CMPS 142 Project Guidelines, Spring 2015

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

Types of projects

The 142 course projects can take a variety of forms. Typical projects are experimentation with a challenging and interesting **labeled** dataset. **I expect that most groups will perform a study based on the KaggleTitanic survivor dataset or the first year success dataset posted to the class web page.** Alternatives are a dataset from a (current or past) data-mining, information retrieval, or machine learning contest, or a particularly good dataset related to your research or 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, with or without the help of others, is strongly discouraged**. 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 is to first experiment with a couple of learning methods on the data set, and then focus on trying to improve the best one or two methods. The algorithm itself could be either implemented by the students or be from a machine learning library such as Weka. Be sure to include comparison with a standard baseline method (naive Bayes and logistic regression are often good choices). It is also good to compare your results with other work if available. This helps 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)
- using techniques/ideas from the current literature.

An analysis of what good performance should be and why the algorithm did (or did not) perform well is expected for this kind of project.

Students are expected to work in small groups (2 to 4 students). 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 - optional - due Tuesday May 12

Project proposals are only for those using datasets other than the Titanic or First Year Performance data.

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 and data set to be used. It should have a descriptive title like "Learning the value of baseball players" (rather than a generic "project proposal") followed by a listing of the project members in alphabetical order along with their email addresses and class (142/242). The rest of the proposal should describe:

1. the problem you are applying learning to,
2. the data you intend on learning from (number of examples, kinds of features, labels). Indicate if it is already in your possession, and if not how and when you will obtain it.
3. a first idea as to the methods you will use,
4. and how you will evaluate the success of learning.

Be sure to define any application-area jargon you use, and recall that I may not be familiar with your area of research. Students may include an alternative project idea if they wish, but clearly label where one project idea ends and the other begins. If you are doing a project that overlaps with other coursework, an independent study, or a research assistantship, this needs to be disclosed at this point (along with the other supervising faculty member).

For those of you doing the suggested Kaggle competition, the proposal can be very short.

If you would like to provide more background and/or details, structure your proposal so that an executive summary on the first page answers the four questions above, and the elaboration is contained on the following pages.

In the proposal, you can reference papers, books, or web-pages in footnotes or parentheses with enough information to easily locate the source rather than creating a formal bibliography. For example, you could say "using confidence-rated AdaBoost as described by Shapire and Singer in "Improved Boosting Algorithms Using Confidence-rated Predictions", Machine Learning, 1999"

Progress Report - due Tuesday May 26th

The progress report is a less formal (and much briefer, perhaps 2 pages of text) version of the final report. It should have (at least) three sections: an introduction (like that for the final report, with a complete problem statement and background although you may not have many results to report), a methodology/plans section, and a progress/problems section. The progress report should include a formal bibliography with complete references to the cited work.

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. the parameters of the learning algorithm you have tuned or need to tune
5. what experiments you plan, and how you will evaluate the results
6. an estimate of how much computer time your experiments require

The progess/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.

Final Project Reports – due June 11 at 9 am

Project reports must be typeset **in 12pt font**. Although not mandatory, I prefer them double-sided with page numbers.

Unless you have made other arrangements, you need to 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.* (This is also true for the proposal and progress report.) 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 elses work word-for-word OR an indirect quotation that puts someone elses 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 elses idea, even if you dont 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 both in the caption and 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 can be helpful, 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 an 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

[1] Available at `http://www.amersol.edu.pe/hs/english/howtohandlequotes.asp`

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 goal of the abstract in a technical paper is to allow researchers to decide if the paper is on an interesting topic to them.

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 must 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. If there has been previous work on your data set, this is the place to say who tried what and how successful they were. 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. If your course project is part of an on-going research effort, then the related work section should describe what has gone on before, delineate what is the part done for this course, and indicate how it fits into the larger project.

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. Questions like “Did you use cross validation or a held-out test set?” should be answered here. 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 or default situation and compare each of the variants to this 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). If you used software other than your own, I would like an evaluation of it in the conclusions (or perhaps another section if the evaluation part becomes too long). The evaluation should indicate how easy the software was to use and any work-arounds you had to make. These evaluations will help let me know if I should be recommending the software to future Machine Learning students.

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. See ICML or NIPS papers for examples of bibliography styles.

Oral Presentation

Groups will make an oral presentation of their methods and results on Thursday June 11 from 9-11:30 in the classroom (instead of a final exam during exam week). Each group should plan on a 10-12 minute presentation, the exact amount of time will depend on the number of presenting groups. All students are expected to attend all of the presentations.