RESEARCH ARTICLE OPEN ACCESS

An Efficient Adaptive Graph Time-Variant Classification (AGTVC) Using Roughset Based Online Streaming Feature Selection Algorithm

N. Arul Kumar[1], S.Vinodkumar[2]
[1]M.Phil Research Scholar, Department of Computer Science
[2]Assistant Professor, PG Department of Computer Science
Sree Saraswathi Thyagaraja College, Pollachi, Coimbatore, India

Abstract:

Adaptive incremental sub-graph classification is a significant tool for examining data with structure dependence. In this paper proposed a novel Adaptive Graph Time-Variant Classification (AGTVC) Using Roughset Based Online Streaming Feature Selection algorithm mines the relevant features from the synthetic real-world social network data sets which improve the graph learning prediction accuracy than previous methods. The proposed AGTVC algorithm divide feature selection and load into memory in a mini-batch manner, which is a important reduction in memory and running time. Experiments results shows the AGTVC algorithms can decrease both the processing time and memory cost. The ROSFS algorithm learns both feature selection and possible results to converts a high dimensional problem into a big constraint problem with respect to feature vector. The experimental results are reanalyzed with several constrains such as number of dimensions versus objective, running time and accuracy. Based on the results generated on this paper, it concludes that AGTCV accuracy and performance increases compared to the previous method of ISJF algorithm.

Keywords — **Data Mining, Feature Selection, Graph Classification, Roughset.**

I. INTRODUCTION

Data mining is the process of finding previously an unknown patterns and trends in databases and using that information to build predictive models. Data mining combines statistical analysis, machine learning and database technology to extract hidden patterns and relationships from large databases.

A data graph offers a very attractive way of representing high-dimensional data and discovering the underlying information, e.g., the neighborhood structure, essential for the data. In the literature, neighborhood graphs have demonstrated wide usage in a great number of machine learning, data mining, and computer vision problems including classification, clustering, regression, dimensionality reduction, relevance ranking, and so on. However, scalability of data graphs to the explosive growth of data sets remains challenging. For example, the time complexity for constructing a neighborhood graph is quadratic in the database size, which is computationally infeasible for practical large-scale data sets of millions up to billions of samples [12] [13].

Graph based Feature selection is a field of Machine Learning and Pattern Recognition and consists in reducing the dimensionality of the data by eliminating those features which are noisy, redundant or irrelevant for a classification problem [7] [11]. Similarly feature weighting is a generalization of feature selection, which assigns weights to each feature. For the case of feature selection the weights are 0 or 1. Another term is feature extraction which in the literature refer both to the calculus of the feature values from the original data (e.g. the extraction of cornerness or SIFT descriptors from images) and to the combination of basic features into high level features. In the literature the term feature extraction also refers to a different pattern recognition process: the transformation of the feature space also referred to as feature transform. Some widely used feature transform methods are Principal Component Analysis and Independent Component Analysis

Graph classification is an important branch of data mining research, given much structured and semi-structured data can be represented as graphs. Images, text and biological data are just a few examples [8-10]. However, recently, time-variant graph data generated from dynamic graphs has entered the domain. For example, for the information propagation of graphs (i.e., a series of graphs in time series), when analyzing the information propagation in time windows, a sequence of graphs showing information propagation over time, where the target is to predict the class label that describes

the outbreak or non-outbreak of an information propagation record. At a specific point in time, the status of the information diffusion is a graph; however the graph is evolutionary over time. Therefore, information diffusion at different stages forms a time-variant graph.

A time-variant graph can be used to characterize the changing nature of a structured object [14]. As both the node attributes and graph structure keep changing in each time-variant graph, in this thesis, we refer to a sequence of graphs whose node content and/or network structure continuously change as a time variant graph. The first address a simple case where the graph structure is fixed but node content continuously evolves, which becomes the networked time series problem. In this case, only the attribute of nodes changes over time but the structure of the graph is static [15]. The variation of the nodes attribute over time forms the time series. By combining with the static graph structure, we call this kind of time-variant graph a networked time series.

II. RELATED WORK

In [1] authors discussed a model for data stream classification observations the data stream classification problem from the end of view of a active approach in which synchronized training and test streams are utilizes for dynamic classification of data sets. This framework replicates real-life conditions efficiently, since it is attractive to categorize test streams in actual time over developing training and test stream. The plan here is to produce a classification system in which the training form can adjust quickly to modify of the underlying data stream. In order to attain this objective, authors proposed an on-demand classification procedure which can enthusiastically select the suitable window of past training data to construct the classifier.

In [2] article, authors investigated the problems of substructure correspondence search using indexed attributes in graph databases. By changing the edge relaxation proportion of a query graph into the maximum authorized attribute misses, the structural filtering algorithm can filter graphs lacking performing pair-wise similarity multiplication. It is an additional exposed that using moreover too little or too several attributes can outcome in poor filtering presentation. Thus the challenge is to plan an effectual attribute set collection scheme that could maximize the filtering ability. The authors proved that the difficulty of optimal attribute set selection is $\Omega(2m)$ in the worst case, where m is the amount of attributes for collection. In preparation, they identified some criteria to build successful feature sets for filtering, and show that merging attributes with comparable dimension and selectivity can progress the filtering and investigate performance extensively within a multi-filter composition structure.

In [3] authors proposed a new online feature selection structure for applications with streaming features where the information of the complete feature space is unidentified in advance. The authors defined streaming features as features that current flow in one by one above time whereas the amount of training instances remains fixed. This is in difference with traditional online learning schemes that only contract with successively additional interpretation, with little concentration being paid to streaming features. The significant challenges for Online Streaming Feature Selection (OSFS) include 1) the continuous development of feature quantities over time, 2) a large attribute space, probably of unidentified or infinite size, and 3) the unavailability of the whole feature set prior to learning starts.

In [4] Graph matching plays an essential role in numerous real applications. In this paper, authors studied how to equivalent two large graphs by maximizing the quantity of matched edges, which is recognized as maximum frequent subgraph matching and is NP-hard. To discover correct matching, it cannot a graph with extra than 30 nodes. To locate an approximate matching, the value can be very poor. The authors proposed a novel two-step approach that can resourcefully match two large graphs above thousands of nodes with high identical quality. In the first step, they proposed an anchor-selection/expansion method to calculate a good preliminary matching. In the second step, they proposed a new approach to process the initial matching.

In [5] authors considered proposed a novel online feature selection framework for applications with streaming features where the information of the occupied feature space is unidentified in advance. To describe streaming features as features that current in one by one over time whereas the quantity of training examples remains permanent. This is in difference with conventional online learning methods that simply compact with consecutively added observations, with little consideration being paid to streaming features. The significant challenges for Online Streaming Feature Selection (OSFS) include 1) the permanent growth of feature volumes over time, 2) a large feature space, probably of unidentified or infinite dimension, and 3) the unavailability of the complete feature set before learning starts. In the paper, authors presented a novel Online Streaming Feature Selection method to choose powerfully relevant and non-redundant features on the fly. A resourceful Fast-OSFS algorithm is proposed to progress feature selection performance. The proposed algorithms are assessed comprehensively on high-dimensional datasets and also with a real-world case study on impact crater detection.

In [6] authors proposed Data stream classification creates numerous challenges to the data mining community. In this paper, authors addressed four such main challenges, namely, unlimited length, concept-drift,

concept-evolution, and feature-evolution. Because a data stream is tentatively infinite in duration, it is unreasonable to accumulate and use all the past data for training. Concept-drift is a frequent phenomenon in data streams, which happens as a consequence of changes in the underlying concepts. Concept-evolution happens as a outcome of new classes developing in the stream. Feature-evolution is a commonly occurring process in several streams, such as text streams, in which novel features shows as the stream progresses.

III. PROPOSED METHODOLOGY

In this paper, the proposed methodology recognizes the Adaptive Graph time-variant classification (AGTVC) method uses the MATLAB framework to establish the competence of the graph learning algorithm is applied in the MemeTracker dataset. This proposed framework is illustrated in figure 1.1.

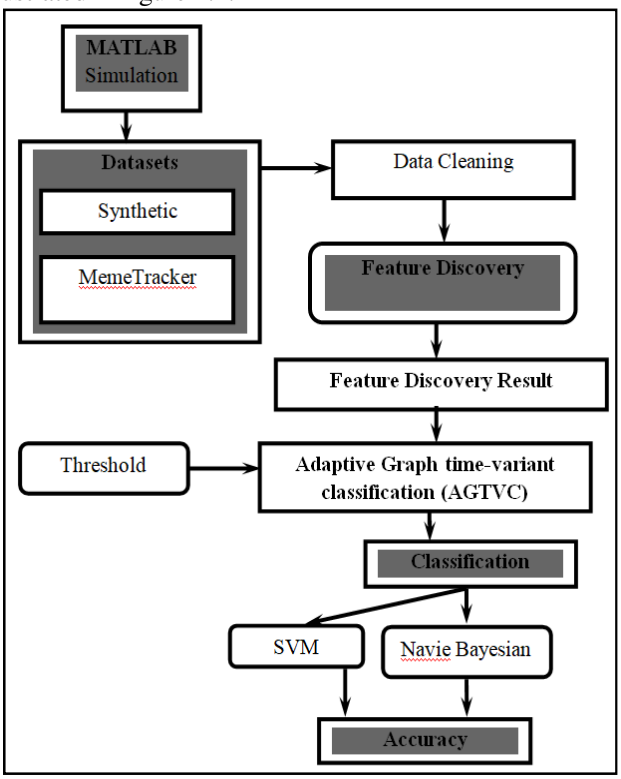

Figure 1.1: Adaptive Graph learning classification

A. *Data Cleaning*

The need for data preprocessing can be seen from the fact that redundant data and insignificant features may often confuse the classification algorithm, leading to the discovery of inaccurate or ineffective knowledge. Moreover, the processing time will increase when all features are used.

Finally, preprocessing helps to remove the redundant data, incomplete data and transforms the data into a uniform format.

The preprocessing module of the proposed system performs the following functionalities:

- Performs redundancy check and handles null values
- Converts categorical data to numerical data

In this proposed methodology the data cleaning procedure on synthetic and MemeTracker data set. For each corpus we performed the following normalization operations:

- Taking posts during only one month;
- Filtering out of non-English posts;
- Removal of empty words with empty words list
- Stemming using Porter stemming;
- Filtering out of words appearing less than five times

The above data cleaning then yields a standard word vector for each post. The vector for a cascade is then computed by averaging the vectors of all the posts that compose a cascade. The profile of each user is computed by averaging the vectors of the cascades diffused by the user on the training set. In order to evaluate the different models, we use a 5-fold cross validation scheme (4 blocks for training, one for testing). Training blocks are used to estimate models parameters and the last one is used for the evaluation. Noise and the amount of data are reduced and extra information is added during preprocessing to the data. The data is transformed into the occurrence domain, where the noise is reduced. Data cleaning schedules effort to clean data by satisfying in smoothing noisy data, missing values, recognizing or removing anomalies, and resolving unpredictability.

B. *Feature Selection Process*

A feature selection process is can be divided into four basic disjoint subsets - (1) irrelevant features, (2) redundant feature, (3) weakly relevant but non-redundant features, and (4) strongly relevant features. An optimal feature selection algorithm should non-redundant and strongly relevant features. For streaming features, it is difficult to find all strongly relevant and non-redundant features. A roughset based online streaming feature selection algorithm ROSFS finds an optimal subset using

a two-phase scheme - online relevance analysis and redundancy analysis.

A general framework of ROSFS is presented as follows:

Algorithm1: Feature selection of ROSFS

Input: collection of input features.
Output: trained features
Process
Step 1: Initialize best candidate features = ϕ
Step 2: Generate a new feature f_n
Step 3: ROSFS analysis
Ignore f_n, **if** f_n is irrelevant to the class labels
Finest candidate feature = Fest candidate feature $\cup f_n$
else
Finest candidate feature = Fest candidate feature $\cap f_n$
End if
Step 4: Online Redundancy Analysis
Step 5: Alternate Step 1 to Step 3 until the stopping criteria are satisfied.

In the feature selection analysis phase, ROSFS discovers strongly and weakly relevant features, and adds them into finest candidate features (FCF). If a new coming feature is irrelevant to the class label, it is discarded; otherwise it is added to FCF. In the redundancy analysis, ROSFS dynamically eliminates redundant features in the selected subset. For each feature fn in FCF, if there exists a subset within FCF making fn and the class label conditionally independent, fn is removed from FCF. An alternative way to improve the efficiency is to further divide this phase into two analysis - inner-redundancy analysis and outer-redundancy analysis. In the inner-redundancy analysis, ROSFS only re-examines the feature newly added into FCF, while the outer-redundancy analysis re-examines each feature of FCF only when the process of generating a feature is stopped.

C. Adaptive Graph Time-Variant Classification (AGTVC)

The adaptive graph time-variant classification performs sub-time-variant graphs are potential graph-outline candidates. The maximum similarity of a candidate to all training sequences can be used for graph prediction by ranking the similarity of a candidate (a testing time-variant graph). Therefore, in order to find graph-outline patterns, the algorithm needs to generate potential sub-time-variant graphs as candidates and then find the one, which has maximum similarity (other subsequences from the candidate sub-time-variant graphs), as graph-shapelet patterns

To calculate graph-outline patterns from sub-time-variant graphs, time-variant graphs should be converted to transformation sequences so that we can use edit similarity as the measure. The transformation sequences between two graphs in a given time-variant graph can be represented by a series of operations, e.g., node and edge insertions and deletions based on policy 1 and policy 2.

Algorithm 2 shows the detailed procedures for finding sub-time-variant graph candidates from time-variant graphs. Given a time-variant graph database G along with user-defined graph-outline length l. Line 1 converts time-variant graphs G to time series D based on the number of nodes and edges in each graph. After that, the algorithm generates all candidates within a sliding window (lines 3-6). For each outline candidate, the algorithm calculates the Euclidean distance between the candidate and other time series and records the locations with minimum distance in each time series (lines 7-8). Meanwhile, the algorithm updates the distance threshold and sub-time-variant graph locations, if any candidate has a smaller distance than the current distance threshold. Finally, the algorithm returns the sub-time-variant graphs and labels in each time-variant graph based on the sub-time-variant graph locations (lines 9-12).

Algorithm 2: AGTVC: Generate graph sub-time-variant candidates
Input: *G*: A set of time-variant graphs; *l*: Length of graph-shapelet patterns;
Result: *G´*: Graph subsequence candidates;
Process
Step 1: *D* ← Apply *G* to obtain corresponding time series *D*;
Step 2: *cSet* ← ∅; // Time series candidates with length l
Step 3: for *k* in *D* **do**
Step 4: Gp_k^l← Extract graph time series subsequence with length *l* from *k*
Step 5: *cSet* ← cSet ∪ Gp^l_k;
Step 6: end for
Step 7: Outline ← Apply *cSet* to *D* based on graph edge mining.
Step 8: Save the graph outline edge locations in each time series
Step 9: for each g_i in *G* **do**
Step 10: g_i´← Apply recoded graph outline edge location to the related sub-time-variant graphs in g_i;
Step 11: *G´* = *G´* ∪ g_i
Step 12: end for
Step 13: return *G´*

IV. PERFORMANCE EVALUATION

In this paper was carried on the synthetic and MemeTracker dataset and their performance was evaluated. The purpose of the experiments is to: 1) behavior a parameter study for the purpose of choosing optimal parameter settings for the experiments; 2) compare the proposed algorithms with benchmark methods to validate the performance of our method; and 3) test our method on real-world social network applications. All experiments are tested on a Windows 10 operating system with 3.2 GHz CPU and 8 GB memory.

Prediction Accuracy: The five algorithms are evaluated under different support thresholds (30, 50, 70 and 90) in the *AGTVC* subgraph mining. The settings for the time stamps are 3.61, 3.62, 3.63 and 3:64 $*10^5$ seconds. The experimental results report the performance of precision and recall measures of Off-line Subgraph Feature selection (OSF), Incremental Subgraph Feature (ISF), incremental subgraph join feature selection algorithm (*ISJF*), *AGTVC* and three benchmarks on the realworld data set in Figs. 1.2 and 1.3 listed in Table 1.

PRECISION: Precision is a technique to recover samples that are relavant which is based on accepting the similarity. While the precision significance is high then the algorithm outcome in a more relavant results.

$$precision = \frac{tp}{tp + fp} \quad eqn.(1)$$

This is also known as completely predictive value. The graph given below is an evaluate for the MemeTracker dataset taken as input. The precision for a class is the amount of true positives (i.e. the amount of items properly labeled as fit in to the positive class) separated by the whole number of elements labeled as fit in to the positive class (i.e. the amount of true positives and false positive, which are items wrongly labeled as belonging to the class).

RECALL: Recall technique is used as a significant instance that are recovered which is also based on accepting and relavance. At this time if the recall value is larger than the algorithim returns a relavant results.

$$Recall = \frac{tp}{tp + fn} \quad eqn.(2)$$

Recall is also assumed to be sensitivity value. The graph determines of the input dataset. Recall in this framework is defined as the amount of true positives separated by the total amount of essentials that essentially belong to the positive class (i.e. the amount of true positives and false negative, which are items which were not labeled as fit in to the positive class but must have been).

Table 1: Precision Score under the Parameter Support = 30 on the MemeTracker Data Sets

Methods	3.61	3.612	3.62	3.63	3.64
OSF	0.7	0.75	0.83	0.86	0.89
ISF	0.75	0.77	0.79	0.8	0.82
ISJF	0.72	0.75	0.82	0.83	0.91
AGTVC	**0.79**	**0.80**	**0.88**	**0.92**	**0.93**

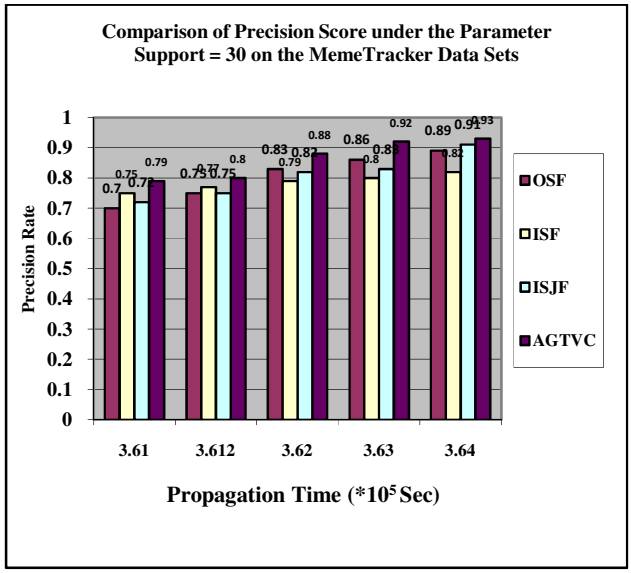

Fig. 1.2: Precision comparison under Support = 30

Table 2: Recall Score under the Parameter Support = 30 on the MemeTracker Data Sets

Methods	3.61	3.612	3.62	3.63	3.64
OSF	0.86	0.87	0.89	0.893	0.92
ISF	0.89	0.91	0.93	0.932	0.93
ISJF	0.90	0.92	0.93	0.94	0.95
AGTVC	**0.92**	**0.93**	**0.95**	**0.96**	**0.98**

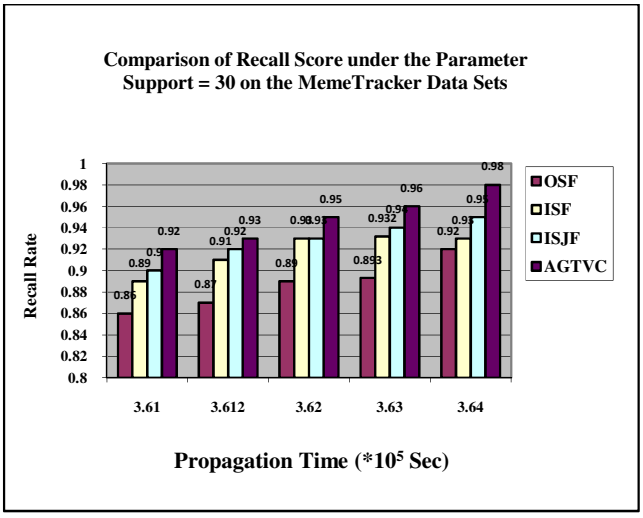

Fig. 1.3: Recall comparison under Support = 30

V. CONCLSUION

In this paper, presents an Adaptive Graph Time-Variant Classification (AGTVC) with roughset based online streaming feature selection algorithm (ROSFS). Based on the examination that ROSFS features selection follow the sliding closure property and long-pattern roughset features are frequently buried below short-pattern subgraph features, the research work proposed a novel Adaptive Graph Time-Variant Classification (AGTVC) for mining incremental subgraph features, and a subgraph join feature selection algorithm (ISJF) to exact long-pattern subgraphs. The proposed AGTVC algorithm divide feature selection and load into memory in a mini-batch manner, which is a important reduction in memory and running time. Experiments results shows the AGTVC algorithms can decrease both the processing time and memory cost. The ROSFS algorithm learns both feature selection and possible results to converts a high dimensional problem into a big constraint problem with respect to feature vector. AGTVC adds a new constraint on the graph time-variant classifier and forces the classifier to select long pattern subgraphs by joining short-pattern subgraph features.

REFERENCES

[1] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A framework for on demand classification of evolving data streams," IEEE Trans. Knowl. Data Eng., vol. 18, no. 5, pp. 577–589, May 2006.

[2] X. Yan, F. Zhu, P. S. Yu, and J. Han, "Feature-based similarity search in graph structures," ACM Trans. Database Syst., vol. 31, no. 4, pp. 1418–1453, 2006

[3] X. Wu, K. Yu, H. Wang, and W. Ding, "Online streaming feature selection," presented at the 27th Int. Conf. Mach. Learn., Haifa, Israel, 2010.

[4] Y. Zhu, L. Qin, J. X. Yu, Y. Ke, and X. Lin, "High efficiency and quality: Large graphs matching," in Proc. 20th ACM Int. Conf. Inf. Knowl. Manage., 2011, pp. 1755–1764.

[5] X. Wu, K. Yu, W. Ding, H. Wang, and X. Zhu, "Online feature selection with streaming features," IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 5, pp. 1178–1192, May 2013.

[6] M. M. Masud, et al., "Classification and adaptive novel class detection of feature-evolving data streams," IEEE Trans. Knowl. Data Eng., vol. 25, no. 7, pp. 1484–1497, Jul. 2013.

[7] M. Tan, I. W. Tsang, and L. Wang, "Towards ultrahigh dimensional feature selection for big data," J. Mach. Learn. Res., vol. 15, pp. 1371–1429, 2014.

[8] M. Takac, A. Bijral, P. Richtarik, and N. Srebro, "Mini-batch primal and dual methods for SVMs," in Proc. 30th Int. Conf. Mach. Learn., 2013, pp. 2059–2067.

[9] K. Riesen and H. Bunke, Graph Classification and Clustering Based on Vector Space Embedding. River Edge, NJ, USA: World Sci. Publishing Company, 2010.

[10] H. Fei and J. Huan, "Structured sparse boosting for graph classification," ACMTrans. Knowl. Discovery Data, vol. 9, 2014, Art. no. 4.

[11] X. Kong and P. S. Yu, "Semi-supervised feature selection for graph classification," in Proc. 16th ACM SIGKDD Conf. Knowl. Discovery Data Mining, 2010, pp. 793–802.

[12] S. Pan, J. Wu, and X. Zhu, "CogBoost: Boosting for fast cost-sensitive graph classification," IEEE Trans. Knowl. Data Eng., vol. 27, no. 11, pp. 2933–2946, Nov. 2015.

[13] S. Zhang, X. Luo, J. Xuan, X. Chen, and W. Xu, "Discovering small-world in association link networks for association learning," World Wide Web, vol. 17, no. 2, pp. 229–254, 2014.

[14] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. Classification and Regression Trees. Wadsworth International Group, 1994.

[15] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 2(2):121–167, June 1998.