Ludwig-Maximilians-Universität München 01.04.2014
Institut für Informatik
Prof. Dr. Volker Tresp
Gregor Jossé

Maschinelles Lernen und Data Mining

Sommersemester 2014

Übungsblatt 1

Besprechung des Übungsblattes am 24.04.2014

Aufgabe 1-1 Lineare Algebra

Gegeben seien die Vektoren $\mathbf{a} = (1, 2, 1)^T$ und $\mathbf{b} = (2, 2, 1)^T$, sowie die Matrizen

$\mathbf{A} = \begin{pmatrix} 1 & 2 \\ 2 & 1 \\ 1 & 1 \end{pmatrix}$, $\mathbf{B} = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$ und $\mathbf{C} = \begin{pmatrix} 1 & 3 \\ 0 & 1 \end{pmatrix}$.

a) Berechnen Sie per Hand oder in einer Programmiersprache Ihrer Wahl (gerne mit bereits verfügbaren Matrixoperationen):
$\mathbf{a}^T\mathbf{b}$, $\mathbf{a}\mathbf{b}^T$, $\mathbf{AC}$, $\mathbf{CA}^T$, $\mathbf{A}^T\mathbf{a}$, $\mathbf{a}^T\mathbf{A}$.

b) Invertieren Sie $\mathbf{B}$ und überprüfen Sie, ob für die gewählte Sprache gilt $\mathbf{B}^{-1}\mathbf{B} = \mathbf{B}\mathbf{B}^{-1} = I$

c) Generieren Sie eine 3×3 orthonormale Matrix. Prüfen Sie ob Spalten und Zeilen orthonormal sind.

Aufgabe 1-2 Die ADALINE Lernregel

Das *ada*ptive *lin*ear *e*lement (ADALINE) Modell verwendet die *least mean square* Kostenfunktion

$$\text{cost} = \frac{1}{2}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 ,$$

für N Trainingselemente, wobei y_i das tatsächliche und $\hat{y}_i$ das berechnete Klassenlabel von Muster i ist. Im Gegensatz zum einfachen Perceptron erfolgt die Klassifizierung nicht über eine Signum-Funktion sondern direkt: $\hat{y} = h$. (Zur Erinnerung: M ist die Anzahl der Merkmale der Muster $x_i \in \mathbb{R}^M$ und des Gewichtungsvektors $w \in \mathbb{R}^M$, wobei $x_0 = 1$ konstant ist und dem Bias entspricht.)

a) Leiten Sie das gradientenbasierte Lernverfahren für einen ADALINE Prozess her. (Optimieren Sie analog zur Perceptron-Lernregel nach w.)

b) Geben Sie hierzu die musterbasierte Lernregel an.

c) Welche Vorteile haben musterbasierte Lernregeln?

d) Nennen Sie die wichtigsten Unterschiede des ADALINE-Modells zur in der Vorlesung kennengelernten Perceptron-Lernregel.

Aufgabe 1-3 Anwendung der Perceptron-Lernregel

Gegeben seien zwei Klassen A und B, die jeweils 2 Muster umfassen:

$$A = \left\{ p_1 = \begin{pmatrix} 2 \\ 4 \end{pmatrix},\ p_2 = \begin{pmatrix} 1 \\ 0.5 \end{pmatrix} \right\}, \qquad B = \left\{ p_3 = \begin{pmatrix} 0.5 \\ 1.5 \end{pmatrix},\ p_4 = \begin{pmatrix} 0 \\ 0.5 \end{pmatrix} \right\}$$

Die Klassenlabels y der Klasse A sind deklariert mit 1, die von B mit -1.

Lösen Sie die folgenden Aufgaben mittels einer Programmiersprache Ihrer Wahl oder mit der guten alten Pen-and-Paper-Methode und geben Sie die Zwischenschritte graphisch mit an:

a) Wieviele Iterations-Schritte benötigt die musterbasierte Perceptron-Lernregel um die Klassen A und B zu separieren wenn der Gewichtungsvektor w mit $(0, 1, -1)$ initialisiert wird und eine Schrittweite $\eta = 0.1$ verwendet wird?

b) Wieviele Iterations-Schritte brauchen wir für $\eta = 0.25$? Spielt hierbei die Reihenfolge der betrachteten Muster eine Rolle? Wenn ja geben Sie ein Beispiel zweier unterschiedlicher Einfügereihenfolgen, ansonsten beweisen Sie das Gegenteil.

c) Nach wievielen Iterationen terminiert die gradientenbasierte Lernregel jeweils für beide η? Spielt hier die Einfügereihenfolge eine Rolle?

Kleiner Hinweis: Bei korrekter Bearbeitung dieser Aufgaben sollten nicht mehr als jeweils 10 Iterationen notwendig werden.

Aufgabe 1-4 Das Perceptron in mehr als zwei Dimensionen

a) Die Zahlen von 1 bis 9 und 0 wurden in der Vorlesung als Pixel-Arrays repräsentiert. Die Datenmatrix hierzu finden Sie in der Datei `numberMatrix.RData`. Trainieren Sie ein Perceptron darauf, gerade von ungeraden Zahlen zu unterscheiden. Variieren Sie w und η. Terminiert das Perceptron auch für das Problem "ist ein Vielfaches von drei"?

b) Was ist die Trainings-Komplexität eines Perzeptron für einen M-dimensionalen Datensatz der Größe N? Was kostet danach eine Vorhersage?