

Übungen zum Modul WPM6: Algorithmisches Lernen

SS 2010 Blatt 4

Ausgabe: 08.06.2011, Besprechung: 22.06.2011

Aufgabe 4.1 Präsenzübung: (Manuelles) Setzen der Trennlinie:

Gegeben seien zwei Klassen:

Klasse 1: $\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \end{pmatrix}$

Klasse 2: $\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

Stellen Sie die Punkte Grafisch dar (z.B. gnuplot) und bestimmen Sie die optimale Trennlinie. Welches sind die Support-Vektoren? Wie groß ist der Margin?

Aufgabe 4.2 Präsenzübung: Einarbeitung in LIBSVM:

Laden Sie sich von der Seite www.csie.ntu.edu.tw/~cjlin/libsvm/ (oder über google "libsvm") die libsvmtool herunter. Machen Sie sich mit folgenden Tools vertraut: svm-train, svm-predict, svm-scale, subset.py (in /tools) Informationen hierzu liefert auch die Einführung "A practical guide to SVM classification".

Aufgabe 4.3 Präsenzübung: Lernen der XOR-Funktion:

Erzeugen Sie sich im Texteditor Ihrer Wahl einen Datensatz zur XOR-Funktion. Diese Textdatei hat folgenden Inhalt:

-1 1:0.000 2:0.000

-1 1:1.000 2:1.000

1 1:1.000 2:0.000

1 1:0.000 2:1.000

Die erste Zahl in der Spalte gibt die Klasseneinteilung an, die Parameter sind mit 1,2,... durchnummeriert, die Werte sollten normiert werden $[0/ + 1]$.

Trainieren sie nun diese Funktion mit den verschiedenen Kernels und testen Sie, indem Sie das erzeugte Modell auf den Ursprungsdatensatz anwenden.

Testen Sie das Modell mit manuell verrauschten Daten.

Aufgabe 4.4 Verifikation von Aufgabe 4.1: Lernen Sie die Klassenzuteilung aus Aufgabe 4.1 mit einer SVM analog zu der Vorgehensweise in Aufgabe 4.3. Vergleichen Sie die Ergebnisse.

Aufgabe 4.5 Test mit echten Daten:

Probieren Sie einen oder mehrere Test-Datensätze von der Seite aus. Benutzen Sie das Werkzeug subset.py zur Aufteilung in Trainings- und Messdatensatz. Untersuchen Sie folgende Aspekte:

- Welches Verhältnis Trainingsdaten/Messdaten erscheint bei Ihrem gewählten Datensatz sinnvoll?
- Welche Auswirkung hat vorheriges Skalieren?
- Welche Kernel bringen ein gutes Ergebnis?
- Kann Cross-Validation das Ergebnis verbessern?

