Author2Vec: Learning Author Representations by Combining Content and Link Information

Ganesh J[1] Soumyajit Ganguly[1] Manish Gupta[1,2] Vasudeva Varma[1] Vikram Pudi[1]
[1]IIIT, Hyderabad, India, {ganesh.j, soumyajit.ganguly}@research.iiit.ac.in, {vv, vikram}@iiit.ac.in
[2]Microsoft, Hyderabad, India, gmanish@microsoft.com

ABSTRACT

In this paper, we consider the problem of learning representations for authors from bibliographic co-authorship networks. Existing methods for deep learning on graphs, such as DeepWalk, suffer from link sparsity problem as they focus on modeling the link information only. We hypothesize that capturing both the content and link information in a unified way will help mitigate the sparsity problem. To this end, we present a novel model 'Author2Vec' [1], which learns low-dimensional author representations such that authors who write similar content and share similar network structure are closer in vector space. Such embeddings are useful in a variety of applications such as link prediction, node classification, recommendation and visualization. The author embeddings we learn are empirically shown to outperform DeepWalk by 2.35% and 0.83% for link prediction and clustering task respectively.

1. INTRODUCTION

Recently, there has been an increasing interest in embedding information networks [1, 2] into low-dimensional vector spaces. The motivation is that once the embedded vector form is obtained, the network mining tasks can be solved by off-the-shelf machine learning algorithms. In an attempt to construct good representation in a scalable way, researchers have started using deep learning as a tool to analyze graphs. For instance, DeepWalk [2], a recent model, transforms a graph structure into a sample collection of linear sequences containing vertices using uniform sampling (truncated random walk). They treat each sample as a sentence, run the Skip-gram model [5], originally designed for learning word representations from linear sequences to learn the representation of vertices, from such samples.

The main drawback of DeepWalk is the link sparsity problem [6] inherent in a real world information network. For example, two authors who write scientific articles related to the field 'Machine Learning' are not considered to be similar by DeepWalk if they are not connected. In this paper, we aim to overcome the above mentioned problem by fusing the textual information with the link information in a synergistic fashion, for creating author representations. Our experiments on a large dataset show that harnessing the content and link information alleviates the link sparsity problem.

[1]Code is publicly accessible at `https://github.com/ganeshjawahar/author2vec`

WWW'16 Companion, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4144-8/16/04.
http://dx.doi.org/10.1145/2872518.2889382.

2. AUTHOR2VEC MODEL

Consider a co-authorship network $G = (V, E)$ in which each vertex represents the author and edge $e = \langle u, v \rangle \in E$ represents an interaction between author u and author v. Two authors are connected if they co-author at least one article. Let us denote the set of articles published by each author u by $P_u = \{p_{u1}, .., p_{uN_p}\}$, containing N_p papers. For every paper, we also have the abstract and the year of publication. Then the goal of our proposed model, Author2Vec, is to learn author representations $\mathbf{v}_u \in \mathbb{R}^d$ $(\forall u \in V)$, where d is the embedding size. The model learns the author embedding in an unsupervised way, using two types of models: Content-Info and Link-Info model, which are explained below. As the name suggests, the former model learns the textual concepts, while the latter model enriches the social dimensions further by fusing the relational concepts.

Content-Info Model: This model aims to capture the author representation purely by the textual content, represented by the abstracts of her papers. The model takes an author u (associated with embedding $\mathbf{v}_u$) and paper p (associated with embedding $\mathbf{v}_p$) as inputs and predicts whether u wrote p or not. Our training tuples consist of a set of positive input pairs (where p is a publication by the author u) and negative input pairs (where p is not a publication by the author u). The intuition to do this is to push the author representations closer to her content, and away from irrelevant content. More formally, we predict the author-paper relationship $r_C(u, p)$, taking the value $l \in [1, 2]$, where '1' and '2' denote the negative and positive input pair respectively. We predict using a neural network that considers both the angle (Eq. 1) and the distance (Eq. 2) between the input pair $(\mathbf{v}_u, \mathbf{v}_p)$:

$$h_C^{(\times)} = \mathbf{v}_u \odot \mathbf{v}_p \tag{1}$$

$$h_C^{(+)} = \mid \mathbf{v}_u - \mathbf{v}_p \mid \tag{2}$$

$$h_C = tanh(W_C^{(\times)} h_C^{\times} + W_C^{(+)} h_C^{+} + b_C^{(h)}) \tag{3}$$

where $W_C^{(\times)} \in \mathbb{R}^{n_h \times d}$, $W_C^{(+)} \in \mathbb{R}^{n_h \times d}$, $b_C^{(h)}$ are the parameters of this model. Note n_h defines the hidden layer size. The usage of distance metrics, $h_C^{(\times)}$ and $h_C^{(+)}$ is empirically motivated and similar strategies have been successfully used

in Tai et al. [7]'s work to capture semantic relatedness of sentence pairs. The objective function of the Content-Info model can be written as follows.

$$\mathcal{L}_C = \mathbb{P}[r_C(u,p) = l] = softmax(U_C.h_C + b_C^{(p)}) \quad (4)$$

where $U_C \in \mathbb{R}^{2\times n_h}$, $b_C^{(p)}$ are the new parameters of this model. We learn the unknown parameters $W_C^{(\times)}$, $W_C^{(+)}$, $b_C^{(h)}$, U_C, $b_C^{(p)}$, $\mathbf{v}_u \in \mathbb{R}^d$ (i.e. author embeddings), $\mathbf{v}_p \in \mathbb{R}^d$ (i.e. paper embeddings) by maximizing the likelihood function in Eq. 4. The paper embeddings ($\mathbf{v}_p$) are pre-initialized with output obtained from running Paragraph2Vec [3] on all the abstracts.

Link-Info Model: The goal of the Link-Info model is to enrich the author representations obtained from the previous model by fusing the link information. This model takes as input both the author embeddings ($\mathbf{v}_u$ and $\mathbf{v}_v$). Similar to the Content-Info model, the training tuples consist of positive input pairs (where u has collaborated with v) and negative input pairs (where u has never collaborated with v in the training set). This setup effectively pushes the authors who share similar network structure closer in vector space from the irrelevant authors. We predict the author-author relationship $r_L(u,v)$ using a different neural network:

$$h_L^{(\times)} = \mathbf{v}_u \odot \mathbf{v}_v \quad (5)$$

$$h_L^{(+)} = \mid \mathbf{v}_u - \mathbf{v}_v \mid \quad (6)$$

$$h_L = tanh(W_L^{(\times)} h_L^{\times} + W_L^{(+)} h_L^{+} + b_L^{(h)}) \quad (7)$$

where $W_L^{(\times)} \in \mathbb{R}^{n_h \times d}$, $W_L^{(+)} \in \mathbb{R}^{n_h \times d}$, $b_L^{(h)}$ are the parameters of this model. The objective function of the Link-Info model can be written as follows.

$$\mathcal{L}_L = \mathbb{P}[r_L(u,v) = l] = softmax(U_L.h_L + b_L^{(p)}) \quad (8)$$

where $U_L \in \mathbb{R}^{2\times n_h}$, $b_L^{(p)}$ are the new parameters of this model. We learn the unknown parameters $W_L^{(\times)}$, $W_L^{(+)}$, $b_L^{(h)}$, U_L, $b_L^{(p)}$, $\mathbf{v}_u \in \mathbb{R}^d$ (i.e. author embeddings) by maximizing the likelihood function in Eq. 8.

Training Details: We can trivially connect both the models by sharing the author embedding weights. Thus, the overall objective function of Author2Vec, maximizing the data likelihood can we written as follows.

$$\mathcal{L} = \mathcal{L}_C + \mathcal{L}_L \quad (9)$$

We use stochastic gradient descent with mini-batch size of 256 and learning rate of 0.1 to learn the unknown parameters of the model.

3. EXPERIMENTS

We validate our learned author embeddings using two tasks: link prediction and clustering. In all the experiments, we empirically set d, n_h to 100, 50 respectively. We use Citation Network Dataset (CND) proposed in Chakraborty [4] for both the tasks. CND contains 711810 computer science papers (along with abstracts) written by a total of 500361 authors. Each paper is tagged with one of the 24 computer science fields. We use the best experimental settings for DeepWalk reported in [2].

Link Prediction: For link prediction, we use 20 years of CND data from 1990-2009, where the last year is used for testing and the remaining as training. The positive classes are the author pairs who co-authored in the training years.

Table 1: Performance comparison on link prediction and clustering tasks

Task	Link Prediction	Clustering
Model \Metric	Accuracy (%)	NMI (%)
DeepWalk	81.965	19.956
Content-Info	80.707	19.823
Link-Info	72.898	19.163
Author2Vec	**83.894**	**20.122**

For every positive pair, we choose one negative pair randomly. Test set contains the pair of authors who did not publish together in the training years, but in the test year. The resulting dataset contains 2485764 training pairs and 15342 test pairs. We use logistic regression to solve this binary classification problem. We report the accuracies for this task.

Clustering: CND contains manually annotated research area for each paper. For simplicity, we associate a field to each author by picking the field in which the author publishes the most. We employ K-Means algorithm (with k=24, denoting the number of computer science fields) using the embeddings as features and report the cross-validation scores using Normalized Mutual Information (NMI) metric.

Analysis: Results in Table 1 lead to the following observations: (1) Using only the content information fails to perform well without linkage knowledge. (2) Model which learns only using the link information gives poor results. This is because without the global content information, the author embeddings tend to be sensitive to noisy links. (3) DeepWalk outperforms the previously discussed naïve models, mainly due to the superiority of random walk based approach over the negative sampling approach. (4) However, Author2Vec outperforms DeepWalk thanks to the fusion of the content and link information. The performance improvement of 2.35% for link prediction and 0.83% for the clustering task clearly shows the superior quality of author embeddings learned by Author2Vec.

4. CONCLUSIONS

Author2Vec fuses content and link information to learn high quality author embeddings given a bibliographic network. We plan to extend the model for weighted graphs, where edge weights indicate the number of papers co-authored. It would also be interesting to incorporate the global network information to enhance the embeddings.

5. REFERENCES

[1] Ahmed, A., Shervashidze, N., Narayanamurthy, S., Josifovski, V., Smola, A.J.: Distributed Large-scale Natural Graph Factorization. In: WWW. (2013) 37–48

[2] Perozzi, B., Al-Rfou, R., Skiena, S.: DeepWalk: online learning of social representations. In: KDD. (2014) 701–710

[3] Le, Q., Mikolov, T.: Distributed Representations of Sentences and Documents. In: ICML. (2014) 1188–1196

[4] Chakraborty, T., Sikdar, S., Tammana, V., Ganguly, N., Mukherjee, A.: Computer Science Fields as Ground-truth Communities: Their Impact, Rise and Fall. In: ASONAM. (2013) 426–433

[5] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed Representations of Words and Phrases and their Compositionality. In: NIPS. (2013) 3111–3119

[6] Nowell, D.L., Kleinberg, J.: The link-prediction problem for social networks. In: Journal of the American Society for Information Science and Technology. (2007) 1019-1031

[7] Tai, K.S., Socher, R., Manning, C.D.: Improved semantic representations from tree-structured long short-term memory. In: ACL. (2015) 1556–1566