CS/ECE/ME 532
Fall 2018 Final Project

Overview. The goal of the project is to create a tutorial and learning activity on a topic related to machine learning. You will work in groups of two or three students. Your target audience is students that have just finished taking the class. Your project should either (1) introduce/explain/demonstrate a machine learning technique that we didn't cover in class but uses the mathematical foundations we laid, or (2) compare and contrast several techniques covered in class but applied to a type of problem we didn't consider.

The project will be in the style of an active learning exercise. You will provide information about your topic and activities (problems and/or computer based) that will help develop understanding of the new concepts. Completion of your project by a classmate - digesting the information and performing the activities - should take about 1.5 hours.

Each student will evaluate and score two of their peers' projects using the rubric provided. This will result in approximately 4-5 peer evaluations of each project.

Timeline

1. 11:59pm Tuesday, November 6, 2018: Groups selected in Canvas

2. 11:59pm Sunday, November 11, 2018: Group identity and proposals (1/2 page description of topic) submitted to Canvas

3. Monday, November 19, 2018: Feedback on proposals provided

4. 11:59pm Thursday, December 13, 2018: All project materials submitted to Canvas

5. 2:45pm Tuesday, December 18, 2018: Your peer evaluations submitted in Canvas

Grading

Your final grade on the project will be based on the following components:

- Project proposal: 10%
- Project quality: 75%
- Your peer evaluation of two other projects: 15%

Project Proposal

Your one-page project proposal should include: (1) project title; (2) names, email addresses, and student numbers of group members; (3) A short (1/2 page maximum) explanation of the learning objectives for the project. The learning objectives should describe what someone will be able to do after completing your project activities. For example, "Students will be able to use matrix completion for image in-painting". I suggest having two or three (related) learning objectives and providing a sentence or two that elaborates on each of them.

Note that since someone needs to be able to complete your project activities in about 1.5 hours, you should not expect them to write more than a few lines of code.

Project Materials

Your project will involve a written component and may also include video, software, and data. There will be one submission per group. You should organize your materials as follows:

1. Title page: contain the title for the project and the names and email addresses of the group members.

2. Abstract/Executive Summary: This short section (about 1/2 page) should clearly explain the machine learning topics studied in the project and the learning objectives. This is an important part of your project because it is the first thing that the reader sees. Be clear and concise and use this opportunity to convey what makes this project interesting and unique.

3. Background: This section should explain the context for the main idea behind the project. You may create a short video ($<$ 10 minutes long) if you prefer that to a written explanation. If you use a video, then instruct the reader to view the video at the appropriate place in your document. Describe the problem that will be studied, a brief history of how the problem came about (with citations), why it is important/interesting, and any other relevant facts. You may use images/diagrams/etc from research papers, the internet, or other sources to help with your explanation as long as you cite your references. Finally, the background section should give the mathematical description of the problem with equations as needed.

4. Warm-up: This section should pose a few (3 or 4) short problems that are designed to test whether the reader was paying attention while studying the previous section. They can be math or computational problems. They should be short-answer, not too difficult, and should serve to prepare the reader for the tasks that lie ahead. For example, if the reader is expected to use a particular MATLAB script/function later, this would be a good place to introduce that script/function and use it to solve a simple problem. Solutions to the warm-up exercises should be included in the appendix.

5. Main activity: This section should contain three to five tasks that call on the reader to solve, analyze, visualize, and discuss problems related to the main theme of the project. Each problem should be accompanied by additional explanations if needed. You may intersperse problems within text, but be sure to label them clearly so the reader knows what they are being asked to do. If you are using external data files/sources, be sure to explain where the data is coming from (internet, synthetically generated, etc). Complete solutions to these problems should be included in the appendix.

6. References: Contains citations for references given in the Background or Lab sections, sources of code and data.

7. Appendix: Complete solutions (code, figures, equations, etc) for the warm-up problems and main activity. If virtually identical code is used multiple times, there is no need to repeat it. Rather make a note of any change.

An important note about code: You may use any programming language you wish for developing the project. However, you will not know in advance what computing system will be running the code you provide, so I strongly suggest using MATLAB for any code you wish your reader to run. You may assume everyone evaluating your project is able to run MATLAB through the UW license. You may not assume your readers have access to Python, Julia, Fortran 66, Latin or ancient Greek.

Topic Ideas

You are encouraged to come up with your own ideas for topics. However, if you are struggling to identify something, here are some ideas to help get you started:

- Recommender Systems and Collaborative Filtering `http://www.slideshare.net/erikbern/collaborative-filtering-at-spotify-16182818`
- Matrix Completion `http://statweb.stanford.edu/~candes/papers/SVT.pdf`
- Nonlinear Dimensionality Reduction `http://www.beermapperapp.com`
- Support Vector Machines `https://www.cs.cmu.edu/~cga/ai-course/svm.pdf`
- Deep Learning `http://arxiv.org/abs/1406.3332`
- Sparse Coding and Dictionary Learning `http://www.di.ens.fr/willow/pdfs/icml09.pdf`
- Neuronal Spike Sorting `http://www.scholarpedia.org/article/Spike_sorting`
- Sparse Methods for Machine Learning `http://www.di.ens.fr/~fbach/nips2009tutorial/nips_tutorial_2009_sparse_methods.pdf`
- Topic Modeling `http://www.cl.uni-heidelberg.de/courses/ss12/topicmodels/intro.pdf`
- Independent Component Analysis `http://cs229.stanford.edu/notes/cs229-notes11.pdf`

- Spectral Clustering http://cs.nyu.edu/~dsontag/courses/ml14/notes/Luxburg07_tutorial_spectral_clustering.pdf
- Climate Data Analysis http://www.princeton.edu/~rvdb/tex/LocalWarming/LocalWarming.pdf
- Image Segmentation http://www.cis.upenn.edu/~jshi/GraphTutorial/
- Anomaly Detection http://www.cs.bu.edu/faculty/crovella/paper-archive/sigc04-network-pdf
- Deconvolution and Deblurring http://www.mathcs.emory.edu/~nagy/courses/fall06/ID_lecture1.pdf
- Genomic Data Analysis and Classification https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2253491/
- Spectral Learning Algorithms for Natural Language Processing http://www.cs.columbia.edu/~scohen/naacl13tutorial/