Temporal-Relational Classifiers for Prediction in Evolving Domains

Umang Sharan
Purdue University
Department of Computer Science
usharan@cs.purdue.edu

Jennifer Neville
Purdue University
Departments of Computer Science and Statistics
neville@cs.purdue.edu

Abstract

Many relational domains contain temporal information and dynamics that are important to model (e.g., social networks, protein networks). However, past work in relational learning has focused primarily on modeling static "snapshots" of the data and has largely ignored the temporal dimension of these data. In this work, we extend relational techniques to temporally-evolving domains and outline a representational framework that is capable of modeling both temporal and relational dependencies in the data. We develop efficient learning and inference techniques within the framework by considering a restricted set of temporal-relational dependencies and using parameter-tying methods to generalize across relationships and entities. More specifically, we model dynamic relational data with a two-phase process, first summarizing the temporal-relational information with kernel smoothing, and then moderating attribute dependencies with the summarized relational information. We develop a number of novel temporal-relational models using the framework and then show that the current approaches to modeling static relational data are special cases within the framework. We compare the new models to the competing static *relational methods on three real-world datasets and show that the temporal-relational models consistently outperform the relational models that ignore temporal information—achieving significant reductions in error ranging from 15% to 70%.*

1 Introduction

Recent research has demonstrated the utility of modeling relational information for domains such as web analytics [5], marketing [8] and fraud detection [19]. This work has demonstrated that incorporating the characteristics of *related* instances into statistical models improves the accuracy of attribute predictions. However, this work has focused primarily on modeling *static* relational data. For the numerous relational domains that have temporal dynamics, researchers have generally analyzed static versions of the data, which comprise all the objects, links, and attributes that have occurred up to and including a specific time t. These approaches have ignored the temporal information present in the data even though, in many datasets, there are likely to be dependencies in the temporal-relational information that can be exploited to improve model performance. For example, in fraud detection a pair of individuals that communicate regularly over time should have a stronger relationship, and thus the attributes of the individuals are more likely to exhibit correlation, than a pair of individuals that communicate for a brief time period. To date, there are few available data mining tools that can exploit these types of temporal-relational dependencies.

Relational data can exhibit temporal dynamics in a number of ways. First, the instances in the data may appear and disappear over time. For example, web pages are created as web sites are developed, expanded, and modified over time. It may be important to model object creation times if recently added instances exhibit different characteristics than older instances (e.g., new vs. established accounts). Second, the links (or relations) in the data may represent events at a particular time. If this is the case, then the time associated with the event may be important to model (e.g., the time of a phone call). Third, the attribute values in the data may change over time. For example, a sensor may be recording the position of an object moving through a building and this may inform predictions about the properties of the object.

Recent work has only just begun to incorporate temporal information into statistical relational models. Some initial work has focused on transforming temporal-varying links and objects into static aggregated features [19] and other work has focused on modeling the temporal dynamics of time-varying attributes in static link structures [13]. There have been some recent efforts to model temporally-varying links to improve automatic discovery of relational communities or groups [4, 15] but this work has not attempted to exploit temporal link information in a classification context.

The goal of this work is to improve attribute prediction in dynamic domains by incorporating the influence of time-

U. Sharan and J. Neville. Temporal-Relational Classifiers for Prediction in Evolving Domains. In Proceedings of the 8th IEEE International Conference on Data Mining (ICDM'08), pp. 540-549, 2008.

varying links into statistical relational models. One motivation for modeling time-varying links is the identification of influential relationships in the data. Since much of the success of relational models is predicated on the correlation between attribute values of linked instances, a method that prunes away spurious relationships and highlights stronger relationships will lead to higher levels of correlation, and thus, more significant increases in model performance compared to methods for independent and identically-distributed (i.i.d.) data.

We conjecture that the temporal link information will be useful for disambiguating relationships in this fashion. In particular, we look for patterns of *temporal locality* and *temporal recurrence* to identify stronger relationships that are more likely to exhibit correlation among the associated attribute values. Temporal locality refers to the notion that events in the recent past are more influential than events in the distant past. Temporal recurrence refers to the notion that a regular series of events between two instances is more likely to indicate a stronger underlying relationship than an event isolated in time.

As illustration, consider the Cora database of computer science research papers extracted automatically from the web [18]. Each paper has an associated topic and citations to other papers that have been published in the past. Figure 1 shows the autocorrelation between the topics of papers published in the year 1996 with the topics of the papers they cite in the past. The x-axis represents the time interval between 1996 and the year of publication of the cited papers. Observe that the autocorrelation between topics decreases as the time lag increases. Thus, the topics of recent references are likely to be better indicators than the topics of references that were published farther in the past.

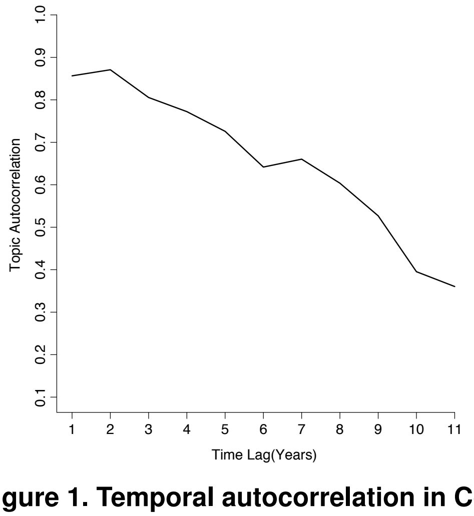

Figure 1. Temporal autocorrelation in Cora.

In this work, we propose the Time Varying Relational Classifier (TVRC) framework—a novel approach to incorporating temporal dependencies into statistical relational models. TVRC uses a two-step process that first transforms a dynamic relational graph into a static weighted summary graph using kernel smoothing. The second phase then incorporates the static link weights into a modified relational classifier to moderate the influence of attributes throughout the relational data graph. Our contributions are as follows:

- We propose a novel approach to temporal-relational classification for domains where link events occur over time. Our approach calculates a measure of relational influence for each pair of objects, based on their interactions over time, and then uses this measure to moderate the influence of associated attributes.

- We show that our framework subsumes current *static* relational modeling approaches. In particular, we show that two common *static* modeling approaches for relational data are special cases within in our framework.

- We demonstrate that models derived from our framework consistently perform well compared to current state-of-the-art methods that ignore the temporal dimension of the data, achieving a 15-70% reduction in error.

The rest of the paper is organized as follows. In Section 2, we review related work. We present our proposed methods and experimental analysis in Sections 3 and 4 respectively. Finally, we offer conclusions in Section 5.

2 Background and Related Work

In this work, we consider relational data represented as a directed, attributed multi-graph $G = (V, E)$, with V (nodes) representing objects and E (edges) representing relations, having one or more connected components. The nodes V represent objects in the data (e.g., people, organizations, events) and the edges E represent relations among the objects (e.g., works-for, located-at). Each node $v \in V$ and edge $e \in E$ are associated with a type $G(v) = g_v,\ G(e) = g_e$. Each object or link type $g \in G$ has a number of associated attributes $\mathbf{X^g} = (X_1^g, ..., X_m^g)$ (e.g., age, gender).

When relational data has a temporal component, there are three aspects of the data that may vary over time. First, attribute values may vary over time $\mathbf{X_i} = (X_{i_1}, X_{i_2}, ..., X_{i_t})$. Second, relationships may vary over time. This results in a different data graph $G_t = (V, E_t)$ for each time step t, where the nodes remain constant but the edge set may vary (i.e., $E_{t_i} \neq E_{t_j}$ for some t_i, t_j). Third, objects existence may vary over time (i.e., objects may be added or deleted). This can also represented as a set of data graphs $G'_t = (V_t, E_t)$, but in this case both the objects and links sets may vary.

In this work, we focus on the second and third case, considering datasets where the objects and links vary over time. We assume that attribute values are static and do not change over time. Focusing on this type of data, we consider the task of attribute prediction and develop methods to model dependencies between the temporal evolution of the link structure and the observed attribute values. Related work can be categorized into the following areas: link analysis, graph models, and Markov decision processes.

In the area of link analysis, there have been a number of researchers who have proposed and analyzed models that exploit temporal changes in link structure (e.g., in the World Wide Web or citation graphs). In particular, many researchers have studied the characteristics of the Web by representing its evolution as a series of graph snapshots (see e.g., [2]). Similar data structures have been used to study the evolution of communities and behavior in Blogspace [17].

Temporal changes in link structure have also been modeled by Cortes et al. [6] and Hill et al. [15]. This work was motivated by the problem of analyzing large-scale dynamic networks such as telecommunications call networks. In order to process massive volumes of data efficiently while accounting for the dynamic nature of transactional data, they proposed a method of summarization based on exponential decay of link weights. Their relational representation captures relational changes in a concise way that evolves smoothly through time and has been utilized for fraud detection analysis.

Temporal locality has been studied for many years in citation analysis (see e.g., [9, 11])—it is well-known that a scientific paper gets the majority of its citations soon after it is published and as time passes, receives fewer citations. Amitay et al. [1] have also identified aspects of temporal locality in Web domains, showing that incorporating hyperlink timestamps into link-based page-ranking algorithms can improve retrieval accuracy.

In the area of graph models, recent work has modeled dynamic graphs with sequential linear models [14]. This work represents temporal sequences of network structures as first-order Markov chains where each network instance is generated by an exponential random graph model [23]. The probability distribution for a graph at time t only depends on the graph at time $(t-1)$: $P(G^t|G^{t-1}) = \frac{1}{Z(\theta,G^{t-1})} exp\{\theta' \psi(G^t, G^{t-1})\}$. However, this model was designed for link prediction tasks and is not geared for attribute prediction. In addition, although Markov Chain Monte Carlo estimation techniques have been developed to learn the parameters of the model in small datasets, the approach is not tractable for large datasets.

In the area of Markov decision processes, some recent work has focused on modeling attributes that change over time. Guestrin et al. [13] combine probabilistic relational models (PRM) [10] with dynamic Bayesian networks [12] to model the state transitions in relational Markov decision processes (RMDP). In essence, RMDPs are a series of PRMs, one for each time step, linked together through temporal slot chains following a first-order Markov assumption, where the state of an object at the next time step $(t{+}1)$ can depend on the state of the object (and related objects) at the current time step t. Although these models can be used for attribute prediction, they cannot model the effect of time-varying link structures—they assume that the links among instances are fixed throughout time.

In our framework, we build on past work that has identified the importance of temporal locality in dynamic graph structures and extend the work of Cortes et al. [6] to develop a summary representation that captures temporal changes in relational structures. We then use this representation to moderate the influence of attributes in relational graphs through modified relational learning techniques. Our approach moves beyond past work in link analysis to use the temporal information in a novel way to improve attribute prediction. We also avoid the restrictive Markov assumptions of past work in graph models and MDPs by summarizing the temporal information before incorporating it into the models. This facilitates reasoning about patterns of temporal locality and temporal recurrence that we believe are crucial for identifying and exploiting influential relationships in the data.

3 TVRC Framework

Our approach, the Time Varying Relational Classifier (TVRC), is a two-phase modeling framework, which consists of a *graph summarization* phase and a *relational classification* phase. The key idea of TVRC is to exploit the temporal influence of relationships across snapshots by summarizing them suitably and utilizing the summarized values in a modified relational classification model. Figure 2 illustrates the general framework of TVRC approach. We represent the data as a sequence of graph "snapshots." Let $\mathbf{G} = \{G_1, G_2, \cdots, G_n\}$ be the sequence of temporal snapshots of the data at consecutive time steps t, from $t = 1,..,n$. Each temporal snapshot G_t contains the relationships between the objects in the dataset at time t. New objects and links may have been added or deleted from G_{t-1} to G_t.

3.1 Graph Summarization

The graph summarization phase summarizes a temporal sequence of relational graphs into a weighted summary graph. Let $G_t = (V_t, E_t, W_t)$ be the relational graph at time step t, where $W_t = \{w_{ij} = 1 : e_{ij} \in E_t\}$ is a set of unit weights on the edges of G_t. We define the summary graph $G_t^S = (V_t^S, E_t^S, W_t^S)$ at time t as a weighted sum of the snapshot graphs up to time t as follows:

$$V_t^S = V_1 \cup V_2 \cup \cdots \cup V_t$$

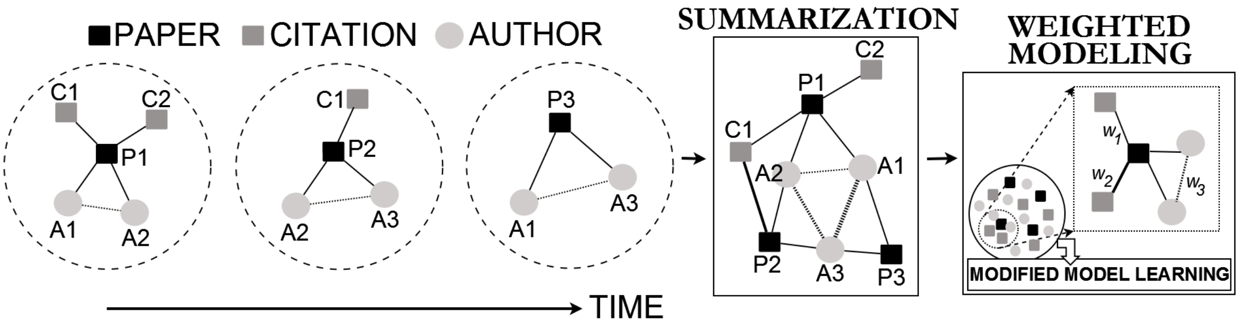

Figure 2. TVRC framework.

$$E_t^S = E_1 \cup E_2 \cup \cdots \cup E_t$$

$$W_t^S = \alpha_1 W_1 + \alpha_2 W_2 + \cdots + \alpha_t W_t = \sum_{i=1}^{t} K(G_i; t, \theta)$$

where V_t and E_t are the vertex and edge sets respectively of the temporal snapshot G_t, and W_t are the unit weights associated with the edges of G_t. The α weights determine the contribution of each temporal snapshot in the summary graph. We use a kernel function K with parameters θ to determine the influence of each edge in the summary. Representing the summary operation through kernel smoothing gives us the freedom to explore and choose a suitable weighing scheme from a wide range of kernel functions, so as to best capture and exploit the temporal variations in the subsequent classification phase. To model temporal locality and temporal recurrence, we employ a weighting scheme based on decaying kernels. In this work, we explore the following three kernels for summarization—exponential kernels, linear kernels, and inverse linear kernels.

Our representation based on kernel summarization is general enough to subsume the two current approaches to modeling temporally-varying link structure in static relational models. The first approach ignores the temporal information on the links and include all links up to time t uniformly. The second approach uses only the links in the current time step t and ignore all links that occurred in the past (i.e., $t' < t$). We define the uniform and pulse kernels below to capture these two approaches.

Exponential Kernel

The exponential kernel weighting scheme is similar to the approach used by Cortes et al. [6] for graph summarization. The kernel function K_E is defined as: $K_E(G_i; t, \theta) = (1-\theta)^{t-i}\theta W_i$. The exponential kernel weighs the recent past highly and decays the weight exponentially as time passes. The parameter θ determines the rate of decay. Figure 3 shows an example of how quickly the unit weight for an event in 1998 decays over time in the summary graph.

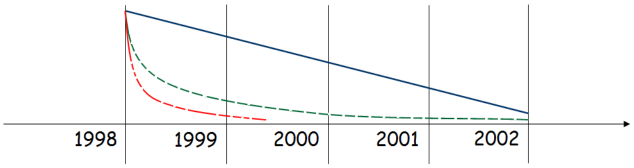

Figure 3. Example weight decay for 1998 link event. Solid curve=K_L, uniform dash curve=K_{IL}, non-uniform dash curve=K_E.

Inverse Linear Kernel

The inverse linear kernel K_{IL} is defined as: $K_{IL}(G_i; t, \theta) = (\frac{1}{t_i - t_o + 1})\theta W_i$ where t_o refers to the first time step considered in the time window. The inverse linear kernel decays more gently than the exponential (see Figure 3) and retains the historical information longer. Again, the parameter θ determines the rate of decay.

Linear Kernel

The linear kernel K_L decays much less rapidly than either the exponential or the inverse linear (see Figure 3) and is defined as: $K_L(G_i; t, \theta) = (\frac{t_* - t_i + 1}{t_* - t_o + 1})\theta W_i$ where t_* refers to the final time step considered in the time window and t_o is defined as above.

Uniform Kernel

The uniform kernel K_U is defined as: $K_U(G_i; t, \theta) = W_i$. The uniform kernel uses the original unit weights on the edges in each snapshot graph and does not attempt to reweight them based on time.

Pulse Kernel

The pulse kernel K_P is defined as: $K_P(G_i; t, \theta) = \{W_i \text{ if } t = t_*; 0 \text{ otherwise}\}$. The pulse kernel only uses the weights on edges in the final step of the time window under consideration.

Note that the kernel summarization approach is general enough to handle cases when more than one link exists between a pair of nodes at a given time step. However, it does assume that if there are multiple links, they are all of the same type and thus can be summarized into a single relationship. For example, if two authors A and B publish two papers and administer a grant together in year t, there will be one summary edge between A and B even though the coauthor links and the co-PI links do not necessarily indicate the same type of relationship between A and B. This case can be easily dealt with by partitioning the graph into a separate graph for each link type g before summarization.

3.2 Weighted Relational Classification

The second phase of the TVRC algorithm is relational classification. Once we have summarized the temporal-relational information into link weights on the summary graph, we learn a predictive model on the summarized data exploiting the temporal information to improve the prediction results. Any relational model that can be extended to use weighted instances is suitable for this phase. In this work, we extend the relational Bayes classifier (RBC) and the relational probability tree (RPT) because of their relative simplicity and efficiency.

Weighted Relational Bayes Classifier

RBCs [21] extend naive Bayes classifiers to relational settings by treating heterogeneous relational subgraphs as homogenous sets of attribute multisets. For example, when modeling the dependencies between the topic of a paper and the topics of its references, the related topics form multisets of varying size depending on the number of associated references (e.g., {GA, NN}, {GA, NN, NN, NN, RL}). The RBC models these heterogenous multisets by assuming that each value of the multiset is independently drawn from the same multinomial distribution. This approach is designed to mirror the independence assumption of the naive Bayesian classifier [7]. In addition to the conventional assumption of attribute independence, the RBC also assumes attribute value independence within each multiset.

More formally, for a class label C, attributes $\mathbf{X}$, and related items R, the RBC calculates the probability of C for an item i of type $G(i)$ as follows:

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

In the equation above, each attribute value is given an implicit weight of 1. In order to incorporate the weights from the summary graph we modify this equation to define the weighted RBC as follows:

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

where w_{ij}^t is the product of the weights in the summary graph G_t^S on the path from node i to the related node j.

Weighted Relational Probability Trees

RPTs [20] extend standard probability estimation trees to a relational setting in which data instances are heterogeneous and interdependent. The RPT algorithm searches over a space of relational features that use aggregation functions (e.g., AVERAGE, MODE, COUNT) to dynamically propositionalize heterogeneous relational multisets and create binary feature splits within the RPT. Similar to RBCs, each attribute value is implicitly given a weight of 1 in the RPT feature construction and evaluation phase. In order to incorporate the weights from the summary graph, the modified RPT uses the weighted multisets to calculate its aggregate relational features. For example, a paper that cites five papers with the following topics: {GA, NN, NN, NN, RL} would have the following feature value $MODE(citedPaper.topic) = NN$ in the original RPT. In the modified RPT, the multiset will be augmented with the summary weights on the citations to those papers: {GA:0.1, NN:0.2, NN:0.1, NN:0.4, RL:0.8} and the feature value will become $MODE(citedPaper.topic) = RL$.

In the modified versions of both the RBC and the RPT, the weights can be viewed as probabilities that a particular relationship in the data is still active at the current time step t, given that the link was observed at time $(t-k)$.

3.3 TVRC Learning

There are three steps to learning a TVRC model: (1) calculating the summary graph, (2) estimating the parameters of the relational model, and (3) estimating the parameters of the kernel function.

The graph summarization can be expressed as a recursive computation on the sequence of graph snapshots $\{G_1, \cdots, G_t\}$. This means that the summary graph at time t can be written as a weighted sum of the graph at time t and the summary graph at time $(t-1)$. For example, the exponential kernel can be computed from the link weights $\{W_1, W_2, \cdots, W_t\}$ as follows:

$$W_t^S = \begin{cases} (1-\theta)W_{t-1}^S + \theta W_t & \text{if } t > t_o \\ \theta W_t & \text{if } t = t_o \end{cases}$$

where t_o is defined as the initial time step in the time window. This enables efficient computation of the summary weights by simply considering the temporal sequence of link events between pairs of linked instances in the data. Note that it does not require considering all pairs of nodes in the data.

Once we have computed the weighted summary graph, we can estimate the parameters of the the relational models using relatively simple modifications to their learning techniques. The weighted RBC uses standard maximum likelihood learning where the sufficient statistics for each conditional probability distribution are computed as weighted sums of counts, based on the edge weights from the summarization phase. The weighted RPT uses the standard RPT learning algorithm except that the aggregate functions are computed after weighing each attribute value count with the corresponding summary edge weight.

We use k-fold cross validation to estimate the kernel parameter θ. We divide the training sample into k folds by sampling the instances independently. We then learn a TVRC model by training it on $(k-1)$ folds and applying the model on the remaining fold using a range of values of θ. In the experiments below, we consider 10 values of θ in the set $\{0.1, 0.2, ..., 1.0\}$ and k=10. The θ value that achieves the maximum cross-validated accuracy on each fold is selected and the average of these values is used as the final θ value. For the kernels considered in this work, there is a single θ parameter that is tied across all the edges in the graphs. Future work will consider the use of different θ parameters for each link type in the data.

The time complexity of learning the TVRC models is $O((t+m)E)$, where t is the number of time steps, m is the number of attributes used for prediction, and E is the number of edges in the summary graph G_t^S. The time complexity for the summary graph computation is $O(tE)$ since each edge must be considered in each summary calculation. In the worst case, when the graph is close to completely connected, the number of edges will be $O(V^2)$. However, in many relational datasets, node degree is bounded by a constant (i.e., a node's neighbors do not grow as a function of dataset size), thus $E << V^2$ in practice. The time complexity for the RBC and RPT learning algorithms are $O(mV)$ where V is the number of target nodes (i.e., instances being predicted) assuming that the number of neighbors for each node (used to get a particular set of attribute values) can be bounded by a constant.

3.4 TVRC Inference

There are two steps to applying a TVRC model for prediction. First, we compute the summary graph $G_{t'}^S$ for the time step t' at which the model will be applied. Then we apply the learned model to $G_{t'}^S$. The prediction phase of RBCs and RPTs is appropriately augmented to incorporate the link weights W_t^S in the multiset representations as described above for learning.

4 Experiments

We report the results of TVRC prediction on three real world datasets. We evaluate the performance of the five kernels and two relational models. The uniform and pulse kernels are included to represent current competing relational models (i.e., RBC/RPT) that ignore the temporal dimension of the data. The results show a significant improvement in prediction accuracy over the competing models, regardless of the choice of decay kernel or relational model.

4.1 Data

We considered three real world datasets for our experiments. The Cora database contains authorship and citation information about computer science research papers extracted automatically from the Web [18]. We considered the set of 16,153 papers published in the time window 1981-1998, along with their associated references and authors.

Figure 4 shows the relational query we used for classification, using modified QGraph [3] notation. The *topic* and *area* attributes are supplied to the relational classification model; the summary weights on reference and coauthor links are computed using the *year* publication dates on papers. The prediction task is to identify whether a paper is a machine learning paper. Note that we use the class label on related objects for prediction but only the labels corresponding to the previous time steps are available to the model (this also applies to the other two datasets).

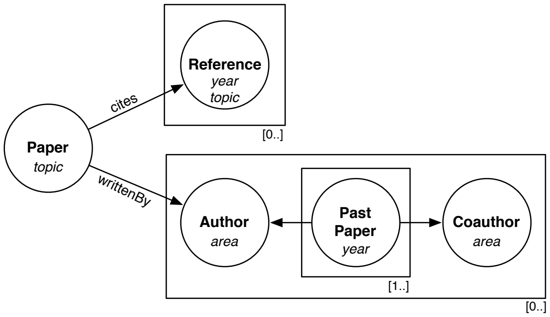

Figure 4. Cora query with attributes used for summarization and classification.

The Reality Mining database contains telephone call and mobile device proximity records among a set of 97 students, faculty, and researchers at MIT over the course of the 2004-2005 academic year (www.reality.mit.edu). Each participant of the study was equipped with a bluetooth cellphone. The data records the duration of phone calls between pairs of subjects as well as the period of time in which pairs of individuals were in proximity of one another.

Figure 5 shows the relational query we used for classification. We considered the five month period from July to November 2004, which contained 443,553 call edges, and 285,512 proximity edges. The prediction task was to predict whether a person is a student or not based on his/her call patterns and device proximity measurements. We defined the granularity of our temporal dimension to a month in these data (compared with a year for the other two datasets). The summary weights are computed using the *date* stamp on the links.

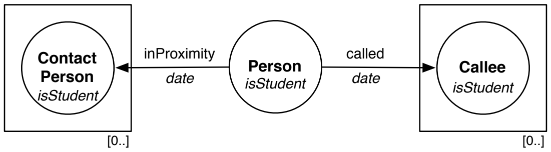

Figure 5. Reality Mining query with attributes used for summarization and classification.

The Internet Movie Database (IMDb: www.imdb.com) contains movie release information, including their earnings, actors, studios, directors, etc. For this work, we selected the set of 5,301 movies that were released in the US in the time period 1981-2007. Each movie has an attribute *earnings*, which records the amount the movie grossed in total. We adjusted these values to account for inflation and make the values comparable across different years. Furthermore, we only considered movies with (adjusted) gross earnings $> \$1mil$.

Figure 6 shows the relational query we used for classification. The classification task was to predict whether a movie would be a blockbuster (earnings $> \$90mil$). The summary weights on the links are computed using the release *year* of the movies.

4.2 Methodology

For each dataset, we divided the data into disjoint temporal samples or 'snapshots' $\{G_1, G_2, \cdots, G_t\}$ where each snapshot G_i corresponds to the events that happened at time i. For IMDb and Cora, we chose snapshots sizes of one year (i.e., G_i consists of all events within a single year): IMDb—$t \in$ [2002,2007], Cora—$t \in$ [1993,1998]. For Reality Mining, we chose a snapshot size of one month: $t \in$ {July, August, September, October, November}. These snapshots sizes were chosen empirically, based on the quality of timestamp information (e.g., in Cora many publication months are missing) and the number of events within each snapshot (e.g., telephone calls are more frequent than movie releases). Future work will consider more fully the impact of snapshot size on model performance. Also note that for each dataset, the summarization phase uses data from a larger time window than the set of windows considered for evaluation (e.g., in IMDB the summarization also considers the data from 1981-2001).

The temporal snapshots are summarized using the kernel functions described in Section 3.1. The classification experiments are set up as follows: we learn the model (RBC and RPT) on the sample corresponding to time t (G_t^S) and apply the model on the subsequent sample corresponding to time $(t+1)$ (G_{t+1}^S). We compare the performance of different methods using area under the ROC curve (AUC).

4.3 Results and Analysis

Our first set of experiments evaluate the performance of TVRC models on the Cora, IMDb and Reality Mining datasets. Figures 7a-c shows the performance of TVRCs using each of the five different kernels for summarization and the RBC model for classification. Figures 7d-f shows the results for the TVRC models that use RPTs as the classification model. Recall that we have included the uniform and pulse kernels to compare to state-of-the-art relational models which use a static representation of dynamic data.

The results show a consistent improvement of the TVRC models with decay kernels compared to the baseline uniform and pulse kernels. Overall, the decay kernels achieve a 15-70% reduction in error, regardless of the choice of relational model. In addition, the exponential kernel outperforms the other decay kernels irrespective of the model used for classification.

To test the significance of the improvements in AUC, we used two-tailed, paired t-tests to compare pairs of models across the set of trials for a given dataset, with a null hypothesis that there is no difference in performance. All TVRC models (with decay kernels) are significantly better ($p < 0.01$) than the competing relational models (i.e., uniform and pulse kernels). This is consistent across all three datasets and the two relational models. In addition, the TVRC with exponential kernel is significantly better ($p < 0.01$) than the TVRC with the linear or inverse-linear kernels. There is no significant difference between the TVRC models that use the RPT and RBC.

We performed another set of experiments to assess the cross-validation aspect of TVRC. We compared the TVRC to its "ceiling" model which used the optimal value for the summary parameter θ (chosen as the θ that performed best on the test set). Figure 7g shows the average AUC perfromance for different values of θ using RBC for relational classification (similar results are observed while using RPT—the plot is omitted for brevity). The squares ($\Box$) show the optimal choice of θ for the TVRC while the diamonds ($\diamond$) show the θ picked using k-fold cross validation. The plot shows an inverted curve with a unique maxima for all the three datasets—however, the optimal point is different for each dataset. It is clear that the parameter setting chosen via cross-validation is not significantly different from the optimal (ceiling) choice of θ. This is notable because we used a more efficient (but potentially less accurate) method of i.i.d. cross-validation rather than designing a more computationally-intensive *relational* cross-validation approach. Although it has been shown that ignoring dependencies among instances in relational domains

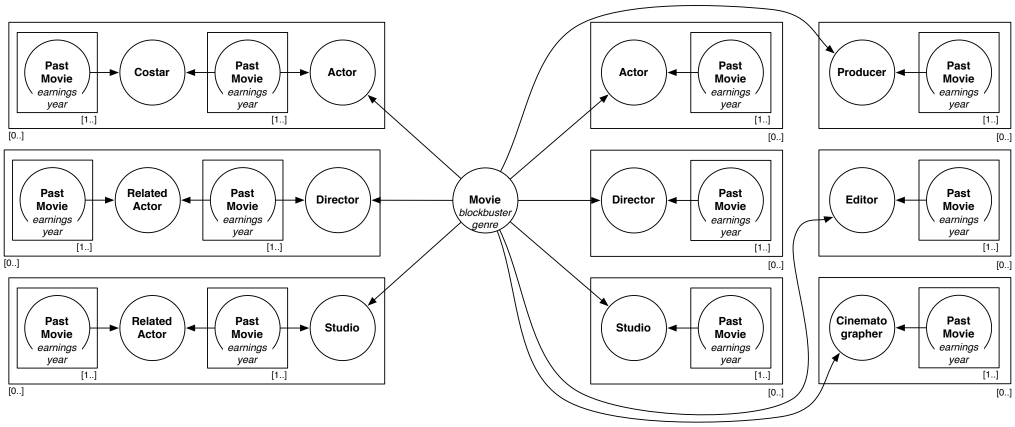

Figure 6. IMDb query with attributes used for summarization and classification.

can result in statistical biases [16], we conjecture that i.i.d. cross-validation is effective in this situation because we are focused on selecting a single parameter—so all choices of that parameter value are biased uniformly and thus, it does not affect the optimal parameter choice adversely.

We also compared the models with ablated data to assess the temporal content in each type of link. Figures 7h-i compare the performance of the models on the Cora data using the RBC as a component model. We graph the performance of the models when only the reference links or the coauthor links are considered in isolation. We differentiate between these two types of links as follows. Reference links occur only once in the snapshot G_t where t is the time when the paper was published. Thus, reference links are examples of *temporal isolated events*. However, links between coauthors are examples of *temporally recurring events* as they can recur everytime a new paper is published. Figure 7h-i shows similar patterns as before–in both cases we see a significant improvement by using the TVRC with an exponential kernel. By examining Figures 7a, h, and i, it can also be seen that TVRC performance is more accurate using both the reference and coauthor links than either alone. This indicates that *temporal isolated events* and *temporally recurring events* are both useful for improving the performance of the model and also that both types of links exhibit temporal dependencies. This effect also holds for the RPT model and the IMDb datasets. (There are only temporally recurring events in the Reality Mining dataset.)

5 Conclusions

This paper presents a new approach to modeling relational data with time-varying link structure. To date, work on statistical relational models has focused primarily on static snapshots of relational datasets even though most relational domains have temporal dynamics that are important to model. Although there has been some work modeling domains with time-varying attributes, to our knowledge this is the first model that exploits information in dynamic relationships between entities to improve prediction. This work has demonstrated that significant performance gains can be achieved by incorporating temporal-relational information into statistical relational models.

We evaluated our algorithm on three real world domains and showed that the TVRC approach achieves significant performance gains compared to competing relational approaches that ignore the temporal component of the data. The improvement is achieved regardless of the decay kernel function used for summarization or the relational model used for classification. Overall, the TVRC results in a 15-70% reduction in error.

One strength of TVRC approach lies in the modularity of the framework whereby each phase is independent of the other, enabling us to choose the appropriate kernel function and relational classifier to best match the domain under consideration. Notably, the static representations generally used to apply relational models to dynamic datasets are special cases within our framework. In this work we have chosen to explore three decay kernels in order to best capture the notions of temporal locality and recurrence. However, our future work will examine the temporal-relational autocorrelation patterns in additional real-world datasets to explore whether alternative kernel functions (e.g., step functions) could produce additonal performance gains. We have chosen to use RBCs and RPTs as the relational models in this work due to their simplicity. However, the TVRC framework is flexible enough that it can be used with other statistical relational models (e.g., [10, 22, 24]) as long as the models can be modified to deal with weighted instances.

Another strength of our approach is that it is a relatively simple and efficient way of incorporating time into statistical relational models. While our algorithm doesn't make a Markov assumption about the temporal dependencies, it is predicated on the assumption that events in the recent past

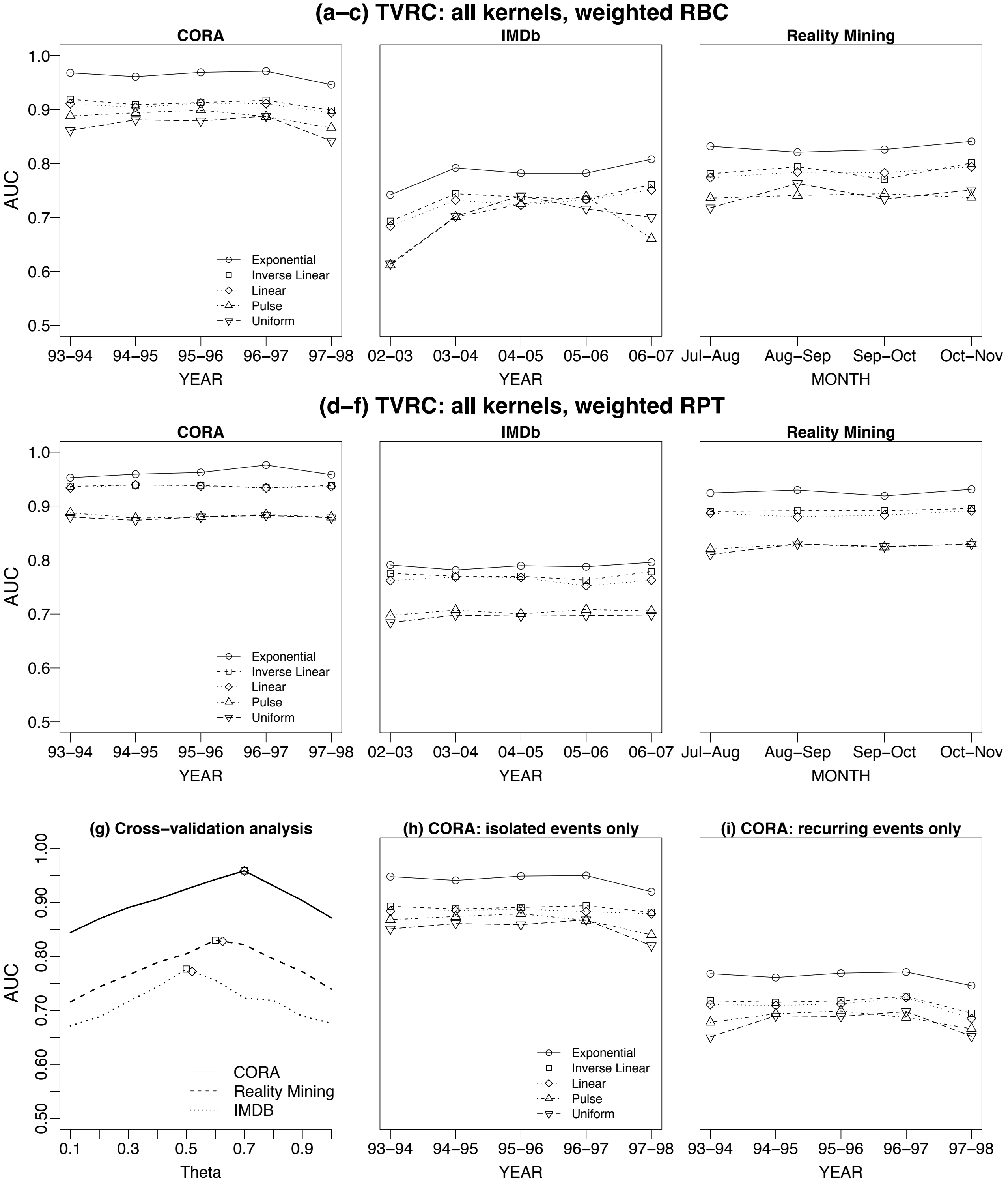

Figure 7. TVRC Experimental results.

are more informative than events in the distant past. A full joint *temporal-relational* model may be able to represent the dependencies in the data more accurately. However without a means to limit either the temporal or relational dependencies, the dimensionality of such a joint model will be far too large for accurate estimation with reasonably-sized datasets.

This work attempts to model temporal dependencies by specifying a limited space of temporal patterns to moderate the relational dependencies. Additional efforts to identify and exploit temporal *motifs* for use as relational features may be a promising means to extend the relational model space in a restricted way while still capturing most of the relevant temporal information in an efficient manner.

Our future work will include an extension to the the temporal summarization scheme to model temporally varying attributes and an investigation of alternative kernels and relational models. In addition, we will cast the model in a more principled graphical model framework, formulating it as a latent variable model where the summary "influence" weights between pairs of nodes are hidden variables that change over time and affect the statistical dependencies between attribute values of incident nodes.

Acknowledgments

This research is supported by DARPA, IARPA, and AFRL under contract numbers FA8750-07-2-0158 and NBCH1080005. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements either expressed or implied, of DARPA, IARPA, AFRL, or the U.S. Government.

References

[1] E. Amitay, D. Carmel, M. Herscovici, R. Lempel, and A. Soffer. Trend detection through temporal link analysis. *J. Am. Soc. Inf. Sci. Technol.*, 55(14):1270–1281, 2004.

[2] K. Bharat, B. W. Chang, M. R. Henzinger, and M. Ruhl. Who links to whom: Mining linkage between web sites. In *Proc. of IEEE Int'l Conference on Data Mining*, 2001.

[3] H. Blau, N. Immerman, and D. Jensen. A visual query language for relational knowledge discovery. Technical Report 01-28, University of Massachusetts Amherst, Computer Science Department, 2001.

[4] D. Cai, Z. Shao, X. He, X. Yan, and J. Han. Community mining from multi-relational networks. In *Proc. of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases*, 2005.

[5] S. Chakrabarti, B. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. In *Proc. of the ACM SIGMOD Int'l Conference on Management of Data*, 1998.

[6] C. Cortes, D. Pregibon, and C. Volinsky. Communities of interest. In *Proc. of the 4th International Symposium of Intelligent Data Analysis*, 2001.

[7] P. Domingos and M. Pazzani. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 29:103–130, 1997.

[8] P. Domingos and M. Richardson. Mining the network value of customers. In *Proc. of the 7th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, 2001.

[9] L. Egghe. A noninfometric analysis of the relationship between citation age and journal productivity. *J. Am. Soc. Inf. Sci. Technol.*, 52(5), 2001.

[10] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *Proc. of the 16th International Joint Conference on Artificial Intelligence*, 1999.

[11] E. Garfield. Citation analysis as a tool in journal evaluation. *Science*, 178:471–479, 1972.

[12] Z. Ghahramani. Learning dynamic Bayesian networks. In *Adaptive Processing of Sequences and Data Structures*, pages 168–197. Springer-Verlag, 1998.

[13] C. Guestrin, D. Koller, C. Gearhart, and N. Kanondia. Generalizing plans to new environments in relational mdps. In *Proc. of the International Joint Conference on Artificial Intelligence*, 2003.

[14] F. Guo, S. Hanneke, W. Fu, and E. Xing. Recovering temporally rewiring networks: A model-based approach. In *Proceedings of the 24th International Conference on Machine Learning (ICML)*, 2007.

[15] S. Hill, D. Agarwal, R. Bell, and C. Volinsky. Building an effective representation of dynamic networks. *Journal of Computational and Graphical Statistics*, Sept, 2006.

[16] D. Jensen and J. Neville. Linkage and autocorrelation cause feature selection bias in relational learning. In *Proc. of the 19th International Conference on Machine Learning*, 2002.

[17] R. Kumar, J. Novak, P. Raghavan, and A. Tomkins. On the bursty evolution of blogspace. In *Proc. of the 12th International WWW Conference*, 2003.

[18] A. McCallum, K. Nigam, J. Rennie, and K. Seymore. A machine learning approach to building domain-specific search engines. In *Proc of the 16th International Joint Conference on Artificial Intelligence*, 1999.

[19] J. Neville, Ö. Şimşek, D. Jensen, J. Komoroske, K. Palmer, and H. Goldberg. Using relational knowledge discovery to prevent securities fraud. In *Proc. of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2005.

[20] J. Neville, D. Jensen, L. Friedland, and M. Hay. Learning relational probability trees. In *Proc. of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003.

[21] J. Neville, D. Jensen, and B. Gallagher. Simple estimators for relational Bayesian classifers. In *Proc. of the 3rd IEEE International Conference on Data Mining*, 2003.

[22] C. Perlich and F. Provost. Aggregation-based feature invention and relational concept classes. In *Proc. of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003.

[23] T. Snijders. Markov chain monte carlo estimation of exponential random graph models. *Journal of Social Structure*, 3(2), 2002.

[24] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *Proc. of the 18th Conference on Uncertainty in Artificial Intelligence*, 2002.