Machine Learning 1 WS 2014/15

Gedächtnisprotokoll

1. April 2015

Bearbeitungszeit: 120 Minuten, Punkte: 100.

Aufgabe	Punkte
1	15
2	15
3	25
4	20
5	25

Lösungen sind nicht offiziell, falls du Fehler entdeckst bitte bei der Freitagsrunde melden / Wiki eintragen!

Multi Choice

Aufgabe 1

15 Punkte

Es gibt nur eine richtige Antwort. Falsche Antworten geben 0 Punkte genauso wie keine Antwort.

(a) (3 Punkte) Let $f_1(x) \dots f_N(x)$ be a set of descriminats for classification. The classification decision is given by $c* = \underset{i}{\text{argmax}}\ f_i(x)$ Which of the following sets would produce the same classification as the one above?

- ☐ $g_i(x) = (f_i(x))^2$
- ☐ $h_i(x) = log(1 + exp(f_i(x)))$
- ☐ None of them
- ☐ Both of them

(b) (3 Punkte) Consider a two-class classification problem. A sufficient condition for the Bayes optimal classifier to be linear is:

- ☐ The data generating distributions for both classes are equivalent except for the mean
- ☐ The data generating distributions for both classes are Gaussian
- ☐ The data generating distributions for both classes have the same covariance
- ☐ None of the above

(c) (3 Punkte) The Fisher linear discriminant finds the projection that:

- ☐ Maximizes the margin between the two data generating distribution
- ☐ Maximizes the margin between the mean of the two data generating distribution
- ☐ Maximizes the ratio between the within-class variance and the between-class variance
- ☐ Minimizes the ratio between the within-class variance and the between-class variance

(d) (3 Punkte) Which is **false**? Let k be a Gaussian Kernel. A Gram Matrix K associated to this Kernel always satisfies:

- ☐ $K = K^T$
- ☐ $KK^T = I$
- ☐ All Eigenvalues are non-negative
- ☐ $\forall u \in R^N : u^T K u \geq 0$

(e) (3 Punkte) Error Backpropagation is a technique to:

- ☐ Efficiently compute the error gradient in a multilayer neural network.
- ☐ Efficiently compute the error gradient in restricted Boltzmann machine.
- ☐ Efficiently compute the prediction error of a multilayer neural network.
- ☐ Efficiently compute the prediction error of a restricted Boltzmann machine.

Lösung:

(a) Both of them

(b) None of the above

(c) Minimizes the ratio between the within-class variance and the between-class variance

(d) ?

(e) Efficiently compute the error gradient in a multilayer neural network.

Models and Datasets

Aufgabe 2

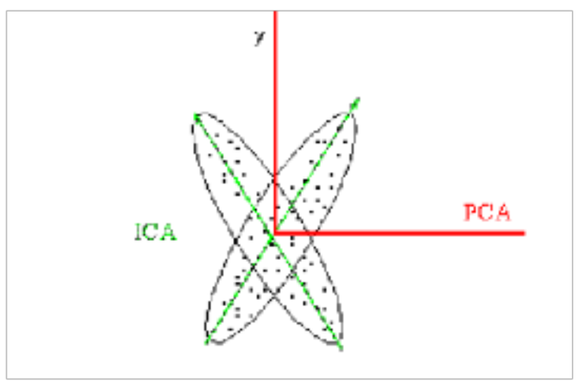
15 Punkte

(a) (7 Punkte) Sketch a two-dimensional unsupervised dataset for which the first PCA component project to a different direction thanthe ICA components learned on the same dataset. Show using arrows the components learned by both methods. Your example must be stereotyping in order to show the difference between the components learned by both methods.

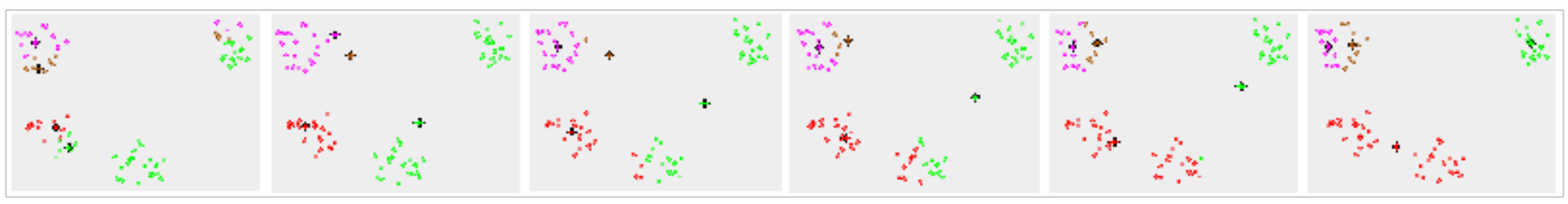
(b) (7 Punkte) Sketch a two-dimensional unsupervised dataset for which K-means (K=10) does not return the correct cluster structure. Your example must be stereotyping in order to show the difference between the two types of clusters.

Lösung:

(a)



(b)



pic from wiki

Bayes Classification

Aufgabe 3

20 Punkte

Consider a two-class problem where each class w_1, w_2 generates binary data $x \in \{0,1\}^d$ corresponding to some propability distribution. In particular each dimension x_i is drawn independently from a Bernoulli distribution.

$$Pr(x_i = 0|w_1) = 1 - p_i$$

$$Pr(x_i = 1|w_1) = p_i$$

$$Pr(x_i = 0|w_2) = p_i$$

$$Pr(x_i = 1|w_2) = 1 - p_i$$

The data generating propability distribution for each class can also be written as:

$$Pr(x|w_1) = \prod_{i=1} p_i^{x_1}(1-p_i)^{1-x_1}$$

$$Pr(x|w_2) = \prod_{i=1}^{d}(1-p_i)^{x_1} p_i^{1-x_1}$$

(a) (5 Punkte) Assume both classes occurring with the same probability ($Pr(w_1) = Pr(w_2) = 0.5$). Show that the liklihood ratio given by $l = \frac{Pr(w_1|x)}{Pr(w_2|x)}$ can be rewritten as $l = \frac{Pr(x|w_1)}{Pr(x|w_2)}$

(b) (10 Punkte) A likelihood ratio of $l > 1$ indicates that class w_1 is more likely then class w_2. given x. Thus if satisfied, it leads to the decision of classifying x as w_1. Show that the inequality $l > 1$ can after multiple transformations be rewritten as a linear inequality of type $a^T x + b > 0$ where $a \in R^d$ $b \in R$ are parameters that depend on probabilities $p_1 \dots p_d$.

(c) (5 Punkte) Give an explicit formula for a and b.

Lösung:

(a) TODO

(b)

(c)

Lagrance multipliers

Aufgabe 4

25 Punkte

For an unsupervised dataset x_N we would like to solve the optimization problem:

$$\min_w ||w||^2$$

subject to the inequality constrains

$$w^T x_i \geq 1 \quad \text{for } 1 \leq i \leq N$$

where $w \in R^d$

(a) (5 Punkte) Give a geometrical interpretation to the optimization problem above and its solution

(b) (5 Punkte) Write down the Lagrangian function $L(w, \alpha_1 \dots \alpha_N)$ associated to the optimization problem where $\alpha_1 \dots \alpha_N$ are Lagrange multipliers for each constrain

(c) (15 Punkte) Derive the Lagrange dual of the program above. Describe how the solution of the primal problem can then be recovered from the solution of the dual.

Lösung:

(a) x

(b) x

(c) x

Programmning Gradient Descent

Aufgabe 5

25 Punkte

We consider the regression model

$$f(x, w) = log(1 + exp(w^T x))$$

that is specially tailored for positive output data. Assume a labeled dataset with $x_i \in R^d$ and $y_i > 0$, we would like to use gradient descent to find the parameter vector w given the regularisation objection:

$$E = \sum_{i=1}^{N} (f(x_i, w) - y_i)^2 + \lambda ||w||^2$$

where λ is some fixed hyperparameter.

(a) (5 Punkte) Compute the gradient of the error function E with respect to the parameter vector w

(b) (15 Punkte) Write a program that performs gradient descent in the parameter space until convergence.

```
X,y = getData()
w = numpy.zeros((d))
# YOUR CODE HERE
return w
```

(c) (5 Punkte) Explain what can be done in order to address the case where large values of x would cause the exponential function to numerically overflow. (Here we assume that we do not have access to bignum or other precision types than the usual float32 float64)

Lösung:

(a)

$$\nabla_w E = 2\lambda w + \sum_{i=1}^{N} 2(f(x_i, w) - y_i) \frac{exp(w^T x_i)}{1 + exp(w^T x_i)} x_i$$

(b)
```
X,y = getData()
w = numpy.zeros((d))
# YOUR CODE HERE
return w
```

(c) I dont know