

COMP3308/3608, Lecture 6a

ARTIFICIAL INTELLIGENCE

Statistical-Based Learning

(Naïve Bayes)

Reference: Witten, Frank, Hall and Pal, ch.4.2: p.96-105

Russell and Norvig, ch.20: p.802-810

Outline

- **Bayes theorem**
- **Naïve Bayes algorithm**
- **Naïve Bayes - issues**
 - **Zero probabilities - Laplace correction**
 - **Dealing with missing values**
 - **Dealing with numeric attributes**

What is Bayesian Classification?

- Bayesian classifiers are statistical classifiers
- They can predict the **class membership probability**, i.e. the probability that a given example belongs to a particular class
- They are based on the **Bayes Theorem**



Thomas Bayes (1702-1761)

Bayes Theorem

- Given a **hypothesis** H and **evidence** E for this hypothesis, then the probability of H given E , is:

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

- Example: Given are instances of fruits, described by their color and shape. Let:**
 - E is red and round
 - H is the hypothesis that E is an apple
- What are:**
 - $P(H/E)=?$
 - $P(H)=?$
 - $P(E/H)=?$
 - $P(E)=?$



Bayes Theorem – Example (cont. 1)

- $P(H/E)$ is the probability that E is an apple, given that we have seen that E is red and round
 - Called *posteriori probability* of H conditioned on E
- $P(H)$ is the probability that any given example is an apple, regardless of how it looks
 - Called *prior probability* of H
- The posteriori probability is based on more information than the prior probability which is independent of E



Bayes Theorem – Example (cont. 2)

- What is $P(E/H)$?
 - the posteriori probability of E conditioned on H
 - the probability that E is red and round, given that we know that E is an apple
- What is $P(E)$?
 - the prior probability of E
 - The probability that an example from the fruit data set is red and round



Bayes Theorem for Problem Solving

- **Given: A doctor knows that**
 - Meningitis causes stiff neck 50% of the time
 - Prior probability of any patient having meningitis is 1/50 000
 - Prior probability of any patient having stiff neck is 1/20
- **If a patient has a stiff neck, what is the probability that he has meningitis?**

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

Bayes Theorem for Problem Solving - Answer

- **Given: A doctor knows that**
 - Meningitis causes stiff neck 50% of the time $P(S | M)$
 - Prior probability of any patient having meningitis is 1/50 000 $P(M)$
 - Prior probability of any patient having stiff neck is 1/20 $P(S)$
- **If a patient has a stiff neck, what is the probability that he has meningitis?** $P(M | S) = ?$

$$P(M | S) = \frac{P(S | M)P(M)}{P(S)} = \frac{0.5 (1 / 50000)}{1 / 20} = 0.0002$$

Naïve Bayes Algorithm

- The Bayes Theorem can be applied for classification tasks = Naïve Bayes algorithm
- While 1R makes decisions based on a single attribute, Naive Bayes uses all attributes and allows them to make contributions to the decision that are *equally important and independent* of one another
- Assumptions of the Naïve Bayes algorithm
 - 1) Independence assumption – (the values of the) attributes are conditionally independent of each other, given the class (i.e. for each class value)
 - 2) Equally importance assumption – all attributes are equally important
- Unrealistic assumptions! => it is called *Naive* Bayes
 - Attributes are dependent of one another
 - Attributes are not equally important
- But these assumptions lead to a simple method which works surprisingly well in practice!

Naive Bayes on the Weather Example

- Given: the weather data →
- Task: use Naïve Bayes to predict the class (*yes* or *no*) of the new example

outlook=sunny, temperature=cool,
humidity=high, windy=true

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

• The Bayes Theorem:
$$P(H | E) = \frac{P(E | H)P(H)}{P(E)}$$



- What are H and E?
 - the evidence **E** is the new example
 - the hypothesis **H** is **play=yes** (and there is another H: **play=no**)



- How to use the Bayes Theorem for classification?
 - Calculate $P(H/E)$ for each **H** (class), i.e. $P(\text{yes}|E)$ and $P(\text{no}|E)$
 - Compare them and assign **E** to the class with the highest probability
 - OK, but for $P(H/E)$ we need to calculate $P(E)$, $P(H)$ and $P(E/H)$ – how to do this? From the given data (this is the training phase of the classifier)

Naive Bayes on the Weather Example (2)

- We need to calculate and compare $P(\text{yes}|E)$ and $P(\text{no}|E)$

$$P(\text{yes} | E) = \frac{P(E | \text{yes})P(\text{yes})}{P(E)}$$
$$P(\text{no} | E) = \frac{P(E | \text{no})P(\text{no})}{P(E)}$$

where E

outlook=sunny, temperature=cool,
humidity=high, windy=true

1) How to calculate $P(E|\text{yes})$ and $P(E|\text{no})$?

Let's split the evidence E into 4 smaller pieces of evidence:

- E1 = outlook=sunny, E2 = temperature=cool
- E3 = humidity=high, E4 = windy=true

Let's use the Naïve Bayes's independence assumption: E1, E2, E3 and E4 are independent given the class. Then, their combined probability is obtained by multiplication of per-attribute probabilities:

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

$$P(E | \text{no}) = P(E_1 | \text{no}) P(E_2 | \text{no}) P(E_3 | \text{no}) P(E_4 | \text{no})$$

Naive Bayes on the Weather Example (3)

- Hence:

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

$$P(\text{no} | E) = \frac{P(E_1 | \text{no}) P(E_2 | \text{no}) P(E_3 | \text{no}) P(E_4 | \text{no}) P(\text{no})}{P(E)}$$

- In summary:
 - **Numerator** - the probabilities will be estimated from the data
 - **Denominator** – the two denominators are the same ($P(E)$) and since we are comparing the two fractions, we can just compare the numerators => there is no need to calculate $P(E)$

Calculating the Probabilities from the Training Data

E1 = outlook=sunny, E2 = temperature=cool

E3 = humidity=high, E4 = windy=true

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

- $P(E_1|\text{yes})=P(\text{outlook}=\text{sunny}|\text{yes})=?$

- $P(E_2|\text{yes})=P(\text{temp}=\text{cool}|\text{yes})=?$

- $P(E_3|\text{yes})=P(\text{humidity}=\text{high}|\text{yes})=?$

- $P(E_4|\text{yes})=P(\text{windy}=\text{true}|\text{yes})=?$

- $P(\text{yes})=?$

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

Calculating the Probabilities from the Training Data

E1 = outlook=sunny, E2 = temperature=cool

E3 = humidity=high, E4 = windy=true

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

- $P(E_1|\text{yes})=P(\text{outlook}=\text{sunny}|\text{yes}) = ?/9 = 2/9$

- $P(E_2|\text{yes})=P(\text{temp}=\text{cool}|\text{yes})=?$

- $P(E_3|\text{yes})=P(\text{humidity}=\text{high}|\text{yes})=?$

- $P(E_4|\text{yes})=P(\text{windy}=\text{true}|\text{yes})=?$

- $P(\text{yes})=?$

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

Calculation the Probabilities (2)

- Weather data - counts and probabilities:

	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								

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

proportions of days when
humidity is normal and play is yes
i.e. the probability of humidity to
be normal given that play=yes

proportions of days
when play is yes

Calculation the Probabilities (3)

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

	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								

⇒ $P(E_1 | \text{yes}) = P(\text{outlook} = \text{sunny} | \text{yes}) = 2/9$

$P(E_2 | \text{yes}) = P(\text{temperature} = \text{cool} | \text{yes}) = 3/9$

$P(E_3 | \text{yes}) = P(\text{humidity} = \text{high} | \text{yes}) = 3/9$

$P(E_4 | \text{yes}) = P(\text{windy} = \text{true} | \text{yes}) = 3/9$

- $P(\text{yes}) = ?$ - the probability of a yes without knowing any E , i.e. anything about the particular day; the **prior probability** of yes; $P(\text{yes}) = 9/14$

Final Calculations

- By substituting the respective evidence probabilities:

$$P(\text{yes} | E) = \frac{\frac{2}{9} \frac{3}{9} \frac{3}{9} \frac{3}{9}}{P(E)} = \frac{0.0053}{P(E)}$$

- Similarly calculating $P(\text{no} | E)$:

$$P(\text{no} | E) = \frac{\frac{3}{5} \frac{1}{5} \frac{4}{5} \frac{3}{5}}{P(E)} = \frac{0.0206}{P(E)}$$

- $\Rightarrow P(\text{no} | E) > P(\text{yes} | E)$
- \Rightarrow for the new day **play = no** is more likely than **play = yes**

Another Example

- **Use the NB classifier to solve the following problem:**
- **Consider a volleyball game between team A and team B**
 - **Team A has won 65% of the time and team B has won 35%**
 - **Among the games won by team A, 30% were when playing on team B's court**
 - **Among the games won by team B, 75% were when playing at home**
- **If team B is hosting the next match, which team is most likely to win?**

Solution

- **host** – the team hosting the match {A, B}
- **winner** – the winner of the match {A, B}
- Using NB, the task is to compute and compare 2 probabilities:
 $P(\text{winner}=A|\text{host}=B)$
 $P(\text{winner}=B|\text{host}=B)$

$$P(\text{winner} = A | \text{host} = B) = \frac{P(\text{host} = B | \text{winner} = A)P(\text{winner} = A)}{P(\text{host} = B)}$$

$$P(\text{winner} = B | \text{host} = B) = \frac{P(\text{host} = B | \text{winner} = B)P(\text{winner} = B)}{P(\text{host} = B)}$$

Solution (2)

$$P(\text{winner} = A \mid \text{host} = B) = \frac{P(\text{host} = B \mid \text{winner} = A)P(\text{winner} = A)}{P(\text{host} = B)}$$

$$P(\text{winner} = B \mid \text{host} = B) = \frac{P(\text{host} = B \mid \text{winner} = B)P(\text{winner} = B)}{P(\text{host} = B)}$$

- **Do we know these probabilities:**
 - **$P(\text{winner}=A)= ?$ //probability that A wins**
 - **$P(\text{winner}=B)=?$ //probability that B wins**
 - **$P(\text{host}=B|\text{winner}=A)=?$ //probability that team B hosted the match, given that team A won**
 - **$P(\text{host}=B|\text{winner}=B)=?$ //probability that team B hosted the match, given that team B won**

Solution (3)

$$P(\text{winner} = A \mid \text{host} = B) = \frac{P(\text{host} = B \mid \text{winner} = A)P(\text{winner} = A)}{P(\text{host} = B)}$$

$$P(\text{winner} = B \mid \text{host} = B) = \frac{P(\text{host} = B \mid \text{winner} = B)P(\text{winner} = B)}{P(\text{host} = B)}$$

- Do we know these probabilities:
 - $P(\text{winner}=A)= ?$ //probability that A wins **=0.65**
 - $P(\text{winner}=B)=?$ //probability that B wins **=0.35**
 - $P(\text{host}=B|\text{winner}=A)=?$ //probability that team B hosted the match, given that team A won **=0.30**
 - $P(\text{host}=B|\text{winner}=B)=?$ //probability that team B hosted the match, given that team B won **=0.75**

Solution (4)

$$\begin{aligned} P(\text{winner} = A \mid \text{host} = B) &= \frac{P(\text{host} = B \mid \text{winner} = A)P(\text{winner} = A)}{P(\text{host} = B)} = \\ &= \frac{0.3 * 0.65}{P(\text{host} = B)} = 0.195 \end{aligned}$$

$$\begin{aligned} P(\text{winner} = B \mid \text{host} = B) &= \frac{P(\text{host} = B \mid \text{winner} = B)P(\text{winner} = B)}{P(\text{host} = B)} = \\ &= \frac{0.75 * 0.35}{P(\text{host} = B)} = 0.2625 \end{aligned}$$

=>NB predicts team B

i.e. NB predicts that if team B is hosting the next match, then team B is more likely to win

Three More Things About Naïve Bayes

- **How to deal with probability values of zero in the numerator?**
- **How do deal with missing values?**
- **How to deal with numeric attributes?**

Problem – Probability Values of 0

- Suppose that the training data was different:
outlook=sunny had always occurred together with **play=no** (i.e. **outlook=sunny** had never occurred together with **play=yes**)

- Then:

$P(\text{outlook}=\text{sunny}|\text{yes})=0$ and

$P(\text{outlook}=\text{sunny}|\text{no})=1$

$$P(\text{yes} | E) = \frac{\underbrace{P(E_1 | \text{yes})}_{=0} P(E_2 | \text{yes}) P(E_3 | \text{yes}) P(E_4 | \text{yes}) P(\text{yes})}{P(E)}$$

	yes	...
sunny	0	...
overcast	4	...
rainy	3	...
		...
sunny	0/9	...
overcast	4/9	...
rainy	3/9	...

- => final probability **$P(\text{yes}|E)=0$** no matter of the other probabilities
- This is not good!
 - The other probabilities are completely ignored due to the multiplication with 0
 - I.e. the prediction for new examples with **outlook=sunny** will always be **no**, regardless of the values of the other attributes

A Simple Trick to Avoid This Problem

- Assume that our training data is so large that adding 1 to each count would not make difference in calculating the probabilities ...
- but it will avoid the case of 0 probability
- This is called the Laplace correction or Laplace estimator



“What we know is not much. What we do not know is immense.”

Pierre-Simon Laplace (1749-1827)

Image from http://en.wikipedia.org/wiki/File:Pierre-Simon_Laplace.jpg

Laplace Correction

- Add 1 to the numerator and k to the denominator, where k is the number of attribute values for the given attribute
- Example:
 - A dataset with 2000 examples, 2 classes: *buy_Mercedes=yes* and *buy_Mercedes=no*; 1000 examples in each class
 - 1 of the attributes is *income* with 3 values: *low*, *medium* and *high*
 - For class *buy_Mercedes=yes*, there are 0 examples with *income=low*, 10 with *income=medium* and 990 with *income=high*
- Probabilities without the Laplace correction for class *yes*:
 $0/1000=0$, $10/1000=0.01$, $990/1000=0.99$
- Probabilities with the Laplace correction:
 $1/1003=0.001$, $11/1003=0.011$, $991/1003=0.988$
- The correct probabilities are close to the adjusted probabilities, yet the 0 probability value is avoided!

Laplace Correction – Modified Weather Example

	yes	...
sunny	0	...
overcast	4	...
rainy	3	...
		...
sunny	0/9	...
overcast	4/9	...
rainy	3/9	...

$P(\text{sunny}|\text{yes})=0/9 \rightarrow \text{problem}$

$P(\text{overcast}|\text{yes})=4/9$

$P(\text{rainy}|\text{yes})=3/9$

Laplace correction

- Assumes that there are 3 more examples from class *yes*, 1 for each value of *outlook*
- This results in adding 1 to the numerator and 3 to the denominator of all probabilities
- Ensures that an attribute value which occurs 0 times will receive a nonzero (although small) probability

$$P(\text{sunny} | \text{yes}) = \frac{0+1}{9+3} = \frac{1}{12}$$

$$P(\text{overcast} | \text{yes}) = \frac{4+1}{9+3} = \frac{5}{12}$$

$$P(\text{rainy} | \text{yes}) = \frac{3+1}{9+3} = \frac{4}{12}$$

Generalization of the Laplace Correction: M-estimate

- Add a small constant m to each denominator and mp_i to each numerator, where p_i is the prior probability of the i values of the attribute:

$$P(\text{sunny} \mid \text{yes}) = \frac{2 + mp_1}{9 + m}$$

$$P(\text{overcast} \mid \text{yes}) = \frac{4 + mp_2}{9 + m}$$

$$P(\text{rainy} \mid \text{yes}) = \frac{3 + mp_3}{9 + m}$$

- Note that $p_1 + p_2 + \dots + p_n = 1$, n - number of attribute values
- Advantage of using prior probabilities – it is rigorous
- Disadvantage – computationally expensive to estimate prior probabilities
- Large m - the prior probabilities are very important compared with the new evidence coming in from the training data; small m - less important
- Typically we assume that each attribute value is equally probable, i.e. $p_1 = p_2 = \dots = p_n = 1/n$
- The Laplace correction is a special case of the m-estimate, where $p_1 = p_2 = \dots = p_n = 1/n$ and $m = n$. Thus, 1 is added to the numerator and m to the denominator.

Handling Missing Values - Easy

- Missing attribute value in the new example – do not include this attribute

- e.g. **outlook=?**, temperature=cool, humidity=high, windy=true

- Then:

$P(\text{yes} E) = \frac{\overline{3} \ \overline{3} \ \overline{3} \ \overline{9}}{\overline{9} \ \overline{9} \ \overline{9} \ \overline{14}} = \frac{0.0238}{P(E)}$	$P(\text{no} E) = \frac{\overline{1} \ \overline{4} \ \overline{3} \ \overline{5}}{\overline{5} \ \overline{5} \ \overline{5} \ \overline{14}} = \frac{0.0343}{P(E)}$
--	---



- **outlook** is not included. Compare these results with the previous results!
 - As one of the fractions is missing, the probabilities are higher but the comparison is fair - there is a missing fraction in both cases
- Missing attribute value in a training example – do not include this value in the counts
 - Calculate the probabilities based on the number of values that actually occur (are not missing) and not on the total number of training examples

Handling Numeric Attributes

outlook			temperature		humidity		windy			play	
	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					
			64	72	65	95					
			69	71	70	91					
			75		80						
			75		70						
			72		90						
			81		75						
sunny	2/9	3/5	mean	73 74.6	mean	79.1 86.2	false	6/9	2/5	9/14	5/14
overcast	4/9	0/5	std dev	6.2 7.9	std dev	10.2 9.7	true	3/9	3/5		
rainy	3/9	2/5									

numeric

- We would like to classify the following new example:
outlook=sunny, temperature=66, humidity=90, windy=true



- Question: How to calculate
 $P(\text{temperature}=66|\text{yes})=?$, $P(\text{humidity}=90|\text{yes})=?$
 $P(\text{temperature}=66|\text{no})=?$, $P(\text{humidity}=90|\text{no})=?$

Using Probability Density Function

- **Answer:** By assuming that numerical values have a *normal* (Gaussian, bell curve) probability distribution and using the probability density function
- For a *normal* distribution with mean μ and standard deviation σ , the probability density function is:

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

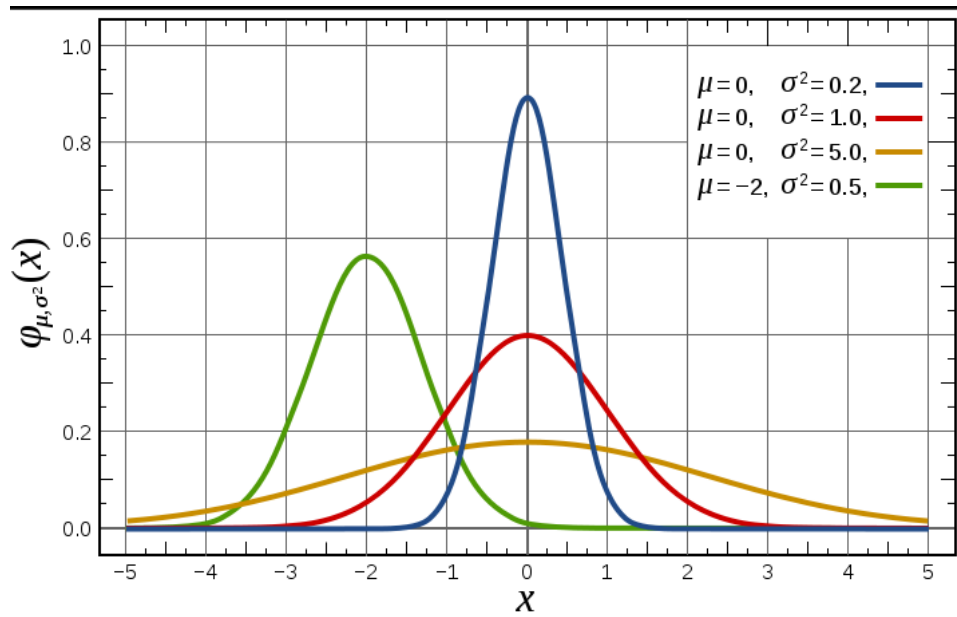


Image from http://en.wikipedia.org/wiki/File:Normal_Distribution_PDF.svg

More on Probability Density Functions



What is the meaning of the probability density function of a continuous random variable?

- closely related to probability but not exactly the probability (e.g. the probability that x is exactly 66 is 0)
- = the probability that a given value $x \in (x-\epsilon/2, x+\epsilon/2)$ is $\epsilon \cdot f(x)$

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

Calculating Probabilities Using Probability Density Function

$$f(\text{temperature} = 66 \mid \text{yes}) = \frac{1}{6.2\sqrt{2\pi}} e^{-\frac{(66-73)^2}{2*6.2^2}} = 0.034$$

mean for temp. for class=yes

$$f(\text{humidity} = 90 \mid \text{yes}) = 0.0221$$

$$P(\text{yes} \mid E) = \frac{\frac{2}{9} 0.034 \frac{0.0221}{9} \frac{3}{14}}{P(E)} = \frac{0.000036}{P(E)}$$

std.dev. for temp. for class=yes

=> $P(\text{no} \mid E) > P(\text{yes} \mid E)$

=> no play

$$P(\text{no} \mid E) = \frac{\frac{3}{5} 0.0291 \frac{0.038}{5} \frac{3}{14}}{P(E)} = \frac{0.000136}{P(E)}$$



• Compare with the categorical weather data!

Mean and Standard Deviation - Reminder

- A reminder how to calculate the mean value μ and standard deviation σ :

X is a random variable with values, x_1, x_2, \dots, x_n

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

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

Note that the denominator is $n-1$ not n

Naive Bayes - Advantages

- **Simple approach – the probabilities are easily computed due to the independence assumption**
- **Clear semantics for representing, using and learning probabilistic knowledge**
- **Excellent computational complexity**
 - **Requires 1 scan of the training data to calculate all statistics (for both nominal and continuous attributes assuming normal distribution):**
 - **$O(pk)$, p - # training examples, k -valued attributes**
- **In many cases outperforms more sophisticated learning methods
=> always try the simple method first!**
- **Robust to isolated noise points as such points are averaged when estimating the conditional probabilities from data**

Naive Bayes - Disadvantages

- **Correlated attributes reduce the power of Naïve Bayes**
 - **Violation of the independence assumption**
 - **Solution: apply feature selection beforehand to identify and discard correlated (redundant) attributes**
- **Normal distribution assumption for numeric attributes - many features are not normally distributed – solutions:**
 - **Discretize the data first, i.e. numerical -> nominal attributes**
 - **Use other probability density functions, e.g. Poisson, binomial, gamma, etc.**
 - **Transform the attribute using a suitable transformation into a normally distributed one (sometimes possible)**
 - **Use kernel density estimation – doesn't assume any particular distribution**

COMP3308/3608, Lecture 6b

ARTIFICIAL INTELLIGENCE

Evaluating and Comparing Classifiers

Reference: Witten, Frank, Hall and Pal, ch.5: p.161-194

Russell and Norvig, p.695-697

Outline

- **Evaluating and comparing classifiers**
 - **Empirical evaluation**
 - **Measures: error rate and accuracy**
 - **Single holdout estimation, repeated holdout**
 - **Stratification**
 - **Cross-validation**
 - **Comparing classifiers – t-paired significance test**
 - **Other performance measures: recall, precision and F1**
 - **Cost-sensitive evaluation**
- **Inductive learning**

Evaluation: the Key to Success

- How to *evaluate* the performance of classifiers?
 - What performance measures to use?
 - What procedures to follow?
- How to *compare* the performance of 2 classifiers?
- Recall our goal: a classifier which *generalises well on new data*
 - i.e. classifies correctly new data, unseen during training

Holdout Procedure

- Simple way to evaluate performance of a classifier
- Holdout procedure:
 - Split data into 2 **independent (non-overlapping)** sets: *training* and *test* (usually 2/3 and 1/3)
 - Use the training data to build the classifier
 - Use the test data to evaluate how good the classifier is
 - Calculate performance measures such as *accuracy* on test data

outlook	temp.	humidity	windy	play
sunny	85	85	false	no
sunny	80	90	true	no
overcast	83	86	false	yes
rainy	70	96	false	yes
rainy	68	80	false	yes
rainy	65	70	true	no
overcast	64	65	true	yes
sunny	72	95	false	no
sunny	69	70	false	yes
rainy	75	80	false	yes
sunny	75	70	true	yes
overcast	73	90	true	yes
overcast	81	75	false	yes
rainy	71	91	true	no

training data: 9
examples (2/3)

test data: 5
examples (1/3)

Accuracy and Error Rate

- **Accuracy:** proportion of **correctly** classified examples (i.e. their class is predicted correctly)
- **Error rate:** complimentary to accuracy - proportion of **incorrectly** classified examples (i.e. their class is predicted incorrectly)
- Accuracy and error rate sum to 1; typically given in % => sum to 100
- Evaluated on training and test set
 - **accuracy on training data** is overly optimistic, not a good indicator of performance on future data
 - **accuracy on test data** is the performance measure used

Accuracy - Example

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

training data

- 1R classifier

if outlook=sunny then play=no

elseif outlook=overcast then play=yes

elseif outlook=rainy then play=yes

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

test data

Accuracy on training data=?

Accuracy on test data =?

Validation Set

- Sometimes we need to use a 3rd set: validation set
 - E.g. some classification methods (DTs, NNs) operate in two stages:
 - Stage 1: build the classifier
 - Stage 2: tune its parameters
 - The test data can not be used for parameter tuning (stage 2)!
- Rule: the test data should not be used in any way to create the classifier
- Proper procedure uses 3 non-overlapping data sets
 - 1) Training set - to build the classifier
 - 2) Validation set - to tune parameters
 - 3) Test set - to evaluate accuracy
 - Examples
 - DTs – training set is used to build the tree, validation set is used for pruning, test set – to evaluate performance
 - NNs – validation set is used to stop the training (to prevent overtraining)

Making the Most of the Data

- **Generally**
 - The larger the training data, the better the classifier
 - The larger the test data, the better the accuracy estimate
- **Dilemma - ideally we want to use as much data as possible for**
 - training to get a good classifier
 - testing to get a good accuracy estimate
- **Once the evaluation is completed, all the data can be used to build the final classifier**
 - i.e. training, validation and test sets are joined together, a classifier is built using all of them for actual use (i.e. give to the customer)
 - But the accuracy of the classifier must be quoted to the customer based on test data

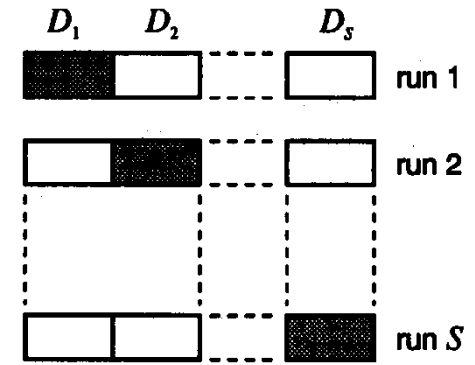
Stratification

- An improvement of the holdout method
- The holdout method reserves a certain amount for testing and uses the remainder for training
- Problem: the examples in the training set might not be representative of all classes
 - E.g.: all examples with a certain class are missing in the training set => the classifier *cannot* learn to predict this class
- Solution: *stratification*
 - Ensures that each class is represented with approximately equal proportions in both data sets (training and test)
- Stratification is used together with the evaluation method (e.g. holdout or cross validation)

Repeated Holdout Method

- Holdout can be made more reliable by repeating the process
 - E.g. 10 times: In each of the 10 iteration, a certain proportion (e.g. 2/3) is randomly selected for training (possibly with stratification) and the reminder is used for testing
 - The accuracy on the different iterations are averaged to yield an overall accuracy
- This is called *repeated holdout method*
- Can be improved by ensuring that the test sets do not overlap -> *cross validation*

Cross-Validation



from Neural Networks for Pattern Recognition,
C. Bishop, Oxford Uni Press, 1995

- Avoids overlapping test sets
- *S-fold-cross validation*:

Step 1: Data is split into S subsets of equal size

Step 2: A classifier is built S times. Each time the testing is on 1 segment (black) and the training is on the remaining $S-1$ segments (white)

Step 3: Average accuracies of each run to calculate overall accuracy.

- *10-fold cross-validation* – a standard method for evaluation

Split data into 10 non-overlapping sets $set1, \dots, set10$ of approx. equal size

Run1: train on $set1 + \dots + set9$, test on $set10$ and calculate accuracy ($acc1$)

Run2: train on $set1 + \dots + set8 + set10$, test on $set9$ and calculate accuracy ($acc2$)

....


Run10: train on $set2 + \dots + set10$, test on $set1$ and calculate accuracy ($acc10$)

final cross validation accuracy = average ($acc1, acc2, \dots, acc10$)

More on Cross-Validation

- **Better to be used with stratification**
 - the subsets are stratified before the cross-validation is performed
 - Weka does this
- **Neither the stratification, nor the split into 10 folds needs to be exact**
 - 10 approximately equal sets, in each of which the class values are represented in approximately the right proportion
- **Even better: repeated stratified cross-validation**
 - e.g. 10-fold cross-validation is repeated 10 times and results are averaged
 - Reduces the effect of random variation in choosing the folds

Leave-one-out Cross-Validation

- n -fold cross-validation, where n is the number of examples in the data set
-  How many times do we need to build the classifier?
- **Advantages**
 - The greatest possible amount of data is used for training => increases the chance for building an accurate classifier
 - Deterministic procedure – no random sampling is involved (no point in repeating the procedure – the same results will be obtained)
- **Disadvantage**
 - high computational cost => useful for small data sets

Comparing Classifiers

- Given two classifiers C1 and C2, which one is better on a given task?
- Step 1: Use 10-fold CV and then compare the 2 accuracies
- How much can we trust this comparison? Are the 10-fold CV estimates significantly different?
- Step 2: Run a statistical test to find if the differences are significant
 - paired t-test can be used

	C1	C2
Fold 1	95%	91%
...		
Fold 2	82%	85%
=====		
mean	91.3%	89.4%
(10-fold CV)		

Is C1 better than C2? Is this difference significant?

Comparing Classifiers (2)

	C1	C2	
Fold 1	95%	91%	$d_1 = 95 - 91 = 4$
...			
Fold 2	82%	85%	$d_2 = 82 - 85 = 3$
=====			
mean	91.3%	89.4%	$d_{mean} = 3.5$

$$\sigma = \sqrt{\frac{\sum_i^k (d_i - d_{mean})^2}{k - 1}}$$

1. Calculate the differences d_i
2. Calculate the standard deviation of the differences (an estimate of the true standard deviation)

If k is sufficiently large, d_i is normally distributed

3. Calculate the confidence interval Z: t is obtained from a probability table

$$Z = d_{mean} \pm t_{(1-\alpha)(k-1)} \frac{\sigma}{\sqrt{k}}$$

1- α – confidence level
k-1 – degree of freedom

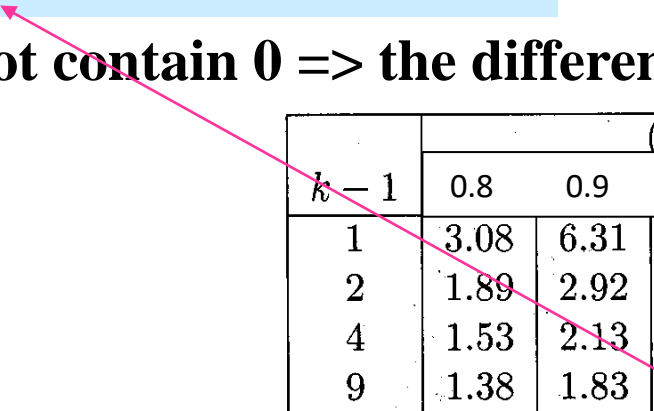
4. Interval contains 0 – difference not significant, else significant

Comparing Classifiers - Example

Suppose that:

- We use 10 fold CV $\Rightarrow k=10$
- $d_{mean}=3.5; \frac{\sigma}{\sqrt{k}} = 0.5$
- We are interested in significance at 95% confidence level
- Then: $Z = 3.5 \pm 2.26 \times 0.5 = 3.5 \pm 1.13$

\Rightarrow The interval does not contain 0 \Rightarrow the difference is statistically significant



$k - 1$	$(1 - \alpha)$				
	0.8	0.9	0.95	0.98	0.99
1	3.08	6.31	12.7	31.8	63.7
2	1.89	2.92	4.30	6.96	9.92
4	1.53	2.13	2.78	3.75	4.60
9	1.38	1.83	2.26	2.82	3.25
14	1.34	1.76	2.14	2.62	2.98
19	1.33	1.73	2.09	2.54	2.86
24	1.32	1.71	2.06	2.49	2.80
29	1.31	1.70	2.04	2.46	2.76

Confusion Matrix

- 2 class prediction (classes *yes* and *no*) – 4 different outcomes

- Confusion matrix:

examples	# assigned to class yes	# assigned to class no
# from class yes	true positives (tp)	false negatives (fn)
# from class no	false positives (fp)	true negatives (tn)



- How can we express *accuracy* in terms of *tp*, *fn*, *fp*, *tn*?
- Where are the correctly classified examples?
- Where are the misclassifications?
- Weka – iris data classification using 1R (3 classes); confusion matrix:

a b c <-- classified as

50 0 0 | a = Iris-setosa

0 44 6 | b = Iris-versicolor

0 3 47 | c = Iris-virginica

accuracy=?

Other Performance Measures: Recall, Precision, F1

- Information retrieval (IR) uses *recall (R)*, *precision (P)* and their combination *F1 measure (F1)* as performance measures

$$P = \frac{tp}{tp + fp} \quad R = \frac{tp}{tp + fn} \quad F1 = \frac{2PR}{P + R}$$

examples	# assigned to class yes	# assigned to class no
# from class yes	true positives (tp)	false negatives (fn)
# from class no	false positives (fp)	true negatives (tn)

IR - Example

- Blocking spam e-mails

email	# classified as spam	# classified as not spam
# spam	24 (tp)	1 (fn)
# not spam	70 (fp)	5 (tn)

- Spam precision** - the proportion of spam e-mails that were classified as spam, in the collection of all e-mails that were classified as spam:
 $P=24/(24+70)=25.5\%$, low
- Spam recall** - the proportion of spam e-mails that were classified as spam, in the collection of all spam e-mails:
 $R=24/(24+1)=96\%$, high
- Accuracy** – $(24+5)/(24+1+70+5)=29$, low
- Is this a good result?

R vs P - Extreme Examples

- **Extreme example 1: all e-mails are classified as spam and blocked**

email	# classified as spam	# classified as not spam
# spam	25 (tp)	0 (fn)
# not spam	75 (fp)	0 (tn)

- Spam precision = 25%, Spam recall = 100%, Accuracy = 25%

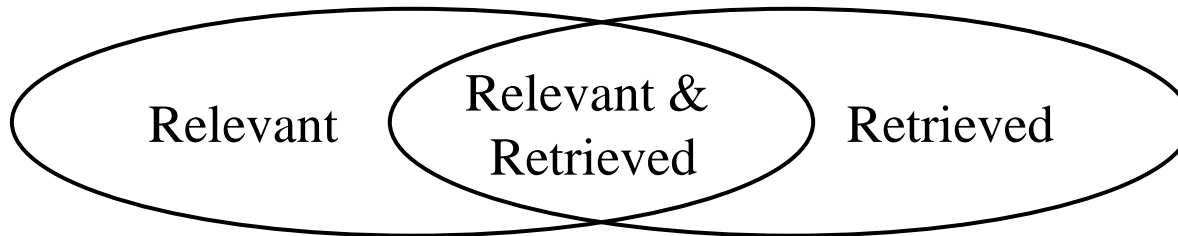
- **Extreme example 2: all but one e-mail are classified as not-spam**

email	# classified as spam	# classified as not spam
# spam	1 (tp)	79 (fn)
# not spam	0 (fp)	20 (tn)

- Spam precision = 100%, Spam recall = 12.5%, Accuracy = 21%
- **Trade-off between R and P - typically we can maximize one of them but not both**

IR Example 2: Text Retrieval

- Given a query (e.g. “Cross validation”), a text retrieval system retrieves a number of documents
- *Retrieved* – the number of all retrieved documents
- *Relevant* – the number of all documents that are relevant



- Recall, Precision and F1 are used to assess the accuracy of the retrieval

$$precision = \frac{Relevant \& Retrieved}{Retrieved}$$

$$recall = \frac{Relevant \& Retrieved}{Relevant}$$

Cost-Sensitive Evaluation

- Misclassification may have different cost
- Which is higher?
 - The cost of misclassifying a legitimate e-mail as spam and blocking it
 - The cost of misclassifying spam e-mail as legitimate
- To reflect the different costs, we can calculate weighed (adjusted) accuracy, precision and F1: blocking a legitimate e-mail counts as X errors (e.g. 10, 100 , etc. – depends on the application)
- Other examples where the cost of different errors is different (most applications)
 - Loan approval: the cost of lending to a defaulter > cost of refusing a loan to a non-defaulter
 - Oil spills detection: the cost of failing to detect oil spills > false alarms
 - Promotional mail: the cost of sending junk mail to a customer who doesn't respond < cost of sending mail to a customer who would respond

Inductive Learning

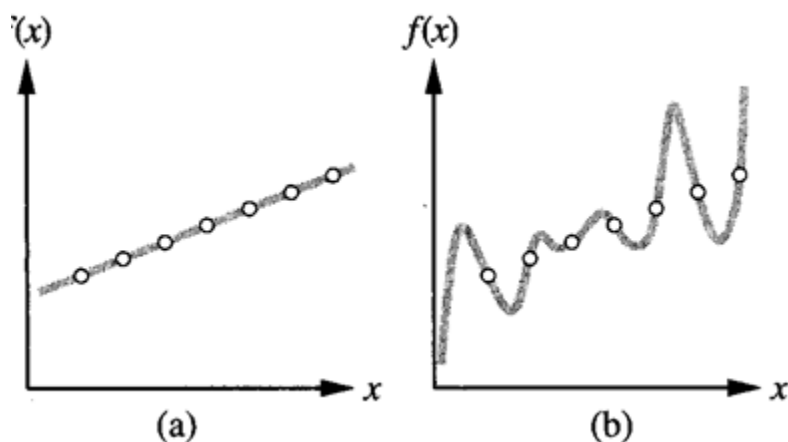
- Actually, why do we need to do empirical evaluation?
- Supervised learning is *inductive* learning
- Induction: inducing the universal from the particular
All apples I have seen are red. \Rightarrow All apples are red.
- Given a set of examples $(\mathbf{x}, f(\mathbf{x}))$,
 - \mathbf{x} is the input vector
 - $f(\mathbf{x})$ is the output of a function f applied to \mathbf{x}
 - we don't know what $f(\mathbf{x})$ is
- Find: a function h (hypothesis) that is a good approximation of f (R&N, p.651)
- Ex.: Given: $([1,1],2), ([2,1],3), ([3,1],4)$, find $h \approx f$

Inductive Learning - Difficulty

- We can generate many hypotheses h
 - the set of all possible hypotheses h form the hypothesis space H
- How to tell if a particular h is a good approximation of f ?
 - A good h will **generalize** well, i.e. will predict new examples correctly
 - That's why we measure its performance on new examples (test set)
- A good h does not necessary imply fitting the given samples x perfectly - we want to extract patterns not to memorize the given data

Which Consistent Hypothesis is Best?

- **Given:** a set of 7 examples $(x, f(x))$, x and $f(x)$ are real numbers
- **Task:** find $h \approx f$ such as
 - h is a function of a single variable x
 - the hypothesis space is the set of polynomials of degree at most k , e.g. $2x+3$ – a degree-1 polynomial; $6x^2+3x+1$ – degree 2
- Consider these 2 solutions:



- Both hypotheses are *consistent* with training data (agree with it, fit the examples perfectly)
- Which one is better? How do we choose from several consistent hypotheses?



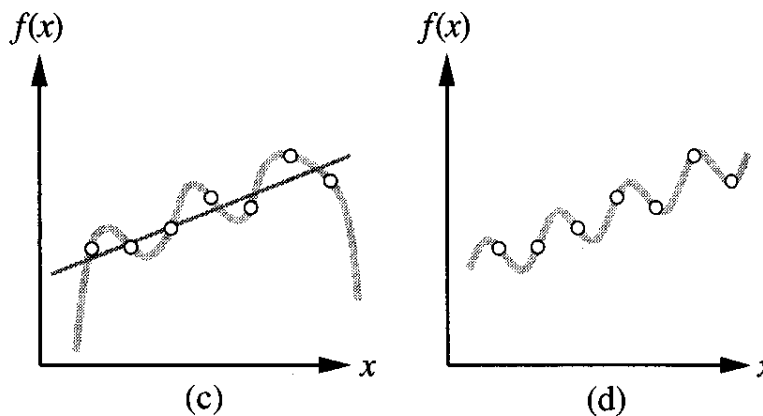
fit with a line
(degree-1)

fit with a degree-7 polynomial

Multiple Consistent Hypotheses

- Recall that we'd like to *extract pattern* from data
- **Ockham razor: prefer the simplest hypothesis consistent with training data**
 - It is also simpler to compute
- **How to define simplicity? Not easy in general.**
 - Easy in this case - obviously degree-1 polynomial (line) is simpler than a higher degree polynomial

Importance of Hypotheses Space



- (d) shows the true function: $\sin(x)$
- In (c) we are searching the hypothesis space H consisting of polynomial functions – \sin is not there

- (c): a degree-1 polynomial function (line) cannot perfectly fit data
- (c): a degree-6 polynomial perfectly fits data but is this a good choice?



- How many parameters? Does it seem to find any pattern in data?

- The choice of hypotheses space is important
 - *The \sin function cannot be learnt accurately using polynomials*
- A learning problem is *realizable* if H contains the true function
- In practice we can't tell if the problem is realizable because we don't know the true hypothesis! (we are trying to learn it from examples)

Empirical Evaluation

- This is the best thing we can do
- Empirical evaluation - review
 - Examples used to build the model are called *training set*
 - Success (how good the fit is) is typically measured on fresh data for which class labels are known (*test data*) as the proportion of correctly classified examples (*accuracy*)