# Homework 4
# CS 6375: Machine Learning

Spring 2012

Due date: Wednesday, April 16, midnight

## 1 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 that implements various image input/output operations. You have to implement the function kmeans in the template. See the file for more details. (If you are not comfortable with Java, you are free to use a language of your choice. But you will have to write the support code for manipulating images yourself.)

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.

## 2 EM algorithm [40 points]

In this problem, you will use the EM algorithm and Gaussian mixture models to cluster the data into exactly three classes (or clusters). As discussed in class, in this approach, we assume a Gaussian mixture model with $K$ components for the data and we find the parameters of the model using maximum likelihood. The equations for updating are provided on the class Web page and in your textbook (Bishop, Chapter 9).

- Download the data from the class website.
- Implement the EM algorithm for general Gaussian mixture models (assume that the data is an array of doubles or long doubles). Use the algorithm to cluster the given data. I recommend that you run the algorithm multiple times from a number of different initialization points (different $\theta^0$ values) and pick the one that results in the highest log-likelihood (since EM in general only finds local maxima). One heuristic is to select $r$ different randomly-chosen initialization conditions. For example, for each start, select the initial $K$ Gaussian means by randomly selecting $K$ initial data points, and select the initial $K$ covariances as all being some multiple of the overall data covariance—the selection of initial covariances is not as critical as the initial means). Another option for initialization is to randomly assign class labels to the training data points and then calculate $\theta^0$ based on this initial random assignment (or begin the iterations by executing a single M-step, which is also fine).

  Report the parameters you get for different initializations. What initialization strategy did you use? How sensitive was the performance to the initial settings of parameters.
- Now assume that variance equals 1.0 for all the three clusters and you only have to estimate the means of the three clusters using EM. Report the parameters you get for different initializations. Which approach worked better, this one or the previous one.

What to turn in for this part:

- Your code. EM for general GMMs and EM for GMMs with known variance.
- A report containing answers to the questions above.

# 3 Boosting [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 WEKA's data directory. Note that the following set up does not work with all datasets available there and therefore you will have to choose the three carefully.
- For Bagging, set numIterations to 30.
- AdaboostM1: set maxIterations to 30. Set weightThreshold to a reasonable value and report what you used.

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:

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?

What to turn in:

1. A report containing the tables and answers to the three questions posed above. Each table and question is worth five points.