Information Retrieval

WS 2015 / **2016**

Lecture 10, Tuesday January 12th, 2016

(Classification, Naive Bayes)

Prof. Dr. Hannah Bast
Chair of Algorithms and Data Structures
Department of Computer Science
University of Freiburg

UNI FREIBURG

Overview of this lecture

- Organizational
  - Your results + experiences with ES9 k-means
  - Date and place for the **exam**
- Contents
  - Classification introduction and examples
  - Probability recap two one-slide crash courses
  - Naïve Bayes algorithm, example, implementation
  - Exercise Sheet 10: learn to predict the **genre** and **rating** from a given movie description using Naïve Bayes

Your experiences with ES9

- Summary / excerpts
  - Quite hard and time-consuming for many of you
  - Bug in sparse normalization we provided

    Was pointed out on the **forum** and then fixed soon
  - Using numpy / scipy in the right way was non-trivial
  - Easy to make small mistakes which are hard to debug
  - Still lots of problems getting the linear algebra right
  - For the geometric toy example from the lecture, normalizing the points gives different clusters … was explained in **forum**
  - Great support on the **forum**, at all times

Your results for ES9

- For our dataset (≈ 200.000 docs, 50 clusters)
  - Relatively few iterations (20 - 30) are enough
  - Pretty fast, even with Python (≈ 1 second / iteration)

    That is the power of linear algebra: two years ago, even the C++ implementations were 10 x slower on a smaller dataset
  - Some centroids are meaningful, others not so much:

    written silent british comedy American starring frank

    it was at won nominated awards academy award best

    s an the her in to of story about who

    jung jin park korea soo ki lee kim south korean

Exam

- Written exam
  - For all **except** the B.Sc. Computer Science students
  - Date: Tuesday, February 23, 2016   14 – 16 h
  - Place: HS026 … and maybe also HS036

    Depends on the number of participants
- Oral exam
  - **Only** for the B.Sc. Computer Science students
  - Date: Wednesday, February 24, 2016, afternoon

    A time slot will be allocated to you by the Prüfungsamt
  - Place: my office (building 51, second floor, room 28)

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
  - Two examples on the next two slides

Classification 2/5

- 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

    Tainted is a 1988 low-budget suspense drama about a school teacher married to the owner of a crematorium … 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?**

Classification 3/5

- Example 2 (artificial documents)
  - Training set of documents, each labeled with its class

aba	A
baabaaa	A
bbaabbab	B
abbaa	A
abbb	B
bbbaab	B

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

  - Prediction

abababa	which class?
baaaaaa	which class?

Classification 4/5

- 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

Classification 5/5

- Quality evaluation
  - Given a **test set** of labeled documents, and the predictions from a classification algorithm
  - For each class c let:

    D_c = documents labeled c (in the test set)

    D'_c = 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 $F := 2 \cdot P \cdot R \,/\, (P + R)$

    Note that $P = R = F = 100\%$ if and only if $D_c = D'_c$

Probability recap 1/3

- 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

Probability recap 2/3

head

tail

- Maximum Likelihood Estimation (MLE)
  - Consider a sequence of coin flips, for example

    HHTTTTTTTHTTTTTHTTHTT (5 times H, 15 times T)
  - Which Pr(H) and Pr(T) are the most likely?
  - Looks like Pr(H) = ¼ and Pr(T) = ¾ ... let's prove this

$x := Pr(H)$, then $Pr(T) = 1 - x$

Note: ln is a monotone function

$Pr(HHTT \ldots HTT) = x^5 \cdot (1-x)^{15}$

Find x such that $x^5 \cdot (1-x)^{15}$ is maximized

Equiv.: such that $f(x) := \ln(x^5 \cdot (1-x)^{15})$ is max.

$= 5 \cdot \ln x + 15 \cdot \ln(1-x)$

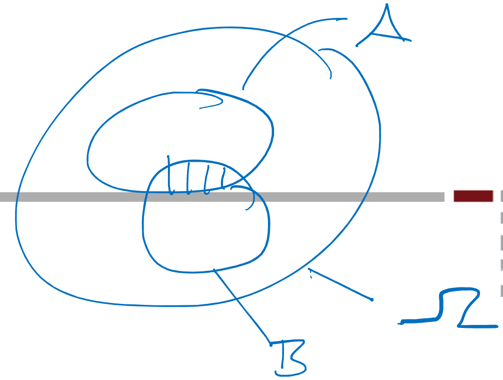
$f'(x) = \frac{5}{x} - \frac{15}{1-x} \stackrel{!}{=} 0 \Rightarrow 5(1-x) = 15 \cdot x$

$\Rightarrow 5 = (5+15) \cdot x \Rightarrow Pr(H) = x = \frac{5}{5+15} = \frac{1}{4}$

$\Rightarrow Pr(T) = \frac{15}{5+15} = \frac{3}{4}$

$Pr(A) = \frac{|A|}{|\Omega|} = \frac{3}{6} = \frac{1}{2}$ $Pr(A|B) = \frac{|A \cap B|}{|B|} = \frac{1}{3}$

Probability recap 3/3



E.g. rolling a dice
$\Omega = \{1,2,3,4,5,6\}$
$A = \{2,4,6\}$ „even"
$B = \{1,2,3\}$ „≤ 3"

- Conditional probabilities
  = subsets of Ω
  - Let A and B be events in a probability space Ω
  - Denote by Pr(A | B) the probability of A ∩ B in the space B

    (1) Pr(A | B) := Pr(A n B) / Pr (B)

    (2) $\underbrace{Pr(A \mid B) \cdot Pr(B)}_{= Pr(A \cap B)} = \underbrace{Pr(B \mid A) \cdot Pr(A)}_{= Pr(A \cap B)}$

  - The latter is called **Bayes Theorem**, after Thomas Bayes, 1701 – 1760

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

$$Pr(A|B) = \frac{|A \cap B|}{|B|} = \frac{|A \cap B| \setminus |\Omega|}{|B| \setminus |\Omega|} = \frac{Pr(A \cap B)}{Pr(B)}$$

$$Pr(B|A) = \frac{|A \cap B|}{|A|} = \frac{|A \cap B| \setminus |\Omega|}{|A| \setminus |\Omega|} = \frac{Pr(A \cap B)}{Pr(A)}$$

Naive Bayes 1/11

- 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 $W_1 \ldots W_m$ 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 "relativity" is present, "theory" is more likely
  - Anyway, this gives us something well-defined

Naive Bayes 2/11

- Learning phase
  - For a **training set** T of objects, let:

    T_c = the set of documents from class c

    n_{wc} = #occurrences of word w in documents from T_c

    n_c = #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 10

    $p_c := |T_c| / |T|$ global likeliness of a class

    $p_{wc} := n_{wc} / n_c$ likeliness of a word for a class

    Beware: n_{wc} and hence p_{wc} are often zero … see slide 20

Naive Bayes 3/11

- Learning phase, example
  - Consider Example 2 (artificial documents)

aba	A
baabaaa	A
bbaabbab	B
abbaa	A
abbb	B
bbbaab	B

this is what NB „learns" (and remembers for prediction)

$|T_A| = 3, \quad |T_B| = 3, \quad |T| = 6 \Rightarrow p_A = p_B = \frac{3}{6} = \frac{1}{2}$

$n_{aA} = 10, \quad n_{bA} = 5, \quad n_A = 15 \Rightarrow p_{aA} = \frac{2}{3}, \quad p_{bA} = \frac{1}{3}$

$n_{aB} = 6, \quad n_{bB} = 12, \quad n_B = 18 \Rightarrow p_{aB} = \frac{1}{3}, \quad p_{bB} = \frac{2}{3}$

Naive Bayes 4/11

- Prediction
  - For a given document d we want to compute

    Pr(C=c | D=d) … for each class c

    The probability of class c, given document d
  - Using Bayes Theorem, we have:

    $\Pr(C=c \mid D=d) = \underline{\Pr(D=d \mid C=c)} \cdot \Pr(C=c) / \Pr(D=d)$
  - Using our (naïve) probabilistic assumptions, we have:

    $\underline{\Pr(D=d \mid C=c)} = \Pr(W_1=w_1 \cap \ldots \cap W_m=w_m \mid C=c)$

    „naive“ assumption $= \Pi_{i=1,\ldots,m} \Pr(W_i=w_i \mid C=c)$

Naive Bayes 5/11

- Prediction ... continued
  - We thus obtain that $\Pr(C=c \mid D=d)$

    $= \Pi_{i=1,\ldots,m} \Pr(W_i=w_i \mid C=c) \cdot \Pr(C=c) / \Pr(D=d)$

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

    this part is enough

    For the product in the front just take the p_{wc} for all words w in the document and multiply them (if a word w occurs multiple times, also take the factor p_{wc} multiple times)
  - Note that the $\Pr(D=d)$ is the same for all c

    We can hence compute the class c with the largest $\Pr(C=c \mid D=d)$ entirely from the learned p_{wc} and p_c

Naive Bayes 6/11

- Prediction, example

  $q_A > q_B$

  $\Rightarrow$ predict A !

  - Consider Example 2 (artificial documents)

aba	A
baabaaa	A
bbaabbab	B
abbaa	A
abbb	B
bbbaab	B

Recall from the training:

$p_A = p_B = \frac{1}{2}$

$p_{aA} = \frac{2}{3}$, $p_{bA} = \frac{1}{3}$

$p_{aB} = \frac{1}{3}$, $p_{bB} = \frac{2}{3}$

another example:

ababababa $\rightarrow$?

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

class

$q_A := \Pr(C = A \mid D = aab) = p_{aA} \cdot p_{aA} \cdot p_{bA} \cdot p_A / \Pr(D = d)$

$= \frac{2}{3} \cdot \frac{2}{3} \cdot \frac{1}{3} \cdot \frac{1}{2} / \Pr(D = d)$

$q_B := \Pr(C = B \mid D = aab) = p_{aB} \cdot p_{aB} \cdot p_{bB} \cdot p_B / \Pr(D = d)$

$= \frac{1}{3} \cdot \frac{1}{3} \cdot \frac{2}{3} \cdot \frac{1}{2} / \Pr(D = d)$

Naive Bayes 7/11

this is just $\Sigma_w (n_{wc} + \varepsilon)$

- 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

    $p_{wc} := (n_{wc} + \varepsilon) / (n_c + \varepsilon \cdot \#vocabulary)$

    This is like adding every word ε times for every class

    For ES10, take ε = 1/10 … for short docs, a larger ε would add too much noise

Naive Bayes 8/11

- 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

$\log \prod_i p_i$
$= \sum_i \log p_i$ ← MUCH nicer

- 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 ≈ 10^{-308} , $\approx 2^{-1024}$

    Then multiplying 52 probabilities < 10^{-6} is already zero
  - Therefore, compute the **log**-probabilities … then products of probabilities translate into sums of log-probabilities

    Log-probabilities also give you the most likely class, because log is a monotone function

    Beware: don't take exp in the end !

    $\exp(-1000) = 0$
    (with a double)

Naive Bayes 10/11

- 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 Σ_w $tf.idf_{wc}$
  - For ES10, none of these are required … but feel free to play around with them

Naive Bayes 11/11

- Linear algebra (LA)
  - Assume the documents are given as a term-document matrix, like we have seen it many times now

    For ES10, 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 ES10 too

References

- Further reading
  - Textbook Chapter 13: Text classification & Naive Bayes

    http://nlp.stanford.edu/IR-book/pdf/13bayes.pdf
  - Advanced material on the whole subject of learning

    Elements of Statistical Learning, Springer 2009
- Wikipedia
  - http://en.wikipedia.org/wiki/Naive_Bayes_classifier
  - http://en.wikipedia.org/wiki/Bayes'_theorem
  - http://en.wikipedia.org/wiki/Maximum_likelihood