A Framework for Predicting Oil-Palm Yield from Climate Data

A. Majid Awan[1], Mohd. Noor Md. Sap[2]

Faculty of Computer Science & Information Systems, University Technology Malaysia
Skudai 81310, Johor, Malaysia

[1]awanmajid@hotmail.com, [2]mohdnoor@fsksm.utm.my

***Abstract*—**Intelligent systems based on machine learning techniques, such as classification, clustering, are gaining wide spread popularity in real world applications. This paper presents work on developing a software system for predicting crop yield, for example oil-palm yield, from climate and plantation data. At the core of our system is 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. 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. In this paper we present a robust weighted kernel k-means algorithm incorporating spatial constraints for clustering the data. The proposed algorithm can effectively handle noise, outliers and auto-correlation in the spatial data, for effective and efficient data analysis by exploring patterns and structures in the data, and thus can be used for predicting oil-palm yield by analyzing various factors affecting the yield.

***Keywords*—**Pattern analysis, clustering, kernel methods, spatial data, crop yield

1. Introduction

The contribution of agriculture in the economic growth of Malaysia can be substantially improved through better management practices. Oil-palm has become an important crop in Malaysia. However, oil-palm production potential is reduced when trees are exposed to stressful weather conditions. Low moisture is the most common stressful condition oil-palm faces, so monitoring rainfall and other related parameters (e.g. temperature, pressure, soil moisture, sun-shine duration, humidity, etc) is useful in predicting oil palm yield levels. The lagged effect of weather in Malaysia has implications for global vegetable oil prices in general and for the palm oil market in particular. Moreover, not enough is known about the daily patterns of rainfall or sunshine illumination levels to determine what may mitigate the expected negative effects of the heavy or below-normal rainfall [1]. Keeping this importance in view, this study is aimed at investigating the impacts of hydrological and meteorological conditions on oil-palm plantation using computational machine learning techniques. Resulting improved understanding of the factors affecting oil-palm yield would not only help in accurately predicting yield levels but would also help substantially in looking for mitigating solutions.

Clustering and classification are very useful machine learning techniques which can capture meaningful patterns in the agro-hydrological data. However, complexity of geographic data precludes the use of general purpose pattern discovery and data analysis techniques. 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, 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 [2-10]. However, much research effort is being put up for improving these techniques and in applying these techniques to various application domains. 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 [2,5,11], 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 (i.e., measured values that are close in time and space tend to be highly correlated, or similar), non-linear separability, outliers, noise, high-dimensionality, and when the data has clusters of widely differing shapes and sizes [12-14]. The popular clustering algorithms, like k-means, have some limitations for this type of data [14,15]. Therefore, we present

a weighted kernel k-means clustering algorithm incorporating spatial constraints bearing spatial neighborhood information in order to handle spatial auto-correlation, outliers and noise in the spatial data.

2. Application Area and Methods

This work is focusing on clustering spatial data, e.g. for finding patterns in rainfall, temperature, pressure data so that their impact on other objects like vegetation (specifically oil-palm yield) etc could be explored. A very simplified view of the problem domain looks as shown in Figure 1. The data consists of a sequence of snapshots of the earth areas taken at various points in time. Each snapshot consists of measurement values for a number of variables e.g., temperature, pressure, precipitation, crop yield, etc. All attribute data within a snapshot is represented using spatial frameworks, i.e., a partitioning of the study region into a set of mutually disjoint divisions which collectively cover the entire study region. This way we would be dealing with spatial time series data.

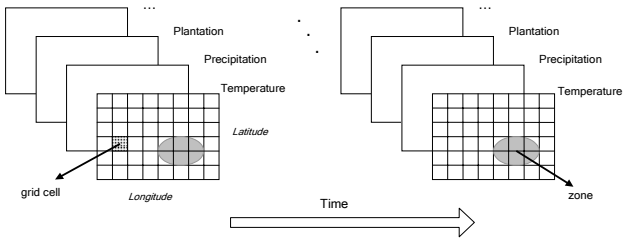

Fig. 1. A simplified view of the problem domain

Clustering, often better known as spatial zone formation in this context, segments land into smaller pieces that are relatively homogeneous in some sense. A goal of the work is to use clustering to divide areas of the land into disjoint regions in an automatic but meaningful way that 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 phenomena or parameters, such as precipitation, influence the climate and oil-palm produce (for example) of different areas.

The spatial and temporal nature of our target data poses a number of challenges. For instance, such type of data is noisy. And displays autocorrelation In addition, such data have high dimensionality (for example, if we consider monthly precipitation values at 1000 spatial points for 12 years, then each time series would be 144 dimensional vector), clusters of non-convex shapes, outliers.

Classical statistical data analysis algorithms often make assumptions (e.g., independent, identical distributions) that violate the first law of Geography, which says that everything is related to everything else but nearby things are more related than distant things. Ignoring spatial autocorrelation may lead to residual errors that vary systematically over space exhibiting high spatial autocorrelation [13]. The models derived may not only turn out to be biased and inconsistent, but may also be a poor fit to the dataset [14].

One way to model spatial dependencies is by adding a spatial autocorrelation term in the regression equation. This term contains a neighborhood relationship contiguity matrix. Such spatial statistical methods, however, are computationally expensive due to their reliance on contiguity matrices that can be larger than the spatial datasets being analyzed [13].

If we apply a clustering algorithm to cluster time series associated with points on the land, we obtain clusters that represent land regions with relatively homogeneous behavior. The centroids of these clusters are time series that summarize the behavior of these land areas, and can be represented as indices. 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. This way clustering can better help in detailed analysis of our problem.

A simplified architecture of the agro-hydrological system looks as shown below:

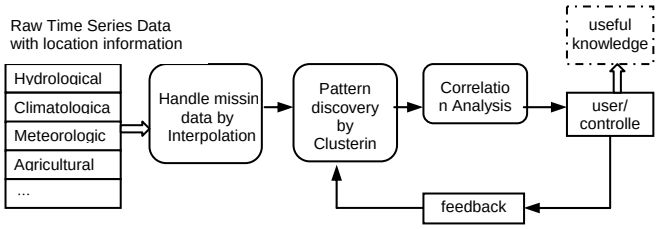

Fig. 2. A simplified architecture for the system

It is worth mentioning that we use Rough sets approach for handling missing data in the preprocessing stage of the system. However, in order to contain the contents, the main component of the system, namely clustering algorithm, is elaborated in this paper.

3. 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. Typical examples are support vector machines [16-18], kernel Fisher linear discriminant analysis [25], kernel principal component analysis [11], kernel perceptron algorithm [20], 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 [5,7,18].

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 [18,21]. 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. 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 3 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 [8].

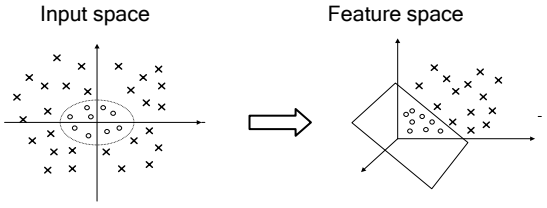

Fig. 3. Mapping nonlinear data to a higher dimensional feature space where a linear separating hyperplane can be found, eg, via the nonlinear map $\Phi(x) = (z_1, z_2, z_3) = \left([x]_1^2, [x]_2^2, \sqrt{2}[x]_1[x]_2\right)$

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

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

4. Proposed Weighted Kernel K-Means with Spatial Constraints

It has been reviewed in [19] that there are some limitations with 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. Both [16,2] are computationally very intensive, unable to handle large datasets and autocorrelation in the spatial data. The method proposed in [16] is not feasible to handle high dimensional data due to computational overheads, whereas the convergence of [2] 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.

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 [2,3,5,6,16,19,22-25]. 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 with the Voronoi tessellation induced by the codebook W can be written as

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

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

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

where $|V_j|$ denotes the cardinality of V_j . Therefore, k-means is guaranteed to find a local minimum for the quantization error [26,27].

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 [8]. The kernel k-means algorithm can be generalized by introducing a weight for each point *x*, denoted by *u*(*x*). 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:

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

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

***Why weighting*?** As the weather station readings or agriculture related attribute values do not often represent equal areas, in such cases the use of weighted means instead of means becomes necessary. A 'weight' attributed to a mean or other value usually signifies importance. If weighting is not used, then each value that enters into a calculation of mean would have the same importance. Sometimes this is not a realistic or fair way to calculate a mean. Individual observations often, for various reasons, are of varying importance to the total, or average value.

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.

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

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 [28]. 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, with the penalty term, 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.

The following can be written for kernel functions, using the kernel trick for distances [29]

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

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. (7) can be 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.

4.2 Scalability

The pruning procedure used in [30,31] can be adapted to speed up the distance computations in the weighted kernel k-means algorithm. The acceleration scheme is based on the idea that we can use the triangle inequality to avoid unnecessary computations. According to the triangle inequality, for a point x_i, we can write, $d(x_i, w_j^n) \geq d(x_i, w_j^o) - d(w_j^o, w_j^n)$. The distances between the corresponding new and old centers, $d(w_j^o, w_j^n)$ for all j, can be computed. And this information can be stored in a $k \times k$ matrix. Similarly, another $k \times n$ matrix can be kept that contains lower bounds for the distances from each point to each center. The distance from a point to its cluster centre is exact in the matrix for lower bounds. Suppose, after a single iteration, all distances between each point and each center, $d(x_i, w_j^o)$, are computed. In the next iteration, after the centers are updated, we can estimate the lower bounds from each point x_i to the new cluster center, w_j^n, using $d(w_j^o, w_j^n)$ calculations and the distances from the previous iteration, i.e., we calculate the lower bounds as $d(x_i, w_j^o) - d(w_j^o, w_j^n)$. The distance from x_i to w_j^n is computed only if the estimation is smaller than distance from x_i to its cluster center. This estimation results in sufficient saving in computational time. Once we have computed lower bounds and begin to compute exact distances, the lower bound allows us to determine whether or not to determine remaining distances exactly.

5. Experimental Results

The system is implemented in C++. We get very hopeful results regarding analyzing various factors impacting oil-palm yield. Because of space constraints, here we mention the results in a brief manner, especially of the clustering algorithm as the SWK algorithm is at the core of the overall system.

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 4. 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.

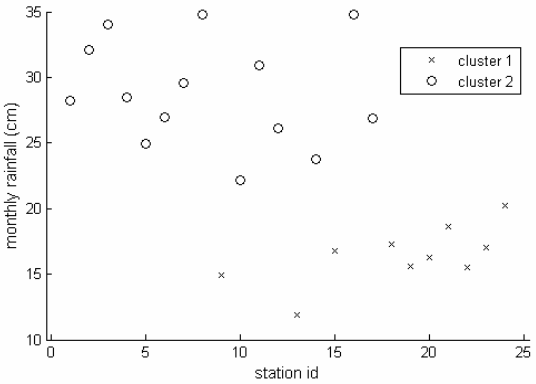

Fig. 4. Clustering results of SWK-means algorithm showing two clusters of monthly rainfall (average) of 24 stations

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 also validates proper working of the algorithm.

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

For the overall system, the information about the landcover areas of oil palm plantation is gathered. The yield values for these areas over a span of time constitute time series. The analysis of these and other time series (e.g., precipitation, temperature, pressure, etc) is conducted using clustering technique. Clustering is helpful in analyzing the impact of various hydrological and meteorological variables on the oil palm plantation. 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. Our initial study shows that the rainfall patterns alone affect oil-palm yield after 6-7 months. This way we are able to predict oil-palm yield for the next 1-3 quarters on the basis of analysis of present plantation and environmental data.

6. Conclusions

It is highlighted how computational machine learning techniques, like clustering, can be effectively used in analyzing the impacts of various hydrological and meteorological factors on vegetation. Then, a few challenges especially related to clustering spatial data are pointed out. 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 kernel methods are helpful for clustering complex and high dimensional data that is non-linearly separable in the input space. Consequently for developing a system for oil-palm yield prediction, an algorithm, weighted kernel k-means incorporating spatial constraints, is presented which is a central part of the system. The proposed algorithm has the mechanism to handle spatial autocorrelation, noise and outliers in the spatial data. We get promising results on our test data sets. It is hoped that the algorithm would prove to be robust and effective for spatial (climate) data analysis, and it would be very useful for oil-palm yield prediction.

References

[1] US Department of Agriculture, Production Estimates and Crop Assessment Division, 12 November 2004. htttp://www.fas.usda.gov/pecad2/highlights//2004/08/maypalm/

[2] F. Camastra, A. Verri. A Novel Kernel Method for Clustering. IEEE Trans. on Pattern Analysis and Machine Intelligence, vol 27, pp 801-805, May 2005.

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

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

[5] M. Girolami. Mercer Kernel Based Clustering in Feature Space. IEEE Trans. on Neural Networks. Vol 13, 2002.

[6] A. Majid Awan and Mohd. Noor Md. Sap. An Intelligent System based on Kernel Methods for Crop Yield Prediction. To appear in Proc. 10th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2006), 9-12 April 2006, Singapore.

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

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

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

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] M. Steinbach, P-N. Tan, V. Kumar, S. Klooster and C. Potter. Data Mining for the Discovery of Ocean Climate Indices. Proc of the Fifth Workshop on Scientific Data Mining at 2nd SIAM Int. Conf. on Data Mining, 2002.

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

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

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

[19] M.N. Md. Sap, A. Majid Awan. Finding Patterns in Spatial Data Using Kernel-Based Clustering Method. In proc. Int. Conf. on Intelligent Knowledge Systems (IKS-2005), Istanbul, Turkey, 06-08 July 2005.

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

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

[22] A. Majid Awan, Mohd. Noor Md. Sap. Weighted Kernel K-Means Algorithm for Clustering Spatial Data. To appear in WSEAS Transactions n Information Science and Applications, 2006.

[23] A. Majid Awan, Mohd. Noor Md. Sap. A Software Framework for Predicting Oil-Palm Yield from Climate Data. To appear in International Journal of Computational Intelligence, Vol 3, 2006.

[24] M.N. Md. Sap, A. Majid Awan. Finding Spatio-Temporal Patterns in Climate Data using Clustering. Proc 2005 Int. Conf. on Cyberworlds (CW'05), 23-25 November 2005, Singapore.

[25] 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.

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

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

[28] 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.

[29] B. Scholkopf. The kernel trick for distances. In Advances in Neural Information Processing Systems, volume 12, pages 301--307. MIT Press, 2000.

[30] I.S. Dhillon, J. Fan, and Y. Guan. Efficient clustering of very large document collections. In Data Mining for Scientific and Engineering Applications, pp.357–381. Kluwer Academic Publishers, 2001.

[31] C. Elkan. Using the triangle inequality to accelerate k-means. In Proc. of 20th ICML, 2003.