EE3-23: Assignment 4

Spring, 2017

Instructions:

To get a full mark for this assignment, you need to solve only 80% (36 points). However, you must specify which questions you want to have marked.

You need to submit a single zip file on Blackboard, named `assignment4_<name>.zip` where `<name>` is your name. The file should contain a report in PDF (write your name on your solution) named `assignment4_<username>_<name>.pdf` where `<username>` is your blackboard user name, and your code and other relevant files. Your code must include a main file named `assignment4_<username>` with a possible extension (such as `.m` or `.py`) or a Makefile which produces the main file on any standard linux system: the main file should be either an executable or a script file, which generates all the data and plots required in your solution.

Justify your answers. You are allowed to consult with others (mention their names at the beginning of your solution), but you need to work out and write up your solution alone. For the last problem, you can use any available library functions, but make sure you explain what they do if it is asked. For the other problems, no external sources should be used (other than the recommended textbooks and any external sources related to matrix calculus). It is specifically considered cheating to search for the solutions of the problems and copy them.

Include the following pledge into your answer:

I, <YOUR NAME>, pledge that this assignment is completely my own work, and that I did not take, borrow or steal work from any other person, and that I did not allow any other person to use, have, borrow or steal portions of my work. I understand that if I violate this honesty pledge, I am subject to disciplinary action pursuant to the appropriate sections of Imperial College London.

If you don't include your pledge, you will get 0%.

Deadline: March 24, 23:59:59, followed by a 5-minute grace period.

Problem 1. Consider the following modified version of linear regression: We are given a dataset $\{(x_i, y_i), i = 1, \dots, n\}$, where $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}^p$, thus, for each feature vector x_i we want to estimate p values. Furthermore, we want to give different weights $\gamma_i > 0$ to errors made on different data points, and we want to learn the weight matrix $w \in \mathbb{R}^{d \times p}$ to minimize the weighted empirical error with Tykhonov regularization:

$$J(w) = \sum_{j=1}^{p} \sum_{i=1}^{n} \gamma_i \Big((w^\top x_i)_j - y_{i,j} \Big)^2 + \sum_{j=1}^{p} \sum_{k=1}^{d} w_{k,j}^2 \ .$$

(a) Give a closed for expression for the minimizer w^* minimizing the augmented weighted empirical error $J(w)$ using the data matrix $X = (x_1, \dots, x_n)^\top$, the matrix of the target values $Y = (y_1, \dots, y_n)^\top$(each column corresponds to the different targets), and the diagonal weight matrix $\Gamma = \mathrm{diag}(\gamma_1, \dots, \gamma_n)$. **[6 points]**

Hint: Standard formulas for matrix differentiation are available, e.g., at `http://www.atmos.washington.edu/~dennis/MatrixCalculus.pdf` or `http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/3274/pdf/imm3274.pdf`.

(b) For $\gamma_i = \gamma$ for all i, show that a kernelized version of the linear prediction is possible, where first a transformation $\phi : \mathbb{R}^d \to \mathcal{Z}$ is applied to the input features x, where ϕ is the transformation corresponding to the kernel $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ (and $\mathcal{Z}$ is a reproducing kernel Hilbert space), and predictions are of the form $w^\top \phi(x)$. That is, how the resulting weight can be expressed using only K, without the need for computing ϕ for any feature vector. **[4 points]**

Problem 2 (Support vectors)**.** Suppose we have a binary classification problem with features in $\mathbb{R}^d$, where the classes are linearly separable with some positive margin.

(a) Show that the maximum margin linear classifier does not change if we remove a data point from our training set which is not a support vector or add a new data point that does not violate the current maximum margin. **[6 points]**

Hint: Show that the maximum margin classifier is unique.

(b) Show that there can be at most $d+1$ support vectors such that the maximum margin classifier changes if the support vector is removed from the data. **[3 points]**

(c) Conclude that the expected test error of the maximum margin classifier (taken over randomly selected data sets of size n) is at most $\frac{d+1}{n+1}$. **[2 points]**

(d) Can you make similar statements about soft-margin SVMs? **[4 points]**

Problem 3. Consider a binary classification problem where the training set is linearly separable. Given this knowledge, can you argue that using a hard-margin SVM is always better than using soft-margin SVM? Describe some scenarios supporting your argument. **[5 points]**

Problem 4 (Digit classification)**.** Consider the handwritten digit classification problem considered in Assignment 1, and use the datasets provided there.

(a) Learn a large margin separator using SVM with Gaussian (RBF) kernel to separate the digits 2 and 8 based on the raw data (zip.train). Report your results: Specifically, comment on how you computed cross-validation for finding the parameters of the SVM and plot the cross-validation error. Compare the results to what is achieved by the perceptron algorithm in Assignment 1 (just use previous results, e.g., from the sample solution or your own). **[4 points]**

(b) Use principal component analysis to reduce the dimension of the data: Determine the principal component directions $V = [v_1, \ldots, v_d]$ (here d is the dimension of the original data set and the v_i vectors are in a decreasing order relative to their corresponding singular values (eigenvalues)–do not forget centering the input features!). You do not need to report these. For some $1 \le k \le d$, we use $(v_1^\top x, \ldots, v_k^\top x)$ as the k-dimensional approximation of the input vector x.

Write a program that, for any given k, trains a soft-margin SVM on the data for digits 2 and 8 using the k-dimensional approximation of the input features. Describe the training procedure (refer to part (a) and describe any difference if necessary) and plot the training/cross-validation/test error of the models learned as a function of k (no other figures are necessary). Comparing the above graphs, argue if more training data would be helpful for different values of k. Can you make the same conclusion during training (i.e., based on the training and the cross-validation error)? **[8 points]**

(c) Comparing digit 1 against all the others: In this problem one class is formed by digit 1, while all other digits correspond to the other class. Show a scatter plot of this problem based on (i) the hand-crafted feature vectors provided; and (ii) the 2-dimensional PCA approximation of the data. Train an SVM classifier for both cases and show the decision boundary on the scatter plots. Comment on your findings. **[3 points]**