Introduction to Machine Learning

CS4375 --- Fall 2018

Course Project

Overview

- 12% of course grade
- Spans the rest of the semester: you should start now
- Provides hands-on experience with employing machine learning techniques to tackle a prediction task
  - However, you may choose not to use machine learning
- Everyone will work on the same task
  - You may work in a group of two or individually

Task: Binary Prediction Task

- Training Data (Yours ---Complete---):
  - 5 sets of 1,800 instances = 9,000 instances
- Test set for preliminary evaluation (Ours ---Missing Class---):
  - 4,000 instances
- Test set for final evaluation(Ours ---Missing Class---):
  - 4,000 instances

- Class value: 0/1 (last column)
- 205 features
- .attr file includes the names and ranges of the features

Missing Values

- This is real data, so it contains missing values

- Two versions of the training sets and test set provided
  - In one set (mv), missing values are indicated by “?”
  - In the other set (nmv), the missing values have been calculated using a simple mean/median/mode procedure
  - You may do learning using one or both of these files, or you may fill in the missing values in whatever way you like

What you need to do

- Predict the class values of the test instances
  - You may use any algorithms and techniques you want, including those that are not introduced in this course
    - You may even employ non-learning-based methods
  - You may use any publicly available software packages

Key Dates

- Preliminary evaluation: Saturday, Nov 24
- Final evaluation: Sunday, Dec 9
- Project report: Thursday, Dec 13

Preliminary Evaluation (Due: Nov 24)

- 5% of the course grade

- The test set for the preliminary evaluation will be available on Nov 17 (one week before the deadline)

- Need to submit (1) your prediction file, and (2) a README file listing the names of all group members to eLearning

Prediction File Format

- The file should be plain text
- Each line should contain **only** the predicted class value (0 or 1) of a test instance
- # lines in prediction file should be the same as # lines in test file
- You must return predictions to us in the same order as the instances in the test file
- Sample prediction file:

  0

  0

  1

  …

Scoring

- We will compute the accuracy (i.e., percentage of correctly classified test instances) of the system submitted by each team

- Your score in the preliminary evaluation will be based on the normalized test accuracy of your system
  - Score =

$$\frac{accuracy - baseline\ accuracy}{max\ accuraxy - baseline\ accuracy}$$

More on Scoring

- Your score will depend on a) how much you can improve beyond our baseline accuracy, and b) how your results compare to other groups in the class.
- We will provide our baseline's accuracy for each of the 5 training sets (training on 4 sets and "testing" on the 5th), so you can get an idea of how good your system is before the preliminary testing phase.

Final Evaluation (Due: Dec 9)

- 5% of the course grade
- **Goal**: a second chance for you to improve your system after seeing how your team performs relative to the other teams
- The test set for the final evaluation will be available on Dec 2 (one week before the deadline)
- Need to submit (1) your prediction file; (2) your code; and (3) a README file containing instructions on how to compile and run your system, as well as a listing the names of all group members to eLearning
  - Prediction file format and scoring are the same as those in the preliminary evaluation

Project Report (Due: Dec 13)

- 2% of the course grade

- The project report should describe everything that the team did for the project. It should
  - include approaches that were attempted but were ultimately not employed because of their poor performance, for instance
  - the approach that was chosen for final evaluation
  - lessons learned

- More details on the project report later …

Data Sets

- Under the assignments section in the course website.

Challenges

- Parameter tuning
  - Almost all learning algorithms have their own set of parameters
    - E.g., for neural nets, we need to specify the number of hidden layers and the number of hidden units per layer
    - The performance of a learner is to a large extent determined by these parameters
    - You probably want to tune them on validation data

What else can you try?

- Different learning algorithms
  - You can employ multiple learning algorithms to make predictions
  - Some learners may perform better than others
    - You may consider discarding the bad ones or putting less weights on them
    - How can I determine which learners are bad?
      - Use your validation data
    - How do I know how much weight I should give to a learner’s prediction?
      - Use your validation data

Anything else?

- Feature selection
  - Motivation: using all the available features may not always yield better results than using a subset of them
  - While many learners can (implicitly) select features, in many cases it may be good to explicitly identify and filter out the irrelevant features
  - How can we select good features?
    - Information gain
    - Let the decision tree learner tell you
    - Other methods (consult the literature)

Summary

- You can use whatever approach you want to do make predictions
  - Be creative!

- You can even discuss your approach with other groups
  - But you are not allowed to share your team’s predictions with other groups