CS1675: Homework 2

Due: 9/13/2018, 11:59pm

This assignment is worth 50 points.

In this assignment, you will implement the K-means algorithm. You will then use it to perform image clustering, to test your implementation.

Part I: Clustering (25 points)

Write a function `my_kmeans.m` to implement a basic version of the K-means algorithm.

Inputs: [5 pts for correct format of input/output]

- an NxD data matrix `A` where N is the number of samples and D is the dimensionality of your feature representation,
- the number `K` denoting how many clusters to output, and
- a value `iters` saying how many iterations to run for K-means.

Outputs:

- an Nx1 output `ids` containing the data membership IDs of each sample (denoted by indices randing from 1 to K, where K is the number of clusters),
- a KxD matrix `means` containing the mean/center for each cluster, and
- a scalar `ssd` measuring the final SSD error of the clustering, i.e. the sum of the squared distances between points and their assigned means, summed over all clusters.

Instructions:

1. [5 pts] First, initialize the cluster means randomly. Get the range of the feature space, separately for each feature dimension (compute max and min and take the difference) and use this to request random numbers in that range. Check the documentation for `rand`.
2. [5 pts] Then, iterate over the following two steps. The first step is to compute the memberships for each data sample. Use Matlab's function `pdist2` to efficiently compute distances (check its documentation to see what inputs it expects). Then for each sample, find the min distance and the cluster that gives this min distance.
3. [5 pts] The second step is to recompute the cluster means, simply taking the average across samples assigned to that cluster, for each feature dimension.
4. [5 pts] Finally, compute the overall SSD error. It helps to keep track of the min distance per sample as you iterate.

Part II: Random restarts (5 points)

K-means is sensitive to the random choice of initial clusters. To improve your odds of getting a good clustering, implement a wrapper function `restarts.m` to do `R` random restarts and return the clustering with the lowest SSD error.

Inputs: same as for `my_kmeans.m`, plus

- a scalar `R` denoting how many random restarts to perform.

Outputs: same as for `my_kmeans.m`, but

- `ssd` is the lowest SSD across all random restarts.

Part III: Image segmentation using clustering (20 points)

You will next test your implementation by applying clustering to segment and recolor an image. Write your code in a script `segment.m`.

1. [5 pts] Download the following images: `panda`, `cardinal`, and `pittsburgh`. Load them in Matlab using `im = imread(filename);`. This will return a HxWx3 matrix per image, where H and W denote height and width, and the image has three channels (R, G, B). Convert the image to `double` format. To avoid a long run of your code, downsample the images (reduce their size) e.g. using `im = imresize(im, [100 100]);`.
2. [5 pts] To perform segmentation, you need a representation for every image pixel. We will use a three-dimensional feature representation for each pixel, consisting of the R, G and B values of each pixel. Use `im = reshape(im, H*W, 3);` to convert the 3D matrix into a 2D matrix with pixels as the rows and channels (features) as the columns. Use the random restarts function you wrote above, to perform clustering over the pixels of the image.
3. [5 pts] Then recolor the pixels of each image according to their cluster membership. In particular, replace each pixel with the average R, G, B values for the cluster to which the pixel belongs (i.e. recolor using the cluster means). Show the recolored image using `imshow`, but convert it to format `uint8` before displaying.
4. [5 pts] Experiment with at least five different combinations of settings for `K, iters, R`. Write a brief report (`report.pdf` or `report.docx`) documenting your findings about these, and include the image results inside the document.

Submission: Please include the following files:

- `my_kmeans.m`
- `restarts.m`
- `segment.m`
- `report.pdf/docx`