Proceedings of the 5th WSEAS International Conference on Applied Computer Science, Hangzhou, China, April 16-18, 2006 (pp560-565)

Kernel-based Algorithm for Clustering Spatial Data

[1]A. MAJID AWAN, [2]MOHD. NOOR MD. SAP, [3]M. OMAR MANSUR
Faculty of Computer Science & Information Systems
University Technology Malaysia
Skudai 81310, Johor
MALAYSIA

[1]

Abstract: - This paper presents a method for unsupervised partitioning of data for finding spatio-temporal patterns in climate data using kernel methods which offer strength to deal with complex data non-linearly separable in input space. 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. In this paper we present a robust weighted kernel k-means algorithm incorporating spatial constraints for clustering climate data. The proposed algorithm can effectively handle noise, outliers and auto-correlation in the spatial data, for effective and efficient data analysis.

Keywords: - Clustering, K-means, Kernel methods, spatial data, unsupervised learning

1 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. 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 [3, 7, 8, 9, 15, 20]. Generally speaking, a 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 [4, 9, 16], 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 [11, 17, 21]. With this in view, the intention of this paper is, firstly, to analyze selective kernel-based clustering techniques 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.

2 Kernel-based methods

The kernel methods are among the most researched subjects within machine-learning community in recent years and have been widely applied to pattern recognition and function approximation [2, 5, 6, 14, 16, 19]. 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.

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 [18, 19]. When the classes are linearly separable, the linear SVM performs adequately.

There are instances where a linear hyperplane cannot separate classes without misclassification, an instance 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. 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 [15].

Examples of some well-known kernel functions are given below in table 1.

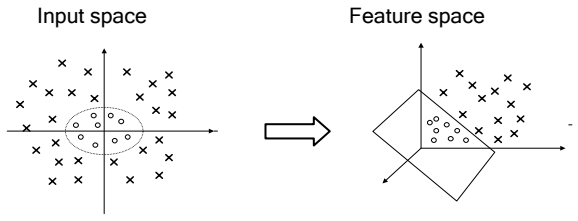

Fig. 1. Mapping nonlinear data to a higher dimensional feature space where a linear separating hyperplane can be found, eg, via the nonlinear map $\Phi(x) = (z_1, z_2, z_3) = \left([x]_1^2, [x]_2^2, \sqrt{2}[x]_1[x]_2\right)$

TABLE 1. Some well-known kernel functions

Polynomial	$K(x_i, x_j) = < x_i, x_j >^d$	*d* is a positive *integer*
Radial Basis Function (RBF)	$K(x_i, x_j) = \exp(-\|x_i - x_j\|^2 / 2\sigma^2)$	σ is a user defined value
Sigmoid	$K(x_i, x_j) = \tanh(\alpha < x_i, x_j > + \beta)$	α, β are user defined values

3 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 [2, 4, 9]. Before discussing two kernel-based algorithms [2, 4] 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.

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

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. 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.

The clustering level can be controlled by changes in the width parameter of the Gaussian kernel (σ). 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.

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. However, the algorithm has the following drawbacks:

- One problem with the algorithm is its extreme dependence on σ. Finding the right value of σ is time-consuming and very delicate.
- Another disadvantage of the algorithm is its complexity. 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. The kernel k-means algorithm [4] uses k-means like strategy in the feature space using one class support vector machine. The algorithm can find more than one clusters. 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.
- The algorithm requires the solution of a quite number of quadratic programming problems, so takes heavy computation time.
- 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 Weighted kernel k-means with spatial constraints

As we have illustrated above, there exist some problems in the 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.

The k-means clustering algorithm can be enhanced by the use of a kernel function; by using an appropriate nonlinear mapping from the original (input) space to a higher dimensional feature space, one can extract clusters that are non-linearly separable in input space. Usually the extension from k-means to kernel k-means is realised by expressing the distance in the form of kernel function [15]. The kernel k-means algorithm can be generalized by introducing a weight for each point x, denoted by $u(x)$ [7]. This generalization would be powerful for making the algorithm more robust to noise and useful for handling auto-correlation in the spatial data. Using the non-linear function ϕ, the objective function of weighted kernel k-means can be defined as:

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

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

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

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

In the above expression, the last term is needed to be calculated once per each iteration of the algorithm, and is representative of cluster centroids. If we write

$$C_k = \frac{\sum_{x_j, x_l \in V_j} u(x_j)u(x_l)K(x_j, x_l)}{(\sum_{x_j \in V_j} u(x_j))^2} \quad (6)$$

With this substitution, eq (5) can be re-written as

$$\left\| \phi(x_i) - \frac{\sum_{x_j \in V_j} u(x_j)\phi(x_j)}{\sum_{x_j \in V_j} u(x_j)} \right\|^2 = K(x_i, x_i) - 2\frac{\sum_{x_j \in V_j} u(x_j)K(x_i, x_j)}{\sum_{x_j \in V_j} u(x_j)} + C_k \quad (7)$$

For increasing the robustness of fuzzy c-means to noise, an approach is proposed in [1]. Here 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 also helpful in finding clusters in the data corrupted by noise. The objective function (3) can, thus, be written as:

$$E(W) = \sum_{j=1}^{k}\sum_{x_i \in V_j} u(x_i)\left\|\phi(x_i) - w_j\right\|^2 + \frac{\gamma}{N_R}\sum_{j=1}^{k}\sum_{x_i \in V_j} u(x_i)\sum_{r \in N_k}\left\|\phi(x_r) - w_j\right\|^2 \quad (8)$$

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.

For kernel functions, the following can be written

$$\left\|\phi(x_i) - w_j\right\|^2 = K(x_i, x_i) - 2K(x_i, w_j) + K(w_j, w_j)$$

If we adopt the Gaussian radial basis function (RBF), then $K(x, x) = 1$, so eq. (8) can be simplified as

$$E(W) = 2\sum_{j=1}^{k}\sum_{x_i \in V_j} u(x_i)(1 - K(x_i, w_j)) + \frac{\gamma}{N_R}\sum_{j=1}^{k}\sum_{x_i \in V_j} u(x_i)\sum_{r \in N_k}(1 - K(x_r, w_j)) \quad (9)$$

The distance in the last term of eq. (8), can be calculated as

$$\left\| \phi(x_r) - \frac{\sum_{x_j \in V_j} u(x_j)\phi(x_j)}{\sum_{x_j \in V_j} u(x_j)} \right\|^2 = 1 - 2\frac{\sum_{x_j \in V_j} u(x_j)K(x_r, x_j)}{\sum_{x_j \in V_j} u(x_j)} + \frac{\sum_{x_j, x_l \in V_j} u(x_j)u(x_l)K(x_j, x_l)}{(\sum_{x_j \in V_j} u(x_j))^2} \quad (10)$$

As first term of the above equation does not play any role for finding minimum distance, so it can be omitted, however.

$$\left\| \phi(x_r) - \frac{\sum_{x_j \in V_j} u(x_j)\phi(x_j)}{\sum_{x_j \in V_j} u(x_j)} \right\|^2 = 1 - 2\frac{\sum_{x_j \in V_j} u(x_j)K(x_r, x_j)}{\sum_{x_j \in V_j} u(x_j)} + C_k = 1 - \beta_r + C_k \quad (11)$$

For RBF, eq. (5) can be re-written as

$$\left\| \phi(x_i) - \frac{\sum_{x_j \in V_j} u(x_j)\phi(x_j)}{\sum_{x_j \in V_j} u(x_j)} \right\|^2 = 1 - 2\frac{\sum_{x_j \in V_j} u(x_j)K(x_i, x_j)}{\sum_{x_j \in V_j} u(x_j)} + C_k \quad (12)$$

As first term of the above equation does not play any role for finding minimum distance, so it can be omitted.

We have to calculate the distance from each point to every cluster representative. This can be obtained from eq. (8) after incorporating the penalty term containing spatial neighborhood information by using eq. (11) and (12). Hence, the effective minimum distance can be calculated using the expression:

$$-2\frac{\sum_{x_j \in V_j} u(x_j)K(x_i, x_j)}{\sum_{x_j \in V_j} u(x_j)} + C_k + \frac{\gamma}{N_R}\sum_{r \in N_k}(\beta_r + C_k) \quad (13)$$

Now, the algorithm, weighted kernel k-means with spatial constraints, can be written as follows.

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

SWK_means (K, k, u, N, γ, ε)

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 cluster, compute $C(k)$ using expression (6)
4. For each point x, find its new cluster index as

   $j(x) = arg\,min_j \left\|\phi(x) - w_j\right\|^2$ using expression (13),
5. Compute the updated clusters as

   $w_j^{(i+1)} = \{x : j(x)=j\}$
6. 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.

4.1 Handling outliers

This section briefly discusses about spatial outliers, i.e., observations which appear to be inconsistent with their neighborhoods. Detecting spatial outliers is useful in many applications of geographic information systems and spatial databases, including transportation, ecology, public safety, public health, climatology, location-based services, and severe weather prediction. Informally, a spatial outlier is a local instability (in values of non-spatial attributes) or a spatially referenced object whose non-spatial attributes are extreme relative to its neighbors, even though the attributes may not be significantly different from the entire population.

We can examine how eq. (13) 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.

5 Experimental Results

Given a data matrix, whose rows consists of time series from various points on the land (rainfall stations), the objective is to discover temporal and/or spatial patterns in the data. If we apply clustering algorithm to the rainfall time series associated with points on the land (surroundings of rainfall stations), we obtain clusters that represent land regions with relatively homogeneous behaviour. The centroids of these clusters are time series that summarize the bahaviour of those land areas.

For experimentation we selected 24 rainfall stations. A 12-month moving average is used for removing seasonality from the data. For monthly rainfall values for 5 years, we get a data matrix of 24×60. SWK-means partitioned it into 2 clusters. We also applied the algorithm to the monthly average rainfall values of this period, for easy visualization of results. Its results are shown in Figure 2. As the locations of rainfall stations are known, the clustering results can be easily mapped on the physical locations on the map. Actually the clusters will summarize the time series associated with relevant regions, and when results are plotted for a longer period, it will form some contiguous regions.

Since the kernel matrix is symmetric, we only keep its upper triangular matrix in the memory. For the next five year periods of time for the selected 24 rainfall stations we may get data matrices of 48×60, 72×60 and so on. The algorithm proportionally partitioned the data into two clusters. The corresponding results are shown in table 2 (a record represents 5-year monthly rainfall values taken at a station). It validates proper working of the algorithm.

We use the clustering algorithm as a part of a software system for analyzing the impact of various hydrological and meteorological variables on the oil palm plantation [22]. It enables us to identify regions of the land whose constituent points have similar short-term and long-term characteristics. Given relatively uniform clusters we can then identify how various parameters, such as precipitation, temperature etc, influence the climate and oil-palm produce of different areas using correlation.

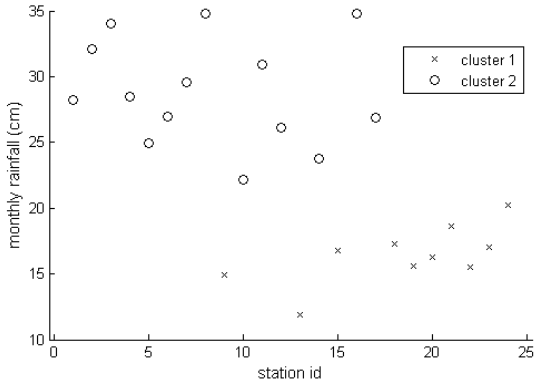

Fig. 2. Clustering results of SWK-means algorithm showing two clusters of monthly rainfall of 24 stations

TABLE 2. Results of SWK-means on rainfall data at 24 stations for 5, 10, 15, 20, 25, 30, 35 years

No. of Records	No. of records in cluster 1	No. of records in cluster 2
24	10	14
48	20	28
72	30	42
96	40	56
120	50	70
144	60	84
168	70	98

6 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. The proposed algorithm has the mechanism to handle spatial autocorrelation, noise and outliers in the spatial data. We are getting promising results on our test data sets. It is very much hoped that the algorithm would prove to be robust and effective for spatial (climate) data analysis. 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. Medical Imaging*, vol. 21, pp.193–199, 2002.

[2] A. Ben-Hur, D. Horn, H. Siegelman, and V. Vapnik, "Support vector clustering," *J. Machine Learning Research*, vol. 2(Dec.), pp. 125–137, 2001.

[3] F. Camastra, "Kernel methods for unsupervised learning," PhD thesis, Univ. of Genova, Italy, 2004.

[4] F. Camastra, A. Verri, "A novel kernel method for clustering," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, pp. 801-805, May 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, and B. Kulis, "Kernel kmeans, spectral clustering and normalized cuts," in *Proc. 10th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pp. 551–556, 2004.

[8] C. Ding and X. He, "K-means clustering via principal component analysis,"in *Proc. 21st Int. Conf. Machine Learning*, July 2004, pp. 225–232.

[9] M. Girolami, "Mercer kernel based clustering in feature space," *IEEE Trans. Neural Networks*, vol. 13(3), pp. 780–784, May 2002.

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

[11] J. Han, M. Kamber and K. H. Tung, "Spatial clustering methods in data mining: a survey," in *Geographic Data Mining and Knowledge Discovery*, Harvey J. Miller and Jiawei Han (eds.). Taylor and Francis, 2001.

[12] S.P. Lloyd, "An algorithm for vector quantizer design," *IEEE Trans. Communications*, vol. 28, no. 1, pp. 84–95, 1982.

[13] M.N. Md. Sap, and A. Majid Awan, "Finding spatio-temporal patterns in climate data using clustering," in *Proc. IEEE 2005 Int. Conf. on Cyberworlds*, Singapore, 23-25 November 2005, pp. 155–164.

[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.). MIT Press, 2000, pp. 568–574.

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

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

[17] S. Shekhar, P. Zhang, Y. Huang, and 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.

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

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

[20] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans, "Maximum margin clustering," in *Proc. Advances in Neural Information Processing Systems* (NIPS2004), 2004.

[21] 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.

[22] A.M. Awan and M.N. Md. Sap, "An intelligent system based on kernel methods for crop yield prediction," in *Proc. 10th Pacific-Asia Conference on Knowledge Discovery and Data Mining* (PAKDD 2006), 9-12 April 2006, Singapore, pp. 841–846.