A Weighted Majority Voting based on Normalized Mutual Information for Cluster Analysis

Meshal Shutaywi and Nezamoddin N. Kachouie

Department of Mathematical Sciences, Florida Institute of Technology

Abstract

Due to advancements in data acquisition, large amount of data are collected in daily basis. Analysis of the collected data is an important task to discover the patterns, extract the features, and make informed decisions. A vital step in data analysis is dividing the objects (elements, individuals) in different groups based on their similarities. One way to group the objects is clustering. Clustering methods can be divided in two categories, linear and non-linear. K-means is a commonly used linear clustering method, while Kernel K-means is a non-linear technique. Kernel K-means projects the elements to a new space using a kernel function and then group them in different clusters. Different kernels perform differently when they are applied to different data sets. Choosing the right kernel for an application could be challenging, however applying a set of kernels and aggregating the results could provide a robust performance for different data sets. In this work, we address this issue and propose a weighted majority voting to ensemble the results of three different kernels.

Introduction

Cluster analysis in one of the most common unsupervised learning methods in the machine learning which is used to discover underlying patterns or grouping in data. Cluster analysis is performed to discover distinct individuals that share common features within a large population and group them in the same cluster (Monti et al. 2003). Clustering has been increasingly used in the past decades to address multidisciplinary problems as an important step in data analysis (Jain, Murty, and Flynn 1999) and data mining (Dhillon, Guan, and Kulis 2004). Clustering techniques attempt to find an optimal solution based on a clustering criterion (Nguyen and Caruana 2007). To this end, several clustering methods have been developed (Jain, Murty, and Flynn 1999) such as K-means, Fuzzy c-means, mixture models, and spectral clustering.

In this paper, we introduce a weighted majority voting clustering based on normalized mutual information (NMI). We aggregate the results of several clustering methods to conclude a final clustering results. The focus of this study is on the class of K-means clustering methods including K-means, K-means++, and kernel K-means. K-means is a well known clustering method that is commonly used for clustering analysis. It is based on minimizing Euclidean distance between each point and the center of the cluster to which it belongs. The advantages of K-means are its simplicity and speed while it suffers from random initialization of cluster centers (Arthur and Vassilvitskii 2007). K-means++ is an extension of K-means to address its shortcoming of random initialization (De La Vega et al. 2003), (Har-Peled and Mazumdar 2004), (Kumar, Sabharwal, and Sen 2004), and (Matoušek 2000). K-means can discover clusters that are linearly separable. Kernel K-means is a non-linear extension of K-means clustering method. Kernel K-means clustering, as the name implies, uses a Kernel function to project nonlinearly separable clusters into a space to make them linearly separable.

Method

We study K-means based clustering techniques and aggregate their results to provide a robust clustering analysis method. These clustering methods include K-means, K-means++, and kernel K-means with three different kernels.

K-means

K-means choose K centers such that the total squared distance between each point and its cluster center is minimized. K-means technique can be summarized by first selecting K arbitrary centers, which are usually, as Lloyds algorithm suggests, uniformly selected at random from the data. Second, each point is assigned to a cluster that is closest to it, and this is determined by calculating the Euclidean distance between each point and the cluster centers. Third, each new cluster center is calculated based on the average of all points belong to that cluster. Finally, the second and third steps are repeated until the algorithm reaches stability. K-means objective function can be written as $\sum_{j=1}^{k} \sum_{x_i \in \pi_j} \|x_i - \mu_j\|^2$,where π_j is the cluster j, and $\|.\|$ is denoted for Euclidean distance throughout the paper.

K-means++

K-means++ is based on a particular way of choosing centers for the K-means algorithm. Suppose that n data points $\chi \subset \mathbb{R}^d$ are given, and there are K clusters. K-means++

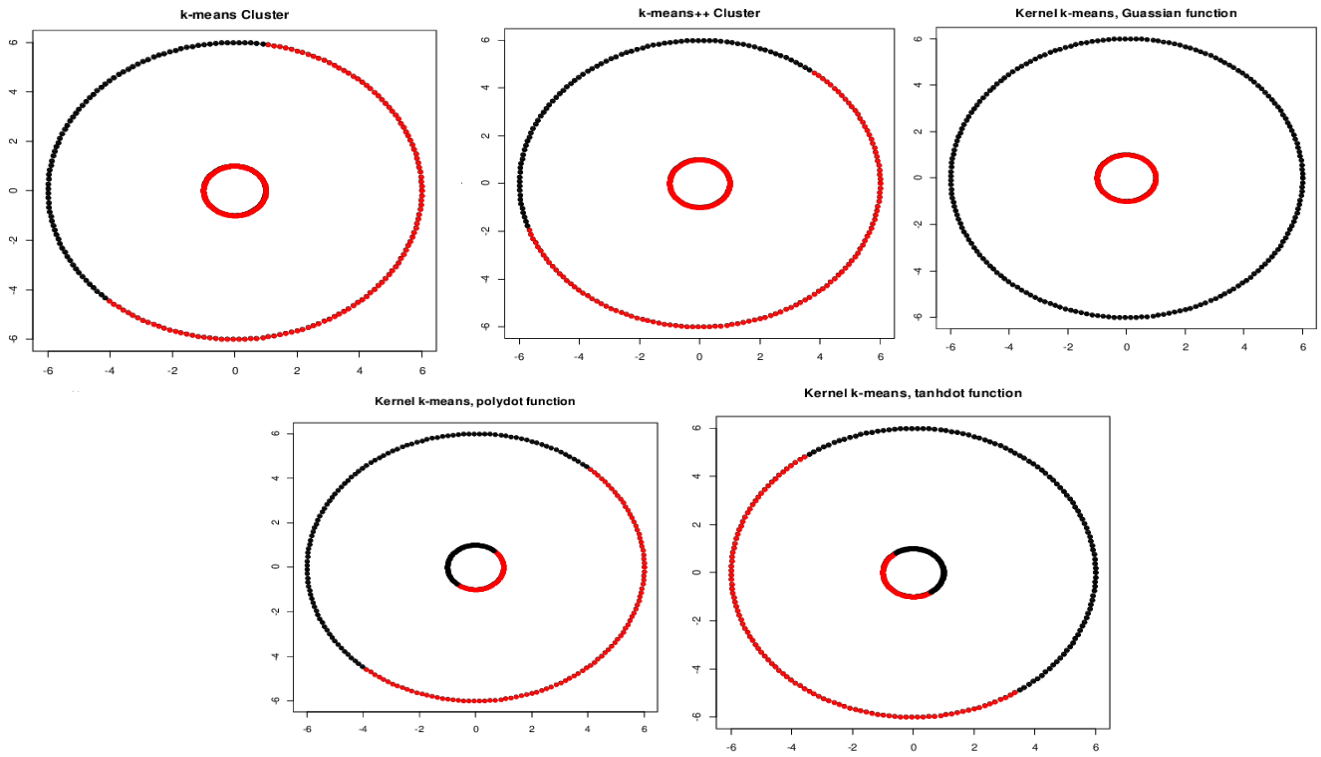

Figure 1: Clustering results obtained by applying K-means, K-mean++, and kernel K-means with Gaussian, polynomial, and tangent kernel functions to two noiseless inner circles.

Table 1: NMI scores for different clustering methods for two noiseless inner circles, and associated weights for kernel K-means with the three kernel functions.

	K-means	K-means++	Kernel K-means		
			Gaussian	Polynomial	Hyperbolic Tangent
Two noiseless inner circles	0.278	0.278	1	0	0
Weights			1	0	0

algorithm can be summarized as follows (Arthur and Vassilvitskii 2007): First, choose one center uniformly at random from the data points χ. Second, compute $D(x)$ for each data point x, where $D(x)$ is the distance between x and the nearest center that has already been chosen. Third, choose one new data at random as a new center with probability $\frac{D(x)^2}{\sum_{x \in \chi} D(x)^2}$. Then, repeat the second and third steps until K centers have been chosen. After obtaining the initial centers, proceed as the standard k-means clustering.

Kernel K-means

Kernel K-means separates clusters that are nonlinearly separable. The idea of kernel K-means clustering relies on transforming the data into a higher-dimensional feature space using a nonlinear function to project the points such that they will be linearly separable in the projected space. Kernel K-means algorithm follows (Dhillon, Guan, and Kulis 2004):

1. Let $\{x_1, x_2, ..., x_n\}$ be the set of data points, k be the number of clusters, π_j be the cluster j, $\{\pi_j\}_{j=1}^{k}$ be a partitioning of points, and ϕ be the non-linear function. Then, kernel matrix K can be constructed. Each elements in the matrix is a dot-product in the kernel defined features space as follow,

$$\kappa(x_i, x_z) = \phi(x_i).\phi(x_z), \qquad i, z = 1, 2, ..., n. \quad (1)$$

   where $\phi(x_i)$ denotes the data point x_i in transformed space. The dot products $\phi(x_i).\phi(x_z)$ are computed using kernel function κ. Some popular kernel functions are Radial Basis Function (RBF) (Campbell 2001) known as gaussian, polynomial, and sigmoid.

2. Randomly initialize each cluster center.

3. Compute Euclidean distance from each data point to the

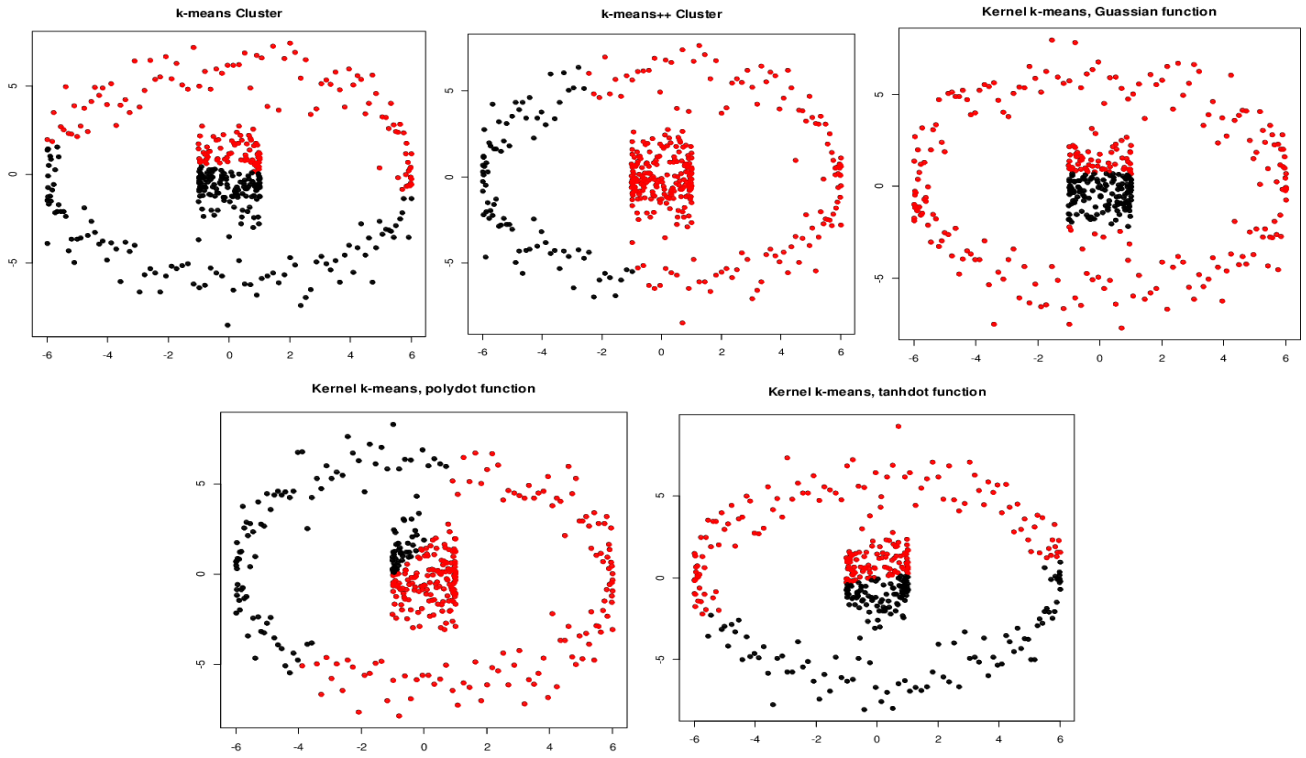

Figure 2: Clustering results obtained by applying K-means, K-mean++, and kernel K-means with Gaussian, polynomial, and tangent kernel functions to two noisy inner circles.

Table 2: NMI scores for different clustering methods for two noisy inner circles, and associated weights for kernel K-means with the three kernel functions.

	K-means	K-means++	Kernel K-means		
			Gaussian	Polynomial	Hyperbolic Tangent
Two noisy inner circle	0.068	0.062	0.769	0.011	0.001
Weights			0.985	0.014	0.001

cluster center μ_j in the transformed space as follow:

$$\begin{aligned}\phi(x_i) - \mu_j &= \phi(x_i) - \sum_{x_i \in \pi_j} \frac{\phi(x_i)}{|\pi_j|} \\ &= \phi(x_i).\phi(x_i) - \frac{2\sum_{x_z \in \pi_j} \phi(x_i).\phi(x_z)}{|\pi_j|} \\ &\quad + \frac{2\sum_{x_z, x_c, \in \pi_j} \phi(x_z).\phi(x_c)}{|\pi_j|^2} \end{aligned} \quad (2)$$

where $|\pi_j|$ is the number of elements in cluster π_j.

4. Assign data points to that cluster whose distance is minimized.
5. Compute the new cluster centers μ_j as the average points in transformed space belong to cluster π_j as

$$\mu_j = \sum_{x_i \in \pi_j} \frac{\phi(x_i)}{|\pi_j|}, \qquad j = 1, 2, ..., k \quad (3)$$

6. Repeat from step 3.

So, the objective function of kernel K-means is defined as:

$$D(\{\pi_j\}_{j=1}^k) = \sum_{j=1}^{k} \sum_{x_i \in \pi_j} \|\phi(x_i) - \mu_j\|^2 \quad (4)$$

After applying the aforementioned clustering methods, Normalized Mutual Information (NMI) is used to evaluate their performance (Dhillon, Guan, and Kulis 2004). NMI is calculated from a confusion matrix, whose entry $(i, j), n_i^{(j)}$ represents the number of points in cluster i and true class j, as follow

$$\frac{2\sum_{l=1}^{k}\sum_{h=1}^{c} \frac{n_l^{(h)}}{n} log \frac{n_l^{(h)} n}{\sum_{i=1}^{k} n_i^{(h)} \sum_{i=1}^{c} n_l^{(i)}}}{H(\pi) + H(\zeta))}, \quad (5)$$

where c is the number of classes, $H(\pi) = -\sum_{i=1}^{k} \frac{n_i}{n} log \frac{n_i}{n}$, and $H(\zeta) = -\sum_{j=1}^{c} \frac{n^{(j)}}{n} log \frac{n^{(j)}}{n}$. Also, $n_i, n^{(j)}$, and n are

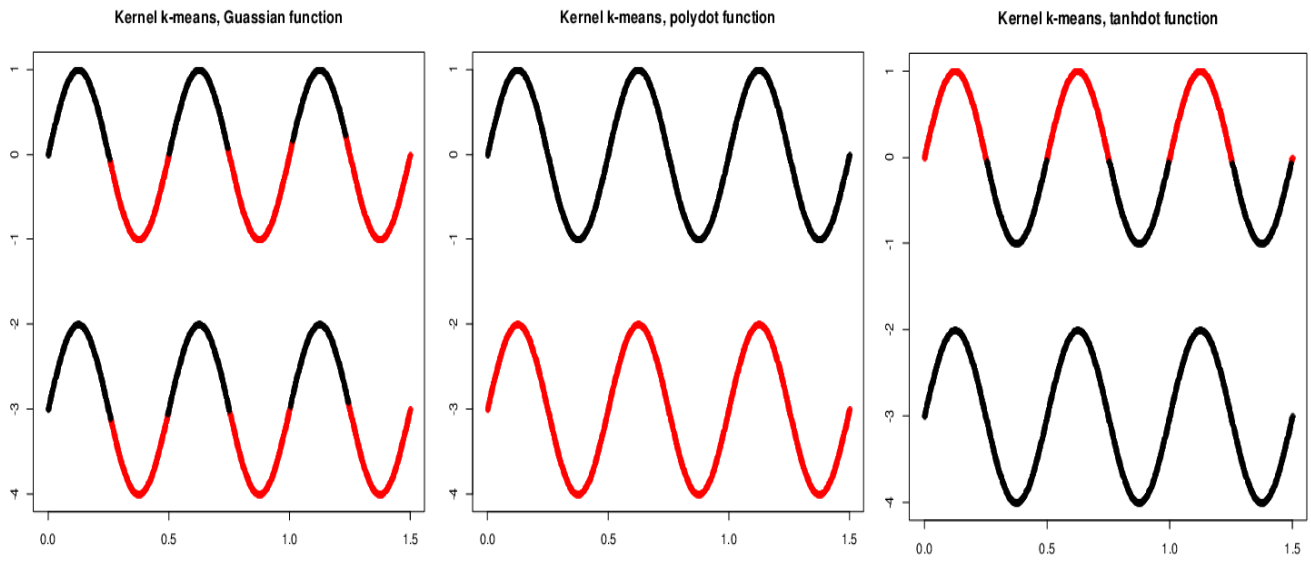

Figure 3: Clustering results obtained by applying kernel K-means with Gaussian, polynomial, and tangent kernel functions to two noiseless inner sine waves.

Table 3: NMI scores for different clustering methods for two noiseless sine waves, and associated weights for kernel K-means with three kernel functions.

	Kernel K-means		
	Gaussian	Polynomial	Hyperbolic Tangent
Two noiseless inner sine waves	0.385	1	0.351
Weights	0.222	0.576	0.202

Table 4: NMI scores for several clustering methods for two noisy separate sine waves, and associated weights for kernel K-means with the three kernel functions.

	Kernel K-means		
	Gaussian	Polynomial	Hyperbolic Tangent
Two noisy inner sine waves	0.61	0.785	0.347
Weights	0.35	0.451	0.199

total points in the i^{th} cluster, total points in the j^{th} class, and total sample size, respectively.

NMI value ranges from zero to one. High NMI value means that the true classes and identified clusters are consistent. That is, most of the observations in the same class are clustered in the same cluster (Dhillon, Guan, and Kulis 2004). Next is an explanation of weighted majority voting.

Weighted Majority Voting of Clustering Algorithms

Majority voting is based on the idea that the judgment of a group is superior to those of individuals. Majority voting approach has been used in supervised learning (classification) to combine classifiers so that more accurate results are produced. The underlining assumption is that neighboring samples within a "natural" cluster are very likely to be co-located in the same group by a clustering algorithm.

The algorithms can be summarized as follow: considering the first clustering method, a data of size n is partitioned , and paired of samples are voted for association. Then, the results of clustering method are mapped into a co-association matrix of size $n \times n$, whose $(i,j)^{th}$ element (at the end) represents the number of time the given sample pair has co-occurred in a cluster. Each co-occurrence is considered a vote toward their being in the same cluster. The previous steps are repeated for each clustering algorithm considered, with keeping track (accumulating) of the co-association matrix. The co-association matrix is normalized by dividing its elements by the number of methods. Then, majority voting associations are found for each sample pair (x_i, x_j) by comparing the the $(i,j)^{th}$ element in the association matrix ($(i,j)^{th}$normalized vote) with the fixed threshold 0.5. If it is greater than 0.5, then the sample pair is joined in the same cluster; if the sample pair were in different previously formed clusters, join the clusters. For each remaining sample not included in a cluster, form a single element cluster (Fred 2001).

In majority voting combination of general clustering algorithms, if a particular sample pair is voted to be located in cluster A using, for example, three clustering methods while it is voted to be located in cluster B using two methods, then this pair will be in cluster A even if the correct cluster is

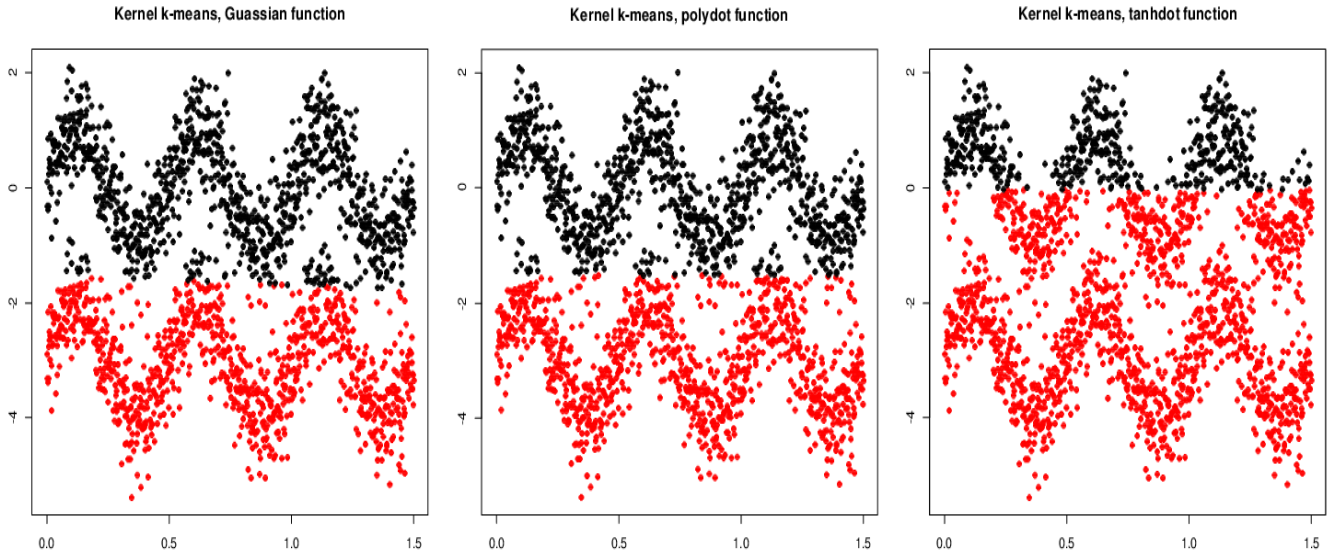

Figure 4: Clustering results obtained by applying kernel K-means with Gaussian, polynomial, and tangent kernel functions to two noisy sine waves.

cluster B. To overcome this shortcoming, we introduce the concept of using "weights" based on normalized mutual information (NMI) with majority voting algorithm.

After obtaining the clustering results of kernel K-means for the three kernel functions, we combine these results using the proposed weighted majority voting where the weights are computed using NMI. Suppose that ζ_1, ζ_2, and ζ_3 are the NMI's associated with Gaussian kernel, polynomial kernel, and hyperbolic tangent kernel respectively. Let w_1, w_2 and w_3 be the weights associated with these three kernels respectively and are computed by

$$w_i = \frac{\zeta_i}{\sum_{j=1}^{3} \zeta_j}, \qquad i = 1, 2, 3. \tag{6}$$

such that

$$\sum_{i=1}^{3} w_i = 1 \tag{7}$$

Simulation

Kernel K-means clustering method with three different kernels is compared with K-means and K-means++. Two-dimensional simulations are generated for two nonlinear clusters. First simulation contains two noiseless inner circles, while the second simulation is the noisy version of the first simulation. Third and forth simulations are noiseless and noisy inner sine waves, respectively. After generating the simulated data, the clustering methods including K-means, K-means++, and kernel K-means with three different kernel functions (Gaussian, polynomial, and hyperbolic tangent) are applied. Each simulation has 100 iterations and an NMI score is computed for each cluster method after each iteration. The results are then combined using the proposed weighted majority voting.

Results

We have applied three clustering methods, K-means, K-means ++, and kernel K-means with three different kernels to the simulated data and calculated NMI for each method. Next, the weights based on NMI for each kernel function is computed using Eq. (6).

First, we perform the cluster methods with two noiseless inner circles and compute the corresponding NMI's. Table 1 shows the NMI scores for the considering clustering methods with two noiseless inner circles. Based on NMI scores, Kernel K-means clustering performs the best with Gaussian kernel function as its NMI score is 1. Also, figure 1 shows the resulted clusters for different methods for this simulation where each color represents a cluster. It is clear from the top right of the figure 1 that Gaussian kernel was able to detect clusters that are nonlinearly separable as two inner circles are clustered in different clusters. NMI scores for polynomial kernel and hyperbolic tangent kernel functions are zeros since half of the observations in the first cluster come from first class and the rest of observations come from the second class as it can be seen on the bottom of the figure 1. It is clear that kernel K-means performance depends on the selected kernel function. So, it is imperative to aggregate the clustering results of these kernel functions to produce robust outcomes. We have combined the results by computing the associated weight to each kernel based on NMI score. Gaussian kernel receives a weight of 1 since the NMI's associated with the other two kernels are zero (Table 1). Since Gaussian kernel has the weight of one for this simulation, the aggregated clustering result by performing majority voting is the same as kernel K-means with Gaussian kernel. That is, regardless of the kernel function, the aggregated results is the optimal solution. As these clusters are not linearly separable, K-means and K-means++ perform poorly with low

NMI's trying to linearly separate the clusters in two groups as it is depicted in figure 1.

Even when the inner circles are corrupted with noise, kernel K-means with Gaussian kernel performance is robust with NMI of 0.769 (Table 2). Other methods are not able to separate the noisy inner circles into two clusters successfully with corresponding low NMI's (figure 2). Similar to the first simulation the aggregated result is the same as clustering results of Gaussian kernel. It is clear that Gaussian kernel is performing better than other kernel functions, and it gets the highest weight based on its NMI score.

In the next simulation two inner sine waves with the same frequency are generated. Kernel K-means with polynomial kernel outperforms Gaussian and tangent kernels as it can be seen in the clustering results (Figure 3) as well as the NMI scores (Table 3). The computed weights using NMI scores are 0.222, 0.576, and 0.202 for Gaussian, polynomial, and tangent kernels respectively.

Next, the inner sine waves are corrupted with noise and kernel K-means with different kernels are applied to this noisy signal (figure 4). The NMI scores are reported in the Table 4. Polynomial kernels perform better than the other kernels with NMI scores 0.785. The aggregated result is similar to the clustering results of polynomial kernel.

Conclusion

An important task in data analysis is dividing data into different groups based on their similarities by discovering underlying patterns and extracting features. To this end several clustering methods have been introduced for cluster analysis. K-means and its extensions are broadly used for cluster analysis. While K-means can identify the clusters that are linearly separable, Kernel K-Means is introduced to separate the clusters that are not linearly separable by projecting the data elements to a new space using a kernel function in which the groups are linearly separable. However, different kernel functions do not perform the same when they are applied to different data sets. Therefore, choosing the right kernel for an application is a challenging task. To address this, one can apply a set of kernels and aggregate the results to provide a robust performance for different data sets. In this study, we introduced a weighted majority voting to combine the clustering results of three different kernels. The proposed method provides promising results since the weights proposed to be assigned to kernel functions work appropriately, and we are going to apply it to real applications where the data cannot be linearly separated in different groups such as Genomic datasets as well as climate data.

References

Arthur, D., and Vassilvitskii, S. 2007. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, 1027–1035. Society for Industrial and Applied Mathematics.

Campbell, C. 2001. An introduction to kernel methods. *Studies in Fuzziness and Soft Computing* 66:155–192.

De La Vega, W. F.; Karpinski, M.; Kenyon, C.; and Rabani, Y. 2003. Approximation schemes for clustering problems. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, 50–58. ACM.

Dhillon, I. S.; Guan, Y.; and Kulis, B. 2004. Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 551–556. ACM.

Fred, A. 2001. Finding consistent clusters in data partitions. In *International Workshop on Multiple Classifier Systems*, 309–318. Springer.

Har-Peled, S., and Mazumdar, S. 2004. On coresets for k-means and k-median clustering. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, 291–300. ACM.

Jain, A. K.; Murty, M. N.; and Flynn, P. J. 1999. Data clustering: a review. *ACM computing surveys (CSUR)* 31(3):264–323.

Kumar, A.; Sabharwal, Y.; and Sen, S. 2004. A simple linear time (1+/spl epsiv/)-approximation algorithm for k-means clustering in any dimensions. In *Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on*, 454–462. IEEE.

Matoušek, J. 2000. On approximate geometric k-clustering. *Discrete & Computational Geometry* 24(1):61–84.

Monti, S.; Tamayo, P.; Mesirov, J.; and Golub, T. 2003. Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Machine learning* 52(1):91–118.

Nguyen, N., and Caruana, R. 2007. Consensus clusterings. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, 607–612. IEEE.