CS 472/572: Machine Learning
Final Project Instructions

1 Description

Instead of a final exam, CS 472/572 has a final project which counts for 30% of the grade. It is intended to provide realistic experience in using or researching machine learning.

There are several different ways to do this project:

- **Kaggle Competition.** (Recommended.) Pick one of the current contests on kaggle.com, download the data, and try to develop the best model you can. Kaggle provides training data, an initial set of features, a testing framework, and a leaderboard to compare your results to many other teams. Therefore, you can focus all of your time on developing better features, expanding the training data, selecting appropriate models and tuning them, and building an ensemble from a collection of models.
- **New Domain.** Identify an interesting problem, collect data, design a feature representation, apply several machine learning algorithms (being careful not to train on test data), and analyze the results.
- **Algorithm Development** Develop and evaluate a new machine learning algorithm, representation, regularizer, optimization method, etc. It is hard to do this well, since most of the easy and obvious ideas have been tried already. Therefore, this kind of project is not recommended for most students.

If you want help picking a project, feel free to ask me questions. It's best if you already have some idea of what you want to do.

2 Methods and Results

Your project must contain theoretical or empirical results. Coming up with new theoretical results of interest is difficult, so I expect that most of you will only present empirical results.

For an application paper, you should evaluate and justify the choices you made. Here are some questions to think about:

- How did you select your data? How much data? What cleaning or processing did you do to the data? (For some problems, you may need to be creative about integrating data from multiple sources, or making do with noisy labels.)
- How did you formulate your problem as a machine learning problem? Is this problem best posed as classification? Regression? Clustering? Ranking? Probability estimation?

- What features did you select and why?
- What algorithms did you use? (You should almost always use more than one, in order to have a comparison.)
- What baselines did you use (if proposing a novel algorithm or feature set or problem formulation)?
- How did you set up the training/tuning/testing data? Did you do cross-validation? How did you tune the parameters?
- How do you choose to measure performance? Accuracy? Learning curves? ROC curves? Confusion matrix? Precision/recall/F1 measure? Running time?
- Which algorithm performs best? Can you determine why that algorithm works best?

You do not need to implement everything yourself. scikit-learn and weka are popular open-source toolkits that already include many common classifiers. Other popular open-source tools include vowpal wabbit (especially for large, high-dimensional data), LIBLINEAR (for linear models), LIBSVM (for SVMs), and Keras, Pytorch, Tensorflow (for neural networks).

Please do follow the scientific method. Develop appropriate experiments to validate or refute your hypotheses, as well as to provide more insight. For example, which feature representation worked best? Which classifiers or combinations of classifiers worked best? Why do you think this is? What evidence do you have for this explanation?

An accuracy with no explanation is not interesting. An explanation of how you obtained that accuracy, what worked and what didn't, and what you learned is more interesting. Please include some quantitative measures, in tables, charts, and graphs.

This does not need to be publishable research, but it should demonstrate that you understand how to apply machine learning to a real problem (for an application paper) or how to develop and evaluate novel algorithms (for an algorithms paper).

Negative results are acceptable. If you get a negative result, explore what happened and why. Not enough data? Overfitting? Bad features? Noisy labels? Different distribution at test time? Explore what led to the poor results and try to determine if that could be overcome.

You can also try ambitious projects that might be hard to complete during this course, but please talk to the instructor about this in advanced. The ambitious projects will be evaluated based on the status of the projects at the time of the final presentations. A detailed plan for the project, where you are at the presentation time and how you plan to finish it should be presented in the final presentations and reports.

3 Writing

All papers are expected to be clearly written with a good structure. I will hold graduate students to a higher standard of formal, technical writing and analysis of experimental results. This project should be doable by a single person, so I expect that larger groups will have correspondingly more experiments and more analysis.

Many machine learning papers use a structure similar to the following:

1. Abstract: Summarize the entire paper (including results) in 50-250 words.
2. Introduction: Identify the problem you're trying to solve, describe why it's important, and outline the key method or strategy that you will use to solve it.
3. Background: Describe the technologies or ideas that you will build on in your method. For an application paper, this could simply be a detailed description of the problem you're trying to solve. For an algorithm paper, this could be the machine learning methods that you're extending.
4. Methods: Describe your approach to solving the problem. This should contain your key contributions.
5. Experiments: Evaluate your approach experimentally. Describe your methods in enough detail that another researcher could replicate them. How well does your method work? Does your method outperform reasonable baselines? How does your method compare to simplified versions of your method? What kinds of errors remain? What interesting things do you learn from your experiments? Tables of results are useful, but charts and figures are often better. This can also be integrated with the methods, so that each aspect of the model is evaluated as it is introduced (e.g., feature selection, classifier selection, ensemble construction).
6. Conclusion: Summarize your contributions and discuss future work (50-500 words).
7. References: Works that you cite in the body of your paper. You may use any standard citation style as long as it is consistent.

I recommend that you use a structure similar to this one, unless you have a good reason.

I do not require perfect English, but I greatly appreciate clear writing. Your methods should be described clearly enough to replicate your results. Your conclusions should be supported by evidence. Your arguments should follow logically. Each paragraph should discuss a single idea. If you're having trouble, there is writing tutoring available on campus for all students.

Learning to write a good technical paper is an extremely valuable skill in both graduate school and industry. Writing well is very difficult, even for experienced writers, but it does get easier with practice.

Students are encouraged to use the NAACL template to write the report for this class: `http://naacl.org/naacl-pubs`.