COMPSCI 689
Lecture 26: Project Requirements Review and Course Wrap-Up

Benjamin M. Marlin

College of Information and Computer Sciences
University of Massachusetts Amherst

Slides by Benjamin M. Marlin (marlin@cs.umass.edu).

Implementation Projects

- Should select a recent and non-trivial machine learning model and re-implement existing learning, prediction, and/or inference algorithms in a novel computational context.
- For example, on a smart phone, wearable device, embedded microcontroller, within a browser, in a multi-core/parallel/distributed computing environment, or in a language where existing implementations do not exist).

Algorithms Projects

- Algorithms projects should select an existing machine learning model and investigate new learning, prediction, and/or inference algorithms for it based on different optimization or approximation methods.
- The proposed algorithms should be well-motivated according to some criteria (i.e., potential for reduced computational complexity, improved speed-accuracy trade-offs, etc.).

Modeling Projects

- Modeling projects should start from an existing machine learning model and propose extensions or modifications that are well-motivated by the needs of a particular application.
- For example, dealing with different types of outputs, handling missing data, better representing latent structures, improved regularization, etc.
- Projects of this type will also need to develop learning, prediction, and/or inference algorithms and are better suited for advanced students or teams.

Applications Projects

- Do not meet the project requirements.

Specific Requirements: Title, Authors, Abstract

- Title: Select an informative title for your project.
- Author(s): List the names of all group members (your name if working alone).
- Abstract (100-200 words): A brief summary of your problem, approach, and results. A clear statement of what is *novel* in your project.
- Note 1: The term *problem* here referes to the *machine learning* implementation, algorithm or modeling problem you are solving.
- Note 2: Your project may be *motivated* by the requirements of a particular application, but your primary contribution in this report must be your novel implementation, algorithm or modeling work and you report needs to be framed to reflect this.

Specific Requirements: Introduction

- An introduction describing the problem your project is solving, a discussion of the motivation for selecting this problem, a brief statement about what is novel in your work relative to the related literature, and a summary of your methodology, experiments, and results.
- As for the abstract, the *problem* must be the *machine learning* problem you are solving.
- Implementation projects should be motivated by the *need* for an implementation in a given language or on a given platform.
- Algorithms projects should be motivated by the *need* to achieve better perfromance according to some criteria (speed, accuracy, speed-accuracy tradeoff, privacy, fairness, etc.).

Specific Requirements: Introduction

- Modeling projects should be motivated by the modeling *needs* of a specific application or set of applications.
- Recommended page length: 0.5-1 pages.

Specific Requirements: Related Work

- A related work section describing **at least 5** pieces of prior research from the **primary machine learning literature** related to the problem your project addresses.
- You can cite additional work related to the application you are addressing, but this does not count against the above requirement unless that work is published in the primary ML literature.
- Your related work selection should include a mix of both well-cited and more recent articles when possible. Do your best to identify the current state-of-the-art methods related to your work.
- Since your project is focused on methodological contributions within machine learning, machine learning papers addressing similar modeling, algorithms, or implementation issues that look at different applications *are related work*.

Specific Requirements: Related Work

- This section is **not** just a separate summary of each of your papers.
- Think of this section as an extended justification for the novelty of your work.
- It should be written as **narrative description** of how your selected papers relate to each other and to your project contributions.
- Focus on what problems they are solving and what solutions are proposed. Point out the weaknesses that your project addresses.
- You should provide **mathematical and/or algorithm** details for at least one model that you will compare to in your experiments.
- Recommended page length: 1-2 pages.

Specific Requirements: Methodology

- The methodology section is the core of your paper.
- It is where you communicate the details of the solution to the problem you are addressing.
- Your goal is to convince the reader that the details of your solution are *technically correct* by supplying sufficient mathematical descriptions, derivations, proofs, pseudo-code, etc.
- You need to write in a way that is simultaneously extremely clear, concise and correct.
- Recommended page length: 1-6 pages.

Specific Requirements: Methodology

For implementation projects, this section will describe the models and/or algorithms that you decided to implement including:

- mathematical descriptions
- a description of the hardware/software platform or programming language you decided to implement the models and/or algorithms in
- your design methodology or implementation architecture
- any libraries or existing resources you leveraged for your implementation.

Specific Requirements: Methodology

For algorithms projects, this section should describe:

- the model that you will work with including mathematical descriptions
- derivations and/or proofs for the algorithms you are proposing
- pseudo-code description of the proposed algorithm
- any required implementation-level details

Specific Requirements: Methodology

For models projects, this section should describe: , as well as

- your proposed model including a mathematical description (and where possible, a graphical depiction)
- derivations for the associated algorithms that you are proposing for learning, inference, and/or prediction
- pseudo-code description of the proposed algorithm
- any required implementation-level details

Specific Requirements: Data Sets

- All projects require empirical comparisons and thus all projects need to describe the data sets used.
- Describe your data was obtained from. Give a url if you downloaded it from the web.
- Describe the number of data cases, the number of features, what the features represent and what their data types are, etc.
- For data sets with large numbers of features, you should provide a summary of the features and not an exhaustive listing (include a reference to a published paper or website that describe the data in more detail for large data sets if possible).
- Briefly explain why you picked each data set.
- Recommended page length: 0.5-1 pages.

Specific Requirements: Experiments

- You must perform experiments to validate your work.
- You should discuss the rationale for performing the selected set of experiments and what your hypotheses were for the outcomes. This rationale should follow from the problem you are solving and the motivation for your work.
- You should select experiments with the ability to validate or refute your hypotheses.
- In ML, these hypotheses typically take the form "Implementation/Algorithm/Model A is more performant than implementation/Algorithm/Model B" with respect to some perfromance metric.
- Recommended page length: 2-3 pages.

Specific Requirements: Experiments

- Implementation projects should validate the correctness of the proposed implementations against existing reference implementations and then consider additional hypotheses.
- Algorithms projects should compare the motivating aspects of performance (accuracy, convergence time, speed-accuracy trade-offs, etc.) relative to existing baseline and state-of-the-art algorithms.
- Modeling projects should conduct experiments to validate that the proposed modeling extensions relative to existing baseline and state-of-the-art algorithms.

Specific Requirements: Experiments

Machine learning experiments involve many details. You should make sure to describe:

- What data sets you used in each experiment and how the data were processed.
- What hyper-parameter settings you investigated for each model and how free hyper-parameters were selected (e.g., use of cross-validation).
- What performance metrics were used and why they were chosen.

Specific Requirements: Experiments

There are several common problems with machine learning experiments you need to avoid:

- Optimizing hyper-parameters for your approach, but not for comparison methods.
- Not using similar optimization convergence criteria for your method and for comparison methods.
- Not using a proper generalization assessment method (training on the test data, selecting hyper-parameters on test data).
- Using different train/test or cross-validation splits for different methods.
- Comparing run times for non-comparable implementations. Not controlling timing properly to account for multi-core vs single threaded code, etc.

Specific Requirements: Results

- In this section, you will describe and discuss the results of each of your experiments.
- Use suitable figures and tables (your report must contain at least 2/4/6 results figures or tables for groups of 1/2/3).
- Pick your graphs and visualizations with care to support the conclusions you wish to draw from your experiments as clearly as possible.

Specific Requirements: Results

- Avoid common problems like showing performance across a set of plots with different y-axes so that results are not visually comparable.
- Make sure to label axes, include informative titles and/or captions, use line styles/color and legends appropriately.
- Make sure to interpret your results with care. Provide a fair discussion that is supported by the results you obtain.
- **Whether your results are negative or positive is less important than whether you conducted a well-motivated and technically correct experiment in the first place.**
- Recommended page length: 2-3 pages.

Specific Requirements: Discussion and Conclusions

- In this section, you will discuss the overall outcomes of your project.
- How do your results relate to what has been reported in the literature previously?
- What seemed to work well and what didn't? Did you run into any particular problems?
- What else would you have done if you had more time (e.g., future work)?
- What should others learn from your results?
- Recommended page length: 0.5-1 pages.

Specific Requirements: References

- Provide a list of references to support your assessment of related work.
- You can provide unlimited additional pages of references.
- Use a standard format for references that includes authors, title, publication venue, year, page numbers, etc.
- A link to an arxiv version of a paper is not an acceptable reference format. Provide proper references to the published versions of all papers.

Specific Requirements: Formatting

- Max of 8/16/20 pages excluding references for groups of 1/2/3.
- Nothing will be graded beyond page limit except for references.
- Use standard NIPS format templates for Latex or Word. Do not modify the template.
- Use the top-level section headings just described.
- Use a standard format for citations and for references. If using Latex, use bibtex to manage your references.

Grading Breakdown

- Title/Abstract [5%]
- Introduction [10%]
- Related work [20%]
- Methodology [20%]
- Experiments and Results [20%]
- Conclusions [5%]
- Novelty [10%]
- Clarity [10%]

Project Submission

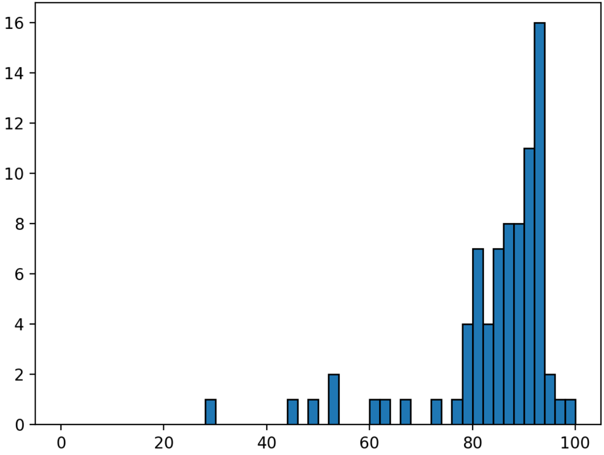
- Projects are due Dec 19, 2017 by 11:59pm. No late days. No late submissions accepted.
- Upload one project report per group and one code zip file containing code per group. See the video below posted by Gradescope that shows how to upload a group assignment:

  `https://www.youtube.com/watch?v=a6DERS94qPY`

Announcements

- Grading for Q11 and Q12 will be complete today.
- We will start the process of verifying pending and final regrade requests and re-importing final gradescope grades into Moodle this week.
- We will post to Piazza when this work is complete.
- Current Moodle grades...

Current Grade Distribution



Course Metrics: Scaling

- Final class size of 80 students.
- 66% growth over 2016 (48 students).
- 2nd largest PhD-level CS course this semester behind 682.
- 1,600+ pages of homework reports, 320 code submissions, and 900+ quizzes graded.
- Average homework report turn-around time of about 1.5 weeks.
- 1200+ Total piazza posts and follow-ups.
- 350+ Instructor answers.
- 100 minute average response time on Piazza.

Course Metrics: Content Generation

Completely new course content in F2017 including:

- 18 demos.
- 500+ pages of lecture slides.
- 5000+ lines of code related to assignments (data processing, solutions, autograder, etc.).

Content: Review

- **Foundations:** numerical optimization, MCMC methods, variational methods.
- **Learning Frameworks:** MLE, MAP, ERM, RRM, Bayesian Inference, Variational Bayesian Inference.
- **Supervised Learning:** Linear Regression, Logistic Regression, GLMS, SVM/SVR, MLP, CNNs, LSTMs.
- **Unsupervised Learning:** Mixture Models, Factor Analysis, RMBs, GANs, Autoencoders, VAEs.
- **Implementing custom models.**

What Didn't We Cover?

- A number of more advanced learning paradigms including semi-supervised learning, weakly supervised learning, active learning, reinforcement learning, and multi-task learning.
- A number of more advanced modeling and learning frameworks including structured prediction models and graphical models.
- A few general models and algorithms, and many, many more specialized models and algorithms.

Where to go from here? - ML Seminars/Courses

- CS 590M: Introduction to Simulation (Spring 18)
- CS 682: Deep Learning (Spring 18)
- CS 687: Reinforcement Learning
- CS 688: Probabilistic Graphical Models (Spring 18)
- CS 690M: Machine Learning Theory

Where to go from here? - Applications Courses

Where to go from here? - Applications Courses

- CS 585: Natural Language Processing
- CS 597R: Applied Information Retrieval
- CS 603: Robotics (Spring 18)
- CS 646: Information Retrieval
- CS 670: Computer Vision
- CS 690IV: Intelligent Visual Computing (Spring 18)
- CS 690N: Advanced Natural Language Processing (Spring 18)
- CS 690V: Visual Analytics

Course Evaluations

Please be sure to take some time to complete the course evaluation:

`http://owl.oit.umass.edu/partners/courseEvalSurvey/uma`

The End

Thank You!