Overview of this lecture

- Organizational
  - Your results + experiences with ES10

- Contents
  - Classification: introduction and examples
  - Probability recap: two crash courses
  - Naïve Bayes: algorithm, example, implementation

- Exercise Sheet 11: learn to predict the genre and rating from a given movie description using Naïve Bayes
Your experiences with ES10

Summary / excerpts

- "seeing the term-pairs was pretty mind-blowing"
- Most of you are starting to appreciate the Linear Algebra...
- ... and numpy
- Performance problems with average_p() from ES 2 ML
- Only marginal improvements with added LSI, very bad results if you only use LSI.
- Many pairs not really synonyms:

  her – he, won – for, awards – for, nominated – best, international – festival, ...
Your experiences with ES10

How to go to bed early?

- **Advantages**: you feel like a functioning member of society again, you can see the sun rise...
- Just get up earlier... painful
- Drugs (red wine, sleeping pills...) unhealthy in the long run
- Go-to-bed-algorithm ... spend 8h debugging it
- Don't post on the forum late at night

- **Sleep cycle reset**: just stay awake until you are in sync again with a socially acceptable sleeping pattern... works, but you will be completely exhausted for some days
But should you go to bed early?

- Each of you has a private internal clock (Circadian rhythm) offset... your Chronotype
- Some flexibility (+/- 2 hours), but anything greater than that can lead to problems
- "Night owls" may be better in intuitive intelligence, creative thinking and inductive reasoning...
- ...but they lag behind early-risers in academic performance
- The human circadian rhythm may actually be not 24h long, but 24h and 11m (free running sleep)

So maybe night owls are just insensitive to external zeitgebers
Classification  1/5

Problem

– Given objects and classes
– Goal: given an object, predict to which class it belongs
– To achieve that, we are given a training set of objects, each labeled with the class to which it belongs
– From that we can (try to) learn which kind of objects belong to which class
Example 1 (natural language text)

- Training set of documents, each labeled with its class

  Flying Saucer Rock n Roll from 1998 is a 12-minute spoof of a 1950s black and white science fiction B-movie ... Comedy

  The Conversation is a 1974 American psychological thriller film written, produced and directed ... Thriller

  Toby the pup in the museum is he first cartoon in a series of twelve. Toby works as a janitor in a museum ... Animation

- Prediction

  Heavy Times one summer afternoon, out of boredom and peer pressure, three best friends go to visit ... which class?
Example 2 (artificial documents)

- Training set of documents, each labeled with its class

<table>
<thead>
<tr>
<th>Document</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>aba</td>
<td>A</td>
</tr>
<tr>
<td>baabaaa</td>
<td>A</td>
</tr>
<tr>
<td>bbaabbab</td>
<td>B</td>
</tr>
<tr>
<td>abbaa</td>
<td>A</td>
</tr>
<tr>
<td>abbb</td>
<td>B</td>
</tr>
<tr>
<td>bbbaab</td>
<td>B</td>
</tr>
</tbody>
</table>

Just two words (a and b, spaces omitted), and two classes

- Prediction

<table>
<thead>
<tr>
<th>Document</th>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>abababa</td>
<td>which class?</td>
</tr>
<tr>
<td>baaaaaaa</td>
<td>which class?</td>
</tr>
</tbody>
</table>
Difference to K-means

- K-means can also be seen as assigning (predicting) a class label for each object ... each cluster = one class

- **Difference 1:** the clusters have no "names"

- **Difference 2:** k-means has no learning phase (where it could learn how objects and classes relate)

  This is called **unsupervised** learning ... in contrast, a method like Naïve Bayes does **supervised** learning

- **Difference 3:** classification methods do soft clustering = for each object, output a probability for each class

  But one often wants only the most probable class
Quality evaluation

- Given a **test set** of labeled documents, and the predictions from a classification algorithm
- For each class $c$ let:
  
  $D_c = \text{documents labeled } c$ (in the test set)
  
  $D'_c = \text{documents classified as } c$ (by the algorithm)
- Then (note that these are **per class**)

  - **Precision** $P := |D'_c \cap D_c| / |D'_c|$
  - **Recall** $R := |D'_c \cap D_c| / |D_c|$
  - **F-measure** $F := 2 \cdot P \cdot R / (P + R)$
Motivation

In this lecture, we will look at Naïve Bayes, one of the simplest (and most widely used) classification algorithms.

Naïve Bayes makes **probabilistic assumptions**

For that, two very basic concepts from probability theory need to be understood:

- Maximum Likelihood Estimation (MLE)
- Conditional probabilities and Bayes Theorem

The following two slides are to refresh your memory concerning both of these
Maximum Likelihood Estimation (MLE)

- Consider a sequence of coin flips, for example
  
  HHTTTTTTTTTTTTTTTTTTTT (5 times H, 15 times T)

- Which Pr(H) and Pr(T) are the most likely?

- Looks like Pr(H) = \( \frac{1}{4} \) and Pr(T) = \( \frac{3}{4} \) ...

\[
\begin{align*}
  x & := P_r(H) \quad P_r(T) = 1 - x \\
  \rho & = P_r(HHHTTTT...T) = x^5 \cdot (1-x)^{15} \quad \text{\textless; maximize this}
\end{align*}
\]

Equiv:

\[
\begin{align*}
  f(x) & = \ln(\rho) = \ln(x^5 \cdot (1-x)^{15}) = 5\ln x + 15\ln(1-x) = 5\ln x + 15\ln(1-x) \\
  f'(x) & = \frac{5}{x} + \frac{15}{1-x} \cdot (-1) = \frac{5}{x} - \frac{15}{1-x} = 0 \quad \Rightarrow 5 - 5x = 75x \\
  & \Rightarrow 5 = 20x \quad \Rightarrow x = \frac{1}{4} 
\end{align*}
\]
Conditional probabilities

Let $A$ and $B$ be events in a probability space $\Omega$

For example, rolling a dice.

$$\Omega = \{1, 2, 3, 4, 5, 6\}$$

$$A = \{2, 4, 6\} \quad \text{"even"}$$

$$B = \{1, 2, 3\} \quad \text{"\leq 3"}$$

$$P_r(A) = \frac{|A|}{|\Omega|} = \frac{3}{6} = \frac{1}{2}$$

$$P_r(B) = \frac{|B|}{|\Omega|} = \frac{3}{6} = \frac{1}{2}$$

$$\frac{|A \cap B|}{|B|} = \frac{1}{3} = P_r(A \mid B)$$
Conditional probabilities ... continued

- Denote by $\Pr(A \mid B)$ the probability of $A \cap B$ in the space $B$

  \[
  \text{(1)} \quad \Pr(A \mid B) := \frac{\Pr(A \cap B)}{\Pr(B)}
  \]

  \[
  \text{(2)} \quad \Pr(A \mid B) \cdot \Pr(B) = \Pr(B \mid A) \cdot \Pr(A)
  \]

- The latter is called **Bayes Theorem**, after Thomas Bayes, 1701 – 1761

- For an intuitive understanding, assume that $\Omega$ is finite, and all $x$ in $\Omega$ equiprobable:

  \[
  \Pr(v(A \mid B)) = \frac{|A \cap B|}{|B|} = \frac{|A \cap B|}{|B|} \cdot \frac{\sqrt{|\Omega|}}{} = \frac{|A \cap B|}{|B|} \cdot \frac{|A \cap B|}{|A|} = \frac{\Pr(A \cap B)}{\Pr(B)}
  \]

  \[
  \Pr(v(B \mid A)) = \frac{|B \cap A|}{|A|} = \frac{|A \cap B|}{|A|} \cdot \frac{\sqrt{|\Omega|}}{} = \frac{|A \cap B|}{|A|} \cdot \frac{|A \cap B|}{|A|} = \frac{\Pr(A \cap B)}{\Pr(A)}
  \]

\[\]
Probabilistic assumption

- Underlying **probability distributions**:
  
  A distribution $p_c$ over the classes ... where $\Sigma_c p_c = 1$

  For each $c$, a distr. $p_{wc}$ over the words ... where $\Sigma_w p_{wc} = 1$

- Naïve Bayes assumes the following process for generating a document $D$ with $m$ words $W_1...W_m$ and class label $C$

  Pick $C=c$ with prob. $p_c$, then pick each word $W_i=w$ with probability $p_{wc}$, independent of the other words

  This is clearly unrealistic (hence the name **Naive** Bayes): e.g. when "Bielefeld" is present, "existence" is less likely
Learning phase

- For a training set $T$ of objects, let:

  - $T_c = \text{the set of documents from class } c$
  - $n_{wc} = \#\text{occurrences of word } w \text{ in documents from } T_c$
  - $n_c = \#\text{occurrences of all words in documents from } T_c$

- We compute the $p_c$ and $p_{wc}$ using simple maximum likelihood estimation (MLE), as explained on Slide 12

\[
p_c := \frac{|T_c|}{|T|} \quad \text{global likeliness of a class}
\]
\[
p_{wc} := \frac{n_{wc}}{n_c} \quad \text{likeliness of a word for a class}
\]
Learning phase, example

Consider Example 2 (artificial documents)

<table>
<thead>
<tr>
<th>Document</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>aba</td>
<td>A</td>
</tr>
<tr>
<td>baabaaa</td>
<td>A</td>
</tr>
<tr>
<td>bbaabbab</td>
<td>B</td>
</tr>
<tr>
<td>abbaa</td>
<td>A</td>
</tr>
<tr>
<td>abbb</td>
<td>B</td>
</tr>
<tr>
<td>bbbaab</td>
<td>B</td>
</tr>
</tbody>
</table>

\( |T_A| = 3\) ; \( |T_B| = 3\) ; \( |T| = 6\) \(\Rightarrow\)

\[
\begin{align*}
|T_A| &= 3 \\
|T_B| &= 3 \\
|T| &= 6
\end{align*}

\(n_A = 15\) \(n_B = 18\)

\(n_{aA} = 10\) \(n_{bA} = 5\)

\(n_{aB} = 6\) \(n_{bB} = 12\)

\(P_A = \frac{|T_A|}{|T|} = \frac{3}{6} = \frac{1}{2}\)

\(P_B = \frac{|T_B|}{|T|} = \frac{3}{6} = \frac{1}{2}\)

\(P_{aA} = \frac{n_{aA}}{n_A} = \frac{10}{15} = \frac{2}{3}\)

\(P_{bA} = \frac{n_{bA}}{n_A} = \frac{5}{15} = \frac{1}{3}\)

\(P_{aB} = \frac{n_{aB}}{n_B} = \frac{6}{18} = \frac{1}{3}\)

\(P_{bB} = \frac{n_{bB}}{n_B} = \frac{12}{18} = \frac{2}{3}\)
**Prediction**

- For a given document \( d \) we want to compute

\[
\Pr(C=c \mid D=d) \quad \text{... for each class} \quad c
\]

The probability of class \( c \), given document \( d \)

- Using Bayes Theorem, we have:

\[
\Pr(C=c \mid D=d) = \Pr(D=d \mid C=c) \cdot \Pr(C=c) / \Pr(D=d)
\]

- Using our (naïve) probabilistic assumptions, we have:

\[
\Pr(D=d \mid C=c) = \Pr(W_1=w_1 \cap \ldots \cap W_m=w_m \mid C=c)
\]

\[
= \prod_{i=1,...,m} \Pr(W_i=w_i \mid C=c)
\]
Prediction ... continued

- We thus obtain that $\Pr(C=c | D=d)$

  $$= \prod_{i=1,...,m} \Pr(W_i=w_i | C=c) \cdot \Pr(C=c) / \Pr(D=d)$$

  $$= \prod_{i=1,...,m} p_{w_i|c} \cdot p_c / \Pr(D=d)$$

  For the product in the front just take the $p_{w_c}$ for all words $w$ in the document and multiply them (if a word $w$ occurs multiple times, also take the factor $p_{w_c}$ multiple times)

- Note that the $\Pr(D=d)$ is the same for all $c$

  We can hence compute the class $c$ with the largest $\Pr(C=c | D=d)$ entirely from the learned $p_{w_c}$ and $p_c$
Prediction, example

Consider Example 2 (artificial documents)

<table>
<thead>
<tr>
<th>Document</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>aba</td>
<td>A</td>
</tr>
<tr>
<td>baabaaa</td>
<td>A</td>
</tr>
<tr>
<td>bbaababb</td>
<td>B</td>
</tr>
<tr>
<td>abbaa</td>
<td>A</td>
</tr>
<tr>
<td>abbb</td>
<td>B</td>
</tr>
<tr>
<td>bbbabaab</td>
<td>B</td>
</tr>
</tbody>
</table>

Recall from training:

- $p_A = p_B = 1/2$
- $p_{aA} = 2/3$
- $p_{bA} = 1/3$
- $p_{bB} = 2/3$

Let us predict the class for $aab$ ... A or B?

\[
q_A = \frac{2}{3} \times \frac{2}{3} \times \frac{1}{3} \times \frac{1}{2} / Pr(D = aab)
\]

\[
q_B = p_{aB} \times p_{aB} \times p_{bB} \times p_B / Pr(D = aab) = \frac{1}{3} \times \frac{1}{3} \times \frac{2}{3} \times \frac{1}{2} / Pr(D = aab)
\]
Smoothing

- Problem: when only one \( p_{wc} = 0 \), then \( \Pr(C=c \mid D=d) = 0 \).

  This happens rather easily, namely when \( d \) contains a word that did not occur in the training set for class \( c \).

- Therefore, during training we actually compute

\[
p_{wc} := \frac{n_{wc} + \varepsilon}{n_c + \varepsilon \cdot \# \text{vocabulary}}
\]

This is like adding every word \( \varepsilon \) times for every class.

For ES11, take \( \varepsilon = 1/10 \).
Smoothing ... continued

- What about $p_c = 0$ for a class $c$?

  This means, that $|T_c| = 0$, that is, there was no document from class $c$ in the training set.

- When $p_c = 0$, then $\Pr(C=c \mid D=d) = 0$ for any document $d$.

  But that is reasonable: if we did not see any document from a particular class $c$ during training, we can learn nothing for that class, and we cannot meaningfully predict it.

So no smoothing needed for that case.
Naive Bayes  9/11

- **Numerical stability**
  - Problem: a product of many small probabilities quickly becomes zero due to limited precision on the computer.

  For example, the smallest positive number that can be represented with an 8-byte double is $\approx 10^{-308} \approx 2^{-1024}$

  Then multiplying 52 probabilities $< 10^{-6}$ is already zero.

- Compute the \textbf{log}-probabilities! ... then products of probabilities translate into sums of log-probabilities.

  Instead of comparing $q_A$ and $q_B$, let's compare $\log(q_A)$ and $\log(q_B)$.

  $\log \prod_i p_i = \sum_i \log p_i$
Some possible refinements

- Instead of words, we could take any other quantifiable aspect of a document as so-called "feature"
  
  For example, also consider all (two-word) phrases

- Omit non-predictive words like "and"
  
  For example, omit the most frequent words

- In training, replace the word frequencies $n_{wc}$ by $tf.idf_{wc}$
  
  And correspondingly, replace $n_c$ by $\sum_w tf.idf_{wc}$

- For ES11, none of these are required ... but feel free to play around with them
Linear algebra (LA)

- Assume the documents are given as a term-document matrix, like we have seen it many times now.

For ES11, we provide you with the code to construct the document-term matrix with simple tf entries.

- Then all the necessary computations can again be done very elegantly and efficiently using matrix operations.

Whenever you have to compute a large number of (weighted) sums in a uniform manner, this calls for LA.

However, if you feel more comfortable with (boring and inefficient) for-loops, you can use those for ES11 too.
References

- Further reading
  - Textbook Chapter 13: Text classification & Naive Bayes
  - Advanced material on the whole subject of learning
    [Elements of Statistical Learning, Springer 2009](http://en.wikipedia.org/wiki/Naive_Bayes_classifier)

- Wikipedia