Assignment 2 - Machine Learning

Andreas Timürtas

April 2021

Task T1

i	1	2	3	4
x_i	-2	-1	1	2
y_i	+1	-1	-1	+1

Table 1: Data for our one-dimensional binary classification problem

The data provided is seen in table 1 and the feature map, $\boldsymbol{\phi}(x)$, we will use can be seen in equation 1.

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

To calculate the kernel matrix $\mathbf{K}$ using our data we will use the formula

$$\mathbf{K} = [k(x_i, x_j)]_{1 \leq i,j \leq 4}$$

where $k(x, y) = \boldsymbol{\phi}(x)^T \boldsymbol{\phi}(y)$. We begin by simplifying the expression of $k(x_i, x_j)$.

$$k(x_i, x_j) = \boldsymbol{\phi}(x_i)^T \boldsymbol{\phi}(x_j) = \begin{pmatrix} x_i & x_i^2 \end{pmatrix} \begin{pmatrix} x_j \\ x_j^2 \end{pmatrix} = x_i x_j + x_i^2 x_j^2 \tag{2}$$

The kernel matrix can then be written as:

$$\mathbf{K} = \begin{pmatrix} k(x_1, x_1) & k(x_1, x_2) & k(x_1, x_3) & k(x_1, x_4) \\ k(x_2, x_1) & k(x_2, x_2) & k(x_2, x_3) & k(x_2, x_4) \\ k(x_3, x_1) & k(x_3, x_2) & k(x_3, x_3) & k(x_3, x_4) \\ k(x_4, x_1) & k(x_4, x_2) & k(x_4, x_3) & k(x_4, x_4) \end{pmatrix} \tag{3}$$

If we now insert our data from table 1 using the simplification in equation 2 we get the resulting kernel matrix in equation 4.

$$\mathbf{K} = \begin{pmatrix} 20 & 6 & 2 & 12 \\ 6 & 2 & 0 & 2 \\ 2 & 0 & 2 & 6 \\ 12 & 2 & 6 & 20 \end{pmatrix} \tag{4}$$

Task T2

We want to solve the maximization problem in equation 5 using the data in table 1, given the information that all α values are equal.

$$\begin{aligned} \underset{\alpha}{\text{maximize}} \quad & \sum_{i=1}^{4} \alpha - \frac{1}{2} \sum_{i,j=1}^{4} \alpha^2 y_i y_j k(x_i, x_j) \\ \text{subject to} \quad & \alpha \geq 0 \text{ and} \sum_{i=1}^{4} y_i \alpha = 0, \quad \forall i \end{aligned} \tag{5}$$

We begin by simplifying the expression in the equation and then inserting the values from our data.

$$\begin{aligned} &\sum_{i=1}^{4} \alpha - \frac{1}{2} \sum_{i,j=1}^{4} \alpha^2 y_i y_j k(x_i, x_j) = 4\alpha - \frac{\alpha^2}{2} \sum_{i,j=1}^{4} y_i y_j k(x_i, x_j) = \\ &4\alpha - \frac{\alpha^2}{2}(20 - 6 - 2 + 12 - 6 + 2 + 0 - 2 - 2 + 0 + 2 - 6 + 12 - 2 - 6 + 20) = 4\alpha - \frac{\alpha^2}{2} \cdot 36 \end{aligned} \tag{6}$$

To maximize the expression above we differentiate with respect on α and set it equal to zero. We then solve for α.

$$\frac{\partial}{\partial \alpha} 4\alpha - \frac{\alpha^2}{2} \cdot 36 = 36\alpha - 4 = 0 \iff \alpha = \frac{4}{36} = \frac{1}{9} \tag{7}$$

This α satisfies both the constraints in 5 and because the expression in equation 6 is negative definite we know that it is concave and $\alpha = \frac{1}{9}$ maximizes therefore the equation.

Task T3

The equation for the classifier is given by

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

We want to simplify this expression for our data in table 1. From the previous task we could use that all α values were equal with this data set and we will use the same information here. We recall that $k(x_j, x)$ is equal to $x_j x + x_j^2 x^2$.

$$\begin{aligned} g(x) &= \sum_{j=1}^{4} \alpha y_j k(x_j, x) + b = \\ &= \alpha \left(y_1 k(x_1, x) + y_2 k(x_2, x) + y_3 k(x_3, x) + y_4 k(x_4, x) \right) + b = \\ &= \alpha \left((-2x + 4x^2) - (-x + x^2) - (x + x^2) + (2x + 4x^2) \right) + b = \\ &= \alpha \left(-2x + 4x^2 + x - x^2 - x - x^2 + 2x + 4x^2 \right) + b = 6\alpha x^2 + b \end{aligned} \tag{8}$$

We can now use the result from task T2 and put $\alpha = \frac{1}{9}$ as we proceed to calculate what b should be. We know that the expression

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

must hold for any support vector (x_s, y_s). We can therefore insert the support vector $(x_s, y_s) = (1, -1)$ from our data and solve for b.

$$y_s(\sum_{j=1}^{4} \alpha_j y_j k(x_j, x_s) + b) = y_s(\frac{2}{3}x^2 + b) = -(\frac{2}{3} + b) = 1 \Longrightarrow b = -\frac{5}{3} \tag{9}$$

Putting it all together we get the expression

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

This is the simplest form of the classifier $g(x)$. In figure 1 the four data points are plotted in the (x, x^2) space with the decision boundary. Here we can see that all points are support vectors because the decision boundary has the largest margin to each class.

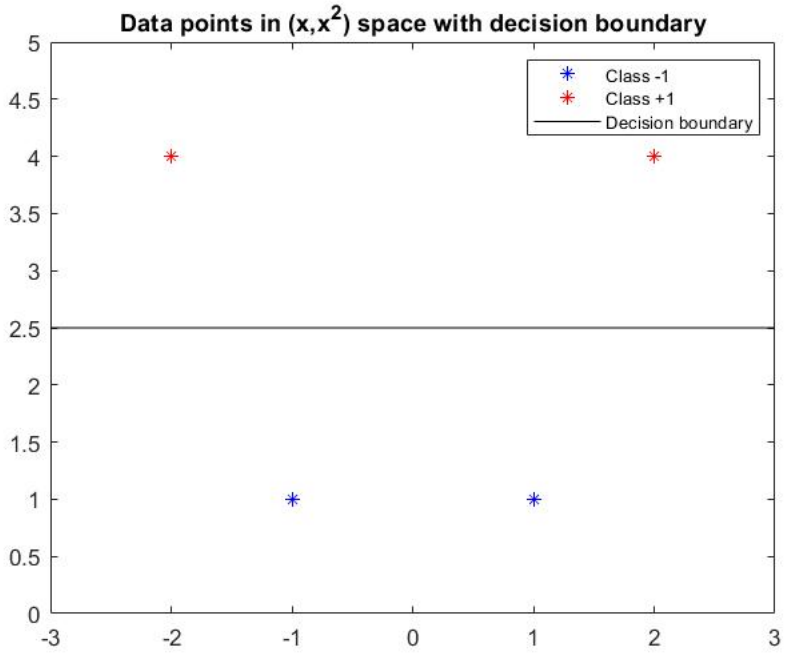

Figure 1: The data points plotted in (x, x^2) space with the decision boundary. The blue dots belongs to class "-1" and the red dots to class "+1".

Task T4

In figure 2 a plot of the points in our new data set can be seen. We notice that the four points that are closest to the decision boundary is the same points as in the old data set. This means that these four points will define our support vectors and therefore will result in the same classifier $g(x)$ as in task T3.

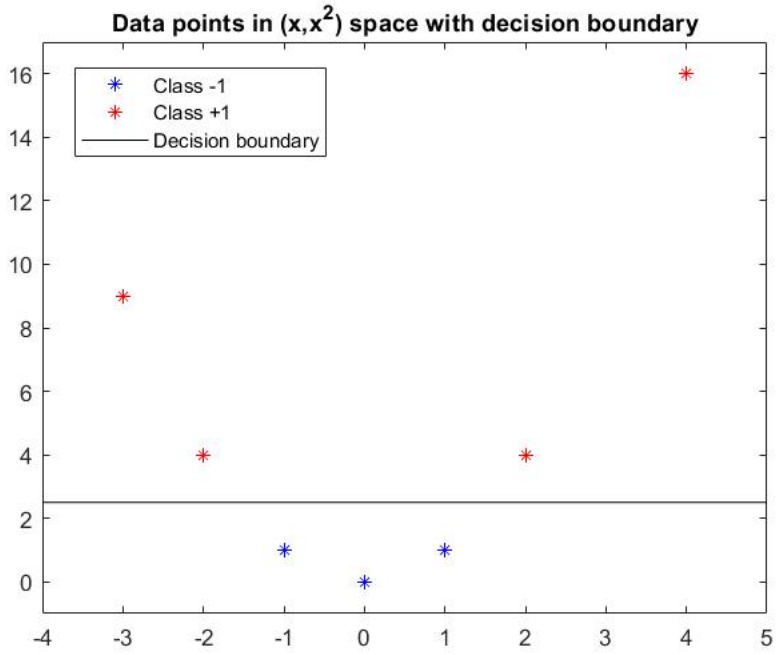

Figure 2: The data points plotted in the (x, x^2) space. The blue dots belongs to class "-1" and the red dots to class "+1".

Task T5

The linear soft margin classifiers primal formulation is seen in equation 10.

$$\begin{aligned} \underset{\mathbf{w},b,\xi}{\text{minimize}} \quad & \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i \\ \text{subject to} \quad & y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{aligned} \tag{10}$$

We want to show that the Lagrangian dual problem is given by equation 11.

$$\begin{aligned} \underset{\alpha_1,\ldots,\alpha_n}{\text{maximize}} \quad & \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j \\ \text{subject to} \quad & 0 \leq \alpha_i \leq C \\ & \sum_{i=1}^{n}\alpha_i y_i = 0 \end{aligned} \tag{11}$$

We begin by formulating the corresponding Lagrangian equation $L(\mathbf{w}, \xi, \mathbf{b}, \mathbf{a})$ with the constraints from equation 10.

$$L(\mathbf{w}, \xi, \mathbf{b}, \alpha) = \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i(y_i(\mathbf{w}^T\mathbf{x}_i + \mathbf{b}) - 1 + \xi_i) - \sum_{i=i}^{n}\mu_i\xi_i \tag{12}$$

The corresponding KKT conditions is given by

$$\alpha_i \geq 0 \tag{13}$$
$$y_i(\mathbf{w}^T\mathbf{x}_i + \mathbf{b}) - 1 + \xi_i \geq 0 \tag{14}$$
$$\alpha_i(y_i(\mathbf{w}^T\mathbf{x}_i + \mathbf{b}) - 1 + \xi_i) = 0 \tag{15}$$
$$\mu_i \geq 0 \tag{16}$$
$$\xi_i \geq 0 \tag{17}$$
$$\mu_i\xi_i = 0 \tag{18}$$

for $i = 1, ..., n$. We proceed by optimizing $\mathbf{w}$, $\mathbf{b}$ and ξ_i from $L(\mathbf{w}, \xi, \mathbf{b}, \alpha)$ by differentiating with respect to those variables and set the result equal to zero.

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \tag{19}$$
$$\frac{\partial L}{\partial \mathbf{b}} = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{20}$$
$$\frac{\partial L}{\partial \xi_i} = 0 \Rightarrow \alpha_i = C - \mu_i \tag{21}$$

We use these expression to simplify the Lagrangian equation.

$$\begin{aligned} L(\mathbf{w}, \xi, \mathbf{b}, \alpha) &= \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i(y_i(\mathbf{w}^T\mathbf{x}_i + \mathbf{b}) - 1 + \xi_i) - \sum_{i=i}^{n}\mu_i\xi_i \\ &= \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{n}\alpha_i y_i(\mathbf{w}^T\mathbf{x}_i + \mathbf{b}) + \sum_{i=1}^{n}\alpha_i + \sum_{i=1}^{n}\xi_i(C - \alpha_i - \mu_i) \\ &\Rightarrow \frac{1}{2}\mathbf{w}^T\mathbf{w} - \mathbf{w}^T\mathbf{w} + \sum_{i=1}^{n}\alpha_i y_i \mathbf{b} + \sum_{i=1}^{n}\alpha_i \\ &= -\frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{i=1}^{n}\alpha_i \\ &= \sum_{i=1}^{n}\alpha_i - \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i\mathbf{x}_j \end{aligned} \tag{22}$$

The remaining part of the Lagrangian dual problem is the constraints. We already have shown that $\sum_{i=1}^{n}\alpha_i y_i = 0$ and need to show that $0 \leq \alpha_i \leq C$ for $i = 1, ..., n$. Because α_i are Lagrangian multipliers we know that they need to be larger or equal to 0. We can also see in equation 21 that α_i need to be less or equal to C because μ_i are also Lagrangian multipliers. This gives us the constraints and we have therefore derived the Lagrangian dual problem in equation 11.

Task T6

When support vectors with $y_i(\mathbf{w}^T\mathbf{x}_i + b)$ is less the 1 we can see from the second KKT condition that ξ_i must be positive (this is because $y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1$ is negative). We can then use the fourth and sixth conditions to conclude that μ_i must be equal to 0. If equation 21 must hold then $\alpha_i = C$ if μ_i is equal to 0.

Task E1

In this task and the task that follows we will use the MNIST data set, a set that are images of handwritten digits. We will only use images that either contains the digits one or zero and will

have 12665 images for our training set and 2115 for our test set. The images have a resolution of 28x28 pixels giving us a total of 784 dimensions for each image, which makes it hard to visualize the images next to each other. The method we will use to help us visualize the images is called Principal component analysis (PCA) and uses the singular value decomposition to project the images to smaller dimensions. To compute our linear PCA we first had to make our data zero mean which we did by subtracting the mean of each pixel. We could then use MATLABs build in function *svd()* that calculates and returns the left eigenvectors U, the diagonal matrix with the eigenvalues S and the right eigenvector V. We could then use the two columns in U with the largest corresponding eigenvalues to project the images in to two dimensions, making it possible to plot the *images* next to each other. The resulting plot is seen in figure 3 where we clearly see that the images create two clusters, one for each digit.

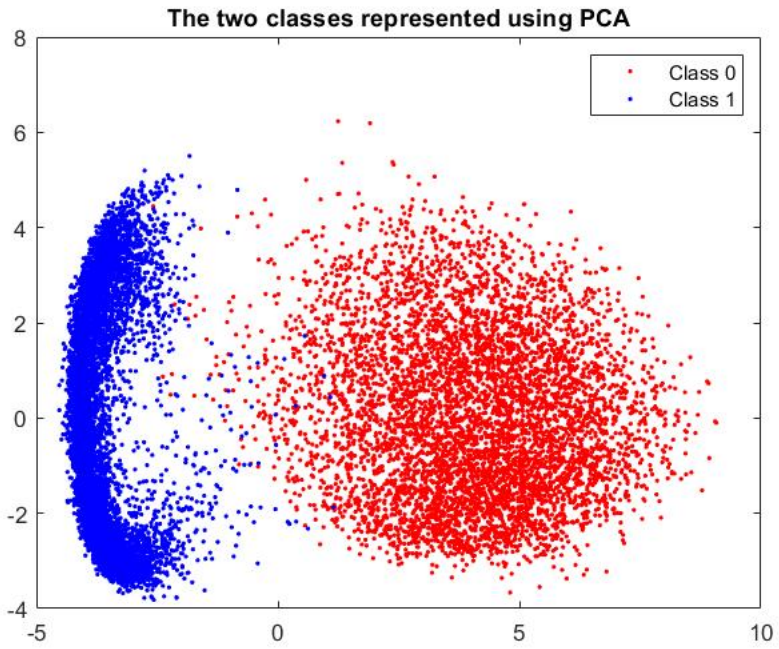

Figure 3: The samples visualized in two dimensions using PCA. The red dots are images of the digit zero while the blue dots are images of the digit one.

Task E2

In this task we wanted to define a number of clusters of our data using the method K-means clustering. This is an unsupervised method, meaning no labels of the samples are used, that uses the distance to K numbers of centroids to assign each sample to a cluster. A centroid is simply just an image that, from the beginning, is randomized using the mean and standard deviation of our data. The method then proceeds to assign each sample a cluster (the closest centroid) and then moves the centroid to the mean of those images. This will make the centroids resemble the images that belongs to that cluster. This is iterated until the centroids converges. In our method we used the L_2-norm on the difference of a centroid and the sample as our distance function. In figure 4 the method using K = 2 have assigned the samples to two clusters and the been visualized using the same PCA as task E1. The green dots are the centroids for the two clusters. In figure 5 we have done the same but with K = 5. The overlap that is clearly visible in the plot can be explained by realising that the distance is calculated using all all dimensions while the plot is made using a projecting, this means that the samples that overlap may only be visible when we project the images and therefore be a result of our lack of ability to visualize the samples better. An easy example to explain this is by thing of a sphere that is projected

into two dimensions making a circle. If the sphere was split into two halves a projection could be made that would result in that the two halves fully overlaps in two dimensions.

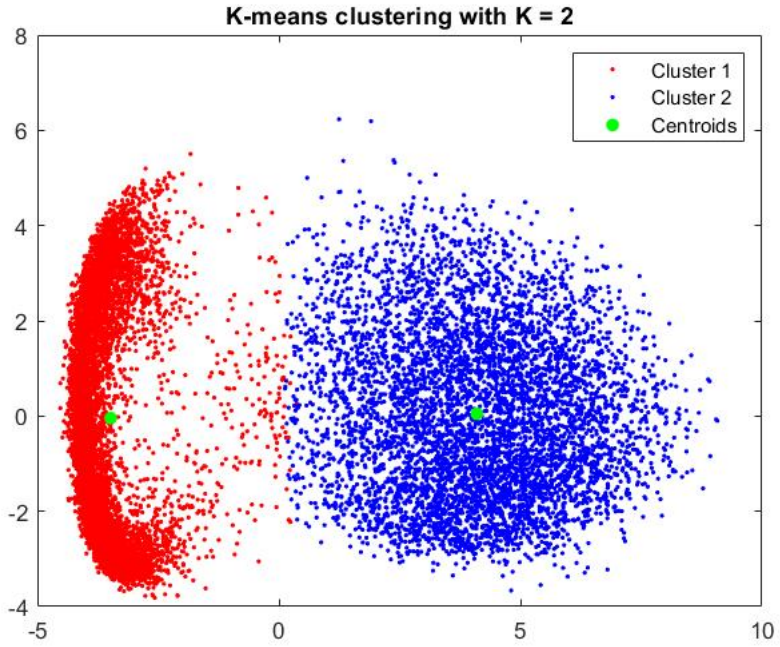

Figure 4: The two clusters after using the K-means clustering method. The red dots belongs to cluster 1 and the blue dots to cluster 2. The green dots are the centroids for each cluster.

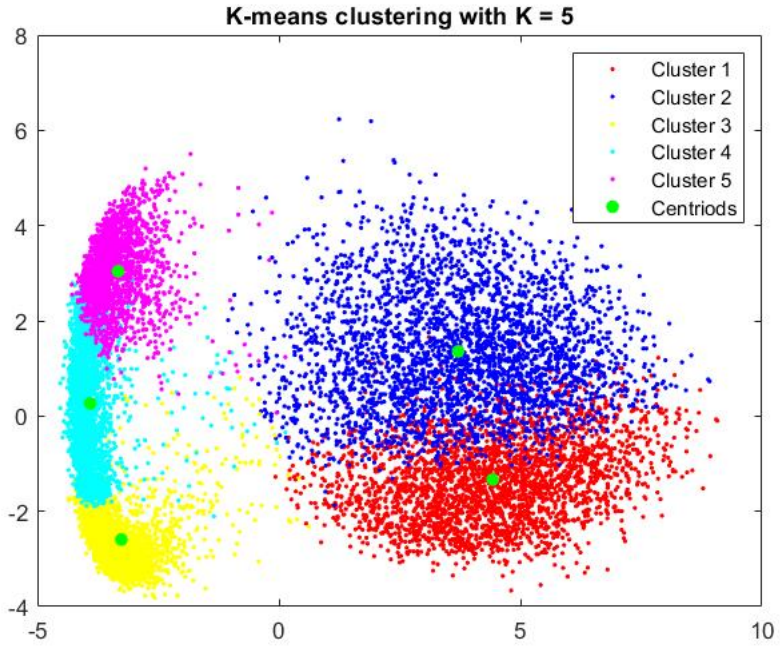

Figure 5: The resulting clusters after using K-means clustering with K=2. The red and blue dots can clearly be seen overlapping.

Task E3

The green dots in figure 4 and 5 are the centroids for each cluster. In figure 6 and 7 these can be seen as images and we can clearly see how each centroid resembles either the digit one or zero.

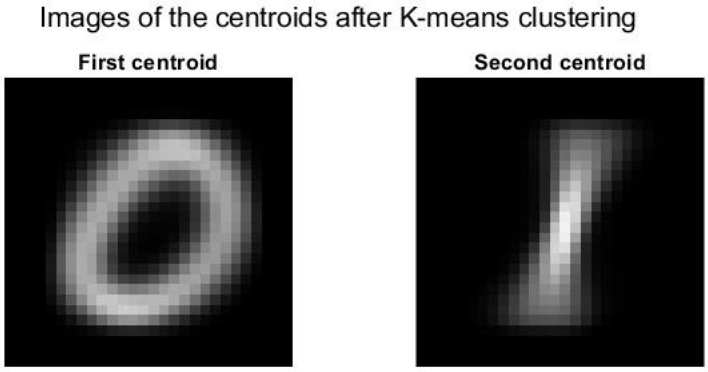

Figure 6: The centroids from the K-means clustering with K=2. The left images clearly resembles the digit zero while the right image resembles the digit one.

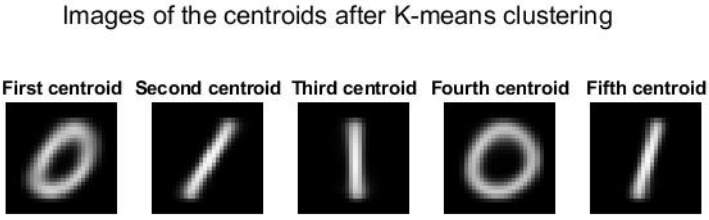

Figure 7: The five centroids after K-means clustering. The second, third and fifth images resembles the digit one and the first and fourth images resembles the digit zero.

Task E4

The next function we implemented was K-means classifier. This function first assigned the training samples into K clusters using the same method as in task E2. We then assigned each cluster a label based on the label that was mostly included in the cluster and at the same time assigned the centroids a label. These centroids could then be used to predict which digit a number contained by calculating the distance to each centroid and assigning the sample to the label of the closest centroid. In figure 8 this is done for every sample in the testing data set. In table 2 the result can be seen and the classifier performed with a misclassification rate of 0.0076 % on the test set, a rate better then the rate for the training set and got assigned 16 samples wrong.

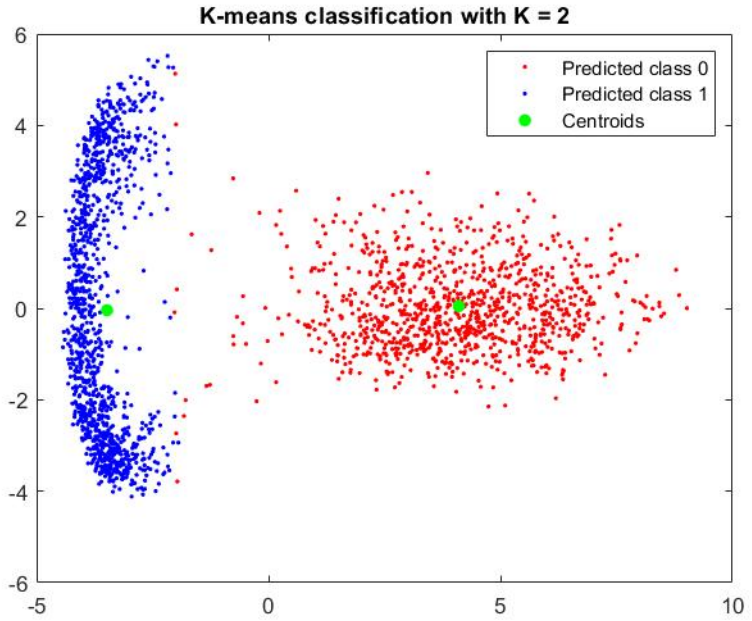

Figure 8: Predicted class using the centroids with assosiated labels. Here we used the K-means classifer with K = 2.

Table 2: K-means classification results

Training data	Cluster	# '0'	# '1'	Assigned to class	# misclassified
	1	5811	6	0	6
	2	112	6736	1	112
$N_{\text{train}} = 12665$				Sum misclassified:	118
				Misclassification rate (%):	0.0093
Testing data	Cluster	# '0'	# '1'	Assigned to class	# misclassified
	1	980	16	0	16
	2	0	1119	1	0
$N_{\text{test}} = 2115$				Sum misclassified:	16
				Misclassification rate (%):	0.0076

Task E5

We tested to increase the number of cluster from two clusters to 12, increasing with one sample each test. We got the best performance on the test set when K was equal to 10 with only 5 samples being misclassified, giving us a rate of 0.0023641 %. The improvement could be because extra clusters can specify in capturing different variations of the same digit. An example is the digit 1 that can be written with or without the lines at the top and bottom. Here for example three clusters could help to capture the three different ways the digit can be written. There could be a larger number of clusters that would perform better but for larger K values the time it takes to run the function increases significantly, making it hard to test. Taking it to the extreme one could imagine a classifier with a value of K equal to the number of samples. This would perhaps perform excellent but many of the clusters would be redundant and the value we get would be small against the computing power necessary to calculate and operate such big classifier. Another problem with searching for a K value that minimizes the misclassified samples on the test set will be discussed in task E8.

Task E6

In this task we used a supervised support vector machine (SVM) to classify our data. This was done through the built in MATLAB function *fitcsvm()* that returned a binary linear soft-margin SVM model using the samples and the corresponding labels. Here we used the parameter C equal to 1 and a linear kernel. To the predict which class an image belonged to be used the build in function *predict()* and the result for our training and test data can be seen in table 3. The model performed relatively good with only misclassifying two samples.

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'	979	1
	'1'	1	1134
$N_{\text{test}} = 2115$		Sum misclassified:	2
		Misclassification rate (%):	0.00094563

Task E7

To continue we trained an non-linear SVM using the Gaussian kernel. This was again done by using the function *fitcsvm()* but this time by specifying the kernel to be the Gaussian. This kernel has an parameter σ^2 that can be adjusted by specifying the parameter $\beta = \sqrt{1/\sigma^2}$. When we first produced an model with β equal to 1 we got 388 misclassified samples on the test data, resulting in a misclassification rate of 0.1835 %. After some testing with the value of β we achieved zero misclassifications on the test data when β was set to 5. The result for this model is seen in table 4.

Table 4: Gaussian kernel 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

Task E8

As seen in the previous task our Gaussian kernel SVM with β equal to 5 performed flawlessly on both the training and test sets. This could nevertheless be misleading due to the fact that

we searched for a value of β that performed excellent on the test data. This means that there is a risk that the parameter is being overfitted for this set and may not produce as flawless on new images. One way to minimize this risk is to use a third set of images, a validation set, to get a more general error.

0.1 Table 1

Table 5: K-means classification results

Training data	Cluster	# '0'	# '1'	Assigned to class	# misclassified
	1	5811	6	0	6
	2	112	6736	1	112
$N_{\text{train}} = 12665$				Sum misclassified:	118
				Misclassification rate (%):	0.0093
Testing data	Cluster	# '0'	# '1'	Assigned to class	# misclassified
	1	980	16	0	16
	2	0	1119	1	0
$N_{\text{test}} = 2115$				Sum misclassified:	16
				Misclassification rate (%):	0.0076

0.2 Table 2

Table 6: 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 (%):	0.00094563

0.3 Table 3

Table 7: Gaussian kernel 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