Machine Learning for User-Adaptive Systems: CSC 591/791

Self-Selected Project Reports Guidelines

Min Chi
Department of Computer Science
North Carolina State University

Self-Selected Project Guidelines

Your class project is an opportunity for you to explore an interesting machine learning problem of your choice in the context of a real-world data set. Below, you will find some project ideas, but the best idea would be to combine machine learning with problems in your own research interest. Your class project must be about new things you have done this semester; you can't use results you have developed in previous semesters or in other classes.

Projects can be done individually, or in teams of two to three students. For a group, group members are responsible for dividing up the work equally and making sure that each member contributes. For each project, the TA will be your a project consultant/mentor. Please consult with the TA before finalizing the project proposal. The final responsibility to define and execute an interesting piece of work is yours. You are strongly urged to consult the TA or the instructor early on if your project will rely purely on simulated data or if you intend to do a learning theory related project.

Your project will be worth 25% of your final class grade, and will have 3 deliverables:

- Proposal, Due Date: Oct 22nd: 1 page (2%)
- Project Presentation (Date: Dec 2nd Classtime)(10%)
- Project Final Report (Date: Dec 6th): 8 pages (13%)

Note that all write-ups in the form of a NIPS paper.
https://nips.cc/Conferences/2019/PaperInformation/StyleFiles

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:

- The novelty of the project ideas and applications. The groups are encouraged to come up with original ideas and novel applications for the projects. A project with new ideas (algorithms, methods, theory) on ML or new, interesting applications of existing algorithms is scored higher than a project without much new idea/application.

- The extensiveness of the study and experiments. A project that produces a more intelligent system by combining several ML techniques together, or a project that involves extensive experiments and thorough analysis of the experimental results, or a project that nicely incorporates various real world applications, are scored higher.
- The writing style and the clarity of the written paper.

Project Proposal (Due Date: Oct 22nd)

Page limit: Proposals should be one page maximum.
Include the following information:

- Project title
- Data set
- Project idea. This should be approximately two paragraphs.
- Software you will need to write.
- Papers to read.Include 1-3 relevant papers.You will probably want to read at least one of them before submitting your proposal
- Teammate (if any) and work division. We expect projects done in a group to be more substantial than projects done individually.

Project Presentation (Date: Dec 2nd, Classtime)

All project members should be present. The session will be open to everybody.

Project Final Report (Date: Dec 6th)

See seperate documents

Project Suggestions

Here are some project ideas and datasets for this year:

Semi-Supervised Learning

In many applications, it is easy to obtain a large amount of unlabeled data, but difficult or costly to label this 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. When this assumption is 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 (1) for a survey of the many approaches to semi-supervised learning.) You may compare several semi-supervised and supervised algorithms on your data set, and perhaps draw some general conclusions about semi-supervised learning.

This project can use essentially any data set. For some ideas, we recommend consulting the UC Irvine Machine Learning Repository.

(1) Xiaojin Zhu. Semi-supervised Learning Literature Survey.

Twenty Newgroups text data

This data set contains 1000 text articles posted to each of 20 online newgroups, for a total of 20,000 articles. For documentation and download, see this website: http://www.cs.cmu.edu/afs/cs/project/theo-11/www/naive-bayes.html. This data is useful for a variety of text classification and/or clustering projects. The "label" of each article is which of the 20 newsgroups it belongs to. The newsgroups (labels) are hierarchically organized (e.g., "sports", "hockey").

Available software: The same website provides an implementation of a Naive Bayes classifier for this text data. The code is quite robust, and some documentation is available, but it is difficult code to modify.

Project ideas:

EM text classification in the case where you have labels for some documents, but not for others (see McCallum et al, and come up with your own suggestions)

Spectral algorithms for learning Hidden Markov Models

Hidden Markov Models (HMMs) are a common tool for analyzing dynamic data. The most widely-used algorithm for learning HMMs is the Baum-Welch algorithm, which is essentially doing expectation maximization. Like all EM procedures, the Baum-Welch algorithm suffers from local optima, and in addition the trouble of determining the number of hidden states. Some researchers proposed alternative learning algorithms ([1] and [2]) that are free from local optima, and demonstrated their performance in a number of applications, including robot vision and audio classification. The goal of this project is to implement these algorithms and apply them to other dynamic data, such as brain wave signals , video streams (e.g., CMU motion capture database , which contains both video data and biomarker data), and/or any dynamic data in your field of interest. Compare results with those obtained by either standard HMMs or domain specific extensions/ enhancements.

For those who are looking for a specific application, the UCI machine learning archive has a section on time series data, which may be a good starting point.

References:

[1] A spectral algorithm for learning hidden Markov models, COLT 2009.

[2] Hilbert Space Embeddings of Hidden Markov Models, ICML 2010.

KDD cup 2010 (Provided by Datashop in Pittsburgh Science Learning Center)

The KDD Cup is the annual Data Mining and Knowledge Discovery competition in which some of the best data mining teams in the world compete to solve an practical data mining problem of some importance. Year 2010's challenge is about how generally or narrowly do students learn? How quickly or slowly? Will the rate of improvement vary between students? What does it mean for one problem to be similar to another? It might depend on whether the knowledge required for one problem is the same as the knowledge required for another. But is it possible to infer the knowledge requirements of problems directly from student performance data, without human analysis of the tasks? This year's challenge asks you to predict student performance on mathematical problems from logs of student interaction with Intelligent Tutoring Systems. This task presents interesting technical challenges, has practical importance, and is scientifically interesting.

[Download the datasets:]https://pslcdatashop.web.cmu.edu/KDDCup/downloads.jsp.

More information about the datasets can be found: https://pslcdatashop.web.cmu.edu/ KDDCup/rules.jsp

You will apply the various machine learning techniques to the KDD datasets to improve the accuracy on predicting "Correct First Attempt values" for the test portion of the data. Please report the Root Mean Squared Error (RMSE).

30-40 Datasets for Education Data Mining

[Dataset Webpage:] Go to the webpage https://pslcdatashop.web.cmu.edu/ (You may need log in through WebISO) and click "Public Datasets".

There are about 30-40 public datasets available from the webpage (Only select the datasets whose status is labeled "complete"). If you clicking each datasets, you will find a general description and the related publications. To look at the datasets, click "Export" link.

Project Idea 1:
For each dataset, you can compare various machine learning techniques (at least five to seven different ML methods) on predicting "Correct First Attempt values" (Generally listed in the column "Outcome"). Please report the Root Mean Squared Error (RMSE).

Project Idea 2:
Across datasets, you can compare several machine learning techniques (at least two to three different ML methods) on predicting "Correct First Attempt values"(Generally listed in the column "Outcome"). Please report the Root Mean Squared Error (RMSE) on the test data. The hypothesis here is that there may not be an absolute winner, different machine learning techniques may be effective on different task domains. For example, you can split the datasets into science (physics & math) vs. second language learning (Chinese, French).

Brain-Computer Interface data set

Description: This data set was used in the BCI Competition III (dataset V): http://www.bbci.de/competition/iii/ Using a cap with 32 integrated electrodes, EEG data were collected from three subjects while they performed three activities: imagining moving their left hand, imagining moving their right hand, and thinking of words beginning with the same letter. As well as the raw EEG signals, the data set provides precomputed features obtained by spatially filtering these signals and calculating the power spectral density.

Size:

31216 records in training data, 10464 in test data
Each record has 96 continuous values and a numerical label
63 MB as uncompressed text

Reference:

On the need for on-line learning in brain-computer interfaces by J. del R. Millán

Task: Perform exploratory data analysis to get a good feel for the data and prepare the data for data mining. Train at least two different classifiers to assign class labels to the test data to indicate which activity the subject was performing while the data were collected.

Challenges: This data set represents time series of EEG readings. A baseline approach could be based on the given precomputed features. It might also be possible to train a classifier on a window of some size around each time step. Both of these approaches ignore the fact that the data is really a time series; one might consider using an explicit time-series model such as a Hidden Markov Model.