Homework #6

CS 260: Machine Learning Algorithms
Prof. Ameet Talwalkar
Due: 12/4/15, noon
Please abide by the Academic Integrity Policy

1 Clustering

Given a set of data points $\{\mathbf{x}_n\}_{n=1}^N$, the k-means clustering minimizes the following distortion measure (or objective or clustering cost):

$$D = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|_2^2$$

where $\boldsymbol{\mu}_k$ is the prototype of the k-th cluster and r_{nk} is a binary indicator variable. If $\mathbf{x}_n$ is assigned to the cluster k, r_{nk} is 1 otherwise r_{nk} is 0. For each cluster, $\boldsymbol{\mu}_k$ is the representative for all the data points assigned to that cluster.

a. In the lecture, we stated but did not prove that $\boldsymbol{\mu}_k$ is the mean of all points associated with the kth cluster, thus motivating the name of the algorithm. You will now prove this statement. Assuming all r_{nk} are known (i.e., assuming you know the assignments of all N data points), show that if $\boldsymbol{\mu}_k$ is the mean of all data points assigned to cluster k, for any k, then the objective D is minimized. This justifies the iterative procedure of k-means[1].

b. We now change the distortion measure to

$$D = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|_1$$

In other words, we use the L_1 norm ($\|\mathbf{z}\|_1 = \sum_d |z_d|$) to measure the "closeness" of each point to the cluster prototypes. Under this new cost function, and again assuming that all r_{nk} are known, show that the $\{\boldsymbol{\mu}_k\}_{k=1}^K$ which minimize D are the elementwise medians of all data points assigned to the k-th cluster. Note that the elementwise median of a set of vectors is defined as a vector whose d-th element is the median of all vectors' d-th elements.

2 Mixture Models

Suppose $X_1, \ldots, X_N$ are i.i.d random variables with the density function $f_X(x, \lambda) = \lambda e^{-\lambda x}$, for $x \geq 0$ and 0 otherwise. We observe $Y_i = \min\{X_i, c_i\}$ for some fixed and known c_i. Assume (for simplicity) that this thresholding occurs only for the last $n - r$ variables; i.e. $Y_i = c_i$ for $i = r+1, \ldots, N$. The goal is to estimate the value of λ using EM algorithm.

(a) Write the log-likelihood in terms of unobserved variables X_i.

[1] More rigorously, one would also need to show that if all $\boldsymbol{\mu}_k$ are known, then r_{nk} can be computed by assigning $\mathbf{x}_n$ to the nearest $\boldsymbol{\mu}_k$. You are not required to do so.

(b) Write down the E-Step and take the expectation, i.e., compute $Q(\lambda|\lambda^t)$. Hint: You may want to use the memorylessness property of the exponential distribution.

(c) Write down the M-Step and calculate the value of λ^{t+1}.

3 Eigenfaces

Face recognition is an important task in computer vision and machine learning. In this question you will implement a classical method called Eigenfaces. You will use face images from the `Yale Face Database B` (`http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html`) which contains face images from 10 people under 64 lighting conditions.

a. **Dataset**. Download the data file `face_data.mat`. It contains three sets of variables:

- `image`: each element is a face image (50×50 matrix). You can use matlab function `imshow` to visualize the image. The data is stored in a cell array.
- `personID`: each element is the ID of the person, which takes values from 1 to 10.
- `subsetID`: each element is the ID of the subset which takes values from 1 to 5. Here the face images are divided into 5 subsets. Each subset contains face images from all people, but with different lighting conditions.

b. **Implement PCA**. Fill in the function `pca_fun` in the `pca_fun.m` file. The function takes the data matrix (each row being a sample) and target dimensionality d (lower than or equal to the original dimensionality) as the input, and outputs the eigenvectors.

c. **Compute Eigenfaces**. Take each 50×50 training image and vectorize it into a 2500-dimensional vector. Perform PCA on all vectorized face images, and retain the first $d = 200$ eigenvectors. These eigenvectors are called *eigenfaces* (when displayed as images). Please display the top 5 eigenfaces (use `imshow`) in your report.

d. **Classification**. First project each image into the eigenspace to obtain a d-dimensional vector. For each $d \in \{20, 50, 100, 200\}$, train a classifier and report its classification performance. Evaluate classification performance using the *leave-one-subset-out* strategy: treat each subset as the test set and the remaining four subsets as the training set, and then report the average accuracy. Please experiment with the linear SVM (use LIBSVM toolbox `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`), tuning the hyperparameter C (pick an appropriate range of values to choose among). Note that when tuning this hyperparameter, you should also apply the *leave-one-subset-out* strategy on the training set. To summarize, you should report (a) the optimal parameter for linear SVMs (b) average test accuracy for each $d \in \{25, 100, 200\}$.

Submission

Please provide the following as part of your submission:

- Provide your answers to problems 1, 2, 3c, and 3d in hardcopy. The papers need to be stapled and submitted at the beginning of section on the due date.
- Please put all of your code in a single folder named `[lastname]_[firstname]_hw6`, and submit a single `.zip` file containing this folder called `[lastname]_[firstname]_hw6.zip`. The only acceptable languages are MATLAB and Octave.

- You MUST include the main function called `CS260_hw6.m` in your root folder in your zip file. After running this main file, your program should be able to generate all of the results needed for this programming assignment, either as plots or console outputs. You can have multiple files (i.e your sub-functions), however, we require that once we unzip your folder and execute your main file, your program should execute correctly. Please double-check your program before submitting.
- Please submit this zip file by the due date by following the instructions on this form.
- You are encouraged to collaborate, but collaboration must be limited to discussion only and you need to write down / implement your solution on your own. You also need to list with whom you have discussed the HW problems.