Übung zur Vorlesung

Maschinelles Lernen 2

Sommersemester 2007

Abteilung Maschinelles Lernen
Institut für Softwaretechnik und
theoretische Informatik
Fakultät IV, Technische Universität Berlin
Prof. Dr. Klaus-Robert Müller
Email: krm@cs.tu-berlin.de

Blatt 2

Abgabe **verlängert bis zum 15. 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 Semster stabil bleiben. Für praktische Aufgaben bitte ebenfalls Code und Ausgabe (d.h. Ergebnisse, Plots) abgeben. Verwende Matlab oder Octave. Lehrversionen von Matlab sind im cs-Netz (IRB) installiert. Zum Start `ml/bin/matlab` aufrufen. Octave ist ein freie Matlabclone, der unter `www.octave.org` verfügbar ist. Für die Abgabe von praktischen Aufgaben bitte die Coding-Richtlinien beachten, die für das Praktikum gelten (siehe hierzu `http://ml.cs.tu-berlin.de/de/ss07_ml_praktikum.html`, Hinweise auf dem ersten Übungsblatt).

Aufgaben

In diesem Übungsblatt wollen wir die Verbindungen zwischen Boosting und Gradientenabstieg ausnutzen, um Boosting-Methoden auf Regressionsprobleme zu erweitern, d.h. für die Beobachtungen $(\boldsymbol{x}_i, y_i)$ gilt $\boldsymbol{x}_i \in \mathbb{R}^d$ und $y_i \in \mathbb{R}$. In diesem Fall verwendet man gewöhnlich die quadratische Verlustfunktion

$$L_2(y, y') \quad = \quad \frac{1}{2}(y - y')^2 \; .$$

(Der Faktor 1/2 ist irrelevant für die Minimierung des quadratisschen Verlusts, vereinfacht aber die Notation im Gradientenverfahren.)

Im Gegensatz zur Klassifikation gibt es in Regressionsproblemen nur eine informelle Definition von schwachen Lernern. Ein schwacher Lerner ist ein Verfahren, dass „underfittet“. Schwache Lerner sind beispielsweise Ridge-Regression oder *penalized least squares* (vgl. Blatt 9 des Wintersemesters) mit hohen Werten für den Regularisierungsparameter λ. Einem weiteren schwachen Lerner werden wir in Aufgabe 3 begegnen.

1. **L_2-Gradientenabstieg (5 Punkte)** Zeige, dass der Gradientenabstieg für die quadratische Verlustfunktion äquivalent zu einem iterativen Lernen auf den Residuen ist, d.h. der negative Gradient u_i aus Gleichung (2) von Blatt 1 entspricht den Residuen

   $$r_i \quad = \quad y_i - f_m(\boldsymbol{x}_i) \, .$$

   Bemerkung: Für die meisten schwachen Lerner kann man zeigen, dass die optimale Schrittlänge $\alpha_m = 1$ ist. Eine Variante von L_2-Boosting besteht darin, die Schrittlänge vorher festzulegen (s. Aufgabe 3).

2. **Linearität von L_2-Boosting (10 Punkte)** Im letzten Wintersemester (Blatt 9) haben wir den Begriff „linearer Lerner“ kennengelernt. „Linear“ bedeutet in diesem Fall, dass man den Lerner in der Form

   $$\widehat{\boldsymbol{y}} \quad = \quad \boldsymbol{H}\boldsymbol{y}$$

   schreiben kann, wobei $\boldsymbol{y}$ der Vektor ist, der alle Label des Trainingsdatensatzes enthält, $\widehat{\boldsymbol{y}}$ die Vorhersage auf dem Trainingsdatensatz ist, und $\boldsymbol{H}$ eine Matrix ist, die von $\boldsymbol{X}$, aber nicht von $\boldsymbol{y}$ abhängt.

   Zeige, dass L_2-Boosting ein lineares Lernverfahren ist, falls die einzelnen schwachen Lerner linear sind. Zeige genauer: Falls in jeder Iteration m ein linearer (schwacher) Lerner $\boldsymbol{H}_m$ angewandt wird, so wird L_2-Boosting durch die Matrix

   $$\boldsymbol{B}_M = \sum_{m=1}^{M} \boldsymbol{H}_m \prod_{m'=1}^{m-1} (\boldsymbol{I} - \boldsymbol{H}_{m'})$$

   repräsentiert.

   Gehe hierbei wie folgt vor:

(a) Zeige, daß der Residuenvektor im $m$ten Schritt $\boldsymbol{r}_m$ (d.h. der Vektor $(r_1, \ldots, r_n)$ im $m$ten Schritt) folgende Rekursionsformel erfüllt:

$$\boldsymbol{r}_m = \boldsymbol{r}_{m-1} - \boldsymbol{H}_{m-1}\boldsymbol{r}_{m-1} = (\boldsymbol{I} - \boldsymbol{H}_{m-1})\boldsymbol{r}_{m-1}$$

Verwende hierzu, daß

$$\boldsymbol{r}_m = \boldsymbol{y} - \boldsymbol{f}_{m-1}, \qquad \boldsymbol{f}_m = \boldsymbol{f}_{m-1} + \boldsymbol{h}_m, \qquad \boldsymbol{h}_m = \boldsymbol{H}_m\boldsymbol{r}_m, \qquad \boldsymbol{f}_0 = \boldsymbol{0}.$$

wobei $\boldsymbol{h}_m = (h_m(\boldsymbol{x}_1), \ldots, h_m(\boldsymbol{x}_n))$, und entsprechend $\boldsymbol{f}_m$ der Vektor aller Vorhersagen auf dem Trainingsdatensatz ist.

(b) Zeige nun, daß

$$\boldsymbol{f}_M = \boldsymbol{B}_M\boldsymbol{y} = \sum_{m=1}^{M} \boldsymbol{H}_m \prod_{m'=1}^{m-1} (\boldsymbol{I} - \boldsymbol{H}_{m'})\boldsymbol{y}. \tag{1}$$

(c) *(Zusatzaufgabe)* Zeige, daß sich die Matrix $\boldsymbol{B}_M$ auch schreiben läßt als

$$\boldsymbol{B}_M = \boldsymbol{I}_n - (\boldsymbol{I}_n - \boldsymbol{H}_M)\,(\boldsymbol{I}_n - \boldsymbol{H}_{M-1}) \ldots (\boldsymbol{I}_n - \boldsymbol{H}_1)$$

Bemerkung: Es ist also möglich, die Freiheitsgrade (vgl. Blatt 9 des Wintersemesters) von L_2-Boosting zu definieren.

3. **Implemetierung (10 Punkte)** Implementiere den L_2-Algorithmus. Verwende als schwachen Lerner *komponentenweise Kleinste-Quadrate-Regression*: Wähle in jeder Boosting-Iteration die optimale Variable X_{j^*} aus. Für diese Variable ist h_m das Ergebnis der Kleinste-Quadrate-Regression

$$\gamma_0 + \gamma_1 X_{j^*}$$

angewandt auf $\boldsymbol{x}_i$ und die Residuen r_i (siehe unten). Dabei ist die Variable X_{j^*} optimal, falls sie von allen d Variablen den Trainingsfehler am meisten reduziert. Wähle als konstante Schrittlänge $\alpha_m = 0.1$.

Bemerkung: Der Vorteil dieses schwachen Lerners ist, dass i.A. nur eine kleine Anzahl der d Variablen in der endgültigen Lösung auftauchen, L_2-Boosting liefert also i.A. dünn besetzte Lösungen.

4. **Anwendung (5 Punkte)** Wende L_2-Boosting auf den `prostate`-Datensatz an, den man unter

   http://www-stat.stanford.edu/~tibs/ElemStatLearn/

   findet. Verwende dabei nur die Trainingsdaten. Plotte die Anzahl der Variablen, die in der Boosting-Lösung auftauchen, als Funktion der Anzahl der Boosting-Iterationen. Bestimme die optimale Anzahl der Boosting-Iterationen, indem Du den Testfehler für alle Boostingschritte bestimmst. Plotte den Trainings- und Testfehler als Funktion der Anzahl der Boosting-Iterationen. Gib die ausgewählten Variablen für die optimale Anzahl von Boosting-Iterationen aus.

Hinweise

Wir betrachten ein eindimensionales Regressionsproblem

$$Y \quad = \quad \gamma_0 + \gamma_1 X + \text{Fehler}$$

d.h. $\boldsymbol{x}_i \in \mathbb{R}$. Der Kleinste-Quadrate-Schätzer ist die Lösung des Minimierungsproblems

$$\min_{\gamma_0,\gamma_1} \sum_{i=1}^{n} (y_i - \gamma_0 - \gamma_1 \boldsymbol{x}_i)^2 \ .$$

Die Lösung lautet

$$\begin{aligned} \widehat{\gamma}_1 &= \frac{\operatorname{cov}(\boldsymbol{X}, \boldsymbol{y})}{\operatorname{var}(\boldsymbol{X})} \\ \widehat{\gamma}_0 &= \overline{y} - \widehat{\gamma}_1 \overline{\boldsymbol{X}} \, . \end{aligned}$$