Assignment 2: Classification

Due Feb 23 at 11:59pm
100 marks total

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)

Question 4.9 in PRML.

Start by writing down the log-likelihood. You then need to make use of a constraint on the π_k, using a Lagrange multiplier.

Question 2 (10 marks)

The use of a logistic regression model is closely related to using the *Naive Bayes Assumption.* The NBA states that the input features $x_1, \ldots, x_m$ are mutually independent given the class label:

$$p(\boldsymbol{x}|\mathcal{C}_i) = \prod_{j=1}^{m} p(x_j|\mathcal{C}_i)$$

for each class label $i = 1, ..k$. Intuitively, the NBA implies a model that considers correlations between the class label and the features but does *not* consider correlations between the features alone. Considering the binary case with $k = 2$, show that the Naive Bayes Assumption leads to a logistic regression model in that the log-odds

$$ln\left(\frac{p(\mathcal{C}_1|\boldsymbol{x})}{p(\mathcal{C}_2|\boldsymbol{x})}\right)$$

are a sum of the form

$$ln\left(\frac{p(\mathcal{C}_1|\boldsymbol{x})}{p(\mathcal{C}_2|\boldsymbol{x})}\right) = t_0 + t_1 + \cdots + t_m$$

where t_0 depends on the class labels only, and each term $t_i, i > 0$ depends only on the class labels and on x_i.

Question 3 (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 4 (40 marks)

In this question you will compare 2 methods for optimization for logistic regression.

1. Download the assignment 2 code and data from the website. Run the script `logistic_regression.m` in the lr directory.

   This code performs gradient descent to find $\boldsymbol{w}$ which maximizes the likelihood (more precisely, minimizes negative log-likelihood).

   Include the final output of Figures 2 and 3 (plot of separator path in slope-intercept space; plot of neg. log likelihood over iterations) in your report.

   Why are these plots oscillating? **Briefly explain why in your report.**

   How might you fix this? **Fix this, and include new plots in your report and an explanation of your fix.**

2. Modify this code to use iterative reweighted least squares (IRLS, Eqn. 4.99). The built-in MATLAB function `diag` is useful for Eqn. 4.98.

   Note that this only takes about 3 lines of code to implement. If you're doing more work, stop, read the textbook, or ask me or Majid for help.

   Include new plots of Figures 2 and 3 using IRLS in your report.

   Yes, it is that fast.

Question 5 (30 marks)

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. If you have problems, please ask in class or e-mail Majid. **I advise installing this early.**

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.

Experiment with different kernels and values for parameter C, 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');
```

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

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!

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 *new* online assignment submission server at: `https://courses.cs.sfu.ca`.