Structural Neighborhood Based Classification of Nodes in a Network

Sharad Nandanwar
Department of Computer Science and Automation
Indian Institute of Science
Bangalore, India
sharadnandanwar@csa.iisc.ernet.in

M. N. Murty
Department of Computer Science and Automation
Indian Institute of Science
Bangalore, India
mnm@csa.iisc.ernet.in

ABSTRACT

Classification of entities based on the underlying network structure is an important problem. Networks encountered in practice are sparse and have many missing and noisy links. Statistical learning techniques have been used in intra-network classification; however, they typically exploit only the local neighborhood, so may not perform well. In this paper, we propose a novel structural neighborhood-based classifier learning using a random walk. For classifying a node, we take a random walk from the node and make a decision based on how nodes in the respective k^{th}-level neighborhood are labeled. We observe that random walks of short length are helpful in classification. Emphasizing role of longer random walks may cause the underlying Markov chain to converge to a stationary distribution. Considering this, we take a lazy random walk based approach with variable termination probability for each node, based on the node's structural properties including its degree. Our experimental study on real world datasets demonstrates the superiority of the proposed approach over the existing state-of-the-art approaches.

CCS Concepts

•Computing methodologies → Semi-supervised learning settings; Statistical relational learning; •Human-centered computing → Social networks;

Keywords

Graph-based semi-supervised learning; Relational learning; Collective classification

1. INTRODUCTION

With widespread availability and exponential growth of network data in recent years, there has been a surge of interest in mining network and graph data; this has shown up in many guises. Classification, although a well defined problem in machine learning and pattern recognition literature, has not been studied extensively in the context of networks. In the network domain, the classification problem, often addressed as relational classification, is to assign each node of the network to a well defined community (based on interests, demographics, concepts, etc.) in the presence of its intrinsic features as well as the link structure. Some of the popular applications of node classification are, `Wikipedia` page categorization based on intra-wiki links, genre identification of movies in a movie-actor network, social circle learning in a friendship network etc. [25, 17]

To motivate the problem, we use a toy example of a citation network shown in figure 1, where nodes in the network represent papers published in the past proceedings of a conference, and edges depict the citations made by these papers. Each track in the conference where papers are published represents a class, which are distinguished by the color in figure 1. Nodes with missing color information correspond to new submissions which are unlabeled. For a new submission in an upcoming conference, the classification task is to identify the track based on the citations made by the authors in the paper. Networks generally observe homophily, i.e., nodes have a tendency to connect to other nodes that are similar to them. Because of this, it can be claimed that citations made in a paper are from an identical or a correlated track. Further, citations made by authors may range over multiple publishers who opt for different schema standards. This leads to heterogeneity in the metadata. Obtaining precise track information for such papers may have its own cost overheads. In the real world, getting labeled data for learning a supervised model is always expensive and the process involved is also tedious. So, the problem boils down to learning from a small set of labeled examples.

Generally, the study of networks revolves around social networking websites because of their popularity. However, in this paper, by "network" we refer to a generic network, be it a co-citation network, friendship network, biological network, hyper-link structure of the Internet, astronomical network, etc. Network Analysis is not only limited to data showing linked structure; other unstructured data expressed in high dimensional space can also be transformed into weighted graphs so as to understand the topology of the data in a better way. Several approaches have been suggested in the literature for transforming non-graph data into graph data, by using measures like Gaussian similarity [15].

KDD '16, August 13-17, 2016, San Francisco, CA, USA
 ISBN 978-1-4503-4232-2/16/08…$15.00
DOI: http://dx.doi.org/10.1145/2939672.2939782

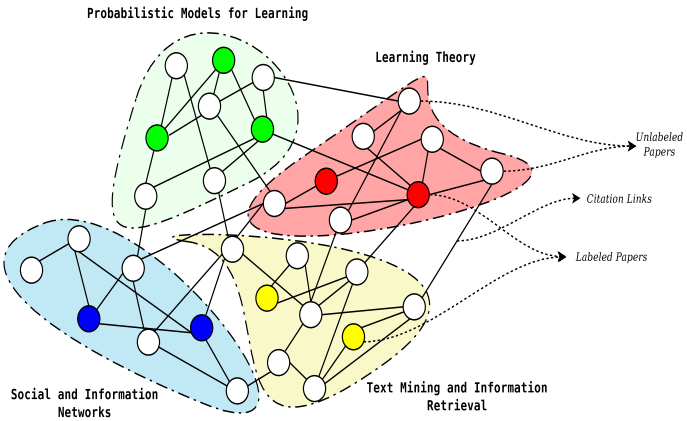


Figure 1: Illustration of multi-class classification in a Citation Network

Conventionally, networks are represented by graphs, defined as follows.

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 $\mathcal{V}_u = \mathcal{V} \setminus \mathcal{V}_l$.*

Many times, structure of the graph is noisy because of missing and noisy edges. While using traditional statistical learning tools like Support Vector Machines (SVM), we may be losing crucial information if we learn on the basis of a node's local neighborhood alone. Such locally oriented models may not be able to capture global or long distance relationships. Techniques from link prediction (another sub-domain in network analysis) can be leveraged to make the representation more precise. On these lines, many graph kernels [8, 9, 19] were proposed, which compute global structural similarity between pairs of nodes. However, when the network size is in the order of millions, such graph kernel based approaches also suffer due to the high computational requirements and overhead involved in searching over the parameter space.

Unlike the traditional supervised learning techniques, the features in a relational data scenario correspond to the data instances themselves. Hence, since features and instances are interdependent, so are the data instances, rendering the otherwise usual *i.i.d.* assumption invalid. Even though the network is sparsely labeled ($|\mathcal{V}_l| << |\mathcal{V}|$), the connectivity between nodes imparts additional knowledge, strengthening the case for a classifier. In accordance with the principle of homophily [1], neighboring nodes in a network are supposed to be similar to each other. Also, as per the clustering hypothesis, nodes forming a dense sub-network should be correlated [1]. These assumptions, along with the knowledge of connectivity, can be used to formulate the problem as a semi-supervised or transductive learning problem in graph. 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. A random walk in the limit gets attracted towards dense sub-networks in the graph. Hence, a majority of nodes in such a dense sub-network share the same class label.

The principal contributions of our work are as follows:

- *We propose a novel approach for intra-network classification, which enriches the adjacency structure of a node by exploiting its global neighborhood.*
- *We formulate this as an optimization problem that exploits the network structure.*
- *We provide a learning algorithm based on the stochastic gradient descent approach for the same.*
- *We show effectiveness of the proposed approach by making an exhaustive comparative study with state-of-the-art approaches for intra-network classification.*

Rest of the paper is organized as follows: section 2 describes some of the state-of-the-art techniques in the field of relational classification including collective classification, graph kernels, and social representation learning. The proposed approach for discriminative learning, viz. *Structural Neighborhood Based Classification*, is described in section 3. Section 4 describes how to extend the proposed framework from exploiting the local neighborhood to that of the global neighborhood. A detailed comparative study with existing baseline approaches is made in section 5, followed by discussion and conclusion in section 6 and section 7 respectively. All the source code and datasets used for evaluation are available at `https://github.com/sharadnandanwar/snbc`.

2. STATE-OF-THE-ART

Traditional statistical relational learning methods may not work well in cases where the data is sparsely labeled. In the past decade, several approaches that exploit the relational dependency among the nodes have been proposed. We summarize some of the influential methods below.

Lu and Getoor [12] proposed an approach on the lines of logistic regression, which computes the conditional probability of a class as the product of posterior probabilities conditioned on the node attributes and the node links respectively, and finds a MAP estimate for the class variable. However, similar to the other local classifiers like the k-Nearest Neighbor Classifier and SVM, this approach too cannot view beyond the local neighborhood of the nodes and is hence not robust to noise.

Collective Classification: Collective classification approaches utilize additional knowledge about attributes and class information, based on relationships among entities. Macskassy and Provost [13] introduced a simple relational classifier for network data, the weighted-vote relational neighbor (wvRN) classifier. It computes the class membership as a weighted average of the estimated class membership probabilities of the neighboring nodes. Multi Rank Walk [11], yet another relational classifier, is based on the principle of random graph walks similar to Page Rank [16]. The class of

any unlabeled node is decided as the one which has the highest probability of containing terminal nodes of the random walk.

Social Dimensions for Classification: Networks encountered in practice generally exhibit a community structure, i.e. a certain group of nodes or entities have stronger connections among themselves, as compared to rest of the network. SocioDim [24] is a framework which exploits the cluster hypothesis, which posits that the nodes in the same community are likely to be of the same class. SocioDim attempts to find the latent social dimensions of the network corresponding to the communities and performs classification in these dimensions. However, in networks where community structure has a high conductance, the obtained social dimensions would be inexact, leading to poor classification performance. Wang et al. [25] propose relational learning in a multi-label setting by extracting social context based features. The extracted social context features correspond to hidden causes which make nodes collaborate among themselves. Based on these features, an iterative probabilistic process similar to wvRN is adopted. DeepWalk [17] and LINE [22] are recently proposed approaches for Social Representation Learning based on random walks. These methods try to capture neighborhood similarity and community membership in latent representations. These label independent representations are then used in multi-label classification.

Random Walk based approaches: Zhou et al. [27] proposed a globally consistent learning approach on the lines of spectral clustering. The proposed iterative method updates the label of a node using information the node receives from its neighbors. Label propagation algorithm [28] adapts a probabilistic perspective using Markov random walks, wherein structure is used to compute transition probabilities. Hidden labels of a node are thereafter inferred using its ancestors' hidden/observed labels. In [2], Baluja et al. proposed a controlled random walk over a graph, with three possible actions (*inject, continue, abandon*) at each step. *Inject* causes the random walk to stop and return its current label, *continue* continues the random walk to its neighbors, and *abandon* terminates the walk without performing any labeling. [21] proposed a modification to this by defining a well-behaved objective function which such a framework minimizes, thus guaranteeing convergence.

Learning using Graph Kernels: In addition to the above schemes, many graph kernels that exploit structural information have also been proposed. Link prediction techniques based on Common Neighbors, Adamic-Adar, Resource Allocation Index, etc. [10] provide means for computing similarity between nodes. They, however, consider only the local neighborhood of nodes and neglect long range relationships. Graph kernels, on the other hand, capture the notion of global similarity between the nodes. A simple kernel would count the number of paths of fixed length between the nodes. It is well-known that longer paths between two nodes are less significant compared to shorter ones. Exponential and von-Neumann graph kernels [8] compute a weighted mean of all such path counts giving higher weights to shorter paths. The number of paths of length n between nodes v_i and v_j are reflected in A^n_{ij}, where A is the adjacency matrix. Then, a von Neumann kernel is defined as,

$$\sum_{i=0}^{\infty} \gamma^i A^i = (I - \gamma A)^{-1},$$

where $\gamma \in (0, 1)$. Exponential graph kernel, very similar to the above, is defined as,

$$\sum_{i=0}^{\infty} \frac{\gamma^i}{i!} A^i = \exp(\gamma A).$$

Laplacian counterparts of the above kernels have also been defined, namely, regularized Laplacian kernel [19] and heat diffusion kernel [9] respectively. Graph kernels usually work well on smaller networks. For learning on large networks, these approaches require inverting large matrices, which is not feasible when the resources are limited.

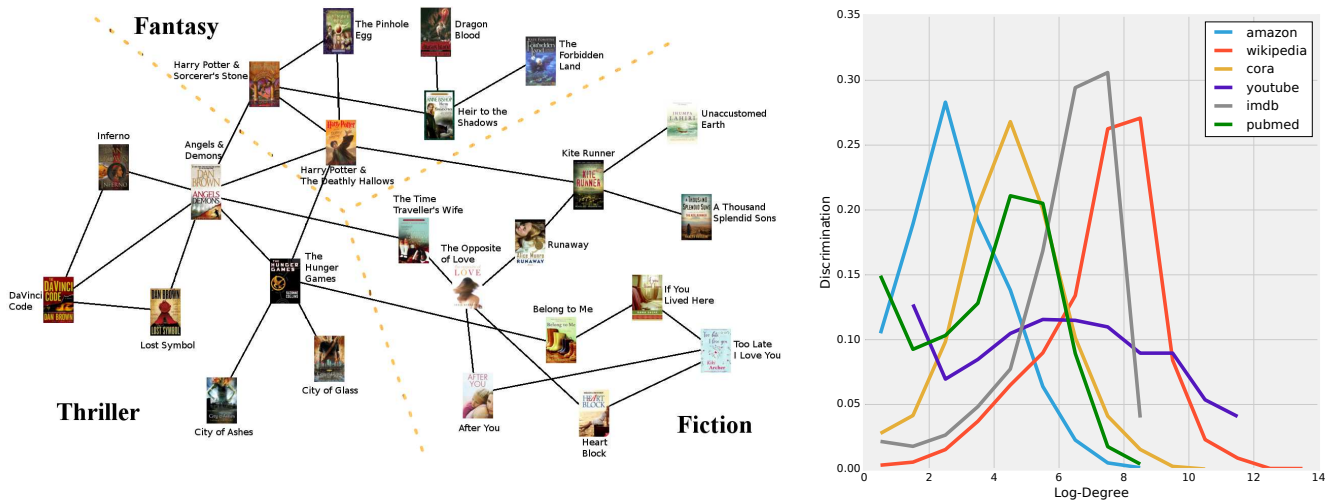
On a side note, relational classifiers described above do not have good generalization abilities and suffer when the given network is sparsely labeled.

A heterogeneous information network [20] is a network composed of multiple types of nodes and relationships between them. Study of these networks has lately gained prominence. Chakrabarti et al. [3] showed that making use of hyper-links along with hypertext while classifying linked text content leads to an enhanced performance. Ming Ji et al. [7] proposed a framework for transductive learning in heterogeneous networks based on two assumptions: local consistency (class assignment of neighbors is similar) and ground truth (pre-assigned class labels are correct). In [6], ranking and classification in heterogeneous information networks are combined based on the intuition that highly ranked objects within a class should play a more important role in classification.

3. STRUCTURAL NEIGHBORHOOD BASED CLASSIFICATION (SNBC)

Traditional learning approaches like the k-Nearest Neighbor classifier and SVM use local adjacency information in classification. In a sparsely labeled network, it becomes difficult to classify a node if there are not sufficient number of labeled nodes in its vicinity. In accordance with homophily, two nodes connected by an edge are expected to be similar to each other. We assume that the behavior of a node is defined by the average behavior of its neighboring nodes, and come up with a novel way of learning in networks. Most of the statistical classifier learning problems are inherently binary classification problems, which can be easily extended for multi-class classification using standard approaches like one-vs-one and one-vs-all [14]. Henceforth, in this paper, we will be working with a binary classification problem, which can be extended to deal with multi-class classification like others.

Let $\mathcal{C}^+$ and $\mathcal{C}^-$ represent the sets of positive and negative examples respectively in the binary classification problem under consideration. A simple approach to classify a node would be to count the number of neighbors from the respective classes and label the node based on majority voting. However, this approach has its own limitations. If the network is sparsely labeled, nodes may lack sufficient number of labeled neighbors. Macskassy [13] showed that instead of counting votes, taking a weighted vote of each neighbor is more useful. The weight of a vote expresses the confidence with which a neighbor can be assigned to a class. In a probabilistic model, this confidence can be measured using conditional class probabilities. In classifiers where a real decision function is learned, a larger distance from the decision boundary signifies a higher weight.



(a) (b)

Figure 2: (a)Toy network of books modeling similarity relationship among books based on user preferences. (b)Plot showing expected contribution of nodes with degree d towards discrimination i.e. $\Delta(d)$

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,

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

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 $j \in \{1, 2, \cdots, n\}$. For node v_i, we define the first-level/local neighborhood as follows:

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., $(v_i, v_j) \in \mathcal{E}$.*

Further, let $\mathcal{N}_{i,+}^1 \subseteq \mathcal{C}^+$ and $\mathcal{N}_{i,-}^1 \subseteq \mathcal{C}^-$ denote the sets of neighboring nodes from positive class and negative class respectively in the binary labeled network. In this setting, we assume a linear separating hyperplane (like the one in perceptron, logistic regression, SVM) of the form $w^\mathsf{T} \cdot x + b$. Then, given the parameters w and b of an optimal decision boundary, for an unknown example x_u, the predicted label is given by

$$\hat{y}_u = sign(w^\mathsf{T} \cdot x_u + b)$$

Because of homophily in the network, a node is expected to have its class label (defined by its properties) similar to majority of its neighbors. For a target node v_i, the above scoring function assigns a positive or negative value ($\hat{y}_j$) to its neighboring node (v_j). As most of the neighboring nodes are expected to have their class label same as that of the target node, the aggregated sum of scores from these neighboring nodes should also have the same sign as the target label (y_i).

Also, in real networks, variability in the node degree is inherent and follows a power law distribution. It is observed that high degree nodes are generally the source of linkage noise in networks. For example consider the toy network of books shown in figure 2a. The network models "users who liked this also liked" similarity relation among books. In the graph, **Angels and Demons** shares a similarity relation with most of the books in the **Thriller** category. This entices one to become prejudiced that any book that shares such a relation with **Angels and Demons** will belong to the **Thriller** category. However, contrary to this intuition, the same book shares many relations with books in other categories too, as is visible in the graph. This can be attributed to the *"rich getting richer"* phenomenon in networks, i.e., the book went popular and was enjoyed by readers across the genres. Based on this observation, we argue that a link to a low or medium degree node should be considered more reliable in comparison with a link to a high degree node. Thus, while learning the weight vector w, if an almost equivalent performance can be achieved by (1) assigning larger weights to a large number of medium degree nodes, than (2) assigning larger weights to a small number of high degree nodes, then the former should be preferred. Figure 2b shows the expected discriminatory power (defined below) with varying degree. The graph shows how the relative average mutual information varies for nodes lying in different degree zones. We define a measure of discrimination Δ for degree d as follows:

$$\Delta(d) = \frac{1}{Z} \sum_{i \in \{j : d_j = d\}} \max_{c \in \mathcal{C}} MI(a_i, y_c),$$

where Z is the normalization constant, and $MI(X, Y)$ computes the mutual information of random variables X and Y, d_j is the degree of node v_j, and y_c refers to the label vector of class c. We observe a consistent behavior across all datasets considered in our experiments, however the peaking range is different for each dataset, which would depend on the network size and its structure, more specifically on its sparsity. Based on these observations, we state the following conjectures:

Conjecture 1. *For a node v, its neighbors $\mathcal{N}_v$ and the decision boundary ($\mathcal{H}$), sum of distances between $\mathcal{H}$ and neighbors $\mathcal{N}_v^{(s)}$ similar to v (i.e., of the same class as v) should be greater than the same for other neighbors ($\mathcal{N}_v \setminus \mathcal{N}_v^{(s)}$).*

Conjecture 2. *The role of high degree nodes in discrimination is smaller compared to that of medium degree nodes.*

We formalize our intuition as follows:

(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,

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

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 $m_i = A\frac{a_i}{d_i}$, 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

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

(2) Degree Dependent Regularization: Weight w_i for node v_i with degree d_i should be affected in a manner directly proportional to a monotonically increasing function of d_i. Let $g : (\mathbb{R}^+ \bigcup \{0\}) \times \mathbb{Z}^+ \to \mathbb{R}$ be the penalty function such that penalty for node v_i is $p_i = g(w_i, d_i)$. We define penalty for the network as a whole by the ℓ_2 norm of vector $p = (p_1, p_2, \ldots, p_n)$, and try to minimize the squared penalty. In this work we consider the following penalty functions:

- 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):**

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

Regularizing the objective in (1) corresponding to empirical loss with the above penalty using regularization parameter λ would lead to,

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

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

4. DIVING DEEP INTO THE NETWORK

The objective in (2) attempts to learn parameters (w and b), while labeling the node under consideration based on the collective behavior of its first-level neighborhood. We further explore the significance of far away neighbors in this setting. We begin by giving a recursive definition of the r^{th}-level neighborhood of a node:

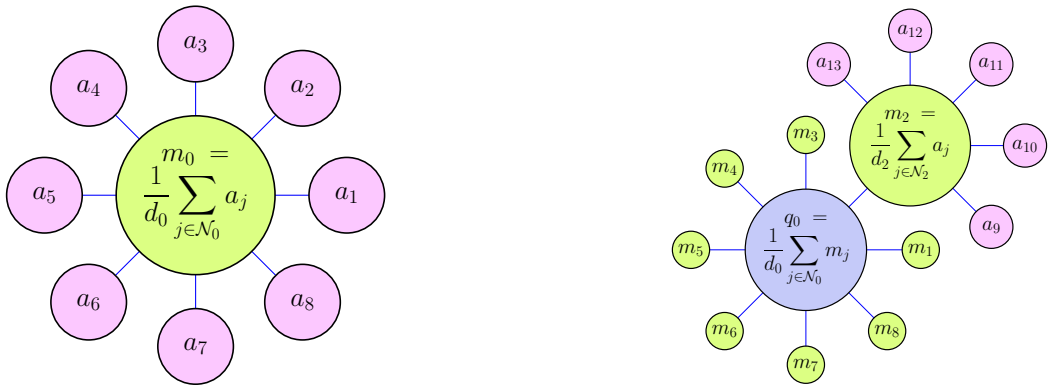
DEFINITION 4. ***r^{th}-Level Neighborhood*** *of a node v_i is defined as a multiset $\mathcal{N}_i^r$ s.t. $v_k \in \mathcal{N}_i^r$ if and only if there is an edge in graph $\mathcal{G}$ connecting nodes v_k and v_j where node $v_j \in \mathcal{N}_i^{r-1}$, and multiplicity of v_k in $\mathcal{N}_i^r$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

Note that the multiplicity of a node v_k in multiset $\mathcal{N}_i^r$ indicates the number of paths of length r possible between nodes v_i and v_k. Proceeding on similar lines as in section 3, we deal with the second-level neighborhood, which can be generalized for any depth in the network. While considering the first-level neighborhood $\mathcal{N}^1$, we were interested in how nodes in $\mathcal{N}^1$ get classified based on adjacency information they have. When using second-level neighborhood, we first accumulate scores at first-level neighbors using decision values of second-level neighbors. These scores from first-level neighbors are further propagated to the node under consideration to perform classification. Figure 3 illustrates the involved process (3a) for first-level neighborhood based classification, and (3b) when using second-level neighborhood in classification.

In section 3 we have derived the representation m_i for node v_i based on its first-level neighborhood as $m_i = A\frac{a_i}{d_i}$. We carry forward this representation while delving into higher level neighborhoods. Using similar homophily arguments at each node, we have,

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

which simplifies to



(a) First-Level Neighborhood

(b) Second-Level Neighborhood

Figure 3: For an unweighted graph Figure 3(a) illustrates how representation for node v_0 based on its first-level neighbors is derived while 3(b) illustrates the representation for the same based on its second level neighbors. $\mathcal{N}_i$ denotes the set of first-level neighbors for node v_i with degree $d_i = |\mathcal{N}_i|$

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

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

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 $q_i = A^2 D^{-1} \frac{a_i}{d_i}$ 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 $A(AD^{-1})^{r-1}\frac{a_i}{d_i}$. For classification using r-level neighbors, the adjacency information contained in the matrix A as a whole gets mapped to $A(AD^{-1})^r$.

Random Walk Based View: The above scheme can also be viewed as taking random walks of fixed length r starting from the node under consideration, and classifying the node by accumulating the decision values for respective neighbor nodes. With higher values of r, the representation derived above tends to become more global. However, with increasing values of r, the transition probabilities between any pair of nodes start converging towards the stationary probability distribution. Because of this, in a connected graph for sufficiently large values of r, every node will eventually end up having a similar representation. Because of this, one can expect some loss in discrimination for higher values of r.

Lazy Random Walk: To avoid the situation where all nodes have identical representation, we consider taking lazy random walks of arbitrary lengths instead of a fixed length r. We start by introducing a dampening factor at each hop, which controls the termination of the random walk. If dampening factor is denoted by $\gamma \in [0, 1]$, the random walk terminates at each hop with probability $(1 - \gamma)$, and continues to the next hop with probability γ. The representation q_i for node v_i obtained using the above procedure is given by

$$q_i = (1-\gamma)A\Big(e_i + \gamma\frac{a_i}{d_i} + \gamma^2(AD^{-1})\frac{a_i}{d_i} + \gamma^3(AD^{-1})^2\frac{a_i}{d_i} + \ldots\Big), \tag{3}$$

e_i being a n-dimensional unit vector with i^{th} entry as 1 and remaining as zeros. Other notations used have already been defined. This closely resembles the regularized graph laplacian kernel [19] using unnormalized graph laplacian. However, instead of computing structure based similarity matrix, we are interested in a structural similarity based representation. In equation 3, 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), higher powers will dominate. This may lead to every node having similar representation as mentioned previously. We empirically find that taking such an approach does not lead to a significant improvement in the performance of the learned classifier.

Structured Random Walk: As stated earlier, networks observed in practice exhibit a variability in degree. It is easier to classify higher degree nodes than it is to classify low degree nodes, since the former have a richer local neighborhood. Low degree nodes have limited information owing to their smaller local neighborhood. In addition, if the dampening factor is high, then continuing from a high degree node may lead to accumulation of more noise at each successive hop. Considering these consequences, we decide upon having a separate dampening factor specific to a node, based on its structural property. Intuitively, on encountering a high degree node, a random walk should observe higher termination probability $(1 - \gamma)$, and the same should be lower in case of low degree nodes. We use randomness of a node as a measure to derive respective dampening factors, which is defined for node v_i as follows:

$$H_i = -\sum_{j \in \mathcal{N}_i} p_{ij} \log_2(p_{ij}),$$

where $\mathcal{N}_i$ denotes the set of neighbors of node v_i, as defined earlier. Assuming uniform transition probabilities for all neighbors of a node, we have $p_{ij} = \frac{1}{|\mathcal{N}_i|}$, which gives

$$H_i = \log_2 |\mathcal{N}_i| = \log_2 d_i$$

As stated earlier, higher randomness implies a higher termination probability and a lesser dampening factor. Taking this into account, we choose

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

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

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

4.1 Structural Neighborhood Based Classification (SNBC) Algorithm

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

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

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

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 $\eta_t = \frac{1}{2+\lambda t}$. Algorithm 1 provides the pseudocode for the adopted approach with a LWRLD penalty function.

Algorithm 1 Training Structural Neighborhood Based Classifier with LWRLD penalty using stochastic gradient descent

procedure SNBC_TRAIN $(\boldsymbol{A}, \boldsymbol{y}, \lambda, imax, k)$

Input:
- $\boldsymbol{A}_{n\times n}$ ▷ Adjacency matrix representation of network
- $\boldsymbol{y}_{n\times 1}$ ▷ Label vector, $\boldsymbol{y}_i \in \{-1,\ 1\}$ if i^{th} node is labeled, $\boldsymbol{y}_i = 0$ otherwise
- λ ▷ Regularization parameter
- $imax$ ▷ Maximum number of iterations for SGD
- k ▷ Sample size for SGD

Output:
- $\boldsymbol{w}_{imax}, b$ ▷ Learned decision function parameter

$Tr \leftarrow \{i : \boldsymbol{y}_i \neq 0\}$ ▷ Set of training points
$\boldsymbol{d}_{n\times 1} \leftarrow \boldsymbol{A}\ \boldsymbol{1}$ ▷ Degree Vector
$\boldsymbol{D} \leftarrow diag(\boldsymbol{d})$
Define log degree vector $\boldsymbol{l}_{n\times 1}$ s.t. $\boldsymbol{l}_i \leftarrow \log_2(2 + \boldsymbol{d}_i)$
$\boldsymbol{L} \leftarrow diag(\boldsymbol{l})$
$\boldsymbol{\Gamma} \leftarrow diag^{-1}(\boldsymbol{l})$
$\boldsymbol{X} \leftarrow \boldsymbol{A}(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{D}^{-1}\boldsymbol{\Gamma})^{-1}(\boldsymbol{I} - \boldsymbol{\Gamma})$
$\boldsymbol{w}_0 \leftarrow \boldsymbol{0}_{n\times 1}, b \leftarrow 0$ ▷ Initialization
for $t = 1, 2, \ldots, imax$ **do**
  $b \leftarrow \frac{1}{k}\sum_{i\in Tr}(\boldsymbol{y}_i - \boldsymbol{x}_i^\mathsf{T}\boldsymbol{w}_{t-1})$
  Choose $Smp \subseteq Tr$, where $|Smp| = k$, at random
  $Mc \leftarrow \{i \in Smp : \boldsymbol{y}_i(\boldsymbol{x}_i^\mathsf{T}\boldsymbol{w}_{t-1} + b) < 1\}$
  $\boldsymbol{w}_t \leftarrow (\boldsymbol{I} - \eta_t\boldsymbol{L})\boldsymbol{w}_{t-1} + \frac{\eta_t}{k}\sum_{i\in Mc}\boldsymbol{y}_i\boldsymbol{x}_i$
end for
return $\boldsymbol{w}_{imax}, b$
end procedure

5. EXPERIMENTS

We empirically evaluate the results of the proposed approach SNBC over various real world datasets. In order to show the effectiveness of SNBC we compare it with some of the recent and state-of-the-art intra-network classification schemes.

5.1 Datasets:

We used some of the popular relational datasets described below.

`Youtube:` A subset of `Youtube` users with grouping information made available by Lei Tang[1] is used. The graph models *"knows"* friendship relations amongst users who are assigned to multiple groups according to their interests.

`PubMed:` The `PubMed` data set[2] consists of scientific publications from the `PubMed` database pertaining to diabetes, where each publication is classified into one of three classes: "Diabetes Mellitus, Experimental", "Diabetes Mellitus Type 1", "Diabetes Mellitus Type 2".

`CoRA:` A collection of research articles[3] in the computer science domain classified into pre-defined research topics. For our study, we ignored the citations for which complete information is not available.

`IMDb:` We crawled information about movies from `IMDb`[4]. For generating the graph, we took a subset of English movies released after the year 1990. Then, we defined the similarity between the movies based on the top 5 billed stars in them. Genre assigned to the movies is used as the target variable.

`Amazon Books:` We extracted a subset of books from the amazon co-purchasing network data[5]. Books having less than 5 reviews were ignored. For each book, the dataset also provides a list of other similar books, which is used to build a network. The Genre of books gives a natural

[1] `leitang.net/code/social_dimension/data/youtube.mat`

[2] `linqs.cs.umd.edu/projects/projects/lbc/index.html`

[3] `people.cs.umass.edu/~mccallum/data/CoRA-classify.tar.gz`

[4] `www.imdb.com/interfaces`

[5] `snap.stanford.edu/data/amazon-meta.html`

categorization, which we use as class labels in our learning problem.

Wikipedia: We use a crawled dump of Wikipedia pages from different areas of computer science using the Wikimedia API[6]. For crawling, we choose 16 top level category pages, and recursively crawled subcategories up to a depth of 3. The top level categories are used as class labels.

Some of the statistics of the datasets are summarized in Table 1.

Dataset	#Nodes	#Edges	#Classes	Label Cardinality
Amazon	83742	190097	30	1.546
CoRA	24519	92207	10	1.004
IMDb	19359	362079	21	2.300
PubMed	19717	44324	3	1.000
Wikipedia	35633	495388	16	1.312
Youtube	22693	96361	47	1.707

Table 1: Datasets used for experiments

5.2 Comparative Study:

There is a lot of work on semi-supervised learning in graphs and on collective classification. However, we focus on comparing our work with the state-of-the-art approaches. We restrict our study to some of the recent advances in the field of intra-network classification, and use linear Support Vector Machine (SVM) as the baseline classifier. We briefly describe below, the approaches considered by us for the empirical study.

Linear SVM [4]: We learn from local adjacency vector using Support Vector Machine(SVM) and use this as a benchmark.

SocioDim-Modularity [23]: This approach uses modularity matrix and tries to extract social dimensions hidden in relations among nodes by computing eigenvectors of the modularity matrix. We fix the number of social dimensions as 200 for our experiments.

SocioDim-EdgeClustering [24]: As an extension to SocioDim-Modularity, this approach performs edge clustering by using incidence matrix of the given graph. The hard clusters for edges thus obtained are used to compute latent dimensions of the node. In our experiment, we partition edges into 5000 clusters.

SCRN [25]: SCRN, an approach for collective classification, exploits social context features in relational learning. The social features are computed using edge clustering, in a way similar to SocioDim-EdgeClustering.

Deepwalk [17]: Deepwalk is a recently proposed approach for deep learning in networks. The approach is useful for learning low dimensional embeddings in large networks. We obtain 128 dimensional embeddings for a node using Deepwalk, and use these embeddings further in classification.

5.3 Setup and Parameter-tuning:

To study the robustness of the proposed framework on a sparsely labeled network, we learn a model by holding out the labels of 90%, 70%, 50%, 30% and 10% of nodes respectively. However, because of lack of space here, we show results obtained on using 10% of the labeled nodes for training. Set of nodes used for training is sampled using the Snowball sampling [5] technique. For datasets having directed edges, we remove directionality by adding an edge in the opposite direction also. The datasets used in experiments are multi-labeled, i.e., a node can belong to more than one class. To train SVM in this setting, we use the one-vs-all approach for training a model corresponding to each class. For tuning linear SVM, we restricted the search space to $2^{-10} < \lambda < 2^5$. For each node in the test set, decision values are obtained from the respective class models. We assign s most probable classes to the node using these decision values, where s is equal to the number of labels assigned to the node originally. However, the decision values given by different SVM models cannot be compared directly. We use Platt's Scaling [18] to convert these decision values into probability scores. Probability scores from different models, being on the same scale, can be compared. For validating our results, we use three popular evaluation measures for multi-label classification: Hamming Score, Micro-F_1 Score, and Macro-F_1 Score [26]. If for the i^{th} node, T_i is the set of true labels, and P_i is set of predicted labels, then we have

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

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

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

To verify robustness of various approaches, we take 100 random splits of the network into test and train datasets. We report average Hamming Score, Micro-F_1 Score, and Macro-F_1 Score over these random splits.

6. RESULTS AND DISCUSSION

In this section, we explain the behavior of the proposed approach SNBC based on a systematic experimental study described above. Table 2 reports the results using different penalty functions. We choose the functions LWD, LWRD, and LWRLD as described in section 3. Although the penalty function can have many more forms, our purpose of choosing these functions is to demonstrate the impact of rate of increase of function value with degree. In all the cases, we found LWRLD penalty to be outperforming. The reason is that the other two functions, in addition to penalizing high degree nodes, also heavily penalize medium degree nodes that carry maximum discriminative power. This heavy penalization inhibits learning larger weights for medium degree nodes. However, LWRD was found to be competent in case of small networks like IMDb and PubMed, where the average degree is small. The behavior can be seen as an implication of $\sqrt{d_{avg}} \approx \log_2(d_{avg})$ for small average degree (d_{avg}).

Figures 4, 5, and 6 show respectively the Hamming Score, Macro-F_1 Score, and Micro-F_1 score for multi-label classification of the datasets. In most of the cases, the proposed approach improves over the existing ones. We investigate the reason behind this. As we move from local neighborhood to global neighborhood, the ratio of number of between-class

[6] en.wikipedia.org/w/api.php

Dataset → Penalty Func. ↓	*Measure*	Youtube	PubMed	CoRA	IMDb	Amazon	Wikipedia
LWD	*Hamming Sc.*	18.27 ± 0.09	41.13 ± 1.28	37.74 ± 0.10	32.77 ± 0.15	15.18 ± 0.03	16.13 ± 0.04
	Micro-F1 Sc.	24.29 ± 0.16	41.12 ± 1.37	37.71 ± 0.10	40.93 ± 0.55	17.60 ± 0.15	21.32 ± 0.11
	Macro-F1 Sc.	8.09 ± 0.08	21.63 ± 2.39	6.33 ± 0.11	9.84 ± 0.25	2.84 ± 0.02	7.32 ± 0.19
LWRD	*Hamming Sc.*	35.05 ± 2.79	78.55 ± 0.37	60.46 ± 2.56	32.76 ± 0.12	61.08 ± 0.36	16.08 ± 0.16
	Micro-F1 Sc.	40.64 ± 1.12	78.55 ± 0.37	60.45 ± 2.55	40.91 ± 0.46	61.53 ± 0.45	21.26 ± 0.22
	Macro-F1 Sc.	33.43 ± 2.20	76.35 ± 0.77	49.33 ± 4.38	9.82 ± 0.22	59.67 ± 0.74	7.22 ± 0.27
LWRLD	*Hamming Sc.*	**36.15 ± 2.58**	**80.04 ± 0.19**	**67.39 ± 2.23**	**32.90 ± 0.36**	**61.24 ± 0.19**	**69.93 ± 0.16**
	Micro-F1 Sc.	**41.26 ± 1.34**	**80.04 ± 0.19**	**67.39 ± 2.24**	**41.30 ± 0.44**	**61.71 ± 0.25**	**71.32 ± 0.12**
	Macro-F1 Sc.	**35.66 ± 3.31**	**78.28 ± 0.24**	**57.47 ± 3.96**	**16.56 ± 0.31**	**59.69 ± 0.31**	**61.37 ± 1.58**

Table 2: Evaluation of multi-label classification with different penalty functions, using 10% of labeled data for training.

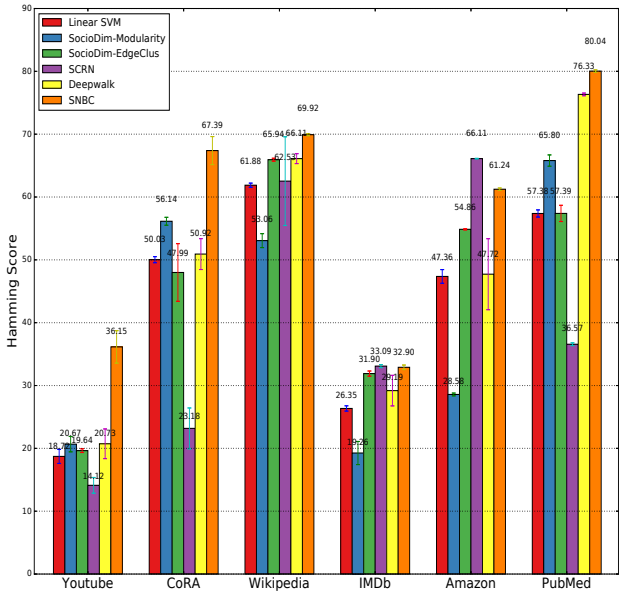


Figure 4: Hamming score for multi-label classification compared with the state-of-the-art approaches using 10% of nodes for training

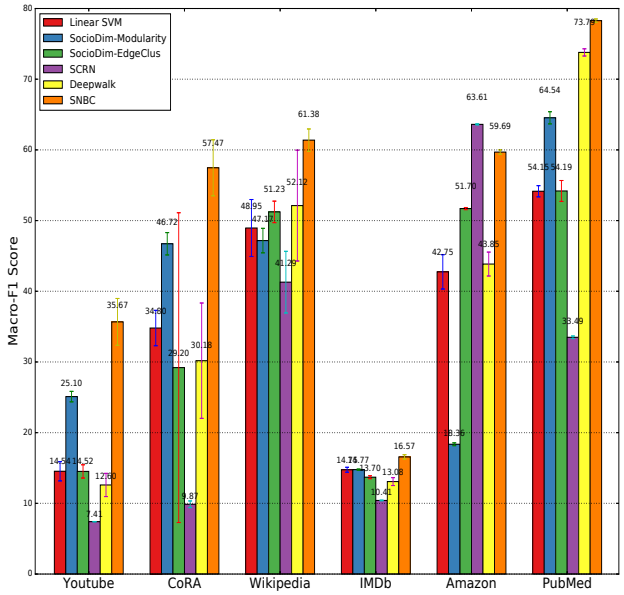


Figure 5: Macro-F1 score for multi-label classification compared with the state-of-the-art approaches using 10% of nodes for training

to within-class paths keeps increasing, i.e., the boundary between classes starts diminishing in an expected sense. Despite all this, SNBC is able to perform well. The reason is that even though inter-class separability diminishes in an expected sense, low degree nodes which earlier had very less information for classification, now have a better representation.

Citation networks are generally sparse, and many low degree nodes can be expected to occur in such a network. In our experimental results, we find that SNBC was able to improve remarkably over all other techniques in both `CoRA` and `PubMed` citation networks. In our setting, where the network is sparsely labeled, some of the features important to low degree nodes got ignored, leading to misclassification. While learning using global neighborhood, these features are extended by exploiting neighborhood information, which leads to improved performance. Similar to citation networks, in case of the `Youtube` social network also, we record a significant improvement for all metrics. `Youtube` data being a real world data, has a similar behavior, i.e., presence of large number of low degree nodes, which get aided when we look into global neighborhood. However, networks like `Amazon` and `IMDb` are synthesized based on similarity. These networks have enough link information for most of the nodes. For example, in the `Amazon` network, for each book, we have a list of five similar books; this makes most of the books to have almost the same degree. Expanding representation using global neighborhood in such a case will only lead to accumulation of noise and hence, will show only marginal or no improvement. Similar is the case for `IMDb` movie network. The `Wikipedia` hyperlink network is a highly noisy network with lots of between-class edges. We find performance of the proposed approach to be significantly better in this case also. In the `Wikipedia` network, most of the noisy links come out of high degree nodes. But the adopted structured walk approach causes the random walk not to move further from high degree nodes, thus inhibiting accumulation of noise.

7. CONCLUSION

The work here proposes and demonstrates an approach for classifying network data based on homophily, cluster hypothesis, and structural properties including degrees of nodes. We propose a structured random walk based ap-

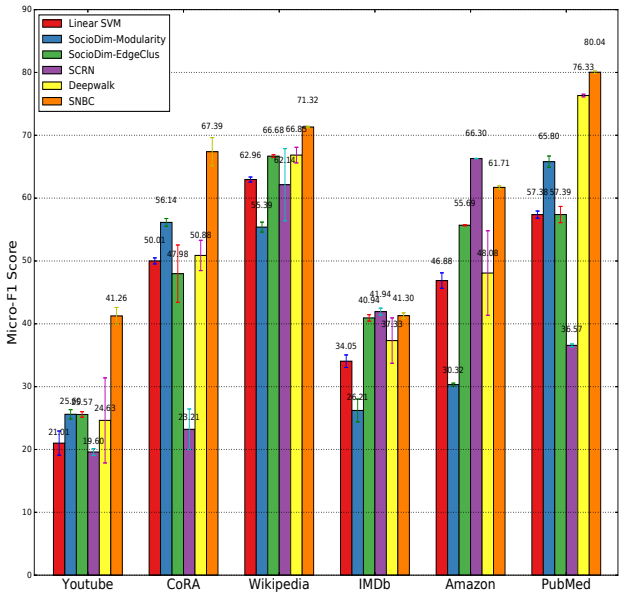


Figure 6: Micro-F1 score for multi-label classification compared with the state-of-the-art approaches using 10% of nodes for training

proach to classification, while emphasizing the role of medium degree nodes in classification. A regularizer that underplays the importance of high degree and low degree nodes is implicitly used. This helps us in achieving robust classification by regulating the noise. A comparative study is made using some of the state-of-the-art intra-network classification approaches, and a baseline classifier demonstrates the effectiveness of our approach.

8. REFERENCES

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

[2] S. Baluja, R. Seth, D. Sivakumar, Y. Jing, J. Yagnik, S. Kumar, D. Ravichandran, and M. Aly. Video suggestion and discovery for youtube: taking random walks through the view graph. In *WWW*, pages 895–904. ACM, 2008.

[3] S. Chakrabarti, B. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. 27(2):307–318, 1998.

[4] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *JMLR*, 9:1871–1874, 2008.

[5] L. A. Goodman. Snowball sampling. *The annals of mathematical statistics*, pages 148–170, 1961.

[6] M. Ji, J. Han, and M. Danilevsky. Ranking-based classification of heterogeneous information networks. In *KDD*, pages 1298–1306. ACM, 2011.

[7] M. Ji, Y. Sun, M. Danilevsky, J. Han, and J. Gao. Graph regularized transductive classification on heterogeneous information networks. In *ECML PKDD*, pages 570–586. Springer, 2010.

[8] J. Kandola, N. Cristianini, and J. S. Shawe-taylor. Learning semantic similarity. In *NIPS*, pages 657–664, 2002.

[9] R. I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *ICML*, pages 315–322, 2002.

[10] D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *JASIST*, 58(7):1019–1031, 2007.

[11] F. Lin and W. W. Cohen. Semi-supervised classification of network data using very few labels. In *ASONAM*, pages 192–199. IEEE, 2010.

[12] Q. Lu and L. Getoor. Link-based classification. In *ICML*, pages 496–503, 2003.

[13] S. A. Macskassy and F. Provost. A simple relational classifier. In *MRDM at KDD*, 2003.

[14] C. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[15] A. Y. Ng, M. I. Jordan, Y. Weiss, et al. On spectral clustering: Analysis and an algorithm. In *NIPS*, pages 849–856. MIT; 1998, 2002.

[16] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: bringing order to the web. 1999.

[17] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *KDD*, pages 701–710. ACM, 2014.

[18] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[19] A. J. Smola and R. Kondor. Kernels and regularization on graphs. In *Learning theory and kernel machines*, pages 144–158. Springer, 2003.

[20] Y. Sun and J. Han. Mining heterogeneous information networks: a structural analysis approach. *ACM SIGKDD Explorations Newsletter*, 14(2):20–28, 2013.

[21] P. P. Talukdar and K. Crammer. New regularized algorithms for transductive learning. In *ECML PKDD*, pages 442–457. Springer, 2009.

[22] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. Line: Large-scale information network embedding. In *WWW*, pages 1067–1077, 2015.

[23] L. Tang and H. Liu. Relational learning via latent social dimensions. In *KDD*, pages 817–826. ACM, 2009.

[24] L. Tang and H. Liu. Leveraging social media networks for classification. *DMKD*, 23(3):447–478, 2011.

[25] X. Wang and G. Sukthankar. Multi-label relational neighbor classification using social context features. In *KDD*, pages 464–472. ACM, 2013.

[26] M.-L. Zhang and Z.-H. Zhou. A review on multi-label learning algorithms. *TKDE*, 26(8):1819–1837, 2014.

[27] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *NIPS*, pages 321–328, 2004.

[28] X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, Citeseer, 2002.