GA-TVRC: A Novel Relational Time Varying Classifier to Extract Temporal Information Using Genetic Algorithms

İsmail Güneş, Zehra Çataltepe, Şule Gündüz Öğüdücü
{gunesi,cataltepe,sgunduz}@itu.edu.tr

Computer Engineering Department, Istanbul Technical University, Istanbul, Turkey

Abstract. Almost all networks in real world evolve over time, and analysis of these temporal changes may help in understanding or explanation of some properties or processes of a network. This paper presents GA-TVRC, a novel Relational Time Varying Classifier which uses Genetic Algorithms to extract temporal information. GA-TVRC uses Evolutionary Strategies to optimize the influence of each previous time period on classification of new nodes. A Relational Bayesian Classifier (RBC) that is proposed by Neville et.al. [3] is utilized to compute the fitness function. The performance of GA-TVRC is compared with both the RBC, which ignores the time effect and the time varying relational classifier (TVRC) that is proposed by Sharan and Neville [20]. TVRC improves the RBC by taking the time effect into account using different predetermined weights. According to the experiments on two real world datasets, GA-TVRC extracts time effect better than the previous methods and improves the classification performance by up to 5% compared to TVRC and up to 10% compared to RBC.

Keywords: Relational Bayesian Classifier, Time-Varying Relational Classifier, Evolving Networks, Genetic Algorithms, Evolutionary Strategies

1 Introduction

Real world networks often have complex network structures with interacting, multi-type components and these interactions and components may change over time. Examples of such evolving networks include world wide web network, many types of biological networks, movie-actor networks, scientific collaboration networks, power grids, and telephone call networks [1, 28].

The change on the network involves critical information about the network and should not be ignored. A change on the network may occur in three different ways:

- Addition of a new node or removal of a node
- Addition or removal of an edge
- Change of the node attributes

These types of changes in a network may provide additional information about the network. For instance, two objects which interact during a long time period are likely

to have a strong relationship. Similarly, more recent relationships among objects are probably more important than relationships that occurred in the distant past. The patterns which can not be extracted from only a single snapshot of the network may be revealed using the network snapshots for a long period of time. Hence, it is inevitable to utilize the time information for a more accurate analysis of the network. Usually, only additions or removals of edges between a fixed set of nodes are handled by the methods which deal with evolving networks. In addition to change of edges, our method benefits from both the changes of nodes and the changes of attributes to improve the classification accuracy.

Although there have been many efforts to understand the growth and dynamics of networks [1], [2], there aren't many studies that explore the evolution of the network for different purposes such as classification. To be able to make a classification on an evolving network, relations between nodes and the evolution of these relations should be taken into account. The best way of representing these relations is utilizing the relational data which involves information about the associated objects in addition to the object to be classified. Classification on heterogeneous relational data requires additional effort and differs from classification on homogeneous network data in which all objects have the same number of attributes and all attributes have the same number of values. In this paper, the classification on the relational data will be optimized by analyzing the evolution of the network.

In this paper, a classifier called Genetic Algorithm enhanced Time Varying Relational Classifier (GA-TVRC) is proposed in order to improve the classification performance in evolving complex networks. A detailed analysis is performed to understand the influence of heterogenous, time varying data on classification. It is assumed that interactions in each time period in the past have different influence on the classification accuracy on the present network. Therefore, we focus on extracting the influence of different time periods on classification in a network. For this task, a novel and fairly general genetic algorithm based framework is proposed. Genetic Algorithms are employed to optimize the influence values of different time periods. Experimental results on two real datasets show that the usage of influence values, computed by our method for different time periods, improves the classification accuracy. The proposed framework is general in that it may be used to improve the performance of any classification method. In this study, a relational classifier, the Relational Bayesian Classifier (RBC) [3], is chosen to be improved by adding time effect. The RBC is an extension of the simple Bayesian classifier that can be applied on relational data. The relational classification is likely to perform better than the traditional classification because not only the objects of the same type but also related objects of different types contribute to the classification task.

We compare our method with RBC [3] which does not take the time effect into account and TVRC [20] which is a time varying relational classifier that extends RBC. We also use a baseline method which utilizes random time influence values for comparison. Experimental results show that GA-TVRC outperforms all these methods by using the most accurate temporal information in the network and provides higher classification performance than other methods.

The rest of the paper is organized as follows. Section 2 describes the background information which provides the basis for the proposed method. In Section 3, related work on evolving networks and relational classification is given. The details of our algorithm is explained in Section 4. Section 5 gives the experiments and analysis of the results, while Section 6 concludes with comments and future work.

2 Background

In this section, we first provide the techniques we use for relational classification. Then, the basic information about the genetic algorithm is given to provide the required background.

2.1 Classification Method

We represent our relational dataset as an attributed multi-graph $G = (V, E)$ where V is the set of nodes of different types and E is the set of edges between them. The main purpose of a relational classifier is the classification of a node of given type in V. Among relational classifiers, RBC applies the simple Bayesian classifier to relational data with some estimation techniques. Among these techniques, the independent value estimator (INDEPVAL) which assumes each attribute value to be independently drawn from the same distribution achieves the best results. Using this assumption, attributes with different numbers of values can be handled. RBC estimates the conditional probability of a node *i* belonging to a class *C* given the attributes *X* and related nodes *R*:

$$P(C^i|X,R) \propto \prod_{X_m \in X^{G(i)}} P({X_m}^i|C).\prod_{j \in R}\prod_{X_k \in X^{G(j)}} P({X_k}^j|C).P(C) \qquad (1)$$

The related nodes are the objects which are connected to node *i* directly or through some other nodes. The attributes are the attributes of both node *i* to be classified and other related nodes. In order to calculate conditional probability of a class label ($P(C|X,R)$), conditional probabilities of the attributes (($P(X|C)$) are used. The conditional probabilities given a class label is calculated for each attribute of the related nodes. There are two types of nodes: *G(i)* denotes the node type to be classified and *G(j)* denotes the types of other nodes. X_m and X_k are the attributes of these node types, respectively. In the first part of multiplication, the conditional probabilities for attribute values of the node of type *G(i)* are calculated. Then, the conditional probabilites for attribute values of the node of type *G(i)* are computed in the second part of the multiplication. Finally, these conditional probabilities are also multiplied with class label probability *P(C)*. Although, for normalization, the formula should be divided with the attribute value probabilities *P(X)*, since the INDEPVAL assumes that each attribute is independently drawn from the same distribution, this denominator can be ignored.

TVRC extends RBC by including the time effect as follows [20] :

$$P(C^i|X,R) \propto \prod_{X_m \in X^{G(i)}} P({X_m}^i|C).\prod_{j \in R}\prod_{X_k \in X^{G(j)}} {w_{ij}}^t P({X_k}^j|C).P(C) \qquad (2)$$

The only difference between RBC and TVRC is the usage of weight value w_{ij}^t. This weight value is determined according to the relation between node *i* and related nodes *R*. The conditional probabilities for attribute values of a related node are multiplied with the weight value of the relation. In TVRC, in order to give more emphasis to more recent relations, the weight values decay in time according to a pre-defined function.

In our work, the weight values are determined using the Genetic Algorithms. Please see Section 4 for details of our method, Genetic Algorithm enhanced Time Varying Relational Classifier (GA-TVRC).

2.2 Genetic Algorithms

Genetic algorithms were first proposed by Holland [4] as an optimization method. They are especially suitable when the solution space of a problem is very large and an exhaustive search for the solution is impractical. In a genetic algorithm, potential solutions should be shown in a suitable representation. Each possible solution is represented in a data structure, which is called an individual and the algorithm tries to find the best fitting solution. In order to improve the quality of a solution, the algorithm uses genetic operations on individuals for some number of iterations.Genetic Algorithms provide better results than traditional optimization algorithms do because they are less likely to get stuck into a local maxima.

The main operations of the genetic algorithm are:

- Reproduction: Passing a candidate solution to the next generation.
- Mutation: Changing genes of an individual.
- Cross-over: Swapping the genes of any two individuals.

After applying these operations to an individual which is a candidate solution, the fitness function is used to evaluate how good a solution is. The solutions are chosen statistically according to their fitness values. This behavior is similar to the real world so that it is likely but not guaranteed for a strong individual to survive.

Genetic algorithms can be used not only for binary genes but also for real numbers. These kinds of numerical optimization problems are solved using Evolutionary Strategies [5]. The Evolutionary Strategies are similar to Genetic Algorithms and also based on adaptation and evolution. Here, they are used to optimize the influence of interaction time on classification.

3 Related Work

Although the usage of network information for classification has been getting more common, the change in network structure has mostly been ignored during classification process. This maybe partly due to the reason that weighting and combining the network data on different time periods, in the best way for the specific problem is difficult. The change of the network has been analyzed for different purposes and using different methods [6]. For example, in link prediction, there are methods [7–9] that use the change information to better predict the new links. In addition to link prediction, clustering results have also been improved using the change of the network [10–13].

Another work attempts to model the evolution of social networks mathematically and emphasizes the order of interactions instead of summarizing the static networks [14]. The effects of particular network properties on network evolution was examined in [15].

Using the time information was shown to improve relational classification by some previous methods. Time Varying Classifier (TVRC) [19, 20] was shown to improve the RBC [3, 16, 17] classifier. RBC primarily makes use of the related nodes as well as the node to be classified. TVRC merges nodes and edges from different time periods and generates a summary graph. Then RBC is applied with the weight values. TVRC utilizes kernel functions for merging the weight values from different snapshots, each of which represents the network in a given time period. The most accurate results are obtained using decaying kernel functions, i.e. the information from the snapshots taken in the recent past have more influence on classification. While this strategy was shown to improve accuracies obtained, TVRC can not handle more complicated and uneven cases of interaction. For instance, co-authorship of two scientists in $t - 2\Delta t$ may be more important than the co-authorship of these two in $t - \Delta t$. Besides, it is impossible to assign optimum importance values for snapshots using this method even if they are really all decaying, because the actual decay function may be different for different kinds of networks. Hence, determining the optimum values representing the effect of the past network data could produce more accurate classification results than time-decaying kernels. In this work, evolutionary strategies are used to determine the optimum weights for each network snapshot from different past time periods.

4 Genetic Algorithm Enhanced Time Varying Relational Classifier (GA-TVRC)

Before the GA-TVRC algorithm starts, the network data need to be prepared to be used as input to a time varying classifier. The network data are represented as distinct sub-graphs. Each sub-graph includes the node to be classified and other related nodes which have interacted with the node in previous time periods. After gathering the input in the required form, the method is initialized.

The overall framework of the proposed time varying relational classifier method is shown in Figure 1. The framework consists of three phases: training phase, validation phase and test phase. The classifier is learned in training phase of the framework. In validation phase, the influence of temporal data on classification is determined using Evolutionary Strategies. This phase is the main focus of our work. The contribution of this validation phase on classification accuracy is evaluated in the test phase.

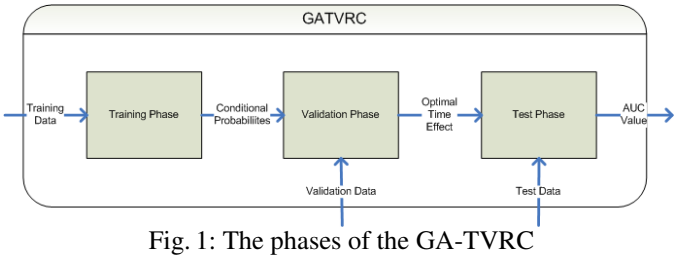


Fig. 1: The phases of the GA-TVRC

4.1 Training Phase

The inputs of this phase are training data as distinct sub-graphs. Each sub-graph includes the node to be classified and other related nodes. In this phase, the probabilities which will be used to compute the result in Equation 1 are calculated from training data. The conditional probabilites of each attribute given a class label $(P(X|C))$ and the probabilities of each class label $((P(C))$ are estimated using the training data. The ouputs of the phase are these probabilities which will be used in validation phase.

4.2 Validation Phase Using Evolutionary Strategies

In validation phase, the probabilities which are calculated in training phase are taken as inputs and applied on validation data. Learned probabilities are used to classify nodes in new sub-graphs using Equation 3. Time influence on classification is adjusted based on the classification performance. The outputs of the validation phase are the optimal time influence values which will be used in the test phase.

In this phase, Evolutionary Strategies are used for validating the proposed method by adjusting the influence of each interaction time on classification. The classification method is applied many times using Equation 3 with different time influence values in this phase and the best combination is sought.

We add the time influence into RBC equation, but unlike TVRC, the time influence is optimized by evolutionary strategies instead of using a kernel function. The nodes to be classified are evaluated according to:

$$P(C^i|X,R) \propto \prod_{X_m \in X^{G(i)}} P({X_m}^i|C).\prod_{j \in R}\prod_{X_k \in X^{G(j)}} {c_{ij}}^t P({X_k}^j|C).P(C). \quad (3)$$

Here, c_{ij}^t indicates the *influence value* of the time period in which the interaction between the node *i* and another related node *j* occurred. Our algorithm explores the optimal influence values of each time period using the Evolutionary Strategies. The steps to obtain influence values are as follows:

- An initial random population is generated. An individual represents the influence values for different time periods. Initially, each individual has random values.
- Following steps are applied until the algorithm terminates:
  - The classifier is realized for actual valuess of each individual and the area under the ROC curve (AUC) is obtained as the fitness value.
  - The genetic algorithm operations are applied to the individuals and new individuals are created.
  - New population is constructed by selecting fittest individuals among all individuals
- The individual with the highest fitness value is presented as the solution.

Representation Each individual is a solution candidate and it is an array of influence values of each time period. Each gene location on an individual represents the influence value of a time period. Here, we prefer to use individuals with 5 genes. The genes can take on real values showing the effect of each interaction time on classification.

Mutation Mutation is performed by adding a random value to a gene of an individual. The values to be added are drawn from the normal distribution $N(\xi, \sigma)$ where the mean ξ is set to 0 and the variation σ is called mutation step size. The mutation step size (variation) σ is also updated after each iteration according to the mutation success rate p_s [18]:

$$\sigma = \begin{cases} \sigma/c & if\, p_s > 1/5 \\ \sigma.c & if\, p_s < 1/5 \\ \sigma & if\, p_s = 1/5 \end{cases}$$

p_s is computed as the ratio of successful mutations and *c* is a coefficient that is generally set between 0.85 and 1 [18]. In our experiments the c value was set to be 0.9. The vector values are changed by adding random noise drawn from the normal distribution: $x_i' = x_i + N(0, \sigma)$.

The mutation step size is evolved so that the search space's traversal can be adjusted according to the mutation performance. The step size gets bigger by successful mutations and therefore the diversity in Genetic Algorithm increases.

Cross-over Two children are created by cross-over as in the traditional cross-over mechanism. The cross-over is applied to one gene in a random position from each parent and these genes are switched in order to form new children. All individuals are exposed to the mutation operator and both the individual before the mutation and the individual after the mutation are preserved in the population. Then, the cross-over operation is applied to individuals which are selected uniformly and new individuals are created. After mutation and cross-over operations, the fittest individuals are selected as the survivors and they constitute the new population.

Fitness The fitness to be used is the AUC value which results after each classification. AUC denotes the area under the ROC (Receiver Operating Characteristics) curve. The ROC curve is the plot of the true positive rate versus false positive rate for a binary classifier. It is formally proven that AUC is statistically consistent and more discriminating than accuracy while evaluating learning algorithms [29].

A fitness value is computed for each individual representing a solution for time influences. This fitness value is used to optimize the solution. The usage of AUC as a fitness function provides the selection of best coefficients giving the unbiased and optimum solution. Besides, there is no need for a normalization of AUC value because the value of AUC is between 0 and 1 by its nature just like the fitness value. There are many examples of different aspects using the AUC as a fitness function [21], [22], [23].

4.3 Test Phase

The inputs of test phase are the optimal time influence values from validation phase. These optimal values and the probabilites which are learned in training phase are applied to new nodes in test set using Equation 3. The output of test set is the AUC value after classification of test set.

After computing the optimal influence values in validation phase, the values are used to classify further nodes and it is checked whether using the computed values

improves the classification performance or not. The test phase requires the computation of conditional probabilities of belonging to each class label for a node and comparison of them to determine true class label of the node. Here, the time of each past interaction for this node and their influences on classification are also taken into account to compute these probabilities for classification.

5 Experimental Results

In our experiments, the main goal was evaluation of the classification performance of the proposed method GA-TVRC. The performance of four different methods were compared. The first method to be tested was Relational Bayesian Classifier (RBC) and the second one was Time Varying Relational Classifier (TVRC) that utilizes the time information in relational classification. Then, our method GA-TVRC that uncovers the optimal influence values, was tested. In addition to these methods, a new method Random Time Varying Relational Classifier (R-TVRC) in which the effects of each time period are set randomly was also tested. The aim of the tests with R-TVRC was assessing the contribution of other methods compare to random time influence values.

Another goal of the experiments was extraction of the pattern behind the effect of different time periods on classification, if there is one. This pattern can be used to understand the real time effect on classification.

5.1 Datasets

The experiments were performed on High-Energy Physics literature (HEP-Th) dataset and P2P file sharing (Can-o-sleep) dataset. Both datasets have been used in different studies and they also contain time information.

HEP-Th Dataset The HEP-Th dataset includes information on papers in theoretical high-energy physics. This dataset was provided for the 2003 KDD Cup competition [24].

The HEP-Th Database has 42319 nodes which are composed of 29555 papers, 9200 authors, 448 journals and 3116 e-mail domains. The edges which connect these nodes are 352,807 citations, 87794 co-authorships, 58,515 authorship, 20,816 publications and 12,487 e-mail affiliations. There are a total of 532,429 edges.

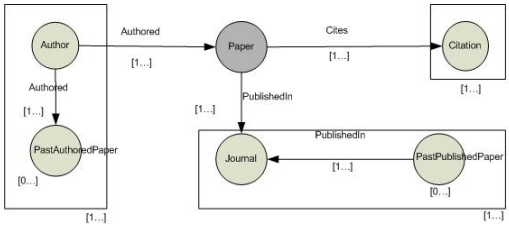


Fig. 2: The sub-graph structure of HEP-Th dataset

In Figure 2, the visual query which is prepared with QGraph [27] is shown. This query is used for acquiring the sub-graphs which satisfy the required conditions. In a sub-graph, there is not only the *Paper* node to be classified but also its related nodes of types *Journal*, *Citation*, *PastPublishedPaper* and *PastAuthoredPaper*. The *PastPublishedPaper* indicates the papers which have been published in the related journal before while the *PastAuthoredPaper* indicates the past papers of the related author. The usage of these types of nodes and the past interactions let us use the previous time information which they have. A sample sub-graph which results after this query is shown in Figure 3. This sample sub-graph is an example sub-graph which includes nodes of given types and their interactions between 1994 and 1998. Here, the classification task is to predict whether a paper is of the area *Quantum Algebra*.

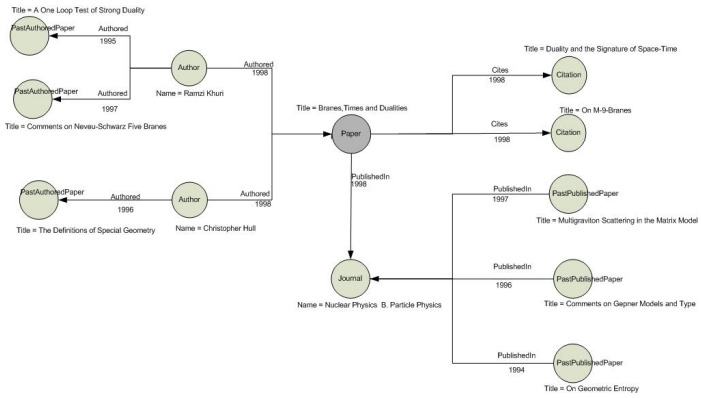


Fig. 3: A sample sub-graph from HEP-Th dataset

The Proximity HEP-Th database is based on data from the arXiv archive and the Stanford Linear Accelerator Center SPIRES-HEP database provided for the 2003 KDD Cup competition with additional preparation performed by the Knowledge Discovery Laboratory, University of Massachusetts Amherst.

Can-o-sleep Dataset The P2P file sharing dataset (Can-o-sleep) includes information on files which were transferred in a campus network through P2P file sharing server. The dataset contains mp3 files shared between users for 81 days in 2003. There are 563409 nodes which are composed of 291925 files, 6528 users, 221152 transfers, and 43804 queries. The edges which connect these nodes are ownership, transfer, making queries etc. and overall there are more than 6 million edges.

In Figure 4, the visual query for Can-o-sleep dataset is shown. In addition to *File* node to be classified there are also its related nodes of types *Query*, *Transfer*, *User*, *PastTransfer* and *UsersPastTransferFile*. The *UsersPastTransferFile* indicates other files which have been transferred in the past by related users. Here, the classification task is to predict whether a file will be transferred mored than 10 times a week.

The Proximity Can-o-sleep database is based on data collected by the Privacy, Internetworking, Security, and Mobile Systems Laboratory at the University of Mas-

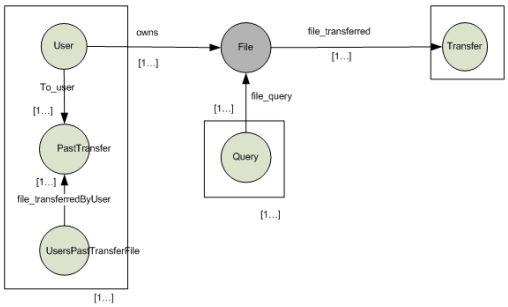


Fig. 4: The sub-graph structure of Can-o-sleep dataset

sachusetts Amherst with additional preparation by the Knowledge Discovery Laboratory, University of Massachusetts Amherst.

5.2 Methodology

The training and test sets are determined based on the sliding windows technique. That is, the time period which includes the train and test set for an experiment, is shifted in time to generate new train and test sets. The timeline including HEP-Th datasets which are used in our experiments is shown in Figure 5. The required information, such as conditional probabilities, are acquired on training set and the classifiers are tested on the test set. The validation set is only used by GA-TVRC in order to explore the time influence values. For instance, after learning conditional probabilities between 1994 and 1997, the papers published in 1998 and their past interactions are used for extracting the influence of each year between 1994 and 1998. After this validaton phase, the computed influence values are tested on the papers published in 1999. The sizes of each training, validation and test sets which were used in our experiments are given in Table 1.

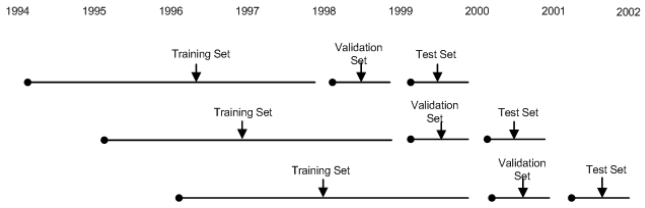


Fig. 5: The HEP-Th training, validation and test sets used in the experiments

Table 1: The sizes of HEP-Th training, validation and test sets in terms of number of sub-graphs

Period	Training Set Size	Validation Set Size	Test Set Size
$1994-1999$	3495	840	886
$1995-2000$	4220	864	1031
$1996-2001$	4638	999	1133

Timeline that denotes the training, validation and test sets for Can-o-sleep datasets is similar with HEP-Th timeline and it is shown in Figure 6. The sizes of each training, validation and test sets are given in Table 2.

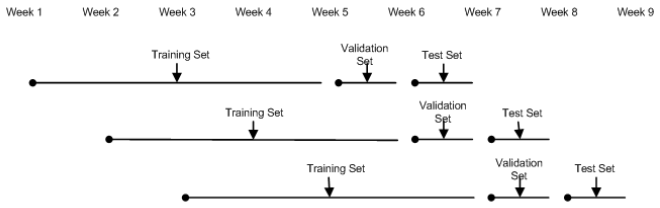


Fig. 6: The Can-o-sleep training, validation and test sets used in the experiments

Table 2: The sizes of Can-o-sleep training, validation and test sets in terms of number of sub-graphs

Period	Training Set Size	Validation Set Size	Test Set Size
$week1 - week6$	1587	952	1085
$week2 - week7$	1326	934	1102
$week3 - week8$	1435	1013	1816

The performances of RBC, TVRC, R-TVRC and GA-TVRC were compared in our experiments. Each method was applied on the same train and test sets. RBC was used with the estimator INDEPVAL which assumes that each value is independently drawn from the same distribution. It was experimentally shown that INDEPVAL provides the best results compared to other estimators in [3]. Similarly, TVRC was also used with the configuration that had been shown to give the best results. It was previously shown that, among the kernel functions which determine the influence of time period, decaying kernels provide best results [19]. The Exponential Kernel which decay the influence values exponentially is the best kernel function among the decaying kernels. Hence, Exponential Kernel was used with TVRC in our experiments.

In validation phase of GA-TVRC, Evolutionary Strategies have been run for 50 generations, each of which includes 20 individuals. The validation and test phases of GA-TVRC have been repeated 10 times and the performance of GA-TVRC has been determined by averaging the results of these runs. Similar with GA-TVRC, the performance of R-TVRC has been determined by averaging the consecutive 10 runs. The performance of each method have been quantified by evaluating the resulting AUC values.

5.3 Results and Analysis

The first set of experiments demonstrates the AUC values for RBC, TVRC, R-TVRC and GA-TVRC. The results of all methods are shown in Table 3 and Table 4. As it is seen from these tables, GA-TVRC outperforms other methods for all datasets and provides an improvement of 10% compared to RBC and 5% compared to TVRC. TVRC slightly achieves better than RBC by utilizing time effect. RBC algorithm which has equal unit influence values provides similar results with R-TVRC which has random values. Hence, it can be claimed that using decaying time influence values in TVRC increases the AUC values compared to RBC and R-TVRC but optimizing time influence in GA-TVRC gives the best results.

Extracting the trend of time influence on classification is one of the purposes of our work. The optimal influence values of each year which were computed by 10 distinct

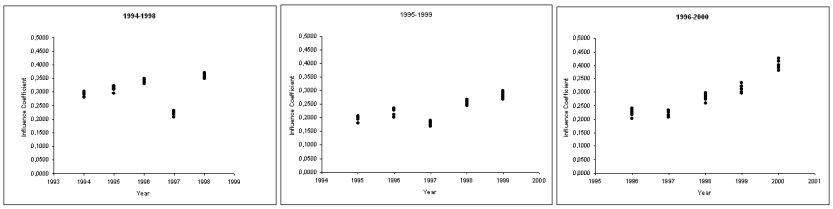
Table 3: The AUC values for each method on HEP-Th datasets

Test Set \ Method	RBC	TVRC	R-TVRC	GA-TVRC
1999	0.804	0.839	0.801	0.900
2000	0.722	0.749	0.733	0.786
2001	0.815	0.828	0.794	0.840

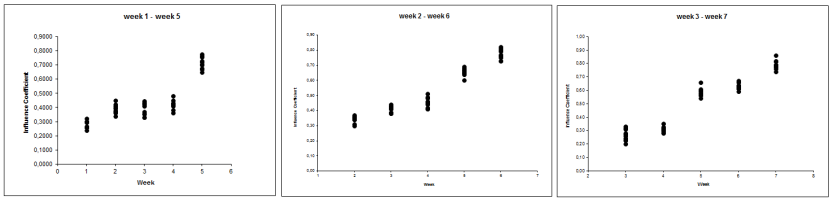
Table 4: The AUC values for each method on Can-o-sleep datasets

Test Set \ Method	RBC	TVRC	R-TVRC	GA-TVRC
week1	0.801	0.829	0.796	0.894
week2	0.769	0.797	0.753	0.841
week3	0.792	0.821	0.802	0.881

runs of the algorithm was given in Figure 7. We can say that the effect of an interaction time on classification increases with time and more recent interactions have more influence but there are also some exceptions. The effect of year 1997 in Figure 7(a),(b),(c) is relatively low in all three time periods and it is hard to discover this type of exceptions without using genetic algorithms.



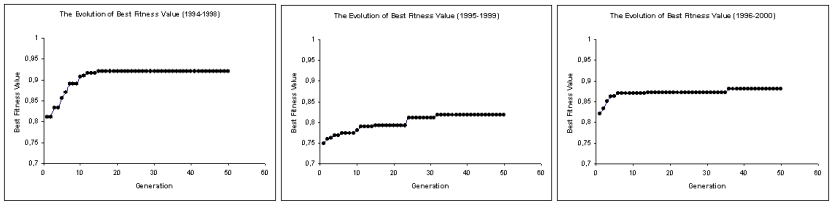
(a) HEP-Th 1994-1998 (b) HEP-Th 1995-1999 (c) HEP-Th 1996-2000



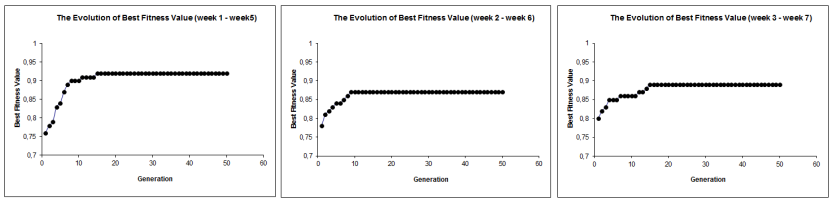
(d) P2P week1-week5 (e) P2P week2-week6 (f) P2P week3-week7

Fig. 7: The influences of each time period on classification

Another test set was performed to evaluate the progress of Genetic Algorithms. Figure 8 shows the best fitness values in each generation to check whether the fitness value converges to some fitness value or not. The figure indicates that the fitness value is likely to converge to it's maximum value before termination of the algorithm. In addition, change in mutation step size which was evolved in GA-TVRC algorithm is also examined. It is shown in Figure 9 that mutation step size increases in earlier generations after successful mutations. As the algorithm converges to solution, mutations start to get fail and the mutation step size decreases and converges to 0.

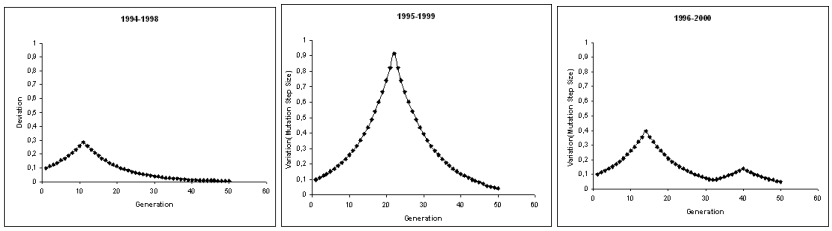


(a) HEP-Th 1994-1998 (b) HEP-Th 1995-1999 (c) HEP-Th 1996-2000

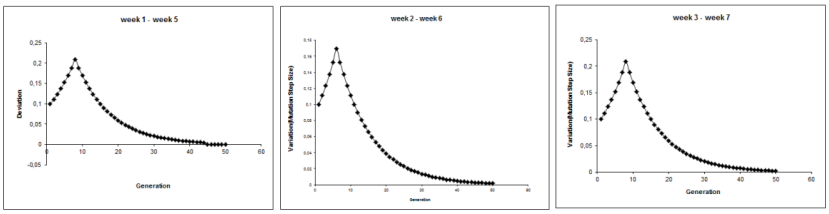


(d) P2P Week1-Week5 (e) P2P Week2-Week6 (f) P2P Week3-Week7

Fig. 8: The evolution of the best fitness value by the generations



(a) HEP-Th 1994-1998 (b) HEP-Th 1995-1999 (c) HEP-Th 1996-2000



(d) P2P Week1-Week5 (e) P2P Week2-Week6 (f) P2P Week3-Week7

Fig. 9: The evolution of the mutation step size by the generations

6 Conclusions and Future Work

In this paper, we have demonstrated a novel framework for finding patterns in the influences of interaction times within a complex network in order to enhance the ability of a classifier for the network. The proposed framework utilizes Genetic Algorithms for efficiently extracting the effects of time periods in which the interactions in a network occured. The optimal time effects which have been obtained by Evolutionary Strategies have been, in turn, used to improve the classification performance. As far as we know, Genetic Algorithms have never been used to explore the time influence on classi-

fication. We have shown that Genetic Algrithms may be used to improve classification performance. It has seen that the classification performance may be improved by this way. Besides, the analysis of evolution of time effect on the classification by the time is another contribution and it may inspire the future works which will use the time effect on classification.

Our method was evaluated on two real world networks. It was shown that, by using time influence values, which have been derived by Evolutionary Strategies, the classification performance was improved in terms of AUC. Experiments showed that GA-TVRC improves the classification performance of RBC by up to 10% with contribution of utilizing time influence. In addition to this improvement, GA-TVRBC outperforms TVRC which also utilizes temporal data according to a decaying exponential kernel strategy which was reported to perform best. The assumption of generally decaying time influence was mostly verified by results of our experiments. However, there are also some exceptions to this observation and there is no definite patterns on these coefficients. The lack of a pattern prevents using a mathematical expression and encourages use of Genetic Algorithms.

We are in the process of extending our proposed framework in several diretions. Each link type in a network may change in a different way and their effect should be computed distinctly for each link type to improve the classification performance. For example, the effect of a recent co-authorships may be more than the effect of a recent publication relation. We are also planning to experiment on different kinds of networks to better understand the time influence on classification.

References

1. Albert, R., Barabasi, A.L.: Statistical mechanics of complex networks. Reviews in Modern Physics, 74, 47-97 (2002)
2. Newman, M.E.J.: The structure and function of complex networks. SIAM Review, 45, 167-256 (2003)
3. Neville, J., Jensen, D., Gallagher, B., Fairgrieve, R.: Simple Estimators for Relational Bayesian Classifiers. In: Proceedings of the 3rd IEEE International Conference on Data Mining, pp. 609-612 (2003)
4. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, Michigan (1975)
5. Rechenberg, I.: Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution, Frommann-Holzboog, Stuttgart (1973)
6. Borgnat, P., Fleury, E., Guillaume, J., Robardet, C., Scherrer, A.: Proceedings of NATO Advanced Study Institute on Mining Massive Data Sets for Security, IOS Press (2008)
7. O'Madadhain, J., Hutchins, J., Smyth, P.: Prediction and Ranking Algorithms for Event-based Network Data. In: ACM SIGKDD Explorations Newsletter, vol. 7, Issue 2, pp. 2330 (2005)
8. Tylenda, T., Angelova, R., Bedathur, S.: Towards Time-aware Link Prediction in Evolving Social Networks. In: SNA-KDD '09: Proceedings of the 3rd Workshop on Social Network Mining and Analysis, pp. 1-10 (2009)
9. Hsu, W.H., Weninger, T., Paradesi, M.S.R.: Predicting Links and Link Change in Friends Networks: Supervised Time Series Learning with Imbalanced Data. In: Proceedings of the 18th International Conference on Artificial Neural Networks in Engineering (ANNIE 2008), St. Louis, MO (2008)

10. Gaertler, M., Grke, R., Wagner, D., Wagner, S.: How to Cluster Evolving Graphs. In: Proceedings of the European Conference of Complex Systems, ECCS06 (2006)
11. Palla, G., Barabasi, A. L., Vicsek, T.: Quantifying Social Group Evolution. Nature 446, 664-667 (2007)
12. Chakrabarti, D., Kumar, R., Tomkins, A.: Evolutionary Clustering. In: Proc. ACM SIGKDD 06, pp. 554-560 (2006)
13. Chan, S.Y., Hui, P., Xu, K.: Community Detection of Time-Varying Mobile Social Networks. In: Proc. of the First Int. Conf. on Complex Sciences: Theory and Applications, Complex 2009 (2009)
14. Berger-Wolf, T., Saia, J.: A Framework for Analysis of Dynamic Social Networks. DIMACS Technical Report, vol. 28 (2005)
15. Csardi, G., Strandburg, K.J., Zalanyi, L., Tobochnik, J., Erdi, P.: Estimating the Dynamics of Kernel-Based Evolving Networks. In: Proceeding of the sixth international Conference on complex Systems (2006)
16. Neville, J., Jensen, D., Friedland, L., Hay, M.: Learning Relational Probability Trees. In: KDD 03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 625630. ACM, New York (2003)
17. Neville, J., Jensen, D.: Dependency Networks for Relational Data. In: Proceedings of the 4th IEEE International Conference on Data Mining, pp. 170177 (2004)
18. Beyer, H., Schwefel, H.: Evolution Strategies: A Comprehensive Introduction. Natural Computing 1, no.1 (2002)
19. Sharan, U., Neville, J.: Temporal-Relational Classifiers for Prediction in Evolving Domains. In: ICDM 2008, pp. 540-549 (2008)
20. Sharan, U., Neville, J.: Exploiting Time-varying Relationships in Statistical Relational Models. In: WebKDD/SNA-KDD '07: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis, pp. 9-15. (2007)
21. Weninger, T., Hsu, W.H., Xia, J., Aljandal, W.: An Evolutionary Approach to Constructive Induction for Link Discovery. In: Genetic and Evolutionary Computation Conference (GECCO-2009), Montral, Canada (2009)
22. Segond, M., Fonlupt, C., Robilliard, D.: Genetic Programming for Protein Related Text Classification. In: Genetic and Evolutionary Computation Conference (GECCO-2009), pp. 1009-1106. (2009)
23. Shao, H., Zheng, G.: Construction of Bayesian Classifiers with GA for Response Modeling in Direct Marketing. In: 2nd IEEE International Conference on Computer Science and Information Technology, Beijing-China (2009)
24. Knowledge Discovery Laboratory Website, University of Massachusetts Amherst, Department of Computer Science, http://kdl.cs.umass.edu/data/hepth/hepth-info.html
25. Newman, M.E.J.: Scientific Collaboration Networks, I. Network Construction and Fundamental Results. Phys Rev E 64, 016131-1016131-8 (2001)
26. Newman, M.E.J.: Scientific Collaboration Networks, II. Shortest Paths, Weighted Networks, and Centrality. Phys Rev E 64, 016132-1016132-7 (2001)
27. Blau, H., Immerman, N., Jensen, D.: A Visual Query Language for Relational Knowledge Discovery. University of Massachusetts Amherst, Computer Science Department Technical Report 01-28 (2001)
28. Banerjee, A.: The Spectrum of the Graph Laplacian as a Tool for Analyzing Structure and Evolution of Networks. Ph.D. Thesis, University of Leipzig (2008)
29. Ling, C.X., Huang, J., Zhang, H.: AUC: a Statistically Consistent and more Discriminating Measure than Accuracy. In: Proceedings of the Eighteenth International Joint Conference of Artificial ntelligence (IJCAI) (2003)