Ludwig-Maximilians-Universitaet Muenchen 24.04.2015
Institute for Informatics
Prof. Dr. Volker Tresp
Gregor Jossé
Johannes Niedermayer

Machine Learning and Data Mining

Summer 2015

Exercise Sheet 1

Presentation of Solutions to the Exercise Sheet on the 29.04.2015

Exercise 1-1 Linear Algebra

Let $\mathbf{a} = (1, 2, 1)^T$ and $\mathbf{b} = (2, 2, 1)^T$ be vectors and let

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

a) Calculate the following results (either with pen and paper or a programming language of your choice):
$\mathbf{a}^T\mathbf{b}$, $\mathbf{a}\mathbf{b}^T$, $\mathbf{AC}$, $\mathbf{CA}^T$, $\mathbf{A}^T\mathbf{a}$, $\mathbf{a}^T\mathbf{A}$.

b) Invert $\mathbf{B}$ and check if $\mathbf{B}^{-1}\mathbf{B} = \mathbf{B}\mathbf{B}^{-1} = I$ holds.

c) Generate an orthonormal 3×3 matrix. Check if rows and columns are indeed orthonormal.

Exercise 1-2 Recap: Vector Calculus

Compute $\frac{\partial g(\mathbf{x})}{\partial \mathbf{x}}$ the functions below. *Hint:* For a function $g(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ miti $\mathbf{x} \in \mathbb{R}^n$ holds:

$$\frac{\partial g(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial g(\mathbf{x})}{\partial \mathbf{x}_1} \\ \frac{\partial g(\mathbf{x})}{\partial \mathbf{x}_2} \\ \vdots \\ \frac{\partial g(\mathbf{x})}{\partial \mathbf{x}_n} \end{bmatrix}.$$

a) $g(\mathbf{x}) = \sum_{i=1}^{n} \mathbf{x}_i$,

b) $g(\mathbf{x}) = \langle \mathbf{x}, \mathbf{x} \rangle$, the standard scalar product of $\mathbf{x}$ with itself,

c) $g(\mathbf{x}) = (\mathbf{x} - \mu)^2$ für $\mu \in \mathbb{R}^n$.

Exercise 1-3 Boolean Function as Perceptron

Consider the boolean function *or* ($\vee$) for two binary inputs.

- a) Illustrate the different inputs as well as possible separating hyperplanes grafically.
- b) Given the above picture, guess weights for a perceptron (with outputs 0 and 1) such that the perceptron is a classifier for the $\vee$ function. Instead of using the $sign$ function, as in the lecture, use the Heaviside function f for classification:
$$f(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$$
- c) Initialize the weight vector as $w = (0, 0, 0)$ and learn the right weights employing the algorithm of the lecture and a learning rate $\eta = 0.2$. Use the following learning rule:
$$w_i \leftarrow w_i + \eta \cdot (y_i - \hat{y}_i)x_{i,j}$$
. Start training vector x_3 and proceed with increasing index (in contrast to the principle of random sampling).

Exercise 1-4 Applying the perceptron learning rule

Let A and B be two classes, both comprising two patters:

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

Classes A and B are labelled with 1 and -1, respectively.

Solve the following exercises either using pen and paper or a programming language of your choice. Also, visualize the partial results.

a) How many iterations are required by the pattern-based perceptron learning rule in order to seperate classes A and B correctly if the weight vector w is initialized as $(0, 1, -1)$ and step size η is set to 0.1?

b) How many iterations are required if $\eta = 0.25$? Is the order of the considered patterns relevant? If so, give an example, otherwise, prove it.

c) After how many iterations does the gradient-based learning rule terminate for both η? In this case: Is the order of the considered patterns relevant?

Hint: If you need more than 10 iterations, you miscalculated.

Exercise 1-5 The ADALINE learning rule

The *ada*ptive *lin*ear *e*lement (ADALINE) model uses the *least mean square* cost function

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

for N training set elements, where y_i is the actual and $\hat{y}_i$ the computed class label of pattern i. In contrast to the simple perceptron, classification is not realized by the signum-function. Instead, it is done directly: $\hat{y} = h$. (As a reminder: M is the number of input features of patterns $x_i \in \mathbb{R}^M$ and the dimensionality of the weight vector $w \in \mathbb{R}^M$, where $x_0 = 1$ is constant and corresponds to the bias or offset.)

a) Deduce the gradient descent-based learning rule (or: adaption rule) for the ADALINE process (analoguously to the perceptron learning rule).

b) Specify the corresponding sample-based learning rule.

c) What advantages do sample-based learning rules have?

d) Name the most distinctive characteristics between the ADALINE model and the perceptron model.