Introduction to learning and analysis of big data
Exercise 3

Dr. Sivan Sabato

Fall 2019/20

Submission guidelines, **please read and follow carefully**:

- You may submit the exercise in pairs.
- Submit using the submission system.
- The submission should be a zip file named "ex3.zip".
- The zip file should include **only the following files in the root - no subdirectories please**.
- The files in the zip file should be:
  1. A file called "answers.pdf" - The answers to the questions, including the graphs.
  2. An ".m" file for each of the requested Matlab functions.

  Anywhere in the exercise where Matlab is mentioned, you can use the free software Octave instead.
- For questions use the course Forum, or if they are not of public interest, send them via the course requests system.
- Grading: Q1(a): 19 points; Q1(b),Q1(c): 9 points each. 7 points for each sub-question in Q2-Q5.

Question 1 For this question, use the data file `EX3q1_data.mat`, which contains data points $x_i \in \mathbb{R}^2$ and labels $y_i \in \{-1, 1\}$. There are 2000 training examples and 200 test points.

(a) Implement the soft-margin kernel SVM routine described in class, using MATLAB's `quadprog` command. Use the Gaussian (RBF) kernel. The function should be implemented in the submitted file called "softsvmrbf.m". The first line in the file (the signature of the function) should be:

```
function alpha = softsvmrbf(lambda, sigma, m, d, Xtrain, Ytrain)
```

The input parameters are:

- `lambda` - the parameter λ of the soft SVM algorithm.
- `sigma` - the bandwidth parameter σ of the RBF kernel.

- m - the size of the training sample $S = ((x_1, y_1), \ldots, (x_m, y_m))$, an integer $m \geq 1$.
- `d` - the number of features in each example in the training sample, and integer $d \geq 1$. So $\mathcal{X} = \mathbb{R}^d$.
- `Xtrain` - a 2-D matrix of size $m \times d$ (note: first number is the number of rows, second is the number of columns). Row i in this matrix is a vector with d coordinates that describes example x_i from the training sample.
- `Ytrain` - a column vector of length m (that is, a matrix of size $m \times 1$). The i's number in this vector is the label y_i from the training sample. You can assume that each label is either -1 or 1. **Important: The labels in the input to soft SVM should be -1 or 1, and not 0 or 1.**

The function returns the variable `alpha`. This is the vector of coefficients found by the algorithm, $\alpha \in \mathbb{R}^m$, a column vector of length `m`.

(b) Perform 10-fold cross-validation to tune λ and σ. Try the values $\lambda \in \{0.01, 0.1, 1\}$ and $\sigma \in \{0.01, 0.05, 1, 2\}$ — a total of 12 parameter pairs to try. For each of the 12 (λ, σ) pairs, report the following:

- The average cross-validation error
- The error, as measured on the test set, when using this pair to train a classifier on the entire training set.
- The difference, in absolute value, between the two values above

Now, answer the following questions:

- What is the optimal pair according to the cross validation procedure?
- Is it different from the optimal pair as measured on the test set?
- Based on you answers to the questions above, do you think the cross-validation procedure was sufficiently successful? Explain.

(c) set $\lambda = 0.01$ and consider $\sigma \in \{0.01, 0.05, 1, 2\}$. For these values, run the soft-margin RBF SVM and plot the function that each of the resulting classifiers induces on the original $\mathbb{R}^2$ space. To do this, divide the space into a fine grid, and color the grid points red or green, depending on whether they are labeled positive or negative by the classifier. You can use Matlab's `heatmap` routine.

Discuss in words the differences that you see between the various values of σ, and try to explain them by referring to the different behavior of smaller and larger values of σ that we discussed in class.

Question 2 Let $\mathcal{X}$ be the set of all undirected graphs over n vertices numbered $1, \ldots, n$ with degree at most 5. Let $\mathcal{Y} = \{0, 1\}$. For a graph $x \in \mathcal{X}$, define the mapping $g : \mathcal{X} \to \mathbb{N}^n$, where coordinate i in the vector $g(x)$ is the degree of vertex i in the graph x. Let $\mathcal{H} = \{h_v : \mathcal{X} \to \mathcal{Y} \mid v \in \mathbb{N}^n, h_v \not\equiv 0\}$, where $h_v(x) := \mathbb{I}[g(x) = v]$.

Suppose that $\mathcal{D}$ is a distribution over $\mathcal{X} \times \mathcal{Y}$, and suppose that in this distribution, the label of a graph x is a deterministic function of $g(x)$.

(a) Use the **size** of the hypothesis class and the PAC-learning upper bound that we showed in class to show that the sample complexity of learning $\mathcal{H}$ as a function of n is $O(n)$.

(b) What is the best dependence on ϵ that you can get in the sample complexity using the PAC bounds that we learned in class? Why?

(c) What is the VC dimension of $\mathcal{H}$? Use this value to get a better upper bound for the sample complexity of learning $\mathcal{H}$ as a function of n and ϵ.

Question 3 Let $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$. Consider a Gradient Descent algorithm that attempts to minimize the following objective:

$$\text{Minimize}_{w \in \mathbb{R}^d} \lambda \|w\| + \sum_{i=1}^{m} (\langle w, x_i \rangle - y_i)^2.$$

(a) Show that the objective is convex. What does it mean about running gradient descent?

(b) Suppose that Gradient Descent is run on S with a step size η. Calculate the the formula for $w^{(t+1)}$ as a function of $w^{(t)}$ and η. Explain the steps of your derivation.

(c) What would the update step for $w^{(t+1)}$ be in Stochastic Gradient Descent for the same objective?

Question 4 **Kernel functions.** Consider a space of examples $\mathcal{X} = \mathbb{R}^d$. Let $x, x' \in \mathcal{X}$.

(a) Prove that the following function *cannot be* a kernel function for any feature mapping ψ:

$$K(x, x') := -x(1)x'(1).$$

Hint: consider the case of $x = x'$.

(b) Prove that the following function *cannot be* a kernel function for any feature mapping ψ:

$$K(x, x') := (x(1) + x(2))(x'(3) + x'(4)).$$

Question 5 In an election poll, n random draws of voters are selected. Each of the voters is asked to which party they like. There are 30 possible parties. Assume that all people gave one of the parties as an answer. Let $\hat{p}_i$ be the fraction of people in the poll who answered that they will vote for party i. Use Hoeffding's bound to calculate the smallest number n such that with a probability at least 97%, for all $i \in \{1, \ldots, K\}$ the proportion in the population of people who like party i is within $\pm 5\%$ from $\hat{p}_i$.