VSB-E ALGORITHM USING WEKA

Pooja Saharan[1], Rashmi Mishra[2], Isha Jauhari[3]

[1,2,3]Assistant Professor, CSE Dept, ABES Engineering College, Ghaziabad, U.P, (India)

ABSTRACT

Evolutionary clustering is a research area addressing the problem of clustering time stamped data. In this paper propose an algorithm for evolutionary clustering using WEKA. A variance score based approach for evolutionary clustering must satisfy the two criteria of evolutionary clustering. Paper provide theoretical as well as experimental proofs to support our claims.

Keywords: Evolutionary Clustering, Snapshot Cost, Temporal Cost, Variance Score, Time Stamped Data, Variance Score Based Evolutionary

I. INTRODUCTION

The discovery of evolving communities in dynamic networks is an important research topic that poses challenging tasks [3]. In data mining, clustering can be done offline to extract usage patterns and give recommendations that are highly dependent on the quality of clustering solution. Clustering is a grouping of data objects such that the objects within the group are similar to one another and different from the objects in other groups. Clustering is used in image processing, bioinformatics, web mining and many more areas [1]. Evolutionary clustering is the problem of processing time stamped data to produce a sequence of clustering, that is, a clustering for each time step of the system [1].An evolutionary clustering should take care of two conflicting criteria: Preserving the current cluster quality and not deviating too much from the recent history.

1.1 Problem Statement

Evolutionary clustering was introduced by Chakrabarti.et.al. in [1] as the problem of clustering data coming at different time steps to produce a sequence of clustering. At each time step a new clustering must be produced by simultaneously optimizing two conflicting criteria. The first is that the clustering should reflect as accurately as possible the data coming during the current time step. The second is that each clustering should not shift dramatically from one time step to the successive. For these the snapshot cost (SC) and the temporal cost (T C), is defined respectively. The snapshot cost SC measures how well a cluster structure represents the data at time t. The temporal cost T C measures how similar the cluster structure (C) is with the previous cluster (C_{t-1}). Chakrabarti et al. defined the cost function for generic data objects.

$\text{Cost} = a * SC + (1 - a) * T\,C$

Where “a” is an input parameter used by the user to emphasize one of the two objectives. When a = 1 the approach returns the clustering without temporal smoothing. When a = 0, however, the same clustering of the previous time step is produced.

II. RELATED WORK

Vozalis and Margaritis from University of Marcedonia in Greece proposed that they provide a review of the experiments we conducted on two contrasting recommender systems' algorithms: classic Collaborative Filtering and Item-based Filtering [12]. Remco.et.al proposed that WEKA is a popular machine learning workbench with a development life of nearly two decades. A recent development in WEKA is the inclusion of package management, so that packages can easily be added to a given installation [5]. The first stable release of the WEKA 3 software coincided with the publication of the first edition of *Data Mining* by Witten and Frank. Moreover, a simple experiment was conducted using a data mining application. Wekato apply data mining algorithms to recommender system..Han.et.al proposed that our ability to generate and collect data has been increasing rapidly. This explosive growth has generated an even more urgent need for new techniques and automated tools that can help us transform this data into useful information and knowledge [6]. Folino and Pizzuti proposed evolutionary based clustering methods try to maximize cluster accuracy, with respect to incoming data of the current time step, and minimize clustering drift from one time step to the successive one. In order to optimize both these two competing objectives.Shankar.et.al proposed an algorithm for evolutionary clustering using frequent item sets.

III. FRAMEWORK

3.1 Overview of the Framework

User satisfaction is the most important factor of the success of a recommender system which is an accurate recommendation within a reasonable time. In commercial systems, it is measured by number of recommended items that has been bought. For non-commercial systems, it is measured by asking for users feed-back. To properly employ a Recommender

System, it is important to study the domain for which it is being used. Evolutionary clustering algorithm is evaluated by testing against various datasets available on sites like movielens.com, amazon.com, AlloCine, Zagat, LibraryThing, Last.fm, Pandora, StumbleUpon, Netix etc.

3.2 Data Mining Tool: WEKA

In implementation a comparative study of variety of feature selection methods for data mining, using WEKA (data mining tool) is done. WEKA was developed at the University of Waikato in New Zealand, and the name stands for *Waikato Environment for Knowledge Analysis*. WEKA is a machine learning workbench that supports many activities of machine learning practitioners [5] .Machine learning is the study of algorithms that automatically improve their performance with experience [13].Machine learning algorithms can be broadly characterized by the language used to represent learned knowledge. WEKA is machine learning/data mining software written in Java language (distributed under the GNU Public License) [5].

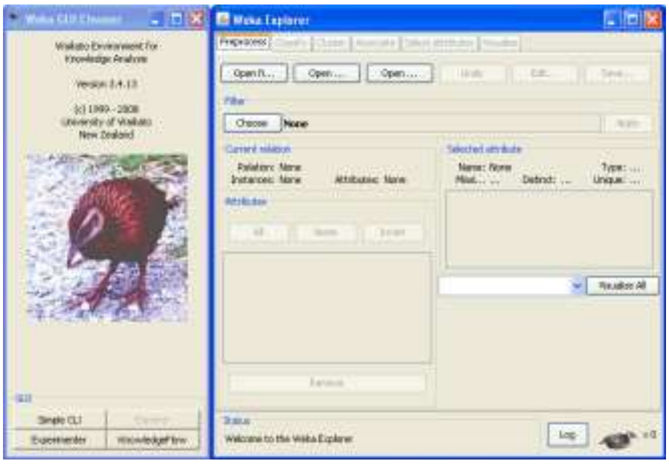

Fig.1 Weka Interface

WEKA’s Functionality [5]:

1. Data preprocessing. As well as a native file format (ARFF). WEKA supports various other formats (for instance CSV, Matlab ASCII files).
2. Classification. Classifiers are divided into “Bayesian” methods (Naive Bayes, Bayesian nets, etc.), lazy methods (nearest neighbor and variants), rule-based methods (decision tables, OneR, RIPPER), tree learners function-based learners
3. Clustering. Unsupervised learning is supported by several clustering schemes, including EM based mixture models, k-means, and various hierarchical clustering algorithms.
4. Attribute selection. The set of attributes used is essential for Classification performance.
5. Data visualization. Data can be inspected visually by plotting attribute values against the class, or against other attribute values. There are specialized tools for visualization, such as a tree viewer for any method that produces classification trees, a Bayes network viewer.

3.3 Design Tool: NetBeans

NetBeans refers to both a platform framework for Java desktop applications and an integrated development environment for developing. Netbeans is a GUI design tool. We can also create new modules for NetBeans IDE itself. For example, one can write modules that make your favorite cutting-edge technologies available to users of NetBeans IDE. Alternatively, one might create a module to provide an additional editor feature. NetBeans IDE not only provides great productivity tools, but also includes sample applications and tutorial solutions that show you complex technologies at work. These samples are provided as ready-to-use NetBeans IDE projects and each comes with an informative readme file so you can get started quickly.

IV. ALGORITHM

Here, we are presenting our Variance Score Based Evolutionary-Clustering Algorithm that satisfies the below conditions:

Input: User-Item rating table.

Output: Clusters of items. Variance of user similarity inside each cluster is small.

Definition of optimal clustering: In each group, try removing items, one after another according to a predefined order. After each removal of item, recompute user similarity. Finally compute the variance of user similarity during this process of item removal. The optimal clustering is the clustering with smallest average user similarity variance among all clusters.

Algorithm:

1. Cluster initializing: Assigns top 10% items that are rated by most users to each cluster consecutively. Initially each group will have approximately the same number of items.
2. Compute variance score in each cluster.
3. For each unassigned item i:

3.1. Try assigning *i* to all clusters, in each group compute average user similarity variance score.

3.2. Actually assign *i* to the cluster with smallest average user similarity variance score.

3.3 Update user similarity in that cluster.

4. Start again from the beginning of item list, for each item *i*:

4.1. Try removing *i* from the current cluster, compute average user similarity variance score in that cluster.

1.2. Try assigning *i* to all other clusters, in each group compute average user similarity variance score.

1.3. Move *i* to the cluster with smallest average user similarity variance. If the current cluster has smallest user similarity variance score, item will remain in that cluster. Go to step 5.

1.4. Update user similarity in previous and new cluster of item *i.*

5. Repeat step 4 until no item moves from one cluster to another in a loop.

V. EXPERIMENT

Evolutionary clustering algorithm is implementing using WEKA and NETBEANS tool. In NETBEANS tool, implement the objective function for the evolutionary clustering algorithm. The evolutionary algorithm is based upon the cost function .Cost function is composed of two competing objectives, the snapshot cost (SC) and the temporal cost(TC) .Folino and Pizzuti proposed the cost function .

Cost=a*SC+ (1-a)*TC

To maximize the snapshot cost (SC), we employ Variance function and to minimize the temporal cost (TC), we employ Normalized Mutual Information (NMI). To implement the cost function, we employ variance score. In WEKA tool, a comparison is made between evolutionary clustering algorithm and various recommendation algorithms.

VI. CONCLUSIONS

In this paper, provided an algorithm to perform evolutionary clustering using WEKA. Considered the problem of clustering time stamped data. Plan to look into how this algorithm helps us with improving the quality of the recommender systemand to make a comparison between evolutionary clustering algorithm and various recommendation algorithms.

REFERENCES

[1] Chakrabarti, D., Kumar, R., & Tomkins, A. (2006). Evolutionary clustering. Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining KDD 06, 8(4), 554. ACM Press.

[2] Demir, G. N., Uyar, A. S., & Oguducu, S. (2007). Graph-based sequence clustering through multiobjective evolutionary algorithms for web recommender systems. Computer Engineering, 2, 1943-1950. ACM Press.

[3] Folino, F., & Pizzuti, C. (2010). Multiobjective Evolutionary Community Detection for Dynamic Networks. Proceedings of the 12th annual conference on Genetic and Evolutionary Computation GECCO2010 (pp. 535-536). ACM Press.

[4] Hall, M. A., & Frank, E. (2010). WEKA — Experiences with a Java Open-Source Project.Journal of Machine Learning Research, 11, 2533- 2541.

[5] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. (2009).The WEKA data mining software.ACM SIGKDD Explorations Newsletter, 11(1), 10.

[6] Han, J., & Kamber, M. (2006). Data Mining: Concepts and Techniques. Techniques (p. 770). Morgan Kaufmann.

[7] Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems.ACM, Transactions on Information Systems, 22(1), 5-53. ACM Press.

[8] Hruschka, E. R., Campello, R. J. G. B., Freitas, A. A., & De Carvalho, A.C. P. L. F. (2009). A survey of evolutionary algorithms for clustering. IEEE Transactions on Systems Man and Cybernetics Part C Applications and Reviews, 39(2), 133-155. IEEE Press.

[9] Kwon, Y. (2008). Improving top-n recommendation techniques using Rating variance. Proceedings of the 2008 ACM conference on Recommender systems RecSys 08, 307.ACM Press.

[10] O’Connor, M., & Herlocker, J. (2001). Clustering items for collaborative filtering. Human Factors.Citeseer.

[11] Shankar, R., Kiran, G. V. R., & Pudi, V. (2010). Evolutionary clustering using frequent itemsets. Proceedings of the First International Workshop on Novel Data Stream Pattern Mining Techniques StreamKDD 10 (pp. 25-30). ACM Press.

[12] Vozalis, E. G., & Margaritis, K. G. (2003). Recommender Systems: An Experimental Comparison of two Filtering Algorithms.Compute..

[13] Yang, Y., & Pedersen, J. O. (1997). A Comparative Study on Feature Selection in Text Categorization.Methods (Vol. 20, pp. 412-420). Citeseer.

[14] Bellogín, A., Cantador, I., Castells, P., & Ortigosa, Á. (2008). Discovering Relevant Preferences in a Personalised Recommender System using Machine Learning Techniques. Work, 82-96