Use training set
- Supplied test set
- Cross-validation (e.g. tenfold)
- Percentage split

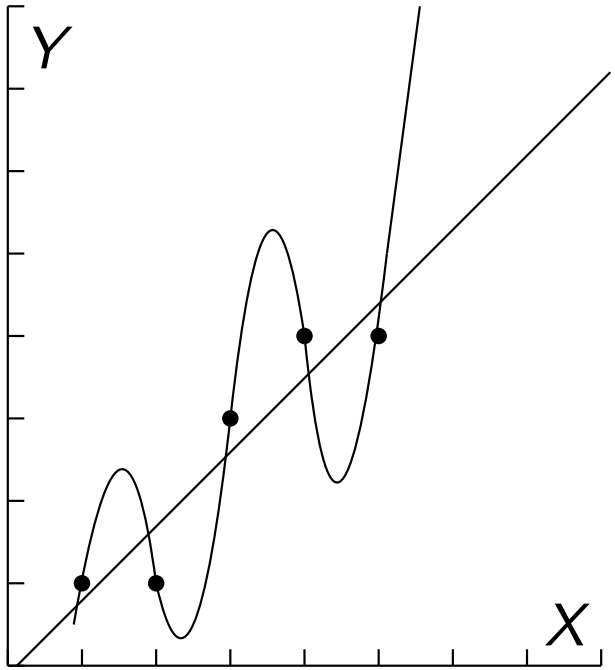
Underfitting et overfitting

Eviter l'overfitting : Ne pas se laisser séduire par un modèle qui colle très (trop) bien à l'ensemble d'apprentissage A.



Overfitting dans une régression

Prédire Y à partir de X se basant sur 5 points.



Overfitting

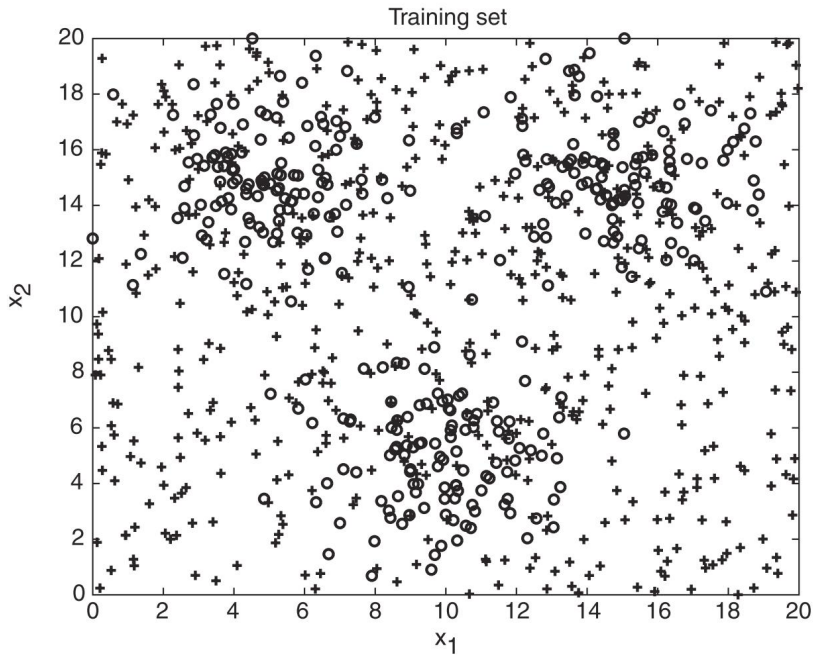


Figure 4.22. Example of a data set with binary classes.

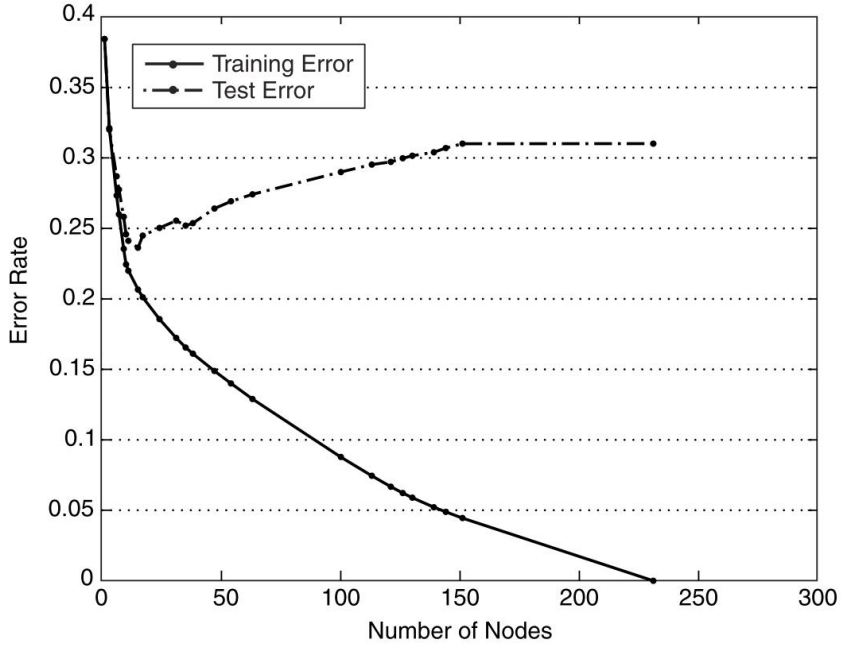
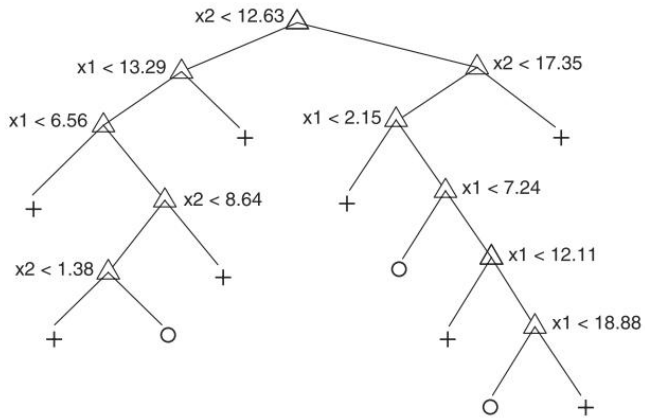


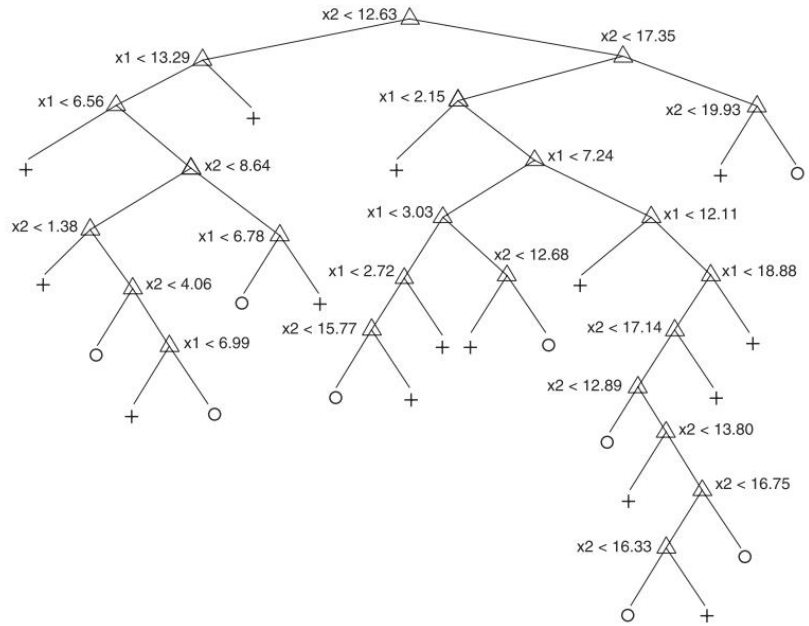
Figure 4.23. Training and test error rates.

Source: Pang-Ning Tan, Michael Steinbach, and Vipin Kumar: *Introduction to Data Mining*. Addison Wesley, 2006

Overfitting



(a) Decision tree with 11 leaf nodes.



(b) Decision tree with 24 leaf nodes.

Figure 4.24. Decision trees with different model complexities.

Source: Pang-Ning Tan, Michael Steinbach, and Vipin Kumar: *Introduction to Data Mining*. Addison Wesley, 2006

Outline

Feature Selection

Features are also called attributes or variables.

> *" In classification learning, the choice of attributes used to define examples is by far the single most determining factor of the success or failure of the learning algorithm."*
>
> Source: Usama M. Fayyad et al.: *From Digitized Images to Online Catalogs. Data Mining a Sky Survey* AI Magazine. 17(2), 1996

Relational Data Mining

Possession of credit cards may be an important feature.

C#	Salary	...	Sex	Class
1	25000	...	M	bad
2	24000	...	F	good
⋮	⋮	⋮	⋮	⋮

C#	Card
1	VISA
1	CB
1	AE
2	CB
⋮	⋮

Feature Selection

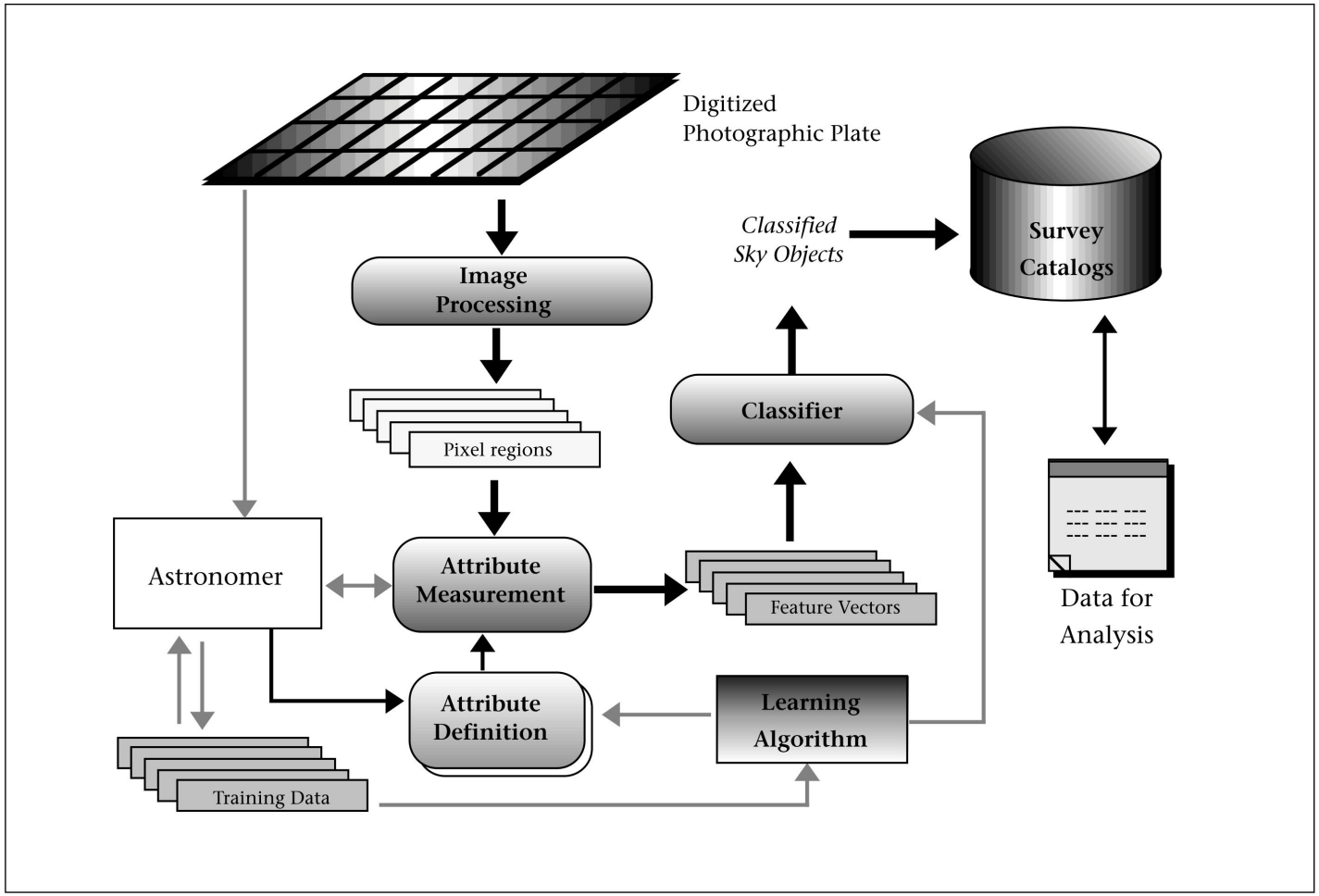


Figure 3. An Overview of the SKICAT Plate-Cataloging Process.

Source: Usama M. Fayyad et al.: *From Digitized Images to Online Catalogs. Data Mining a Sky Survey* AI Magazine. 17(2), 1996

Feature Selection

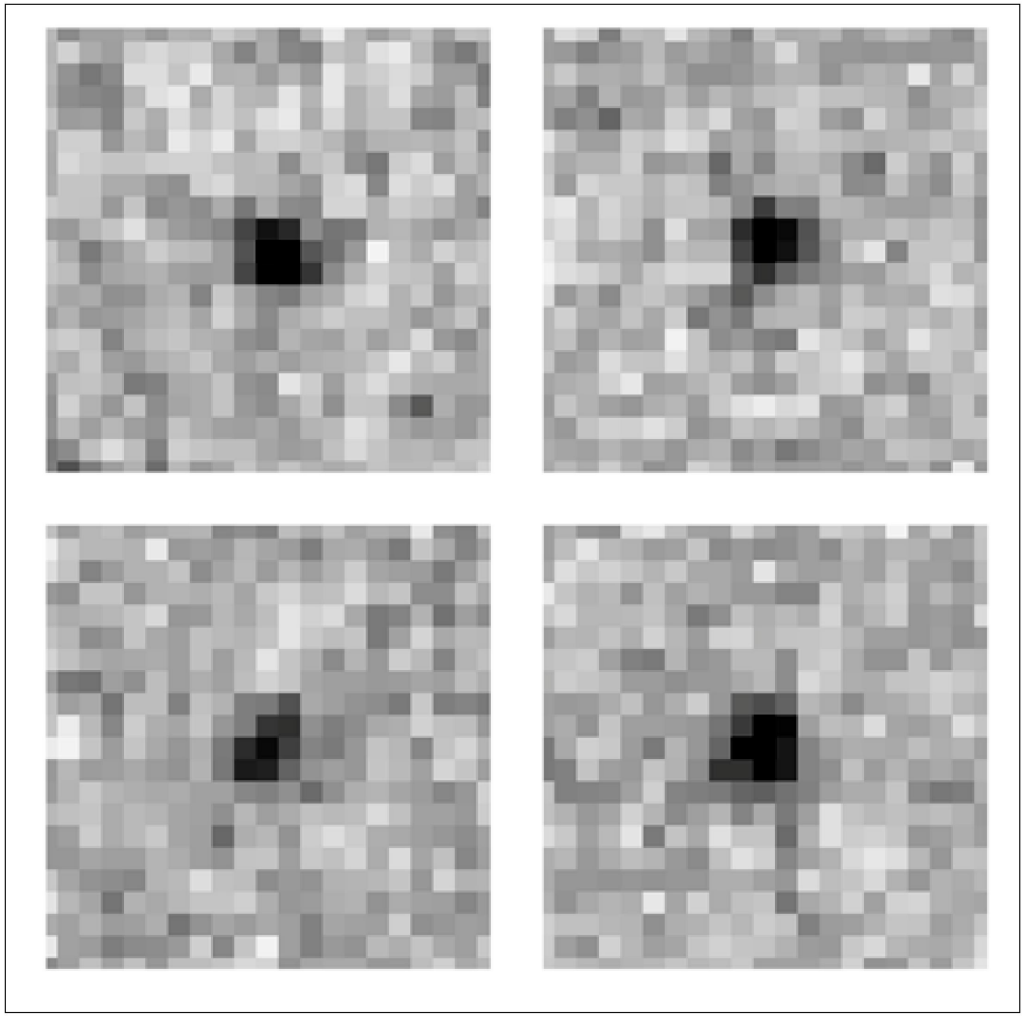


Figure 6. An Illustrative Example: Four Faint Sky Objects.

Source: Usama M. Fayyad et al.: *From Digitized Images to Online Catalogs. Data Mining a Sky Survey* AI Magazine. 17(2), 1996

Outline

Qu'est-ce qu'un arbre de décision ?

Nodes. Test a particular attribute (usually *propositional*, i.e., comparison of attribute with constant).

- Nominal attribute: one child for each possible attribute value; no testing of the same attribute further down the tree.
- Numeric attribute: two-way split ($<, \geq$) or three-way split ($<$, $>$, and $=$ or "between").

Leaf nodes. Classification.

Treatment of missing values:

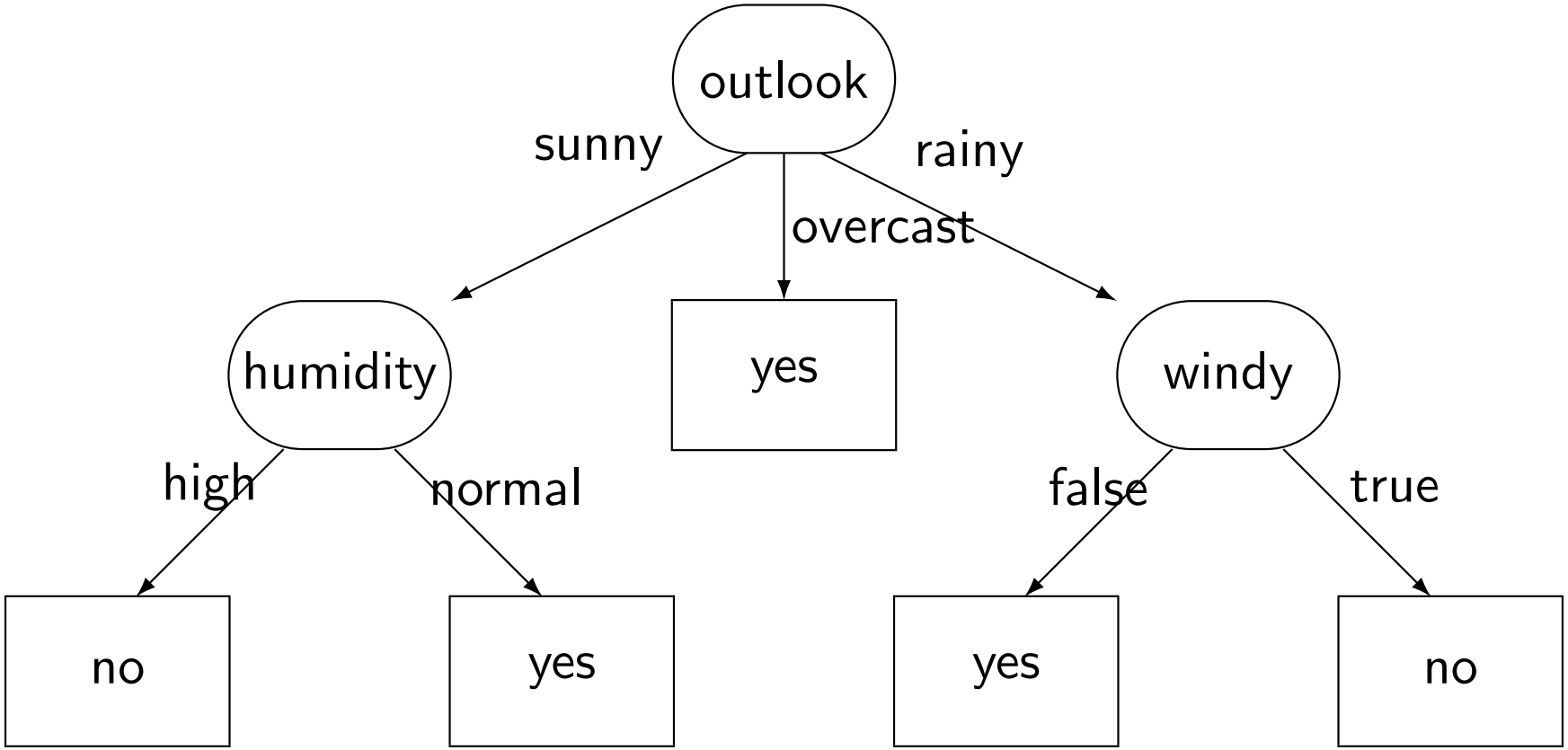
- treat "NULL" as attribute value in its own right, or
- use most popular branch.

Properties:

Completeness. At least one classification per example (because of exhaustive split).

Soundness or consistency. At most one classification per example (because of non-overlapping split).

Arbre de décision pour les Weather Data



Divide and Conquer: `classifiers.trees.Id3`

Information gain for creating a branch on the `outlook` attribute:

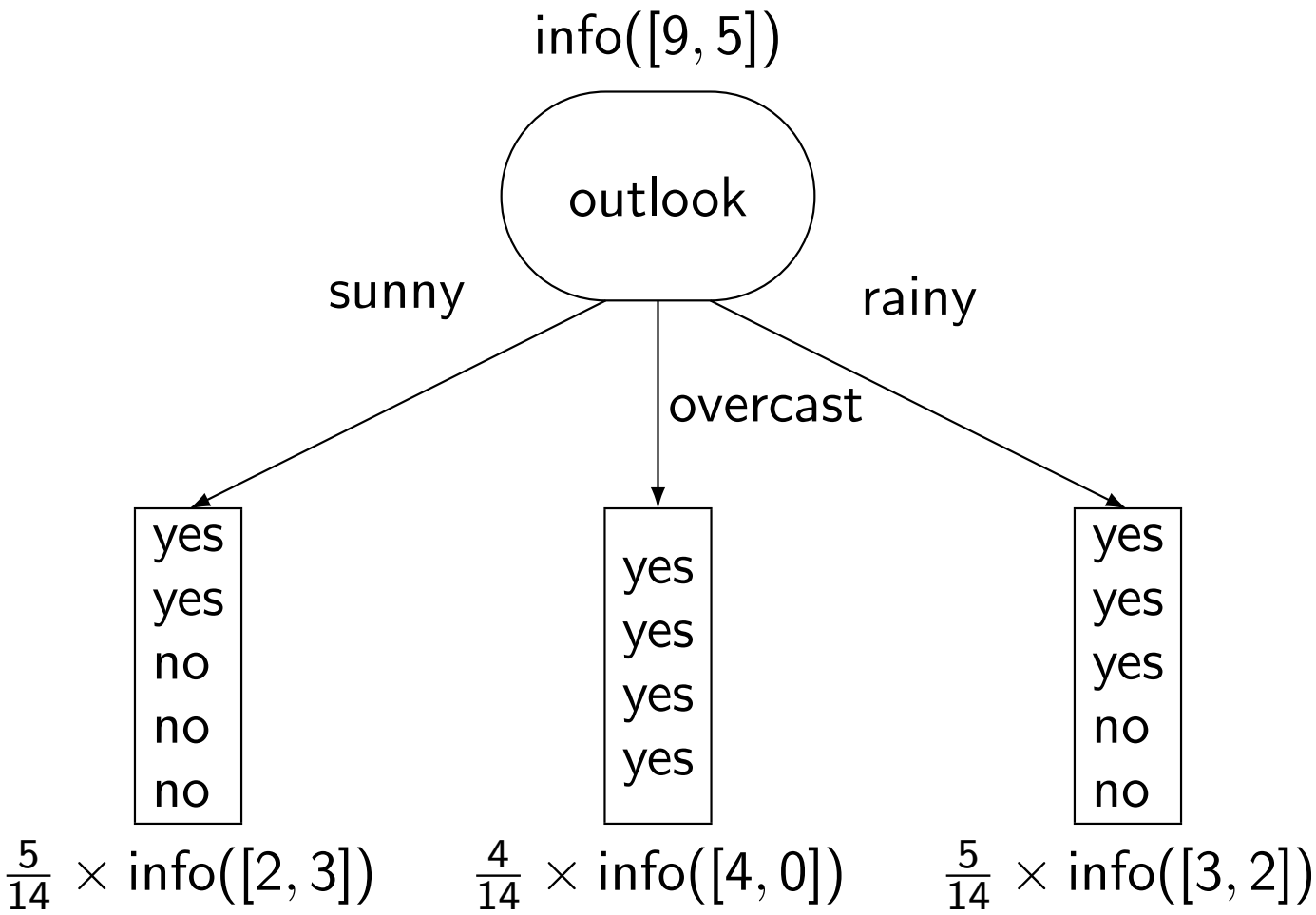
$$\text{info}([9,5]) - \frac{5}{14}\text{info}([2,3]) - \frac{4}{14}\text{info}([4,0]) - \frac{5}{14}\text{info}([3,2]) \ ,$$

which is equal to 0.247 (see further). In the same way, we obtain:

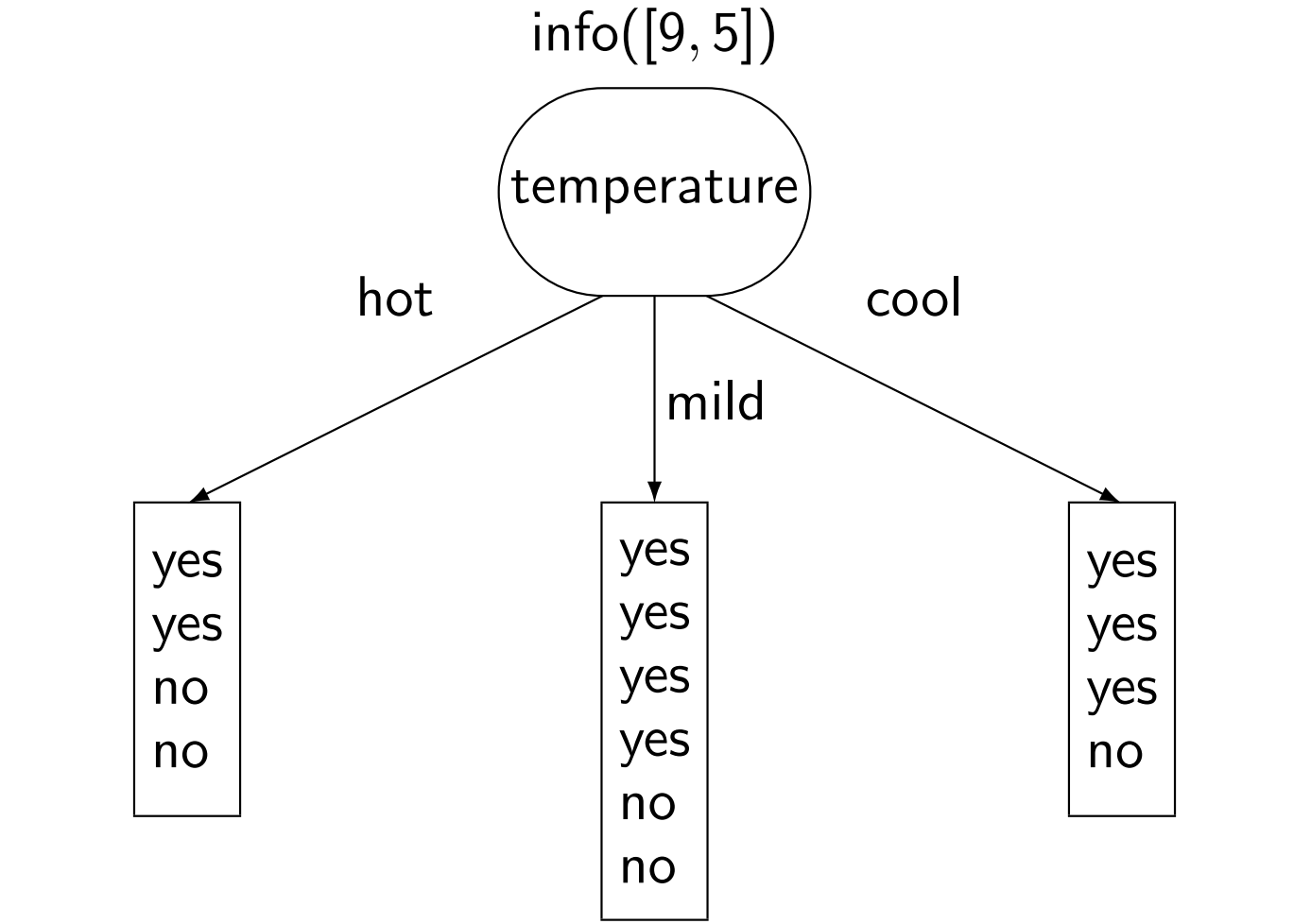
Attribute	Information gain
`outlook`	0.247
`temperature`	0.029
`humidity`	0.152
`windy`	0.048

So we select `outlook` as the splitting attribute at the root of the tree. Now the process can be repeated recursively for each branch, using only those instances that actually reach the branch. If at any time all instances at a node have the same classification, stop developing that part of the tree.

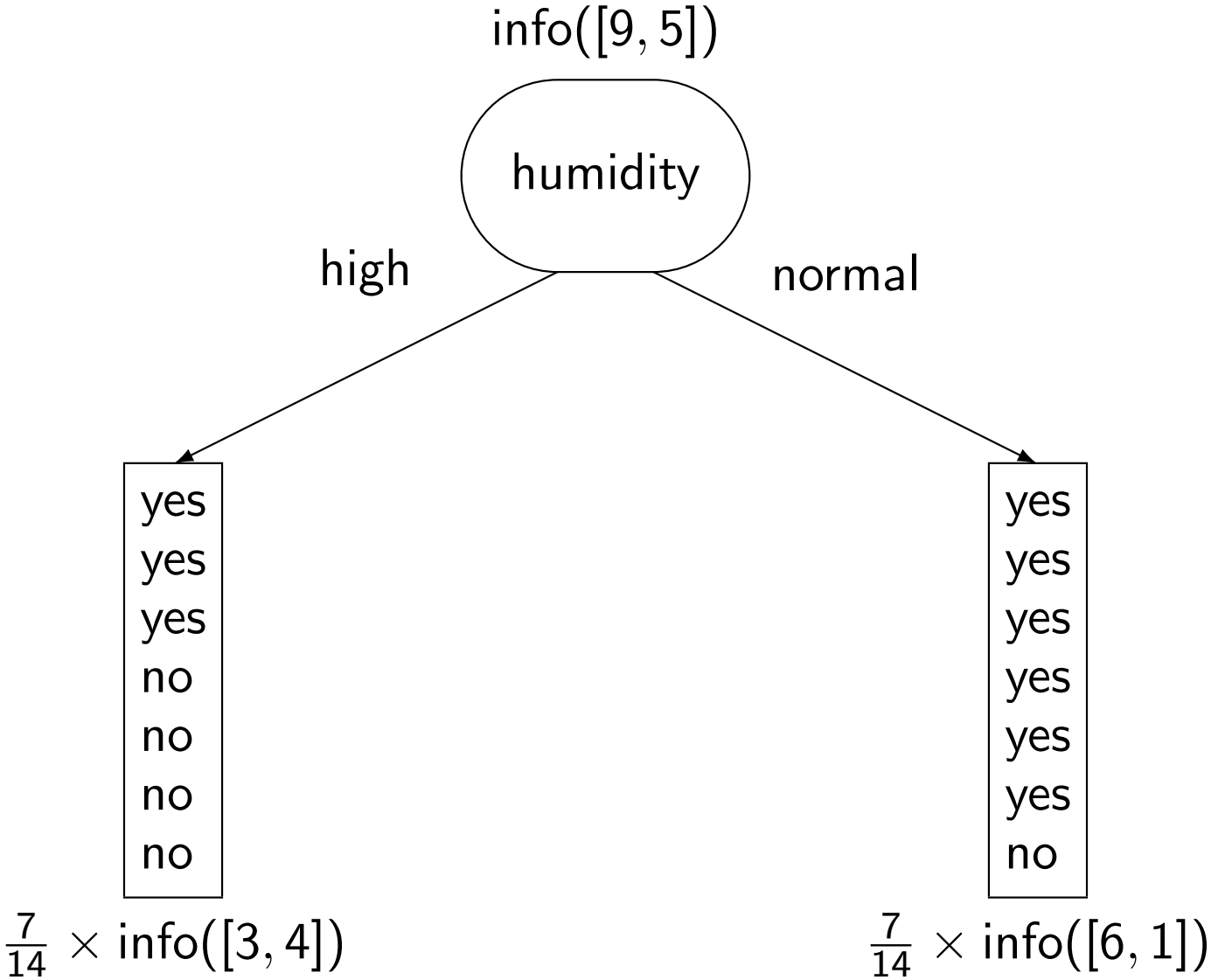
Choosing the Root I



Choosing the Root II



Choosing the Root III



Calculating Information I

Intuitively, info($[m_1, m_2]$), associated with a node of the tree, represents the expected amount of information that would be needed to specify whether a new instance should be classified `yes` or `no`, given that the example reached that node. What kind of properties would we expect info($[m_1, m_2]$) to have?

- Suppose that $2m$ examples reach a certain node. Then info($[2m, 0]$) should be zero, and info($[m, m]$) should reach a maximum (why?).
- The measure should be applicable to multiclass situations, not just to two-class ones. For example, info($[m_1, m_2, m_3]$) for three classes. Moreover, we should be able to make the decision involved in info($[m_1, m_2, m_3]$) in two stages:
  1. First decide whether it's the first case or one of the other two cases, info($[m_1, m_2 + m_3]$) ,
  2. and then decide which of the other two cases it is: info($[m_2, m_3]$) .

Calculating Information II

In some cases the second decision will not need to be made, namely where the decision turns out to be the first one. Taking this into account leads to the *multistage property*:

$$\begin{aligned}\text{info}([m_1, m_2, m_3]) &= \text{info}([m_1, m_2 + m_3]) \\ &\quad + \frac{m_2 + m_3}{m_1 + m_2 + m_3}\text{info}([m_2, m_3]) \ .\end{aligned}$$

Remarkably, it turns out that there is only one function that satisfies all these properties, and it is known as the *information value* or *entropy*:

Entropy

$$\text{info}([m_1, m_2, \ldots, m_n]) \quad = \quad \text{entropy}([p_1, p_2, \ldots, p_n])$$

where

$$\begin{aligned} \text{entropy}([p_1, p_2, \ldots, p_n]) &= -p_1 \log p_1 - p_2 \log p_2 \cdots - p_n \log p_n \\ p_i &= \frac{m_i}{\sum_{j=1}^{n} m_j} \quad (\forall i, 1 \leq i \leq n) \end{aligned}$$

For example,

$$\begin{aligned} \text{info}([2,3,4]) &= -\frac{2}{9}\log\frac{2}{9} - \frac{3}{9}\log\frac{3}{9} - \frac{4}{9}\log\frac{4}{9} \\ &= \frac{1}{9}(-2\log 2 - 3\log 3 - 4\log 4 + 9\log 9) \end{aligned}$$

Gini index

$$\text{info}([m_1, m_2, \ldots, m_n]) = \text{gini}([p_1, p_2, \ldots, p_n])$$

where

$$\begin{aligned} \text{gini}([p_1, p_2, \ldots, p_n]) &= 1 - (p_1)^2 - (p_2)^2 \cdots - (p_n)^2 \\ p_i &= \frac{m_i}{\sum_{j=1}^{n} m_j} \quad (\forall i, 1 \leq i \leq n) \end{aligned}$$

Classification error

$$\text{info}([m_1, m_2, \ldots, m_n]) = \text{error}([p_1, p_2, \ldots, p_n])$$

where

$$\begin{aligned} \text{error}([p_1, p_2, \ldots, p_n]) &= 1 - \max\{p_1, p_2, \ldots, p_n\} \\ p_i &= \frac{m_i}{\sum_{j=1}^{n} m_j} \quad (\forall i, 1 \leq i \leq n) \end{aligned}$$

Entropy vs Gini vs Error

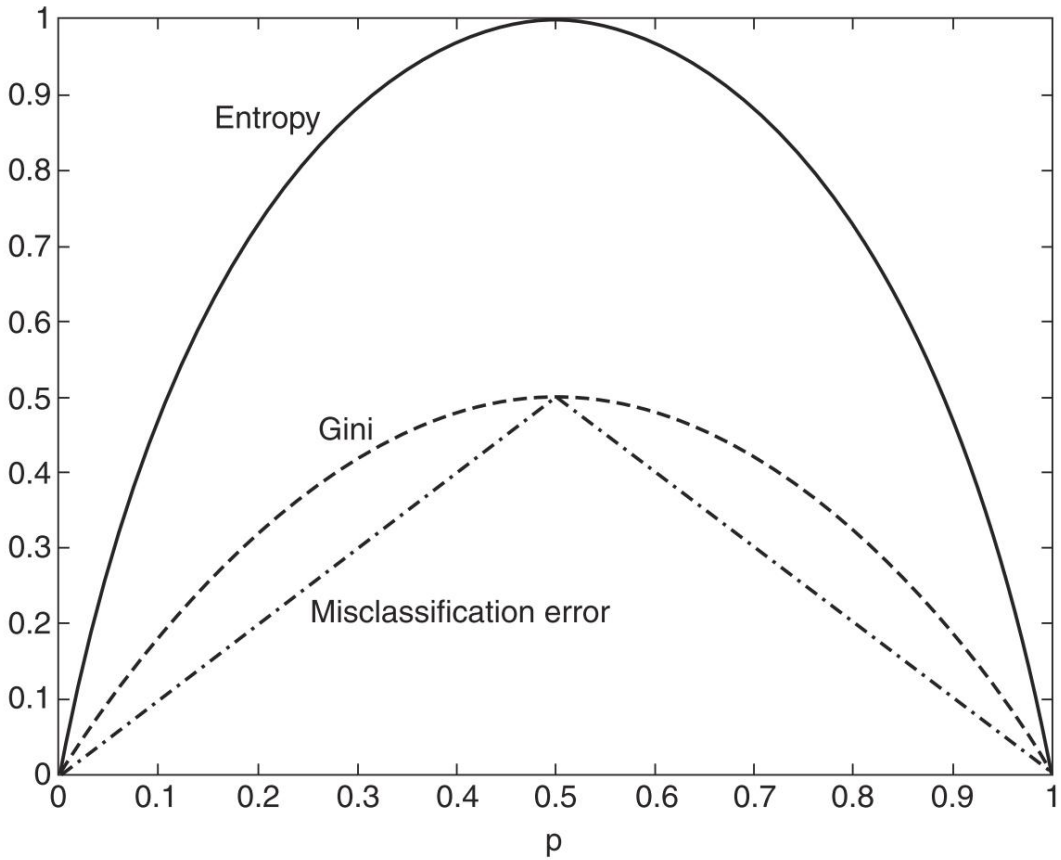


Figure 4.13. Comparison among the impurity measures for binary classification problems.

Source: Pang-Ning Tan, Michael Steinbach, and Vipin Kumar: *Introduction to Data Mining*. Addison Wesley, 2006

ID3 and Key Attributes

Table 4.6 The weather data with identification codes.

ID code	Outlook	Temperature	Humidity	Windy	Play
a	sunny	hot	high	false	no
b	sunny	hot	high	true	no
c	overcast	hot	high	false	yes
d	rainy	mild	high	false	yes
e	rainy	cool	normal	false	yes
f	rainy	cool	normal	true	no
g	overcast	cool	normal	true	yes
h	sunny	mild	high	false	no
i	sunny	cool	normal	false	yes
j	rainy	mild	normal	false	yes
k	sunny	mild	normal	true	yes
l	overcast	mild	high	true	yes
m	overcast	hot	normal	false	yes
n	rainy	mild	high	true	no

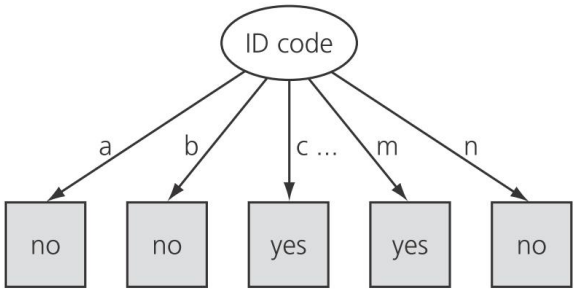


Figure 4.5 Tree stump for the *ID code* attribute.

Source: Ian H. Witten and Eibe Frank: *Data Mining. Practical Machine Learning Tools and Techniques* (2nd Edition). Morgan Kaufmann, 2005

Information Gain Ratio

Idea: penalize attributes with many children.

For an attribute with n children, let $p_1, p_2, \ldots, p_n$ be the fractions of instances arriving at each node.
Then,

$$\text{gain ratio} = \frac{\text{information gain}}{\text{entropy}([p_1, p_2, \ldots, p_n])}$$

Intuition: If $p_1 = p_2 = \cdots = p_n = 1/n$,
then $\text{entropy}([p_1, p_2, \ldots, p_n]) = n * \left(-\frac{1}{n} \log\left(\frac{1}{n}\right)\right) = \log n$.

Recall: Entropy is lower (and hence gain ratio is higher) for unbalanced tests (e.g.,, $\text{entropy}([\frac{1}{2}, \frac{1}{2}]) > \text{entropy}([\frac{8}{9}, \frac{1}{9}])$).

In conclusion, gain ratio favors unbalanced attributes with few values.

Splitting Numeric Attributes

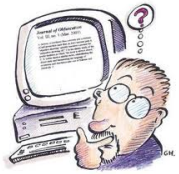
Class	No	No	No	Yes	Yes	Yes	No	No	No	No
	Annual Income									
Sorted Values →	60	70	75	85	90	95	100	120	125	220
Split Positions →	55	65	72	80	87	92	97	110	122	172	230
	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >
Yes	0 3	0 3	0 3	0 3	1 2	2 1	3 0	3 0	3 0	3 0	3 0
No	0 7	1 6	2 5	3 4	3 4	3 4	3 4	4 3	5 2	6 1	7 0
Gini	0.420	0.400	0.375	0.343	0.417	0.400	*0.300*	0.343	0.375	0.400	0.420

Figure 4.16. Splitting continuous attributes.

Source: Pang-Ning Tan, Michael Steinbach, and Vipin Kumar: *Introduction to Data Mining*. Addison Wesley, 2006

Test Options in Weka

- Use training set
- Supplied test set
- Cross-validation (e.g. tenfold)
- Percentage split



Travaux Pratiques

17.1 INTRODUCTION TO THE EXPLORER INTERFACE

- The Classify Pannel (page 562)
  Exercises 17.1.8–17.1.10

Source: Ian H. Witten and Eibe Frank: *Data Mining. Practical Machine Learning Tools and Techniques* (3rd Edition). Morgan Kaufmann, 2011

Outline

Evaluation metrics

		classe prédite: play=yes	classe prédite: play=no
classe observée	play=yes	6 (TP)	3 (FN)
	play=no	2 (FP)	3 (TN)

$$
\begin{aligned}
P &= TP + FN \\
N &= FP + TN \\
TPR &= TP/P \qquad \text{(True-Positive-Rate)} \\
FPR &= FP/N \qquad \text{(False-Positive-Rate)}
\end{aligned}
$$

$$
\begin{aligned}
\text{Recall} &= TP/P \\
\text{Precision} &= TP/(TP + FP) \\
\text{Accuracy} &= (TP + TN)/(P + N)
\end{aligned}
$$

$$
\text{F-measure} = 2 \cdot \frac{\text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}
$$

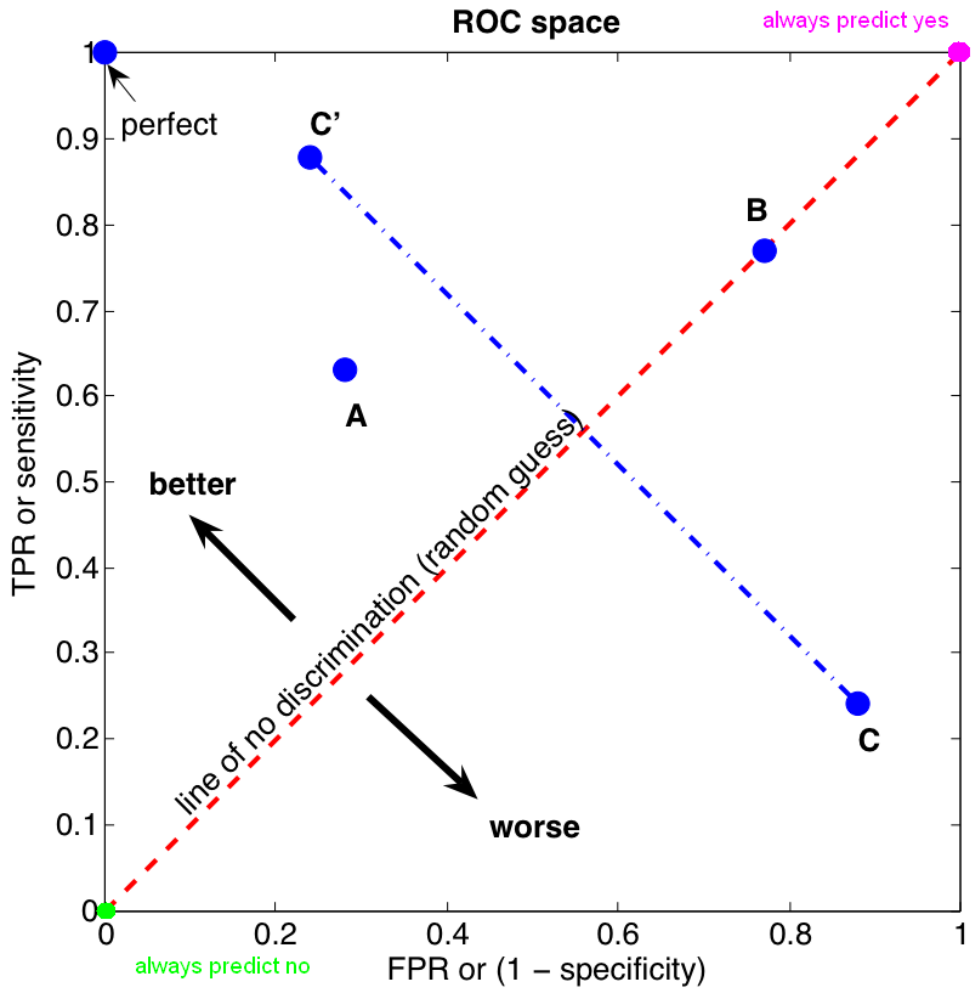
Note: Recall = TPR.

Note harmonic mean: $\frac{1}{\text{F-measure}} + \frac{1}{\text{F-measure}} = \frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}$.

$$\begin{aligned} \text{FPR} &= \frac{\text{FP}}{\text{N}} \\ &= \frac{\text{N} - \text{TN}}{\text{N}} \\ &= 1 - \frac{\text{TN}}{\text{N}} \end{aligned}$$

ROC Space

Each instance of a confusion matrix represents one point in the ROC space.



Source: Wikipedia

ROC AUC (Area Under Curve)

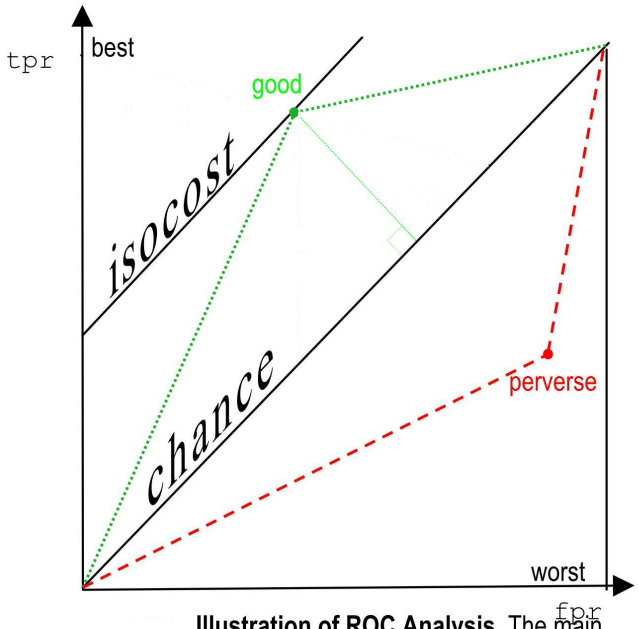


Illustration of ROC Analysis. The main diagonal represents chance with parallel isocost lines representing equal cost-performance. Points above the diagonal represent performance better than chance, those below worse than chance. For a single good (dotted=green) system, AUC is area under curve (trapezoid between green line and x=[0,1]). The perverse (dashed=red) system shown is the same (good) system with class labels reversed.

- $0 \leq \text{AUC} \leq 1$;
- for random guessing, we obtain $\text{AUC} = 0.5$.

Source: D. Powers: *Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation* Journal of Machine Learning Technologies. 2(1), 2011

Étude de cas

J. Dairy Sci. 97:731–742
http://dx.doi.org/10.3168/jds.2013-6693

Prediction of insemination outcomes in Holstein dairy cattle using alternative machine learning algorithms

Saleh Shahinfar,*[1] David Page,† Jerry Guenther,* Victor Cabrera,* Paul Fricke,* and Kent Weigel*
*Department of Dairy Science, and
†Department of Biostatistics and Medical Informatics and Department of Computer Science, University of Wisconsin, Madison 53706

Objective and Outcome

The objective of this study was to compare the performance of different machine learning algorithms for predicting the insemination outcomes of lactating dairy cows using production, reproduction, health, and genetic information. Identification of specific environ-

data sets. Overall, results of this paper suggest that, although prediction of the insemination outcome for individual lactating dairy cows is extremely difficult, information regarding health, reproductive history, production level, and other environmental features can be used to identify highly fertile subsets of cows. Decision support tools developed using this methodology may allow dairy farmers to optimize their breeding programs by targeting animals that are most likely to become pregnant. Such tools could be especially valu-

Methodology

Machine Learning Algorithms

No systematic approach exists that one can use, a priori, to find the most suitable machine learning method for a particular task. Therefore, a common approach in machine learning studies is to test multiple leading algorithms on a new application. In this study, the leading algorithms for learning Bayesian networks and decision trees, including bagging and random forest algorithms that learn ensembles (groups) of trees were tested. The algorithms tested herein are among the most widely used in machine learning today. To classify insemination events into pregnant or nonpregnant outcomes based on the aforementioned explanatory variables, 5 types of machine learning algorithms were used: naïve Bayes, Bayesian networks, decision trees, bagging (ensemble of decision trees), and random forests. A brief explanation of each technique follows.

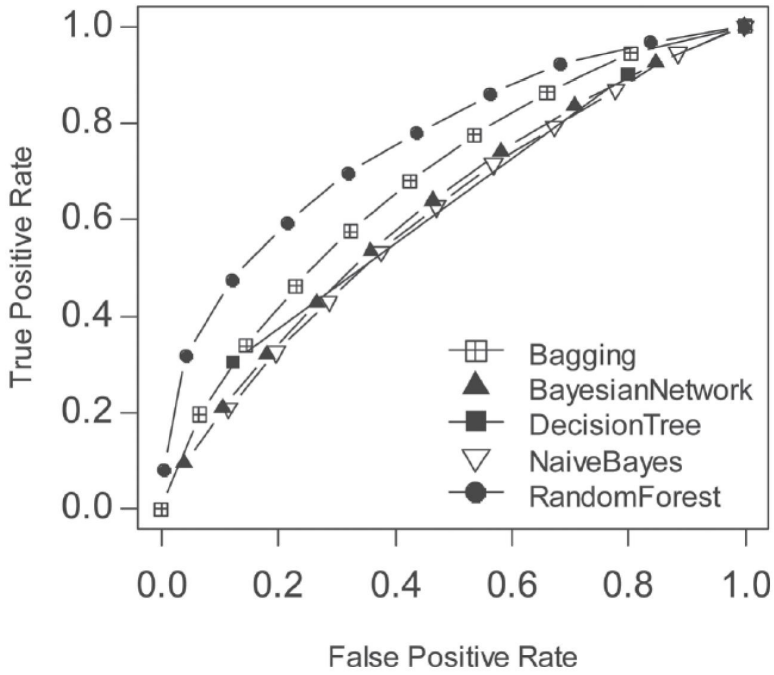


Figure 3. Receiver operating characteristic curves for 5 types of machine learning algorithms used to predict insemination outcomes in primiparous cows.

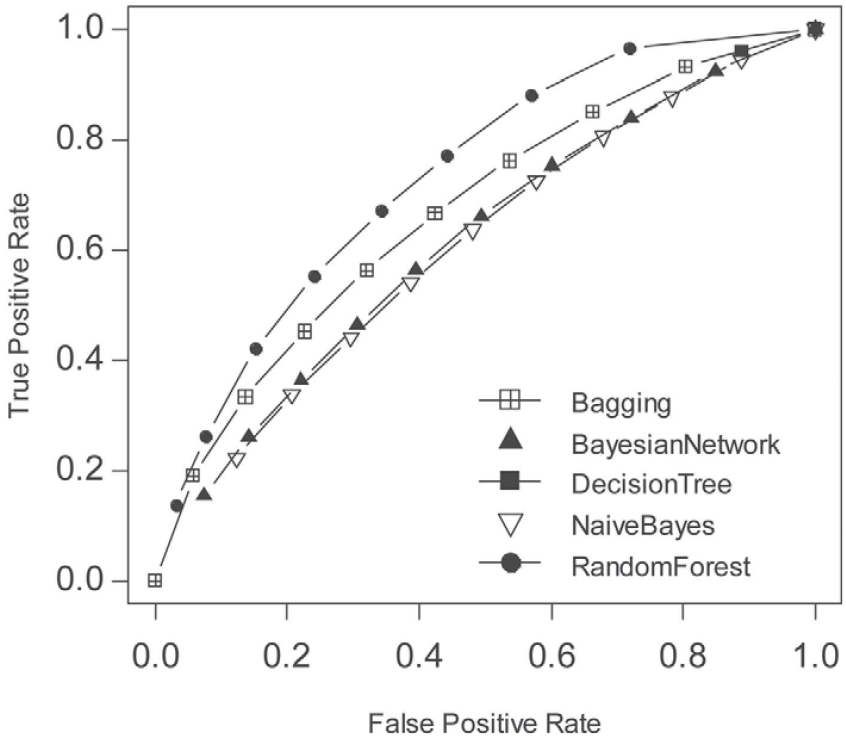


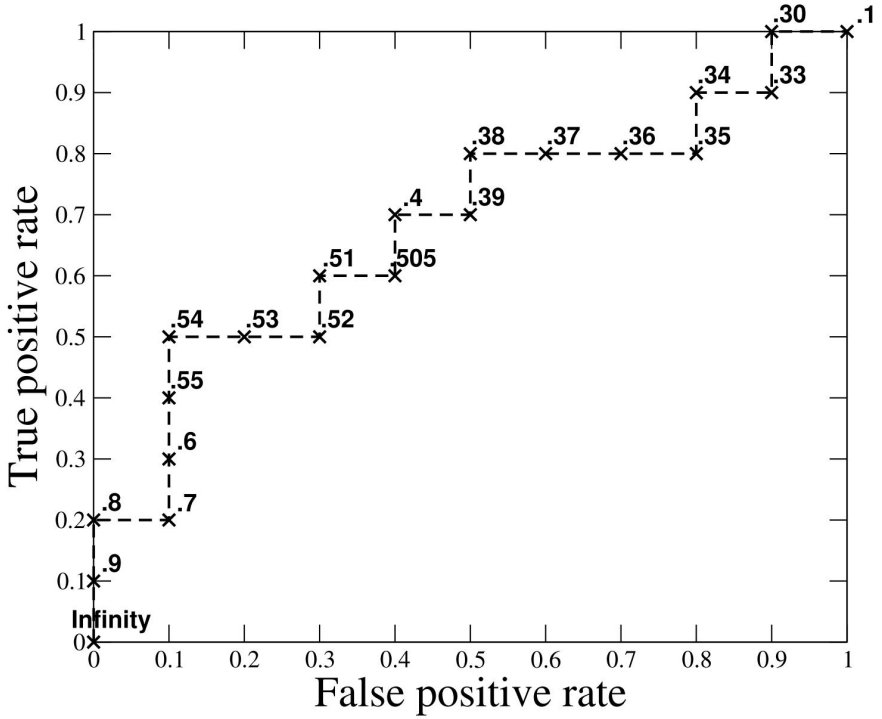
Figure 4. Receiver operating characteristic curves for 5 types of machine learning algorithm used to predict pregnancy outcomes in multiparous cows.

Different Costs of FP and FN

misclassified instances (FP and FN). In our study, the cost of FN instances (failing to inseminate a cow that would have become pregnant) is much greater than the cost of FP instances (inseminating a cow that will not become pregnant). Therefore, FN should be avoided more precisely than FP; in other words operating in

ROC Curve created by thresholding a test set

For classifiers that predict, for each instance, the degree (called *score* or *probability*) to which an instance belongs to the class "yes".



Inst#	Obs.	Score
1	yes	.9
2	yes	.8
3	no	.7
4	yes	.6
5	yes	.55
6	yes	.54
7	no	.53
8	no	.52
9	yes	.51
10	no	.505
11	yes	.4
12	no	.39
13	yes	.38
14	no	.37
15	no	.36
16	no	.35
17	yes	.34
18	no	.33
19	yes	.30
20	no	.1

Some "optimal" scoring

Inst#	Obs.	Score′
6	yes	.95
9	yes	.92
1	yes	.90
2	yes	.88
4	yes	.77
5	yes	.66
11	yes	.33
13	yes	.22
17	yes	.19
19	yes	.18
3	no	.17
14	no	.16
15	no	.15
16	no	.14
7	no	.13
8	no	.12
10	no	.11
12	no	.10
18	no	.09
20	no	.08

Inst# is an identifier for each instance;

Obs. is the observed class; *Score* is the predicted score.

Source: T. Fawcett: *ROC Graphs: Notes and Practical Considerations for Data Mining Researchers* Technical Report. HP, 2003

ROC Curve computation

Let threshold value be $(0.51 + 0.52)/2$:

Class=**yes** *if* **Score** $> (0.51 + 0.52)/2$;
Class=**no** *if* **Score** $< (0.51 + 0.52)/2$.

This results in TP $= 5$, FP $= 3$, TN $= 7$, FN $= 5$.

$$\begin{aligned} \text{P} &= 10 \\ \text{N} &= 10 \\ \text{TPR} &= 5/10 \\ \text{FPR} &= 3/10 \\ \text{Accuracy} &= 12/20 \end{aligned}$$

Note: the highest accuracy (14/20) is obtained for threshold value $(0.53 + 0.54)/2$.

Comparison Tasks

- Classifier M_A correctly classifies 24 of 32 records (accuracy = 0.75).
- Classifier M_B correctly classifies 3500 of 5000 records (accuracy = 0.70).

Two tasks:

1. Estimating a confidence interval for accuracy.
2. Comparing the performance of two classifiers.

Processus de Bernoulli

Prosaïquement, un processus de Bernoulli consiste à tirer à pile ou face (*success* ou *failure*) plusieurs fois (*n* fois) de suite, éventuellement avec une pièce truquée (probabilité *p*, éventuellement $p \neq 0.5$).

Un example est la classification :

- prédire la classe pour *n* nouvelles instances;
-
  - *success* = prédiction correcte (TP ou TN);
  - *failure* = prédiction incorrecte (FP ou FN);
- *p* est la "vraie" exactitude du classificateur (*accuracy*).

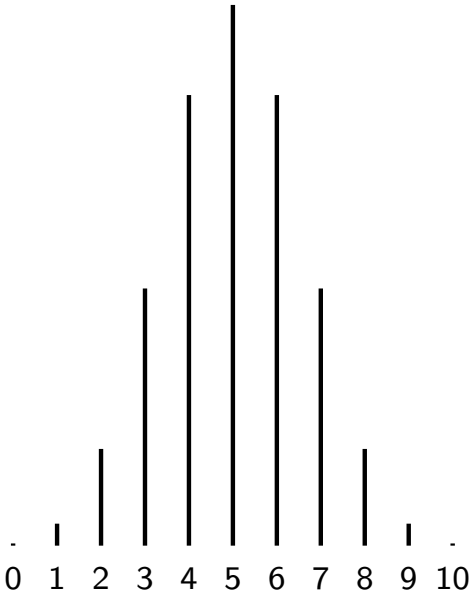
Notes:
Le nombre de classes possibles peut être supérieur à 2.
Ne pas confondre success/failure avec des classes yes/no!

Estimating a Confidence Interval for Accuracy

- Classifying n records $\rightsquigarrow$ n independent success/failure experiments, each of which yields success with probability p.
- The probability of exactly k successes is: $\binom{n}{k} p^k (1-p)^{n-k}$
- For example, flipping a coin 10 times and counting the number of heads. The probability of exactly 2 heads is: $\binom{10}{2} (0.5)^2 (0.5)^8$

Binomial Distribution $B(n, p)$

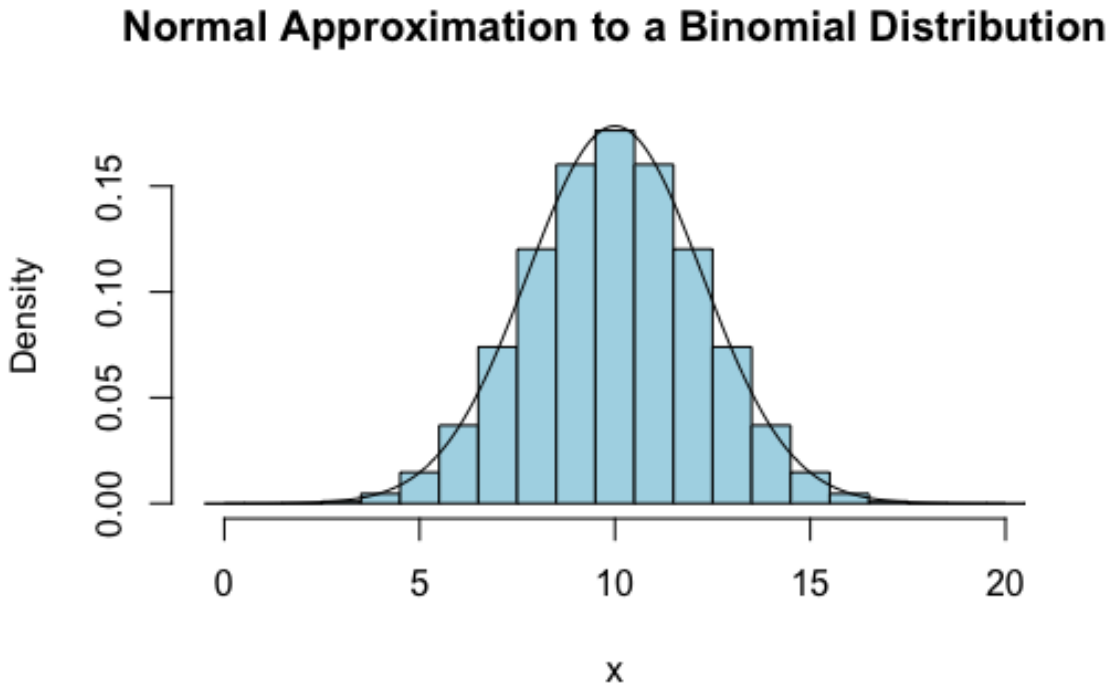
Probability mass function
($n = 10$, $p = 0.5$)



$$\begin{aligned} \text{Mean} &= np \\ \text{Variance} &= np(1-p) \end{aligned}$$

- For large[1] n, an approximation to $B(n, p)$ is given by $N(np, np(1-p))$.
- $N(\mu, \sigma^2)$ is the normal distribution with mean μ and variance σ^2.
- Recall: B is discrete; N is continuous.

[1] A commonly used rule is that both $np \geq 5$ and $n(1-p) \geq 5$.

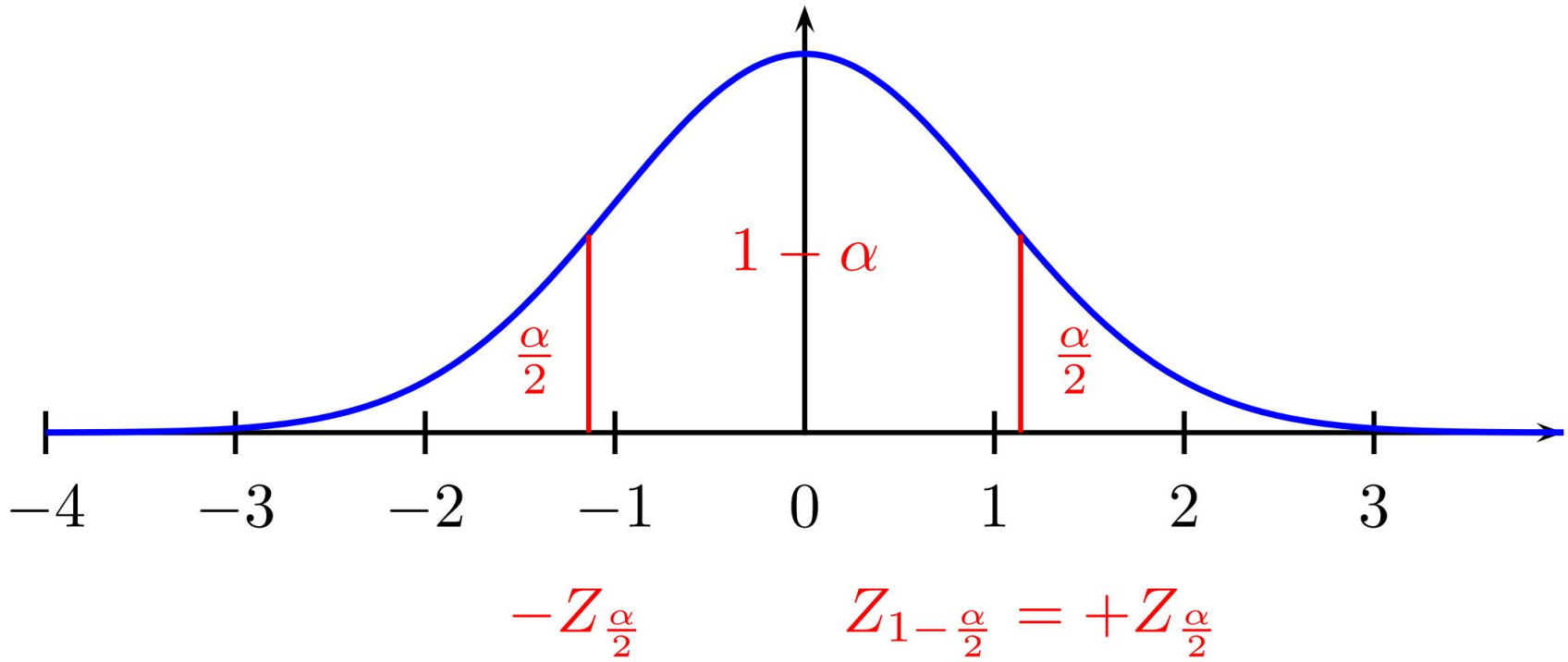


Source: `http://mathcenter.oxford.emory.edu/`

Try it yourself at:

`https://shiny.rit.albany.edu/stat/binomial/`

$N(0,1)$



$1-\alpha$	0.99	0.95	0.9	0.7	0.5
$+Z_{\frac{\alpha}{2}}$	2.58	1.96	1.65	1.04	0.68

- We observe that X instances out of n are correctly classified. (Note: there can be two or more classes.)
- $X \sim N(np, np(1-p))$, where p is the real accuracy.
- Thus, $\frac{X-np}{\sqrt{np(1-p)}} \sim N(0,1)$.
- Fix α.
- $Pr(-Z_{\frac{\alpha}{2}} \leq \frac{X-np}{\sqrt{np(1-p)}} \leq +Z_{\frac{\alpha}{2}}) = 1-\alpha$
- We obtain:

$$p_{1,2} = \frac{2X + (Z_{\frac{\alpha}{2}})^2 \pm Z_{\frac{\alpha}{2}}\sqrt{(Z_{\frac{\alpha}{2}})^2 + 4X - \frac{4X^2}{n}}}{2(n + (Z_{\frac{\alpha}{2}})^2)}$$

- For example, $n = 100$, $X = 80$ (accuracy $= 0.80$)
  $\rightsquigarrow$ if $1-\alpha = 0.95$ then $p_1 = 0.711$ and $p_2 = 0.867$.

Comparing Performance

- We observe that M_A wrongly classifies a fraction e_1 of n_1 instances.
  We observe that M_B wrongly classifies a fraction e_2 of n_2 instances.
- For large n_1 and n_2, the observed difference $d = e_1 - e_2$ in the error rate is normally distributed with mean d_t and variance[2]
  $\sigma^2 = \frac{e_1(1-e_1)}{n_1} + \frac{e_2(1-e_2)}{n_2}$.
- $\frac{d-d_t}{\sigma} \sim N(0,1)$
- Fix α.
- $Pr(-Z_{\frac{\alpha}{2}} \leq \frac{d-d_t}{\sigma} \leq +Z_{\frac{\alpha}{2}}) = 1 - \alpha$
- We obtain:

$$d_t = d \pm Z_{\frac{\alpha}{2}} \sigma$$

Note: The subscript t in d_t stands for "true."

[2]For independent random variables, the variance of their difference is the sum of their variances. The variance of e_1 is $\frac{e_1(1-e_1)}{n_1}$.

Example: Not Significant

- 8 records out of 32 are wrongly classified $\rightsquigarrow n_1 = 32$, $e_1 = 0.25$. 1500 records out of 5000 are wrongly classified $\rightsquigarrow n_2 = 5000$, $e_2 = 0.30$.
- $\sigma = \sqrt{\frac{e_1(1-e_1)}{n_1} + \frac{e_2(1-e_2)}{n_2}} = 0.0768$
- $d = 0.25 - 0.30 = -0.05$
- For $1 - \alpha = 0.95$,

$$\begin{aligned} d_t &= -0.05 \pm 1.96 \times 0.0768 \\ &= -0.05 \pm 0.151 \end{aligned}$$

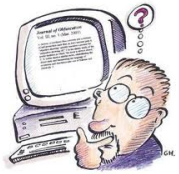
- Since the interval includes zero, the observed difference is not statistically significant at a 0.95 confidence level.

Example: Significant

- 2000 records out of 5000 are wrongly classified $\rightsquigarrow n_1 = 5000$, $e_1 = 0.40$.
  2 records out of 100 are wrongly classified $\rightsquigarrow n_2 = 100$, $e_2 = 0.02$.
- $\sigma = \sqrt{\frac{e_1(1-e_1)}{n_1} + \frac{e_2(1-e_2)}{n_2}} = 0.156$
- $d = 0.40 - 0.02 = 0.38$
- For $1 - \alpha = 0.95$,

$$\begin{aligned} d_t &= 0.38 \pm 1.96 \times 0.156 \\ &= 0.38 \pm 0.03 \end{aligned}$$

- Since the interval does not include zero, the observed difference is statistically significant at a 0.95 confidence level.



Exercice

Pour le même jeu de données, deux algorithmes, A and B, donnent les matrices de confusion suivantes.

A	Y	N
Y	12	4
N	3	13

B	Y	N
Y	13	3
N	2	14

Peut-on conclure que B est meilleur ?

Outline

Labor Data

Table 1.6 The labor negotiations data.

Attribute	Type	1	2	3	...	40
duration	years	1	2	3		2
wage increase 1st year	percentage	2%	4%	4.3%		4.5
wage increase 2nd year	percentage	?	5%	4.4%		4.0
wage increase 3rd year	percentage	?	?	?		?
cost of living adjustment	{none, tcf, tc}	none	tcf	?		none
working hours per week	hours	28	35	38		40
pension	{none, ret-allw, empl-cntr}	none	?	?		?
standby pay	percentage	?	13%	?		?
shift-work supplement	percentage	?	5%	4%		4
education allowance	{yes, no}	yes	?	?		?
statutory holidays	days	11	15	12		12
vacation	{below-avg, avg, gen}	avg	gen	gen		avg
long-term disability assistance	{yes, no}	no	?	?		yes
dental plan contribution	{none, half, full}	none	?	full		full
bereavement assistance	{yes, no}	no	?	?		yes
health plan contribution	{none, half, full}	none	?	full		half
acceptability of contract	{good, bad}	bad	good	good		good

Source: Ian H. Witten and Eibe Frank: *Data Mining. Practical Machine Learning Tools and Techniques* (2nd Edition). Morgan Kaufmann, 2005

Pruning

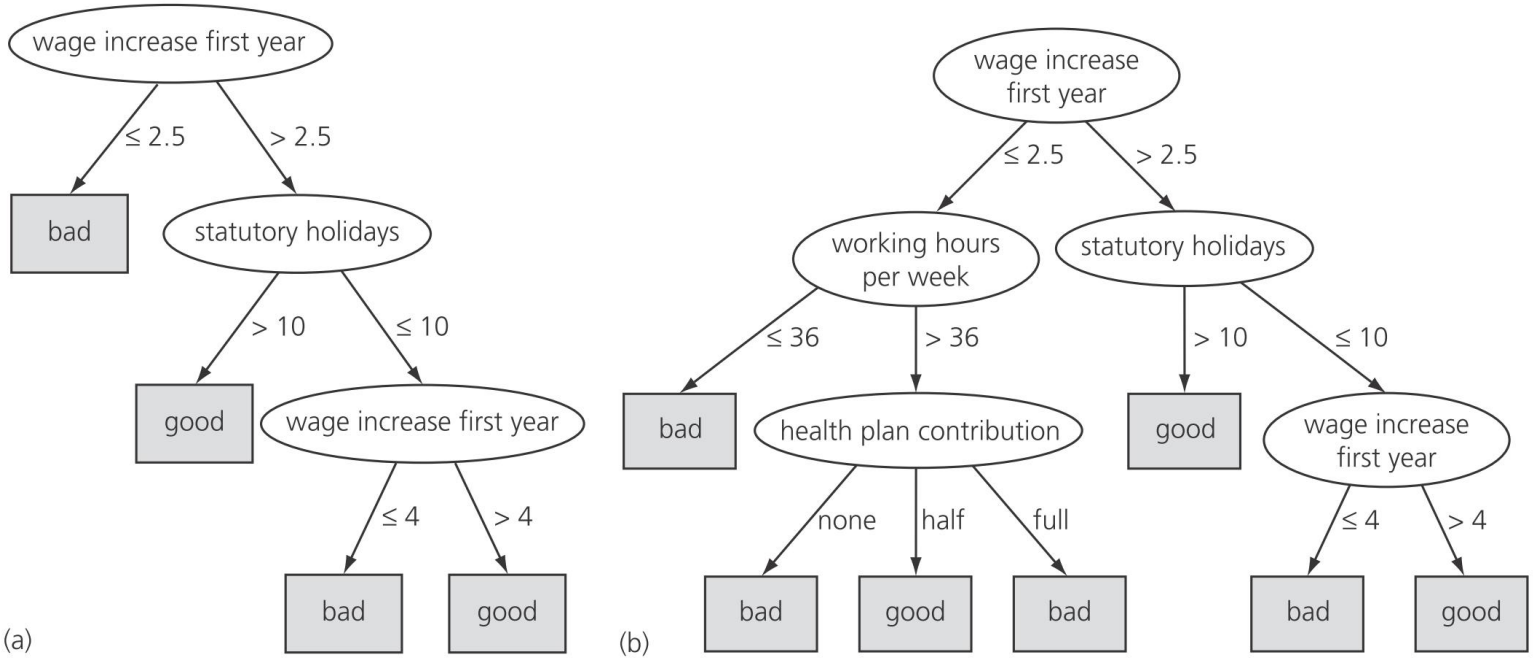


Figure 1.3 Decision trees for the labor negotiations data.

Source: Ian H. Witten and Eibe Frank: *Data Mining. Practical Machine Learning Tools and Techniques* (2nd Edition). Morgan Kaufmann, 2005

Pruning in C4.5

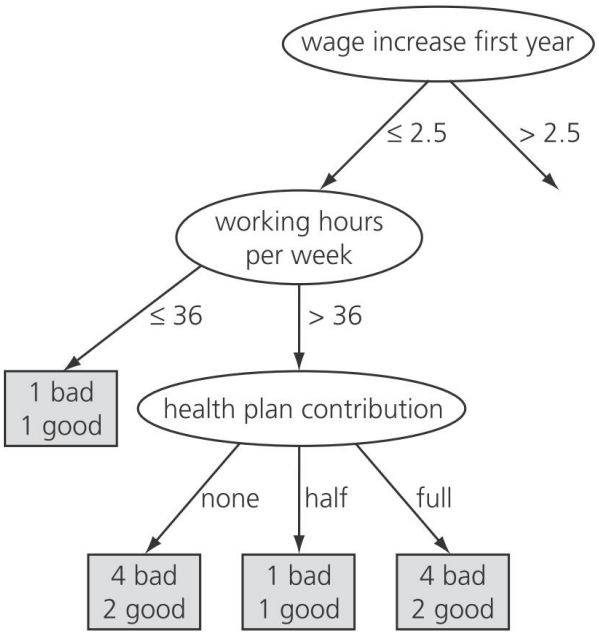


Figure 6.2 Pruning the labor negotiations decision tree.

$$1 - \alpha = 0.5, z = 0.68$$

n	X	p_1	p_2
6	4	0.528	0.781
2	1	0.283	0.717
14	9	0.552	0.724

Pessimistic precision estimate of "health plan contribution" is:

$$\frac{6}{14} \times 0.528 + \frac{2}{14} \times 0.283 + \frac{6}{14} \times 0.528 = 0.493$$

Since $0.493 < 0.552$, we prune away "health plan contribution".

Source: Ian H. Witten and Eibe Frank: *Data Mining. Practical Machine Learning Tools and Techniques* (2nd Edition). Morgan Kaufmann, 2005

Outline

Outline

Le principe

In distance-based learning, each new instance is compared with existing ones using a distance metric, and the closest existing instance is used to assign the class to the new one :

- nearest-neighbor classification, and
- k-nearest-neighbor classification.

Instance-based learning is lazy, deferring the real work as long as possible.

La distance entre deux instances

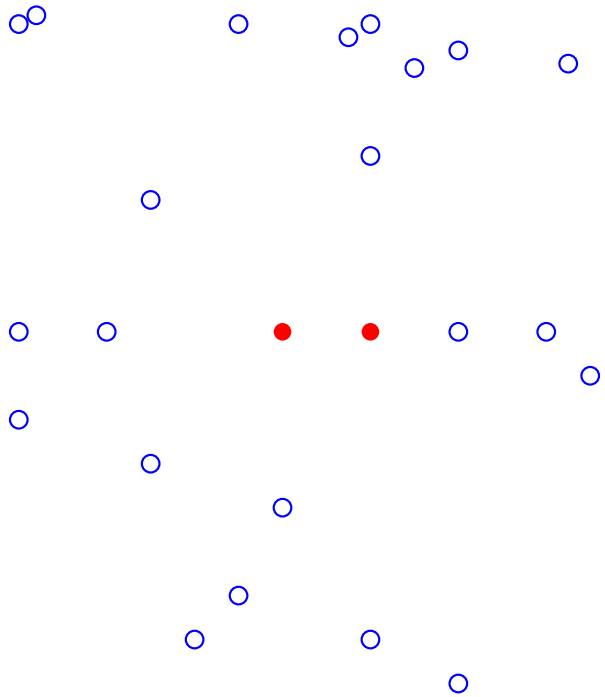
Computing the distance between two examples :

- One numeric attribute: difference.
- Several numeric attributes: normalization + Euclidean distance.
- Nominal attributes: distance of 0 if the values are identical, otherwise the distance is 1. However, it may be desirable to use a more sophisticated representation of the attributes.

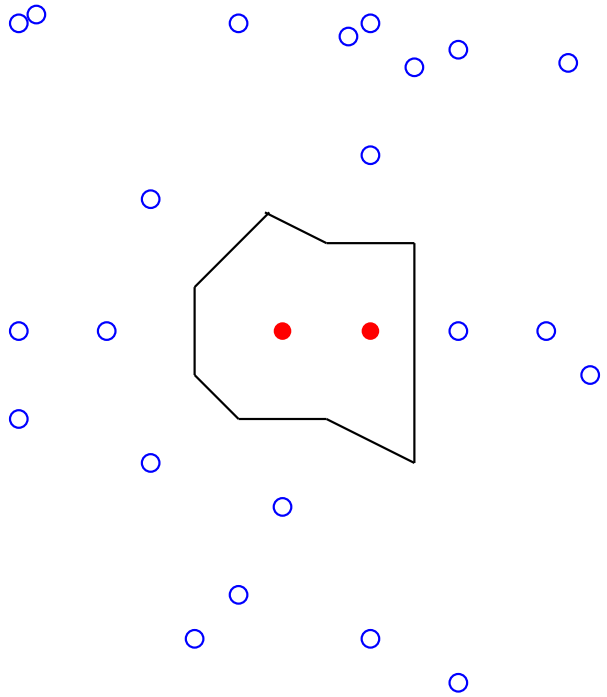
Some attributes will be more important than others, and this is usually reflected in the distance metric by some kind of attribute weighting. Deriving attribute weights from the training set is a key problem in instance-based learning.

Exemple en $\mathbb{R}^2$

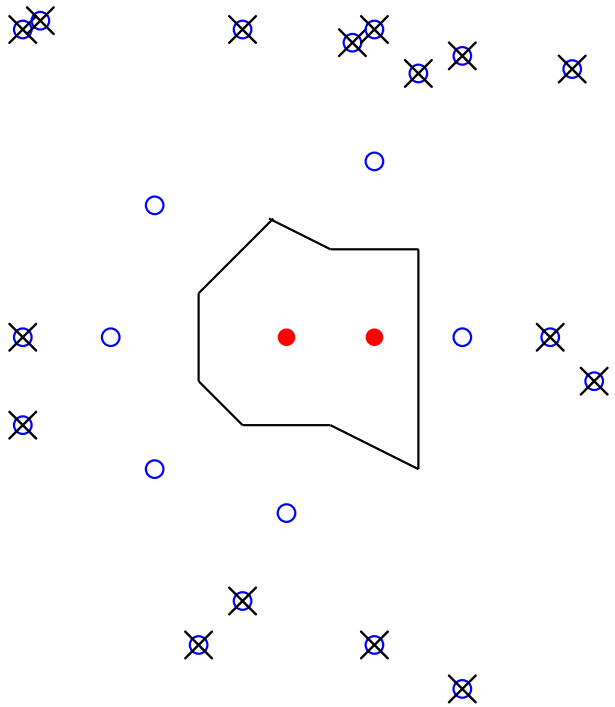
Filled-circle class and open-circle class.



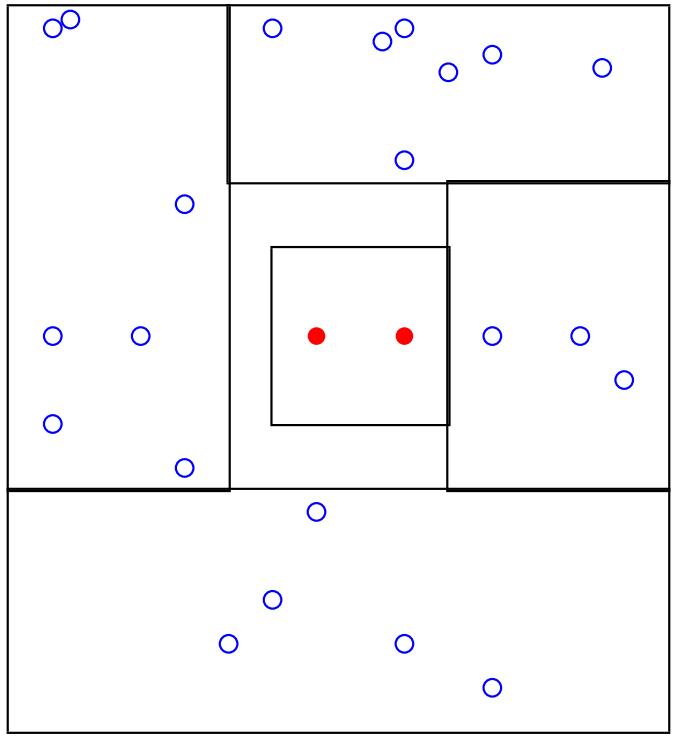
Eight-sided polygon separates the filled-circle class from the open-circle-class. This polygon is implicit in the operation of the nearest-neighbor rule.

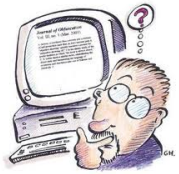


Save just a few prototypical examples of each class—the ones that actually get used in the nearest-neighbor decisions.



Creating rectangular regions that enclose examples of the same class. Examples that fall outside all rectangles will be subject to the usual nearest-neighbor rule.



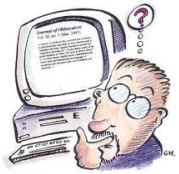


Travaux Pratiques

17.2 NEAREST-NEIGHBOR LEARNING AND DECISION TREES

- Exercises 17.2.1–17.2.3

Source: Ian H. Witten and Eibe Frank: *Data Mining. Practical Machine Learning Tools and Techniques* (3rd Edition). Morgan Kaufmann, 2011



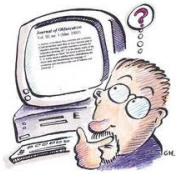
Travaux Pratiques

In problems where the goal is to learn from examples, high dimensionality is a big problem.

17.2 NEAREST-NEIGHBOR LEARNING AND DECISION TREES

- Attribute Selection
  Exercises 17.2.4–17.2.5

Source: Ian H. Witten and Eibe Frank: *Data Mining. Practical Machine Learning Tools and Techniques* (3rd Edition). Morgan Kaufmann, 2011

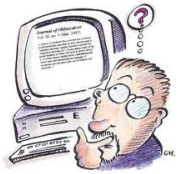


Travaux Pratiques

17.2 NEAREST-NEIGHBOR LEARNING AND DECISION TREES

- Class Noise and Nearest-Neighbor Learning
  Exercises 17.2.6–17.2.8

  Source: Ian H. Witten and Eibe Frank: *Data Mining. Practical Machine Learning Tools and Techniques* (3rd Edition). Morgan Kaufmann, 2011

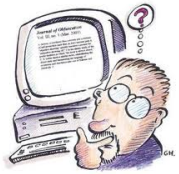


Travaux Pratiques

17.2 NEAREST-NEIGHBOR LEARNING AND DECISION TREES

- Varying the Amount of Training Data
  Exercises 17.2.9–17.2.11

Source: Ian H. Witten and Eibe Frank: *Data Mining. Practical Machine Learning Tools and Techniques* (3rd Edition). Morgan Kaufmann, 2011



Travaux Pratiques

17.2 NEAREST-NEIGHBOR LEARNING AND DECISION TREES

- Interactive Decision Tree Construction
  Exercise 17.2.12

Source: Ian H. Witten and Eibe Frank: *Data Mining. Practical Machine Learning Tools and Techniques* (3rd Edition). Morgan Kaufmann, 2011

Outline

Qu'est-ce que les règles de classification ?

Antecedent or precondition. Series of tests, usually ANDed together.

Consequent or conclusion. Class(es).

Set of rules. Usually ORed together.

For example,

```
If outlook=sunny and humidity=high then play=no
If outlook=sunny and humidity=normal then play=yes
If outlook=overcast then play=yes
If outlook=rainy and windy=false then play=yes
If outlook=rainy and windy=true then play=no
```

or,

```
if outlook = sunny and humidity = high then play = no
elsif outlook = rainy and windy = true then play = no
elsif outlook = overcast then play = yes
elsif humidity = normal then play = yes
else play = yes
```

1R: `classifiers.rules.OneR`

Supposons C est l'attribut cible. Pour tout attribut A avec comme valeurs possibles $v_1, v_2, \ldots, v_n$, créer un classificateur :

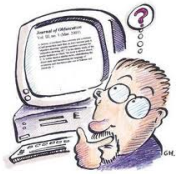
$$\begin{aligned} &\texttt{if } A = v_1\texttt{, alors } C = c_{i_1} \\ &\texttt{if } A = v_2\texttt{, alors } C = c_{i_2} \\ &\qquad\qquad \vdots \\ &\texttt{if } A = v_n\texttt{, alors } C = c_{i_n} \end{aligned}$$

où $c_{i_1}, c_{i_2}, \ldots, c_{i_n}$ sont les valeurs pour C qui "collent" le mieux.
Finalement, choisir le classificateur qui minimise le taux d'erreur.

Données numériques

To avoid overfitting, when discretizing a numeric attribute, a minimum limit is imposed on the number of examples of the majority class in each partition.

64	65	68	69	70	71	72	72	75	75	80	81	83	85	*minBucketSize*
yes	no	yes	yes	yes	no	no	yes	yes	yes	no	yes	yes	no	1
yes	no	yes	yes	yes	no	no	yes	yes	yes	no	yes	yes	no	2

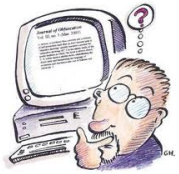


Travaux Pratiques

17.3 CLASSIFICATION BOUNDARIES

- Exercise 17.3.1–17.3.4

Source: Ian H. Witten and Eibe Frank: *Data Mining. Practical Machine Learning Tools and Techniques* (3rd Edition). Morgan Kaufmann, 2011



Travaux Pratiques

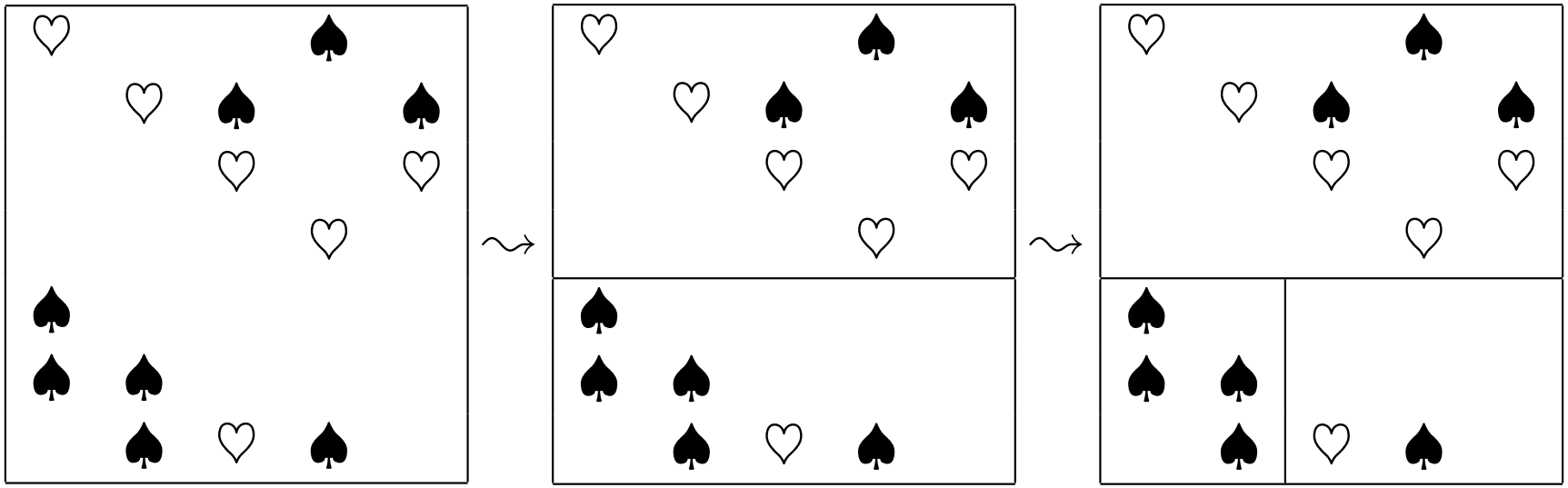
17.3 CLASSIFICATION BOUNDARIES

- Exercise 17.3.5–17.3.6

Source: Ian H. Witten and Eibe Frank: *Data Mining. Practical Machine Learning Tools and Techniques* (3rd Edition). Morgan Kaufmann, 2011

Covering Algorithm: `classifiers.rules.Prism`

Take each class in turn and seek a way of covering all instances in it, at the same time excluding instances not in the class.



We use the weather data, and form rules that cover each of the two classes `yes` and `no` in turn. To begin, we seek a rule:

```
If ?  then play = no
```

For the unknown term ?, we have several choices:

Condition	Accuracy
`outlook = sunny`	3/5
`outlook = overcast`	0/4
`outlook = rainy`	2/5
`temperature = hot`	2/4
`temperature = mild`	2/6
`temperature = cool`	1/4

Condition	Accuracy
`humidity = high`	4/7
`humidity = normal`	1/7
`windy = false`	2/8
`windy = true`	3/6

We select the largest fraction, 3/5, and create the rule:

```
If outlook = sunny then play = no
```

Weather date for which `outlook = sunny`:

outlook	temperature	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
sunny	mild	high	false	no
sunny	cool	normal	false	yes
sunny	mild	normal	true	yes

Should we stop here? Perhaps. But let's say we are going for exact rules. So we refine the rule further, to:

```
If outlook = sunny and ?  then play = no
```

For the unknown term ?, we have several choices:

Condition	Accuracy
`temperature = hot`	2/2
`temperature = mild`	1/2
`temperature = cool`	0/1
`humidity = high`	3/3
`humidity = normal`	0/2
`windy = false`	2/3
`windy = true`	1/2

We need to choose between `temperature = hot` and `humidity = high`. We choose `humidity = high` because it has the greater coverage:

```
If outlook = sunny and humidity = high then play = no
```

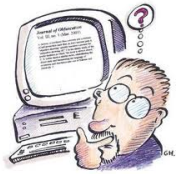
This rule is 100% accurate, but only covers three out of the five `no` instances. So we delete these three from the set of instances and start again, looking for another rule of the form:

```
If ?  then play = no
```

Along the same lines, we obtain a second rule:

```
If windy = true and outlook = rainy then play = no
```

This rules covers the two `no` instances not covered by the previous rule; its accuracy is 100%. Now that all the `no` cases are covered, the next step is to proceed with the `yes` ones in just the same way.



Exercice

1. L'ensemble de règles créé par Prism, est-il toujours *sound* et *complete* ?

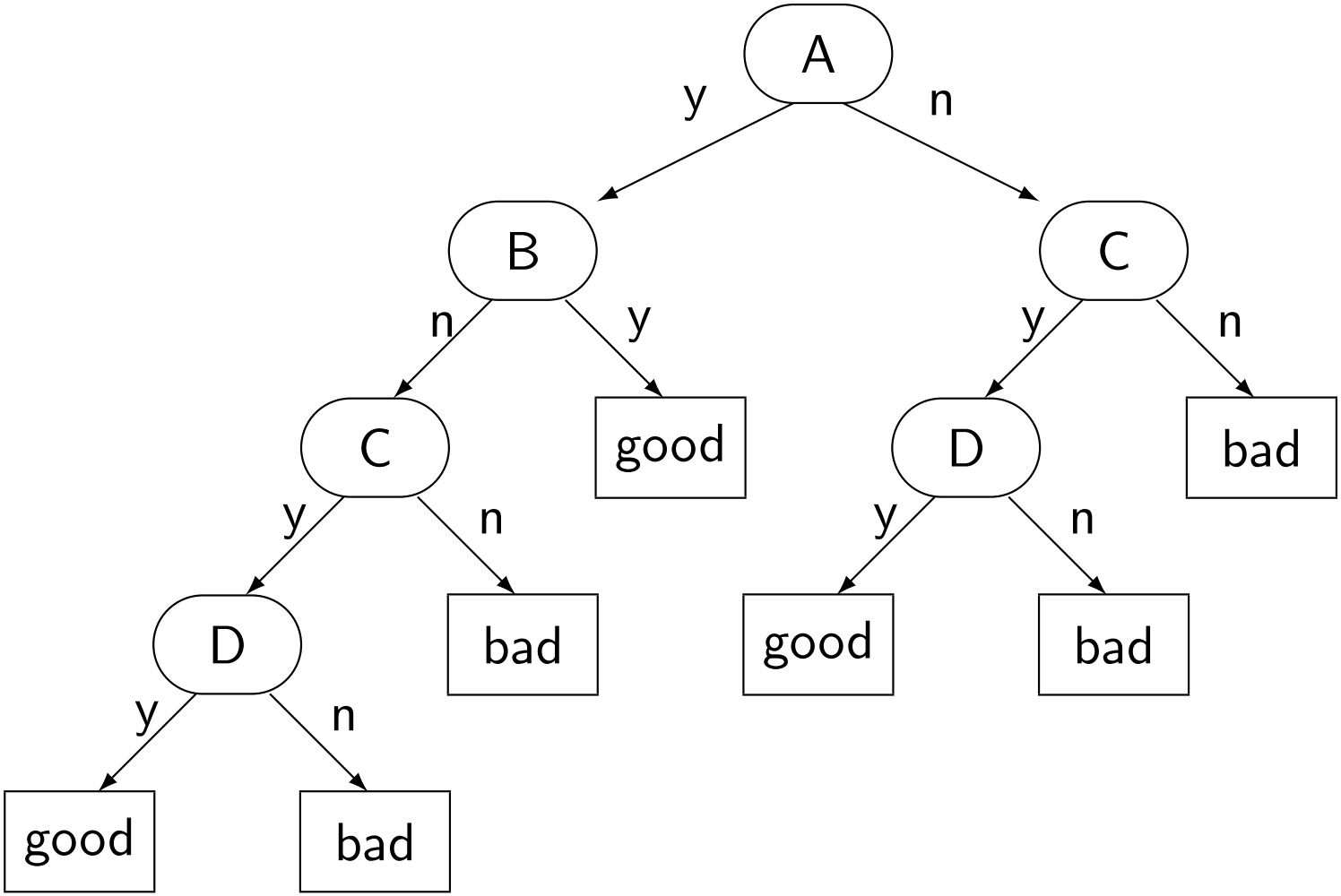
Outline

Replicated Subtree Problem

```
If A=y and B=y then good
If C=y and D=y then good
```

Rules that are read directly of a decision tree are far more complex than necessary:

```
If A=y and B=n and C=y and D=y then good
If A=y and B=n and C=y and D=n then bad
...
```


A
y
n
B
C
n
y
y
n
C
good
D
bad
y
n
y
n
D
bad
good
bad
y
n
good
bad

Rules=Independent Nuggets of Knowledge!?

Caution:

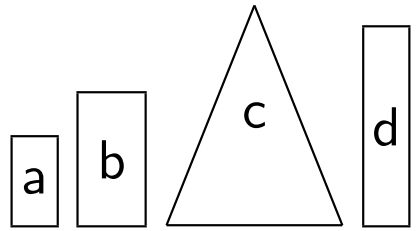
- Often decision lists or preferences among conflicting rules. For example, the following rules are meant to be interpreted in order:

```
If outlook=sunny and humidity=high then play=no
If outlook=rainy and windy=true then play=no
If outlook=overcast then play=yes
If humidity=normal then play=yes
If none of the above then play=yes
```

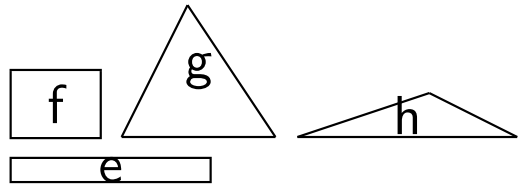
- Soundness and completeness is not guaranteed (unless rules are directly read of decision tree).

Note concernant la puissance d'expression

ID	Width	Height	Sides	Class
a	2	4	4	standing
b	3	6	4	standing
c	8	10	3	standing
d	2	9	4	standing
e	9	1	4	lying
f	4	3	4	lying
g	7	6	3	lying
h	10	2	3	lying



Standing



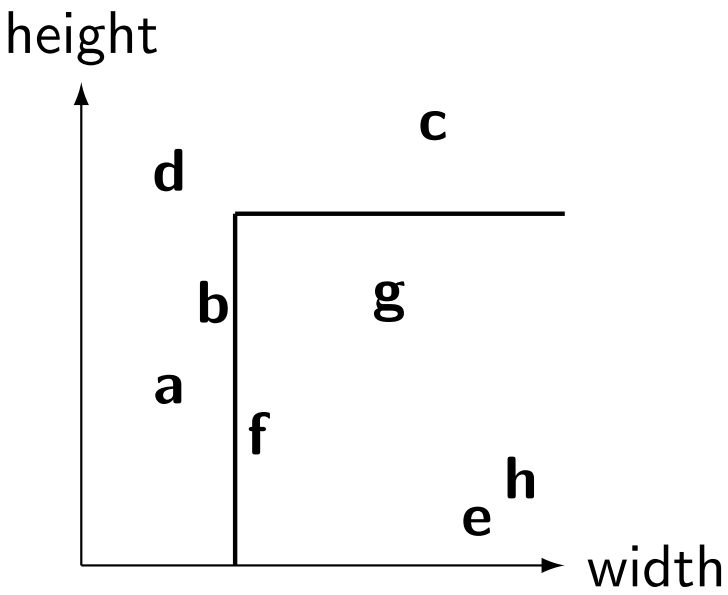
Lying

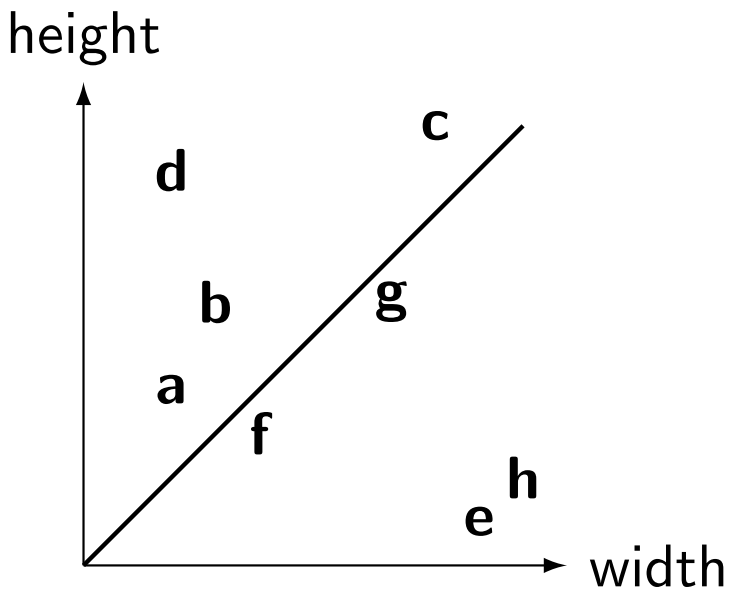
Les systèmes de classification classiques comparent *des attributs avec des constantes :*

```
If width > 3.5 and height
< 8 then lying
```

La règle suivante compare *les attributs entre-eux :*

```
If width > height then
lying
```





Outline

Probabilité conditionnelle

$$Pr(H \mid E) = \frac{Pr(H \wedge E)}{Pr(E)}$$

- Par exemple, 100 personnes dont 40 Etudiants; 10 des 40 étudiants sont des Hommes. La probabilité $Pr(E)$ qu'une personne soit étudiant est de $40/100$. La probabilité $Pr(H \wedge E)$ qu'une personne soit un étudiant masculin est de $10/100$. La probalité $Pr(H \mid E)$ qu'un étudiant soit masculin est de $10/40 = \frac{10/100}{40/100}$.
- Qu'est-ce que $Pr(E \mid H)$?

classifiers.bayes.NaiveBayesSimple

Voir transparent 4. Utilisons les abbréviations suivantes :

E_1	outlook=sunny
E_2	temperature=cool
E_3	humidity=high
E_4	wind=true
E	$E_1 \& E_2 \& E_3 \& E_4$
H_n	play=no
H_y	play=yes

On calcule $Pr(H_n \mid E)$ et $Pr(H_y \mid E)$. Si $Pr(H_n \mid E) > Pr(H_y \mid E)$, on prédit play=no; sinon play=yes.

$$Pr(H_n \mid E)$$

$$= \frac{Pr(E \mid H_n)Pr(H_n)}{Pr(E)} = \frac{Pr(E_1 \& E_2 \& E_3 \& E_4 \mid H_n) \times Pr(H_n)}{Pr(E)}$$

$$\approx \frac{Pr(E_1 \mid H_n) \times Pr(E_2 \mid H_n) \times Pr(E_3 \mid H_n) \times Pr(E_4 \mid H_n) \times Pr(H_n)}{Pr(E)}$$

$$= \frac{\frac{3}{5} \times \frac{1}{5} \times \frac{4}{5} \times \frac{3}{5} \times \frac{5}{14}}{Pr(E)} = 0.0206/Pr(E)$$

De même manière, $Pr(H_y \mid E) \approx 0.0053/Pr(E)$. Donc,

$$\frac{Pr(H_n \mid E)}{Pr(H_y \mid E)} = \frac{0.0206}{0.0053} \ .$$

En plus, $Pr(H_n \mid E) + Pr(H_y \mid E) = 1$.
Dès lors, $Pr(H_n \mid E) = 0.795$ et $Pr(H_y \mid E) = 0.205$.

On utilise la règle de Bayes :

$$Pr(H \mid E) = \frac{Pr(E \mid H) \times Pr(H)}{Pr(E)}$$

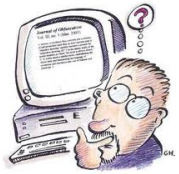
La naïveté se trouve en :

$$Pr(E_1 \& E_2 \mid H) \approx Pr(E_1 \mid H) \times Pr(E_2 \mid H)$$

On suppose que E_1 et E_2 soient indépendants (pourvu que H).
Prenons, par ex., $E_1 = E_2 = E$.

$$Pr(E \mid H) = Pr(E \& E \mid H) \approx Pr(E \mid H) \times Pr(E \mid H)$$

$$Pr(E \mid H) \overset{???}{\approx} Pr(E \mid H) \times Pr(E \mid H)$$



Exercice

1. Construire un fichier avec trois attributs (dont un attribut cible) où l'hypothèse $Pr(E_1 \& E_2 \mid H) \approx Pr(E_1 \mid H) \times Pr(E_2 \mid H)$ n'est manifestement pas justifiée.
2. Quelle des deux méthodes, ID3 ou Naive Bayes, est la plus biaisée par des corrélations entre colonnes ?

Données numériques

outlook	temperature	humidity	windy	play
...	...	...	...	no
...	...	...	...	no
...	27	...	...	yes
...	23	...	...	yes
...	15	...	...	yes
...	...	...	...	no
...	14	...	...	yes
...	...	...	...	no
...	15	...	...	yes
...	22	...	...	yes
...	23	...	...	yes
...	24	...	...	yes
...	28	...	...	yes
...	...	...	...	no

Qu'est-ce que $Pr(\text{temperature}=21 \mid \text{play}=\text{yes})$?

Moyenne et variance

$S = \{27, 23, 15, 14, 15, 22, 23, 24, 28\}$

$$\textit{sample mean} \quad \mu_S = \frac{1}{n}\sum_{i=1}^{n} x_i = 21.222$$

$$\textit{sample variance} \quad \sigma_S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu_S)^2 = 27.944$$

Distribution normale

Distribution normale avec moyenne μ et écart type σ :

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

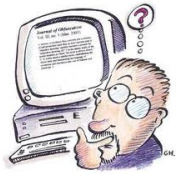
La probabilité qu'une valeur se trouve dans l'interval $[a, b]$:

$$\int_a^b f(x)dx$$

On a $\int_{-\infty}^{+\infty} f(x)dx = 1$ $\int_{\mu-\sigma}^{\mu+\sigma} f(x)dx \approx 0.68$ $\int_{\mu-2\sigma}^{\mu+2\sigma} f(x)dx \approx 0.95$
Pour petit ε,

$$\int_{21-\varepsilon/2}^{21+\varepsilon/2} f(x)dx \approx \varepsilon \cdot f(21)$$

Donc, $Pr(\text{temperature}{=}21 \mid \text{play}{=}\text{yes}) \sim f(21)$



Travaux Pratiques

17.5 DOCUMENT CLASSIFICATION

- Exercises 17.5.1–17.5.9

Source: Ian H. Witten and Eibe Frank: *Data Mining. Practical Machine Learning Tools and Techniques* (3rd Edition). Morgan Kaufmann, 2011

Outline

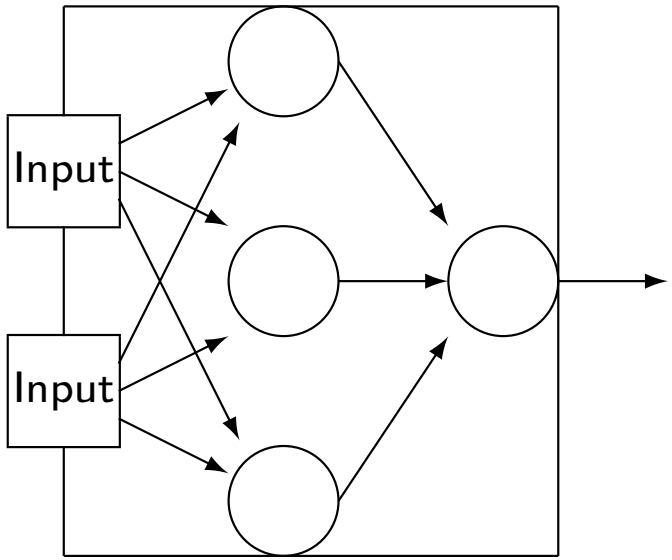
classifiers.functions.MultilayerPerceptron

Vue utilisateur :

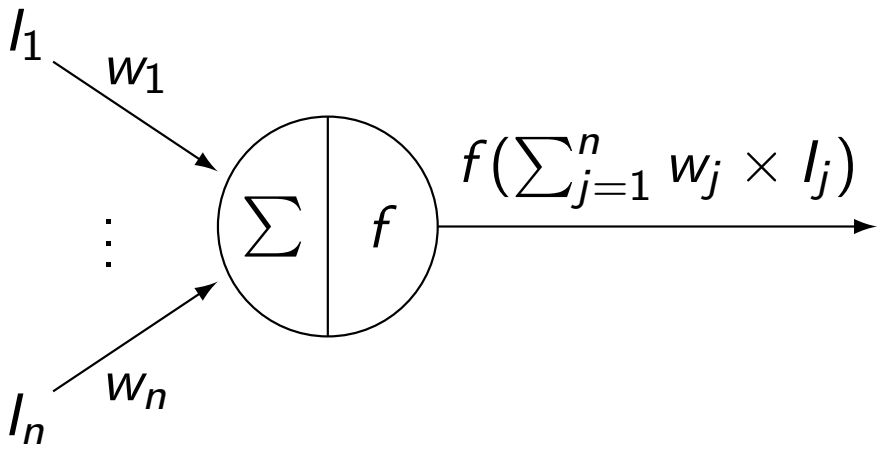
Attention :

Experience shows that in many applications [. . .], the explicit knowledge structures that are acquired, the structural descriptions, are at least as important, and often very much more important, than the ability to perform well on new examples. People frequently use data mining to gain knowledge, not just predictions. [Witten and Frank, pp. 7–8]

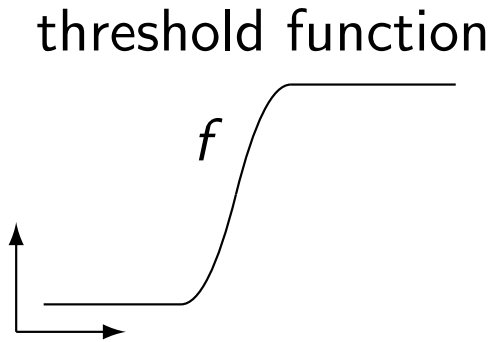
Un coup d'œil à l'intérieur...



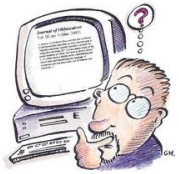
Neurone



où :

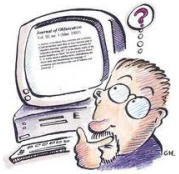


Une fois que la topologie (nombre de couches, nombre de neurones par couche) du réseau est fixée, le poids w_j de chaque connexion est choisi pour optimiser la prédicition sur les données d'apprentissage.



Exercice

- Prédire la classe des fichiers `straight.arff` et `segment.arff`.
- Déterminer le réseau le plus simple qui donne de bons résultats.
- Comparer avec la complexité de l'arbre construit par `classifiers.trees.J48`.



Travaux Pratiques

1. Voir `nhl.pdf`.
2. Voir `hepatitis.pdf`.
3. Voir `TPneurones.pdf`.