

# Machine Learning for Cyber-Security & Artificial Intelligence

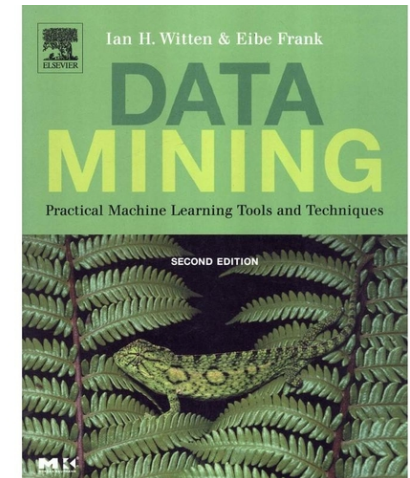
## Part 1 – Bayesian Networks

Hermes Senger

Adapted from:

**Chapters 4.2, 9.1, 9.2,** Bayesian networks

Data Mining: Practical Machine Learning Tools and Techniques,  
4th Edition. By Ian H. Witten, Eibe Frank, Mark A. Hall, Christopher J. Pal.  
Morgan Kauffman, 2017.



# Agenda

- Simple probabilistic modelling – Chapter 4.2
- Bayesian networks – Chapter 9.2

# Simple probabilistic modeling

- “Opposite” of 1R: use all the attributes
- Two assumptions: Attributes are
  - *equally important*
  - *statistically independent* (given the class value)
    - This means knowing the value of one attribute tells us nothing about the value of another takes on (if the class is known)
- Independence assumption is almost never correct!
- But ...
- The scheme is easy to implement in a program and very fast
- It is known as *naïve Bayes*
  - this scheme often works surprisingly well in practice!!!

# Probabilities for weather data

Outlook			Temperature			Humidity			Windy			Play	
<i>Yes</i>		<i>No</i>	<i>Yes</i>		<i>No</i>	<i>Yes</i>		<i>No</i>	<i>Yes</i>		<i>No</i>	<i>Yes</i>	<i>No</i>
Sunny	2	3	Hot	2	2	High	3	4	False	6	2	9	5
Overcast	4	0	Mild	4	2	Normal	6	1	True	3	3		
Rainy	3	2	Cool	3	1								
Sunny	2/9	3/5	Hot	2/9	2/5	High	3/9	4/5	False	6/9	2/5	9/14	5/14
Overcast	4/9	0/5	Mild	4/9	2/5	Normal	6/9	1/5	True	3/9	3/5		
Rainy	3/9	2/5	Cool	3/9	1/5								

Outlook	Temp	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

# Probabilities for weather data

Outlook			Temperature			Humidity			Windy			Play	
Yes No			Yes No			Yes No			Yes No			Yes	No
Sunny	2	3	Hot	2	2	High	3	4	False	6	2	9	5
Overcast	4	0	Mild	4	2	Normal	6	1	True	3	3		
Rainy	3	2	Cool	3	1								
Sunny	2/9	3/5	Hot	2/9	2/5	High	3/9	4/5	False	6/9	2/5	9/14	5/14
Overcast	4/9	0/5	Mild	4/9	2/5	Normal	6/9	1/5	True	3/9	3/5		
Rainy	3/9	2/5	Cool	3/9	1/5								

- A new day:

Outlook	Temp.	Humidity	Windy	Play
Sunny	Cool	High	True	?

Likelihood of the two classes

For "yes" =  $2/9 \times 3/9 \times 3/9 \times 3/9 \times 9/14 = 0.0053$

For "no" =  $3/5 \times 1/5 \times 4/5 \times 3/5 \times 5/14 = 0.0206$

Conversion into a probability by normalization:

$P(\text{"yes"}) = 0.0053 / (0.0053 + 0.0206) = 0.205$

$P(\text{"no"}) = 0.0206 / (0.0053 + 0.0206) = 0.795$

# Can combine probabilities using Bayes's rule

- Famous rule from probability theory due to

**Thomas Bayes**

**Born:** 1702 in London, England

**Died:** 1761 in Tunbridge Wells, Kent, England

- Probability of an event  $H$  given observed evidence  $E$ :

$$P(H | E) = P(E | H)P(H) / P(E)$$

- *A priori* probability of  $H$  :  $P(H)$ 
  - Probability of event *before* evidence is seen
- *A posteriori* probability of  $H$  :  $P(H | E)$ 
  - Probability of event *after* evidence is seen

# Naïve Bayes for classification

- Classification learning: what is the probability of the class given an instance?
  - Evidence  $E$  = instance's non-class attribute values
  - Event  $H$  = class value of instance
- **Naïve assumption**: evidence splits into parts (i.e., attributes) that are conditionally ***independent***
- This means, given  $n$  attributes, we can write Bayes' rule using a product of per-attribute probabilities:

$$P(H | E) = P(E_1 | H)P(E_2 | H) \dots P(E_n | H)P(H) / P(E)$$

# Weather data example

Outlook	Temp.	Humidity	Windy	Play
Sunny	Cool	High	True	?

← ***Evidence E***

***Probability of  
class “yes”***

$$P(\text{yes}|E) = \frac{P(E_1|\text{yes}) \times P(E_2|\text{yes}) \times P(E_3|\text{yes}) \times P(E_4|\text{yes}) \times P(\text{yes})}{P(E)}.$$

$$= \frac{2/9 \times 3/9 \times 3/9 \times 3/9 \times 9/14}{P(E)}$$

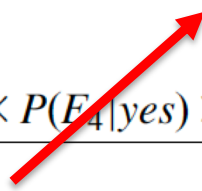


# The “zero-frequency problem”

- Naïve Bayes doesn't work well with this!
- What if an attribute value does not occur with every class value? (e.g., suppose “Humidity = high” never happens for class “yes”)

- Probability will be zero:

$$P(\text{Humidity} = \text{High} \mid \text{yes}) = 0 \quad \text{ZERO}$$

$$P(\text{yes} \mid E) = \frac{P(E_1 \mid \text{yes}) \times P(E_2 \mid \text{yes}) \times P(E_3 \mid \text{yes}) \times P(E_4 \mid \text{yes}) \times P(\text{yes})}{P(E)}$$


$$P(\text{yes} \mid E) = 0$$

- A *posteriori* probability will also be zero:  
(Regardless of how likely the other values are!)
- **Remedy**: add 1 to the count for every attribute value-class combination  
(Laplace estimator)
- **Result**: probabilities will never be zero
- Additional advantage: **stabilizes probability** estimates computed from **small samples of data**

# Modified probability estimates

- In some cases adding a constant different from 1 might be more appropriate
- Example: attribute *outlook* for class *yes*

$$\frac{2 + \mu/3}{9 + \mu}$$

**Sunny**

$$\frac{4 + \mu/3}{9 + \mu}$$

**Overcast**

$$\frac{3 + \mu/3}{9 + \mu}$$

**Rainy**

- Weights don't need to be equal (but they must sum to 1)

$$\frac{2 + \mu p_1}{9 + \mu}$$

$$\frac{4 + \mu p_2}{9 + \mu}$$

$$\frac{3 + \mu p_3}{9 + \mu}$$

# Missing values

- Not a problem at all for Bayesian formulation!
  - **Simply omit the attribute!!**
- Training: instance is not included in frequency count for attribute value-class combination
- Classification: attribute will be omitted from calculation
- Example:

Outlook	Temp.	Humidity	Windy	Play
?	Cool	High	True	?

Likelihood of "yes" =  $3/9 \times 3/9 \times 3/9 \times 9/14 = 0.0238$

Likelihood of "no" =  $1/5 \times 4/5 \times 3/5 \times 5/14 = 0.0343$

$P(\text{"yes"}) = 0.0238 / (0.0238 + 0.0343) = 41\%$

$P(\text{"no"}) = 0.0343 / (0.0238 + 0.0343) = 59\%$

# (Missing) Numeric attributes

- Not a problem!
  - Assume probability distribution!
- Usual assumption: attributes have a **normal** or **Gaussian** probability distribution (given the class)
- The *probability density function* for the normal distribution is defined by two parameters:

- Sample mean  $\mu$

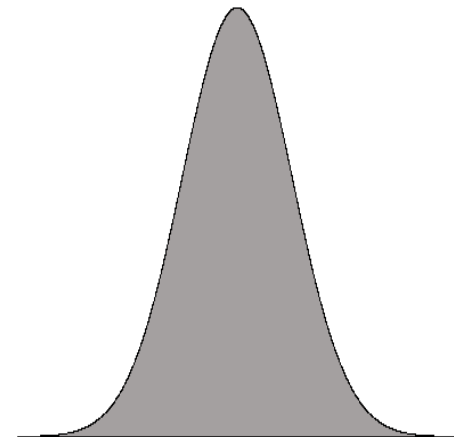
$$\mu = \frac{1}{N} \sum_{i=1}^N x_i$$

- Standard deviation  $\sigma$

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \mu)^2}$$

- Then the density function  $f(x)$  is

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



# Statistics for weather data

Outlook			Temperature		Humidity		Windy			Play	
			Yes	No	Yes	No				Yes	No
Sunny	2	3	64, 68,	65,71,	65, 70,	70, 85,	False	6	2	9	5
Overcast	4	0	69, 70,	72,80,	70, 75,	90, 91,	True	3	3		
Rainy	3	2	72, ...	85, ...	80, ...	95, ...					
Sunny	2/9	3/5	$\mu = 73$	$\mu = 75$	$\mu = 79$	$\mu = 86$	False	6/9	2/5	9/14	5/14
Overcast	4/9	0/5	$\sigma = 6.2$	$\sigma = 7.9$	$\sigma = 10.2$	$\sigma = 9.7$	True	3/9	3/5		
Rainy	3/9	2/5									

- Example density value:

$$f(\text{temperature} = 66|\text{yes}) = \frac{1}{\sqrt{2\pi} \cdot 6.2} e^{-\frac{(66-73)^2}{2 \cdot 6.2^2}} = 0.0340$$

# Classifying a new day

- A new day:

Outlook	Temp.	Humidity	Windy	Play
Sunny	66	90	true	?

Likelihood of "yes" =  $2/9 \times 0.0340 \times 0.0221 \times 3/9 \times 9/14 = 0.000036$

Likelihood of "no" =  $3/5 \times 0.0221 \times 0.0381 \times 3/5 \times 5/14 = 0.000108$

$P(\text{"yes"}) = 0.000036 / (0.000036 + 0.000108) = 25\%$

$P(\text{"no"}) = 0.000108 / (0.000036 + 0.000108) = 75\%$

- Missing values during training are not included in calculation of mean and standard deviation

# Naïve Bayes: discussion

- Naïve Bayes works surprisingly well even if independence assumption is clearly violated
- Why? Because *classification does not require accurate probability estimates as long as maximum probability is assigned to the correct class*
- However: *adding too many redundant attributes will cause problems* (e.g., identical attributes)
- Note also: many numeric attributes are not normally distributed (*kernel density estimators* can be used instead)

# Agenda

- Simple probabilistic modelling – Chapter 4.2
- Bayesian networks – Chapter 9.2



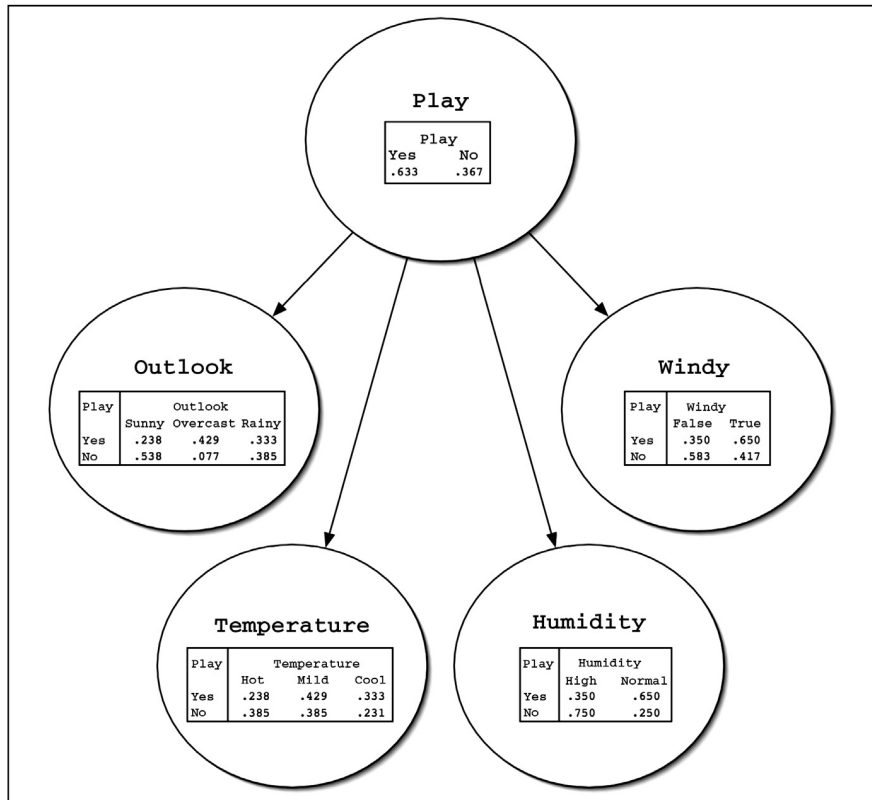
# Bayesian networks

- The chain rule holds for any order for the  $A_i$ s
- A Bayesian network is an acyclic graph,
- Therefore its nodes can be given an ordering where ancestors of node  $A_i$  have indices  $< i$
- Thus a Bayesian network can be written

$$P(A_1, A_2, \dots, A_n) = \prod_{i=1}^n P(A_i | \text{Parents}(A_i))$$

- When a variable has no parents, we use the unconditional probability of that variable

# Bayesian network #1 for the weather data



Random Variables

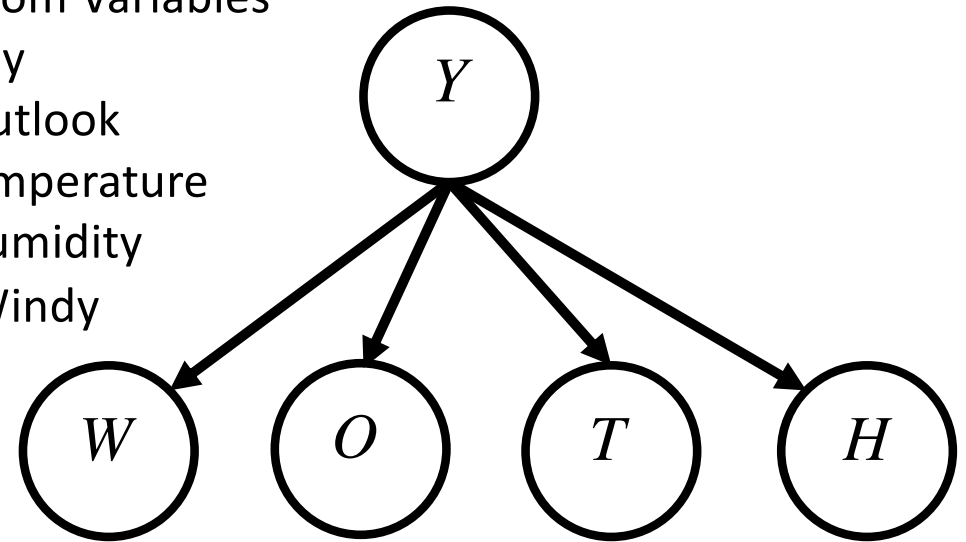
Y: Play

O: Outlook

T: Temperature

H: Humidity

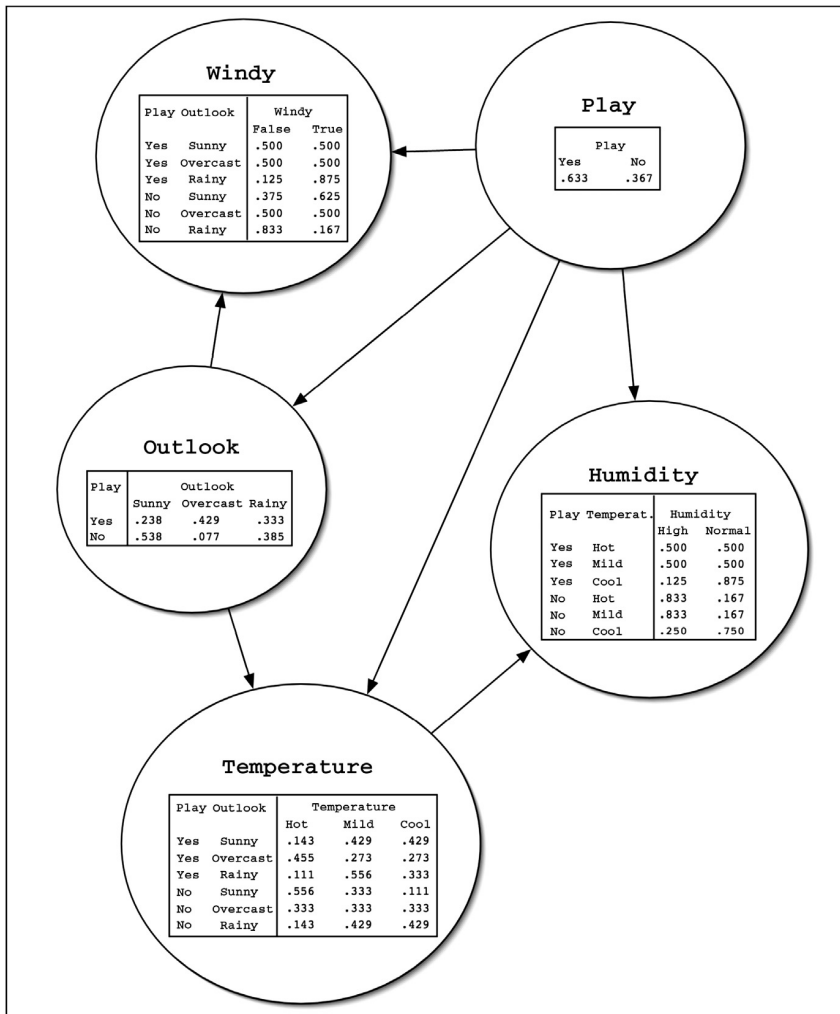
W: Windy



The graphs express the factorization below:

$$P(Y, O, T, H, W) = P(W | Y)P(O | Y)P(T | Y)P(H | Y)P(Y)$$

# Bayesian network #2 for the weather data



Random Variables

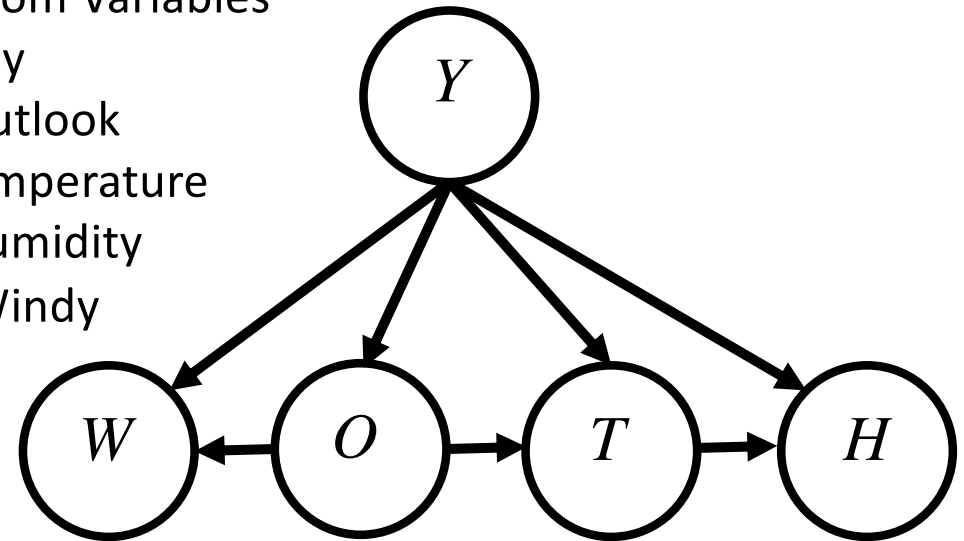
Y: Play

O: Outlook

T: Temperature

H: Humidity

W: Windy



The graphs express the factorization below:

$$P(Y, O, T, H, W) = P(W | O, Y) P(O | Y) P(T | O, Y) P(H | T, Y) P(Y)$$

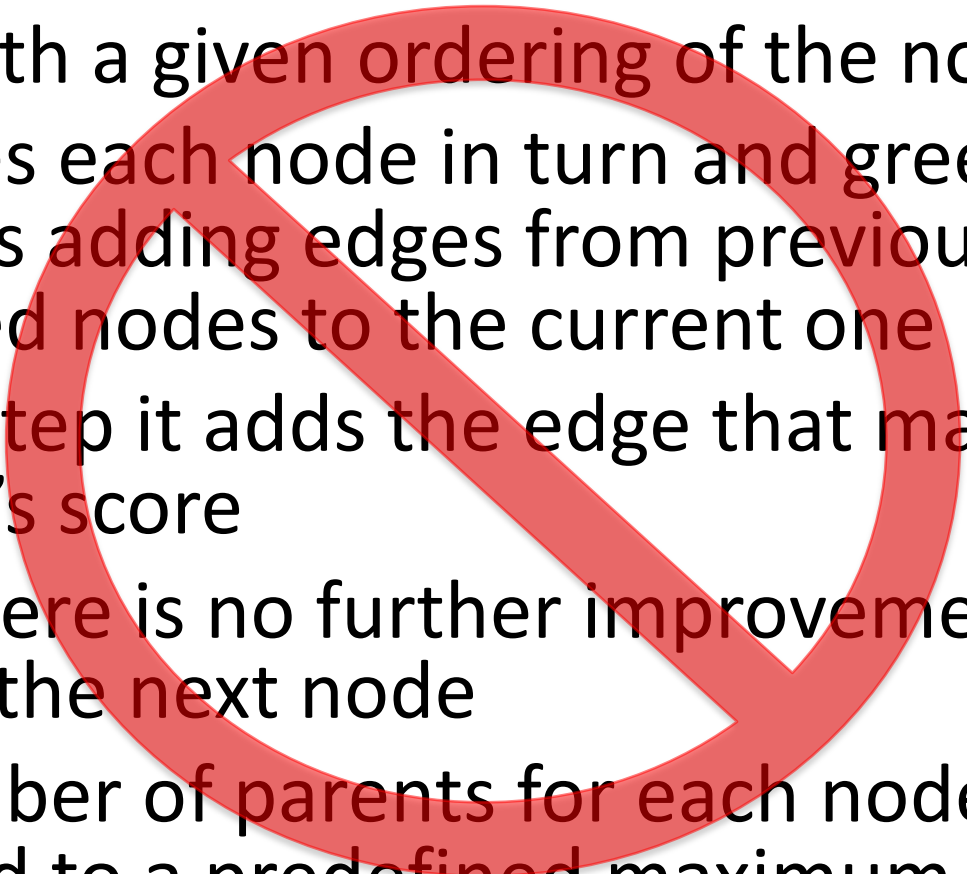
# Estimating conditional distributions

- Estimating conditional distributions in Bayesian networks is equally easy and amounts to simply counting configurations and dividing, ex.

$$P(B = b \mid A = a) = \frac{P(B = b, A = a)}{P(A = a)} = \frac{\sum_{i=1}^N \mathbf{1}(\tilde{A}_i = a, \tilde{B}_i = b)}{\sum_{i=1}^N \mathbf{1}(\tilde{A}_i = a)}.$$

- Zero counts cause problems and this motivates the use of Bayesian priors

# Network structure learning algorithm

- *K2*: a simple and very fast learning algorithm,
  - Starts with a given ordering of the nodes
  - Processes each node in turn and greedily considers adding edges from previously processed nodes to the current one
  - In each step it adds the edge that maximizes the network's score
  - When there is no further improvement, attention turns to the next node
  - The number of parents for each node can be restricted to a predefined maximum to mitigate overfitting
- 

# Tree augmented naïve bayes (TAN)

- Another good learning algorithm for Bayesian network classifiers
- Takes the Naïve Bayes (NB) classifier and adds edges to it
- The class attribute is the sole parent of each node in a NB model: TAN considers adding a second parent to each node
- If the class node and all corresponding edges are excluded from consideration, and assuming that there is exactly one node to which a second parent is not added, the resulting classifier has a tree structure rooted at the parentless node—hence the name
- For this restricted network type there is an efficient algorithm based on computing a maximum weighted spanning tree
- Algorithm is linear in the number of instances and quadratic in the number of attributes

# Naïve Bayes: discussion

- Naïve Bayes works surprisingly well even if independence assumption is clearly violated
- Why? Because *classification does not require accurate probability estimates as long as maximum probability is assigned to the correct class*
- However: *adding too many redundant attributes will cause problems* (e.g., identical attributes)
- Note also: many numeric attributes are not normally distributed (*kernel density estimators* can be used instead)