Structural Neighborhood Based Classification of Nodes in a Network

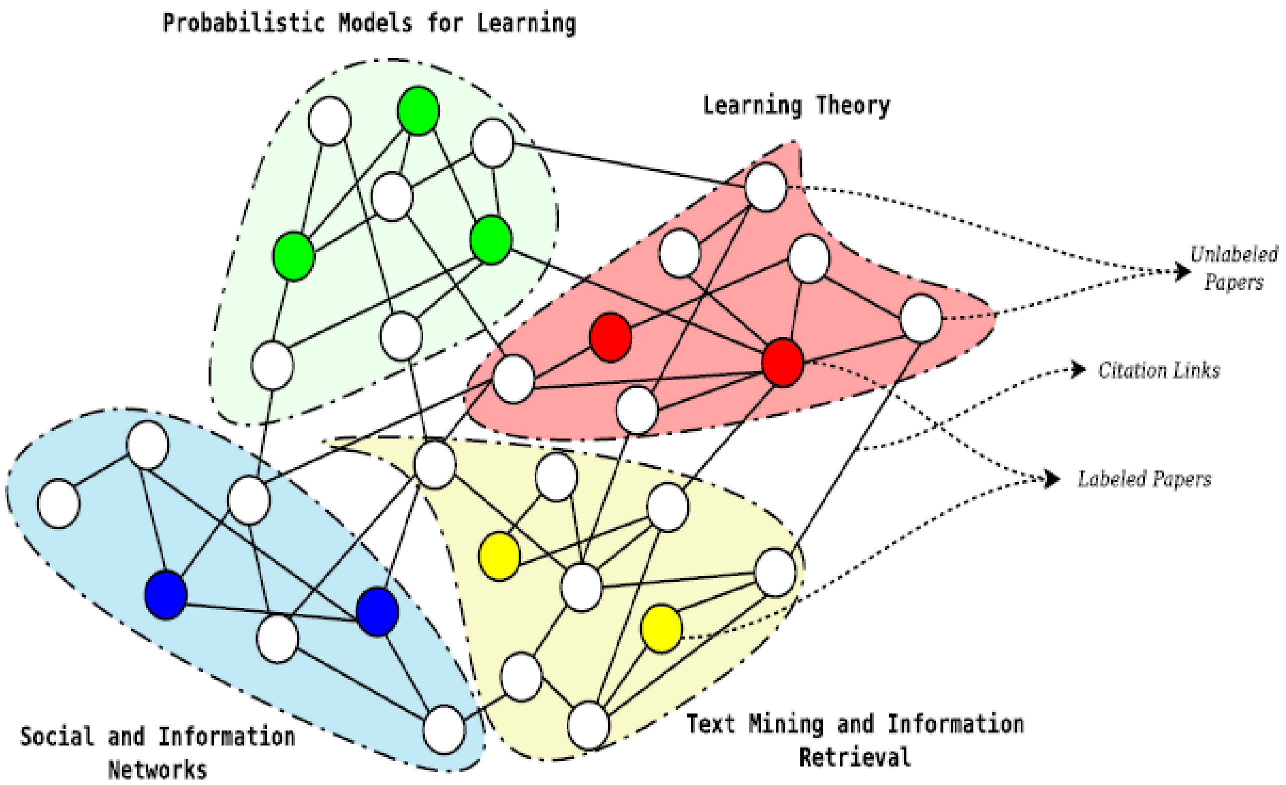
基于结构邻域的网络节点分类

KDD

Sharad Nandanwar

M. N. Murty

Purpose



- propose a novel structural neighborhood-based classifier learning using a random walk under sparse network

(1)

- homophily [1]: neighboring nodes in a network are supposed to be similar to each other.

- To exploit homophily in networks, we take a comprehensive view of classification, where a node is classified based on how other nodes in its extended neighborhood are labeled.

[1] A. Anagnostopoulos, R. Kumar, and M. Mahdian.Influence and correlation in social networks. In KDD,pages 7–15. ACM, 2008.

(2)

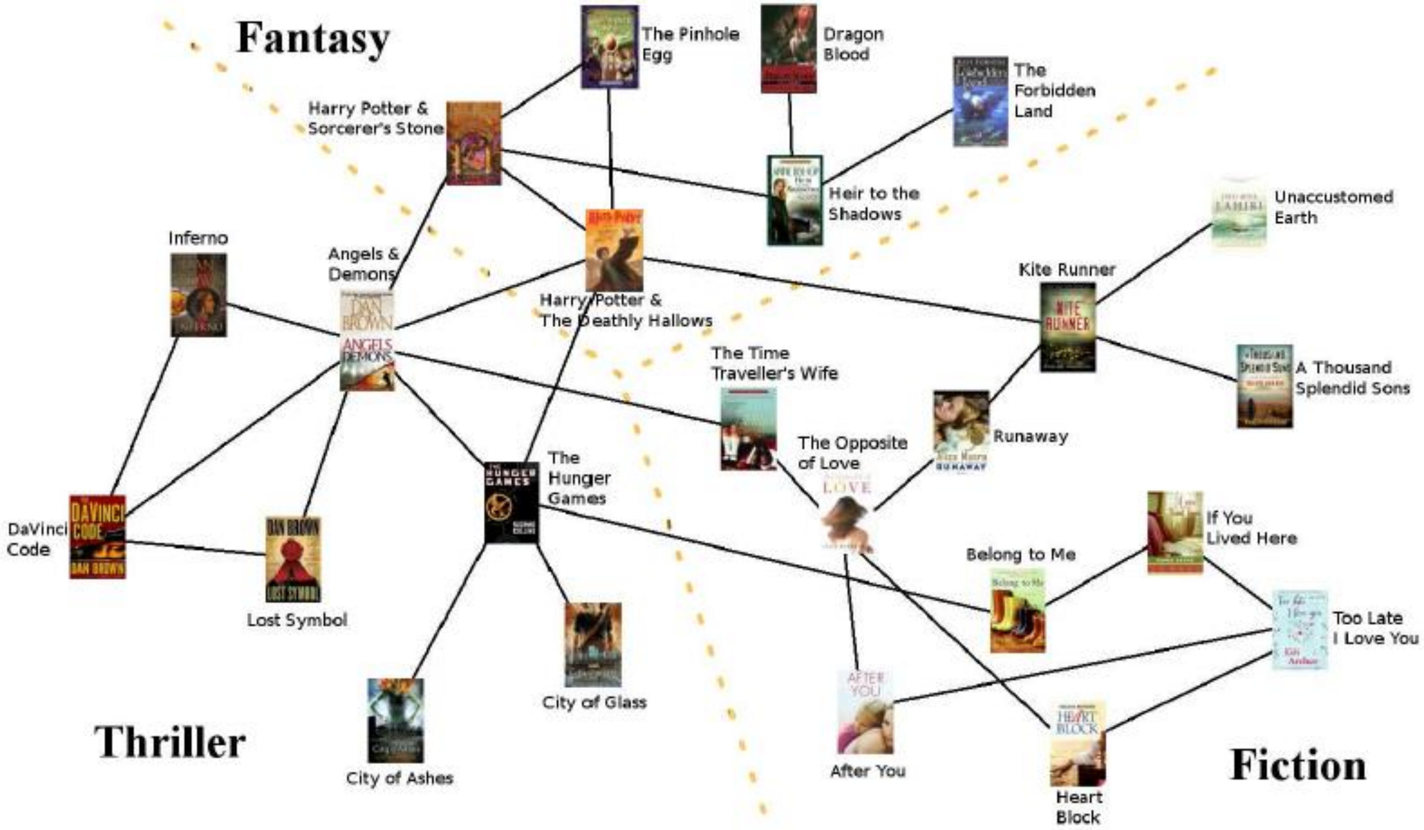
- A simple approach: use the number of neighbors from the respective classes and label the node based on majority voting.

- However,if the network is sparsely.It did not well.the label is sparse too.

What affect weight？

- Propose a method: weight vote instead of count

S. A. Macskassy and F. Provost. A simple relationalclassifier. In MRDM at KDD, 2003.



- It is observed that high degree nodes are generally the source of linkage noise in networks.

- They think should a link to a low or medium degree node should be considered more reliable

- So！！！Degree could affect the weight

Some Prepared

- In this paper, we will be working with a binary classification problem

- Let $C+$ and $C-$ represent the sets of positive and negative examples respectively in the binary classification problem under consideration.

DEFINITION 1. *A network is modeled as a graph* $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{W})$, *where* $\mathcal{V}$ *is the set of* $|\mathcal{V}| = n$ *interacting units (nodes or actors), and* $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ *is the set of edges indicating relationship or interactions among the nodes.* $\mathcal{W} \in \mathbb{R}^{n \times n}$, *where* $\mathcal{W}_{ij}$ *indicates affinity or strength of the relationship between nodes* $v_i, v_j \in \mathcal{V}$.

For a sparsely labeled network, the classification problem is formally stated as follows.

DEFINITION 2. *Given a graph* $\mathcal{G}(\mathcal{V}, \mathcal{E})$ *and a set of labeled nodes* $\mathcal{V}_l$ $(\subsetneq \mathcal{V})$ *with corresponding (ordered) set of labels* $\mathcal{Y}_l \in \mathcal{C}^{|\mathcal{V}_l|}$, *where* $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k\}$, *the set of* k *labels* $\mathcal{C}_1$ *to* $\mathcal{C}_k$. *The objective is to learn a model for inferring labels of the unlabeled nodes* $\mathcal{V}_u = \mathcal{V} \setminus \mathcal{V}_l$.

We start by defining the adjacency based representation of a graph. Given an undirected and binary-weighted graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, the corresponding adjacency matrix A is defined as follows,

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

In the case of a weighted graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{W})$, links carry a non-negative weight specified by $\mathcal{W}$. Adjacency Matrix A for these graphs is modified as $A_{ij} = \mathcal{W}_{ij}$. Based on the link attributes of a node, we define node v_i in vector space notation as $a_i = [A_{ji}]_{n \times 1}$ where $j \in \{1, 2, \cdots, n\}$.

Definition 3. ***First-Level Neighborhood*** *of a node* v_i *is the set* $\mathcal{N}_i^1$ *s.t.* $v_j \in \mathcal{N}_i^1$ *if and only if there exists an edge in the graph connecting* v_i *and* v_j*, i.e.,* $(v_i, v_j) \in \mathcal{E}$.

the follow problem

约束最优化问题

$$\min_{w,b} \quad \frac{\lambda}{2}||p||^2 \quad + \quad \frac{1}{|\mathcal{V}_l|}\sum_{i\in\mathcal{V}_l} \varepsilon_i \tag{2}$$

$$\begin{aligned} \text{such that} \quad & y_i(w^\intercal m_i + b) \geq 1 - \varepsilon_i \;, \\ & \varepsilon_i \geq 0 \;, \text{ and} \\ & m_i = A\frac{a_i}{d_i} \end{aligned}$$

(1) Structural Neighborhood: For a node v_i having class label $y_i \in \{-1, 1\}$, the aggregated scores from the nodes having label y_i in the first-level neighborhood ($\mathcal{N}_i^1$) should be more than the aggregated scores from rest of the nodes in the first-level neighborhood. Thus, for optimal w and b, we have, for all i,

$$r_i = y_i \left(w \bullet x_i + b \right)$$

$$y_i \sum_{j \in \mathcal{N}^1_{i,y_i}} A(v_i, v_j)(w^\mathsf{T} \cdot a_j + b) \;\geq\; -y_i \sum_{j \in \mathcal{N}^1_{i,-y_i}} A(v_i, v_j)(w^\mathsf{T} \cdot a_j + b),$$

where $A(v_i, v_j) = A_{ij}$ indicates the weight of edge joining nodes v_i and v_j. This can be equivalently rewritten as,

$$y_i \left(\sum_{j \in \mathcal{N}^1_{i,y_i}} A_{ij}(w^\mathsf{T} \cdot a_j + b) \;+\; \sum_{j \in \mathcal{N}^1_{i,-y_i}} A_{ij}(w^\mathsf{T} \cdot a_j + b) \right) \geq\; 0,$$

$$\implies y_i \left(\sum_{j \in \mathcal{V}} A_{ij}(w^\mathsf{T} \cdot a_j + b) \right) \;\geq\; 0.$$

Rearranging the terms, we get,

$$y_i \;\left(\; w^\mathsf{T} \cdot A \cdot a_i + d_i \; b \;\right) \;\geq\; 0,$$

where $d_i \;=\; \sum_{j \in \mathcal{V}} A_{ij}$.

$$\implies y_i \left(\frac{1}{d_i} w^\mathsf{T} \cdot A \cdot a_i + b \right) \;\geq\; 0$$

Let $M = [m_1, m_2, \cdots, m_n] = A^2 D^{-1}$, where $D \in \mathbb{R}n \times n$ defined as $D_{ij} = \begin{cases} d_i & \text{if } \; i = j \\ 0 & \text{otherwise} \end{cases}.$

Then, we have the following in a linearly separable case:

$$y_i(w^\mathsf{T} m_i + b) \geq 0.$$

The above can be interpreted as mapping the adjacency information a_i to a new space as $m_i = A\dfrac{a_i}{d_i}$, and then learning a decision boundary. For node v_i, we define empirical loss ε_i (≥ 0) such that

$$y_i(w^\mathsf{T} m_i + b) \geq 1 - \varepsilon_i.$$

Mean empirical loss, that is to be minimized, is given by

$$f(w, b) = \frac{1}{|\mathcal{V}_l|} \sum_{i \in \mathcal{V}_l} \varepsilon_i. \tag{1}$$

转化为了约束
最优化问题

Degree Dependent Regularization

- W depends on degree
- Add Degree Dependent Regularization p
- $p = (p_1, p_2, \ldots, p_n)$ 其中 penalty for node v_i is $p_i = g(w_i, d_i)$.

- Linear Weighted Degree **(LWD):**

$$g(w_i, d_i) := |w_i| d_i$$

- Linear Weighted Root Degree **(LWRD):**

$$g(w_i, d_i) := |w_i| \sqrt{d_i}$$

- Linear Weighted Root Log Degree **(LWRLD):**

$$g(w_i, d_i) := |w_i| \sqrt{\log_2 d_i}$$

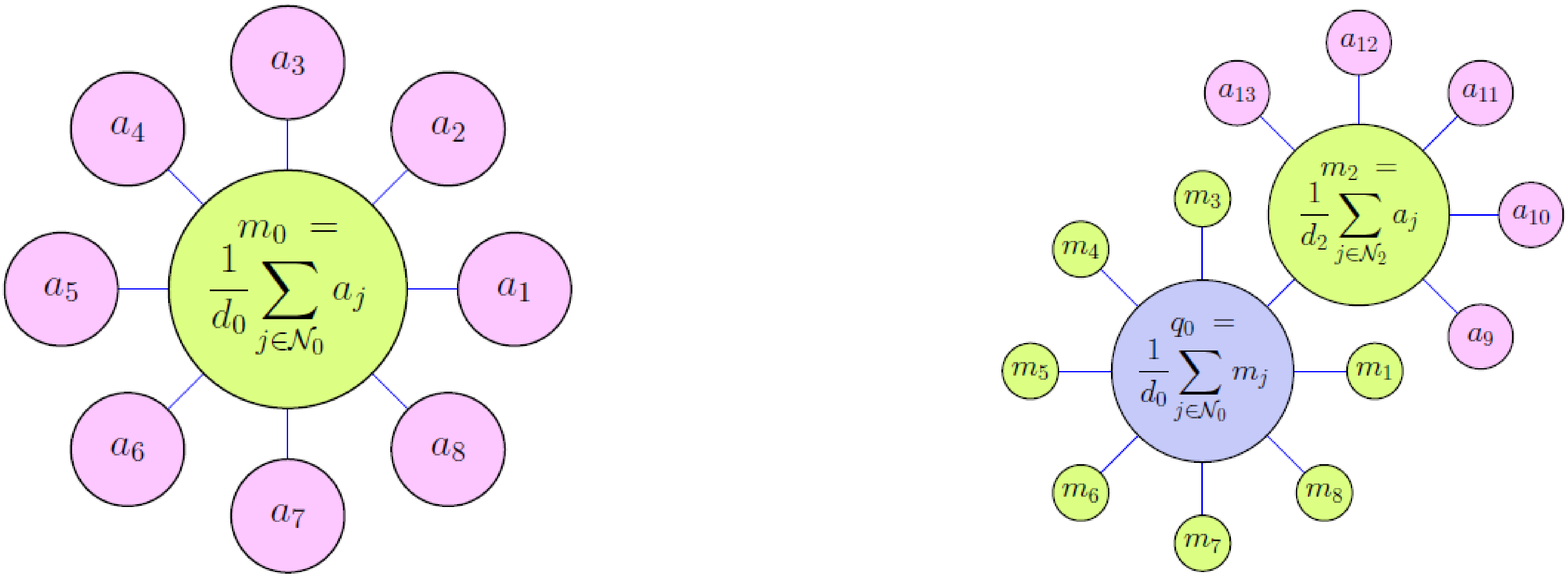
The objective function

Add a parameter λ

$$\min_{w,b} \quad \frac{\lambda}{2}||p||^2 \quad + \quad \frac{1}{|\mathcal{V}_l|}\sum_{i\in\mathcal{V}_l} \varepsilon_i \qquad (2)$$

$$\text{such that} \quad y_i(w^\mathsf{T} m_i + b) \geq 1 - \varepsilon_i \ ,$$
$$\varepsilon_i \geq 0 \ , \text{ and}$$
$$m_i = A\frac{a_i}{d_i}$$

Definition 4. ***r^{th}-Level Neighborhood*** *of a node v_i is defined as a multiset $\mathcal{N}_i^r$ s.t. $v_k \in \mathcal{N}_i^r$ if and only if there is an edge in graph $\mathcal{G}$ connecting nodes v_k and v_j where node $v_j \in \mathcal{N}_i^{r-1}$, and multiplicity of v_k in $\mathcal{N}_i^r$ is given by the cardinality of set $\{v_j | (v_k, v_j) \in \mathcal{E}$ and $v_j \in \mathcal{N}_i^{r-1}\}$.*



(a) First-Level Neighborhood

(b) Second-Level Neighborhood

$$y_i \left(\sum_{j \in \mathcal{N}^1_{i,y_i}} A_{ij}(w^\mathsf{T} \cdot m_j + b) \; + \sum_{j \in \mathcal{N}^1_{i,-y_i}} A_{ij}(w^\mathsf{T} \cdot m_j + b) \right) \geq \; 0, \quad \forall i = 1, \ldots, n$$

$$y_i \quad \left(\frac{1}{d_i} w^\mathsf{T} \cdot A \cdot m_i + b \right) \quad \geq \quad 0, \quad \forall i = 1, \ldots, n$$

where, $d_i \;\; = \;\; \sum_{j \in \mathcal{V}} A_{ij}$.

Let $Q = [q_1, q_2, \cdots, q_n] = MAD^{-1} = A(AD^{-1})^2$, where M and D are defined in section 3. Then, for the linearly separable case, we have,

$$y_i(w^\mathsf{T} q_i + b) \geq 0, \quad \forall i = 1, \ldots, n$$

where $q_i = A^2 D^{-1} \dfrac{a_i}{d_i}$ based on second-level neighborhood of a node. We can argue using inductive logic that while using r-level neighbors, the same will be mapped to $A(AD^{-1})^{r-1} \dfrac{a_i}{d_i}$. For classification using r-level neighbors, the adjacency information contained in the matrix A as a whole gets mapped to $A(AD^{-1})^r$.

How to Random Walk

- Random Walk：

with increasing values of r, the transition probabilities between any pair of nodes start converging towards the stationary probability distribution.

- Lazy Random Walk:

（1） length is not fixed.

（2） Use dampening factor at each hop, which controls the termination of the random walk

- If dampening factor is (… the random walk terminates at each hop …), and … the next hop with probab… enta… obtained using the above pr… given b…

if γ is chosen to be close to zero, it is equivalent to learning with the given adjacency representation alone.

On the other extreme, if gamma is large (≈ 1), the same as random walk

$$q_i = (1-\gamma)A\left(e_i + \gamma\frac{a_i}{d_i} + \gamma^2(AD^{-1})\frac{a_i}{d_i} + \gamma^3(AD^{-1})^2\frac{a_i}{d_i} + \dots\right)$$

e_i being a n-dimensional unit vector with i^{th} entry as 1 and remaining as zeros.

Structured RandomWalk

- higher degree implies a higher termination probability and a lesser dampening factor

$$\gamma_i = \frac{1}{\log_2 d_i}.$$

We define matrix Γ as,

$$\Gamma = diag(\gamma_1, \gamma_2, \ldots, \gamma_{|\mathcal{V}|}).$$

The representation then becomes

$$q_i = A\Big(e_i + \Gamma\tfrac{a_i}{d_i} + (AD^{-1})\Gamma^2\tfrac{a_i}{d_i} + (AD^{-1})^2\Gamma^3\tfrac{a_i}{d_i} + \ldots\Big)(I - \Gamma).$$

Let Q be the matrix obtained by stacking column vectors q_i corresponding to all the nodes. Then, we have

$$Q = A(I - AD^{-1}\Gamma)^{-1}(I - \Gamma).$$

The core of Algorithm

Taking structured random walk into account the objective in (2) is modified as:

$$\min_{w,b} \quad \frac{\lambda}{2}||p||^2 \quad + \quad \frac{1}{|\mathcal{V}_l|}\sum_{i\in\mathcal{V}_l} \quad \varepsilon_i \tag{4}$$

$$\text{s.t.} \quad y_i(w^{\mathsf{T}}q_i + b) \geq 1 - \varepsilon_i \ ,$$
$$\varepsilon_i \geq 0 \ , \text{ and}$$
$$q_i = A\Big(e_i + \Gamma\tfrac{a_i}{d_i} + (AD^{-1})\Gamma^2\tfrac{a_i}{d_i} + (AD^{-1})^2\Gamma^3\tfrac{a_i}{d_i} + \ldots\Big)(I-\Gamma)$$

Subgradient of the above is given by,

$$\nabla_t = \lambda p\frac{\partial p}{\partial w} - \frac{1}{|\mathcal{V}_l|}\sum_{i\in\mathcal{V}_l} \mathbb{1}[y_i w_t^{\mathsf{T}} q_i < 1] y_i q_i,$$

where $\mathbb{1}$ denotes indicator function. Using gradient descent, the iterative update rule for w is given by

$$w_{t+1} = w_t - \eta_t \nabla_t$$

where η_t is the learning rate for the t^{th} iteration. We use stochastic gradient descent mini-batch update algorithm with a variable learning rate η_t given by $\eta_t = \dfrac{1}{2+\lambda t}$.

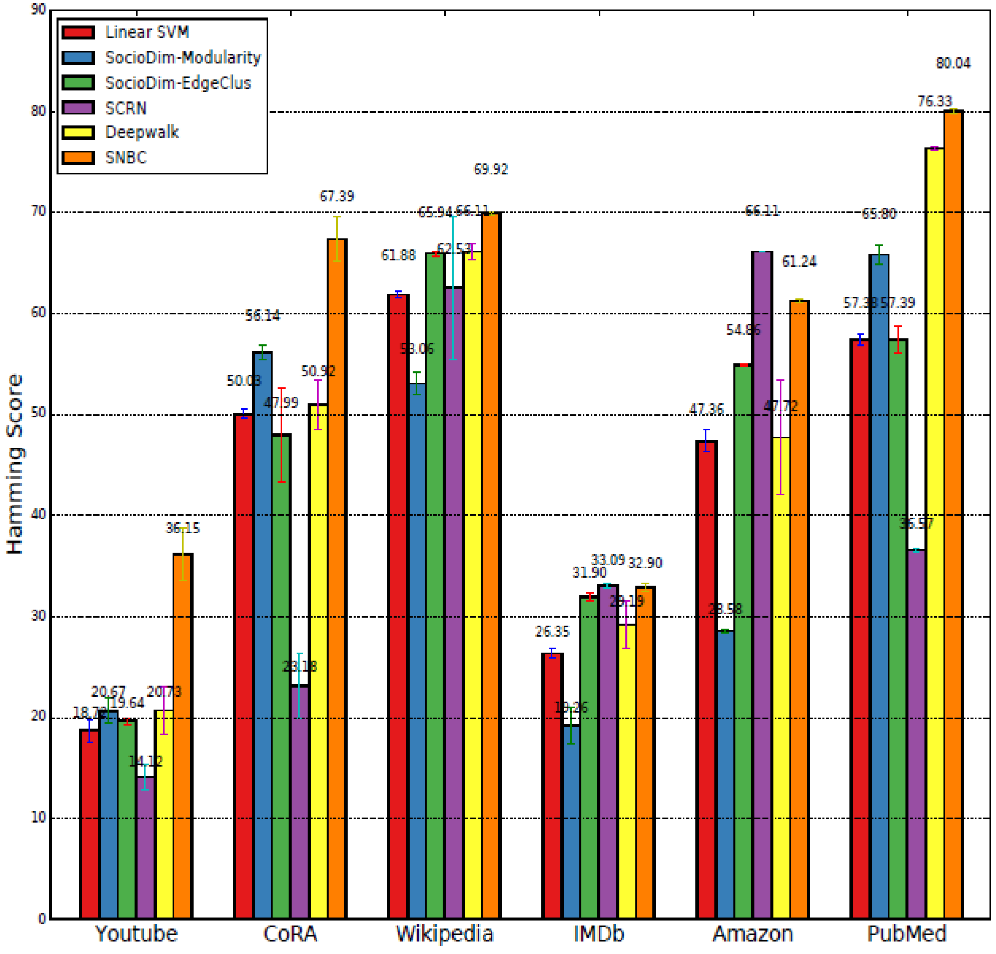
Experienment

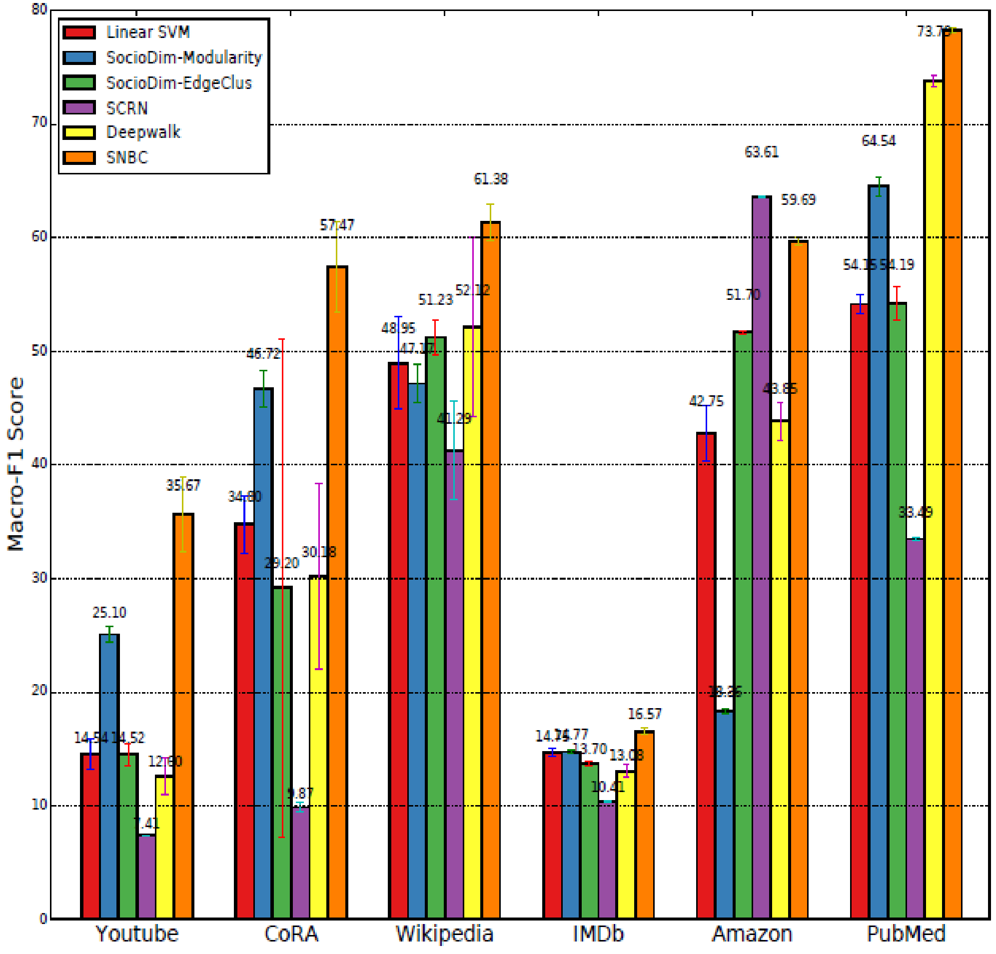
If for the i_{th} node, T_i is the set of true labels, and P_i is set of predicted labels

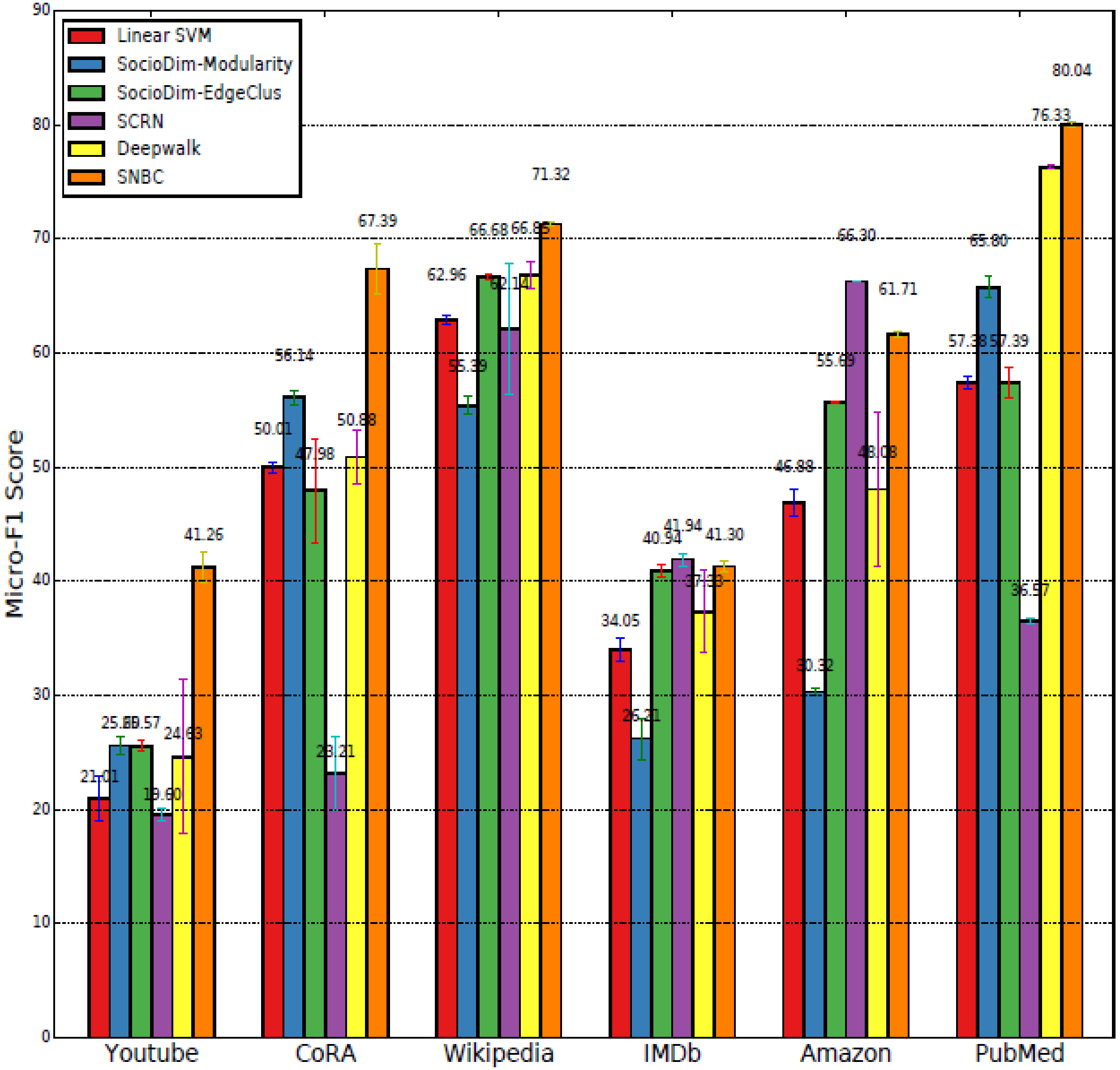
$$\text{Hamming Score} = \sum_i \frac{|T_i \cap P_i|}{|T_i \cup P_i|}$$

$$\text{Micro-}F_1 \text{ Score} = \frac{2\sum_i |T_i \cap P_i|}{\sum_i |T_i| + \sum_i |P_i|}$$

$$\text{Macro-}F_1 \text{ Score} = \frac{1}{k}\sum_{j=1}^{k} \frac{2\sum_{i\in\mathcal{C}_j} |T_i \cap P_i|}{\sum_{i\in\mathcal{C}_j} |T_i| + \sum_{i\in\mathcal{C}_j} |P_i|}$$


Hamming Score
Linear SVM
SocioDim-Modularity
SocioDim-EdgeClus
SCRN
Deepwalk
SNBC
Youtube
CoRA
Wikipedia
IMDb
Amazon
PubMed


Macro-F1 Score
Linear SVM
SocioDim-Modularity
SocioDim-EdgeClus
SCRN
Deepwalk
SNBC
Youtube
CoRA
Wikipedia
IMDb
Amazon
PubMed


Linear SVM
SocioDim-Modularity
SocioDim-EdgeClus
SCRN
Deepwalk
SNBC
Micro-F1 Score
Youtube
CoRA
Wikipedia
IMDb
Amazon
PubMed
21.01
19.60
24.63
41.26
50.01
56.14
47.98
23.21
50.88
67.39
62.96
55.39
66.68
66.85
71.32
34.05
26.21
40.94
41.94
41.30
46.88
30.32
55.69
66.30
48.08
61.71
57.38
65.80
57.39
36.57
76.33
80.04

Thank you！