CSE 250B Project 1

You should work in a group of either two or three students for this project. For multiple reasons, working alone or in a larger team is not appropriate. The joint report for your team must be submitted in hard copy at the start of class on Thursday, January 23, 2014.

The goal of this project is for you to understand logistic regression and gradient-based optimization, and practical issues that arise in classifier training. You should implement your own code for training logistic regression models with L_2 regularization. The user should be able to choose between two optimization methods, namely stochastic gradient descent (SGD) and L-BFGS. Implement SGD yourself, but for L-BFGS use a standard library package. See `http://en.wikipedia.org/wiki/Limited-memory_BFGS` for some options. A good implementation in Matlab is Mark Schmidt's minFunc available at `http://www.di.ens.fr/~mschmidt/Software/minFunc.html`. Libraries for L-BFGS are available in other programming languages also. You do not need to understand the mathematical details of the L-BFGS algorithm.

Your code should implement grid search for finding values for hyperparameters. For both SGD and L-BFGS, the strength of regularization is a hyperparameter. For SGD, one or more other hyperparameters define the learning rate. In general, setting the learning rate well is quite difficult; this is a major drawback of SGD. For relevant advice, see `http://leon.bottou.org/projects/sgd`. For general advice on setting hyperparameters, see Section 3 of *A Practical Guide to Support Vector Classification* by Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin, available at `http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf`.

It is vital to be confident that basic algorithms are implemented correctly. The report must convince the reader that this is true. To verify that derivatives are correct, use a function such as the one available at `http://people.csail.mit.edu/jrennie/matlab/checkgrad2.m`. Be sure you understand this code before using it.

After your code is working, apply it to the dataset named Gender Recognition [DCT] that is available at `http://mlcomp.org/datasets/1571`. You will need to register on this web site in order to download the data. Be sure to understand the format of the data. The web site says that the svmlight-linear software achieves an error rate of 0.084. Can you do better?

For this project and for later ones, the only deliverable is a well-written joint report. As is the case for a research paper, your job is to inform and convince the reader fully.

Please bring a printed copy to the start of class on January 23. To quote Sanjoy Dasgupta, "Discuss your results in precise and lucid prose. Content is king, but looks matter too!" The report should be self-contained and complete, but concise. There should be separate sections for at least the following: (i) introduction, (ii) design and analysis of algorithms, (iii) design of experiments, (iv) results of experiments, and (v) findings and lessons learned. The report should be typeset with LaTeX and have appropriate equations, tables, figures, citations, and references. Explain in reproducible detail your algorithms, especially any refinements that you invent, and the design of your experiments. These explanations should be separate from actual experimental results. In your report, analyze experimentally and theoretically the big-O time and space complexity of your implementations. Explain any non-trivial obstacles that you did or did not overcome in trying to achieve good time complexity, and provide brief empirical timing results.