Lund University
Faculty of Engineering (LTH)

FMAN45

Machine Learning

Assignment 2

Author:
Marcel Attar, 941127-2173

The report was handed in on: May 8, 2019

LUNDS UNIVERSITET
Lunds Tekniska Högskola

Solving a nonlinear kernel SVM with hard constraints

We are looking at support vector machines with hard constraints and are given the data

Table 1: The provided data

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

Task T1

We want to compute the kernel matrix using the provided data in table 1, using the feature map $\phi(x) = (x, x^2)^T$ and kernel $k(x,y) = \phi(x)^T\phi(y)$. Computing the values for all the indices generates the following kernel matrix

$$\mathbf{K} = [k(x_i,x_j)]_{1\leq i,j\leq 4} = \begin{bmatrix} 20 & 6 & 2 & 12 \\ 6 & 2 & 0 & 2 \\ 2 & 0 & 2 & 6 \\ 12 & 2 & 6 & 20 \end{bmatrix} \tag{1}$$

Task T2

We want to solve

$$\max_{\alpha_1,\ldots,\alpha_4} \left(\sum_{i=1}^{4} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{4} \alpha_i\alpha_j y_i y_j k(x_i,x_j) \right) := \max_{\alpha_1,\ldots,\alpha_4} \mathcal{L}(\alpha_1,\ldots,\alpha_4) \tag{2}$$

$$\text{subject to} \quad \alpha_i \geq 0 \quad \text{and} \quad \sum_{i=1}^{4} y_i\alpha_i = 0$$

and we know that $\alpha := \alpha_1 = \alpha_2 = \alpha_3 = \alpha_4$. Then equation 2 becomes

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

If we now use the data in table 1 and the kernel matrix from task T1 we get

$$\max_{\alpha} \left(4\alpha - \frac{1}{2}\alpha^2(24 - 6 - 6 + 24) \right) = \max_{\alpha} \left(4\alpha - 18\alpha^2 \right)$$

By differentiating with respect to α and setting to 0 we will get the maximum point.

$$\frac{\partial \mathcal{L}}{\partial \alpha} = 0 \quad \Leftrightarrow \quad 4 - 36\alpha = 0 \quad \Leftrightarrow \quad \alpha = \frac{4}{36} = \frac{1}{9} \tag{3}$$

Task T3

We know that, for any support vector $_s$, we have

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

and that the classifier is given by

$$g(x) = \sum_{j=1}^{4} \alpha_j y_j k(x_j, x_s) + b \tag{5}$$

We now want to reduce the classifier to the simplest form as an equation of x. From before we know that $\alpha := \alpha_1 = \alpha_2 = \alpha_3 = \alpha_4$, therefore we can move the α_j out of the sum

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

The summation term becomes

$$\sum_{j=1}^{4} y_j k(x_j, x_s) = -2x + 4x^2 + x - x^2 - x - x^2 + 2x + 4x^2 = 6x^2$$

Therefore

$$g(x) = \alpha 6x^2 + b = \frac{2}{3}x^2 + b$$

Where we used the previously calculated value of $\alpha = \frac{1}{9}$ from task T2. Since all point in x_i are support vectors (by looking at table 1 you can see this) we can chose anyone of them as support vectors x_s in equation 4. If we chose $x_s = x_1 = -2$ and use in equation 4 we get

$$1 \cdot \left(\frac{2}{3}(-2)^2 + b \right) = 1 \quad \Leftrightarrow \quad b = -\frac{5}{3}$$

Finally our classifier becomes

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

Task T4

We are now given a new set of data

Table 2: The provided data

i	1	2	3	4	5	6	7
x_i	-3	-2	-1	0	1	2	4
y_i	$+1$	$+1$	-1	-1	-1	$+1$	$+1$

And are asked to calculate our classifier $g(x)$. By looking at the new dataset we see that the four points where the class shift happens, x_2, x_3, x_5 and x_6, have the same values as the previous dataset and with the same corresponding classes. Therefore, the support vectors are the same as previously and because of that the classifier function is the same

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

The Lagrangian dual of the soft margin SVM

We are now using a soft margin for the support vector machine, hence a new term has been added to the primal formulation of the linear soft margin classifier is given by

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \tag{6}$$
$$\text{subject to} \quad y_i(\mathbf{w}^T\mathbf{x_i} + b) \geq 1 - \xi_i \quad \text{and} \quad \xi_i \geq 0$$

Task T5

We now want to write equation 6 as the Lagrangian dual problem and simplify it as much as possible.

$$\mathcal{L}(\mathbf{w},b,\xi) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i\left(y_i(\mathbf{w}^T\mathbf{x_i} + b) - 1 + \xi_i\right) - \sum_{i=1}^{n}\beta_i\xi_i \tag{7}$$

where $\alpha_i \geq 0$ and $\beta_i \geq 0$. Equation 6 now can be written as

$$\max_{\alpha_1,\ldots,\alpha_n}\min_{\mathbf{w},b,\xi}\mathcal{L}(\mathbf{w},b,\xi) \tag{8}$$

We get the KKT conditions by taking the gradient and setting it to 0

$$\frac{\partial\mathcal{L}}{\partial\mathbf{w}} = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^{n}\alpha_i y_i \mathbf{x_i} \tag{9}$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \quad \Rightarrow \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{10}$$

$$\frac{\partial \mathcal{L}}{\partial \xi_i} = 0 \quad \Rightarrow \quad C - \beta_i - \alpha_i = 0 \quad \Rightarrow \quad C = \beta_i + \alpha_i \tag{11}$$

Let's use these three equations and that $\|\mathbf{w}\|^2 = \mathbf{w}^T\mathbf{w}$, giving us

$$\mathcal{L} = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{n}\xi_i + B$$

and B we define as

$$B := -\sum_{i=1}^{n}\left(\alpha_i y_i \left(\sum_{j=1}^{n}\alpha_j y_j \mathbf{x_j}\right)^T \mathbf{x_i} + \alpha_i y_i b - \alpha_i + \alpha_i \xi_i\right) - \sum_{i=1}^{n}\beta_i\xi_i$$

We can simplify this to

$$B = -\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j {\mathbf{x_i}}^T\mathbf{x_j} - \sum_{i=1}^{n}\alpha_i\xi_i + \sum_{i=1}^{n}\alpha_i - \sum_{i=1}^{n}\beta_i\xi_i$$

If we now plug this in to the Lagrange function and substitute C with equation 11 we get

$$\max_{\alpha_1,\ldots,\alpha_n}\mathcal{L} = \max_{\alpha_1,\ldots,\alpha_n}\left(\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j {\mathbf{x_i}}^T\mathbf{x_j}\right) \tag{12}$$

$$\text{subject to} \quad \sum_{i=1}^{n}\alpha_i y_i = 0$$

and we know that α_i, $\beta_i \geq 0$ and that

$$C = \beta_i + \alpha_i \quad \Rightarrow \quad \alpha_i = C - \beta_i$$

Therefore, the maximum value of α_i is C, meaning that

$$0 \leq \alpha_i \leq C$$

Task T6

We want to show that support vectors with $y_i(\mathbf{w}^T\mathbf{x_i} + b) < 1$ have coefficient $\alpha_i = C$.

Proof. The complementary slackness is

$$\alpha_i\left(y_i(\mathbf{w}^T\mathbf{x_i} + b) - 1 + \xi_i\right) = 0 \tag{13}$$

$$\beta_i\xi_i = 0 \tag{14}$$

and from equation 11 we have $C = \beta_i + \alpha_i$. Using this in equation 14 we get

$$(C - \alpha_i)\xi_i = 0$$

For $\alpha_i = C$ the condition above is satisfied if $\xi_i > 0$. Condition 13 is only satisfied if

$$y_i(\mathbf{w}^T\mathbf{x_i} + b) - 1 + \xi_i = 0 \quad \Rightarrow \quad \xi_i = 1 - y_i(\mathbf{w}^T\mathbf{x_i} + b)$$

This can then be used in $\xi_i > 0$ leading to

$$1 - y_i(\mathbf{w}^T\mathbf{x_i} + b) > 0 \quad \Rightarrow \quad y_i(\mathbf{w}^T\mathbf{x_i} + b) < 1$$

□

Dimensionality reduction on MNIST using PCA

Task E1

See attached code for implementation. The plot that was produced using Principal Component Analysis (PCA) is shown below.

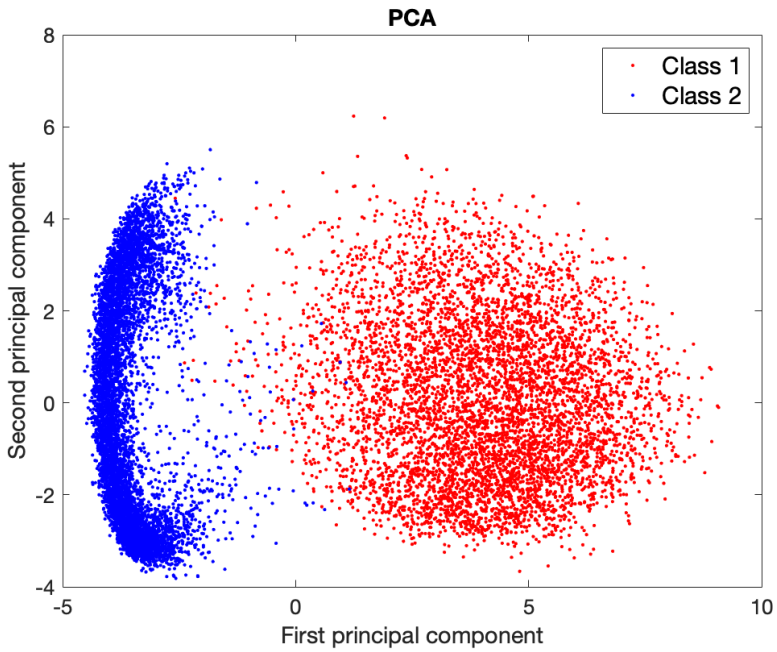

Figure 1: A plot of the data 'train_data_01' being projected down to the two principal components using PCA. The two classes are almost two distinct clusters, although with some overlap.

Task E2

See attached code for implementation. The two plots below are for clustering the data using the K-means clustering algorithm, the first plot is for when we divide into two clusters and the second is for when we divide into five clusters.

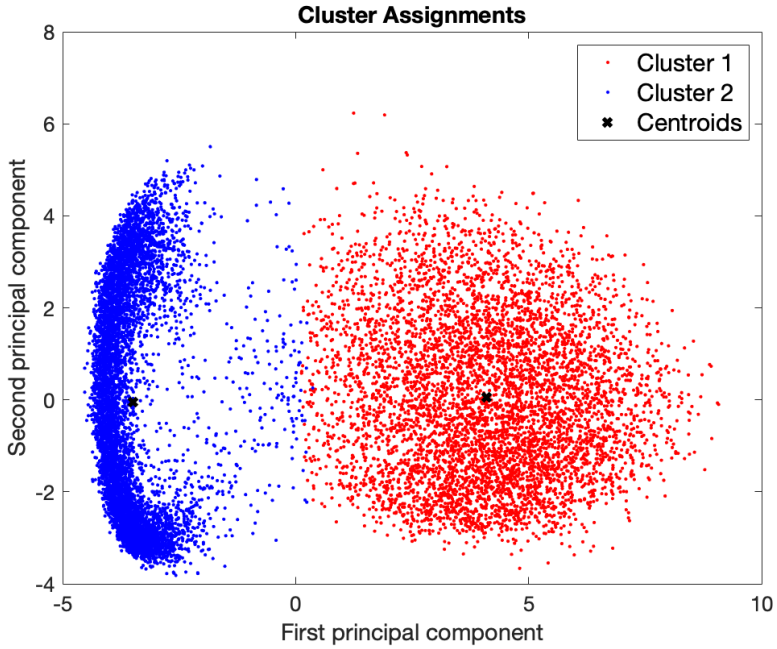

Figure 2: Two clusters have been assigned using the K-means clustering approach on the data 'train_data_01'. Some small overlap occur with the two clusters.

Comparing this with figure 1 we know that there are some samples that would be misclassified if we used this approach.

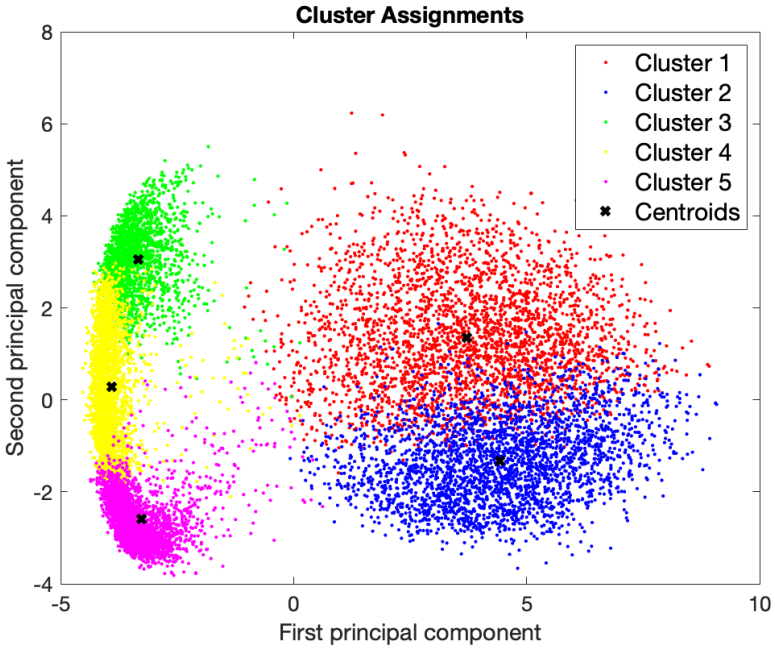

Figure 3: Five clusters have been assigned using the K-means clustering approach on the data 'train_data_01'. This time a lot of overlap occurs.

The reason for the big overlap, when we have five clusters, is that we performed the K-means clustering algorithm on the data *before* we projected it onto the principal components. If we would have done it the other way around we would have gotten straight lines as separators (i.e. no overlap) in 2D. This way is better because we use all the data, before any dimensionality reduction that would lead to information loss.

Task E3

See attached code for implementation.

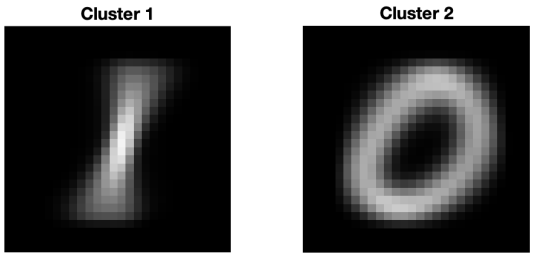

Figure 4: Here the data 'train_data_01' has been divided into two clusters using the K-means approach. The figure represents the two centroids of the two clusters.

Figure 5: Here the data 'train_data_01' has been divided into five clusters using the K-means approach. The figure represents the five centroids of the five clusters.

As can be observed from the figures, when dividing into more clusters, we get

sharper images.

Task E4

See attached code for implementation. The result of the classification can be seen below for two clusters on two different datasets.

Table 3: 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 (%):	0.93%
Testing data	Cluster	# '0'	# '1'	Assigned to class	# misclassified
	1	11	1135	1	11
	2	969	0	0	0
$N_{\text{test}} = 2115$				Sum misclassified:	11
				Misclassification rate (%):	0.52%

E5

If we instead use five clusters we get the following results.

Table 4: K-means classification results

Training data	Cluster	# '0'	# '1'	Assigned to class	# misclassified
	1	1850	0	0	0
	2	23	3692	1	23
	3	20	3042	1	20
	4	2275	0	0	0
	5	1755	8	0	8
$N_{\text{train}} = 12665$				Sum misclassified:	51
				Misclassification rate (%):	0.40%
Testing data	Cluster	# '0'	# '1'	Assigned to class	# misclassified
	1	544	0	0	0
	2	1	387	1	1
	3	3	325	1	3
	4	3	423	1	3
	5	429	0	0	0
$N_{\text{test}} = 2115$				Sum misclassified:	7
				Misclassification rate (%):	0.33%

As we can see this yields better results for both datasets.

Classification of MNIST digits using SVM

E6

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%

The linear SVM produces no misclassifications on the two datasets.

E7

Table 6: Gaussian kernel SVM classification results with $\beta = 1$

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%

The Gaussian works just as well as the linear SVM on these datasets, however, the Gaussian was considerably slower.

E8

We can't expect the same, low to none errors in misclassification due to the fact that kernel specification is very unique to each problem in SVM classification.