DBCSVM: Density Based Clustering Using Support Vector Machines

Santosh Kumar Rai[1], Nishchol Mishra[2]
[1] SOIT, RGPV
Bhopal, M.P., India

[2] SOIT, RGPV
Bhopal, M.P., India

Abstract
Data categorization is challenging job in a current scenario. The growth rate of a multimedia data are increase day to day in an internet technology. For the better retrieval and efficient searching of a data, a process required for grouping the data. However, data mining can find out helpful implicit information in large databases. To detect the implicit useful information from large databases various data mining techniques are use. Data clustering is an important data mining technique for grouping data sets into different clusters and each cluster having same properties of data. In this paper we have taken image data sets and firstly applying the density based clustering to grouped the images, density based clustering grouped the images according to the nearest feature sets but not grouped outliers, then we used an important super hyperplane classifier support vector machine (SVM) which classify the all outlier left from density based clustering. This method improves the efficiency of image grouping and gives better results.
Keywords: *Classification, Clustering, DBSCAN, SVM*

1. Introduction

The fast progress of internet technology has increasing amounts of multimedia data produced and stored in databases. To extract the implicit and useful information through databases concern in data mining, Data clustering is an important data mining techniques which clusters the data into different groups and data into same group having the similar properties. The goal of clustering is to discover both the dense and the sparse regions in the data set [2, 3].
The objectives of clustering are:

- To discover expected groupings
- To instigate assumption about the data
- To find reliable and valid organization of the data

There are two main approaches to clustering data sets [4,13]
I. Hierarchical Clustering
I.1 Agglomerative clustering
I.2 Divisive clustering
II. Partitioning Clustering
II.1 K- means clustering
II.2 K- medoid clustering
II.2.1 PAM (Partition Around Medoid)
II.2.2 CLARA (Clustering Large Applications)
II.2.3 CLARANS (Clustering Large Applications Based on Randomized **S**earch)
II.2.4 Density Based Clustering

- DBSCAN (Density Based Spatial Clustering of Application of Noise)
- GDBSCAN
- IDBSCAN
- KIDBSCAN
- OPTICS (Ordering Points to Identify the Clustering Structure)

1.1 Hierarchical Clustering vs. Partitioning Clustering Techniques

I. The partitioning clustering [2, 3] techniques partition the database into a predefine number of clusters. In contrast the hierarchical clustering techniques do a sequence of partition. They create a hierarchy of clusters from small too big or big too small.
II. Hierarchical and Partitioning Clustering have key differences in running time, assumptions, input parameters and resultant clusters. Typically, partitioning clustering is faster than hierarchical clustering.
III. Hierarchical clustering requires only a similarity measure, while partitioning clustering requires stronger assumptions such as number of clusters and the initial centers.

IV. Hierarchical clustering does not require any input parameters, while partitioning clustering algorithms require the number of clusters to start running.
V. Hierarchical clustering returns a much more meaningful and subjective division of clusters but partitioning clustering results in exactly k clusters. Hierarchical clustering algorithms are more suitable for categorical data as long as a similarity measure can be defined accordingly [2, 3].

In this paper we have taken image data set and two partitioning clustering methods such as k- means and density based clustering DBSCAN, which group the images into different clusters and each cluster having same properties of data. K-means is a well-known partitioning method; it takes as input a set S of objects and an integer k, and outputs a partition of S into subsets $S_{1,...}S_{2,}S_k$[15]. Through k-means we find cluster id of each cluster. K-means can be easily implemented, and quickly identifies data clustering, but cannot accurately recognize arbitrary shapes. DBSCAN is an important density based clustering algorithm, which takes two input parameters Eps and Minpts [1]. The key idea of DBSCAN is that for each object of cluster, the neighbourhood around an object of given radius (Eps) must include at least minimum number of objects (minpts). DBSCAN starts with arbitrary object q. The neighbourhood of q is obtained by executing a query [1, 11]. If neighbourhood of q is greater than minpts objects then a new cluster with q as the core object is created, and all data objects in neighbourhood of q as seed are assigned to this cluster , otherwise the cluster q is labelled as non-core objects, in turn, are either border objects or noise objects [1,12]. Hence, DBSCAN can locate arbitrary shapes and noise objects. To group the border or noise objects we used hyper-plane classifier Support Vector Machines (SVM), which grouped the all border or noise or outlier left from DBSCAN. On the other hand, the complexity of DBSCAN is very high, because each object must check all data objects to discover its neighbourhood. To reduce the complexity of DBSCAN, this paper present new algorithm DBCSVM (a new grouping schemes density based clustering using Support Vector Machines). The proposed DBCSVM algorithm can efficiently be used for large image data sets and faster grouped the image data sets into different clusters.
The rest of this paper is organized as follows: section 2 describes clustering methods k-means and DBSCAN. In section 3 describe the Support Vector Machines. The proposed algorithm DBCSVM is presented in section 4. Next section 5 shows the result and analysis. Conclusion is drawn in section 6.

2. Clustering Methods

2.1 K-means

In k-means algorithm [1, 13, 14] each cluster is represented by centre of gravity of cluster. K-means clustering is a data mining machine learning algorithm used to cluster observations into groups of related observations without any prior knowledge of those relationships. The k-means algorithm is one of the simplest clustering techniques and it is commonly used in medical imaging, biometrics and related fields. K-means procedure is as follows: (1) Select k data points as the initial centroids. (2) (Re) Assign all points to their closest centroids (3) recomputed the centroid of each newly assembled cluster (4) Repeat steps 2 and 3 until the centroid do not change. However, K-means has some advantages and disadvantages. Advantages: With a large number of variables, K-Means may be computationally faster than hierarchical clustering (if K is small), it may produce tighter clusters than hierarchical clustering, especially if the clusters are globular, K-Means algorithm is its favorable execution time. Disadvantages: Difficulty in comparing quality of the clusters produced (e.g. for different initial partitions or values of K affect outcome), Fixed number of clusters can make it difficult to predict what K should be, It does not work well with non-globular clusters, It cannot accurately recognize arbitrary shapes, it cannot filter out noise.

2.2 DBSCAN

DBSCAN [1, 4] is a density based special clustering of application with noise. Density based clustering locates regions of high density that are separated from one another by regions of low density. Density: Number of points within a specified radius (Eps).
DBSCAN having certain concepts [1-4, 10]:
1. Ɛ- Neighborhood of an Object: The number of objects within a non negative value Ɛ from object is called Ɛ-Neighborhood of an Object, the Ɛ- Neighborhood of an Object p , denoted by $N_ε(p)$.In figure 1 show the Ɛ-Neighborhood of p and q.

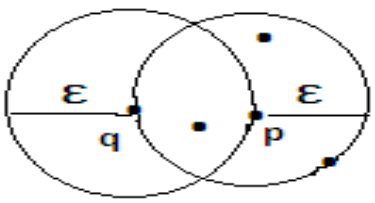

Figure 1 Ɛ-Neighbourhood of p and q

2. Core Object: An object is called core object if the Ɛ-Neighborhood of an object contains at least a minimum of points (threshold), Minpts. That is $N\varepsilon(p) \geq$Minpts. In figure 5 shows the core object.
3. Directly-density-reachable: An object q is directly-density-reachable from an object p with respect to Ɛ and Minpts, if p is acore object and q is in its Ɛ-Neighborhood. That is $q \in N\varepsilon(p)$ and $N\varepsilon(p) \geq$ Minpts.

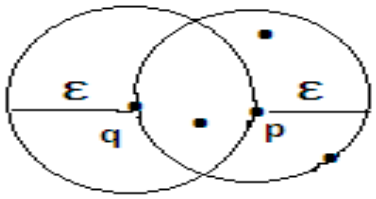

Figure 2 Directly density reachable

In figure 2 Ɛ represents the radius of the circles and given minpts. q is directly density reachable from p because q is within the Ɛ- Neighbourhood of p and p is a core object , P is not directly density reachable from q because p is within the Ɛ- Neighborhood of q but q is not a core object.
4. Density-reachable: An object p is density reachable from q with respect to Ɛ and Minpts if there is a chain of objects $p_{1,\ldots}p_n$ with p_1=q, p_n=p such that p_{1+1} is directly density-reachable from p_i with respect to Ɛ and Minpts for $1 \leq i \leq n$. In figure 3 q is density reachable from p; p is not density reachable from q.

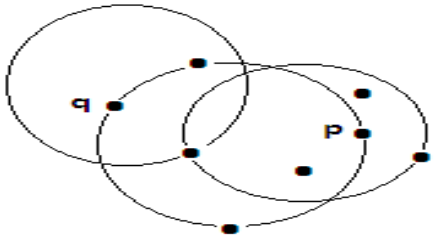

Figure 3 Density-reachable

5. Density-connected: An object p is density reachable from q with respect to Ɛ and Minpts if there is an object O such that both p and q are density-reachable from O with respect to Ɛ and Minpts. In figure 4**,** p and q are density connected

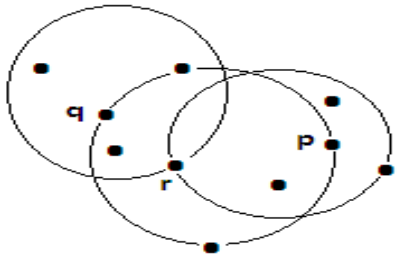

Figure 4 Density-connected

6. Border object: A border object is noncore object, which is always density-reachable from core object. Two border objects are not density-reachable from each other. In figure 5 show the border object.
7. Noise object: A noise object is a non-core object, which is not density-reachable from other core objects.

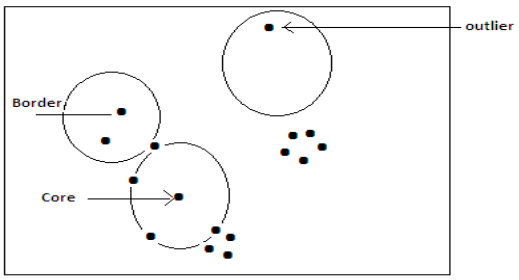

Figure 5 Border object

DBSCAN algorithm takes two parameters Eps and Minpts. The main procedure for DBSCAN is that a neighborhood around an object of a given radius (Eps) has contained at least a minimum number of data objects. If the neighborhood of p contains greater than or equal to Minpts objects, then a new cluster with P as core object is created. DBSCAN collects directly density-reachable objects from core objects and also collects few density-reachable clusters. The process terminates when no new object can be added to any cluster. DBSCAN has some advantages and disadvantages. Advantages [5]: DBSCAN does not require prior knowledge of data cluster, as opposed to k-means; it can find arbitrarily shaped clusters, it has a notion of noise, and it requires just two parameters and is mostly insensitive to the ordering of the points in the database. Disadvantages [5]: The complexity of DBSCAN is very high for large databases; it can only result in a good clustering as good as its distance measure is in the function Euclidean distance measure this distance metric can be rendered almost useless, it does not respond well to data sets with varying

densities (called hierarchical data sets), it cannot group the outlier objects or border objects.

3. Support Vector Machines

In this section, we give brief discussion of SVM. Support Vector Machine is a concept related to the set of supervised learning method, used for classification of the data sets. The basic idea is to find a hyper plane which separates the d-dimensional data perfectly into its two classes [6], which is illustrated in figure 6.Support VectorMachines, were introduced by Vladimir Vapnik and colleagues (AT&T Bell Labs, 1985) [9].

Consider the problem of separating the set of training vectors belonging to two classes, $(x_1 , y_1),\ldots\ldots\ldots, (x_m , y_m)$, where $x_i \in R^n$ is a feature vector and $y_i \in \{+1,-1\}$ is a class label, e.g., image classification problem, +1 denotes indoor image, -1 denotes the outdoor image. If the two classes are linearly separable, the hyper-plane that does the separation is [6, 7]

$$W. x+ b=0 \quad (1)$$

The goal of a SVM is to find the parameter w_o and b_o for an optimal hyper-plane to maximize the distance between the hyper-plane and the closest data point [6, 8]

$$y_i(\omega .x_i + b) \geq 1, i =1,\ldots\ldots m \quad (2)$$

For a given w_o and b_o, the distance of a point x from the optimal hyper-plane defined in is (2) of all the boundaries

$$d(\omega_o, b_o, x) = \frac{|\omega_o.x+b_0|}{||\omega_o||} \quad (3)$$

determined by w and b, the one that maximizes the margin will generalize better than other possible separating hyper-planes.

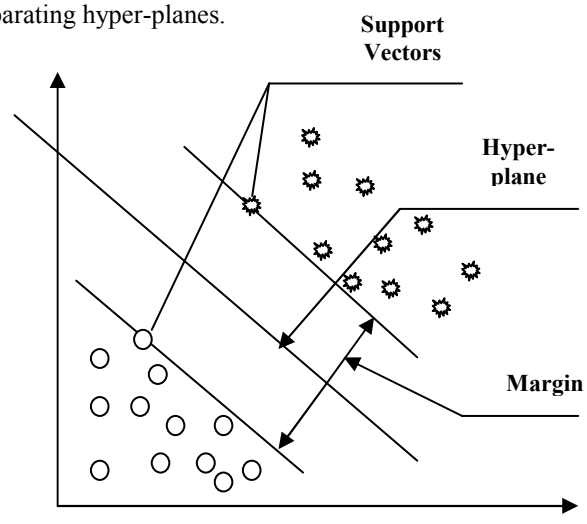

Figure 6 Illustrate the idea of an optimal hyper-plane for linear separable.

4. Proposed Algorithm

4.1 Algorithm

Phase-I

(1) Input: Image data set
(2) Get value of R, G & B (From RGB model)
(3) Convert RGB color space to HSV color space (Hue Saturation Value)

Where:

$$H = \begin{cases} undefined, & if\ MAX = MIN \\ 60 \times \frac{G-B}{MAX-MIN} + 0, & if\ MAX = R\ and\ G \geq B \\ 60 \times \frac{G-B}{MAX-MIN} + 360, & if\ MAX = R\ and\ G < B \\ 60 \times \frac{B-R}{MAX-MIN} + 120, & if\ MAX = G \\ 60 \times \frac{R-G}{MAX-MIN} + 240, & if\ MAX = B \end{cases}$$

$$S = \begin{cases} 0, & if\ MAX = 0 \\ 1 - \frac{MIN}{MAX}, & otherwise \end{cases}$$

$$V = MAX$$

(4) HSV space quantization
(5) Calculate histogram
(6) Calculate DCD extraction (Dominant color descriptor)
(7) Features set of data (F_1, F2, F3.....) →FD
(8) Indexing the features set of image data set (FD)

Phase-II

K_ means (FD, K)
For (i=1 to K) do
Center= Random Generate ();
// randomly select K points from the feature set as centers
End for
While (center <> previous center) do
//For each point of cluster point p.
For (i =1 to K) do
//Compute the distance between p and Center;
End for
$K_{id}=K_1$, K2 ….K_k
Recalculate each cluster center
End While

Phase III

//Apply DBSCAN and Support Vector Machine for classify the Image Data set
DBCSVM (Kid, Eps, Minpts)
Near=Find Nearest ($K_{id,}$ Eps)
//Kid= Cluster id of data set
//Eps=Neighborhood
//Minpts=Threshold value
If (near. Size>=Minpts)
Mark

Class label
else
Unclass label
//Set Data into a SVM Classifier**:**
f: $R^N \rightarrow \{\pm 1(C_1,C_2) \in FD\}$//Function
//C_1=Classify data
//C_2= Unclassified data
Imp [I] = $1/l \sum_{i=1}^{l} | f(c1) - f(c2) |$
// Imp=Imperial error
$w.c_1+b \geq +1$ for $u_i = +1$
$w.c_2+b < -1$ for $u_i = -1$
//where w is weight matrix
// b= biase constant of vector
// *l*=total length of vector,
// u_i = margin

4.2 Flow graph for proposed work

User input:

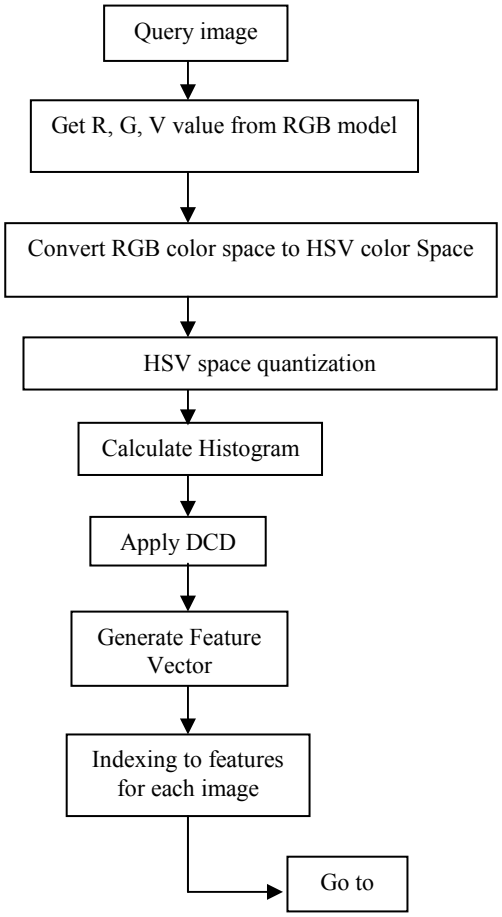

Figure 7 User input for proposed work

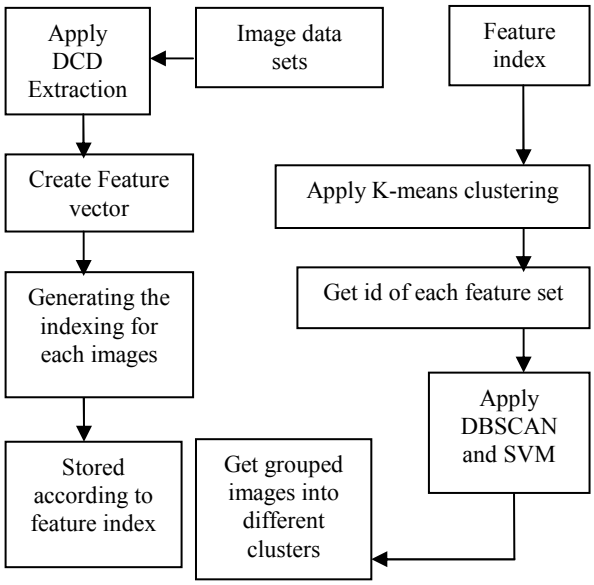

Figure 8 Database for proposed work

5. Experimental Results and Analysis

Experiments were conducted on a desktop computer with 3 GB RAM and Pentium (R) 2.3 GHz Dual-Core CPU and running Microsoft Windows7 Ultimate. All algorithms were implemented in MATLAB 7.8.0 (R2009a).In this method we take UCI or MCI image data sets and format of these images are in JPG.

Figure 9 illustrate that the generation of features sets on the basis of: color, shape, texture, histogram, dimension and direction for the each images in dataset. In the next step we applied k-means algorithm to find id of each features sets, and then enter any id according to the feature id generated, for this corresponding id image is selected from image data sets. In figure 10 first clusters generated, which having five images and all having similar properties such as color, shape, texture. Similarly, we found others cluster such as clusters second, third, fourth and fifth, which is shown in figure 11, 12, 13 and 14 respectively. The grouping of images into different clusters is done through our proposed DBCSVM algorithm. It shows better grouping of images into different clusters. This proposed DBCSVM algorithm using two algorithms DBSCAN and SVM simultaneously.

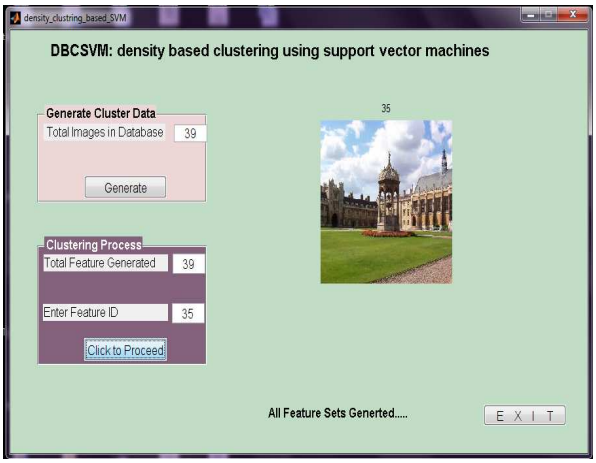

Figure 9 Total Features Generated and Enter Feature ID

Figure 10 First Cluster

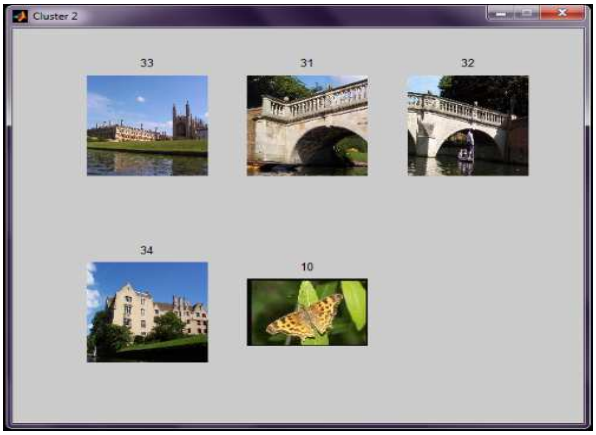

Figure 11 Second Cluster

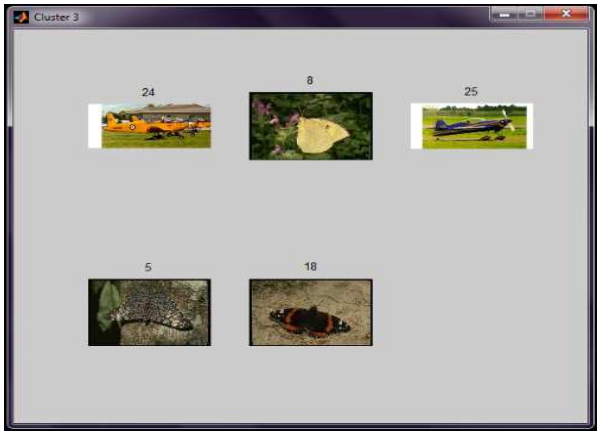

Figure 12 Third Cluster

Figure 13 Fourth Cluster

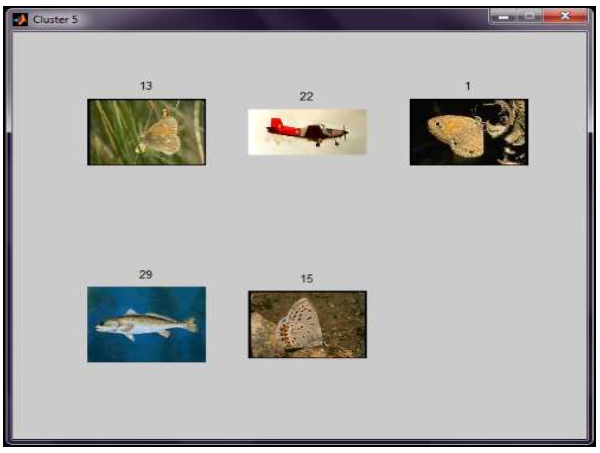

Figure 14 Fifth Cluster

6. Conclusion

This method presents an improved DBSCAN clustering algorithm named DBCSVM: Density Based Clustering Using Support Vector Machines. In the process of feature extraction generator, huge amount of matrix for the calculation of description of feature for the purpose of clustering, for this purpose previous density based clustering take more time and does not give better result. From this method the separation of farer and nearer points are very efficient. The farer points jumps into the next step of clustering. Our method gives better result and takes less time comparison to previous DBSCAN clustering methods.

The work can be summarized as follows:

- The proposed DBCSVM algorithm can efficiently be used for large image data sets.
- The proposed DBCSVM algorithm is faster than another clustering algorithm.
- Through this algorithm we can find better result.

References

[1] Cheng-Fa Tsai, Heng-Fu Yeh, Jui-Fang Chang and Ning-Hang Liu. 2010 PHD: An efficient Data Clustering Scheme using Partition Space technique for Knowledge Discovery in Large Databases, Applied Intelligence, Volume 33, Number 1, Pages 39-53, Springer

[2] Jiawei Han and Micheline Kamber. 2001 Data mining: Concept and Techniques, Morgan Kaufmann Publishers, San Francisco, USA, ISBN 15558604898

[3] Arun K. Pujari 2001 Data Mining Techniques University Press (India) Private Limited.

[4] Pavel Berkhin A Survey of Clustering Data Mining Techniques

[5] K. Mumtaz1 and Dr. K. Duraiswamy An Analysis on Density Based Clustering of Multi Dimensional Spatial Data ,Indian Journal of Computer Science and Engineering Vol. 1 No 1 8-12

[6] Yanni Wang, Bao-Gang Hu.(2002): Hierarchical Image Classification Using Support Vector Machines**,** The 5th Asian Conference on Computer Vision, 23--25, Melbourne, Australia

[7] Dustin Boswell 2002 Introduction to Support Vector Machines

[8] Steve R. Gunn 1998 Support Vector Machines for Classification and Regression ,Technical Report

[9] Vapnik 1995 The nature of statistical learning theory, Springer-Verlag, New York

[10] Wang T-P and Tsai C-F 2006 GDH: An effective and efficient approach to detect arbitrary patterns in clusters with noises in very large databases. Master thesis, National Pingtung University of Science and Technology, Taiwan

[11] Tsai C-F and Yen C-C 2007 ANGEL: A new effective and efficient hybrid clustering technique for large databases. Lect Notes Comput Sci (LNCS) 4426:817–824

[12] EsterM, Kriegel HP, Sander J and Xu X. 1996 A density-based algorithm for discovering clusters in large spatial databases with noise.In: Proceedings of the 2nd international conference on knowledge discovery and data mining, pp 226–231

[13] Leo Wanner 2004 Introduction to Clustering Techniques ,IULA

[14] Andrew W. Moore 2001 K-means and Hierarchical Clustering , Associate Professor School of Computer Science Carnegie Mellon University

[15] Osama Abu Abbas 2008 Comparisons between Data Clustering Algorithms, The International Arab Journal of Information Technology,vol.5,no.3