Multimodal Learning Based Approaches for Link Prediction in Social Networks

Feng Liu(✉), Bingquan Liu, Chengjie Sun, Ming Liu, and Xiaolong Wang

School of Computer Science and Technology,
Harbin Institute of Technology, Harbin, China
{fengliu,liubq,cjsun,mliu,wangxl}@insun.hit.edu.cn

Abstract. The link prediction problem in social networks is to estimate the value of the link that can represent relationship between social members. Researchers have proposed several methods for solving link prediction and a number of features have been used. Most of these models are learned with only considering the features from one kind of data. In this paper, by considering the data from link network structure and user comment, both of which could imply the concept of link value, we propose multimodal learning based approaches to predict the link values. The experiment results done on dataset from typical social networks show that our model could learn the joint representation of these datas properly, and the method MDBN outperforms other state-of-art link prediction methods.

Keywords: Link prediction · Social networks · Multimodal learning · Deep Belief Networks · Support vector classifier

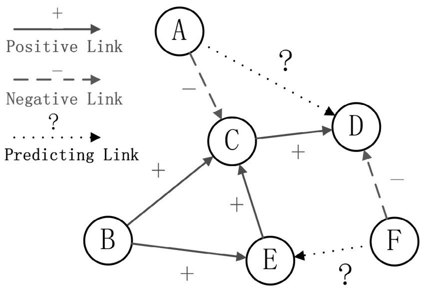
1 Introduction

With the explosive growth of SNS(Social Network Services) websites, there are large scale data of social media [3]. The mass data includes the interactions among social members, such as comments and links. The comment is always a short paragraph with only one or a few sentences, which are sent from one user to another. The link is usually a label with sign value that represent one user's certain kind of opinion to another, such as expressing support or oppose.

Taking social members as vertexes and links as directed edges between them, the link network can be represented as a graph. The classical link prediction is the problem of predicting the existence of a link between two entities, based on attributes of the objects and other observed links [6,13]. In this paper, the predicting task based on link network structure, as shown in Fig. 1(a), is to predict the relation of one user toward another from the evidence provided by their relations with other members from the surrounding social network.

Many state-of-art link prediction methods are based on machine learning models [20]. The machine learning based methods treat the link prediction problem as a classification task. Logistical Regression model is used to classify the

© Springer International Publishing Switzerland 2015
J. Li et al. (Eds.): NLPCC 2015, LNAI 9362, pp. 123–133, 2015.
DOI: 10.1007/978-3-319-25207-0_11



(a) predicting via link network

Link	Comments
(+)	Great contributions.
(+)	Unconvincing reasons for adminship, but generally good contributions.
(-)	User is too new, too "speedy"-happy.
(+)	Too smart to be an admin.
(-)	I do not trust him with the tools.
(?)	Solid answers, strong contributions, good content editor.
(?)	A good editor, but quite lacking in the areas admins inevitably should work in.

(b) predicting via user comments

Fig. 1. The Link Prediction Problem.

link values in [11,12]. Support vector machine is used to analyse how link network structure features effect link's values in [17]. Deep belief network based approaches for link prediction are introduced in [15,16]. However, these methods only used the data from link network structure.

The sentiment analysis or opinion mining is the computational study of people's opinions and attitudes toward entities and individuals [14]. Taking the 'link' as one user's opinion or attitude to another user, the link prediction task is some kind like the sentiment analysis or opinion mining as shown in Fig. 1(b). User comments are used in [2,5,21] to improve the performance of link prediction. It shows that the link prediction problem could be solved by using user comments with opinion mining methods.

As introduced in [19], a good multimodal learning model must satisfy certain properties. The joint representation must be such that similarity in the representation space implies similarity of the corresponding concepts. It is also desirable that the joint representation be easy to obtain even in the absence of some modalities. It should also be possible to fill-in missing modalities given the observed ones. In addition, it is also desirable that extracted representation be useful for discriminative tasks. Multimodal learning model focus on learning representations for speech audio which are coupled with videos of the lips is introduced in [18]. And multimodal models for image and text described the image are introduced in [10,22].

Assuming that one user have given good comments to the other, that user would be more likely to give a positive link. While the one who got bad comment are more likely to get a negative link. As a result, by learning the joint representation between the comments and links properly, we could predict the sign value of user links with high accuracy. At that same time, this joint representation could help to classify the polarity of user comment. This research would benefit link prediction and sentiment analysis, both of which are hot topic in study of social computing.

For the link prediction problem in this paper, the predicting via link network shown in Fig. 1(a), and predicting via user comments shown in Fig. 1(b), they imply similarity of the corresponding 'concept' as link value. This inspires us to use multimodal learning model to get the joint representation of link network features and user comments. And we assume that joint representation could improve the performance of link prediction based on only one kind of data. The multimodal learning based approaches for link prediction are proposed in this paper. The experiment results show that our method works well and the performance of link prediction is improved.

2 Background

In this section, we review the basic principles of RBM and DBN, which is used as a layer-wise building block for our models.

2.1 Restricted Boltzmann Machine

A Restricted Boltzmann Machine(RBM) is a neural network that contains two layers. It has a single layer of hidden units h that are not connected with each other. And the hidden units have undirected, symmetrical connections w to a layer of visible units v. As shown in Fig. 2(a), $\{V_1, W_1, H_1\}$ constructs a RBM. The model defines a probability distribution over v, h as

$$-\log P(v,h) \propto E(v,h) = -\sum_i v_i a_i - \sum_j h_j b_j - \sum_{i,j} v_i w_{ij} h_j \tag{1}$$

where a_i is the bias of visible unit i, and b_j is the bias of hidden unit j. When input a vector v $(v_1, v_2 ... v_i ...)$ to the visible layer, the binary state h_j of each hidden unit is set to 1 with probability by

$$p(h_j = 1|v) = \varphi(b_j + \sum_i v_i w_{ij}) \tag{2}$$

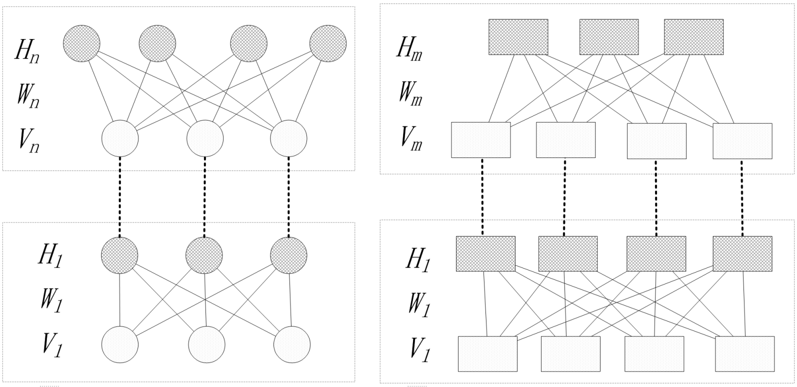
where $\varphi(x) = 1/(1 + e^{-x})$. When input a vector h $(h_1, h_2, ... h_j ...)$ to the hidden layer, the binary state v_i of each visible unit is set to 1 with probability by

$$p(v_i = 1|h) = \varphi(a_i + \sum_j h_j w_{ij}) \tag{3}$$

The parameters $\{w, a, b\}$ are usually trained by using the Contrastive Divergence(CD) learning procedure, which is introduced in [1,7].

2.2 Deep Belief Network

The Deep Belief Network(DBN) is a multilayer, stochastic generative model that is created by learning a stack of RBMs, as shown in Fig. 2. It can be understood as only one RBM may not have enough abstracting ability to solve some complex problems for there are only two layers in one RBM. And the two layers can be though as transforming the input(visible layer) into another space(hidden layer)



(a) link DBN (b) comment DBN

Fig. 2. The Structure of DBN.

only once, so one RBM's ability is limited. One DBN is built up with a stack of RBMs that could have more abstracting ability, because each layer of RBM in DBN can make a space transformation [9].

When learning a DBN, only the first RBM is trained on the original samples by CD learning procedure. Then the second RBM is trained by the first RBM's hidden activation vectors which are generated from the original samples. Do that iteratively until the top RBM is learned. This greedy, layer-by-layer learning can be repeated as many times as desired [8]. If a sample vector is imputed to the first RBM of that DBN, the highly abstracted vector of that sample would be gotten from the top RBM's hidden layer.

3 Link Prediction Problem

In this paper, the link prediction problem is defined as follows. Taking the whole network as a directed graph $G = (V, E)$, V is the set of users and E is the set of edges. Each edge linking two nodes has a sign value(either positive or negative). Supposing there are two nodes u and v and an edge linking from u to v. Denote that edge as $e(u, v)$ and assume the sign value of $e(u, v)$ is "lost". Supposing there is a sub graph G', whose edges have the same assumption as $e(u, v)$. Meanwhile, the sign values of edges in $G - G'$ are known. We infer the sign value of edges in G' by using the information from the structure of G and the patterns of link values from $G - G'$. For illustration, in Fig.1(a), a small part of the whole SSN is shown to illustrate the link prediction problem.

4 Methodology

In this section, we describe how to extract features from link networks and user comments. Then we introduce how to learn discriminative DBN for predicting link values and the method of MDBN.

4.1 Link Network Structure Features

The link network could be represented as a directed graph $G = (V, E)$, where V is the set of users and E is the set of edges, and each edge e has a link value (either positive or negative) as shown in Fig. 1(a). The edge directly linking from user u to v is denoted as $e(u, v)$. And denote $Ne(u)$ as the set of u's neighbour nodes and $CNe(u, v)$ as the common neighbours shared with u and v.There are total 26 link network structure features, including node features and neighbour features.

The node features for u is counting the in-degrees and out-degrees with sign values. Denote $d^+_{in}(u)$ for positive in-degree, $d^-_{in}(u)$ for negative in-degree, $d^+_{out}(u)$ for positive out-degrees and $d^-_{out}(u)$ for negative out-degrees. There are 8 node features.

The neighbour features includes statistical information from $CNe(u, v)$, such as $c^N_{Ne}(u, v)$ is the number of nodes w in $CNe(u, v)$ and $c^E_{Ne}(u, v)$ is the number of edges between w and u, v. Then select any node w from $CNe(u, v)$, whose edges could have any direction with any sign value connected with u and v, denote $c(u\overset{+}{\to}w\overset{+}{\leftarrow}v)$ as the number of nodes who get positive links from both u and v. There are 2 directions and 2 kinds of sign values, so the relationships of u, v and w can be divided into 16 kinds. There are total 18 neighbour features.

4.2 User Comment Features

In some online social networks, users are allowed to make some comments when they tag a link to others, such as Wikipedia[1]. A small part of Wikipedia contributors are administrators, who are users with access to additional technical features that aid in maintenance. In order for a user to become an administrator a Request for Adminship (RfA) is issued and the Wikipedia community via a public discussion or a vote decides who to promote to adminship. Some examples of the RfA votes are shown in Fig. 1(b). The vote can be represented as a link with sign value, and users can also make some comments in their votes.

The text, such as comment, could be represented by the Bag of Words model(BOW). It treats each comment as the bag of its words, and represents text as a vector of words via the word dictionary. The word dictionary Dic contains all the appeared words, and the dictionary size is $lenDic$. The set of words appeared in comment from u to v is denoted as $W(u, v)$. Then build a word vector $w(u, v)$ with dimension $lenDic$, and set the ith position to '1' if the Dic's $ith\ word \in W(u, v)$, while all other positions are set to '0'.

The $lenDic$ is always very large for there are so many words appeared in comments. But most of these words are only used few times, and it makes $w(u, v)$ very sparse. Because the number of first layer RBM's visible units should be equal to $lenDic$. We select $number$ of top frequency words to build another Dic named as $top\ number$, and the new $lenDic$ is equals to $number$.

[1] https://www.wikipedia.org

4.3 Discriminative Deep Belief Networks

In order to use the unsupervised learned DBN to solve link prediction problem , we added a layer of linear output units for class labels at the top of link DBN and comment DBN. Each of the output unit stands for a class label, and the sample's label should be the output unit with the largest value. This output layer works as a linear Softmax classifier, when input a sample vector v to the bottom of DBN, the value of the jth output unit is

$$O_j(v) = \sum_{h_i \in H_{top}} w_{ij} h_i \tag{4}$$

where H_{top} is the hidden layer activations of the top RBM in DBN, and can be calculated by Eq.2 though all the layers by inputting v into the bottom RBM. And w is the weights between H_{top} and output layer.

Then the possibility of the class label of input vector v is

$$p(label = o_i|v) = \frac{e^{O_i}}{\sum\limits_j e^{O_j}} \tag{5}$$

The Softmax classifier is learned with minimizing the cross-entropy loss error as $-\sum_i y_i \log p_i$ where y_i is the class label and p_i is the predicted link value. In order to get a better classification performance, the unsupervised DBN is fine-tuned when we update the output layer's weights.

4.4 Multimodal Deep Belief Networks

There are two approaches to learn a Multimodal Deep Belief Networks(MDBN) model, a shallow one as shown in Fig. 3 and a deep one as shown in Fig. 4. The circle neural 'Networks Unit' means this neural unit is learned with link network structure data, the rectangle neural 'Comment Unit' means this neural unit is learned with user comment data, and the diamond neural 'Multimodal Unit' means this unit is learned with both kinds of data. The main difference between the two approaches is that the shallow one is learned by original data, while the deep one is based on well learned DBNs, which are used for discrimination in Sect.4.3.

The shallow MDBN is a direct approach that trains a DBN over the concatenated link network data and user comment data. We joint the vector of link structure features and word vector as the input for the first layer RBM, and train the shallow MDBN with the greedy layer-by-layer learning, as shown in Fig. 3. As the RBM can model the distribution of input samples, the 1st RBM could represent the joint distribution of the link network and user comments. Then the joint input is abstracted by above RBMs. However, the correlations between the link network and user comments are highly non-linear. We found it is difficult for the 1st RBM to represent their relations properly, so we designed deep MDBN by taking the learned DBNs into account.

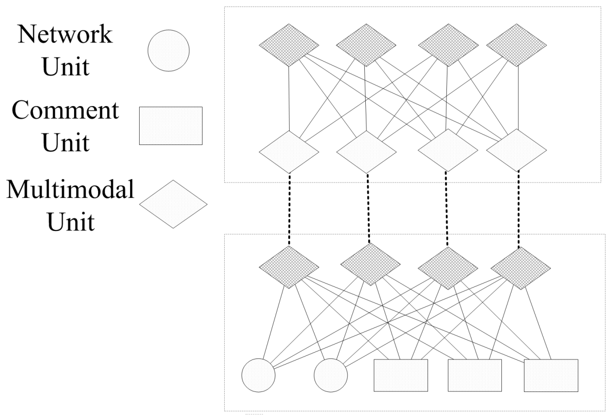


Fig. 3. Structure of Shallow Multimodal Deep Belief Networks.

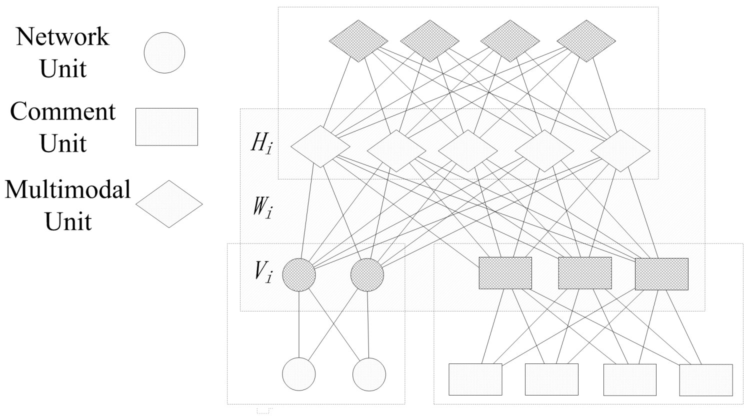


Fig. 4. Structure of Deep Multimodal Deep Belief Networks.

As introduction in Sect.2.2, well learned DBN could highly abstracted the inputs, and the results of discriminative DBN show that the represented inputs become linear classifiable. It inspires us to learn a multimodal model on higher RBMs. As shown in Fig. 4, we reuse the DBNs learned for link prediction introduced in Sect.4.3 as bottom RBMs. The link structure features is represented by the link DBN(in Fig. 2(a)) firstly, and the user comment word vector is represented by the comment DBN (in Fig. 2(b)). Then we joint the highly abstracted features together, shown as V_i in Fig. 4, and use it to learn higher RBMs. It can be easier for the model to learn higher-order correlations between link network data and user comments data.

5 Experiments and Analysis

5.1 Experiment Setup

In our experiments, the social media data of Wikipedia RfA prepared by West [21] is used. Before building the comment word vector, we removed the words such as 'support', 'supporting', 'supported' and similar words for 'oppose'. Removing such words is because that these words could imply the sentiment ploarity directly, and the model may nelgect all the other words. To avoid the samples imbalance effects, we randomly selected 50000 balanced samples. We use 40000 samples for training and the other 10000 for testing.

In order to make a more completely comparison with other state-of-art link prediction methods. We learned Support Vector based Classifier(SVC), which is widely used for link prediction [20], on the same dataset and features. We extracted features by programming in Python, train DBNs and MDBNs in Matlab, and the toolkit 'LIBLINEAR' by [4] is used for learning SVCs. The parameters of LIBLINEAR is set as $(`-s\ 1\ \ -c\ 1\ \ -B\ 1')$. The parameter $-s\ 1$ means using L2-regularized L2-loss support vector classification model, c is the cost and $-B\ 1$ means adding bias to the samples.

5.2 Experiment Results and Analysis

The experiment results are shown in Table 1. The first column is the name of method. We use 2 rows to introduce a DBN or MDBN, the first row is name and the second row is the network structure as dimension of each layer in RBM. The shallow MDBN has $26(link) + 2000(word)$ as input dimension. And the deep MDBN, which reused the link DBN and comment DBN, has $100(represented\ link) + 500(represented\ word)$ visible units. The second column is the features used to learn the model, and top 2000 means word vector built with top 2000 frequency words.

The experiment results is shown in Table 1. The first column is the name of method. SVC it a multi-class classifier based on support vectors. We use 2 rows to introducing a DBN or MDBN, the first row is name and the second row is the network structure as dimension of each layer in a RBM. For example, (26-25-50-100) means the first RBM has a visible layer of 26 units and 25 hidden units. Then the second RBM has 25 visible units and 50 hidden units, and the top RBM has 50 visible units and 100 for hidden. The first RBM always has a visible layer with the same dimension as the feature vector's. As a result, the shallow MDBN has $26(link) + 2000(word)$ as input dimension. And the deep MDBN, which reused the link DBN and comment DBN, has $100(representedlink) + 500(representedword)$ visible units. The second column is the features used to learn the model, including link structure features, word vector features, word vector built with top 2000 frequency words, and the combine of them.

First of all, the models learned on multiple source of data outperform the ones learned on single source of data. Both the results of SVC with concatenated features and MDBN have better accuracy than using only one kind of features.

Table 1. Experiment Results

Methods	Features	Accuracy %
SVC	link structure	81.17
SVC	word vector (all words)	84.81
SVC	word vector (top 2000)	82.65
SVC	link + all words	85.96
discriminative DBN (26-25-50-100)	link structure	82.53
discriminative DBN (2000-2000-1000-500)	word vector (top 2000)	85.22
shallow MDBN (2026-2000-1000-500)	link + top 2000	87.75
deep MDBN (600-600-800-1000)	link + top 2000	88.50

It shows that use the joint representation from different data spaces, which imply similarity concepts, could improve models' performance.

Secondly, by learning on the same data, discriminative DBN outperforms SVC. What is more, the DBN with top 2000 frequency words has better performance than SVC learned with all words. It means that the highly abstracted features are more suitable for discrimination. As the discriminative DBN's output layer is linear unit in our experiment, it means that the original features become more linear classifiable after transformation by DBN.

Thirdly, the MDBN's performance is best of all methods, it shows that our multimodal learning method is effective for link prediction problem. Both the shallow and deep MDBN have good performance, and the deep one has about 0.8% higher accuracy. This result shows it is better to learn a multimodal model on higher RBMs. Because the correlations between data from different space became easier for learning after they abstracted by DBN.

Another aspect need to care is the computing cost. Learning RBMs is a time and space costly process for a large number of weights need to be adjusted. We run the experiments on a 4 core 3.5 Ghz CPU and 16GB RAM, and learning the deep MDBN needs about 40 hours. So the SVC learned with concatenated features maybe a choice when fast training is required. It cost less than 10 minutes with an acceptable performance.

6 Conclusion

In this paper, we proposed multimodal learning approaches for predicting link values in social networks. By taking both the link network structure and user comment data into account, the MDBN outperforms state-of-art link prediction methods, such as support vector based classifier and discriminative deep belief network. From the analysis of the different network structure of MDBNs, shallow and deep, we found that it is better to learn a multimodal model on data represented by RBMs.

Our further work includes try to find some other data that imply link values. And we would try some other method for multimodal learning.

Acknowledgement. This work is supported by the National Natural Science Foundation of China (61272383 and 61300114), Specialized Research Fund for the Doctoral Program of Higher Education (No. 20132302120047), the Special Financial Grant from the China Postdoctoral Science Foundation (No.2014T70340), China Postdoctoral Science Foundation (No.2013M530156), and Natural Science Foundation of Heilongjiang Province(F201132).

References

1. Carreira-Perpinan, M.A., Hinton, G.E.: On contrastive divergence learning. In: Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics. vol. 2005, pp. 17–24. Society for Artificial Intelligence and Statistics, Savannah Hotel, Barbados (2005)
2. Chelmis, C., Prasanna, V.K.: Social link prediction in online social tagging systems. ACM Transactions on Information Systems (TOIS) **31**(4), 20–46 (2013)
3. Demailly, C.: The business impacts of social networking. AT&T White Papers, pp. 1–8 (2008)
4. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. The Journal of Machine Learning Research **9**, 1871–1874 (2008)
5. Freno, A., Garriga, G., Keller, M.: Learning to recommend links using graph structure and node content. In: Neural Information Processing Systems Workshop on Choice Models and Preference Learning, pp. 1–7. NIPS (2011)
6. Getoor, L., Diehl, C.P.: Link mining: a survey. ACM SIGKDD Explorations Newsletter **7**(2), 3–12 (2005)
7. Hinton, G.E.: Training products of experts by minimizing contrastive divergence. Neural Computation **14**(8), 1771–1800 (2002)
8. Hinton, G.E.: To recognize shapes, first learn to generate images. Progress in Brain Research **165**, 535–547 (2007)
9. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
10. Kiros, R., Salakhutdinov, R., Zemel, R.: Multimodal neural language models. In: Proceedings of the 31st International Conference on Machine Learning (ICML 2014), pp. 595–603 (2014)
11. Leskovec, J., Huttenlocher, D., Kleinberg, J.: Predicting positive and negative links in online social networks. In: Proceedings of the 19th International Conference on World Wide Web, pp. 641–650. ACM (2010)
12. Leskovec, J., Huttenlocher, D., Kleinberg, J.: Signed networks in social media. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1361–1370. ACM (2010)

13. Liben-Nowell, D., Kleinberg, J.: The link-prediction problem for social networks. Journal of the American Society for Information Science and Technology **58**(7), 1019–1031 (2007)
14. Liu, B., Zhang, L.: A survey of opinion mining and sentiment analysis. In: Mining Text Data, pp. 415–463. Springer (2012)
15. Liu, F., Liu, B., Sun, C., Liu, M., Wang, X.: Deep learning approaches for link prediction in social network services. In: Neural Information Processing, pp. 425–432. Springer (2013)
16. Liu, F., Liu, B., Sun, C., Liu, M., Wang, X.: Deep belief network-based approaches for link prediction in signed social networks. Entropy **17**(4), 2140–2169 (2015)
17. Liu, F., Liu, B., Wang, X., Liu, M., Wang, B.: Features for link prediction in social networks: A comprehensive study. In: 2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 1706–1711. IEEE (2012)
18. Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., Ng, A.Y.: Multimodal deep learning. In: Proceedings of the 28th International Conference on Machine Learning (ICML-11), pp. 689–696 (2011)
19. Srivastava, N., Salakhutdinov, R.R.: Multimodal learning with deep Boltzmann machines. In: Advances in neural information processing systems, pp. 2222–2230 (2012)
20. Wang, P., Xu, B., Wu, Y., Zhou, X.: Link prediction in social networks: the state-of-the-art. Science China Information Sciences, pp. 1–38 (2014)
21. West, R., Paskov, S.H., Leskovec, J., Potts, C.: Exploiting social network structure for person-to-person sentiment analysis. Transactions of the Association of Computational Linguistics **2**(1), 297–310 (2014). http://aclweb.org/anthology/Q14-1024
22. Xing, E.P., Yan, R., Hauptmann, A.G.: Mining associated text and images with dual-wing harmoniums, pp. 1–9 (2012). CoRR, arxiv:abs/1207.1423