Node Classification in Graph Data using Augmented Random Walk

Hossein Rahmani
Department of Knowledge Engineering (DKE),
Maastricht University, The Netherlands
h.rahmani@maastrichtuniversity.nl

Gerhard Weiss
Department of Knowledge Engineering (DKE),
Maastricht University, The Netherlands
Gerhard.Weiss@maastrichtuniversity.nl

***Abstract*—Node classification in graph data plays an important role in web mining applications. We classify the existing node classifiers into Inductive and Transductive approaches. Among the Transductive methods, the Majority Rule method (MRM) has a prominent role. This method considers only the class labels of the neighboring nodes, neglecting the informative connectivity information in the graph data. In this paper, we propose an Augmented Random Walk (ARW) based approach to resolve main limitations of MRM. In our proposed method, first, we augment the initial graph by adding class labels as new nodes to the graph and then we connect each classified node to its corresponding class label nodes. Second, we apply a Random Walk algorithm to find the similarity score of each un-classified node to different class labels. Third, we predict class labels with the highest scores for the un-classified node. Empirical results show that our proposed method clearly outperforms the Majority Rule method in six graph datasets with high homophily.**

***Index Terms*—Node Classification, Majority Rule, Graph Augmentation, Random Walk.**

I. Problem Statement

Consider an undirected graph $G < V, E >$ with node set V and edge set E, where each node $v \in V$ is annotated with a description $d(v) \in D$ and, optionally, a label $l(v) \in L$. We assume that there exists a "true" labelling function λ from which l is a sample, that is, $l(v) = \lambda(v)$ where l(v) is defined. The task of node classification [1] is to predict the labeling set $l(v_i)$ for each un-classified node v_i. If $|L| = 2$ then the classification problem is called binary classification while if $|L| > 2$ then it is called multi-class classification. In case l(v) associated with a set of labels $Y \subseteq L$ then the classification problem is called multi-label classification [7]. There are two main approaches, Transductive and Inductive, for node classification problem.

In Transductive approach, the task is to predict the label of all the nodes. That is, given the graph $G = (V, E, d, l)$, with l being a partial function, the task is to construct a completed version $G' = (V, E, d, l')$ with l' being a complete function that is consistent with l where l(v) is defined. In practice, there is an additional constraint that l' should approximate λ. This is imposed by some optimization criterion o, the exact form of which expresses assumptions about λ. For instance, o could express that nodes that are directly connected to each other tend to have similar labels by stating that the number of $\{v_1, v_2\}$ edges where $l'(v_1) \neq l'(v_2)$ should be minimal. The assumptions made about λ are called the bias of the Transductive learner.

In the Inductive approach, the task is to learn a function $f: D \rightarrow L$ that maps a node description d(v) to its label l(v). That is, given $G = (V, E, d, l)$, we need to construct $f: D \rightarrow L$ such that $f(d(v)) = l(v)$ when l(v) is defined and f is defined for all elements of D. Note that f differs from l in that it maps D, not V , onto L. This implies, for instance, that it can also make predictions for a node v that was not in the original network, as long as d(v) is known.

Besides the bias expressed by the optimization criterion o (which may still be present), there is now also a bias imposed by the choice of D: whenever two different nodes have the same description, they are assumed to have the same labels: $d(v_1) = d(v_2) \rightarrow \lambda(v_1) = \lambda(v_2)$. Additionally, the learning algorithm used to learn f has its own Inductive bias [4]: given exactly the same inputs, two different learning algorithms may learn different functions f, according to assumptions they make about the likely shape of f. Thus we have three types of biases. Transductive learners have a transductive bias, which is implied by the choice of the optimization criterion o. Inductive learners have a description bias, imposed by the choice of d, as well as an Inductive bias, imposed by the choice of the learning algorithm that is used to learn f from (d(v), l(v)) pairs.

Considering the biases (description and learning algorithm) of inductive approach in addition to difficulties of evaluating inductive methods in graph data, in this paper, we focus on Transductive approach and in particularly the Majority Rule method (MRM). This method and its limitations are discussed in the Section II. Section III explains precisely our proposed Augmented Random Walk (ARW) based approach to resolve main limitations of MRM. In Section IV, we compare ARW with MRM in six graph datasets. Section V concludes.

II. Majority Rule Method and Its Limitations

Among the Transductive node classifiers, MRM has a prominent role [6]. This method assigns to each node those labels that occur most frequently among its neighbors (typically a fixed number of labels is predicted). As an

example, Figure 1 shows a simple graph with four nodes V = $\{v_1, v_2, v_3, v_4\}$ and labeling set $L(v_i)$ for each node v_i. MRM predicts $\{l_1\}$ for node v_1 as it occurs most in the neighborhood of v_1.

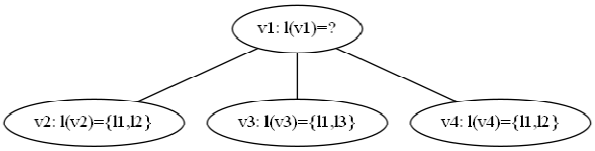

Fig. 1. Simple graph with node set V = $\{v_1, v_2, v_3, v_4\}$ and labeling set $l(v_i)$ for each node v_i. MRM predicts $\{l_1\}$ for node v_1 as it occurs most in the neighborhood of v_1.

However, this method suffers from several limitations. First, this method only considers the local neighborhood of the v_i ignoring the remaining information in the network. In Figure 2, Majority Rule method can not discriminate three labels l_1, l_2 and l_3 from each other since they all occur two times in the neighborhood of v_1. Although one might prioritize l_1 over $\{l_2, l_3\}$ by considering the labeling information of the second level neighboring nodes (node v_5 in Figure 2).

Second, MRM does not take into account the connectivity of the neighboring nodes in the prediction process. In Figure 3, independent from the existence of edge e_1, Majority Rule method can not discriminate three labels l_1, l_2 and l_3 from each other. However, one might give more priority to class label l_1 since it is more reachable to un-classified node v_1.

Third, MRM does not consider the confidence of class labels in the neighborhood of the un-classified nodes. In Figure 4, all three labels l_1, l_2 and l_3 occur two times in the local neighborhood of v_1 but one might consider l_3 as a class label with higher confidence since it occurs in nodes with a less number of class labels. In general, we assume that the confidence of node's class labels decreases as the number of its class labels increases.

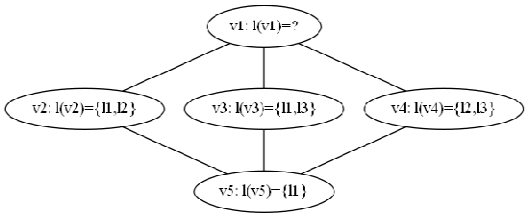

Fig. 2. MRM considers only the first neighborhood level and accordingly can not discriminate three labels l_1, l_2 and l_3 from each other.

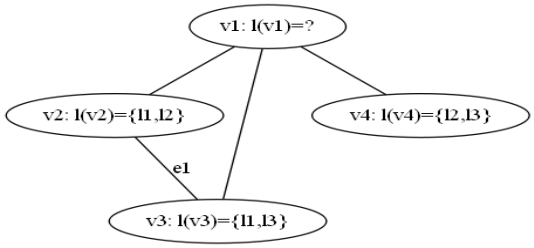

Fig. 3. MRM neglects the connectivity of neighboring nodes and accordingly can not discriminate three labels l_1, l_2 and l_3 from each other.

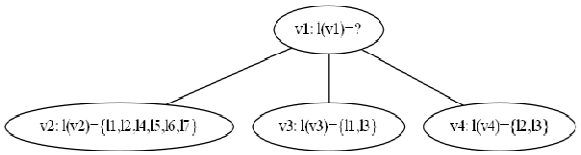

Fig. 4. MRM neglects the confidence of class labels and accordingly can not discriminate three labels l_1, l_2 and l_3 from each other.

III. AUGMENTED RANDOM WALK (ARW) METHOD

In this section, we propose a novel method to resolve the limitations of Majority Rule method discussed in Section II. There are two main steps in our proposed method. In the first step, we augment the initial graph by adding new nodes and edges to the initial graph and in the second step, we apply the Random Walk method for the node classification problem. These steps are discussed in the following sections.

A. Graph Augmentation Process

The initial graph G (which is formally described in Section I) is augmented as follows. If $L = \{l_1, l_2, \dots, l_n\}$ is the set of all the class labels in the initial graph then, (i) we add a new node l_i to G ($V = V \cup \{l_i\}$) for each $l_i \in L$ and (ii) we add a new edge $e_{ij} = \{v_j, l_i\}$ to G ($E = E \cup \{e_{ij}\}$) for each classified node v_j in which $l_i \in l(v_j)$. If each node v_j is annotated, on average, with k class labels, then the augmented graph G' will have $|V| + |L|$ nodes and, on average, $|E| + |V| * K$ edges. Figure 5 shows the augmented graph of Figure 1.

B. Applying Random Walk

To predict the class labels for each un-classified node $v_i \in G'$, we apply the steady state distribution of the RandomWalk with Restarts (RWR) technique [2] to calculate the graph similarity between v_i and each labeling node l_j . We simulate the trajectory of a random walker that starts from v_i and moves to its neighbors with uniform probability. We keep the random walker close to the original node v_i by allowing transition to the original node with probability r as the restart probability. Formally, the RWR technique can be represented by following formula:

$$x_{k+1} \leftarrow (1 - r) A x_k + c x_0 \qquad (1)$$

where x_k denotes the proximity vector at iteration t (i.e., a vector which contains the probability of reaching each node from v_i in k steps in the corresponding element). Therefore, x_0 is a vector with all elements being zero except the ith element which is one, and A is the adjacency matrix. This formula is used iteratively to generate the steady state RWR proximity vector (for more details see [2]).

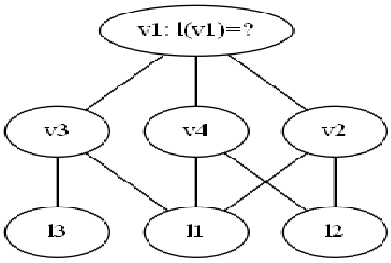

Fig. 5. Augmented graph G' of the Graph G shown in Figure 1. G is augmented with three nodes $\{l_1, l_2, l_3\}$ and six edges $\{\{v_2,l_1\}, \{v_2,l_2\}, \{v_3,l_1\}, \{v_3,l_3\}, \{v_4,l_1\}, \{v_4,l_2\}\}$.

IV. Empirical Results

We compare ARW with MRM in six well-known annotated graph datasets with respect to leave-one-out cross validation (LOOCV). LOOCV is the particular case of cross validation where a single node from the graph is considered as a validation data, and the remaining nodes as the training data. We repeat this process for each node in the graph data and we average the N (= number of the examined nodes) results to produce a single estimation. All the graph datasets are described and downloadable from http://linqs.cs.umd.edu/projects/projects/lbc/. Among the six datasets, Citeseer [8] and Cora [9] are two real-world bibliographic data sets in which publications are classified into six and seven class labels, respectively. The documents in the WebKB are webpages collected from computer science departments of four US universities (Cornell, Texas, Washington and Wisconsin) in 1997. Then, the webpages were manually classified into seven different classes. We apply no parameter tuning for our method and we choose 0.5 as the restart value. Figure 6 compares the MRM with our proposed ARW in six different graph datasets. ARW outperforms MRM by 4.66% with respect to the average Fmeasure calculated over six datasets with high homophily. Applying paired t-test results in p-value 0.02 which indicates that MRM and ARW methods are significantly different. However due to lack of enough samples this claim should be investigated in more details.

V. Conclusions and Future Work

In this paper, we addressed the problem of node classification in graph datasets. We discussed in detail the Majority Rule method as one of the most known Transductive node classifier in graph data. To resolve main limitations of the Majority Rule method, we proposed a new Random Walk based method which considers connectivity information of the neighboring nodes in addition to confidence values of their class labels in the prediction process. Without applying any parameter tuning and in 6 graph datasets with high homophily [3], our proposed method outperforms MRM by 4.66% with respect to the Fmeasure value calculated through leave-one-out cross validation process.

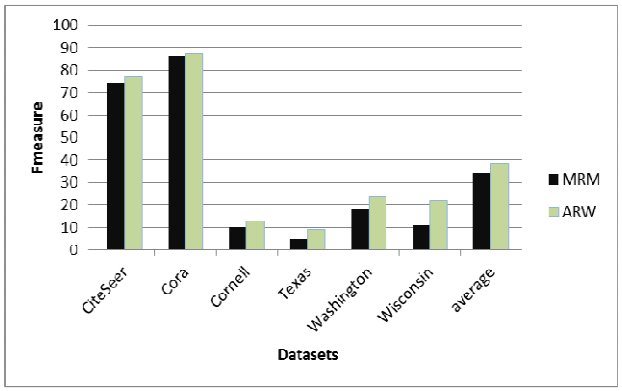

Fig. 6. Comparing MRM with ARW in six graph datasets. ARW outperforms MRM with respect to average Fmeasure calculated in LOOCV process.

Regarding future research induced by our work, we see four particularly important directions for refinement and extension of our approach. First, comparing our proposed method with MRM in more varied graph datasets. The variations could be the number of nodes, number of edges, degree distribution, number of class labels for each classified node and etc. Second, tuning the parameters to achieve more accurate prediction result. In the current implementation, we assume r = 0.5 and no weight for the graph edges. However, we could examine Random Walk method with different r values and choose the one which leads to best prediction result. The value of r could be even variable for different nodes in the same graph. For example, if there is enough information in the local neighborhood of un-classified node $v_i \in G$ (with respect to number of the classified neighboring nodes, their connectivity and their class labels), then Random Walker could explore more in the local neighborhood of v_i (high r values) while in other cases with limited information in the neighborhood, Random Walker could explore more widespread neighboring nodes (small r values). Third, investigating the robustness of our proposed method with respect to noisy edges (False Positive and False Negative edges) in the graph dataset. A way to do this is to randomly add or remove some portion of the edges in the graph and then compare the robustness of the MRM and ARW methods in the generated noisy graphs. Fourth, using the augmented graph for cluster analysis [5] of the class labels. The resulting clusters will provide more insights about the relationships among the class labels and could be used in multi-label classification [7].

References

[1] S. Bhagat, G. Cormode, and S. Muthukrishnan, "Node classification in social networks", CoRR, abs/1101.3291, 2011.

[2] L. Lovász, "Random walks on graphs: A survey, " in: Bolyai Society Mathematical Studies, 2, in: Combinatorics, Pál Erd˝ os is Eighty, vol. 2, Bolyai Mathematical Society, 1996, pp. 353–397.

[3] M. McPherson, L. Smith-Lovin, and J. M Cook, " Birds of a feather: Homophily in social networks", Annual Review of Sociology, 27(1):415–444, 2001.

[4] T. M. Mitchell, "Machine Learning", McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.

[5] S. E. Schaeffer, "Graph clustering", Computer Science Review, 1(1):27 – 64, 2007.

[6] B. Schwikowski, P. Uetz, and S. Fields, "A network of protein-protein interactions in yeast", Nat Biotechnol, 18(12):1257–1261, December 2000.

[7] M. Zhang and Z. Zhou, "A review on multi-label learning algorithms". Knowledge and Data Engineering, IEEE

[8] C. L. Giles, K. Bollacker, and S. Lawrence, "CiteSeer: An Automatic Citation Indexing System". In Digital Libraries 98: The Third ACM Conference on Digital Libraries. New York: Association for Computing Machinery.

[9] A. K. McCallum, K. Nigam, J. Rennie and K. Seymore, "Automating the Construction of Internet Portals with Machine Learning". Information Retrieval Journal 3(2): 127–163.