# Information Retrieval

## WS 2013 / 2014

### Lecture 11, Tuesday January 21st, 2014

### (Support Vector Machines)

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

# Overview of this lecture

- Organizational
  - Your results + experiences with Ex. Sheet 10 (Naive Bayes)
  - The **oral exams** (ONLY for the computer science bachelor students) are on February 21 + March 27 (you can choose)
- Support Vector Machines (SVMs)
  - Another linear classifier, just like Naive Bayes
  - But with a different objective function (max. margin size)
  - Some more linear algebra … you will love it
  - We will play around with the SVM Light software
  - **Exercise Sheet 11:** Prove that Naive Bayes is a linear classifier + compute the margin size on the given dataset

# Experiences with ES#10 (Naive Bayes)

- Summary / excerpts last checked January 21, 15:50
  - Easier again then the last sheet
  - Very interesting topic
  - Most of you have time issues and start late
  - What kind of questions in the oral exams ... see next slide

# Your results for ES#10 (Naive Bayes)

- For our dataset (50.562 docs, 10 classes)
  - Training time (10% of the docs): around 0.1 seconds
  - Prediction time (90% of the docs): around 1 second
  - **Bottom line 1:** Naive Bayes is quite efficient, namely: 50K documents / second for both training and prediction
  - The precision is between 80% and 90%
  - **Bottom line 2:** Sounds good, but without having seen other methods, it's hard to tell how good exactly

    Today we will see a comparison with SVMs

# Exam (applies to both written and oral exam)

- Three types of questions
  - **Type 1:** Do the steps of an algorithm, or a variant thereof, like we did in the lecture
  - **Type 2:** Write a small program, or understand what a given small program does
  - **Type 3:** Small calculations or proofs
  - The emphasis is on (basic) understanding, not on learning things by heart

    You can use course materials during the exam
  - To prepare, understand how it was done in the lecture, then put the solution away, then **try do to it yourself**

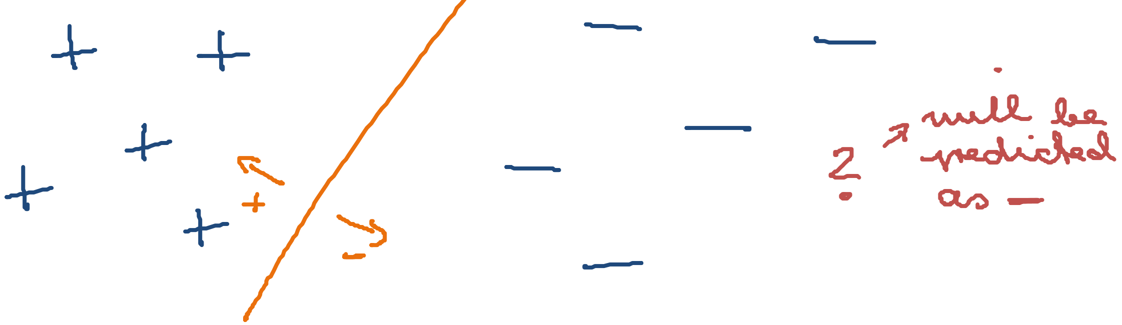
    If you did the exercises, not much left to do for you

# Linear Classifiers 1/5

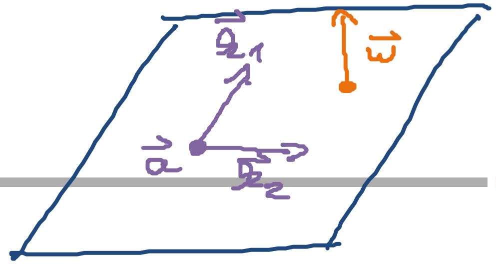
we assume two classes + and –

- **Informally**
  - Assume the objects are points in d dimensions
  - Let's assume we have only two classes for now
  - A linear classifier tries to separate the data points by a (d-1)-dimensional **hyperplane** … definition on next slide

    For d = 2 this means: try to separate by a **straight line**
  - Predictions are made based on which side of the hyperplane / straight line the object lies on
  - Note: points in the training set may not be separable



# Linear Classifiers 2/5




- Formal definition of a hyperplane

  dot product

  - A hyperplane H in $\mathbf{R}^d$ if defined by an anchor point $a \in \mathbf{R}^d$, and linearly independent $h_1, \ldots, h_{d-1}$ and consists of all linear combinations $a + \Sigma_i \alpha_i h_i$ for arbitrary $\alpha_1, \ldots, \alpha_{d-1} \in \mathbf{R}$
  - **Lemma:** For each such H, there exists a $w \in R^d$ orthogonal to $h_1, \ldots, h_{d-1}$ and $b \in \mathbf{R}$ such that $H = \{ x \in \mathbf{R}^d : w \bullet x = b \}$
  - **Proof:** Pick any $w$ orthogonal to $h_1, \ldots, h_{d-1}$ and let $b = w \bullet a$ Then we can show that $x \in H \Leftrightarrow w \bullet x = b$

"$\Rightarrow$": $x \in H \Rightarrow x = a + \sum_{i=1}^{d-1} \alpha_i h_i \Rightarrow w \cdot x = \underbrace{w \cdot a}_{= w \cdot a\, = b} + \sum_{i=1}^{d-1} \alpha_i \underbrace{w \cdot h_i}_{=0 \text{ because } w \perp h_i}$

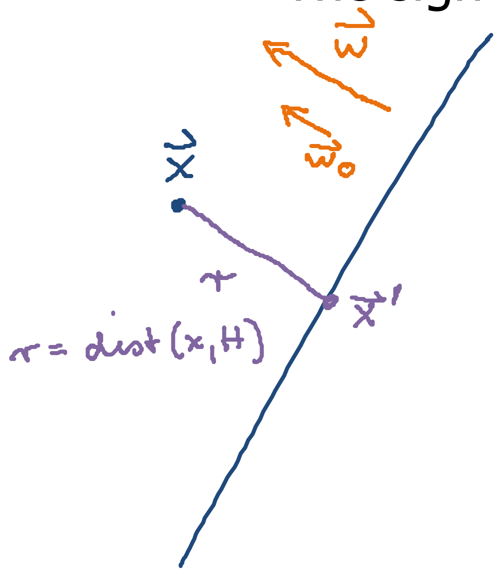
"$\Leftarrow$": $w \cdot x = b \Rightarrow \sum_{i=1}^{d-1} \alpha_i \underbrace{w \cdot h_i}_{=0} + \alpha_d \cdot |w|^2 = b$

Since $x \in \mathbb{R}^d$ and $h_1, \ldots, h_{d-1}, w$ are lin. indep. we can write $x = \sum_{i=1}^{d-1} \alpha_i h_i + \overbrace{\alpha_d \cdot w}^{=a} \Rightarrow x \in H$

$\Rightarrow \alpha_d \cdot |w|^2 = w \cdot a \Rightarrow \alpha_d \cdot w \cdot \cancel{|w|^2} = \cancel{|w|^2} \cdot a$

# Linear Classifiers 3/5

- Distance from a point to a hyperplane
  - Let $H = \{ x \in \mathbf{R}^d : w \bullet x = b \}$ be a hyperplane in $\mathbf{R}^d$
  - Then the distance of a point $x \in \mathbf{R}^d$ to $H$ is $|w \bullet x - b| / |w|$
  - The sign of $w \bullet x - b$ says on which side of $H$ lies $x$




CASE 1: $\vec{x}$ is on the side of H where $\vec{w}$ points

$w_0 = w/|w|$, so $|w_0| = 1$

$\Rightarrow x = x' + r \cdot w_0 \Rightarrow w \cdot x = \underbrace{w \cdot x'}_{=b} + r \cdot \underbrace{w \cdot \overset{w/|w|}{w_0}}_{=|w|}$

because $x' \in H$

$\Rightarrow w \cdot x = b + r \cdot |w|$

$\Rightarrow r = (w \cdot x - b)/|w|$

CASE 1: $\vec{x}$ is on the opposite side of H

$\Rightarrow x = x' - r \cdot w_0 \Rightarrow \dots$

$\Rightarrow r = -(w \cdot x - b)/|w|$ ■

# Linear Classifiers 4a/5

- Two-class Naive Bayes (NB) is a linear classifier
  - Recall how NB predicts the probability of a class C for d

    $\Pr(C \mid d) = \Pr(C) \cdot \Pi_{i=1,\ldots,|d|} \Pr(w_i \mid C)$, $|d|$ = #words in d

    where $w_i$ is the i-th word of d

    Note: #features = #words in the document

    Note: if $f_i = 0$, then $\Pr(w_i \mid C)^{f_i} = 1$

  - We can equivalently write this as

    $\Pr(C \mid d) = \Pr(C) \cdot \Pi_{i=1,\ldots,|V|} \Pr(w_i \mid C)^{fi}$, V = vocabulary

    where $w_i$ is the i-th word in V, and fi = #occ of $w_i$ in d

    Note: #features = size of the vocabulary

# Linear Classifiers 4b/5

- Two-class Naive Bayes (NB) is a linear classifier
  - **Lemma:** Assume our two classes are called $A$ and $B$, and define $b \in \mathbf{R}$ and $w \in \mathbf{R}^{|V|}$ as follows:

    $b = -\log_2 (\Pr(A) / \Pr(B))$, $w_i = \log_2 (\Pr(w_i \mid A) / \Pr(w_i \mid B))$

    Then NB predicts $A$ for $x$ if $w \bullet x - b > 0$, and $B$ otherwise
  - **Proof: Exercise 11.1**

    This is a good exercise for understanding the linear algebra behind linear classifiers. It's not hard, but you have to understand the basic concepts, so perfect exercise :-)

# Linear Classifiers 5a/5

- The toy example from our last lecture again:

| | | |
|---|---|---|
| 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 (recap):

$Pr(A) = Pr(B) = \frac{1}{2}$

$Pr(a|A) = \frac{10}{15} = \frac{2}{3}$, $Pr(b|A) = \frac{5}{15} = \frac{1}{3}$

$Pr(a|B) = \frac{6}{18} = \frac{1}{3}$, $Pr(b|B) = \frac{12}{18} = \frac{2}{3}$

PREDICTION:

Arbitrary doc $d$ with $m_a \times a$, $m_b \times b$

$Pr(A|d) = Pr(A) \cdot Pr(a|A)^{m_a} \cdot Pr(b|A)^{m_b}$

$= \frac{1}{2} \cdot \left(\frac{2}{3}\right)^{m_a} \cdot \left(\frac{1}{3}\right)^{m_b}$

$Pr(B|d) = Pr(B) \cdot Pr(a|B)^{m_a} \cdot Pr(b|B)^{m_b}$

$= \frac{1}{2} \cdot \left(\frac{1}{3}\right)^{m_a} \cdot \left(\frac{2}{3}\right)^{m_b}$

$Pr(A|d) > Pr(B|d)$

$\Longleftrightarrow \frac{Pr(A|d)}{Pr(B|d)} > 1$

$\Longleftrightarrow 2^{m_a} / 2^{m_b} = 2^{m_a - m_b} > 1$

$\Longleftrightarrow m_a - m_b > 0$

$\Longleftrightarrow m_a > m_b$

# Linear Classifiers 5b/5

- The toy example from our last lecture again:

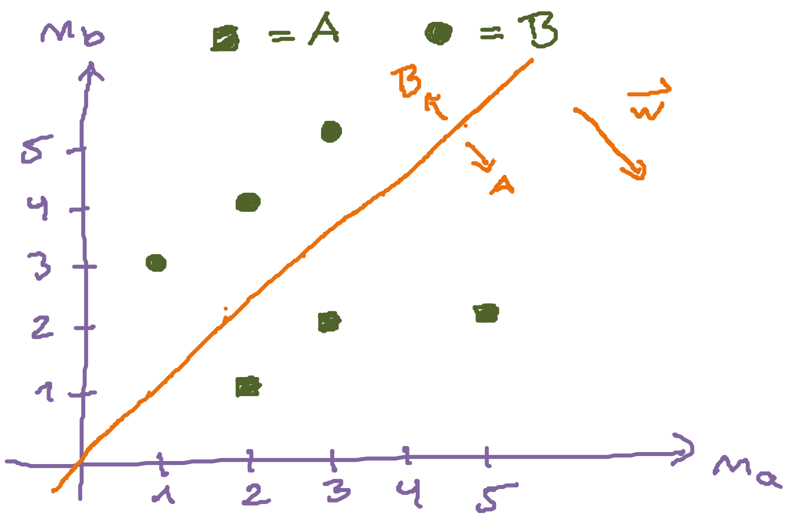
| | | | $(m_a, m_b)$ |
|---|---|---|---|
| Doc 1: | aba | class A | $(2, 1)$ |
| Doc 2: | baabaaa | class A | $(5, 2)$ |
| Doc 3: | bbaabbab | class B | $(3, 5)$ |
| Doc 4: | abbaa | class A | $(3, 2)$ |
| Doc 5: | abbb | class B | $(1, 3)$ |
| Doc 6: | bbbaab | class B | $(2, 4)$ |

$$w = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

$$w_1 = \log_2 \frac{Pr(a|A)}{Pr(b|A)} = 1$$

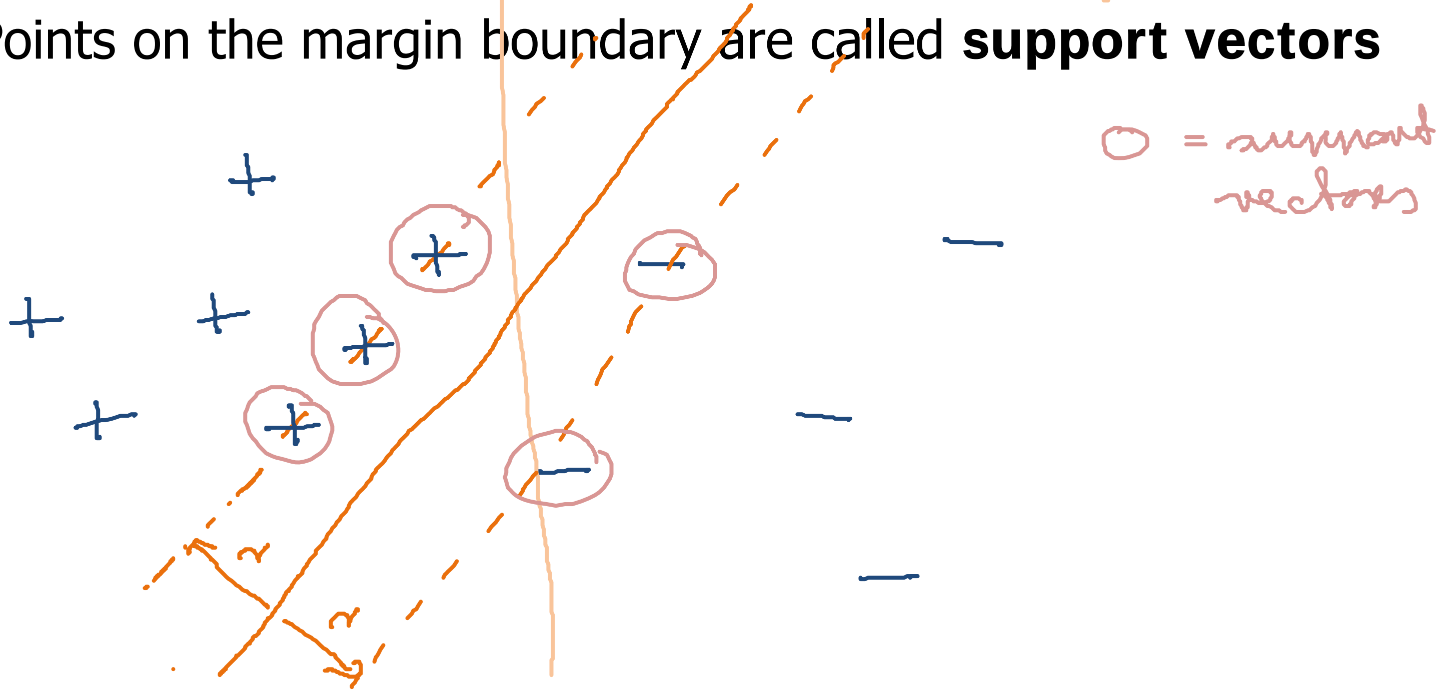
$$w_2 = \log_2 \frac{Pr(b|A)}{Pr(b|B)} = -1$$

$$b = -\log_2 \frac{Pr(A)}{Pr(B)} = 0$$



# Support Vector Machines 1/8

- Intuition
  - Place the separating hyperplane H such that the symmetric margin around H until the next points is as large as possible
  - In $\mathbf{R}^2$ this means: try to separate the point sets with not just a line, but a "band" of width $2r$, with $r > 0$ as large as possible
  - Points on the margin boundary are called **support vectors**



# Support Vector Machines 2/8

- Derivation of formal optimization problem
  - Let $x_1, \ldots, x_m \in \mathbf{R}^d$ be the objects from the training set
  - Let $y_i = +1$ if $x_i$ is in class A, $y_i = -1$ if $x_i$ is in class B
  - Let $H = \{ x \text{ in } \mathbf{R}^d : w \bullet x = b \}$ be a separating hyperplane, such that $w \bullet x_i - b > 0$ for $x_i$ from A, and $< 0$ for $x_i$ from B
  - Then $\text{dist}(x_i , H) = y_i \cdot (w \bullet x_i - b) / |w|$ (see slide 7)
  - This gives rise to the following maximization problem:

    Maximize $2r$, such that $y_i \cdot (w \bullet x_i - b) / |w| \geq r$ for all $i$
  - We can equivalently formulate this as … proof on next slide

    Minimize $|w|^2$, such that $y_i \cdot (w \bullet x_i - b) \geq 1$ for all $i$
  - This is a well-known kind of optimization problem … slide 14

# Support Vector Machines 3/8

GIVEN: $x_i, y_i$

- Proof of equivalence of
  - Maximize $2r$, such that $y_i \cdot (w \bullet x_i - b) / |w| \geq r$ for all $i$
  - Minimize $|w|^2$, such that $y_i \cdot (w \bullet x_i - b) \geq 1$ for all $i$

$$\max_{r,w,b} 2r \quad \text{s.t.} \quad y_i \cdot (w \cdot x_i - b) / |w| \geq r$$

$\alpha := r \cdot |w|$

$r = \frac{\alpha}{|w|}$

$$\max_{\alpha,w,b} \frac{2\alpha}{|w|} \quad \text{s.t.} \quad y_i \cdot (w \cdot x_i - b) \geq \alpha$$

$2/|w|$

$$y_i \cdot \left(\frac{w}{\alpha} \cdot x_i - \frac{b}{\alpha}\right) \geq 1$$

OBSERV: if $\alpha, w, b$ is optimal, then so is $1, \frac{w}{\alpha}, \frac{b}{\alpha}$

So, we might as well set $\alpha = 1$

$$\max_{w,b} \frac{2}{|w|} \quad \text{s.t.} \quad y_i \cdot (w \cdot x_i - b) \geq 1$$

$$\max \frac{2}{|w|} \iff \max \frac{1}{|w|} \iff \min |w| \Rightarrow \min |w|^2$$

# Support Vector Machines 4/8

e.g.
$3x_1 - x_2 + 5x_3 \geq -1$

- We now have a quadratic optimization problem
  - The $|w|^2 = w \bullet w$ is a **quadratic** objective function
  - The $y_i \cdot (w \bullet x_i - b) \geq 1$ are **linear** constraints
  - There are established numerical methods for this kind of problem, but the details are beyond the scope of this course
  - Similar as for the SVD, we will use third-party software:

    SVM Light … see next slide

# Support Vector Machines 5a/8

- SVM Light Software
  - Solves the described quadratic optimization problem
  - Download from http://svmlight.joachims.org
  - Usage for training:

    ./svm_learn <training data> <model>

    The file with the training data contains one line per document (label + features with their counts), see live demo for exact format

    The mode file stores the optimal w and b … the console outputs provides |w| and the number of outliers

    Note from slide 15, that the size of the margin is 2 / |w|

# Support Vector Machines 5b/8

- SVM Light Software
  - Usage for classification:

    ./svm_classify <testing data> <model file> <output file>

    Format for testing data is like for training data

    The output file contains the value

# Support Vector Machines 6/8

- So far complete linear separation or nothing
  - The optimization problem can be easily extended to incorporate **outliers =** objects in the training set that lie inside of the margin or even on the wrong side of it:

    Minimize $|w| / 2 + C \cdot \Sigma_i \xi_i$

    such that $y_i \cdot (w \bullet x_i - b) / |w| \geq 1 - \xi_i$ for all $i$

    where $\xi_i > 0$ and the $C > 0$ is a user-defined parameter

# Support Vector Machines 7/8

- Multi-Class Support Vector Machines
  - Assume we have an arbitrary number of k classes again
  - **Option 1:** Build k classifiers, one for each class, with the i-th one doing the classification: Class i OR not Class i

    Drawback: Need to "vote" when more than one class wins
  - **Option 2:** Build $k \cdot (k - 1) / 2$ classifiers, one for each subset of two classes

    Drawback: For large k, that's a lot of classifiers !
  - **Option 3:** Extend the SVM theory to be able to deal with more than two classes directly

    Drawback: optimization problem becomes more complex

# Support Vector Machines 8/8

- What if the data is not at all linearly separable
  - … even when allowing for a few outliers
  - Standard trick: map objects to a different vector space, where they become (almost) linearly separable again
  - For SVMs, this can be done particularly efficiently, with the so-called "kernel" trick … see machine learning lecture

# References

- Further reading
  - Textbook Chapter 15: Support vector machines

    http://nlp.stanford.edu/IR-book/pdf/15svm.pdf
- Wikipedia
  - http://en.wikipedia.org/wiki/Linear_classifier
  - http://en.wikipedia.org/wiki/Support_vector_machine
- SVM Light Software
  - http://svmlight.joachims.org