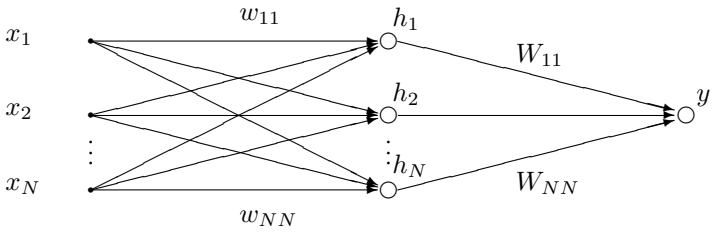
Neural Networks and Learning Systems

TBMI26, Kernel Methods

2017



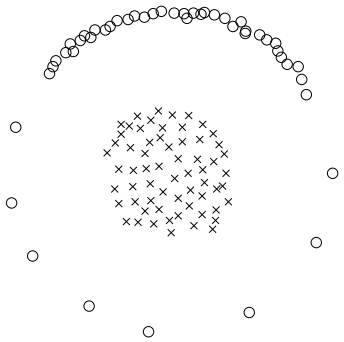
Neural Networks and Learning Systems
Exercise Collection

Exercises

1. Kernel Methods

1.1. 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.



1.2. 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.

1.3. Kernel definition

What is defined by a kernel function?

1.4. 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?

1.5. Explicit mapping vs. implicit mapping with kernel

Consider the following non-linear 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?

1.6. 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?

1.7. 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.

1.8. 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.

1.9. Kernel trick

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

1.10. 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.

1.11. Support vectors

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

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

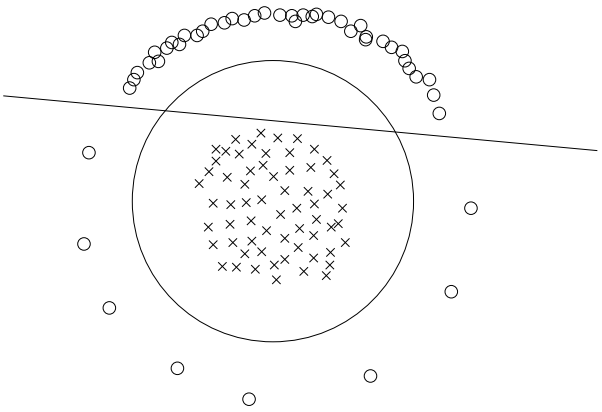
The training samples $\mathbf{x}$ have been mapped according to: $\varphi_1(\mathbf{x}) = x_1^2$, $\varphi_2(\mathbf{x}) = x_2^2$, $\varphi_3(\mathbf{x}) = x_1x_2$, and $\varphi_4(\mathbf{x}) = 1$.
Is any of the two samples $\mathbf{x} = (1 \quad 1)^T$ and $\mathbf{y} = (1 \quad -1)^T$ a support vector?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

Solutions

2. Kernel Methods

Answer 1.1

Since the linear classifier produce a line the placement is pretty straight forward and some misclassifications will occur. With access to the squared feature values $\varphi_1 = x_1^2$ and $\varphi_2 = x_2^2$, a linear classifier with a circular boundary in the original space is easily generated: $f(\varphi_1, \varphi_2) = sign(w_1\varphi_1 + w_2\varphi_2) = sign(w_1x_1^2 + w_2x_2^2)$ that provides perfect classification.



Answer 1.2

a) The scalar product is defined as $\mathbf{x}_1 \cdot \mathbf{x}_2 = \mathbf{x}_1^T\mathbf{x}_2$.

b) The length of the vector $\|\mathbf{x}\| = \sqrt{\|\mathbf{x}\|^2} = \sqrt{\mathbf{x}^T\mathbf{x}} = \sqrt{\mathbf{x} \cdot \mathbf{x}}$.

c) The distance $\|\mathbf{x}_1 - \mathbf{x}_2\| = \sqrt{(\mathbf{x}_1 - \mathbf{x}_2)^T(\mathbf{x}_1 - \mathbf{x}_2)} = \sqrt{\mathbf{x}_1^T\mathbf{x}_1 + \mathbf{x}_2^T\mathbf{x}_2 - 2\mathbf{x}_1^T\mathbf{x}_2} = \sqrt{\mathbf{x}_1 \cdot \mathbf{x}_1 + \mathbf{x}_2 \cdot \mathbf{x}_2 - 2\mathbf{x}_1 \cdot \mathbf{x}_2}$

d) The vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ span a triangle. The Law of Cosines from trigonometry states that if the lengths of the triangle sides are a, b and c, the following relation holds: $c^2 = a^2 + b^2 - 2ab\cos\theta$. If $\mathbf{x}_1$ represents a and $\mathbf{x}_2$ represents b, c is $\mathbf{x}_1 - \mathbf{x}_2$. The Law of Cosines then gives $\|\mathbf{x}_1 - \mathbf{x}_2\|^2 = \|\mathbf{x}_1\|^2 + \|\mathbf{x}_2\|^2 - 2\|\mathbf{x}_1\|\|\mathbf{x}_2\|\cos\theta$. Rearranging and using the results above gives the angle θ between $\mathbf{x}_1$ and $\mathbf{x}_2$ in terms of scalar products $\cos\theta = \frac{\mathbf{x}_1 \cdot \mathbf{x}_2}{\sqrt{\mathbf{x}_1 \cdot \mathbf{x}_1}\sqrt{\mathbf{x}_2 \cdot \mathbf{x}_2}}$.

Answer 1.3

The kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$ defines a scalar product between two vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ that have been mapped to a (usually high-dimensional) feature space via a function $\varphi()$.

$$\varphi(\mathbf{x}_i)^T \varphi(\mathbf{x}_j) = k(\mathbf{x}_i, \mathbf{x}_j) = \text{ some function.}$$

The clue here is that we never deal with $\varphi(\mathbf{x}_i)$ and $\varphi(\mathbf{x}_j)$ explicitly, i.e., we never calculate them or store them in the computer, we only deal the scalar products defined by $k(\mathbf{x}_i, \mathbf{x}_j)$. As the kernel function defines the scalar product, which in turn defines the distance between vectors according to 1.2, the kernel function can also be seen as a similarity function.

Answer 1.4

A kernel function defines the inner product $\kappa(\mathbf{x}_1, \mathbf{x}_2) = \Phi(\mathbf{x}_1)^T \Phi(\mathbf{x}_2)$ in the new feature space. Thus, $\kappa(\mathbf{x}_1, \mathbf{x}_2)$ specifies the feature space by defining how distances and angles are measured, instead of explicitly stating the mapping function $\Phi(\mathbf{x})$.
The distance between $\mathbf{x}_1$ and $\mathbf{x}_2$ in the new feature space is

$$\begin{aligned}
\|\Phi(\mathbf{x}_1) - \Phi(\mathbf{x}_2)\| &= \sqrt{(\Phi(\mathbf{x}_1) - \Phi(\mathbf{x}_2))^T (\Phi(\mathbf{x}_1) - \Phi(\mathbf{x}_2))} \\
&= \sqrt{\Phi(\mathbf{x}_1)^T \Phi(\mathbf{x}_1) - 2\Phi(\mathbf{x}_1)^T \Phi(\mathbf{x}_2) + \Phi(\mathbf{x}_2)^T \Phi(\mathbf{x}_2)} \\
&= \sqrt{\kappa(\mathbf{x}_1, \mathbf{x}_1) - 2\,\kappa(\mathbf{x}_1, \mathbf{x}_2) + \kappa(\mathbf{x}_2, \mathbf{x}_2)} \\
&= \sqrt{1 - 2\,e^{-\frac{1}{4}} + 1} = 0.6651
\end{aligned}$$

Answer 1.5

$$\varphi(\mathbf{x})^T \varphi(\mathbf{y}) = x_1^2 y_1^2 + x_2^2 y_2^2 + 2x_1 x_2 y_1 y_2 = (x_1 y_1 + x_2 y_2)^2 = \left(\mathbf{x}^T \mathbf{y}\right)^2 \tag{1}$$

Answer 1.6

a) The kernel matrix $\mathbf{K}$ contains all scalar products between all training data feature vectors (generally mapped through a nonlinear function $\varphi()$). For example, if we have three training data samples $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$, the kernel matrix is

$$\mathbf{K} = \begin{pmatrix} \varphi(\mathbf{x}_1)^T\varphi(\mathbf{x}_1) & \varphi(\mathbf{x}_1)^T\varphi(\mathbf{x}_2) & \varphi(\mathbf{x}_1)^T\varphi(\mathbf{x}_3) \\ \varphi(\mathbf{x}_2)^T\varphi(\mathbf{x}_1) & \varphi(\mathbf{x}_2)^T\varphi(\mathbf{x}_2) & \varphi(\mathbf{x}_2)^T\varphi(\mathbf{x}_3) \\ \varphi(\mathbf{x}_3)^T\varphi(\mathbf{x}_1) & \varphi(\mathbf{x}_3)^T\varphi(\mathbf{x}_2) & \varphi(\mathbf{x}_3)^T\varphi(\mathbf{x}_3) \end{pmatrix} = \begin{pmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & k(\mathbf{x}_1, \mathbf{x}_3) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & k(\mathbf{x}_2, \mathbf{x}_3) \\ k(\mathbf{x}_3, \mathbf{x}_1) & k(\mathbf{x}_3, \mathbf{x}_2) & k(\mathbf{x}_3, \mathbf{x}_3) \end{pmatrix}.$$

One can note that $\mathbf{K}$ is symmetric as $\mathbf{x}_i^T \mathbf{x}_j = \mathbf{x}_j^T \mathbf{x}_i$.

b) 10×10.

Answer 1.7

We have the following kernel matrix

$$\mathbf{K} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 2 & 2 \\ 0 & 2 & 4 \end{pmatrix} = \begin{pmatrix} \mathbf{x}_1^T\mathbf{x}_1 & \mathbf{x}_1^T\mathbf{x}_2 & \mathbf{x}_1^T\mathbf{x}_3 \\ \mathbf{x}_2^T\mathbf{x}_1 & \mathbf{x}_2^T\mathbf{x}_2 & \mathbf{x}_2^T\mathbf{x}_3 \\ \mathbf{x}_3^T\mathbf{x}_1 & \mathbf{x}_3^T\mathbf{x}_2 & \mathbf{x}_3^T\mathbf{x}_3 \end{pmatrix}.$$

We have three training data vectors that we denote $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$. Using the derivations in 1.2 above, we can extract the following information from $\mathbf{K}$:

- $\|\mathbf{x}_1\| = \sqrt{\mathbf{x}_1 \cdot \mathbf{x}_1} = 1$, i.e., this point is 1 unit length from the origin of the space. Similarly, $\|\mathbf{x}_2\| = \sqrt{2}$ and $\|\mathbf{x}_3\| = 2$.

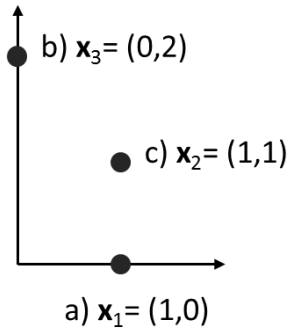
- The distances $\|\mathbf{x}_1 - \mathbf{x}_2\| = 1$, $\|\mathbf{x}_1 - \mathbf{x}_3\| = \sqrt{5}$ and $\|\mathbf{x}_2 - \mathbf{x}_3\| = \sqrt{2}$.
- The angle between $\mathbf{x}_1$ and $\mathbf{x}_2$ is 45°. The angle between $\mathbf{x}_1$ and $\mathbf{x}_3$ is 90° (orthogonal as the scalar product is 0).

Now we can reconstruct the data as follows (see figure c below):

a) Let's begin by placing $\mathbf{x}_1$ at any point with distance 1 from the origin.

b) Next, place $\mathbf{x}_3$ on an axis orthogonal (angle 90°) to $\mathbf{x}_1$, a distance 2 from the origin.

c) Finally, $\mathbf{x}_2$ must be at one of 2 possible positions that are at an angle 45° and a distance 1 relative $\mathbf{x}_1$. As the distance between $\mathbf{x}_2$ and $\mathbf{x}_3$ must also be $\sqrt{2}$, there is only one possible point left. Note that



there is an infinite number of solutions depending on where we start with $\mathbf{x}_1$.

Answer 1.8

a)

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

b) The vectors in the feature space become

$$\varphi(\mathbf{x}_1) = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \varphi(\mathbf{x}_2) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \varphi(\mathbf{x}_3) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

The average vector in the feature space then becomes

$$\overline{\varphi}(\mathbf{x}) = \frac{1}{3} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

The centered feature vectors the become $\varphi'(\mathbf{x}_i) = \varphi(\mathbf{x}_i) - \overline{\varphi}(\mathbf{x})$:

$$\varphi'(\mathbf{x}_1) = \frac{1}{3} \begin{bmatrix} 2 \\ 0 \\ 0 \\ -1 \end{bmatrix}, \quad \varphi'(\mathbf{x}_2) = \frac{1}{3} \begin{bmatrix} -1 \\ 0 \\ 0 \\ -1 \end{bmatrix}, \quad \varphi'(\mathbf{x}_3) = \frac{1}{3} \begin{bmatrix} -1 \\ 0 \\ 0 \\ 2 \end{bmatrix}$$

The centered kernel matrix then becomes

$$\text{K'} = \frac{1}{9} \begin{bmatrix} 5 & -1 & -4 \\ -1 & 2 & -1 \\ -4 & -1 & 5 \end{bmatrix}$$

This is the same matrix as we get if we subtract the row- and column averages from K and add the total average.

Answer 1.9

The problem must be possible to formulate in terms of scalar products between the samples so that we can swap in a nonlinear kernel function here, the so-called kernel trick. A first step to do this is typically to show that the optimal solution can be written as a linear combination of the training data vectors $\mathbf{w}^* = \sum_{i=1}^{N} \alpha_i \mathbf{x}_i$ for some values of the α's.

Answer 1.10

The original cost function is:

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

The kernelized function can be derived for problems for which it can be shown that the optimal solution $\mathbf{w}^*$ lies in the span of the input training data, i.e., $\mathbf{w}^* = \sum_{i=1}^{N} \alpha_i \mathbf{x}_i$, where α_i are some number, $\mathbf{x}_i$ is a traing data example and N is the number of training examples. Inserting this relationship in the original cost function yields $\mathbf{w}^T\mathbf{w} = \sum_{i=1}^{N}\sum_{i=j}^{N} \alpha_i\alpha_j\mathbf{x}_i^T\mathbf{x}_j$ (which can also be written in a vector form $\boldsymbol{\alpha}^T\mathbf{K}\boldsymbol{\alpha}$ using a kernel matrix $\mathbf{K}$). The entire kernelized cost function is

$$\begin{aligned} &\min \sum_{i=1}^{N}\sum_{i=j}^{N} \alpha_i\alpha_j\mathbf{x}_i^T\mathbf{x}_j \\ &\text{subject to the constraint } y_i \left(\alpha_i\mathbf{x}_i^T\mathbf{x}_i + \alpha_0\right) \geq 1 \text{ for all } i. \end{aligned}$$

In non-linear kernel methods, the inner products $\mathbf{x}_i^T\mathbf{x}_j$ is replaced with a non-linear kernel function $\kappa(\mathbf{x}_i, \mathbf{x}_j)$.

Answer 1.11

The classification function in SVM has the value ± 1 for the support vectors. Hence, $\mathbf{x}$ is a support vector since $f(\mathbf{x}) = 1$