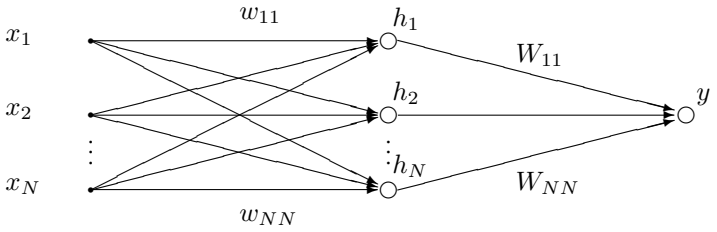
Neural Networks and Learning Systems
TBMI26 / 732A55

Exercise Collection

2022



Neural Networks and Learning Systems
Exercise Collection

Contents

Exercises

1. Introduction and k-nearest neighbors

1.1. Learning types

Machine learning can be divided into three main types. Which?

1.2. Feature

What is a *feature* in the context of machine learning?

1.3. Classification function

a) Mathematically, a classifier can be described as a function $f(\mathbf{x}; w_1, w_2, \ldots, w_n) \rightarrow \Omega$. What is $\mathbf{x}$, $w_1, w_2, \ldots, w_n$ and Ω?

b) What is the difference between regression and classification?

c) How is learning of a classifier for a pattern recognition problem performed?

1.4. Generalization & overtraining

a) It is important that a learning method is able to *generalize*. What does this mean?

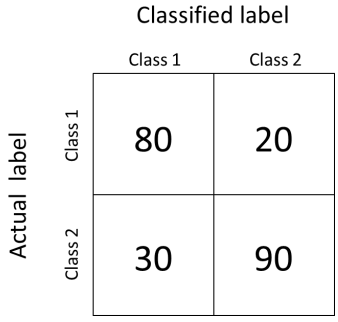
b) In supervised learning, we are usually training a classifier to minimize the error on training data. But what we really want is a low generalization error. How can we estimate the generalization error of a classifier?

c) Why is it important to divide the data set into a training set and a test set when training a learning system?

d) How can you notice if a supervised machine learning algorithm has overtrained?

1.5. Confusion matrices

You evaluate a classifier you have just trained and find the following *confusion matrix*. What is the classification accuracy?



1.6. Cross-validation

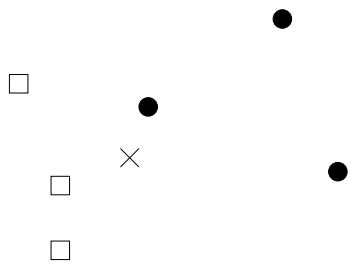
You have 900 labeled training samples and want to evaluate how well different supervised classification algorithms perform for this data. Explain/sketch how you would do this with *3-fold cross-validation*.

1.7. k-Nearest Neighbors

Mention one advantage and one disadvantage of the k-nearest neighbor classifier.

1.8. k-Nearest Neighbors

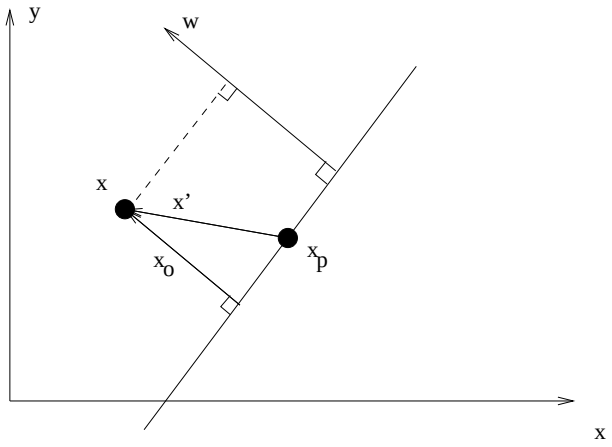
Which class does X belong to using a kNN classifier with $k = 1$? And $k = 3$? Why? How can we handle $k = 2$?



2. Linear classifiers

2.1. Linear algebra reminder

This exercise should work as a small repetition of planes, projections and the scalar products. Define the plane equation in vector format using the normal $\mathbf{w}$, then define how to calculate the distance from the plane to any point $\mathbf{x_0}$.



2.2. The perceptron

a) Write the mathematical expression for the perceptron.

b) What is the purpose of the "bias weight" in a perceptron?

c) What happens if the input to the bias weight in a perceptron is set to the constant value 2?

d) Construct a classification example with two classes which are linearly separable, but which requires a non-zero bias weight for a perfect classification.

2.3. Batch vs. online learning

What is the difference between *batch learning* and *online learning*?

2.4. Partial derivatives of vectors

Find the derivative:

a) $\frac{\partial y}{\partial \mathbf{w}}$ if $y = \mathbf{w}^T\mathbf{x}$

b) $\frac{\partial y}{\partial \mathbf{x}}$ if $y = \mathbf{x}^T\mathbf{W}\mathbf{x}$

c) $\frac{\partial y}{\partial \mathbf{w}}$ if $y = \|\mathbf{w}\|^4$

2.5. Optimization

Minimize the cost function f with respect to $\mathbf{x}$:

a) if $f(\mathbf{x}) = 3\|\mathbf{x}\|^2 + \begin{bmatrix}1 & 1\end{bmatrix}\mathbf{x} - 4$

b) with the gradient descent method for (a) when $\eta = 0.05$ and $x_{old} = \begin{bmatrix}1\\1\end{bmatrix}$. Only two iterations need to be described.

2.6. Gradient descent step length

In gradient descent optimization, for example when training a neural network, what may be the effect of

- a too long step length?
- a too short step length?

Illustrate with figures.

2.7. Cost function

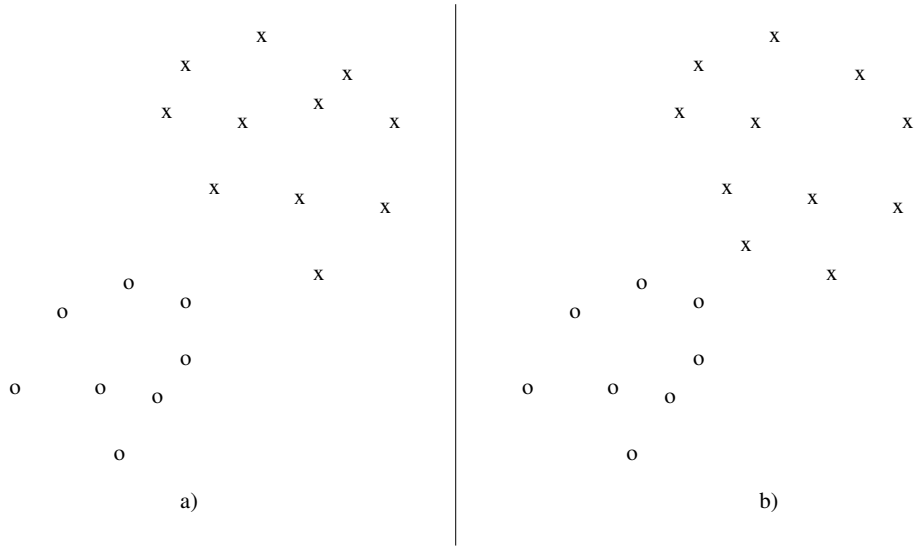
In supervised learning, why is it a problem to train a classifier by minimizing the number of false classifications using gradient descent? That is, to minimize $\sum_{k=1}^{N} I(z_k \neq y_k)$ where z_k is the output from the classifier, y_k is the correct label, $I(z_k \neq y_k)$ is equal to 1 if $z_k \neq y_k$ and 0 otherwise, and N is the number of training examples.

2.8. SVM

What is the *maximum margin* principle that is used, for example, in SVM?

2.9. SVM

Draw the optimal hyperplane (decision boundary) and mark the support vectors in the following two cases:



2.10. SVM

In SVM, an entity is minimized under the following constraint:

$$d_i(\mathbf{w}^T\mathbf{x}_i + w_0) \geq 1.$$

What is being minimized, and for which $\mathbf{x}_i$ is the constraint fulfilled with equality?

2.11. SVM

Are SVMs particularly useful when the training data set is rather small or when it is very large? Motivate!

3. Neural networks and nonlinear classifiers

3.1. XOR

a) Sketch the so-called "XOR problem".

b) Why cannot the "XOR problem" be solved by a one layer perceptron?

3.2. Cover's theorem

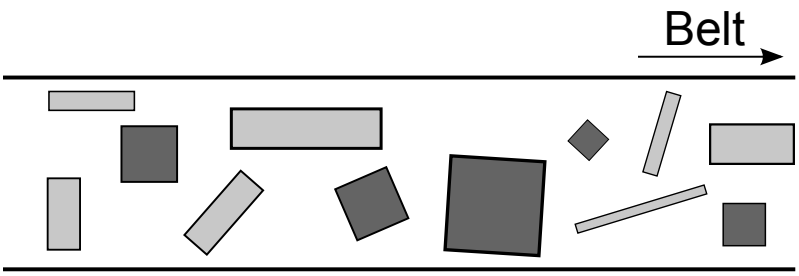
What is the meaning of Cover's theorem in words?

3.3. Nonlinear classification

In an industrial application, two types of objects with square and rectangular (non-square) shapes respectively are transported on a belt. The size of the objects varies. A camera observes the objects and you have access to a function that measures the lengths x_1 and x_2 of two neighboring sides of each object. There is a small amount of noise in the measurement, so that x_1 and x_2 will be slightly different also for square objects.

Using the above measurements, your task is to classify each object as rectangular or square:

- Suggest features and sketch how the features distribute in the feature space for both classes (only the approximate structure is asked for!).
- Suggest a suitable classifier based on this.



*Note: The image is just for illustration, you are **not** required to measure in it!*

3.4. Prediction using a neural network

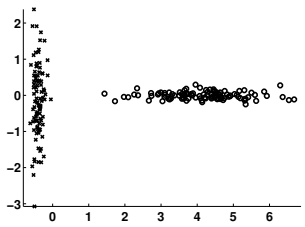
Describe briefly how a neural network could be trained to predict the temperature for the next day.

3.5. Activation functions

a) Is it appropriate to train the network with $y = \pm 1$ if we, for example, use tanh() as activation function?

b) Would we gain anything by using a nonlinear activation function in the output layer, instead of a linear function?

c) Draw the approximate decision boundaries you would get if you trained a linear classifier using the error functions $\epsilon_1 = \sum (y_i - \mathbf{w}^T \mathbf{x}_i)^2$ and $\epsilon_2 = \sum (y_i - \tanh(\mathbf{w}^T \mathbf{x}_i))^2$, $\mathbf{x}_i$ are the feature vectors, $y_i = \pm 1$ are the class labels, and $\mathbf{w}$ are the boundary parameters. Explain how the difference in error functions affects the location of the boundary.



3.6. Neural network training

a) Describe (by an equation) how a momentum term is used in gradient descent, for example when training a neural network.

b) How can the problem of overfitting be avoided when training a neural network?

3.7. Metrics

Distance can be measured in a number of ways and is of great importance in classification. The most intuitive distance from a class with mean μ is the Euclidian distance $d_e = \sqrt{(x-\mu)^T(x-\mu)}$. Two other common distances are RBF $d_r = e^{-d_e^2/\sigma}$ and the Mahalanobis distance $d_m = \sqrt{(x-\mu)^T C^{-1}(x-\mu)}$, where C is the covariance matrix of the class. Which of these three metrics is most suitable for classification of the data in the figure below, and why?

3.8. Backpropagation derivation

a) Derive the batch training error gradient for a two-layer neural network with an arbitrary number of neurons, classes an inputs. The hidden layer shall have an nonlinear activation function while the output layer shall have a linear activation function.

Hint: First do the derivation component-wise for the output layer then component-wise for the hidden layer. Last but not least transfer the result to matrix form.

Also, derive the corresponding update rules for the hidden and output layers.

You will implement this in the computer assignment 1.

b) Rewrite the previous result in on-line training format.

3.9. Manual neural network

You have the following training samples:

$\mathbf{X}$	=	-1.6	-1.4	-1.2	-0.8	-0.4	0	0.3	0.7	0.9	1.1
$\mathbf{D}$	=	-1	-1	-1	1	1	1	1	-1	-1	-1

where $\mathbf{D}$ is the desired output for the samples in $\mathbf{X}$. Create a neural network to separate the data. Use one node in the output layer, and $sign$ as activation function, both in the hidden layer and the output layer. Sketch the network and assign $\mathbf{W}$ (weights in the hidden layer) and $\mathbf{V}$ (weights in the output layer) so that the output gives 100% accuracy.

4. Deep learning

This class will focus on preparing for the Deep Learning computer assignment. We will make sure everyone can access their cloud computer on Microsoft Azure, or alternatively help setup the Python environment on your own computers. After this you are encouraged to go through the introductory Jupyter Notebook that explains how the Keras deep learning API works. You will not have time to do these preparations during the lab.

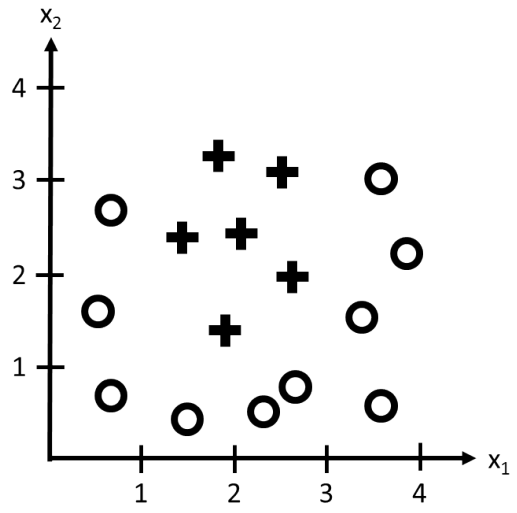
5. Ensemble learning and boosting

5.1. Decision tree & CART

a) What is a *decision tree / CART*?

b) What is the difference between a classification tree and a regression tree?

c) The figure below shows a feature space with two classes. Draw a classification and regression tree (CART) that classifies the two classes and draw the classification boundaries in the figure.



5.2. Decision stump

a) What is a *decision stump*?

b) Which parameters are we optimizing? (Assume one dimension for now)

c) What is the cost function we want to minimize?

d) Why is this cost function always ≤ 0.5 after optimization?

e) How many thresholds do we need to test?

5.3. Ensemble learning

What is the basic idea of ensemble learning?

5.4. Boosting vs bagging.

a) What is *bagging*?

b) What is *boosting*?

5.5. Weak classifier

a) What is a *weak classifier*?

b) Describe one weak classifier.

c) Construct/sketch a very simple "toy" classification task with two classes.

d) How will one instance of your weak classifier solve this classification task?

5.6. Boosting algorithm

Describe/sketch the flow of a general boosting algorithm using pseudo-code. Omit details and equations but include the logic.

5.7. AdaBoost

You have the following data:

$$\mathbf{X} = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \qquad \mathbf{Y}_a = \begin{bmatrix} 1 & 1 & -1 & 1 \end{bmatrix} \qquad \mathbf{Y}_b = \begin{bmatrix} 1 & -1 & -1 & 1 \end{bmatrix}$$

where $\mathbf{X}$ contains four 2d-samples (one per column), and $\mathbf{Y}_a \& \mathbf{Y}_b$ contain classification labels for the corresponding samples.

a) Perform the first AdaBoost iteration on the data $\mathbf{X}$ using the labels $\mathbf{Y}_a$. Sketch the classification problem. Use 'decision stumps' as weak classifiers. Does AdaBoost work well in this setting? (2p)

b) Perform the first AdaBoost iteration on the data $\mathbf{X}$ using the labels $\mathbf{Y}_b$. Sketch the classification problem. Use 'decision stumps' as weak classifiers. Does AdaBoost work well in this setting? (2p)

c) Why is outliers a problem for the standard AdaBoost method? (1p)

Hint: The standard way of updating the weights in the standard AdaBoost method is $d_{t+1}(i) \propto d_t(i) e^{-\alpha_t y_i h_t(\mathbf{x})}$, where $\alpha_t = \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}$. *Note:* This was a question worth 5 points on the exam from 2012-03-22.

6. Recap class

This class is a class where you can catch up to previous classes and also ask question regarding the assignments. No new problems are added for this class.

7. Reinforcement learning

7.1. Feedback

Describe the difference in feedback between supervised and reinforcement learning.

7.2. Delayed feedback

Describe the "temporal credit assignment problem".

7.3. V vs. Q

What is the difference between the value function V and the Q-function in reinforcement learning?

7.4. Discount factor

What does the *discount factor* control in reinforcement learning?

7.5. Exploration vs. exploitation

Explain the principle of ϵ-*greedy exploration*.

7.6. Get the V- and Q-functions 1

The figure shows three different deterministic state models and their corresponding reward functions. The states are numbered from 1 to 5 and arrows represent actions denoted as "up", "same" and "right". The numbers close to the arrows show the reward. If the system reaches a state called "End" no more rewards are given, i.e. the V-function is defined as 0 in these states. With optimal means the policy that maximizes the reward.

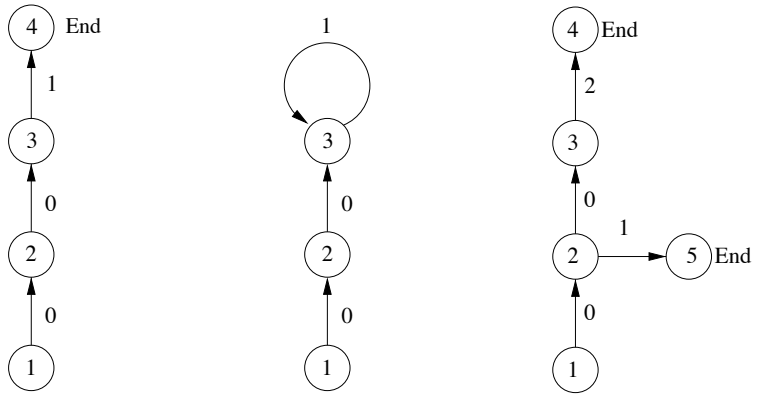


Figure 1: The state models A, B and C.

a) Calculate the optimal Q- and V-function for system A as a function of γ. (1p)

b) Calculate the optimal Q- and V-function for system B as a function of γ. Interpret the result in words. (2p)

c) Calculate the optimal Q- and V-function for system C as a function of γ. Interpret the result in words. (2p)

7.7. Get the V- and Q-functions 2

The figure shows two different deterministic state models and their corresponding reward functions. The states are numbered from 1 to 6 and arrows represent actions denoted as "forward" and "shortcut". The numbers close to the arrows show the reward.

a) Describe the optimal policy for system A and B respectively. (1p)

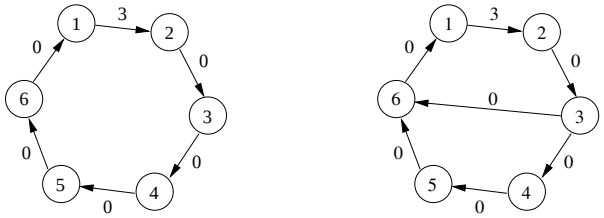


Figure 2: The state models A and B.

b) Calculate the optimal Q- and V-functions for system A as a function of γ ($0 < \gamma < 1$). (2p)

c) Calculate the optimal Q- and V-functions for system B as a function of γ ($0 < \gamma < 1$). (2p)

7.8. Get the V- and Q-functions 3

The figure shows two different deterministic state models and their corresponding stochastic reward functions. The states are numbered from 1 to 6 and arrows represent actions denoted as "up" and "left". The numbers close to the arrows show the reward. In the cases where the reward is stochastic a percentage states the probability of a certain reward.

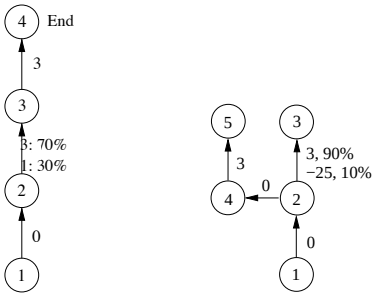


Figure 3: The state models A and B.

a) Calculate the optimal Q- and V-functions for system A as a function of γ ($0 < \gamma < 1$). (2p)

b) Calculate the optimal Q- and V-functions for system B as a function of γ ($0 < \gamma < 1$). (3p)

7.9. Get the V- and Q-functions 4

The figure below shows a state model of a system were the task is to get from state "1" to an end node with maximal reward over time. Find the optimal path using Q-learning with the allowed actions "up", "down" and "right". When trying to move right from state "1" there is a chance p of ending up in state "3", and a chance $1 - p$ of ending up in state "4" ($0 \leq p \leq 1$).

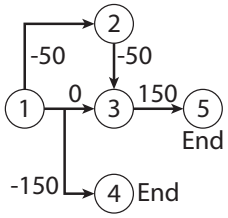


Figure 4: The state model. The numbers next to the arrows are the rewards for moving from one state to another.

a) Calculate the expected Q-function after a very large number of iterations with a learning rate $0 < \alpha < 1$ and discount factor $0 < \gamma \leq 1$.

b) Calculate the expected V function using $\gamma = 1$.

c) Why can we not use $\alpha = 1$ when we train this system?

8. Unsupervised learning and dimensionality reduction

8.1. Cluster belonging

What is meant by *hard clustering* and *soft/fuzzy clustering* respectively?

8.2. Clustering algorithms

a) What is the difference between k-means clustering and mixture of Gaussians clustering?

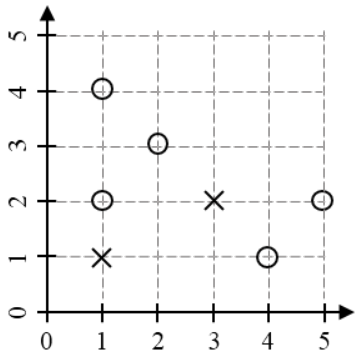
b) What is the optimization algorithm called that is used in k-means clustering and mixture of Gaussians clustering?

8.3. The k parameter

What is determined by the k parameter in the k-nearest neighbors and k-means algorithms respectively?

8.4. k-means algorithm

Perform one iteration of the k-means algorithm on the data plotted below. Circles (o) are input data and crosses (x) are the current prototype vectors. How many more iterations must be made before the algorithm converges?



8.5. Multidimensional Gaussian function

a) What is the density function of the multidimensional Gaussian (normal) distribution?

b) Given a mean vector of $\mathbf{0}$ and covariance matrix $\begin{pmatrix} 3 & -1 \\ -1 & 3 \end{pmatrix}$, calculate the density function for $\mathbf{x_1} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $\mathbf{x_2} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$ and $\mathbf{x_3} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$.

8.6. k-means & Mixture of Gaussians

Assume the following six 2d-samples (written as columns): $\mathbf{X} = \begin{bmatrix} -1 & -1 & -1 & 1 & 1 & 1 \\ -1 & 0 & 1 & -1 & 0 & 1 \end{bmatrix}$.

Assume we have two prototypes ($k = 2$): $\mathbf{p}_1 = \begin{bmatrix} -0.5 \\ 0 \end{bmatrix}$ and $\mathbf{p}_2 = \begin{bmatrix} 0.5 \\ 0 \end{bmatrix}$.

a) Draw a sketch of the samples and prototypes.

b) Perform two iterations of k-means.

c) Perform two iterations of mixture of Gaussians (MoG). Assume initial $\mathbf{C_1} = \mathbf{C_2} = \mathbf{I}$.

d) This exercise is probably more fun in Matlab...

8.7. Statistical concepts: 1-dimensional

Give the expression for the following statistical concepts (one dimensional case):

a) Covariance (according to the definition, between the stochastic variables X and Y using the expectation operator.)

b) Covariance estimation (if you have N samples x_i and y_i respectively)

c) Variance (according to the definition, for the stochastic variables X using the expectation operator.)

d) Variance (expressed as a covariance)

e) Variance estimation (if you have N samples x_i)

f) Correlation (expressed in terms of covariance and variance)

8.8. Statistical concepts: multi-dimensional

Give the expression for the following statistical concepts (multi dimensional case):

a) Covariance (according to the definition, for the stochastic variable $\mathbf{X}$ using the expectation operator.)

b) Covariance matrix estimation (if you have N samples $\mathbf{x_i}$)

c) Cross-covariance (according to the definition, between the stochastic variables $\mathbf{X}$ and $\mathbf{Y}$ using the expectation operator.)

8.9. Covariance matrix interpretation

a) You have a covariance matrix $\mathbf{C}$ describing a set of three dimensional signal samples $\mathbf{x_i}$. Identify the elements of the matrix in terms of variances and covariances of the signal components. Symmetries?

b) You have the same matrix as above. How would you change it to get the correlation matrix?

c) You have calculated the covariance and correlation matrix $\mathrm{Cov}(\mathbf{x})$ and $\mathrm{Corr}(\mathbf{x})$ respectively. Describe $\mathrm{Cov}(5\mathbf{x})$ and $\mathrm{Corr}(5\mathbf{x})$.

d) Assume a (time-varying) vector with signals $\mathbf{z}(t)$. If the covariance matrix $\mathbf{C}_{zz}$ is diagonal, what does this mean?

8.10. Covariance matrix estimation

Find the mean value $\mathbf{m}$ and the covariance matrix $\mathbf{C}$ which describes $\mathbf{x}$ when we have the data samples $\mathbf{x}_1 = \begin{bmatrix} -1 \\ 3 \end{bmatrix}$, $\mathbf{x}_2 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ and $\mathbf{x}_3 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$.

8.11. Dimensionality reduction

You have a data set with 3-dimensional feature vectors and calculate the correlation matrix to be

$$\mathbf{C} = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}$$

Based on this information, explain what we can tell about the relationship between the features. Illustrate how this knowledge can be used.

8.12. Eigendecomposition of the covariance matrix

a) Find the eigenvalues and the eigenvectors to the covariance matrix $\mathbf{C} = \begin{bmatrix} 4 & \sqrt{3} \\ \sqrt{3} & 2 \end{bmatrix}$.

b) How are the eigenvectors related?

c) How are the eigenvalues interpreted?

8.13. Principal component analysis derivation

a) Derive the eigenvalue problem of PCA on the basis of maximizing the variance of the projection $\hat{\mathbf{w}}^T\mathbf{x}$. $\hat{\mathbf{w}}$ is the direction the data $\mathbf{x}$ should be projected upon.

b) Show that maximizing the variance of the projection is equal to minimizing the mean square error $|\mathbf{x} - (\hat{\mathbf{w}}^T\mathbf{x})\hat{\mathbf{w}}|^2$.

8.14. Principal direction

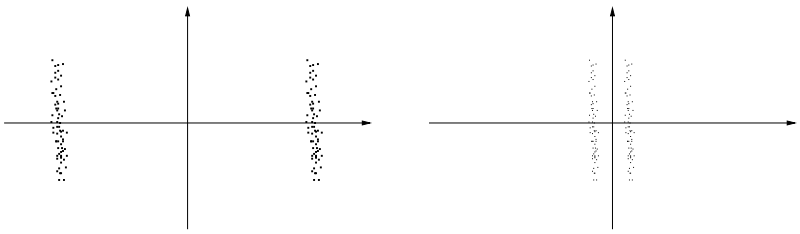
Assume data of three dimensions $\mathbf{z} = (z_1, z_2, z_3)^T$. We calculate the first principal direction (largest eigenvector of the covariance matrix) as $\mathbf{w}_1 = \frac{1}{3}(2, 1, 2)^T$. What does the result mean?

8.15. Shape of a Gaussian distribution

A Gaussian distribution of two dimensions has the mean $\mathbf{0}$ and the covariance $\begin{pmatrix} 3 & -1 \\ -1 & 3 \end{pmatrix}$.
Calculate the principal directions and draw, based on these directions, a sketch of the distribution with the help of contour lines. Include the principal directions in the sketch.

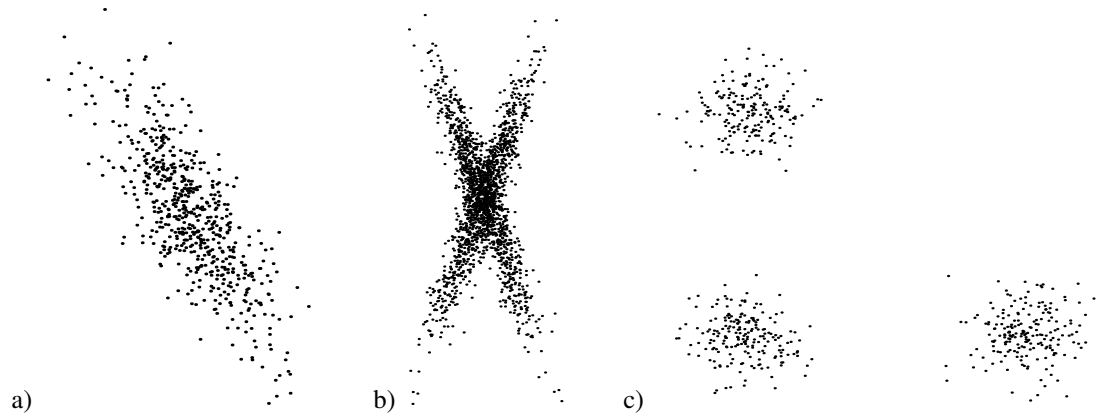
8.16. More principal directions

Draw the first principal direction in the two data sets below. What conclusions can you make about PCA as a preprocessing step before classification? The two data sets consist of samples from two classes, separated by the y-axis.



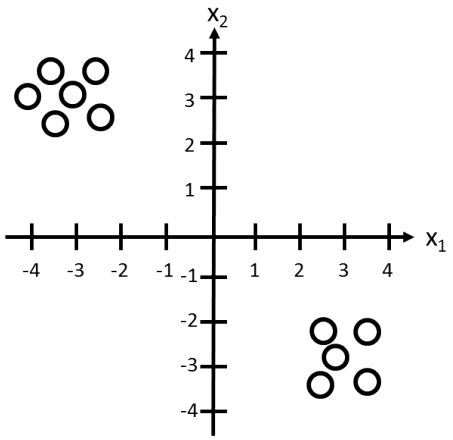
8.17. Even more principal directions

Assume three sources producing two dimensional data according to the figures below. Draw the principal directions.



8.18. PCA + k-means

The figure below shows 2-dimensional training examples. Sketch and draw what happens if you first apply PCA to reduce the dimension of the training data to 1, and then apply k-means clustering to the result. Also give approximate numbers of what would be the result of the k-means algorithm.



8.19. Data compression

Assume we receive packets of 3D-data from a source $\mathbf{x} = (x_1, x_2, x_3)^T$. We should transmit these data on a channel only capable of transmitting packets of 1D-data. PCA has been used to find the first principal direction $\mathbf{e}_1 = (-1, 1, 1)^T$ also known on the receiver side.
Assume that we receive a packet with the following data

$$\mathbf{x} = \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix}$$

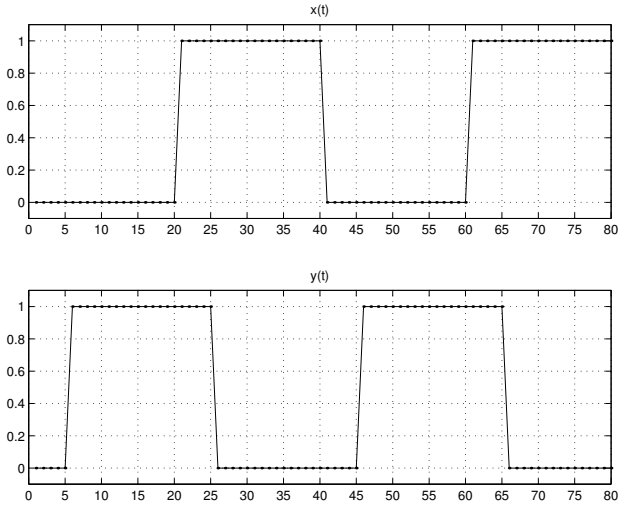
With the help of the first principal direction we can do a dimension reduction so that we can transmit this package. Show how to do this!
In what sense is this an optimal method to reduce the dimensionality of the data?

8.20. PCA of signals

A source gives a two dimensional signal

$$\mathbf{s}(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}$$

where $x(t)$ and $y(t)$ are shown in the figures below:



Find two normalized vectors $\hat{\mathbf{w}}_1$ and $\hat{\mathbf{w}}_2$ so that

$$\hat{s}_1(t) = \hat{\mathbf{w}}_1^T \mathbf{s}(t)$$
and
$$\hat{s}_2(t) = \hat{\mathbf{w}}_2^T \mathbf{s}(t)$$

get maximum and minimal variance respectively. Sketch the new signals.

8.21. More feature dimensions than training samples

Assume that we have 2000 samples of data of 50 dimensions. We represent the data with a matrix $\mathbf{X}$ of size 50×2000 (one training sample per column). The covariance matrix $\mathbf{C}_{xx} = \mathbf{X}\mathbf{X}^T$ (when X is mean-centered) then has the form 50×50, which makes the eigenvalue problem in PCA easy to solve. If we instead have 50 training samples of data of 2000 dimensions, the size of the covariance matrix is 2000×2000, which isn't that fun to calculate. Show that it is ok to instead use the matrix of scalar products between the samples ($\mathbf{X}^T\mathbf{X}$) in order to find the principal directions.

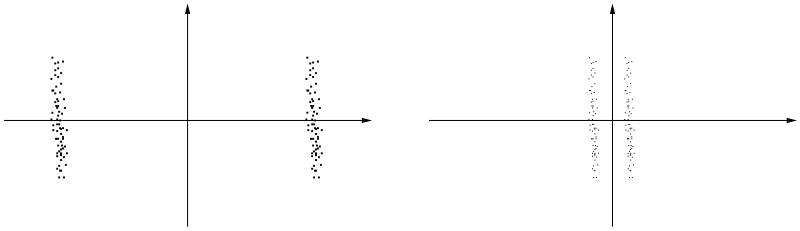
8.22. Fisher linear discriminant

a) Write the cost function being optimized with Fisher linear discriminant/linear discriminant analysis. Explain the notations you use!

b) What assumption is made about the distributions of the two classes in linear discriminant analysis?

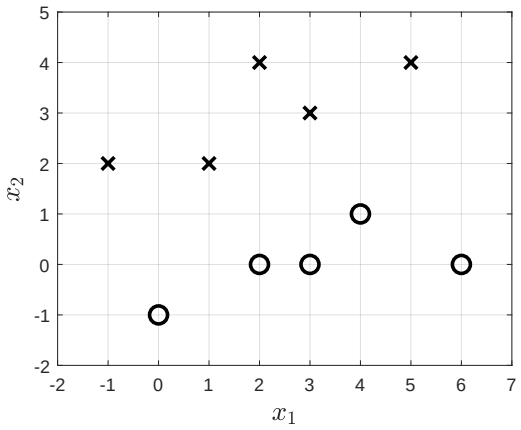
8.23. FLD vs. PCA

Fisher linear discriminant analysis results in a vector that can be used for classification or dimension reduction. Draw the resulting vectors in the two datasets below. Explain the principal differences between LDA and PCA, if any. The data represent two classes separated by the y-axis.



8.24. LDA example

The data points in the figure have two features (x_1 and x_2) and belong to either the class "crosses" or the class "circles":



Perform linear discriminant analysis (LDA) on the data to reduce the dimensionality to one dimension that separates the two classes optimally. Draw the reduced data.

Hint: The inverse of a 2×2-matrix $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ is $\frac{1}{ad-bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$.

8.25. Dimensionality reduction in Matlab

In this task you will learn about PCA and LDA in Matlab by analyzing a dataset about country demographics. This is a simplified version of an old computer assignment in this course, and might therefore take a bit longer than the rest of the questions. Start by downloading the DimReduction zip file from the LISAM course page and unpack the files. The *dimReduction* script is where you will implement your analysis. If you rather just play with the data to answer the questions below you can instead run the *dimReduction_solution* script. The *sorteig* function is used by the provided code to calculate sorted eigenvalues and eigenvectors.

Read the script and follow the instruction in the comments to implement your analysis. Note that some of the operations you will implement are trivial if you use the built-in functions in Matlab (such as cov), however we recommend that you don't use these and instead use matrix operations from scratch.

Now try to answer the following questions about the data:

a) Why do we start by normalizing the data per feature?

b) You will probably find that the covariance matrix and correlation matrix are the same. Why is this the case?

c) Which two features are the most correlated? Which are the most uncorrelated?

d) How much information is kept when transforming the data to the first two principle components?

e) How many principle components are required to keep 90% of the data after the transformation?

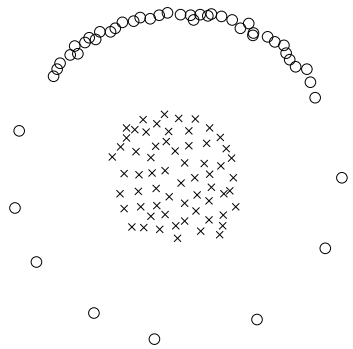
f) There is one country that seems to be mislabeled. Which?

g) Which feature is the most important for separating the Developing countries from the Industrialized? Which is the least important?

9. Kernel methods

9.1. Classification with a nonlinear mapping

The figure below shows the data for a classification problem in the xy-plane. Draw the optimal decision boundaries for a linear classifier that have access to $1, x_1, x_2$. Also draw the optimal boundaries for a classifier based on a linear combination of $1, x_1, x_2, x_1x_2, x_1^2, x_2^2$. The optimal solution is the one with least number of misclassifications and largest margins if possible.



9.2. Scalar product

a) What is the scalar product (a.k.a. as dot product or inner product) between two vectors $\mathbf{x}_1$ and $\mathbf{x}_2$?

b) Show how the length of a vector $\mathbf{x}$ can be expressed in terms of the scalar product.

c) Show how the distance between two points $\mathbf{x}_1$ and $\mathbf{x}_2$ can be expressed in terms of the scalar product.

d) Show how the angle between $\mathbf{x}_1$ and $\mathbf{x}_2$ can be expressed in terms of the scalar product.

The conclusion of this exercise is that if we know the values of the scalar products between vectors, we also know how these vectors are geometrically positioned relative to each other.

9.3. Kernel definition

What is defined by a kernel function?

9.4. Using the kernel function

Consider a Gaussian kernel function $\kappa(\mathbf{x}_1, \mathbf{x}_2) = e^{-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|^2}{4}}$. What is the distance between two feature vectors

$$\mathbf{x}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \text{ and } \mathbf{x}_2 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

in the new feature space defined by this kernel function?

9.5. Explicit mapping vs. implicit mapping with kernel

Consider the following nonlinear mapping of the input data $\mathbf{x}$:

$$\begin{aligned} \varphi_1(\mathbf{x}) &= x_1^2 \\ \varphi_2(\mathbf{x}) &= x_2^2 \\ \varphi_3(\mathbf{x}) &= \sqrt{2} \cdot x_1 x_2 \end{aligned}$$

You want to analyse this data with a kernel method. How is the scalar product $\varphi(\mathbf{x}_1)^T \varphi(\mathbf{x}_2)$ expressed in the input data space?

9.6. Kernel matrix

a) What is the kernel matrix $\mathbf{K}$ (a.k.a. Gram matrix or similarity matrix)?

b) We have 10 training samples in a 20-dimensional space that we want to analyse with a kernel method. How large is the kernel matrix?

9.7. Interpreting the kernel matrix geometrically

You get the following kernel matrix

$$\mathbf{K} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 2 & 2 \\ 0 & 2 & 4 \end{pmatrix}.$$

Plot the training data vectors that could have generated this kernel matrix.

9.8. Removing the mean

Suppose we have a quadratic mapping of the input signal $\mathbf{x}$ to a high-dimensional feature space $\mathbf{x} \rightarrow \varphi(\mathbf{x})$ according to $\varphi(\mathbf{x}) = \mathbf{x} \times \mathbf{x}$ where "$\times$" means that you take the outer product and then make a vector of the resulting matrix, which will contain all products between the components of the input vectors.

a) What will the kernel matrix look like if we have the following three data vectors?

$$\begin{pmatrix} -1 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

b) In some algorithms, such as PCA, it is required to remove the mean from the data. However, the kernel matrix above corresponds to the original and non-centered samples in the feature space. Show that you get a kernel matrix that corresponds to the samples being centered in the feature space by, from the non-centered kernel matrix subtracting the column mean from each column, then subtract the row mean from each row, and finally add the total mean value of the whole matrix, i.e.

$$k'_{ij} = k_{ij} - \frac{1}{n}\sum_i k_{ij} - \frac{1}{n}\sum_j k_{ij} + \frac{1}{n^2}\sum_{ij} k_{ij}$$

where k' are the components in the centered kernel matrix.

9.9. Kernel trick

What is required of an optimization problem in order to be able to solve it with kernel methods?

9.10. Kernel trick applied to SVM

The optimal plane separating two linearly separable classes is in a support vector machine found by optimizing the cost function

$$\min \|\mathbf{w}\|^2 = \mathbf{w}^T\mathbf{w}$$
$$\text{subject to the constraint } y_i\left(\mathbf{w}^T\mathbf{x}_i + w_0\right) \geq 1 \text{ for all } i,$$

where $\{\mathbf{x}_i, y_i\}$ are the training examples and $y_i = \pm 1$. Show how the kernel trick is applied to optimize a nonlinear SVM classifier.

9.11. Support vectors

After training an SVM, the resulting discriminant function $f(\varphi)$ is given by:

$$f(\varphi) = 4\varphi_1 + 9\varphi_2 + 4\varphi_3 - 16\varphi_4$$

The training samples $\mathbf{x}$ have been mapped according to: $\varphi_1(\mathbf{x}) = x_1^2$, $\varphi_2(\mathbf{x}) = x_2^2$, $\varphi_3(\mathbf{x}) = x_1x_2$, and $\varphi_4(\mathbf{x}) = 1$.
Is any of the two samples $\mathbf{x} = \begin{pmatrix} 1 & 1 \end{pmatrix}^T$ and $\mathbf{y} = \begin{pmatrix} 1 & -1 \end{pmatrix}^T$ a support vector? Why/why not?

Solutions