CLUSTERING CST TELEPHONE BILL USING HYBRID PCAK ALGORITHM

DechenPelki[1], Dechen Wangmo[2], Tshering[3]

Department of Information Technology,College of Science and Technology
Royal University of Bhutan, Bhutan
Email: [1]2010015@cst.edu.bt, [2]2010015@cst.edu.bt, [3]tshering.cst@rub.edu.bt

ABSTRACT

In this paper, we chose to study Principal Component Analysis(PCA) and K-mean clustering algorithm(K) to investigate a set of real world telephone data. The raw data we received showed high variation between maximum and minimum data. Hybrid PCAK was thus proposed. The PCA normalize the data range and dimension. While K-mean clusters the normalized and dimension reduced data into k cluster. The clustered output from PCAK showed telephone use pattern of CST staff.

Keywords:*PCA, Clustering, K-mean, PCAK*

1. INTRODUCTION

Data analysis is becoming increasingly difficult in data-rich fields due to large data dimensions. For such fields, hybrid implementation proved to classify and organize data into easy to visualize output.

Principal Components Analysis is a standard tool in data analysis that constructs a representation of the data with a set of orthogonal basis vectors that are the eigenvectors of the covariance matrix generated from the data. By projecting the data onto the dominant eigenvectors, the dimension of the original dataset can be reduced with little loss of information.

K-means clustering is a type of data mining algorithm involving the clustering of various observations into different groups. It is very simple and the group clustering can also be done without any knowledge of variable relationships. It results in a more efficient and faster way of determining patterns especially when the data or information involved is large. With hybrid clustering technique, extraction of data and analysis become much easier, efficient, and faster. This proposed method is numerical, unsupervised, non-deterministic and iterative.

In this study, the MATlab software is chosen and used to construct PCAK hybrid algorithm and analyze real world telephone bill of staff's office telephone bills.

2. THEORETICAL BACKGROUND

2.1 PCA

Principal component analysis is a variable reduction procedure. When obtaining data with a large number of variables, there must be *redundancy* in those variables.

Redundancy means the variables are correlated with each other as they are measuring the same construct. Therefore, to reduce the observed variables into a smaller number of principal components or artificial variables that are responsible for most of the variance in the observed variables, PCA is performed.

The PCA:

a. Simplifies data by reducing dimensions of data space

b. Finds the most informative viewpoint from which to visualize the data from a scatter plot

c. Produces low-dimensional images of high dimensional shapes

d. Shows amount of variance between axes

2.2 PCA ALGORITHM

According to (Smith, 2002), PCA algorithm consists of six steps:

a. Generate a set of data

b. Subtract the mean: subtract the mean fromeach of the data dimensions.

c. Calculate the covariance matrix: In the statistical analysis, covariance is computed to find out how much the dimensions vary from the mean with respect to each other. It is always measured between two or more dimensional data sets. The size of the covariance matrix depends on dimension of data. For example for 2 dimensional data, covariance matrix will be 2x2 matrix, similarly for 3-dimensional data it will be 3x3 matrixes.

Let the subtracted mean be represented in matrix B, and covariance matrix C can be computed as:

$$C = \frac{1}{n-1} B^{'} B \quad \text{.....(}$$

a) where, $B^{'}$ = transpose matrix of B
n= no. of data samples

$$C = \begin{pmatrix} \text{cov}(x,x) & \text{cov}(x,y) \\ \text{cov}(y,x) & \text{cov}(y,y) \end{pmatrix}$$

.....(b)

d. Calculate the eigenvectors and eigenvalues of the covariance matrix:
The eigenvector with the highest eigenvalue is the principle component of the data set. It is the most significant relationship between the data dimensions. In general, once eigenvectors are found from the covariance matrix, the next step is to order them by eigenvalue, highest to lowest. This gives us the components in order of significance. Now, if we like, we can decide to ignore the components of lesser significance. We do lose some information, but if the eigenvalues are small, we don't lose much. If we leave out some components, the final data set will have fewer dimensions than the original.

e. Choose components and form a feature vector: once eigenvectors are found from the covariance matrix, the next step is to order them by eigenvalue, highest to lowest. Thus the components are sorted in order of significance. The number of eigenvectors chosendecides the number of dimensions of the new data set, thereby constructs a feature vector(matrix of vectors). From the list of eigenvectors take the eigenvectors and form a matrix in the columns:

FeatureVector =
(eig_1,eig_2,.eig_n).....(c)

f. Derive the new data set: Take the transpose of the FeatureVector and multiply with the original data set, transposed:

*FinalData = RowFeatureVector*x

RowDataAdjusted.....(d)

whereRowFeatureVector is the matrix with the eigenvectors in the columns transposed (the eigenvectors are now in the rows and the most significant are in the top) and

RowDataAdjusted is the mean-adjusted data transposed (the data items are in each column, with each row holding a separate dimension).

2.3 COMPUTING THE PRINCIPAL COMPONENTS

In computational terms the principal components are found by calculating the eigenvectors and eigenvalues of the data covariance matrix. This process is equivalent to finding the axis system in which the co-variance matrix is diagonal. The eigenvector with the largest eigenvalue is the direction of greatest variation, the one with the second largest eigenvalue is the (orthogonal) direction with the next highest variation and so on.

2.4 K-MEAN CLUSTERING

K-Means clustering generates a specific number of clusters. From given unsupervised data,it randomlychooses k clusters if user donot specify number of cluster. At the end of each iteration, data which are close to each other form groups in to cluster. K-means is a partitioned clustering algorithm.

Let the set of data points (or instances) D be $\{\mathbf{x}1, \mathbf{x}2, \ldots, \mathbf{x}n\}$,

where$\mathbf{x}i = (xi1, xi2, \ldots, xir)$ is a vector in a real-valued space $X \subseteq Rr$, and r is the number of attributes (dimensions) in the data.

The *k*-means algorithm partitions the given data into *k* clusters.

a. Each cluster has a cluster **center** called **centroid**.
b. K is either specified by the user or randomly pickedfrom data.

2.5 K-MEAN CLUSTERING PROPERTIES

a. There are always K clusters.
b. There is always at least one item in each cluster.
c. The clusters are non-hierarchical and they do not overlap.
d. It uses Euclidean distance to calculate distance between Centre and data point. Euclidean Distance is the most common use of distance. It examines the root of square differences between coordinates of a pair of objects. The Euclidean Distance between point's p and q is the length of the segment connecting them:

2.6 K-MEAN CLUSTERING ALGORITHM

a. Cluster number (K) and k-cluster centroids are initialized using one of the methods listed below:

i. Random initialization
ii. Sampling initialization

b. The distances between cluster centroids and each objects are calculated, group is assigned, comparing distances of each object with the closest cluster center.

c. Determine the new center coordinate from the new group.

d. Compute new distance with the new centroid and objects, assign group to the objects closest to cluster center.

e. Check for convergence, if not converged, repeat procedure from step.c.

If the number of data is less than the number of cluster then each data is assumed as the centroid of the cluster. Each centroid will have a cluster number. If the number of data is bigger than the number of cluster, for each data, distance to all centroid is calculated and get the minimum distance. This data is said belong to the cluster that has minimum distance from this data.

Note: The convergence will always occur if the following condition is satisfied:

i. Each switch in step 2 the sum of distance from each training sample to that training sample's group centroid is decreased.

ii. There are only finitely many partitions of the training examples into k clusters.

f. Stop

2.7 ARCHITECTURE OF K-MEAN ALGORITHM

The fig.1 represents the architecture of k-means. In its operation, number of cluster is either set or generated from the data. Centroid is initialized, distance between data and centroid is computed. Cluster is formed based on minimum distance. New centroid is selected and keeps on shifting data till it stops.

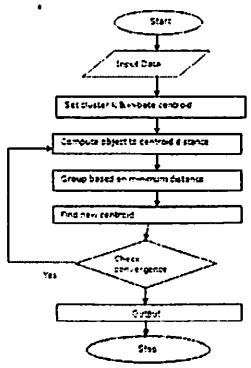


Fig. 1: Architecture of k-mean algorithm.

1.8 K-MEAN TIME COMPLEXITY

a. Computing distance between the vectors is O(M), where M is the dimensionality of vectors.

b. Reassigning clusters: O(KN) distance computations, O(KNM).

c. Computing centroids: each vectors get added to some centroid. O(NM)

d. Assume two steps are done once for I iterations: O(IKNM)

3. METHODOLOGY

3.1 SYSTEM DESCRIPTION

The design of system is shown in Fig.2. The dimension reduction and normalization of telephone data is done using PCA. The preprocessed data is passed through k-mean algorithm and cluster is generated as output.

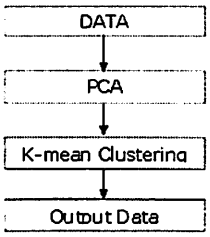


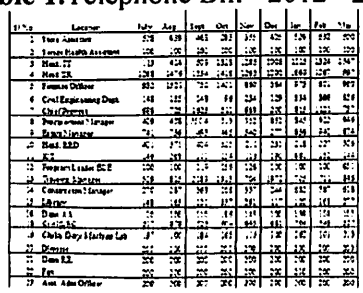
Fig. 2: System diagram.

4. EXPERIMENT

4.1DATA

The telephone bill of nine month from July 2012 to March 2013 is collected and tabulated as shown in table 1.

Table 1. Telephone Bill - 2012 - 2013



4.2 APPLYING PCA TO THE DATA SET

a. Subtract the mean

PCA subtracts the mean from each of the data dimensions. Here, we basically transformadata set producing new data set(Fig 3) whose mean is zero.

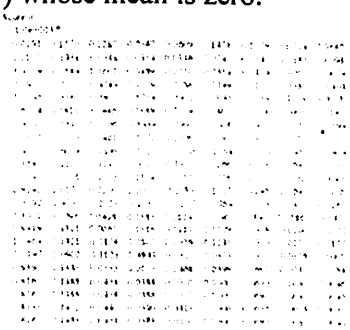


Fig. 3: Transformed data set

b. Calculate the covariance matrix

The covariance matrix is computed usingMathlab function, covMatrix=cov(A). covariance matrix from data set is shown in fig. 4.

CovMatrix =

1.0e-005 *

0.9611	1.2247	1.1856	1.0065	0.7517	0.6834	0.9856	0.5609	0.5683
1.2247	1.7597	1.6192	1.7221	1.1675	1.1629	1.3992	1.1806	0.9035
1.1856	1.6190	3.1565	2.3280	1.3218	2.0690	1.4624	1.5331	1.1075
1.0065	1.7221	2.3280	2.8031	1.5043	2.2377	1.5505	1.4349	1.3720
0.7517	1.1675	1.3218	1.5043	1.2721	1.2033	1.3163	1.3691	1.1471
0.6834	1.1629	2.0692	2.2377	1.2033	2.1936	1.2562	1.1529	1.0567
0.9856	1.3992	1.4624	1.5506	1.3163	1.2562	1.5877	1.4992	1.1879
0.5609	1.1806	1.5331	1.4349	1.3691	1.1529	1.4992	1.6781	1.3691
0.5683	0.9036	1.1075	1.3720	1.1471	1.0567	1.1879	1.3691	1.4333

Fig. 3:covariance matrix

c. Calculate the eigenvectors and eigenvalues of the covariance matrix

The eigenvectors and eigenvalues of the covariance matrix of data set is computed using Mathlab function tools eighnvectors and eigenvalues. eigenvectors and eigenvalues generated is shown in fig. 4.

eigenvectors =

-0.2139	0.1499	-0.5051	0.0763	-0.0343	0.3142	-0.2770	-0.4163	-0.5663
-0.3150	0.1472	-0.4927	0.3995	-0.1819	0.1873	-0.0241	0.2819	0.5716
-0.4290	-0.4764	-0.3039	-0.6309	-0.1363	-0.1227	0.2235	-0.0675	0.1024
-0.4332	-0.2820	0.2027	0.4735	-0.2949	-0.3296	0.0996	0.2662	-0.4373
-0.2911	0.2242	0.1520	0.1123	-0.0060	-0.5371	-0.1917	-0.6372	0.3132
-0.3551	-0.4286	0.3608	0.1356	0.4568	0.4572	-0.2954	-0.0995	0.1614
-0.3203	0.3661	-0.0610	0.0220	0.6320	-0.3592	0.5752	0.0704	-0.1303
-0.3122	0.3946	0.1160	-0.3942	0.0661	-0.1395	-0.5615	0.4797	-0.0935
-0.2656	0.3545	0.4452	-0.1603	-0.4977	0.4712	0.2976	-0.1445	0.0276

eigenvalues =

1.0e-006 *

1.2803
0.1682
0.1173
0.0750
0.0295
0.0156
0.0075
0.0034
0.0018

Fig. 4:eigenvectors and eigenvalues

d. Choosing components and forming a feature vector

the feature vector of data set is computed using matlab tool function feature_vector. The generated feature vector is shown in fig. 5.

feature_vector =

-0.2139	0.1499
-0.3180	0.1472
-0.4290	-0.4764
-0.4332	-0.2820
-0.2911	0.2242
-0.3551	-0.4286
-0.3203	0.3661
-0.3122	0.3946
-0.2656	0.3545

Fig. 4:feature vector

e. Deriving the new data set

Final Data = Row Feature Vector x Row Data Adjust

Row Feature Vector is the matrix with the eigenvectors in the columns *transposed* so that the eigenvectors are now in the rows, with the most significant eigenvector at the top.

Row Data Adjust is the mean-adjusted data *transposed*, ie. the data items are in each column, with each row holding a separate dimension.

Final Data is the final data items in columns, and dimensions along rows. It is the original data ***solely in terms of the vectors.***(Sayad, 2010)

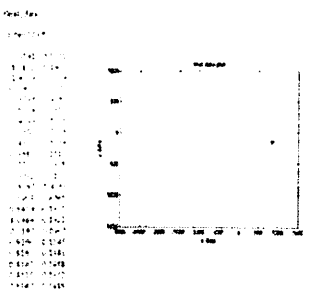
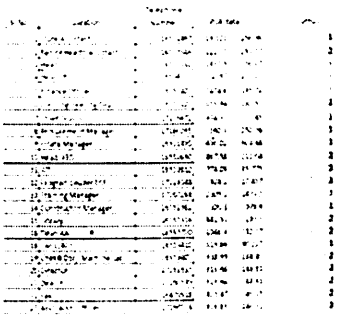


Figure 4. Final data plot

1. Applying K-mean Algorithm to the data set

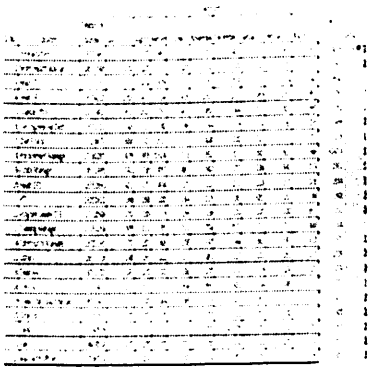
The data is supplied as input to K mean, by choosing three clusters. After iterations the data is grouped into three

clusters, the highest cluster number indicates the person making more calls.



After data is given as input, we group the data into three K i.e. 1, 2, and 3.The above figure shows the data clustered into k=3 ,red has four data indicating 4 people making highest call.

The output is overlapping because distance between the data are more, however, with the help of PCA (Principal reduction analysis) the data set is reduced into two dimensions, hence in the output we can see distinct three clusters with different colors (my case red for highest call made, green for low and blue for medium calls made by staff of CST). Also, with PCA, the group remains unchanged i.e. the person making highest or lowest call also remains the same.



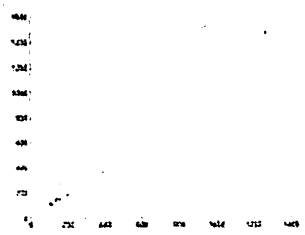


Fig. plot before PCA with the same cluster

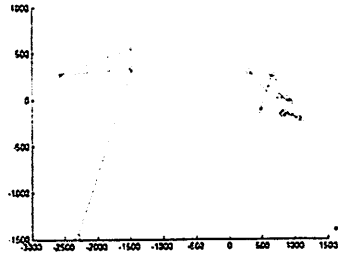


Fig. Plot after PCA with the same cluster

It displays non-overlapping data grouped into three clusters.

1. RESULT AND CONCLUSION

Using PCA, multidimensional data set was normalized from the data ranges of Nu.900 (lowest) to Nu.12,092 (highest) and reduced to 2-dimensional data set. Preprocessed data was clustered using k-mean algorithm in to three clusters. The study showed that it was difficult for single tools to analyze data. Upon proposing the hybrid, it revealed distinct three groups without overlap. It shows that hybrid performs better than the single tools.

2. ACKNOWLEDGEMENT

We would like to extend our gratitude towards Mr. Tshering, Dean RIL, for his continued support and guidance without which this study would have been impossible. We would also like to thank our IT department for allowing us to use the IT facilities. Finally, we thank our

college accounts section for providing us with the telephone bill data.

3. REFERENCE

Abdi, H. & Williams L.J. (2010). *Principal Component Analysis.*

Unknown. Retrieved on April 9th, 2013 from *http://www.vitutor.com/statistics/descriptive/formulas_statistics.html*

Sayad, S.(Dr.).(2010). *PCA Principal Component Analysis.*University of Toronto.

Smith, L.I. (February 26th, 2002). *A Tutorial on Principal Components Analysis.*

Unknown.(2010). *Principal Components Analysis.*36-490.

Halko, N., Martinson,P., Shkolnisky, Y. &Tygert, M. (2010). *An Algorithm for the principal component analysis of large data sets.*

Unknown. Retrieved on March 16th from *www.doc.ic.ac.uk/~dfg/ProbabilisticInference/IDAPILecture15.pdf*

Liu, S. (May 7th, 2011). *The PCA implementation in MATLAB.*

Anh, T. & Magi, S. (June 6th, 2009). *Principal Component Analysis: Final Paper in Financial Pricing.*

Brubaker, S. C. (August, 2009). *Extensions of Principal Component Analysis.*Georgia Institute of Technology.

Tshering. (2007). *Hybrid clustering algorithm: HSOM.* KhonKaen University.

Unknown. Retrieved on March 16th 2013 from *http://www.nlpca.org/pca-principal-component-analysis-matlab.html*

Unknown. Retrieved on March 29th 2013 from *http://www.mathworks.com/help/stats/pca.html#bti6n7k-2*

Dorji, G. Y. (2010). HYBRID KSOM APPROACH TO HIGH IMPEDANCE FAULT DETECTION IN THE DISTRIBUTION FEEDER. College of Science and Technology, Phuntsholing.

Unknown. Retrieved on March 12th 2013 from *http://www.improvedoutcomes.com/docs/WebSiteDocs/Clustering/Performing_K_means_Clustering.htm*

Moore. (2001). cs.cmu.edu/~awm. Retrieved on March 12th 2013 from *https://www.cs.cmu.edu/~awm*

Shmueli, P. a. (n.d.).Data Mining for Business Intelligence.

Tan, S. K. (2005). Retrieved on April 10th 2013 from *http://www-users.cs.umn.edu/~kumar/dmbook/index.php*

Teknomo, K. (2007, July). Retrieved on March 16th 2013 from *http:\\people.revoledu.com\kardi\tutorial\kMean*