



Blatt 13

Aufgabe 13.1

Skizzieren Sie qualitativ die verschiedenen Fälle, die bei der Trennung von 4 Punkten in zwei Klassen durch eine orientierte Gerade auftreten.

Zeigen Sie (skizzenhaft), daß die VC-Dimension von $\text{sign}(\vec{\omega}^\top \vec{x})$,

wobei \vec{x} mit Bias dargestellt wird $\vec{x} = (1, x_1, x_2)$, (die Indikatorfunktion trennt also die Ebene an der Geraden) maximal 3 sein kann.

Aufgabe 13.2

Zeigen Sie, daß die VC-Dimension (beim Zweiklassenproblem) für folgende Indikatorfunktion *mindestens* $N+1$ ist:

$$f_i(x) = \text{sign}\left(\sum_{i=1}^N \omega_i \phi_i(x) + \omega_0\right)$$

Dabei sind die ϕ_i radiale Basisfunktionen (also z.B. Gauss Funktionen mit Parametern Mittelwert und Varianz), die ω_i sind Gewichte und x ist die eindimensionale Eingabe.

Aufgabe 13.3

Unter `/home/williams/public_html/svm-toy` finden Sie ein Demonstrationsprogramm, mit dem man eine Supportvektormaschine mit 2D Merkmalen trainieren und sich die entstehenden Klassengebiete anzeigen lassen kann. Es gehört zur LIBSVM, einer Library für SVMs, im Netz unter <http://www.csie.ntu.edu.tw/~cjlin/libsvm/> zu finden. Es sollte sich im Linux-Pool ohne Probleme direkt ausführen lassen.

- (a) Spielen Sie ein wenig mit verschiedenen Punktmengen und Einstellungen.
- (b) Laden Sie Punktmenge1 (unter `/home/williams/public_html/Punktmenge1.txt`).
Vergleichen Sie die Klassengebiete bei folgenden Parametern miteinander:
 - (i) `-t 1 -c 100`
 - (ii) `-t 1 -c 10000`

Erklären Sie, woher die Unterschiede kommen. Welches Ergebnis ist besser? Warum?

- (c) Benutzen Sie Punktmenge2 und als Parameter `-t 1 -c 100000 -d 3`. Erklären Sie, wieso die Trennfläche nicht-linear sein kann! Welchen Einfluß hat der Parameter C?
-

options:

`-s svm_type` : set type of SVM (default 0)

0 -- C-SVC

1 -- nu-SVC

2 -- one-class SVM

3 -- epsilon-SVR

4 -- nu-SVR

`-t kernel_type` : set type of kernel function (default 2)

0 -- linear: $u \cdot v$

1 -- polynomial: $(\gamma \cdot u \cdot v + \text{coef0})^{\text{degree}}$

2 -- radial basis function: $\exp(-\gamma \cdot |u-v|^2)$

3 -- sigmoid: $\tanh(\gamma \cdot u \cdot v + \text{coef0})$

`-d degree` : set degree in kernel function (default 3)

`-g gamma` : set gamma in kernel function (default $1/k$)

`-r coef0` : set coef0 in kernel function (default 0)

`-c cost` : set the parameter C of C-SVC, epsilon-SVR, and nu-SVR (default 1)

`-n nu` : set the parameter nu of nu-SVC, one-class SVM, and nu-SVR (default 0.5)

`-p epsilon` : set the epsilon in loss function of epsilon-SVR (default 0.1)

`-m cachesize` : set cache memory size in MB (default 40)

`-e epsilon` : set tolerance of termination criterion (default 0.001)

`-h shrinking`: whether to use the shrinking heuristics, 0 or 1 (default 1)

`-b probability_estimates`: whether to train a SVC or SVR model for probability estimates, 0 or 1 (default 0)

`-wi weight`: set the parameter C of class i to $\text{weight} \cdot C$, for C-SVC (default 1)

The k in the `-g` option means the number of attributes in the input data.

option `-v` randomly splits the data into n parts and calculates cross validation accuracy/mean squared error on them.