Technische Universität Dortmund
Fakultät Statistik
Lehrstuhl Computergestützte Statistik
Fakultät Informatik
Lehrstuhl für Künstliche Intelligenz

Prof. Dr. Claus Weihs
Dr. Thomas Liebig
Sebastian Buschjäger
Malte Jastrow
Jennifer Neuhaus-Stern

Übung zur Vorlesung

Wissensentdeckung in Datenbanken

Sommersemester 2018

Übungsblatt Nr. 5

Der Abgabetermin ist Dienstag der 05.06.2018 bis 10:00 Uhr im moodle-Raum

Aufgabe 1 (Gradientenabstieg) **(10 Punkte)**

In dieser Aufgabe sollen Sie einen Gradientenabstieg implementieren. Gegeben sei dazu die folgende Modellfunktion $f_{c,\boldsymbol{a},\boldsymbol{b}} : \mathbb{R}^d \to \mathbb{R}$ mit Koeffizienten $c, \boldsymbol{a}, \boldsymbol{b}$

$$f_{c,\boldsymbol{a},\boldsymbol{b}}(\boldsymbol{x}) = c + \sum_{i=1}^{d} \boldsymbol{a}_i \cdot \boldsymbol{x}_i + \sum_{i=1}^{d} \sum_{j=i}^{d} \boldsymbol{b}_{i,j} \cdot \boldsymbol{x}_i \cdot \boldsymbol{x}_j,$$

wobei $\boldsymbol{x}_i$ den i-ten Eintrag des Vektors $\boldsymbol{x}$ bezeichne.
Zum Testen Ihrer Implementierung benutzen Sie bitte den `housing.csv` Datensatz, der im Moodle-Raum verfügbar ist. Dieser Datensatz enthält 506 Trainingsbeispiele mit 13 Attributen. Beachten Sie, dass die Attributwerte (nicht das Label!) bereits auf das Intervall $[-1, 1]$ normiert sind.
Ziel dieses Datensatzes ist es anhand von regionalen Eigenschaften den mittleren Hauspreis in 1000er Schritten (14te Spalte) in Boston vorherzusagen. Weitere Informationen zu diesem Datensatz finden Sie unter `https://archive.ics.uci.edu/ml/datasets/Housing`.
Bitte dokumentieren Sie ihre Implementierung entsprechend und stellen Sie sicher, dass ihre Implementierung eigenständig ausführbar ist.

a) (1 Punkt) Berechnen Sie den Gradienten von f bzgl. $\boldsymbol{a}, \boldsymbol{b}$ und c.

b) (5 Punkte) Implementieren Sie einen Gradientenabstieg in `R`, um die Koeffizienten c, $\boldsymbol{a}$ und $\boldsymbol{b}$ zu lernen. Benutzen Sie hierzu als Verlustfunktion den Residual Sum of Squares Error (RSS):

$$\ell(c, \boldsymbol{a}, \boldsymbol{b}; \mathcal{D}) = \frac{1}{2} \sum_{(\boldsymbol{x},y)\in\mathcal{D}} \left(y - f_{c,\boldsymbol{a},\boldsymbol{b}}(\boldsymbol{x})\right)^2 .$$

Welchen Fehler erreichen Sie nach 100 Schritten mit einer Schrittweite $\eta_t = 0.0002$?
Was passiert bei einer Schrittweite $\eta_t = 0.002$?

Hinweis: Überlegen Sie sich zunächst wie viele Parameter das Modell f hat und wie sie diese am geschicktesten darstellen.

c) (2 Punkte) In der Vorlesung wurde gezeigt, dass wenn der Gradient der Verlustfunktion ℓ Lipschitz-Stetig mit Konstante L ist, führt die Wahl der Schrittweite von $\eta_t = \frac{1}{L}$ bei konvexen Funktionen zur Konvergenz zu einem globalen Optimum. Bestimmen Sie L für $\nabla\ell(f_{c,\boldsymbol{a},\boldsymbol{b}}; \mathcal{D})$ und geben Sie eine geeignete Schrittweite η_t an.
Welchen Fehler erreichen Sie jetzt nach 100 Schritten mit der neuen Schrittweite?

Hinweis: Folgende obere Schranke gilt: Falls $L \leq \sup_{\boldsymbol{\beta}} ||\nabla F(\boldsymbol{\beta})||_2$, so ist die Funktion $F : \mathbb{R}^n \to \mathbb{R}$ Lipschitz-Stetig mit Konstante L. Beachten Sie, dass die Frobenius-Norm zu verwenden ist, falls ∇F eine Matrix ist. Es genügt also die Frobenius-Norm der Hesse-Matrix nach oben hin abzuschätzen, wobei sie die Tatsache verwenden dürften dass die Beispiele bereits auf das Intervall $[-1, 1]$ normiert sind.

d) (1 Punkt) Welche Änderungen müssten Sie an Ihrer Implementierung durchführen, wenn Sie einen l_1 Regularisierer zu ℓ hinzufügen? Welche Besonderheit gibt es hier?

<u>Hinweis:</u> Sie müssen den regularisierten Gradientenabstieg nicht implementieren. Bitte beschreiben Sie ihr Vorgehen dennoch so detailliert wie möglich und geben Sie die notwendigen Formeln exakt an.

e) (1 Punkte) Inwiefern ändert sich die Lipschitz-Konstante L_{alt} aus Aufgabe c), wenn Sie eine l_1 Regularisierung hinzufügen?

<u>Hinweis:</u> Sie müssen L_{alt} nicht explizit kennen um diese Aufgabe zu lösen. Verwenden Sie die Definition der Lipschitz-Stetigkeit: $|\nabla\ell(\boldsymbol{\beta};\mathcal{D}) - \nabla\ell(\boldsymbol{\beta}';\mathcal{D})| \leq L|\boldsymbol{\beta} - \boldsymbol{\beta}'|$. Des Weiteren ist die Dreiecksungleichung an dieser Stelle hilfreich, d.h. die Ungleichung
$|x+y| \leq |x| + |y| \; \forall x, y \in \mathbb{R}$.

Aufgabe 2 (Support Vector Machine (SVM)) (5 Punkte)

In dieser Aufgaben sollen Sie die SVM besser kennenlernen. Im Moodle finden Sie die Datei `wid_SoSe18_tm05.Rmd`, in der eine Ebene $\beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 = \beta_0$ definiert wird. Mit Hilfe dieser Ebene werden synthetische Daten erzeugt, welche sich mit einer Ebenengleichung perfekt trennen lassen (100% Klassifikationsgenauigkeit).

a) (2 Punkte) Plotten Sie die generierten Daten und färben Sie diese der Klasse entsprechend ein. Visualisieren Sie anschließend die Entscheidungsebene mit folgendem Vorgehen:

1) Stellen Sie zunächst die Ebenengleichung nach einer Variable x_i um.

2) Ziehen Sie zufällig M zwei-dimensionale Punkte im Interval $[-1, 1]^2$ für die übrigen beiden Variable x_j und x_k

3) Setzen Sie die generierten Punkte in die umgestellte Ebenengleichung ein um so die Ebene "abzutasten".

<u>Hinweis:</u> Für dynamische 3D Plots können Sie den `scatter3Drgl` Befehl des `plot3Drgl` Paketes verwenden.

b) (3 Punkte) Benutzen Sie die `svm` Methode des `e1071` Paketes um auf den generierten Daten eine lineare SVM zu trainieren. Berechnen Sie anschließend den (primalen) Gewichtsvektor der trainierten SVM aus den dualen Gewichten und den Support-Vektoren. Vergleichen Sie die Gewichte der vorgegebenen Ebene mit den von der SVM berechneten Gewichten. Erklären Sie den Unterschied!
<u>Hinweis:</u> Das `svm` Methode skaliert die Daten intern, was zu Problemen bei der Interpretation führen kann. Stellen sie dies mit dem Parameter `scale=c(0,0,0)` aus.

Aufgabe 3 (Mean-Squared-Error vs. Log-Likelihood) (5 Punkte)

Angenommen Sie trainieren ein lineares Regressionsmodell (Folie 6, Foliensatz `WiD 180522 Lineares Modell, Bias-Varianz.pdf`) der Form $f_{\boldsymbol{\beta}}(\boldsymbol{x}^i) = \langle \boldsymbol{x}^i, \boldsymbol{\beta} \rangle$ mit Hilfe eines Trainingsdatensatze $\mathcal{D} = \{(\boldsymbol{x}^1, y^1), \ldots, (\boldsymbol{x}^N, y^N)\}$. Da Ihre Messungen nicht perfekt sind, nehmen Sie an, dass die Labels mit einem normalverteilten Fehler behaftet sind, d.h: $y^i \sim \mathcal{N}(\mu^i, \sigma^2)$. Hier bezeichnet μ^i den wahren Messwert (ohne Fehler) und σ^2 die Varianz des Fehlers (welcher für alle Messungen gleich ist).
Zeigen Sie unter Annahme, dass Ihre Modellannahmen stimmen, d.h. $f(\boldsymbol{x}^i) = \mu^i$, dass das Minimieren des RSS (siehe Aufgabe 1) zur gleichen Lösung wie der Maximum Liklihood-Schätzer führt:

$$\arg\max_{\mu \in \mathcal{M}} \log \mathcal{L}(p_{\mu,\sigma}; \mathcal{D}) = \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^n} \mathrm{RSS}(f_{\boldsymbol{\beta}}; \mathcal{D})$$

<u>Hinweis:</u> Überlegen Sie sich zunächst, über welche Menge $\mathcal{M}$ der Parameter μ optimiert wird