# Semi-supervised learning for clusterable graph embeddings with NMF

**Priyesh Vijayan[1]*, Anasua Mitra[2]†, Srinivasan Parthasarathy[3] and Balaraman Ravindran[1]**
[1] Dept. of CSE, Indian Institute of Technology Guwahati, India
[2]Dept. of CSE and Robert Bosch Centre for Data Science and AI
Indian Institute of Technology Madras, India
[3] Dept. of CSE and Dept. of Biomedical Informatics, Ohio State University, USA

## Abstract

We propose a Semi-Supervised Learning (SSL) model for learning cluster invariant node representations that enforce high label smoothness within the clusters. This work focuses on learning semi-supervised node representations in non-attributed graphs. Specifically we compare and analyze different SSL models that use Non-Negative Matrix Factorization. The clustering assumption of SSL has not been explored much in the network representation learning literature. We show that explicitly encoding the clustering requirement provides improved performance on node classification task across a variety of datasets. Further, we demonstrate the superior clusterability of the learned node representations quantitatively with clustering task and qualitatively with t-SNE visualizations.

## 1 Introduction

Chapelle et al. (2009) mentions that efficient SSL requires that the data lie in (1) a low-dimensional manifold, (2) exhibit high label smoothness characterized by homogeneous high-density clusters of the same class (3) which are well separated from the clusters of different classes. Here, we focus on graph based SSL for node classification in non-attributed graphs to learn clusterable node representations. We propose a novel Semi-Supervised Non-negative Matrix Factorization (SS-NMF) model that learns cluster invariant node representations that enhance high label smoothness within these learned clusters. Clusterability though a well known prior, was either largely ignored or not explicitly handled in SSL. To the best of our knowledge, we are the first to explicitly learn semi-supervised cluster invariant representations in the network representation learning setup.

## 2 Proposed Work

### 2.1 Notations

Let, $G = (V, A, Y)$ be a networked data, where $V$ is the set of vertices and $A \in \mathbb{R}^{N \times N}$ is the matrix of (un)weighted, (un)directed edges. $C$ denotes the set of $q$ possible classes. $Y \in \mathbb{R}^{q \times N}$ is the one-hot representation of classes for all nodes. We have $L$ labeled data $\{(i, Y_i)\}_{i=1}^{L}$ and $UL$ unlabeled data $\{(i)\}_{i=L+1}^{L+UL}$ with total number of nodes $N = L + UL$. $m$ is the dimension of representation space. Let, $D_{N \times N}$ be the diagonal degree matrix of the adjacency matrix $A$ defined as $d_{\mathrm{ii}} = \sum_j a_{\mathrm{i,j}}$. Thus, the unnormalized Laplacian operator on the graph $G$ can be defined as $L_{N \times N} = D - A$.

*equally contributing authors
†equally contributing authors

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

### 2.2 SS-NMF Framework

In this section, we will build the proposed semi-supervised model step by step.

**Learning network structure by encoding local context:** The rudimentary component of the model is that which learns locally invariant node representations, i.e., nodes which are connected have similar representations. We obtain locally invariant representations by factorizing a proximity matrix that encodes the similarity between the nodes. Herein, we consider the Pointwise Mutual Information matrix, $S$, defined in Tu et al. (2016) for Matrix Factorized DeepWalk (MFDW) model. It is the average transition probability in a window size of $(t)$. Herein, different from Tu et al. (2016), we resort to Non-negative Matrix Factorization framework as the proximity matrix is all positive. We factorize the proximity $S$ into two non-negative basis matrices - the node representation matrix $U \in \mathbb{R}^{m \times N}$ and the context/ neighbourhood representation matrix $M \in \mathbb{R}^{m \times N}$ as given below.

$$L_1 = \min_{M,U} \|S - U^T M\|^2 : M \geq 0, U \geq 0 \tag{1}$$

**Encoding supervision knowledge:** In order to learn semi-supervised representations, we need to jointly factorize the label matrix, $Y$ along with $S$. We define the label matrix factorization term in Eqn: 2. Where $W \in \mathbb{R}^{q \times N}$ is the weight penalty matrix that zeros out all the label information of test instances. Specifically, $W_i$ is equal to 0 if the corresponding $Y_i$ is unknown and 1 otherwise. $\odot$ is the hadamard or element-wise multiplication. $Q \in \mathbb{R}^{q \times m}$ is the label basis matrix. The supervision component is defined as follows:

$$L_2 = \min_{Q,U} \|W \odot (Y - QU)\|^2 : Q \geq 0, U \geq 0 \tag{2}$$

**Encoding cluster level label smoothness:** Here, we define a novel component that allows for learning cluster invariant representation. In essence, it allows the model to learn clusterable representations such that there is high label smoothness within the cluster. This component primarily comprises of two components:

- **Learn cluster assignment:** Let, $H \in \mathbb{R}^{k \times N}$ represents the cluster membership indicator matrix defined for $k$ number of clusters. We obtain H by projecting node embeddings, $U$ on cluster basis, $H = CU$. We can control the soft assignment to clusters with $Trace(HH^T) = N$ term. By setting $HH^T = I$, we relax the block diagonal assignments to orthogonal assignments similar to Wang et al. (2017).
- **Encode Cluster Invariance Property (C.I.P):** We enforce this constraint by applying Laplacian regularization on $H$ with label similarity based proximity matrix, $E$. We define the label similarity network defined over train data as $E = (W \odot Y)^T (W \odot Y) \in \mathbb{R}^{N \times N}$, where $\mathcal{L}(\mathcal{E}) = D - E$ is the unnormalized Laplacian operator on $E$.

The label based similarity matrix introduces new edges between nodes of similar labels which may be far away or not even connected in the original network, $S$. In this way, clusters can capture global information. Unlike models which enforce explicit Laplacian regularization on the embedding space or label space, we enforce this constraint on an abstract space, clusters.

Cluster invariant representations that enforce similar cluster assignments to nodes of same/ similar labels are obtained in Eqn: 3. There is a circular enforcement between $H$ and $U$, i.e., $U$ learns from $H$, $H$ implicitly learns from $U$ and $H$ explicitly learns from the cluster overlap regularization term. Therefore, $H$ pushes two nodes with same labels and similar neighbourhood together into the same cluster and ensures that these two nodes have similar representations.

$$L_{group} = \min_{H,C,U} \beta \|H - CU\|^2 + \phi Tr\{H\mathcal{L}(E)H^T\} + \zeta \|HH^T - I\|^2 : H \geq 0 \tag{3}$$

### 2.3 Optimization

In SS-NMF, the node representations, $U$ are learned by jointly factorizing the local neighbourhood proximity matrix $S$, label matrix $Y$ and inferred cluster assignment matrix $H$. We learn SS-NMF by combining the objective terms corresponding to each component along with L2 regularization on the learned factor matrices as given below:

$$L = \alpha L_{network} + \theta L_{label} + L_{group} + \lambda(L2_{reg}) : M, U, Q, C, H \geq 0 \tag{4}$$

$\alpha, \beta, \theta, \zeta, \phi, \lambda$ are hyperparameters controlling the importance of respective terms in equation. Multiplicative update rules can be derived for $M, U, C, Q, H$ to minimize the above objective function as in Lee and Seung (2001).

# 3 Experiments

## 3.1 Baselines & Related Work

The state of the art methods for semi-supervised classification on non-attributed graphs are limited only to MaxMargin-DeepWalk (**MMDW**) Tu et al. (2016) and (**Planetoid**) Yang et al. (2016). Besides, we also introduce two more semi-supervised baselines built on top of the unsupervised community preserving embedding model (**MNMF**) Wang et al. (2017) and the Matrix Factorization version of DeepWalk (**MFDW**) Tu et al. (2016) viz: **MNMFL** and **MFDWL** respectively, where the original objectives are jointly factorized with label matrix as in Eqn: 2. For reference's sake, we have also included random walk sampling based DeepWalk (**DW**) Perozzi et al. (2014).

Planetoid was defined for multi-class classification problem on stratified labelled set (which can be unrealistic). We empirically observed Planetoid to perform poorly in comparison to other baselines when the labelled set is randomly drawn (not reported here) and is not directly extensible for multi-label problems. Hence, we do not report results for Planetoid here, rather we define a similar semi-supervised NMF model, MF-Planetoid (**MF-Plan**) (refer to Appendix for details) with Planetoid's semi-supervised learning objective i.e., explicitly enforcing embeddings of nodes of the same label to be similar. To be fair here, we merely report the comparison of our model, SSNMF against the original Planetoid model on their train/ test split below in terms of Micro-F1 scores in percentage: a) On Cora dataset; Planetoid: 69.1 and SSNMF: **78.8** b) On Citeseer dataset; Planetoid: 49.3 and SSNMF: **50.6** c) On Pubmed dataset; Planetoid: 66.4 and SSNMF: **79.6**. We found balanced class distribution to be beneficial for our model as we obtain extraordinary performance improvement $10\%$ on Cora and $13\%$ on Pubmed. Albeit, we primarily report results in Table: 1, 2 on a realistic setup with randomly sampled train-test data. Dataset details are provided in the Appendix.

## 3.2 Node classification

We report classification performance with Micro-F1 scores averaged over $5$ runs with randomly sampled train and test sets. For all models, we learn an external Logistic Regression classifier (LR) that makes label predictions from the model's learned node representations. Though we can obtain label predictions internally for the supervised models by reconstructing the label matrix, we found that using an external classifier further improves the performance in all models. However, we noticed that models with explicit SS label based constraints, SS-NMF and MF-Plan, were less sensitive to the external LR unlike other models. We define two aggregate metrics to measure the overall performance of models across datasets viz: *Rank* and *Penalty*. *Rank* of a model is defined as the average position of the model when the results are ordered in descending order in each dataset and *Penalty* of the model is defined as the average difference from the best performing model in each dataset. The lower the rank and penalty, the better is the performance of the model.

Table 1: Node Classification Results | Micro-F1 Scores

| | Non-negative Matrix Factorization models | | | | | | | Sampling |
|---|---|---|---|---|---|---|---|---|
| | Proposed | SOTA | Proposed Baseline Variants | | | SOTA | SOTA | SOTA |
| Datasets | SS-NMF | MMDW | MFDWL | MF-Plan | MNMFL | MNMF | MFDW | DW |
| Cora | **85.84** | 83.92 | 83.69 | 84.38 | 84.55 | 82.66 | 80.37 | 80.15 |
| Citeseer | **69.75** | 67.25 | 68.62 | 69.52 | 69.00 | 63.57 | 59.71 | 57.27 |
| Wiki | **67.02** | 66.69 | 66.18 | 66.42 | 66.75 | 65.75 | 63.94 | 63.01 |
| Washington | **66.09** | 61.13 | 62.61 | 62.83 | 62.96 | 62.61 | 59.13 | 59.13 |
| Wisconsin | **54.14** | 50.67 | 50.38 | 52.13 | 52.76 | 51.13 | 49.02 | 48.12 |
| Texas | **61.70** | 56.38 | 58.51 | 59.36 | 57.45 | 57.45 | 56.38 | 58.51 |
| Cornell | **52.04** | 51.22 | 50.94 | **52.04** | **52.04** | 51.45 | 50.00 | 38.78 |
| PPI | 23.09 | **23.58** | 22.19 | 22.16 | 21.45 | 21.23 | 21.75 | 22.22 |
| Blogcatalog | 36.35 | 34.72 | 34.36 | 34.53 | 34.88 | 34.42 | 32.05 | **40.59** |
| Rank | **1.33** | 3.67 | 5.00 | 3.33 | 3.44 | 5.78 | 7.11 | 6.33 |
| Penalty | **0.7267** | 2.3144 | 2.6700 | 2.0156 | 2.1856 | 3.4711 | 5.4622 | 5.9700 |

In Table: 1, the bold-ed entries in a dataset column denote the best score achieved in that dataset and the underlined entries denote the second best score. Matrix factorization based DW, MFDW performs better than sampling-based DW as shown in Tu et al. (2016) on all but PPI and Blogcatalog. In these datasets, we found the Cross-Entropy (CE) loss used in DW to be an attributing factor for improved performance. We believe a CE based Matrix reconstruction loss could improve the performance on all the models in the MF framework. *All supervised models obtain better ranking and lower penalty over unsupervised models*. The SS variants of the unsupervised models are better than their unsupervised counterparts on all datasets, i.e., MNMFL > MNMF and MFDWL > MFDW. *SS-NMF outperforms its base model*, MFDWL on all datasets which is a clear indicator that learning cluster invariant representations are useful. *SS-NMF is ranked first in* $7/9$ *datasets*, while being ranked second on the remaining two. Thus, obtaining an average rank score of $1.33$ and the lowest penalty of $0.7267$. On the datasets where SS-NMF is ranked first, as per the paired t-test, there exists no case where $p - value < 0.05$ and t-scores are positive (i.e., no competing method significantly beats SS-NMF). The two datasets where SS-NMF failed can be attributed to its competitor's superiority. Ignoring SS-NMF, on PPI all the other models perform similarly and MMDW easily beats the best among them by more than $1.3\%$. This shows the effectiveness of using Max-margin based label prediction loss. It is expected (based on results) that modifying MMDW to have our proposed within-cluster label smoothness can achieve SOTA results across all datasets.

### 3.3 Clusterability of learned representations

We validate superior clusterability of the learned node representations quantitatively in Table: 2 and qualitatively with t-SNE plots provided in Appendix. In Table: 2 we report the averaged cluster quality of learned embeddings of models trained with $50\%$ labeled data over $5$ folds, and $5$ runs with different initialization techniques (random, k-means++, PCA based). The clusters were obtained with k-means and Fuzzy c-means algorithms for multi-class and multi-label datasets correspondingly. The optimal number of clusters was obtained using gap statistics Tibshirani et al. (2001). We evaluate the obtained clusters against gold standard classes as gournd truth clusters and report the NMI scores. We used Overlapping NMI Lancichinetti et al. (2009), McDaid et al. (2011) for overlapping clusters to evaluate the multi-label datasets.

From Table: 2, it is evident that SS-NMF performs well in semi-supervised node clustering. It is the best performing model on 7 datasets where it beats the second best model by 1-7%, and it is the second best performing model on the other two datasets where it is falling short of the best by a mere $0.1\%$. All the semi-supervised NMF models outperform the unsupervised NMF models except for MMDW which is outperformed by MNMF. Both MFDWL and MNMFL outperform their unsupervised counterparts, MFDW and MNMF. The supervised MMDW outperforms the simple unsupervised MFDW in all but Cornell. However, it is thoroughly washed out in comparison against the unsupervised MNMF. Though MMDW's max-margin representations outperformed unsupervised MNMF in many of the datasets for node classification task, it seems that they are not well clusterable. Consistent superior performance of SS-NMF & MF-Plan suggests that the label similarity based clusterability criteria can learn informative node representations beyond the graph structure. This is supported by the t-SNE plots too, especially that of SS-NMF which provides superior high-quality visualizations of well separable homophilous clusters.

Table 2: Node Clustering | (O)NMI Scores

| | Non-negative Matrix Factorization models | | | | | | | Sampling |
|---|---|---|---|---|---|---|---|---|
| | Proposed | SOTA | Proposed Baseline Variants | | | SOTA | SOTA | SOTA |
| Dataset | **SS-NMF** | MMDW | MFDWL | MF-Plan | MNMFL | MNMF | MFDW | DW |
| Cora | **54.40** | 36.44 | 51.38 | 51.80 | 53.21 | 39.29 | 34.40 | 34.28 |
| Citeseer | **50.94** | 22.61 | 28.94 | 48.19 | 41.19 | 29.96 | 17.71 | 19.04 |
| Wiki | **52.60** | 35.68 | 47.80 | 47.80 | 48.38 | 45.62 | 28.31 | 32.57 |
| Washington | **40.27** | 13.65 | 18.45 | 31.41 | 33.52 | 19.90 | 09.93 | 02.88 |
| Wisconsin | **31.52** | 07.79 | 06.81 | 28.38 | 17.89 | 11.20 | 06.09 | 05.04 |
| Texas | **36.30** | 07.63 | 10.61 | 28.72 | 15.14 | 09.00 | 02.85 | 02.70 |
| Cornell | 04.77 | 03.70 | 04.49 | **04.89** | 04.14 | 03.99 | 04.16 | 03.53 |
| PPI | **09.76** | 08.44 | 08.26 | 09.36 | 09.19 | 08.77 | 07.91 | 09.44 |
| Blogcatalog | **14.31** | 06.07 | 06.18 | 07.36 | 08.61 | 06.93 | 03.06 | 03.71 |
| Rank | **1.11** | 6.00 | 4.77 | 4.67 | 2.89 | 4.67 | 7.11 | 7.00 |
| Penalty | **0.0133** | 16.9977 | 12.4522 | 4.1200 | 7.0800 | 13.3700 | 20.0633 | 20.2000 |

## References


Chapelle, O.; Scholkopf, B.; Zien, A. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks* **2009**, *20*, 542–542.

Tu, C.; Zhang, W.; Liu, Z.; Sun, M. Max-Margin DeepWalk: Discriminative Learning of Network Representation. IJCAI. 2016; pp 3889–3895.

Wang, X.; Cui, P.; Wang, J.; Pei, J.; Zhu, W.; Yang, S. Community Preserving Network Embedding. AAAI. 2017; pp 203–209.

Lee, D. D.; Seung, H. S. Algorithms for non-negative matrix factorization. Advances in neural information processing systems. 2001; pp 556–562.

Yang, Z.; Cohen, W. W.; Salakhutdinov, R. Revisiting semi-supervised learning with graph embeddings. *arXiv preprint arXiv:1603.08861* **2016**,

Perozzi, B.; Al-Rfou, R.; Skiena, S. Deepwalk: Online learning of social representations. Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. 2014; pp 701–710.

Tibshirani, R.; Walther, G.; Hastie, T. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **2001**, *63*, 411–423.

Lancichinetti, A.; Fortunato, S.; Kertész, J. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics* **2009**, *11*, 033015.

McDaid, A. F.; Greene, D.; Hurley, N. Normalized mutual information to evaluate overlapping community finding algorithms. *arXiv preprint arXiv:1110.2515* **2011**,

Chakrabarti, S.; Dom, B.; Indyk, P. Enhanced hypertext categorization using hyperlinks. ACM SIGMOD Record. 1998; pp 307–318.

Craven, M.; McCallum, A.; PiPasquo, D.; Mitchell, T.; Freitag, D. *Learning to extract symbolic knowledge from the World Wide Web*; 1998.

McCallum, A. K.; Nigam, K.; Rennie, J.; Seymore, K. Automating the construction of internet portals with machine learning. *Information Retrieval* **2000**, *3*, 127–163.

Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; Eliassi-Rad, T. Collective classification in network data. *AI magazine* **2008**, *29*, 93.

Stark, C.; Breitkreutz, B.-J.; Reguly, T.; Boucher, L.; Breitkreutz, A.; Tyers, M. BioGRID: a general repository for interaction datasets. *Nucleic acids research* **2006**, *34*, D535–D539.

Zafarani, R.; Liu, H. Social Computing Data Repository at ASU. 2009; http://socialcomputing.asu.edu.

Tang, L.; Liu, H. Scalable learning of collective behavior based on sparse social dimensions. Proceedings of the 18th ACM conference on Information and knowledge management. 2009; pp 1107–1116.

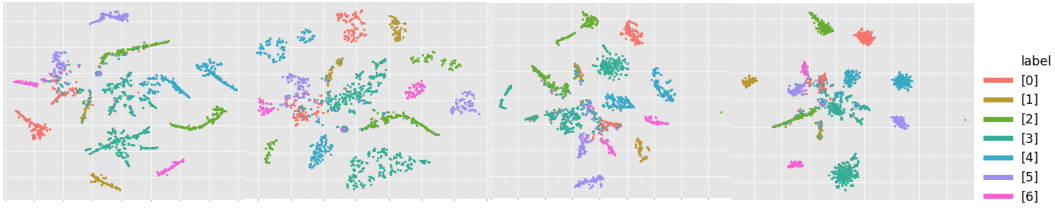
Tang, L.; Liu, H. Relational learning via latent social dimensions. Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. 2009; pp 817–826.

# 4 Appendix

## 4.1 t-SNE Visualization

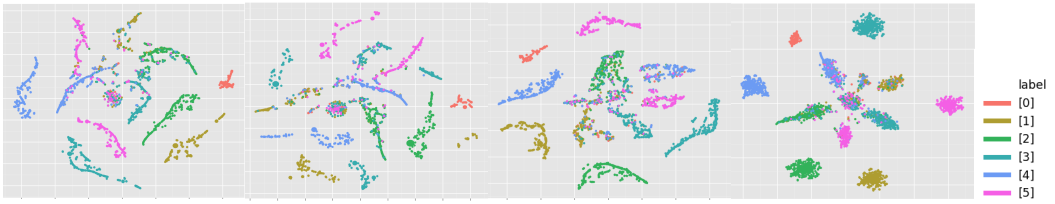
Further, we also present the details of t-SNE experiment on the learned node embeddings for Citeseer and Cora dataset in Figure: 2 & 1. t-SNE plots are specially well-suited for the visualization of high dimensional data. As t-SNE algorithm scales quadratically in terms of the number of nodes N, we first reduced the dimension of learned node embeddings to 64 retaining as much information as possible using Principal Component Analysis (PCA) algorithm. Next, we fed this reduced data to t-SNE algorithm. In t-SNE, the perplexity term controls the number of neighbours for each sample to take into consideration while preserving the local structure in the reduced dimension space. We experimented with perplexity in the range of $10 - 100$ increasing by a step size of 10 and found that it did not affect the visualizations much from 30 onwards. So we fixed 40 as a common value of perplexity for all the competing methods. It can be seen that our proposed model obtains better clusters visually compared to other SS methods due to label similarity based cluster-enhanced node embeddings learned.

Figure 1: t-SNE Visualization of embeddings on Cora dataset




(a) MFDWL (b) MF-Plan (c) MNFL (d) SS-NMF

Figure 2: t-SNE Visualization of embeddings for Citeseer dataset




(a) MFDWL (b) MF-Plan (c) MNFL (d) SS-NMF

## 4.2 Datasets details

Description of the datasets used are provided below with summary statistics tabulated in Table: 3.

*WWW networks:* WebKB Chakrabarti et al. (1998) consists of four small networks collected from four different universities - Washington, Wisconsin, Texas and Cornell. The networks are a collection of web pages as nodes where the task is to predict the type of webpage.

*Citation networks:* In the following citation networks, nodes are the research papers and edges exist if one paper cites another. Cora Craven et al. (1998), Citeseer McCallum et al. (2000), Wiki Sen et al. (2008) and Pubmed are four bibliographic datasets where the task is to predict the research area of the paper.

*Biological network:* PPI Stark et al. (2006) is a protein-protein interaction biological dataset for Homo Sapiens where the task is to predict the functional properties for proteins from the hallmark gene sets.

*Social network:* BlogCatalog Zafarani and Liu (2009) Tang and Liu (2009) Tang and Liu (2009) is a social network dataset with entities as bloggers and edges depicting friendship between them.

Classes are the topic of interests of the bloggers inferred through their blogs. PPI and BlogCatalog are multi-label classification datasets while the rest are all multi-class classification datasets.

Table 3: Datasets used in this experiment
$V$: nodes, $E$: edges, $Y$: labels, $ML$: multi-label dataset

| Dataset | $\|V\|$ | $\|E\|$ | $\|Y\|$ | ML | Avg Degree |
|---|---|---|---|---|---|
| Washington | 230 | 596 | 5 | F | 4.88 |
| Wisconsin | 265 | 724 | 5 | F | 5.0 |
| Texas | 186 | 464 | 5 | F | 4.51 |
| Cornell | 195 | 478 | 5 | F | 4.12 |
| Cora | 2,708 | 5,278 | 7 | F | 2.00 |
| Citeseer | 3,312 | 4,732 | 6 | F | 1.42 |
| Wiki | 2,405 | 17,981 | 19 | F | 6.87 |
| PPI | 3,890 | 76,584 | 50 | T | 19.69 |
| Blogcatalog | 10,312 | 3,33,983 | 39 | T | 64.78 |
| Pubmed | 19,717 | 44,338 | 3 | T | 4.50 |

### 4.3 Derivation of multiplicative update rules

Here, we give the detailed derivation for Eqn: 4 in order to get the multiplicative update equations for learned factor matrices $M, U, Q, C$ & $H$ respectively.

$$\begin{aligned}O &= \alpha Tr[(S - U^T M)(S - U^T M)^T] + \beta Tr[(H - CU)(H - CU)^T] \\ &+ \theta Tr[W \odot \{(Y - QU)(Y - QU)^T\}] + \zeta Tr[(HH^T - I)(HH^T - I)^T] \\ &+ \phi Tr\{H\mathcal{L}(E)H^T\} + \lambda Tr(MM^T + QQ^T + CC^T + UU^T + HH^T)\end{aligned}$$

$$\begin{aligned}\mathcal{L} &= \alpha Tr[SS^T - 2SM^T U + U^T MM^T U] \\ &+ \beta Tr[HH^T - 2HU^T C^T + CUU^T C^T] \\ &+ \theta Tr[W \odot \{YY^T - 2YU^T Q^T + QUU^T Q^T\}] \\ &+ \zeta Tr[HH^T HH^T - 2HH^T + I] + \phi Tr\{H\mathcal{L}(E)H^T\} \\ &+ \lambda Tr(MM^T + QQ^T + CC^T + UU^T + HH^T) \\ &+ Tr[\psi_1 M^T + \psi_2 U^T + \psi_3 C^T + \psi_4 Q^T + \psi_5 H^T]\end{aligned}$$

Let $\psi_1, \psi_2, \psi_3, \psi_4, \psi_5$ be the Lagrange multipliers for the non-negative constraints on factor matrices $M, U, Q, C, H$ respectively. We then have the Lagrange function $\mathcal{L}$ and obtaining partial derivatives of $\mathcal{L}$ with respect to the respective factor matrices,

$$\frac{\partial \mathcal{L}}{\partial M} = -2\alpha US + 2\alpha UU^T M + 2\lambda M + \psi_1$$

$$\frac{\partial \mathcal{L}}{\partial C} = -2\beta HU^T + 2\beta CUU^T + 2\lambda C + \psi_3$$

$$\frac{\partial \mathcal{L}}{\partial Q} = -2\theta(W \odot Y)U^T + 2\theta(W \odot QU)U^T + 2\lambda Q + \psi_4$$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial U} &= -2\alpha MS^T - 2\theta Q^T(W \odot Y) - 2\beta C^T H + 2\alpha MM^T U + 2\beta C^T CU \\ &+ \theta Q^T(W \odot QU) + 2\lambda U + \psi_2\end{aligned}$$

$$\frac{\partial \mathcal{L}}{\partial H} = 2\beta H - 2\beta CU + 4\zeta HH^T H - 4\zeta H + 2\lambda H + 2\phi HD - 2\phi HE + \psi_5$$

Using the KKT conditions, $\psi_{1ab} m_{ab} = 0$, $\psi_{2ab} u_{ab} = 0$, $\psi_{3ca} c_{ca} = 0$, $\psi_{4da} q_{da} = 0$ and $\psi_{5cb} h_{cb} = 0$, where $M = [m_{ab}], U = [u_{ab}], C = [c_{ca}], Q = [q_{da}], H = [h_{cb}]$ s.t. a, b, c, d are the respective row & column indices. Solving Eqn: 4, We get the following update equations,

$$M = M \odot \left(\frac{\alpha US}{\alpha UU^T M + \lambda M}\right) \tag{5}$$

$$C = C \odot (\frac{\beta H U^T}{\beta C U U^T + \lambda C}) \tag{6}$$

$$Q = Q \odot (\frac{\theta(W \odot Y)U^T}{\theta(W \odot QU)U^T + \lambda Q}) \tag{7}$$

$$U = U \odot (\frac{\alpha M S^T + \beta C^T H + \theta Q^T (W \odot Y)}{\alpha M M^T U + \beta C^T C U + \theta Q^T (W \odot QU) + \lambda U}) \tag{8}$$

$$H = H \odot (\frac{\beta C U + 2\zeta H + \phi H E}{\beta H + 2\zeta H H^T H + \phi H D + \lambda H})^{1/4} \tag{9}$$

### 4.4 More on Baselines and Variants with the experiment setup

In this section, we briefly give the details of experiment setup we followed for the baselines already described in section 3.1 along with the equation of baseline variants we have used in our experiment. The semi-supervised experiment is set up with $50\%$ labeled data are drawn randomly $5$ times. All classification and clustering results reported here are an average over these five sets. The results reported are an average of these five sets. For all the competing algorithms we set the dimension of the node embeddings as $128$. We also extensively searched for optimal hyperparameter values for all the competing methods using $20\%$ of the training data as a validation set.

For original sampling based DeepWalk we set the window size to $5$, the number of walks per source node and the walk length in the range of $[10, 40, 80]$ and report the best results. We also have MFDW Eqn: 1 - the matrix factorized DeepWalk, as we build our model incrementally on top of it. For all the matrix factorization based baselines, we vary the hyper-parameter values (the respective weightage terms for each component in the objective function) in the range of $[0.1, 0.5, 1.0, 5.0, 10.0]$.

**Max-Margin DeepWalk (MMDW):** In this paper Tu et al. (2016), max-margin loss is incorporated in the objective function of MFDW to learn discriminative representations of vertices in networks. It has one important hyper-parameter alpha-bias $(\eta)$ that balances the importance of primal gradient and biased gradient to induce max-margin loss based bias into random walk. We varied bias in the range of $\eta = [10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}]$ and weightage of proximity matrix factorization term as $\alpha = [0.1, 0.5, 1.0, 5.0, 10.0]$.

**MFDWL:** We build a variant of MMDW which also incorporates supervised information into node embeddings by jointly optimizing Eqn: 1 & 2. It works competitively as compared to MMDW.

**Planetoid & MF-Planetoid:** Planetoid Yang et al. (2016) learns an embedding space for nodes by jointly enforcing label and neighborhood similarity. Planetoid uses random walks to enforce structural similarity. Embeddings of nodes which appear in the random walks of a node are made to be similar to the node while others are made to be dissimilar. It can be seen as a similar approach to ours that uses the notion of label similarity to keep node embeddings close in the manifold space but does not explicitly learn and incorporate any global structure (clusters/ communities) into the node embeddings. As we observed poor performance of Planetoid on random test-train splits, we derive a matrix-factorized version of Planetoid as an alternative baseline. It enforces label smoothness $E$, i.e, train-label similarity on embedding space $U$, unlike ours as in Eqn: 3 on cluster space.

$$L_{MF-Plan} = O(MFDWL) + Tr\{U\mathcal{L}(E)U^T\} \tag{10}$$

**MNMF & MNMFL:** MNMF Wang et al. (2017), as introduced earlier, one recent state-of-the-art matrix factorization approach that incorporates both mesoscopic and microscopic structure of network into node representations by discovering communities through modularity maximization. We build one semi-supervised variant of MNMF, viz. MNMFL by jointly optimizing its objective function along with Eqn: 2. Unlike the original MNMF paper that factorizes a combination of first order and cosine similarity based second order node proximity to learn node representations, here, for sake of fair comparison, we stick to a combination of first order and second order transition probability based proximity matrix as $S$, following MMDW Tu et al. (2016).

One recent state-of-the-art non-negative matrix factorization approach that incorporates both mesoscopic and microscopic structure of network into node representations by discovering communities

through modularity maximization and jointly optimizing for it. It preserves the pairwise interaction by factorizing the first order and cosine similarity based proximity in the second order neighbourhood. It preserves the community structures in graph data by modularity maximization based community detection.

$$O = \min_{M,U,C,H \geq 0} \|S - MU^T\|^2 + \|H - UC^T\|^2 - \beta Tr\{H^T \mathcal{L}(B) H\} + \zeta \|HH^T - I\|^2 \quad (11)$$