

Statistical modeling

- “Opposite” of 1R: use all the attributes
- Two assumptions: Attributes are
 - ◆ *equally important*
 - ◆ *statistically independent* (given the class value)
 - ★ This means that knowledge about the value of a particular attribute doesn’t tell us anything about the value of another attribute (if the class is known)
- Although based on assumptions that are almost never correct, this scheme works well in practice!

Probabilities for the weather data

Outlook	Temperature		Humidity		Windy		Play						
	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:

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

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

Bayes's rule

- Probability of event H given evidence E :

$$\Pr[H | E] = \frac{\Pr[E | H]\Pr[H]}{\Pr[E]}$$

- *A priori* probability of H : $\Pr[H]$
 - ◆ Probability of event *before* evidence has been seen
- *A posteriori* probability of H : $\Pr[H | E]$
 - ◆ Probability of event *after* evidence has been seen

Naïve Bayes for classification

- Classification learning: what's the probability of the class given an instance?
 - ◆ Evidence E = instance
 - ◆ Event H = class value for instance
- Naïve Bayes assumption: evidence can be split into independent parts (i.e. attributes of instance!)

$$\Pr[H | E] = \frac{\Pr[E_1 | H]\Pr[E_2 | H] \dots \Pr[E_n | H]\Pr[H]}{\Pr[E]}$$

The weather data example

Outlook	Temp.	Humidity	Windy	Play
Sunny	Cool	High	True	?

← Evidence E

$$\Pr[\text{yes} | E] = \Pr[\text{Outlook} = \text{Sunny} | \text{yes}] \times$$

$$\Pr[\text{Temperature} = \text{Cool} | \text{yes}] \times$$

$$\Pr[\text{Humidity} = \text{High} | \text{yes}] \times$$

$$\Pr[\text{Windy} = \text{True} | \text{yes}] \times$$

$$\frac{\Pr[\text{yes}]}{\Pr[E]}$$

$$= \frac{2/9 \times 3/9 \times 3/9 \times 3/9 \times 9/14}{\Pr[E]}$$

*Probability for
class “yes”*

The “zero-frequency problem”

- What if an attribute value doesn't occur with every class value (e.g. “Humidity = high” for class “yes”)?
 - ◆ Probability will be zero! $\Pr[\text{Humidity} = \text{High} \mid \text{yes}] = 0$
 - ◆ *A posteriori* probability will also be zero! $\Pr[\text{yes} \mid E] = 0$
(No matter 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! (also: stabilizes probability estimates)

Modified probability estimates

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

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

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

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

Sunny

Overcast

Rainy

- Weights don't need to be equal (if they 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

- 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” = $1/5 \times 4/5 \times 3/5 \times 5/14 = 0.0343$

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

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

Dealing with numeric attributes

- 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:

- ◆ The *sample mean* μ :

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

- ◆ The *standard deviation* σ :

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

- ◆ The density function $f(x)$:

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

Statistics for the weather data

	Outlook		Temperature		Humidity		Windy		Play			
	Yes	No	Yes	No	Yes	No	Yes	No	Yes	No		
Sunny	2	3	83	85	86	85	False	6	2	9	5	
Overcast	4	0	70	80	96	90	True	3	3			
Rainy	3	2	68	65	80	70						
								
Sunny	2/9	3/5	<i>mean</i>	73	<i>mean</i>	79.1	86.2	False	6/9	2/5	9/14	5/14
Overcast	4/9	0/5	<i>std dev</i>	6.2	<i>std dev</i>	10.2	9.7	True	3/9	3/5		
Rainy	3/9	2/5										

- Example density value:

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

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” = $3/5 \times 0.0291 \times 0.0380 \times 3/5 \times 5/14 = 0.000136$

$P(\text{“yes”}) = 0.000036 / (0.000036 + 0.000136) = 20.9\%$

$P(\text{“no”}) = 0.000136 / (0.000036 + 0.000136) = 79.1\%$

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

Probability densities

- Relationship between probability and density:

$$\Pr\left[c - \frac{\varepsilon}{2} < x < c + \frac{\varepsilon}{2}\right] \approx \varepsilon * f(c)$$

- But: this doesn't change calculation of *a posteriori* probabilities because ε cancels out
- Exact relationship:

$$\Pr[a \leq x \leq b] = \int_a^b f(t) dt$$

Discussion of Naïve Bayes

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