Visualizing and Understanding the Relationship between PCA, Auto encoder and K-Means Clustering

Nikhil Jangamreddy
IIT Ropar
Punjab, India
2018csm1011@iitrpr.ac.in

ABSTRACT

Principal Component Analysis (PCA) is a widely used technique in the area of Unsupervised Dimensionality Reduction. In Unsupervised data Clustering, one of the popular technique is K-means clustering. C Ding et al. proved that K-means Clustering can be approximated as a super-sparse PCA. Authors also proved that that the relaxed solution of K-means Clustering, specified by the Cluster Indicators, is given by Principal Component Analysis (PCA). Although PCA is not a Clustering method, it is generally used to reveal Clusters. In General, both methods, PCA and K-means Clustering are used together. This is because in case of higher Dimension data, PCA helps in reducing the Dimension of data on which we can apply K-means Clustering to reduce Computation cost. In a nutshell, we aim to establish the Relationship between PCA and K-means Clustering along with needed proofs. Later we Visualise graphs, for this established Relationship on IMDB Movie dataset. Later we extend to Understand relationship between PCA and auto-encoder i.e.., under what constraints PCA is equivalent to Auto encoder using IRIS dataset.

CCS CONCEPTS

• **Machine Learning → Unsupervised learning**; • **Dimensionality Reduction → Principal Component Analysis**.

KEYWORDS

Unsupervised learning, Neural networks, Dimensionality reduction, Clustering

ACM Reference Format:
Nikhil Jangamreddy. 2019. Visualizing and Understanding the Relationship between PCA, Auto encoder and K-Means Clustering. In *Proceedings of Project report*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION AND MOTIVATION

This paper[3] new insights to the observed effectiveness of PCA-based data reductions, beyond the conventional noise-reduction explanation. The need for Data reductions are characterized by Continuously evolving data in high dimension space. In Unsupervised learning, most popular technique is K means Method[5] , which use Centroids to characterise clusters and optimise the Total cluster Distance (within Cluster distance and Between cluster distance).

In Unsupervised dimensonality reduction, PCA is one among most often used technique. In this technique, High dimension input is reduced to low Dimension input data. In [9], It is shown that applying PCA first when applied on data and we perform K-means is applied on it to Cluster the data.

In Principal Component Analysis(PCA), we pick the Attributes which give maximum variance. It is proven to be Equivalent to the best low rank approximation to Singular value decomposition[4]. Although the reasoning why PCA is applied first before applying K-means can be explained with noise reduction, however effectiveness of PCA can be explained through formal proof as discussed[3].

Later, we express the relation between PCA and linear Auto encoder. It is proven that under the constraints by using Linear activation function for encoder, decoder along with squared loss function in auto encoder is equivalent to PCA.

2 RELATED WORK

Applying Dimensionality reduction technique before applying clustering is common for noise reduction. As mentioned [8], principal component analysis is applied before K-means clustering for gene expression data. Later Ding proposed formal proof[3] to prove the idea of applying PCA first in K-means clustering is approximately equivalent to K-means clustering. Essentially taking maximum variance is proven to be equivalent to best low rank approximation to singular value decomposition[4].

Dimensionality reduction method PCA is proven to be equivalent to linear auto encoder under particular constraints[2]. It is also shown that sub space spanned by principal component analysis is same sub space spanned by auto encoder weights. Later [7], proved that first m singular vectors in auto encoder weights are the principal components in PCA. Instead of linear encoder and decoder, It is also shown that using non linear encoder,decoder can perform very well compared to PCA, linear auto encoder[6].

3 METHODOLOGY

As specified in Section 1, initially we prove how K-means clustering can be performed via principal component analysis and show how it is equivalent to K-means clustering. In subsection 3.1, we prove principal components are the continuous solutions to the discrete cluster membership indicators for K-means clustering. Later in subsection 3.2, we prove that how PCA is equivalent to Linear auto encoder.

Project report, CS503, 2019

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

3.1 K-means Clustering via Principal Component Analysis

Below description formally proves that principal components computed in the PCA technique are equivalent to Cluster membership indicators in K-Means clustering algorithm[3]. In other words, PCA is performing Clustering according to Objective function defined in K-means clustering. This provides formal argument to performing K-means Clustering via PCA.

K means clustering is characterised by Centroids for individual Clusters. The objective function defined in K-means clustering is given by :

$$J_{\mathrm{K}} = \sum_{k=1}^{K} \sum_{i \in C_k} (\mathbf{x}_i - \mathbf{m}_k)^2$$

Here (x1,x2,...,xn) denote Data matrix,

m_k denote centroid of Cluster k where

$$m_k = \sum_{i \in C_k} \mathbf{x}_i / n_k$$

n_k denote number of points in a cluster k.

Formulation for PCA :

Let the data matrix be X = (x1,x2,...,xn)

Y = (y1,y2....yn) and

$$\mathbf{y}_i = \mathbf{x}_i - \overline{\mathbf{x}}$$

and

$$\overline{\mathbf{x}} = \sum_i \mathbf{x}_i / n$$

Covariance matrix is designed as

$$\sum_i (\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^T / n = YY^T$$

principal directions be A_k, Eigen values as B_k

Condition for PCA can be specified as :

$$YY^T \mathbf{A}_k = \lambda_k \mathbf{A}_k, Y^T Y \mathbf{B}_k = \lambda_k \mathbf{B}_k, \mathbf{B}_k = Y^T \mathbf{A}_k / \lambda_k^{1/2}$$

3.1.1 *Principal Components in PCA are equivalent to cluster membership indicators in K-means clustering.*

2 way Clustering :

Let us assume K = 2, that implies

$$d(\mathrm{C}_k, \mathrm{c}_\ell) \equiv \sum_{i \in C_k} \sum_{j \in C_\ell} (\mathrm{x}_i - \mathrm{x}_j)^2$$

Then after further Simplification we get i.e..,

$$J_{\mathrm{K}} = \sum_{k=1}^{K} \sum_{i \in C_k} \frac{(\mathbf{x}_i - \mathbf{x}_j)^2}{2n_k} = n\overline{\mathbf{y}^2} - \frac{1}{2} J_{\mathrm{D}}$$

and also,

$$J_{\mathrm{D}} = \frac{n_1 n_2}{n} \left[2 \frac{d(\mathrm{c}_1, \mathrm{C}_2)}{n_1 n_2} - \frac{d(\mathrm{c}_1, \mathrm{C}_1)}{n_1^2} - \frac{d(\mathrm{c}_2, \mathrm{C}_2)}{n_2^2} \right]$$

and

$$\frac{d(\mathrm{C}_1, \mathrm{C}_2)}{n_1 n_2} = \frac{d(\mathrm{C}_1, \mathrm{C}_1)}{n_1^2} + \frac{d(\mathrm{C}_2, \mathrm{C}_2)}{n_2^2} + (\mathrm{m}_1 - \mathrm{m}_2)^2$$

where $\overline{\mathbf{y}^2} = \sum_i \mathbf{y}_i^T \mathbf{y}_i / n$

Since J_D is always positive, so minimising J_{K} is equivalent to maximising J_{D}

Theorem 1 :

In [3] it is proved for K=2, minimising the within cluster distance J_{K} is equivalent to maximising the the distance objective function J_{D} which is always positive.

Note :

1) Later in [3], they prove that maximising J_{D} is equivalent to applying Principal component analysis.

Theorem 2 :

For K = 2, Clustering membership solution is equivalent to principal component v1.

Similarly we can extend this idea to any arbitrary K using regularised relaxation method[3].

3.2 PCA is equivalent to Linear auto encoder

Auto Encoder is a popular method in Deep learning. As discussed earlier, PCA is a dimensionality reduction technique used for noise reduction[1]. It is proven that PCA is equivalent to Linear auto encoder. In PCA the subspace spanned by Principal components is the same subspace spanned by Auto encoder which use linear activation function, and Squared loss function.

3.2.1 *Proof : How PCA equivalent to Linear Auto encoder with Squared loss Function ?*

In case of PCA, minimum reconstruction error is defined as :

$$\mathcal{L}(\mathbf{x}, \tilde{\mathbf{x}}) = \|\mathbf{x} - \tilde{\mathbf{x}}\|^2$$

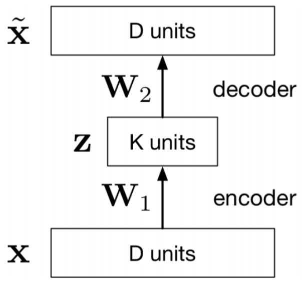

Figure 1: Auto encoder which takes d dimension input with one hidden layer

$$\mathbf{z} = f(W_1 \mathbf{x}); \quad \hat{\mathbf{x}} = g(W_2 \mathbf{z})$$

Squared loss error is given by :

$$\min_{\mathbf{W}_1, \mathbf{W}_2} \frac{1}{2N} \sum_{n=1}^{N} \left\| \mathbf{x}^{(n)} - \hat{\mathbf{x}}^{(n)} \right\|^2$$

If we Assume functions f and g linear Squared loss can be rewritten as,

$$\min_{\mathbf{W}_1, \mathbf{W}_2} \frac{1}{2N} \sum_{n=1}^{N} \left\| \mathbf{x}^{(n)} - W_2 W_1 \mathbf{x}^{(n)} \right\|^2$$

If we Consider in Auto encoder we have,

$$\tilde{\mathbf{x}} = \mathbf{W}_2\mathbf{W}_1\mathbf{x}$$

Under the Constraint,

$$\mathbf{W}_2\mathbf{W}_1 = \mathbf{I}$$

The above Optimisation problem of Auto encoder is equivalent to PCA as specified below.

$$\mathcal{L}(\mathbf{x}, \tilde{\mathbf{x}}) = \|\mathbf{x} - \tilde{\mathbf{x}}\|^2$$

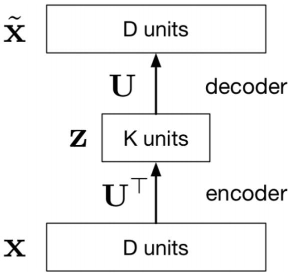

Figure 2: Linear Auto encoder which is equivalent to PCA

Note : Principal components weights not equal to Linear Auto encoder weights.

The method on how to retrieve Principal components from Linear auto Encoder weights is shown to be first m singular vectors of W_2 are the first m principal components of X[7].

4 EXPERIMENTAL SETTINGS

In order to evaluate the performance of two methods described in Section 3, For K means Clustering via principal component analysis specified in section 3.1 we choose IMDB movie data. For PCA is equivalent to Linear auto encoder specified in section 3.2, we choose IRIS data. Further details are specified below.

4.1 K-means Clustering via Principal Component Analysis

In this experiment we have considered IMDB movie dataset. Dataset includes whether a movie belongs to different genres. It contains 14332 × 44 data. After removing the columns which contain strings like URL we get 14332 × 38 data. Later we center the data around origin.

To apply Principal component analysis, we need to know number of Principal components to choose. In order to know the number of principal components we use the technique called explained variance. Later we apply Sum of squares distance as metric to measure the performance. To find the best K i.e.., number of Clusters, we use elbow method.

4.2 PCA is equivalent to Linear auto encoder

In this experiment we have considered IRIS data. IRIS data includes 150 × 5 data. It contains sepal width, sepal length, petal width, petal length as columns. The data includes 50 rows corresponding to each flower type. Flower types include Iris Setosa, Iris Versicolour and Iris Virginica.

5 RESULTS AND DISCUSSION

For the Experiment settings discussed in section 4.1,4.2 below section discusses results regarding the same. In experiments mentioned in section 5.1, we compare the cluster shapes as well as Squared error obtained in comparing PCA guided K-means clustering with K-means clustering. In experiments mentioned in section 5.2, we perform PCA and linear auto encoder on IRIS data to visualise the clusters.

5.1 K-means Clustering via Principal Component Analysis

To choose the appropriate number of Principal components we choose Explained variance metric and cumulative variance metric. Below is the plot corresponding to number of principal components vs explained variance, cumulative variance. As per the plot we have choosen number of components = 27, which explains 90% cumulative variance. Later we apply PCA on the data which will reduce data to 14332 × 27. We apply K-means on this data which we refer as PCA guided K-means clustering. Below is plot for PCA guided K-means clustering Vs K-means clustering with number of clusters = 3. Below is plot for PCA guided K-means clustering Vs K-means clustering with number of clusters = 4. In order to find

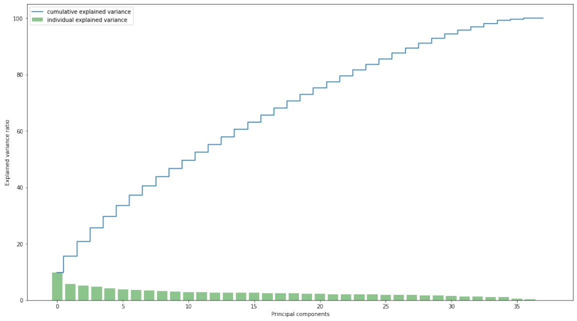

Figure 3: Principal components vs Explained variance

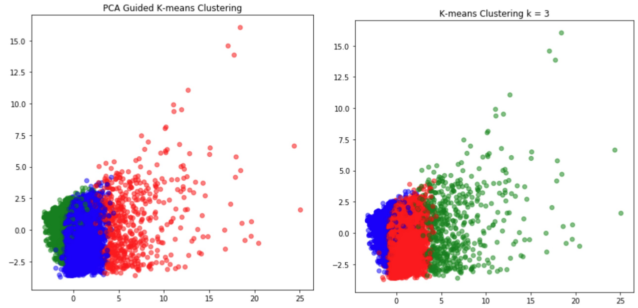

Figure 4: PCA guided K-means clustering vs K-means clustering for number of clusters = 3

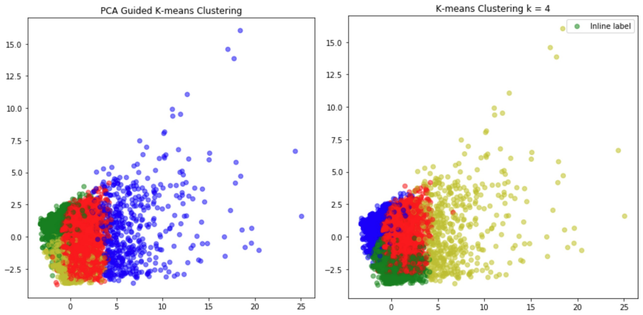

Figure 5: PCA guided K-means clustering vs K-means clustering for number of clusters = 4

optimal number of clusters k for PCA guided K-means clustering vs K-means clustering for different number of principal components. Left plot is taken for number of principal components = 27, right plot is taken for number of principal components = 32.

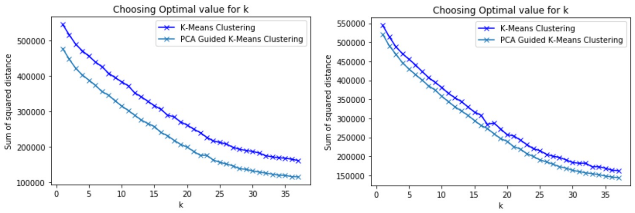

Figure 6: choosing optimal k PCA guided K-means clustering vs K-means clustering for different number of principal components

5.2 PCA is equivalent to Linear auto encoder

As specified in specified in section 4.2, we use IRIS data to demonstrate relation between PCA and linear auto encoder. We standardise the IRIS data and then run PCA on it, later we apply auto encoder with one hidden layer with linear encoder, decoder with mean squared error. Below is plot for PCA vs Linear auto encoder on IRIS dataset.

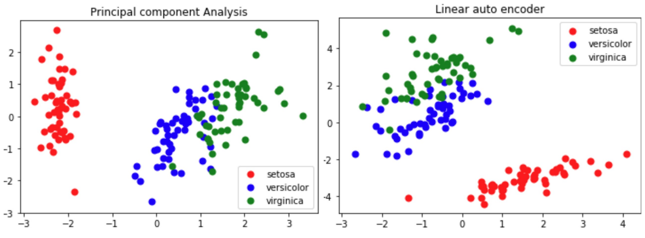

Figure 7: PCA vs Linear auto encoder on IRIS dataset

6 SUMMARY AND FUTURE WORK

In section 3, we have proved that PCA guided K-means clustering is equivalent to K-means clustering assuming sufficient number of principal components are choosen. In order to simulate this we have taken IMDB movie dataset to simulate the results described in [3].Later we have proved that PCA is equivalent to auto encoder assuming one hidden layer, linear encoder, linear decoder and mean squared error. Later we simulated PCA linear encoder equivalence using IRIS dataset. It is also shown that linear encoder weights are not equal to principal component weights. For future work, we can study how to retrieve principal components from auto encoder weights. Also we can work on non linear encoding,decoding function in auto encoder and evaluate its performance with respect to PCA. It is proved in [6] that non linear auto encoder can perform very well compared to PCA for complex data.

REFERENCES

[1] [n. d.]. PCA Auto encoder idea Kernel Description. https://www.cs.toronto.edu/~urtasun/courses/CSC411/14_pca.pdf.

[2] Pierre Baldi and Kurt Hornik. 1989. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks* 2, 1 (1989), 53–58.

[3] Chris Ding and Xiaofeng He. 2004. K-means clustering via principal component analysis. In *Proceedings of the twenty-first international conference on Machine learning*. ACM, 29.

[4] Carl Eckart and Gale Young. 1936. The approximation of one matrix by another of lower rank. *Psychometrika* 1, 3 (sep 1936), 211–218. https://doi.org/10.1007/BF02288367

[5] J. A. Hartigan and M. A. Wong. 1979. Algorithm AS 136: A K-Means Clustering Algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 28, 1 (1979), 100–108. http://www.jstor.org/stable/2346830

[6] Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *science* 313, 5786 (2006), 504–507.

[7] Elad Plaut. 2018. From principal subspaces to principal components with linear autoencoders. *arXiv preprint arXiv:1804.10253* (2018).

[8] Ka Yee Yeung and Walter L. Ruzzo. 2001. Principal component analysis for clustering gene expression data. *Bioinformatics* 17, 9 (2001), 763–774.

[9] Hongyuan Zha, Xiaofeng He, Chris Ding, Horst Simon, and Ming Gu. 2002. Spectral Relaxation for K-means Clustering. *Adv. Neural Inf. Process. Syst.* 14 (2002).