Linear Classifiers and Perceptron

CS678 Advanced Topics in Machine Learning
Thorsten Joachims
Spring 2003

Outline:
- Linear classifiers
- Example: text classification
- Perceptron learning algorithm
- Mistake bound for Perceptron
- Separation margin
- Dual representation

Text Classification

E.D. And F. MAN TO BUY INTO HONG KONG FIRM
The U.K. Based commodity house E.D. And F. Man Ltd and Singapore’s Yeo Huap Seng Ltd jointly announced that Man will buy a substantial stake in Yeo’s 71.1 pct held unit, Yeo Hiap Seng Enterprises Ltd. Man will develop the locally listed soft drinks manufacturer into a securities and commodities brokerage arm and will rename the firm Man Pacific (Holdings) Ltd.

About a corporate acquisition?
JA
NEIN

Learning Text Classifiers

Training Set
Real-World Process
New Documents
Learner
Classifier

Goal:
- Learner uses training set to find classifier with low prediction error.

Generative vs. Discriminative Training

Process:
- Generator: Generates descriptions \( \hat{x} \) according to distribution \( P(\hat{x}) \).
- Teacher: Assigns a value \( y \) to each description \( \hat{x} \) based on \( P(y|\hat{x}) \).

\[ (x_1, y_1), \ldots, (x_n, y_n) \sim P(\hat{x}, y) \quad \hat{x} \in \mathbb{R}^N \quad y \in \{1, -1\} \]

Generative Training
- make assumptions about the parametric form of \( P(x, y) \).
- estimate the parameters of \( P(x, y) \) from the training data
- derive optimal classifier using Bayes’ rule
- example: naive Bayes

Discriminative Training
- make assumptions about the set \( H \) of classifiers
- estimate error of classifiers in \( H \) from the training data
- select classifier with lowest error rate
- example: SVM, decision tree
**Representing Text as Attribute Vectors**

**Attributes:** Words (Word-Stems)

**Values:** Occurrence-Frequencies

---

The ordering of words is ignored!

---

**Linear Classifiers (Example)**

Text Classification: Physics (+1) versus Receipts (-1)

<table>
<thead>
<tr>
<th>ID</th>
<th>nuclear (x1)</th>
<th>atom (x2)</th>
<th>salt (x3)</th>
<th>pepper (x4)</th>
<th>water (x5)</th>
<th>heat (x6)</th>
<th>and (x7)</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>+1</td>
</tr>
<tr>
<td>D2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>D3</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>+1</td>
</tr>
<tr>
<td>D4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
</tr>
</tbody>
</table>

| w,b | 2          | 3          | -1        | -3          | -1         | -1        | 0        | b=1 |

D1: $\sum_{i=1}^{7} w_i x_i + b = [2 \cdot 1 + 3 \cdot 2 + (-1) \cdot 0 + (-3) \cdot 0 + (-1) \cdot 2 + (-1) \cdot 0 + 0 \cdot 2] + 1$

D2: $\sum_{i=1}^{7} w_i x_i + b = [2 \cdot 0 + 3 \cdot 0 + (-1) \cdot 0 + (-3) \cdot 3 + (-1) \cdot 0 + (-1) \cdot 1 + 0 \cdot 1] + 1$

---

**Linear Classifiers**

**Rules of the Form:** weight vector $\vec{w}$, threshold $b$

$$h(x) = \text{sign} \left[ \sum_{i=1}^{N} w_i x_i + b \right] = \begin{cases} 1 & \text{if } \sum_{i=1}^{N} w_i x_i + b > 0 \\ -1 & \text{else} \end{cases}$$

**Geometric Interpretation (Hyperplane):**

---

**Perceptron (Rosenblatt)**

Input: $S = (x_1 y_1), \ldots, (x_n y_n) \in \mathbb{R}^N \ y_i \in \{1, -1\}$ (linear separable)

1. $w_0 \leftarrow 0; b_0 \leftarrow 0; k \leftarrow 0$
2. $R = \max \| \vec{x} \|$
3. repeat
   - for $i=1$ to $n$
     - if $y_i (w_k \cdot \vec{x}_i + b_k) \leq 0$
       - $w_{k+1} \leftarrow w_k + \eta y_i \vec{x}_i$
       - $b_{k+1} \leftarrow b_k + \eta y_i^2$
       - $k \leftarrow k + 1$
     - endif
   - endfor
4. until no mistakes made in the for loop
5. return $(w_k, b_k)$
**Analysis of Perceptron**

**Definition (Margin of an Example):** The margin of an example \((\hat{x}_i, y_i)\) with respect to the hyperplane \((\hat{w}, b)\) is

\[ \delta_i = y_i (w \cdot x_i + b) \]

**Definition (Margin of an Example):** The margin of a training set \(S = (\hat{x}_1, y_1), \ldots, (\hat{x}_n, y_n)\) with respect to the hyperplane \((\hat{w}, b)\) is

\[ \delta = \min_i y_i (w \cdot x_i + b) \]

**Theorem (Novikoff):** If for a training set \(S\) there exists a weight vector with margin \(\delta\), then the perceptron makes at most

\[ \frac{\delta_0^2}{4\delta^2} \]

mistakes before returning a separating hyperplane.

**Dual Perceptron**

- For each example \((\hat{x}_i, y_i)\), count with \(\alpha_i\) the number of times the perceptron algorithm makes a mistake on it. Then

\[ \hat{w} = \sum_{i=1}^{n} \alpha_i y_i \hat{x}_i \]

- \(\alpha = 0; b_0 \leftarrow 0\) and \(R = \max_i \|x_i\|\)

- repeat
  - for \(i = 1\) to \(n\)
    - if \(y_i \sum_{i=1}^{n} \alpha_i y_i (\hat{x}_j \cdot \hat{x}_i) + b_0 \leq 0\)
      - \(\alpha_i \leftarrow \alpha_i + 1\)
      - \(b \leftarrow b + y_i R^2\)
    - endif
  - endfor
- until no mistakes made in the for loop
- return \(\hat{w}, b\)

---

**Experiment: Perceptron for Text Classification**

Train on 1000 pos / 1000 neg examples for “acq” (Reuters-21578).