# An Overview on Network Representation Learning

**S Jafar Ali Ibrahim[1], B Siva Reddy[2], K Venkata Yaswanth[3], K Muneswar[4], S Jaswanth[5]**

1 Professor, Dept of CSE, Qis College of Engineering and Technology, Ongole, Prakasam (Dt)
2, 3, 4, 5 Students, Dept of CSE, Qis College of Engineering and Technology, Ongole, Prakasam (Dt)

***ABSTRACT***: *Representation learning has proven its usefulness in many activities such as photography and text mining. The goal of network representation learning is to learn distributed vector representation for each vertex in the networks, an essential feature of network analysis is now increasingly recognised. Some techniques of network representation research network systems for learning. In effect, vertices of the network contain rich data (such as text), that cannot be used with the traditional algorithmic frameworks. We suggest DeepWalk in text-associated form, by showing that DeepWalk, a high-tech network representation solution, is equal to matrix factorisation (TADW). In the context of matrix factorisation, TADW introduce text features of vertices in network representation research. Through applying them to the multi classifying of vertices, we compare our system and different baseline methods.The experimental results show that, our method outperforms other baselines on all three datasets, especially when networks are noisy and training ratio is small.*

***INDEX TERMS*** Network representation learning, recommendation algorithm

## I. INTRODUCTION

In the era of big data, it turns out to be progressively hard to retrieve related data from gigantic unstructured data. Subsequently recommender frameworks have become a viable way to settle the problem of data over-burden. As of late, such exploration headings have drawn extraordinary consideration from the scholarly community and industry. Commonplace utilizations of recommender frameworks incorporate Amazon's item proposal, Netix's film suggestion, last. fm's music proposal, LinkedIn's companion suggestion, and Google's news suggestion. Communitarian separating (CF) [2] is the most generally utilized proposal strategy in the examination of recommender frameworks. Nonetheless, the issues of information sparsity and cold beginning have altogether negative effect on the presentation of cooperative sifting techniques.

For instance, attributable to information sparsity, conventional cooperative separating calculations can't precisely compute the similitudes between clients or between things; or can't precisely learn inert client and thing highlight vectors from clients' past exercises. The development of interpersonal organizations carries an occasion to lighten the issues of information sparsity and cold beginning in conventional community separating calculations. A few analysts use the rich data contained in interpersonal organizations to propose some informal community based suggestion calculations. Run of the mill interpersonal organization based proposal calculations incorporate SoRec [3], RSTE [4], SocialMF [5], TrustMF [6], etc. Informal organization based suggestion calculations by and large accept that clients with trust relations as a rule share basic interests. Nonetheless, in the first informal community, the trust relationship is normally double, that is, just 0 or 1 is utilized to mean the trust connection between clients where 1 signifies there is a trust connection between two clients, and the level of trust is 1 and 0 shows that there is no trust connection between clients. Naturally, the granularity of such a portrayal is too coarse to even consider specifying the various degrees of trust among clients.. Indeed, numerous clients are probably going to confide in each other due to their shared associations, in spite of the fact that they have not assembled any immediate trust associations. During the time spent planning suggestion models, the nature of proposal calculations can be upgraded by thinking about such aberrant and verifiable trust connections. Be that as it may, such understood trust connections between clients are regularly overlooked in the conventional interpersonal organization based suggestion models. To handle the above issues, this exploration professional proposal calculation. Specifically, we initially embrace an organization portrayal strategy [7] to install informal community into a low-dimensional space, and afterward use the low-dimensional portrayals of clients to surmise fine-grained and thick trust connections between clients. At last, we coordinate the fine-grained and thick trust connections into the informal organization based proposal model to learn inactive component vectors of clients and things all the more correctly. The observational outcomes on genuine world datasets show that our proposed approach out-performs customary interpersonal organization based suggestion calculations.

## **II.** RELATED WORK

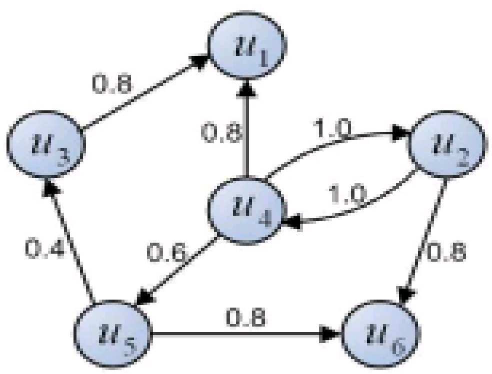
In this section, we review a few significant methodologies for recommender frameworks, particularly for community sifting. Two sorts of community separating approaches are generally examined: memory-based and model-based.

The memory-based methodologies are the most famous expectation techniques and are broadly embraced in business synergistic separating frameworks [12, 16]. The most examined instances of memory-based community oriented separating incorporate userbased approaches [2, 7, 10, 15] and thing based methodologies [4, 12, 19]. Client based methodologies anticipate the appraisals of dynamic clients dependent on the evaluations of comparative clients found, and itembased approaches foresee the appraisals of dynamic clients dependent on the registered data of things like those picked by the dynamic client. Client based and thing based methodologies regularly utilize the PCC calculation [16] and the VSS calculation [2] as the similitude calculation strategies. PCC-based synergistic sifting for the most part can accomplish better than the other mainstream calculation VSS, since it considers the distinctions of client rating style.

In the model-based methodologies, preparing datasets are utilized to prepare a predefined model. Instances of model-based methodologies incorporate the bunching model [21], viewpoint models [8, 9, 20] and the dormant factor model [3]. [11] Presented a calculation for community oriented sifting dependent on various leveled grouping, which attempted to adjust heartiness and precision of expectations, particularly when not many information were accessible. [8] Proposed a calculation dependent on a speculation of probabilistic idle semantic examination to persistent esteemed reaction factors. As of late, a few network factorization strategies [15, 17, 18, 20] have been proposed for cooperative separating. These techniques all emphasis on fitting the client thing rating grid utilizing low-position approximations, and use it to make further expectations. The reason behind a low-dimensional factor model is that there is just few elements affecting inclinations, and that a client's inclination vector is dictated by how each factor applies to that client.

All the above techniques for recommender frameworks depend on the supposition that clients are autonomous and indistinguishably circulated, and overlook the social exercises between clients, which isn't predictable with the truth that we regularly approach companions for suggestions. In view of this instinct, numerous specialists have as of late began to investigate trust-based recommender frameworks. In [14], a trust-mindful shared separating strategy for recommender frameworks is proposed. In this work, the community separating measure is educated by the standing of clients which is processed by spreading trust. Trust esteems are figured notwithstanding comparability measures between clients. The investigations on an enormous genuine dataset shows that this work expands the inclusion (number of appraisals that are unsurprising) while not diminishing the exactness (the blunder of forecasts). Bedi et al. in [1] proposed a trust-based recommender framework for the Semantic Web; this framework runs on a worker with the information circulated over the organization as ontologies, and utilizations the Web of trust to produce the proposals. These strategies are all memory-based techniques which utilize just heuristic calculations to create suggestions. There are a few issues with this methodology, nonetheless. The connection between the trust organization and the client thing network has not been concentrated deliberately. Also, these techniques are not adaptable to extremely enormous datasets since they may have to figure the pair astute client similitudes and pair insightful client trust scores.

In this paper, by directing dormant factor investigation utilizing probabilistic framework factorization, we get familiar with the client idle component space and thing inert element space by utilizing a client informal community and a client thing network all the while and consistently. Albeit as of late, comparative factor investigation techniques have been utilized in [7, 8] for archive recovery and report characterization, our methodology has three basic contrasts contrasted and these strategies: (1) Our strategy can manage missing worth issue, while their strategies can't. (2) Our technique is deciphered utilizing a probabilistic factor examination model. (3) Complexity investigation shows that our strategy is more proficient than their techniques and can be applied to extremely huge datasets.




(a) Social Network Graph

| | $i_1$ | $i_2$ | $i_3$ | $i_4$ | $i_5$ | $i_6$ | $i_7$ | $i_8$ |
|---|---|---|---|---|---|---|---|---|
| $u_1$ | 5 | 2 | | 3 | | 4 | | |
| $u_2$ | 4 | 3 | | | 5 | | | |
| $u_3$ | 4 | | 2 | | | | 2 | 4 |
| $u_4$ | | | | | | | | |
| $u_5$ | 5 | 1 | 2 | | 4 | 3 | | |
| $u_6$ | 4 | 3 | | 2 | 4 | | 3 | 5 |

(b) User-Item Matrix

| | $i_1$ | $i_2$ | $i_3$ | $i_4$ | $i_5$ | $i_6$ | $i_7$ | $i_8$ |
|---|---|---|---|---|---|---|---|---|
| $u_1$ | 5 | 2 | 2.5 | 3 | 4.8 | 4 | 2.2 | 4.8 |
| $u_2$ | 4 | 3 | 2.4 | 2.9 | 5 | 4.1 | 2.6 | 4.7 |
| $u_3$ | 4 | 1.7 | 2 | 3.2 | 3.9 | 3.0 | 2 | 4 |
| $u_4$ | 4.8 | 2.1 | 2.7 | 2.6 | 4.7 | 3.8 | 2.4 | 4.9 |
| $u_5$ | 5 | 1 | 2 | 3.4 | 4 | 3 | 1.5 | 4.6 |
| $u_6$ | 4 | 3 | 2.9 | 2 | 4 | 3.4 | 3 | 5 |

(c) Predicted User-Item Matrix

Figure 1: Example for Toy Data

## III. SOCIAL RECOMMENDATION FRAMEWORK

In this section, we first demonstrate our social recommendation framework using a simple but illustrative toy example. Then we introduce the factor analysis method for social recommendation using probabilistic matrix factorization.

### A. Toy Example

Let us first consider the typical social network graph in Fig. 1(a). There are 6 users in total (nodes, from u1 to u6) with 8 relations (edges) between users in this graph, and each relation is associated with a weight wij in the range [0, 1] to specify how much user ui knows or trusts user uj . In an online social network Web site, the weight wij is often explicitly stated by user ui. As illustrated in Fig. 1(b), each user also rates some items (from i1 to i8) on a 5-point integer scale to express the extent of favor of each item. The problem we study in this paper is how to predict the missing values of the user-item matrix effectively and efficiently by employing two different data sources. As mentioned in Section 1, motivated by the intuition that a user's social connections will affect this user's behaviors on the Web, we therefore factorize the social network graph and user-item matrix simultaneously and seamlessly using U T Z and U T V , where the shared low-dimensional matrix U denotes the user latent feature space, Z is the factor matrix in the social network graph, and V represents the low-dimensional item latent feature space. If we use 5 dimensions to perform the matrix factorization for social recommendation, we obtain

$$U = \begin{bmatrix} 1.55 & 1.22 & 0.37 & 0.81 & 0.62 & -0.01 \\ 0.36 & 0.91 & 1.21 & 0.39 & 1.10 & 0.25 \\ 0.59 & 0.20 & 0.14 & 0.83 & 0.27 & 1.51 \\ 0.39 & 1.33 & -0.43 & 0.70 & -0.90 & 0.68 \\ 1.05 & 0.11 & 0.17 & 1.18 & 1.81 & 0.40 \end{bmatrix},$$

$$V = \begin{bmatrix} 1.00 & -0.05 & -0.24 & 0.26 & 1.28 & 0.54 & -0.31 & 0.52 \\ 0.19 & -0.86 & -0.72 & 0.05 & 0.68 & 0.02 & -0.61 & 0.70 \\ 0.49 & 0.09 & -0.05 & -0.62 & 0.12 & 0.08 & 0.02 & 1.60 \\ -0.40 & 0.70 & 0.27 & -0.27 & 0.99 & 0.44 & 0.39 & 0.74 \\ 1.49 & -1.00 & 0.06 & 0.05 & 0.23 & 0.01 & -0.36 & 0.80 \end{bmatrix},$$

Since this example is a toy example, we cannot evaluate the accuracy of the prediction. However, the experimental analysis in Section 4 based on Epinions dataset tests the effectiveness of our approach. In the following sections, we will present the details of how we conduct factor analysis for social recommendation using probabilistic matrix factorization.

**B. Social Network Matrix Factorization**

Suppose we have a directed social network graph G = (V, E), where the vertex set V = {vi} m i=1 represents all the users in a social network and the edge set E represents the relations between users. Let C = {cik} denote the m × m matrix of G, which is also called the social network matrix in this paper. For a pair of vertices, vi and vk, let cik ∈ (0, 1] denote the weight associated with an edge from vi to vk, and cik = 0, otherwise. The physical meaning of the weight cik can be interpreted as how much a user i trusts or knows user k in a social network. Note that C is an asymmetric matrix, since in a social network, especially in a trust-based social network, user i trusting k does not necessary indicate user k trusts i.

The idea of social network matrix factorization is to derive a high-quality l-dimensional feature representation U of users based on analyzing the social network graph G. Let U ∈ R l×m and Z ∈ R l×m be the latent user and factor feature matrices, with column vectors Ui and Zk representing user-specific and factor-specific latent feature vectors, respectively. We define the conditional distribution over the observed social network relationships as

$$p(C|U,Z,\sigma_C^2) = \prod_{i=1}^{m}\prod_{k=1}^{m} \mathcal{N}\left[\left(c_{ik}|g(U_i^T Z_k),\sigma_C^2\right)\right]^{I_{ik}^C},$$

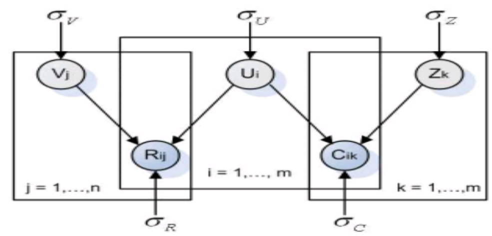



Figure 2: Graphical Model for Social Recommendation

In online social networks, the value of cik, which is mostly explicitly stated by user i with respect to user k, and it cannot accurately describe the relations between users since it contains noises and it ignores the graph structure information of social network. For instance, similar to the Web link adjacency graph in [26], in a trust-based social network, the confidence of trust value cik should be decreased if user i trusts lots of users.

**C. Matrix Factorization for Social Recommendation**

As analyzed in Section 1, in order to reflect the phenomenon that a user's social connections will affect this user's judgement of interest in items, we model the problem of social recommendation using the graphical model described in Fig. 2, which fuses both the social network graph and the user-item rating matrix into a consistent and compact feature representation. Based on Fig. 2, the log of the posterior distribution for social recommendation is given by

$$
\begin{aligned}
&\ln p(U, V, Z | C, R, \sigma_C^2, \sigma_R^2, \sigma_U^2, \sigma_V^2, \sigma_Z^2) = \\
&-\frac{1}{2\sigma_R^2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij}^R (r_{ij} - g(U_i^T V_j))^2 \\
&-\frac{1}{2\sigma_C^2} \sum_{i=1}^{m} \sum_{k=1}^{m} I_{ik}^C (c_{ik}^* - g(U_i^T Z_k))^2 \\
&-\frac{1}{2\sigma_U^2} \sum_{i=1}^{m} U_i^T U_i - \frac{1}{2\sigma_V^2} \sum_{j=1}^{n} V_j^T V_j - \frac{1}{2\sigma_Z^2} \sum_{k=1}^{m} Z_k^T Z_k \\
&-\frac{1}{2} \left( \left( \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij}^R \right) \ln \sigma_R^2 + \left( \sum_{i=1}^{m} \sum_{k=1}^{m} I_{ik}^C \right) \ln \sigma_C^2 \right) \\
&-\frac{1}{2} \left( m l \ln \sigma_U^2 + n l \ln \sigma_V^2 + m l \ln \sigma_Z^2 \right) + C, \qquad (8
\end{aligned}
$$

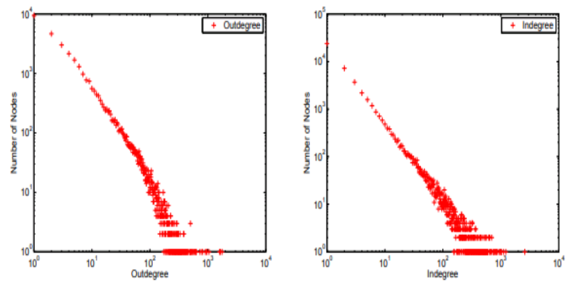



Figure 3: Degree Distribution of User Social Network

where C is a constant that does not depend on the parameters. Maximizing the log-posterior over three latent features with hyper parameters (i.e. the observation noise variance and prior variances) kept fixed is equivalent to minimizing the following sum-of-squared-errors objective functions.

## IV. EXPERIMENTAL RESULTS AND ANALYSIS

Here "-" indicates TSVM cannot converge in 12 hours because of low quality of representation (TSVM can always converge in 5 minutes for TADW). We did not show the results of semi-supervised learning on Wiki dataset because supervised SVM has already attained a competitive and even better performance with small training ratio on this dataset. Thus we only report the results of supervised SVM for Wiki. Wiki has much more classes than the other two datasets, which requires more data for sufficient training; hence we set the minimum training ratio to 3%. From these tables, we have following observations:

(1) TADW consistently outperforms all the other baselines on all three datasets. Furthermore, TADW can beat other baselines with 50% less training data on Cora and Citeseer datasets. These experiments demonstrate that TADW is effective and robust.

(2) TADW has more significant improvement for semi supervised learning. TADW outperforms the best baseline, i.e. naive combination, by 4% on Cora and 10% to 20% on Citeseer. This is because the quality of network representations is poor on Citeseer, while TADW is more robust for learning from noisy data than naive combination.

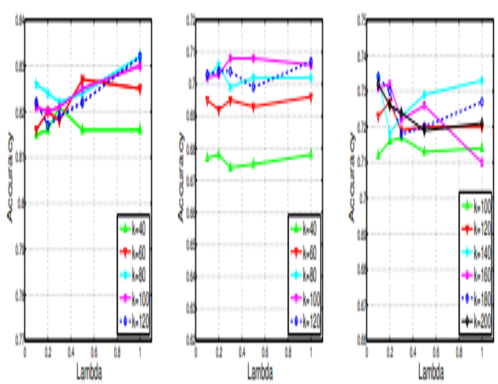
(3) TADW has an encouraging performance when training ratio is small. The accuracies of most baselines drop quickly as training ratio decreases because their vertex representations are much noisy and inconsistent for training and testing. Instead, since TADW learns representation jointly from both network and text information, the representations have less noise and are more consistent.

These observations demonstrate the high quality of representations generated by TADW. Moreover, TADW is not task specific and the representations can be conveniently used for different tasks, such as link prediction, similarity computation and vertex classification. The classification accuracy of TADW is also competitive with several recent collective classification algorithms [Shi et al., 2011; McDowell and Aha, 2012; 2013] though we don't perform specific optimization for the tasks when we learn representations.

**A. Parameter Sensitivity TADW has two hyper parameters**: dimension k and weight of regularization term λ. We fix training ratio to 10% and test classification accuracies

Table 3: Evaluation results on Wiki dataset.

| Classifier | SVM | | | | | | |
|---|---|---|---|---|---|---|---|
| % Labeled Nodes | 3% | 7% | 10% | 20% | 30% | 40% | 50% |
| DeepWalk | 48.4 | 56.6 | 59.3 | 64.3 | 66.2 | 68.1 | 68.8 |
| PLSA | 58.3 | 66.5 | 69.0 | 72.5 | 74.7 | 75.5 | 76.0 |
| Text Features | 46.7 | 60.8 | 65.1 | 72.9 | 75.6 | 77.1 | 77.4 |
| Naive Combination | 48.7 | 62.6 | 66.3 | 73.0 | 75.2 | 77.1 | 78.6 |
| NetPLSA | 56.3 | 64.6 | 67.2 | 70.6 | 71.7 | 71.9 | 72.3 |
| TADW (k=100) | **59.8** | **68.2** | **71.6** | **75.4** | **77.3** | **77.7** | **79.2** |
| TADW (k=200) | 60.4 | 69.9 | 72.6 | 77.3 | 79.2 | 79.9 | 80.3 |



(a) Cora Dataset (b) Citeseer Dataset (c) Wiki Dataset

Figure 4: Parameter sensitivity of k and λ

We let k vary from 40 to 120 and λ vary from 0.1 to 1 for Cora and Citeseer datasets; k vary from 100 to 200 and λ vary from 0.1 to 1 for Wiki dataset. Figure 2 shows the variation of classification accuracies with different k and λ. The accuracies range within 1.5%, 1% and 2% for fixed k on Cora, Citeseer and Wiki, respectively. The accuracies are competitive when k ≥ 80 on Cora and Citeseer and k ≥ 140 on Wiki. Therefore TADW can keep stable when k and λ vary within a reasonable range.

**B. Case Study**

To better understand the effectiveness of text information for NRL, we present an example in Cora dataset. The document title is "Irrelevant Features and the Subset Selection Problem". We call this paper IFSSP for short. The class label of IFSSP is "Theory". As shown in Table 4, using representations generated by Deep Walk and TADW, we find 5 most similar documents of IFSSP ranked by cosine similarity. We find that, all these documents are cited by IFSSP. However, 3 of the 5 documents found by Deep Walk have different class labels while the first 4 documents found by TADW have the same label "Theory". This indicates that, as compared to pure network-based Deep Walk, TADW can learn better network representations with the help of text information. The 5th document found by Deep Walk also shows another limitation of considering only network information. "MLC Tutorial A Machine Learning library of C classes" (MLC for short) is a document describing a general toolbox, which may be cited by many works in different topics. Once some of these works cite IFSSP as well, Deep Walk will tend to give IFSSP a similar representation with MLC even though they are totally on different topics.

Table 4: Five nearest documents by Deep Walk and TADW

| Top 5 nearest documents by DeepWalk | |
|---|---|
| Title | Class Label |
| Feature selection methods for classifications | Neural Network |
| Automated model selection | Rule Learning |
| Compression-Based Feature Subset Selection | Theory |
| Induction of Condensed Determinations | Case Based |
| MLC Tutorial A Machine Learning library of C classes | Theory |
| **Top 5 nearest documents by TADW** | |
| Title | Class Label |
| Feature subset selection as search with probabilistic estimates | Theory |
| Compression-Based Feature Subset Selection | Theory |
| Selection of Relevant Features in Machine Learning | Theory |
| NP-Completeness of Searches for Smallest Possible Feature Sets | Theory |
| Feature subset selection using a genetic algorithm | Genetic Algorithms |

## V. CONCLUSIONS

Traditional social-network-based recommendation algorithms generally utilize the coarse-grained trust relationships to generate a recommendation, which seriously hinders the performance of recommendation algorithms. To tackle this problem, we proposed a network representation learning enhanced recommendation algorithm in this study. Specially, we first adopt a network representation learning tech unique to embed a social network into a low-dimensional space, and then utilize the low-dimensional representations of users to infer fine-grained dense trust relationships between them. Finally, we integrate the fine-grained dense trust relationships into the classic matrix factorization model to learn latent user and item feature vectors. Experimental results on real-world datasets show that our proposed approach out-performs traditional social-network-based recommendation algorithms. As mentioned above, our proposed recommendation algorithm is a two-stage approach, i.e. firstly adopting a network representation technique to embed a social network into a low-dimensional space, and then integrating the fine-grained dense trust relationships inferred from embedded representations of users into the matrix factorization model.

## VI. REFERENCES

*[1] [Bhuyan et al., 2014] Monowar H Bhuyan, DK Bhattacharyya, and Jugal K Kalita. Network anomaly detection: methods, systems and tools. IEEE Communications Surveys & Tutorials, 16(1):303–336, 2014.*

*[2] [Chen et al., 2007] Mo Chen, Qiong Yang, and Xiaoou Tang. Directed graph embedding. In Proceedings of IJCAI, pages 2707–2712, 2007.*

*[3] [Fan et al., 2008] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. JMLR, 9:1871–1874, 2008.*

*[4] [Fouss et al., 2007] Francois Fouss, Alain Pirotte, J-M Renders, and Marco Saerens. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. IEEE TKDE, 19(3):355–369, 2007.*

*[5] [Hofmann, 1999] Thomas Hofmann. Probabilistic latent semantic indexing. In Proceedings of SIGIR, pages 50–57, 1999.*

*[6] [Joachims, 1999] T. Joachims. Making large-scale SVM learning practical. In B. Scholkopf, C. Burges, and ¨ A. Smola, editors, Advances in Kernel Methods -*

*Support Vector Learning, chapter 11, pages 169–184. MIT Press, Cambridge, MA, 1999.*

*[7] [Krizhevsky et al., 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Proceedings of NIPS, pages 1097–1105, 2012.*

*[8] [Levy and Goldberg, 2014] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In Proceedings of NIPS, pages 2177–2185, 2014.*

*[9] [Lin et al., 2015] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In Proceedings of AAAI, 2015.*

*[10] [Lu and Zhou, 2011 ¨ ] Linyuan Lu and Tao Zhou. Link pre- ¨ diction in complex networks: A survey. Physica A: Statistical Mechanics and its Applications, 390(6):1150–1170, 2011.*

*[11] [McDowell and Aha, 2012] Luke McDowell and David Aha. Semi-supervised collective classification via hybrid label regularization. In Proceedings of ICML, 2012.*

*[12] [McDowell and Aha, 2013] Luke K McDowell and David W Aha. Labels or attributes?: rethinking the neighbors for collective classification in sparsely-labeled networks. In Proceedings of CIKM, pages 847–852, 2013.*

*[13] [Mei et al., 2008] Qiaozhu Mei, Deng Cai, Duo Zhang, and ChengXiang Zhai. Topic modeling with network regularization. In Proceedings of WWW, pages 101–110, 2008.*

*[14] [Mikolov et al., 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In Proceedings of NIPS, pages 3111–3119, 2013.*

*[15] [Morin and Bengio, 2005] Frederic Morin and Yoshua Bengio. Hierarchical probabilistic neural network language model. In Proceedings of AISTATS, volume 5, pages 246– 252, 2005.*

*[16] [Natarajan and Dhillon, 2014] Nagarajan Natarajan and Inderjit S Dhillon. Inductive matrix completion for predicting gene–disease associations. Bioinformatics, 30(12):i60–i68, 2014.*

*[17] [Perozzi et al., 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In Proceedings of SIGKDD, pages 701–710, 2014.*

*[18] [Sen et al., 2008] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. AI magazine, 29(3):93, 2008.*

*[19] [Shi et al., 2011] Xiaoxiao Shi, Yao Li, and Philip Yu. Collective prediction with*

*latent graphs. In Proceedings of CIKM, pages 1127–1136, 2011.*

*[20] [Tang and Liu, 2009] Lei Tang and Huan Liu. Relational learning via latent social dimensions. In Proceedings of SIGKDD, pages 817–826, 2009.*

*[21] [Tang and Liu, 2011] Lei Tang and Huan Liu. Leveraging social media networks for classification. Proceedings of SIGKDD, 23(3):447–478, 2011.*

## *Authors Profile*

*S Jafar Ali Ibrahim working as Professor of CSE Department in QIS College of Engineering and Technology (Autonomous), Ongole, Andhra Pradesh, India.*

*B Siva Reddy pursuing B Tech in computer science engineering from Qis college of Engineering and Technology (Autonomous & NAAC 'A' Grade), Ponduru Road, Vengamukkalapalem, Ongole, Prakasam Dist, Affiliated to Jawaharlal Nehru Technological University, Kakinada.*

*K Venkata Yaswanth pursuing B Tech in computer science engineering from Qis college of Engineering and Technology (Autonomous & NAAC 'A' Grade), Ponduru Road, Vengamukkalapalem, Ongole, Prakasam Dist, Affiliated to Jawaharlal Nehru Technological University, Kakinada.*

*K Muneswar pursuing B Tech in computer science engineering from Qis college of Engineering and Technology (Autonomous & NAAC 'A' Grade), Ponduru Road, Vengamukkalapalem, Ongole, Prakasam Dist, Affiliated to Jawaharlal Nehru Technological University, Kakinada.*

*S Jaswanth pursuing B Tech in computer science engineering from Qis college of Engineering and Technology (Autonomous & NAAC 'A' Grade), Ponduru Road, Vengamukkalapalem, Ongole, Prakasam Dist, Affiliated to Jawaharlal Nehru Technological University, Kakinada.*