Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 10, October 2014, pg.497 – 504

RESEARCH ARTICLE

COMPARATIVE STUDY OF VARIOUS CLUSTERING TECHNIQUES

Akshay S. Agrawal
P.G. Scholar, Department of Computer Engg.
ARMIET, Sapgaon, India.
akshay1661@gmail.com

Prof. Sachin Bojewar
Associate Professor,
VIT, Wadala.
sachin.bojewar@vit.edu.in

Abstract:
Clustering is a process of dividing the data into groups of similar objects and dissimilar ones from other objects. Representation of data by fewer clusters necessarily loses fine details, but achieves simplification. Data is model by its clusters. Clustering plays an significant part in applications of data mining such as scientific data exploration, information retrieval, text mining, city-planning, earthquake studies, marketing, spatial database applications, Web analysis, marketing, medical diagnostics, computational biology, etc. Clustering plays a role of active research in several fields such as statistics, pattern recognition and machine learning. Data mining adds complications to very large datasets with many attributes of different types to clustering. Unique computational requirements are imposed on relevant clustering algorithms. A variety of clustering algorithms have recently emerged that meet the various requirements and were successfully applied to many real-life data mining problems.

Key terms: ***Clustering, Feature subset selection, Minimum Spanning Tree***

1. INTRODUCTION

The goal of this study is to provide a universal review of various clustering techniques in data mining. A technique for grouping set of data objects into multiple groups/clusters so that objects within the cluster have high similarity, but are very dissimilar to objects in the other clusters is known as 'clustering'. Clustering is a technique of removing any attribute that is known to be very noisy or not interesting. Dissimilarities and similarities are estimated based on the attribute values representing the objects. Clustering algorithms are used to organize and categorize data for data concretion and model construction, detection of deviation, etc. Common approach of clustering is to find centroid that will represent a certain cluster. Cluster centre will be represented with input vector which measures a similarity unit between input vector and all cluster centroid and determining which cluster is nearest or most similar one. To gain penetration into the data distribution or as a preprocessing step for other data mining algorithms operating on the detected clusters, cluster analysis can be used as a standalone data mining tool. Clustering is unsupervised learning of a hidden data concept. Data mining deals with large databases that can enforce on clustering analysis for additional severe computational requirements. These challenges led to the emergence of powerful broadly applicable data mining clustering methods. Many clustering algorithms have been developed and are categorized from several aspects such as partitioning methods, hierarchical methods and grid-based methods. Data set can be either numeric or

© 2014, IJCSMC All Rights Reserved

categorical. Inherent geometric properties of numeric data can be employed to naturally define distance function between data points and categorical data that can be derived from either quantitative or qualitative data where observations are directly observed from counts.

2. EXISTING DATA CLUSTERING APPROACHES

2.1. Hierarchical Clustering:

Hierarchical clustering builds a cluster hierarchy or a tree of clusters also known as a 'dendogram', an inverted tree that describes the order in which points are merged or clusters are split. Each cluster node contains child clusters and sibling clusters partitions the point covered by their common parents. Such approach allows exploring data on different levels of granularity. In this each item is assigned to a cluster such that '*N*' items can have '*N*' clusters which finds the closest pair of clusters and merge them into a single cluster and compute distance between a new cluster and each of old clusters. Repeat these steps until all items are clustered into *K* no. of clusters. [3]

2.1.1. Agglomerative:

It starts with the points as an individual clusters and, at each step, closest pair of the clusters is merged. This requires defining the notion of cluster distance. Agglomerative hierarchical clustering is a bottom-up clustering method where clusters have sub-clusters, which in turn have sub-clusters, etc. It finds the two clusters that are closest to each other and form one.

Algorithm:

1) Compute the proximity graph, if necessary.
(Sometimes the proximity graph is all that is available.)
2) Merge the closest (most similar) two clusters.
3) Update the proximity matrix to reflect the proximity between the new cluster and the original clusters.
4) Repeat steps 3 and 4 until only a single cluster remains.

2.1.2. Divisive:

It starts with one, all-inclusive cluster and, at each step; a cluster is split into a unity clusters of individual points. In this case, at each step which cluster is to be spited is decided. Divisive technique is a top-down clustering method and is less commonly used. It works in a similar way to agglomerative clustering but in the opposite direction. [3]

Algorithm:

1) Compute a minimum spanning tree for the proximity graph.
2) Create a new cluster by breaking the link corresponding to the largest distance (smallest similarity).
3) Repeat step 2 until only singleton clusters remain.

2.2. Partitioning Clustering:

Partitioning methods are classified into two subcategories viz. centroid and medoid algorithms. Centroid algorithms represent each cluster by using the gravity centre of the instances. The Mediod algorithm represents each cluster by means of the occurrences closest to gravity centre. Partitioning clustering algorithms try to locally improve a certain criterion. They compute the values of the similarity or distance, they order the results, and pick the one that optimizes the criterion. Hence, the majority of them could be considered as greedy-like algorithms. [7]

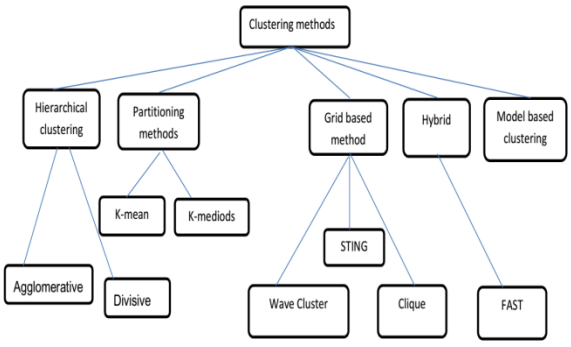

Fig. 2.1.: Clustering Techniques

© 2014, IJCSMC All Rights Reserved

2.2.1. K-means:

It is a partitioning method technique which finds mutual exclusive clusters of spherical shape and generates a specific number of disjoint, flat (non-hierarchical) clusters. Statistical method can be used to cluster and the assign rank values to the cluster categorical data. [2]

Algorithm:

1. Choose A as number of clusters.

2. Initialize the codebook vectors of K-clusters (randomly from instances).

3. For every new sample vector:

3.1. Compute the distance between the new vector and every cluster's codebook vector.

3.2. Recompute the closest codebook vector with the new vector using a learning that decreases in time.

Here categorical data have been converted into numeric by assigning rank value. K-Means algorithm organizes objects into k-partitions where each partition represents a cluster. The algorithm starts out with initial set of means and classifies cases based on their distances to their centers. Next, it computes the cluster means again using the cases that are assigned to the clusters; then, based on the new set of means reclassification of all cases is done. This step keeps repeating until cluster means don't change between successive steps. Finally, the means of cluster is calculated once again and the cases are assigned to their permanent clusters.

2.2.2. K-mediods:

The objective of K-medoid clustering is to find a non-overlapping set of clusters such that each cluster has a most representative point. In this algorithm, rather than calculating the mean of the items in each cluster, a representative item or medoid is chosen for each cluster at each iteration. Medoids for each cluster are calculated by finding object '*I*' within the cluster that minimizes: $\sum_{j \in C}$ **d(i,j)**

where, 'C_i' is the cluster containing object '*i*' and '*d*(*i, j*)' is the distance between objects '*i*' and '*j*'. Medoids for each cluster are calculated by finding object '*i*' within the cluster.[3]

Algorithm:

1. Choose k objects at random to be the initial cluster medoids.

2. Assign each object to the cluster associated with the closest Mediod.

3. Recalculate the positions of the k medoids.

4. Repeat Steps 2 and 3 until the medoids become fixed.

2.3. Grid-based Clustering:

These algorithms mainly focuses on spatial data, *i.e.*, data that model the geometric structure of objects in space, their relationships, properties and operations. The main objective of this algorithm is to quantize the data set into a number of cells and then work with objects belonging to these cells. They build several hierarchical levels of groups of objects instead of relocating points. In this sense, they are closer to hierarchical algorithms but the merging of grids, and consequently clusters, does not depend on a distance measure but it is decided by a predefined parameter. A dense cell contains more than certain numbers of points are connected to form the clusters. Basically grid-based clustering algorithms are classified into: STatistical INformation Grid-based method (STING), WaveCluster, and CLustering In QUEst(CLIQUE). [6]

2.3.1. Wave Cluster:

A multi-resolution clustering algorithm, wave cluster is used to find clusters for very large spatial databases. The goal of the algorithm is to detect clusters from the given set of spatial objects. The data is summarized by imposing a multi dimensional grid structure on to the data space. [3] Transforming the original feature by applying wavelet transform and then finding the dense regions in the new space is the main idea. The signal processing technique that decomposes a signal into different frequency sub bands is known as a wavelet transform. The first step of the wavelet cluster algorithm is to quantize the feature space. In the second step, discrete wavelet transform is applied on the quantized feature space and hence new units are generated. Wave Cluster works with a large number of numerical attributes. [8]

2.3.2. STING:

STING (STastical INformation Grid) is a grid-based

multi-resolution clustering technique in which the embedded spatial area of input object are divided into rectangular cells. Statistical information regarding the attributes in each grid cell, such as the mean, maximum, and minimum values are stored as statistical parameters in these rectangular cells. Statistical parameters of higher level cells can easily be computed from the parameters of lower level cells. The quality of STING clustering depends on the granularity of the lowest level of grid structure as it uses a multi resolution approach to cluster analysis. Moreover, for construction of a parent cell STING does not consider the spatial relationship between the children and their

© 2014, IJCSMC All Rights Reserved

neighboring cells. As a result, the shapes of the resulting clusters are isothetic i.e. all the cluster boundaries are either horizontal or vertical, and 'np' diagonal boundary is detected. Dense clusters can be identified approximately using count and cell size information. STING divides the spatial area into several levels of rectangular cells and irrelevant cells are removed. [3]

2.3.3. Clique:

A clustering algorithm that finds high-density regions by partitioning the data space into cells (hyper-rectangles) and finding the dense cells is CLIQUE (CLustering In QUEst) clustering. Clusters are found by taking the union of all adjacent and high-density cells. Clusters are described by expressing the cluster as a DNF (disjunctive normal form) expression and then simplifying the expression for simplicity and ease of use. CLIQUE is based on the following simple property of clusters: Since a cluster represents a dense region in some subspace of the feature space, there will be dense areas corresponding to the cluster in all lower dimensional subspaces. CLIQUE generates the possible set of *k*-dimensional cells that might possibly be dense by looking at dense $(k-1)$ dimensional cells, since each *k*-dimensional cell must be associated with a set of *k* dense $(k-1)$ dimensional cells. [2]

2.4. Model-based Clustering:

These algorithms find good estimations of model parameters that best fit the data. They can be either partitioned or hierarchical, depending on the structure or model they anticipate about the data set and the way they refine this model to identify partitioning. They are closer to density-based algorithms, in that they grow particular clusters so that the ideal model is improved. However, it sometimes starts with a fixed number of clusters and they do not use the same concept of density. Since, the distance in input space from other units are large; the outlier can be easily detected in model-based clustering. [4]

3. PROPOSED SYSTEM

Feature selection involves identifying a subset of the most useful features that produces compatible results as the original entire set of features. A feature selection algorithm is basically evaluated from the efficiency and effectiveness points of view. The time required to find a subset of features is concerned with the efficiency while the effectiveness is related to the quality of the subset of features. Some feature subset selection algorithms can effectively eliminate irrelevant features but fail to handle redundant features yet some of others can remove the irrelevant while taking care of the redundant features. A Fast clustering-based feature selection algorithm (FAST) is proposed which is based on above criterion handling redundancy and irrelevancy. [1] The Minimum Spanning tree (Kruskal's algorithm) is constructed from the F-Correlation value which is used to find correlation between any pair of features. Kruskal's algorithm is a greedy algorithm in graph theory that finds a minimum spanning tree for a connected weighted graph. It finds a subset of the edges that forms a tree that includes every vertex, where the total weight of all the edges in the tree is minimized. [1]

Methodology:

1. Create a forest F (a set of trees), where each vertex in the graph is a separate tree.
2. Create a set S containing all the edges in the graph.
3. While S is nonempty and F is not yet spanning, an edge with minimum weight from S is removed. If that edge connects two different trees, then add it to the forest, combining two trees into a single tree, otherwise discard that edge. At the termination, the forest forms a minimum spanning forest of the graph.

© 2014, IJCSMC All Rights Reserved

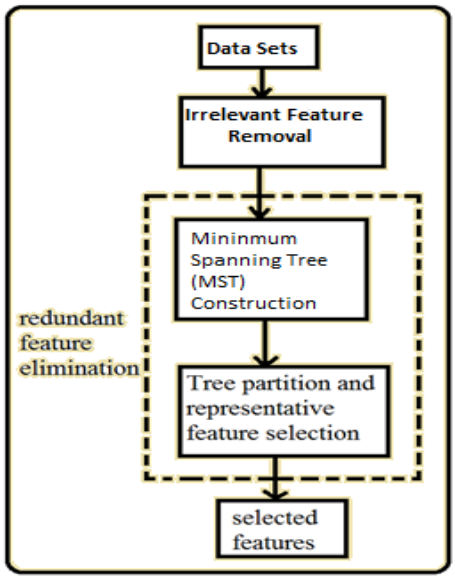

Fig. 3.1.: Feature Subset Selection Algorithm

Features in different clusters are relatively independent; the clustering-based strategy of FAST has a high probability of producing a subset of useful and independent features. To ensure the efficiency of FAST, we adopt the efficient minimum-spanning tree (MST) clustering method.

Proposed Algorithm:

Inputs: D (F1, F2 … Fm, C) (High Dimensional Dataset).

Output: S-Selected feature subset for searching.

Part 1: Removing irrelevant features

The features whose SU(Fi,C) values are greater than a predefined threshold(Ø) comprise the target relevant feature subset. Consider feature input dataset (F).

F´= { F_1´,F_2´,…F_k´} (k<=M)

1. for i = 1 to m do
2. T-Relevance = SU (Fi, C)
3. if T-Relevance > θ then
4. S = S ∪{ };

Part 2: Removing redundant features

The F-correlation SU(Fi´,Fj´) value for each pair of features.

5. G = NULL; //G is a complete graph
6. for each pair of features {F'i, F'j} ⊂ S do
7. F-Correlation = SU (F'i , F'j)
8. F'i and/or F'j to with F-Correlation as the weight of the corresponding edge;
9. MinSpanTree = Kruskal's (G); //Using Kruskal's algorithm to generate minimum spanning tree.

Part 3 : Feature selection.

10. Forest = minSpanTree
11. for each edge $E_{i,j}$ ∈ Forest do
12. if SU (F'i , F'j) < SU(F'i ,C) ∧ SU(F'i , F'j)< SU(F'j , C) then
13. Forest = Forest − E_{ij}
14. S = ϕ
15. for each tree T_i ∈ Forest do
16. F_R^j = argmax F_k∈ SU(F'k , C)
17. S = S ∪{ F_R^j };
18. Return S.

© 2014, IJCSMC All Rights Reserved

The algorithm can be expected to be divided into 3 major parts:
The first part is concerned with removal of irrelevant features;
The second part is used for removing the redundant features and
The final part of the algorithm is concerned with feature selection based on the value of the Forest.
The detailed classification and structural output of the algorithm is described below.

Working:

A. First step:

The data set 'D' with 'm' features *F*= (F1,F2,…Fm) and class 'C', 'I' compute the *T-Relevance* '***SU***' ($\mathcal{F}i$, $\mathcal{C}$) value for every feature ($1 \leq i \leq m$).

B. Second step:

Here the first step is to calculate the *F-Correlation* '***SU***' ($\mathcal{F}'_i$, $\mathcal{F}'_j$) value for each pair of features $\mathcal{F}'_i$ and $\mathcal{F}'_j$ Then, seeing features $\mathcal{F}'_i$ and $\mathcal{F}'_j$ as vertices and '***SU***' ($\mathcal{F}'_i$, $\mathcal{F}'_j$) the edge between vertices $\mathcal{F}'_i$ and $\mathcal{F}'_j$ a weighted complete graph *G= (V,E)* is constructed which is an undirected graph. The complete graph reflects the correlations among the target-relevant features.

C. Third step:

Here, unnecessary edges can be removed. Each tree $T_j \in Forest$ shows a cluster that is denoted as $V(T_j)$, which is the vertex set of T_j . For each cluster $V(T_j)$, select a representative feature whose *T-Relevance* $SU(F_jR, C)$ is the highest. All F_jR (j = 1...|*Forest*|) consist of the final feature subset ∪ F_jR .

4. ADVANTAGES OF FAST CLUSTERING

Table 4.1.: Advantages and Disadvantages

SR.NO.	Techniques (or) Algorithms	Advantages	Disadvantages
1.	FAST Algorithm	Improve the performance of the classifiers. The efficiently and effectively deal with both irrelevant and redundant features, and obtain a good feature subset.	--
2.	Consistency Measure	Fast, Remove noisy and irrelevant data.	Unable to handle large volumes of data.
3.	Wrapper Approach	Accuracy is high.	Computational complexity is large.
4.	Filter Approach	Suitable for very large features.	Accuracy is not guaranteed.
5.	Agglomerative linkage algorithm	Reduce Complexity.	Decrease the Quality when dimensionality becomes high.
6.	INTERACT Algorithm	Improve Accuracy.	Only deal with irrelevant data.
7.	Distributional clustering	Higher classification accuracy.	Difficult to evaluation.
8.	Relief Algorithm	Improve efficiency and Reduce Cost.	Powerless to detect redundancy.
9.	Grid based method	Jobs can automatically restart if a failure occurs.	You may need to have a fast interconnect between compute resources.
10.	Model based method	Clusters can be characterized by a small number of parameters.	Need large data sets. Hard to estimate the number of clusters.

5. COMPARATIVE STUDY OF EXISTING METHODS

The clustering algorithms are compared on the basis of:

1. The data type used for clustering;
2. The shape of the clusters;
3. Time and space complexity for clustering;
4. The size of data set;

© 2014, IJCSMC All Rights Reserved

Table 5.1.: Comparative Study

Clustering Techniques	Methods	Data Type	Cluster Shape	Complexity	Data Set	Measure	Advantages	Disadvantages
Hierarchical Method	CURE	Numerical	Arbitrary	$O(N^{-2})$	Large	Similarity Measure	1. Attempt to address the scalability problem and improve the quality of clustering results.	1. Do not scale well with the number of data objects.
	BIRICH	Numerical	Spherical	O(N)(Time)	Large	Feature Tree	1. Suitable large scale data sets in main memory. 2. Minimize number of scans. 3. I/O costs are very low.	1. Suffers from identifying only convex or spherical clusters of uniform size.
	CHAMELEON	Discrete	Arbitrary	$O(N^{-2})$	Large	Similarity Measure	1. Strongly relies on graph partitioning.	1. Issue of scaling to large data set that cannot fit into main memory.
	ROCK	Mix	Graph	$O(KN^{-2})$	Small	Similarity Measure	1. Introduces some global property and thus provides better quality.	1. It can breakdown if the choice of parameter is incorrect w.r.t. the data set being clustered.
Partitioning Method	K-means	Numerical	Spherical	O(NKD)(T) O(N+K)(S)	Large	Mean	1. Is relatively scalable and efficient in processing large data sets.	1. Different sized cluster. 2. Clusters of different densities. 3. Non globular clusters. 4. Wrong number of clusters. 5. Outliers and empty clusters.
	K-mediods	Numerical	Arbitrary	O(TKN)	Large	Mediod	1. Is more robust in presence of outliers and noise. 2. Perform better for large data sets.	1. More costly. 2. Requires user to specify 'k'. 3. Do not scale well for large number of data sets.
	CLARA	Numerical	Arbitrary	O(K(40+K2) + -K (N-K))+	Sample	Mediod	1. Deals with larger data sets. 2. More scalable and efficient.	1. If sampling is biased then we cannot have a good clustering. 2. Trade off-of efficiency.
	CLARANS	Numerical	Arbitrary	Quadratic in total performance	Sample	Mediod	1. More effective than CLARA. 2. Handle outliers.	1. Computational complexity is $O(N^2)$ where n is the number of objects. 2. Clustering quality depend upon sampling method.
Grid-based Method	STING	Spatial	Vertical and Horizontal Boundaries	O(N)	Large	Distance	1. Grid environments are much more modular and don't have single points of failure. 2. If one of the servers/desktops within the grid fail there are plenty of other resources able to pick the load. 3. Jobs can automatically restart if a failure occurs.	1. You may need to have a fast interconnect between compute resources (gigabit Ethernet at a minimum). 2. Licensing across many servers may make it prohibitive for some apps.
	Wave cluster	Spatial	Arbitrary					
	CLIQUE	High Dimensional	Arbitrary					
Hybrid Method	FAST	High Dimensional	Spherical	Sub Quadratic Time Complexity	Large	Feature Subset Selection (MST) Tree	1. Improve the performance of the classifiers. 2. The efficiently and effectively deal with both irrelevant and redundant features, and obtain a good feature subset.	--

N = number of objects, K = number of clusters, S = size of sample.

6. FUTURE SCOPE

As a future work, a FAST clustering algorithm for removing redundancy and irrelevancy from feature subset selection algorithm can be developed and implemented. More challenging domains with more features and a higher proportion of irrelevant ones will require more sophisticated methods for feature selection. Although further increases in efficiency would increase the number of states examined such constant factor improvements cannot eliminate problems caused by exponential growth in the number of feature sets. However viewing these problems in terms of heuristic search suggests some places to look for solutions in general we must,

1. Invent more intelligent techniques for selecting an initial set of features from which to start the search.
2. Formulate search control methods that take advantage of structure in the space of feature sets.
3. Devise improved frameworks better even than the wrapper method for evaluating the usefulness of alternative feature sets
4. Design better halting criteria that will improve efficiency without sacrificing useful feature sets.

7. CONCLUSION

In this paper, we have proposed a clustering algorithm, FAST for high dimensional data. The algorithm includes (i) irrelevant features removal (ii) construction of a minimum spanning tree (MST) from, and (iii) partitioning the MST and selecting the representative features. Feature subset selection should be able to recognize and remove as much of the unrelated and redundant information. In the proposed algorithm, a cluster will be used to develop a MST for faster searching of relevant data from high dimensional data. Each cluster will be treated as a single feature and thus volume of data to be processed is drastically reduced.

© 2014, IJCSMC All Rights Reserved

FAST algorithm will obtain the best proportion of selected features, the best runtime, and the best classification accuracy. Overall the system will be effective in generating more relevant and accurate features which can provide faster results.

REFERENCES

[1] Qinbao Song, Jingjie Ni and Guangtao Wang, A Fast Clustering-Based Feature Subset Selection Algorithm for High Dimensional Data, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING VOL:25 NO:1 YEAR 2013.

[2] Osama Abu Abbas, Comparision between Data Clustering Algorithms, The International Arab journal of Information Technology, Vol. 5, No. 3,July 2008.

[3] A Review: Comparative Study of Various Clustering Techniques in Data Mining, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 3, March 2013.

[4] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. Automatic

Subspace Clustering of High Dimensional Data for Data Mining Applications. In *Proceedings of the International Conference on Management of Data, (SIGMOD)*, volume 27(2) of *SIGMOD Record*, pages 94–105, Seattle,WA, USA, 1–4 June 1998. ACM Press.

[5] Jiawei Han and Michelle Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2001.

[6] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases. KDD'96.

[7] Literature Survey on Clustering Techniques, http://www.slideshare.net/IOSR/a0310112-26684753.

[8] BOTTOU, L. and BENGIO, Y. 1995. Convergence properties of the K-means algorithms. In Tesauro, G. and Touretzky, D. (Eds.) Advances in Neural Information Processing Systems 7, 585-592, The MIT Press, Cambridge, MA.

[9] DHILLON, I., FAN, J., and GUAN, Y. 2001. Efficient clustering of very large document collections. In Grossman, R.L., Kamath, C., Kegelmeyer, P., Kumar, V., and Namburu, R.R. (Eds.) Data Mining for Scientific and Engineering Applications, Kluwer Academic Publishers.

[10] Alexander Hinneburg and Daniel A. Keim. An Efficient Approach to Clustering in Large Multimedia Databases with Noise. In *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining, (KDD)*, pages 58–65, New York, NY, USA, 27–31 August 1998. AAAI Press.

© 2014, IJCSMC All Rights Reserved