Information Retrieval

WS 2013 / 2014

Lecture 10, Tuesday January 14th, 2014

(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 Ex. Sheet 9 (k-means)
  - Date for the exam: **Wednesday, February 19, 2014**

    Time: 14 – 16 h, Room: to be announced
- Classification using Naive Bayes
  - Like clustering, but **learns** from a training set
  - This is then called **classification**
  - Naive Bayes is one of the simplest classification methods
  - Exercise Sheet 10: Classify the documents from ES#10 (100K articles about people) using Naive Bayes

Experiences with ES#9 (k-means)

- Summary / excerpts last checked January 14, 15:30
  - Ok conceptually, but quite challenging in the details
  - The difficulty is not k-means, but treating documents as objects of which one can compute the average
  - Can be parallelized very well; one student implemented a multi-threaded version: k threads → almost k times faster
  - Good thing that we made no new year's resolution … we would have failed them already
  - Point distribution is uneven sometimes, and so is the distribution of the level of detail in the TIP file
  - Many of you have time stress it seems

Your results for ES#9 (k-means)

UNI FREIBURG

- For our dataset (100.000 docs, 1000 terms, 50 clusters)
  - Relatively few iterations (10 – 20) are enough
  - A single iteration is quite time-intensive (10-20 seconds)
  - Typical RSS was around 68.500, that is, 0.68 per document, that is, an average score difference of 0.03 per term
  - For many centroids, words belong to same intuitive "topic"

    chinese china hong kong dynasty han republic li zhou people

    singer songwriter music pop american is an album born albums
  - For some centroids, the similarity is of a different kind

    his he to in as of with on that by (all frequent)

    irish ireland o dublin dála teachta td an fianna fáil (same language)

Naive Bayes 1/10

- High-level view
  - Given a set of **objects** and a set of **classes**
  - For each object from a given so-called **training set**, we know to which class it belongs
  - Learn from this training set, and then predict the class for arbitrary other objects, from a so-called **testing set**
- Difference to K-means
  - Naive Bayes is **supervised** = gets some input to learn from; K-means is **unsupervised** = gets no such input
  - Naive Bayes does **soft clustering** = each object may be assigned to more than one class

    Typically, one is only interested in the "top" class though

Naive Bayes 2/10

- Example
  - Training set of documents with known class

    Thomas Houldsworth was a Tory, and then Conservative Party, politician in England. He was a Member of Parliament (MP) for 34 years, … **Politician**

    Ann May was a silent film star who made motion pictures from 1919 - 1925. Her given name was Anna Max and she was born in Cincinnati, Ohio. **Actor**

  - Testing set of documents, predict class for each

    George Siegmann was an American actor in the silent film era. He is listed as having been in over 100 films. **which class ?**

    Harvey McLane was a Canadian provincial politician. He was the Liberal member of … **which class?**

Naive Bayes 3/10

- Three basic steps
  - **Step 1:** **represent** each object as a vector

    We take one dimension per word in a document … next slide

    In the context of learning, these are often called feature vectors (each dimension = one feature)

  - **Step 2:** **learn** how "likely" each feature is for each class, e.g.

    Prob(film | Actor) = 0.05
    Prob(parliament | Actor) = 0.01

  - **Step 3:** **predict**, using the probabilities from Step 2, how likely a class is for a given feature vector

    Prob(Politician | Document on George Siegmann) = 0.8
    Prob(Actor | Document on Georges Siegmann) = 0.2

Naive Bayes 4/10

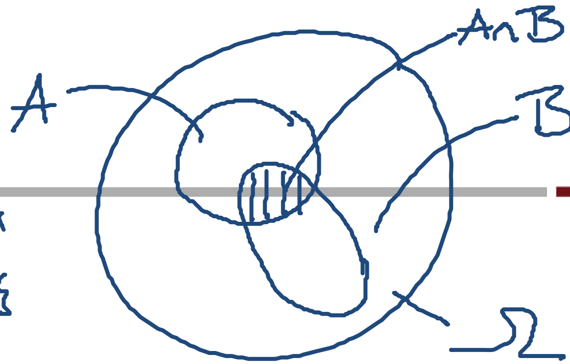
- Probabilistic model ... so that the "likely" becomes precise
  - We assume the following random process for generating a document with m words

    Pick class c with probability p_c ... where $\Sigma_c\, p_c = 1$

    Pick the i-th word as w with prob. p_{cw} ... where $\Sigma_w\, p_{cw} = 1$
  - Each word is picked independently of the other words

    This is clearly unrealistic (hence the name **Naive** Bayes): e.g. when "relativity" is present, "theory" is more likely
  - However unrealistic, these assumptions give us well-defined probabilities to compute with ...

Conditional Probabilities



e.g. $\Omega = \{1,2,3,4,5,6\}$ "rolling a dice" $A = \{2,4,6\}$ "even number"

- A one-slide crash course $B = \{1,2,3\}$ "number ≤ 3"
  - Let A and B be events in a probability space Ω
  - Denote by Pr(A | B) the probability of A n B in the space B

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

(2) Pr(A | B) · Pr(B) = Pr (B | A) · Pr(A)

  - 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:

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

$$Pr(A|B) = \frac{|A \cap B|}{|B|} = \frac{|A \cap B| / |\Omega|}{|B| / |\Omega|} = \frac{Pr(A \cap B)}{Pr(B)} \Rightarrow Pr(A \cap B) = Pr(B) \cdot Pr(A|B)$$

$$Pr(B|A) = \frac{Pr(A \cap B)}{Pr(A)} \Rightarrow Pr(A \cap B) = Pr(A) \cdot Pr(B|A)$$

(2)

Maximum Likelihood Estimation (MLE)

- Another one-slide crash course
  - Consider a sequence of coin flips, for example

    HHTTTTTTHTTTTTHTTHTT (5 times H, 15 times T)

    Heads

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

$p = Pr(H)$, $1-p = Pr(T)$

$m_H = \#\text{Heads} = 5$

$m_T = \#\text{Tails} = 15$

$\Rightarrow Pr(HHTT\ldots TT) = p^{m_H} \cdot (1-p)^{m_T}$

For which p is this probability maximal

Let us maximize instead $f(p) = \ln\left(p^{m_H} \cdot (1-p)^{m_T}\right)$

OK, since ln is monotone

$= m_H \cdot \ln p + m_T \cdot \ln(1-p)$

$f'(p) = \frac{m_H}{p} - \frac{m_T}{1-p} \overset{!}{=} 0 \Rightarrow \frac{m_H}{p} = \frac{m_T}{1-p} \Rightarrow (1-p)\, m_H = p \cdot m_T$

$\Rightarrow m_H = p \cdot (m_H + m_T) \Rightarrow p = \frac{m_H}{m_H + m_T}$ ☒

Naive Bayes 5/10

- Step 2: learning from a training set
  - We need to compute the following "prior" probabilities

    $\Pr(C = c)$ (global likeliness of a class)

    $\Pr(W = w \mid C = c)$ (likeliness of a feature for a class)
  - For a training set T of objects let

    T_c be 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
  - Then we compute the priors as follows using **MLE**

    $\mathbf{Pr(C = c)} := |T_c| / |T|$ note that $\Sigma_c |T_c| = |T|$

    $\mathbf{Pr(W = w \mid C = c)} := n_{wc} / n_c$ note that $\Sigma_w n_{wc} = n_c$

BEWARE: n_{wc} is zero quite often, see slide 14

Naive Bayes 6/10

- Step 3: prediction based on the learned priors
  - For a document D we want to compute for each class c

    $Pr(C = c \mid W_1 = w_1$ n ... n $W_m = w_m)$

    where w_i is the value of the i-th feature (word) of D
  - Using Bayes Theorem, we can prove (next slide) that

    $Pr(C = c \mid W_1 = w_1$ n ... n $W_m = w_m)$ **=** p'_c / P

    where p'_c = $\mathbf{Pr(C = c)} \cdot \Pi_{i=1,\ldots,m}$ $\mathbf{Pr(W_i = w_i \mid C = c)}$

    and $P = \Sigma_c \, p'_c$

Naive Bayes 7/10

$\Pr(A) \cdot \Pr(B|A) = \Pr(B) \cdot \Pr(A|B)$

$\Pr(B|A) = \frac{\Pr(B)}{\Pr(A)} \cdot \Pr(A|B)$

- Proof of $\Pr(C = c \mid W_1 = w_1 \cap \ldots \cap W_m = w_m) = p'_c / P$
  - where $p'_c = \Pr(C = c) \cdot \Pi_{i=1,\ldots,m} \Pr(W_i = w_i \mid C = c)$
  - and $P = \Sigma_c\, p'_c$

$$\Pr(C=c \mid W_1 = w_1 \cap \ldots \cap W_m = w_m)$$

$$\overset{\text{Bayes Term.}}{=} \frac{\Pr(C=c)}{\underbrace{\Pr(W_1=w_1 \cap \ldots \cap W_m = w_m)}_{=:P}} \cdot \Pr(W_1 = w_1 \cap \ldots \cap W_m = w_m \mid C = c)$$

$$\overset{\text{Indep. Ass.}}{=} \frac{1}{P} \cdot \underbrace{\Pr(C=c) \cdot \prod_{i=1}^{m} \Pr(W_i = w_i \mid C = c)}_{=:p'_c} \quad \blacksquare$$

For finding the most likely c, suffices to look at p'_c, since $\frac{1}{P}$ is indep. of c.

Naive Bayes 8a/10

- Important implementation advice 1/2
  - **Problem 1:** when only one of the $\Pr(W = w \mid C = c)$ is zero, the whole product is zero, and c will be out of the game

    Therefore, instead of $\Pr(W = w \mid C = c) := n_{wc} / n_c$ do

    $\Pr(W = w \mid C = c) :=$ **$(n_{wc} + \varepsilon) / (n_c + \varepsilon \cdot \#\text{vocabulary})$**

    This is like adding every word ε times for every class

    For ES#10, take $\varepsilon = 1/10$

    Our docs are short, so a larger ε would add too much noise

    Note: when $\Pr(C = c) = 0$, the whole product is also zero, and c will be out of the game; but that is **ok**, since this only happens if there was no doc from class c in the training set

Naive Bayes 8b/10

$\log \prod_i p_i = \sum_i \log p_i$

e.g. $p_i = \frac{1}{1000}$

$\Rightarrow \log_{10} p_i = -1000$

- Important implementation advice 2/2
  - **Problem 2:** A product of many small probabilities quickly becomes zero due to limited precision on the computer

    Therefore, instead of $\mathbf{\Pi_i}\ p_i$ compute $\mathbf{\Sigma_i} \log p_i$

    This also gives you the most likely class, because log is a monotone function

    In particular, don't take exp in the end, since already exp(-1000) is zero on most computers

Naive Bayes 9/10

- An small but complete example

  $\varepsilon := 0$ for this EXAMPLE

  – 6 documents, only words are a or b, 2 classes: A and B

Doc 1: aba	class A
Doc 2: baabaaa	class A
Doc 3: bbaabbab	class B
Doc 4: abbaa	class A
Doc 5: abbb	class B
Doc 6: bbbaab	class B

TRAINING:

$n_A = n_B = 3 \Rightarrow Pr(A) = Pr(B) = \frac{3}{6} = \frac{1}{2}$

$n_{aA} = 10,\ n_{bA} = 5 \quad 10+5 = 15$

$n_{aB} = 6,\ n_{bB} = 12 \quad 6+12 = 18$

$\Rightarrow Pr(a|A) = \frac{10}{15} = \frac{2}{3}$

$Pr(b|A) = \frac{5}{15} = \frac{1}{3}$

$Pr(a|B) = \frac{6}{18} = \frac{1}{3}$

$Pr(b|B) = \frac{12}{18} = \frac{2}{3}$

PREDICT:

say Doc: aab → A or B ?

class A: $Pr(A) \cdot Pr(a|A)^2 \cdot Pr(b|A)$

$= p'_A = \frac{1}{2} \cdot \left(\frac{2}{3}\right)^2 \cdot \frac{1}{3} = \frac{4}{2 \cdot 3^3}$

class B: $Pr(B) \cdot Pr(a|B)^2 \cdot Pr(b|B)$

$= p'_B = \frac{1}{2} \cdot \left(\frac{1}{3}\right)^2 \cdot \frac{2}{3} = \frac{2}{2 \cdot 3^3}$

$\Rightarrow p'_A > p'_B$

$\Rightarrow$ predict A ☐

Naive Bayes 10/10

- Feature Design
  - In our example: one feature for each word in the doc.
  - Alternative: feature vector of size M, M = #vocab.
  - Other alternatives: pick all 3-grams, consider word positions, consider part-of-speech tags (verb, noun, …)
- Feature Selection
  - Some words are not very predictive, like "and"
  - Considering them adds unnecessary noise to our decision
  - One simple remedy: remove very frequent (stop) words

    For ES#10, simply take all words though

Quality Evaluation

- How do we measure how good our classification is?
  - For each class c we do the following
  - Let D_c = #documents from class c (ground truth)
  - Let D'_c = #documents classified as c
  - Then, as usual (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 $D_c = D'_c$

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