CMPS 142/242 Project Guidelines, Spring 2011

This document describes the project requirements, the project proposal, the progress report, and the final project writeup.

Types of projects

The 141/242 course projects can take a variety of forms. Perhaps the most typical project is experimentation with a challenging labeled dataset. This may be a dataset from a (current or past) data-mining, information retrieval, or machine learning contest, or a dataset related to your research or particular interests. In the past, students have worked on data ranging from sports salaries, text reviews of movies, and data-base schemata. It is most important that you have sufficient labeled data – hand labeling your own data is not recommended. The most common difficulties arise from data arriving too late or the required labeling taking too much time or effort. Next most common is computation time – running the algorithm on the full dataset takes weeks rather than days. A typical project of this type would include a main learning method that appeared to fit the learning problem well. The algorithm itself could be either implemented by the students or be from a machine learning library. Comparison with a standard baseline method (perhaps naive Bayes or logistic regression) or previously published work is used to evaluate the effectiveness of the learning and to calibrate the difficulty of the problem. Things that add to the difficulty of this kind of project include:

- an interesting dataset/learning problem
- feature selection/creation
- the sophistication of the main algorithm
- the need for parameter tuning (e.g. to avoid over-fitting)

An analysis of why the algorithm did (or did not) perform well is expected for this kind of project.

A similar kind of project is a comparative study of multiple (probably 3 or 4) learning methods across several data sets. The data sets can be standard ones like those in the UCI libraries. Typically Weka another package is used for the learning implementations, and the project compares of ease-of-use, generalization effectiveness, and running time of the various algorithms. Strive for insight such as why particular algorithms perform better on certain datasets.

A third kind of project is a literature survey/review. The survey should have a focus like learning of EEG time-series data or learning with random forests. The topic should be advanced enough so that I learn something from this kind of report.

Finally, for 142 students, I am also willing to accept "tutorial" projects that provide infrastructure and documentation to assist future students with their projects. One example of this might be a comparison of Support Vector Machine packages focussing on which easiest to use together with a local guide describing how to use it.

Students may work in small groups (usually a max of 3 or 4 students), but group projects will be held to a higher standard than individual projects. It is possible for projects in this course to overlap with projects in other courses/independent studies, but such overlap must be disclosed to all relevant faculty members and significant overlap can raise expectations.
Project Proposals - due Monday April 18

The goal of the project proposal is to ensure that the proposed projects have sufficient depth while remaining feasible.

The project proposal should be a short (2-3 paragraphs to 1 page) description of a project idea. It should:

1. describe the problem you are applying learning to,
2. the data you intend on learning from,
3. a first idea as to the methods you will use,
4. and how you will evaluate the success of learning.

Students may include one or two alternative project ideas if they wish.

Progress Report - due Friday May 13th

The progress report is a less formal (and much briefer, perhaps 3–4 pages of text) version of the final report. It should have (at least) three sections: an introduction (like that for the final report, but with a complete problem statement and you probably won’t have results to report), a methodology/plans section, and a progress/problems section.

The methodology/plans section should describe:

1. the data you are using and how it was obtained
2. the pre-processing and/or feature extraction you have done (or are planning to do)
3. the learning methods and tools you are using (or implementing)
4. what parameters of the learning algorithm you will need to tune
5. what experiments you plan, and how you will evaluate the results

The progress/problems section should list the progress you have made as well as any significant problems/difficulties you have encountered or can visualize down the road. One purpose of the progress report is to get you to think about any potential difficulties while there is still time to work around them.

Please attach a copy of your proposal with my comments to your progress report.

Final Project Reports – due Monday June 6th at 4pm

Project reports must be typeset in 12pt font. I would like you turn in a hardcopy of your report to me (in my office) as well as e-mailing me a soft-copy (probably .pdf) file. See me if you have code or unusual data that you would like to make available to future classes. You can leave your project reports in the bin outside my office door if I am not there, but please send me an e-mail indicating that you have done so.

The text in your report must be in your own words. Quoted text must be set off by quotes (“””) and the source clearly attributed, even if the text is as small as a single phrase. Alternatively,
quoted material can be acknowledged and then displayed in an indented paragraph. For example, the following is from *How To Handle Quotes, MLA-Style*¹:

It is important to know how to effectively use quotations in your papers. The following are examples of how to properly use quotations. Note that every quotation – whether a direct quotation that exactly copies someone else’s work word-for-word OR an indirect quotation that puts someone else’s work into your own words – needs to be documented. That means that you give credit to the source. FDR uses the MLA system for this. Keep in mind that if you use someone else’s idea, even if you don’t directly or indirectly quote it, you must still give that person credit. You do that in the same way that you handle quotations.

If you use someone else’s figures or tables the appropriate attribution must appear in the caption as well as in the text where you discuss the figure/table.

In the past, project reports have been about 7 to 15 pages long (not counting appendices and large tables, which can add quite a bit of length). Please do not turn in large sections of code listings or massive tables of raw data (although some information on the data is important, and a table indicating what a few typical examples look like would be OK, especially if you have an unusual data set). The report should be easy to read, if it is hard to tell what you are trying to say, then it will be hard to give you a good grade. Every figure or table in the report body should be discussed in the report body. If you would like to present additional experiments that are not evaluated in the body of your report, include them as an appendix (additional data or experimental results can also be included as an appendix).

The title of your report should indicate the learning problem it addresses. Your report should have an abstract as well as introduction/problem description, related work, Data, methods used, results, and conclusion sections. It must also contain a bibliography. I am flexible on the exact section breakdown, you may add or merge sections if it makes writing/understanding the report easier. Readability is important, so be sure to define your terms *before* using them and present things in a logical order. Target the level of your report so that it can be understood by a typical CS senior – i.e. limit your use of jargon and provide an brief overview of those concepts that a CS senior is unlikely to be familiar with.

Your report should start with a short 1-paragraph abstract that mentions the problem you attacked, your main methodology, and your results (perhaps 3-5 sentences total).

The introduction should contain a description of your problem at a level that typical upper division CS students should be able to understand. Any area-specific jargon should be explained/defined when first used. The introduction can also give an overview of your results, how you obtained your data, etc. However this additional information is likely to appear elsewhere, and so should be just summarized in the introduction to avoid too much redundancy. If your particular problem is technical or difficult to describe precisely then you might give just an overview of it in the introduction and use a different section to describe the detailed questions you attempt to answer. The introduction should provide an overview of what the problem is, why it is interesting/important (why did you choose it) how you attacked the problem, and an idea of the success and/or failure of your methods.

The related work section should contain a survey of relevant previous work for your problem and possibly the methods you used. This is sometimes a good place to clearly spell out what you

did for the course as opposed to what was done by others or outside of the course. Feel free to cite textbooks or articles etc. for descriptions of algorithms. However, the best related work sections are not just lists of references, but evaluate and put into context the previous contributions, as well as relating them to the current work. Graduate student projects are expected to have better/more extensive related work sections.

The methodology section should describe the details of your experiments. It should start with a description of the data, including the data source, the number of examples, the features and labels, and what preprocessing was done. If cross validation is done or a held-out test set was used, then that should be described as well. Describe the learning techniques used and what software packages (such as Weka or SVM light) you used. Either here or in the conclusions you could indicate any difficulties or problems using packages and how they were resolved. There will be enough information here so that another student could reproduce your results. Although I am not interested in a printout of any code you wrote, you could include a link or pointer to where it could be obtained (as well as your datasets). You should also explain here (if not earlier in your report) why you picked the methods you did.

The experimental results section should describe what happened. Is it what was expected? What were the surprises/anomalies? In retrospect, why do you think the results come out the way they did? How do your results compare with others? Ideally, each experiment is a question and the results provide an answer. Tables and graphs are appropriate ways to summarize information. If you are doing many experiments or varying many parameters, a good way of structuring your presentation is to have a baseline situation and compare each of the individual experiments to the baseline.

The conclusions section should include a short self-evaluation of your project (what went right and what went wrong) together with a summary of what was learned from the experiments and what you yourself learned and a recap of what you accomplished. If there are other things you would have liked to try but didn’t get around to, you can include future work in the conclusions section (or even make further work its own section).

You should acknowledge any help you have been given on the project and anything else from others that made the project possible (such as data or machinery/code).

The bibliography should contain relevant publications (articles, books, web pages, etc.) and other resources that you read or used in conjunction with your project. Part of the project is to identify the relevant literature and read in more detail about some aspect of machine learning.