

Maschinelles Lernen 2

Sommersemester 2008

Blatt 11

Abgabe 1. Juli 2008 in der Vorlesung. Praktische Übungsaufgaben über PASS abgeben (<https://ml01.zrz.tu-berlin.de/~mikio/pass.pl?conf=blatt11.conf>). Verwende matlab (/home/ml/ml/bin), oder octave (<http://www.octave.org> frei verfügbar).

Aufgaben

1. (10 Punkte) Das SVM-Optimierungsproblem mit “Hard-Margin” lautet

$$\min_w \frac{1}{2} \|w\|^2$$

so dass $y_i \langle w, x_i \rangle \geq 1, \quad 1 \leq i \leq n.$

Formuliere dieses Optimierungsproblem so um, daß der margin ρ maximiert wird:

$$\max_{w, \rho} \rho$$

so dass $y_i \langle w, x_i \rangle \geq \rho, \quad 1 \leq i \leq n,$
 $\|w\|_2 = 1.$

2. (5 Punkte) Zeige, dass der Gradientenabstieg (siehe Rückseite) für die quadratische Verlustfunktion

$$L(y, y') = \frac{1}{2} (y - y')^2$$

äquivalent zu einem iterativen Lernen auf den Residuen ist, d.h. der negative Gradient

$$u_i = - \left. \frac{\partial L(y, f)}{\partial f} \right|_{f=f_m(x_i)}, \quad 1 \leq i \leq n,$$

entspricht dem Residuum $r_i = y_i - f_m(x_i)$.

3. (15 Punkte) Implementiere Boosting für Regression (also Gradientenabstieg mit dem L_2 -Fehler, siehe Rückseite), wobei Du als schwachen Lerner lineare Regression in einer einzelnen Koordinate verwendest (d.h., man betrachtet nur eine Dimension in x). Wähle diese Koordinate so aus, dass der Fehler minimal ist. Addiere den schwachen Lerner jedoch mit einem Faktor $\alpha < 1$. Schreibe hierzu eine Funktion

```
C = regboostlinear(X, Y, alpha, M)
```

wobei X und Y die Daten sind, und M die Anzahl der Boostingiterationen. Der Rückgabewert C ist eine Struktur, die alle Informationen, die zur Vorhersage nötig sind, enthalten soll.

Schreibe außerdem eine Funktion

```
Y = predict_regboostlinear(C, X)
```

welche auf X die Vorhersagen Y berechnet.

4. (10 Punkte) Wende `regboostlinear` auf die Datensätze `prostate` und `banana` von der Vorlesungswebseite an. Schreibe hierfür ein Skript `aufgabe3.m`, die die Datensätze lädt, und jeweils `regboostlinear` mit geeigneten Parametern lädt, und dann die gelernten Gewichte pro Dimension ausgibt.

Algorithm 1 Gradientenabstieg

Eingabe: Verlustfunktion $L(y, y')$, Anzahl der Iterationen M

Wende einen schwachen Lerner h_1 auf die Daten an und setze $f_1 = h_1$

for $m = 2, \dots, M$ **do**

 Berechne den negativen Gradienten der Verlustfunktion

$$u_i = - \left. \frac{\partial L(y, f)}{\partial f} \right|_{f=f_m(x_i)}, i = 1, \dots, n. \quad (1)$$

 Wende einen schwachen Lerner h_m auf die **modifizierten** Daten (\mathbf{x}_i, u_i) an

 Bestimme die optimale Schrittweite

$$\alpha_m = \arg \min_{\alpha} \left\{ \sum_{i=1}^n L(y_i, f_{m-1}(\mathbf{x}_i) + \alpha h_m(\mathbf{x}_i)) \right\}. \quad (2)$$

 Aktualisiere

$$f_m = f_{m-1}(\mathbf{x}) + \alpha_m h_m(\mathbf{x}),$$

end for

return f_m
