

Technical Report

Pattern Recognition and Image Processing Group
Institute of Computer Aided Automation
Vienna University of Technology
Favoritenstr. 9/183-2
A-1040 Vienna AUSTRIA
Phone: +43 (1) 58801-18351
Fax: +43 (1) 58801-18392
E-mail:
URL: <http://www.prip.tuwien.ac.at/>

January 22, 2010

NC LU - Abgabe 2

Kern, Weichselbaum

Abstract

Documentation for the first exercise of Neural Computation LU, Group 8.

1 Einfhrung

2 SVM, Aufgabe 2

Die Daten erzeugen wir mit derselben Funktion wie bei Aufgabe 1. Dies wird ueber ie Std. Abweichung und Durchschnitt geregelt.

Die *alpha* werden mit Matlab's quadprog erzeugt. Diese quadprog Funktion wird gefuettert mit der Lagrange-Funktion in Matrix-Form $-0.5\boldsymbol{\alpha}^T \mathbf{H}\boldsymbol{\alpha} + \mathbf{f}^T \boldsymbol{\alpha}$. H in unserem Fall ist $H_{ij} = y_i y_j (\mathbf{x}_i \mathbf{x}_j)$ und f ist der Einheitsvektor $f = \mathbf{1} = [1 \dots 1 \dots 1]^T$. Als Constraints verwenden wir $\alpha > 0$ und $y^T \alpha = 0$.

Da wir den Gewichtsvektor nicht augmentieren berechnen wir uns den Bias mit den Supportvektoren. Dies geschieht ueber

$$b_0 = \frac{1}{N_{SV}} * \sum_{s=1}^{N_{SV}} \left(\frac{1}{y_s} - \mathbf{x}_s^T \mathbf{w}_0 \right), s = 1, N_{SV}$$

und wird innerhalb der decision rule, welche wir in *predictSVM()* implementiert haben, verwendet.

Laut Folien wird bei der decision rule das innere Produkt von \mathbf{X} und \mathbf{x}_i verwendet. Dies tun wir nicht und speichern den Gewichtsvektor direkt ab um nicht jedesmal zu summieren bzw. um den Code wiederverwenden zu koennen.

Bei linear seperierbare Daten ist bei einer Crossvalidation von 80/20 eine 100%ige Vorhersage des Klassenlabels gegeben.

Abbildung 1 zeigt die 10 Datensaeetze und die Hyperplane fuer die 10 geforderten linear seperierbaren Datensaeetze.

Fuer **Beispiel 3** haben wir einige Daten kopiert und fuer den RBF-Kernel angepasst. Die Funktion *rbfkernel()* wird mit dem regulaeren Kernel (dh. ein gausscher RBF-Kernel) implementiert und im umgeschriebenen *predictSVM* verwendet.

Anders als in Beispiel 2.2 wurde hier das innere Produkt verwendet und das *sigma* als Parameter erweitert, so dass ein Test der Vorhersagefaehigkeit unserer SVM durchgefuehrt werden konnte.

Leider wurde dadurch die Faehigkeit verloren, den Gewichtsvektor im vorhinein

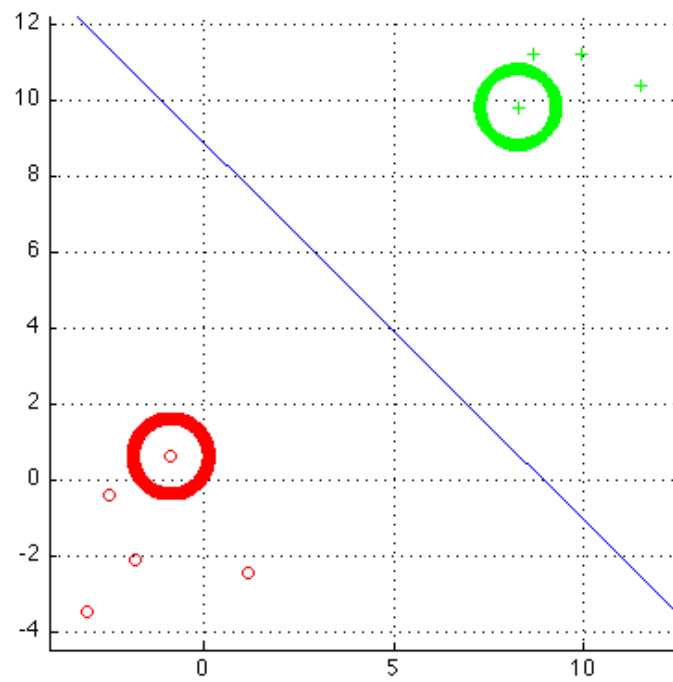


Figure 1: Datensatz, linear seperierbar, geplottet in Matlab. + ist Klasse 1, o signalisiert Klasse -1. Vektoren mit einem Kreis zeigen die Supportvektoren. Die Hyperplane ist optimal, da der Margin zwischen den Supportvektoren maximal ist.

zu berechnen und nur mehr mit der neuen Beobachtung zu multiplizieren. Allerdings hat diese Einschränkung keinen Einfluss auf die Performance des Systems, da wir lediglich 100 2D-Test-Datensätze erzeugt haben.

Die Werte für σ haben keinen Einfluss auf die Entscheidung bei linear separierbaren Daten, wenn sie so deutlich voneinander entfernt sind wie in unserem Fall. Erst wenn die Supportvektoren nahe der Hyperplane sind ist die Auswahl von σ entscheidend.

σ stellt die Standardabweichung innerhalb der Klassen dar, welche wir erwarten. Liegen die beiden Klassen nun nah beieinander besteht die Möglichkeit, dass diese sich bei einer großen Standardabweichung überschneiden und Datensätze nicht mehr eindeutig einer Klasse zugeordnet werden können.

Dies wurde probiert mit Werten von 0.1 bis 100 in 0.1 Schritten, keine neuen Beobachtungen wurden falsch vorhergesagt.

Bei den Schlupfvariablen haben wir die Lektüre von Kecan [1] verwendet. Diese besagt, dass sich die Hesse-Notation der Lagrange-Funktion nicht verändert. Lediglich die Constraints für α ändern sich. Der Upperbound von α ist nun nicht unendlich, sondern der Regularisierungsparameter C .

Dies ist insofern bedeutend, als dass die Vektoren, welche nicht linear geteilt werden können, zu Supportvektoren der Größe C gemacht werden.

Abbildung 2 zeigt die Vektoren in 2D-Darstellung und deren tatsächlichen vorhergesagten Wert (d.h. $\text{predictSVM}()$ ohne signum).

References

- [1] Vojislav Kecan. *Learning and Soft Computing*. MIT Press, 2001.

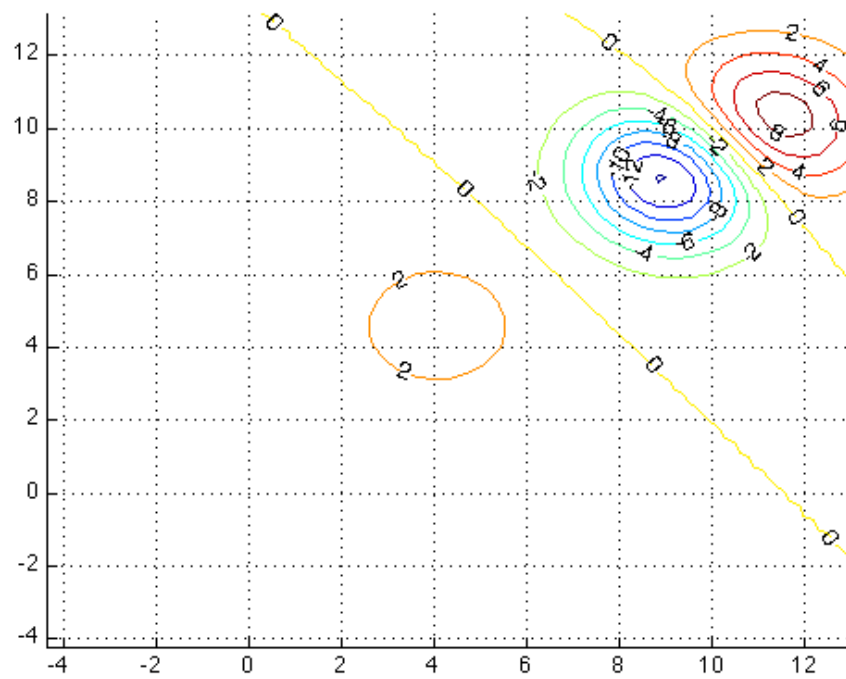


Figure 2: Datensatz, nicht linear seperierbar, mit contour in Matlab geplottet. Zeigt die Struktur des Datensatzes an Hand der gefunden Klassen.