FMAN45 - Machine Learning - Assignment 2

Wiliam Lindskog

April 2021

Task T1

First, one must compute the Kernel matrix $\mathbf{K}$. Considering the (non-linear) feature map

$$\phi(x) = \begin{pmatrix} x \\ x^2 \end{pmatrix} \tag{1}$$

This results in the corresponding kernel

$$\begin{aligned}\mathbf{K} = k(x_i, x_j) = \phi(x_i)^T \phi(x_j) = x_i x_j + (x_i x_j)^2 \qquad (2) \\ = \begin{pmatrix} 20 & 6 & 2 & 12 \\ 6 & 2 & 0 & 2 \\ 2 & 0 & 2 & 6 \\ 12 & 2 & 6 & 20 \end{pmatrix} \qquad (3)\end{aligned}$$

Task T2

Since $\alpha_{\{1,\ 2,\ 3,\ 4\}} = \alpha$, a new optimization problem is received.

$$\max_\alpha \Big(4\alpha - \frac{\alpha^2}{2} \sum_{i,j=1}^{4} y_i y_j k(x_i x_j)\Big) \tag{4}$$

subject to $\alpha \geq 0$ and $\alpha \sum_{i=1}^4 y_i = 0$. What is inside the summation term equals:

$$\begin{aligned}\sum_{i,j=1}^{4} y_i y_j k(x_i x_j) &= 1 \cdot (2 \cdot 20 + 2 \cdot 12 + 2 \cdot 2 + 2 \cdot 0) - 1 \cdot (4 \cdot 6 + 4 \cdot 2) && (5) \\ &= (40 + 24 + 4 + 0) - (24 + 8) && (6) \\ &= 44 - 8 = 36 && (7)\end{aligned}$$

Inserting this in equation (4) gives

$$\max_\alpha \Big(4\alpha - \frac{36\alpha^2}{2}\Big) = \max_\alpha \big(4\alpha - 18\alpha^2\big) \tag{8}$$

Acknowledging the fact the the second derivative of the function inside the max expression is concave (negative definite), see equation (9) below,

$$\frac{d^2}{d\alpha^2}\big(4\alpha - 18\alpha^2\big) = \frac{d}{d\alpha}\big(4 - 36\alpha\big) = -36 < 0 \tag{9}$$

and that $\alpha \geq 0$, means that we can find a maximum point where the first derivative equals 0

$$\begin{aligned} 0 &= \frac{d}{d\alpha}\big(4\alpha - 18\alpha^2\big) && (10) \\ &= 4 - 36\alpha && (11) \\ \alpha &\iff \frac{4}{36} = \frac{1}{9} && (12) \end{aligned}$$

Task T3

In this task, one should reduce the classifier function

$$g(x) = \sum_{j=1}^{4} \alpha_j y_j k(x_j, x) + b \tag{13}$$

also presented in the instructions. From the previous task we have that $\alpha = \frac{1}{9}$. Inserting this in equation (13) gives

$$g(x) = \frac{1}{9}\sum_{j=1}^{4} y_j k(x_j, x) + b \tag{14}$$

Expanding previous equation we have

$$\begin{aligned} &\frac{1}{9}(1 \cdot (-2x + 4x^2) - 1 \cdot (-x + x^2) - 1 \cdot (x + x^2) + 1 \cdot (2x + 4x^2)) + b && (15) \\ &= \frac{1}{9}(-2x + 4x^2 + x - x^2 - x - x^2 + 2x + 4x^2) + b && (16) \\ &= \frac{1}{9}(x^2(4 \cdot 2 - 2 \cdot 1)) + b = \frac{1}{9}6x^2 + b = \frac{2x^2}{3} + b && (17) \end{aligned}$$

Using equation (6) presented in instructions, also below

$$y_s\Big(\sum_{j=1}^{4} \alpha_j y_j k(x_j, x) + b\Big) = 1 \tag{18}$$

and the result from equation (17) we can find that

$$1 = y_s\Big(\sum_{j=1}^{4}\alpha_j y_j k(x_j, x_s) + b\Big) \tag{19}$$

$$= y_s\Big(\frac{2x_s^2}{3} + b\Big) = 1 \cdot \Big(\frac{2(-2)^2}{3} + b\Big) \tag{20}$$

$$= \frac{8}{3} + b \iff b = 1 - \frac{8}{3} = -\frac{5}{3} \tag{21}$$

Therefore, g(x) can be described as

$$g(x) = \frac{2x^2}{3} - \frac{5}{3} \tag{22}$$

Task T4

One may draw the binary classification problem in a 2D-space. Noticing that $x_2,\ x_3,\ x_5,\ and\, x_6$ are instances on a decision boundary equal to g(x) we can use the same function g(x) as a classifier. Also, presented in equation (22). The rest of the data points are thereafter separated into different classes. See figure below:

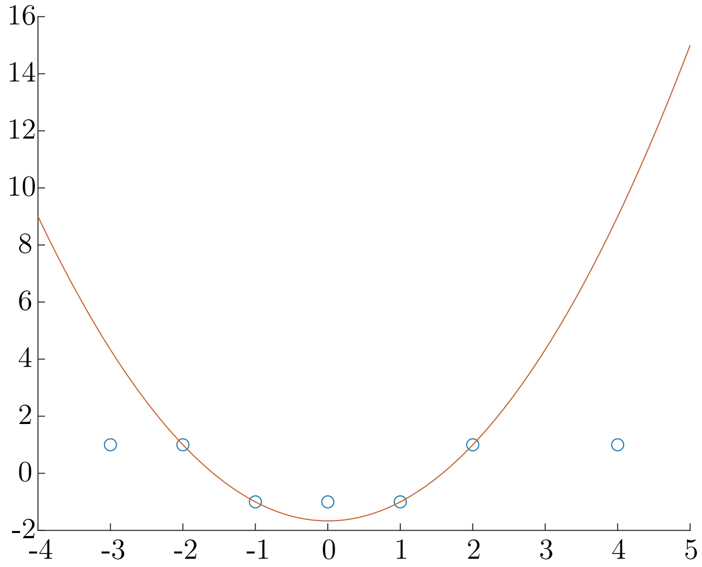


Figure 1: g(x) plotted with data points

Task T5

The primal formulation of the linear soft margin classifier presented in the instructions can be rewritten as:

$$\min_{\omega,b,\xi} \frac{1}{2}||\omega||^2 + C\sum_{i=1}^{n}\xi_i \tag{23}$$

subject to $0 \geq 1 - \xi - y_i(\omega^T x_i + b)$ and $-\xi \leq 0$. The presented minimization problem can be derived to respective Lagrangian function:

$$L = \frac{1}{2}||\omega||^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i(\xi - 1 + y_i(\omega^T x_i + b)) - \sum_{i=1}^{n}\lambda_i\xi_i \tag{24}$$

subject to $\alpha_i,\ \lambda_i \geq 0$. To derive the sought dual problem, it is necessary to minimize L with respect to $\omega,\ b,\ \xi$. Setting the differentiation to zero results in:

$$\frac{dL}{d\omega} = 0 = \omega - \sum_{i=1}^{n}\alpha_i y_i x_i \iff \omega = \sum_{i=1}^{n}\alpha_i y_i x_i \tag{25}$$

$$\frac{dL}{db} = 0 = -\sum_{i=1}^{n}\alpha_i y_i \iff \sum_{i=1}^{n}\alpha_i y_i = 0 \tag{26}$$

$$\frac{dL}{d\xi_i} = 0 = C - \alpha_i - \lambda_i \iff \lambda_i = C - \alpha_i \tag{27}$$

and $0 \leq \alpha_i \leq C$ due to constraints. Using equation (25) and imputing it in equation (24) we receive:

$$\inf_{\omega,b,\xi} L = \frac{1}{2}||\sum_{i=1}^{n}\alpha_i y_i x_i||^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i(\xi - 1 + y_i((\sum_{i=1}^{n}\alpha_i y_i x_i)^T x_i + b)) - \sum_{i=1}^{n}\lambda_i\xi_i \tag{28}$$

$$= -\frac{1}{2}\sum_{i=1}^{n}\sum_{i=j}^{n}\alpha_i\alpha_j y_i y_j x_i x_j + C\sum_{i=1}^{n}\xi_i(C - \alpha_i - \lambda_i) + \sum_{i=1}^{n}\alpha_i - \sum_{i=1}^{n}\alpha_i y_i b \tag{29}$$

$$= \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{i=j}^{n}\alpha_i\alpha_j y_i y_j x_i x_j \tag{30}$$

This, together with constraints derived in equation (26) and (27) gives the dual problem.

Task T6

We seek to show that support vectors with $y_i(\omega^T x_i + b) < 1$ have coefficient $\alpha_i = C$. For specified support vectors, we have that $\xi_i > 0$. Using complementary slackness (of the KKT conditions) we have that $\lambda_i = 0$[1] and therefore $\alpha_i = C$, from equation (27).

[1] Since the relevant constraint on λ for $\xi \geq 0$ is not active

Task E1

For this task one must compute a linear PCA (2-dimensional) and visualize the 784-dimensional MNIST dataset. The data is firstly transformed to zero-mean data. If data is represented by $X \in \mathbb{R}^2$, then it is computed

$$X = X - \overline{X} \tag{31}$$

where

$$\overline{X} = \frac{1}{N}\sum_{i=1}^{N} X_i, \; X = \{X_1, \; X_2, \; ..., \; X_N\} \tag{32}$$

Identifying the two largest eigenvalues, it is possible to project the data used on the related eigenvectors.

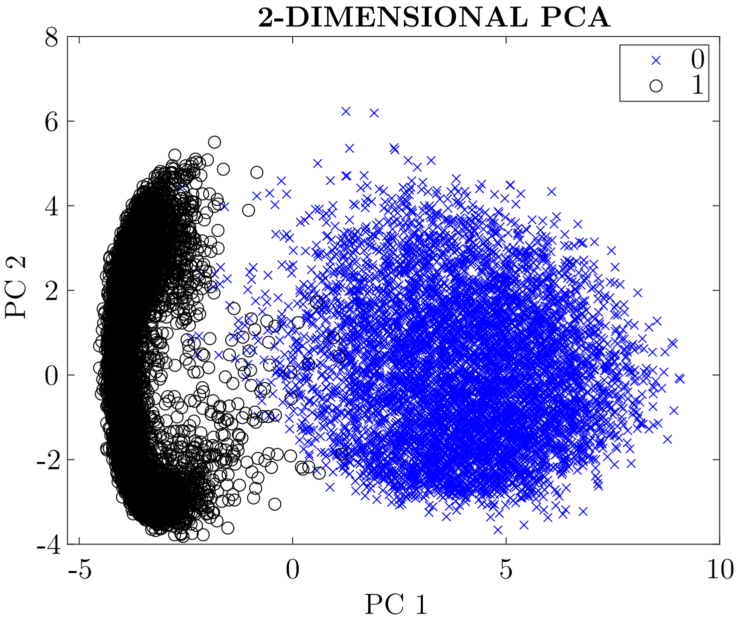


Figure 2: 2-dimensional PCA of selected labels 0 and 1 from MNIST data

In figure 2 it is possible to identify principal component (PC) values for the data with different labels.

Task E2

For this task one must cluster the dataset into 2 and 5 clusters using K-means. Tolerance is set $\epsilon = 10^{-6}$ and the algorithm process quit when a pairwise distance function $dist < \epsilon$. Figure 3 visualizes a PCA representation of the data using K-means with 2 clusters. It is quite identical to figure 2.

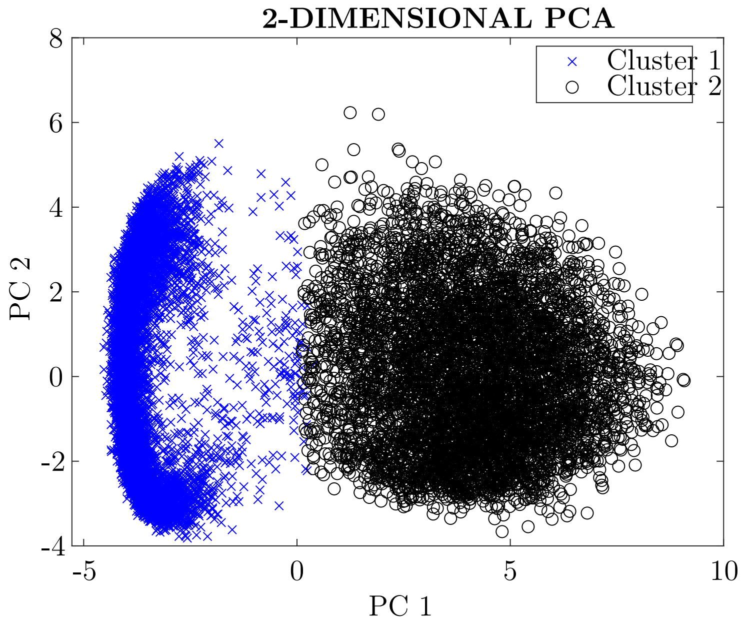
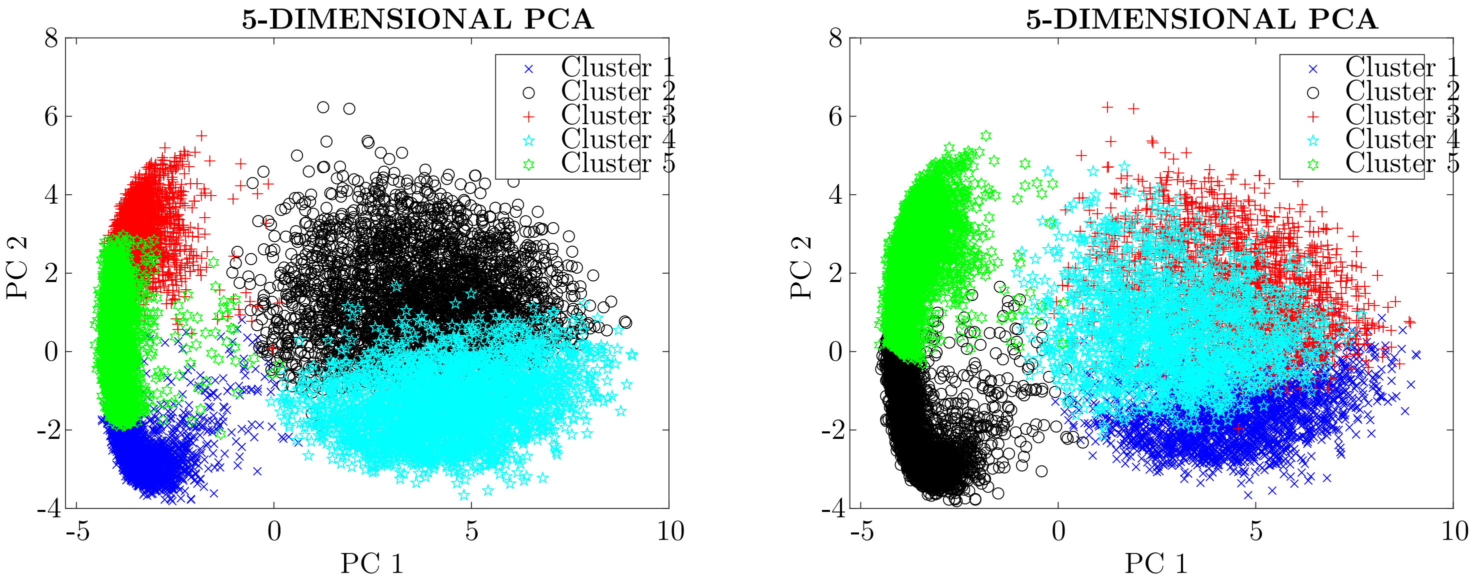


Figure 3: K-means representation of PCA for 2 clusters

Using K-means for 5 clusters, one receives following varying results.



(a) One K-means representation of PCA for 5 clusters

(b) Another K-means representation of PCA for 5 clusters

Figure 4: Two different cluster representations

An overlap for the cluster is visible in both subplot in figure 4. This is due to PCA dimensionality reduction which is put onto data used. The data points may be within a certain distance to many clusters, and therefore we get an overlap. The clusters for the original dimension should not be overlapping. Nevertheless, in a lower dimensional space, they might.

Task E3

For this task we wish to visualize the centroids for 2 and 5 clusters.

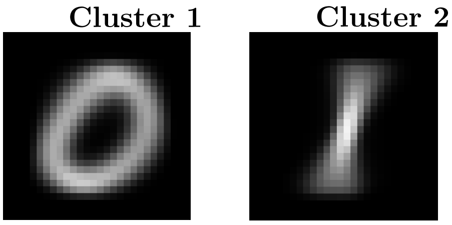


Figure 5: Representation of images using 2 cluster K-means

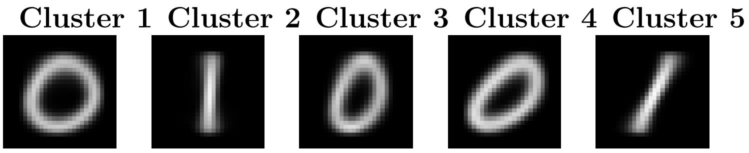


Figure 6: Representation of images using 5 cluster K-means

It is clear that the centroids shown as images in figure 5 and 6 correspond to figure 3 and 4b.

Task E4

Utilizing the centroids retrieved after having used 2 cluster K-means, it is possible to classify data points as 0s or 1s. Result is presented in table 1. It seems to be learning

Table 1: K-means classification results

Training data	Cluster	# ’0’	# ’1’	Assigned to class	# misclassified
	1	114	6736	1	114
	2	5809	6	0	6
$N_{\text{train}} = 12665$				Sum misclassified:	120
				Misclassification rate (%):	0.95
Testing data	Cluster	# ’0’	# ’1’	Assigned to class	# misclassified
	1	12	1135	1	12
	2	968	0	0	0
$N_{\text{test}} = 2115$				Sum misclassified:	12
				Misclassification rate (%):	0.57

well. The test misclassification rate is 0.57. The train misclassification is somewhat higher, 0.95, but still less than 1%.

Task E5

Using a max number of cluster equal to 20, it is possible to iterate from 2 to 20 and plot misclassification rate. The misclassification rate is derived from the test data and is illustrated in figure

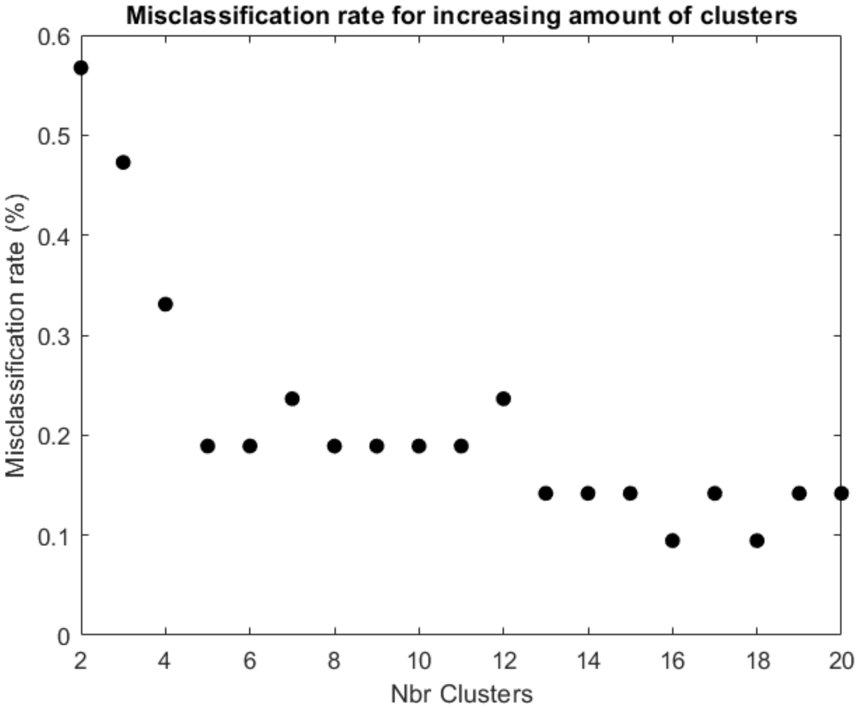


Figure 7: Misclassification rate for K = 2, 3, ..., 20clusters

From figure 7 it is clear that *generally* for a higher number of clusters, the missclassification decreases. Nevertheless, it is not a steady decrease and "jumps" towards the end of the iterations. Lastly, it should be said the generally it performs well for all chosen amount of clusters. It must be taken into consideration that highest missclassification rate is approximately 0.55%.

Task E6

Here, one must classify the MNIST data using a linear support vector machine (SVM). The SVM classifier should be used to classify the 0s and 1s for both training and test data.

Table 2: Linear SVM classification results

Training data	Predicted class	True class: # ’0’	# ’1’
	’0’	5923	0
	’1’	0	6742
$N_{\text{train}} = 12665$		Sum misclassified:	0
		Misclassification rate (%):	0
Testing data	Predicted class	True class: # ’0’	# ’1’
	’0’	979	1
	’1’	1	1134
$N_{\text{test}} = 2115$		Sum misclassified:	2
		Misclassification rate (%):	0.095

From table 2 it is evident that the linear SVM preforms very well on both training and test data.

Task E7

For this task an SVM using a Gaussian kernel is utilized, when classifying the data. Setting $\beta = 1$ initially gives quite poor results for test data, see table 4.

Table 3: Linear SVM classification results

Training data	Predicted class	True class: # ’0’	# ’1’
	’0’	5923	0
	’1’	0	6742
$N_{\text{train}} = 12665$		Sum misclassified:	0
		Misclassification rate (%):	0
Testing data	Predicted class	True class: # ’0’	# ’1’
	’0’	980	388
	’1’	0	747
$N_{\text{test}} = 2115$		Sum misclassified:	388
		Misclassification rate (%):	18.35

Table 4: Misclassification results $\beta = 1$

By increasing β sporadically, using intervals of 0.2 $\beta = \{1,\ 1.2,\ ...,\ 5.8,\ 6.0\}$ and also setting a constraint that if misclassification rate is 0 for test data, then the process stops and desirable results are found. Having run the process until 4.8, it stops. Thereafter some testing shows that $\beta_{opt} = 4.72$. See figure

Table 5: Linear SVM classification results

Training data	Predicted class	True class:	# ’0’	# ’1’
	’0’		5923	0
	’1’		0	6742
$N_{\text{train}} = 12665$	Sum misclassified:			0
	Misclassification rate (%):			0
Testing data	Predicted class	True class:	# ’0’	# ’1’
	’0’		980	0
	’1’		0	1135
$N_{\text{test}} = 2115$	Sum misclassified:			0
	Misclassification rate (%):			0

Table 6: Misclassification results $\beta = 4.72$

Task E8

In this case, one may choose a β with which it is possible to to achieve 0 or almost 0 misclassification rate. This is the case using data provided for tasks. Nevertheless, there is a risk of overfitting the classifier to the data in question. β can be referred to as a training data point’s influence on new data points. If one were to use a relative large value for the scaling parameter σ, β would consequently be lower, and the variance for the SVM with Gaussian kernel would be high. Nevertheless, the bias (from variance/bias trade-off) is higher. Introducing a new data set, one would not achieve the same misclassification results.