ABSTRACT

Social network analysis has garnered increasing attention in recent years, with implications across a broad range of applications. One particular subfield, link prediction, has become a key part of modern counter-terrorism efforts. However, the difficulty of finding reliable and publicly accessible terrorist network data has limited academic research in this direction. Previous work at the University of Maryland has investigated supervised link classification on a publicly available dataset, the Profiles in Terror (PIT) knowledge base, using Relational Markov Networks (RMNs). This paper introduces models which combine singular value decomposition and logistic regression to yield improved results across all tasks. We discuss issues discovered in the processing method used by the UMD group and suggest improvements. Finally, we apply the models to a new dataset published at Carnegie Mellon University, but conclusions are undermined by the small size of the dataset and the scarcity of positive examples.

1. INTRODUCTION

In a social network where nodes represent individuals and edges indicate relationships between these individuals, the link prediction problem involves assessing the likelihood that there exists a yet unobserved connection between two nodes. This is particularly applicable when dealing with missing data and uncertainty. For example, in the domain of online social networks, users’ personal networks are broad and constantly changing, and users may need to be encouraged to update their online network to reflect their true personal network. Accurate recommendations about people with whom the user may already have a personal relationship can facilitate this process, leading to richer network data.

In general, social network analysis has attracted much attention from the counter-terrorism field. This is indicated by a growing amount of research and the existence of forums such as the Workshop on Link Analysis for Adversarial Data Mining (formerly the Workshop on Link Analysis, Counter-terrorism, and Security). The link prediction problem itself is highly relevant; due to limited investigative resources, it is useful to not only identify likely links but to also rank them in order of likelihood. Accurate leads can prevent catastrophes, but false leads can lead to wasted efforts and increased public distrust, so precision is invaluable.

One difficulty in researching this problem is the scarcity of real-world datasets that are both publicly available and of good quality. The situation is inherently adversarial, in that it is in the interests of the terrorist groups to keep their activity obscured. Even when data does exist, it may be classified information. In previous work, [1] discusses the results of experiments conducted with the ongoing Profiles in Terror¹ (PIT) semantic web project at the University of Maryland. The original PIT dataset contains relevant terrorism-related information extracted from public news sources such as newspapers, broadcasts, and court proceedings, and presents opportunities for various types of research. Part of the work performed by Zhao et al. involves formulating a multi-label link classification task using the PIT dataset. [1] focuses on models based on Relational Markov Networks (RMN), which present complexity difficulties as discussed in the paper.

First, we use the Terrorists² dataset produced by Zhao et al. and demonstrate the effectiveness of low-rank approximation methods in extracting structural information from networks. We use logistic regression models to combine this information with descriptive features from the data known about each agent in the network, yielding results that improve upon RMN accuracy in all tasks. In one particular task, we improved performance by more than 10%. In order to learn more about the data, we also investigate the original

PIT knowledge base. We discuss issues found with missing and inconsistent data.

In addition to the PIT and derived datasets, we also investigate the West Bank dataset published by the Center for Computational Analysis of Social and Organizational Systems (CASOS) lab at Carnegie Mellon University. This dataset contains network information about terrorist groups in the West Bank area, and is similar to but much simpler and smaller than the PIT dataset. We apply our models to this new dataset, with positive results.

2. PROFILES IN TERROR

Profiles in Terror is an ongoing semantic web project to collect and encode public counter-terrorism data in a semi-structured format that is accessible to both humans and computers. The basic elements of the dataset are entities, which can have arbitrary descriptive properties and relationships with other entities. By combining XML, the Resource Description Framework (RDF), and the Web Ontology Language (OWL), the PIT knowledge base supports the creation of type hierarchies and semantic rules. The relevant types and subtypes that currently exist in the dataset are listed below, along with the number of entities matching those types:

- **Organization** (283)
- **Terrorist_Group** (67): An Organization of terrorists, such as Hamas and Hezbollah.
- **Person** (962)
- **Terrorist** (539): A Person that has engaged in terrorist activity.
- **TerroristLeader** (129): A Terrorist that is the leader of a group, such as Osama Bin Laden.
- **Facility** (51): Training camps, etc.
- **Event** (1885)
- **Terror_Plot** (1214): A subclass of Event that indicates real and suspected terrorist plots
- **Terror_Attack** (1252): A subclass of Event that includes attack types such as bombings.

Note that entities may be tagged with more than one type. Also, although a specific type may exist, it may not necessarily be applied consistently. For example, Al Qaeda, which is a high-profile terrorist group, is only marked as an Organization, rather than the more specific Terrorist_Group. This is one of the difficulties of working with the PIT data.

3. THE TERRORISTS DATASET

To use the PIT dataset for link classification, Zhao et al. construct a graph \( G \) consisting only of agents connected to other agents. They define four possible relationship labels between terrorists:

- **Colleague**: The agents are members of the same organization.
- **Family**: The agents are members of the same family.
- **Contact**: The agents have contacted one another.
- **Congregate**: The agents have used the same facility.

Links between agents may have any combination of the above labels, making this a multi-label classification problem. However, their approach is to train four separate binary classifiers, each testing the existence of a particular link type between two nodes; thus, the problem is easily considered as four link prediction problems.

In their experiments, Zhao et al. only train and test their models on links which have at least one label. They restrict their experiments to 851 such agent-agent edges. In this set, the four labels have the following base rates (positive):

- **Colleague**: 487 (53.1%)
- **Family**: 136 (14.8%)
- **Contact**: 114 (19.6%)
- **Congregate**: 180 (12.4%)

The examples indicate terrorist pairs using concatenations of unique URL identifiers. In order to construct an adjacency matrix for each task, we must first parse these identifiers into an ordered terrorist vocabulary. We find 244 unique terrorist identifiers, so the resulting adjacency matrix is \( 244 \times 244 \). We use 1 to represent a positive link, \(-1\) for a non-link, and 0 for unspecified or unknown links. Figure 1 is a visualization of the adjacency matrix for the **colleagues** label. Note that the matrix is symmetric, since a pair relationship does not depend on the order of the two terrorists. Also note that terrorists are not assumed to be colleagues with themselves, i.e. they are unspecified.

**Terrorists** contains an example set for each of the four classification tasks. An example contains, in addition to the URL identifiers and a binary label, 612 binary-valued attributes for each terrorist in the pair (for a total of 1224 attributes). These are extracted from the descriptive terrorist attributes in the PIT dataset, with some preprocessing required. For example, a portion of the attribute set comes from keyword analysis of the terrorist biographies.
4. THE WEST BANK DATASET

Similar to the PIT dataset, the West Bank dataset consists of entities of various types connected by detailed relationships. However, the dataset is much smaller, having only the following entity types and counts:

- **Task** (22): Objectives such as to raid, kidnap, arrest, or confiscate.
- **Resource** (59): Terrorist resources such as training camps, financial sources, and weapons.
- **Knowledge** (32): Sources of information, such as interviews, broadcasts, videotapes, and reports.
- **Agent** (47): Terrorists such as Osama Bin Laden and Saddam Hussein.
- **Location** (66): Political and geographic regions.
- **Organization** (38): Organizations such as Al-Qaeda, the Taliban, and Hezbollah.

Any entity may have a connection to any other entity, regardless of type. As with the PIT dataset, this network must be converted into one consisting only of agents connected to other agents. This leads to \( \binom{47}{2} = 1081 \) unique edges. Due to data sparsity, we only consider the task of determining whether two agents share the same location. There are 25 such connections, out of the possible 1081 edges.

Similar to *Terrorists*, we produce a dataset where an example represents an edge between two terrorists, and contains 34 attributes for each terrorist as well as a label. In *West Bank*, descriptive features for agents indicate the roles an agent has served, such as leader, senior, bodyguard, or suicide bomber.

5. APPROACH

The agent-agent networks we are considering consist of two main types of information: descriptive features for each terrorist, and structural network information. [1] describes models based on Relational Markov Networks for combining this information. The use of these models requires converting the agent-agent graph, \( G \), into a graph where a node exists for each pair of terrorists; thus, the edge-labeling problem becomes a node-labeling problem. Two nodes in the graph (two pairs of terrorists) are linked in the graph if they have any member in common. One drawback of this method is that the graphs become complex, requiring the use of approximation algorithms such as loopy belief propagation. Due to the presence a large number of small cliques, this results in poor approximations [1]. To address this issue, the authors try training a model on a reduced version of the graph, considering only cliques with three members, and this “triad” model performs better than their previous model on all four labeling tasks. However, the results still leave room for improvement.

We use a simple logistic regression model to combine the descriptive and structural information in the network. In order to extract structural features, we use low-rank approximation based on singular value decomposition (SVD). We then experiment using different transformations and combinations of the descriptive and structural features.

5.1. Logistic Regression and Descriptive Features

Given a feature vector \( x \) and weight vector \( w \), logistic regression computes the sigmoid function:

\[
f(z) = \frac{1}{1 + e^{-w \cdot x}}
\]

A threshold can then be used to yield a positive or negative classification. One of the benefits of logistic regression is that output ranges smoothly between 0 and 1 and accepts a domain of \((-\infty, \infty)\). Another benefit is that training is relatively straightforward. The goal is to find the weight vector that minimizes training loss:

\[
w = \underset{w \in \mathbb{R}^d}{\text{argmin}} \|w\|^2 + C \sum_{i=1}^{n} \text{loss}(f(x), y_i)
\]

where \( C \) is a regularization parameter. Training loss is defined by

\[
\text{loss} = -\log(p(x)) \quad \text{with} \quad p(x) = f(x) \text{ if } y_i = 1 \quad \text{and} \quad 1 - f(x) \text{ if } y_i = -1.
\]
However, because logistic regression is a linear classifier, the input features may require transformations for the best performance. Suppose we want to predict a link between terrorist $i$ and $j$. Let $\mathbf{ta}$ and $\mathbf{tb}$ be the descriptive features of terrorist $i$ and $j$, respectively, each with length $n$. We can simply concatenate these features and use them in a classifier. However, since we are using logistic regression, this will not capture interaction between the two terrorists’ features. We may, for instance, be interested not only in the region that each terrorist lives in, but whether or not they live in the same region. Then we can also try taking the pointwise, or element-wise, product of $\mathbf{ta}$ and $\mathbf{tb}$. Thus, we define the following two feature transformations:

- $\text{Tcon}: \langle \mathbf{ta}, \mathbf{tb} \rangle$
- $\text{Tpt}: \langle \mathbf{ta}_1 \mathbf{tb}_1, \mathbf{ta}_2 \mathbf{tb}_2, ..., \mathbf{ta}_n \mathbf{tb}_n \rangle$

We use combinations of both to determine their utility in the link prediction problem.

### 5.2. Extracting network information using low-rank approximation

We also need a way to extract network information into usable features for logistic regression. For an $m \times n$ matrix $A$, the Singular Value Decomposition (SVD) is the factorization

$$A = USV^\top,$$

where $U$ is an orthonormal $m \times m$ matrix, $V$ is an orthonormal $n \times n$ matrix, and $S$ is an $m \times n$ diagonal matrix of “singular values” sorted from highest value to lowest. These indicate the importance of columns in $U$ and $V$ in determining the output $A$.

The number of non-zero singular values is equal to the rank of $A$. However, by setting all but the top $k$ singular values in $S$ to zero, we can produce a rank-$k$ approximation

$$A \approx A' = US'V^\top,$$

where $S'$ is the modified singular value matrix. The low-rank approximation method highlights similarities and patterns in the original matrix, and has been used effectively for predicting missing values in matrices of movie ratings [2]. In our case, we are interested in predicting entries in a terrorist adjacency matrix. We use the following method described by [2]:

1. Impute unknown values in $A$ using the column averages$^4$ $\bar{c}_j$, where $j = 1, 2, ..., n$.

2. Subtract the row averages $\bar{r}_i$ from each row, creating a mean-centered matrix $B$.

3. Compute the SVD, $B = USV^\top$.

4. Compute the low-rank approximation, $B'$.

5. Add the corresponding row averages $\bar{r}_i$ back to $B'$ to yield the prediction matrix $P$.

The entries in $P$ are directly usable as predictions, since they estimate entries in $A_{i,j}$. However, since we are using logistic regression models, we may gain more expressive power by deriving more interesting features. First, factorize $B'$ as follows$^5$:

$$B' = US'V^\top = U\sqrt{S'}\sqrt{S'V^\top}.$$  

Let $U_s$ and $V_s^\top$ represent $U\sqrt{S'}$ and $\sqrt{S'V^\top}$, respectively. Then, consider the value of a particular prediction in $P$:

$$P_{ij} = \bar{r}_i + B'_{ij} = \bar{r}_i + \mathbf{p}_a \cdot \mathbf{p}_b$$

where $\mathbf{p}_a$ is the $i$th column of $U_s$ and $\mathbf{p}_b$ is the $j$th column of $V_s$. We can then construct a feature vector using the pointwise, or element-wise, product of $\mathbf{p}_a$ and $\mathbf{p}_b$:

$$\langle \bar{r}_i, \mathbf{pa}_1 \mathbf{pb}_1, \mathbf{pa}_2 \mathbf{pb}_2, ..., \mathbf{pa}_n \mathbf{pb}_n \rangle$$

We also include the row average as a feature of the vector so a logistic regression classifier can recreate the original prediction $P_{ij}$. This occurs if all features are given weight 1. However, if we are using the features in conjunction with features about the terrorist entities in a logistic regression model, it may be that some features of the low-rank decomposition give information that overlaps with the entity features. We would want to give these features less weight in the model, to emphasize the unique features.

Thus, when considering two terrorists $i$ and $j$, we have two additional feature sets to use in our logistic regression models:

- $\text{SVD-dot}: P_{ij}$

- $\text{SVD-pt}: \langle \bar{r}_i, \mathbf{pa}_1 \mathbf{pb}_1, \mathbf{pa}_2 \mathbf{pb}_2, ..., \mathbf{pa}_n \mathbf{pb}_n \rangle$

Again, because the effect of features is effectively additive in a logistic regression classifier, a simple concatenation of the features, i.e. $\langle \mathbf{pa}, \mathbf{pb} \rangle$, will not be very useful. It would only capture the individual propensities of terrorist $i$ and terrorist $j$ to have links; each feature is equivalent to the next, so it gives no more information than a simple degree count for each terrorist.

$^4$In some columns, there are no ratings available in the training set, so $\bar{c}_j$ cannot be computed. In this case, we use the average of the column averages.

$^5$Note that $S'$ is a square matrix since $m = n$. 

4
6. EXPERIMENTS

We use 10-fold stratified cross validation for our experiments. We run cross validation for multiple values of the logistic regression regularization parameter, $C$, in order to find the best setting for each model. We perform z-normalization of each feature, then train and test the logistic regression model under evaluation. Performance is measured by both AUC and accuracy. We also determine the optimal rank $k$ for the low-rank approximation, according to mean absolute error.

We perform two main experiments. First, we want to determine the most useful combination of feature sets from $T_{con}$, $T_{pt}$, $SVD-dot$, and $SVD-pt$. There are a total of 15 possible combinations, but because $SVD-pt$ contains all the information in $SVD-dot$, we omit combinations which use both. We are left with 11 different feature sets and, hence, 11 logistic regression models.

For this experiment, we focus on the colleague prediction task for two reasons: first, it is the most balanced, with 54% positive examples. Second, it presents the most room for improvement, with [1] achieving less than 88% accuracy. We also perform the experiment on the West Bank dataset, not only to compare models but to also test the feasibility of the dataset.

Our second experiment is to compare our best models with the triad RMN model in [1]. For this, we consider all four prediction tasks.

7. RESULTS

7.1. Optimal Rank $k$

For the low-rank decomposition, we must first determine the rank parameter $k$ to be used. We do this using 10-fold cross validation. We split the known examples into training and test folds; for each training fold, we populate $A$ with the known examples and impute the rest of the values, compute the low-rank approximation, create the prediction matrix $P$, and measure the mean absolute error (MAE) between the original matrix $A$ and the resulting prediction $P$ for entries in the corresponding test set.

For the colleagues task in the Terrorists dataset, Figure 2 shows the MAE as a function of the rank $k$; the value which minimizes MAE is $k = 9$. We choose this value for all four tasks.

For the West Bank dataset, the optimal rank was $k = 1$, shown in Figure 3. This suggests that higher values were prone to overfitting, likely due to the small amount of data.

7.2. Comparison with RMN Triad

Figure 4 shows that $SVD-dot$ and $T_{con} SVD-pt$ outperform the RMN triad model accuracy on all tasks, and by over
10% on some tasks. Interestingly, the SVD-dot yields better accuracy only on the congregate task. The family task is less useful because it is too easy; each model reaches nearly 99% accuracy.

RMN performance in terms of AUC is unknown, but Figure 5 compares our three models. The rankings match on all tasks except family, for which Tcon performs the best. Because all models had nearly the same accuracy, AUC is more meaningful here. This suggests that the terrorist descriptive features were better at separating positive and negative examples.

7.3. Comparison of Feature Set Combinations

Figure 6 shows that Tpt SVD-pt yields the highest AUC on the colleagues classification task. In general, those with the SVD-pt feature set performed better than those with SVD-dot, which itself performed better than combinations of only descriptive features. Another interesting result is that Tpt offers little information when used alone, and offers virtually no unique information if other attributes are also being used in the model.

Results on the West Bank dataset, in Figure 7 are similar. Again, the SVD-based feature sets yield much better performance than those created from the descriptive features. Also, the Tpr feature set is not very useful. In this dataset, there is little distinction between the SVD-dot and SVD-pt feature sets because they were composed from rank-1 approximations. However, they are not identical; SVD-dot consists of \( \langle P_{ij} \rangle \), and SVD-pt consists of \( \langle \tilde{r}_i, B'_{ij} \rangle \). Finally, note that all accuracies are only slightly better than the base negative rate of the dataset, 97.7%.

8. REVISITING MODEL PREDICTIONS IN 2010

The Terrorist dataset was created by Zhao et al. in 2006. Since then, the PIT dataset has been updated. For example, as of this writing there are 539 Terrorists, up from 435. The counts of various terrorist attack types have also changed significantly from those reported by [1]; for example, the number of Bombings has fallen by 68, but the number of NBCR Attacks (Nuclear, Biological, Chemical, or Radioactive attacks) has risen by 34. The cause of these updates may be from new information, from data loss, or perhaps information censorship.
Following the process explained in [1], we created a colleagues label graph from the updated PIT dataset. We found 9 new links and 130 missing links. This presents an opportunity: if our best model predicted any of these changes with a likelihood greater than random, then this would provide additional evidence of the models utility.

Consider this confusion matrix generated by the Tcon SVD-pt model on the colleagues dataset:

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted T</td>
<td>346</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>44</td>
<td>436</td>
</tr>
</tbody>
</table>

That is, on the positive examples, our model is correct 89% of the time, and 95% for the negative examples. Now, if we look at the confusion matrix for just those examples that, in the 2010 version of PIT, have had their labels flipped, we would expect to see these same recall rates if the selection of the flipped examples was random with respect to the models predictions. If the rates are worse, then the changes were biased toward examples which the model got wrong, which means the model will soon be made correct. Conversely, if the rates are better, then the changes were biased toward examples which the model got right, and the model will soon be made incorrect.

The actual confusion matrix for these soon-to-flip examples is below:

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted T</td>
<td>120</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>3</td>
</tr>
</tbody>
</table>

Thus, the model did not predict the disappearing links, but it appears it may have done well in predicting the appearing links. Unfortunately, the small sample size limits the conclusions we can draw.

9. LIMITATIONS AND CONCLUSIONS

There are a few details about our experimental process which might lead to slightly optimistic results. To alleviate the computational intensity of the experiments, we do not perform cross validation on the parameter search optimization process. Similarly, we do not perform cross validation on the rank $k$ optimization process. The selection of the best parameters may be overfitting to the entire dataset, and not generalize as well to additional data. However, this still allows for a meaningful comparison of the different approaches, including the RMN triad model, which is also not guaranteed to be avoiding overfitting to the dataset.

Overall, our experiments demonstrate the efficacy of the low-rank approximation method as well as the added benefit of using derived features in a logistic regression model. $Tcon$ does fairly well in isolation, and does in general add some performance gain to the SVD-based features, but there is a significant amount of overlap in the Terrorists dataset.

The PIT dataset is a rich source of information, but performing machine learning tasks with it is troublesome. Sparse features and inconsistent labeling make it difficult to ensure that the correct information is being extracted. However, results do show that the classification tasks considered with the Terrorists dataset are non-trivial and contain exploitable structure and information.

Finally, the West Bank dataset provided an interesting second dataset, and does support the general results from the Terrorists dataset. However, future work is better spent finding larger, more complex datasets.

10. PERSONAL CONTRIBUTIONS

Our shared work includes the initial research, the processing of the Terrorists data, the training of the models, and the analysis of the results; these were split fairly evenly. Individually, I also took the tasks of parsing the West Bank and PIT datasets, training the models for the West Bank dataset, and investigating the results.

11. REFERENCES
