MLSS Coding Contest

CMU Summer 2014

DEADLINE: 0900 Friday July 18

Email: cmdowney@cs.cmu.edu if you have any questions

Welcome to the 2014 MLSS Classification Contest! In this contest you will have a chance to practice the algorithms you have learned over the last two weeks, as well as a chance to demonstrate your machine learning prowess.

In this contest your goal is to write a *Classifier.* This classifier should take as input a set of feature vectors, and produce as output a set of class labels. You are given a set of labelled training data to use to train your classifier. Your classifier will be evaulated on a held out test set.

There are three twists to this contest:

1. There are a large number of class labels - the dataset you'll be working with has several hundred class labels. This means you'll have to think outside the box, as traditional binary classification algorithms will not work.

2. The class labels are heirarchical - in other words they belong to a heirarchy. The structure of this heirarchy is also provided to you and you are welcome to use this information to assist with classification. Note that the heirarchy forms a Directed Acyclic Graph.

3. Instances can have more than one label. In other words each instance can have anywhere from $1 - n$ labels, where n is the number of distinct labels. Your classifier needs to predict all of the labels for each unlabelled instance.

IMPORTANT Before proceding, please log on to the Autolab system at http://identity.andrew.cmu.edu using the username and temporary password provided and change your password. **If you do not do this your temporary password will expire in 5 days**.

1 Dataset

In this contest you will work with *Text Data* in the form of a set of wikipedia articles. For your convenience these articles have already been featurized: each article has been converted to a list of stemmed word counts for a dictionary of words. For example if the dictionary consists of the 3 words "I", "like", and "apple", and the first document was "apple apples I", then after featurizing it would be represented as the feature vector [1 0 2].

The dataset has the following properties:

- 17238 instances, split evenly between the training set and the test set.
- 346299 features (stemmed words).
- 419 classes in the test/train data.
- 2722 classes in the heirarchy.

The data is provided in the following files:

- XTrain.mat this file contains the training data in an 8619×346299 matrix X of real numbers. $X_{i,j}$ is the number of times word j appears in document i.
- yTrain.mat this file contains the training labels in a 8619×419 matrix y of booleans. $y_{i,j}$ is 1 if document i has label j, and zero otherwise. Note that every instances has at least one label.
- h.mat this file contains the heirarchy information in a 2722×2 matrix h of integers. This is a list of parent-child relationships. If h contains the line $[1, 2]$ then label 1 is the parent of label 2 in the heirarchy.

2 Coding and Submission

You will modify and submit two files: classify.m and parameters.txt. classify.mat should contain your code, while parameters.txt is an empy file you are free to do with as you want. We provide it so that you can upload any trained parameters for your model. Please see below for more details:

- **Octave:** You must submit your code in Octave (though you can use other environments to develop the algorithms). Octave is a free scientific programming language, with syntax identical to that of MATLAB. Installation instructions can be found on the Octave website. (You can develop your code in MATLAB if you prefer, but you *must* test it in Octave before submitting, or it may fail in the autograder.)

- **Autograding:** This problem is autograded using the CMU Autolab system. The code which you write will be executed remotely against a suite of tests, and the results used to automatically assign you a score. To make sure your code executes correctly on our servers, you should avoid using libraries which are not present in the *basic* Octave install.

- **Coding Stubs:** In the code handout, we have provided you with a single folder containing the two files classify.m and parameters.txt. If there are n instances in XTest, and m distinct labels, then classify.m should output a $n \times m$ binary matrix. Complete classify.m to create your classifier. Store any trained parameters in parameters.txt *Do not modify the structure of this directory or rename these files.* When you download the files, you should confirm that the autograder is functioning correctly by submitting the stubs. This should result in an extremely low score (The stub predicts each unlabelled instance belongs to every class).

- **Submission:** Submit your work at Submission Website. Log on to this website using the username provided and your new password. To submit your work, tar the code directory provided (e.g. ”tar -cvf code.tar code” from the terminal) and upload the resulting tar file using the ”hand in your work” option. **Do not submit the data files**

- **Scoring:** Your classifier will be ranked based on its F1 score. The F1 score takes into account both the recall (number of class labels correctly identified) and the precision (number of correct class labels divided by the total number of class labels predicted). For example if your classfiier predicts every label for every instance it will get perfect recall, but terrible precision. To get a high F1 score you need high recall and high precision.

- **Leaderboard:** You can view the current leaderboard for the contest at: Leaderboard.

- **SUBMISSION CHECKLIST**
  - Submission executes on our machines in less than 20 minutes.
  - Submission is smaller than 5MB.
  - Submission is a `.tar` file.
  - Submission returns a binary matrix of the correct size.