Exercise 3

Prof. Kontorovich and Dr. Sabato

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 **exactly two 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. A file called "softsvmrbf.m" - The Matlab/Octave code for the requested function. Note that you can put several auxiliary functions in this file after the definition of the main function. **Make sure that the single file works in Matlab/Octave before you submit it.**
- **Anywhere in the exercise where Matlab is mentioned, you can use Octave instead**.
- Grading: Question 1: 50 points. Question 2: 50 points.
- For questions use the course Forum, or email `inabd161@gmail.com`.

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 1000 training examples and 100 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 `d`.

(b) Perform 10-fold cross-validation to tune λ and σ. Try the values $\lambda \in \{0.01, .1, 1\}$ and $\sigma \in \{0.05, 1, 2\}$ — a total of 9 parameter pairs to try. Report the 9 validation error values for each of the pairs (λ, σ) as well as the optimal pair (i.e., the one that achieves the lowest validation error) and its performance on the test set. Note: the data contains 1000 labeled examples, but MATLAB's `quadprog` cannot handle such large matrices. Experiment to see what sample sizes it can handle. Report what sample size you end up using.

(c) set $\lambda = 0.01$ and consider $\sigma \in \{0.05, 1, 2\}$. For these values, run the soft-margin RBF SVM and plot the decision boundary in $\mathbb{R}^2$. Think about how to draw a decision boundary in $\mathbb{R}^2$. One idea is to take a fixed region (roughly the region in which the training data resides), divide it into a fine grid, and color the grid points red or green, depending on where they fall under the classifier's prediction. You can use the `heatmap` routine for visualizing this.

Question 2. For this question, you may use the PAC bounds proved in class.

For $n \geq 1$, let $\mathcal{H}_n$ be the collection of all MATLAB programs of length n (in ASCII characters, of which there are 128) that receive a natural number $x \in \mathbb{N}$ and halt in finitely many steps with output $y \in \{-1, 1\}$.

(a) Let program length n be fixed and suppose that the distribution $\mathcal{D}$ over $\mathbb{N} \times \{-1, 1\}$ is realizable by $\mathcal{H}_n$ — that is, there is an $h \in \mathcal{H}_n$ that achieves $\mathrm{err}(h, \mathcal{D}) = 0$. Give a PAC sample complexity bound for $\mathcal{H}_n$. Meaning: what sample size $S \sim \mathcal{D}^m$ suffices to guarantee, with probability at least $1 - \delta$, that $\mathrm{err}(\hat{h}_S, \mathcal{D}) \leq \epsilon$? Your answer will depend on ϵ, δ, n.

(b) Now consider the hypothesis class

$$\mathcal{H} = \bigcup_{n=1}^{\infty} \mathcal{H}_n$$

and assume that $\mathcal{D}$ is realizable by $\mathcal{H}$.

Give an inuitive argument for why $\mathcal{H}$ is not PAC-learnable. Meaning: one cannot specify any sample size $m_0(\epsilon, \delta)$ (as a function of ϵ, δ alone) so as to guarantee, with probability at least $1 - \delta$, that $\mathrm{err}(\hat{h}_S, \mathcal{D}) \leq \epsilon$.

(c) Consider the following "Occam learner" for $\mathcal{H} = \bigcup_{n=1}^{\infty} \mathcal{H}_n$ in the realizable case. Given a labeled sample $S = (x_i, y_i)$, $i \leq m$, OCCAM selects the shortest $\hat{h}_S \in \mathcal{H}$ that is consistent with S (i.e., achieves $\mathrm{err}(h, S) = 0$). Prove the following claim: with probability at least $1 - \delta$, the OCCAM learner's $\hat{h}_S$ satisfies

$$\mathrm{err}(\hat{h}_S) \leq \frac{|\hat{h}_S| \log 256 + \log(1/\delta)}{m},$$

where $|h|$ is the length of the program h.

Hint: for fixed $\delta > 0$ and $n = 1, 2, \ldots$, define the sequence

$$\epsilon_n = \frac{n \log 128 + \log(2^n/\delta)}{m}$$

and consider the sequence of "bad events" B_n that some $h \in \mathcal{H}_n$ fools OCCAM by being consistent with S while having $\mathrm{err}(h, \mathcal{D}) > \epsilon_n$. Proceed by (i) bounding $\mathbb{P}(B_n)$ and (ii) showing that

$$\sum_{n=1}^{\infty} \mathbb{P}(B_n) \leq \delta.$$