A Sampling-based Framework for Transductive Classification in Information Networks

Brucce N. dos Santos
Institute of Mathematical and Computer Sciences
University of São Paulo
São Carlos, SP, Brazil
Email: brucce.neves@usp.br

Ricardo M. Marcacini
Laboratory of Scientific Computing (LivES)
Federal University of Mato Grosso do Sul
Três Lagoas, MS, Brazil
Email: ricardo.marcacini@ufms.br

Solange O. Rezende
Institute of Mathematical and Computer Sciences
University of São Paulo
São Carlos, SP, Brazil
Email: solange@icmc.usp.br

***Abstract*—Knowledge extraction from large information networks has received increasing attention in recent years. Among existing methods for knowledge extraction, transductive classification is a well-known semi-supervised learning method, where both labeled and unlabeled vertices are used in the learning process. However, transductive classification tasks become impractical in large information networks and the use of network sampling techniques in the transductive classification setting is not a trivial task, since it is required that all the vertices of the original network be classified during the transductive learning — and not only the vertices of the sample. In this paper, we present a framework called TCSN (Transductive Classification for Sampled Networks). TCSN allows the use of various network sampling techniques, as well as enables the use of various methods of transductive classification for information networks. We present a variation of the Chernoff Bounds method to calculate the minimum size of a sampled network, thereby bounding sampling error within a pre-specified tolerance level. Moreover, TCSN extends the concept of evidence accumulation to combine the results of several rounds of transductive classification into a final classification. Experimental results from different information networks reveals that TCSN statistically outperformed the classification performance in the whole original network. These promising results show that the TCSN enables transductive classification in large information networks without loss of quality in the knowledge extraction process.**

***Index Terms*—network sampling, classification, regularization**

I. Introduction

Information networks are very useful for modeling relationships between real-world entities through vertices and edges [1]. There are many applications involving information networks, such as co-author networks or paper citation networks extracted from bibliographic databases, social networks, networks of financial transactions for fraud detection, and interaction networks between users and items for recommendation systems. Information Networks can also be interpreted as graphs, where vertices contain extra information, such as labels and features [2].

Knowledge extraction from large information networks has received increasing attention in recent years [3]. Among existing methods for knowledge extraction, it is worth mentioning semi-supervised learning for information networks, since it allows learning from labeled and unlabelled data [4]. Thus, given a small set of labeled vertices, a large set of unlabelled vertices is classified considering the structure of the information network. This technique is also called a transductive classification [5], since the entire data set is known in advance during the learning process. However, transductive classification tasks become impractical in large information networks [6]. In this scenario, sampling techniques for networks are essential.

We can define the machine learning task from sampled networks as follows. Let $\mathcal{N} = (V, E, W)$ be an information network, where V is a set of vertices, E is the set of edges between vertices, and W is the set of weights of the edges. Sampling techniques aim to obtain a reduced network $\mathcal{N}_s = (V_s, E_s, W_s)$, with $|V_s| \ll |V|$ and similar performance P in some machine learning task L, i.e., $P(L, \mathcal{N}) \cong P(L, \mathcal{N}_s)$. In fact, network sampling techniques obtain (sub)networks with representative vertices in order to maintain the main characteristics of the original network [7]. In a traditional scenario, inductive classification methods can directly use the sampled network as a training set to obtain a classification model. On the other hand, the use of network sampling techniques in transductive classification setting is not a trivial task, since it is required that all the vertices of the original network be classified during the transductive learning. Moreover, there are other important research questions such as (i) what is the best network sampling technique?; (ii) how to define the minimum size of the sampled network?; and (iii) how to extend transductive classification for sampled networks?

In this paper, we present a framework called TCSN (Transductive Classification for Sampled Networks). TCSN allows the use of various network sampling techniques, such as vertex-based sampling or edge-based sampling, as well as the use of various methods of transductive classification for information networks. To the best of our knowledge, this would represent the first attempt of a practical integration between network sampling techniques and transductive classification methods. Our main contributions are two-fold:

- We present a variation of the Chernoff Bounds [8] to calculate the minimum size of a sampled network, given (i) a confidence level of the approximation in relation to the original network and (ii) the number of classes.

Thus, our proposed TCSN has the advantage of bounding sampling error within a pre-specified tolerance level.

- We proposed a method to perform repeated sampling in the information network in order that each vertex is reached at least once. A transductive classification method is applied to each sampling. Next, we use the concept of evidence accumulation [9] to combine the results of several rounds of transductive classification into a final classification, in which all vertices are classified. Moreover, we demonstrate that the computational complexity of the TCSN is proportional to the size of the sampled network and the number of sampling repetitions.

We carried out a thorough experimental evaluation of the proposed TCSN framework, involving twelve real-world information networks, six network sampling techniques, and a state-of-the-art method for transductive network classification. We statistically compared the results of transductive classification in sampled networks with transductive classification in original networks. The analysis of the results reveals that edge-based sampling techniques and some techniques based on random walk have achieved an impressive classification performance (Macro-F1), outperforming even the classification performance in the original network. These promising results show that the TCSN enables classification in large information networks by using sampling techniques, without loss of quality in the knowledge extraction process.

II. Background and Related Work

In this section, we discuss the basic concepts and related work involving transductive classification for networks, as well as network sampling techniques.

A. Transductive Classification for Information Networks

Transductive classification for information networks has received great attention in recent years, where the central idea is to use labeled vertices, unlabeled vertices and the network topology to infer a class confidence vector for all network vertices [10]. Many popular transductive classification methods have been applied successfully in different areas. Zhu et al. [4] proposed a transductive learning method using Gaussian fields and harmonic functions. Zhou et al. [11] proposed a novel transductive learning method based on local and global consistency. Belkin et al. [12] developed a general-purpose regularization framework for transductive classification in information networks. Although there are differences in these proposals, two properties are common to the methods of transductive classification [5]: First, the estimated class confidence vectors of two vertices must be similar if these two vertices are linked in the information network. Second, the estimated class confidence vectors of labeled vertices should be similar to real class information.

Transductive classification methods for information networks can be generically defined through a regularization framework [12]. Let $\mathcal{N} = (V, E, W)$ be an information network, where V is a set of vertices, E is the set of edges between vertices, and W is the set of weights of the edges. Let V_L be a set of labeled vertices, with $V_L \subset V$. Equation 1 defines the regularization framework for transductive learning [13], where the first term ($\Omega(.)$) calculates the proximity of the class confidence vectors between each pair of vertices in the network. The second term ($\Omega'(.)$) calculates the proximity between the estimated class confidence vector of labeled vertices and their real class information. Moreover, $w_{u,v}$ indicates the weight of the relation between the vertices and μ indicates the importance of the real class information during the classification process. The $\mathbf{f}_v$ indicates the estimated class confidence vector of a vertex v; and $\mathbf{y}_u$ indicates real class information of a labeled vertex u. The regularization function is a minimization problem that aims to obtain a class confidence matrix $\mathbf{F}$, which represents the estimated class confidence of the entire information network.

$$Q(\mathbf{F}) = \frac{1}{2} \sum_{u,v \in V} w_{u,v} \Omega(\mathbf{f}_u, \mathbf{f}_v) + \mu \sum_{u \in V^L} \Omega'(\mathbf{f}_u, \mathbf{y}_u) \quad (1)$$

A promising approach to instantiating this regularization framework was proposed by Ji et al. [5], called GNetMine. GNetMine considers different levels of importance for the vertices, as well as the level of importance of the labeled data. In practice, GNetMine generalizes other regularization functions proposed in the literature. Equation 2 defines the GNetMine regularization function, where the $\lambda_{(u,v)}$ defines the importance level between vertices u and v, with $0 \leq \lambda_{(u,v)} \leq 1$. To suppress popular vertices (high degree) from dominating the class vector confidence estimations, $d(.)$ is used to sum the edge weights of all neighbors of a vertex u belonging to the same relationship[1] of (u, v). The importance of real class information of a labeled vertex u is defined by $\alpha_{(u)}$, with $0 < \alpha_{(u)} \leq 1$.

$$Q(\mathbf{F}) = \sum_{u,v \in V} \lambda_{(u,v)} w_{u,v} \left\| \frac{\mathbf{f}_u}{\sqrt{d(u,v)}} - \frac{\mathbf{f}_v}{\sqrt{d(v,u)}} \right\|^2 + \sum_{u \in V^L} \alpha_{(u)} (\mathbf{f}_u - \mathbf{y}_u) \quad (2)$$

GNetMine can be solved through iterative solutions called *label propagation*. In this case, vertices gradually propagate their class information to neighboring vertices considering their relation weights. In the label propagation, the class confidence vector $\mathbf{f}$ of unlabeled vertices is initialized with 0 and $\mathbf{f} = \mathbf{y}$ for the labeled vertices. The stopping criterion of the label propagation is obtained when there are no significant changes in the class matrix confidence $\mathbf{F}$ of the vertices (or a maximum number of iterations).

B. Network Sampling

Network sampling techniques have been used for a wide variety of applications. Previous work uses network sampling to improve visualization tasks [14]. Studies in graph clustering

[1]For homogeneous information networks, all vertices belong to the same type of relationship. Heterogeneous networks organize vertices in different types of relationships.

use sampling for speed up algorithms, such as spectral clustering [15]. Network sampling has also been employed for noise removal and speed up inductive relational classification algorithms [16], but is still underexplored in the context of transductive classification.

Although different sampling techniques have been proposed in recent years, the most popular are known as sampling by exploration (e.g. random walk) and edge-based sampling. In this section, we describe six popular techniques that we consider appropriate for large information networks, due to the low computational cost [7], [17].

- **Edge Sampling (Edge) [7]:** Randomly select a subset of edges $E_s \subset E$. The set of sampled vertices is $V_s = \{u, v | (u, v) \in E_s\}$, which is added in the $\mathcal{N}_s$ network. This process is repeated until the desired network sample size is reached.
- **Simple Random Walk Sampling (SRW) [7], [18]:** Randomly select an initial vertice $v \in V$. Let $Z(v)$ be the set of neighboring vertices of v. Randomly select a neighbor $u \in Z(v)$ and add (u, v) to the sampled network $\mathcal{N}_s$. Repeat the process from u and stop the random walk when the desired network sample size is reached. If the random walk stuck on isolated component of the network, then restart the walk from an unvisited (random) vertex.
- **Random Walk Sampling with Fly Back Probability (RWF) [7]:** Performs a random walk similar to the SRW, but considering a probability p of returning to some vertex already visited.
- **Induced Subgraph Random Walk Sampling (ISRW) [19]:** First, it performs network sampling using the SRW. Next, all edges E of the original network $\mathcal{N}$ that connect $u, v \in \mathcal{N}_s$ are added to the sampled network $\mathcal{N}_s$, if $E(u, v) \notin \mathcal{N}_s$. Thus, the average degree of the $\mathcal{N}_s$ gets closer to the original network $\mathcal{N}$.
- **Snowball (SB) [20]:** In the first stage, select randomly a set of k vertices and add to $V^{(0)}$. In the next stage i, obtain a sample of edges $E^{(i)}$ from the k neighbors of each vertex in $V^{(i-1)}$. Vertices selected in this stage are $V^{(i)} = \{u, v | (u, v) \in E^{(i)}\}$. The final sample V_{SB} consists of the union of the vertices selected in each stage t (until reaching the sample size), i.e., $V_{SB} = \cup_{i=0}^{t} V^{(i)}$.
- **Forest Fire (FF) [21]:** Randomly select a initial vertex and begin "burning" associated edges and the corresponding neighbor vertices. If an edge gets burned, the neighbors vertices get a chance to burn its own edges, and so on recursively until reaching the desired sample size. FF has the burning probability p parameter.

III. Transductive Classification for Sampled Networks (TCSN)

In this section, we present details of our TCSN framework, which is divided into three steps: (1) compute the size of the sampled network, (2) repeat the sampling process until each vertex $v \in V$ is present in some sampled network; and (3) combine the result of the individual transductive classification from each sampled network using evidence accumulation.

In the first step of the TCSN, our goal is to define a lower bound regarding the number of vertices of the sampled network, i.e, a minimum number of vertices $|\mathcal{N}_s|$ to represent the original network $\mathcal{N}$ with a certain level of confidence α. To determine this lower bound, we propose a variation method based on Chernoff bounds, where we also consider the class information, in particular, the expected fraction of labeled vertices by each class.

Let $B(\mathcal{N})$ be the lower bound for the number of vertices $|\mathcal{N}_s|$, where the original network $\mathcal{N}$ has c classes and a total of $|V|$ vertices. Let e be a fraction of the minimum number of vertices expected in each class, where $0 < e < 1$. Thus, Equation 3 calculates the minimum number of vertices so that the sampled network $\mathcal{N}_s$ can maintain the class distribution of the original network (given a confidence level α)[2].

$$B(\mathcal{N}) \geq \left\lfloor e|V| + c \cdot log\Big(\frac{1}{\alpha}\Big) + c\sqrt{log\Big(\frac{1}{\alpha}\Big)^2 + 2e\frac{|V|}{c}log\Big(\frac{1}{\alpha}\Big)} \right\rfloor \quad (3)$$

Once a minimum number of vertices has been determined to guarantee the class distribution, the network sampling techniques aim to maintain the topological properties of the original network. Thus, in the second step of the TCSN, we perform repeated network samplings until each vertex is present in some sampled network.

Let $S = \{\mathcal{N}_{s_1}, \mathcal{N}_{s_2}, ..., \mathcal{N}_{s_m}\}$ be a set of m sampled networks from a network $\mathcal{N}$. Let V_{s_j} be the set of vertices of the j-th sampled network $\mathcal{N}_{s_j}$. Let V^L be the set of labeled vertices of the network $\mathcal{N}$, with $V^L \subset V$. The sampling repetition of the TCSN is performed until two criteria are satisfied:

1) The labeled vertices V^L must be present in each sampled network (Equation 4); and

$$\bigcap_{j=1}^{m} (V_{s_j} \cap V^L) = V^L \quad (4)$$

2) The union of the sets of vertices of each sampled network must be equal to the set of vertices of the original network (Equation 5).

$$\bigcup_{j=1}^{m} V_{s_j} = V \quad (5)$$

While the first criterion allows to maintain labeled information for transductive classification in each sampling, the second criterion guarantees that a vertex will be classified at least once in some sampled network.

In our proposed TCSN, a given sampling technique is applied repeatedly until these two criteria are satisfied. The first criterion can be reached more easily by starting the sampling technique from V^L or by adding V^L in the sampled network at the end of the sampling process. The latter strategy is used in the TCSN. Although the number m of sampled

[2]A proof is available along with the technical documentation and source code of the TCSN framework at https://github.com/BrucceNeves/TCSN

networks may vary according to the sampling technique, the size of each sampled network is close to the value determined by the estimated lower bound.

In the third step of the TCSN, each sampled network is used as input to a transductive classification process resulting in m class confidence matrices (one for each sampled network) $R = \{\mathbf{F}_1, \mathbf{F}_2, ..., \mathbf{F}_m\}$, where $\mathbf{F}_i$ corresponds to the class confidence matrix obtained with the transductive learning (e.g. GNetMine regularization) from the sampled network $\mathcal{N}_{s_i}$.

To obtain the final class confidence matrix $\hat{\mathbf{F}}$, we use the idea of evidence accumulation, where $\Gamma = \{\gamma_1, \gamma_2, ..., \gamma_m\}$ indicates the importance levels of each individual class confidence matrix $\mathbf{F}_i \in R$, with $\gamma_i > 0$, $\forall \gamma_i \in \Gamma$. Equation 6 defines the calculation of the final class confidence matrix, where $\mathbf{F}_i^{(v)}$ indicates the class confidence vector of an vertex v in a sampled network $\mathcal{N}_{s_i}$. Note that if the Γ importance levels was equal to 1 for all samplings, then the evidence accumulation of the $\hat{\mathbf{F}}$ is an unweighted mean. However, this importance can be estimated in alternative ways, such as accuracy of the transductive classifier of each sampling, thereby resulting in a weighted average.

$$\hat{\mathbf{F}} \leftarrow \frac{\sum_{i=1}^{m} \sum_{v \in V_{s_i}} \gamma_i(\mathbf{F}_i^{(v)})}{\sum_{i=1}^{m} \gamma_i} \tag{6}$$

After obtaining the indicator class matrix $\hat{\mathbf{F}}$ by evidence accumulation from each sampling, we can define the final label of a vertex $v \in V$ based on the class label with the highest confidence value in the estimated class vector $\hat{\mathbf{f}}^{(v)} \in \hat{\mathbf{F}}$, as defined in Equation 7, where the function cl returns the class label c for a given class confidence vector. Thus, all the vertices of the original information network can be classified considering the combination of class information confidence in each sampled network.

$$label = cl\Big(\arg\max_{1 \le i \le c} \hat{\mathbf{f}}_i^{(v)}\Big) \tag{7}$$

The computational complexity of the transductive classification on the original information network is given by the complexity of the network regularization, i.e., $\Theta(ct(|V| + |E|))$, where c is the number of classes and t is the number of iterations.

The TCSN depends on the time complexity of each step, defined as $\mathcal{T}_{spl}$ (network sampling technique), $\mathcal{T}_{reg}$ (regularization in a sampled network) and $\mathcal{T}_{ec}$ (evidence accumulation). Thus, the complexity of the TCSN is $\Theta(\mathcal{T}_{spl} + \mathcal{T}_{reg} + \mathcal{T}_{ec})$. While time complexity for network sampling and evidence accumulation techniques are linear ($|V| + |E|$), the time complexity for network regularization is $\mathcal{T}_{reg} = mct(|V_s| + |E_s|)$. In this case, m is the number of sampled networks. Also, the values $|V_s|$ and $|E_s|$ are defined by Chernoff Bounds of the Equation 3. From a practical point of view, transductive classification executions are independent of each other and can be obtained in parallel for each sampled network, thereby allowing the TCSN to be an alternative for scalability in large information networks.

IV. Experimental Evaluation

A. Datasets

We carried out an experimental evaluation with twelve information networks representing events extracted from Reuters[3]. These information networks were used in the Websensors Project[4] and are appropriate for experimental analysis in our scenario due to the different types of domain, size and topology. Table I presents an overview of these information networks, including the network domain type, number of vertices ($|V|$), number of edges ($|E|$) and number of classes.

TABLE I
Overview of the information networks used in the experimental evaluation.

Information Network ($\mathcal{N}$)	$\|V\|$	$\|E\|$	#Classes
Business Transactions (BT)	27604	322989	4
Commodity Markets (CM)	45615	857476	3
Consumer Finances (CF)	2526	22004	3
Crimes And Justice (CJ)	81202	1311486	3
Exchange Markets (EM)	114681	1515632	3
General Subjects (GS)	34482	414872	7
Government Indicators (GI)	31534	438813	4
Inflation (INF)	4016	39173	2
Lawsuits (LAW)	29516	384018	2
Natural Disasters (ND)	20047	263648	3
Reports (REP)	33502	402657	4
Trade Reserves (TR)	13799	178773	3

All the information networks used in this work are undirected and unweighted. These information networks are organized into vertices representing events, as well as vertices representing textual information, geographic information and temporal information. For transductive learning, each vertex type is used with a certain level of importance in the regularization process.

B. Experimental Setup

In the network sampling step, we used the Equation 3 of the TCSN framework to define the minimum sampling size, with the expected fraction of vertices by class $e = 0.15$. The confidence level for the lower bound was defined as 95% ($\alpha = 0.95$). Table II summarizes the percentage of the size ($100 \times \frac{|V_s|+|E_s|}{|V|+|E|}$) of the sampled networks in relation to the original network according to each sampling technique.

All sampling techniques receive the minimum sample size as input parameter. Here, we present the other parameters used in each technique.

- **Edge:** The number of sampled edges starts with the same value as the minimum sample size. The number of trials to reach the minimum number of vertices was defined as $k = 5$.
- **SRW, ISRW:** These techniques use only the minimum sample size as parameter.
- **RWF:** Minimum sample size and fly back probability $p = 0.5$.

[3]RCV1 (Reuters Corpus Volume 1)
[4]https://websensors.net.br/

TABLE II
PERCENTAGE OF THE SAMPLED NETWORKS SIZE IN RELATION TO THE ORIGINAL NETWORK ACCORDING TO EACH SAMPLING TECHNIQUE.

$\mathcal{N}$	Edge	FF	ISRW	RWF	SB	SRW
BT	3.52%	3.22%	12.69%	3.60%	3.54%	3.62%
CM	2.01%	1.87%	13.48%	2.26%	2.37%	2.28%
CF	10.56%	8.15%	15.98%	8.95%	9.24%	8.92%
CJ	2.13%	2.00%	12.17%	2.40%	2.35%	2.42%
EM	2.33%	2.65%	17.64%	2.98%	3.28%	3.01%
GS	3.79%	3.34%	12.05%	3.70%	3.51%	3.71%
GI	2.96%	2.76%	12.40%	3.12%	3.09%	3.14%
INF	6.50%	5.52%	14.39%	6.08%	6.25%	6.08%
LAW	2.82%	2.62%	11.30%	2.88%	2.84%	2.88%
ND	3.35%	2.97%	11.19%	3.16%	3.20%	3.17%
REP	3.32%	3.11%	12.38%	3.37%	3.24%	3.38%
TR	3.78%	3.40%	12.01%	3.63%	3.78%	3.65%

- **SB:** Minimum sampling size and number of neighbors $k = 100$.
- **FF:** Minimum sampling size and burning probability calculated as described in [7].

We implemented a classifier based on the GNetMine regularization to perform the transductive classification. The GNetMine parameters for each dataset were tuned by means of 10-fold cross validation. In the evidence accumulation, we define the same level of importance for each sampled network.

The source code of the sampling techniques, sampled networks, transductive classifier, and parameter analysis are publicly available in the Git repository of the TCSN framework at https://github.com/BrucceNeves/TCSN.

$$F1_{macro} = \frac{2 \times P_{macro} \times R_{macro}}{P_{macro} + R_{macro}} \tag{8}$$

$$P_{macro} = \frac{1}{|C|} \sum_{c_i \in C} \frac{TP_{c_i}}{(TP_{c_i} + FP_{c_i})} \tag{9}$$

$$R_{macro} = \frac{1}{|C|} \sum_{c_i \in C} \frac{TP_{c_i}}{(TP_{c_i} + FN_{c_i})} \tag{10}$$

We used precision and recall measures to analyze the classification performance. In particular, we adopted Macro-F1 [22], which is defined as the harmonic mean (Equation 8) between Precision-Macro (Equation 9) and Recall-Macro (Equation 10). In this case, TP_{c_i} indicates the true positives of the class c_i, FN_{c_i} the false negatives of the class c_i, and C indicates the set of labels of the network, where $c_i \in C$.

For all information networks, we used 50 randomly labeled vertices and the 10-fold cross validation process to estimate the Macro-F1 measure.

C. Results and Discussion

We present and discuss the experimental results considering two aspects: (1) the Macro-F1 performance of the transductive classification of the TCSN in comparison with the transductive classification in the original information network; and (2) the computational cost of each sampling technique used in the TCSN, considering the number of sampling repetitions.

Table III presents the Macro-F1 results for each information network and sampling technique. The last column describes the Macro-F1 results for the original information network (without sampling). While it was expected that the classification in sampling scenarios would obtain an approximation of the classification performance without sampling, we obtained the impressive results of the improvement of classification performance using TCSN for some sampling techniques. We believe that the step of evidence accumulation from different sampled networks of the proposed TCSN is also a way to reduce the impact of outliers and noises, consequently improving transductive learning.

TABLE III
MACRO-F1 RESULTS OF THE PROPOSED TCSN IN COMPARISON WITH TRANSDUCTIVE CLASSIFICATION WITHOUT SAMPLING.

	TCSN						No
$\mathcal{N}$	Edge	FF	ISRW	RWF	SB	SRW	Sampling
BT	0.655	0.500	0.626	0.643	0.570	0.638	0.548
CM	0.874	0.735	0.850	0.862	0.821	0.860	0.833
CF	0.887	0.716	0.877	0.875	0.847	0.870	0.830
CJ	0.733	0.573	0.671	0.714	0.647	0.711	0.637
EM	0.616	0.537	0.549	0.609	0.544	0.607	0.562
GS	0.656	0.442	0.623	0.631	0.560	0.630	0.514
GI	0.661	0.495	0.603	0.641	0.553	0.641	0.531
INF	0.864	0.789	0.849	0.856	0.827	0.853	0.802
LAW	0.759	0.692	0.740	0.753	0.725	0.751	0.724
ND	0.853	0.709	0.841	0.838	0.794	0.839	0.758
REP	0.594	0.454	0.565	0.578	0.517	0.577	0.496
TR	0.789	0.647	0.775	0.774	0.709	0.771	0.671

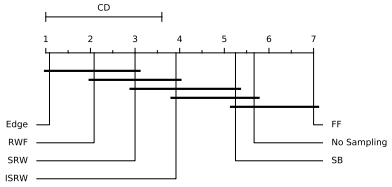

Fig. 1. Critical difference diagram between each sampling technique.

A statistical analysis of Macro-F1 results is presented in Figure 1, which illustrates the critical difference diagram between each sampling technique (Friedman Non-Parametric Test with Nemenyi post-hoc test [23]). In this case, each sampling technique is organized into an average ranking based on Macro-F1. If there is no statistically significant difference between two techniques, then we connect these two techniques by means of a line. The statistical analysis reveals that the techniques Edge, RWF and SRW used in the TCSN obtains superior results to the transductive classification without sampling. On the other hand, the sampling techniques ISRW, SB and FF used in the TCSN did not obtain Macro-F1 performance with a statistically significant difference in comparison to the transductive classification without sampling.

Regarding the computational cost of each sampling technique in TCSN, Figure 2 illustrates an overview of the computational cost (average number of sampling repetitions) in relation to the overall rank average. Each sampling technique

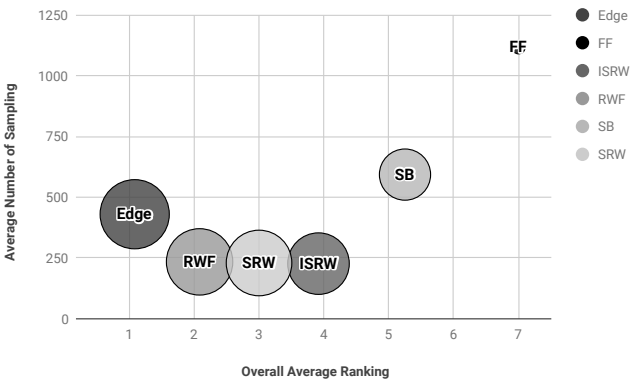

Fig. 2. Comparison of the computational cost (average number of sampling repetitions) in relation to the overall rank average for each sampling technique. Each sampling technique is shown as a circle, in which the larger the radius of the circle, the better the Macro-F1 performance.

is shown as a circle, in which the larger the radius of the circle, the better the Macro-F1 performance of the technique (i.e. average Macro-F1 value in all information networks). The FF and SB sampling techniques did not achieve good Macro-F1 performance. In addition, these techniques present the highest computational cost. We observed that these two sampling techniques have low diversity in each round of sampling, thereby requiring many repetitions to reach all vertices of the information network. Moreover, evidence accumulation is impaired in scenarios with low diversity.

We note that Edge Sampling achieves the highest performance Macro-F1, but with a slightly higher computational cost than Random Walk based techniques (RWF, SRW, and ISRW). In practice, Edge, RWF and SRW showed the best performances (Macro-F1 and computational cost) in most settings, being the most appropriate for transductive classification in large information networks.

V. Concluding Remarks

Generating smaller and representative information networks from large networks is an important task for knowledge extraction in practical scenarios. Our experimental results revealed that the simple sampling techniques based on Edge Sampling and Random Walk (RWF and SRW) are the most suitable for transductive classification in sampled networks. In addition to achieving good Macro-F1 classification results, such techniques presented lower computational cost. TCSN using Edge Sampling, RWF or SRW achieved a minium average improvement of 12% in Macro-F1 performance (considering all datasets) compared to the transductive classification without sampling. This improvement is obtained with a significant reduction in computational time, since the transductive classification can be performed in parallel for each sampled network.

Directions for future work include extending sampling techniques for weighted information networks. We also plan to evaluate other network regularization algorithms.

Acknowledgements

This work was supported by National Council for Scientific and Technological Development (CNPq) [426663/2018-7 and 433082/2018-6], CAPES, and Birdie (https://birdie.ai/).

References

[1] C. Chen, C. X. Lin, M. Fredrikson, M. Christodorescu, X. Yan, and J. Han, "Mining large information networks by graph summarization," in *Link Mining: Models, Algorithms, and Applications*. Springer, 2010, pp. 475–501.

[2] C. Shi, Y. Li, J. Zhang, Y. Sun, and S. Y. Philip, "A survey of heterogeneous information network analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 1, pp. 17–37, 2016.

[3] M. Wang, W. Fu, S. Hao, H. Liu, and X. Wu, "Learning on big graph: Label inference and regularization with anchor hierarchy," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 5, pp. 1101–1114, 2017.

[4] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proceedings of the 20th Int. Conf. on Machine learning (ICML-03)*, 2003, pp. 912–919.

[5] M. Ji, Y. Sun, M. Danilevsky, J. Han, and J. Gao, "Graph regularized transductive classification on heterogeneous information networks," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2010, pp. 570–586.

[6] I. Atastina, B. Sitohang, G. P. Saptawati, and V. S. Moertini, "A review of big graph mining research," in *IOP Conf. Series: Materials Science and Engineering*, vol. 180, no. 1. IOP Publishing, 2017, pp. 1–10.

[7] J. Leskovec and C. Faloutsos, "Sampling from large graphs," in *Proceedings of the 12th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*. ACM, 2006, pp. 631–636.

[8] T. Hagerup and C. Rüb, "A guided tour of chernoff bounds," *Information Processing Letters*, vol. 33, no. 6, pp. 305–308, 1990.

[9] A. L. Fred and A. K. Jain, "Combining multiple clusterings using evidence accumulation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 835–850, 2005.

[10] B. N. d. Santos, R. G. Rossi, and R. M. Marcacini, "Transductive event classification through heterogeneous networks," in *23rd Brazillian Symposium on Multimedia and the Web*. ACM, 2017, pp. 285–292.

[11] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," *Advances in Neural Information Processing Systems NIPS*, vol. 16, no. 16, pp. 321–328, 2004.

[12] M. Belkin, I. Matveeva, and P. Niyogi, "Regularization and semi-supervised learning on large graphs," in *Proceedings of the 17th Conference on Learning Theory*. Springer, 2004, pp. 624–638.

[13] X. Zhu, "Semi-supervised learning literature survey," Computer Sciences, University of Wisconsin-Madison, Tech. Rep. 1530, 2005.

[14] Y. Wu, N. Cao, D. Archambault, Q. Shen, and W. Cui, "Evaluation of graph sampling: A visualization perspective," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 401–410, 2016.

[15] R. Mall, R. Langone, and J. Suykens, "Kernel spectral clustering for big data networks," *Entropy*, vol. 15, no. 5, pp. 1567–1586, 2013.

[16] L. Berton, D. A. Vega-Oliveros, J. C. Valverde-Rebaza, A. T. da Silva, and A. de Andrade Lopes, "The impact of network sampling on relational classification." in *Int. Conf. on Information Management and Big Data (SIMBig)*, 2016, pp. 62–72.

[17] M. Al Hasan, N. K. Ahmed, and J. Neville, "Network sampling: Methods and applications," in *KDD'2013 Tutorials*, 2013.

[18] B. D. Hughes, *Random walks and random environments*. Clarendon Press. Oxford University Press, 1995.

[19] X. Lu and S. Bressan, "Sampling connected induced subgraphs uniformly at random," in *Int. Conf. on Scientific and Statistical Database Management*. Springer, 2012, pp. 195–212.

[20] L. A. Goodman, "Snowball sampling," *The Annals of Mathematical Statistics*, pp. 148–170, 1961.

[21] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graphs over time: densification laws, shrinking diameters and possible explanations," in *Proceedings of the eleventh ACM SIGKDD Int. Conf. on Knowledge Discovery in Data Mining*. ACM, 2005, pp. 177–187.

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

[23] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7.