ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayaShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

MST Clustering and Relevancy Analysis for Key Element Identification Process

Satheesh V[1], Poongothai T[2]

M.Tech, Dept of IT, K.S.R. College of Engineering, Tamilnadu, India[1]

Associate Professor, Dept of IT, K.S.R. College of Engineering, Tamilnadu, India[2]

ABSTRACT: Data items are grouped with reference to the similarity under the clustering process. Similarity measures are used to analyze the relationship between the transactions. Vector based similarity models are suitable for low dimensional data values. High dimensional data values are clustered using subspace clustering methods. Feature selection involves identifying a subset of the most useful features that produces compatible results. A feature selection algorithm is constructed with the consideration of efficiency and effectiveness factors. The efficiency concerns the time required to find a subset of features. The effectiveness is related to the quality of the subset of features.

High dimensional data clustering and feature selection process is carried out using the Fast clustering algorithm. FAST algorithm is divided into two steps. In the first step, features are divided into clusters by using graph-theoretic clustering methods. In the second step, the most representative feature is selected from each cluster to form a subset of features. 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. Minimum-Spanning Tree (MST) clustering method is adopted to ensure the efficiency of FAST. Feature subset selection algorithm is used to identify the features from the clusters.

Transaction similarity analysis is carried out with different type of correlation measures in the feature selection process. Dynamic feature intervals can be used to distinguish features. Redundant feature filtering mechanism is used to filter the similar features. Custom threshold is used to improve the cluster accuracy.

I. INTRODUCTION

The high dimensionality of data poses a challenge to learning tasks such as classification. In the presence of many irrelevant features, classification algorithms tend to overfit training data. Many features can be removed without performance deterioration. Feature selection is one effective means to remove irrelevant features. Optimal feature selection requires an exponentially large search space. Researchers often resort to various approximations to determine relevant features. However, a single feature can be considered irrelevant based on its correlation with the class; but when combined with other features, it becomes very relevant. Unintentional removal of these features can result in the loss of useful information and thus may cause poor classification performance. This is studied as attribute interaction. For example, MONK1 is a data set involving feature interaction. There are six features in MONK1 and the target concept of MONK1 is: $(A_1 = A_2)$ or $(A_5 = 1)$. Here A_1 and A_2 are two interacting features. Considered individually, the correlation between A_1 and the class C is zero, measured by mutual information. Hence, A_1 or A_2 is irrelevant when each is individually evaluated. However, if we combine A_1 with A_2, they are strongly relevant in defining the target concept. An intrinsic character of feature interaction is its irreducibility, i.e., a feature could lose its relevance due to the absence of its interacting feature(s).

A clustering is essentially a set of such partions, usually containing all objects in the data set. Additionally, it may specify the relationship of the clusters to each other, for example a hierarchy of clusters embedded in each other. Clusterings can be roughly distinguished in:

- Hard clustering: each object belongs to a cluster or not

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayaShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

- Soft clustering: each object belongs to each cluster to a certain degree

Data clustering algorithms can be hierarchical or partitional. Hierarchical algorithms find successive clusters using previously established clusters, whereas partitional algorithms determine all clusters at once. Hierarchical algorithms can be agglomerative or divisive. Agglomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters. Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters. Two-way clustering, co-clustering or bi-clustering are the names for clusterings where not only the objects are clustered but also the features of the objects, i.e., if the data is represented in a data matrix, the row and columns are clustered simultaneously. Another important distinction is whether the clustering uses symmetric or asymmetric distances. A property of Euclidean space is that distances are symmetric. In other applications, this is not the case.

The graph based clustering is also referred as hierarchical clustering scheme. Hierarchical clustering builds, or breaks up, a hierarchy of clusters. The traditional representation of this hierarchy is a tree, with individual elements at one end and a single cluster containing every element at the other. Agglomerative algorithms begin at the top of the tree, whereas divisive algorithms begin at the bottom. Cutting the tree at a given height will give a clustering at a selected precision. In the following example, cutting after the second row will yield clusters {a} {b c} {d e} {f}. Cutting after the third row will yield clusters {a} {b c} {d e f}, which is a coarser clustering, with a smaller number of larger clusters.

In the Machine Learning (ML) scientific community there is a need for rigorous and correct statistical analysis of published results, due to the fact that the development or modifications of algorithms is a relatively easy task. The main inconvenient related to this necessity is to understand and study the statistics and to know the exact techniques which can or cannot be applied depending on the situation, that is, type of results obtained. In a recently published paper in JMLR by Demsar, a group of useful guidelines are given in order to perform a correct analysis when we compare a set of classifiers over multiple data sets. Demsar recommends a set of non-parametric statistical techniques for comparing classifiers under these circumstances, given that the sample of results obtained by them does not fulfill the required conditions and it is not large enough for making a parametric statistical analysis. He analyzed the behavior of the proposed statistics on classification tasks and he checked that they are more convenient than parametric techniques.

II. RELATED WORK

Feature subset selection can be viewed as the process of identifying and removing as many irrelevant and redundant features as possible. This is because 1) irrelevant features do not contribute to the predictive accuracy and 2) redundant features do not redound to getting a better predictor for that they provide mostly information which is already present in other feature(s).

Of the many feature subset selection algorithms, some can effectively eliminate irrelevant features but fail to handle redundant features [1] yet some of others can eliminate the irrelevant while taking care of the redundant features [2]. Our proposed FAST algorithm falls into the second group. Traditionally, feature subset selection research has focused on searching for relevant features. A well-known example is Relief which weighs each feature according to its ability to discriminate instances under different targets based on distance-based criteria function. However, Relief is ineffective at removing redundant features as two predictive but highly correlated features are likely both to be highly weighted. Relief-F extends, enabling this method to work with noisy and incomplete data sets and to deal with multiclass problems, but still cannot identify redundant features.

However, along with irrelevant features, redundant features also affect the speed and accuracy of learning algorithms, and thus should be eliminated. FCBF [2] and CMIM [3] are examples that take into consideration the redundant features. CFS is achieved by the hypothesis that a good feature subset is one that contains features highly correlated with the target, yet uncorrelated with each other. FCBF ([2], [4]) is a fast filter method which can identify relevant features as well as redundancy among relevant features without pairwise correlation analysis. CMIM [3] iteratively picks features which maximize their mutual information with the class to predict, conditionally to the response of any feature already

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayaShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

picked. Different from these algorithms, our proposed the FAST algorithm employs the clustering-based method to choose features.

Recently, hierarchical clustering has been adopted in word selection in the context of text classification (e.g., [5]). Distributional clustering has been used to cluster words into groups based either on their participation in particular grammatical relations with other words by Pereira et al. or on the distribution of class labels associated with each word by Baker and McCallum. As distributional clustering of words are agglomerative in nature, and result in suboptimal word clusters and high computational cost, Dhillon et al. [5] proposed a new information-theoretic divisive algorithm for word clustering and applied it to text classification. Butterworth et al. [8] proposed to cluster features using a special metric of Barthelemy-Montjardet distance, and then makes use of the dendrogram of the resulting cluster hierarchy to choose the most relevant attributes. Unfortunately, the cluster evaluation measure based on Barthelemy-Montjardet distance does not identify a feature subset that allows the classifiers to improve their original performance accuracy. Furthermore, even compared with other feature selection methods, the obtained accuracy is lower.

Hierarchical clustering also has been used to select features on spectral data. Van Dijck and Van Hulle [7] proposed a hybrid filter/wrapper feature subset selection algorithm for regression. Krier et al. [6] presented a methodology combining hierarchical constrained clustering of spectral variables and selection of clusters by mutual information. Their feature clustering method is similar to that of Van Dijck and Van Hulle [7] except that the former forces every cluster to contain consecutive features only. Both methods employed agglomerative hierarchical clustering to remove redundant features.

Quite different from these hierarchical clustering-based algorithms, our proposed FAST algorithm uses minimum spanning tree-based method to cluster features. Meanwhile, it does not assume that data points are grouped around centers or separated by a regular geometric curve. Moreover, our proposed FAST does not limit to some specific types of data.

III. PROBLEM STATEMENT

Fast clustering-based feature selection algorithm (FAST) is used to cluster the high dimensional data and feature selection process. FAST algorithm is divided into two steps. In the first step, features are divided into clusters by using graph-theoretic clustering methods. In the second step, the most representative feature is selected from each cluster to form a subset of features. 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. Minimum-Spanning Tree (MST) clustering method is adopted to ensure the efficiency of FAST. Feature subset selection algorithm is used to identify the features from the clusters. The following drawbacks are identified from the existing system.

- Correlation measures are not optimized
- Limited clustering accuracy
- Feature relevance is low
- Threshold is not optimized

IV. MINIMUM SPANNING TREE CLUSTERING

The collection of bridges in a LAN can be considered a graph whose nodes are the bridges and whose edges are the cables connecting the bridges. To break loops in the LAN while maintaining access to all LAN segments, the bridges collectively compute a spanning tree. The spanning tree is not necessarily a minimum cost spanning tree. A network administrator can reduce the cost of a spanning tree, if necessary, by altering some of the configuration parameters in such a way as to affect the choice of the root of the spanning tree. The spanning tree that the bridges compute using the Spanning Tree Protocol can be determined using the following rules. The example network at the right, below, will be used to illustrate the rules.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayaShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

The first algorithm for finding a minimum spanning tree was developed by Czech scientist Otakar Borůvka in 1926. Its purpose was an efficient electrical coverage of Moravia. There are now two algorithms commonly used, Prim's algorithm and Kruskal's algorithm. All three are greedy algorithms that run in polynomial time, so the problem of finding such trees is in P. Another greedy algorithm not as commonly used is the reverse-delete algorithm, which is the reverse of Kruskal's algorithm. The fastest minimum spanning tree algorithm to date was developed by Bernard Chazelle, and based on Borůvka's. Its running time is O(e α(e,v)), where *e* is the number of edges, *v* refers to the number of vertices and α is the classical functional inverse of the Ackermann function. The function α grows extremely slowly, so that for all practical purposes it may be considered a constant no greater than 4; thus Chazelle's algorithm takes very close to O(e) time.

More recently, research has focused on solving the minimum spanning tree problem in a highly parallelized manner. For example, the pragmatic 2003 Fast Shared-Memory Algorithms for Computing the Minimum Spanning Forest of Sparse Graphs" by David A. Bader and Guojing Cong demonstrates an algorithm that can compute MSTs 5 times faster on 8 processors than an optimized sequential algorithm. Typically, parallel algorithms are based on Boruvka's algorithm Prim's and especially Kruskal's algorithm does not scale as well to additional processors.

It has been shown by J. Michael Steele based on work by A. M. Frieze that given a complete graph on n vertices, with edge weights chosen from a continuous random distribution *f* such that $f'(0) > 0$, as *n* approaches infinity the size of the MST approaches $\zeta(3) / f'(0)$, where ζ is the Riemann zeta function. For uniform random weights the exact expected size of the minimum spanning tree has been computed for small complete graphs.

V. CLUSTER BASED FEATURE SELECTION SCHEME

With the aim of choosing a subset of good features with respect to the target concepts, feature subset selection is an effective way for reducing dimensionality, removing irrelevant data, increasing learning accuracy, and improving result comprehensibility. Many feature subset selection methods have been proposed and studied for machine learning applications. They can be divided into four broad categories: the Embedded, Wrapper, Filter, and Hybrid approaches. The embedded methods incorporate feature selection as a part of the training process and are usually specific to given learning algorithms, and therefore may be more efficient than the other three categories. Traditional machine learning algorithms like decision trees or artificial neural networks are examples of embedded approaches. The wrapper methods use the predictive accuracy of a predetermined learning algorithm to determine the goodness of the selected subsets, the accuracy of the learning algorithms is usually high. However, the generality of the selected features is limited and the computational complexity is large. The filter methods are independent of learning algorithms, with good generality. Their computational complexity is low, but the accuracy of the learning algorithms is not guaranteed. The hybrid methods are a combination of filter and wrapper methods using a filter method to reduce search space that will be considered by the subsequent wrapper. They mainly focus on combining filter and wrapper methods to achieve the best possible performance with a particular learning algorithm with similar time complexity of the filter methods. The wrapper methods are computationally expensive and tend to overfit on small training sets. The filter methods, in addition to their generality, are usually a good choice when the number of features is very large. Thus, we will focus on the filter method in this paper.

With respect to the filter feature selection methods, the application of cluster analysis has been demonstrated to be more effective than traditional feature selection algorithms. Pereira et al., Baker and McCallum and Dhillonet al. employed the distributional clustering of words to reduce the dimensionality of text data. In cluster analysis, graph-theoretic methods have been well studied and used in many applications. Their results have, sometimes, the best agreement with human performance. The general graph-theoretic clustering is simple: compute a neighborhood graph of instances, then delete any edge in the graph that is much longer/shorter than its neighbors. The result is a forest and each tree in the forest represents a cluster. In our study, we apply graph-theoretic clustering methods to features. In particular, we adopt the minimum spanning tree (MST)-based clustering algorithms, because they do not assume that data points are grouped around centers or separated by a regular geometric curve and have been widely used in practice.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayaShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Based on the MST method, we propose a Fast clustering bAsed feature Selection algoriThm (FAST). The FAST algorithm works in two steps. In the first step, features are divided into clusters by using graph-theoretic clustering methods. In the second step, the most representative feature that is strongly related to target classes is selected from each cluster to form the final subset of features. 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. The proposed feature subset selection algorithm FAST was tested upon 35 publicly available image, microarray, and text data sets. The experimental results show that, compared with other five different types of feature subset selection algorithms, the proposed algorithm not only reduces the number of features, but also improves the performances of the four well-known different types of classifiers.

VI. FAST CLUSTERING AND FEATURE SELECTION PROCESS

The proposed FAST algorithm logically consists of three steps: 1) removing irrelevant features, 2) constructing an MST from relative ones, and 3) partitioning the MST and selecting representative features. For a data set D with m features $F = \{F_1, F_2, \ldots, F_m\}$ and class C, we compute the T-Relevance $SU(F_i, C)$ value for each feature F_i $(1 \leq i \leq m)$ in the first step. The features whose $SU(F_i, C)$ values are greater than a predefined threshold Φ comprise the target-relevant feature subset $F' = \{F'_1, F'_2, \ldots, F'_k\}$ $(k \leq m)$.

In the second step, we first calculate the F-Correlation $SU(F'_i, F'_j)$ value for each pair of features F'_i and F'_j $(F'_i, F'_j, \varepsilon F'_i \cap i \neq j)$. Then, viewing features F'_I and F'_j as vertices and $SU(F'_i, F'_j)$ $(i \neq j)$ as the weight of the edge between vertices F'_i and F'_j, a weighted complete graph G = (V,E) is constructed where $V = \{ F'_i | F'_i \varepsilon F'_i \cap [1, k]\}$ and $E = \{(F'_i, F'_j)\}(F'_i, F'_j \varepsilon F'_i \cap, I, j \varepsilon [1, k] \cap i \neq j$. As symmetric uncertainty is symmetric further the FCorrelation $SU(F'_i, F'_j)$ is symmetric as well, thus G is an undirected graph.

The complete graph G reflects the correlations among all the target-relevant features. Unfortunately, graph G has k vertices and k(k - 1)/2 edges. For high-dimensional data, it is heavily dense and the edges with different weights are strongly interweaved. Moreover, the decomposition of complete graph is NP-hard. Thus for graph G, we build an MST, which connects all vertices such that the sum of the weights of the edges is the minimum, using the wellknown Prim algorithm. The weight of edge (F'_i, F'_j) is F-Correlation $SU(F'_i, F'_j)$.

After building the MST, in the third step, we first remove the edges $E = \{(F'_i, F'_j) | (F'_i, F'_j \varepsilon F'_i \cap, I, j \varepsilon [1, k] \cap i \neq j$, whose weights are smaller than both of the T-Relevance $SU(F'_i, C)$ and $SU(F'_j, C)$, from the MST. Each deletion results in two disconnected trees T_1 and T_2. Assuming the set of vertices in any one of the final trees to be V (T), we have the property that for each pair of vertices $(F'_i, F'_j \varepsilon V (T))$, $SU(F'_i, F'_j) \geq SU(F'_i, C)$ ν $SU(F'_i, F'_j) \geq SU(F'_j, C)$ always holds. From Definition 6, we know that this property guarantees the features in V (T) are redundant.

This can be illustrated by an example. Suppose the MST is generated from a complete graph G. In order to cluster the features, we first traverse all the six edges, and then decide to remove the edge (F_0, F_4) because its weight $SU(F_0, F_4) = 0:3$ is smaller than both $SU(F_0, C) = 0:5$ and $SU(F_4, C) = 0:7$. This makes the MST is clustered into two clusters denoted as V (T_1) and V (T_2). Each cluster is an MST as well. Take V (T_1) as an example. We know that $SU(F_0, F_1) > SU(F_1, C)$, $SU(F_1, F_2) > SU(F_1, C)$ ^ $SU(F_1, F_2) > SU(F_2, C)$, $SU(F_1, F3) > SU(F_1, C)$ ^ $SU(F_1, F3) > SU(F3, C)$. We also observed that there is no edge exists between F_0 and F_2, F_0 and F_3, and F_2 and F_3. Considering that T_1 is an MST, so the $SU(F_0,F_2)$ is greater than $SU(F_0, F_1)$ and $SU(F_1, F_2)$, $SU(F_0, F_3)$ is greater than $SU(F_0, F_1)$ and $SU(F_1, F3)$, and $SU(F_2, F_3)$ is greater than $SU(F_1, F_2)$ and $SU(F_2, F_3)$. Thus, $SU(F_0, F_2) > SU(F_0, C)$^ $SU(F_0, F_2) > SU(F_2, C)$, $SU(F_0, F_3) > SU(F_0, C)$ ^ $SU(F_0, F_3) > SU(F_3, C)$, and $SU(F_2, F_3) > SU(F_2, C)$ ^ $SU(F_2, F_3) > SU(F3, C)$ also hold. As the mutual information between any pair $(F_i, F_j)(i, j = 0, 1, 2, 3$ ^ $i \neq j)$ of F_0, F_1, F_2, and F_3 is greater than the mutual information between class C and Fi or Fj, features F_0, F_1, F_2, and F_3 are redundant.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayaShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

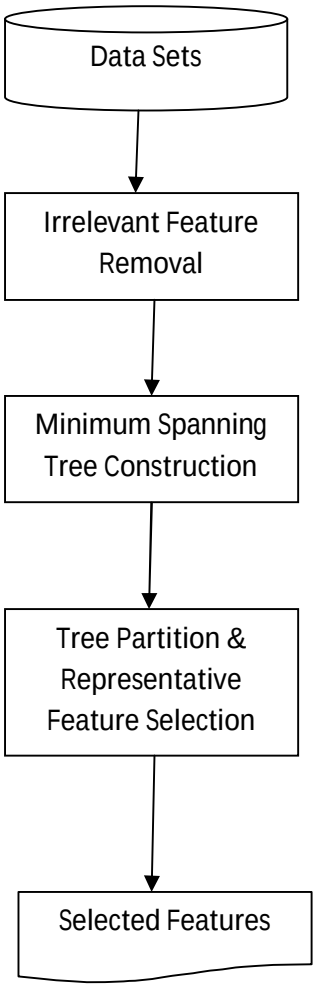

Fig. 1. Framework of the feature subset selection algorithm.

After removing all the unnecessary edges, a forest Forest is obtained. Each tree T_j ε Forest represents a cluster that is denoted as V (T_j), which is the vertex set of T_j as well. As illustrated above, the features in each cluster are redundant, so for each cluster V (T_j) we choose a representative feature F^j_R whose T-Relevance SU(F^j_R, C) is the greatest. All F^j_R (j = 1 . . . |Forest|) comprise the final feature subset $\dot{U}F^j_R$.

The details of the FAST algorithm is shown in Algorithm 1.

Inputs: D(F_1,F_2,…,F_m, C) – the given data set θ – the T-Relevance threshold.

Output: S – selected feature subset

1. for i = 1 to m do
2. T-Relevance = SU (F_i, C)
3. If T-Relevance > θ then

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayaShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

4. S = S Ụ {F_i};
5. G = NULL;
6. For each pair of features {F'$_i$, F'$_j$} C S do
7. F-Correlation = SU {F'$_i$, F'$_j$}
8. Add F'$_i$, and/or F'$_j$ to G with F-Correlation as the weight of the corresponding edge;
9. minSpanTree = Prim (G);
10. Forest = minSpanTree
11. For each edge E_{ij} Є Forest do
12. If SU(F'$_i$, F'$_j$) < SU (F'$_i$, C) ʌ SU (F'$_i$, F'$_j$) < SU (F'$_i$, F'$_j$) then
13. Forest = Forest – E_{ij}
14. S = ɸ
15. For each tree T_i Є Forest do
16. F^j_R = argmax$_{F'k \in Ti}$ SU(F'$_k$,C)
17. S = S U {F^j_R};
18. return S

VII. FEATURE SELECTION USING MST CLUSTERS

The feature selection process is improved with a set of correlation measures. Dynamic feature intervals can be used to distinguish features. Redundant feature filtering mechanism is used to filter the similar features. Custom threshold is used to improve the cluster accuracy.

The graph based clustering algorithm is designed with Minimum Spanning Tree (MST). Correlation measures are optimized to improve the cluster results. Feature selection process is enhanced with dynamic threshold values. The system is designed with five major modules. They are data preprocess, irrelevant filtering, MST construction, cluster process and feature selection.

Data preprocess is designed to perform data cleaning with missing value assignment Process. Irrelevant filtering module is designed to filter irrelevant features with correlation analysis. MST construction module is designed to construct Minimum Spanning Tree (MST) with transactions. Cluster process module is designed to partition the MST with boundaries. Feature selection module is designed to fetch features from cluster results.

7.1. Data Preprocess

Noisy data remove and missing data update operations are carried out under the data preprocess. Redundant data values are removed from the transactional data collection. Aggregation based data substitution mechanism is used for missing data update process. Dimensionality analysis is performed for high dimensional data values.

7.2. Irrelevant Filtering

Irrelevant filtering process is carried out to remove irrelevant features. Correlation measures are used in the relevancy analysis process. Relevancy is analyzed for all features. A threshold value is used to filter the feature values.

7.3. MST Construction

Graph theoretic method is applied for the tree construction. Minimum Spanning Tree (MST) is constructed with the neighborhood information. Shorter/longer edges are removed with reference to its neighbors. The MST produces a forest with a set of trees.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayaShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

7.4. Cluster Process

Features are divided into clusters by using graph-theoretic clustering methods. Fast clustering algorithm is used for the data partitioning process. The Minimum Spanning Tree is used in the clustering process. Trees under the MST are separated with interval values as clusters.

7.5. Feature Selection

Feature selection is applied to filter a subset of useful and independent features. Features in different clusters are relatively independent. Similar features are filtered from the cluster results using redundant feature filtering method. The features strongly related to target classes is selected from each cluster to form the final subset of features.

VIII. CONCLUSION

The high dimensional data values are grouped using the clustering technique. Feature selection methods are used to select key elements in the transactions. FAST algorithm is used to select features from high dimensional data values. Correlation measures are used to improve the feature selection process. The system achieves high feature selection quality. Process time is low in the feature selection scheme. The selected features can be applied for classification process. Cluster accuracy is high in the correlation measures based feature selection process.

REFERENCES

[1] G. Forman, "An Extensive Empirical Study of Feature Selection Metrics for Text Classification," J. Machine Learning Research, vol. 3, pp. 1289-1305, 2003.
[2] L. Yu and H. Liu, "Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution," Proc. 20th Int'l Conf. Machine Leaning, vol. 20, no. 2, pp. 856-863, 2003.
[3] F. Fleuret, "Fast Binary Feature Selection with Conditional Mutual Information," J. Machine Learning Research, vol. 5, pp. 1531-1555, 2004.
[4] L. Yu and H. Liu, "Efficient Feature Selection via Analysis of Relevance and Redundancy," J. Machine Learning Research, vol. 10, no. 5, pp. 1205-1224, 2004.
[5] I.S. Dhillon, S. Mallela, and R. Kumar, "A Divisive Information Theoretic Feature Clustering Algorithm for Text Classification," J. Machine Learning Research, vol. 3, pp. 1265-1287, 2003.
[6] C. Krier, D. Francois, F. Rossi, and M. Verleysen, "Feature Clustering and Mutual Information for the Selection of Variables in Spectral Data," Proc. European Symp. Artificial Neural Networks Advances in Computational Intelligence and Learning, pp. 157-162, 2007.
[7] G. Van Dijck and M.M. Van Hulle, "Speeding Up the Wrapper Feature Subset Selection in Regression by Mutual Information Relevance and Redundancy Analysis," Proc. Int'l Conf. Artificial Neural Networks, 2006.
[8] R. Butterworth, G. Piatetsky-Shapiro, and D.A. Simovici, "On Feature Selection through Clustering," Proc. IEEE Fifth Int'l Conf. Data Mining, pp. 581-584, 2005.
[9] 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, January 2013.