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 an idea/task that you describe clearly, relate to existing work, implement and test on a dataset. 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 one week’s worth of work (spread out over a longer time to allow the computers to do some work in the interim!), and about a day writing it up after that.

Ideally, we prefer group sizes of 2-3 people. Exceptions are possible with the instructor’s permission. Please feel free to contact either Richard, Kaustav, Wenjie, or Yujia about project ideas.

The projects will be presented during the last day of class, April 5th. Then the write-ups are due April 20th.

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 you still must compare it to at least one other approach. You can either pick a project of your own design, or you can choose from the set of pre-defined projects described below. 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 3 references to previous published papers or book sections. 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 paper format (using Latex is encouraged but not required), and must be no longer than 6 pages (10 for pair projects) including the figures and tables and references. Do not hand in any code of any kind.

3 Project Proposal

You must turn in a brief project proposal (1-2 paragraphs). Your project proposal should either say which of the pre-defined projects you plan to pursue, or describe the idea behind your self-defined project. You should also briefly describe software you will need to write, and papers (2-3) you plan to read. Please also say who else is in your group. Include your email address on your proposal. We need this to contact you and arrange meetings to discuss your proposal.

The project proposal is due in class on February 23.
4 Marking Scheme

The projects will be marked out of 30, with each point being worth 1% of your grade. The following criteria will be taken into account when marking:

- Clarity/Relevance of problem statement and description of approach.
- Discussion of relationship to previous work and references.
- Design and execution of experiments.
- Figures/Tables/Writing: easily readable, properly labeled, informative.

5 Friendly Advice

- Be selective! Don’t choose a project that has nothing to do with machine learning. Don’t investigate an algorithm that is clearly doomed to failure or attack a problem that is ill-defined or unsolvable.

- Be honest! You are not being marked on how good the results are. It doesn’t matter 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 too 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 cross-checks 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 report less boring.