EC 503 COURSE PROJECT
"Learning from Data"
Date: March 3, 2018
Deadlines:
Team formation and topic selection: March 16, 2018
Project summary due: March 26, 2018
Progress report due: April 9, 2018
Presentations: Apr 30 & May 1, 2018
Final report due: May 2, 2018

The course project is one of the core elements of the learning process in EC503, and a major contributor to the final course grade. Thus, it is very important to devote a sufficient amount of effort and time to do well in the project. The project has four main objectives:

1. to test and extend your knowledge acquired in the course on a contemporary machine learning problem,
2. to test your skills in literature search while correlating the knowledge you acquired in the course with the work of others,
3. to test your ability to map a machine learning problem onto a mathematical formulation, analyze it, and then to map this formulation onto a set of instructions in *Matlab*, *R, Python*, *C/C++/C#*, *Java*, or *Weka*,
4. to test your ability to communicate the results in written and oral form.

TEAM SIZE AND COMPOSITION

A balanced team is one where (1) analytical, (2) computational, and (3) presentation skills are adequately represented and where team members work together conscientiously and cheerfully with mutual respect and aim to learn from each other and grow. The recommended (and preferred) team size is 3 due to a combination of pedagogical and logistical considerations.

TIMELINE

The project will be executed in three phases as outlined below. Please pay careful attention to the deadlines, which are also listed at the top of this page.

- *Phase I*: You select two teammates and a topic according to the guidelines listed in the PROJECT GUIDELINES section of this document. By March 16, 2018, each team *emails* Prof. Ishwar the team members' names, BU email addresses, and an informal 1–2 paragraph description (i.e., an informal abstract) of the topic they plan to work on. Then each team carefully researches the literature using conference and journal papers, books, and the Internet, and prepares a formal 2-page summary. This phase must be completed by March 26, 2018 when the summary is due. Details as to what to include in the summary are explained in the REPORTS section of this document.
- *Phase II*: Each team further researches the topic, now in more depth, using various resources, analyzes, implements and then evaluates one or more algorithms in *Matlab*, *R,*

Python, *C/C++/C#*, *Java*, or *Weka* on one or more datasets. You may use native functions in *Matlab, R, Python, etc.,* or those available on the Internet (except for a complete solution to your problem, if it exists). Each team prepares a progress report that extends the original summary by describing steps completed thus far. This report, no more than 4 pages long, is due on April 9, 2018. See the REPORTS section for details on report content.

- *Phase III*: Based on the completed research, each team prepares a 20-minute oral presentation followed by a short question period to be given on Apr 30 & May 1, 2018 (see PRESENTATION section for details). In parallel with getting ready for your presentation, prepare a final report (up to 6 pages; see REPORTS section for details), which is due on May 2, 2018. You are also required to submit the slides of your presentation and the project-code to the course website by May 2, 2018.

PROJECT GRADE BREAKDOWN

5%	Project Summary
5%	Progress Report
5%	Project Software
35%	Project Presentation
50%	Final Report

It is expected that each team member will participate and contribute to every aspect of the project from problem formulation, literature review, algorithm development, and mathematical analysis, to coding, discussion and comparison of results, report-writing, and the presentation. Different team members may assume leadership roles and do the "heavy lifting" in different aspects of the project. However, everyone is expected to be fully aware of what everyone else is doing. The roles and contributions of each team member should be clearly discussed in the final report, e.g., who wrote which part of the final report, who was responsible for which aspect of coding, who did which part of the literature survey, who analyzed which algorithm, etc.

The final report, presentation, and code will be subjected to a plagiarism check. Failing any of these checks can have a serious consequence on your academic status at BU. If you are not a native speaker of English, please note that reproducing text from the literature verbatim in your report will be treated as plagiarism even if you cite the source and enclose the borrowed text within quotation marks. **You are expected to explain ideas using your own words.** You may take the help of your team members to refine your explanation, but copying from published work is absolutely not permitted.

Although communication between teams is permitted, collaboration is not. All the work should be carried out by each team independently. The remaining classmates and other presentation participants (other faculty, students, etc.) will participate in the evaluation of your project presentation.

REPORTS

The project summary, progress, and final reports must be typeset and submitted electronically in pdf format (no paper copies). The pdf file must be uploaded to the course web site (only one electronic submission per team) in the project section (each summary/report is an assignment).

Project summary

The project summary should not exceed 2 pages of text (figures, tables, references, and the division of labor section are extra). It should describe the problem under investigation, briefly review the literature and outline the proposed work (including datasets and code you need to write). The description of proposed work should be at two levels: an ideal objective, that the team hopes to achieve, and a fall-back plan, in case the ideal turns out to be unattainable. A list of references should be included at the end of the summary. The last section must clearly explain the division of labor, i.e., what each member is primarily responsible for, preferably as a bulleted list.

Progress report

The progress report should expand upon the summary by elaborating on work accomplished, work in progress, and work that needs to be done. This includes experimental results, augmenting the literature review, explaining challenges ahead, changes to the division of labor, etc. It should not exceed 4 pages of text (figures, tables, references, detailed mathematical derivations, and the division of labor section are extra).

Final report

You must use one of the report templates, either in Microsoft Word or in Latex, available in the "Project" section of the course web site. Please do not change the report format. These templates have been taken from the author-kit of a CVPR conference. The final report should be structured as follows:

1. *Introduction*: Introduce the goals of the project (no detail, just overview). Motivate the study and discuss how it is similar to or different from other problems in machine learning. Specifically, place your work within a broad context of the field, discuss connections to other problems, and the importance/relevance of the specific problems you are proposing to study. Identify any novel or challenging aspects of your project.

2. *Literature Review*: Briefly review the literature by discussing related work. Distinguish between works that are only broadly related to the topic of your project and a smaller subset of key papers (this could be just one or two) that are most directly and closely related to your project. Discuss how the key papers are related to your project and highlight similarities and differences.

3. *Problem Formulation and Solution Approaches*: This could span multiple sections or subsections.

    - Describe, in precise mathematical language, the learning and inference problems being studied in detail, and the solution methodologies that you are investigating. Clearly identify the underlying constraints and assumptions: what is assumed to be known? what is unknown? what is to be learned using what? what is to be decided using what? are there any constraints on what you are allowed to do or not do? Describe the underlying probabilistic models (if any).
    - Describe the learning and inference algorithms and analyze their properties mathematically. Specifically, discuss the final optimization problems (maximization or minimization of suitable objective functions) associated with the algorithms. Are

there any theoretical properties, e.g., unbiasedness, asymptotic consistency, sample complexity, convergence rate, etc., available for the algorithms?

- Discuss the underlying intuition and key ideas behind the models and algorithms.
- Discuss any overfitting issues (curse of dimensionality) and the use of priors or other forms of regularization to mitigate them.
- Discuss the time/memory complexities of the algorithms.
- Use consistent mathematical conventions and notation throughout. Think carefully about what mathematical symbols are necessary for the exposition and what are unnecessary. Every symbol and index that appears in the report must be explained at the location of its first occurrence or very close to it.

4. *Implementation*: Describe how you have implemented the proposed methods. Include the source code in the appendix (not counted towards the page limit).

5. *Experimental Results*: Describe the experiments you have conducted on synthetic and real data and the results you have obtained (datasets, performance measures used, sensitivity to initialization or other algorithm parameters, impact of real-world degradations like missing data, etc.). Discuss the appropriateness of the performance metrics that you have chosen. Discuss if your results are consistent with your understanding, i.e., are they expected or unexpected? do they make sense?

6. *Conclusion*: Describe what you have learned from the project and what further improvements are possible.

7. *Description of Individual Effort*: At the end of the report, please include a brief description of each project member's contribution to the project. In particular, discuss who wrote which part of the final report, who was responsible for which aspect of coding, who did which part of the literature survey, and who analyzed (theoretically or intuitively) which algorithm. Like the appendix, this falls outside the page limit.

8. *References*: Include the list of references you used (again not counted towards the page limit).

9. *Appendix*: **Extra** figures, **extra** tables, detailed mathematical derivations, and *Matlab*, *R, Python*, *C/C++/C#*, *Java*, or *Weka* code.

   Note: Unlike the summary and progress reports where figures and tables were not counted towards the page limit, in the final report, all the **key** figures and tables must be included within the 6 page limit. If there are some finer points that you wish to make which require additional figures and tables, you may include them in the Appendix.

Criteria on which the final report will be evaluated:

1. *Organization, neatness, clarity, logic, and flow:* 5 points

2. *Abstract and Introduction:* 5 points

3. *Literature Review:* 5 points

4. *Problem Formulation, Solution Approaches, and Implementation*: 15 points

5. *Experimental Results:* 15 points

6. *Project difficulty:* 5 points. How challenging was the project in terms of mathematical and algorithmic difficulty, implementation difficulty, and the difficulty of datasets.

PRESENTATION

- The presentation, in PowerPoint or PDF, should consist of about 20 slides. Depending on the content, more slides are acceptable if many figures or plots are shown. But you **must not exceed** the 20-minute time limit for your presentation. As a rule of thumb, you should have no more than about 1 slide per minute of presentation.

- The 20 minutes of the presentation should be shared approximately equally by all team members.

- You can bring your presentation on a USB drive, but if you are planning to run a live demo, then you should use your own laptop (please test your laptop with the projector in advance to avoid delays during presentation).

- You need to upload your presentation to the project section of the course web site.

- Your presentation should be structured similarly to the report and should contain: introduction, related work, problem formulation and solution approaches, description of your implementation, presentation and discussion of experimental results, and conclusions. The discussion of related work can be much shorter than in the final report and you need not present detailed mathematical derivations (unless that is crucial to your project).

- Use bullet lists, figures and diagrams. Do not put a lot of text in any slide. Short phrases are preferred to long complete sentences. Choose the equations that you want to display carefully (do not blindly dump every single equation from the final report into your presentation). Any symbol that you display must be explained at some level otherwise it need not be displayed at all. Aim to provide an intuitive explanation or interpretation of the underlying mathematics.

- Express your ideas clearly. Remember that your fellow students may be less familiar with the topic, so your team must give a clear, well-organized presentation. Please practice a "dry run" of your presentation in front of your friends.

Please note that you will be asked questions, some of them quite detailed, about your work during the oral presentation. Therefore, you must come prepared to the presentation in terms of both theoretical underpinnings of your work as well as your implementation. In addition to the course instructor and your classmates, other faculty and students may be present.

Criteria on which the presentation will be evaluated by the audience:

1. Are the project goals and significance clear?

2. Is the problem formulation clear? (assumptions, constraints, model, data, etc.)

3. Do you understand the solution methodology (algorithms) and the key insights?

4. Are the implementation details and main experimental results clear?
5. Is the flow of the presentation logical and the slides neat, clear and easy to follow?
6. Is this a challenging project? (mathematically, algorithmically, implementation, or datasets)

PROJECT GUIDELINES

- Focus on exploring learning-frameworks and algorithms instead of specific applications or datasets, i.e., do not tie your project down too strongly to a specific dataset or a specific application. Aim to explore and compare (pros and cons of) a few learning methods.
- You must explore at least one algorithm that is not part of the course syllabus and at least one that is part of the syllabus.
- You must compare performance according to more than one metric and ideally also report confidence bounds.
- You should test algorithms on least 2 datasets: one clean small-scale dataset on which you can quickly develop and debug your code and one larger more challenging dataset.
- Choose datasets wisely: do not select datasets that require an enormous amount of "black-art" pre-processing for selecting good features for learning. Although very important in practice, it is not the focus of this course. Preprocessing can quickly turn into a huge time sink. It is a quagmire that you should avoid getting sucked into. You want to focus on understanding and comparing different learning algorithms.
- That said, it would be good to study the impact of at least one type of real-world degradation (e.g., missing feature components, data variability, etc.) on the performance of different algorithms and methods to mitigate the degradation.
- It would be good to explore the value of some kind of regularization (use of a prior) in your problem to mitigate the impact of high data dimensionality or decision-rule complexity relative to the size of the training set.
- It would be good to have some discussion and comparison of the time/memory complexities of the algorithms.
- The scale of experimental validation, e.g., number of algorithms, number of datasets, size and complexity of datasets, etc., should be commensurate with the size of the team.

PROJECT IDEAS (incomplete list; you can propose your own project)

- Generalized Linear Models and Exponential Family
- Mixture Models and the EM Algorithm
- Hidden Markov Models
- Boosting

- Ensemble Methods
- Active Learning
- Online Learning
- Zero-shot Learning
- Reinforcement Learning with Applications to Computer Games
- Recommendation Systems
- Topic Models
- Word Embeddings for Natural Language Processing Tasks
- t-SNE Emebdding for Visualization
- Independent Component Analysis and Blind Signal Separation
- Generative-Adversarial Networks and their applications
- Social Biases in Machine Learning and methods for their Removal

Main machine learning conferences

- JMLR Workshop and Conference Proceedings: `http://www.jmlr.org/proceedings/`
- NIPS: `https://nips.cc/`
  NIPS papers: `http://papers.nips.cc/`
- ICML: `https://icml.cc/Conferences/2018/`
  ICML papers (pre-2015): `http://www.machinelearning.org/icml.html`
- AIStats: `http://www.aistats.org/past.html`
- KDD conferences: `http://kdd.org/conferences`

DATASET SOURCES (incomplete list)

- UCI Machine Learning Repository: `http://archive.ics.uci.edu/ml/`
- UCI KDD Archive: `http://kdd.ics.uci.edu/`
- `mldata.org`
- Kaggle: `www.kaggle.com`