Homework 3
Machine Learning

Due date: Sunday, April 8, 11:59 p.m.

1 Collaborative Filtering on Netflix Ratings (40 points)

- Read the paper Empirical Analysis of Predictive Algorithms for Collaborative Filtering linked on the course web page. You need to read up to Section 2.1, and are encouraged to read further if you have time.

- The dataset we will be using is a subset of the movie ratings data from the Netflix Prize. You need to download it from the course web page. It contains a training set, a test set, a movies file, a dataset description file, and a README file. The training and test sets are both subsets of the Netflix training data. You will use the ratings provided in the training set to predict those in the test set. You will compare your predictions with the actual ratings provided in the test set. The evaluation metrics you need to measure are the Mean Absolute Error and the Root Mean Squared Error. The dataset description file further describes the dataset, and will help you get started. The README file is from the original set of Netflix files, and has been included to comply with the terms of use for this data.

- Implement the collaborative filtering algorithm described in Section 2.1 of the paper (Equations 1 and 2; ignore Section 2.1.2) for making the predictions. You may program in C, C++, Java or Python.

2 Boosting and Bagging [30 points]

In this part of the homework, you will experiment with Bagging and Boosting. Bagging and AdaboostM1 are available under the "Meta" category in WEKA. Use the following settings:

- Choose three classifiers of your choice. Example: J48, Logistic regression, Decision stump, etc. If you are using decision trees, turn pruning ON.
- Choose three datasets from the UCI machine learning repository. Note that the following set up does not work with all datasets available there and therefore you will have to choose the three quite carefully. The larger the data size the better (but be reasonable, don't try very large datasets).
- For Bagging, set numIterations to 30.
- AdaboostM1: set maxIterations to 30. Set weightThreshold to 100000.

You will run the three classifiers by themselves (Vanilla) and then with bagging and boosting on each of the three datasets and report results for 10-fold cross validation in the following table:

```
Dataset1:
Base learner     Vanilla       Bagging        Boosting
Classifier1       xxx          yyy            zzz
Classifier2       xxx          yyy            zzz
Classifier3       xxx          yyy            zzz

Dataset2:
Base learner     Vanilla      Bagging        Boosting
Classifier1       xxx          yyy            zzz
Classifier2       xxx          yyy            zzz
Classifier3       xxx          yyy            zzz

Dataset3:
Base learner     Vanilla       Bagging        Boosting
Classifier1       xxx          yyy            zzz
Classifier2       xxx          yyy            zzz
Classifier3       xxx          yyy            zzz
```

where xxx, yyy and zzz are the error rates; replace them by the error rates that you get. Replace Classifier1, Classifier2, Classifier3 and Dataset1, Dataset2 and Dataset3 by the specific classifiers and datasets chosen.

Repeat the experiment for 2 other settings for number of iterations: 100 and 150.

Answer the following questions (5 points each):

1. Which algorithms+data set combination is improved by Bagging?
2. Which algorithms+data set combination is improved by Boosting?
3. Can you explain these results in terms of the bias and variance of the learning algorithms applied to these domains? Are some of the learning algorithms unbiased for some of the domains? Which ones?

3 K-means clustering on images [30 points]

In this problem, you will use K-means clustering for image compression. We have provided you with two images.

- Display the images after data compression using K-means clustering for different values of K (2, 5, 10, 15, 20).
- What are the compression ratios for different values of K? Note that you have to repeat the experiment multiple times with different initializations and report the average as well as variance in the compression ratio.
- Is there a tradeoff between image quality and degree of compression. What would be a good value of K for each of the two images?

We have provided you java template KMeans.java which implements various image input/output operations. You have to implement the function kmeans in the template. See the file for more details.

What to turn in for this question:

- Your source code for the kmeans algorithm.
- A report containing your write up and plots.

Note that your program must compile and we should be able to replicate your results. Otherwise no credit will be given.