Understanding the relationships between biomedical terms like viruses, drugs, and symptoms is essential in the fight against diseases. Many attempts have been made to introduce the use of machine learning to the scientific process of hypothesis generation (HG), which refers to the discovery of meaningful implicit connections between biomedical terms. However, most existing methods fail to truly capture the temporal dynamics of scientific term relations and also assume unobserved connections to be irrelevant (i.e., in a positive-negative (PN) learning setting). To break these limits, we formulate this HG problem as future connectivity prediction task on a dynamic attributed graph via positive-unlabeled (PU) learning. Then, the key is to capture the temporal evolution of node pair (term pair) relations from just the positive and unlabeled data. We propose a variational inference model to estimate the positive prior, and incorporate it in the learning of node pair embeddings, which are then used for link prediction. Experiment results on real-world biomedical term relationship datasets and case study analyses on a COVID-19 dataset validate the effectiveness of the proposed model.

1 Introduction

Recently, the study of co-relationships between biomedical entities is increasingly gaining attention. The ability to predict future relationships between biomedical entities like diseases, drugs, and genes enhances the chances of early detection of disease outbreaks and reduces the time required to detect probable disease characteristics. For instance, in 2020, the COVID-19 outbreak pushed the world to a halt with scientists working tediously to study the disease characteristics for containment, cure, and vaccine. An increasing number of articles encompassing new knowledge and discoveries from these studies were being published daily [1]. However, with the accelerated growth rate of publications, the manual process of reading to extract undiscovered knowledge increasingly becomes a tedious and time-consuming task beyond the capability of individual researchers.

In an effort towards an advanced knowledge discovery process, computers have been introduced to play an ever-greater role in the scientific process with automatic hypothesis generation (HG). The study of automated HG has attracted considerable attention in recent years [41, 25, 45, 47]. Several previous works proposed techniques based on association rules [25, 18, 47], clustering and topic modeling [45, 44, 5], text mining [43, 42], and others [28, 49, 39]. However, these previous works fail to truly utilize the crucial information encapsulated in the dynamic nature of scientific discoveries and assume that the unobserved relationships denote a non-relevant relationship (negative).

To model the historical evolution of term pair relations, we formulate HG on a term relationship graph $G = \{V, E\}$, which is decomposed into a sequence of attributed graphlets $G = \{G^1, G^2, ..., G^T\}$, where the graphlet at time $t$ is defined as,
We highlight the contributions of this work as follows. We address the HG problem by modeling how the observed co-occurrence between two terms is determined by application needs. In PN learning, the empirical risk minimizer. Given an input representation, the goal is to minimize the expected classification risk to obtain an empirical risk minimizer. In recent years, the use of unbiased risk estimator has gained attention. Completely at Random) assumption since the noise for negative samples is constant. However, estimating this probability value from the positive-unlabeled data is nontrivial. We propose a variational inference model to estimate the positive prior and incorporate it in the learning of node pair embeddings, which are then used for link prediction (hypothesis generation).

We highlight the contributions of this work as follows.

1) Methodology: we propose a PU learning approach on temporal graphs. It differs from other existing approaches that learn in a conventional PN setting on static graphs. In addition, we estimate the positive prior via a variational inference model, rather than setting by prior knowledge.

2) Application: to the best of our knowledge, this is the first application of PU learning on the HG problem, and on dynamic graphs. We applied the proposed model on real-world graphs of terms in scholarly publications published from 1945 to 2020. Each of the three graphs has around 30K nodes and 1-2 million edges. The model is trained end-to-end and shows superior performance on HG. Case studies demonstrate our new and valid findings of the positive relationship between medical terms, including newly observed terms that were not observed in training.

2 Related Work of PU Learning

In PU learning, since the negative samples are not available, a classifier is trained to minimize the expected misclassification rate for both the positive and unlabeled samples. One group of study proposed a two-step solution: 1) identifying reliable negative samples, and 2) learning a classifier based on the labeled positives and reliable negatives using a (semi)-supervised technique. Another group of studies considered the unlabeled samples as negatives with label noise. Hence, they place higher penalties on misclassified positive examples or tune a hyperparameter based on suitable PU evaluation metrics. Such a proposed framework follows the SCAR (Selected Completely at Random) assumption since the noise for negative samples is constant.

PU Learning via Risk Estimation Recently, the use of unbiased risk estimator has gained attention. The goal is to minimize the expected classification risk to obtain an empirical risk minimizer. Given an input representation and the loss function calculating the incurred loss of predicting an output when the true value is $y$. Function $l$ has a variety of forms, and is determined by application needs. In PN learning, the empirical risk minimizer is obtained by minimizing the PN risk w.r.t. a class prior of $\pi_p$:

$$\hat{R}(f) = \pi_p \hat{R}_p(f) + \pi_N \hat{R}_N(f),$$

where $\pi_N = 1 - \pi_p$, $\hat{R}_p(f) = \frac{1}{n_p} \sum_{i=1}^{n_p} l(f(h^p_i), +1)$ and $\hat{R}_N(f) = \frac{1}{n_N} \sum_{i=1}^{n_N} l(f(h^N_i), -1)$. The variables $n_p$ and $n_N$ are the numbers of positive and negative samples, respectively.
Algorithm 1 by

\[ \hat{\pi}_N R_N^{-1}(f) = \hat{R}_U^{-1} - \pi_p \hat{R}_P^{-1}(f), \]

where \( \hat{R}_U^{-1} = \mathbb{E}_{h \sim p(h)}[l(f(h), -1)] \) and \( \hat{R}_P^{-1} = \mathbb{E}_{h \sim p(h_{y=+1})}[l(f(h), -1)] \). Furthermore, the classification risk can then be approximated by:

\[ \hat{R}_{PU}(f) = \pi_p \hat{R}_P^{-1}(f) + \hat{R}_U^{-1}(f) - \pi_p \hat{R}_P^{-1}(f), \]

where \( \hat{R}_P^{-1}(f) = \frac{1}{\pi_p} \sum_{i=1}^{n}\{l(f(h^i_1), -1), \hat{R}_U^{-1}(f) = \frac{1}{n_\mathcal{U}} \sum_{i=1}^{n}l(f(h^i_U), -1), \) and \( n_\mathcal{U} \) is the number of unlabeled data sample. To obtain an empirical risk minimizer \( \hat{f}_{PU} \) for the PU learning framework, \( \hat{R}_{PU}(f) \) needs to be minimized. Kiryo et al. noted that the model tends to suffer from overfitting on the training data when the model \( f \) is made too flexible \[29\]. To alleviate this problem, the authors proposed the use of non-negative risk estimator for PU learning:

\[ \hat{R}_{PU}(f) = \pi_p \hat{R}_P^{-1}(f) + \max\{0, \hat{R}_U^{-1} - \pi_p \hat{R}_P^{-1}(f)\}. \]

It works in fact by explicitly constraining the training risk of PU to be non-negative. The key challenge in practical PU learning is the unknown of prior \( \pi_P \).

Prior Estimation The knowledge of the class prior \( \pi_P \) is quintessential to estimating the classification risk. In PU learning for our node pairs, we represent a sample as \( \{h, s, y\} \), where \( h \) is the node pair representation (to be learned), \( s \) indicates if the pair relationship is observed (labeled, \( s = 1 \)) or unobserved (unlabeled, \( s = 0 \)), and \( y \) denotes the true class (positive or negative). We have only the positive samples labeled: \( p(y = 1 | s = 1) = 1 \). If \( s = 0 \), the sample can belong to either the positive or negative class. PU learning runs commonly with the Selected Completely at Random (SCAR) assumption, which postulates that the labeled sample set is a random subset of the positive sample set \[16, 6, 8\]. The probability of selecting a positive sample to observe can be denoted as: \( p(s = 1 | y = 1, h) \). The SCAR assumption means: \( p(s = 1 | y = 1, h) = p(s = 1 | y = 1) \). However, it is hard to estimate \( \pi_P = p(y = 1) \) with only a small set of observed samples \( (s = 1) \) and a large set of unobserved samples \( (s = 0) \) \[7\]. Solutions have been tried by i) estimating from a validation set of a fully labeled data set (all with \( s = 1 \) and knowing \( y = 1 \) or \( -1 \)) \[29, 10\]; ii) estimating from the background knowledge; and iii) estimating directly from the PU data \[16, 6, 8, 27, 14\]. In this paper, we focus on estimating the prior directly from the PU data. Specifically, unlike the other methods, we propose a scalable method based on deep variational inference to jointly estimate the prior and train the classification model end-to-end. The proposed deep variational inference uses KL-divergence to estimate the parameters of class mixture model distributions of the positive and negative class in contrast to the method proposed in \[14\] which uses penalized L1 divergences to assign higher penalties to class priors that scale the positive distribution more than the total distribution.

3 PU learning on Temporal Attributed Networks

3.1 Model Design

The architecture of our Temporal Relationship Predictor (TRP) model is shown in Fig. 1. For a given pair of nodes \( a^{ij} = \langle v_i, v_j \rangle \) in any temporal graphlet \( G^t \), the main steps used in the training process of TRP for calculating the connectivity prediction score \( p^t(a^{ij}) \) are given in Algorithm 1. The testing process also uses the same Algorithm 1 (with \( t=\tau \)), calculating \( p^t(a^{ij}) \) for node pairs that have not been connected in \( G^{\tau-1} \). The connectivity prediction score is calculated in line 6 of Algorithm 1 by \( p^t(a^{ij}) = f_C(h^t_{a^{ij}}; \theta_C) \), where \( \theta_C \) is the classification network parameter, and the embedding vector \( h^t_{a^{ij}} \) for the pair \( a^{ij} \) is iteratively updated in lines 1-5. These iterations of updating \( h^t_{a^{ij}} \) are shown as the recurrent structure in Fig. 1 (a), followed by the classifier \( f_C(\cdot; \theta_C) \).

The recurrent update function \( h^\tau_{a^{ij}} = f_A(h^\tau_{a^{ij}}; z^\tau_{v_i}, z^\tau_{v_j}; \theta_A), \tau = 1...t, \) in line 4 is shown in Fig. 1 (b), and has a Gated recurrent unit (GRU) network at its core.

\[ \mathcal{P} = \sigma_g(W^z f_m(z^\tau_{v_i}, z^\tau_{v_j}) + U^p h^\tau_{a^{ij}}^{-1} + b^p), \]

\[ r = \sigma_g(W^r f_m(z^\tau_{v_i}, z^\tau_{v_j}) + U^r h^\tau_{a^{ij}}^{-1} + b^r), \]

\[ \bar{h}^\tau_{a^{ij}} = \sigma_h(W^f f_m(z^\tau_{v_i}, z^\tau_{v_j}) + r \circ U h^\tau_{a^{ij}}^{-1} + b), \]

\[ h^\tau_{a^{ij}} = \mathcal{P} \circ \bar{h}^\tau_{a^{ij}} + (1 - \mathcal{P}) \circ h^\tau_{a^{ij}}^{-1}. \]
Algorithm 1: Calculate the future connection score for term pairs $a^{i,j} = <v_i, v_j>$

Input: $G = \{G^1, G^2, \ldots, G^T\}$ with node feature $x^t_v$, a node pair $a^{i,j} = <v_i, v_j> \in G^t$, and an initialized pair embedding vector $h_{a^{i,j}}^{0}$ (e.g., by zeros) 

Result: $p_{a^{i,j}}^{t}$ the connectivity prediction score for the node pair $a^{i,j}$

1. for $\tau \leftarrow 1 \cdots T$
2. Obtain the current node feature $x^{\tau}_v$ $(v = v_i, v_j)$ of both nodes (terms) $v_i, v_j$; as well as $x^{\tau}_{Nr(v)}$ $(v = v_i, v_j)$ for the node feature of sampled neighboring nodes for $v_i, v_j$;
3. Aggregate the neighborhood information of node $v = v_i, v_j$, $z^{\tau}_v = f_G(x^{\tau}_v, x^{\tau}_{Nr(v)}; \theta_G)$;
4. Update the embedding vector for the node pair $h^{\tau}_{a^{i,j}} = f_A(h^{\tau-1}_{a^{i,j}}, z^{\tau}_v, z^{\tau}_{v_j}; \theta_A)$;

5. end
6. Return $p_{a^{i,j}}^{t} = f_C(h^{t}_{a^{i,j}}; \theta_C)$

where $\circ$ denotes element-wise multiplication, $\sigma$ is a nonlinear activation function, and $f_m(.)$ is an aggregation function. In this study, we use a max pool aggregation. The variables $\{W, U\}$ are the weights. The inputs to the function $f_A$ include: $h^{\tau-1}_{a^{i,j}}$, the embedding vector in previous step; $\{z^{\tau}_v, z^{\tau}_{v_j}\}$, the representation of node $v_i, v_j$ after aggregation their neighborhood, $z^{\tau}_v = f_G(x^{\tau}_v, x^{\tau}_{Nr(v)}; \theta_G)$, given in line 3. The aggregation function $f_G$ takes input the node feature $x^{\tau}_v$, and the neighboring node feature $x^{\tau}_{Nr(v)}$ and goes through the aggregation block shown in Fig. 1(c). The aggregation network $f_G(\theta_G)$ is implemented following GraphSAGE [21], which is one of the most popular graph neural networks for aggregating node and its neighbors.

The loss function in our problem $l(p^t(a^{i,j}), y)$ evaluates the loss incurred by predicting a connectivity $p^t(a^{i,j}) = f_C(h^{t}_{a^{i,j}}; \theta_C)$ when the ground truth is $y$. For constructing the training set for PU learning in the dynamic graph, we first clarify the label notations. For one pair $a^{i,j}$ from a graph $G^t$, its label $y^{ij}_t = 1 \ (\text{positive})$ if the two nodes have a link observed in $G^{t+1}$ (they have an edge $e^{t+1}$, observed in next time step), i.e., $s^{i,j}_t = 1$. Otherwise when no link is observed between them in $G^{t+1}$, $a^{i,j}$ is unlabeled, i.e., $s^{i,j}_t = 0$, since $y^{ij}_t$ can be either 1 or -1. Since we consider insertion
only graphlets sequence, $V^1 \subseteq V^2, ..., \subseteq V^T$ and $E^1 \subseteq E^2, ..., \subseteq E^T$, $y^{ij}_t = 1$ maintains for all future steps after $t$ (once positive, always positive). At the final step $t = T$, all pairs with observed connections already have $y^{ij}_T = 1$, our objective is to predict the connectivity score for those pairs with $s^{ij}_T = 0$. Our loss function is defined following the unbiased risk estimator in Eq. (3),

$$L^R = \pi_P \hat{R}_P^+(f_C) + \hat{R}_U(f_C) - \pi_P \hat{R}_P(f_C),$$

where $\hat{R}_P^+(f_C) = \frac{1}{|\mathcal{D}_P|} \sum_{a^{ij} \in \mathcal{D}_P} 1/(1 + \exp(p_t^{ij})), \hat{R}_U(f_C) = \frac{1}{|\mathcal{D}_U|} \sum_{a^{ij} \in \mathcal{D}_U} 1/(1 + \exp(-p_t^{ij}))$, and $\hat{R}_P(f_C) = \frac{1}{|\mathcal{D}_P|} \sum_{a^{ij} \in \mathcal{D}_P} 1/(1 + \exp(p_t^{ij}))$ with the positive samples $\mathcal{D}_P$ and unlabeled samples $\mathcal{D}_U$, when taking $l$ as sigmoid loss function. $L^R$ can be adjusted with the non-negative constraint in Eq. (4), with the same definition of $\hat{R}_P^+(f), \hat{R}_U$, and $\hat{R}_P(f)$.

### 3.2 Prior Estimation

The positive prior $\pi_P$ is a key factor in $L^R$ to be addressed. The samples we have from $G$ are only positive $\mathcal{D}_P$ and unlabeled $\mathcal{D}_U$. Due to the absence of negative samples and of prior knowledge, we present an estimate of the class prior from the distribution of $h$, which is the pair embedding from $f_A$. Without loss of generality, we assume that the learned $h$ of all samples has a Gaussian mixture distribution of two components, one is for the positive samples, while the other is for the negative samples although they are unlabeled. The mixture distribution is parameterized by $\beta$, including the mean, co-variance matrix and mixing coefficient of each component. We learn the mixture distribution using stochastic variational inference [24] via the “Bayes by Backprop” technique [9]. The use of variational inference has been shown to have the ability to model salient properties of the data generation mechanism and avoid singularities. The idea is to find variational distribution variables $\theta^*$ that minimizes the Kullback-Leibler (KL) divergence between the variational distribution $q(\beta|\theta)$ and the true posterior distribution $p(\beta|h)$:

$$\theta^* = \arg\min_\theta L^E,$$

where, $L^E = KL(q(\beta|\theta)||p(\beta|h)) = KL(q(\beta|\theta)||p(\beta)) - \mathbb{E}_{q(\beta|\theta)}[\log p(h|\beta)]$.

The resulting cost function $L^E$ on the right of Eq. (7) is known as the (negative) “evidence lower bound” (ELBO). The second term in $L^E$ is the likelihood of $h$ fitting to the mixture Gaussian with parameter $\beta$: $\mathbb{E}_{q(\beta|\theta)}[\log p(h|\beta)]$, while the first term is referred to as the complexity cost [9]. We optimize the ELBO using stochastic gradient descent. With $\theta^*$, the positive prior is then estimated as

$$\pi_P = q(\beta^*_i|\theta^*), i = \arg\max_{k=1,2} |C_k|$$

where $C_1 = \{h \in \mathcal{D}_P, p(h|\beta^1) > p(h|\beta^2)\}$ and $C_2 = \{h \in \mathcal{D}_P, p(h|\beta^2) > p(h|\beta^1)\}$.

### 3.3 Parameter Learning

To train the three networks $f_A(:,\theta_A), f_G(:,\theta_G), f_C(:,\theta_C)$ for connectivity score prediction, we jointly optimize $L = \sum_{t=1}^T L^R_t + L^E_t$, using Adam over the model parameters. Loss $L^R$ is the PU classification risk as described in section 3.1, and $L^E$ is the loss of prior estimation as described in section 3.2. Note that during training, $y^{ij}_a = y^{ij}_{a-1}$ since we do not observe $G^T$ in training. This is to enforce prediction consistency.

### 4 Experimental Evaluation

#### 4.1 Dataset and Experimental Setup

The graphs on which we apply our model are constructed from the title and abstract of papers published in the biomedical fields from 1949 to 2020. The nodes are the biomedical terms, while the edges linking two nodes indicate the co-occurrence of the two terms. Note that we focus only on the co-occurrence relation and leave the polarity of the relationships for future study. To evaluate the model’s adaptivity in different scientific domains, we construct three graphs from papers relevant to COVID-19, Immunotherapy, and Virology. The graph statistics are shown in Table 1. To set up the
We measure the performance using four different metrics. These metrics are the Macro-F1 score, F1 score of observed connections (F1-S), F1 score adapted to PU learning (F1-P) [7, 30], and the label ranking average precision score (LRAP), where the goal is to give better rank to the positive node pairs. In all metrics used, higher values are preferred.

At each \( t = 1, ..., T - 1 \), graph \( G^t \) is incrementally updated from \( G^{t-1} \) by adding new nodes (biomedical terms) and their links. For the node feature vector \( x_v^t \), we extract its term description and convert to a 300-dimensional feature vector by applying the latent semantic analysis (LSI). The missing term and context attributes are filled with zero vectors. If this node already exists before time \( t \), the context features are updated with the new information about them in discoveries, and publications. In the inference (testing) stage, the new nodes in \( G^T \) are only presented with their feature vectors \( x_v^T \). The connections to these isolated nodes are predicted by our TRP model.

We implement TRP using the Tensorflow library. Each GPU based experiment was conducted on an Nvidia 1080TI GPU. For each neural network based model, we performed a grid search over the learning rate \( lr = \{1e-2, 5e-3, 1e-3, 5e-2\} \). For the prior estimation, we adopted Gaussian, square-root inverse Gamma, and Dirichlet distributions to model the mean, co-variance matrix and mixing coefficient variational posteriors respectively.

### 4.2 Comparison Methods and Performance Matrices

We evaluate our proposed TRP model in several variants and by comparing with several competitors:

1) **TRP variants:** a) TRP-PN - the same framework but in PN setting (i.e., treating all unobserved samples as negative, rather than unlabeled); b) TRP-nnPU - trained using the non-negative risk estimator Eq. (4) or the equation defined in section 3.1 for our problem; and c) TRP-uPU - trained using the unbiased PU risk estimator Eq. (3), the equation defined in section 3.1 for our problem. The comparison of these variants will show the impact of different risk estimators.

2) **SOTA PU learning:** the state-of-the-art (SOTA) PU learning methods taking input \( h \) from the SOTA node embedding models, which can be based on LSI [11], node2vec [20], DynAE [19] and GraphSAGE [21]. Since node2vec learns only from the graph structure, we concatenate the node2vec embeddings with the text (term and context) attributes to obtain an enriched node representation. Unlike our TRP that learns \( h \) for one pair of nodes, these models learn embedding vectors for individual nodes. Then, \( h \) of one pair from baselines is defined as the concatenation of the embedding vector of two nodes. We observe from the results that a concatenation of node2vec and LSI embeddings had the most competitive performance compared to others. Hence we only report the results based on concatenated embeddings for all the baselines methods. The used SOTA PU learning methods include [16] by reweighting all examples, and models with different estimation of the class prior such as SAR-EM [8] (an EM-based SAR-PU method), SCAR-KM2 [38], SCAR-C [8], SCAR-TkE [6], and pen-L1 [14].

3) **Supervised:** weaker but simpler logistic regression applied also \( h \).

We measure the performance using four different metrics. These metrics are the Macro-F1 score (F1-M), F1 score of observed connections (F1-S), F1 score adapted to PU learning (F1-P) [7, 30], and the label ranking average precision score (LRAP), where the goal is to give better rank to the positive node pairs. In all metrics used, higher values are preferred.

### 4.3 Evaluation Results

Table 2 shows that TRP-uPU always has the superior performance over all other baselines across the datasets due to its ability to capture and utilize temporal, structural, and textual information (learning better \( h \)) and also the better class prior estimator. Among TRP variants, TRP-uPU has higher or equal F1 values comparing to the other two, indicating the benefit of using the unbiased risk estimator. On the LRAP scores, TRP-uPU and TRP-PN have the same performance on promoting the rank of the positive samples. Note that the results in Table 2 are from the models trained with their best learning rate, which is an important parameter that should be tuned in gradient-based optimizer, by either exhaustive search or advanced auto-machine learning [35]. To further investigate the performance of
Table 1: Three graph dataset statistics, with their number of nodes and edges

<table>
<thead>
<tr>
<th></th>
<th>#nodes</th>
<th>#edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>COVID-19</td>
<td>27,325</td>
<td>2,474,624</td>
</tr>
<tr>
<td>Immunotherapy</td>
<td>28,823</td>
<td>919,004</td>
</tr>
<tr>
<td>Virology</td>
<td>38,956</td>
<td>1,117,118</td>
</tr>
</tbody>
</table>

Table 2: Evaluation results on the COVID-19, Immuniotherapy and Neurology datasets, respectively

<table>
<thead>
<tr>
<th></th>
<th>COVID-19</th>
<th>Virology</th>
<th>Immunotherapy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F1-S</td>
<td>F1-M</td>
<td>LRAP</td>
</tr>
<tr>
<td>Supervised</td>
<td>0.82</td>
<td>0.86</td>
<td>1.73</td>
</tr>
<tr>
<td>SCAR-C [8]</td>
<td>0.82</td>
<td>0.86</td>
<td>1.73</td>
</tr>
<tr>
<td>SCAR-KM [19]</td>
<td>0.76</td>
<td>0.82</td>
<td>1.52</td>
</tr>
<tr>
<td>SCAR-TEG [9]</td>
<td>0.77</td>
<td>0.80</td>
<td>1.51</td>
</tr>
<tr>
<td>SAR-EM [10]</td>
<td>0.78</td>
<td>0.85</td>
<td>1.68</td>
</tr>
<tr>
<td>Elkan [15]</td>
<td>0.82</td>
<td>0.86</td>
<td>1.74</td>
</tr>
<tr>
<td>TRP-PN</td>
<td>0.84</td>
<td>0.87</td>
<td>1.80</td>
</tr>
<tr>
<td>pen.l-nnPU</td>
<td>0.71</td>
<td>0.70</td>
<td>1.35</td>
</tr>
<tr>
<td>TRP-uuNU</td>
<td>0.80</td>
<td>0.82</td>
<td>1.88</td>
</tr>
<tr>
<td>pen.l-uuPU</td>
<td>0.85</td>
<td>0.88</td>
<td>1.86</td>
</tr>
<tr>
<td>TRP-uPU</td>
<td>0.86</td>
<td>0.88</td>
<td>1.88</td>
</tr>
</tbody>
</table>

TRP variants, we show in Figure 2 their F1-S at different learning rate in trained from 1 to 10 epochs. We notice that TRP-uPU has a stable performance across different epochs and learning rates. This advantage is attributed to the unbiased PU risk estimation, which learns from only positive samples with no assumptions on the negative samples. We also found interesting that nnPU was worse than uPU in our experimental results. However, it is not uncommon for uPU to outperform nnPU in evaluation with real-world datasets. Similar observations were found in the results in [14, 17]. In our case, we attribute this observation to the joint optimization of the loss from the classifier and the prior estimation. Specifically, in the loss of uPU (Eq. (3)), \( \pi_{P} \) affects both \( \hat{R}_{P}(f) \) and \( \hat{R}_{U}(f) \). However, in the loss of nnPU (Eq. (4)), \( \pi_{P} \) only weighted \( \hat{R}_{P}(f) \) when \( \hat{R}_{U} - \pi_{P}\hat{R}_{P}(f) \) is negative. In real-world applications, especially when the true prior is unknown, the loss selection affects the estimation of \( \pi_{P} \), and thus the final classification results, TRP-PN is not as stable as TRP-uPU due to the strict assumption of unobserved samples as negative.

Figure 2: Stability comparison of TRP-PN, TRP-nnPU and TRP-uPU, showing the F1-S performance of the models (Y-axis) with different learning rates (X-axis) on 10 epochs.
4.4 Incremental Prediction

In Figure 3, we compare the performance of the top-performing PU learning methods on different year splits. We train TRP and other baseline methods on data until $t-1$ and evaluate its performance on predicting the testing pairs in $t$. It is expected to see performance gain over the incremental training process, as more and more data are used. We show F1-P due to the similar pattern on other metrics. We observe that the TRP models display an incremental learning curve across the three datasets and outperformed all other models.

![Figure 3: F1-P per year. The models are incrementally trained with data before the evaluation year.](image)

4.5 Qualitative Analysis

We conduct qualitative analysis of the results obtained by TRP-uPU on the COVID-19 dataset. This investigation is to qualitatively check the meaningfulness of the paired terms, e.g., can term covid-19 be paired meaningfully with other terms. We designed two evaluations. First, we set our training data until 2015, i.e., excluding the new terms in 2016-2020 in the COVID-19 graph, such as covid-19, sars-cov-2. The trained model then predicts the connectivity between covid-19 as a new term and other terms, which can be also a new term or a term existing before 2015. Since new terms like covid-19 were not in training graph, their term feature were initialized as defined in Section 4.1. The top predicted terms predicted to be connected with covid-19 are shown in Table 3 top, with the verification in COVID-19 graph of 2016-2020. We notice that the top terms are truly relevant to covid-19, and we do observe their connection in the evaluation graph. For instance, Cough, Fever, SARS, Hand (washing of hands) were known to be relevant to covid-19 at the time of writing this paper.

In the second evaluation, we trained the model on the full COVID-19 data ($\leq 2020$) and then predict to which terms covid-19 will be connected, but they haven’t been connect yet in the graph until 2020. We show the results in Table 3 bottom, and verified the top ranked terms by manually searching the recent research articles online. We did find there exist discussions between covid-19 and some top ranked terms, for example, [3] discusses how covid-19 affected the market of Chromium oxide and [23] discusses about caring for people living with Hepatitis B virus during the covid-19 spread.

4.6 Pair Embedding Visualization

We further analyze the node pair embedding learned by TRP-uPU on the COVID-19 data by visualizing them with t-SNE. To have a clear visibility, we sample 800 pairs and visualize the learned embeddings in Figure 3. We denote with colors the observed labels in comparison with the predicted

---

1 Dataset used in this analysis was downloaded in early March 2020 from https://www.semanticscholar.org/cord19/download
Table 3: Top ranked terms predicted to be connected with term *covid-19*, trained until 2015 (the top table) and until 2020 (the bottom table). Verification of existence (Ex) was conducted in the graph in 2020 when trained until 2015, and by manual search otherwise. Sc is the predicted connectivity score.

<table>
<thead>
<tr>
<th>Terms</th>
<th>Sc</th>
<th>Ex</th>
<th>Terms</th>
<th>Sc</th>
<th>Ex</th>
<th>Terms</th>
<th>Sc</th>
<th>Ex</th>
<th>Terms</th>
<th>Sc</th>
<th>Ex</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leukocyte count</td>
<td>0.98</td>
<td>Yes</td>
<td>Air</td>
<td>0.85</td>
<td>Yes</td>
<td>Infection control</td>
<td>0.96</td>
<td>Yes</td>
<td>Serum</td>
<td>0.84</td>
<td>Yes</td>
</tr>
<tr>
<td>Fever</td>
<td>0.94</td>
<td>Yes</td>
<td>Lung</td>
<td>0.81</td>
<td>Yes</td>
<td>Population</td>
<td>0.93</td>
<td>Yes</td>
<td>Ventilation</td>
<td>0.76</td>
<td>Yes</td>
</tr>
<tr>
<td>Hand</td>
<td>0.91</td>
<td>Yes</td>
<td>SARs</td>
<td>0.70</td>
<td>Yes</td>
<td>Public health</td>
<td>0.88</td>
<td>Yes</td>
<td>Cough</td>
<td>0.71</td>
<td>Yes</td>
</tr>
<tr>
<td>Antibodies</td>
<td>0.99</td>
<td>Yes</td>
<td>Lymph</td>
<td>0.99</td>
<td>Yes</td>
<td>Tobacco</td>
<td>0.99</td>
<td>Yes</td>
<td>Adaptive immunity</td>
<td>0.99</td>
<td>Yes</td>
</tr>
<tr>
<td>A549 cells</td>
<td>0.99</td>
<td>Yes</td>
<td>White matter</td>
<td>0.99</td>
<td>Yes</td>
<td>Serum albumin</td>
<td>0.99</td>
<td>Yes</td>
<td>Allopurinol</td>
<td>0.99</td>
<td>Yes</td>
</tr>
<tr>
<td>Hepatitis b virus</td>
<td>0.99</td>
<td>No</td>
<td>Alkaline phosphatase</td>
<td>0.99</td>
<td>Yes</td>
<td>Macrophages</td>
<td>0.99</td>
<td>Yes</td>
<td>Liver function tests</td>
<td>0.99</td>
<td>Yes</td>
</tr>
<tr>
<td>Mycoplasma</td>
<td>0.99</td>
<td>Yes</td>
<td>Zinc</td>
<td>0.99</td>
<td>Yes</td>
<td>Bacteroides</td>
<td>0.99</td>
<td>No</td>
<td>Chromium dioxide</td>
<td>0.96</td>
<td>No</td>
</tr>
</tbody>
</table>

Figure 4: Pair embedding visualization. The blue color denotes the *true positive samples*, the red points are *unobserved negative*, the green points are *unobserved positive*.

labels. We observe that the *true positives* (observed in $G^T$ and correctly predicted as positives - blue) and *unobserved negatives* (not observed in $G^T$ and predicted as negatives - red) are further apart. This clear separation indicates that the learned $h$ appropriately grouped the *positive* and *negative (predicted)* pairs in distinct clusters. We also observe that the *unobserved positives* (not observed in $G^T$ but predicted as positives - green) and *true positives* are close. This supports our motivation behind conducting PU learning: the *unlabeled* samples are a mixture of *positive* and *negative* samples, rather than just *negative* samples. We observe that several *unobserved positives* are relationships like between *Tobacco* and *covid-19*. Although they are not connected in the graph we study, several articles have shown a link between terms [2, 46, 37].

5 Conclusion

In this paper, we propose TRP - a temporal risk estimation PU learning strategy for predicting the relationship between biomedical terms found in texts. TRP is shown with advantages on capturing the temporal evolution of the term-term relationship and minimizing the unbiased risk with a positive prior estimator based on variational inference. The quantitative experiments and analyses show that TRP outperforms several state-of-the-art PU learning methods. The qualitative analyses also show the effectiveness and usefulness of the proposed method. For the future work, we see opportunities like predicting the relationship strength between drugs and diseases (TRP for a regression task). We can also substitute the experimental compatibility of terms for the term co-occurrence used in this study.
Acknowledgments and Disclosure of Funding

The research reported in this publication was supported by funding from the Computational Bioscience Research Center (CBRC), King Abdullah University of Science and Technology (KAUST), under award number URF/1/1976-31-01, and NSFC No 61828302. Additional revenue related to this work: student internship at Sony Computer Science Laboratories Inc. We would like to acknowledge the great contribution of Sucheendra K. Palaniappan and The Systems Biology institute to this work for the initial problem definition and data collection.

6 Broader Impact

TRP can be adopted to a wide range of applications involving node pairs in a graph structure. For instance, the prediction of relationships or similarities between two social beings, the prediction of items that should be purchased together, the discovery of compatibility between drugs and diseases, and many more. Our proposed model can be used to capture and analyze the temporal relationship of node pairs in an incremental dynamic graph. Besides, it is especially useful when only samples of a given class (e.g., positive) are available, but it is uncertain whether the unlabeled samples are positive or negative. To be aligned with this fact, TRP treats the unlabeled data as a mixture of negative and positive data samples, rather than all be negative. Thus TRP is a flexible classification model learned from the positive and unlabeled data.

While there could be several applications of our proposed model, we focus on the automatic biomedical hypothesis generation (HG) task, which refers to the discovery of meaningful implicit connections between biomedical terms. The use of HG systems has many benefits, such as a faster understanding of relationships between biomedical terms like viruses, drugs, and symptoms, which is essential in the fight against diseases. With the use of HG systems, new hypotheses with minimum uncertainty about undiscovered knowledge can be made from already published scholarly literature. Scientific research and discovery is a continuous process. Hence, our proposed model can be used to predict pairwise relationships when it is not enough to know with whom the items are related, but also learn how the connections have been formed (in a dynamic process).

However, there are some potential risks of hypothesis generation from biomedical papers. 1) Publications might be faulty (with faulty/wrong results), which can result in a bad estimate of future relationships. However, this is a challenging problem as even experts in the field might be misled by the faulty results. 2) The access to full publication text (or even abstracts) is not readily available, hence leading to a lack of enough data for a good understanding of the studied terms, and then inaccurate h in generation performance. 3) It is hard to interpret and explain the learning process, for example, the learned embedding vectors are relevant to which term features, the contribution of neighboring terms in the dynamic evolution process. 4) For validating the future relationships, there is often a need for background knowledge or a biologist to evaluate the prediction.

Scientific discovery is often to explore the new nontraditional paths. PU learning lifts the restriction on undiscovered relations, keeping them under investigation for the probability of being positive, rather than denying all the unobserved relations as negative. This is the key value of our work in this paper.

References


