Image Compression by K-Means Clustering and Dimensionality Reduction by Principal Component Analysis

Swapna Pillai
Assistant Professor
Department of Electronics & Communication Engineering
Acropolis Institute of Technology and Research, India

Abstract— Image compression is a technique of data compression applied on digital images. Here we will implement image compression using k-means clustering algorithm and dimensionality reduction by Principal Component Analysis.

Keywords: Image compression, K-Means Clustering

I. Introduction

Image compression is a technique of data compression applied on digital images. It is mainly employed to reduce the cost of data storage or transmission. There are several literatures available for image compression using various techniques. [1],[2],[3],[5],[7],[9] Most of the image compression algorithms takes the advantage of visual perception or the statistical properties of the digital data. Here we will implement image compression using k-means clustering algorithm and dimensionality reduction by Principal Component Analysis.

An image is made up of different intensity values of pixels. Basically in a colored image each pixel is of 3 bytes containing RGB (Red-Green-Blue).

A. K means clustering:

It is an unsupervised machine learning algorithm. Typically, unsupervised algorithms make inferences from datasets using only input vectors without referring to known, or labelled, outcomes[1],[2],[3]. This method group similar data points together and discover underlying patterns. To achieve this objective, K-means looks for a fixed number (k) of clusters in a dataset [1]. A cluster refers to a collection of data points aggregated together because of certain similarities [1]. You'll define a target number k, which refers to the number of centroids you need in the dataset. A centroid is the imaginary or real location representing the center of the cluster. Every data point is allocated to each of the clusters through reducing the in-cluster sum of squares. In other words, the K-means algorithm identifies k number of centroids, and then allocates every data point to the nearest cluster, while keeping the centroids as small as possible. The 'means' in the K-means refers to averaging of the data; that is, finding the centroid.

How the K-means algorithm works:

To process the learning data, the K-means algorithm in data mining starts with a first group of randomly selected centroids, which are used as the beginning points for every cluster, and then performs iterative (repetitive) calculations to optimize the positions of the centroids. It halts creating and optimizing clusters when either:

– The centroids have stabilized — there is no change in their values because the clustering has been successful.
– The defined number of iterations has been achieved.

Concretely, you are given a training set {x(1),...,x(m)} (where x(i) ∈Rn), and want to group the data into a few cohesive "clusters".

The K-means algorithm is as follows:

The K-means algorithm is as follows:

```
% Initialize centroids
centroids = k MeansInitCentroids (X, K);
for iter = 1: iterations
% Cluster assignment step: Assign each data point
% closest centroid. idx(i) corresponds to c(i), the index
% of the centroid assigned to example i idx = find ClosestCentroids(X, centroids);
% Move centroid step: Compute means based on centroid
% assignments
centroids = computeMeans(X, idx, K);
end
```

The inner-loop of the algorithm repeatedly carries out two steps: (i) Assigning each training example x(i) to its closest centroid, and (ii) Recomputing the mean of each centroid using the points assigned to it. The K-means algorithm will always converge to some final set of means for the centroids. Note that the converged solution may not always be ideal and depends on the initial setting of the centroids. Therefore, in practice the K-means algorithm usually run a few times with diff erent random initializations. One way to choose between these diff erent solutions from diff erent random initializations is to choose the one with the lowest cost function value (distortion).

We will implement this in two parts

1) Finding Closest Centroids

In the "cluster assignment" phase of the K-means algorithm, the algorithm assigns every training example x(i) to its closest centroid, given the current positions of centroids. Specifically, for every example i we set c(i) := j that minimizes ||x(i) −μj||2, where c(i) is the index of the centroid that is closest to x(i), and μj is the position (value) of the j'th centroid

2) Computing Centroid Means

The second phase of the algorithm recomputes, for each centroid, the mean of the points that were assigned to it.

Results:

K-means on example dataset

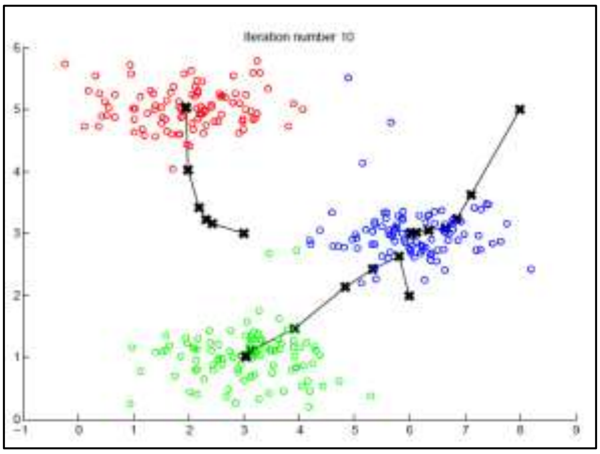

Fig. 1: K-means clustering

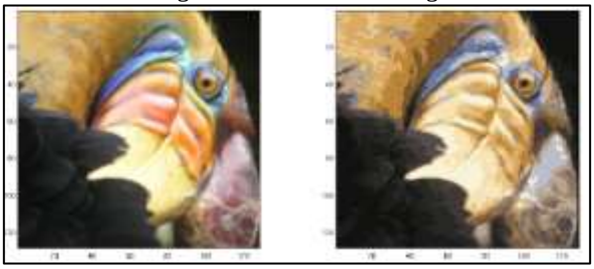
Fig. 2: Original and Reconstructed images

Finally, we can view the eff ects of the compression by reconstructing the image based only on the centroid assignments. Specifically, we can replace each pixel location with the mean of the centroid assigned to it. Figure [2] shows the reconstruction we obtained. Even though the resulting image retains most of the characteristics of the original, we also see some compression artifacts.

II. Principal Component Analysis

Principal Component Analysis (PCA) is a basis transformation to diagonalize an estimate of the covariance matrix of the data.[5],[6],[7]. The new coordinates in tile Eigenvector basis, i.e. the orthogonal projections onto the Eigenvectors, are called principal components.

A. Dimensionality Reduction with PCA:

Dimensionality reduction is the transformation of high-dimensional data into a meaningful representation of lower dimensionality. It is an important field in machine learning, since it facilitates classification, compression, and visualization of high-dimensional data, by mitigating various undesired properties of highdimensional spaces. Traditionally, dimensionality reduction is performed by means of linear techniques such as PCA and LDA. Recent advances in dimensionality reduction are based on the intuition that highdimensional data lies on or near a low-dimensional manifold that is embedded in the high-dimensional space [5]. This intuition has led to the proposal of a large number of techniques that are based on the evaluation of local properties of the data, under the assumption that the data manifold is locally linear. These techniques can be subdivided into three different types: (1) techniques that preserve global properties of the datasuchasIsomapandFastMVU,(2)techniquesthatpreservelocalpropertiesofthedatasuchasLLEand Laplacian Eigenmaps, and (3) techniques that perform global alignment of a mixture of local linear models such as manifold charting and LLC.

B. Face Image Dataset:

In this part,we will run PCA on face images to see how it can be used in practice for dimension reduction. We will apply this on the face image data set. Now that you have computed the principal components for the face dataset, you can use it to reduce the dimension of the face dataset.

III. Results


Fig. 3: Principal Components


Fig. 4: Original Image


Fig. 5: Reconstructed Image

From the reconstruction, we can observe that the general structure and appearance of the face are kept while the fine details are lost.

REFERENCES

[1] P.K. Agarwal, and C.M. Procopiuc, "Exact and Approximation Algorithms for Clustering," Proc. Ninth Ann. ACM-SIAM Symp. Discrete Algorithms, pp. 658-667, Jan. 1998.
[2] S. Arya, D.M. Mount, N.S. Netanyahu, R. Silverman, and A.Y. Wu, "An Optimal Algorithm for Approximate Nearest Neighbor Searching," J. ACM, vol. 45, pp. 891-923, 1998.
[3] P.S. Bradley, and U. Fayyad, "Refining Initial Points for K-means Clustering," Proc. 15th Int'l Conf. Machine Learning, pp. 91-99, 1998.
[4] M. A. Aizerman, E. M. Braverman, & L. I. Rozonoér. Theoretical foundations of the potential function method in pattern recognition learning. Automation and Remote Control, 25:821–837, 1964.
[5] B. E. Boser, I. M. Guyon, & V Vapnik. A training algorithm for optimal margin classifiers. In Fifth Annual Workshop on COLT, Pittsburgh, 1992. ACM.
[6] M. Kirby & L. Sirovich. Application of the Karhunen-Loève procedure for the characterization of human faces. IEEE Transactions, PAMI-12(1):103–108, 1990.
[7] P. Simard, Y. LeCun. & J. Denker. Efficient pattern recognition using a new transformation distance. In S. J. Hanson, J. D. Cowan, & C. L. Giles, editors, Advances in NIPS 5, San Mateo, CA, 1993. Morgan Kaufmann.
[8] J.B. Tenenbaum, V. de Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. Science, 290(5500):2319–2323, 2000.
[9] L.J.P. van der Maaten. An introduction to dimensionality reduction using Matlab. Technical Report 07-06, MICC-IKAT, Maastricht University, Maastricht, The Netherlands, 2007.
[10] L.J.P. van der Maaten, E.O. Postma, and H.J. van den Herik. Dimensionality reduction: A comparative review. Preprint. 2007.

