COMS 4721: Machine Learning for Data Science

Columbia University, Spring 2021

Homework 3: Due April 1, 2021 by 11:59pm

Please read these instructions to ensure you receive full credit on your homework. Submit the written portion of your homework as a *single* PDF file through Courseworks (less than 5MB). In addition to your PDF write-up, submit all code written by you in their original extensions through Courseworks (e.g., .m, .r, .py, .c). Any coding language is acceptable, but do not submit notebooks. Also, do not wrap your files in .rar, .zip, .tar and do not submit your write-up in .doc or other file type. When resubmitting homeworks, please be sure to resubmit *all files*. Your grade will be based on the contents of one PDF file and the original source code. Additional files will be ignored. We will not run your code, so everything you are asked to show should be put in the PDF file. Show all work for full credit.

Late submission policy: Late homeworks will have 0.1% deducted from the final grade for each minute late. *Your homework submission time will be based on the time of your **last** submission to Courseworks. I will not revert to an earlier submission time!* Therefore, do not re-submit after midnight on the due date unless you are confident the new submission is significantly better to overcompensate for the points lost. Submission time is non-negotiable and will be based on the time you submitted your last file to Courseworks. The number of points deducted will be rounded to the nearest integer.

Problem 1 (Boosting coding) – 30 points

In this problem you will implement boosting for the least squares classifier that we briefly discussed in Lecture 8. Recall that this "classifier" performed least squares linear regression by treating the ± 1 labels as if they were real-valued responses. Also recall that we criticized this classifier as being not a very good one to use in practice (i.e., "weak") on its own, and so boosting this classifier can be a good illustration of the method.

Using the toy data provided, implement the AdaBoost algorithm on the least squares classifier. You should use the bootstrap method as discussed in the slides to do this, where each bootstrap set $\mathcal{B}_t$ is the size of the training set. In the data, I have added a dimension equal to 1 for the intercept term. Recall that if the value of $\epsilon_t > 1/2$, you can simply change the sign of the regression vector you learned in iteration t (including the offset term) and recalculate to make $\epsilon_t < 1/2$.

a) Run your boosted least squares classifier for $T = 2500$ rounds and plot the empirical training error of $f_{boost}^{(t)}(\cdot)$ for $t = 1, \ldots, T$. In the same plot, show the upper bound on the training error as a function of t. (This upper bound is given in the slides for Lecture 13.)

b) Show a stem plot of the average of the distribution on the data across all 2500 iterations. (The empirical average of w_t in the slides over t.)

c) In two separate figures, plot ϵ_t and α_t as a function of t.

Problem 2 (K-means) – 15 points

Implement the K-means algorithm discussed in class. Generate 500 observations from a mixture of three Gaussians on $\mathbb{R}^2$ with mixing weights $\pi = [0.2, 0.5, 0.3]$ and means μ and covariances Σ,

$$\mu_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \mu_2 = \begin{bmatrix} 3 \\ 0 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \mu_3 = \begin{bmatrix} 0 \\ 3 \end{bmatrix}, \Sigma_3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

a) For $K = 2, 3, 4, 5$, show on the same plot the value of the K-means objective function per iteration for 20 iterations (the algorithm may converge before that).

b) For $K = 3, 5$, plot the 500 data points and indicate the cluster of each for the final iteration by marking it in some way.

Problem 3 (Bayes classifier revisited) – 30 points

In this problem, you will implement the EM algorithm for the Gaussian mixture model, with the purpose of using it in a Bayes classifier. The data is a processed version of the spam email data you looked at in Homework 2. Now, each labeled pair (x, y) has $x \in \mathbb{R}^{10}$. We discussed how the Bayes classifier learns class-conditional densities, and unsupervised learning algorithms can be useful here. In this problem, the class conditional density will be the Gaussian mixture model (GMM). In these experiments, please initialize all covariance matrices to the empirical covariance of the data being modeled. Randomly initialize the means by sampling from a single multivariate Gaussian where the parameters are the mean and covariance of the data being modeled. Initialize the mixing weights to be uniform.

a) Implement the EM algorithm for the GMM described in class. Using the training data provided, for each class separately, plot the log marginal objective function for a 3-Gaussian mixture model over 10 different runs and for iterations 5 to 30. (In other words, don't show iterations 1 through 4.) There should be two plots, each with 10 curves.

b) Using the best run for each class after 30 iterations, predict the testing data using a Bayes classifier and show the result in a 2×2 confusion matrix, along with the accuracy percentage. Repeat this process for a 1-, 2-, 3- and 4-Gaussian mixture model. *Show these results nearby each other. You don't need to repeat Part (a) for these other cases.* Note that a 1-Gaussian GMM doesn't require an iterative algorithm, although your implementation will likely still work in this case.