Task-Guided and Semantic-Aware Ranking for Academic Author-Paper Correlation Inference

Chuxu Zhang+, Lu Yu*, Xiangliang Zhang* and Nitesh V. Chawla+
+University of Notre Dame, IN 46556, USA
*King Abdullah University of Science and Technology, Thuwal, 23955, SA
{czhang11,nchawla}@nd.edu, {lu.yu,xiangliang.zhang}@kaust.edu.sa

Abstract

We study the problem of author-paper correlation inference in big scholarly data, which is to effectively infer potential correlated works for researchers using historical records. Unlike supervised learning algorithms that predict relevance score of author-paper pair via time and memory consuming feature engineering, network embedding methods automatically learn nodes’ representations that can be further used to infer author-paper correlation. However, most current models suffer from two limitations: (1) they produce general purpose embeddings that are independent of the specific task; (2) they are usually based on network structure but out of content semantic awareness. To address these drawbacks, we propose a task-guided and semantic-aware ranking model. First, the historical interactions among all correlated author-paper pairs are formulated as a pairwise ranking loss. Next, the paper’s semantic embedding encoded by gated recurrent neural network, together with the author’s latent feature is used to score each author-paper pair in ranking loss. Finally, a heterogeneous relations integrative learning module is designed to further augment the model. The evaluation results of extensive experiments on the well known AMiner dataset demonstrate that the proposed model reaches significant better performance, comparing to a number of baselines.

1 Introduction

Due to the growing evolution of scientific research and increasing data collections by various online services such as Google Scholar, Microsoft Academic or AMiner, the problems of mining big scholarly data have gained a lot of attention in the past decade. Examples include scientific impact modeling and prediction [Wang et al., 2013; Shen et al., 2014; Dong et al., 2015], heterogeneous bibliographic network analysis [Sun et al., 2012; Huang et al., 2016; Chen and Sun, 2017], etc.

In this work, we consider the problem of author-paper correlation inference in big scholarly data. Specifically, given an author’s previous correlated papers (e.g., publications or references), we would like to effectively infer the potential relevant works for him/her, such that the author will interact (e.g., cite) with those papers in the future. Solutions of the problem bring important implications to researchers with different knowledgeable levels. For example, an effective algorithm provides suitable academic paper reading lists for new PhD students with little historical feedback in database, or helps active scientists track the related or following works of their previous publications. In addition, it can be a good reference for recommender system design in digital libraries such as Elsevier or Springer.

As one of the representative solutions, supervised leaning algorithms can be applied to predict the correlation score between author and paper, as they were used in 2013 KDD cup author-paper pair identification challenge [Efimov et al., 2013; Li et al., 2015]. However, such methods heavily rely on time and memory consuming feature engineering, and the extracted features may be too simple to capture complicated relations. In recent years, unlike traditional supervised leaning, several network embedding models [Perozzi et al., 2014; Grover and Leskovec, 2016; Chen and Sun, 2017; Dong et al., 2017] have been proposed to automatically learn nodes’ representations that can be further used for various applications in scholarly data such as author-paper correlation inference. Although the proximity among nodes is preserved by dense vectors, most of the existing embedding models suffer from following two drawbacks:

- They produce general purpose embeddings that are independent of task, no matter what kind (homogeneous or heterogeneous) of networked data is used. However, when it comes to the author-paper correlation inference problem, nodes should be embedded under the guidance of target for generating task-specific representations.
- Even though they take account of the task for embedding generation, e.g., TaskE [Chen and Sun, 2017], they are purely based on network structure and out of content awareness. Actually, content (e.g., abstract of paper) contains useful semantic information that should be used for encoding nodes into a better feature space.

To address the above issues and solve the given problem, we propose a task-guided and semantic-aware ranking model. First, we model the historical interactions among correlated author-paper pairs via pairwise ranking according to the spe-
cific task. Next, we introduce the gated recurrent neural network to encode paper’s content, and combine the obtained semantic embedding with author’s latent feature to score each author-paper pair in ranking loss. Moreover, we design a heterogeneous relations integrative learning module to formulate the indirect correlations among authors and papers, and further augment the model. Finally, a relations sampling based mini-batch gradient descent algorithm is designed for model training. The main contributions of this paper are summarized as follows:

- We study the author-paper correlation inference problem in big scholarly data, which brings important implications to academic community.
- To solve the problem, we propose a model by jointly content semantic encoding and heterogeneous relations augmented ranking, and design the corresponding learning algorithm.
- We conduct extensive experiments to evaluate the performance of the proposed model on the well known AMiner dataset. The results show that our model significantly outperforms a number of baselines.

2 Problem

We introduce few notations that will be used throughout this paper. Specifically, we denote the sets of authors and papers as \( U \) and \( I \), respectively. Let \( l_{u,T}^v \) be the set of both author \( u \)'s publications and references before a given timestamp \( T \). Similarly, \( l_{u,T}^v \) represents \( u \)'s papers and references after \( T \). In this work, the correlated papers of each author are assumed as both publications and references in dataset. The problem is formalized as:

**Academic Author-Paper Correlation Inference.** Given \( l_{u,T}^v \) of each author \( u \in U \) and the content (i.e., abstract) of each paper \( v \in I \), the goal is to learn a model to rank all potential papers \( v' \in \hat{l}_{u,T}^v \) for \( u \), such that its top return papers are in \( l_{u,T}^v \).

Note that each author can cite previous works or write new papers, thus the return papers can be published both before and after \( T \). In addition, the overlapping between \( l_{u,T}^v \) and \( l_{u,T}^v \) are removed from \( l_{u,T}^v \), so that papers in \( l_{u,T}^v \) are never cited by \( u \) before \( T \).

3 Proposed Model

In this section, we present how to design task-guided and semantic-aware ranking model for solving the problem, and use historical academic data to construct a heterogeneous network for capturing indirect author-paper relations which benefit and augment the model.

3.1 Pairwise Ranking with Gated Recurrent Neural Network

We model historical interactions among correlated author-paper pairs via pairwise ranking optimization [Rendle et al., 2009]. Specifically, for a given author \( u \), the correlated paper \( v \in \hat{l}_{u,T}^v \) should be ranked higher than the uncorrelated paper \( v' \in \hat{l}_{u,T}^v \). In other words, the relevance score \( s_{u,v} \) of \( \langle u, v \rangle \) pair should be larger than that of \( \langle u, v' \rangle \) pair as much as possible, leading to an author-paper pairwise ranking loss as follows:

\[
\mathcal{L}_{\text{rank}} = \sum_{u \in U} \sum_{v \in \hat{l}_{u,T}^v} \sum_{v' \notin \hat{l}_{u,T}^v} \left[ -\log \sigma(s_{u,v} - s_{u,v'}) \right]
\]

where \( \sigma \) is the sigmoid function. For each author-paper pair \( \langle u, v \rangle \), we introduce author \( u \)'s latent feature \( q_u \in \mathbb{R}^d \) (the embedding dimension) and represent paper \( v \) as semantic embedding \( E_{p_v} \in \mathbb{R}^d \) via content encoder \( f \); \( E_{p_v} = f(p_v) \), where \( p_v \) represents paper \( v \)'s content. The inner product of \( q_u \) and \( f(p_v) \) is used to measure the relevance score, i.e., \( s_{u,v} = q_u^T f(p_v) \).

To encode papers’ contents to fixed size embeddings \( E_p \in \mathbb{R}^{1 \times d} \), we use the gated recurrent units (GRU), a specific type of recurrent neural network (RNN), which has been widely adopted for many applications such as machine translation [Cho et al., 2014]. Specifically, a paper is represented by a sequence of word embeddings: \( \{x_1, x_2, \ldots, x_{m_{\text{max}}} \} \), where \( x_t \) denotes the \( t \)-th word embedding pre-trained by word2vec [Mikolov et al., 2013] and \( m_{\text{max}} \) is the maximum length of paper’s abstract. For each step \( t \) with the input of word embedding \( x_t \) and hidden state vector \( h_{t-1} \), GRU computes the updated hidden state vector via:

\[
\begin{align*}
\mathbf{z}_t &= \sigma(\mathbf{N}_t x_t + \mathbf{M}_t h_{t-1}) \\
\mathbf{r}_t &= \sigma(\mathbf{N}_t x_t + \mathbf{M}_t h_{t-1}) \\
\mathbf{h}_t &= \tanh[\mathbf{N}_t x_t + \mathbf{M}_t (\mathbf{r}_t \odot h_{t-1})] \\
\end{align*}
\]

where \( \sigma \) is the sigmoid function, the operator \( \odot \) denotes element-wise multiplication, \( \mathbf{N} \) and \( \mathbf{M} \) are parameter matrices of \( \text{GRU} \) network, \( \mathbf{z}_t \) and \( \mathbf{r}_t \) are updated gate vector and reset gate vector, respectively. We apply the above \( \text{GRU} \) network to encode words’ contextual embeddings \( \mathbf{h} \in \mathbb{R}^{1 \times d} \) and use a mean pooling layer to obtain the general semantic embedding of each paper. All of these steps construct the paper’s content encoder \( f \), as illustrated by Figure 1(a). Note that, we also explore other encoding architectures such as LSTM or attention-based GRU but obtain similar result. Thus we choose GRU since it has a concise structure for reducing training time.

Using the above \( \text{GRU} \) encoder for papers’ embeddings and the authors’ latent features, we further minimize the pairwise ranking loss (i.e., Eq. (1)) via gradient descent approach, as illustrated by Part-1 of Figure 1(b). The process leads to task-guided and semantic-aware ranking and we name it TSR.

3.2 Heterogeneous Relations based Integrative Learning Augmentation

TSR trains model by only using direct correlations, i.e., correlated author-paper pairs in \( \hat{l}_{u,T}^v \) for each author \( u \). However, there are multiple indirect author-paper relations, which can be inferred from direct correlations and useful for improving TSR. Inspired by DeepWalk [Perozzi et al., 2014], we apply random walk to collect those indirect correlations. Figure 1(b) Part-2 gives the illustration. First, we use the correlated papers set \( \hat{l}_{u,T}^v \) (contains both author \( u \)'s publications and references before \( T \)) of each author \( u \) to create a heterogeneous
network (HetNet) with two kinds of nodes (author and paper) and two kinds of undirected edges (author-paper and paper-paper). Then we perform random walk sampling over HetNet to generate a set of node sequences. For example, in Figure (b) Part-2, a heterogeneous walk: \(w_a = \{a \rightarrow b \rightarrow c \rightarrow 4 \rightarrow 3 \rightarrow 5\}\) is generated and the surrounding context of each node in \(w_a\) implies different relations among authors and papers within \(w_a\). We use sub-sequence \(w_1 = \{1 \rightarrow b \rightarrow 2 \rightarrow c \rightarrow 4 \rightarrow 3 \rightarrow 5\}\) (window size equals to 3) centered at author \(c\) as an illustration. Besides the direct connections, e.g., author \(c\) writes paper 2, \(w_1\) also captures multiple indirect relations. For example, author \(c\) has indirect citation relation with paper 5 since s/he has citation relation with paper 3. Therefore, each heterogeneous walk contains both direct correlations and indirectly transitive relations among authors and papers.

To formulate indirect relations within each walk, we design a heterogeneous relations integrative learning module (HRIL) to augment TSR based on a reasonable assumption that the relevance scores of indirectly correlated author-paper pairs should be larger than those of uncorrelated pairs. Specifically, we introduce a hinge loss to formulate the difference of author \(u\)’s correlations to indirectly correlated paper \(v\) and uncorrelated paper \(v’\):

\[
H(u, v, v') = \left[\xi + s_{uv'} - s_{uv}\right]_+	ag{3}
\]

where \(\{x\}_+ = \max(x, 0)\) and \(\xi\) is a positive margin value. A loss penalty will incur if the score of \((u, v)\) is not at least \(\xi\) larger than that of \((u, v')\). Such formulation has been widely adopted in recent works [Chen and Sun, 2017; Zhang et al., 2017; 2018] for modeling preference differences. Thus the overall loss in each walk \(w\) is formulated as:

\[
L_{hinge}(w) = \sum_{u \in w} \sum_{v \in w} \sum_{v' \in [v]\tau} \left[\xi + s_{uv'} - s_{uv}\right]_+	ag{4}
\]

where \(\tau\) is the window size of surrounding context used for relations extraction, and \(I_u\) indicates the position of \(u\) in \(w\). Therefore constraining \(L_{hinge}\) obeys our assumption.

### 3.3 Model Training

We generate a plenty of walks rooted at each author node to collect indirect correlations among authors and papers, and let network (HetNet) with two kinds of nodes (author and paper) and two kinds of undirected edges (author-paper and paper-paper). Then we perform random walk sampling over HetNet to generate a set of node sequences. For example, in Figure (b) Part-2, a heterogeneous walk: \(w_a = \{a \rightarrow b \rightarrow c \rightarrow 4 \rightarrow 3 \rightarrow 5\}\) is generated and the surrounding context of each node in \(w_a\) implies different relations among authors and papers within \(w_a\). We use sub-sequence \(w_1 = \{1 \rightarrow b \rightarrow 2 \rightarrow c \rightarrow 4 \rightarrow 3 \rightarrow 5\}\) (window size equals to 3) centered at author \(c\) as an illustration. Besides the direct connections, e.g., author \(c\) writes paper 2, \(w_1\) also captures multiple indirect relations. For example, author \(c\) has indirect citation relation with paper 5 since s/he has citation relation with paper 3. Therefore, each heterogeneous walk contains both direct correlations and indirectly transitive relations among authors and papers.

To formulate indirect relations within each walk, we design a heterogeneous relations integrative learning module (HRIL) to augment TSR based on a reasonable assumption that the relevance scores of indirectly correlated author-paper pairs should be larger than those of uncorrelated pairs. Specifically, we introduce a hinge loss to formulate the difference of author \(u\)’s correlations to indirectly correlated paper \(v\) and uncorrelated paper \(v’\):

\[
H(u, v, v') = \left[\xi + s_{uv'} - s_{uv}\right]_+\tag{3}
\]

where \(\{x\}_+ = \max(x, 0)\) and \(\xi\) is a positive margin value. A loss penalty will incur if the score of \((u, v)\) is not at least \(\xi\) larger than that of \((u, v')\). Such formulation has been widely adopted in recent works [Chen and Sun, 2017; Zhang et al., 2017; 2018] for modeling preference differences. Thus the overall loss in each walk \(w\) is formulated as:

\[
L_{hinge}(w) = \sum_{u \in w} \sum_{v \in w} \sum_{v' \in [v]\tau} \left[\xi + s_{uv'} - s_{uv}\right]_+\tag{4}
\]

where \(\tau\) is the window size of surrounding context used for relations extraction, and \(I_u\) indicates the position of \(u\) in \(w\). Therefore constraining \(L_{hinge}\) obeys our assumption.

### 3.3 Model Training

We generate a plenty of walks rooted at each author node to collect indirect correlations among authors and papers, and let

\[
\begin{align*}
L &= L_{\text{rank}} + \sum_{w \in W} L_{\text{hinge}}(w) + L_{\text{reg}}
\end{align*}
\]

where \(L_{\text{reg}}\) is the regularization term for avoiding overfitting. We denote all model parameters including GRU network coefficients (for generating papers’ embeddings) and authors’ latent features as \(\Theta\), and let \(C_{\text{rank}}\) and \(C_{\text{hinge}}\) be the sets of \((u, v, v')\) triples in \(L_{\text{rank}}\) and \((u, v, v')\) triples in \(L_{\text{hinge}}\), respectively. Thereafter we can rewrite the objective of TSR+ as:

\[
\begin{align*}
\mathcal{L} &= \sum_{(u, v, v') \in C_{\text{rank}}} \left\{-\log \sigma \left[ q_u^T f(p_u) - q_u^T f(p_{v'}) \right] \right\} \\
&\quad + \sum_{(u, v, v') \in C_{\text{hinge}}} \left\{ \xi + q_u^T f(p_{v'}) - q_u^T f(p_u) \right\}_+ + \lambda \| \Theta \|^2
\end{align*}
\]

where parameter \(\lambda\) controls regularization penalty. To minimize the above objective function, we design a relations sampling based mini-batch Adam Optimizer [Kingma and Ba, 2014], as illustrated by the pseudocode in Algorithm 1.

### 4 Experiments

In this section, we conduct extensive experiments to evaluate the proposed model and various baselines. Case studies are also provided.
4.1 Experimental Design

Dataset.

We use the public available dataset\footnote{https://aminer.org/citation} AMiner [Tang et al., 2008] between 2005 and 2015, and remove the papers published in venues (e.g., workshop) with limited number of publications and the instances without content (i.e., abstract). Besides, considering most of researchers pay attention to papers published in top venues and each research area has its own community, we extract a subset data of six areas according to Google Scholar Metrics\footnote{https://scholar.google.com/citations?view-op=metrics_intro&hl=en}, namely Artificial Intelligence (AI), Computer Vision (CV), Data Mining (DM), Databases (DB), Computational Linguistics (CL) and Information System (IS). For each area, we choose three top venues\footnote{AI: ICML, AAAI, IJCAI. CV: CVPR, ICCV, ECCV. DM: KDD, WSDM, ICDM. DB: SIGMOD, VLDB, ICDE. CL: ACL, EMNLP, NAACL. IS: WWW, SIGIR, CIKM.} that are considered to have influential publications. The main statistics of two datasets (AMiner-T and AMiner-F) are summarized in Table 1. Note that, there are some missing citations in the dataset. It makes the inference task more difficult due to potential noise in model training.

Baselines.

We compare TSR and TSR+ with eight baseline methods that span three categories: (1) feature-based supervised learning, (2) content-based ranking and (3) network embedding.

- **Feature-based supervised learning.** It first extracts author-paper paired features and then applies supervised learning algorithms to predict the relevance score of each author-paper pair. We extract 14 kinds of features (as shown in Table 2) and choose Bayesian Regressor (BayesR), Neural Network (NeuNet) and Random Forest (RandomF) as learning algorithms. For each correlated author-paper pair, we randomly sample 5 negative pairs to train model.

- **Content-based ranking.** It first encodes each paper’s content (i.e., abstract) via language modeling and then applies pairwise ranking BPR [Rendle et al., 2009] to learn each author’s latent feature. We employ two popular models word2vec [Mikolov et al., 2013] and paragraph vector (doc2vec) [Le and Mikolov, 2014] to learn papers’ embeddings. Note that word2vec encodes embedding of each word in content, we connect the output of word2vec with a mean pooling layer to obtain general embedding of each paper.

- **Network embedding.** It learns embeddings of both authors and papers based on the structure of author-paper heterogeneous network (same as the HetNet in TSR+). We use both homogeneous model Deepwalk [Perozzi et al., 2014] and heterogeneous model metapath2vec [Dong et al., 2017]. In addition, a task-guided network embedding model (TaskE) [Chen and Sun, 2017] for author identification is introduced for comparison.

Evaluation Metrics.

As described in problem definition, for each author $u \in U$, papers in $l_{u,T}$ are treated as training data and papers in $l_{u,T}$ are left for evaluation. The overlapping between $l_{u,T}$ and $l_{u,T}$ are removed from $l_{u,T}$. We use three popular metrics, i.e., Recall@$k$, Precision@$k$ and AUC, to evaluate the performance of each method. The Recall@$k$ shows the ratio of true correlated papers returned in the top-k list, which is defined as:

$$Rec@k = \frac{1}{|U|} \sum_{u \in U} \frac{|l_{u,T} \cap l_{u,T}^T|}{|l_{u,T}|},$$

where $l_{u,T}$ denotes the set of top-k papers for author $u$. The Precision@$k$ reflects the accuracy of top-k papers by a method and it can be computed according to:

$$Pre@k = \frac{1}{|U|} \sum_{u \in U} \frac{|l_{u,T} \cap l_{u,T}^T|}{k}.$$

The AUC measures the accuracy of pairwise orders between correlated and uncorrelated papers of each author, which is formulated as:

$$AUC = \frac{1}{|U|} \sum_{u \in U} \sum_{u' \in U} \sum_{v \in l_{u,T}, v' \in l_{u',T}} \delta(s_{uv} > s_{uv'}) - \delta(s_{uv} < s_{uv'})$$

where $E(u) = \{(v, v') | v \in l_{u,T}, v' \notin l_{u,T} \cup l_{u',T}\}$ and $\delta$ is indicator function which equals to 1 when the condition holds otherwise 0. The $k$ is set to 10 and a larger Recall@$k$, Precision@$k$ or AUC value means a better performance.

Experimental Settings.

All of information used for model training such as triple/pair samples in our models or the selected features in supervised learning baselines, are extracted from training data. We design two different training/test splits by setting $T = 2012$ and 2013. Besides, two key issues of the experiments are set as:

- **Reproducibility.** For the fair of comparison, we use the same feature dimension $d = 128$ for all content-based ranking and network embedding models. In the proposed models, the regularization parameter $\lambda$ equals to 0.001. For HRIL module of TSR+, we set the number of walks (start from each author node) as 10 and the walk length as 20. Besides, we fix window size $\tau = 5$ and hinge loss margin $\xi = 0.1$. In addition, we employ TensorFlow to implement the proposed model and further conduct it via NVIDIA TITAN X GPU.
### 4.2 Result Comparison

The performances of all methods are reported in Table 3, where the best results are highlighted in bold and the best baselines are indicated by star notation. TSR+ achieves the best results (highlighted in bold) in all cases.

**Evaluation candidates.** It is time and memory consuming to extract and store features for all author-paper pairs (which amounts to over \(2.7 \times 10^{13}\) pairs in AMiner-F). Thus the supervised learning baselines cannot scale up to such large amount of data. To deal with this issue and reduce evaluation time, we follow the setting [Chen and Sun, 2017] that randomly samples a set of negative (uncorrelated) papers and combines it with the set of correlated papers to form a candidate set of total 200 papers for each author. In addition, we eliminate the authors who have few correlated papers (less than 3) in test set to avoid noise. After that, the average sizes of authors’ correlated papers sets respectively equal to 14.7 and 12.5 in AMiner-F test data for \(T = 2012\) and \(2013\), and the corresponding values in AMiner-T test data are 15.7 and 12.7. Those relative small values make the 200 candidates large enough for a convincing evaluation. The reported results are averaged over 10 experiments of such setting.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>T</th>
<th>Metric</th>
<th>BayesR</th>
<th>NeuNet</th>
<th>RandomF</th>
<th>word2vec +BPR</th>
<th>doc2vec +BPR</th>
<th>Deepwalk</th>
<th>metapath2vec</th>
<th>TaskE</th>
<th>TSR</th>
<th>TSR+</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMiner-T</td>
<td>2012</td>
<td>Rec@10</td>
<td>0.211</td>
<td>0.220</td>
<td>0.288</td>
<td>0.267</td>
<td>0.224</td>
<td>0.230</td>
<td>0.305*</td>
<td>0.296</td>
<td>0.312</td>
<td>0.330</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pre@10</td>
<td>0.248</td>
<td>0.263</td>
<td>0.341</td>
<td>0.311</td>
<td>0.274</td>
<td>0.312</td>
<td>0.364*</td>
<td>0.355</td>
<td>0.350</td>
<td>0.374</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AUC</td>
<td>0.674</td>
<td>0.663</td>
<td>0.705</td>
<td>0.801*</td>
<td>0.739</td>
<td>0.675</td>
<td>0.723</td>
<td>0.738</td>
<td>0.831</td>
<td>0.846</td>
</tr>
<tr>
<td>AMiner-F</td>
<td>2013</td>
<td>Rec@10</td>
<td>0.224</td>
<td>0.255</td>
<td>0.301</td>
<td>0.272</td>
<td>0.228</td>
<td>0.189</td>
<td>0.302</td>
<td>0.303*</td>
<td>0.329</td>
<td>0.343</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pre@10</td>
<td>0.221</td>
<td>0.246</td>
<td>0.304</td>
<td>0.276</td>
<td>0.244</td>
<td>0.237</td>
<td>0.325*</td>
<td>0.321</td>
<td>0.320</td>
<td>0.341</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AUC</td>
<td>0.694</td>
<td>0.693</td>
<td>0.730</td>
<td>0.785*</td>
<td>0.735</td>
<td>0.647</td>
<td>0.715</td>
<td>0.723</td>
<td>0.853</td>
<td>0.842</td>
</tr>
<tr>
<td></td>
<td>2012</td>
<td>Rec@10</td>
<td>0.333</td>
<td>0.361</td>
<td>0.378*</td>
<td>0.372</td>
<td>0.341</td>
<td>0.245</td>
<td>0.365</td>
<td>0.351</td>
<td>0.444</td>
<td>0.461</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pre@10</td>
<td>0.344</td>
<td>0.375</td>
<td>0.379</td>
<td>0.377</td>
<td>0.357</td>
<td>0.327</td>
<td>0.391*</td>
<td>0.397</td>
<td>0.433</td>
<td>0.450</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AUC</td>
<td>0.709</td>
<td>0.726</td>
<td>0.708</td>
<td>0.844*</td>
<td>0.808</td>
<td>0.698</td>
<td>0.740</td>
<td>0.721</td>
<td>0.869</td>
<td>0.878</td>
</tr>
<tr>
<td></td>
<td>2013</td>
<td>Rec@10</td>
<td>0.353</td>
<td>0.406</td>
<td>0.404</td>
<td>0.399</td>
<td>0.362</td>
<td>0.286</td>
<td>0.412*</td>
<td>0.398</td>
<td>0.475</td>
<td>0.496</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pre@10</td>
<td>0.331</td>
<td>0.383</td>
<td>0.368</td>
<td>0.366</td>
<td>0.344</td>
<td>0.331</td>
<td>0.423*</td>
<td>0.405</td>
<td>0.427</td>
<td>0.445</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AUC</td>
<td>0.721</td>
<td>0.742</td>
<td>0.720</td>
<td>0.850*</td>
<td>0.817</td>
<td>0.726</td>
<td>0.774</td>
<td>0.739</td>
<td>0.880</td>
<td>0.883</td>
</tr>
</tbody>
</table>

| Imps. over Baseline | 37.7% | 28.3% | 18.1% | 16.9% | 28.9% | 43.8% | 13.0% | 16.2% | - | - |

Table 3: Performance comparisons of different models. The last row reports the average improvements (%) of TSR+ over baselines. The best baseline of each case is indicated by star notation. TSR+ achieves the best results (highlighted in bold) in all cases.

- With the increment of \(\tau\), \(Rec_{@10}\) and \(Pre_{@10}\) increase first at since a larger window means more useful indirect correlations among authors and papers. But when \(\tau\) goes beyond a certain value, the results decrease with the further increment of \(\tau\) due to the possible involvement of uncorrelated noise. The best \(\tau\) is around 5.
- Similar to \(\tau\), an appropriate value should be set for \(d\) such that the best representations of authors and papers are learned. The optimal value of \(d\) is around 128.

Besides \(d\) and \(\tau\), we also investigate the impacts of other hyper-parameters such as regularization parameter \(\lambda\), and reveal the similar point. Therefore the certain settings of the hyper-parameters lead to the best performance of TSR+.

**Performances on Different Author Groups.**

As presented in introduction, an author-paper inference model should be effective for researchers with different knowledgeable levels. In order to validate the effectiveness of TSR+ on different author groups (from “cold-start” to active), we classify all authors into 6 groups (i.e., \(1\sim3, 4\sim6, 7\sim9, 10\sim12, 13\sim15\) and \(16\sim17\) based on the number of observed correlated papers they have in training data, then evaluate the performance in each group. The performances (in
2013

### Venue
2013

### Paper
- **Rank**: 4
- **Venue**: WSDM
- **Year**: 2013
- **Title**: "Yes, there is a correlation: from social networks to personal behavior on the web.

### Venue
2012

### Paper
- **Rank**: 3
- **Venue**: KDD
- **Year**: 2012
- **Title**: "No country for old members: user lifecycle and linguistic change in online communities.

### Venue
2012

### Paper
- **Rank**: 4
- **Venue**: WWW
- **Year**: 2012
- **Title**: "Fast mining and forecasting of complex time-stamped events.

### Venue
2010

### Paper
- **Rank**: 5
- **Venue**: WWW
- **Year**: 2010
- **Title**: "Earthquake shakes Twitter users: real-time event detection by social sensors.

### Venue
2008

### Paper
- **Rank**: 5
- **Venue**: WWW
- **Year**: 2008
- **Title**: "Inferring user demographics and social strategies in mobile social networks.

### Venue
2012

### Paper
- **Rank**: 4
- **Venue**: KDD
- **Year**: 2012
- **Title**: "Confluence: conformity influence in large social networks.

### Venue
2010

### Paper
- **Rank**: 3
- **Venue**: KDD
- **Year**: 2010
- **Title**: "Predicting emerging social conventions in online social networks.

Table 4: Two case studies of TSR+’s result. Notations □, △ and × represent writing, citing and uncorrelated relations between author and paper, respectively.

![Graphs](image1.png)

Figure 2: The impacts of window size and embedding dimension on TSR+. TSR+ performs best when \( T = 2013 \) and \( d = 128 \).

![Graphs](image2.png)

Figure 3: Performances of our models and three selected baselines in different author groups. TSR+ consistently reaches the best performance in all groups.

5 Related Work

In the past decade, some works have devoted to academic data mining problems, such as heterogeneous bibliographic network analysis [Sun et al., 2012; Huang et al., 2016], citation recommendation [He et al., 2010; Ren et al., 2014] or collaborator recommendation [Tang et al., 2012; Li et al., 2014]. In this paper, we study the problem of author-paper correlation inference in big scholarly data.
GRU-based content encoding and task-guided ranking, and is further augmented by a heterogeneous relations integrative learning module. The extensive experiments on the well known AMiner data demonstrate that TSR+ achieves significant better performance, comparing to a number of baselines. Some potential future work includes: (1) TSR+ can be extended by using more context information like publication venue of paper; (2) the dynamics of authors’ embeddings should be considered for the task since authors keep publishing new papers, and citing more papers.

Acknowledgments

We would like to thank Yuxiao Dong for suggestions. This work is supported by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053 and the National Science Foundation (NSF) grant IIS-1447795. This work is partially supported by King Abdullah University of Science and Technology (KAUST).

References


[Ren et al., 2014] Xiang Ren, Jialu Liu, Xiao Yu, Urvashi Khandelwal, Quanquan Gu, Lidan Wang, and Jiawei Han. Cluscite: Effective citation recommendation by information network-based clustering. In KDD, pages 821–830, 2014.


3647