DS 5230: Unsupervised machine learning and data mining (Fall 2020) Roi Yehoshua
(Due) December 21, 2020

Final Project

1 Project Description

Your final project can be on anything that has some relation to unsupervised machine learning. We suggest a few example topics in a later section of this document.

There are two general types of projects you may choose from:

- **Practical**: Find and formulate a problem, develop/implement/apply algorithms we studied in class (or perhaps other methods that we did not study!) to a real-life problem.
- **Theoretical**: Do an independent study of an advanced topic not covered in the course. Could also be used to deeply study / reproduce a research paper.

2 Teams

Teams should have 1–3 members. Our recommendation:

- Practical project: We strongly recommend forming teams of 2, which in past experience has led to much more interesting projects. Overall, we expect each team member to spend about 40–60 hours on the project. This is why teams of two are recommended – you can accomplish much more with 100 total hours compared to 50.
- Theoretical project: Teams of 1 or 2 are recommended. If in a team of 2, the chosen topic should be broader, and there should be a rationale as to why 2 people are needed.

If you are working in a group, each student should work on a different part of the project and what each student will work on should be explicitly listed in the project proposal.

3 Potential Topics

This section provides some general topic categories, but you are welcome to choose anything else that interests you.

Topic Categories

- (Practical) Find a publicly released dataset (e.g., all of Wikipedia text, Yelp data, Arnetminer), identify some questions you would like to answer (e.g., what types of food preferences correlate), apply multiple techniques on the dataset (clustering, dimensionality reduction, association rule mining, recommendation systems), and provide detailed discussion of the results.

- (Practical) Find some interesting problem / dataset on Kaggle or some other competition platform related to one of the course topics and tackle it. If you do well, you might be able to enter/win a competition!

- (Theoretical) Learn extensively about a problem / model / algorithm that is just beyond the scope of the course. Develop / apply and convey the understanding on illustrative examples. Produce pedagogical materials about the topic.

- (Theoretical) Find a paper that tackles an interesting problem, and try to re-implement it (initially without referring to existing code, if any is available). This can be quite challenging because you may find that some crucial details (e.g., particular settings of constants or hyperparameters) may not be present in the paper!

Projects that are *not* recommended

- Apply an existing algorithm to an existing dataset and report the results. This is not enough – you need to analyze the results, see what was good and what could be improved, and iterate.

- Projects on unrelated topics – if you are unsure, ask us!

- Projects that are too broad. 40-60 hours (or 80-120) is not a lot of time. Start small, and if you succeed early, extend and iterate from there. If there is an interesting problem that is likely to take too much time (this is true for many interesting problems), identify the first step in the problem and make that your project.

4 Project Proposal

The project proposal is a short document that organizes your team's intentions and communicates that to the course staff, such that we can provide appropriate guidance. The proposal must show that you have read background material on your topic and are qualified to undertake what you propose to do. It should include full references for the papers and other sources that you have consulted and that will form the foundation for your work.

There is no specific page limit, but we expect that a 1-2 pages should suffice. The proposal must specify all the following items:

- Project participants: Who is on the team? If you are working in a team of two or three, is there a clear division of labor? If team is smaller/larger than recommended, provide a rationale.
- Problem description: What problem are you trying to solve? Describe the problem *formally* from a computational perspective. What are the inputs and outputs? Why is the problem interesting?
- Algorithms: What algorithms do you expect to use? Why are these algorithms appropriate? How are these algorithms typically used, and how are you using them? Have other people used similar algorithms to solve your problem before?
- Datasets: What data sets are you using? Where will this come from? Is it an existing dataset, or are you making a new one? If the latter, do you have the resources to do so?
- Libraries and tools: What libraries / platforms, if any, will you have to learn in order to undertake your project? Provide references where applicable.
- Results: What is the ideal outcome of the project? What results do you expect to show? What comparisons will you do? Are there risks for not getting all the results? If so, what will you do about it?

All team members should submit the proposal on Canvas. We will review all project proposals, and they will be part of your grade on the project.

5 Project Presentation

The presentation should be 5 minutes long, with an additional 1-2 minutes for questions. Ideally, all members of each team will present. There are some cases where inevitably only one member is available, which is fine; but if are all present, you should decide how to split up the presentation.

Suggested presentation structure

- (1 min) Describe the problem on a high level. What are you trying to do? When is the input and the output?
- (1.5 mins) Provide a technical problem formulation. This does not have to be mathematical, although it can be. If you have a supervised learning problem, this section should explain what features you are using, what form of output are you

trying to predict, what the dataset looks like, etc. Basically, at the end of this section, we should start to see how we might obtain a solution for your problem (e.g., by applying an algorithm from class that works for the type of problem you have).

- (1 min) Describe the algorithm you chose to apply to solve your problem. If you have tried more than one, you can tell us more, or you may want to just focus on the most interesting one.
- (1 min) Explain results obtained from this algorithm, ideally compared against some baseline. Did the algorithm do well? If so, what made it work? If not, why not (and what should be the next thing to try)? Remember to take the time to explain what the experimental setup is, and explain what empirical numbers / graph axes mean.
- (0.5 min) Optional: Future directions / some high-level thoughts about your project so far / conclusion.

The above structure is only a suggestion. You can choose to present in a different way. Ultimately, try to tell an interesting story. If the algorithm is not too interesting, but you have interesting empirical results and insights, focus your time on the latter. Negative results (i.e., we could not get it to work) are completely acceptable too! Why did things not work – was it the algorithm, the features, the data, etc.? What surprised or frustrated you? Did certain aspects of the project take much longer than expected? Your classmates (and we) will greatly appreciate whatever insights you share, including how difficult it is to get something to work.

As you can see from the above, we do not expect project to be complete by the presentation. But certainly everyone would have put in enough time by this point to say 6-7 minutes about their project. Tell an interesting story, hear about others' stories, and have a good time!

6 Final Report

The papers should be written using the AAAI format (for the AAAI Conference on Artificial Intelligence): `http://www.aaai.org/Publications/Templates/AuthorKit20.zip`. Project reports must be detailed and self-contained, explaining the problem, methods and results. Your report can be organized differently, but the general organization is the following:

1. Abstract: A short summary of what problem you are solving, how you solved it and what the results are.
2. Introduction: A longer description motivating the problem and solution method.

3. Background: Any background information needed to understand the methods used in the project (e.g., a description of the theoretical framework or existing algorithms that you build off of).

4. Related work: If you are basing your project on someone else's work, explain on a coarse level what they have done, and if you are doing anything different. Also, what other methods could be applied to your problem, why didn't you use them and how they relate to your method.

5. Project description: What you actually did in formal detail (with algorithms, equations, etc.).

6. Empirical results: A description of which experiments were run (be precise about what settings/data you ran experiments with), what the results were and why you got these results (under what circumstances does the algorithm solve your problem successfully? when does it fail?) Again, these should be formal, often with graphs. The results could also include analysis such as a comparison of different methods or performance on different variants of the problem.

7. Conclusions / future directions: A summary of the results and what you learned by trying to complete this project. If you had more time to spend on the project, what would you have liked to do next? What advice about the project would you give to future DS 5230 students?

There is no set length / page limit for the report. It would probably take at least 6–8 pages (including figures) to include all of the above. You can write as much as you wish, but do not be excessive – just get to the point and provide a complete description of the project, including the components described above.

7 Timeline and Deliverables

All deadlines, except for the project presentation, are at 11:59 PM.
Submissions should be uploaded to Canvas by all team members. If it is a practical project, also submit the code used to produce your results.

- The project proposal is due 11/10.

- Project presentations will happen in the final two classes on 12/8 (Tue) and 12/11 (Fri). You will be given the option to choose one of the time slots available in these two days. Your team should upload slides *before* lecture on the day of your team's presentation.

- **The final project report / deliverables is due on 12/21.**
  There can not be any extensions on the final project, because final grades are due soon after that, and we need time to ensure every project receives a proper assessment.