Information Retrieval

WS 2012 / 2013

Lecture 11, Wednesday January 23rd, 2013

(SVMs = 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 (Naïve Bayes)
  - Oral exams are on **March 5 + 6** (Tuesday + Wednesday)
- Support Vector Machines (SVMs)
  - Another linear classifier, just like Naïve Bayes
  - But different objective function + harder to optimize
  - Some more linear algebra … you will love it
  - Play around with SVM Light software
  - **Exercise Sheet 11:** Compare SVMs with Naïve Bayes with respect to linear separability and classification accuracy

Experiences with ES#10 (Naive Bayes)

- Summary / excerpts   last checked January 23, 14:45
  - Theory was clear and not too hard to implement
  - Confusion about multiple labels and choice of training set
  - Great observation: better compare the **log** Pr(C = c | doc)

    Reason: They easily become ≤ -1000 , and the exp(...) of any such value is 0 on a typical machine, and all such classes then become indistinguishable → hurts prediction quality badly
  - Ignoring stop-words helps a bit, but not much
  - Another promising idea from you: consider only words that strongly discriminate between classes in the training set

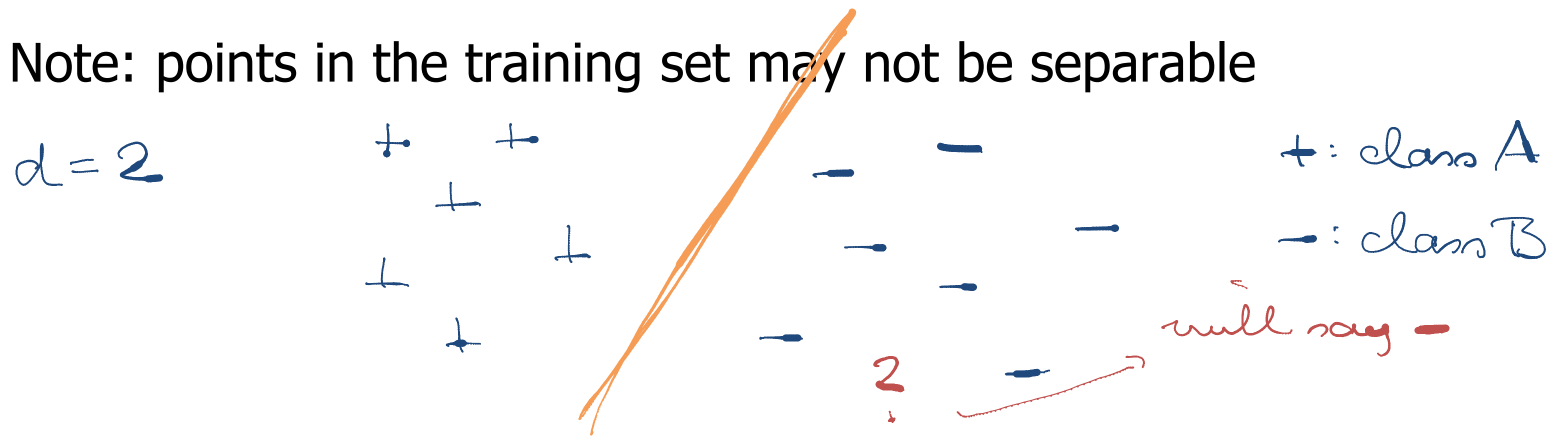
Your results for ES#10 (Naive Bayes)

- For our dataset (38.115 docs, 18 classes)
  - Reading time: on the order of 10 seconds
  - Training time: on the order of 1 second
  - Prediction time: on the order of a few seconds
  - **Bottom line 1:** Naive Bayes is definitely efficient !
  - Quality around 50%
  - With non-exp-trick 60% and more
  - **Bottom line 2:** Without having seen other methods, it's hard to tell whether this is good or bad or so-so

Linear Classifiers 1/6

- 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

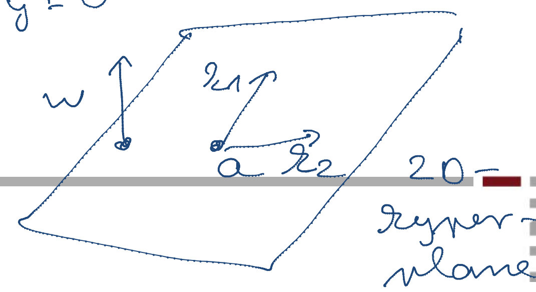
    For d = 2 this means: try to separate by a **straight line**
  - Predictions are made based on which side of the hyperplane / straight line the object lies on
  - Note: points in the training set may not be separable



Linear Classifiers 2/6

$x \perp y :=$ x orthogonal to y $\Leftrightarrow x \bullet y = 0$

$d = 3$



$w, h_1, \ldots, h_{d-1}$ form a base of $\mathbb{R}^d$

- Formal definition of a hyperplane
  - 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 $w \in \mathrm{R}^d$ and $b \in \mathrm{R}$ such that $H = \{ x \in \mathbf{R}^d : w \bullet x = b \}$, $w \perp h_1, \ldots, h_{d-1}$

We have to show $x \in H \Leftrightarrow w \cdot x = b$

"$\Rightarrow$": $x \in H \Rightarrow x = a + \sum \alpha_i \cdot h_i$

$\Rightarrow w \bullet x = w \bullet a + \sum \alpha_i w \bullet h_i = w \cdot a =: b$ ($=0$ because $w \perp h_i$)

"$\Leftarrow$": $x \in \mathbb{R}^d$, $w \bullet x = b$, write $x - a = \alpha \cdot w + \sum \alpha_i \cdot h_i$ (because $w, h_1, \ldots, h_{d-1}$ form a base)

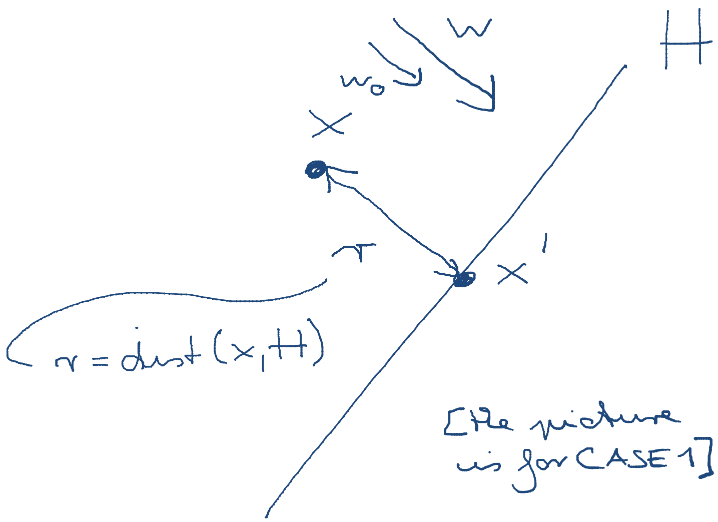
$\Rightarrow w \bullet (a + \alpha \cdot w + \sum \alpha_i h_i) = b = w \bullet a$

$w \bullet a + \alpha \cdot |w|^2 + \underbrace{\sum \alpha_i h_i w}_{=0}$ $\Rightarrow \alpha \cdot |w|^2 = 0$

$\Rightarrow \alpha = 0$ ∎

Linear Classifiers 3/6

- 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



Let $w_0 = w / |w| \Rightarrow |w_0| = 1$

CASE 1: $x' = x + r \cdot w_0$, $r > 0$
(w points from x towards H)

$x' \in H \Rightarrow w \bullet x' = w \bullet x + r \cdot w \bullet w_0 = b$

$w \bullet w_0 = |w| \cdot w_0 \bullet w_0 = |w|$

$$\Rightarrow r = - \frac{w \bullet x - b}{|w|}$$

CASE 2: $x' = x - r \cdot w_0$, $r > 0$
(w points from x away from H)

$$\Rightarrow r = \frac{w \bullet x - b}{|w|}$$

Linear Classifiers 4/6

- Two-class Naïve 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
  - 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
  - **Lemma:** For two classes A and B, define $b \in \mathbf{R}$ and $w \in \mathbf{R}^{|V|}$

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

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

Linear Classifiers 5/6

- Proof of Lemma

  $x = (f_1, \ldots, f_{|V|})^T$
  = feature vector for doc

  - **NB** predicts A for x if $w \bullet x - b > 0$, and B otherwise

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

$$\Pr(A \mid doc) = \Pr(A) \cdot \prod_{i=1}^{|V|} \Pr(w_i \mid A)^{f_i} / P$$

$$\Pr(B \mid doc) = \Pr(B) \prod_{i=1}^{|V|} \Pr(w_i \mid B)^{f_i} / P$$

$$\log \frac{\Pr(A \mid doc)}{\Pr(B \mid doc)} = \underbrace{\log \frac{\Pr(A)}{\Pr(B)}}_{= -b} + \sum_{i=1}^{|V|} f_i \cdot \underbrace{\log \frac{\Pr(w_i \mid A)}{\Pr(w_i \mid B)}}_{= w_i}$$

$$= w \bullet x - b$$

$$\Pr(A \mid doc) > \Pr(B \mid doc)$$

$$\Longleftrightarrow \log \frac{\Pr(A \mid doc)}{\Pr(B \mid doc)} > 0$$

$$\Longleftrightarrow w \bullet x - b > 0 \quad \blacksquare$$

$$x > y \Longleftrightarrow \frac{x}{y} > 1$$

$$\Longleftrightarrow \log \frac{x}{y} > 0$$

Linear Classifiers 6/6

- 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

$n_A = 3$, $n_B = 3$

$n_{aA} = 10$, $n_{aB} = 6$

$n_{bA} = 5$, $n_{bB} = 12$

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

$Pr(a|A) = \frac{2}{3}$; $Pr(a|B) = \frac{1}{3}$

$Pr(b|A) = \frac{1}{3}$; $Pr(b|B) = \frac{2}{3}$

new doc $n_a \times a$, $n_b \times b$

$Pr(A|doc) = \frac{1}{2} \cdot \left(\frac{2}{3}\right)^{n_a} \cdot \left(\frac{1}{3}\right)^{n_b} / P$

$Pr(B|doc) = \frac{1}{2} \cdot \left(\frac{1}{3}\right)^{n_a} \cdot \left(\frac{2}{3}\right)^{n_b} / P$

$\frac{Pr(A|doc)}{Pr(B|doc)} = \left(\frac{2}{3}\right)^{n_a - n_b} \cdot \underbrace{\left(\frac{1}{3}\right)^{n_b - n_a}}_{= 3^{n_a - n_b}}$

$= 2^{n_a - n_b} > 1 \iff n_a > n_b$

w and b from the proof

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

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

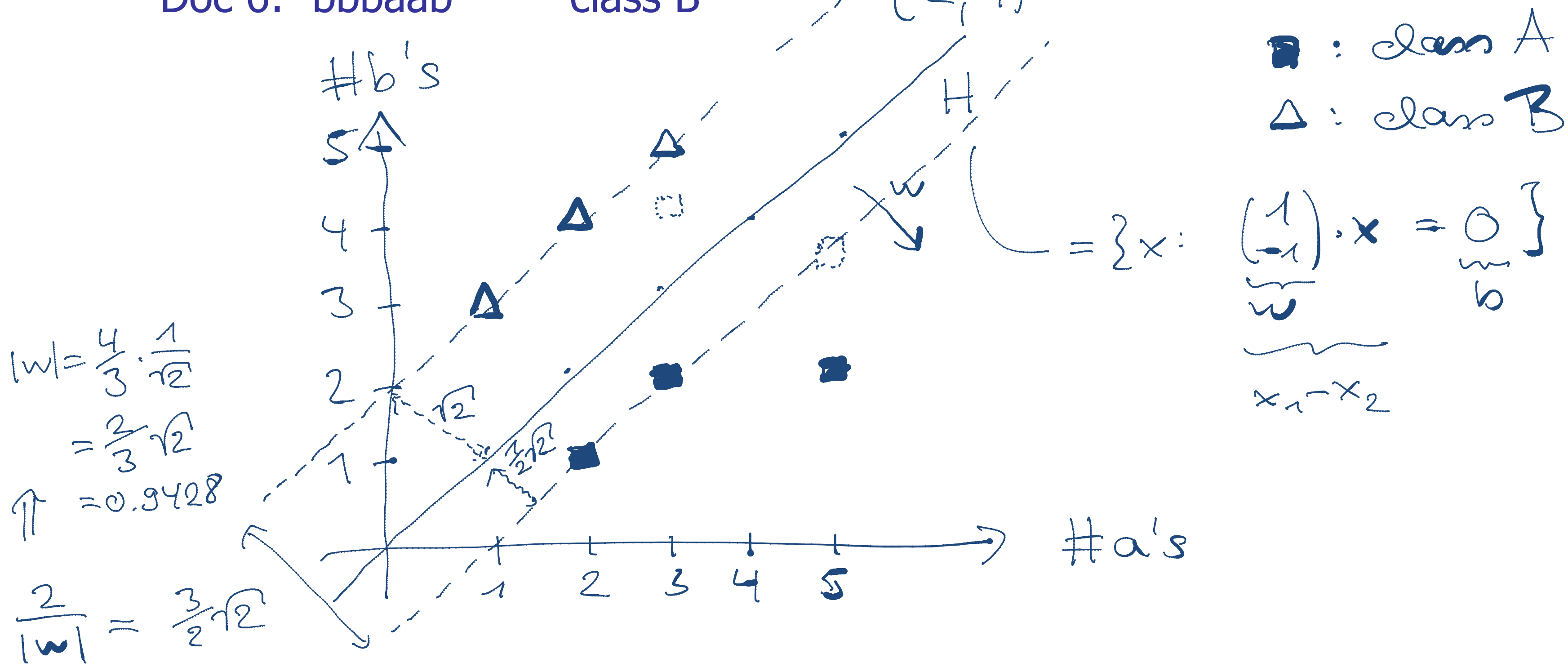
$b = -\log_2 \frac{Pr(A)}{Pr(B)} = 0$

$\underbrace{\begin{pmatrix} n_a \\ n_b \end{pmatrix}}_{= x} \cdot \underbrace{\begin{pmatrix} 1 \\ -1 \end{pmatrix}}_{= w} = n_a - n_b \lessgtr 0$

Feature vectors $(n_a, n_b) \in \mathbb{R}^2$

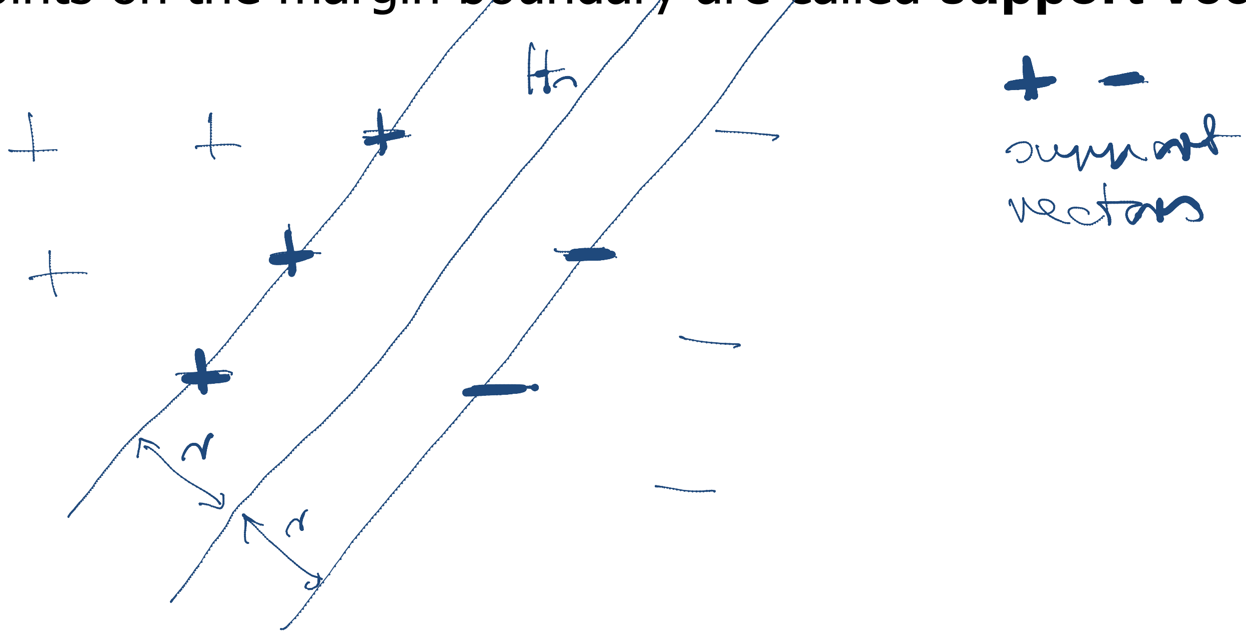
Doc 1:	aba	class A	(2, 1)
Doc 2:	baabaaa	class A	(5, 2)
Doc 3:	bbaabbab	class B	(3, 5)
Doc 4:	abbaa	class A	(3, 2)
Doc 5:	abbb	class B	(1, 3)
Doc 6:	bbbaab	class B	(2, 4)

let's consider these as points in the plane



Support Vector Machines 1/7

- Intuition
  - Place the separating hyperplane H such that on both sides, there is a **margin** $r > 0$ as large as possible to the points
  - 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 2/7

UNI FREIBURG

- 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 3/7

UNI FREIBURG

- 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

$\frac{2}{|w|} = 2r$ is the width of the margin / band (you need that for the exercise)

$$\max 2r \quad \text{s.t.} \quad y_i (w \bullet x_i - b) \geq r \cdot |w| \quad \forall i$$

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

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

Observe: if w, b, α is optimum, so is $\frac{w}{\alpha}, \frac{b}{\alpha}, 1$

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

and objective $\frac{2 \cdot 1}{|w/\alpha|} = \frac{2\alpha}{|w|}$

so just take $\alpha = 1$ and just optimize over w and b

$$\max \frac{2}{|w|} \Longleftrightarrow \min \frac{|w|}{2} \Longleftrightarrow \min |w|^2$$

Support Vector Machines 4/7

- 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 Software
  - Solve this optimization problem
  - Download from http://svmlight.joachims.org
  - I will show to download and install it, then let's apply it to our toy example (the 6 documents, with words a and b)

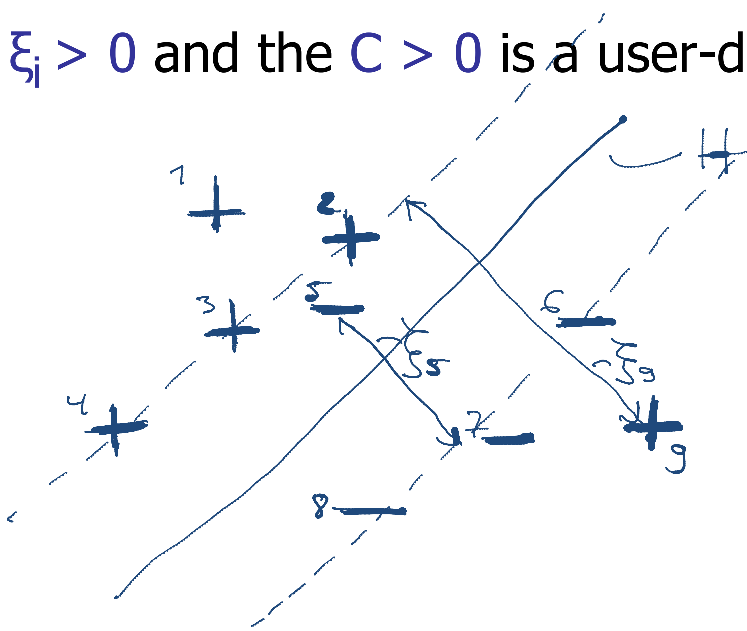
Support Vector Machines 5/7

- 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



In SVM Light, the C can be set with the -c option.
Set to something very large to disallow outliers.

Support Vector Machines 6/7

- 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 7/7

- 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