CSC2515 – Final Project

Due: December 15, 2006 by noon
email (*.pdf or .ps only no MS-Word*) to csc2515@cs Worth: 33%

1 General Guidelines

- The idea of the final project is to give you some experience trying to do a piece of original research in machine learning and writing up your results in a paper style format.
- What we expect to see is *a simple but original idea/task* that you describe clearly, relate to existing work, implement and test on a small scale problem.
- To do this you will need to write code, run it on some data, make some figures, read a few background papers, collect some references, and write a few pages describing your task, the algorithm(s) you used and the results you obtained.
- As a rough rule of thumb, spend about as much time doing the work as you would have studying for an exam and a few hours writing it up after that (instead of actually writing an exam).
- Projects are to be done **individually**. Many students chose to use some aspect of their graduate research as a project, which is wonderful, but whatever you submit for the course project must be entirely your own work (even if the larger project involves other people).

2 Specific Requirements

- Your project must implement one or more machine learning algorithms and apply them to some data. Your project may be a comparison of several existing algorithms, or it may propose a new algorithm in which case it still must compare it to at least one other approach.
- You can either pick a project of your own design (preferred), or you can choose from a list of suggested projects which will be posted partway through the term. Regardless of which way you select a project, you cannot use the excuse that you got a "bad project" to explain doing a poor job on it. So select wisely!
- Your submission must include at least two figures which graphically illustrate quantitative aspects of your results, such as training/testing error curves, learned parameters, algorithm outputs, input data sorted by results in some way, etc.
- Your submission must include at least 5 references to previous published papers or book sections. Please include page numbers for all references to indicate that you actually *saw* the paper you are referencing even if you didn't read it very carefully!
- Your submission should follow the generally accepted style of paper writing: include an introduction section to motivate your problem and algorithm, a section describing your approach and how it compares to previous work, a section outlining the experiments you ran and the results you obtained, and a short conclusions section to sum up what you discovered.
- Your submission must be prepared in the NIPS 2006 paper style (using Latex is encouraged but not required), and must be no longer than 5 pages in length (which is less than the NIPS limit of 8 pages), including all figures, tables, references, etc. Do not hand in any code of any kind.

3 Marking Scheme

- The projects will be marked out of 11, with each point being worth 3% of your grade.
- The following criteria will be taken into account roughly equally when marking:
  1. Clarity/Relevance of problem statement and description of approach.
  2. Discussion of relationship to previous work and references.
  3. Design and execution of experiments.
  4. Figures/Tables/Writing: easily readable, properly labeled, informative.

4 Friendly Advice (for this assignment and for doing reseach in general)

- Be selective! Don't chose a project that has nothing to do with machine learning. Don't investigate an algorithm that is clearly stupid, doomed to failure or un-implementable. Don't attack a problem that is irrelevant, ill-defined or unsolvable.
- Be honest! You are not being marked on how *good* the results are. It doesn't matter one bit if your method is better or worse than the ones you compare to. What matters is that you try something sensible, *clearly describe* the problem, your method, what you did, and what the results were.
- Be modest! Don't pick a project that is way to hard. Usually, if you select the simplest thing you can think of to try, and do it carefully, it will take much longer than you think.
- Be careful! Don't do foolish things like test on your training data, set parameters by cheating, compare unfairly against other methods, include plots with unlabeled axes, use undefined symbols in equations, etc. Do sensible crosschecks like running your algorithms several times, leaving out small parts of your data, adding a few noisy points, etc. to make sure everything still works reasonably well. Make lots of pictures along the way.
- Learn! The point of the project is to give you a chance to "test drive" the process of writing a paper, which many of you have never done, in a low-stress setting, away from the pressures of your thesis and conference deadlines. Consider this an opportunity to learn how to write code to run large experiments, make nice figures, layout readable equations, describe your work concisely to a smart but uninitiated reader, etc.
- Have fun! If you pick something you think is cool, that will make getting it to work less painful and writing up your results less boring.