10-401 Machine Learning: Project

Due 12 noon Thursday, March 24, 2016

100 points

Instructions

- **Late homework policy:** Homework and project are worth full credit if submitted before the due date and zero credit after that. You have four late days to use throughout the entire semester. **Important note about late days:** The late day policy for the project is that, to use one late day for the project, each team member must apply one of their late days. As such, the effective number of late days a group can use is the **minimum remaining number of late days among team members**. However, Autolab does not enforce late days strictly (*i.e.*, you are allowed to submit after using your late days, but your grade will be penalized). This means that if a team member with more late days makes a submission beyond the allowable late days of their partner, their partner's grade will be penalized. Make sure to communicate with your partner about how many late days you would like to use or have available for the project!

- **Collaboration policy:** We recommend forming teams of two students on this project. You need to identify your teammate and create a group for the two of you on Autolab before **February 12**. We will randomly pair up unmatched students at that time. To create a group on Autolab, go to the project page, and click "Group options" on the left sidebar. Here you can set your group name and enter the Andrew email address of your partner. Your partner will need to accept your invitation before your group is official.

- **Online submission:** There will be three components to submission. First, to evaluate the accuracy of your predictions, you will submit a file to Autolab for your predictions on a test set. Second and third, at the final due date, you will make another submission to Autolab consisting of your code and a report.

  You will submit your solutions online on Autolab, just like for homework assignments. On the 10-401 Autolab page, there will be an assessment called "Project: crime prediction." Here you can form a team, make submissions, and see your position on the leaderboard. *Note:* The submission portal and leaderboard are not ready yet, but will be announced within a few days on piazza. Meanwhile, you can make yourself familiar with the dataset and try training/testing classifiers locally on your machine using random splits of the data.

  Your test submission must be a file named `p.csv.zip`, which, unsurprisingly, is a csv file named `p.csv`, compressed with zip. Refer to the project description below for a detailed description of how the csv file should look.

  Either member of a team can make a submission, and the submission will be linked to both team members' accounts. Once you make a submission, the autograder will score your submission and display your team name and score on the project leaderboard. Due to space constraints and the potential for overfitting as you get feedback on the evaluation dataset from your leaderboard position, **each team is limited to 20 submissions**.

  The final submission (due March 24), consisting of your code and written report, will also be submitted on Autolab. Each team should submit a file named `submission.tar`, containing a file named `code.tar` of all your code and a file named `report.pdf` with your project write-up. We will release a submission template for you to follow.

- **Policy on external resources:**
  - External machine learning packages, such as `libsvm`, `weka`, `sklearn` and `theano`, are allowed.
  - You may code in any of the following languages: Matlab/Octave, C/C++, Java, Python.
  - For fairness, we ask that both training and predicting to be done on *a single machine*, typically your laptops, even if some students could have access to cluster machines.
  - You might be able to find data sets online that contain data similar to the training/testing data we provided. **It is explicitly forbidden to use any such online data resource to help design, analyze or train your model. All training, validation and prediction must be done solely using the data sets we provided**. This rule is articulated more precisely in the *policy on reproducibility* section below.
  - If you are not sure whether certain external resource could be used, please contact the TAs via email or on piazza.
- **Policy on reproducibility:** All results should be reproducible, in that they can be reproduced in a timely manner on the data sets we provided. At the end of the semester we will check each team's code and ensure reproducibility of the results. Below are a few important items that we hope every team is aware of:
  - We expect the training and predicting procedure to end in a reasonable amount of time. In general, less than 24 hours is considered reasonable.
  - If you are using a randomized algorithm, please supply a fixed seed so that the behavior of the program is deterministic. If you are using an external learning package, be aware of the inherent randomness in that package and set random seeds accordingly. **As a rule of thumb, make sure running your program twice will produce identical outputs.**
  - You are only allowed to use the labeled training set and unlabeled testing set we provided as inputs to your training/predicting method. **Any use of external data is strictly prohibited, unless approved by the TAs or instructors**.
  - In your code you may set values of *up to five* parameters manually to "magic numbers", except common constants like 0, 1 or infinity (e.g., 1e100). For example, setting $C = 1000$ in a SVM model would be one parameter that is considered setting manually. Parameters or hyper-parameters that are selected using cross-validation on the training data set we provided are not subject to this constraint. However, in this case make sure that you can reproduce the parameter selection procedure on the provided data sets.

Project: predicting crime categories of events

Task description

You are given a data set that consists of many "events". Each event corresponds to one particular category of crime (e.g., theft or murder) or not crime behavior (non-crime). You are request to predict the crime category of each event based on information like event time, day of week, local police district, address and (X, Y) coordinates, etc.

The training data set consists of $100,000$ events, each belonging to one of 39 crime categories (including the non-crime category). Each event is described with the following information:

- *Date*: the date when the event occurs. E.g., 2015-05-13.
- *Day of week*: e.g., Monday, Wednesday.
- *Police district*: which local police office is in charge of the specific event. E.g., RICHMOND, CENTRAL.
- *Address*: the address where the event occurs; e.g., Oak St/ Laguna St.

- *X, Y*: longtitude and latitude of where the event occurs; e.g., -122.4259, 37.7746

The testing data set consists of 40,000 events, each with the above-mentioned information but without crime category labels. Your task is to predict the crime category of each event in the testing data set. For each event, your prediction should be a probability distribution over the 39 crime categories, with higher probability indicating that you are more confident of a particular category. For example, a prediction of (theft, 0.8; robbery, 0.2) means that the event is most likely to be theft, less likely to be robbery, and not possible to be other crimes such as murder or rape.

Data formats

The training data is in csv format. It has 100,000 lines, each corresponding to a training event. Each line contains 7 comma-separated fields: date, category, day of week, police district, address, X and Y. Below is an example line in the training file:

`2015-05-13 21:17:00,ROBBERY,Wednesday,INGLESIDE,1600 Block of VALENCIA ST, -122.420272135283,37.7473316298785`

The testing data is also in csv format. It has 40,000 lines, each corresponding to a testing event. Each line contains 6 comma-separated fields in the same order of the training file, except for the *category* field which is to be predicted. Below is an example line in the testing file:

`2015-05-10 23:59:00,Sunday,BAYVIEW,2000 Block of THOMAS AV,-122.39958770418998,37.7350510103906`

The file you are submitting consists of 40,000 lines; each line contains your predicted probability for the category of the corresponding event in the testing file. There should also be a header line, which is a list of comma separated names of crime categories. The header line is provided in the sample submission file (distributed on piazza) and you should not change the order of the crime categories in your submission. On each line, you should output 39 comma-separated real numbers that are between zero and one. It is not required that they sum to one: we will do the normalization in the testing script. **Your submission must contain exactly 40001 lines, with the first line exactly the same with the first line in the sample submission file and on each of the remaining lines there must be exactly 39 real numbers, separated by 38 commas. Submission with incorrect formats will not be graded.**

The training file, testing file and sample submission file will be distributed on piazza.

Evaluation criterion [75 points]

The testing data set will be split uniformly at random into two sets with equal size (i.e., 20,000 events each). You will not know how we split it. For each of your submission to Autolab, one in the two split sets will be used and your score is computed as

$$\text{score} = -\frac{1}{n}\sum_{i=1}^{n}\ln(p_{c_i}),$$

where c_i is the ground-truth category of the ith event and p_{c_i} is the probability you provided for this crime category of event i. $n = 20000$ is the total number of events in the test split. Clearly, smaller score means better accuracy. Everyone's submission will be tested on the same testing set.

By the due date, you are required to submit a final prediction file and the other test split is used to compute the score of your prediction. Your final score out of the 75 points will be 40% of your best score the Autolab reports, plus 60% of the score of your final prediction on the other held-out test set.

Final report [25 points]

Your final report should be 2 to 3 pages, formatted by LaTeXusing NIPS template. The NIPS template can be downloaded at https://nips.cc/Conferences/2015/PaperInformation/StyleFiles.

The final report should have the following components:

1. Include a detailed description of your algorithm and procedure of training/testing, including procedures of selecting algorithm parameters.

2. Include justification of your choice and why it worked/did not work.

3. Include necessary plots/experiments that support your choice/design of the learning algorithm and parameter choices used.