A Parameter-Free Label Propagation Algorithm Using Bipartite Heterogeneous Networks for Text Classification

Rafael G. Rossi
Institute of Mathematics and Computer Science - University of São Paulo
São Carlos, SP, Brazil
ragero@icmc.usp.br

Alneu A. Lopes
Institute of Mathematics and Computer Science - University of São Paulo
São Carlos, SP, Brazil
alneu@icmc.usp.br

Solange O. Rezende
Institute of Mathematics and Computer Science - University of São Paulo
São Carlos, SP, Brazil
solange@icmc.usp.br

ABSTRACT

A bipartite heterogeneous network is one of the simplest ways to represent a textual document collection. In such case, the network consists of two types of vertices, representing documents and terms, and links connecting terms to the documents. Transductive algorithms are usually applied to perform classification of networked objects. This type of classification is usually applied when few labeled examples are available, which may be worthwhile for practical situations. Nevertheless, for existing transductive algorithms users have to set several parameters that significantly affect the classification accuracy. In this paper, we propose a parameter-free algorithm for transductive classification of textual data, referred to as LPBHN (*Label Propagation using Bipartite Heterogeneous Networks*). LPBHN uses a bipartite heterogeneous network to perform the classification task. The proposed algorithm presents accuracy equivalent or higher than state-of-the-art algorithms for transductive classification in heterogeneous or homogeneous networks.

Categories and Subject Descriptors

G.2.2 [**Graph Theory**]: Graph Labeling; H.2.8 [**Database Applications**]: Data Mining; H.2.4 [**Systems**]: Textual Databases

General Terms

Algorithms, Experimentation

Keywords

Transductive Learning, Automatic Text Classification, Text Representation, Heterogeneous Networks

1. INTRODUCTION

Automatic classification of textual documents is one of the most important tasks to manage, retrieve, and extract knowledge from the huge amount of textual data available nowadays [4]. Usually this task is performed by using inductive learning methods, which induces a function/classification model that are used to classify unlabeled examples. Many labeled examples are required to generate an accurate model. However, to obtain a high number of labeled examples is costly and time consuming. Hence, the use of transductive learning has become worthwhile for real applications. Transductive learning does not create a model to classify new documents. Instead, transductive learning considers a data set of both labeled and unlabeled examples to perform the classification, spreading the class information from labeled to unlabeled data. Unlabeled data labeled during the process also aid the categorization of the remaining unlabeled data. Usually few labeled examples are required to perform transductive learning.

Transductive learning is commonly performed using networks to represent the data. A network is defined as $N = \langle \mathcal{O}, \mathcal{R}, \mathcal{W} \rangle$, in which $\mathcal{O}$ represents the set of objects, $\mathcal{R}$ the set of relations among objects, and $\mathcal{W}$ the weight of the relations. If $\mathcal{O}$ consists of a single type of object, this network is called homogeneous network. When $\mathcal{O}$ consists of h different types of objects ($h \geq 2$), i.e., $\mathcal{O} = \mathcal{O}_1 \cup \mathcal{O}_2 \cup \cdots \cup \mathcal{O}_h$, the network is called heterogeneous network [3].

One of the simplest ways to represent a text collection is by using a bipartite heterogeneous network. To build such network, it is not necessary to compute similarities among documents, co-occurrence of terms, nor explicit information about relations among objects. In this network, $\mathcal{O} = \mathcal{D} \cup \mathcal{T}$, in which $\mathcal{D} = \{d_1, d_2, \ldots, d_m\}$ represents objects of document type and $\mathcal{T} = \{t_1, t_2, \ldots, t_n\}$ represents objects of term type. $\mathcal{D}$ consists of labeled ($\mathcal{D}^L$) and unlabeled documents ($\mathcal{D}^U$), i.e., $\mathcal{D} = \mathcal{D}^L \cup \mathcal{D}^U$. A connection occurs only from an object $d_i \in \mathcal{D}$ to an object $t_i \in \mathcal{T}$ and its weight is the frequency of t_i in d_i. Despite this type of network leads to good classification performance, it has been underexplored [5, 6]. Besides, any textual document collection is easily modeled by a bipartite heterogeneous network.

For most of existing transductive algorithms for classification in heterogeneous networks, users have to set one or more parameters which significantly affect the algorithm performance. Setting the best parameters is a difficult task. Furthermore, these algorithms do not provide a high classification accuracy for a small number of labeled examples. To cope with these drawbacks, here we propose a parameter-free algorithm for transductive classification referred to as LPBHN (*Label Propagation using Bipartite Heterogeneous*

SAC'14 March 24-28, 2014, Gyeongju, Korea.

http://dx.doi.org/10.1145/2554850.2554901

Networks). LPBHN represents a textual document collection by a bipartite heterogeneous network. Existing label information is propagated from documents to terms and then the label information from terms are propagated to the unlabeled documents. This processes iterates in a stochastic way until labels do not change. The proposed algorithm presents classification accuracy equivalent or higher than the state-of-the-art algorithms for transductive classification in heterogeneous networks. LPBHN also presents better accuracy and faster classification than transductive algorithms based on homogeneous document networks. We also verified that LPBHN obtains a better accuracy than inductive algorithms considering a bag-of-words representation using much less labeled examples.

The remainder of this paper is organized as follows. Section 2 presents the algorithms for transductive classification in heterogeneous networks used in this work. Section 3 details the proposed algorithm. Section 4 compares the results obtained by the proposed algorithm with the algorithms presented in Section 2. Finally, Section 5 presents the conclusions and future work.

2. HETEROGENEOUS NETWORK-BASED TRANSDUCTIVE CLASSIFICATION

The algorithms presented here are instantiated for the problem of transductive classification of texts represented by bipartite heterogeneous networks as follows. Let w_{d_i,t_j} be the weight of a link between the document d_i and the term t_j and let $\mathcal{C} = \{c_1, c_2, \ldots, c_l\}$ be the set of classes of the document collection. For each network object o_i is assigned a weight vector $\mathbf{f}_i = \{f_1, f_2, \ldots, f_{|\mathcal{C}|}\}$, which stores the weights of the object o_i for each class of the collection. The weight vector defines the object relevance to discriminating each class. Hence, it is also referred to as class information vector. The matrix $\mathbf{F} = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_{|\mathcal{O}|}\}^T$ stores all the weight vectors of the network objects. The vector $\mathbf{y}_i = \{y_1, y_2, \ldots, y_{|C|}\}$ stores the real class information of a labeled document d_i, where y has value 1 in the position corresponding to the class of document i and 0 in the other positions. The matrix $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_{|D^L|}\}^T$ represents class information of all labeled documents.

TM (`Tag-based classification Model`) algorithm, proposed in [10], classifies objects in a bipartite heterogeneous network composed by target objects ($\mathcal{D}$), objects from other domains that aid the classification ($\mathcal{A}$), e.g. authors and conferences information, and bridge objects ($\mathcal{T}$) which connect objects in $\mathcal{D}$ and $\mathcal{A}$. The classification process is guided by the following assumptions: i) class information assigned for vertices in $\mathcal{A}$ and in $\mathcal{D}^L$ must not differ from their real class information; ii) if there exist an initial estimation of the classes of vertices in $\mathcal{D}^U$, the class information of the vertices in $\mathcal{D}^U$ must be closer to these estimations; and iii) the class information of a vertex o_i must be closer to a vertex o_j if o_j is a neighbor of o_i. Given these assumptions, TM algorithm minimizes the following function:

$$Q(\mathbf{F}) = \alpha \sum_{a_i \in \mathcal{A}} ||\mathbf{f}_i - \mathbf{y}_i||^2 + \beta \sum_{d_i \in \mathcal{D}^L} ||\mathbf{f}_i - \mathbf{y}_i||^2 + \gamma \sum_{d_i \in \mathcal{D}^U} ||\mathbf{f}_i - \mathbf{y}_i||^2 + \sum_{o_i \in \mathcal{D} \cup \mathcal{A}, t_j \in \mathcal{T}} w_{o_i,t_j} ||\mathbf{f}_i - \mathbf{f}_j||^2 \quad (1)$$

in which α, β, and γ are parameters to control the weight of each assumption. The classification of the objects uses the "*Class Mass Normalization*" (CMN) concept [11] (see Equation 12).

In [3], a general framework for classification using heterogeneous network called `GNetMine` (GM) is proposed. GM is based on the following assumptions: i) the class information of connected objects must be similar, and ii) the class information assigned during the classification process must not differ from the real class information. Given these assumption, GM attempts to minimize the following function:

$$Q(\mathbf{F}) = \sum_{d_i \in \mathcal{D}} \sum_{t_j \in \mathcal{T}} w_{d_i,t_j} \left|\left| \frac{\mathbf{f}_i}{\sqrt{\Phi(d_i)}} - \frac{\mathbf{f}_j}{\sqrt{\Phi(t_j)}} \right|\right|^2 + \sum_{d_i \in \mathcal{D}^L} \alpha ||\mathbf{f}_i - \mathbf{y}_i||^2 \quad (2)$$

in which α is the importance given to the real class information and $\Phi(d_i)$ is the degree of the object d_i in the network. The classification uses the arg-max value of the weight vectors.

IRC (`Iterative Reinforcement Categorization`) algorithm [9] performs an iterative reinforcement strategy to classify objects in a bipartite heterogeneous network. The trasductive classification is carried out propagating class information from document in $\mathcal{D}^L$ and $\mathcal{D}^U$ to the terms in $\mathcal{T}$ and from terms to documents in $\mathcal{D}^U$. The goal of IRC algorithm is to reach the condition

$$\lim_{t \to \infty} ||\mathbf{F}(\mathcal{D}^U)^{t+1} - \mathbf{F}(\mathcal{D}^U)^t|| = 0 \quad (3)$$

i.e., the class information of unlabeled documents does not change in consecutive iterations. Propagation of information class from documents to terms is according to Equation 4 and information class from terms to documents is propagated according to Equation 5.

$$f_{j,k} = \alpha \cdot \frac{\sum_{d_u \in \mathcal{D}^L} w_{d_u,t_i} f_{u,k}}{\sum_{d_u \in \mathcal{D}^L} w_{d_u,t_i}} + \beta \cdot \frac{\sum_{d_v \in \mathcal{D}^U} w_{d_v,t_i} f_{v,k}}{\sum_{d_v \in \mathcal{D}^U} w_{d_v,t_i}} \quad (4)$$

$$f_{i,k} = \alpha' f_{i,k} + \beta' \cdot \frac{\sum_{t_u \in \mathcal{T}} w_{d_i,t_u} f_{u,k}}{\sum_{t_u \in \mathcal{T}} w_{d_i,t_u}} \quad (5)$$

Parameters α and β represent the relevance of labeled and unlabeled documents respectively. α' and β' represent the relevance of previous class information of documents and terms respectively. A document d_i is classified according to the maximum $f_{i,j}$ of its weight vector.

3. PROPOSED ALGORITHM FOR LABEL PROPAGATION IN BIPARTITE HETEROGENEOUS NETWORKS

LPBHN (*Label Propagation through Bipartite Heterogeneous Networks*) algorithm considers a text collection represented by a bipartite heterogeneous network, as illustrated in Figure 1. The information class propagation procedure attempts to weight the relevance of each term or document for discriminating each class. Given these relevances, information class from documents are propagated to terms and then from terms to documents. This process iterates until convergence, similarly to IRC (Equation 3). Hence, the label propagation can be solved by $\mathbf{F} = \mathbf{P}\,\mathbf{F}$ [11], where the matrix

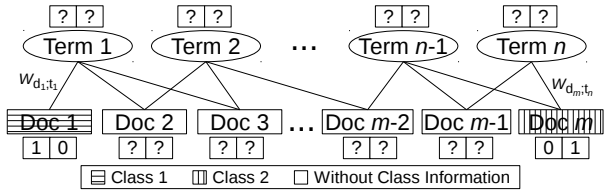


Figure 1: Example of a bipartite heterogeneous network composed by n terms and m documents for transductive classification.

$\mathbf{F}$ (Section 2) stores the class information vectors, and the matrix $\mathbf{P}$ stores the probabilities of the connections among terms and documents computed as follows. The probability of a document d_i be connected to a term t_j is computed by Equation 6. Conversely, Equation 7 computes the probability of a term t_j occurs in an document d_i.

$$p_{d_i,t_j} = \frac{w_{d_i,t_j}}{\sum_{d_k \in \mathcal{D}, w_{d_k,t_j} \in \mathcal{W}} w_{d_k,t_i}} \quad (6) \qquad p_{t_j,d_i} = \frac{w_{d_i,t_j}}{\sum_{t_k \in \mathcal{T}, w_{d_i,t_k} \in \mathcal{W}} w_{d_i,t_k}} \quad (7)$$

Considering that the classification task has labeled documents (D^L), unlabeled documents (D^U), and terms (T), the matrices $\mathbf{F}$ and $\mathbf{P}$ are subdivided considering those distinct types of objects. Thus, the label propagation problem is rewritten by Equation 8.

$$\begin{bmatrix} \mathbf{F}_{D^L} \\ \mathbf{F}_{D^U} \\ \mathbf{F}_T \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{D^L D^L} & \mathbf{P}_{D^L D^U} & \mathbf{P}_{D^L T} \\ \mathbf{P}_{D^U D^L} & \mathbf{P}_{D^L D^L} & \mathbf{P}_{D^U T} \\ \mathbf{P}_{TD^L} & \mathbf{P}_{TD^U} & \mathbf{P}_{TT} \end{bmatrix} \begin{bmatrix} \mathbf{F}_{D^L} \\ \mathbf{F}_{D^U} \\ \mathbf{F}_T \end{bmatrix} \quad (8)$$

We constrain $\mathbf{F}_{D^L} = \mathbf{Y}$ as proposed in [11]. The values of the submatrices $\mathbf{P}_{D^L D^L}$, $\mathbf{P}_{D^L D^U}$, $\mathbf{P}_{D^U D^L}$, $\mathbf{P}_{D^U D^U}$, and $\mathbf{P}_{D^T D^T}$ are 0 since there are no relations among objects of the same type in a bipartite heterogeneous network. Hence, LPBHN performs the transductive classification as follows:

1. Propagate information class from documents to terms: $\mathbf{F}_T \leftarrow \mathbf{P}_{TD^L}\,\mathbf{F}_{D^L} + \mathbf{P}_{TD^U}\,\mathbf{F}_{D^U}$.
2. Propagate information class from terms to documents: $\mathbf{F}_{D^U} \leftarrow \mathbf{P}_{D^U T}\,\mathbf{F}_T$ and $\mathbf{F}_{D^L} \leftarrow \mathbf{P}_{D^L T}\,\mathbf{F}_T$.
3. Redefine de information class of the labeled documents: $\mathbf{F}_{D^L} \leftarrow \mathbf{Y}^L$.
4. Repeat the steps 1, 2, and 3 until convergence.

One can see that $\mathbf{F}_T$ and $\mathbf{F}_{D^U}$ at the n-th iteration can be computed by the Equations 9 and 10 respectively.

$$\mathbf{F}_T^{(n)} = \sum_{i=0}^{n-1} (\mathbf{P}_{D^U T}\mathbf{P}_{TD^U})^i \mathbf{P}_{TD^L}\mathbf{Y}^L + (\mathbf{P}_{TD^U}\mathbf{P}_{D^U T})^{n-1}\mathbf{P}_{TD^U}\mathbf{F}_{D^U}^{(0)} \quad (9)$$

$$\mathbf{F}_{D^U}^{(n)} = \sum_{i=0}^{n-1} (\mathbf{P}_{D^U T}\mathbf{P}_{TD^U})^i \mathbf{P}_{D^U T}\mathbf{P}_{TD^L}\mathbf{Y}^L + (\mathbf{P}_{D^U T}\mathbf{P}_{TD^U})^n \mathbf{F}_{D^U}^{(0)} \quad (10)$$

Each row of matrix $\mathbf{P}$ is row-stochastic or row-normalized [8], i.e., the sum of the values of the row is 1. Thus, the sum of the values of a row in $\mathbf{P}_{D^L T}$ and $\mathbf{P}_{D^U T}$ is lesser than 1 for terms connected to labeled and unlabeled documents, i.e, useful terms for label propagation. The same occurs for the submatrices $\mathbf{P}_{TD^L}$ and $\mathbf{P}_{TD^U}$. Thus, there exist a γ such that

$$\sum_{j=1}^{|\mathcal{D}^U|} (P_{D^U T}P_{TD^U}[i,j]) \leq \gamma < 1 \,.$$

At the n-th iteration we have:

$$\begin{aligned} \sum_j (\mathbf{P}_{D^U T}\mathbf{P}_{TD^U}[i,j])^{(n)} &= \sum_j \sum_k \mathbf{P}_{D^U T}\mathbf{P}_{TD^U}[i,k])^{(n-1)} (\mathbf{P}_{D^U T}\mathbf{P}_{TD^U}[k,j]) \\ &= \sum_k (\mathbf{P}_{D^U T}\mathbf{P}_{TD^U}[i,k])^{(n-1)} \sum_j (\mathbf{P}_{D^U T}\mathbf{P}_{TD^U}[k,j]) \\ &\leq \sum_k (\mathbf{P}_{D^U T}\mathbf{P}_{TD^U}[i,k])^{(n-1)}\gamma \\ &\leq \gamma^{(n)} \,. \end{aligned} \quad (11)$$

For $n \to \infty$,

$$\lim_{n\to\infty} (\mathbf{P}_{D^U T}\mathbf{P}_{TD^U})^n = 0 \,.$$

Thus, the proposed solutions for $\mathbf{F}_T^{(n)}$ and $\mathbf{F}_{D^U}^{(n)}$ converge.

The final classification of the network objects uses the *Class Mass Normalization* (CMN) concept. Hence, the class of a document d_i ($class(d_i)$) using CMN is given by [11]:

$$class(d_i) = \arg\max_{1 \leq l \leq |\mathcal{C}|} Pr[c_l] \cdot \frac{f_{i,l}(\mathcal{D}^U)}{\sum_{d_j \in \mathcal{D}} f_{j,l}(\mathcal{D})} \quad (12)$$

in which $Pr[c_l]$ is the prior probability of the class c_l.

However LPBHN has been designed for text classification, it can be applied to any domain modeled as a bipartite heterogeneous network. Furthermore, we highlight the absence of any parameters for label propagation.

Now we compare LPBHN with the iterative solutions of GM [3], TM [10], and IRC [9]. The strength of label propagation in GM is inversely proportional to the degree of the linked objects. TM set the relevance of real class information by the parameter β. Small value of β decreases the relevance of the real class information from labeled documents if the document has a high degree. On the other hand, if $\beta \to \infty$ there is no change in the information class of labeled documents. Similarly, γ controls the change in the information class of the of unlabeled objects. IRC considers just the weight of the connection to propagating the class information. Another difference is that TM and GM allow to change the class information of labeled objects during the propagation process. In LBPHN it remains unchanged.

4. EXPERIMENTAL EVALUATION

We compared the LPBHN algorithm with the algorithms presented in Section 2. We also compared LPBHN with Gaussian Field with Harmonic Function (GFHF) algorithm [11], which is a traditional algorithm for label propagation in homogeneous network and also performs the transductive classification using $\mathbf{F} = \mathbf{P}\,\mathbf{F}$. For the comparison, we used 31 textual document collections from different domains: e-mails (EM), web pages (WP), news articles (NA), sentiment analysis (SA), technical reports (TR), medical documents

(MD) and TREC documents[1] (TD). Table 1 presents the details of the collections as the number of documents ($|\mathcal{D}|$), the number of terms ($|\mathcal{T}|$), the average number of terms per document ($|\overline{\mathcal{T}}|$), the number of classes ($|\mathcal{C}|$), the standard deviation considering the percentage of the classes ($\sigma(\mathcal{C})$), and the percentage of the majority class ($\max(\mathcal{C})$).

For the collections that belong to 19MClassTextWc set (Fbis, La1s, La2s, Oh0, Oh10, Oh15, Oh5, Ohscal, Tr11, Tr12, Tr21, Tr23, Tr31, Tr41, Tr45, Re0, Re1, Wap) [2] no preprocessing was performed since these collections were already preprocessed. For other collections, single words were considered as terms, stopwords were removed, HTML tags and e-mail headers were removed, terms were stemmed using Porter's algorithm, and only terms with document frequency ≥ 2 were considered. We used term frequency to weight term in documents. The preprocessed collections are available at `http://sites.labic.icmc.usp.br/text_collections/`. More details about the collections can be found at [7].

Table 1: Characteristics of the textual document collections used in the experimental evaluation.

Collection	$\lvert\mathcal{D}\rvert$	$\lvert\mathcal{T}\rvert$	$\lvert\overline{\mathcal{T}}\rvert$	$\lvert\mathcal{C}\rvert$	$\sigma(\mathcal{C})$	$\max(\mathcal{C})$
Classic4 (RE)	7095	7749	35.28	4	1.94	45.16
CSTR (TR)	299	1726	54.27	4	18.89	42.81
Enron-Top-20 (EM)	13199	18194	50.69	20	2.37	9.63
Fbis (NA)	2463	2001	159.24	17	5.66	26.54
Hitech (NA)	2301	12942	141.93	6	8.25	26.21
Industry-Sector (PW)	8817	21490	88.49	12	7.37	11.24
IrishSent (SA)	1660	8659	112.65	3	6.83	39.46
La1s (NA)	3204	13196	144.64	6	8.22	29.43
La2s (NA)	3075	12433	144.83	6	8.59	29.43
MultiDomainSent (SA)	8000	13360	42.36	2	0.00	50.00
NFS (CD)	10524	3888	6.65	16	3.82	13.39
Oh0 (MD)	1003	3183	52.50	10	5.33	19.34
Oh10 (MD)	1050	3239	55.64	10	4.25	15.71
Oh15 (MD)	913	3101	59.30	10	4.27	17.20
Oh5 (MD)	918	3013	54.43	10	3.72	16.23
Ohscal (MD)	11162	11466	60.39	10	2.66	14.52
Polarity (SA)	2000	15698	205.06	2	0.00	50.00
Re0 (NA)	1504	2887	51.73	13	11.56	40.43
Re1 (NA)	1657	3759	52.70	25	5.54	22.39
Reviews (NA)	4069	22927	183.10	5	12.80	34.11
SyskillWebert (WP)	334	4340	93.16	4	10.75	41.02
Tr11 (TD)	414	6430	281.66	9	9.80	31.88
Tr12 (TD)	313	5805	273.60	8	7.98	29.71
Tr21 (TD)	336	7903	469.86	6	25.88	68.75
Tr23 (TD)	204	5833	385.29	6	15.58	44.61
Tr31 (TD)	927	10129	268.50	7	13.37	37.97
Tr41 (TD)	878	7455	195.33	10	9.13	27.68
Tr45 (TD)	690	8262	280.58	10	6.69	23.19
Trec7-3000 (EM)	6000	100464	244.08	2	0.00	50.00
Wap (WP)	1560	8461	141.33	20	5.20	21.86
WebKb (PW)	8282	22892	89.78	7	15.19	45.45

Some of the parameter values used in the experiments were based on the values found in the proposal of the algorithms used for comparison. Other parameter values were defined based on empirical evaluations. For GM we used the parameter $\alpha = \{0.1, 0.3, 0.5, 0.7, 0.9\}$. For TM we used $\alpha = 0$, since there are no objects from different domains, $\beta = \{0.1, 1.0, 10.0, 100.0, 1000.0\}$, and $\gamma = \{0.1, 1.0, 10.0, 100.0, 1000.0\}$. For IRC we used the pairs $[\alpha, \beta] = \{[1, 5], [2, 4], [3, 3], [4, 2], [5, 1]\}$ and $[\alpha', \beta'] = \{[1, 5], [2, 4], [3, 3], [4, 2], [5, 1]\}$. For GFHF we generate document homogeneous networks using a exp-weighted network, in which the weight of the relation between a document d_i and a document d_j (w_{d_i,d_j}) is given by the Gaussin function

$$w_{d_i,d_j} = exp\Big(\frac{-(1 - cos(d_i, d_j)}{\sigma^2}\Big), \tag{13}$$

in which $cos(d_i, d_j)$ is the cosine similarity between the documents d_i and d_j, and σ controls the bandwidth of the Gaussian function. We used $\sigma = \{0.05, 0.2, 0.35, 0.5, 0.75\}$.

The iterative solutions proposed by the respective authors were used for all the algorithms. The maximum number of iterations was set to 1000 since this is a common limit value for iterative solutions. The metric used for comparison was the classification accuracy, i.e., the percentage of correctly classified documents. The accuracies were obtained considering the average accuracies of 10 runs. In each run we randomized the dataset and selected x examples as labeled examples. The remaining $|\mathcal{D}| - x$ examples were used to evaluate the transductive classification. We carried out experiments using 1, 10, and 20 labeled documents for each class to analyze the trade-off between the number of labeled examples and classification accuracy, and the behavior of the algorithms for a different number of labeled examples. The best accuracies obtained by some set of parameters of the algorithms were used for comparison. We used the Friedman's statistical significance test with Nemenyi's post-hoc test and 95% of confidence level [1] to compare the results.

Table 2 presents the highest accuracies for each algorithm using 1, 10, and 20 labeled documents for each class. We notice that LPBHN presented the highest accuracy for a large number of datasets considering all the used numbers of labeled documents. In general, the accuracies obtained by the LPBHN were higher than the other algorithms for domains as sentiment analysis, web pages, e-mails, and medical documents, which typically contain short texts ($|\overline{\mathcal{T}}| \leq 145$).

Figure 2 presents the critical difference diagrams[2] considering the results presented in Table 2. LPBHN obtained the first position in the statistical test ranking for all the used number of labeled documents. Moreover, LPBHN presented better results with statistical significant differences for GM and IRC using 1 labeled document. There was also statistical significant difference when using 10 and 20 labeled documents for IRC.

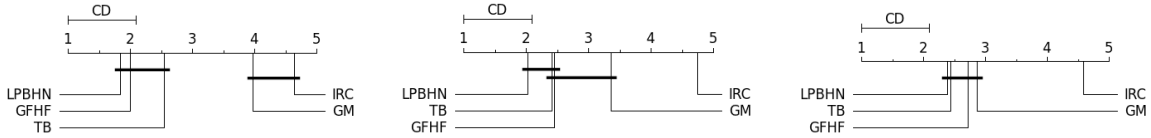
We highlight that LPBHN is parameter-free whereas GM, TM, and IRC requires 1, 2, and 4 parameters respectively, and the comparison of the algorithms considered the highest accuracies obtained by some set of parameters. The choice of the parameters can affect significantly the classification accuracy and the set of parameters that provides the highest accuracy for one collection eventually leads to the lowest accuracy for another collection. To illustrate this point, Figure 3 presents the classification accuracies of the TM algorithm using 20 labeled documents. The set of parameters $[\beta = 1000, \gamma = 100]$ provided the highest accuracy for Classic4 collection and the lowest accuracy for Re0 collection. Besides, the differences among the accuracies using different set of parameters can be high. For instance, the classification accuracies for TM using the set of parameters $[\beta = 1000, \gamma = 100]$ and $[\beta = 100, \gamma = 0.1]$ provided a difference of 23.89 for the Re0 collection.

[1]TREC (*Text Retrieval Conference*): `http://trec.nist.gov/`

[2]Each method is sorted according to the rank of the statistical significant test [1] and the methods connected by a line do not present statistical significant difference among them.

Table 2: Accuracies obtained using 1, 10, and 20 labeled examples for each class.

	1 labeled example					10 labeled examples					20 labeled examples				
Collections	**LPBHN**	**GM**	**TB**	**IRC**	**GFHF**	**LPBHN**	**GM**	**TB**	**IRC**	**GFHF**	**LPBHN**	**GM**	**TB**	**IRC**	**GFHF**
Classic4	**65.83**	51.26	62.71	46.47	62.69	**84.80**	70.82	83.74	72.22	79.13	**86.71**	70.92	85.78	72.14	81.97
CSTR	53.32	46.54	51.22	48.51	**56.71**	71.20	64.48	67.26	68.61	**74.52**	72.92	68.90	69.73	72.37	**74.70**
Enron	**34.17**	27.61	32.92	20.36	32.54	49.91	49.00	**51.42**	35.28	45.13	54.24	54.54	**56.42**	39.09	48.67
Fbis	43.75	28.49	39.22	22.45	**50.54**	64.41	62.82	68.50	42.65	**69.35**	68.37	71.15	**74.51**	52.01	73.75
Hitech	28.77	27.95	29.88	29.05	**40.16**	47.38	44.77	47.68	42.98	**60.12**	64.58	58.46	**67.04**	52.21	**67.04**
IndSec	**35.53**	32.31	35.13	29.13	29.20	**29.21**	27.59	27.46	20.89	21.13	**37.39**	34.91	34.69	25.45	25.58
IrishSent	38.26	37.46	**38.33**	36.81	38.32	**48.39**	46.80	47.73	46.53	47.77	**51.68**	50.64	51.55	48.26	50.19
La1s	44.03	38.79	**44.17**	32.92	42.38	**67.80**	62.50	64.65	54.09	62.88	**72.92**	70.07	71.21	62.05	67.98
La2s	**47.41**	41.15	45.61	33.17	41.70	**69.77**	65.60	67.95	59.06	63.99	74.63	73.39	**74.92**	67.61	68.63
MultiSent	**52.81**	52.07	52.40	51.97	51.98	**56.34**	54.95	55.74	54.03	56.29	58.48	56.38	57.06	56.82	**58.90**
NFS	33.45	30.80	**34.13**	23.83	28.54	51.98	51.01	**52.84**	43.96	50.93	58.26	57.97	**59.17**	50.17	55.59
Oh0	**48.83**	42.01	46.95	39.76	46.70	77.22	75.86	**82.29**	69.05	71.86	81.91	81.34	**82.29**	74.65	76.36
Oh10	38.06	34.99	38.05	34.77	**41.89**	66.03	65.67	**66.91**	61.94	61.86	72.27	72.64	**73.35**	67.44	64.66
Oh15	37.81	33.84	37.56	31.94	**38.32**	**70.44**	67.37	70.26	59.91	68.14	77.42	76.38	**77.66**	67.10	72.92
Oh5	42.25	38.57	42.29	36.49	**44.60**	**71.37**	69.63	71.28	61.30	65.87	**77.77**	76.98	77.60	65.82	69.90
Ohscal	**31.10**	28.55	30.93	27.23	29.96	48.76	45.84	48.05	43.43	**49.09**	**54.66**	52.57	54.23	49.06	54.36
Polarity	**55.10**	52.86	54.13	51.54	53.02	**58.34**	56.23	58.30	52.59	55.59	**60.89**	58.53	60.72	54.54	58.79
Re0	31.86	25.38	30.15	25.57	**42.27**	51.85	48.12	48.96	45.85	**58.10**	53.62	**54.34**	50.29	53.64	61.45
Re1	28.96	26.23	28.43	27.50	**39.19**	49.69	50.64	50.87	48.55	**60.18**	48.44	53.29	49.82	52.83	**59.20**
Reviews	47.25	42.51	47.29	45.99	**55.61**	71.16	70.09	72.81	68.44	**78.92**	76.93	77.39	80.02	75.57	**83.31**
SysWebert	59.24	53.06	57.76	66.27	**86.00**	81.12	74.05	76.22	71.63	**95.31**	85.79	78.31	82.01	75.94	**95.31**
Tr11	48.67	33.63	39.95	26.10	**57.28**	66.80	59.30	67.13	34.33	**76.49**	66.73	65.18	**78.09**	32.61	76.60
Tr12	50.26	34.26	44.13	25.51	**55.57**	68.21	60.85	62.26	27.91	**75.34**	67.87	**76.40**	68.78	41.77	75.55
Tr21	40.33	46.58	35.58	46.52	**55.15**	52.93	51.52	64.56	33.11	**75.97**	50.28	73.40	71.82	34.37	**76.60**
Tr23	46.52	46.26	39.90	39.44	**47.42**	59.59	76.22	50.95	61.76	**76.49**	61.79	**86.52**	58.48	69.73	84.20
Tr31	**59.28**	46.13	44.59	39.28	57.85	81.04	**83.76**	79.32	61.56	82.45	83.73	**88.10**	83.04	68.20	84.93
Tr41	52.94	40.26	46.69	33.79	**53.10**	**78.99**	68.27	68.58	54.60	72.63	**80.58**	72.43	67.81	56.16	78.58
Tr45	**55.96**	46.93	51.19	37.46	55.51	76.68	78.07	**82.73**	41.81	78.24	79.52	86.51	**87.17**	52.79	81.22
Trec7-3000	**69.85**	66.28	67.15	66.07	69.29	**85.72**	85.30	82.31	83.56	83.61	87.27	**87.69**	83.73	84.35	84.89
Wap	**39.05**	26.53	37.18	18.51	33.18	**65.31**	53.55	61.73	41.04	49.47	**69.64**	57.91	64.41	43.29	53.07
WebKb	**25.37**	18.60	24.44	16.93	21.57	**32.99**	33.67	33.47	32.10	24.75	**36.04**	34.12	34.95	33.61	26.25



(a) 1 labeled document. (b) 10 labeled documents. (c) 20 labeled documents.

Figure 2: Critical difference diagrams.

The large number of parameters, the range of the parameter values and the difference in the accuracy provided by different parameters hamper the application of the existing algorithms in practical situations. On the other hand, the proposed algorithm LPBHN, which is parameter-free and obtained an accuracy equivalent or better than the existent algorithms, can be very useful in practical applications.

We also compared LPBHN with a state-of-art inductive algorithm using a bag-of-words (vector space model representation). We consider the best accuracies obtained by Support Vector Machines (SVM) presented in [6]. These accuracies were obtained through a 10-fold cross-validation, i.e., using 90% of labeled examples for training. We also obtained results for SVM using 10 runs with 10% of labeled examples in each run. The parameters used for SVM are the same used in [6]. Figure 4 presents the results obtained by SVM with 10% and 90%, and the results obtained by LPBHN using 10%, 20%, 30%, 40%, 50%, and 60% of labeled documents. We can notice that LPBHN can obtain

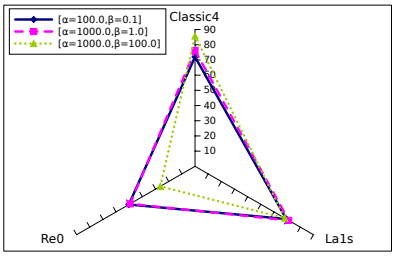


Figure 3: Accuracies obtained using different sets of parameters for TM algorithm.

a higher accuracy than SVM with much less labeled examples. For instance, LPBHN with 10% of labeled examples exceeds the accuracy of SVM with 90% labeled examples for SyskillWebert collection.

The use of a bipartite heterogeneous network can speed

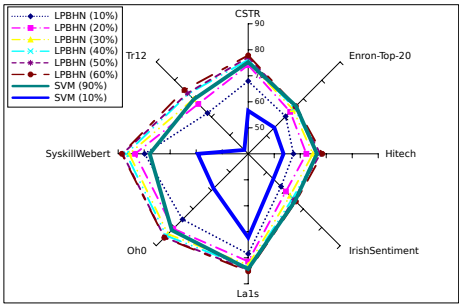


Figure 4: Accuracies obtained by LPBHN using a smaller number of labeled examples than SVM.

up the classification since this network is easily generated (there is no need to compute similarities among documents) and the number of connections among terms and documents in a bipartite network can be smaller than the connections among documents in a homogeneous network. To prove this, we ran LPBHN and GFHF using 5 collections with the highest number of documents (Enron, Ohscal, NFS, IndSec, and WebKb) and 5 collections with the highest number of terms (Trec7-3000, Reviews, WebKb, IndSec, and Enron). We used 20 labeled documents and 1000 as the maximum number of iterations. We obtained the classification time in a computer with Intel Core i7-2600K 3.40GHz processor. Figure 5 presents the square root of the classification time (seconds) to better visualize the results. We notice that the time spent to classify objects in a bipartite heterogeneous network is much smaller than in a document homogeneous network. For intance, LPBHN classified the documents of the NFS collection spending 1% of the time spent by GFHF.

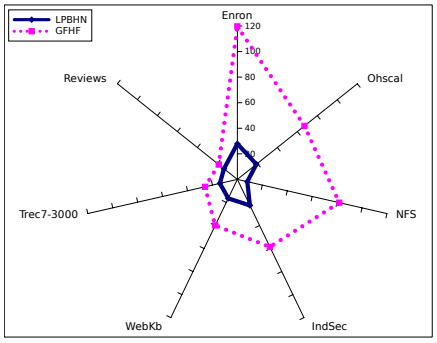


Figure 5: Square root of the time (seconds) to classify documents with LPBHN and GFHF.

5. CONCLUSIONS AND FUTURE WORK

In this work we presented a parameter-free algorithm for classification of texts represented by a bipartite heterogeneous network referred to as LPBHN (*Label Propagation through Bipartite Heterogeneous Network*). LPBHN performs a transductive classification through the label propagation from documents to terms and from terms to documents in a stochastic way.

To evaluate LPBHN we used 31 text collections from different domains. LPBHN outperformed transductive algorithms for classification in bipartite heterogeneous networks or even in homogeneous networks. LPBHN can also exceed the accuracy of a state-of-art inductive algorithm using much less labeled examples. These facts make the application of the LPBHN algorithm interesting in situations in which the label of documents requires a huge human effort.

We highlighted that LPBHN is parameter-free and the comparisons with other parametric algorithms were carried out considering the best combination of parameters, i.e., the set of parameters that provided the highest accuracies. We also highlighted that the different combinations of parameters can provide large differences in the classification accuracy, which impairs the application of the existing algorithms in practical situations.

As future work we intend to generalize the LPBHN algorithm and apply it to other types of data that can be modeled as a bipartite heterogeneous network. We also intend to incorporate relations of the type `document-document` and `term-term` with relations of the type `document-term` in a heterogeneous network and analyze their impact on the accuracy. Moreover, we intend to extend the matrix **F** and **P** to incorporate more than two types of objects.

6. REFERENCES

[1] J. Demsar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[2] G. Forman. 19MclassTextWc dataset, 2006.

[3] M. Ji, Y. Sun, M. Danilevsky, J. Han, and J. Gao. Graph regularized transductive classification on heterogeneous information networks. In *Proc. Eur. Conf. on Machine Learning and Knowledge Discovery in Databases*, pages 570–586. Springer-Verlag, 2010.

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

[5] M. Newman. *Networks: An Introduction*. Oxford University Press, Inc., 2010.

[6] R. G. Rossi, T. P. Faleiros, A. A. Lopes, and S. O. Rezende. Inductive model generation for text categorization using a bipartite heterogeneous network. In *Proc. Int. Conf. on Data Mining*, pages 1086–1091. IEEE, 2012.

[7] R. G. Rossi, R. M. Marcacini, and S. O. Rezende. Benchmarking text collections for classification and clustering tasks. Technical Report 395, Institute of Mathematics and Computer Sciences - University of Sao Paulo, 2013.

[8] E. Seneta. *Non-negative Matrices and Markov Chains*. Springer series in statistics. Springer, 2006.

[9] G.-R. Xue, D. Shen, Q. Yang, H.-J. Zeng, Z. Chen, Y. Yu, W. Xi, and W.-Y. Ma. IRC: An iterative reinforcement categorization algorithm for interrelated web objects. In *Proc. Int. Conf. on Data Mining*, pages 273–280. IEEE, 2004.

[10] Z. Yin, R. Li, Q. Mei, and J. Han. Exploring social tagging graph for web object classification. In *Proc. Int. Conf. on Knowledge Discovery and Data Mining*, pages 957–966, 2009.

[11] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proc. Int. Conf. on Machine Learning*, pages 912–919. AAAI Press, 2003.