CPSC 440/540 Machine Learning (Jan-Apr 2023)
Project Proposal – due Friday March 24 at **11:59pm**

As discussed in the syllabus, we are using the following marking scheme for the course:

1. CPSC 440: 50% assignments and 50% for the best among {final exam, research project, sample lecture/assignment}.
2. CPSC 540: 50% assignments, 25% for the best among {final exam, research project, sample lecture/assignment}, and 25% for the second best among those three categories.

If you choose to do the research project and/or the lecture/assignment project, 10% of the final grade for the corresponding project is for this proposal. That is, if you're a 440 student doing only the research project, the research project proposal will count for 5% of your final course grade, and the research project writeup will be 45% of your final grade. If you're a 540 student doing both the research project and the sample lecture/assignment, you'll do two proposals, each worth 2.5% of your final mark.

If you're a 440 student and confident you want to only do the final, you don't need to hand in a proposal.

Projects are **strongly encouraged** to be done in groups of 2-3. If you really want to do it alone, that's allowed, but **expectations will not be decreased** so it'll be harder, and it will probably be a worse learning experience for you as well – ML research is almost always collaborative.

These proposals are short and lightly graded: if you put some effort in, you'll get full proposal marks. The point is so that (a) you form your project groups now, (b) you think about what you want to do, and (c) we check that the scope and topic of the project is suitable for the course.

If you're doing both projects, the groups don't have to be the same.

It's also okay to do two projects in the same area, e.g. a lecture introducing an ML method and a research project focusing on applying or extending that method. Similarly, it's okay with me for your project to overlap with a project in another simultaneous course, as long as you check with the other instructor (though do an appropriate writeup for each course). Having some overlap with your ongoing graduate research is also okay, but make sure this is a relatively discrete project with a clear scope.

The proposal is not necessarily "binding" – research projects very often shift from the idea you started with, and it's even okay to do something totally unrelated to what you proposed. If so, though, you probably want to check in with me or a TA that the new project is in scope: if you hand in a final project we think is very out of scope for the course without having checked with us, you'll get a bad grade on it.

Submit each proposal separately on Gradescope. The proposal can be done in any format you'd like; default LaTeX style with `\usepackage{fullpage}` is fine.

1 Research Project

The proposal should be a maximum of 2 pages; it's okay to be shorter if you can describe the plan concisely. The proposal should be written for the instructor and the TAs, so you don't need to introduce ML background that's covered in the course or that you would reasonably expect the TAs (graduate students working in ML) to know, but you should introduce any required background for non-ML topics.

There is quite a bit of flexibility in terms of the type of project you can do, since there are many ways that people can make valuable contributions to research. The final deliverable will be a written report consisting of at most 6 pages (in a LaTeX format to be provided), with unlimited additional space for references and possible appendices (which, as with NeurIPS reviewers, you shouldn't count on the graders reading). That report should emphasize one particular "contribution": what has doing this project added to the world?

The three main questions your project proposal needs to answer are:

1. What problem you are focusing on?
2. What do you plan to do?
3. What will the "contribution" be?

For the course project, negative results are acceptable (and often unavoidable). In that case, the paper should probably include something like "here's why we thought this thing we tried would work in this setting, here's us convincing you that it didn't, and here's our best understanding at why we think it failed."

Here are some standard project "templates" that you might want to follow:

- **Application bake-off**: you pick a specific application (from your research, personal interests, or maybe from Kaggle) or a small number of related applications, and try out a bunch of techniques (e.g., random forests vs. logistic regression vs. generative models). In this case, the contribution could be showing that some methods work better than others for this specific application, and hopefully some idea why – or your contribution could be that everything works equally well/badly.
- **New application**: you pick an application where where people aren't using ML, and you test out whether ML methods are effective for the task. In this case, the contribution would be knowing whether ML is suitable for the task (and perhaps how to prepare the data and constructed features).
- **Scaling up**: you pick a specific machine learning technique, and you try to figure out how to make it run faster or on larger datasets. (For example: how do we apply kernel methods when n is very large?) Your improvements might be a new approximation, a distributed version, a smarter implementation, or so on. In this case, the contribution would be the new technique and an evaluation of its performance, or could be a comparison of different ways to address the problem.
- **Improving performance**: you pick a specific machine learning technique, and try to extend it in some way to improve its performance (for example, how can we efficiently use use non-linearities within graphical models). In this case, the contribution would be the new technique and an evaluation of its performance.
- **Generalization to new setting**: you pick a specific machine learning technique, and try to extend it to a new setting (for example, making a graphical model version of random forests). In this case, the contribution could be the new technique and an evaluation of its performance, or could be a comparison of different ways to address the problem.
- **Perspective paper**: you pick a specific topic in ML, read a large number of papers on the topic, then write a report summarizing what has been done on the topic and what are the most promising directions of future work. In this case, the contribution would be your summary of the relationships between the existing works, and your insights about where the field is going.

- **Coding project**: you pick a specific method or set of methods (like independent component analysis), and build an implementation of them. In this case, the contribution could be the implementation itself or a comparison of different ways to solve the problem.
- A **reproducibility report** of a recent paper, as in the 2022 Reproducibility Challenge: we missed the deadline for the official challenge, but you can draw inspiration from the challenge and past submissions there. This is pretty similar to the "coding project," but with slightly different aims.
- **Theory**: you pick a theoretical topic (like the variance of cross-validation or the convergence of stochastic gradient in non-smooth and non-convex setting), read what has been done about it, and try to prove a new result (usually by relaxing existing assumptions or adding new assumptions). The contribution could be a new analysis of an existing method, or why some approaches to analyzing the method will not work.

The above are just suggestions, and many projects will mix several of these templates together, but if you are having trouble getting going then these are all tried-and-true ways to do a course project. Also note that the above includes topics not covered in the course (like random forests): there is flexibility in the topic, it should be closely related to ML.

2 Sample Lecture and Assignment

Throughout CPSC 340 and 440/540, we try to introduce you to a variety of fundamental concepts that we think will stand the test of time, and to give you an idea of what methods people are currently finding useful in a variety of settings. But unfortunately, ML is a huge field, and there just isn't time to cover every relevant topic in the time we have. For the sample lecture/assignment, you will prepare a lecture and an assignment on an ML topic that we do not cover in the course: preparing material is one of the best ways to learn new things, and this also helps us decide what topics to include in future offerings of the course.

(If you do this project type, you're giving the instructors permission to potentially build off your materials for a future iteration of the course. Don't expect it, though!)

If your lecture is about a specific (category of) ML models, it should touch on *most* of the following topics, as in this course:

1. **Motivating problem**: introduce an application that motivates the model.
2. **Model definition**: how is it defined and what are the parameters (and their sizes)?
3. **General framework and other applications**: is this solving an abstract problem, and where else would this model be useful?
4. **Inference**: how do you do things like sampling, decoding, marginalization, conditioning, and test-set predictions/evaluations? (Theoretically and with code.)
5. **MLE**: how do you compute the MLE parameters? (Theoretically and with code.)
6. **MAP**: how do you introduce a prior and compute the MAP parameters?
7. **Bayes**: how do you make Bayesian predictions with the model?
8. **Multivarate** (for one-dimensional distributions): is there a multi-variable version of the model?
9. **MNIST**: what do MNIST samples look like if you use it as a density estimator?
10. **Supervised**: how do we add features to the model and use within a generative or a discriminative classifier?
11. **Deep**: how do we add layers of hidden layers to learn features?

The particular deliverables due with this assignment are:

1. Specify which model you will focus on.
2. Give a 1-sentence-maximum description of how you might cover each of the topics above in the model. Not all topics make sense for all models, so it's fine to say that you won't cover some of them.
3. Give a short outline for what the assignment related to the model will cover.

It's also okay to do a lecture on a topic that isn't "here's a class of ML models." In that case, replace the second bullet point above with giving a rough outline of what sub-parts of the topic you'll address, comparable in detail to the points above. It's of course fine if the final lecture doesn't cover all of these points or do it in the same order or anything – but you should describe at a very high level what kind of sub-topics you'd like to cover in the course of one 80-minute lecture, and why these would give a good overview of the broad topic for a course like 440/540.

As a note: although we're not going to enforce this in any way, I'd like to encourage you to consider *not* doing a topic in which you're already an expert (e.g. are actively researching, or have taken a graduate course on): part of the point of this project is to learn something new. You should still get something out of distilling down your knowledge, though, so it's okay if you do know a topic pretty well already.

Also: do not simply find someone else's lectures or videos and essentially just reproduce them. Reading a textbook is good (many of the topics below are covered at least somewhat in Kevin Murphy's PML series), but ideally you'd find some other sources to pull from as well, even if it's just reading some of the papers he cites.

Lecture Topic Suggestions

The following are some topics that I plan to cover later in the course; please avoid choosing these topics.

- Exponential families and multivariate Laplace/student distributions.
- Rejection sampling and importance sampling.
- Markov chains and MCMC.
- Directed and undirected graphical models ("Bayesian networks" and "Markov random fields"), conditional random fields.
- Latent Dirichlet allocation and variational inference.
- Mixture models and hidden Markov models.
- Boltzmann machines and deep belief networks.
- Variational autoencoders.
- Diffusion models (basic setup).

If you'd like to do something based on recent variations on diffusion models or advanced VAEs, please talk to me (after class or on Piazza) – I haven't prepared those materials yet and it'll be right at the end of the course, so this'll be a little complicated to orchestrate, but since it's such a hot topic at the moment I'd like to allow you to still do it.

Below is a set of topics that you might see in other ML courses that we will probably not cover, as well as some related keywords:

- Sequential Monte Carlo.
- Non-parameteric Bayes.
- Online learning and bandits.
- Reinforcement learning.
- Privacy and security.
- Algorithm fairness.
- Independent component analysis.
- Disentanglement.
- Scaling up Gaussian processes.
- Max-margin Markov networks and structured SVMs.
- Reinforcement learning.
- Meta-learning.
- Active learning.
- Causality.
- Parallel/distributed/federated training.

- GANs.
- Normalizing flows.
- Learning theory.
- Manifold learning.
- Neural ODEs.
- Mutual information estimators.
- Probabilistic context-free grammars.
- Probabilistic programming.
- Graph neural networks.
- Bayesian neural networks.
- Sub-modularity.
- Spectral methods.
- Automatic hyper-parameter tuning.

Some of these topics are too big for one lecture (such as reinforcement learning), and one of the main purposes of doing this proposal is to help you choose the appropriate scope.