ML Ass2

Kuan Teh Wan

May 2024

1 Exercise 1

To compute the kernel matrix **K**, we use the given dataset, the non-linear feature map $\Phi(x)$ and $k(x,y) = \Phi(x)^T\Phi(y)$. Then we have

$$\mathbf{K} = \begin{bmatrix} 20 & 6 & 2 & 12 \\ 6 & 2 & 0 & 2 \\ 2 & 0 & 2 & 6 \\ 12 & 2 & 6 & 20 \end{bmatrix} \tag{1}$$

2 Exercise 2

Using the fact that the solution satisfies $\alpha = \alpha_1 = \alpha_2 = \alpha_3 = \alpha_4$, we can simplify the problem to

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

subject to $\alpha \geq =0$ and$\sum_{i=4} y_i\alpha$=0.

Next, take a look at the submission term, by doing some calculations using the elements from matrix **K** and dataset.

$$\sum_{i,j=1}^{4} y_i y_j k(x_i, x_j) = 1 \cdot (2 \cdot 20 + 2 \cdot 12 + 2 \cdot 2) - 1 \cdot (4 \cdot 6 + 4 \cdot 2) = 36 \tag{3}$$

according to (2) and together with this we have

$$\underset{\alpha}{\text{maximize}}\ 4\alpha - 18\alpha^2 \tag{4}$$

And with this, we know the function is concave since its second derivative is negative

$$\frac{d^2}{d\alpha^2}(4\alpha - 18\alpha^2) = -36 < 0 \tag{5}$$

Thus we can find the maximum value when its first derivative equals to zero

$$\frac{d}{d\alpha}(4\alpha - 18\alpha^2) = 4 - 36\alpha = 0, \alpha = \frac{1}{9} \tag{6}$$

3 Exercise 3

To reduce the classifier function with $\alpha = \frac{1}{9}$ we found in (6)

$$g(x) = \frac{1}{9}\sum_{j=1}^{4} y_j k(x_j, x) + b \tag{7}$$

we begin by plugging values from data set and $k(x_j, x)$

$$g(x) = \frac{1}{9}((-2x+4x^2)-(-x+x^2)-(x+x^2)+(2x+4x^2))+b = \frac{2}{3}x^2+b \tag{8}$$

And b can be found by any support vector with

$$y_s(\frac{2}{3}x^2 + b) = 1 \Rightarrow 1(\frac{2}{3}(-2)^2 + b) = 1 \Rightarrow b = \frac{-5}{3} \tag{9}$$

Thus the form

$$g(x) = \frac{2}{3}x^2 - \frac{5}{3} \tag{10}$$

4 Exercise 4

After observing the new data, we can see that it is a superset of the previously given dataset since $x_2, x_3, x_5 and x_6$ are the same. Therefore, we can use the classifier that we derived in (10).

5 Exercise 5

We begin by writing the Lagrangian

$$L(\omega, b, \xi) = \frac{1}{2}||\omega||^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n} a_i(\xi_i - 1 + y_i(\omega^T x_i + b)) - \sum_{i=1}^{n}\lambda_i\xi_i \tag{11}$$

subject to $\alpha_i, \lambda_i \geq 0$. To derive the dual problem, we will minimize L w.r.b to ω, b, ξ. And setting the derivatives to zero we have:

$$\frac{dL}{dw} = 0 = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0 \iff w = \sum_{i=1}^{n} \alpha_i y_i x_i \tag{12}$$

$$\frac{dL}{db} = 0 = -\sum_{i=1}^{n} \alpha_i y_i = 0 \iff \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{13}$$

$$\frac{dL}{d\xi} = 0 = C - \alpha_i - \lambda_i = 0 \iff \lambda_i = C - \alpha_i \tag{14}$$

And by plugging (12) to (11), we have

$$\begin{aligned}
\max_{\alpha_1,\dots,\alpha_n} L &= \frac{1}{2}||\sum_{i=1}^{n} \alpha_i y_i x_i||^2 + C\sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i(\xi_i - 1 + y_i(\sum_{i=1}^{n} \alpha_i y_i x_i)^T x_i + b)) - \sum_{i=1}^{n} \lambda_i \xi_i \\
&= -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_j y_j x_j^T x_j + \sum_{i=1}^{n} \xi_i(C - \alpha_i - \lambda_i) + \sum_{i=1}^{n} \alpha_i - \sum_{i=1^n} \alpha_i y_i b \\
&= \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_j y_j x_j^T x_j
\end{aligned} \tag{15}$$

Together with $\alpha_i, \lambda_i \geq 0$ from (11) and (14) the constraint is

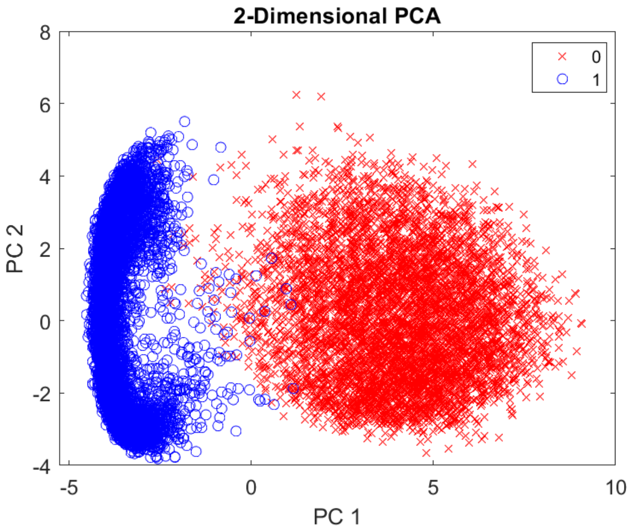
$$0 \leq \alpha_i \leq C, \forall i \tag{16}$$

6 Exercise 6

For the specified support vectors, we have that ξ_i >0. By complementary slackness of the KKT conditions we have that λ_i = 0 because the constraint on λ for $\xi \geq 0$ is not active.Thus α_i = C, from equation (14)

7 Exercise 7

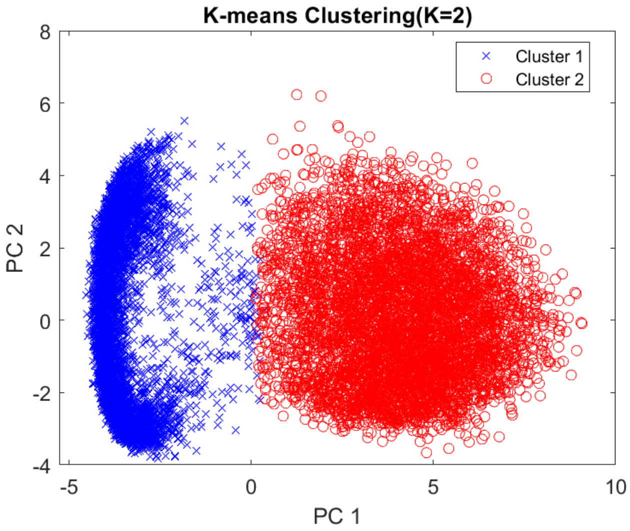
We begin by normalizing data to zero mean before PCA. Using SVD the first two left singular vectors having the largest eigenvalues were found as first and second principal components. Then, the normalized data are projected on PC1 and PC2 as shown in figure 1 below.



Figur 1: MNIST labeled data projected onto PC1, PC2

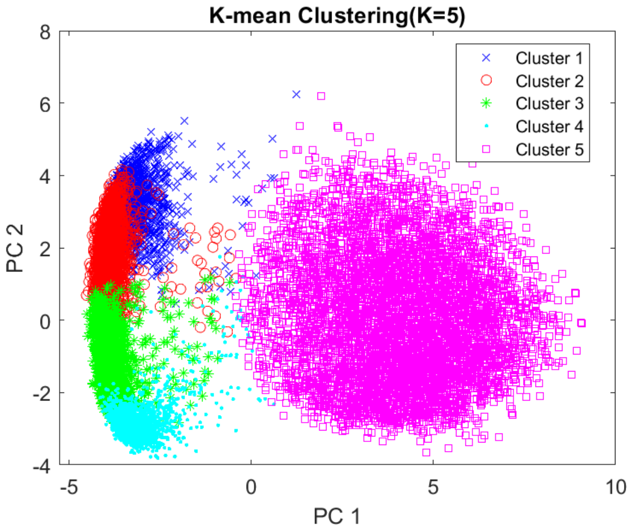
8 Exercise 8

In figure 2, the PCA visualization of K-means clustering(K=2) is shown. We can see that is almost the same as figure 1.



Figur 2: PCA visualization of K-means clustering(K=2)

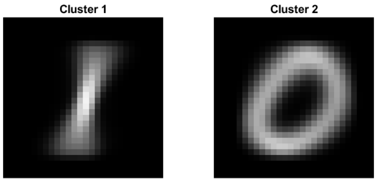
As for figure 3, the PCA visualization of K-means clustering(K=5), there are some clusters overlapping. The reason behind this is that, the classification is done in a higher dimension first then PCA reduces the dimension for better visualization. This can also be seen as loss of information while having Dimensional reduction.



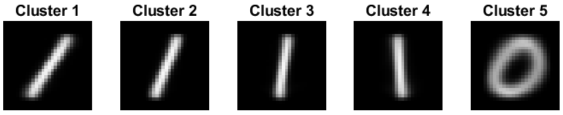
Figur 3: PCA visualization of K-means clustering(K=5)

9 Exercise 9

K = 2 and K = 5 centroids are displayed below, figure 4 and 5 respectively.



Figur 4: K = 2 centroids



Figur 5: K = 5 centroids

10 Exercise 10

Evaluation on classification with K-means classification(K = 2) is shown in the table below.

Tabell 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 (%):	0.947
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 (%):	0.567

11 Exercise 11

Yes, with K=8 we can lower the misclassification rate on test data down to about 0.189% from 0.567% originally (K=2).

12 Exercise 12

Evaluation on classification with Support Vector Machine(SVM) is as shown. We can see that it performs well, better than the results we have previously with K-means.

Tabell 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

13 Exercise 13

Here, we’ll train a non-linear SVM classifier with using a Gaussian kernel and provide its evaluation in a table same as before.

We're using an increment of 0.5 ranging from 1 to 6 find which beta yields the best results. It turns out when $\beta = 5$ we have perfect classification.

Tabell 3: 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

14 Exercise 14

Although with the optimal beta we achieved perfect classification in this case, but it won't happen again if given new unseen images since tuning the hyperparameters tends to lead to overfitting.