Perceptrons and Optimal Hyperplanes

Example: Majority-Vote Function

- Definition: Majority-Vote Function $f_{majority}$
  - N binary attributes, i.e. $x \in \{0,1\}^N$
  - If more than N/2 attributes in x are true, then $f_{majority}(x)=1$, else $f_{majority}(x)=-1$.
- How can we represent this function as a decision tree?
  - Huge and awkward tree!
- Is there an "easier" representation of $f_{majority}$?

Example: Spam Filtering

	viagra	learning	the	dating	lottery	spam?
$\vec{x}_1 = ($	1	0	1	0	0 $)$	$y_1 = 1$
$\vec{x}_2 = ($	0	1	1	0	0 $)$	$y_2 = -1$
$\vec{x}_3 = ($	0	0	0	0	1 $)$	$y_3 = 1$

- Instance Space X:
  - Feature vector of word occurrences => binary features
  - N features (N typically > 50000)
- Target Concept c:
  - Spam (+1) / Ham (-1)
- Type of function to learn:
  - Set of Spam words S, Set of Ham words H
  - Classify as Spam (+1), if more Spam words than Ham words in example.

Example: Spam Filtering

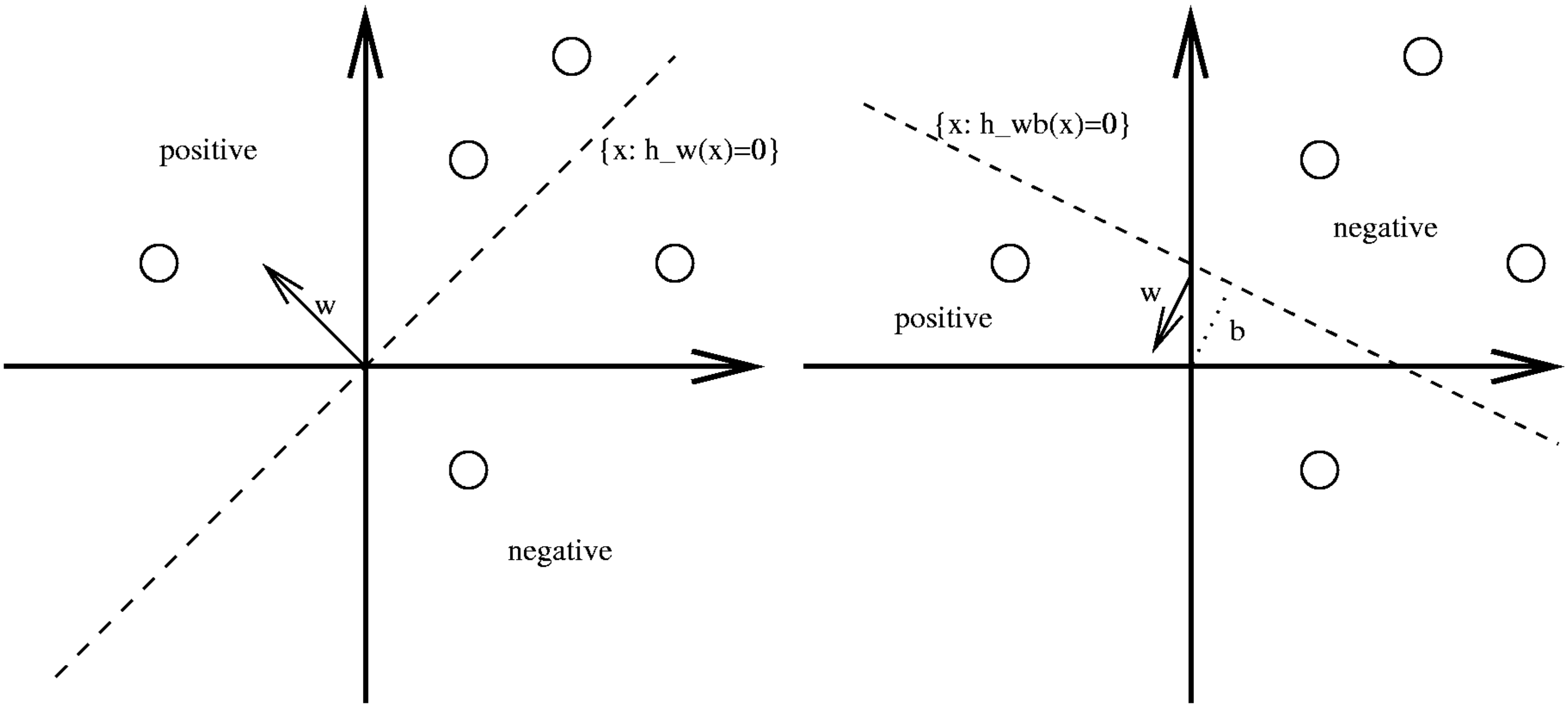
	viagra	learning	the	dating	lottery	spam?
$\vec{x}_1 = ($	1	0	1	0	0 $)$	$y_1 = 1$
$\vec{x}_2 = ($	0	1	1	0	0 $)$	$y_2 = -1$
$\vec{x}_3 = ($	0	0	0	0	1 $)$	$y_3 = 1$

- Use weight vector w=(+1, -1, 0, +1, +1)
  - Compute sign(wx)
- More generally, we can use real valued weights to express "spamminess" of word.
  - w=(+10,-1,-0.3,+1,+5)
  - Which vector is most likely to be spam with this weighting? A=x_1, B=x_2, C=x_3

Linear Classification Rules

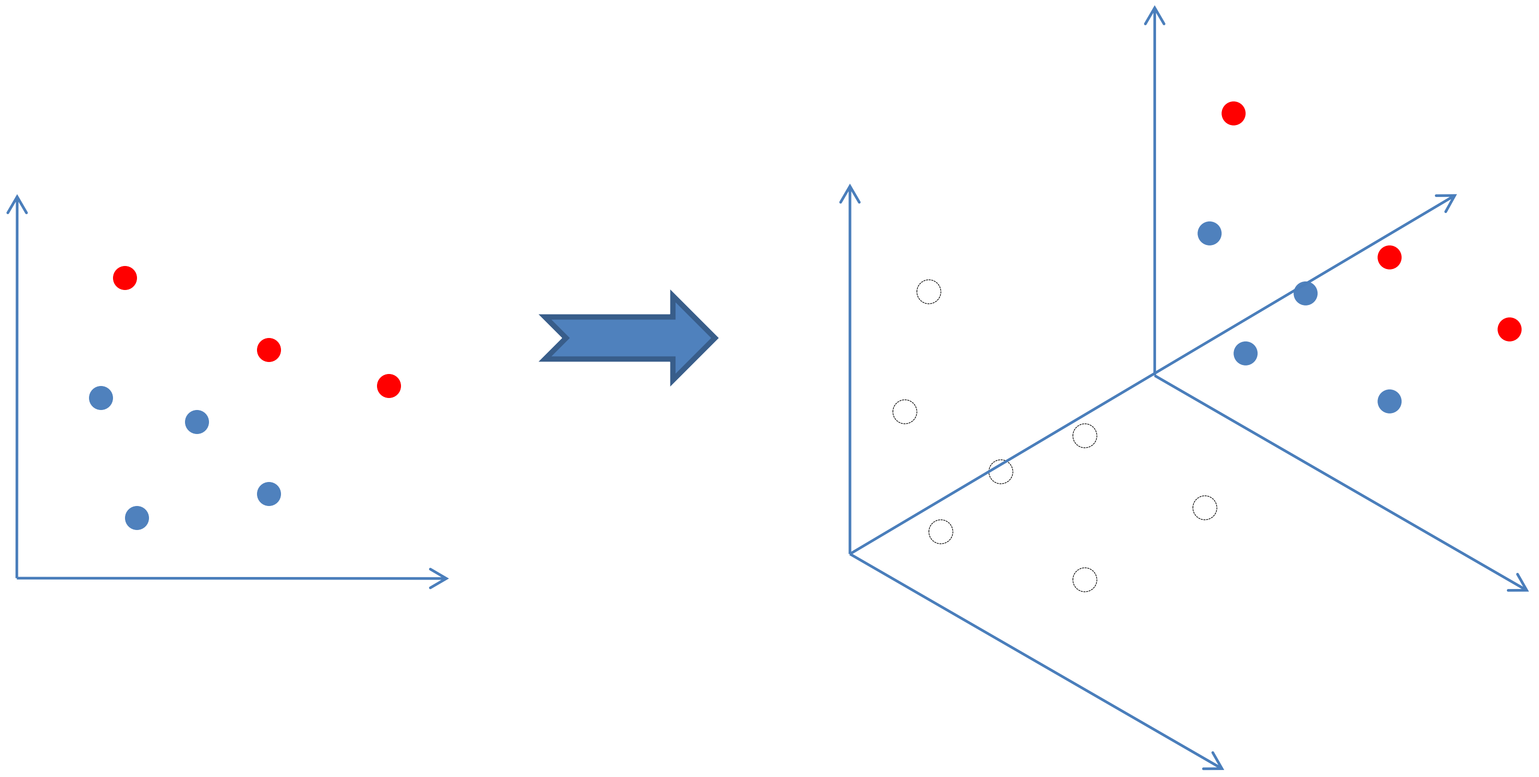
- Hypotheses of the form
  - unbiased: $h_{\vec{w}}(\vec{x}) = \begin{cases} 1 & w_1 x_1 + ... + w_N x_N > 0 \\ -1 & else \end{cases}$
  - biased: $h_{\vec{w},b}(\vec{x}) = \begin{cases} 1 & w_1 x_1 + ... + w_N x_N + b > 0 \\ -1 & else \end{cases}$
  - Parameter vector *w,* scalar *b*
- Hypothesis space H
  - $H_{unbiased} = \{h_{\vec{w}} : \vec{w} \in \Re^N\}$
  - $H_{biased} = \{h_{\vec{w},b} : \vec{w} \in \Re^N \, b \in \Re\}$
- Notation
  - $w_1 x_1 + ... + w_N x_N = \vec{w} \cdot \vec{x}$ and $sign(a) = \begin{cases} 1 & a > 0 \\ -1 & else \end{cases}$
  - $h_{\vec{w}}(\vec{x}) = sign(\vec{w} \cdot \vec{x})$
  - $h_{\vec{w},b}(\vec{x}) = sign(\vec{w} \cdot \vec{x} + b)$

Geometry of Hyperplane Classifiers



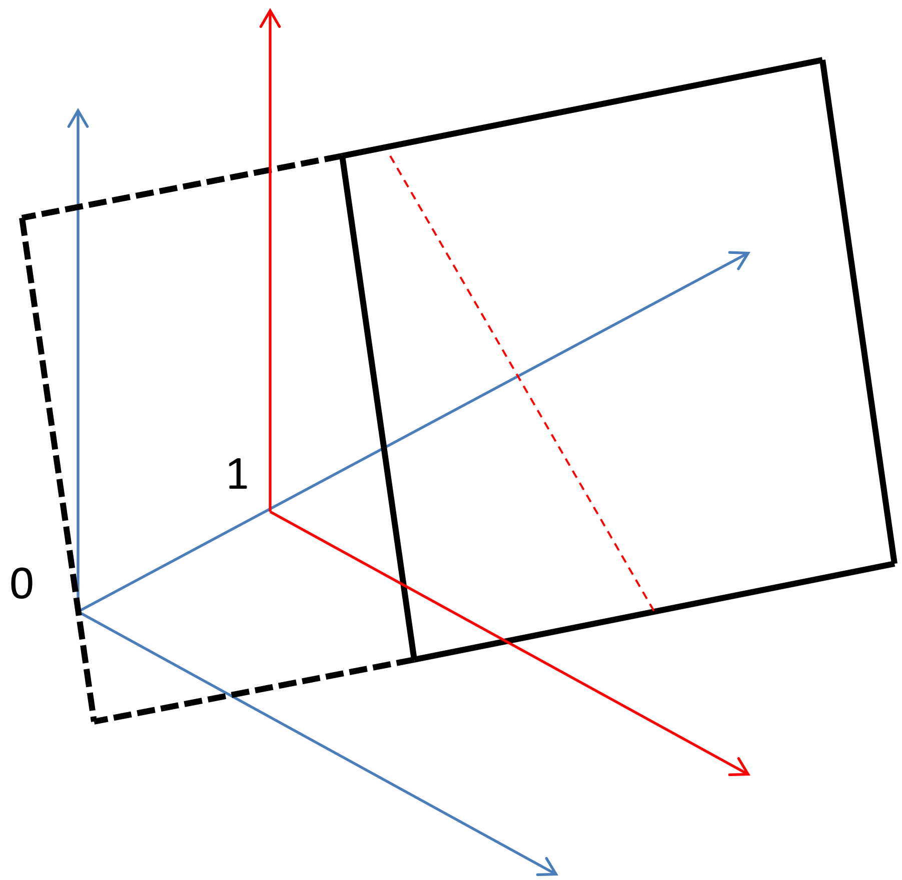
- Linear Classifiers divide instance space as hyperplane
- One side positive, other side negative

Homogeneous Coordinates



$X = (x_1, x_2)$
$W = (w_1, w_2, b)$

$X = (x_1, x_2, 1)$
$W = (w_1, w_2, w_3)$


1
0

(Batch) Perceptron Algorithm

Algorithm:

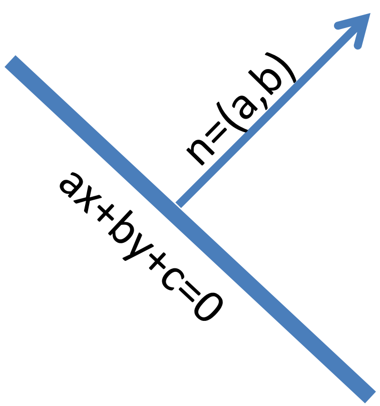
- $\vec{w}_0 = \vec{0},\ k = 0$
- repeat
  - FOR i=1 TO n
    - IF $y_i(\vec{w}_k \cdot \vec{x}_i) \leq 0$ makes mistake
      - $\vec{w}_{k+1} = \vec{w}_k + \eta y_i \vec{x}_i$
      - $k = k + 1$
    - ENDIF
  - ENDFOR
- until I iterations reached

Training Epoch

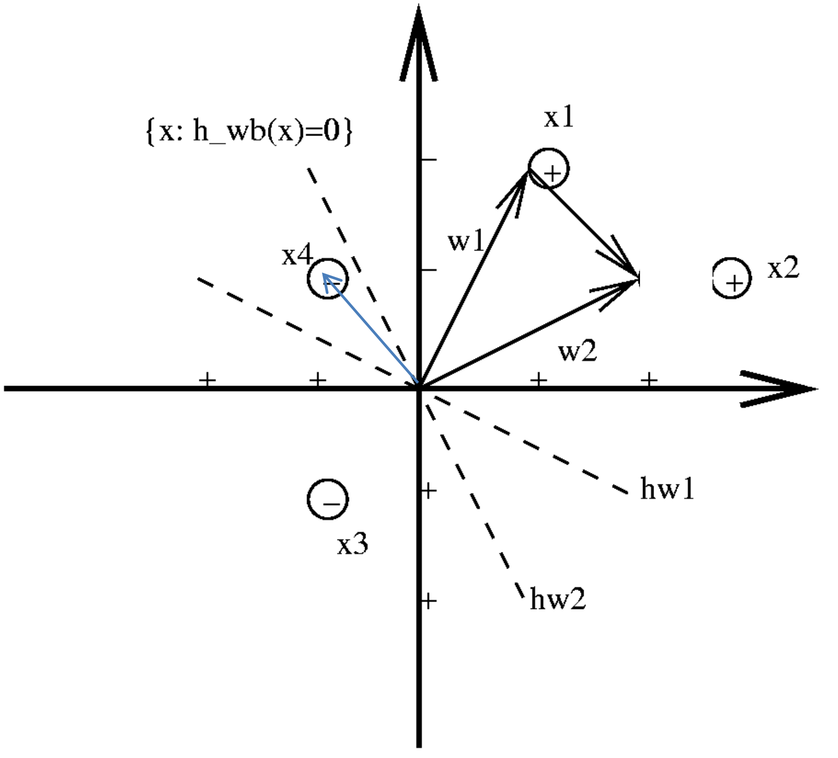
Example: Perceptron

Training Data:

	x_1	x_2	y
$\vec{x}_1 = ($	1	2 $)$	$y_1 = 1$
$\vec{x}_2 = ($	3	1 $)$	$y_2 = 1$
$\vec{x}_3 = ($	-1	-1 $)$	$y_3 = -1$
$\vec{x}_4 = ($	-1	1 $)$	$y_4 = -1$



Updates to weight vector:



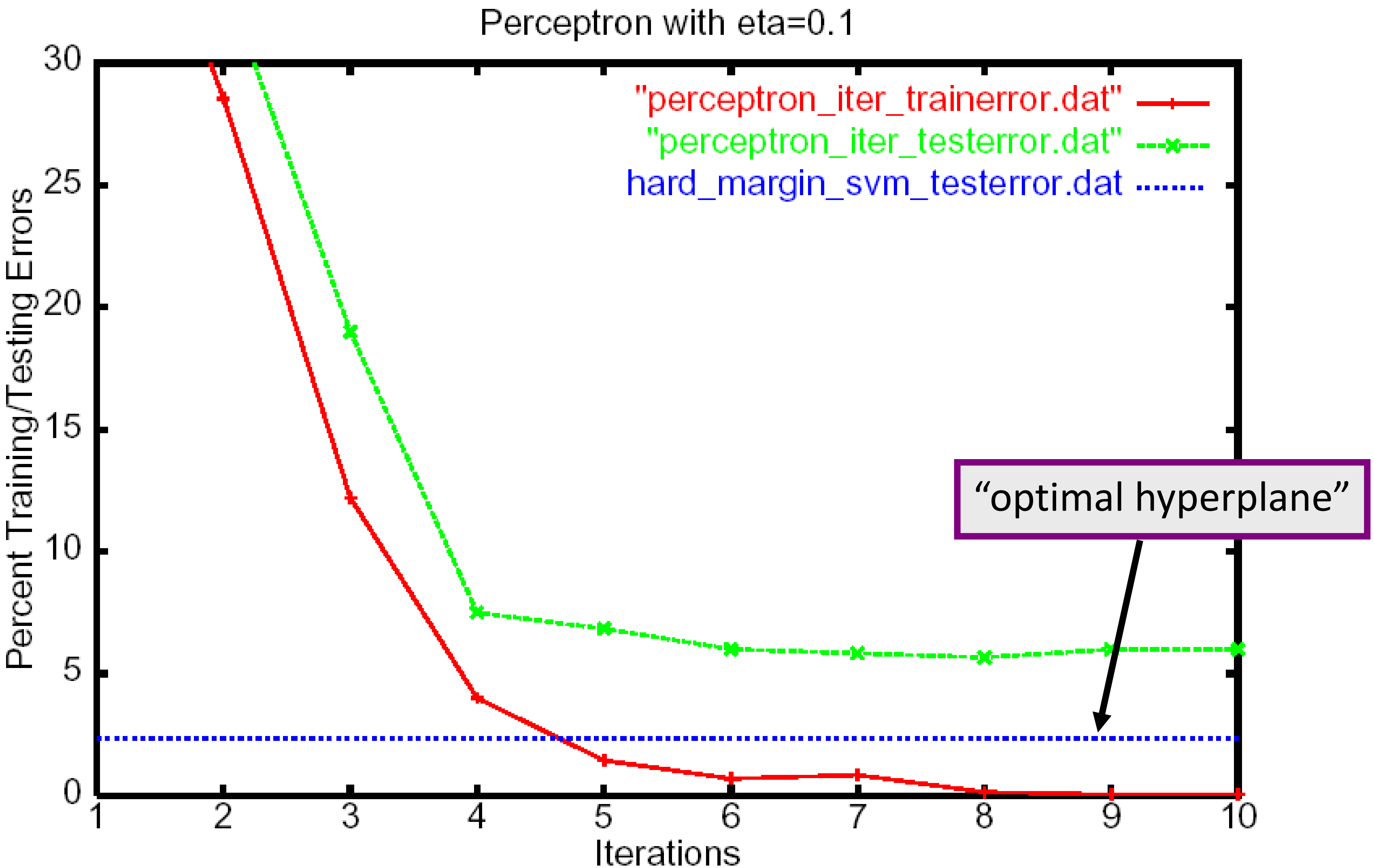
- Init: w=0, η=1
- $(w_0 \ \ x_1) = 0 \rightarrow$ incorrect

  $w_1 = w_0 + \eta\, y_1\, x_1 = 0 + 1*1*(1,2) = (1,2)$

  $\rightarrow h_{w1}x_1 = (w_0+1*1*x_1) * x_1 = h_{w0}(x_1) + 1 * 1 * (x_1*x_1) = 0 + 5$
- $(w_1 \cdot x_2) = (1,2) \cdot (3,1) = 5 \rightarrow$ correct
- $(w_1 \cdot x_3) = (1,2) \cdot (-1,-1) = -3 \rightarrow$ correct
- $(w_1 \cdot x_4) = (1,2) \cdot (-1,1) = 1 \rightarrow$ incorrect
  - $w_2 = (1,2) + \eta\, y_4\, x_4 = (1,2) - (-1,1) = (2,1)$

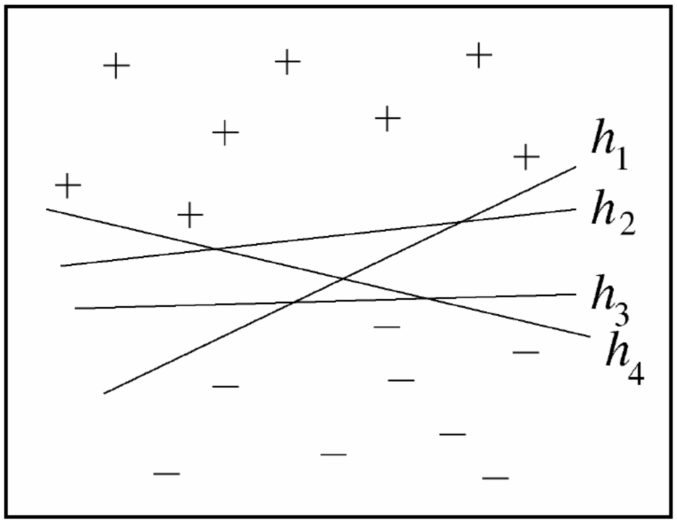
  $\rightarrow h_{w2}\, x_4 = (w_1+1*-1*x_4) * x_4 = h_{w1}(x_4) + 1 * -1 * (x_4 * x_4) = -1$

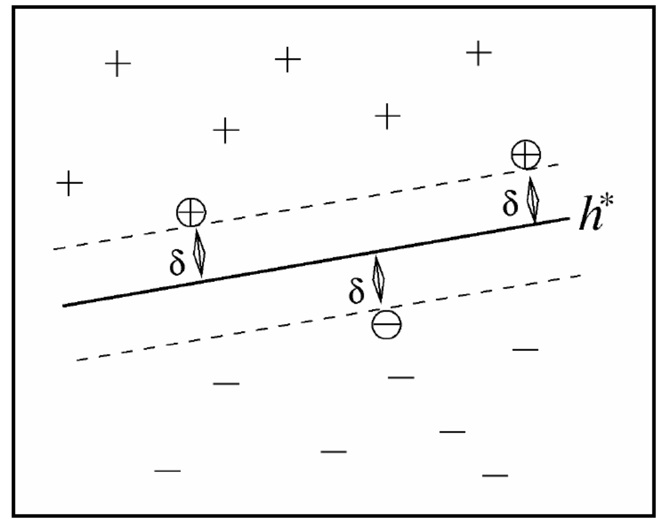
Example: Reuters Text Classification



Optimal Hyperplanes

Assumption: Training examples are linearly separable.





Definition: *For a linear classifier $h_{\vec{w},b}$, the* **margin** δ *of an example* $(\vec{x}, y)$ *is* $\delta = y(\vec{w} \cdot \vec{x} + b)$.

Definition: *The margin is called* geometric margin, *if* $||\vec{w}|| = 1$. *Otherwise,* functional margin.

Hard-Margin Separation

Goal: Find hyperplane with the largest distance to the closest training examples.

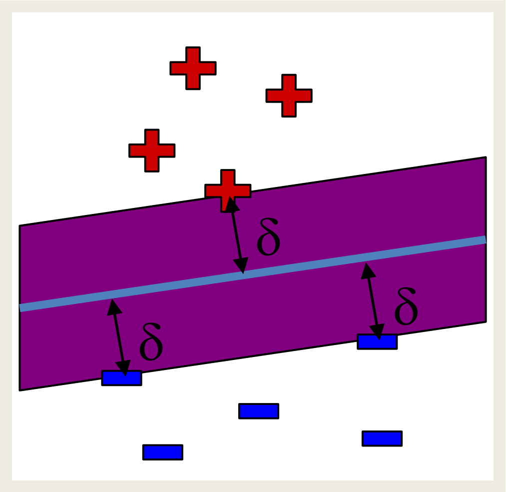
Optimization Problem (Primal):

$$\min_{\vec{w},b} \quad \frac{1}{2}\vec{w}\cdot\vec{w}$$

$$s.t. \quad y_1(\vec{w}\cdot\vec{x}_1 + b) \geq 1$$

$$\ldots$$

$$y_n(\vec{w}\cdot\vec{x}_n + b) \geq 1$$



Support Vectors: Examples with minimal distance (i.e. margin).

Why min ½*w*·*w*?

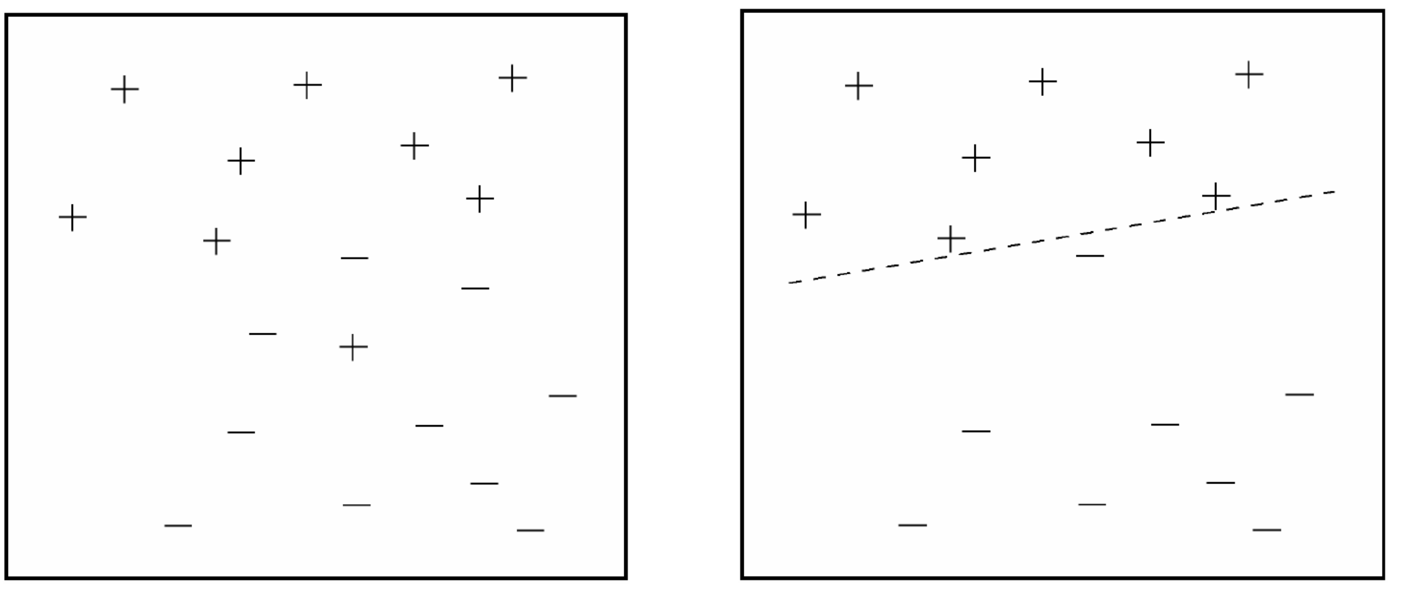
- Maximizing δ and constraining w is equivalent to constraining δ and minimizing w
  - We want maximum margin δ,
    - we don't care about w
    - But because δ=wx, just requiring maximum δ will yield large w…
  - So we ask for maximum δ but constrain w
    - This is equivalent to constraining δ and minimizing w

Definition: *The (hard) margin of a linear classifier* $h_{\vec{w},b}$ *on data* D *is* $\delta = min_{(\vec{x},y)\in D}\,\{y(\vec{w}\cdot\vec{x}+b)\}$.

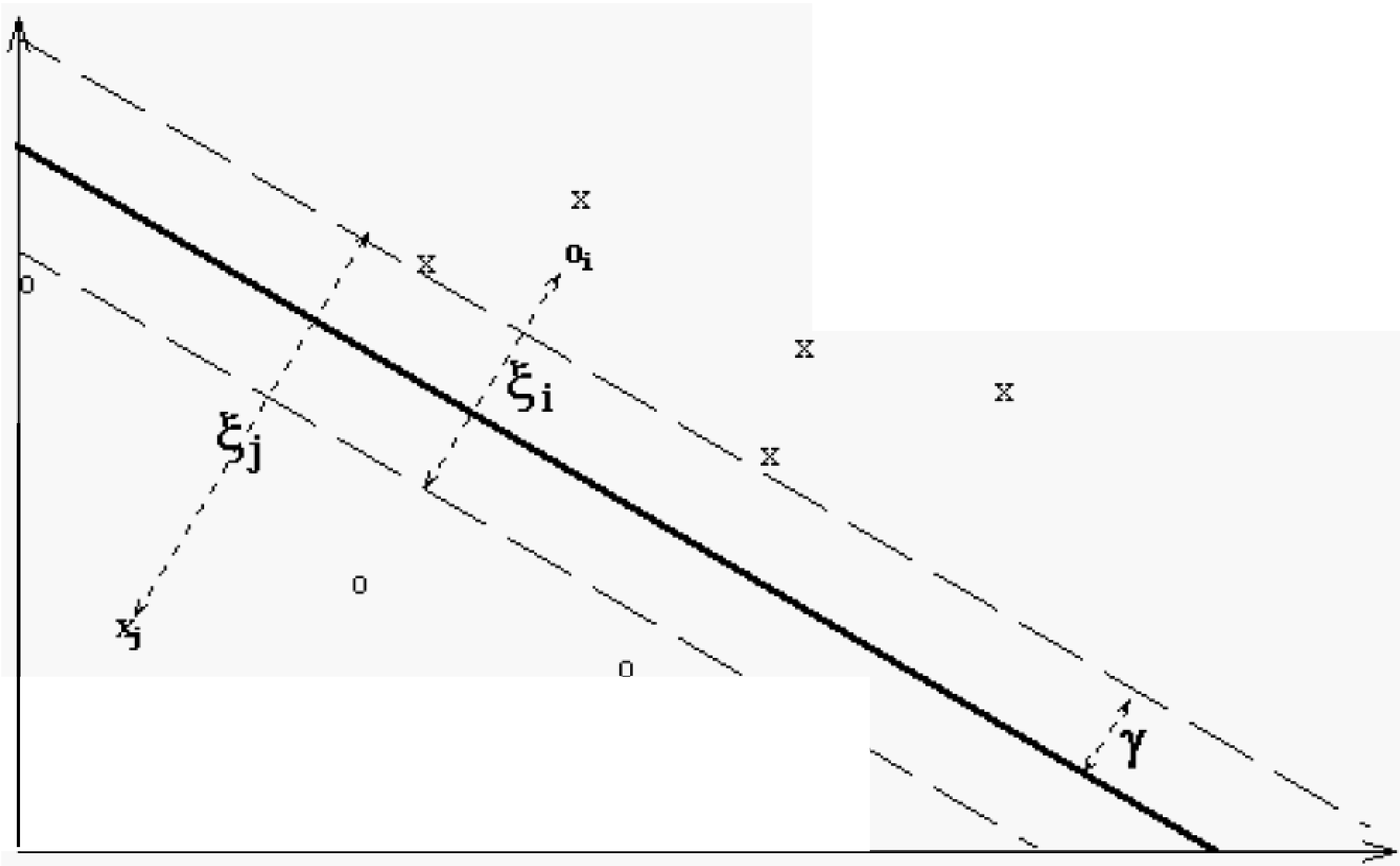
Non-Separable Training Data

Limitations of hard-margin formulation

- For some training data, there is no separating hyperplane.
- Complete separation (i.e. zero training error) can lead to suboptimal prediction error.



Slack



Soft-Margin Separation

Idea: Maximize margin and minimize training

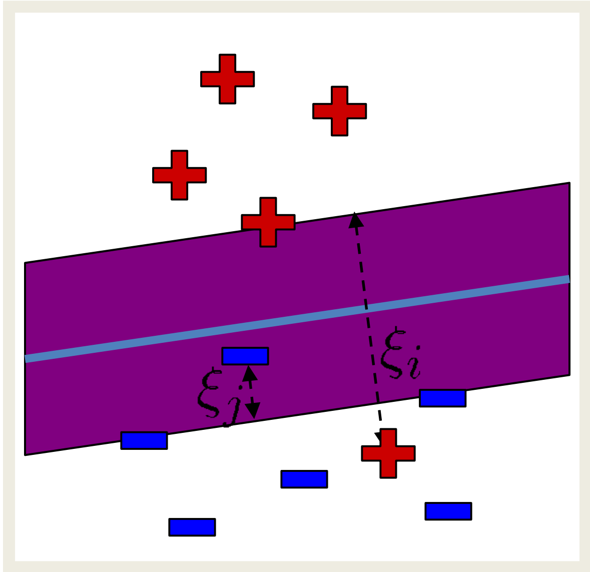
Hard-Margin OP (Primal):

$$\begin{aligned} \min_{\vec{w},b} \quad & \frac{1}{2}\vec{w}\cdot\vec{w} \\ s.t. \quad & y_1(\vec{w}\cdot\vec{x}_1 + b) \geq 1 \\ & \ldots \\ & y_n(\vec{w}\cdot\vec{x}_n + b) \geq 1 \end{aligned}$$

Soft-Margin OP (Primal):

$$\begin{aligned} \min_{\vec{w},\vec{\xi},b} \quad & \frac{1}{2}\vec{w}\cdot\vec{w} + C\sum_{i=1}^{n}\xi_i \\ s.t. \quad & y_1(\vec{w}\cdot\vec{x}_1 + b) \geq 1 - \xi_1 \wedge \xi_1 \geq 0 \\ & \ldots \\ & y_n(\vec{w}\cdot\vec{x}_n + b) \geq 1 - \xi_n \wedge \xi_n \geq 0 \end{aligned}$$

- Slack variable ξ_i measures by how much (x_i, y_i) fails to achieve margin δ
- $\Sigma\xi_i$ is upper bound on number of training errors
- C is a parameter that controls trade-off between margin and training error.



Soft-Margin OP (Primal):

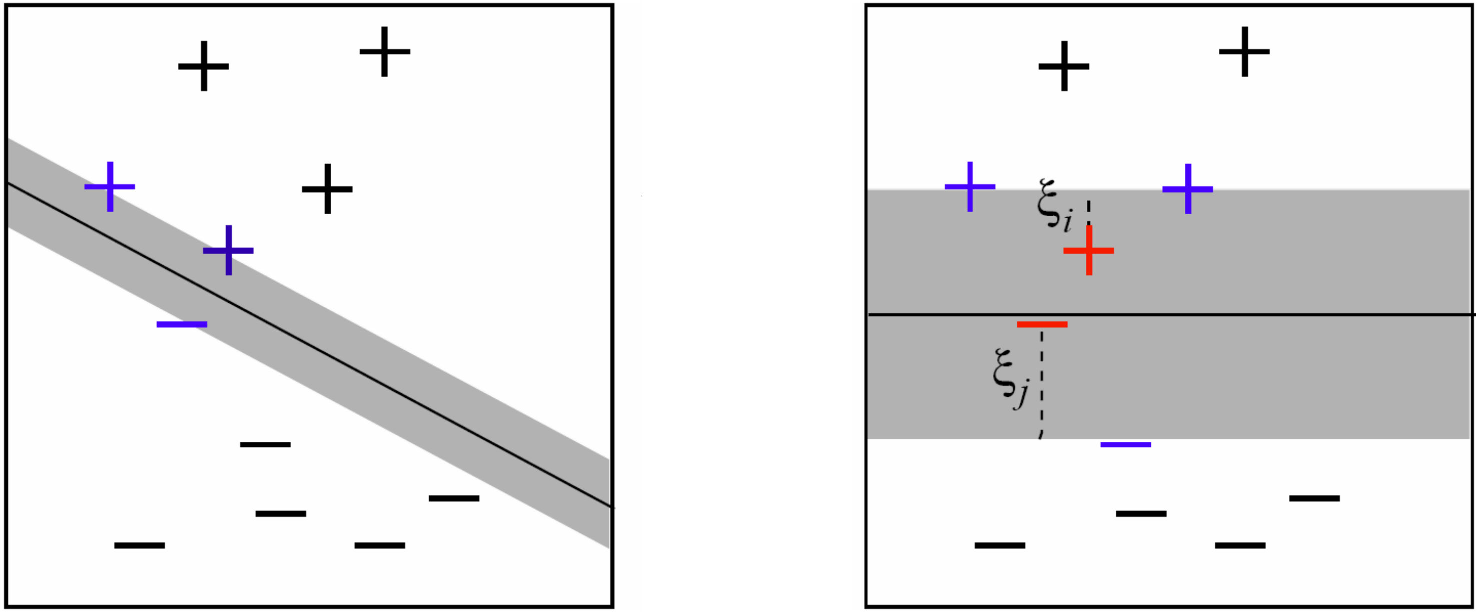
$$\min_{\vec{w},\vec{\xi},b} \frac{1}{2}\vec{w}\cdot\vec{w} + C\sum_{i=1}^{n}\xi_i$$

$$s.t.\ y_1(\vec{w}\cdot\vec{x}_1 + b) \geq 1 - \xi_1 \wedge \xi_1 \geq 0$$

$$\ldots$$

$$y_n(\vec{w}\cdot\vec{x}_n + b) \geq 1 - \xi_n \wedge \xi_n \geq 0$$

Which of these two classifiers was produced using a larger value of C?



A B

Controlling Soft-Margin Separation

- Σξi is upper bound on number of training errors
- C is a parameter

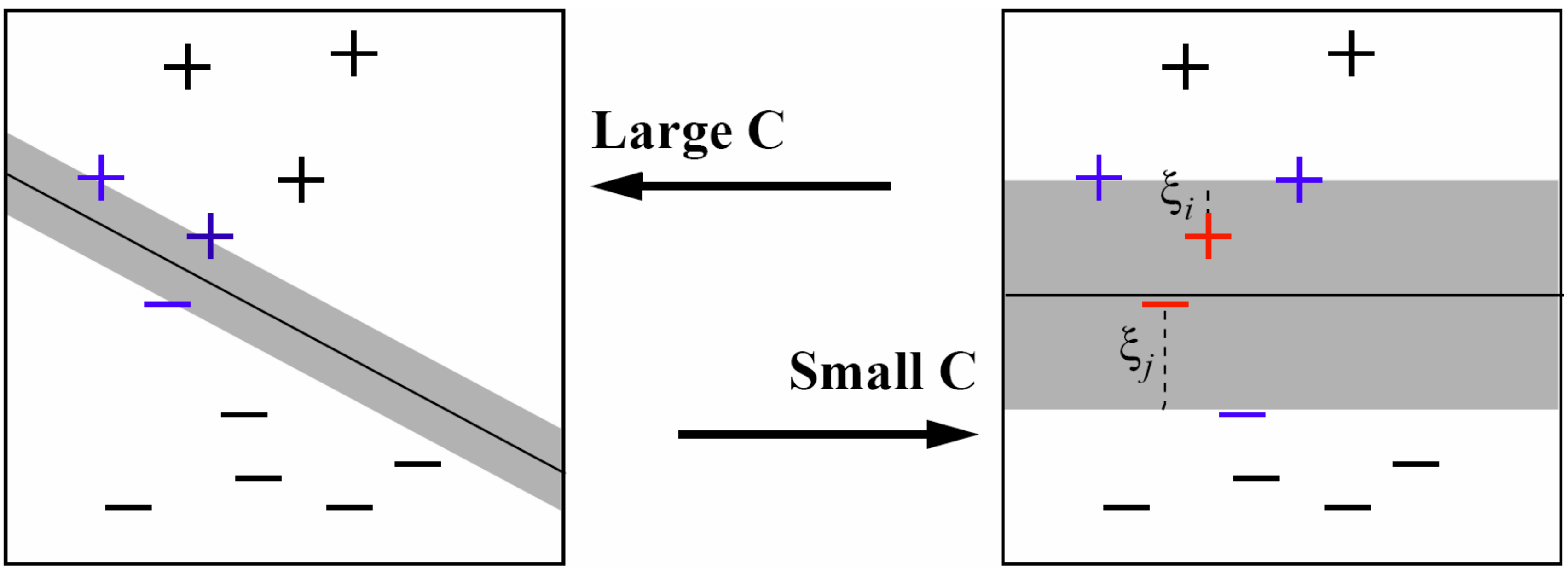
Soft-Margin OP (Primal):

$$\min_{\vec{w},\vec{\xi},b} \frac{1}{2}\vec{w}\cdot\vec{w} + C\sum_{i=1}^{n}\xi_i$$

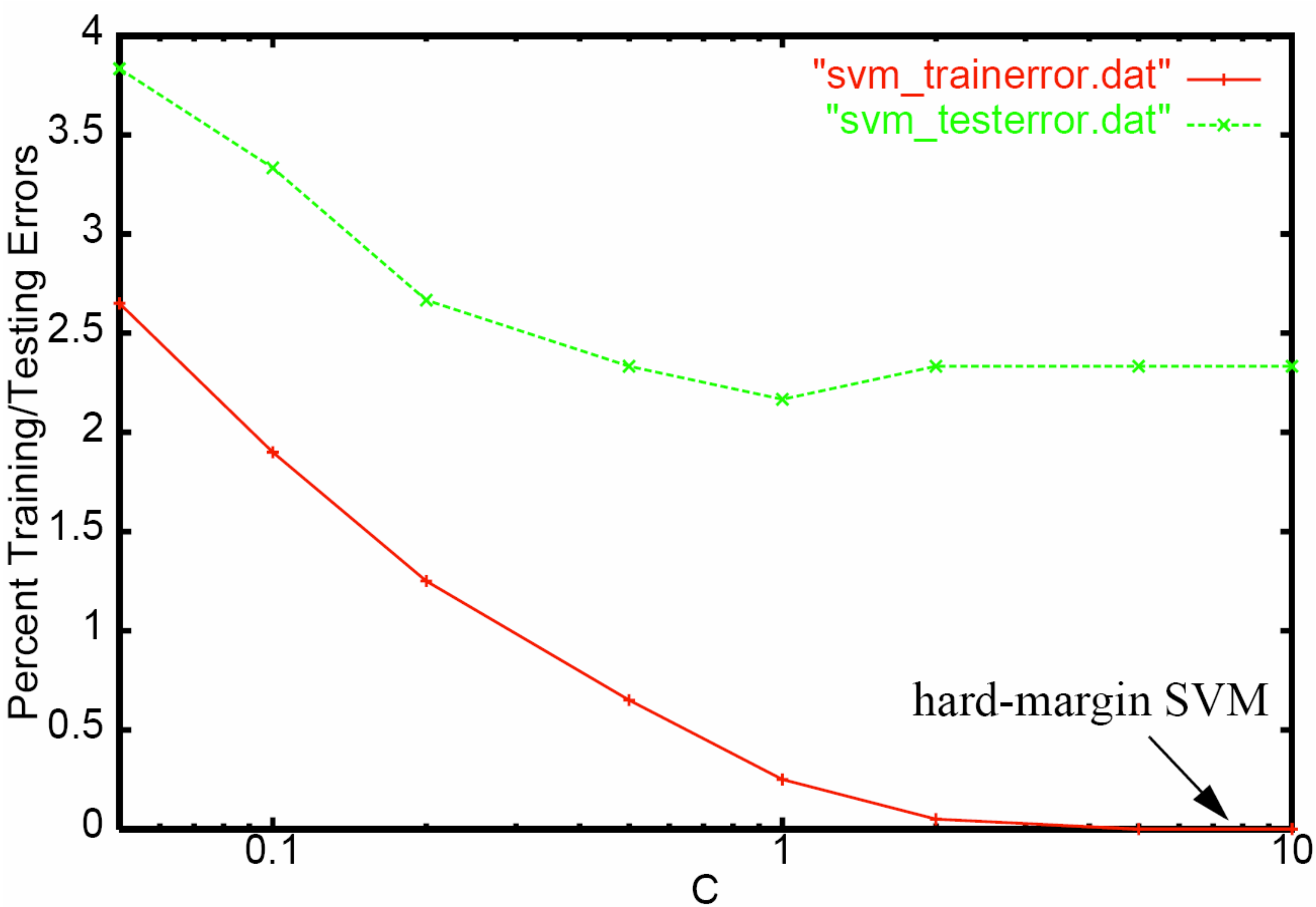
$$s.t.\ y_1(\vec{w}\cdot\vec{x}_1+b) \geq 1-\xi_1 \wedge \xi_1 \geq 0$$

$$\ldots$$

$$y_n(\vec{w}\cdot\vec{x}_n+b) \geq 1-\xi_n \wedge \xi_n \geq 0$$



Example Reuters "acq": Varying C



Example: Margin in High-Dimension

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	y
Example 1	1	0	0	1	0	0	0	1
Example 2	1	0	0	0	1	0	0	1
Example 3	0	1	0	0	0	1	0	-1
Example 4	0	1	0	0	0	0	1	-1
	w_1	w_2	w_3	w_4	w_5	w_6	w_7	**b**
Hyperplane 1	1	1	0	0	0	0	0	2
Hyperplane 2	0	0	0	1	1	-1	-1	0
Hyperplane 3	1	-1	1	0	0	0	0	0
Hyperplane 4	1	-1	0	0	0	0	0	0
Hyperplane 5	0.95	-0.95	0	0.05	0.05	-0.05	-0.05	0

- $\mathbf{Err}_{Dtrain}(h_{\mathbf{w}_1,b_1}) = 2$ and $\sum \xi_i = 8$, $||\mathbf{w}_1|| = \sqrt{2} \implies \delta_1 = -3/\sqrt{2}$
- $\mathbf{Err}_{Dtrain}(h_{\mathbf{w}_2,b_2}) = 0$ and $\sum \xi_i = 0$, $||\mathbf{w}_2|| = \sqrt{4} \implies \delta_2 = 1/\sqrt{4}$
- $\mathbf{Err}_{Dtrain}(h_{\mathbf{w}_3,b_3}) = 0$ and $\sum \xi_i = 0$, $||\mathbf{w}_3|| = \sqrt{3} \implies \delta_3 = 1/\sqrt{2}$
- $\mathbf{Err}_{Dtrain}(h_{\mathbf{w}_4,b_4}) = 0$ and $\sum \xi_i = 0$, $||\mathbf{w}_4|| = \sqrt{2} \implies \delta_4 = 1/\sqrt{2}$
- $\mathbf{Err}_{Dtrain}(h_{\mathbf{w}_5,b_5}) = 0$ and $\sum \xi_i = 0$, $||\mathbf{w}_5|| = \sqrt{2 * 0.9025 + 4 * 0.0025} \implies \delta_5 = 1/\sqrt{1.815}$