Project Requirements

Machine Learning: 10-701

Spring 2020

1 Course Project Guidelines

Your class project is an opportunity to explore a machine learning problem of your choice in the context of a real-world data set. Below you will find specific data sets and potential project ideas from which you may pick one project. If you wish to define your own project with a dataset you already have in hand, you may propose a project as described below.

A typical project consists of (1) picking one of the datasets and tasks below (e.g. text classification over the IMDB Movie Review Dataset), (2) applying one or more basic machine learning approaches to establish a performance baseline (e.g., apply Naive Bayes to the IMDB review bag of words), then (3) going beyond your baseline approach to develop and study more sophisticated and hopefully more successful approaches (e.g., apply Bayes nets with different assumptions than Naive Bayes, try learning latent representations of the review to improve classification accuracy, etc.), and (4) providing a critical assessment of the results as well as an evaluation of their robustness and/or reproducibility. The number of approaches you explore in (3) will probably depend on the number of members in your team.

1. Projects can be done in teams of 2 or 3 students. Team members are responsible for dividing up the work equally and making sure that each member contributes.
2. Each project has a TA as a project consultant/mentor.
3. Can you define your project to use a different data set? Only in the following situation: (1) you want to design a project around a data set associated with your ongoing research, and (2) your faculty advisor associated with that ongoing research writes us an email stating that she will be happy to serve as mentor for your class project, and (3) you write a 1-2 page project proposal, submitting it to Gradescope by February 7.
4. Can you have a project team of just one student? Only if you (1) are using your own data as discussed in point 3 above, and (2) you absolutely cannot find another student in class who is interested in working with you on that data.

Your project will be worth 25% of your final class grade, and will have three deliverables:

1. By Feb 7 you should report to us your team, and which of the projects below you will work on. You can do this on this form. If you are defining your own project, see requirements above or on Piazza.
2. Midway Report, due March 06: maximum of 3 pages (not including references) (20%)
3. Poster Presentation, April 1, times TBA: (20%)
4. Final Report, due May 01: maximum of 8 pages (not including references) (60%)

Grading Criterion

The page limits are strict! Papers over the limit will not be considered. Each deliverable of your project will be evaluated based on several factors:

1. Creativity: You are encouraged to come up with original ideas and novel applications. A project exploring new ideas (algorithms, methods, theory, applications) is scored higher than a project without many new ideas.

2. Completeness: The extensiveness of the study, experiments, and analysis of results. A project that produces a more intelligent system by combining several ML techniques together, or a project that involves well-designed experiments and thorough analysis of the experimental results, or a project that nicely incorporates various real world applications, are scored higher.

3. Clarity of writing: The report should be organized clearly and well written.

4. NeuIPS format: Use NeuIPS conference paper format for all your reports.

5. Structure: See the following section for more details.

Report Structure

Midway Report

1. Title

2. Introduction and Problem Setup: Which project did you choose? What question is your project trying to answer? Clearly define the problem you are trying to solve. Unclude a precise description/background about the problem and data that you chose. It should be clear what the inputs and outputs of your trained model will be, and how you plan to analyze your results.

3. Data: What are the specifics of the dataset that you are using? How was it curated? What problems or biases can it potentially have? Is it characteristic of the real world distribution of the phenomenon you are trying to model? Would this dataset be better suited for another problem?

4. Background/Literature: You should research work related to your own. What problem did they solve and how does it relate to yours? How can you improve on what has already been done?

5. Methods/Model: By the time of the Midway Report, you are expected to have implemented a baseline (i.e., a standard machine learning approach to which you will compare your next method). Describe exactly what method(s) you have implemented and show plots of the performance. Additionally, if you have already gone beyond the baseline, you should describe what work you have completed towards creating a method which beats the baseline.

6. Preliminary Results: Your experimental results. Show plots of the performance of your baseline algorithm and interpret what they mean. Be sure to label and explain this clearly. At this point, we will not provide any code for generating canonical plots, so that you are free to think about the best way to visualize the learning process and the performance of your model.

7. Evaluation of preliminary work: Evaluation of baseline method and any other algorithms you have tried. How do you hope to improve on the work which you have already done? What was successful and what was unsuccessful?

8. Future work: Which techniques do you plan to apply to beat the baseline method? What is your motivation behind these techniques (you are highly encouraged to come up with an original idea of your own or interesting applications rather than simply implementing or applying existing ML algorithms)? How do you plan to evaluate your final method? Goals, timeline, and division of work throughout your team. Provide a rough timeline of your plan ahead and job to be done by each team member.

9. Teammates and work division: We expect projects done in a group to be more substantial than projects done individually. You should outline what everybody in your group will do and by when each task should be complete.

10. References and citations: Clean and correctly formatted citations and bibliography.

Final Report

1. Title
2. Introduction: What is the problem you trying to solve? Why is it important?
3. Data: What does your data look like? Why is it suited to this problem? What biases does it have that should be taken into account while evaluating the results?
4. Background: Briefly summarize the findings from your midway report. Please do not include any "boilerplate" content (e.g., descriptions of the domain, standard background on Machine Learning, explanations of Bayes Nets or other methods implemented in your homeworks, although see part 5 for the details to include in your Methods section).
5. Related work: Previous work related to your topic that you may have referenced to help guide your project.
6. Methods: By the final report, we expect you to have implemented your own ideas beyond the baseline. Additionally, you should describe what work you have completed towards creating a method which beats the baseline.

   More specifically, you should provide a detailed description of your models. You should start by listing all the variables for each model. Then, write the likelihood equations (As in most academic papers, you are not required to show the derivation). In addition, You should define all the optimization steps and equations in training your model (e.g. gradient descent, Newton's method, expectation-maximization, MCMC, etc). You should also describe the algorithm used to solve the optimization problems. For instance, if you come up with your own algorithm or heuristics, write the pseudocode and update equation at each step. From your report, the reader should be able to successfully replicate your results by simply following your descriptions, i.e. there should be no ambiguity as to how you implement your model. You can still use pre-existing implementations/libraries.

   Guideline:

   a. Define your variables and parameters explicitly. Indicate if continuous or binary/categorical. Also the dimensions. This applies even if you are doing feature transformation.

   b. If your model is generative, succinctly describe all the generative processes. This is especially important if you are using a hierarchical model and/or hidden variables.

   c. Write your full objective (the likelihood or other cost function plus any regularization or constraints).

   d. Describe your optimization algorithm stpe-by-step. This will likely involve update equations for each of your parameters.

   e. Report your likelihood/loss progression. If this step is expensive, do it every n-th step.

   f. For nonparametric algorithms, like kernel-SVM, KNN, GP, decision trees, etc, you still have to define and explain all your hyperparameters, kernels, and algorithms.

   g. All of the above applies if you are using deep generative models. In general for neural networks, make a figure fully describing the architecture and all layers.

   In addition to successful approaches, you should briefly detail approaches which you tried and found to not work well. What methods have you completed? What is your motivation behind these techniques (you are highly encouraged to come up with an original idea of your own or interesting applications rather than simply implementing or applying existing ML algorithms)?
7. Results: Your experimental results. Show plots of the performance of your algorithms and interpret what they mean. Be sure to label and explain this clearly. Describe how the current results in each of the experiments align with your expectations. Make sure to provide confidence intervals where appropriate or standard errors while comparing methods. What metrics did you use for evaluation? How do your results compare to prior work? In order to more easily compare training curves across different reports, we will release simple Python plotting code. This will create a plot from a list of $(episode\ number, reward)$ tuples. You will have to use this provided code without modification.

8. Discussion and Analysis: Critically analyze your model and results. Was the method you used appropriate for the question you were asking? Are the results conclusive? Highlight the limitations of your approach (e.g., strong assumptions you had to make, constraints, when your method did not work in practice, etc.). Do the results and the explanation provide insights into the ML models or the environment that you were dealing with? Comment on whether you think there is a way to further improve your method to eliminate these limitations.

9. References and citations: Clean and correctly formatted citations and bibliography.

2 Course Projects

1. **Image classification for Caltech101.**

   Image classification is a supervised learning algorithm where we define a set of target classes (the objects to be identified), and we train a model using previously labeled images. The trained model has to identify the target classes when a new unseen image is presented to it.

   Images on the Caltech101 dataset consist of pictures (around 300×200 pixels) that belong to 101 categories, where each category has around 40 to 800 images (unbalanced classes).

   The dataset can be found here: Caltech101

2. **Text classification for IMDB Movie Review Dataset.**

   The goal of text classification is to automatically classify the text documents into one or more defined categories. You can think of it as an example of supervised machine learning task since a labeled dataset containing text documents and their labels is used to train a classifier. We introduce one of the most popular and fundamental datasets for text classification, IMDB movie review dataset. With this benchmark dataset for sentiment classification, we hope you to explore the pipeline of text modeling, from pre-processing step to make the text easier to process to modeling step to design and implement a model to predict a label based on the vectorized text.

   The IMDB movie review dataset contains movie reviews along with their associated binary sentiment polarity labels. It consists of 50,000 reviews split evenly into the 25k train and 25k test sets. The overall distribution of labels is balanced (25k pos and 25k neg). It also includes an additional 50,000 unlabeled documents for unsupervised learning.

   The dataset and more information about it can be found here.

3. **Predicting NonProfit Funding Decisions**

   DonorsChoose.org is an online charity where teachers in K-12 schools can directly ask for aid for projects to enhance the education of their students. When a project reaches its funding goal, they ship the materials to the school. Suppose you wanted to create a help center to assist teachers in creating projects so that they are always likely to be funded. You want to refer the top 10% of projects that are likely not to be funded to this help center. How do you build a model that will effectively identify the top 10% of projects at risk of not being funded? And how can you make sure that that model is equitable–in that the error rates (or false positive/true negative rates) are equal across a protected attribute of your choice, e.g. poverty level of the school, discipline of project, or gender of the teacher? Data: KDD NonProfit

   Focuses/challenging questions:

   - How do you build a model where we care about being accurate for the top 10% of cases, and not overall?
   - What kind of changes can you make to a model for it to behave more fairly with respect to error rates? (Feature selection, dataset balancing, etc.)

4. **Semi-Supervised Learning**

   In many applications, it is easy to obtain a large amount of unlabeled data, but difficult or costly to label the data. Semi-supervised learning studies algorithms which learn from a small amount of labeled data and a large pool of unlabeled data. Interestingly, semi-supervised learning is not always successful, and unlabeled data points do not always improve performance. Semi-supervised learning algorithms typically make an assumption about the data distribution which enables learning – for example, several algorithms assume that the decision boundary should not pass through regions with high data density or that nearby data points are assigned similar labels. When these assumptions are satisfied, the algorithms perform better than supervised learning.

   The goal of this project is to experiment with semi-supervised learning algorithms on a data set of your choice. Some algorithms you can consider using are: co-training, self-training, transductive SVMS (S3VMs), or one of the many graph-based algorithms. (We recommend reading the following surveys describing the many classical approaches to semi-supervised learning: [1, 2].) It may also be interesting to explore more recent approaches such as self-supervised pre-training, or consistency regularization using data augmentation.

   You may compare several semi-supervised and supervised algorithms on your data set, and perhaps draw some general conclusions about semi-supervised learning. We also recommend exploring at least two modalities (e.g. images and text) to understand how well techniques transfer across modalities.

   This project can use essentially any data set. For some ideas, we recommend consulting the UC Irvine Machine Learning Repository. In addition, the following datasets are often used for semi-supervised image and text classification. (You will have to request access to the Yelp and Amazon datasets.)

   - Image classification datasets: MNIST, CIFAR-10, STL-10, SVHN
   - Text classification datasets: IMDB, Yelp Reviews, Amazon Reviews

5. **Unsupervised classification of datasets**

   Unsupervised classification (also referred to as clustering) is the task of grouping similar data points together when unlabelled data is present. A clustering algorithm should be able to form groups of densely placed (intra-class distance) similar data points and separated by a large distance (inter-class distance) from other groups. Traditional clustering alogrithms include k-means, gaussian mixture models, spectral clustering etc. In this project, the students are expected to apply clustering alogrithms to practical datasets and provide a thorough analysis. The students can also use more modern techniques that apply deep learning for cluster analysis. The project can use any dataset or some of the following datasets:

   - MNIST
   - USPS
   - REUTERS-10K

6. **Neural Style Transfer**

   Neural style transfer is a basic topic in image to image translation, aiming at transferring the style of the source image to the target image. In this context, the source image is called *style* image and the target image is called *content* image. The desired result of style transfer is an image with the combination of the given style and content. Gatys *et al.* showed that it could be achieved by using the back propogation of a convolutional neural network with designed losses, which is the most straightforward method. Based on that, researchers have made efforts to improve the perceptual quality, computation efficiency as well as trying to demystify the underline principle. You may find [1, 2] useful.

   In this project, students are expected to experiment with style transfer algorithms on any datasets and give thorough analysis on the methods, parameters, performances and so on. The topics are not limited to conventional oil-painting style as shown in Gatys' paper. Students can explore any related topics they are interested in such as photo-realistic style transfer, typography transfer, Chinese painting style transfer, face generation, etc. Student can collect data by themselves or use the one we provide [link].