These are the slides of the lecture

Pattern Recognition
Winter term 2011/12
Friedrich-Alexander University of Erlangen-Nuremberg.

These slides are for your personal usage only in order to prepare for the examination.

Publication, reproduction, and distribution of this material is not permitted without prior approval.

Erlangen, April 25, 2018
Dr.-Ing. Stefan Steidl

Pattern Recognition (PR)

Winter Term 2011/12

Stefan Steidl
Computer Science Dept. 5
(Pattern Recognition)

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Rosenblatt’s Perceptron (1957)

- Motivation
- Objective Function
- Minimization of Objective Function
- Remarks on Perceptron Learning
- Convergence of Learning Algorithm
- Lessons Learned
- Further Readings
- Comprehensive Questions

Motivation

- We want to compute a linear decision boundary.
- We assume that classes are linearly separable.
- Computation of a linear separating hyperplane that minimizes the distance of misclassified feature vectors to the decision boundary.

 © 2005-2012 Hornegger, Hahn, Steidl

Objective Function

Assume the following:

- Class numbers are $y = \pm 1$
- The decision boundary is a linear function:

$$y^* = \text{sgn}(\boldsymbol{\alpha}^T \boldsymbol{x} + \alpha_0).$$

Objective Function

Assume the following:

- Class numbers are $y = \pm 1$
- The decision boundary is a linear function:

$$y^* = \mathrm{sgn}(\boldsymbol{\alpha}^T \boldsymbol{x} + \alpha_0).$$

- Parameters α_0 and $\boldsymbol{\alpha}$ are chosen according to the optimization problem

$$\text{minimize} \qquad D(\alpha_0, \boldsymbol{\alpha}) = -\sum_{\boldsymbol{x}_i \in \mathcal{M}} y_i \cdot (\boldsymbol{\alpha}^T \boldsymbol{x}_i + \alpha_0)$$

  where $\mathcal{M}$ includes the misclassified feature vectors.

Objective Function (cont.)

- The elements of the sum in the objective function depend on the set of misclassified feature vectors $\mathcal{M}$.
- In each iteration step the cardinality of $\mathcal{M}$ might change.
- The cardinality of $\mathcal{M}$ is a discrete variable.
- Competing variables: continuous parameters of linear decision boundary and the discrete cardinality of $\mathcal{M}$.

Minimization of Objective Function

Remember the objective function $D(\alpha_0, \boldsymbol{\alpha})$:

$$\text{minimize} \qquad D(\alpha_0, \boldsymbol{\alpha}) = -\sum_{\boldsymbol{x}_i \in \mathcal{M}} y_i \cdot (\boldsymbol{\alpha}^T \boldsymbol{x}_i + \alpha_0)$$

Minimization of Objective Function

Remember the objective function $D(\alpha_0, \boldsymbol{\alpha})$:

$$\text{minimize} \qquad D(\alpha_0, \boldsymbol{\alpha}) = -\sum_{\boldsymbol{x}_i \in \mathcal{M}} y_i \cdot (\boldsymbol{\alpha}^T \boldsymbol{x}_i + \alpha_0)$$

The gradient of the objective function is:

Minimization of Objective Function

Remember the objective function $D(\alpha_0, \boldsymbol{\alpha})$:

$$\text{minimize} \qquad D(\alpha_0, \boldsymbol{\alpha}) = -\sum_{\boldsymbol{x}_i \in \mathcal{M}} y_i \cdot (\boldsymbol{\alpha}^T \boldsymbol{x}_i + \alpha_0)$$

The gradient of the objective function is:

$$\frac{\partial}{\partial \alpha_0} D(\alpha_0, \boldsymbol{\alpha}) \quad = \quad -\sum_{\boldsymbol{x}_i \in \mathcal{M}} y_i$$

Minimization of Objective Function

Remember the objective function $D(\alpha_0, \boldsymbol{\alpha})$:

$$\text{minimize} \qquad D(\alpha_0, \boldsymbol{\alpha}) = -\sum_{\boldsymbol{x}_i \in \mathcal{M}} y_i \cdot (\boldsymbol{\alpha}^T \boldsymbol{x}_i + \alpha_0)$$

The gradient of the objective function is:

$$\begin{aligned} \frac{\partial}{\partial \alpha_0} D(\alpha_0, \boldsymbol{\alpha}) &= -\sum_{\boldsymbol{x}_i \in \mathcal{M}} y_i \\ \frac{\partial}{\partial \boldsymbol{\alpha}} D(\alpha_0, \boldsymbol{\alpha}) &= -\sum_{\boldsymbol{x}_i \in \mathcal{M}} y_i \cdot \boldsymbol{x}_i \end{aligned}$$

Minimization of Objective Function (cont.)

We want to take an update step right after having visited each misclassified observation. The update rule in the $(k+1)$-st iteration step is:

$$\begin{pmatrix} \alpha_0^{(k+1)} \\ \boldsymbol{\alpha}^{(k+1)} \end{pmatrix} =$$

© 2005-2012 Hornegger, Hahn, Steidl

Minimization of Objective Function (cont.)

We want to take an update step right after having visited each misclassified observation. The update rule in the $(k+1)$-st iteration step is:

$$\begin{pmatrix} \alpha_0^{(k+1)} \\ \boldsymbol{\alpha}^{(k+1)} \end{pmatrix} = \begin{pmatrix} \alpha_0^{(k)} \\ \boldsymbol{\alpha}^{(k)} \end{pmatrix} + \lambda \begin{pmatrix} y_i \\ y_i \cdot \boldsymbol{x}_i \end{pmatrix}$$

Here λ is the learning rate which can be set to 1 without loss of generality.

Minimization of Objective Function (cont.)

Input: training data: $S = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), (\boldsymbol{x}_3, y_3), \dots, (\boldsymbol{x}_m, y_m)\}$

Minimization of Objective Function (cont.)

Input: training data: $S = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), (\boldsymbol{x}_3, y_3), \ldots, (\boldsymbol{x}_m, y_m)\}$
initialize $k = 0$, $\alpha_0^{(0)} = 0$ and $\boldsymbol{\alpha}^{(0)} = \mathbf{0}$
repeat
select pair $(\boldsymbol{x}_i, y_i)$ from training set.

Minimization of Objective Function (cont.)

Input: training data: $S = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), (\boldsymbol{x}_3, y_3), \dots, (\boldsymbol{x}_m, y_m)\}$
initialize $k = 0$, $\alpha_0^{(0)} = 0$ and $\boldsymbol{\alpha}^{(0)} = \mathbf{0}$
repeat
select pair $(\boldsymbol{x}_i, y_i)$ from training set.
if $y_i \cdot (\boldsymbol{x}_i^T \boldsymbol{\alpha}^{(k)} + \alpha_0^{(k)}) \leq 0$ **then**

$$\begin{pmatrix} \alpha_0^{(k+1)} \\ \boldsymbol{\alpha}^{(k+1)} \end{pmatrix} = \begin{pmatrix} \alpha_0^{(k)} \\ \boldsymbol{\alpha}^{(k)} \end{pmatrix} + \begin{pmatrix} y_i \\ y_i \cdot \boldsymbol{x}_i \end{pmatrix}$$

$k \leftarrow k + 1$
end if

Minimization of Objective Function (cont.)

Input: training data: $S = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), (\boldsymbol{x}_3, y_3), \ldots, (\boldsymbol{x}_m, y_m)\}$

initialize $k = 0$, $\alpha_0^{(0)} = 0$ and $\boldsymbol{\alpha}^{(0)} = \mathbf{0}$

repeat

select pair $(\boldsymbol{x}_i, y_i)$ from training set.

if $y_i \cdot (\boldsymbol{x}_i^T \boldsymbol{\alpha}^{(k)} + \alpha_0^{(k)}) \leq 0$ **then**

$$\begin{pmatrix} \alpha_0^{(k+1)} \\ \boldsymbol{\alpha}^{(k+1)} \end{pmatrix} = \begin{pmatrix} \alpha_0^{(k)} \\ \boldsymbol{\alpha}^{(k)} \end{pmatrix} + \begin{pmatrix} y_i \\ y_i \cdot \boldsymbol{x}_i \end{pmatrix}$$

$k \leftarrow k + 1$

end if

until $y_i \cdot (\boldsymbol{x}_i^T \boldsymbol{\alpha}^{(k)} + \alpha_0^{(k)}) > 0$ for all i

Output: $\alpha_0^{(k)}$ and $\boldsymbol{\alpha}^{(k)}$

Remarks on Perceptron Learning

- The update rule is extremely simple.
- Nothing happens if we classify all $\boldsymbol{x}_i$ correctly using the given linear decision boundary.
- The parameter α of the decision boundary is a linear combination of feature vectors.

Remarks on Perceptron Learning

- The update rule is extremely simple.
- Nothing happens if we classify all $\boldsymbol{x}_i$ correctly using the given linear decision boundary.
- The parameter α of the decision boundary is a linear combination of feature vectors.
- The decision boundary thus is:

$$F(\boldsymbol{x}) = \left(\sum_{i\in\mathcal{E}} y_i \cdot \boldsymbol{x}_i\right)^T \boldsymbol{x} + \sum_{i\in\mathcal{E}} y_i \quad = \quad \sum_{i\in\mathcal{E}} y_i \cdot \langle \boldsymbol{x}_i, \boldsymbol{x} \rangle + \sum_{i\in\mathcal{E}} y_i$$

where $\mathcal{E}$ is the set of indices that required an update.

Remarks on Perceptron Learning (cont.)

- The final linear decision boundary depends on the initialization, i. e. $\alpha_0^{(0)}$ and $\boldsymbol{\alpha}^{(0)}$.

- The number of iterations can be rather large.

- If data are not linearly separable, the proposed learning algorithm will not converge. The algorithm will end up in hard to detect cycles.