

Enhanced Microblog Network Representation with User-Generated Content

Yue Hu¹, Shu-Qiang Yang¹, Bin Zhou¹

¹College of Computer, National University of Defense Technology, Changsha, China

¹huyue15a@nudt.edu.cn

Abstract: The real-world networks are so sparse that only utilizing the limited edges cannot capture valuable user feature fully. However, the existing network embedding methods usually took the network structure into consideration merely. Therefore, inspired by the assumption that there exist latent relationships between those who generate similar text information, we proposed to enhance the original network structures by incorporating the latent relationships extracted from user-generated content into the raw network topology and furthermore to learn the representation for the revised network. Eventually, we evaluate our proposed method in two inference tasks and the experiment results demonstrate that the network representation generated by our enhanced network embedding method has a better performance than the baseline method on dataset provided by Sina Microblog.

1. Introduction

With the widespread popularity of social network services such as Sina Microblog, networks are ubiquitous in our daily life and play an increasingly important role in the study for our society. However, the diversity and complexity of network information is not propitious for capturing more valuable user features. Moreover, it is difficult to perform these subsequent applications such as interest recommendation, user profile and so on. A straightforward idea is to learn user representation with these available network information. As is known to all, a large-scale network constitutes a high-dimension space so that network representation seeks to embed the networks into a low-dimensional space. In recent year, network representation learning has drawn extensive attention.

1.1 Related Work

Network representation is essentially network embedding in low-dimension space. Many outstanding research effort has been proposed in network embedding. Earlier manifold learning [1] can be considered the originator, such as Isomap, Laplacian Eigenmaps (LE) and Locally Linear Embedding. Inspired by the word embedding [2], DeepWalk [3] exploited Random Walk [4] to generate vertex sequence as sentence in corpus and then introduced Skip-Gram [5] into network representation learning firstly in binary network. Because DeepWalk did not establish a unified objective function, what's worse, the vertex sequences generated randomly are effected by a lot of noise. Hence, a recent work from Tang et al. [6] proposed to model

the first-order and second-order proximity respectively and then presented a simple method to incorporate these two representation vector. Furthermore, GraRep [7] proposed by Cao et al. took the N-order network structure into consideration, where $N > 2$. In order to address the randomness of vertex sequence in DeepWalk, nore2vec [8] added flexibility in exploring neighborhoods to learn richer network representations. As described above, these methods learned network representations only based on the structure information. Certainly, the available network information is not only the network topology but also other information. Thus, Yang et al. presented TADW [9] to model the text feature and network structure simultaneously with inductive matrix completion algorithm to learn better vertex representation. GENE [10] proposed by Chen et al. incorporated the group information into network embedding. Besides, Li et al. [11] proposed multi-faceted representation jointly with diverse information like user-generated contents, user attributes and user-user network graphs.

However, in real world, networks are usually sparse, that is, there are a small quantity of edges between vertex pair. Therefore, only utilizing the limited structure information is difficult to learn the network representation and to have a good performance in the following tasks. As we know, social networks produce not only the user-user network topology, but also other diverse information from each user like text information, image information, video information and emoji information. Assumed that those with similar user-generated text tend to have common interests and concerns, which indicates there exists latent relationship between them.

Hence, motivated by the sparseness problem and inspired by the aforementioned assumption, a feasible

method is to enhance the original network structure with rich user-generated content. Our work proposes how to incorporate the latent relationship extracted from text information into the original network structure and then learn the network representation with the revised network topology based on the classic network embedding model LINE [6]. Finally, we conduct several experiments to evaluate the performance of representation vector with our proposed method in gender and age inference tasks compared with the baseline method.

Generally speaking, the major contribution is that we extract the latent relationship from user-generated text to integrate into the raw network structure information and then learn the network representation with LINE [6], which addresses the sparseness problem of real-world networks.

2. Method

In this section, firstly, we give the definition of the problem. Then we describe the latent relationships extracted from user-generated content and the relationships which integrate the first-order proximity and the second-order proximity for the given network structure. Finally, we introduce how the initial network was revised on the basis of these relationships as above.

1.2 Definition

The vertex corresponding to a user in microblog network usually has rich text information which is published until now.

A network with rich text information is denoted as $G = (V, E, T)$, where $V = \{v_i\}$ is user vertex set and $E = \{(v_i, v_j)\}$ is the edges associated with a weight $w \in \{0, 1\}$. $T = \{t_i\}$ is user-generated blogs set where each entity t_i corresponds to a text paragraph of user v_i .

Therefore, our study aims to capture the latent information from user-generated content and to learn the low-dimensional representation $y_i \in \mathbb{R}_k$ from the revised network G'' for each vertex, where k is expected to be much smaller than $|V|$.

1.3 Latent Relationships from User-generated Content

Considering that users who publish similar blogs usually have several common interests, that is to say, they are likely to be friends or they have the potential to become friends. Thus, the relationship extracted from user-generated contents is named latent relationship.

The extraction of latent relationships can be essentially attributed to a matter of text similarity problem. However, each blog text is not more than 140 words, so it is necessary to integrate all the history blogs generated by each user into a text paragraph. Because the colloquial expression style of blog text makes text analysis vulnerable to noise data, text preprocessing is crucial for the accuracy of subsequent tasks and some targeted measures must be taken such as the stop words filtering, abnormal words replacement, and words segmentation. Here some specific preprocessing measures for user-published blogs are shown as follows,

Some texts separated by symbol “#” in microblog which are named the topic of this blog represents some interests of users, as a result, these topic phrases should be extracted directly used as the keywords for the corresponding user-generated texts.

Contents after symbol “@” usually represent user name and do not need to be cut further.

Filter the useless noise data in raw texts such as punctuation and special symbols.

Replace all abnormal words in contrast to the abnormal vocabulary. Abnormal words are some popular networks-parlances which have been widely recognized. For example, if you want say “thanks”, you can use “3Q” or “3q”. Also, in order to avoid some special expression, you can separate one Chinese character into two or more parts.

Replace the original complex form of simplified Chinese character in contrast to the simple word list.

Remove the stop words.

Calculate TF-IDF for corpus and filter out the low-frequency words according to the pre-set threshold.

HanLP is used to cut the remaining texts which have been preprocessed by above-mentioned measures.

Then we adopt the Latent Dirichlet Allocation (LDA) [2] model which identifies the latent topic information in a large corpus to extract the feature vector, at last, the cosine similarity between any two vectors indicates the weight of latent edge between the corresponding two users.

LDA is a generative probabilistic model covering document, topic and word layers, which is based on the idea that documents are represented as random mixtures over K latent topics where each topic is characterized by a multinomial distribution over words and each document subjects to a multinomial distribution over k latent topics. Hence, the generation process for each document w in a corpus \mathbb{D} consisting of M documents is described as follows,

For each document M_i , choose $\theta \sim \text{Dir}(\alpha)$, where $\text{Dir}(\alpha)$ is the Dirichlet distribution for parameter α and parameter θ is a topic vector, each column of which represents the probability that each topic appears in this document.

For w_{ij} which is the j^{th} word in the i^{th} document, where $j \in \{1, 2, \dots, N_i\}$ and $i \in \{1, 2, \dots, M\}$, then choose a latent topic z_i from the topic vector θ according to the conditional probability distribution $p(z_i|\theta)$ and generate a word w_j according to the conditional probability distribution $p(w_j|z_i, \beta)$.

Given the parameters α and β , the joint distribution of the model is defined as follows,

$$p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta) = p(\theta|\alpha) \prod_{i=1}^M p(z_i|\theta) p(w_j|z_i, \beta).$$

Here w are regarded as the observed variable while θ and z are as the hidden variables, then we use the Expectation Maximization Algorithm (EM) to learn parameters α and β .

Supposed that the top T topics are retained, then each text paragraph is embedded into a feature vector $t_{v_i} = \{w_1, w_2, \dots, w_T\}$, where w_i is the weight value

corresponding to the i^{th} topic and the weight denotes the likelihood that the texts generated by user v_i belongs to the j^{th} topic.

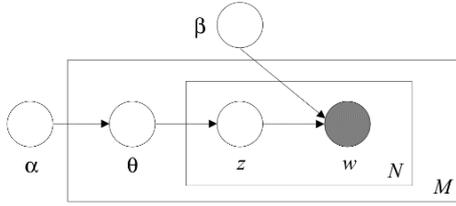


Figure 1. Graphical model representation of LDA which is composed of three layers. α and β are sampled only once during the generation process. The outer plate represents documents and parameter θ is sampled for each document, and therefore the probability that each document generates topic z is different. The inner plate represents the repeated choice of topics and words within a document and the word w is generated by z and β .

Afterwards, each feature vector describes the topics associated with each user-generated contents, in other words, the concerns or interests can be obtained from user-published blogs. Thus we extract the latent relationships from these representation vectors with cosine similarity. Certainly, other similarity measures such as Pearson Correlation Coefficient, Jaccard Coefficient and so on can also be used to calculate the similarity between individuals. Given two representation vector t_{v_i} and t_{v_j} , then w'_{ij} indicates the latent relationship between user v_i and user v_j , and it is defined as follows,

$$w'_{ij} = \frac{\vec{t}_{v_i} \cdot \vec{t}_{v_j}}{\|\vec{t}_{v_i}\| \cdot \|\vec{t}_{v_j}\|}.$$

Accordingly, the latent adjacency matrix extracted from user-generated blogs can be described as matrix $W' \in \mathbb{R}^{|V| \times |V|}$, where each entry $w'_{ij} \in [0,1]$.

1.4 Integrated Relationships from Network Structure

As we all know, real-world social network is usually sparse because there exists a small amount of edges which denotes the friendship between users. Moreover, the direct friendship is voluntarily established by users according to their own preferences so that it plays a significant role in network embedding problem only considered the network structures. However, the direct friendship is not enough to describe the network structure, what's more, it is possible that those who are not friends have something in common. As a matter of fact, there exists a fact that users who share similar friends tend to have common interests or similar characteristics in the social media.

Therefore, LINE [6] took the above facts into consideration and firstly introduced first-order and second-order proximity to characterize the local and global structure information as fully as possible.

First-order Proximity: Given the edge set E , for each pair of vertices in E , the weight on the corresponding edge indicates the first-order proximity. Let w_{ij}^1 be an entry of the first-order proximity matrix W^1 , and then w_{ij}^1 is shown as follows,

$$w_{ij}^1 = \begin{cases} 1, & \text{if } (v_i, v_j) \in E \\ 0, & \text{otherwise} \end{cases}.$$

Second-order Proximity: The number of common neighbors for any pair of vertices is used to define the second-order proximity, which describes the similarity of these two users' neighborhood structure in social network. Given the neighbor vertices set of user v_i and user v_j as \mathcal{N}_{v_i} and \mathcal{N}_{v_j} respectively, then the number of common neighbor vertex is calculated and the second-order proximity between user v_i and v_j is defined as follows,

$$w_{ij}^2 = \frac{2 \times |\mathcal{N}_{v_i} \cap \mathcal{N}_{v_j}|}{|\mathcal{N}_{v_i}| + |\mathcal{N}_{v_j}|}.$$

Now, we exploit the first-order and second-order proximity jointly into the adjacency matrix extracted from the network structure. Accordingly, we introduce W to donate the integrated adjacency matrix where each entry w_{ij} is composed of the two corresponding proximities and is shown as follows,

$$w_{ij} = \lambda \cdot w_{ij}^1 + \mu \cdot w_{ij}^2,$$

Where λ and μ are the normalized coefficient and their value is adjusted iteratively until the result reaches optimal.

1.5 Revised Network with Latent Relationships

In this section, we firstly describe how to revise the network structure with the latent relationships extracted from user-generated contents and then use LINE [6] to learn the embedding representation for the extended network topology. Besides, the original network G have two changes as a result of the revise for the original adjacency matrix: firstly, there exists latent friendship between users who are not friends originally. From the perspective of graph theory, that is, the weight w_{ij} on the edge of pair of vertex (v_i, v_j) changes from 0 to $w(w \in [0,1])$. Secondly, the relationship between users becomes stronger, and thus the weights of some edges increase.

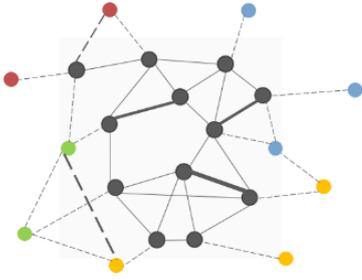


Figure 2. Schematic of enhanced network topology. The sub-graph of gray nodes is the original topology and other color nodes isolated at first. After revised, these edges indicated by dotted line are newly generated, notably, the thickness of line represents the different weight value.

Let W'' be the adjacency matrix of the revised network where each entry w''_{ij} is defined as follows,

$$w''_{ij} = \frac{w_{ij} + w'_{ij}}{\max_{i,j} \{w_{ij}, w'_{ij}\}}$$

However, some entries in the revised adjacency matrix are too small to make much sense and it is more reasonable to remove them. Therefore, we take the final revised adjacency matrix as the input data of LINE [6] to calculate the low-dimension representation. LINE [6] firstly introduced the first-order and second-order proximity and learned the corresponding representation vector for each vertex on the basis of first-order and second-order proximity separately, and then described how to incorporate these two representation vectors into the final vertex representation.

Essentially, the first-order proximity is the weight of edge between the vertices in the network. In order to model the first-order proximity, LINE [6] utilized the direct weight to build the empirical probability and then used the joint probability built by the representation vector to establish the object function. Suppose that \vec{v}_{i1} and \vec{v}_{i2} are the vector representation of vertex v_i and v_j separately, the joint probability between v_i and v_j was defined as follows,

$$p_1(v_{i1}, v_{j1}) = \frac{1}{1 + \exp(-\vec{v}_{i1} \cdot \vec{v}_{j1})}$$

Meanwhile, the empirical probability was defined as follows,

$$\widehat{p}_1(v_{i1}, v_{j1}) = \frac{w''_{ij}}{\sum_{(v_i, v_j \in E)} w''_{ij}}$$

Consequently, LINE [6] adopted Kullback-Leible (K-L) divergence to establish the object function, and then LINE learned network representation by minimizing the object function which was shown as follows,

$$\min \mathcal{L}_1 = \min D_{KL}(\widehat{p}_1(v_{i1}, v_{j1}) || p_1(v_{i1}, v_{j1})),$$

Where measured the difference between the two probability distributions \widehat{p} and p was defined as follows,

$$D_{KL}(\widehat{p} || p) = \sum_{x=1}^n \widehat{p}(x) \cdot \log \frac{\widehat{p}(x)}{p(x)}$$

In order to establish the second-order proximity model, LINE [6] assumed that each vertex play two roles which are the object vertex and the context of other vertices. Hence, \vec{v}_{i2} denotes the representation when v_i is the object vertex while \vec{v}'_{i2} denotes the representation vector when v_i is as the content of other vertices. Similarly, LINE [6] defined the conditional probability of “context” vertex v_j generated by “object” vertex v_i as follows,

$$p_2(v_j | v_i) = \frac{\exp(\vec{v}'_{j2} \cdot \vec{v}_{i2})}{\sum_{k=1}^{|V|} \exp(\vec{v}'_{k2} \cdot \vec{v}_{i2})}$$

Also, the empirical probability was defined as follows,

$$\widehat{p}_2(v_j | v_i) = \frac{w''_{ij}}{d_i}$$

where d_i is the out-degree of vertex v_i . Therefore, LINE [6] established the object function for the second-order proximity by minimizing the K-L divergence to obtain the corresponding network representation. Afterwards, these two representation vectors optimized by Negative Sampling Algorithm(NEG) [13] are put together to denote the final representation \vec{v}_i for each vertex v_i in LINE model, where $\vec{v}_i = \vec{v}_{i1} + \vec{v}_{i2}$.

2. Experiments

In this section, we conduct several experiments for the proposed methods and introduce the experimental results on a real dataset. Moreover, we use some inference tasks to evaluate the quality of the vector representation and compare our method with the baseline method.

2.1 Datasets

We conduct the experiments on dataset extracted from Sina Microblog provided by SMP CUP 2016. The dataset involves about 2.5 million users and covers 550 million friendships, and there are 4, 4000 users with microblogs texts and 3000 users with label among them. For the inference tasks, the dataset are divided into two subsets. One is used for training and the other is for testing. Besides, there are four kinds of data including user information, labels, relationships and user-published contents.

2.2 Baseline Methods

Since our proposed methods exploits the rich text information of each vertex to extend the original network structure, it is necessary to demonstrate the improvements over network embedding only considering the network topology information. However, the real-world social network structure is a binary network, that is, the weight value of each edge is either 0 or 1. Thus, we compare our methods with DeepWalk [3] and LINE only based on the

network structures to validate the performance. Inspired by Word2Vec [2], DeepWalk [3] treated the vertex sequence as the sentence in the corpus. Random walk was used to generate the standard input sequence and then Skip-gram model trained the above sequences to learn the latent representation for each vertex in the network.

2.3 Experiment Settings

For our proposed methods, the number of vertex $|V|$ in the dataset is 1560. The number of topic in LDA topic generation model is set as 60, in other words, the dimension of user-generated text vector is 60. For DeepWalk [3], the number of available vertex is 1155. We used the default parameters, for instance, windows size is 5, walk length is 40, number of random walks to start at each node is set as 10 and the number of latent dimensions to learn for each node is set as 64. For the negative sampling in LINE model, the number of negative samples is set as 5.

2.4 Experiment Results and Analysis

The representation vector extracted from user-generated contents according to LDA indicates the latent relationships. Here, the text representation corresponding user v_i is described as follows,

$$\vec{t}_{v_i} = \{topic_1:w_1, topic_2:w_2, \dots, topic_T:w_T\}$$

In order to visualize the performance of the representation vector, we choose the top three topics for each user-generated blogs and then calculate the corresponding coordinate values in the three axes by adding the weight value to the topic number value. Figure 3 shows the distribution of user-generated texts representation.

At the same time, we visualize the network structure to compare the changes of the network topology before and after revised as shown in Figure 4 and Figure 5.

For the dataset, only five percentage of users are statistically friends each other. Obviously, it can be found from Fig 6 that the second-order proximity is more than the first-order proximity, which is the same as our above-mentioned views for the real-world network.

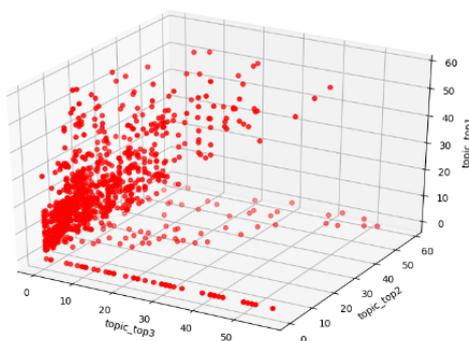
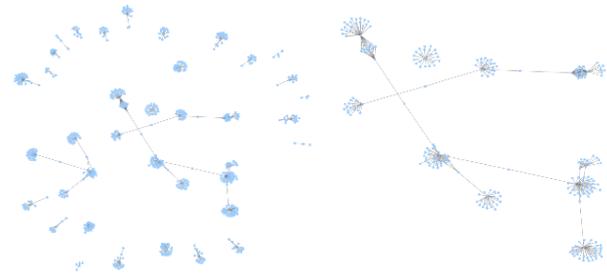


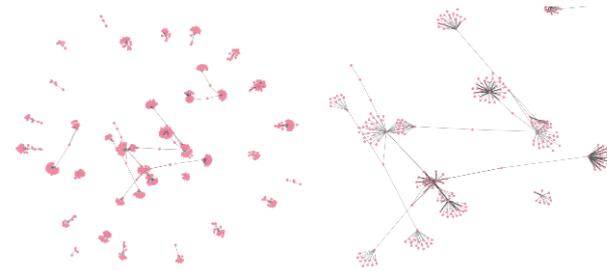
Figure 3. The distribution of user-generated texts representation. The clustering phenomenon of the representation vector indicates the similarity of the user-generated text.



(a). the whole network

(b). the sub-network

Figure 4. Visualization of the latent network topology with user-generated text.



(a). the whole network

(b). the sub-network

Figure 5. Visualization of the final network topology after revised by the user-generated text. Compare with Fig 4, the number of edge increase and the thicker lines indicates that the relationship gets stronger.

2.4.1 Gender Inference

In the real-world network, the task of gender inference is a supervised binary classification problem based on the representation vector generated by our proposed methods.

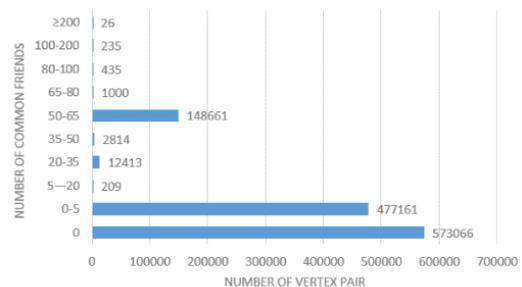


Figure 6. The number of common friends for each vertex pair. The quantity of common friends between user is different, especially, several of which is up to more than 200. Although the half of vertex pairs have no common friends, the second-order proximity plays an significant role in network topology.

Thus, we used linear kernel for SVM model and took the final representation vector as the extracted feature to train the gender classifier. Similarly, we used accuracy, precision, recall and F1-measure to analyses the classification results and evaluate the performance. Accuracy is a metric to measure the number of samples that are classified correctly in the test samples. For classification tasks, precision is referred to as positive predictive value while recall is referred to as the true positive rate or sensitivity. And F1-measure is the harmonic mean of precision and recall.

Table I. Results of Gender Inference Task on Baseline Method

Percentage	Accuracy	Precision	Recall	F-1
13.40%	67.10%	0.6710	1.0	0.8030
22.07%	71.00%	0.7100	1.0	0.8300
30.74%	73.20%	0.7340	0.996	0.8450
39.39%	72.30%	0.7230	1.0	0.8390
48.05%	72.80%	0.7420	0.9730	0.8420
Average	71.38%	0.7160	0.9938	0.8318

Table II. Results of Gender Inference Task on Our Method

Percentage	Accuracy	Precision	Recall	F-1
3.85%	75.00%	0.7500	1.0	0.8570
10.26%	73.40%	0.7340	1.0	0.8466
16.67%	73.10%	0.7320	0.9580	0.8300
23.08%	73.10%	0.7400	0.9730	0.8410
29.49%	72.40%	0.7430	0.9620	0.8380
35.90%	73.20%	0.7510	0.9620	0.8430
42.31%	72.30%	0.7540	0.9380	0.8360
48.72%	72.60%	0.7510	0.9470	0.8380
55.13%	72.70%	0.7520	0.9470	0.8380
Average	72.88%	0.7414	0.9652	0.8408

Table I shows the classification performance based on the DeepWalk model only considering the network structure information. And Table II shows the evaluation results on our proposed methods. From these tables, we can obviously find that the accuracy and precision has been significantly improved, in particular, the average precision increases by 3% based on our enhanced representation method. Moreover, the first column indicates the proportion of the test samples to the whole data set. As the proportion increases, the accuracy rate decreases slightly. As is known to all, the small the percentage, the more training samples, which means that the classifier trained by SVM is more accurate.

2.4.2 Age Inference

For the age inference task, if not considering the false age, the age range of users is extremely wide which vary from children to the older. And the most noteworthy is the majority is the teenager aged between 18 and 30 years old. In order to infer the ages of the test samples more accurately, we divided the age into four levels. Similarly, the inference task is also a supervised classification problem but a multi-labels classification. Therefore, we exploit SVM which is firstly a classic binary classification algorithm to deal this problem. Intuitively, there exists two ways to extend SVM classifier. One is one-to-one and the other is one-to-rest. So we used these two extended SVM algorithms [14] to accomplish the age inference task respectively and used accuracy metric to evaluate the performance of our proposed method compared with the baseline method.

Table III. Results of Age Inference Task on Baseline Method

Percentage	13.04%	21.74%	30.43%	39.13%	47.83%
Accuracy	69.03%	72.54%	70.70%	70.76%	71.53%
	65.82%	70.20%	71.55%	71.65%	70.63%

Table IV. Results of Age Inference Task on Our Method

Percentage	10.26%	23.08%	29.49%	42.31%	48.72%
Accuracy	76.25%	74.72%	75.43%	74.82%	73.95%
	77.50%	75.28%	73.70%	75.18%	73.16%

As is shown in both two tables, the accuracy in the first row is calculated by extended SVM classifier with one-to-one while the accuracy in the second row is with one-to-rest method. Compared the accuracy in Table III to those in Table IV, the result with our proposed method corresponding similar percentage has been greatly improved. For example, when the test sample accounts for about 10 percent of the sample overall, with the SVM algorithm extended by one-to-one method, the accuracy of inferring the age of test samples increased from 69.03% to 76.25%. Furthermore, we compared the results from age inference as described in Figure 7. As can be seen, the user representation extracted by the proposed method achieves better results than the baseline only considering network structure.

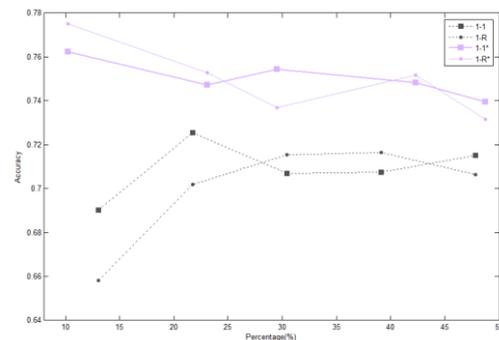


Figure 7. Comparison of age inference results. The gray is accuracy with baseline method while the pink is accuracy with our proposed methods.

3. Conclusion

In this paper, considering the sparseness of network structure in the real world, we study how to exploit other network information to extend the structure information and present a more explicit description for each user. We assume that the user-generated content shows his interests and the similar user-generated texts represent some latent relationships between user pairs. Hence, we propose an enhanced network representation method by integrating the latent relationships extracted from rich texts into the original network structure. Experimental results demonstrate that the representation learned by our proposed method achieves improvement in accuracy metric of user gender and age inference tasks in comparison with the baseline method. For future work, we intend to incorporate multi-faceted network information into the user model and use the feature vector to address other further applications such as hierarchical community detection based on node feature representation and so on.

Acknowledgment

The authors want to thank the relevant outstanding effort and the collaboration from my workmates in our research group. This work was supported by Virtual Community Discovery and Evolution (No 2013CB329606), the National Basic Research Program of China (973 Program No.2013CB329600) and the National Natural Science Foundation of China (NSFC No.61502517).

References

- [1] H. Zhou, W. U. Wei, Q. Z. Teng, X. M. Yang, L. I. Min, and D. Y. Tao, "Research of manifold learning algorithm," *Application Research of Computers*, vol. 24, no. 7, pp. 214–217, 2007.
- [2] R. Collobert and J. Weston, "A unified architecture for natural language processing: deep neural networks with multitask learning," in *International Conference*, 2008, pp. 160–167.
- [3] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: online learning of social representations," pp. 701–710, 2014.
- [4] R. Metzler and J. Klafter, "The random walk's guide to anomalous diffusion: a fractional dynamics approach," *Physics Reports*, vol. 339, no. 1, pp. 1–77, 2000.
- [5] J. R. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain," Ph.D. dissertation, Carnegie Mellon University, 1994.
- [6] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Largescale information network embedding," in *International Conference on World Wide Web*, 2015, pp. 1067–1077.
- [7] S. Cao, W. Lu, and Q. Xu, "Grarep: learning graph representations with global structural information," 2015, pp. 891–900.
- [8] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," 2016, p. 855.
- [9] C. Yang, Z. Liu, D. Zhao, M. Sun, and E. Y. Chang, "Network representation learning with rich text information," in *International Conference on Artificial Intelligence*, 2015, pp. 2111–2117.
- [10] J. Chen, Q. Zhang, and X. Huang, "Incorporate group information to enhance network embedding," in *ACM International on Conference on Information and Knowledge Management*, 2016, pp. 1901–1904.
- [11] J. Li, A. Ritter, and J. Dan, "Learning multi-faceted representations of individuals from heterogeneous evidence using neural networks," *Computer Science*, 2015.
- [12] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.
- [13] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," *Computer Science*, vol. 4, pp. 1188–1196, 2014.
- [14] C. W. Hsu and C. J. Lin, "A comparison of methods for multiclass support vector machines," in *IEEE TRANS. NEURAL NETWORKS*, 2002, pp. 415–425.