Assignment 4: Support Vector Machines, Adaboost and EM

For due date please see course management system
110 marks total, graded part = 60 marks

This assignment is to be done individually.

Important Note: The university policy on academic dishonesty (cheating) will be taken very seriously in this course. You may not provide or use any solution, in whole or in part, to or by another student.

You are encouraged to discuss the concepts involved in the questions with other students. If you are in doubt as to what constitutes acceptable discussion, please ask! Further, please take advantage of office hours offered by the instructor and the TA if you are having difficulties with this assignment.

DO NOT:

- Give/receive code or proofs to/from other students
- Use Google to find solutions for assignment

DO:

- Meet with other students to discuss assignment (it is best not to take any notes during such meetings, and to re-work assignment on your own)
- Use online resources (e.g. Wikipedia) to understand the concepts needed to solve the assignment

---

Question 1 (10 marks)

Show that the exponential kernel $k(\boldsymbol{x}, \boldsymbol{x}') = \exp(k_1(\boldsymbol{x}, \boldsymbol{x}'))$ (Eqn. 6.16) corresponds to a dot product in an infinite dimensional feature space.

- Start by writing down exp as a power series (Taylor expansion around 0).
- You may then make use of Eqn. 6.15, which states that for a polynomial $q(k_1(\boldsymbol{x}, \boldsymbol{x}'))$, e.g. $a_d k_1(\boldsymbol{x}, \boldsymbol{x}')^d$ with $a_d > 0$, there exists a feature space $\boldsymbol{\phi}_d(\boldsymbol{x})$ such that $q(k_1(\boldsymbol{x}, \boldsymbol{x}'))$ acts as a dot product in that space.
- Write down the infinite dimensional space in which $\exp(k_1(\boldsymbol{x}, \boldsymbol{x}'))$ corresponds to a dot product (using the spaces from above).

Question 2 (10 marks)

Consider a linear classifier with decision boundary

$$0 = g(\boldsymbol{x}) = \boldsymbol{w} \bullet \boldsymbol{x} + b,$$

where $\bullet$ denotes the dot product of two vectors (sum of products of components). A key fact for the max-margin classifier is that the unsigned distance from a point (vector) $\boldsymbol{y}$ to the decision boundary is given by

$$\frac{g(\boldsymbol{y})}{||\boldsymbol{w}||}.$$

Prove this fact by using the following approach: Show that if $\boldsymbol{x}$ is the closest point to y on the decision surface, then the distance $||\boldsymbol{y} - \boldsymbol{x}||$ equals the absolute value of $g(\boldsymbol{y})/||\boldsymbol{w}||$. Well break this down into two steps.

1. (4 points) To find the closest point on the decision surface, write down the Lagrangian $L(\boldsymbol{x}, \lambda)$ of the function

   $$f(\boldsymbol{x}) = 1/2||\boldsymbol{y} - \boldsymbol{x}||^2$$

   given the constraint that $0 = g(\boldsymbol{x})$. Write down the stationary point equations of L (i.e. the equations that set the partial derivatives of L to 0).

2. (6 points) Show that assuming that these equations hold, the distance $||\boldsymbol{y} - \boldsymbol{x}||$ equals the absolute value of $g(\boldsymbol{y})/||\boldsymbol{w}||$.

Question 3 (20 marks)

Question 9.19 in PRML.

- Start by introducing a latent variable $\boldsymbol{z}_n$ of appropriate type.
- For the E-step, you should provide equations for calculating the responsibilities $\gamma(\cdot)$
- For the M-step, write down the complete-data log-likelihood
- It will involve the latent variable $\boldsymbol{z}_n$. Replace this with the appropriate responsibility $\gamma(\cdot)$. This is the expected-complete-data log-likelihood.
- Take the derivatives of this with respect to parameters.
- State how you would solve for the optimal parameter values using these derivatives. **You do not need to actually do so.**

If you are stuck, try looking at the example in Sec. 9. 3. 3 of PRML.

Question 4 (10 marks)

Table 1: Data Table for Question 4

positive		negative	
x1	x2	x1	x2
10	0	5	10
0	-10	0	5
5	-2	5	5

Compare nearest neighbour with a clustering approach by doing the following.

1. Plot the data points shown in Table 1.
2. Draw the decision boundary that results from nearest neighbour classification (k-nearest neighbour with $k = 1$).
3. Draw the positive sample mean $\mathbf{m_1}$ (pair of numbers) and the negative sample mean $\mathbf{m_2}$. Draw the decision boundary that results from the following classification rule: Label a point as positive if it is closer to $\mathbf{m_1}$ than it is to $\mathbf{m_2}$, and as negative otherwise.

Question 5 (25 marks)

This question is an extension of a question on the previous assignment. To make the question self-contained, I repeat the instructions here. In this question you will use support vector machines for SPAM email[1] detection. The data are in the tarball on the website, in the spam directory.

The directories easy_ham, and spam (which are .tar.gzipped) contain email messages. These have been parsed into feature vectors for you, and stored in the .mat files. Each .mat file contains a single variable V of word counts. V is $n_{messages}$-by-d, where d is the dictionary size. For interest, the dictionary is also provided.

Download libsvm, which has a MATLAB interface on it, from: `http://www.csie.ntu.edu.tw/~cjlin/libsvm/#matlab`

Download the first link, "A simple MATLAB interface." Instructions for installation are available on the website and included in the readme. I also recommend looking at the authors' user guide, "A Practical Guide to Support Vector Classification", which I have posted on the course site for your convenience. This briefly reviews some of the points we've made in class and discusses questions like how to do cross validation with SVMs. There are also some comments on using SVMs for document classification that you may find relevant.

[1]The data come from the SpamAssasin public mail corpus `http://spamassassin.apache.org/publiccorpus/`.

Experiment with different kernels and values for the SVM parameters, using cross-validation on the easy_ham and spam as your training data. Note that libsvm comes with various utility files that you may find useful. I would suggest using accuracy (percentage of correctly classified instances) as your error measure, but you can use any reasonable measure as long as you explain it.

Choose what you think is the best classifier, then run it on the unlabeled data in `test_data.mat`. This file contains a matrix V of word counts, which is 2796-by-1373. Produce an output vector v that is 2796-by-1, with the target value of -1 for spam messages, and 1 for ham (non-spam) messages.

Save the vector v and your login to identify you in a file spamtest.mat:

```
v = ...
name = 'your login';
save('spamtest.mat','v','name');
```

Include in your report a few plots for kernel/parameter settings showing cross-validation results. I would also like to see plots for training data error, at least for what you decided was your best classifier. By training data error I mean the resubstitution error, which results by applying the trained classifier to the original training sets (easy_ham and spam). Since you are carrying out your own experiments, you have some discretion to include what you think are the most interesting findings. Describe the kernels with which you experimented, in your report. State which kernel/parameter values you used for producing v.

Bonus marks and a prize will be given to the student(s) with the best classification performance!

Question 6 (35 marks)

In this question you will implement AdaBoost for discriminating between 4s and other digits. A code skeleton `boost_digits.m` is provided. There are different versions of the AdaBoost algorithm (see `http://en.wikipedia.org/wiki/Adaboost`). To avoid confusion and to help us grade, please use the one in the book, which is also slightly easier to implement.

Download the data `digits.mat` from the course website (in the tarball). This file contains 1000 images from MNIST, of the digits 0-9.

A function `findWeakLearner.m` is provided. This function chooses the best decision stump (feature, threshold, parity) to minimize weighted 0-1 loss.

Each decision stump is returned from `findWeakLearner.m` as d, p, θ. This represents the weak learner:

$$y_m(\boldsymbol{x}) = \begin{cases} 1 & \text{if} \quad p\boldsymbol{x}(d(1), d(2)) > p\theta \\ -1 & \text{otherwise} \end{cases}$$

Here $\boldsymbol{x}(d(1), d(2))$ gives you the grayscale value of the pixel at location d_1, d_2 in image $\boldsymbol{x}$.

In your report, provide the final plot of training error and test error produced using AdaBoost. Also include the visualization of the final classifier produced using `visualizeClassifier.m`

Note that a second data file `digits10000.mat` is provided if you wish to experiment with more data than the 1000 in `digits.mat`.

Submitting Your Assignment

You should create a report with the answers to questions and figures described above in PDF format. Make sure it is clear what is shown in each figure. **DO NOT INCLUDE SOURCE CODE.**

Include the spamtest.mat file and your PDF report in an archive, and submit it.

Submit your assignment using the online assignment submission server at: `https://courses.cs.sfu.ca`.