Modeling the Evolution of Discussion Topics and Communication to Improve Relational Classification

Ryan Rossi
Department of Computer Science
Purdue University
rrossi@cs.purdue.edu

Jennifer Neville
Department of Computer Science
Purdue University
neville@cs.purdue.edu

ABSTRACT

Textual analysis is one means by which to assess communication type and moderate the influence of network structure in predictive models of individual behavior. However, there are few methods available to incorporate textual content into time-evolving network models. In particular, modeling *both* the evolution of network topology and textual content change in time-varying communication data poses a difficult challenge. In this work, we propose a Temporally-Evolving Network Classifier (TENC) to incorporate the influence of time-varying edges and temporally-evolving attributes in relational classification models. To facilitate this, we use an evolutionary latent topic approach to automatically discover and label communications between individuals in a network with their corresponding latent topic. The topics of the messages are incorporated into the TENC along with time-varying relationships *and* temporally-evolving attributes, using weighted, exponential kernel summarization. We evaluate the utility of the TENC on a real-world classification task, where the aim is to predict the effectiveness of a developer in the python open-source developer network. We take advantage of the textual content in developer emails and bug communications, which both evolve over time. The TENC paired with the latent topics significantly improves performance over the baseline classifiers that only take into account the static properties of the topics and communications. The results show that the TENC can be used to accurately model the complete-set of temporal dynamics in time-evolving communication networks.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data Mining*

General Terms

Algorithms, Experimentation

1st Workshop on Social Media Analytics (SOMA '10), July 25, 2010, Washington, DC, USA.

Keywords

Statistical relational learning, temporal-relational models, topic modeling

1. INTRODUCTION

Dependencies in relational and network domains have been successfully exploited in classification models (see e.g., [6]). This work focused primarily on modeling *static* network data. For the numerous relational domains that have temporal dynamics, researchers have generally analyzed static *snapshots* of the data, which consist of all the objects, links, and attributes that have occurred up to and including time t (e.g., [5, 12]). This approach ignores the temporal information present in the data and limits the applicability of the models. In many datasets there are dependencies between the temporal and relational information that can be exploited to improve model performance.

Recent work in statistical relational learning has started to investigate these temporal dimensions. Some initial work has focused on transforming temporal-varying links and objects into aggregated features [12] and other work has investigated the modeling of the temporal dynamics of network structure [14]. This work has focused on classification, but there has also been some efforts to exploit temporally-varying links to improve automatic discovery of relational communities or groups [3, 7].

In addition to network topology that changes over time, many relational datasets contain textual content that changes over time. As an example, consider email datasets, which naturally evolve over time and often contain noisy link information (as a single email is not necessarily an indication of a strong relationship). There are also other facets of communication that evolve over time such as the topic of messages between two users over time. Although frequency of communication may be indicative of relationship strength, frequency count is limited in its ability to capture the complexity of interpersonal iteractions between people. It is quite possible that the topic, sentiment, and/or intimacy of the email content is a stronger indicator of relationship type and strength, compared to simple frequency.

While textual content offers a wealth of information for predictive network modeling, there has not been a lot of research investigating how to model the network topology and message content, particularly in the case where both are evolving over time. There has been some work that models a *static* network of documents, using bag-of-word methods [4, 15]. In addition, there are also a number of clustering techniques for discovering topics, roles, and groups in social

networks [9, 8]. These approaches learn topic distributions based on email messages sent between users. The email messages are seen to be informative of the individual's role and consequently the groups to which they belong.

There are also other related text mining approaches for social networks. Agrawal et. al [1] use newsgroups to compare between statistical text analysis versus using only the network structure as a basis for a given classification task. Mei and Zhai [10] develop a temporal text mining techniqueto find interesting evolutionary theme patterns. The approach focuses on the evolution and transition of themes across time. In another approach by Mika [11], the traditional bipartite model of ontologies is extended to incorporate a social dimension, resulting in a tripartite model of various actors, concepts and instances.

The goal of our work is to improve attribute prediction in dynamic domains by incorporating time-varying links and temporally-evolving attributes into statistical relational models. In particular we are interested in modeling the influence of evolving latent topics corresponding to the textual semantics of an individuals communication in our sequences of graphs across time. We focus our analysis on the task of predicting effectiveness within a python open-source developer communication network. We posit that the nature or topic of the communication between two developers is crucial in predicting the effectiveness of a developer. Traditional approaches treat the communications uniformly. Our method assigns latent topics to the communications and incorporates the topic information into predictive models to improve performance.

More specifically, we extract and analyze the publicly available data about open-source software development from the Python project (www.python.org). The communications between developers are each labeled with its corresponding latent topic. The effects of the communication patterns with respect to the latent topics are investigated for their impact on developer effectiveness. We derive a temporally-evolving network classifier to incorporate the temporal evolution of both the relationships and attributes. This classifier is used to learn predictive models of developer effectiveness and explore the utility of using the latent topics to predict developer effectiveness. Our analysis indicates the significance of using the textual information and its evolution over time. Furthermore we find that modeling both temporally evolving topic attributes and time-varying relationships improves the accuracy of the models even further, compared to baseline models that only consider static snapshots of the networks.

The main contribution of our work is two-fold. We develop a temporally-evolving network classifier (TENC) capable of modeling the influence of the complete-set of temporal dynamics in relational domains. Secondly, textual analysis techniques are used to derive a network where the edges and nodes contain latent information which is used as the basis to predict individual effectiveness within a communication network. These proposed techniques could be used to explore other networks such as blogs, forums, or other social media. The TENC is general enough to model the complete-set of temporal dynamics for arbitrary relational networks.

2. COMMUNICATION NETWORK

In this work, one of our goals is to derive a temporally-evolving network classifier capable of modeling the evolution of the latent topics for the communications. In this regards, we analyze observational data from a domain that reflects some characteristics of communication and social relationships. Open-source software development is one such domain where the communication among developers is critical to the success of the project. Email is a common form of communication among the developers, and often mailing lists are used to ensure timely delivery of messages to all interested parties. The textual content of both emails and bug messages between developers are extracted and modeled using our temporal classifier.

2.1 Data

We analyze email and bug communication networks extracted from the open-source Python development environment (www.python.org). In Python development, the primary location for communication is the python-dev mailing list, which is publicly available for subscription or download.

We collected a subset of the python-dev mailing list archive for the period 01/01/07−09/30/08. The sample contains 13181 email messages, among 1914 users. Each message contains the following information: message id, sender email address, sender name, timestamp, in-reply-to message id, subject, body. From the set of messages, we constructed an *email* communication network with nodes corresponding to each unique email address, and edges from the sender of each message to the author of the in-reply-to message.

In addition to mailing list discussions, the Python project also has a public bug-tracking database (bugs.python.org), which records information about projects errors (i.e., bugs) and their associated fixes (i.e., solutions). Each bug is associated with the following information: report date, error type, component, status, assigned-developer, resolution date. In addition, there is information describing the details of the bug and the history of activity related to the bug. There is also a set of messages that record discussion among the developers during their attempt to resolve the bug. The format of these messages is similar to the format of the email messages.

Bug reports were collected for the same period listed above (01/01/07−09/30/08) and we constructed a second *bug* discussion network, where nodes consisted of people with unique email addresses and edges consisted of bug comments sent in reply to a previous posting. The sample contained 69435 bug comments among 5108 users.

In addition to the bug and communication networks we also extracted text from the bug and email messages. Using the text from these messages we perform topic modeling and label the links of the network with their appropriate topic. We also assign developers to their most frequently

Python Communication Network Attributes		
Latent Topics	All Topics Email Topics Bug Topics	
Link Attributes	EdgeCount EmailEdgeCount BugEdgeCount	EdgeTopic EmailEdgeTopic BugEdgeTopic
Temporal Snapshots	[Aug - Oct '07] [Feb - April '08] [Aug - Sept '08]	[Nov '07 - Jan '08] [May - July '08]

Table 1: Details of the Python communication network dataset.

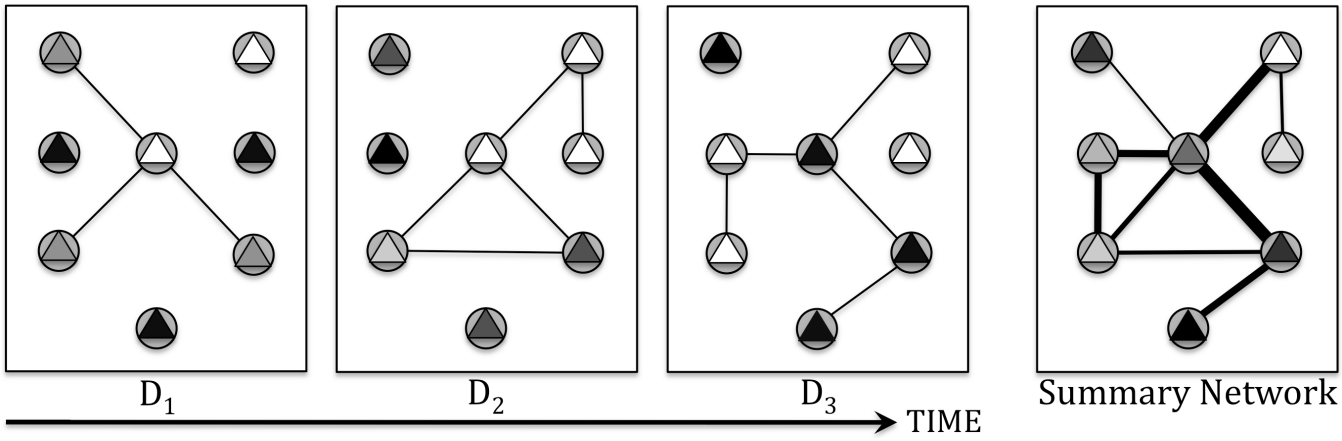


Figure 1: Graph and Attribute Summarization Phase: Temporally evolving edges and attributes.

communicated topic. See Section 4.1 for more detail. For our analysis, we used the set of 185 developers that appeared in both networks (i.e., those that had at least one mailing list message and at least one bug comment).

The defect information (i.e., bugs) associated with the Python dataset enables us to derive measures of individual effectiveness that are consistent with the performance aspect of effectiveness measures. In broad terms, assessing performance on a specific problem-solving task (i.e., problem: errors/bugs and solution: associated fixes) as a measure of effectiveness fits well within the context of a broader theoretical approach to effectiveness in group settings. As a measure of developer effectiveness we used the number of bugs resolved by a developer in a given time window. A binary class label is created that recorded whether a developer has fixed a bug during a three-month period.

2.2 Network Formulation

The Python Communication Network consists of a collection of temporal snapshots. Let $\mathbf{D} = \{D_1, D_2, ..., D_n\}$ be a sequence of temporal snapshots from the relational communication network. Every temporal snapshot corresponds to the events that occured during the time period t, where $t = 1, 2, ..., n$. The size of the temporal snapshots are three month periods where: $D_1 = (Feb07, Mar07, Apr07), D_2 = (May07, Jun07, Jul07), ..., D_7 = (Aug08, Sep08)$. Note that the last temporal snapshot has only two months.

The latent topics of the communications between developers are used to predict individual effectiness ([Has Closed]). Table 1 lists the topic features used in our models. We could have also generated temporal centrality or other types of attributes based on communication patterns. In this work, we are only concerned with using the evolutionary latent topics of the communications between developers in order to predict effectiveness. This allows for the significance of the actual semantics of the communications to be quantified with respect to the measure of effectiveness.

In many relational classification methods, the textual content on links is discarded and the fact that a communication occured between d_i and d_j is the only aspect taken into account. In this work, we are interested in using the textual content of the communications to label *both* the edges and vertices with their corresponding latent topics. Our model uses the relational dependencies between the developers and the latent topics extracted from the textual content of the emails and bugs. More specifically, we focus on predicting individual effectiveness given the latent topics of the communications among individuals.

3. APPROACH

We represent the data as an attributed graph $D = (G, \mathbf{X})$. The graph $G = (V, E)$ represents a set of N individuals as nodes, such that $v_i \in V$ corresponds to individual i and each edge $e_{ij} \in E$ corresponds to a communication event (e.g., email) between individuals i and j. The attribute set:

$$\mathbf{X} = \begin{pmatrix} \mathbf{X^V} = [X^1, X^2, ..., X^{m_v}], \\ \mathbf{X^E} = [X^{m_v+1}, X^{m_v+2}, ..., X^{m_v+m_e}] \end{pmatrix}$$

may contain observed attributes on both the nodes ($\mathbf{X^V}$) and the edges ($\mathbf{X^E}$). Below we use X^m to refer to the generic m^{th} attribute on either nodes or edges.

There are three aspects of relational data that may vary over time. First, the values of attribute X^m may vary over time:

$$\mathbf{X}^m = \begin{bmatrix} x^m_{11} & x^m_{12} & \cdots & x^m_{1t} \\ x^m_{21} & x^m_{22} & \cdots & x^m_{2t} \\ x^m_{31} & x^m_{32} & \cdots & x^m_{3t} \\ \vdots & \vdots & \ddots & \vdots \\ x^m_{n1} & x^m_{n2} & \cdots & x^m_{nt} \end{bmatrix}$$

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 i, j). Third, objects existence may vary over time (i.e., objects may be added or deleted). This is also represented as a set of data graphs $G'_t = (V_t, E_t)$, but in this case both the objects and edge sets may vary.

In this work, we aim to model a communication network that evolves over time, with both edges and attributes (e.g., topics) changing. For simplicity, we assume that the set of nodes remains constant. More specifically, let $D_t = (G_t, \mathbf{X}_t)$ refer to the dataset set at time t, where $G_t = (V, E_t, W^E_t)$ and $\mathbf{X}_t = (\mathbf{X}^V_t, \mathbf{X}^E_t, W^X_t)$. Here W_t refers to a function that

assigns weights on the edges and attributes that are used in TENC below. We define $W_t^E(i,j) = 1$ if $e_{ij} \in E_t$ and 0 otherwise. Similarly, we define $W_t^X(x_i^m) = 1$ if $X_i^m = x_i^m \in \mathbf{X_t^m}$ and 0 otherwise.

3.1 Temporally-Evolving Network Classifier

The Time Varying Relational Classifier (TVRC) [14] is a recent approach that has been developed to incorporate temporal dependencies into statistical relational models. We define the Temporally-Evolving Network Classifier (TENC) as an extension of the TVRC to incorporate the influence of temporally evolving *attributes* (e.g., topics) on both the *edges* and the *nodes*. The TENC uses a two-step process that first transforms the dynamic relational dataset into a statically weighted summary graph and set of summary attributes using kernel smoothing. The second phase then incorporates the static edge and attribute weights into a modified relational classifier to moderate the conditional attribute dependencies throughout the relational data graph.

The graph summarization phase follows the TVRC process by summarizing a temporal sequence of relational graphs into a weighted summary graph. Let $G_t = (V, E_t, W_t^E)$ be the relational graph at time step t, where W_t^E are unit weights on the edges as defined above. We define the summary graph $G_{S_t} = (V, E_{S_t}, W_{S_t}^E)$ at time t as a weighted sum of the temporal graphs up to time t as follows:

$$E_{S_t} = E_1 \cup E_2 \cup \cdots \cup E_t$$

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

where E_t is the edge set of the temporal snapshot G_t, and W_t^E 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. For weighting, we use an exponential kernel function K_E with parameter θ to determine the influence of each edge of snapshot graph G_i in the summary graph:

$$K_E(G_i; t, \theta) = (1-\theta)^{t-i} \theta W_i^E$$

With the exponential kernel function, we can compute the summary graph weights recursively, as a combination of the summary graph weights at time $t-1$ and the snapshot graph weights at time t:

$$W_{S_t}^E = \begin{cases} (1-\theta) W_{S_{t-1}}^E + \theta W_t^E & \text{if } t > t_1 \\ \theta W_t^E & \text{if } t = t_1 \end{cases}$$

where t_1 refers to the first timestep of the data.

In addition to summarizing the graph changes over time as in the TVRC, the TENC also summarizes the attribute changes in a similar fashion. Let $\mathbf{X}_t = (\mathbf{X}_t^V, \mathbf{X}_t^E, W_t^X)$ be the attribute data at time step t, where W_t^X are unit weights on the attribute values as defined above. We define the summary attribute set $\mathbf{X}_{S_t} = (\mathbf{X}_{S_t}^V, \mathbf{X}_{S_t}^E, W_{S_t}^X)$ at time t as a weighted sum of the temporal attribute sets up to time t as follows:

$$\mathbf{X}_{S_t}^V = \mathbf{X}_1^V \cup \mathbf{X}_2^V \cup \cdots \cup \mathbf{X}_t^V$$

$$\mathbf{X}_{S_t}^E = \mathbf{X}_1^E \cup \mathbf{X}_2^E \cup \cdots \cup \mathbf{X}_t^E$$

$$W_{S_t}^X = \beta_1 W_1^X + \beta_2 W_2^X + \cdots + \beta_t W_t^X = \sum_{i=1}^{t} K_X(\mathbf{X}_i; t, \lambda)$$

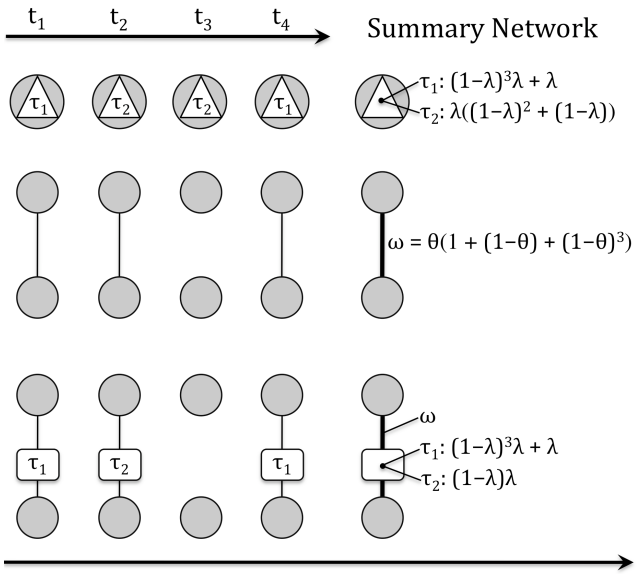


Figure 2: The TENC models the complete-set of temporal dynamics, including node attributes changing over time (top row), edges changes over time (middle row), and link attributes changing over time (bottom row).

where $\mathbf{X}_t$ is the attribute set of the temporal dataset D_t, and W_t^X are the unit weights associated with the attribute values of $\mathbf{X}_t$. The β weights determine the contribution of each temporal snapshot in the summary network. For weighting, we again use an exponential kernel function K_X with parameter λ to determine the influence of each attribute value of $\mathbf{X}_i$ in the summary network:

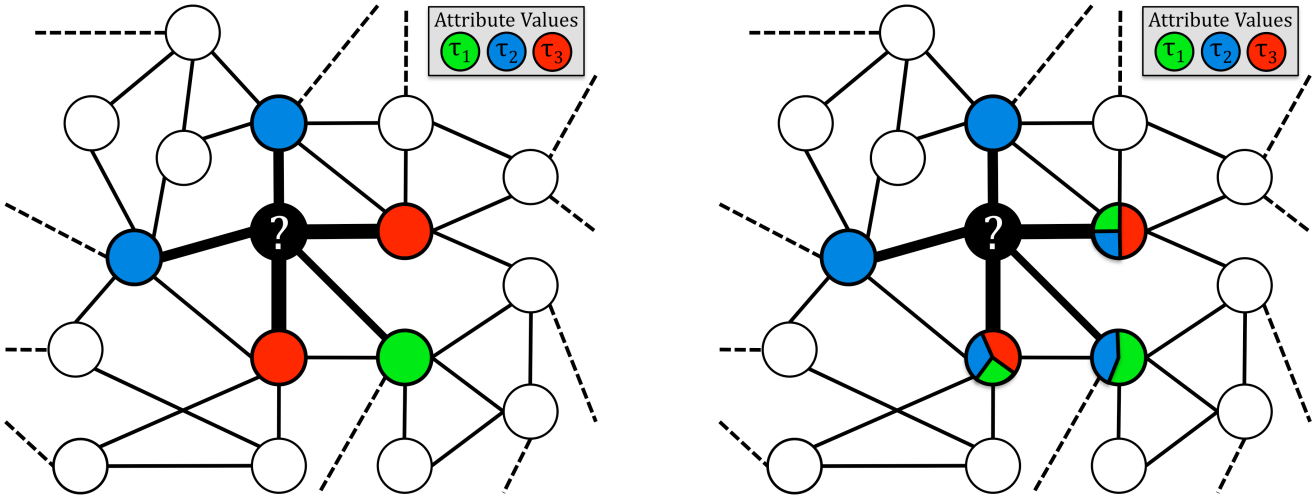
$$K_X(\mathbf{X}_i; t, \lambda) = (1-\lambda)^{t-i} \lambda W_i^X$$

We can also compute the summary attribute weights recursively, as a combination of the summary attribute weights at time $t-1$ and the snapshot attribute weights at time t:

$$W_{S_t}^X = \begin{cases} (1-\lambda) W_{S_{t-1}}^X + \lambda W_t^X & \text{if } t > t_1 \\ \theta W_t & \text{if } t = t_1 \end{cases}$$

The exponential kernel weights attribute values (and edges) that have occurred in the recent past highly and then decays those weights exponentially as time passes. The parameters θ and λ determines the rate of decay for edges and attributes, respectively. The kernel smoothing operation on the input temporal sequence $\{D_1, D_2, \cdots, D_t\}$ can also be expressed as a recursive computation on the weights $\{W_1^E, W_2^E, \cdots, W_t^E\}$ and $\{W_1^X, W_2^X, \cdots, W_t^X\}$ through time as shown above.

The second phase of the TENC algorithm is similar to the weighted relational classification of the TVRC. Once we have summarized the temporal graph and attribute information into edge and attribute weights, respectively, in the summary network $(G_{S_t}, \mathbf{X}_{S_t})$. We then learn a predictive model on the summarized data using the edge and attribute weights to moderate the influence of the weighted relational topic attributes. This approach exploits the temporal information in both the communications and the attribute values to improve the prediction results. More specifically, the



(a) TVRC Model (b) TENC Model

Figure 3: (a) The TVRC feature calculation that includes only the summary edge weights. (b) The TENC feature calculation that incorporates *both* the summary attribute weights and the summary edge weights.

set of attributes that are used for learning include the set of attribute values in $\mathbf{X}_{S_t}$ and each attribute value x_i^m is weighted by its summary weight $W_{S_t}^X(x_i^m)$. When relational attributes are considered by the model, the attributes are weighted by the product of their temporal attribute weight and the link weight of the appropriate link in the summary graph. For example, if node i and j are linked in G_{S_t} and the model uses attribute X^k on node j to predict the class label on node i, then if $W_{S_t}^E(i,j) = \omega$ and $W_{S_t}^X(x_j^k) = \psi$, the attribute value x_j^k will be given a weight of $\omega \cdot \psi$ during learning.

Any arbitrary relational model that can be extended to use weighted instances is suitable for this phase. In this work, we use the relational probability tree (RPT) [13] to compare roughly equivalent models, both with and without reasoning over the temporal dynamics.

The weights can be viewed as probabilities that a particular relationship or attribute value in the network is still active at the current time step t, given that it was observed at time $(t-k)$. Further if we consider the exponential kernel, then we have a geometric distribution with either parameter, θ for edges or λ for attributes. The distribution specifies the probability that an edge or attribute value occured k time steps in the past.

The edges in Figure 3 represent the communications between individuals and the topics on the nodes are denoted τ_1 (green), τ_2 (blue), and τ_3 (red). The thickness of the edge represents the temporal strength of the communication and the colors represent attribute values of either τ_1, τ_2, and τ_3. The TENC Model has probability distributions of attribute values for every node. The weighted relational neighborhood is used in the prediction process.

Figure 3 shows the feature calculations of TVRC and TENC. In particular, consider the weighted mode of the linked topics in TVRC. It is obvious from Figure 3(a) that the combined weight associated with the latent topic τ_3 (red) is greater than the other observed latent topics. Nevertheless, consider the weighted mode of the linked topics in TENC where both the influence of the edges and attributes are incorporated. Clearly, the combined weight associated with τ_2 (blue) is greater than the combined weight of τ_3 (red) as shown in Figure 3(b). Therefore by modeling the complete-set of temporal dynamics we are able to more accurately capture and exploit the evolutionary patterns to improve predictions of relational classifiers.

In order to incorporate the weights from the summary graph and summary attributes, we define a weighted RPT by modifying the counts of the attribute value when computing the RPT aggregate functions. The weighted relational neighborhood is incorporated in the feature calculation. More specifically, the product of the edge weight and the corresponding attribute weight represents the combined influence of an attribute X_{ij} and edge E_{ij} across time.

The RPT classifier takes a sequence of temporal snapshots $\{D_1, D_2, ..., D_t\}$ from the communication network. In every timestep we predict effectiveness using the *Has Closed* attribute. The other objects and links form the relational network used as a basis for our predictions. The RPT algorithm constructs a probability estimation tree for the temporal sample D_t to predict the effectiveness of a future D_{t+1} temporal sample. The temporal latent topic attributes change from D_t to D_{t+1}.

The temporal snapshots in TENC are summarized and then weighted using the exponential kernel function. The graph summarization phase summarizes a temporal sequence of relational graphs which are then used to predict the effectiveness or the HAS CLOSED attribute. In TENC the relational topic attributes are moderated by the summary graph weights and the summary attribute weights.

For every temporal snapshot D_t we learn a model and apply it to D_{t+1} a future temporal snapshot to predict the effectiveness. The TENC uses the current *summarized* temporal snapshot D_{S_t} to predict the future $D_{S_{t+1}}$ *summarized* temporal snapshot.

3.2 Content Analysis

An edge between two developers in the python communication network represents an email or bug communication. We use techniques based on Latent Dirichlet Allocation[2] to extract topics of the communications. The latent topics are used to label the communication links between users. Let $\mathcal{T} = \{\tau_1,\ \tau_2,\ ...,\ \tau_\kappa\}$ be a set of latent topics extracted from the bugs and email communications. In the task of predicting effectiveness between individuals we might find that $\tau_i = \{web\ programming\}$ and $\tau_j = \{sports\}$ therefore it is clear that individuals communicating about sports may be less effective, while communications about web programming might be a significant indicator of effectiveness. The latent topics provide context for the links instead of the uniform notion of a simple communication. This method disambiguates and therefore assigns useful meaning for the edges and consequently the nodes in a relational network.

The bug and email communications are from developers who work on distributed teams. Let $\mathcal{C} = \{c_1,\ c_2, ..., c_m\}$ be a set of bug and email communications and $\mathcal{W} = \{w_1,\ w_2, ..., w_n\}$ be a set of words from the communications between developers. We use the vector-space representation. We define an $n \times m$ matrix denoted M where the coefficents represent the frequency of $w_i \in \mathcal{W}$ appearing in $c_j \in \mathcal{C}$. The matrix is of size 191607 words $\times$ 82616 communications. A standard list of stopwords were removed. Every communication is represented by an edge between two people in the relational network. Using the text from the communication we assign the edge an attribute that corresponds to the topic of communication. We introduce a latent class variable for the topics $\mathcal{T} = \{\tau_1,\ \tau_2,\ ...,\ \tau_\kappa\}$ where κ is the number of topics to be discovered. This can be thought of as a type of dimension reduction where the communications are projected into κ dimensions (or topics). Every communication edge is labeled with its most likely topic and conversely every individual is labeled with their most frequent topic of communication. The topic discovery is extended temporally by considering every timestep separately. Therefore at time t the topic of a communication between d_i and d_j might be τ_1, but at $t+1$ the topic of conversation might evolve into topic τ_2 and a similar temporal transition could occur at the individual or attribute level as well. The next obvious step is to use the full distribution of the topic likelihood instead of selecting the most likely topic. Further details are discussed in section 4.1 on topic discovery.

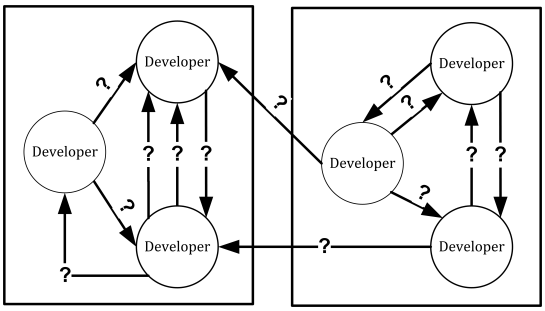


Figure 4: Communication links with uniform semantics.

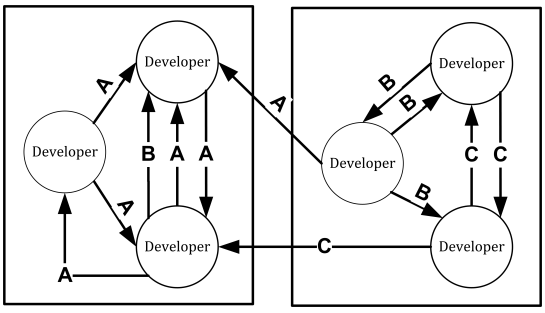


Figure 5: Adding textual information by labeling links in the network with latent topics.

4. RESULTS

In our experiments, we evaluate the various temporal dimensions of communication networks, with respect to the task of predicting node behavior. The PyComm Network is a complete-temporal real-world dataset where nodes, edges, and attributes are rapidly evolving over time. Additionally, the class label based on developer productivity is also changing across time. In our empirical evaluation, we specifically consider the effects of modeling the latent topics of the communication networks and their evolution over time.

4.1 Topic Discovery

The number of topics to estimate depends on many factors including the type of corpus (web content, open source development, ..) and also on the size of the collection. We chose $\kappa = 20$ as the majority of communications are most likely to focus on some aspect of the 18 Python projects under development.

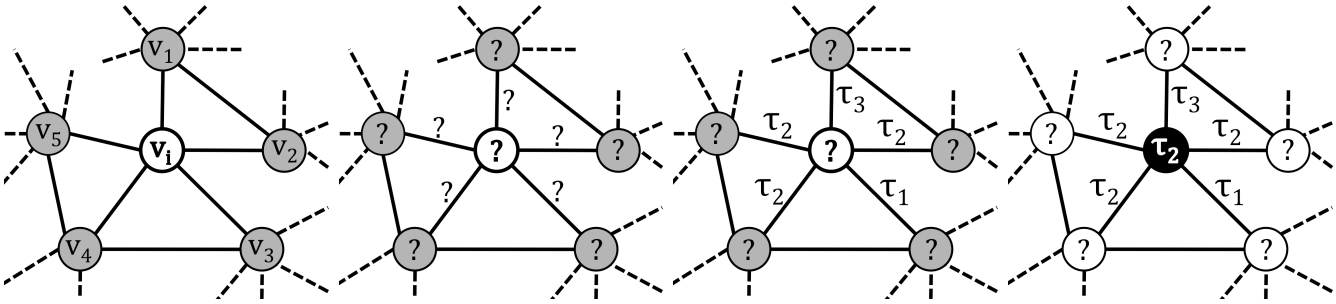
We removed a standard list of stopwords from the communications and also other technical words that appeared with high frequency (e.g., python). A topic can be viewed as a cluster of words that are frequently used together. We used a version of Latent Dirichlet Allocation [2] to model the κ topics. To estimate the parameters we used Expectation-Maximizatiom (EM) and Gibbs sampling for inference.

We modeled the topics in three different sets of communications: (1) the email communication, (2) the bug communications, and (3) the joint set of email and bug messages. After extracting the latent topics from the email and bug communications we label the edges of our relational communication network with their appropriate latent topic. The edges are labeled by performing inference on a communication and assigning it to the topic with the greatest liklihood and consequently a nodal attribute is defined as the mode of the neighbors latent topics.

We generate three object attributes {AllTopic, EmailTopic, BugTopic} where a developer is assigned to the most prevalent topic in a particular set of communications. As an example if a developer most often discusses 'testing' in her bug messages for a particular timestep then she would be assigned to the BugTopic corresponding to testing.

The text of the messages between individuals in the PyComm network are used to predict individual effectiveness. Using this text we automatically extract hidden semantics of the messages and assign an arbitrary message to a latent topic by considering the latent topic with the greatest liklihood.

In Table 2 we list the most likely words for five of the 20 topics when we combine bug and email communication to-



(a) Initial Network (b) Uniform semantics (c) Edges labeled (d) Nodal attributes assigned

Figure 6: Defining a network using only the evolving textual information. The edges are labeled with their corresponding latent topic. (a) The initial network has only the textual information associated with the messages between developers. As an example, $C_{i1} = \{w_1, w_2, ..., w_z\}$. (b) All edges and topic attributes are initially semantically uniform. (c) Edges are labeled with their corresponding latent topic using maximum liklihood. (d) Nodal attributes are assigned a latent topic based on the mode of the neighboring latent topics.

Table 2: The first five topics from the email and bug communications. The number of topics modeled is twenty ($\kappa = 20$).

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
dev	logged	gt	code	test
wrote	patch	file	object	lib
guido	issue	lt	class	view
import	bugs	line	case	svn
code	bug	os	method	trunk
pep	problem	import	type	rev
mail	fix	print	list	modules
release	fixed	call	set	build
tests	days	read	objects	amp
work	created	socket	change	error
people	time	path	imple	usr
make	docu	data	functions	include
pm	module	error	argument	home
ve	docs	open	dict	file
support	added	windows	add	run
module	check	problem	def	main
things	doc	traceback	methods	local
good	doesnt	mailto	exception	src
van	report	recent	ms	directory

gether. The table contains words with both positive and negative connotation such as 'good' or 'doesnt' and also words referring to the network domain such as bugs or exception. An intersting direction to pursue would be to use sentiment analysis to automatically identify the communications with positive and negative tone and use those predictions in the analysis, to moderate relationships among developers.

It is difficult to subjectively assess the high level meaning of a given latent topic from the communications. This is likely due to the fact that we are analyzing communication in a highly specific technical domain. Nevertheless we identify some patterns from the topics. As an example, the first topic seems to be about open source development in a much broader sense. The word 'guido' appears significant, since this is likely a reference to the author of the Python programming language, Guido van Rossum.

4.2 Predicting Effectiveness

The experiments below are intended to investigate the following hypotheses in this domain:

- Relational latent topic information can improve model performance over using simple intrinsic aggregate topic features.
- Information about the temporal dynamics of the latent topics can improve model performance.
- Modeling the topics influence of both the time-varying edges and temporally-evolving attributes improves performance over simply modeling the temporal dynamics of edges.

The main facets of interest are whether incorporating the influence of both time-varying edges and temporally-evolving attributes (in the TENC) improves performance over simply modeling the temporal dynamics of the time-varying edges (in the TVRC). The second facet is if by using TENC, we can successfully predict the effectiveness through the latent topics of the communications between developers. Along these lines, we evaluate the significance of TENC where the influence of both edges and attributes are incorporated into the model. These experiments provide insight into modeling the complete-set of temporal dynamics as well as using textual analysis techniques to incorporate the temporal dynamics of the latent topics into networks.

We evaluate four different models for the classification task of predicting effectiveness using the latent topics. The TVRC, Window, and Union Models are used as baseline classifiers to evaluate the significance of TENC where the influence of the topics from both the temporally evolving edges and attributes are incorporated to predict effectiveness:

– **TENC**: We present the results of the TENC algorithm using an exponential smoothing kernel for summarization of both the relationships and the attributes. The TENC model first summarizes the network into D_{S_t} using all snapshots up to and including t, selecting the weighing parameters θ and λ using k-fold cross validation and then weights the relationships and attributes appropriately during learning and prediction.

This model summarizes both the relationships and the attributes from the latent topics.

– **TVRC**: This model summarizes the graph into G_{S_t} using all timesteps up to and including t, selecting the weighting parameter θ using k-fold cross validation. The attributes are moderated using the temporal edge strengths only.

– **Union Model**: The union model is a baseline model that uses a graph $D_{\leq t}$, which consists of all objects and links up to and including the year t of the sample, for learning. The union model does not weight the attributes or the links.

– **Window Model**: The window model is again a baseline model that uses the network D_{t-1} for prediction on network D_t. In other words, it only uses the immediate past information for prediction and ignores all the other historical data. This model corresponds to the RPT.

The models are learned using the latent topics of the communications and the performance is evaluated using area under the ROC curve (AUC). For every timestep t, we learn a model on D_t and apply the model to D_{t+1}. The utility of incorporating the various granularities of temporal information is measured by comparing TENC to the TVRC. The utility of the temporal information is further evaluated by comparing against two baseline models that ignore the temporal aspects.

We first model the latent network semantics of communications (i.e, textual information) between developers through topic discovery. Using these temporal latent topics, we apply the four relational classifiers. In Figure 7 the results indicate the necessity of appropriately modeling the complete-set of temporal dimensions. Our TENC model that incorporates both the temporal influence of the topic attributes and edges drastically improves model performance over the baseline models. Nevertheless, the TVRC model that incorporates only the influence of time-varying edges also improves the accuracy of our models over the Window and Union Models which ignore the temporal aspects of the dataset.

The temporal dynamics must be modeled and incorporated to successfully predict effectiveness as the static models are seen to be poor predictors. This is an interesting point since it implies that the static latent topics are not very meaningful. The performance is substantially improved only when the temporal-dynamics are incorporated into the predictive models. Modeling the temporal influence of attributes and edges improves performance significantly over all the static models (Window/Union Models) in every timestep as seen above. Therefore we have shown the necessity to model the complete-set of temporal dynamics of datasets to maximize the accuracy of the performance of classifiers.

Evolutionary patterns between the topics of communications and the individuals themselves are clearly present within the communication network. These results indicate that productive developers usually communicate about similar topics or aspects of development. An effective communication has a specific structure or latent meaning that consequently enables others to become more effective. This latent structure is captured more accurately through evolutionary

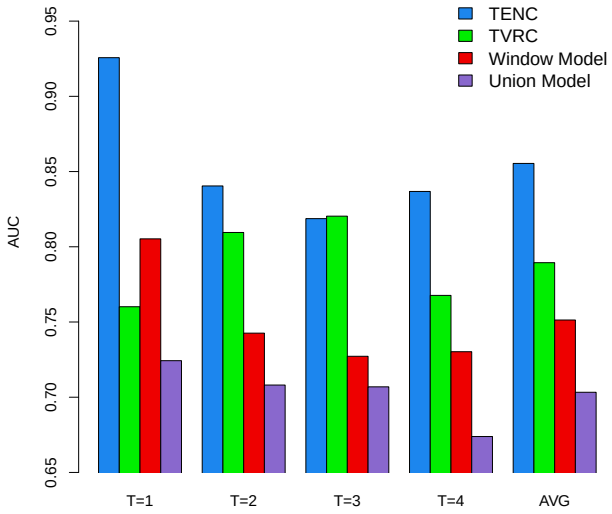


Figure 7: AUC results for each timestep where TENC is compared against the baseline models. The models use only the latent topics of the communications to predict effectiveness. LDA is used to automatically discover the latent topics as well as annotating the communication links and individuals with their appropriate topic in the temporal networks.

topic patterns. Moreover, there are indications of evolutionary patterns within the topics. We posit that as the semantics of a topic evolve over time, an effective individuals messages also transition to corresponding effective topics. An effective topic might evolve over time into an uneffective topic. As an example, suppose an effective topic at time t is mainly about a specific set of functions. However, at time t+1 the functions become depricated and therefore the effective topic has evolved into an ineffective topic of communication. It is likely that the individuals who are still actively pursuing development and discussing the depricated operations are also becoming ineffective.

Our results indicate the necessity of appropriately modeling the full set of temporal dimensions. The TENC performance is significantly better than the static Window and Union models at $p < 0.01$ and better than TVRC at $p < 0.1$ level. We have shown that by modeling and incorporating the temporal influence of attributes and relationships leads to a vast improvement in accuracy and an overall robust model.

5. CONCLUSIONS

We presented a novel approach for modeling time-varying relational data that incorporates textual content by use of latent topics where both the communication links and topics are evolving over time. The TENC framework is not restricted to communication networks, but is applicable to a wide array of relational domains where the relationships between entities change over time as well as the entities attributes. Furthermore, the latent semantic labeling technique is not restricted to natural languages and can be ap-

plied to label the edges and nodes of other types of networks. We have shown the significant increase in performance by including the temporal influence of the latent topics on both the attributes and edges.

The temporal dynamics of the latent topics is a significant aspect in the prediction of individual effectiveness. This is shown using the Temporally-Evolving Network Classifier where the performance is substantially increased when modeling the influence of the temporal dynamics. The utility of modeling a developers latent topic of communications across time is important in determining future communications. This means that a developers past topics of communications are somewhat predictive of their future conversations with their codevelopers and the influence should be modeled accordingly (i.e, decayed over time). These results indicate the necessity of modeling both the time-varying communication edges and the temporally-evolving latent topic attributes. These findings warrant a more systematic investigation of extending relational learning algorithms to take into account the complete-set of temporal dimensions as well as using similar textual analysis approaches for labeling edges and creating attributes through latent temporal information.

Our future work will consider using the entire distribution of topic liklihoods instead of selecting the most likely topic. We are also interested in the temporal evolution of topics with respect to the corresponding individuals communications. In particular the temporal patterns and transitions of topics as they relate to the productivity of the individuals.

6. ACKNOWLEDGMENTS

This research is supported by DARPA and NSF under contract number(s) NBCH1080005 and SES-0823313. 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, NSF, or the U.S. Government. This research was also made with Government support under and awarded by DoD, Air Force Office of Scientific Research, National Defense Science and Engineering Graduate (NDSEG) Fellowship, 32 CFR 168a.

7. REFERENCES

[1] R. Agrawal, S. Rajagopalan, R. Srikant, and Y. Xu. Mining newsgroups using networks arising from social behavior. In *Proceedings of the World Wide Web Conference*, pages 529–535, 2003.

[2] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

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

[4] S. Chakrabarti, B. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 307–318, 1998.

[5] P. Domingos and M. Richardson. Mining the network value of customers. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 57–66, 2001.

[6] L. Getoor and B. Taskar, editors. *Introduction to Statistical Relational Learning*. MIT Press, 2007.

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

[8] A. McCallum, X. Wang, and A. Corrada-Emmanuel. Topic and role discovery in social networks with experiments on enron and academic email. In *Journal of Artificial Intelligence Research (JAIR)*, pages 249–272, 2007.

[9] A. McCallum, X. Wang, and N. Mohanty. Joint group and topic discovery from relations and text. In *Statistical Network Analysis: Models, Issues and New Directions, Lecture Notes in Computer Science 4503*, pages 28–44, 2007.

[10] Q. Mei and C. Zhai. Discovering evolutionary theme patterns from text - an exploration of temporal text mining. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 198–207, 2005.

[11] P. Mika. Ontologies are us: A unified model of social networks and semantics. In *Journal of Web Semantics*, pages 5–15, 2007.

[12] J. Neville, O. Şimşek, D. Jensen, J. Komoroske, K. Palmer, and H. Goldberg. Using relational knowledge discovery to prevent securities fraud. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 449–458, 2005.

[13] J. Neville, D. Jensen, L. Friedland, and M. Hay. Learning relational probability trees. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 625–630, 2003.

[14] U. Sharan and J. Neville. Temporal-relational classifiers for prediction in evolving domains. In *Proceedings of the 8th IEEE International Conference on Data Mining*, 2008.

[15] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *Eighteenth Conference on Uncertainty in Artificial Intelligence*, 2002.