The impact of network sampling on relational classification

Lilian Berton[2] **Didier A. Vega-Oliveros**[1] **Jorge Valverde-Rebaza**[1]
Andre Tavares da Silva[2] **Alneu de Andrade Lopes**[1]

[1]Department of Computer Science
ICMC, University of São Paulo
CEP 13560-970, São Carlos - SP - Brazil
{davo,jvalverr,alneu}@icmc.usp.br

[2]Technological Sciences Center
University of Santa Catarina State
CEP 89219-710, Joinville - SC - Brazil
lilian.2as@gmail.com, andre.silva@udesc.br

Abstract

Many real-world networks, such as the Internet, social networks, biological networks are massive in size, which difficult different processing and analysis tasks. For this reason, it is necessary to apply a sampling process to reduce the network size without losing relevant network information. In this paper, we propose a new and intuitive sampling method based on exploiting the following centrality measures: degree, k-core, clustering, eccentricity and structural holes. For our experiments, we delete 30% and 50% of the vertices from the original network and evaluate our proposal on six real-world networks on relational classification task using six different classifiers. Classification results achieved on sampled graphs generated from our proposal are similar to those obtained on the entire graphs. In most cases, our proposal reduced the original graphs by up to 50% of its original number of edges. Moreover, the execution time for learning step of the classifier is shorter on the sampled graph.

keywords: network sampling, relational classification, centrality measures, complex networks

1 Introduction

Networks are relational structures with a high level of order and organization, despite they display big inhomogeneities (Fortunato, 2010). Furthermore, networks are extremely useful as a representation of a wide variety of complex systems in a lot of real-world contexts, such as social, information, biological and technological domains (Newman, 2010). Formally, a network is denoted by a graph $G = (V, E)$, where V is the set of vertices representing objects in a specific context, and E is the set of edges representing the interactions among these objects. For instance, in a social network, vertices are individuals and edges are the friendships existing among them (Newman, 2010).

Since analyzing and modeling data in relational representation is relevant for different domains, several applications have been studied to obtain more benefits from the network structure, such as community detection (Valejo et al., 2014), link prediction (Valverde-Rebaza and Lopes, 2013; Valverde-Rebaza and Lopes, 2014; Valverde-Rebaza et al., 2015), topic extraction (Faleiros and Lopes, 2015), information diffusion (Vega-Oliveros and Berton, 2015; Vega-Oliveros et al., 2015), and others. Recently, there has been a lot of interest in relational learning, especially related to relational classification techniques (Lu and Getoor, 2003; Macskassy and Provost, 2003; Macskassy and Provost, 2007; Lopes et al., 2009). Relational classifiers have shown best performance than conventional classifiers (Valverde-Rebaza et al., 2014).

However, most of these networks are massive in size, being difficult to be studied in their entirety. In some cases, the network is not totally available, or it is hard to be collected, or even if we have the complete graph, it can be very expensive to run the algorithms on it. Hence, it is necessary to perform and study on network sampling, i.e., selecting a subset of vertices and edges from the full graph, in such way we obtain $G' = (V', E') \in G = (V, E)$ (Leskovec and Faloutsos, 2006; Ahmed et al., 2012; Ahmed et al., 2013).

Considering the assumption that network data fits in memory is not realistic for many real-world domains (*e.g.*, online social networks), different strategies for sampling have been proposed aiming to reduce the number of vertices or edges

of a network. The state-of-the-art technique is called as *random subsampling* method, which select nodes uniformly at random. However, while this technique is intuitive and relatively straightforward, it does not accurately capture properties of networks with power-law degree distributions (Stumpf et al., 2005). To cope with this problem, researchers have also considered other sampling methods based on breadth-first search or random walks. *Snowball sampling* method, for instance, adds nodes and edges using breadth-first search from a randomly selected seed node for accurately maintaining the network connectivity within the snowball (Lee et al., 2006). On the other hand, the *Forest Fire Sampling* (FFS) method uses partial breadth-first search where only a fraction of neighbors are followed for each node (Leskovec and Faloutsos, 2006), and the *degree-based sampling* method selects nodes considering their probabilities to be visited, which is proportional to the node degree (Adamic et al., 2001).

Other techniques have been proposed in the literature, which consider the different sources (*e.g.*, disk-resident/static or streaming) and scale (*e.g.*, small or large). Although there is previous work focusing on evaluating the performance of sampling methods by comparing network statistics, *i.e.* measure the representativeness of the sampled subgraph structure comparing it with the full input network structure (Ahmed et al., 2012; Ahmed et al., 2013), to the best of our knowledge there is no extensive research focused on study the impact of using sampled networks in a specific machine learning task, such as, classification, exploiting a lot of classifiers and datasets. Thus, in this paper, we propose an intuitive sampling method and use different configurations of it to perform an empirical evaluation in six real-world networks and six classifiers. We evaluate the quality of our proposal analyzing: i) how much the full network structure is preserved in the sampled graphs generated, ii) the accuracy obtained by six relational classifiers on entire and sampled graphs; and iii) the execution time in the learning step of the classifiers.

The main contributions of this paper are: i) we propose a new and intuitive method for sampling based on centrality properties of networks, such as, node degree, *k*-core, and others. These measures can be calculated in only part of the graph and have low computational cost; ii) we perform an empirical evaluation that shows the relational classification accuracy obtained by different sampled graphs generated from our proposal is as good as the classification accuracy obtained on entire graphs and taking less time in the learning phase; iii) we also analyze the network topology to exploit which cases the sampled graphs are similar to full graphs.

The remaining of this paper is organized as follows. Section 2 presents some concepts used in the paper encompassing centrality measures and relational classification. Section 3 presents the proposed approach for network sampling. Section 4 presents the experimental evaluation which analyzes the impact of sampling on relational classification and on the network topology. Finally, Section 5 presents the conclusions and future works.

2 Background

In this section, we describe the main centrality measures used as conventional parameters in different sampling methods existent in the literature. Also, we introduce briefly the main concepts on relational classification and six of the most popular relational classifiers.

2.1 Centrality measures

In complex network, some researchers have proposed different measures to analyze the importance of central vertices (Newman, 2010; Dorogovtsev and Mendes, 2002). The centrality measures indicate how much a vertex is important in some scope. Considering that, $n = |V|$ and $m = |E|$, the centrality measures applied in this work are described as follow.

- **Degree (DG)**: The degree or connectivity of vertex i, referred to k_i, is related with the number of edges or connections that go (k_i^{out}) or arrive (k_i^{in}) to vertex i. The *average degree* $\langle k \rangle$ for directed networks is the average of the input or output edges. When the network is undirected, the average degree is the factor $\langle k \rangle = 2 * m/n$, i.e., the sum of all the edges per vertex of the network over the number of vertices. The k_i values can be calculated as follow:

$$k_i = \sum_{i \in N} a_{ij} \,. \tag{1}$$

  Vertices with very high k_i values are called *hubs*, which represent instances strongly connected that impact on the dynamics of the network (Barabasi and Bonabeau, 2003;

Vega-Oliveros and Berton, 2015). For instance, in social networks, hubs are the most popular individuals, like famous actors, politicians, etc. The time complexity for calculating to all the vertices is $O(n * \langle k \rangle)$.

- **K-core (KC)**: The network can be decomposed in terms of sub-networks or cores (Seidman, 1983), where each core of order (H_k) represents the set of vertices that has $k_i \geq k$. Therefore, a vertex i belongs to $Kc(x) = k$ if H_k is the largest core it can be part (Seidman, 1983). The principal core is the set of vertices with the largest k-core value, and they are the most central (Kitsak et al., 2010). In general, vertices with lower KC values are located at the periphery of the network. The KC centrality is obtained by an iterative and incremental process (Batagelj and Zaversnik, 2003) that begins with $k = 1$: (*i*) All the vertices with degree lower or equal than k are removed. Then, (*ii*) the remaining vertices are evaluated several times, in order to remove those with k_i lower or equal than k. After that, (*iii*) the removed vertices are part of the set $Kc(i) = k$, k is incremented, and the process continues with step (*i*). The final set of vertices is the main core of the network, which has the largest KC centrality. Notice that not necessarily the hubs have the highest k-core values. For instance, hubs located in the periphery have small k-core centrality (Kitsak et al., 2010). The algorithm has low computational complexity $O(n + m)$ for calculating the centrality to all vertices.

- **Clustering coefficient (CT)**: In topology terms, it is the presence of triangles (cycles of order three) in the network. The clustering coefficient (Watts and Strogatz, 1998) of a vertex i is defined as the number of triangles centered on i over its maximum number of possible connections, i.e.,

$$CT_i = \frac{2e_i}{k_i(k_i - 1)}. \tag{2}$$

  In the case of $k_i \in \{0, 1\}$, it is assumed a centrality value of zero, and $CT_i = 1$ only if all the neighbors of i are interconnected. The running time complexity of the measure is $O(n * \langle k \rangle^2)$.

- **Eccentricity (EC)**: The shortest path between two vertices is the shortest sequence of edges that connect them, and the distance is the number of edges contained in the path. This problem can be resolved by employing different algorithms, like Dijkstra, Bellman-Ford, Floyd-Warshall, or breadth-first search methods (Cormen et al., 2009). In the case that i and j belong to different components, it is assumed that $\ell_{ij} = n$. In this way, the eccentricity value of a vertex i is the largest distance over all the shortest path to the other vertices, as follow:

$$EC_i = \max_{i \neq j} \{|\ell_{ij}|\} , \tag{3}$$

  where $|\ell_{ij}|$ is the distance of the shortest path between vertices i and j. This measure evaluates how close is a vertex to its most distant vertex. Lower values of EC indicates that the vertex is more central and closer to the others. Therefore, vertices located at the network center have the lowest eccentricity values. For unweighted graphs, the running time complexity of this measure is $O(n * m)$.

- **Structural Holes (HO)**: Some vertices in the network work such as the bridge of clusters or other vertices, and if they are removed a structural hole will occur. The structural hole vertices act as spanners among communities or groups of vertices without direct connections. These individuals are important to the connectivity of local regions. We calculate Burt's constraint scores (Burt, 1992) as the structural holes centrality. The algorithm considers all vertices as ego networks, where connections no related to it have not a direct effect. For each vertex, the score is the fraction of isolated holes will exists associated with it and according to its ego network. The higher the fraction of structural holes associated with the vertex, the more central it is. Therefore, vertices with higher degree centrality tend to have low HO values, given that its ego networks are larger and more densely interconnected, and this diminishes the fraction of isolated holes. The time complexity for calculating the measure to all the vertices is $O(n + n * \langle k \rangle^2)$.

2.2 Relational classification

Conventional classification algorithms learn from a training set formed by independent and identically distributed (i.i.d) data (Mitchell, 1997). Nevertheless, as previously mentioned, a lot of real-world data are relational in nature and can be represented by graphs. Conventional classifiers do not work properly on graphs because they ignore pairwise dependency relations between vertices, *i.e.* relational information. To cope with that, different relational classifiers have been proposed (Lu and Getoor, 2003; Macskassy and Provost, 2003; Macskassy and Provost, 2007; Lopes et al., 2009). Relational classifiers require a fully described graph (vertices and edges) with known labels for some of the vertices to predict the labels of the remaining vertices.

For the domain of relational classification, we redefine the network as the graph $G = (V, E, W)$, where $V = \{v_1, v_2, \ ... \ v_n\}$ is the set of n vertices that describes an object, $E = \{e_1, e_2, \ ... \ e_m\}$ is the set of m edges representing some similarity between a pair of vertices and W is a matrix of weights, which associates to each edge a weight w_{ij} that determines the strength of the connection. For this work, we consider three relational classifiers: weighted vote relational neighbor (wvrn), network-only Bayes (no-Bayes) and network-only link-based (no-lb).

The wvrn classifier estimates class membership probabilities by assuming that linked nodes tend to belong to the same class and considering the weighted mean of the class-membership probabilities for the neighborhood of each node analyzed (Macskassy and Provost, 2007) according to Equation 4.

$$P(v_i = c|N_i) = \frac{1}{N} \sum_{v_j \in N_i} w(v_i, v_j) P(v_j = c|N_j) \tag{4}$$

The no-Bayes classifier employs multinomial naïve Bayes classifier based on the classes of the neighborhood of each vertex (Macskassy and Provost, 2007). The no-Bayes is defined as Equation 5,

$$P(v_i = c|N_i) = \frac{P(N_i|c)P(c)}{P(N_i)} \tag{5}$$

where $P(N_i|c) = \frac{1}{N} \prod_{v_j \in N_i} P(v_j = c_j|v_i = c)^{w(v_i, v_j)}$.

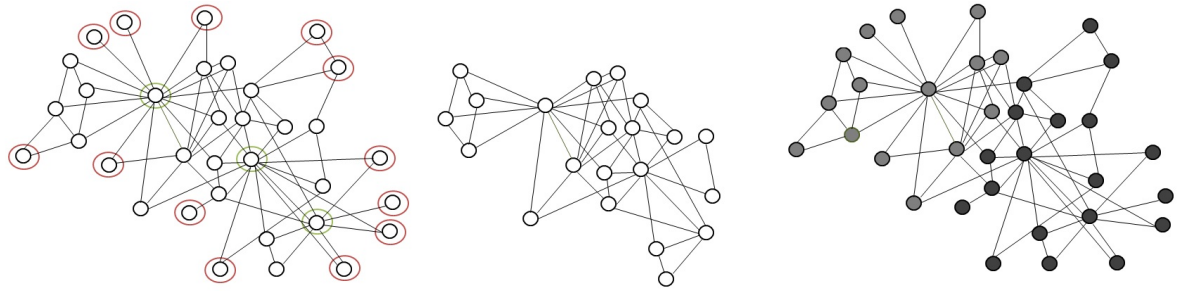
Furthermore, these two relational classifiers use the relaxation label as a collective inference method. The no-lb classifier creates a feature vector for a vertex by aggregating the labels of its neighborhood and then uses logistic regression to build a discriminative model based on those feature vectors (Lu and Getoor, 2003). This learned model is then applied to estimate $P(v_i = c|N_i)$. For no-lb classifier, three aggregation methods have been considered: binary-link (no-lb-binary), mode-link (no-lb-mode), and count-link (no-lb-count). Another aggregation method considered is class-distribution link (no-lb-distrib) (Macskassy and Provost, 2007). All the no-lb aggregations use the iterative classification as a collective inference method.

3 Proposal

As previously mentioned, our proposal consists in an intuitive approach based on exploring the centrality measures of a network to remove some vertices and edges trying to conserve the equivalence between the sampled and the entire network. We aim to obtain a sample from G in such way it does not affect the performance of any learning task. Thus, our proposal generates a sample G' from G, *i.e.* $G' = \sigma(G)$, where σ is the function representing our proposal. It is important to note that G' is a sub-graph from G, so $V' \subset V$ and $E' \subset E$. The size of the sample is relative to the graph size.

The proposed approach is illustrated in Figure 1 and follows these steps: 1) calculate a specific centrality measure for all vertices of the network, in this paper we use DG, KC, CT, EC, HO measures; 2) select some percentage of vertices with the highest (H) or lowest (L) centrality values, in this paper we experiment selecting 30% and 50% of vertices; 3) remove all selected vertices and all their corresponding edges from G, obtaining G'. The sampled graph generated, G', should be equivalent to the entire graph, so learning algorithms should have a similar performance in both the sampled and the entire graph.

All measures used for sampling the graph can be calculated considering only a fraction of the graph, in a direct way or by employing statistical methods. The measures DG, HO and CT, for example, can be calculated for each vertex directly. In the case of EC and KC, there are very precise approaches that consider only the vertex community (part of the network). These measures have low computational cost to be calculated and can be applied on very large networks, moreover, by



(a) (b) (c)

Figure 1: Proposed approach: (a) Select % of vertices with lowest (red) or highest (green) centrality measure values from original graph. (b) Remove all the vertices and all their edges to obtain the sampled network. In this case, we remove vertices with lowest centrality measure values. (c) Use some learning task on the sampled network, for instance, the relational classification.

the experimental results achieved good accuracy.

4 Experimental results

In this section, we present extensive empirical experiments focused on evaluating the quality of sampled graphs generated by different configurations of our proposal, when compared with the original graph. We use six real-world networks and apply six relational classifiers (see Section 2.2) on full and sampled graphs. We perform two types of evaluations, Section 4.2 shows the classification accuracy results, and Section 4.3 shows the topological analysis of sampled and original graphs.

4.1 Data sets and experimental setup

We consider six benchmark data sets[1], which represent real networks and are described in Table 1. We consider that all networks are undirected.

We sample a subgraph G' from a graph G using the centrality measures presented and considering 30% and 50% of vertices with smallest and highest centralities values. For each sample size, we perform 10-fold cross validation and applied the following relational classifiers: weighted vote relational neighbor (wvrn), network-only Bayes (no-Bayes), and network-only link-based (no-lb) classifiers, in their Netkit-SRL implementations with standard configuration. For the network-only link-based classifier we employed models modelink (no-lb-mode), count-link (no-lb-count), binary-link (nolb-binary) and class-distribution-link (no-lb-distrib). The area under the ROC curve (AUC) was used as evaluation measure to compare the accuracy of graph G and sampled graph G'.

[1] http://netkit-srl.sourceforge.net/data.html

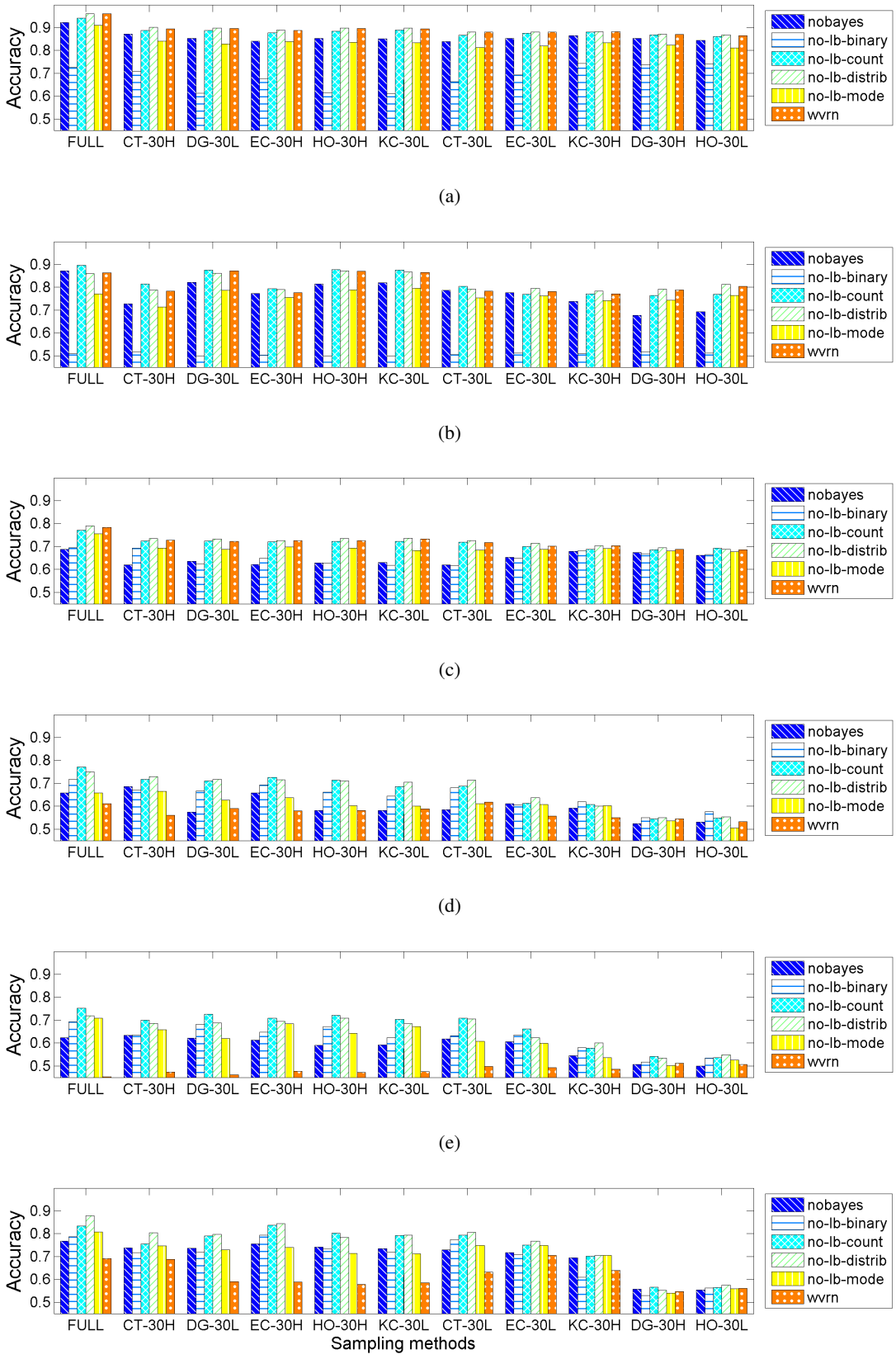
Table 1: Data sets description.

Datasets	$\|V\|$	$\|E\|$	**# Classes**	$\langle k \rangle$
Cora	4240	35912	7	17.84
Cornell	351	1393	6	3.98
Imdb	1441	51481	2	66.99
Industry	2189	6531	12	10.65
Texas	338	1002	6	3.44
Washington	434	1941	6	4.41

4.2 Impact of sampling on classification accuracy

The classification results for the entire graph, 30% and 50% of the sampled networks are shown in Figures 2 and 3 respectively, with the accuracy for the six datasets (Figures (a), (b), (c), (d), (e) and (f)), the six classifiers (bars) and the ten sampling proposed strategies moreover the classification with the entire graph (FULL). For all the sampling strategies the datasets Cora and Imdb achieved the highest accuracy. And the better classifiers, in general, was nolb-lb-count and nolb-lb-distrib.

The Nemenyi post-hoc test (Demšar, 2006) was executed to verify the possibility of detecting statistical differences among the sampling strategies. The results for 30% and 50% of sampled networks are shown in Figures 4 and 5 respectively. On the top of the diagrams is the critical difference (CD) and in the axis are plotted the average ranks of the evaluated techniques, where the lowest (best) ranks are on the left side. When the methods analyzed have no significant difference, they are connected by a black line in the diagram.



(a)

(b)

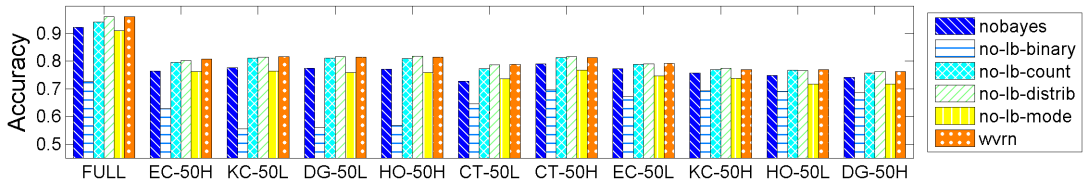
(c)

(d)

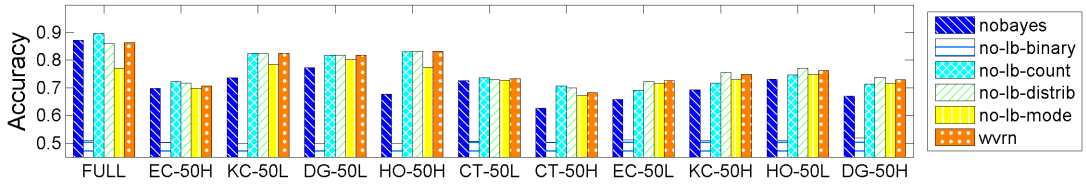
(e)

(f)

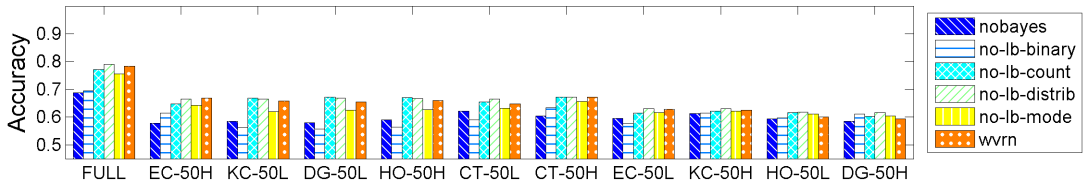
Figure 2: Classification results for the entire graph (FULL) and sampling strategies that remove 30% of vertices for the following datasets: (a) Cora, (b) Cornell, (c) Imdb, (d) Industry, (e) Texas and (f) Washington.



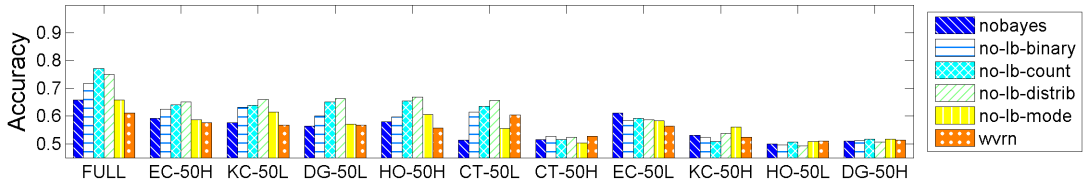
(a)



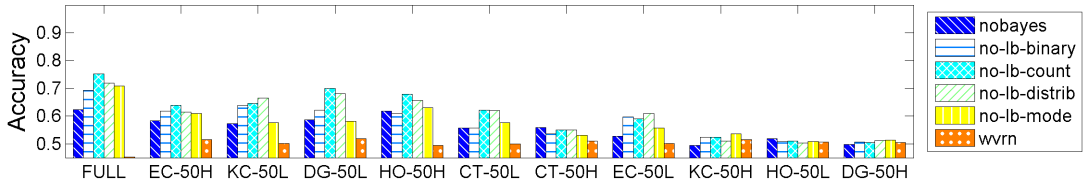
(b)



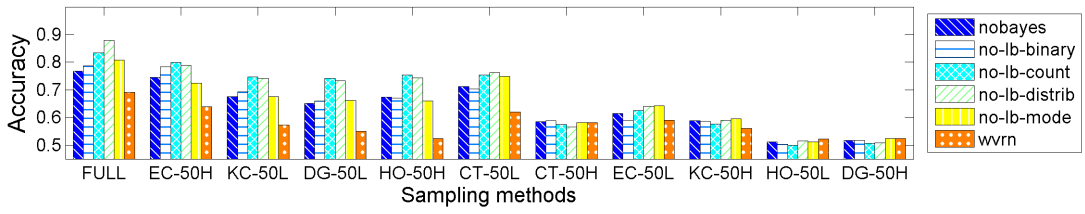
(c)



(d)



(e)



(f)

Figure 3: Classification results for the entire graph (FULL) and sampling strategies that remove 50% of vertices for the following datasets: (a) Cora, (b) Cornell, (c) Imdb, (d) Industry, (e) Texas and (f) Washington.

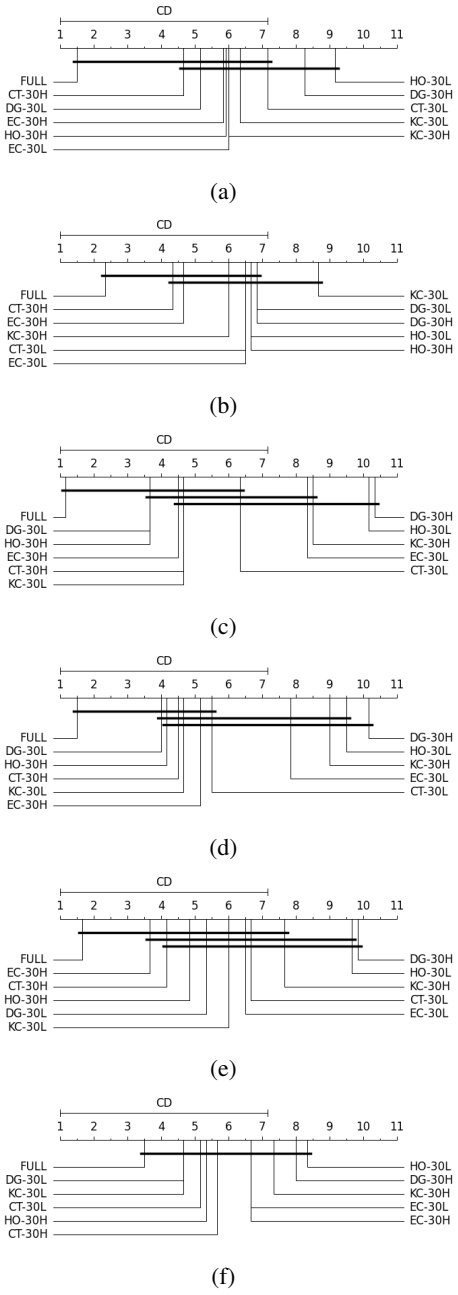


Figure 4: Nemenyi post-hoc test for the entire graph and sampling strategies that removes 30% of vertices for the following relational classifiers: (a) no-Bayes, (b) no-lb-binary, (c) no-lb-count, (d) no-lb-distrib, (e) no-lb-mode and (f) wvrn.

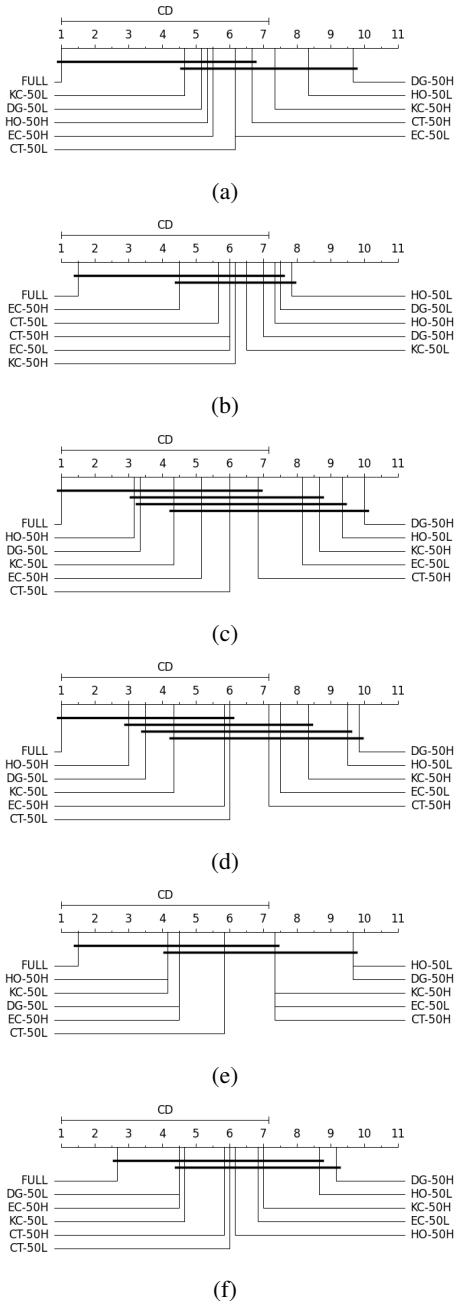


Figure 5: Nemenyi post-hoc test for the entire graph and sampling strategies that removes 50% of vertices for the following relational classifiers: (a) no-Bayes, (b) no-lb-binary, (c) no-lb-count, (d) no-lb-distrib, (e) no-lb-mode and (f) wvrn.

According to the Nemenyi statistics, the critical value for comparing the average-ranking of two different algorithms considering the sampling strategy that removes 30% of vertices (Figure 4) or 50% of vertices (Figure 5) at 95 percentile in all classifiers (no-Bayes, nolb-binary, no-lb-count, no-lb-distrib, no-lb-mode, wvrn) is 6.16.

In all the classifiers there are some sampling strategies that have no statistical difference with the entire graph. It is the case of CT-30H, CT-30L, EC-30H and HO-30H for 30% of vertices sampled, and CT-50L, EC-50H, HO-50H, KC-50L, and DG-50L for 50% of vertices sampled. In particular, the CT-50H only had significance difference with the no-lb-distrib classifier. In terms of accuracy, this result indicates that the CT centrality, for all the analyzed parameters, was more robust and suitable as a sampling strategy.

Table 2: Execution time (ms) for classification training models.

Datasets	Classifiers	CT-30H	EC-30H	HO-30H	KC-30H	DG-30H	FULL
CoRa	no-Bayes	756	824	960	552	**511**	1395
	nolb-binary	13498	13398	13857	13380	**13313**	14547
	no-lb-count	**12213**	12585	13897	12386	12765	13759
	no-lb-distrib	**14218**	14310	14569	14452	14401	15396
	no-lb-mode	13991	13837	14097	13304	**12924**	17516
	wvrn	552	622	713	403	**378**	1032
Cornell	no-Bayes	74	52	52	50	**39**	228
	nolb-binary	724	775	789	**694**	**694**	936
	no-lb-count	718	809	754	697	**667**	1250
	no-lb-distrib	753	759	772	**690**	719	918
	no-lb-mode	729	703	726	**690**	712	2018
	wvrn	38	49	45	**31**	36	145
Imdb	no-Bayes	381	327	558	**214**	175	1042
	nolb-binary	959	810	1260	659	**587**	1538
	no-lb-count	998	843	1258	693	**605**	1786
	no-lb-distrib	947	868	1232	662	**595**	1484
	no-lb-mode	970	830	1086	621	**600**	2720
	wvrn	482	387	695	253	**214**	1050
Industry	no-Bayes	529	635	744	314	**290**	2841
	nolb-binary	23140	**21688**	22061	26874	26101	23196
	no-lb-count	**23751**	23934	27333	25806	26370	27111
	no-lb-distrib	27827	26984	**26721**	29931	28834	28148
	no-lb-mode	27854	25516	**24045**	28688	27881	31027
	wvrn	281	323	346	187	**180**	541
Texas	no-Bayes	54	47	52	44	**40**	206
	nolb-binary	809	730	703	680	**601**	914
	no-lb-count	826	773	765	703	**605**	1125
	no-lb-distrib	766	681	701	680	**633**	895
	no-lb-mode	657	642	666	674	**664**	2283
	wvrn	42	38	42	**37**	40	153
Washington	no-Bayes	68	72	66	45	**42**	277
	nolb-binary	967	963	950	888	**804**	1016
	no-lb-count	1016	977	957	899	**868**	1124
	no-lb-distrib	1043	955	945	880	**794**	1190
	no-lb-mode	870	948	877	897	**834**	2300
	wvrn	51	54	80	**44**	46	218

Table 2 shows the time comparison for the learning step for all classifiers and all datasets. We notice that all sampling strategies proposed, considering 30% of sampling, achieve small time compared with the original graph, especially the strategy DG and KC. The lowest times are in bold.

4.3 Impact of sampling on network topology

We have analyzed the impact of the sampling methods in the structure of the original network. In Table 3, we have the fraction of remaining edges after applying the sampling methods, according to the target vertices (with highest (H) or lowest (L) centrality value) and removal percentage (30 or 50%). The bold values highlight the techniques and parameters that achieve similar accuracy results to the full network, i.e., with no significance difference for all the classifiers. We have observed that the EC and HO measures are inversely proportional to the final fraction of remaining edges. This occurs since for the EC, the most central or closest vertices have the lowest values and for the HO measure, hubs tend to have larger ego-networks; ergo, the centrality values are lower.

We notice that there exists diverse values of removed edges from the original network, without strongly affecting the accuracy of the classifiers (in bold). This variation of removed edges, some larger than 50%, suggest that depending on the expected requirements, it can be privileged in the sampling process:

1. The maximal removal of edges by removing a low proportion of vertices.

2. Equivalent removal proportion of edges and vertices.

3. The minimal removal of edges by removing a high proportion of vertices.

In the first case, by removing 30% of vertices we have the sampling method CT-30H. For the third case, we have the methods DG-50L, KC-50L, and HO-50H. The left bold sampling strategies are in the second case.

Notwithstanding reducing the number of vertices and edges from the original network do not statistically impact the classification results, the topological properties are sensibly affected by the removal. For instance, removing 30% of vertices with the highest degree centrality (k_i) it produces a more homogeneous distributed network (tending to a Poisson or regular graph) and the average degree decays. On the other hand with the same proportion, removing the least connected vertices produce networks with more heterogeneous degree distribution than the original graph.

Table 3: Descriptions of edges on sampled graphs.

Datasets	Measures	#edges 30H	#edges 30L	#edges 50H	#edges 50L
CoRa	DG	0.216	0.916	0.077	**0.780**
	KC	0.266	0.918	0.093	**0.785**
	HO	**0.912**	0.231	**0.767**	0.090
	EC	**0.718**	0.389	**0.477**	0.191
	CT	**0.555**	**0.600**	0.293	**0.368**
Cornell	DG	0.067	0.853	0.017	**0.666**
	KC	0.140	0.849	0.040	**0.670**
	HO	**0.853**	0.080	**0.659**	0.020
	EC	**0.662**	0.326	**0.479**	0.114
	CT	**0.500**	**0.702**	0.051	**0.553**
Imdb	DG	0.226	0.881	0.069	**0.648**
	KC	0.284	0.881	0.086	**0.653**
	HO	**0.872**	0.252	**0.637**	0.096
	EC	**0.499**	0.347	**0.264**	0.146
	CT	**0.601**	**0.553**	0.314	**0.290**
Industry	DG	0.079	0.952	0.029	**0.883**
	KC	0.090	0.951	0.039	**0.887**
	HO	**0.952**	0.094	**0.882**	0.035
	EC	**0.758**	0.354	**0.354**	0.152
	CT	**0.575**	**0.934**	0.203	**0.438**
Texas	DG	0.083	0.835	0.026	**0.634**
	KC	0.182	0.829	0.070	**0.643**
	HO	**0.835**	0.083	**0.626**	0.034
	EC	**0.685**	0.529	**0.492**	0.196
	CT	**0.474**	**0.720**	0.085	**0.559**
Washington	DG	0.082	0.857	0.017	**0.680**
	KC	0.178	0.863	0.074	**0.692**
	HO	**0.855**	0.084	**0.685**	0.028
	EC	**0.724**	0.268	**0.524**	0.114
	CT	**0.467**	**0.745**	0.081	**0.616**

5 Conclusion

In this paper, we proposed a strategy for network sampling by exploring five centrality measures: DG, KC, CT, EC, HO and eliminating vertices with 30% or 50% of lowest or highest centrality values. All centrality measures considered have a low order of complexity and are computationally applicable in real networks scenarios. Moreover, they can be calculated in part of the graph.

The proposed approach reduces the original graph in 50% or even more and the accuracy results remain statistically similar to the obtained with the entire network, i.e. the impact on classification results obtained by entire networks is minimal when compared with those obtained by sampled networks. We have applied the proposed strategy in six real networks considering six different relational classifiers. The CT measure was the most robust in accuracy for all classifiers and on all networks, without statistical significance. Moreover, the execution time for the learning step of the classifiers are smaller in the sampling strategies proposed when compared with the entire graph.

Acknowledgments

This work was partially supported by the São Paulo Research Foundation (FAPESP) grants: $2013/12191-5$ and $2015/14228-9$, National Council for Scientific and Technological Development (CNPq) grants: $302645/2015-2$ and $140688/2013-7$, and Coordination for the Improvement of Higher Education Personnel (CAPES).

References

L.A. Adamic, R.M. Lukose, A.R. Puniyani, and B.A. Huberman. 2001. Search in power-law networks. *Physical Review E*, 64(046135).

Nesreen K. Ahmed, Jennifer Neville, and Ramana Kompella. 2012. Network sampling designs for relational classification. In *In Proceedings of the 6th International AAAI Conference on Weblogs and Social*.

N.K. Ahmed, J. Neville, and T. Kompella. 2013. Network sampling: From static to streaming graphs. *ACM Trans. Knowl. Discov. Data*, 8(2):1–56.

A. Barabasi and E. Bonabeau. 2003. Scale-free networks. *Scientific American*, pages 50–59.

V. Batagelj and M. Zaversnik. 2003. An O(m) algorithm for cores decomposition of networks. *Arxiv preprint cs/0310049*.

R.S. Burt. 1992. *Structural holes: The social structure of competition*. Harvard University Press, Cambridge, MA.

T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. 2009. *Introduction to Algorithms*. The MIT Press, 3 edition.

J. Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *JMLR*, 7:1–30.

S. N. Dorogovtsev and J. F. F. Mendes. 2002. Evolution of networks. In *Adv. Phys*, pages 1079–1187.

T. Faleiros and A. Lopes. 2015. Bipartite graph for topic extraction. In *IJCAI 2015*, pages 4363–4364.

S. Fortunato. 2010. Community detection in graphs. *CoRR*, abs/0906.0612v2.

M. Kitsak, L.K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H.E. Stanley, and A. Makse. 2010. Identification of influential spreaders in complex networks. *Nature Physics*, 6(11):888–893.

S. Lee, P. Kim, and H. Jeong. 2006. Statistical properties of sampled networks. *Physical Review E*, 73(016102).

J. Leskovec and C. Faloutsos. 2006. Sampling from large graphs. SIGKDD 2006, pages 631–636.

A. Lopes, J.R. Bertini, R. Motta, and L. Zhao. 2009. Classification based on the optimal k-associated network. In *Complex Sciences*, volume 4 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunivations Engineering*, pages 1167–1177. Springer Berlin Heidelberg.

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

S.A. Macskassy and F.J. Provost. 2003. A simple relational classifier. In *2nd Workshop on Multi-Relational Data Mining*.

S.A. Macskassy and F.J. Provost. 2007. Classification in networked data: A toolkit and a univariate case study. *JMLR*, 8:935–983.

T.M. Mitchell. 1997. *Machine Learning*. McGraw-Hill, New York.

M. E. J. Newman. 2010. *Networks: an introduction*. Oxford University Press.

S. Seidman. 1983. Network structure and minimum degree. *Social networks*, 5(3):269–287.

M. Stumpf, C. Wiuf, and R. May. 2005. Subnets of scale-free networks are not scale-free: Sampling properties of networks. *Proceedings of the National Academy of Sciences*, 102:4221–4224.

A. Valejo, J. Valverde-Rebaza, and A. Lopes. 2014. A multilevel approach for overlapping community detection. BRACIS 2014, pages 390–395. IEEE.

J. Valverde-Rebaza and A. Lopes. 2013. Exploiting behaviors of communities of Twitter users for link prediction. *Social Network Analysis and Mining*, pages 1–12.

J. Valverde-Rebaza and A. Lopes. 2014. Link prediction in online social networks using group information. In *ICCSA 2014*, volume 8584, pages 31–45.

J. Valverde-Rebaza, A. Soriano, L. Berton, M.C.F. de Oliveira, and A. Lopes. 2014. Music genre classification using traditional and relational approaches. BRACIS 2014, pages 259–264. IEEE.

J. Valverde-Rebaza, A. Valejo, L. Berton, T. Faleiros, and A. Lopes. 2015. A naïve bayes model based on overlapping groups for link prediction in online social networks. In *ACM SAC' 15*, pages 1136–1141.

D. Vega-Oliveros and L. Berton. 2015. Spreader selection by community to maximize information diffusion in social networks. In *SIMBig 2015*, pages 73–82.

D. Vega-Oliveros, L. Berton, A. Lopes, and F. Rodrigues. 2015. Influence maximization based on the least influential spreaders. In *SocInf 2015, co-located with IJCAI 2015*, volume 1398, pages 3–8.

D.J. Watts and S. Strogatz. 1998. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442.