CLUSTERING SPATIAL DATA USING A KERNEL-BASED ALGORITHM

A. Majid Awan, Mohd. Noor Md. Sap

Email: *awanmajid@siswa.utm.my*

Abstract

This paper presents a method for unsupervised partitioning of data using kernel methods which offer strength to deal with complex data non-linearly separable in input space. This work gets inspiration from the notion that a non-linear data transformation into some high dimensional feature space increases the possibility of linear separability of the patterns in the transformed space. Therefore, it simplifies exploration of the associated structure in the data. Kernel methods implicitly perform a non-linear mapping of the input data into a high dimensional feature space by replacing the inner products with an appropriate positive definite function. Firstly, in this paper, selective kernel-based clustering techniques are analyzed and their shortcomings are identified especially for spatial data analysis. Finally, we present a robust weighted kernel k-means algorithm incorporating spatial constraints for clustering spatial data as a case study. The proposed algorithm can effectively handle noise, outliers and auto-correlation in the spatial data. Therefore, this work comes up with new clustering algorithm using kernel-based methods for effective and efficient data analysis by exploring structures in the data.

Keywords:

Clustering algorithms, K-means, Kernel methods, spatial data, unsupervised learning

1.0 Introduction

Data clustering, a class of unsupervised learning algorithms, is an important and applications-oriented branch of machine learning. Its goal is to estimate the structure or density of a set of data without a training signal. It has a wide range of general and scientific applications such as data compression, unsupervised classification, image segmentation for computer vision, anomaly detection, etc. There are many approaches to data clustering that vary in their complexity and effectiveness, due to the wide number of applications that these algorithms have. While there has been a large amount of research into the task of clustering, currently popular clustering methods often fail to find high-quality clusters.

A number of kernel-based learning methods have been proposed in recent years [9, 15, 3, 21, 8, 16, 7]. However, much research effort is being put up for improving these techniques and in applying these techniques to various application domains. Generally speaking, kernel function implicitly defines a non-linear transformation that maps the data from their original space to a high dimensional space where the data are expected to be more separable. Consequently, the kernel methods may achieve better performance by working in the new space. While powerful kernel methods have been proposed for supervised classification and regression problems, the development of effective kernel method for clustering, aside from a few tentative solutions [9, 4, 17], needs further investigation.

Finding good quality clusters in spatial data (e.g, temperature, precipitation, pressure, etc) is more challenging because of its peculiar characteristics such as auto-correlation, non-linear separability, outliers, noise, high-dimensionality, and when the data has clusters of widely differing shapes and sizes [18, 22, 11]. With this in view, the intention of this paper is, firstly, to analyze selective kernel-based clustering techniques properly in order to identify how further improvement can be made especially for spatial data clustering. Finally, we present a weighted kernel k-means clustering algorithm incorporating spatial constraints bearing spatial neighborhood information in order to handle spatial auto-correlation and noise in the spatial data.

This paper is organized as follows. In the next section, it is pointed out how kernel methods can be useful for clustering non-linearly separable and high-dimensional spatial data. The k-means algorithm is briefly described in section 3. In this section, two currently proposed kernel-based algorithms are also reviewed. In section 4, a weighted kernel k-means algorithm with spatial constraints is presented which could be useful for handling noise, outliers and auto-correlation in the spatial data. Finally, the paper concludes with emphasizing the need for an in-depth study for developing a real system.

2.0 Kernel-based Methods

The kernel methods are among the most researched subjects within machine-learning community in recent years and has been widely applied to pattern recognition and function approximation. Typical examples are support vector machines [2, 6, 20], kernel Fisher linear discriminant analysis[14], kernel principal component analysis [17], kernel perceptron algorithm [5], just to name a few. The fundamental idea of the kernel methods is to first transform the original low-dimensional inner-product input space into a higher dimensional feature space through some nonlinear mapping where complex nonlinear problems in the original low-dimensional space can more likely be linearly treated and solved in the transformed space according to the well-known Cover's theorem. However, usually such mapping into high-dimensional feature space will undoubtedly lead to an exponential increase of computational time, i.e., so-called curse of dimensionality. Fortunately, adopting kernel functions to substitute an inner product in the original space, which exactly corresponds to mapping the space into higher-dimensional feature space, is a favorable option. Therefore, the inner product form leads us to applying the kernel methods to cluster complex data [9, 15].

2.1 Support Vector Machines and Kernel-based Methods

Support vector machines (SVM), having its roots in machine learning theory, utilize optimization tools that seek to identify a linear optimal separating hyperplane to

discriminate any two classes of interest [20, 19]. When the classes are linearly separable, the linear SVM performs adequately.

There are instances where a linear hyperplane cannot separate classes without misclassification, relevant to our problem domain; however, those classes can be separated by a nonlinear separating hyperplane. In this case, data may be mapped to a higher dimensional space with a nonlinear transformation function. In the higher dimensional space, data are spread out, and a linear separating hyperplane may be found. This concept is based on Cover's theorem on the separability of patterns. According to Cover's theorem on the separability of patterns, an input space made up of nonlinearly separable patterns may be transformed into a feature space where the patterns are linearly separable with high probability, provided the transformation is nonlinear and the dimensionality of the feature space is high enough. Figure 1 illustrates that two classes in the input space may not be separated by a linear separating hyperplane, a common property of spatial data, e.g. rainfall patterns in a green mountain area might not be linearly separable from those in the surrounding plain area. However, when the two classes are mapped by a nonlinear transformation function, a linear separating hyperplane can be found in the higher dimensional feature space.

Let a nonlinear transformation function ϕ maps the data into a higher dimensional space. Suppose there exists a function *K*, called a kernel function, such that,

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$$

A kernel function is substituted for the dot product of the transformed vectors, and the explicit form of the transformation function ϕ is not necessarily known. In this way, kernels allow large non-linear feature spaces to be explored while avoiding curse of dimensionality. Further, the use of the kernel function is less computationally intensive. The formulation of the kernel function from the dot product is a special case of *Mercer's theorem* [16].

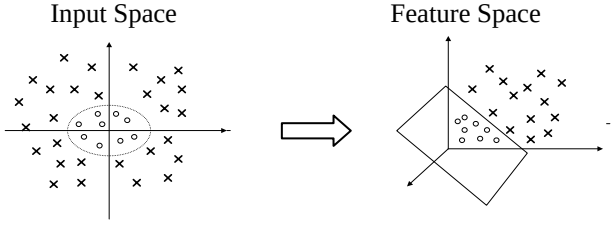

FIGURE 1: Mapping nonlinear data to a higher dimensional feature space where a linear separating hyperplane can be found. When mapped into a feature space via the non-linear map $\Phi(x) = (z_1, z_2, z_3) = \left([x]_1^2, [x]_2^2, \sqrt{2}[x]_1[x]_2\right)$

Examples of some well-known kernel functions are given in below:

Polynomial, $K(x_i, x_j) = \langle x_i, x_j \rangle^d$; *d* is a positive *integer*

Radial Basis Function, $K(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$; σ is a user defined value

Sigmoid, $K(x_i, x_j) = \tanh(\alpha \times \langle x_i, x_j \rangle + \beta)$; α and β are user defined values

3.0 K-Means and Kernel Methods for Clustering

Clustering has received a significant amount of renewed attention with the advent of nonlinear clustering methods based on kernels as it provides a common means of identifying structure in complex data [9, 15, 4, 2]. Before discussing two kernel-based algorithms [4, 2] here, the popular k-means algorithm is described in the next subsection, which is used as predominant strategy for final partitioning of the data.

3.1 K-Means

First we briefly review k-means [12] which is a classical algorithm for clustering. We first fix the notation: let $X = \{ x_i \}_{i=1,\ldots,n}$ be a data set with $x_i \in \mathrm{R}^N$. We call codebook the set $W = \{ w_j \}_{j=1,\ldots,k}$ with $w_j \in \mathrm{R}^N$ and $k << n$. The Voronoi set (V_j) of the codevector w_j is the set of all vectors in *X* for which w_j is the nearest vector, i.e.

$$V_j = \{x_i \in X \mid j = arg \min_{j=1,\ldots,k} \|x_i - w_j\|\}$$

For a fixed training set *X* the quantization error *E*(*W*) associated to the Voronoi tessellation induced by the codebook *W* can be written as

$$E(W) = \sum_{j=1}^{k} \sum_{x_i \in V_j} \|x_i - w_j\|^2 \quad (1)$$

K-means is an iterative method for minimizing the quantization error *E*(*W*) by repeatedly moving all codevectors to the arithmetic mean of their Voronoi sets. It can be proved [10] that a necessary condition for a codebook *W* to minimize the quantization error in eq. (1) is that each codevector w_j fulfills the centroid condition. In the case of finite data set *X* and Euclidean distance, the centroid condition reduces to

$$w_j = \frac{1}{|V_j|} \sum_{x_i \in V_j} x_i \quad (2)$$

where $|V_j|$ denotes the cardinality of V_j . Therefore, k-means is guaranteed to find a local minimum for the quantization error.

However, the k-means does not have mechanism to deal with issues such as:

- Outliers; one of the drawbacks of k-means is lack of robustness with respect to outliers, this problem can be easily observed by looking at the effect of outliers in the computation of the mean in eq. (2).
- non-linear separability of data in input space,
- auto-correlation in spatial data,
- noise, and high dimensionality of data.

3.2 One Class SVM

Support vector clustering (*SVC*) [2], also called one-class SVM, is an unsupervised kernel method based on support vector description of a data set consisting of positive examples only. In SVC, data points are mapped from data space to a high dimensional feature space using a Gaussian kernel. In feature space, SVC computes the smallest sphere that encloses the image of the input data. This sphere is mapped back to data space, where it forms a set of contours, which enclose the data points. These contours are interpreted as cluster boundaries. Points enclosed by each separate contour are associated with the same cluster. (For details about the algorithm, please see [2]).

The clustering level can be controlled by changes in the width parameter of the Gaussian kernel (σ). As this parameter is increased, the number of disconnected contours in data space increases too, leading to an increasing number of clusters. The SVC algorithm can also deal with outliers by employing a soft margin constant that allows the sphere in feature space not to enclose all points. Large values of this parameter, can also deal with overlapping clusters.

Since SVC is using a transformation to an infinite dimension space, it can handle clusters of practically any shape, form or location in space. This is probably its most important advantage. It can easily identify cluster combination, which will cause most other clustering algorithms to fail. A simple example could be concentric ring-like clusters, which will pose a huge problem to the common algorithms (such as k-nearest neighbors, k-means). However, the algorithm has the following drawbacks:

- One problem though, which makes the algorithm hard to tune, is its extreme dependence on the width of the Gaussian σ. Finding the right value of σ is time-consuming and very delicate.
- Another disadvantage of the algorithm is its complexity. Although the calculation of the sphere parameters and the support vectors is relatively rapid, the separation of the sphere to different clusters and determining the adjacency matrix is extremely complicated.
- As the number of dimensions increases, the running time of the algorithm grows dramatically. For a large number of attributes, it is practically not feasible to use this algorithm.

3.3 Mercer Kernel k-Means

In [4], F. Camastra and A. Verri report on extending the SVC algorithm and give a kernel k-means algorithm. The kernel k-means algorithm uses k-means like strategy in the feature space using a one class support vector machine. The algorithm can find more than one clusters. For details about the algorithm, please see [4].
Although the algorithm [4] gives nice results and can handle outliers but it has some drawbacks:

- The convergence of this procedure is not guaranteed and is an open problem. The algorithm does not aim at minimizing the quantization error because the Voronoi sets are not based on the computation of the centroids.
- Another main drawback is the heavy computation time required by the algorithm. The algorithm requires the solution of a quite number of quadratic programming problems.
- Because of the computational overheads, the algorithm might become unstable for high-dimensional data.
- Moreover, there is no mechanism for handling spatial auto-correlation in the data.

4.0 Proposed Weighted Kernel k-Means Incorporating Spatial Constraints

As we have illustrated above, there exist many problems in the current k-means method, especially for handling spatial and complex data. Among these, the important issues/problems that need to be addressed are: i) non-linear separability of data in input space, ii) outliers and noise, iii) auto-correlation in spatial data, iv) high dimensionality of data. Although kernel methods offer power to deal with non-linearly separable and high-dimensional data but the current methods have some drawbacks as identified in section 3. Both [2, 4] are computationally very intensive, unable to handle large datasets and autocorrelation in the spatial data. The method proposed in [2] is not feasible to handle high dimensional data due to computational overheads, whereas the convergence of [4] is an open problem. With regard to addressing these problems, we propose an algorithm—weighted kernel k-means with spatial constraints, in order to handle spatial autocorrelation, noise and outliers present in the spatial data.

Using the non-linear function ϕ, the objective function of weighted kernel *k*-means can be defined as:

$$E(W) = \sum_{j=1}^{k} \sum_{x_i \in V_j} u(x_i) \left\| \phi(x_i) - w_j \right\|^2 \tag{3}$$

where,
$$w_j = \frac{\sum_{x_j \in V_j} u(x_j)\phi(x_j)}{\sum_{x_j \in V_j} u(x_j)} \tag{4}$$

The Euclidean distance from $\phi(x)$ to center w_j is given by (all computations in the form inner products can be replaced by entries of the kernel matrix)

$$\left\| \phi(x_i) - \frac{\sum_{x_j \in V_j} u(x_j)\phi(x_j)}{\sum_{x_j \in V_j} u(x_j)} \right\|^2 = K(x_i, x_i) - 2\frac{\sum_{x_j \in V_j} u(x_j) K(x_i, x_j)}{\sum_{x_j \in V_j} u(x_j)} + \frac{\sum_{x_j \in V_j} \left(u(x_j)\right)^2 K(x_j, x_j)}{\left(\sum_{x_j \in V_j} u(x_j)\right)^2} \tag{5}$$

In [1], an approach is proposed to increase the robustness of fuzzy c-means to noise. Similarly we propose a modification to the weighted kernel k-means to increase the robustness to noise and to account for spatial autocorrelation in the spatial

data. It can be achieved by a modification to eq. (3) by introducing a penalty term containing spatial neighborhood information. This penalty term acts as a regularizer and biases the solution toward piecewise-homogeneous labeling. Such regularization is helpful in finding clusters in the data corrupted by noise. The objective function (3) can, thus, be written as:

$$E(W) = \sum_{j=1}^{k}\sum_{x_i \in V_j} u(x_i)\left\|\phi(x_i) - w_j\right\|^2 + \frac{\gamma}{N_R}\sum_{j=1}^{k}\sum_{x_i \in V_j} u(x_i)\sum_{r \in N_k}\left\|\phi(x_r) - w_j\right\|^2 \quad (6)$$

where N_k stands for the set of neighbors that exist in a window around x_i and N_R is the cardinality of N_k . The parameter γ controls the effect of the penalty term. The relative importance of the regularizing term is inversely proportional to the accuracy of clustering results.

If we adopt the Gaussian radial basis function (RBF), then $K(x, x) = 1$, so we can simplify eq. (6) as

$$E(W) = 2\sum_{j=1}^{k}\sum_{x_i \in V_j} u(x_i)(1 - K(x_i, w_j)) + \frac{\gamma}{N_R}\sum_{j=1}^{k}\sum_{x_i \in V_j} u(x_i)\sum_{r \in N_k}(1 - K(x_r, w_j)) \quad (7)$$

The distance in the last term of eq.(6), can be calculated as

$$\left\|\phi(x_r) - \frac{\sum_{x_j \in V_j} u(x_j)\phi(x_j)}{\sum_{x_j \in V_j} u(x_j)}\right\|^2 = 1 - 2\frac{\sum_{x_j \in V_j} u(x_j)K(x_r, x_j)}{\sum_{x_j \in V_j} u(x_j)} + \frac{\sum_{x_j \in V_j}\left(u(x_j)\right)^2}{\left(\sum_{x_j \in V_j} u(x_j)\right)^2} = \beta_r \quad (8)$$

For RBF, eq. (5) becomes,

$$\left\|\phi(x_i) - \frac{\sum_{x_j \in V_j} u(x_j)\phi(x_j)}{\sum_{x_j \in V_j} u(x_j)}\right\|^2 = 1 - 2\frac{\sum_{x_j \in V_j} u(x_j)K(x_i, x_j)}{\sum_{x_j \in V_j} u(x_j)} + \frac{\sum_{x_j \in V_j}\left(u(x_j)\right)^2}{\left(\sum_{x_j \in V_j} u(x_j)\right)^2} \quad (9)$$

We have to calculate the distance from each point to every cluster representative. For cluster V_j, incorporating the penalty term containing spatial neighborhood information, this can be obtained from eq. (6) by using eq. (8) and (9). Hence, the effective distance to be calculated is:

$$1 - 2\frac{\sum_{x_j \in V_j} u(x_j)K(x_i, x_j)}{\sum_{x_j \in V_j} u(x_j)} + \frac{\sum_{x_j \in V_j}\left(u(x_j)\right)^2}{\left(\sum_{x_j \in V_j} u(x_j)\right)^2} + \frac{\gamma}{N_R}\sum_{r \in N_k}\beta_r \quad (10)$$

We can examine how eq. (10) makes the algorithm robust to outliers. As $K(x_i, x_j)$ measures the similarity between x_i *and* x_j, and when x_i is an outlier, i.e., x_i is far from the other data points, then $K(x_i, x_j)$ will be very small. So, the second term in the above expression will get very low value or, in other words, the weighted sum of data points will be suppressed. The total expression will get higher value and hence results in robustness by not assigning the point to the cluster. For details of the algorithm, please see [13].

Now, the algorithm, weighted kernel k-means with spatial constraints (SWK-means: Spatial Weighted Kernel k-means), can be written as in Figure 2.

Algorithm SWK-means: Spatial Weighted Kernel k-means (weighted kernel k-means with spatial constraints)

SWK_means (K, k, u, N, γ , ε, w_1, ...,w_k)

Input: K: kernel matrix, k: number of clusters, u: weights for each point, set $\varepsilon>0$ to a very small value for termination, N: information about the set of neighbors around a point, γ : penalty term parameter,

Output: w_1, ...,w_k: partitioning of the points

1. Initialize the k clusters: $w_1=0$, ... , $w_k=0$
2. Set $i = 0$.
3. For each point x, find its new cluster index as

   $j(x) = arg\ min_j \left\|\phi(x) - w_j\right\|^2$ using eq. (10),

4. Compute the updated clusters as

   $w_j^{(i+1)} = \{x : j(x)=j\}$

5. Repeat steps 3-4 until the following termination criterion is met:

   $$\left\|W_{new} - W_{old}\right\| < \varepsilon$$

where, $W = \{w_1, w_1, w_1, ..., w_k\}$ are the vectors of cluster centroids.

FIGURE 2: Algorithm SWK-Means (weighted kernel k-means with spatial constraints)

5.0 Discussion and Conclusions

In this paper, a few challenges especially related to clustering spatial data are pointed out. There exist some problems that k-means method cannot tackle, especially for dealing with spatial and complex data. Among these, the important issues/problems that need to be addressed are: i) non-linear separability of data in input space, ii) outliers and noise, iii) auto-correlation in spatial data, iv) high dimensionality of data.

The strengths of kernel methods are outlined, which are helpful for clustering complex and high dimensional data that is non-linearly separable in input space. Two of the currently proposed kernel based algorithms are reviewed and the related research issues are identified. Both [2, 4] are computationally very intensive, unable to handle large datasets and have no mechanism to deal with autocorrelation in the spatial data. The method proposed in [2] is not feasible to handle high dimensional data due to computational overheads, whereas the convergence of [4] is an open problem. With regard to addressing these problems, we presented weighted kernel k-means incorporating spatial

constraints. Theoretically the proposed algorithm has the mechanism to handle spatial autocorrelation, noise and outliers in the spatial data. The implementation, testing and evaluation of the algorithm are underway. It is very much hoped that the algorithm would prove to be robust and effective for spatial data analysis. However, it needs further exploration. In future we plan to investigate the estimation of optimal number of clusters automatically.

References

[1] M.N. Ahmed, S.M. Yamany, N. Mohamed, A.A. Farag and T. Moriarty. A modified fuzzy C-means algorithm for bias field estimation and segmentation of MRI data. IEEE Trans. on Medical Imaging, vol. 21, pp.193-199, 2002.

[2] A. Ben-Hur, D. Horn, H. Siegelman, and V. Vapnik. Support vector clustering. Journal of Machine Learning Research 2, 2001.

[3] F. Camastra. Kernel Methods for Unsupervised Learning. PhD thesis, University of Genova, 2004.

[4] F. Camastra, A. Verri. A Novel Kernel Method for Clustering. To appear in IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI), IEEE Computer Society, 2005.

[5] J. H. Chen and C. S. Chen. Fuzzy kernel perceptron. IEEE Trans. Neural Networks, vol. 13, pp. 1364–1373, Nov. 2002.

[6] N. Cristianini and J.S.Taylor. An Introduction to Support Vector Machines. Cambridge Academic Press, 2000.

[7] I.S. Dhillon, Y. Guan, B. Kulis. Kernel kmeans, Spectral Clustering and Normalized Cuts. KDD 2004.

[8] C. Ding and X. He. K-means Clustering via Principal Component Analysis. Proc. of Int'l Conf. Machine Learning (ICML 2004), pp 225-232. July 2004.

[9] M. Girolami. Mercer Kernel Based Clustering in Feature Space. IEEE Trans. on Neural Networks. 2002.

[10] R.M. Gray. Vector Quantization and Signal Compression. Kluwer Academic Press, Dordrecht, 1992.

[11] J. Han, M. Kamber and K. H. Tung. Spatial Clustering Methods in Data Mining: A Survey. Harvey J. Miller and Jiawei Han (eds.), Geographic Data Mining and Knowledge Discovery, Taylor and Francis, 2001.

[12] S.P. Lloyd. An algorithm for vector quantizer design. IEEE Transaction on Communications, vol. 28, no. 1, pp. 84-95, 1982.

[13] A. Majid Awan, Mohd. Noor Md. Sap. Weighted Kernel K-Means Algorithm for Clustering Spatial Data. Journal of Information Technology, University Technology Malaysia, Dec 2004.

[14] V. Roth and V. Steinhage. Nonlinear discriminant analysis using kernel functions. In Advances in Neural Information Processing Systems 12, S. A Solla, T. K. Leen, and K.-R. Muller, Eds. Cambridge, MA: MIT Press, 2000, pp. 568–574.

[15] D.S. Satish and C.C. Sekhar. Kernel based clustering for multiclass data. Int. Conf. on Neural Information Processing , Kolkata, Nov. 2004.

[16] B. Scholkopf and A. Smola. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, 2002.

[17] B. Scholkopf, A. Smola, and K. R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput., vol. 10, no. 5, pp. 1299–1319, 1998.

[18] S. Shekhar, P. Zhang, Y. Huang, R. Vatsavai. Trends in Spatial Data Mining. As a chapter in Data Mining: Next Generation Challenges and Future Directions, H. Kargupta, A. Joshi, K. Sivakumar, and Y. Yesha(eds.), MIT Press, 2003

[19] V.N. Vapnik. The Nature of Statistical Learning Theory. Springer-Verlag, New York, 1995.

[20] V.N.Vapnik. Statistical Learning Theory. John Wiley & Sons, 1998

[21] L. Xu, J. Neufeld, B. Larson, D. Schuurmans. Maximum Margin Clustering. NIPS 2004.

[22] P. Zhang, M. Steinbach, V. Kumar, S. Shekhar, P-N Tan, S. Klooster, and C. Potter, Discovery of Patterns of Earth Science Data Using Data Mining, as a Chapter in Next Generation of Data Mining Applications, J. Zurada and M. Kantardzic(eds), IEEE Press, 2003.