INTERNATIONAL RESEARCH JOURNAL IN ADVANCED ENGINEERING AND TECHNOLOGY (IRJAET) E - ISSN: 2454-4752 P - ISSN : 2454-4744
VOL 2 ISSUE 6 (2016) PAGES 1310 - 1314
RECEIVED : 29/10/2016. PUBLISHED: 07/11/2016
November 7, 2016

AN EFFECTIVE RE-CLUSTER BASED SUBSET SELECTION USING MST AND HEURISTIC ALGORITHM

[1]R.Sivasankari, [2]S.Vanakovarayan,
[1]PG Scholar, Dept Of Computer Science Engineering, Mailam Engineering College,Villupuram,
[2]Assistant Professor Of Computer Science Engineering, Mailam Engineering College, Villupuram.

Abstract:

Data streams are massive, fast-changing, and in-finite. Clustering is a prominent task in mining data streams, which group similar objects in a cluster. With the aim of choosing a Re-Cluster 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. While the efficiency concerns the time required to find a re-cluster subset of features, the effectiveness is related to the quality of the subset of features. In this project, proposed clustering based subset selection 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 a subset of features. To ensure the efficiency of this algorithm, we are going to use RMR method with heuristic algorithm. A heuristic algorithm used for solving a problem more quickly or for finding an approximate re-cluster subset selection solution. *Minimum Redundancy Maximum Relevance* (mRMR) selection used to be more powerful than the maximum relevance selection. It will provide effective way to predict the efficiency and effectiveness of the clustering based subset selection algorithm.

Keywords : Re-Cluster, MRMR, Graph-theoretic.

1. INTRODUCTION

In earlier days research in Artificial Intelligence (AI) was focused on the development of 2 formalisms, inference mechanisms and tools to operationalize Knowledge-based Systems (KBS). Typically, the development efforts were restricted to the realization of small KBSs in order to study the feasibility of the different approaches. Though these studies offered rather promising results, the transfer of this technology into commercial use in order to build large KBSs failed in many cases. The situation was directly comparable to a similar situation in the construction of traditional software systems, called „software crisis“ in the late sixties: the means to develop small academic prototypes did not scale up to the design and maintenance of large, long living commercial systems. In the same way as the software crisis resulted in the establishment of the discipline Software Engineering the unsatisfactory situation in constructing KBSs made clear the need for more methodological approaches. So the goal of the new discipline Knowledge Engineering (KE) is similar to that of Software Engineering: turning the process of constructing KBSs from an art into an engineering discipline. This requires the analysis of the building and maintenance process itself and the development of appropriate methods, languages, and tools specialized for developing KBSs.Subsequently, we will first give an overview of some important historical developments in KE: special emphasis will be put on the paradigm shift from the so-called transfer approach to the so-called modeling approach. This paradigm shift is sometimes also considered as the transfer from first generation expert systems to second generation expert systems. Based on this discussion will be

INTERNATIONAL RESEARCH JOURNAL IN ADVANCED ENGINEERING AND TECHNOLOGY (IRJAET) E - ISSN: 2454-4752 P - ISSN : 2454-4744
VOL 2 ISSUE 6 (2016) PAGES 1310 - 1314
RECEIVED : 29/10/2016. PUBLISHED: 07/11/2016
November 7, 2016

concluded by describing two prominent developments in the late eighties: Role-limiting Methods and Generic Tasks.

2. EXISTING SYSTEM

Data stream clustering is typically done as a two-stage process with an online part which summarizes the data into many micro-clusters or grid cells and then, in an offline process, these micro-clusters (cells) are re-clustered/merged into a smaller number of final clusters. Since the re-clustering is an offline process and thus not time critical, it is typically not discussed in detail in papers about new data stream clustering algorithms. Most papers suggest using an (sometimes slightly modified) existing conventional clustering algorithm (e.g., weighted k-means in CluStream) where the micro-clusters are used as pseudo points. Another approach used in Den Stream is to use reach ability where all micro-clusters which are less than a given distance from each other are linked together to form clusters. Grid-based algorithms typically merge adjacent dense grid cells to form larger clusters (see, e.g., the original version of D-Stream and MR-Stream).

3. PROPOSED WORK

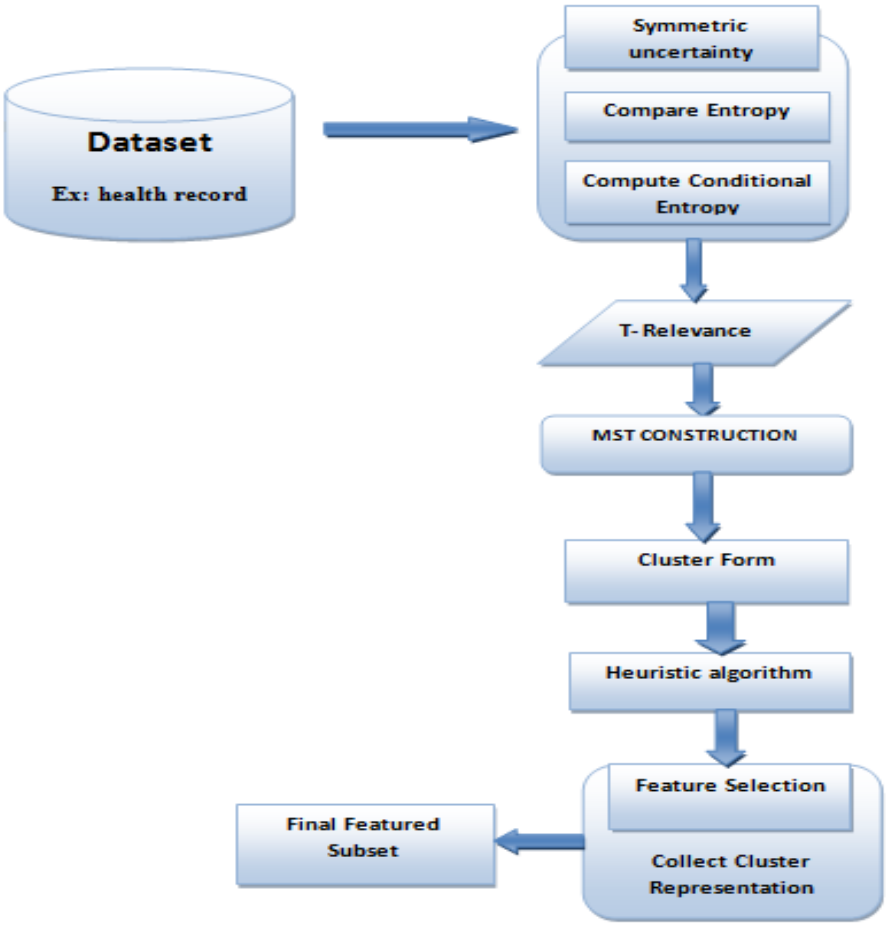

Fig.1.Proposed Architecture

In this project, proposed Clustering based subset Selection algorithm uses minimum spanning tree-based method to cluster features. Moreover, our proposed algorithm does not limit to some specific types of data. Irrelevant features, along with redundant features, severely affect the accuracy of the learning machines. Thus, feature subset selection should be able to identify and remove as much of the irrelevant and redundant information as possible. Moreover, "good feature subsets contain features highly correlated with (predictive of) the class, yet uncorrelated with (not predictive of) each other." In our proposed Cluster based subset Selection algorithm, it involves 1) the construction of the minimum spanning tree from a weighted complete graph; 2) the partitioning of the MST into a forest with each tree representing a cluster; and 3) the selection of representative features from the micro-clusters.

4. CLUSTER ANALYSIS

Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters). It is a main task of exploratory data mining, and a common technique for statistical data analysis, used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, bioinformatics, data compression, and computer graphics. Cluster analysis itself is not one specific algorithm, but the general task to be solved. It can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them. Popular notions of clusters include groups with small distances among the cluster members, dense areas of the data space, intervals or particular statistical distributions. Clustering can therefore be formulated as a multi-objective optimization problem. The appropriate clustering algorithm and parameter settings (including values such as the distance function to use, a density threshold or the number of expected clusters) depend on the individual data set and intended use of the results. Cluster analysis as such is not an automatic task, but an iterative process of knowledge discovery or interactive multi-objective optimization that involves trial and failure. It is often necessary to modify data pre-processing and model parameters until the result achieves the desired properties.

5. MODULES DESCRIPTION

1) Load Data and Classify

Load the data into the process. The data has to be preprocessed for removing missing values, noise and outliers. Then the given dataset must be converted into the arff format which is the standard format for WEKA toolkit. From the arff format, only the attributes and the values are extracted and stored into the database. By considering the last column of the dataset as the class attribute and select the distinct class labels from that and classify the entire dataset with respect to class labels.

2) Information Gain Computation

Relevant features have strong correlation with target concept so are always necessary for a best subset, while redundant features are not because their values are completely correlated with each other. Thus, notions of feature redundancy and feature relevance are normally in terms of feature correlation and feature-target concept correlation. To find the relevance of each attribute with the class label, Information gain is computed in this module. This is also said to be Mutual Information measure. Mutual information measures how much the distribution of the feature values and target classes differ from statistical independence. This is a nonlinear estimation of correlation between feature values or feature values and target classes.

INTERNATIONAL RESEARCH JOURNAL IN ADVANCED ENGINEERING AND TECHNOLOGY (IRJAET) E - ISSN: 2454-4752 P - ISSN : 2454-4744
VOL 2 ISSUE 6 (2016) PAGES 1310 - 1314
RECEIVED : 29/10/2016. PUBLISHED: 07/11/2016
November 7, 2016

3) T-Relevance Calculation

The relevance between the feature Fi € F and the target concept C is referred to as the T-Relevance of Fi and C, and denoted by SU(Fi,C). If SU(Fi,C) is greater than a predetermined threshold , we say that Fi is a strong T-Relevance feature.

$$SU(X,Y) = \frac{2 \times Gain(X|Y)}{H(X) + H(Y)}.$$

After finding the relevance value, the redundant attributes will be removed with respect to the threshold value.

6. ALGORITHM

inputs: $D(F_1, F_2, ..., F_m, C)$ - the given data set
θ - the *T-Relevance* threshold.
output: S - selected feature subset .
//==== Part 1 : Irrelevant Feature Removal ====
for $i = 1$ *to* m **do**
  T-Relevance = SU (F_i, C)
  if *T-Relevance* $> \theta$ **then**
    $S = S \cup \{F_i\}$;
//==== Part 2 : Minimum Spanning Tree Construction ====
G = NULL; *//G is a complete graph*
for *each pair of features* $\{F'_i, F'_j\} \subset S$ **do**
  F-Correlation = SU (F'_i, F'_j)
  Add F'_i *and/or* F'_j *to G with F-Correlation as the weight of the corresponding edge;*
minSpanTree = *Prim* (G); *//Using Prim Algorithm to generate the minimum spanning tree*
//==== Part 3 : Tree Partition and Representative Feature Selection ====
Forest = *minSpanTree*
for *each edge* $E_{ij} \in$ *Forest* **do**
  if $\mathrm{SU}(F'_i, F'_j) < \mathrm{SU}(F'_i, C) \wedge \mathrm{SU}(F'_i, F'_j) < \mathrm{SU}(F'_j, C)$ **then**
    Forest = *Forest* $- E_{ij}$
$S = \phi$
for *each tree* $T_i \in$ *Forest* **do**
  $F_R^j = \mathrm{argmax}_{F'_k \in T_i} \mathrm{SU}(F'_k, C)$
  $S = S \cup \{F_R^j\}$;
return S

Fig.2.Algorithm

After building the MST, in the third step, we first remove the edges whose weights are smaller than both of the T-Relevance SU(Fi`, C) and SU(Fj`, C), from the MST. After removing all the unnecessary edges, a forest Forest is obtained. Each tree Tj € Forest represents a cluster that is denoted as V (Tj), which is the vertex set of Tj as well. As illustrated above, the features in each

cluster are redundant, so for each cluster V (Tj) we choose a representative feature Fj R whose T-Relevance SU(Fj R,C) is the greatest.

CONCLUSION

In this paper, we have developed the first data stream clustering algorithm which explicitly records the density in the area shared by micro-clusters and uses this information for reclustering. We have introduced the shared density graph together with the algorithms needed to maintain the graph in the online component of a data stream mining algorithm. Although, we showed that the worst-case memory requirements of the shared density graph grow extremely fast with data dimensionality, complexity analysis and experiments reveal that the procedure can be effectively applied to data sets of moderate dimensionality. Experiments also show that shared-density reclustering already performs extremely well when the online data stream clustering component is set to produce a small number of large MCs. Other popular reclustering strategies can only slightly improve over the results of shared density reclustering and need significantly more MCs to achieve comparable results.

REFERENCES

[1] Clustering Performance on Evolving Data Streams: Assessing Algorithms and Evaluation Measures within MOA, Author - Philipp Kranen ; Hardy Kremer ; Timm Jansen ; Thomas Seidl .

[2] Organizing multimedia big data using semantic based video content extraction technique, Author - Manju ; P. Valarmathie.

[3] Evaluation Methodology for Multiclass Novelty Detection Algorithms , Author - Elaine R. Faria ; Isabel J. C. R. Goncalves ; Joao Gama .

[4] Joint Image-Text News Topic Detection and Tracking by Multimodal Topic And-Or Graph, Author - Weixin Li ; Jungseock Joo ; Hang Qi ; Song-Chun Zhu.

[5] Performance evaluation of distance measures for preprocessing of set-valued data in feature vector generated from LOD datasets, Author - Rajesh Mahule ; Akshenndra Garg .

[6] Analyzing Enterprise Storage Workloads With Graph Modeling and Clustering, Author - Yang Zhou ; Ling Liu ; Sangeetha Seshadri .