Perceptron

CS478 – Machine Learning
Spring 2008

Thorsten Joachims
Cornell University

Reading: Mitchell Chapter 4.4-4.4.2 & Chapter 7.5
Cristianini/Shawe-Taylor Chapter 2-2.1.1

Outline

- **Linear classification rules**
- **Perceptron learning algorithm**
- **Mistake-bound model**
- **Perceptron mistake bound**

Example: Spam Filtering

	viagra	learning	the	dating	nigeria	*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)

Linear Classification Rules

- **Hypotheses of the form**
  - unbiased: $h_{\vec{w}}(\vec{x}) = \begin{cases} 1 & w_1x_1 + ... + w_Nx_N > 0 \\ -1 & else \end{cases}$
  - biased: $h_{\vec{w},b}(\vec{x}) = \begin{cases} 1 & w_1x_1 + ... + w_Nx_N + b > 0 \\ -1 & else \end{cases}$
  - 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**
  - $w_1x_1 + ... + w_Nx_N = \vec{w} \cdot \vec{x}$ and $sign(a) = \begin{cases} 1 & a > 0 \\ -1 & else \end{cases}$
  - $h_{\vec{w}}(\vec{x}) = sign(\vec{w} \cdot \vec{x})$
  - $h_{\vec{w},b}(\vec{x}) = sign(\vec{w} \cdot \vec{x} + b)$

(Online) Perceptron Algorithm

- Input: $S = ((\vec{x}_1, y_1), ..., (\vec{x}_n, y_n))$, $\vec{x}_i \in \Re^N$, $y_i \in \{-1, 1\}$, $\eta \in \Re$
- Algorithm:
  - $\vec{w}_0 = \vec{0}$, $k = 0$
  - 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
- Output: $\vec{w}_k$

Margin of a Linear Classifier

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

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

Definition: *The (hard) margin of an unbiased linear classifier* $h_{\vec{w}}$ *on a sample S is* $\delta = min_{(\vec{x},y) \in S} y(\vec{w} \cdot \vec{x})$.

Definition: *The (hard) margin of an unbiased linear classifier* $h_{\vec{w}}$ *on a task* $P(X, Y)$ *is*

$$\delta = inf_{S \sim P(X,Y)} min_{(\vec{x},y) \in S} y(\vec{w} \cdot \vec{x}).$$

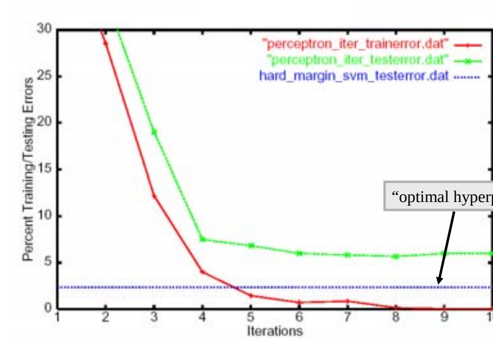
(Batch) Perceptron Algorithm

Input: $S = ((\vec{x}_1, y_1), ..., (\vec{x}_n, y_n))$, $\vec{x}_i \in \Re^N$, $y_i \in \{-1, 1\}$, $\eta \in \Re$, $I \in [1, 2, ..]$

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

Example: Reuters Text Classification



"optimal hyperplane"