

Maschinelles Lernen 2

Sommersemester 2007

Blatt 4

Abgabe 28. Mai 2007 in der Vorlesung, oder bis 23:59:59 Uhr bei `mikio@cs.tu-berlin.de`

Die Aufgaben können auch in Gruppen bearbeitet werden, allerdings sollte die Gruppenzusammenstellung über des Semester stabil bleiben.

Aufgaben

Nachdem auf dem letzten Übungszettel die Grundlagen der Optimierungstheorie behandelt wurden, betrachten wir jetzt typische Probleme, wie sie im Bereich des maschinellen Lernen auftauchen. Am häufigsten treten die folgenden Probleme auf: (1) Optimierungsprobleme als Lineares Programm / Quadratisches Programm umschreiben, (2) die duale Funktion bestimmen, (3) das resultierende Problem an einen Optimierer übergeben.

- Aufstellen eines Optimierungsproblems (10 Punkte)** Tatsächlich lassen sich durch geschicktes Einfügen von Slackvariablen mehr Optimierungsprobleme als lineare oder quadratische Optimierungsprobleme darstellen, als auf den ersten Blick erscheint.

Forme die folgenden (für sich nicht besonders sinnvollen Optimierungsprobleme) entsprechend um, so daß sowohl die zu optimierende Funktion als auch die Nebenbedingung linear sind.

- $\sum_i |\alpha_i|$
- $\sum_i \max(\alpha_i, 0)$
- $\sum_i \max(0, 1 - y_i f(x_i))$ (soft-margin Lossfunktion)
- $\sum_i |y_i - f(x_i)|$ (ℓ_1 -Lossfunktion).

- Berechnen des dualen Problems (10 Punkte)** Zur Erinnerung: Zur Berechnung des dualen Problems wird die Lagrangefunktion bezüglich der ursprünglichen Variablen minimiert und anschließend bezüglich der Lagrangevariablen maximiert.

Bestimme das Duale Optimierungsproblem für die Linear Programming Machine:

$$\begin{aligned} \min_{\alpha, \xi} \quad & \sum_{i=1}^n \alpha_i + C \sum_{i=1}^n \xi_i \\ \text{subject to} \quad & y_i \left(\sum_{j=1}^n \alpha_j x_j^\top x_i + b \right) \geq 1 - \xi_i, \quad 1 \leq i \leq n, \\ & \alpha_i \geq 0, \xi_i \geq 0, \quad 1 \leq i \leq n, \end{aligned}$$

- Aufrufen eines Optimierers (10 Punkte)** Für gewöhnlich operiert man eher mit Mengen von Gleichungen als direkt mit linearen Nebenbedingungen. Der Grund liegt auf der Hand: Die meisten Nebenbedingungen behandeln nur wenige Variablen, so daß die Matrix überdurchschnittlich viele Nullen enthalten.

Möchte man das Problem tatsächlich lösen, so wird man zunächst verfügbare Optimierer verwenden wollen, zum Beispiel die der Optimierungstoolbox von Matlab. Hierfür müssen jetzt die Nebenbedingungsmatrizen aufgestellt werden.

Die `linprog` Funktion von Matlab gibt es in zwei Varianten:

Aufruf	Optimiert
<code>linprog(f, A, b)</code>	$\min_x f^\top x$ mit $Ax \leq b$
<code>linprog(f, A, b, C, d)</code>	$\min_x f^\top x$ mit $Ax \leq b, Cx = d$

Bestimme die Aufrufparameter für die Linear Programming Machine.