Rare Class Detection in Networks

Karthik Subbian* Charu C. Aggarwal† Jaideep Srivastava‡ Vipin Kumar*

Abstract

The problem of node classification in networks is an important one in a wide variety of social networking domains. In many real applications such as product recommendations, the class of interest may be very rare. In such scenarios, it is often very difficult to learn the most relevant node classification characteristics, both because of the paucity of training data, and because of poor connectivity among rare class nodes in the network structure. Node classification methods crucially dependent upon structural homophily, and a lack of connectivity among rare class nodes can create significant challenges. However, many such social networks are content-rich, and the content-rich nature of such networks can be leveraged to compensate for the lack of structural connectivity among rare class nodes. While content-centric and semi-supervised methods have been used earlier in the context of paucity of labeled data, the rare class scenario has not been investigated in this context. In fact, we are not aware of any known classification method which is tailored towards rare class detection in networks. This paper will present a spectral approach for rare-class detection, which uses a distance-preserving transform, in order to combine the structural information in the network with the available content. We will show the advantage of this approach over traditional methods for collective classification.

1 Introduction

The potential of collective classification to identify interesting entities in social networks is now well known, and is utilized widely for product and entity recommendations. A wide variety of methods for collective classification have been proposed in the literature in recent years [3, 5, 6, 8, 10, 11, 15, 17]. Detailed surveys on the topic may be found in [1, 2].

In many scenarios, the class of interest may be very rare. This is quite often the case in many social networks. Some examples of such scenarios are as follows:

- When the goal of the collective classification problem is to determine relevant product adoptions in a social network, very few nodes may be relevant to a particular product.
- In some scenarios, the goal of the collective classification method may be to identify nodes which are under some kind of terror threat. In such cases, the class of interest may be rare.
- The collective classification method may sometimes be used to identify adversarial entities (eg. terrorists) in the network. The presence of such entities may be rare, and the number of labeled examples may be even fewer. Our experimental section actually uses one such data set.

The rare class detection problem has also been studied widely in the context of relational data [7]. Two major challenges of this problem are the difficulty in obtaining sufficient labeled data belonging to the rare class, and the cost-sensitivity of the classification process. In the *network context*, an additional problem is the poor connectivity of rare class nodes. Node classification methods are, after all, based on structural homophily. Despite the obvious importance of the rare class detection problem in networks, and the unique challenges in the network context, we are not aware of any known techniques for this problem.

Conventional network classification often *cannot be used effectively for the problem of rare class detection*, because the effectiveness of label propagation methods reduces with increasing distance from the nodes. Furthermore, the effectiveness of such methods is based on their "*collective*" nature in which consistent labels at the neighbors of a node are used in order to infer labels. However, in the rare class detection problem, this is much more difficult because a given node may not often have neighbors which belong to the same rare class. Therefore, more statistical evidence is needed before a given node may be deemed to belong to a rare class.

One helpful characteristic of social networks is the presence of a modest amount of content at the nodes. The goal is to use this content in order to assist the determination of the class labels of the unlabeled nodes. Unfortunately, since most of the nodes are unlabeled in the first place, the content does not necessarily provide much information in the form of supervision. Nevertheless, the content can be used in order to perform *co-training*, in which the associations between different kinds of links and features are learned in conjunction with the clustering structure of the network. This is helpful for inferring the labels of nodes, because it provides information about the structural and content similarity

[*] University of Minnesota, Minneapolis, MN 55455. {karthik,kumar}@cs.umn.edu

[†] IBM TJ Watson Research, NY, 10598. charu@us.ibm.com

[‡] Qatar Computing Research Institute, Qatar. jsrivastava@qf.org.qa

of unlabeled nodes to the rare class, even when they do not *directly* contain the same content as rare class class nodes, or are not directly connected to a rare class node. This is because of the fact that content and structural correlations in unlabeled nodes can be used to make inferences about the features which are more relevant to the rare classes. For example, if the word "*golf* " is frequently present in labeled nodes of the rare category, then the co-occurrence of the word "*golf*" with other words or links in unlabeled nodes is useful information which should be leveraged in the classification model. Similarly, if the word "*fore*" often co-occurs with "*golf*" (not necessarily in rare class nodes), or unlabeled nodes containing "*fore*" are often linked to by nodes belonging to the rare class, then this information can be used in order to expand the learning information for the rare class. This broader principle is referred to as co-training. As we will see later, a partially supervised spectral clustering approach is an effective way to achieve this goal by embedding the nodes in a new space in which such implicit co-training information is encoded in the form of distances. The rare-class categorization can then be performed in the newly embedded space. It should be pointed out that some network classification methods do use content [7, 6], but not in way which is specifically helpful for rare class detection. In our experimental results, we will show the advantages of using such a focused approach for rare class detection.

1.1 Related Work and Contributions The problem of node classification has been studied in the graph mining literature [11], and especially for relational data in the context of *label or belief propagation* [13, 16, 17]. Such propagation techniques are also used as a tool for semi-supervised learning with both labeled and unlabeled examples [19]. A technique has been proposed in [10], which uses link-based similarity for node-classification. Recently, this technique has also been used in the context of blogs [3].

It has also been shown in [8, 6, 15] that the use of a combination of structure and content during categorization improves the classification accuracy of web pages. There are other semi-supervised statistical relational learning methods that have been proposed to predict unknown labels using a network structure [7, 25, 10]. In most of these methods the class imbalance is not taken into account, and the required statistical evidence for classifying the unknown node from the known neighboring labels, especially for the rare class, is not available sufficiently. The task of cost-sensitive classification is well-known in machine learning and there are plenty of related work [2]. However, there are only a few relational learning methods that address the issues of cost-sensitivity and class imbalance [26]. However, these papers ignore the the content information in each node $\mathcal{D}$, and hence completely missing the aspect of co-training used in our work. For interested readers, many of these methods are discussed in more detail in [11, 21, 1].

Since our approach also uses content, it is possible to use text classifiers such as SVM, particularly when the structure is not available or does not provide much information. The problem of text classification [9, 12, 14] has been studied widely in the information retrieval literature. Detailed surveys may be found in [14]. However, the network structure is completely ignored in most of these related work.

In this paper, we design a method for rare class detection, which uses a combination of structure and content to improve the ability to classify rare classes. We propose an optimization approach to combine several spectral embeddings for such classification. The method is cost-sensitive, and allows the user to specify appropriate costs in order to differentiate between the classification of the rare class and normal class.

2 Rare Class Learning Model

In this section, we will introduce the rare class learning model. The network is denoted by $G = (N, A)$ with node set N and edge set A. A *very small* subset of the nodes $R \subset N$ of the nodes are labeled with the rare class, and another subset $S \subset N$ of the nodes are labeled with the normal class. The remaining nodes $N \backslash (R \cup S)$ are unlabeled. The total number of nodes in N is denoted by n. It is assumed that each node i is associated with content D_i. This content D_i is assumed to be a set of keywords, and may therefore be treated conceptually as a text document. The entire content associated with all nodes is denoted by $\mathcal{D} = \{D_1 \dots D_n\}$.

The content at the nodes could be modeled in a wide variety of ways, depending upon the application-specific scenario. For example, in a social influence analysis scenario [23], the content at each node could correspond to items, that the actor at the node may be interested in. In order to model the labels, a particular item may be considered "special", and may be considered the rare class, which is desired to be found. Note that the presence or absence of this special item at a given node, may only be known for a small subset of the nodes. This implies that the vast majority of the nodes are unlabeled. The use of content and structure in this scenario for classification corresponds to the use of structural correlations between the items at the different nodes in order to predict these rare classes.

It is assumed that the vast majority of the nodes in the network are unlabeled. This formulation implicitly assumes binary labels, though the model can be easily generalized to the case where there are multiple classes, using one-vs-all or one-vs-one approach [2]. Rare class problems are typically posed from a *cost-sensitive perspective* in which misclassification of the rare class incurs cost c_r, whereas misclassification of the normal class incurs the cost c_n. Typically, it is assumed that $c_r >> c_n$. Formally, the cost-sensitive rare class detection problem, with heavily skewed class distribution, may be posed as follows:

Problem 1 (Network Rare Class Detection) *Given an undirected network $G = (N, A)$, with node content $\mathcal{D}$, a set of nodes R labeled with the rare class, a set of nodes S labeled with the normal class, misclassification costs for the rare and normal classes denoted by c_r and c_n respectively, classify the unlabeled nodes in the network, so as to minimize the total misclassification cost.*

One observation is that the number of nodes in the rare class is typically small, whereas the cost of misclassification is rather large. This implies that the smaller amount of training data is usually available for the more important of the two cases. In this context, the use of co-training can be very helpful.

2.1 Broad Overview of Approach

In this section, we will provide a broad overview of a partially supervised spectral clustering approach which is used for creating an embedded representation which can encode the relevant feature-specific information in the form of distances. Note that the use of spectral methods requires the creation of a similarity matrix which encodes a combination of structural and content information. It is important to do this in a semi-supervised way, so as to maximize the cost-sensitive classification accuracy of the rare class. The first step is to add synthetic edges to the network, with weight λ, which are designed to ensure that nodes belonging to rare classes are closer together in the embedded representation. The precise value of λ will be determined later in a semi-supervised way, so as to maximize the cost-sensitive accuracy. Note that such a choice of λ does not necessarily ensure that all the rare nodes cluster together in the embedded representation, but it will ensure that the clustering is performed in a way, so as to maximize the cost-sensitive accuracy. This results in an augmented graph $G = (N, A \cup A_r)$, where A_r is the new set of edges added between nodes belonging to the rare class. Though we have induced (ghost) edges between rare class nodes, the actual contribution of these edges also depends on the content information in these nodes. One may also argue that the induced ghost edges A_r may blow-up the number of edges in the adjacency matrix, however, this is not true, as the number of rare class nodes (especially the ones that are labeled) are much smaller than the number of nodes (n).

The first step is to define a similarity matrix between the nodes in the network. This is needed for the process of spectral analysis. We note that the similarity matrix encodes a significant amount of information using features that are not necessarily present in nodes which belong only to the rare class. Therefore, such an approach implicitly uses co-training by using the full similarity matrix in the analysis process. Therefore, for any pair of nodes $i, j \in N$, we need to define the similarity S_{ij} between i and j. The first step is define the content-based similarity between the node pairs. Let $\mathbf{v}_i$ and $\mathbf{v}_j$ be the frequency weighted vector-space representations of the documents D_i and D_j at the nodes i and j. Then, the content similarity C_{ij} between nodes i and j is defined as the cosine similarity between the documents D_i and D_j:

$$C_{ij} = cosine(\mathbf{v}_i, \mathbf{v}_j) = \frac{\mathbf{v}_i \cdot \mathbf{v}_j}{\|\mathbf{v}_i\| \cdot \|\mathbf{v}_j\|} \tag{2.1}$$

Note that the words chosen for content similarity matrix C_{ij} need to be more representative of the rare class, in order to increase the content-centric connectivity of nodes containing features relevant to the rare class. One way to achieve this, is to retain words that is seen at least once in a rare class labeled document. Therefore, the feature selection, at the very beginning of the algorithm, is also performed in order to maximize the advantages from co-training.

The structural similarity Q_{ij} between nodes i and j is defined on the basis of the edges between i and j, which occur on the basis of both structural and content-based similarity. Let I_{ij} be an indicator variable which takes on the value of 1, if (i, j) occurs in A and 0 otherwise. Similarly, let P_{ij} be an indicator variable which takes on the value of 1, if (i, j) occurs in A_r and 0 otherwise. Then, the structural similarity Q_{ij} is defined as follows:

$$Q_{ij} = I_{ij} + \lambda \cdot P_{ij} \tag{2.2}$$

Then, the total similarity S_{ij} is defined as a weighted combination of the structural similarity Q_{ij} and the content similarity C_{ij}.

$$S_{ij} = Q_{ij} + \nu \cdot C_{ij} \tag{2.3}$$

Here ν is a weighting parameter which decides the relative importance of structure and content in the learning process. The values of ν and λ will be learned later with the use of an iterative approach in order to maximize the cost-sensitive accuracy.

2.2 Creating the Embedding

In order to perform the classification, the first step is to set up the embedding. This is done with the use of the similarity matrix defined by S_{ij}. Therefore, we have a weight matrix $\mathbf{W}$, for which $W_{ij} = S_{ij}$. The diagonal entries of this matrix are set to zero, i.e $W_{ii} = 0$. However, we note that the similarity matrix is defined with the use of the parameters ν and λ, which are unknown in advance. Nevertheless, in order to provide a conceptually comprehensible exposition, the creation of the embedding will be discussed with fixed values of the parameters ν and λ.

The most popular and well-known k-dimensional embedding of the similarity matrix can be done using *Laplacian Eigenmaps*. We refer the readers to well-known texts on this topic [24]. The k-dimensional spectral embedding of $\mathbf{W}$ that minimizes the L_2 norm between the data points (i.e. nodes in our case), can be obtained by computing the leading k Eigen vectors of the (unnormalized) Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{W}$, with necessary scalability constraints on the embedding. Formally, the solution to the following problem (2.4) will provide us the necessary spectral embedding.

$$\begin{aligned} & \min_{\bar{\mathbf{Y}}} && \mathrm{Tr}(\bar{\mathbf{Y}}^T \mathbf{L} \bar{\mathbf{Y}}) \\ & s.t. && \bar{\mathbf{Y}}^T \mathbf{D} \bar{\mathbf{Y}} = \mathbb{I}_k \end{aligned} \tag{2.4}$$

The $\bar{\mathbf{Y}}$ matrix is the k-dimensional spectral embedding of the data points, that are obtained by solving the generalized Eigenvalue problem, $\mathbf{L}\bar{\mathbf{Y}} = \mathbf{D}\mathbf{\Lambda}\bar{\mathbf{Y}}$, which are the k eigenvectors corresponding to the smallest k eigenvalues (ignoring the trivial eigenvalue of 0). Here $\mathbf{\Lambda}$ is diagonal matrix containing the k eigenvalues along the diagonal and $\mathbb{I}_k$ is $k \times k$ identity matrix.

The formulation (2.4) can be further rewritten, in a more general form, as (2.5). We use $\mathbf{L} = \sum_i \alpha_i \left(\mathbf{D}_i - \mathbf{W}_i\right)$ and $\mathbf{W}$ as a linear combination of s different weight matrices $\mathbf{W}_1, \ldots, \mathbf{W}_s$ with corresponding weights $\alpha_1, \ldots, \alpha_s$. The formulation (2.5) clearly shows that our approach considers a linear combination of multiple spectral embeddings $\mathbf{L}_1, \ldots, \mathbf{L}_s$ in order to maximize the with-in cluster and minimize the between-cluster similarity in the overall combined matrix.

$$\begin{aligned} & \max_{\bar{\mathbf{Y}}, \alpha} && \sum_{i=1}^{s} \alpha_i \, \mathrm{Tr}(\bar{\mathbf{Y}}^T \mathbf{W}_i \bar{\mathbf{Y}}) \\ & s.t. && \sum_{i=1}^{s} \alpha_i \bar{\mathbf{Y}}^T \mathbf{D}_i \bar{\mathbf{Y}} = \mathbb{I}_k \end{aligned} \tag{2.5}$$

It should be pointed out that the approach implicitly incorporates supervision during the embedding, because of the impact of the variables α_i, which are used to define the combined Laplacian. As we combine only network and content structure in this paper, α_1 and α_2 corresponds to λ and ν respectively. This framework, however, is more general and can be used for classification problems, where more than one network is needed for supervision. Of course, it has not yet been described, how these variables λ and ν are actually determined. This will be discussed in a later section.

2.3 Illustrative Example We illustrate the notion of combinations of spectral embedding using a simple 1-d spectral embedding example. Consider the content and network structure in Figure 1, the thickness of the edges is proportional to the edge weight.

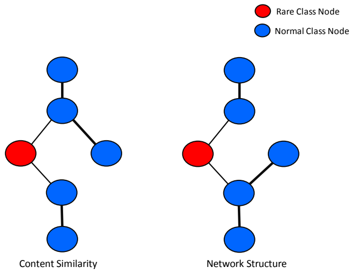
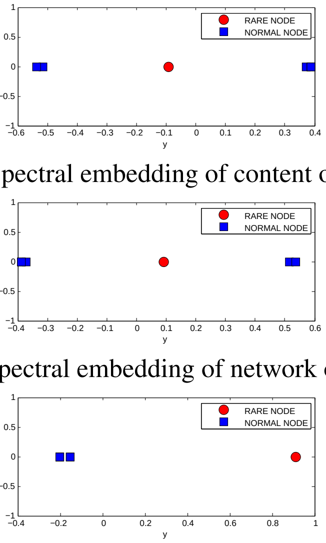


Figure 1: An illustrative example of content and network structure.



(a) Spectral embedding of content only.

(b) Spectral embedding of network only.

(c) Spectral embedding of content and network.

Figure 2: Spectral embeddings of content, network and its combination.

The corresponding 1-d spectral embedding for the content and network structure is shown in Figure 2(a) and (b) respectively. The spectral projection of the nodes are shown along the X-axis. In both the network- and content-only embedding no classifier can correctly classify the rare from the normal class, as there are normal nodes on both sides of the rare node. When the networks are combined with equal weights, the resulting combined embedding clearly improves the classification accuracy and rare class is well separated from the normal class along the real line (X-axis). This is shown in Figure 2(c). In the combined embedding, multiple normal class nodes are overlapping on each other and hence only two of them are visible.

2.4 Classification Approach The multi-dimensional embedding provides an effective way to perform the classification. Since the representation of each node is in multi-dimensional form, in which the implicit relationships between the (unsupervised portion of the) feature space are well represented, it can be used to perform classification effectively. The key here is that the value of λ and ν are not known a-priori. However, we will describe a classification model under the assumption that these values are known. Later, we will describe how to find these parameters so as to maximize the cost-sensitive accuracy.

In our particular case, we use a weighted k-nearest neighbor classifier. For the test node, we find the k nearest neighbors based on the euclidian distances on the embedded representation. Let us assume that the number of nodes belonging to the rare class is k_r and the number of nodes belonging to the normal class is k_n, such that $k_r + k_n = k$. The rare class is reported as the relevant class if $c_r \cdot k_r > c_n \cdot k_n$, and the normal class otherwise.

3 Learning Cost-Sensitive Variables for Embedding

The multidimensional embedding created in the previous case can be converted into a classification model. The key here is that the model is dependent on the parameters λ and ν which define the cost sensitive accuracy. Therefore, the effectiveness of the training model for a particular value of λ and ν needs to be evaluated. For this purpose, a cross-validation approach is used. The training data is divided into q segments. For each value of the parameters ν and λ, the model accuracy is evaluated by training on $q - 1$ segments and testing on the remaining one. The value of q is typically picked to be a small number such as 2 or 3.

The first step is to define the misclassification cost $\mathcal{M}$ as follows:

$$\mathcal{M}(A, \mathbf{C}, G_t, \lambda, \nu) = c_r \cdot m_r + c_n \cdot m_n \tag{3.6}$$

Here m_r and m_n represents the number of nodes misclassified for the rare and normal class respectively. Using the ground truth class labels G_t and the classifier described earlier, classify the points using an optimal combination of network structure A and content similarity matrix $\mathbf{C}$. The value of ν and λ used for the combination are computed with the use of the cross-validation approach. In order to determine the optimal classification accuracy, the algorithm uses an iterative search approach. At the starting point, the value of ν and λ are both set to 0. An arbitrary upper bound is also set for these values at 10^3. This provides the initial range in which to perform the binary search. The values of λ and ν are then alternately adjusted with the use of binary search. In the first step, the optimal value of $\nu = \nu_1$ is computed using binary search, while keeping the value of λ fixed to its initial value of 0. When the locally optimal value of $\nu = \nu_1$ is determined, the optimal value of $\lambda = \lambda_1$ is determined with binary search, while fixing the value of $\nu = \nu_1$. In the kth iteration, the value of $\nu = \nu_k$ is determined by fixing $\lambda = \lambda_{k-1}$, and the optimal value of $\lambda = \lambda_k$ is determined by fixing $\nu = \nu_k$. In each iteration, the difference in the value of ν_k from the last iteration is determined. When the difference between the values in a pair of iterations is less than a pre-defined threshold, it is desired to terminate. The outline of the **RareNet** algorithm is listed in Figure 3.

```
Algorithm RareNet(Network Adjacency List: $A$,
        Content similarity matrix: $C$,
        Ground truth labels: $G_t$)
begin
  Initialize $t = 1$, $\lambda_1 = 10^{-6}$, $\nu_1 = 10^{-6}$;
  repeat
    $\nu_{t+1} = \text{argmin}_\nu\, M(A, C, G_t, \lambda_t, \nu)$;
    $\lambda_{t+1} = \text{argmin}_\lambda\, M(A, C, G_t, \lambda, \nu_{t+1})$;
    $t = t + 1$;
  until($\lambda_t$ and $\nu_t$ not converged);
  Construct $\mathbf{W}$ and $\mathbf{D}$ using $A$, $A_r$, $\mathbf{C}$, $\lambda_t$ and $\nu_t$;
  Compute $\mathbf{L} = \mathbf{D} - \mathbf{W}$;
  $\overline{\mathbf{Y}}$ = Leading $k$ Eigenvectors of $\mathbf{L}$;
  $\mathcal{C}_n$ = $k$-means clusters on normal class in $\overline{\mathbf{Y}}$;
  $\mathcal{C}_r$ = $k$-means clusters on rare class in $\overline{\mathbf{Y}}$;
  for each unknown label node
    Classify using the $k$ nearest neighbor classifier
      on centroids of ($\mathcal{C}_n$) and ($\mathcal{C}_r$);
  endfor
end
```

Figure 3: *RareNet* Outline

4 Experimental Results

In this section, we will present experimental results illustrating the effectiveness of the RareNet algorithm compared to several well established baselines. We will study the effectiveness in terms of rare class accuracy and overall misclassification cost.

4.1 Data sets We used two real-life data sets. One is the *Database List of Publications (DBLP)*, that deals with the co-authorship network, while the other is the terror attack (PIT) data set that describes co-located terror attacks.

DBLP Data Set: We downloaded the publicly available *DBLP* data set [1], and extracted the abstract, venue and author details for each of the published document. Furthermore, we created the ground truth labels for each of the authors, assigning them to one of the 22 areas of computer science as grouped in $academic.research.\ microsoft.com$, using the list of top 10 conferences in each area. We then created a co-authorship network with dominant class as *Data Mining* (DM), and the rare class nodes as *Privacy and Security* (PS) authors. To this end, we considered all published papers that had at least one DM author, and we removed authors from

[1] http://arnetminer.org/citation

other communities except DM and PS. Our final network had 6973 author nodes, which includes 180 authors from PS area, 6793 authors from DM area, and 36614 co-authorship edges. For every author node i, we constructed a word-feature vector $\mathbf{v}_i$ based on all the abstracts of the papers included in our processing. The j-th element of this vector $\mathbf{v}_i$ contains the *tf-idf* score of the j-th word from our dictionary. We removed stop words, stripped off punctuations and stemmed the words while constructing our dictionary. The size of resulting dictionary was 4486 words.

PIT Data set: The *Profiles In Terror (PIT)* data set[2] contains geographically co-located terror attacks, attributes related to each terror attack, and its category. We extracted a co-located terror attack network with *Bombing* as the dominant class and *Arson* as the rare class. Each node in this network corresponds to a terror attack and an edge denotes if they are co-located. The number of nodes in the network was 560 including 31 nodes that correspond to Arson (rare) category. There were 2850 edges in the network. Each node in this data set contains 106 binary attributes corresponding to the attack, which we treat as the content (or attribute) vector $\mathbf{v}_i$ for the corresponding node i.

4.2 Baselines We used different baseline classifiers that considers only content, only network and both content and network structure for classification. Also, our choice of baseline includes both cost-sensitive and insensitive categories. In addition, they represent three popular classes of literatures corresponding to, low dimensional embedding, max-margin and label propagation classifiers. The following is the list of baselines we used in our evaluation.

- **Spectral-NN:** This is a low dimensional embedding classifier, where the network structure using A and A_r undergoes a spectral embedding, followed by a K-nn classifier. This classifier does not use the content information. It uses *only the network structure* available in the training data. We computed the 10 leading eigenvectors for computing the spectral embedding and the value of K was set to 3 in the K-nn algorithm.

- **SVM:** We used the cost-sensitive Support Vector Machine (SVM) classifier to classify the nodes to rare or dominant classes, based only on its content information. In this baseline, we used *only the content information* and excluded the network structure. The content information for node i is the vector $\mathbf{v}_i$, as described in the data set.

- **Iterative Classifier:** We use an iterative classification technique for relational data [7], to classify a node using its immediate neighborhood information. The local classifier used at each node was Naive Bayes with a cost sensitive prior that takes in to account the number of samples in each class along with its cost. When the cost for a class is set higher the prior increases proportionally. This baseline explicitly uses *both the content and the network structure* and is cost sensitive.

4.3 Evaluation Measures We use two important evaluation measures to examine the effectiveness of the classifiers: (1) *Recall* and (2) *Cost sensitive error rate*. The *Recall* measures the fraction of correctly classified rare class nodes out of the total number of rare class nodes. Let the number of test examples for normal class be n_n and those for rare class be n_r. Let m_n and m_r be the number of misclassified normal and rare class nodes respectively. Then, the recall ($\mathcal{R}$) for rare class is defined as follows:

$$\mathcal{R} = 1 - \frac{m_r}{n_r} \tag{4.7}$$

The recall cannot fully capture the effectiveness, because an arbitrary classifier could assign all nodes to the rare class, and do well in terms of recall. Therefore, we also used a cost sensitive error rate (f), which measures the error rate in a cost sensitive way. Formally, let c_n and c_r be the cost of misclassification for normal and rare class respectively. Then, the cost sensitive error rate, f is defined as follows:

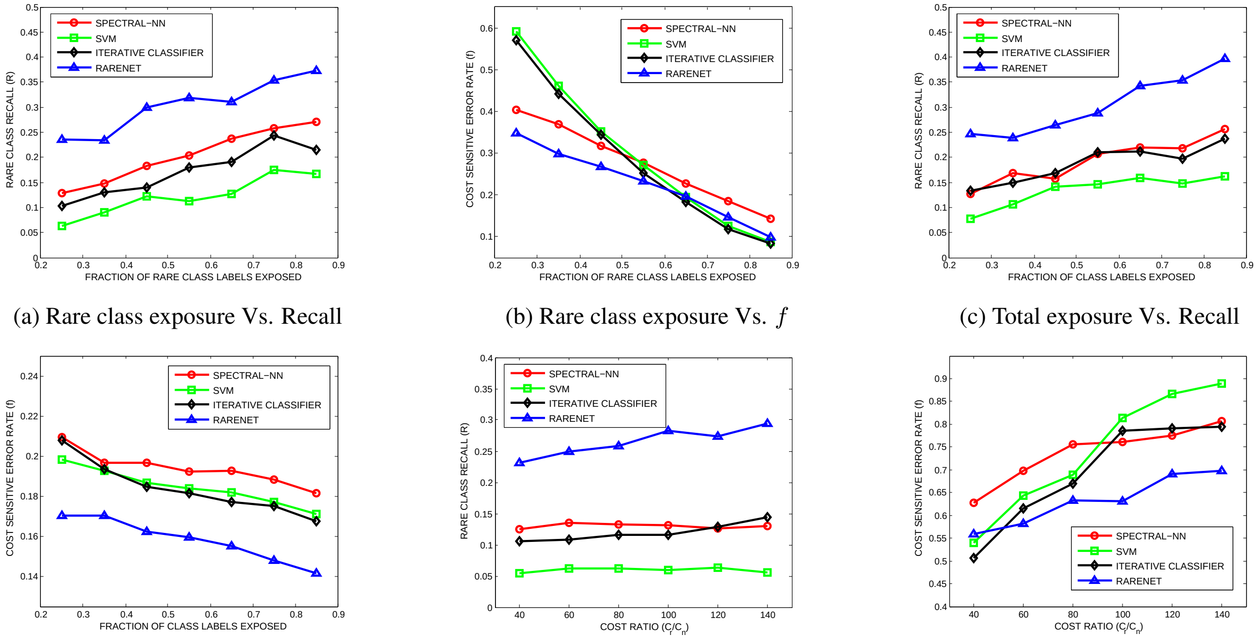
$$f = \frac{c_r m_r + c_n m_n}{c_r n_r + c_n n_n} \tag{4.8}$$

When the costs are equal ($c_n = c_r$), then the value of f is equal to the *error rate*. When the classifier classifies every test example correctly, then $f = 0$. On the other hand, when all test examples are misclassified, then $f = 1$.

4.4 Effectiveness Results The evaluation measures were computed in terms of three control parameters, by varying the amount of training data provided in terms of normal and rare classes, and also varying the cost ratio of the rare class to the normal class. By varying the amount of training data, we are also able to show the impact of *availability* of training data. This is particularly important in the rare class scenario, where rare class examples are hard to obtain. The results of the variations in these parameters are shown in Figure 4 and 5 for *DBLP* and *PIT* data sets respectively.

In Figure 4(a), we gradually increased the fraction of rare class labels from 0.25 to 0.85 for the *DBLP* data set, while keeping the misclassification costs fixed at $c_r = 10$ and $c_n = 1$. We also set the fraction of normal class labels exposed to 0.75. We observe that the rare class recall of all classifiers gradually increased as a greater number of rare class labels were exposed. Our approach (RareNet) outperforms the best performing baseline consistently by up to 10% over all exposure levels. The spectral-NN baseline performs the best in this experiment, because the co-authorship

[2] `http://linqs.cs.umd.edu/projects/projects/lbc/`



(a) Rare class exposure Vs. Recall

(b) Rare class exposure Vs. f

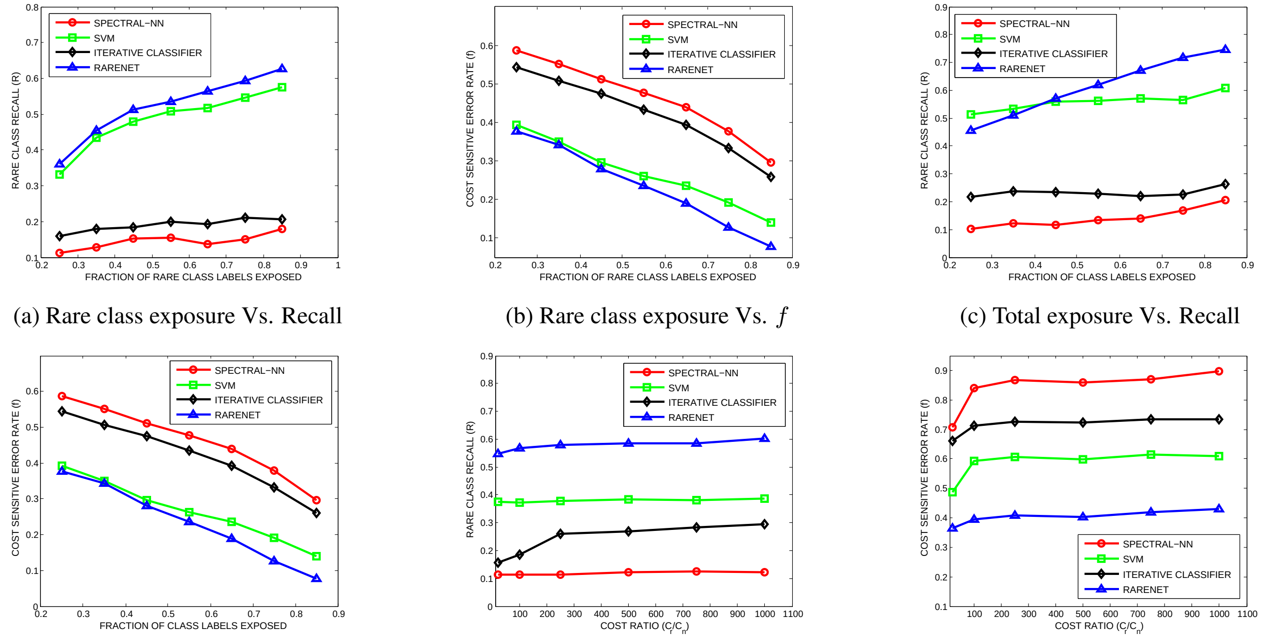
(c) Total exposure Vs. Recall

(d) Total exposure Vs. f

(e) Cost ratio Vs. Recall

(f) Cost ratio Vs. f

Figure 4: Experimental Results for *DBLP* data set



(a) Rare class exposure Vs. Recall

(b) Rare class exposure Vs. f

(c) Total exposure Vs. Recall

(d) Total exposure Vs. f

(e) Cost ratio Vs. Recall

(f) Cost ratio Vs. f

Figure 5: Experimental Results for *PIT* data set

network contains significant information about class associations. The content-only classification using SVM performs quite poorly. The iterative classifier combining the structure and content performs midway between the two baselines.

In the *PIT* data set, increasing the rare class exposure does increase the rare class recall, as shown in Figure 5(a). However, the rate of increase was very different for each baseline. The content-only *SVM* classifier performs extremely well in this data set, as opposed to *DBLP*. Thus, the other baselines performed *inconsistently* across data sets, because of varying impact of content, structure and rare class connectivity. The key observation here is that our approach is able to pick the right combination of network and structure in order to perform the classification. The plots of Figures 4(a) and 5(a) clearly shows, that our approach *RareNet* combines the network and content information appropriately to consistently outperform both content-only and network-only classifier. Another important point to note is that the iterative classifier does moderately improve either the content-only (in DBLP) or network-only classifiers (in *PIT*). This is primarily due to the lack of statistically significant number of rare class nodes in the immediate neighborhood. Our approach overcomes this problem by looking for neighbors in the space formed by the Eigenvectors, and does not necessarily rely on a single channel of information.

The cost sensitive error rate variation with increasing number of exposed rare class labels for the *DBLP* and *PIT* data sets are illustrated in Figures 4(b) and 5(b) respectively. As in the previous case, the order of baselines and the rate of decrease vary with data set. In Figure 4(b) (*DBLP* data set), the iterative and SVM classifiers have lower error rate compared to the *RareNet* algorithm after exposing 65% of the labels, even though *RareNet* is still superior in terms of the recall. Therefore, on an *overall* basis, the *RareNet* approach is much more robust, and varies far less with choice of data set and performance measure. Interestingly, the spectral-NN method performs well in terms of recall, but the overall cost sensitive error of the classifier is poor. As the number of rare class examples are increased, larger number of normal class examples are classified as rare by the spectral-NN method. This problem is not faced by the *RareNet* approach, which is able to redefine locality with both structure and content.

We also varied the fraction of labels exposed in *both* rare and normal class equally. The corresponding performance of the classifiers are illustrated in Figures 4(c) and 5(c) respectively. While, the results have a similar broad trend to that obtained by varying only the rare class labels, the performance of the iterative classifier has improved in this measure. SVM performs poorly in *DBLP* and well in *PIT* data set. In both data sets, our method performs consistently well compared to all baselines. This again underlines the wide variations across different baselines over different data sets and validity measures, whereas *RareNet* is the only baseline, which achieved a high level of consistency. This kind of consistency is important, when using a particular classifier for an arbitrary scenario.

We also evaluated the performance of all methods by varying the ratio of costs $\frac{c_r}{c_n}$. The *RareNet* and *Iterative Classifier* are sensitive to cost. It is evident that the recall increases, as the ratio of cost increases, in Figures 4(e) and 5(e). This is not surprising, because an increase in the cost increases the importance of classifying the rare class accurately. Among the baselines, SVM performs the best in the *PIT* data set, and spectral-NN in terms of rare class recall. Despite the variations in data sets, our method combines the content and network structure optimally and consistently gives good performance in terms of recall and f values in both data sets.

The cost-sensitive error rate levels off, as the classifiers cannot improve the rare class accuracy, despite increasing costs, beyond a certain point. This is clearly illustrated in Figure 5(f), where increasing the cost ratio beyond 300 does not improve the error rate. Most of the increase occurs in the earlier part of the plot. In order to illustrate this more clearly, we show the variations over a smaller range in Figure 4(f) for the *DBLP* data set. As in all other cases, our method consistently outperforms all other baselines.

4.5 Efficiency Results We evaluated the efficiency of our method, using the running time in seconds. Table 1 shows the running time for both data sets. Unlike other classification settings, the network setting is one, where it is difficult to separate the training time clearly from the testing. For example, in the iterative classifier, the training and testing is performed in parallel using an iterative approach. Therefore, the overall running times for classifying all test instances is reported for the different classifiers. All the running times reported are in seconds, and averaged over 10 runs, with the rare and normal class exposure set to 0.25 and 0.75 respectively.

Most of the running time for spectral methods is spent in the Eigen decomposition. This computation is obviously expensive, though the running time is not very different for smaller data sets such as *PIT*. In large data sets, other baselines such as *SVM* also consume significant time to solve the underlying Quadratic Program. We used the interior point method to solve the underlying QP in all our experiments. The running time for the iterative classifier depends significantly on the number of examples exposed. which determines the iterations needed for convergence. Overall, our method is quite comparable with iterative classifier and SVM baselines in terms of runtime, while outperforming these baselines in terms of recall and error rate measures in both data sets. Therefore, our approach provides the best overall performance in terms of effectiveness and efficiency.

Method	DBLP	PIT
Spectral-NN	35.79	9.18
SVM	1446.07	4.62
Iterative Classifier	301.66	8.70
RareNet	566.07	11.88

Table 1: Running Time (seconds) for various classifiers

5 Conclusions

In this paper, we explored the problem of rare class detection in networks. Rare class detection is much more difficult in networks because of the poor connectivity of the nodes and using the concept of homophily effectively. Combining the structure with the content in a careful way, helps in defining a spectral embedding, in which the locality of the rare class nodes are much more informative. This is reflected in the superior and consistent performance of our classifier, with respect to the baseline methods, including network methods which combine structure and content for standard classification scenarios.

Acknowledgements

Research was sponsored by the Army Research Laboratory agreement number W911NF-09-2-0053, Defense Advanced Research Project Agency agreement number W911NF-12-C-0028 and IBM Ph.D. fellowship program. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory, DARPA or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

[1] C. C. Aggarwal, *Social Network Data Analytics*, Springer, (2011).

[2] C. C. Aggarwal. *Data Classification: Algorithms and Applications*, CRC Press, (2014).

[3] S. Bhagat, G. Cormode, and I. Rozenbaum, *Applying link-based classification to label blogs*, WebKDD/ SNA-KDD, (2007), pp. 97–117.

[4] X. Zhu, Z. Ghahramani, *Learning from labeled and unlabeled data with label propagation*, Technical Report CMU-CALD-02-107, Carnegie Mellon University, (2002).

[5] M. Bilgic and L. Getoor, *Effective label acquisition for collective classification*, KDD, (2008), pp. 43–51.

[6] S. Chakrabarti, B. Dom, and P. Indyk, *Enhanced hypertext categorization using hyperlinks*, SIGMOD Conference, (1998), pp. 307–318.

[7] J. Neville, D. Jensen, *Iterative classification in relational data*, AAAI Workshop on Learning Statistical Models from Relational Data, (2000), pp. 13–20.

[8] V. R. de Carvalho and W. W. Cohen, *On the collective classification of email "speech acts"*, SIGIR Conference, (2005), pp. 345–352.

[9] T. Joachims, *Text Categorization with Support Vector Machines: Learning with Many Relevant Features*, ECML Conference, (1998), pp. 137–142.

[10] Q. Lu and L. Getoor, *Link-based classification*, ICML Conference, (2003), pp. 496–503.

[11] S. A. Macskassy, and F. Provost, *Classification in Networked Data: A Toolkit and a Univariate Case Study*, I Journal of Machine Learning Research, Vol. 8, (2007), pp. 935–983.

[12] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell, *Text classification from labeled and unlabeled documents using EM*, Machine Learning, Vol. 39(2–3), (2000), pp. 103–134.

[13] B. Taskar, P. Abbeel, and D. Koller, *Discriminative probabilistic models for relational data*, UAI, (2002), pp. 485–492.

[14] Y. Yang, *An evaluation of statistical approaches to text categorization*, Information Retrieval, Vol. 1(1-2), (1999), pp. 69–90.

[15] T. Zhang, A. Popescul, and B. Dom, *Linear prediction models with graph regularization for web-page categorization*, KDD Conference, (2006), pp. 821–826.

[16] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, *Learning with local and global consistency*, Advances in Neural Information Processing Systems, Vol. 16, (2004), pp. 321328.

[17] D. Zhou, J. Huang, and B. Schölkopf, *Learning from labeled and unlabeled data on a directed graph*, ICML Conference, (2005), pp. 1036–1043.

[18] Y. Zhou, H. Cheng, and J. X. Yu, *Graph clustering based on structural/attribute similarities*, PVLDB, Vol. 2(1), (2009), pp. 718–729.

[19] X. Zhu, Z. Ghahramani, and J. D. Lafferty, *Semi-supervised learning using gaussian fields and harmonic functions* ICML Conference, (2003), pp. 912–919.

[20] N. L. D. Khoa, and S. Chawla, *Large Scale Spectral Clustering Using Resistance Distance and Spielman-Teng Solvers*, Discovery Science, (2012), pp. 7-21.

[21] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad. *Collective classification in network data* AI magazine 29, no. 3 (2008), pp. 93-106.

[22] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. *Spectral grouping using the Nystrom method*, IEEE Trans. Pattern Anal. Mach. Intell., Vol. 26, (2004), pp. 214-225.

[23] K. Subbian, C. Aggarwal, and J. Srivastava. Content-centric flow mining for influence analysis in social streams, CIKM, 2013, pp. 841-846.

[24] M. Belkin, P. Niyogi. *Laplacian eigenmaps for dimensionality reduction and data representation*, Neural computation 15, no. 6 (2003), pp. 1373-1396.

[25] J. Neville, D. Jensen. *Collective classification with relational dependency networks*, In the Second International Workshop on Multi-Relational Data Mining (2003), pp. 77-91.

[26] P. Sen, L. Getoor. *Cost-sensitive learning with conditional Markov networks*, Data Mining and Knowledge Discovery 17, no. 2 (2008), pp. 136-163.