Spectral Clustering Using Optimized Gaussian Kernel Function

Article · May 2014

DOI: 10.14257/ijaiasd.2014.2.1.04

3 authors:

Kanaan Bhissy
Islamic University of Gaza
1 PUBLICATION **3** CITATIONS
SEE PROFILE

Fadi El Faleet
ministry of asra
1 PUBLICATION **3** CITATIONS
SEE PROFILE

Wesam Ashour
Islamic University of Gaza
64 PUBLICATIONS **518** CITATIONS
SEE PROFILE

International Journal of Artificial Intelligence and Applications for Smart Devices
Vol.2 , No.1 (2014), pp. 41-56
http://dx.doi.org/10.14257/ijaiasd.2014.2.1.04

Spectral Clustering Using Optimized Gaussian Kernel Function

Kanaan EL. Bhissy, Fadi EL. Faleet and WesamAshour

kanaan.el.bhissy@gmail.com, ffaleet@gmail.com, washour@iugaza.edu.ps

Abstract

Clustering and segmentation algorithms that depend on Gaussian kernel function as a way for constructing affinity matrix, these algorithms like spectral clustering algorithms suffer from the poor estimation of parzen window σ. The final results depend on this parameter and differ on each time we change it.In this paper we present a new algorithm for estimation σ using optimization techniques, we construct a vector $\overrightarrow{\hat{\sigma}_i}$, each $\hat{\sigma}_i$ corresponding to i^{th}row in a dissimilarity matrix which is used to construct an affinity matrix using Gaussian kernel function. Our algorithm shows that choosing $\hat{\sigma}_i$ as the formula $\hat{\sigma}_i^2 = \frac{max\ d(i,j)^2 - min\ d(i,j)^2}{2\ ln\frac{max\ d(i,j)^2}{min\ d(i,j)^2}}$ is the optimum estimation, and we introduce more than one approach to calculate global value for σ from this vector. The affinity matrix which is produced using our algorithm is very informative and contains addition information like the number of clusters.

Keywords:*Data clustering, Adaptive sigma, Mining algorithms, Spectral clustering, kernel trick, Gaussiankernel, Eigen vectors*

1. Introduction

Data clustering techniques are an important aspect used in many fields such as data mining [1], pattern recognition and pattern classification [2], data compression [3], machine learning [4], image analysis [5], and bioinformatics [6]. The purpose of clustering is to group data points into clusters in which the similar data points are grouped in the same cluster while dissimilar data points are in different clusters. The high quality of clustering is to obtain high intra-cluster similarity and low inter-cluster similarity.

The clustering problems can be categorized into two main types: fuzzy clustering and hard clustering. In fuzzy clustering, data points can belong to more than one cluster with probabilities [7] which indicate the strength of relationships between the data points and a particular cluster. One of the most widely used fuzzy clustering algorithms is fuzzy c-mean algorithm [8]. In hard clustering, data points are divided into distinct clusters, where each data point can belong to one and only one cluster. The hard clustering is subdivided into hierarchal and partitional algorithms. Hierarchal algorithms create nested relation-ships of clusters which can be represented as a tree structure called dendrogram [9]. These algorithms can be divided into agglomerative and divisive hierarchal algorithms. The agglomerative

ISSN: 2288-6710 IJAIASD

hierarchal clustering starts with each data point in a single cluster. Then it repeats merging the similar pairs of clusters until all of the data points are in one cluster, such as complete linkage clustering [10] and single linkage clustering [11]. The divisive hierarchal algorithm reverses the operations of agglomerative clustering, it starts with all data points in one cluster and repeats splitting large clusters into smaller ones until each data point belong to a single cluster such as DIANA clustering algorithm [12].Partitional clustering algorithm divides the data set into a set of disjoint clusters such as Kmeans [13], PAM [12] and CLARA [12].

While clustering and segmentation algorithms are unsupervised learning processes, users are usually required to set some parameters for these algorithms. These parameters vary from one algorithm to another, but mostclustering/segmentation algorithms require a parameter that either directly or indirectly specifies the number of clusters/segments [16]. This parameter is typically either k, the number of clusters/segments to return, or some other parameter that indirectly controls the number of clusters to return, such as an error threshold. Setting these parameters requires either detailed pre-existing knowledge of the data, or time-consuming trial and error. if the data set is very large or is multidimensional, human verification could become difficult. To find a reasonable number of clusters, many existing methods must be run repeatedly with different parameters, and are impractical for real-world data sets that are often quite large. Automatically detects the number of clusters in the data set, and selects the most representative dense prototypes to be initial prototypes even if the clusters are in different shapes with different densities [16].

We desire an algorithm that efficiently determine reasonable number of clusters/segments to return from any spectral clustering algorithm. In order to identify the correct number of cluster, we introduce our algorithm for optimal value of sigma.

Clustering Algorithms: Clustering is anunsupervised machine learning process that creates clusters such that data points inside a cluster are close to each other, and also far apart from data points in other clusters. There are four main categories of clustering algorithms: partitioning, density-based, grid-based, and hierarchical. Partitioning algorithms, such as K-means and PAM [14], iteratively refine a set of k clusters and do not scale well for larger data sets. Density-based algorithms, e.g., DBSCAN [3] and DENCLUE [9], are able to efficiently produce clusters of arbitrary shape, and are also able to handle outliers. If the density of a region is above a specified threshold, those points are assigned to a cluster; otherwise they are considered to be noise. Grid-based algorithms, such as WaveCluster [17], reduce the clustering space into a grid of cells, enabling efficient clustering of very large datasets. Hierarchical algorithms can be either agglomerative or divisive. The agglomerative (bottom-up) approach repeatedly merges two clusters, while the divisive (top-down) approach repeatedly splits a cluster into two. CURE [6] and Chameleon [10] are examples of two hierarchical clustering algorithms. Hierarchical algorithms differ chiefly in the criteria that they use to determine similarity between clusters.

Spectral Clustering: Depends on cutting the graph by mapping the data into a high dimensional feature space, where each coordinate corresponds to one feature

of the data items, effectively transforming the data into a set of points in a Euclidean space(Eigen space). In that space, a variety of methods can be used to find relations in the data. Since the mapping can be quite general (not necessarily linear, for example), the relations found in this way are accordingly very general. This mapping approach is called the kernel trick [33].

The spectral clustering depends on GaussianKernel Function to give the edges between nodes(X(i),X(j)) their weight (Wi,j) the equation is:

$$w_{i,j} = e^{\frac{-\|x_{(i)} - x_{(j)}\|_2^2}{\sigma_x^2}}$$

In spectral clustering, one constructs an affinity or kernel matrix A from the data points and performs a spectral decomposition of A, possibly after normalization. Then the dominant eigenvalues and the corresponding eigenvectors are used for clustering the original data. Spectral clustering may be applied in particular in cases where simple algorithms such as K-means fail.

Our main contributions are: (1) we introduce a novel method for determining the optimal value of sigma by using weak affinity matrix ;(2) there are many existing techniques for comparison : the cross-validation, penalized likelihood estimation, permutation tests, resembling, and finding the knee of an error curve. we compare our method with error index.

The rest of this paper is organized as follows: Section 2 introduces some of the related works. Our proposed algorithm is explained in details in Section 3. Section 4 provides the simulation results and Section 5 gives the conclusions and future works.

2. Related Works

Determining the Number of Clusters/Segments: Five common approaches to estimating the dimension of a model are: cross-validation, penalized likelihood estimation, permutation tests, resembling, and finding the knee of an error curve. Cross-validation techniques create models that attempt to fit the data as accurately as possible. Monte Carlo cross validation [18, 17] has been successfully used to prevent over-fitting (too many clusters/segments). Penalized likelihood estimation also attempts to find a model that fits the data as accurately as possible, but also minimizes the complexity of the model. Permutation tests [22] attempt to prevent the creation of a PLA that over-fits the data by comparing the relative change in approximation error to the relative change of a random time series. If the errors are changing at a similar rate, then more segments would fit noise and not the underlying structure of the time series. Resembling [15] and Consensus Clustering [13] attempt to find the correct number of clusters by clustering many samples of the data set, and determining the number of clusters where clustering's of the various samples are the most "stable." Locating the "knee" of an error curve, in order to determine an appropriate number of clusters or segments, is well known, but it is not a particularly well-studied method. There are methods that statistically evaluate each point in the error curve, and use the point that either minimizes or maximizes some function as the number of clusters/segments to return. Such methods include the Gap statistic [21] and prediction strength [20].

Weak Affinity Matrix: The affinity matrix A is usually interpreted as the adjacency matrix of a graph. Thus spectral clustering algorithms are decomposed into two distinct stages: (a) build a good affinity graph and (b) and a good clustering of the graph. Determining the kernel parameter σ is a pivotal issue and greatly influences on the final clustering result. In some cases the right σ is obvious. However, generally it is non-trivial to find a good σ value. A possible method is trying different values for σ and choosing the one which optimizes some quality measure. The main issue is the construction of a good affinity matrix which is as block diagonal as possible. In that case spectral decomposition is merely the most convenient way to discover this block structure. In order to amplify the block structure of an affinity matrix, new method is called as conductivity method which used a weak affinity matrix where the affinities of many points might be close to zero. So there is weak affinity matrix for min sigma and strong affinity matrix for max sigma. By amplifying weak affinities in matrix A. Instead of considering two points similar if they are connected by a high-weight edge in the graph, we assign them a high affinity if the overall graph conductivity between them is high. The definition for conductivity as for electrical networks, *i.e.*, the conductivity of two points depends on all paths between them.

The connectivity method depends on finding sigma for every point and finding constant τ as shown in the following equation:

$$\sum_{j=1}^{n} \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma_i^2}\right) = \tau$$

The τ is referred as the fixed neighborhood size. Choosing τ is easier than choosing σ since τ is scale invariant. in contrast to σ the clustering result is not very sensitive to the value of τ. In order to obtain weak affinities, τ is set to a small value, such that only immediately neighboring points obtain a significant affinity. The best choosing of τ = 1+2D, where D is the dimension of the data there should be two neighbors for each dimension, plus one because the point is neighbor of itself [34].

3. Proposed Algorithm

In this paper we present an effective way to estimate σ parameter in Gaussian kernel function which has the form $(i,j) = e^{\frac{-d(i,j)^2}{2\sigma^2}}$. This function is essential for constructing a good informative affinity matrix in spectral algorithms, therefore the choice of this parameter is very important and change results dramatically.

Most of spectral algorithms substitute σ by roughly values like 0.5, 1 and 2, but these values are not probable for many datasets, and affect accuracy for spectral clustering algorithms, our algorithm is estimation for σ depending on the dataset itself.

The first stage for any spectral clustering algorithm is to construct similarity matrix which is called affinity matrix using Gaussian kernel function. The main motivation of clustering is to group similar objects in one cluster, and separate objects with less similarity in different clusters. The most important measurement of dissi-

milarity between objects is squared Euclidean distance, and the Gaussian kernel function can considered as a way to convert dissimilarity matrix to similarity one.

Similarity or affinity matrix is a symmetric matrix with NXN dimension where N is the number of samples, and the elements in this matrix have values between 0 and 1.

The i^{th} row in an affinity matrix represents the similarity between sample $x_i$and remainder samples from x_{i+1}tox_n, because the affinity matrix is symmetric with a diagonal which has value 1. So that we need to construct affinity matrix from elements on the right of the diagonal in distance matrix using Gaussian kernel function. We assume that for each i^{th}row of affinity matrix there is σ_i which is called parzen window. In other hand we need to generate two ranges of $\sigma_i$to construct a weak affinity matrix and a strong affinity matrix(1), supposing $\sigma_{i1}$for the weak affinity matrix and $\sigma_{i2}$for the strong affinity one.

To construct weak affinity matrix we need inasmuch as a small σ_{i1} as we can, but not too small that affects relations between neighborhood objects. To achieve that, σ_{i1}must be small and gives strong affinity to closest points. To be mentioned, less distance means more affinity and vice versa.

To have an estimation of σ_{i1}, we use an optimization technique, to get weak affinity we need to minimize $K(i,j) = e^{\frac{-d(i,j)^2}{2\sigma^2}}$, and as a result of the negative(minus sign) power of the exponential, we need to maximize the term $\frac{d(i,j)^2}{2\sigma^2}$ as we can. For this we must choose the dominator to be the farthest distance between $x_i$and remainder$x_j wherej = i+1\ ton$, so the term can be re-written as $\frac{\max d(i,j)^2)}{2\sigma_{i1}^2}$. In this case, the dominator is constant now, we can write Gaussian kernel as a function of $\sigma_{i1} k(i,\sigma_{i1}) = exp^{\frac{-maxd\ (i,j)^2}{2\sigma_{i1}^2}}$. Now we have to minimize $k(i,\sigma_{i1})$. To do that, we can use the simplest way of optimization techniques by taking the first derivative of the function respect to σ_{i1}

$$\frac{dk(i,\sigma_{i1})}{d\sigma_{i1}} = \frac{maxd\ (i,j)^2}{\sigma_{i1}^3} e^{\frac{-max\ d(i,j)^2}{2\sigma_{i1}^2}} \quad \text{.....(1)}$$

In the same way we can construct strong affinity matrix by maximizing $K(i,j) = e^{\frac{-d(i,j)^2}{2\sigma^2}}$, and to do that, we can simply minimize the term $\frac{d(i,j)^2}{2\sigma^2}$. So we choose the dominator to be the closest distance between $x_i$and remainder $x_j wherej = i+1\ toN$ .The term becomes $\frac{min\, d(i,j)^2}{2\sigma_{i2}^2}$.
As the previous one the dominator is constant for the object $x_i$we can write Gaussian kernel as a function of $\sigma_{i1} k(i,\sigma_{i2}) = exp^{\frac{-mind\ (i,j)^2}{2\sigma_{i2}^2}}$,and by the derivation we get formula

$$\frac{dk(i,\sigma_{i2})}{d\sigma_{i2}} = \frac{mind\ (i,j)^2}{\sigma_{i2}^3} e^{\frac{-min\ d(i,j)^2}{2\sigma_{i2}^2}} \quad \text{.....(2)}$$

In the fact we need one value of parzen window σ for each row *i*in the affinity matrix. To achieve that, we replace each $\sigma_{i1}$and $\sigma_{i2}$by one variable $\hat{\sigma}$, and to esti-

mate optimum value for $\hat{\sigma}$ we put eq1 and eq2 equal to zero (optimization technique). Now we have eq1=eq2=0, but not to be forgotten that we have one equation for one unknown$\hat{\sigma}$ as we can see in the following formula:

$$\frac{maxd\ (i,j)^2}{\hat{\sigma}_i^3} e^{\frac{-max\ d(i,j)^2}{2\hat{\sigma}_i^2}} = \frac{mind\ (i,j)^2}{\hat{\sigma}_i^3} e^{\frac{-min\ d(i,j)^2}{2\hat{\sigma}_i^2}} \quad \ldots\ldots\ldots(3)$$

By sampling eq3 we get $e^{\frac{\max\ d(i,j)^2 - \min\ d(i,j)^2}{2\hat{\sigma}_i^2} = \frac{max\ d(i,j)^2}{min\ d(i,j)^2}}$ by taking natural logarithm to both sides $\frac{max\ d(i,j)^2 - min\ d(i,j)^2}{2\hat{\sigma}_i^2} = \ln \frac{max\ d(i,j)^2}{min\ d(i,j)^2}$ so we get

$$\hat{\sigma}_i^2 = \frac{max\ d(i,j)^2 - min\ d(i,j)^2}{2 \ln \frac{max\ d(i,j)^2}{min\ d(i,j)^2}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(4)$$

Note that the last two rows in dissimilarity matrix(N-1 and N), N-1 has one variable after diagonal, and N row has no variable, so we need to calculate $\hat{\sigma}_i^2$i=1,2,3…….,N−2 , after constructing vector $\hat{\sigma}_i^2$ we need to calculate one value for σ^2 we can put it simply equals to $avg(\hat{\sigma}_i^2)$. Suppose that we have 12 samples $x_1, x_2, \ldots\ldots\ldots\ldots x_{12}$with 2 dimension as shown below in Table1 with distribution as shown in Figure 1.

Table 1

Sample	Attribute1	Attribute2
X_1	3	8
X_2	10	11
X_3	18	10
X_4	17	8
X_5	11	10
X_6	5	6
X_7	11	12
X_8	3	6
X_9	19	9
X_{10}	12	11
X_{11}	18	9
X_{12}	5	8

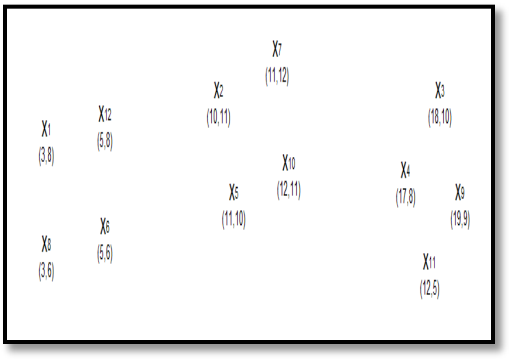

Figure 1. The Distribution of Samples

We calculate dissimilarity matrix between samples as below in Table 2.Then we calculate vector for σ_i*i=1 to 10* with our algorithm using equation (4) then using one $\sigma=avg(\overrightarrow{\hat{\sigma}_i})$, after that we get the following affinity matrix (Table 3). If you look to Table 3 which represents affinity, $x_1$it has

strong affinity with samples: x_6, $x_8$and x_{12}so these samples in one cluster, in other hand sample $x_2$has strong affinity with samples : x_5, $x_7$and x_{10}, so we put these samples in a new cluster.

Sample $x_3$has strong affinity with samples: x_4, $x_9$and x_{11}, these samples form a third cluster and clustering completes. In this example our algorithm construct very optimized affinity matrix and even lead us to do simple clustering with merge samples with strong affinity and obtain true 3 clusters without any prior information about number of clusters as shown below in Table 3.

So our algorithm is capable to construct a very informative affinity matrix and later can be used by any algorithm like spectral clustering to group very large datasets. In our optimized method for Gaussian kernel function the constructed affinity matrix has more than the affinity between samples and it has also an information about the number of clusters as we have shown in the example 1. Spectral clustering try to construct a block diagonal affinity matrix, but the accuracy of determination number of clusters which uses block diagonal(1) or Eigen gap needs a very informative initial affinity matrix that our optimized algorithm capable of do.

We have shown an approach to calculate σ as an average of $\hat{\sigma}_i$ vector. If we have prior information about desired number of clusters k, we can sort $\hat{\sigma}_i$ vector and partition it to k parts and estimate average for all parts to get k values of σ in addition to minimum and maximum values of the same vector to do different clustering with these different values of σ, and each time we use an error measurement like Sum of Squared Error or others and choose the clustering result with minimum error.

4. Simulation and Results

We evaluated our proposed algorithm on several artificial and real data sets.

4.1.Artificial Data Sets

We used two artificial data sets such as 6c2d(500 samples) and 2moon(300 samples) in our tests to show the performance of our algorithm using proposed σ.

a) We generated artificial data set with six clusters and 2 dimension for display, the data set distributed according to non-isotropic Gaussians with different variances as shown in figure 2. In Cluster 1 :70, Cluster 2 :60, Cluster 3 :80, Cluster 4 :90, Cluster 5 :100, Cluster 6 :100 samples. The distribution of samples is showed in Figure 2.

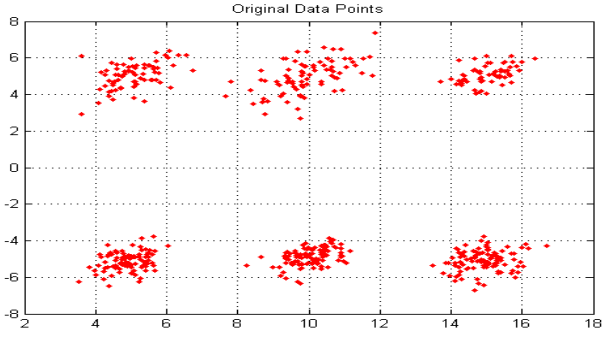

Figure 2. Artificial Data with Six Clusters

As shown in Figure 3 we have six clusters. These clusters are obtained by using our optimized sigma to find good affinity matrix. It is clear from affinity matrix wehavesixclusters, these clusters are shown in the block diagonal matrix in white square color as shown in Figure 4.

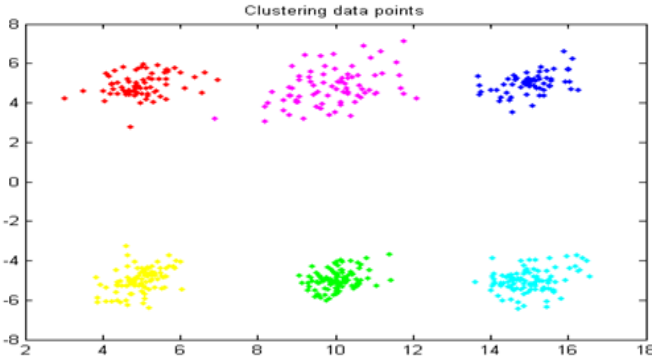

Figure 3. Clustering Dataset with Six Clusters

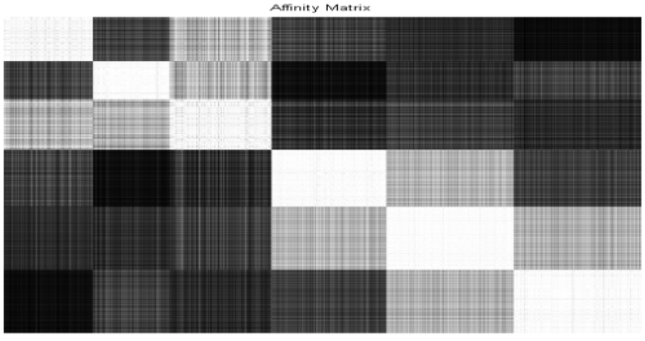

Figure 4.Affinity Matrix with Six Clusters

b) We generated two moons in 2 dimensions with Gaussian noise of variance 0.01 Number of points in Class 1: 155; Class 2: 145. The distribution of samples is shown Figure 5.

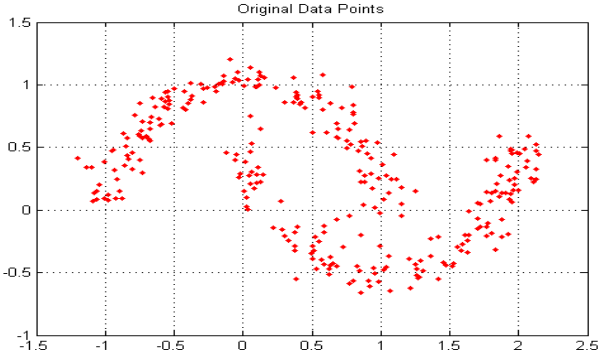

Figure 5.Artificial Data with 2 Clusters

As shown in Figure 6 we have two clusters. These clusters are obtained by using our optimized sigma to find good affinity matrix. It is clear from affinity matrix wehave two clusters, these clusters are shown in the block diagonal matrix in white square color as shown in Figure 7.

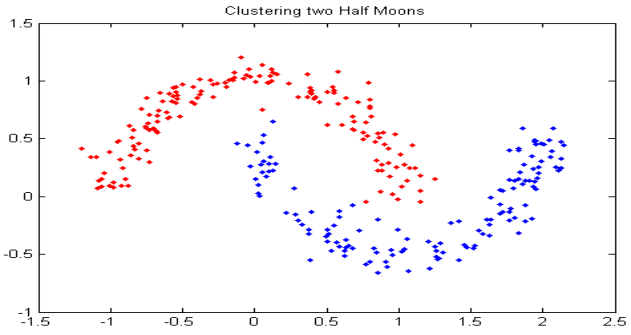

Figure 6. Clustering Dataset with Two Clusters

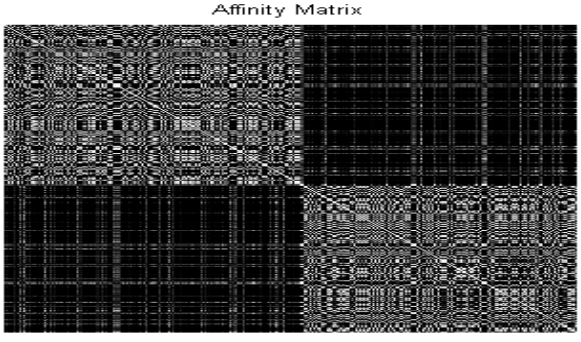

Figure7. Affinity Matrix with Two Clusters

c)We have very complexartificial data set with 2 dimensions and 395 samples as shown in Figure 8 with noise added. This data set is organized from two parts with six clusters. After we found the σ for the first time we used 2nd smallest Eigen vector[37,38] and we divide the complex data set into twoparts as shown in Figure 9.

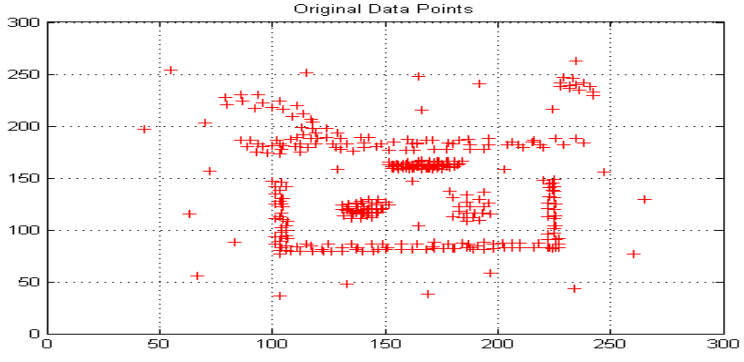

Figure 8. Artificial Data with Six Clusters

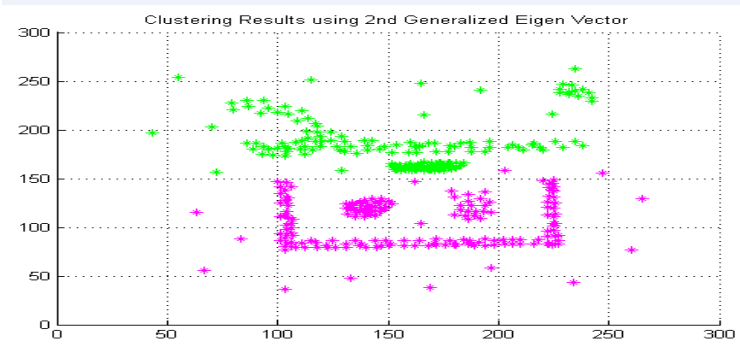

Figure 9. Clustering Dataset with 2 Parts

After thatwe found the new 2σ for two parts and we used k largest Eigen vector of a normalized affinity matrix[39] by using this technique the first part of the complex dataset was as shown in Figure 10, and the second part as shown in Figure 11.We see at the low part there is slightly error because we didn't processing the noise.

Figure 10.Clustering Dataset with First Part

So if we used sigma values from 0.5 to 100 we could't get accurate results,but by using our proposedσ we have accurate results as shown in clustering results.

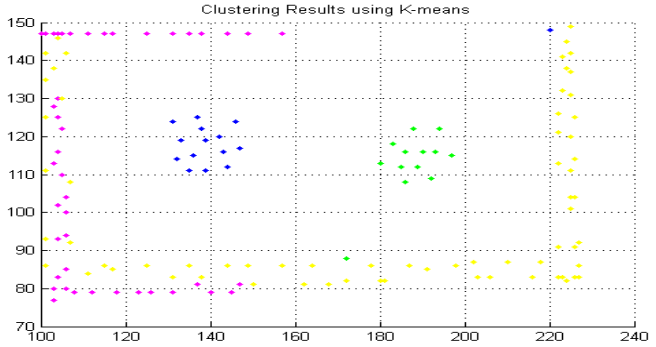

Figure 11. Clustering Dataset with Second Part

4.2. Real Data Sets

a) We used the iris data set from the UCI (http://archive.ics.uci.edu/ml/datasets/Iris) which contains three clusters, 150 data points with 4 dimensions, number of points cluster 1 :50 samples, cluster 2 :50 samples and cluster 3 :50 samples. AS shown in Figure 12we have affinity matrix with three clusters which shown in block diagonal matrix in white square color. These clusters are obtained by using our proposed sigma to find good affinity matrix.

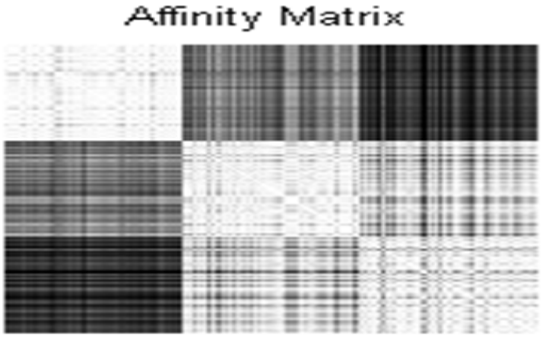

Figure 12. AffinityMatrix with Three Clusters

b) We used the Wine data set from the UCI (http://archive.ics.uci.edu/ml/datasets/wine) which contains three clusters, 178 data points with 13 dimensions. class 1 :59,class 2: 71, class 3 :48 samples. AS shown in Figure 13 we have affinity matrix with three clusters which shown in block diagonal matrix in white square color. These clusters are obtained by using our proposed sigma to find good affinity matrix

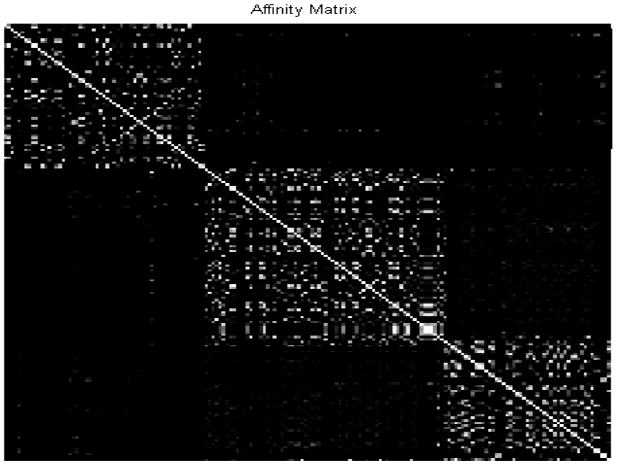

Figure 13. Affinity Matrix with Three Clusters

c) We test 210 samples with 7 images and 19 attributes. This dataset can be found at the following web site:

http://archive.ics.uci.edu/ml/datasets/Statlog+%28Image+Segmentation%29 Description of Datasets. AS shown in Figure 14 we have affinity matrix with seven clusters which shown in block diagonal matrix in white square color. These clusters are obtained by using our proposed sigma to find good affinity matrix.

Figure 14. Affinity Matrix with Seven Clusters

4.3. Error Index

We want to show our strong proposed sigma by measuring the performance using error index. So we will calculate the number of error from the total number of samples in percentage. In an artificial data set 6c2d with 500 samples we get an error index as 0 % by using our proposed sigma, when the sigma is substituted by (0.5,1,2) we get (0,0,0.4)% error index.In an artificial data set 2moon(300 samples)we have an error index as 2.66% by using our proposed sigma, when the sigma is substituted by (0.5,1,2) we get (16,23,33)% error index.In an artificial data set 6c2d(395) samples)we have an error index as 3.26% by using our proposed sigma when the sigma is substituted by (0.5,1,2) we get (99,99,99)% error index.In real data set iris (150 samples) we have an error index as 5.3 % by using our proposed sigma, when the sigma is substituted by (0.5,1,2) we get (45,32,32)% error index.In real data set wine (178 samples) we have an error index as 5.6 % by using our proposed sigma when the sigma is substituted by (0.5,1,2) we get (93,87,38)% error index.and in 7 images error index is 5.8% with our optimized algorithm but with traditional values of σ (0.5,1,2) we get (95,81,73)% error index. The results for artificial and real data set are shown in Table 4. Where c means # clusters, D means # dimension and S means # samples.

Table 4.Error Index

DATA SET	OUR OPTIMIZED σ%	SIGMA=0.5,1 AND 2%		
6C2D(500 S)	0	0	0	0.4
2MOON(300 S)	2.66	16	23	33
6C2D(395 S)	3.2	99	99	99
IRIS (150 S)	5.3	45	32	32
WINE (178 S)	5.6	93	87	83
7 IMAGES (210 S)	5.8	95	81	73

5. Conclusions and Future Works

In this paper we presented a new algorithm for estimating σ using optimization techniques, we constructed a vector $\vec{\hat{\sigma}_i}$, each $\hat{\sigma}_i$ corresponds to i[th]row in a dissimilarity matrix which is used to construct an affinity matrix using Gaussian kernel function. Our algorithm showed that choosing $\hat{\sigma}_i$ as the formula $\hat{\sigma}_i^2 = \frac{max\ d(i,j)^2 - min\ d(i,j)^2}{2\ln\frac{max\ d(i,j)^2}{min\ d(i,j)^2}}$ gives good accuracy in clustering result, and we introduced more than one approach to calculate global value for σ from this vector. The affinity matrix which is produced using our algorithm is very informative and contains addition information like the number of clusters.We hope on future we can develop our algorithm to do complete clustering without depending on other algorithms.

References

[1] M. Eirinaki and M. Vazirgiannis, "Web Mining for Web Personalization", ACM Transactions on Internet Technology (TOIT), vol. 3, no. 1,**(2003)**, pp. 1-27.
[2] B.BahmaniFirouzi, T. Niknam and M. Nayeripour, "A New Evolutionary Algorithm for Cluster Analysis", Proceeding of world Academy of Science, Engineering and Technology, vol. 36,**(2008)**December.
[3] A. Gersho and R. Gray, "Vector Quantization and Signal Compression",KulwerAcadimec, Boston, **(1992)**.

[4] M. Al- Zoubi, A. Hudaib, A. Huneiti and B. Hammo, "New Efficient Strategy to Accelerate k-Means Clustering Algorithm", American Journal of Applied Science, vol. 5, no. 9,**(2008)**, pp 1247-1250.
[5] M. Celebi, "Effecitive Initialization of K-means for Color Quantization", Proceeding of the IEEE International Conference on Image Processing,**(2009)**, pp. 1649-1652.
[6] M. Borodovsky and J. McIninch, "Recognition of genes in DNA sequence with ambiguities",Biosystems, vol. 30, no. 1-3,**(1993)**, pp. 161-171.
[7] J. Bezdek and N. Pal, "Fuzzy Models for Pattern Recognition", IEEE press, New York, NY, USA, **(1992)**.
[8] J. Bezdek, "Pattern Recognition with Fuzzy Objective Function Algorithms", Plenum Press, New York, NY, USA, **(1981)**.
[9] G. Gan, Ch. Ma and J. Wu, "Data Clustering: Theory, Algorithms, and Applications",ASA-SIAM series on Statistics and Applied Probability, SIAM, **(2007)**.
[10] D. Defays, "An Efficeint Algorithm for A Complete Link Method",The Computer Journal, vol. 20,**(1977)**, pp. 364-366.
[11] R. Sibson, "SLINK: an Optimally Efficent Algorithm for the Single Link Cluster Method",The Computer Journal, vol. 16,no. 1,**(1973)**, pp. 30-34.
[12] L. Kaufman and P. Rousseeuw, "Finding Groups in Data: an J. MacQueen, Some Methods for Classification and Analysis of Multivariate Observations", 5th Berkeley Symp. Math. Statist. Prob., vol. 1,**(1967)**, pp. 281-297.
[13] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy, "Advances in Knowledge Discovery and Data Mining",AAAI/MIT press, **(1996)**.
[14] X. Wu, "Top 10 Algorithms in Data Mining", Journal of Knowledge and Information Systems, vol. 14,no. 1-37, **(2008)**.
[15] S. Salvador and P. Chan, "Determining the Number of Clusters/Segments in Hierarchical Clustering/Segmentation Algorithms", Florida Institute of Technology, **(2003)**.
[16] G. Seikholeslami, S. Chatterjee and A. Zhang,"WaveCluster A Multi-Resolution Clustering Approach for Very Large Spatial Databases", Proceedings of the 24th VLDB Conference, **(1998)**.
[17] P. Smyth, "Clustering Using Monte-Carlo Cross-Validation", Proc. 2nd KDD, **(1996)**, pp.126-133.
[18] E. Forgy, "Cluster Analysis of Multivariate Data: Efficiency vs. Interpretability of Classification", Biometrics, vol. 21, **(1965)**.
[19] J. Z.C. Lai and T. J. Huang, "Fast Global Kmeans Clustering Using Cluster Membership and Inequality", Pattern Recognition, vol. 43,**(2010)**, pp. 1954-1963.
[20] G. Frahling and C. Sohler, "A Fast Kmeans Implementation Using Coresets", International Journal of Computational Geometry and Applications, vol. 18, no. 6,**(2008)**, pp. 605-625.
[21] L. Taoying and Y. Chen, "An Improved Kmeans for Clustering Using Entropy Weighting measures", 7th World Congress on Intelligent Control and Automation, **(2008)**.
[22] S. Gupara, K. Rao and V. Bhatnagar, "Kmeans Clustering Algorithm for Categorical Attributes", Proceeding 1st International Conf. on Data Werehousing and Kownledge Discovery, Italy, **(1999)**, pp. 203-208.
[23] Z. Huang, "Extensions to The Kmeans Algorithms for Clustering Large Data Sets with Categorical Values", Data Mining Knowledge Discovery, vol. 2,**(1998)**, pp. 283-304.
[24] W. Barbakh and C. Fyfe, "Inverse Weighted Clustering Algorithm", Computing and Information Systems, ISSN 1352-9404, vol. 11, no. 2,**(2007)**May, pp. 10-18.
[25] W. Barbakh, "The Family of Inverse Exponential Kmeans Algorithms," Computing and Information Systems,ISSN 1352-9404, vol. 11, no. 1, **(2007)**February, pp. 1-10.
[26] W. Barbakh and C. Fyfe, "Clustering and Visualization with Alternative Similarity Functions", The 7th WSEAS international conference on artificial intelligence, knowledge, engineering and data bases, AIKED'08, University of Cambridge, UK,**(2008)**, pp. 238–244.
[27] W. Barbakh and C. Fyfe, "Online Clustering Algorithms", International Journal of Neural Systems (IJNS), vol. 18, no. 3, **(2008)**, pp. 185-194.
[28] I.S. Dhillon, Y. Guan and B. Kulis, "Kernel Kmeans, Spectral Clustering and Normalized Cuts", ACM SIGKDD international conference Knowledge Discovery and Data Mining. Seattle, WA, **(2004)**.
[29] M. Girolami, "Mercer Kernel Based Clustering in Feature Space", IEEE Transactionson Neural Networks, vol. 13, no. 3,**(2002)**, pp. 780-784.

[30] B. Scholkopf, A. Smola and K.R. Muller, "Nonlinear Component Analysis As a Kernel Eigenvalue Problem", Neural Computation,vol. 10,**(1998)**, pp. 1299-1319.
[31] J. Suykens and J. Vandewalle, "Least Squares Support Vector Machine Classifiers", Neural Processing Letters, vol. 9, no. 3,**(1999)**, pp. 293-300.
[32] R.Tibshirani, G. Walther, D. Botstein and P. Brown,"Cluster Validation by Prediction Strength", Technical Report, 2001-21, Dept. of Biostatistics, Stanford Univ, **(2001)**.
[33] R. Tibshirani, G. Walther and T. Hastie,"Estimating the number of clusters in a dataset via th e Gap statistic, In JRSSB, **(2003)**.
[34] K. Vasko and T. Toivonen,"Estimating the number of segments in time series data using permutation tests", Proc. IEEE Intl. Conf.on Data Mining,**(2002)**, pp. 466-473.
[35] T.Zhang, R. Ramakrishnan and M. Livny,"BIRCH: An Efficient Data Clustering Method for Very Large Databases", In ACM.
[36] http://crsouza.blogspot.com/2010/03/kernel-functions-for-machine-learning.html.
[37] http://researchonsearch.blogspot.com/2009/07/spectral-graph-partitioning.html.
[38] M. Saerens, F. Fouss, L. Yen and P. Dupont, "The principal component analysis of a graph, and its relationship to spectral clustering", Proceedings of the 15th European Conference on Machine Learning (ECML 2004) (pp. 371{383). Springer, **(2004)**.
[39] I. Fischer and J. Poland,"Amplifying the Block Matrix Structure for Spectral Clustering", Telecommunications Lab. **(2005)**.
[40] J. Shi and J. Malik, "Normalized Cuts and Image Seg mentation", Proc. IEEE Conf. Computer Vision and Pattern Recognition,**(1997)**, pp. 731-737.
[41] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation", IEEE TransactionsonPAMI, vol. 22, no. 8, **(2000)**August.
[42] A. Ng, M. Jordan and Y. Weiss, "On spectral clustering: analysis and an algorithm", T. Dietterich, S. Becker, and Z. Ghahramani (Eds.), Advances in Neural Information Processing Systems 14, MIT Press, **(2002)**,pp. 849-856.

Table 2

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12
X1	0	58	229	196	68	8	80	4	257	90	226	4
X2	58	0	65	58	2	50	2	74	85	4	68	34
X3	229	65	0	5	49	185	53	241	2	37	1	173
X4	196	58	5	0	40	148	52	200	5	34	2	144
X5	68	2	49	40	0	52	4	80	65	2	50	40
X6	8	50	185	148	52	0	72	4	205	74	178	4
X7	80	2	53	52	4	72	0	100	73	2	58	52
X8	4	74	241	200	80	4	100	0	265	106	234	8
X9	257	85	2	5	65	205	73	265	0	53	1	197
X10	90	4	37	34	2	74	2	106	53	0	40	58
X11	226	68	1	2	50	178	58	234	1	40	0	170
X12	4	34	173	144	40	4	52	8	197	58	170	0

Table 3

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12
X1	1	0.25618	0.00462	0.01003	0.20256	0.82874	0.15282	0.91035	0.00239	0.12084	0.00495	0.91035
X2	0.25618	1	0.21735	0.25618	0.95412	0.30911	0.95412	0.17594	0.13589	0.9105	0.20256	0.45007
X3	0.00462	0.21735	1	0.88922	0.31646	0.01298	0.28809	0.00348	0.95412	0.4196	0.97679	0.01721
X4	0.01003	0.25618	0.88922	1	0.39093	0.03095	0.29493	0.00913	0.88922	0.4507	0.95412	0.03400
X5	0.20256	0.95412	0.31646	0.39093	1	0.29493	0.91035	0.15282	0.21735	0.9541	0.30911	0.39093
X6	0.8287	0.30911	0.01298	0.03095	0.29493	1	0.18440	0.91035	0.00811	0.1759	0.01530	0.91035
X7	0.15282	0.95412	0.28809	0.29493	0.91035	0.18440	1	0.09555	0.18012	0.9541	0.25618	0.29493
X8	0.91035	0.17594	0.00348	0.00913	0.15282	0.91035	0.09555	1	0.00198	0.0829	0.00410	0.82874
X9	0.00239	0.13589	0.95412	0.88922	0.21735	0.00811	0.18012	0.00198	1	0.2880	0.97679	0.00979
X10	0.12084	0.91035	0.41946	0.45007	0.95412	0.17594	0.95412	0.08299	0.28809	1	0.39093	0.25618
X11	0.004959	0.202568	0.976793	0.954124	0.309119	0.015306	0.25618	0.004109	0.976793	0.3909	1	0.018468
X12	0.910353	0.450076	0.017212	0.034007	0.390931	0.910353	0.294938	0.828743	0.009797	0.2561	0.018468	1