Project 1 Report

Vishi Jhalani, Srajan Paliwal, Brianna Posadas, and Rachit Ranjan

April 9, 2018

1 Member Roles

	Cluster Alg	Eval Alg	Cluster Test	Eval Test	Cluster Write	Eval Write	Cluster Edit	Eval Edit
Brianna Posadas	1			1	1			1
Rachit Ranjan	2			2	2			2
Vishi Jhalani		3	3			3	3	
Srajan Paliwal		4	4			4	4	

2 Experiments

2.1 RGB Experiments and Tabular Results

The clustering algorithms were tested on a set of 198 images with corresponding reference images, i.e. each RGB image had a ground truth segmented image to compare the performance of the clustering algorithm with. To quantify the performance of each algorithm, the object-level consistency error (OCE) was calculated. OCE is an improvement over the Martin index because it is optimized for images with exact segment boundaries and sizes. First the partial error is calculated between the two images using equation (6) below from M. Polak, H. Zhang and M. Pi, "An evaluation metric for image segmentation of multiple objects," Image Vision Comput., vol. 27, (8), pp. 1223-1227, 2009.

$$
\begin{aligned}
E_{g,s}(I_g, I_s) &= \sum_{j=1}^{M} \left[1 - \sum_{i=1}^{N} \frac{|A_j \bigcap B_i|}{|A_j \bigcup B_i|} \times W_{ji} \right] W_j, \\
W_{ji} &= \frac{\bar{\delta}(|A_j \bigcap B_i|)|B_i|}{\sum_{k=1}^{N} \bar{\delta}(|A_j \bigcap B_k|)|B_k|}, \\
W_j &= \frac{|A_j|}{\sum_{l=1}^{M} |A_l|},
\end{aligned}
\tag{6}
$$

In the equations above, $I_g = A_1, A_2, ..., A_M$ where A are the fragments that make up the reference image I_g, $I_s = B_1, B_2, ..., B_N$ where B are the fragments that make up the segmented image I_s, and δ is the delta function. The partial error utilizes the Jaccard index to better penalizes over- and under- segmentation by the clustering algorithm. Also, the partial error equation weighs the calculated error from the different segments proportionally by the relative size of the segment with the terms W_{ji} and W_j. The OCE is then calculated using equation (7) from the same paper as illustrated below.

$$OCE(I_g, I_s) = \min(E_{g,s}, E_{s,g}). \quad (7)$$

The OCE gives a value from 0 to 1, where 0 is no error. The average and standard deviation of the OCE values from all 198 images are in the table below. The values are multiplied by 1000 to make comparisons easier. Overall, GMM performed the best on the test images.

Algorithm:	k-means mean	k-means std. dev.	fcm mean	fcm std. dev.	spectral mean	spectral std. dev.	GMM mean	GMM std. dev.	SOM mean	SOM std. dev
Score:	797	73	804	64	808	54	759	92	794	78

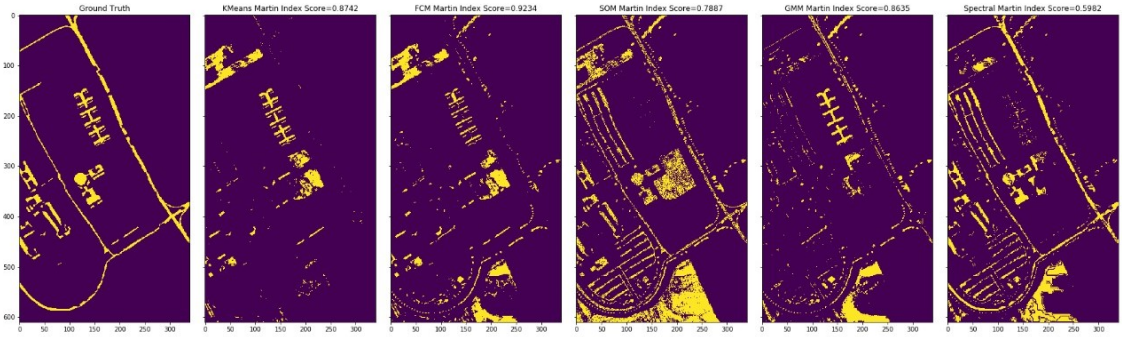
2.2 Hyperspectral Experiments

2.2.1 Pavia Hyperimage

For the Pavia hyperspectral image, our algorithm reduces dimensionality using Principle Component Analysis to a configurable value (currently 8 stored in appConfig.ini). It was observed that any higher than 8 dimensions didn't affect the clustering accuracy while any fewer than 4 dimensions reduced the accuracy when carried out on Pavia and compared with partial ground truth.

The hyperspectral image was then clustered into 9 clusters using KMeans, FCM, SOM, GMM, and spectral clustering. 9 clusters was chosen as the ground truth image was given with 9 clusters. The ground truth image and the resulted clustered images are in the figure below. The output labels were bitwise ANDed with Pavia ground truth mask and then compared to the ground truth using the OCE from the Martin index as explained in section 2.1. These scores are summarized in the table below. In similar fashion as section 2.1, the scores were scaled up by 1000 and a lower score indicates fewer errors. Using this evaluation method, it was found that spectral clustering was better able to match the clustering of the

ground truth image.

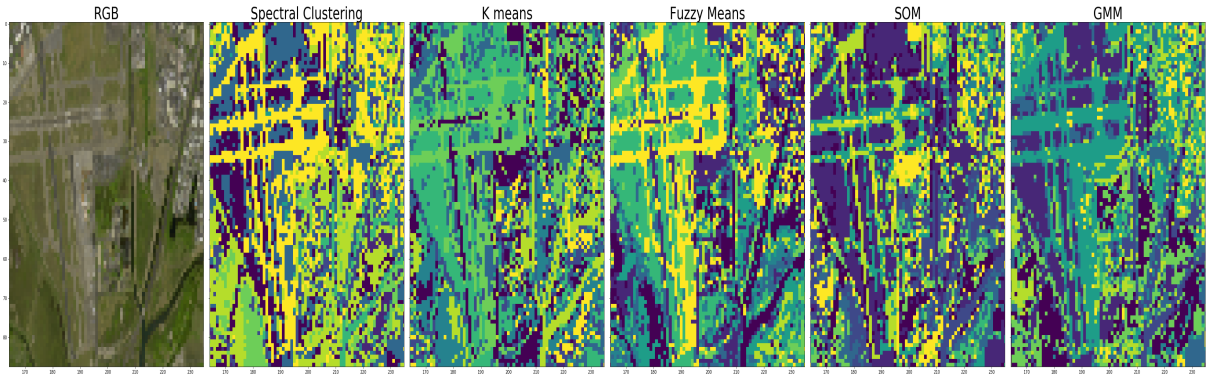


Algorithm	Martin Index Score
Spectral Clustering (Best)	598
SOM	789
GMM	864
K Means	874
FCM	923

2.2.2 Santa Barbara Hyperimage

Because the hyperspectral image from Santa Barbara did not include ground truth data, evaluation of the accuracy of the clustering algorithms were conducted subjectively. First, the hyperspectral image was reduced to 5 dimensions using PCA. It was found, through trial and error, that 5 dimensions was the lowest dimensions the image could be reduced to before data become lost. The results of the clustering algorithms are in the figure below. Comparing the clustered images with the original RGB image, it appears that Fuzzy Kmeans was able to distinguish between the roads, bodies of water, and grassy lands the best among the different clustering algorithms.

To continue ranking the clustering algorithms, subjectively, the next best performing algorithms in order would be SOM, GMM, spectral clustering, and k means. For this image, downsampling was not necessary because the image size itself was already small enough for spectral clustering to run fairly quickly, about 5 minutes. In this case, spectral clustering was able to edge out k means in results.



3 Observations

3.1 K-means advantages and disadvantages

K-means, as a data-clustering algorithm, is ideal for discovering globular clusters where all members of each cluster are in close proximity to each other (in the Euclidean sense). One of the main advantages for K-means is that it is computationally fast. It is relatively faster than other clustering methods when tested on RGB images. For instance with cluster size of 50, K-means took 6.63 minutes to cluster Pavia RGB while FCM took over 32 minutes.

Disadvantages for K-means include the difficulty of pre dicting cluster size K, especially when ground truth is unknown. In case of Pavia RGB partial ground truth and ground truth mask were available. We carried out K-Means on Pavia RGB with different cluster sizes comparing it with the partial ground truth using Structural Similarity. Choosing an elbow point in such a curve would serve well as a value for K. However this can't be generalized for any input image or even the same image with a few augmentations. De terming the value of K is indeed an interesting question to answer. Lastly, K-means was found to be sensitive to outliers; while testing the K-means code with dataset containing outliers, the evaluated centers and cluster labels were quite erroneous.

3.2 FCM advantages and disadvantages

Fuzzy clustering is a form of clustering in which each data point can belong to more than one cluster. This method differs from the k-means objective function by the addition of the membership values and the fuzzifier. The fuzzifier m determines the level of cluster fuzziness. A large m results in smaller membership values, and hence, fuzzier clusters. In the limit m=1, the memberships converge to 0 or 1, which implies a crisp partitioning. In the absence of experimentation or domain knowledge, m is commonly set to 2. The algorithm minimizes

intra-cluster variance as well, but has the same problems as k-means; the minimum is a local minimum, and the results depend on the initial choice of weights.

FCM produces close results to K-Means clustering but it still requires more computation time than K-Means because of the fuzzy measures calculations involvement in the algorithm. In fact, FCM clustering which constitute the oldest component of software computing are really suitable for handling the issues related to understand ability of patterns, incomplete/noisy data, mixed media information, human interaction and it can provide approximate solutions faster. They have been mainly used for discovering association rules and functional dependencies as well as image retrieval. So, overall conclusion is that K-Means algorithm seems to be superior than Fuzzy C-Means algorithm.

3.3 Spectral Clustering advantages and disadvantages

In spectral clustering, we don't cluster data points directly in their native data space like K-means, but instead, form a similarity matrix where the (i,j)−th entry is some similarity distance you define between the i−th and j−th data points in your dataset. So, in a sense, spectral clustering is more general (and powerful) because whenever K-means is appropriate for use then so too is spectral clustering (just use a simple Euclidean distance as the similarity measure). The converse is not true, though.

There are also practical considerations to keep in mind when choosing one of these methods over the other. With K-means you factorize the input data matrix, while with spectral clustering you factorize the Laplacian matrix (a matrix derived from the similarity matrix. Why does it matter? Say you have P data points each with N dimensions/features. Then using K-means you'll be dealing with an N by P matrix, while the input matrix to spectral clustering is of size P by P. The practical implications are: spectral clustering is indifferent to the number of features you use (Gaussian kernel which can be thought of as an infinite-dimensional feature transformation is particularly popular when using spectral clustering). However, it is difficult applying spectral clustering (at least the standard version) to very large datasets (large P). This causes the runtime of the algorithm to be longer than the other clustering algorithms.

3.4 Gaussian Mixture Models advantages and disadvantages

An advantage of GMM is it is the fastest algorithm for learning mixture models. Also, as this algorithm maximizes only the likelihood, it will not bias the means towards zero, or bias the cluster sizes to have specific structures that might or might not apply; thus making this algorithm agnostic. However, GMM does struggle with handling singularities; When

one has insufficiently many points per mixture, estimating the covariance matrices becomes difficult, and the algorithm is known to diverge and find solutions with infinite likelihood unless one regularizes the covariances artificially. Also, this algorithm will always use all the components it has access to, needing held-out data or information theoretical criteria to decide how many components to use in the absence of external cues.

3.5 Self-organizing Map advantages and disadvantages

SOM is a great tool for visualizing high dimensional data as it preserves neighbors from initial mapping, i.e. preserves the topography of the initial dataset in the higher dimension as it projects into the lower dimension. In SOM, initial neuron weights need to be defined before the algorithm can run. These neuron weights need to be precise to cluster inputs. SOM can start with randomized neuron weights, however, this can lead to converges to different final mapping depending on what the initial conditions were. Over multiple runs, an optimal final mapping can be found, but this prolongs runtime. Determining what the initial weights should be beforehand is a challenge. One method for determining the initial neuron weights is by using K Means. K Means can identify initial weights that can then be used in the SOM computation. For extremely large data sets, it may be necessary to employ a dimensionality reduction technique to optimize SOM training. Other disadvantages to SOM is that it does not handle outliers very well. SOM works ideal with large, similar data sets. If there are too many anomalies or not enough data, SOM will generate less than optimal groupings.

3.6 Relative performance of algorithms

A summary of clustering algorithm performance by relative performance is listed in the table below. Overall, there was no one clustering algorithm that worked well for both RGB and hyperspectral images. GMM performed the best for the test RGB images as evaluated by the OCE values. For hyperspectral images, it was found that SOM performed the best from the OCE values of the Pavia image and subjective analysis of the Santa Barbara image. FCM and K means produced mediocre clustering results for all types of images. The spectral clustering algorithm found itself on both sides of the spectrum working relatively well for hyperspectral images but performing poorly on RGB images.

Ranks	RGB Rankings	Hyperspectral Rankings
1 (Best)	GMM	SOM
2	SOM	Spectral Clustering
3	KMeans	FCM
4	FCM	GMM
5	Spectral Clustering	K Means