Quantification in Social Networks

Letizia Milli*†, Anna Monreale*†, Giulio Rossetti*†, Dino Pedreschi*, Fosca Giannotti† and Fabrizio Sebastiani†

*University of Pisa, Italy Email: {milli,rossetti,annam,pedre}@di.unipi.it

†ISTI-CNR, Pisa Italy Email: {name.surname}@isti.cnr.it

***Abstract*—In many real-world applications there is a need to monitor the distribution of a population across different classes, and to track changes in this distribution over time. As an example, an important task is to monitor the percentage of unemployed adults in a given region. When the membership of an individual in a class cannot be established deterministically, a typical solution is the classification task. However, in the above applications the final goal is not determining which class the individuals belong to, but estimating the prevalence of each class in the unlabeled data. This task is called *quantification*. Most of the work in the literature addressed the quantification problem considering data presented in conventional attribute format. Since the ever-growing availability of web and social media we have a flourish of network data representing a new important source of information and by using quantification network techniques we could quantify collective behavior, i.e., the number of users that are involved in certain type of activities, preferences, or behaviors. In this paper we exploit the homophily effect observed in many social networks in order to construct a quantifier for networked data. Our experiments show the effectiveness of the proposed approaches and the comparison with the existing state-of-the-art quantification methods shows that they are more accurate.**

I. Introduction

Many real-world applications require the estimation and monitoring of the distribution of a population across different classes. Sometime we also need to track the changes in this distribution that may derive from varying reasons. An example of such applications is the important task to determining the percentage (or "prevalence") of unemployed people across different geographical regions, or genders, or age ranges, or across different time periods. In the literature, this task has been called *quantification* [6], [7], [9], [25], [28].

Quantification is closely related to *classification*: however, the goal of classification is different, since in classification we are interested in correctly guessing the true class label of each single individual. Instead, in quantification we are interested in classifying our individuals with the goal of estimating the class prevalence where is not strictly necessary to classify correctly each single individual. Classification and quantification are different because, while a perfect classifier is also a perfect quantifier, not necessarily a good classifier is also a good quantifier. Indeed, a classifier that on the test set generates a similar number of misclassified items in the different classes is a good quantifier because the compensation of the misclassifications leads towards a perfect estimation of the class distribution.

Most of the work have been proposed in the literature for addressing the quantification problem taking into consideration data presented in conventional attribute format. Since the ever-growing availability of web and social media we have a flourish of networking data representing a new important source of information. In this paper we want to address the quantification problem in complex networks. The question that we want to answer in this paper is: *how can the quantification task be performed on data describing the relationship among the entities of the system?*

The impact of quantification techniques for networking data is potentially high: this because today we are witnessing an ever more effective dissemination of social networks and social media where people express their interests and disseminate information on their opinions, about their habits, and their wishes. The possibility to analyze and quantify the percentage of individuals with specific characteristics or a particular behavior could help the analysis of many social aspects. For example, analyzing Facebook or Google+, where people can set their education level we could estimate the level of education of a population. Similarly, we could determine the distribution of the political orientation or the geographical origin of the social network population.

Tools for network quantification enable the integration of the so-called *big data analytics* in the consolidated analytical process of the official statistics. An interesting application domain is the monitoring of unemployment by using quantifiers on big data.

In this paper we propose techniques for quantification on networks that exploit the homophily effect observed in many social networks [16], [23], [22]: people tend mostly to relate with others whom they share some interests, ideas or beliefs. Starting from this observation, as a first step our approaches divide of the original network into sub-networks in order to better bound homophily. In particular two different partitioning strategies will be analyzed: one exploiting community discovery while the other adopting the notion of ego-network. The former approach tries to estimate the class prevalence in a networked population taking into account the characteristics of communities composing the entire network and the class frequencies in each community. The latter conversely, partitions the network into ego-networks and tries to infer the class of each unlabeled node in the network by observing the class of its neighborhood.

Our extensive experiments show that our quantification methods for networks, especially that one based on ego-networks, enable more accuracy than existing state-of-the-art quantification methods.

The reminder of the paper is organized as follows. In

Section II we discuss the related works. Section III introduces some background notions and the quantification problem for network data. Section IV describes our methods for quantification in network data based on homophily property. In Section V we provide the complexity analysis for the introduced approaches. In Section VI we show the empirical evaluation of our methods and finally, Section VII concludes the paper.

II. Related Work

The earliest mention of the quantification problem is found in [14], where the task is called *counting*. However, only 10 years later, in 2005, quantification was firstly addressed as a well defined new data mining task [8], [7], [9]. In this series of papers, Forman proposes several quantification methods and an evaluation measure KLD (Kullback-Leibler divergence). Bella et al. [1] moving from those seminal works, later introduced probabilistic versions of Forman's methods.
Quantification has been applied to several domains. For example, [9] uses it to determine the prevalence of support-related issues in incoming telephone calls received at customer support desks, while [5] use it to estimate the prevalence of response classes in open-ended answers obtained in the context of market research surveys. [13] apply quantification for estimating the distribution of support for different political candidates within blog posts. Differently from all of the above, Xue and Weiss [28] use quantification with the goal of improving the accuracy of classification.

To the best of our knowledge [25] is the only work addressing the quantification problem in the context of networking data, where the goal is estimating class prevalence among a population of nodes in a network. The authors propose an approach, inspired by Forman's equation, which uses network connectivity information to forecast the distribution of binary labels (identified in the following as + and -) for a subset of unlabeled nodes. For each vertex i of the test set (i.e. all the unlabeled nodes of the network) they calculate

$$p(+) = \frac{p(i) - p(i|-)}{p(i|+) - p(i|-)} \tag{1}$$

where $p(i)$ identifies the probability for a generic node v of establishing a link with i, while $p(i|-)$ and $p(i|+)$ denote the conditional probability for v, given its label (- or +), to be part of an edge with i. Once that $p(+)$ and $p(-)$ are computed for all the nodes two steps are performed for the quantification: (i) *Cleaning*: all the computed scores that do not belong to $[0, 1]$ are discarded; (ii) *Class Frequency Estimation*: for each class is returned as frequency estimation the *median* of the cleaned values. The major problem of the this approach is due to the choice of the median as frequency indicator: there is no assurance that the estimation provided for each class will return values that sum to 100%. In their experiments (concerning only datasets with a binary class label), the authors overcome such issue computing Eq. 1 only on a single class and defining the estimation for the second one as its complementary. Obviously, the results obtained by this method varies w.r.t. the initial choice of the class for which computing the median of the distribution. Moreover, due to this choice the applicability of their method is restricted only on a binary class scenario.

As will be explained in Section III-A, a straightforward solution to the quantification problem on networks could be the sampling; unfortunately, it does not capture the possible distribution drift. In the literature, many works have been proposed to understand the way to choose qualified samples applicable to a hidden population. These methods does not consider any information about the network structure. Usually, the approaches based on sampling have the form of chain referral sampling [11], [24]. However, the choice on how drawing initial random sample is still a key unsolved problem [4], [27].

Some studies focus on respondent driven sampling [12] for sampling design and population inference in social networks. The process exploits the social structure to expand the initial sample and reduce its independence on it.

A research field that is only apparently related to quantification is *collective classification* [21]. Similarly to quantification, here the classification of instances is not viewed in isolation. However, *collective classification* is radically different from quantification in that its focus is on improving the accuracy of classification by exploiting relationships between the objects to classify (e.g., hypertextual documents that link to each other). The accuracy of *collective classification* is evaluated at the individual level, rather than at the aggregate level as for quantification.

Quantification has also relations with *prevalence estimation from screening tests*, an important task in epidemiology ([19], [29]). A screening test is a test that a patient undergoes in order to check if s/he has a given pathology. Tests are often imperfect, i.e., they may give rise to false positives (the patient is incorrectly diagnosed with the pathology) and false negatives (the test wrongly diagnoses the patient to be free from the pathology). Therefore, testing a patient is akin to classifying a document, and using these tests for estimating the prevalence of the pathology in a given population is akin to performing quantification via classification. The main difference between this task and quantification is that a screening test typically has known and fairly constant recall (that epidemiologists call "sensitivity") and specificity (i.e., recall on the complement of the class), while the same usually does not happen for a classifier. Another related field in statistics is the "randomized response" methodology for conducting privacy-preserving tests, which uses a correction statistics similar to adjusted count post-processing [26].

III. Quantification for Networks

Quantification is closely related to classification, but their final goal differs. Quantification aims at finding the class frequencies in a set of unlabeled data, while classification aims at determining the class of each specific item in the same dataset. In other words, a quantifier does not care about perfectly predicting the class of a single item, but to guess the global trend of the classes in a new set of data.
The problem of quantification in the literature has been addressed considering data presented in conventional attribute format. Recently high-performing approaches based on decision tree variants[17] have been proposed in order to solve it in general contexts. Quantification is an important issue to tackle in order to understand and monitor user behaviors and activities by using social media and web data (i.e., *big data*) as the source of information.

In this paper we tackle the problem to estimate the frequency of each class in a network.

A. Problem Statement

We model the network as a indirect graph that denote by $G = (V, E, L)$, where V is the set of labelled nodes, L is a set of node labels and E is a set of edges, i.e. the set of pairs (u, v) where $u, v \in V$ are nodes. The node labels L represent the class values and a classifier f is a function $f : V \to L$ that assigns a class label $l_i \in L$ to each node $v_j \in V$.

The actual frequency of a class l_i with respect to a network $G = (V, E, L)$ is $freq_V(l_i) = \frac{|\{v_j \in V | v_j.class = l_i\}|}{|V|}$. The estimated frequency using the learnt function f, namely the result of the classifier, is $\widehat{freq_L}(l_i) = \frac{|\{v_j \in V | f(v_j) = l_i\}|}{|V|}$.

Given the set of nodes V, we denote by V_l the subset of labeled nodes while we use V_u to denote the set of unlabeled nodes. In the following sometime we also use "test set" to indicate V_u.

Note that in our setting the classifier is not learnt in an offline phase like typical *eager* learners. Our function f defined above, is an *instance-based* classifier, because it operates on the premises that classification of unknown instances can be done by relating the unknown to the known according to some specific relation between the two kind of instances.

We use the standard notation to indicate the set of *true positives* (TP), *false positives* (FP), *true negatives* (TN) and *false negatives* (FN) of a binary classifier. We use $tpr = \frac{TP}{TP+FN}$ to denote the *true positive rate* and $fpr = \frac{FP}{TN+FP}$ to denote the *false positive rate*.

Now we are ready to formally define the quantification on network problem.

Definition 1 (Network Quantification Problem): Let $L = \{l_1, l_2, \ldots, l_n\}$ be a set of class labels. Given a network $G = (V, E, L)$ and a partition of the nodes V in labeled V_l and unlabeled V_u the network quantification problem consists in finding a classifier f for the best estimation of the class label distribution in V_u, i.e., $\forall l_i \in L$ we want to minimize the difference between the actual frequency $freq_{V_u}(l_i)$ and the estimated one $\widehat{freq_{V_u}}(l_i)$.

The following example highlights the final goal of quantification comparing it with the classification.

Example 1: Consider the network in Figure 1(a) where colored nodes are those whose class values are known. Here, the real frequencies of the two classes are: $freq(A) = \frac{5}{11}$ and $freq(B) = \frac{6}{11}$. Suppose now to apply two different classifiers to predict the class label value of the unlabeled nodes. Figure 1(b) and Figure 1(c) show the result of the two classifiers, where the red dashed nodes represent the misclassified nodes. The percentage of correctly classified node is $\frac{2}{3}$ and $\widehat{freq}(A) = \frac{4}{11}$, so the total percentage of misclassified nodes is $\frac{1}{11}$. In Figure 1(c) we have two misclassified nodes with an accuracy of $\frac{1}{3}$ thus this classification is worse than the previous one. However, if we focus on the quantification, we have $\widehat{freq}(A) = \frac{5}{11}$, that is exactly the real frequency of the class A, i.e., we do not have any quantification error.

This example shows that for quantifying the prevalence of classes accurately, we need to define specific techniques because even if there are some similarities with classification there is not equivalence between the quality of the respective results.

A straightforward solution to the network quantification problem could be the *sampling*, i.e., we could count the number of instances for each class value in the set of labeled nodes and assume that the same proportion of labels in the test set. However, this approach is not suitable when the class label distribution in the test set is different from the training set, that is the case really interesting for quantification. This is the case in above example. Indeed, we can see that the sampling would return the result depicted in Figure 1(b). In detail, if we do not consider the unlabeled nodes, by sampling we obtain $freq(A) = \frac{3}{8}$ and $freq(B) = \frac{5}{8}$ while the real frequencies are $freq(A) = \frac{5}{11}$ and $freq(B) = \frac{6}{11}$.

B. Quantification methods via classification

The typical approach adopted in the literature, to address the quantification problem in data presented in conventional attribute format, is based on standard *classification*. The idea, introduced in [8], [7], [9], is to use a standard classifier and then post-processing the results with specific methods to improve the quantification accuracy.

Moreover, all the methods proposed so far solve quantification via classification address the binary case: anyway, they can be easily extended to deal with single-label multi-class scenarios. More specifically, in [9] the following methods are introduced to post-process the results of classifiers and optimize them for quantification:

Classify & Count (CC). This simple method generates a classifier from the training set $\mathcal{T}_r$, classifies the unlabeled records in the test set, $\mathcal{T}_e$, and estimates for each class label l_i its frequency $freq_{\mathcal{T}_e}(l_i)$ by counting the fraction of records in T_e that have been labeled with l_i. We indicate the estimation computed by this method by $\widehat{freq}^{CC}_{\mathcal{T}_e}(l_i)$.

Adjusted Classify & Count (AC). This method attempts to improve the results obtained by the previous method by adjusting the quantification obtained by *Classify & Count* $\widehat{freq}^{CC}_{\mathcal{T}_e}(l_i)$ with the information about the true positive rate and false positive rate w.r.t. the training set:

$$\widehat{freq}^{AC}_{\mathcal{T}_e}(l_i) = \frac{\widehat{freq}^{CC}_{\mathcal{T}_e}(l_i) - fpr_{\mathcal{T}_r}}{tpr_{\mathcal{T}_r} - fpr_{\mathcal{T}_r}}. \quad (2)$$

Both methods above can also be used in a network setting after the application of a classifier tailored for network classification. In such scenario the training set $\mathcal{T}_r$ became V_l while the test set $\mathcal{T}_e$ become V_u. When a classifier provides a prediction score for each node in the network, a classification-based quantification methods can be applied. However, standard classifiers, optimized for predicting the class of single element, are not optimal for quantification. In this paper, we want to exploit the homophily effect observed in many social networks [16], [23], [22] to construct a quantifier for networking data.

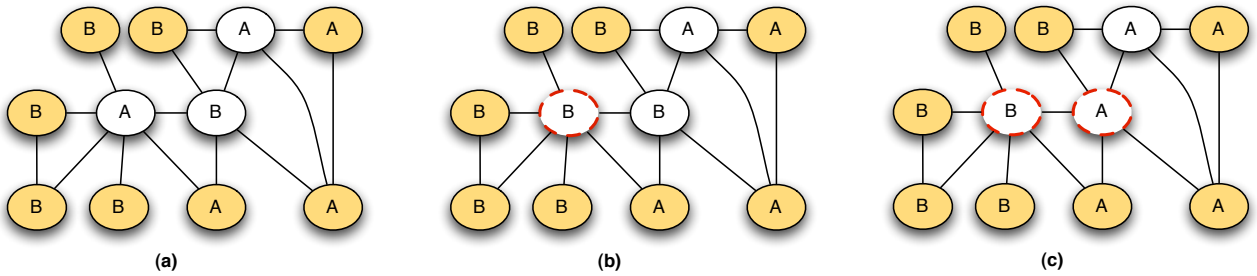


Fig. 1: Network quantification vs Classification

IV. Quantification based on Homophily

In this section we introduce the details of our methods for quantification in networking data based on the notion of homophily. In particular, we propose two categories of methods: one based on community discovery and another based on ego-networks.

A. Community Discovery for Quantification

The methods in this category require the execution of two steps: (i) finding the set of communities; (ii) assigning to the unlabeled nodes the class label by using the information extracted from the communities.

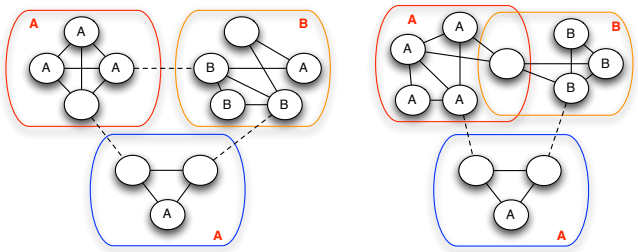
The first step of the algorithm is very simple. Given the whole network G, containing the nodes of the training and test sets, we apply a community detection algorithm that finds clusters of nodes by taking into account the nodes' connections and the topological information about the network. So far, in literature many algorithms have been proposed to address the community discovery problem. The proliferation of these algorithms is due to the fact that there is not a unique definition of community in a network. An exhaustive overview on those definitions and the corresponding algorithms can be found in [2]. All those algorithms returns a set of communities that we denotes by $C = \{c_1, c_2, \ldots, c_n\}$.

To perform the second step, for each community c_i the class label with the highest frequency is identified and assigned to each unlabeled node in the community. In detail, for each community c_i the algorithm computes the frequency of each class $freq_{c_i}(l_i)$ and then identifies the most frequent label that we denote by $Lmax_{c_i}$. Finally, $Lmax_{c_i}$ is assigned to each unlabeled node belonging to the community c_i.

To clarify how this approach works, in Figure 2 (a) we present a simple example. Specifically, in this network the algorithm of community discovery finds three communities that we identify with the colors red, blue and orange. The second step of our method, after having computed the frequency of each label within each community, will assign to the unlabeled nodes belonging to the red community the class label A, to the orange community the class label B and to the blue community the class A.

Even if straightforward, this approach is not suitable when the community discovery algorithm returns overlapping communities like in the network depicted in Figure 2 (b).

In these cases, a node can belong to several communities and each community could have a different majority class



(a) Neat partition (b) Smooth partition

Fig. 2: Community based quantification without overlaps *(a)* and with overlaps *(b)*

label. For example in Figure 2 (b) we have a node belonging to the intersection between the red and orange communities. Moreover, the majority class in the red community is A while in the orange one is B. Now, the question is: *how can we decide the class label for the shared nodes?* We propose two different strategies to decide which class label must be assigned to a node belonging to multiple communities:

Frequency-based Strategy assigning the class label with the greatest overall relative frequency in the labeled nodes. Thus, if a node v_j belongs to m communities and the set of most frequent classes is $\{Lmax_{c_1}, Lmax_{c_2}, \ldots, Lmax_{c_h}\}$ $(h \leq m)$ then, v_j gets the label $Lmax_{c_i}$ if $freq_{c_i}(Lmax_{c_i}) = max_{c_i \in C}\{freq_{c_i}(Lmax_{c_i})\}$.

Density-based Strategy assigning the highest frequency class label of the denser community to which the node belongs.

In the above example, related to Figure 2 (b), adopting the frequency strategy we get: in the red community $Lmax_{red} = A$ with frequency $\frac{4}{5} = 0.8$, while in the orange community $Lmax_{orange} = B$ with frequency $\frac{3}{4} = 0.75$. This implies that we will assign the label A to the shared node because it has a higher frequency. However, following the density policy we get: $density(red) = \frac{7}{10} = 0.7$, $density(orange) = \frac{5}{6} \simeq 0.83$ implying that at the shared node will be assigned the label B.

After the labeling, we have two possibilities: (i) applying the *Classify & Count* strategy, i.e., we compute in the whole network the distribution of the class values by simple counting the nodes labeled with the same label; or (ii) applying the *Adjusted Classify & Count*, i.e., we adjust the quantification obtained by *Classify & Count* with the information about the true positive rate and false positive rate with respect to the

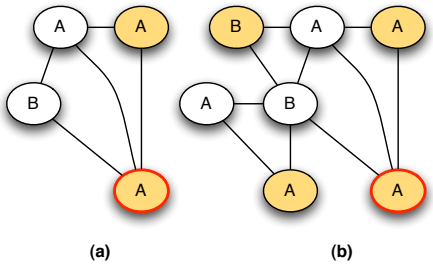


Fig. 3: Ego-network at 1-hop and 2-hops

nodes in the training set.

The weakness of the methodology described until now is that, if in the network we have some isolated and unlabeled nodes, we are not able to assign labels to them because they do not belong to any community. In these particular cases our strategy is to assign to those nodes the class label by following the same class distribution of the training set, i.e. if we have a known distribution of 0.4 for A and 0.6 for B in the training set the isolated nodes of the test set will be assigned respectively 40% to the former class and 60% to the latter.

B. Ego-networks for Quantification

An alternative set of methods that we propose in order to solve the problem of quantification on networks are based on the idea of assigning to each specific node the label that is the most frequent in its neighborhood. In this case we are directly exploiting the homophily property. This approach differs from the previous one where the communities are found without considering the information about the class label, but only the topological structure of the network, and only then, as a post-processing step, is performed a label assignment with the basic assumption about the validity of the homophily within each community.
The quantification process presented in this section is also composed of two main steps: (i) extraction of ego networks, and (ii) class label assignment.

1) Ego-network Extraction: The first step is to generate a partition of the network G in different subgraphs, called *ego-networks*. An ego network is a sub-network centered on a particular node who is the subject of the network. The focal point of the network is called the *ego*. In an ego-network, only nodes that are directly connected to the *ego* form the extracted substructure. An ego-network enables a focused view on the specific properties of a node, highlighting all its interactions with the neighbors. Figure 3 depicts an ego-network example, showing the relations of the ego node A (the node identified with the color red) and its neighbors.

Obviously, the definition of ego-network can be extended by taking into account the k-hop neighborhood; in other words, the size of ego's neighborhood is expanded by including all nodes to whom the ego has a connection at a path length of k, and all the connections among all of these nodes. Intuitively, the more k grows the more the homophily tends to decrease.

In our approach, for each node of the test set we extract its ego-network at k-hops.

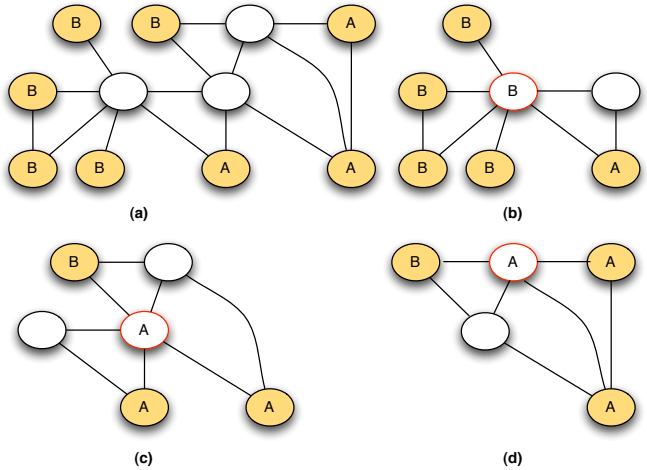


Fig. 4: Ego-network based quantification. *(a)* the full graph with labeled and unlabeled nodes; *(b-c-d)* 1-hop ego-networks of the unlabeled nodes.

2) Class Label Assignment: After this step, for each ego-network $G_{ego} = (V', E', L)$ the algorithm computes the frequency of each class in $freq_{V'}(l_i)$ and then identifies the most frequent label that we denote by $Lmax_{G_{ego}}$. Finally, $Lmax_{G_{ego}}$ is assigned to the ego node.

Also in this case we can have some isolated and unlabeled node that make hard the assignment of the class label because it has no neighbors. As in the previous method, in these particular cases our strategy is to assign to those nodes the class label by following the same class distribution of the training set.

After the assignment of the label to the node, we can apply the *Classify & Count* strategy or the *Adjusted Classify & Count* strategy.

To clarify how this approach works, we discuss a simple example depicted in Figure 4. This figure illustrates the whole process of the algorithm considering ego-network at 1-hop. In particular, Figure 4(a) depicts the original network where the white nodes have unknown class label. This network is the same used in Figure 1. The first step is to extract the ego-networks at 1-hop for each unlabeled (white) node. Figures 4 (b),(c) & (d) show the result of this step. Note that the ego node is indicated by a red border. Then, to each ego node the algorithm assigns the computed label. As a consequence, to the ego node in Figure 4(b) we assign the label B, while to the ego nodes in Figure 4(c) and Figure 4(d) we assign the class label A. This result allows us to obtained a perfect quantifier even if the classification of each node is not perfect as highlighted in Example 1.

V. Complexity Analysis

In this section we discuss the time complexity of the two methods presented above. In the following, we denote by V_l and V_u the set of nodes in the training set and test set respectively.

Theorem 1: Let $G = (V, E, L)$ be a network where $V = V_u \cup V_l$. The network quantification approach based on community discovery computes the class distribution in

$O(CD + |V|)$ time, where $O(CD)$ is the time complexity of the community discovery algorithm.

Proof: The approach based on community discovery, presented in Section IV-A, is composed of two steps. The first one computes the communities in $O(CD)$ time. Note that, $O(CD)$ depends on the algorithm used. The second one visits the communities for computing the most frequent class of each community and assigns to an unlabeled node $v_i \in V_u$ the class label of its community. The visit of all communities can be computed in $O(|V|)$ time where $|V| = |V_u| + |V_l|$. Therefore, we have a whole complexity equal to $O(CD + |V|)$. ■

Now, we can present our analysis for the second approach.

Theorem 2: Let $G = (V, E, L)$ be a network where $V = V_u \cup V_l$. The network quantification approach based on ego-networks computes the class distribution in $O(|V_u| \times |V_l \cup V_u|)$ time.

Proof: The ego-network based approach, presented in Section IV-B, is composed of two steps. The first one computes the ego-networks for each node in the test set V_u. Assuming to implement the network with an hash assigning to each node its 1-hop neighbors, then this step requires $O(V_u)$ accesses to the hash. The second step requires the visit of the ego-networks for computing the most frequent label and this step can be computed in $O(|V_u| \times |V|)$ time, where $|V| = |V_l| + |V_u|$, because each node in the test set can have at most $|V|$ neighbors. Therefore, the whole time complexity of the approach is $O(|V_u| + (|V_u| \times |V|))$. ■

VI. Experiments

In this section we present the evaluation of our methods and the results obtained from our deep experimentation. First, we provide a description of the data used in our experiments (Section VI-A), then we present an overview of the community discovery algorithms used for the evaluation of our method based on community discovery (Section VI-B). Finally, we show the results of the experiments in Section VI-C

A. Datasets

In our empirical evaluation of our methods we used the following datasets:

CoRA[1]: This dataset comprises computer science research papers: the network is built over papers using as relations for the edges both citation and shared-authors. The number of possible different class labels are 7 (their distribution is reported in Fig.5(a)). The network contains 4,240 nodes and 77,824 edges.

IMDb[1]: This dataset comes from the Internet Movie Database, and contains description of movies released in the USA between 1996 and 2001. The class identifies whether the opening weekend box-office sales have exceeded $2 million (class distribution: 57% and 43% respectively). In our network movies are linked if they share a production company, producer, director, or actor. The network contains 1,440 nodes and 51,481 edges.

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

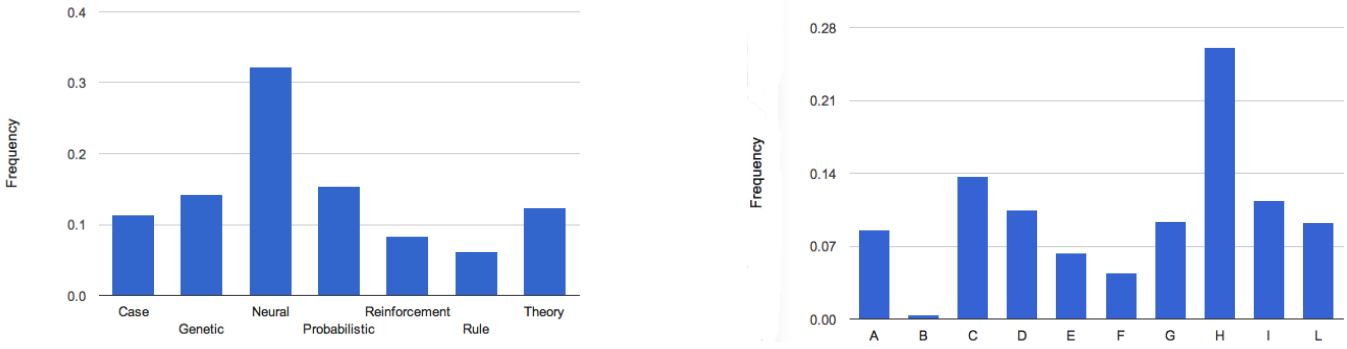
Google+: Social network built on the Google+ service extended with semantic information. The class labels identify the schools attended by the analyzed users (label distribution is reported in Fig.5(b). Different schools are identified with letters from A to L). Our network contains 33,381 nodes and 110,142 edges [10].

B. Community Discovery Methods

To evaluate our quantification methods based on community discovering in our experiments we use two different algorithms for the detection of communities: *DEMON* [3] and *Infohiermap* [20]. They provide the opportunity of testing both the case of overlapping communities (DEMON) and the case of non-overlapping ones (Infohiermap).
DEMON uses a *democratic* approach for discovering the communities in a complex network. This method is based on the notion of ego-network. Each node votes for the communities present in its local view of the network. In practice, the ego network of each node is extracted from the complex network and *Label Propagation* community discovery algorithm [18] is applied on each ego-network ignoring the presence of the ego node itself, since it will be judged by its peer neighbors. After this step, we obtain for each ego node a set of micro-communities. The next step is a phase called *merge*, that combines the vote of every node of the network (e.g. the micro-communities) to obtaining as result a set of overlapping modules. In other words, this phase tries to merge communities following principle: "two communities C_1 and C_2 are merged if and only if at most the $\varepsilon\%$ of the smaller one is not included in the bigger one". Clearly, here ε is a parameter of the algorithm and it is set to 0.25 as suggested by the author in the original paper (moreover, an extensive testing phase has shown how, in our case, the chosen value grant communities having higher average clustering coefficient). This algorithm is incremental, allowing to recompute the communities only for newly incoming nodes and edges in an evolving network. Nevertheless, *DEMON* has also a low theoretical linear time complexity [3]. In following only the results of the micro-communities ones are discussed: this choice was made to reduce the average community size while amplifying the homophily effect.
We choose the second algorith, *Infohiermap* [20] because it is one of the most accurate and best performing non-overlapping clustering algorithms. The basic idea of Infohiermap is combining information theoretic techniques and random walks. Specifically, it uses the probability flow of random walks on a network as a proxy for information flows in the real system and then, decomposes the network into clusters by compressing a description of the probability flow. For the random walks compression the algorithm described the paths with a prefix and a suffix. So, each node belonging to the same cluster of the previous node is described only by its suffix, otherwise by prefix and suffix. Then, the suffixes are reused in all prefixes, just like the street names are reused in different cities. The optimal division in different prefixes represent the optimal community partition.

C. Empirical Evaluation

Our evaluation is organized as follows. First, the compared quantifiers are introduced, and highlights on their implementation are given. Then, the measure used to evaluate their



(a) CoRA (b) Google+

Fig. 5: Label Frequencies. *(a)* CoRA, *(b)* Google+

accuracy is explained; following three different strategies for building the test set are proposed and the experimental results are presented. Moreover, we discuss how network *assortativity* influences the results of the proposed quantification methods. Lastly, we analyze the effects of the overlaps on the performances obtained by community-based approaches.

1) Network Quantifiers: We compare, on the previously introduced networks, six algorithms: (i-ii) Community Discovery based quantification by *Infohiermap* and *DEMON*; (iii) *EG* - Ego-network based labeling[2]; (iv) *LBQ* - Link-based quantification, as defined in [25]; (v) *wvRN* - network classifier, as defined in [15]; and, (vi) *Baseline* - sampling.

The methodologies we propose were tested in their two variants: *Classify & Count* and *Adjusted Classify & Count*. In order to compute the latter for each node $n \in V_l$ the proposed algorithms were applied to newly identify its label: in this way, following a *leave-one-out* strategy, is possible to compute the True Positive Rate and the False Positive Rate on V_l: estimates of tpr and fpr are needed to adjust the label frequency obtained by the standard *Classify & Count*. Once computed the new frequencies, a rescaling step is applied to assure that, for each network, the sum of labels' frequencies is equal to one. It is worth to noting that the Link-based quantification approaches (LBQ), discussed in the related work section II, was slightly modified: the frequencies of labels were assigned using not the median but the mean value of the distribution and, as done for the *Adjusted Classify & Count*, at the end a normalization step was applied to overcome the highlighted issue (i.e. the sum of the estimated frequencies for labels needs to be equal to one).

2) Kullback-Leibler Divergence.: To evaluate the accuracy of a quantifier we need to compare $\widehat{freq_{\mathcal{V}}}(l_i)$, the frequency computed for the new labeled nodes l_i, with $freq_{\mathcal{V}}(l_i)$, its actual frequency. Different measures have been used in the literature for measuring quantification accuracy: the most convincing among the ones proposed so far is the one used by Forman in [9], which uses normalized cross-entropy, better known as Kullback-Leibler Divergence (KLD), defined as:

$$KLD\left(freq_{\mathcal{V}} || \widehat{freq_{\mathcal{V}}}\right) = \sum_{i=1}^{n} freq_{\mathcal{V}}(l_i) \log \frac{freq_{\mathcal{V}}(l_i)}{\widehat{freq_{\mathcal{V}}}(l_i)}$$

[2]Ego-networks were extracted at one and two hops

It aims at evaluating the information loss when $\widehat{freq_{\mathcal{V}}}$ is used as approximation of $freq_{\mathcal{V}}$. It ranges in $[0, +\infty)$: 0 means that the two frequency values are equal for each l_i and $+\infty$ means that their values diverge. If $\widehat{freq_{\mathcal{V}}}(l_i) = 0$ for at least one class, KLD is not defined: therefore, as in [9], we add a small amount ϵ (set to $\frac{0.5}{|V_u|}$) to both numerator and denominator in the log function.

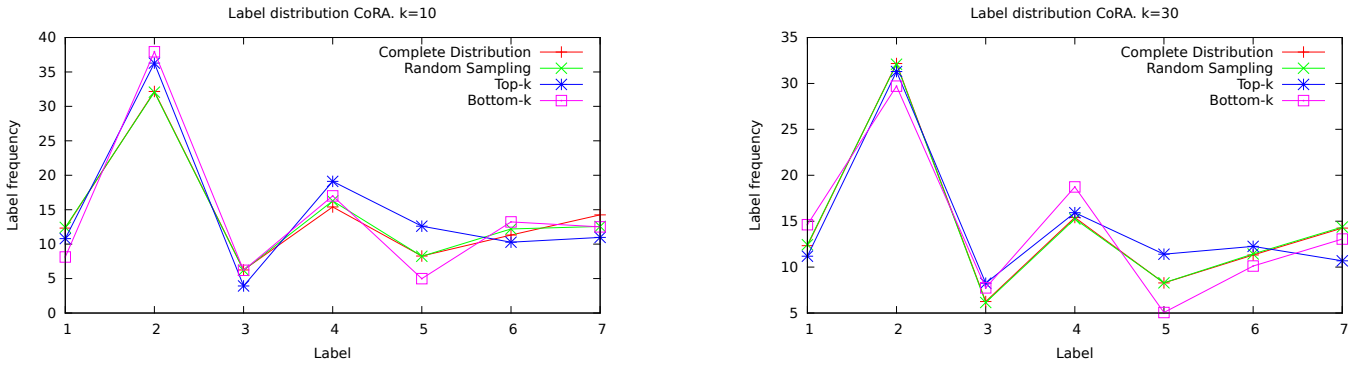
3) Test Set Scenarios.: To better characterize the performance of the compared methodologies we have identified three different scenarios:

- (i) *Random*: the $k\%$ unlabeled nodes are chosen uniformly at random from the whole network;
- (ii) *Top*: the chosen nodes are the $top\text{-}k\%$ w.r.t. the degree distribution;
- (iii) *Bottom*: the chosen nodes are the $bottom\text{-}k\%$ w.r.t. the degree distribution.

Our aim is to capture the average scenario *Random* and two more complex ones which identify those cases in which the neighborhood of unlabeled nodes offers too little (*Bottom-k*) or too much (*Top-k*) information to be exploited for assigning labels. Furthermore, we test all the algorithms by fixing, for each network, the ratio of unlabeled nodes to 10%, 20% and 30% of $|V|$. As shown in Fig. 6, for the CoRA network (as well as for all the other datasets analyzed) the label frequency distributions computed on the *Top-k* and *Bottom-k* node samples show variation w.r.t. the one computed on the whole dataset. Conversely, for the *Random* sampling the frequency distribution does not differ significantly from the complete network's one. We report for each network a table with the KLDs score of the tested approaches: the best results are highlighted in bold for each value k and node sampling scenario.

Random sampling: Extracting $k\%$ of the nodes uniformly at random from the original network ensures that the label distribution of V_u shows a very low drift from the one of V_l. This scenario is very unlikely on real data and in this case also simple approaches, as sampling, can produce good results. In CoRA, as well as in IMDb and Google+ (Table I, II and III), we observe how EG approaches outperform both the baseline and the community-based methods.

Bottom-k sampling: Populating V_u with the *Bottom-k* nodes



(a) (b)

Fig. 6: Label frequency distribution on CoRA for *(a)* $k = 10\%$ and *(b)* $k = 30\%$.

	CoRA			CoRA Bottom			CoRA Top		
Method	10	20	30	10	20	30	10	20	30
Infohiermap	1.369e-2	3.276e-2	4.026e-2	4.213e-2	2.895e-2	3.737e-2	1.211e-2	1.890e-2	1.671e-2
Demon	1.777e-2	2.578e-2	3.830e-2	4.779e-2	3.671e-2	4.173e-2	1.008e-1	5.255e-2	6.055e-2
Demon Density	5.425e-3	1.7375e-2	3.273e-2	5.238e-2	3.695e-2	4.627e-2	5.528e-2	6.258e-2	7.012e-2
EG1h	6.719e-3	1.533e-2	3.003e-2	5.194e-2	3.951e-2	4.762e-2	5.124e-2	6.298e-2	5.958e-2
EG2h	1.459e-2	1.149e-2	4.081e-2	**1.909e-2**	2.111e-2	2.486e-2	1.408e-1	1.184e-1	1.310e-1
Infohiermap AC	6.2387e-3	1.324e-2	1.902e-2	3.724e-2	2.401e-2	3.454e-2	**7.219e-3**	1.593e-2	**1.453e-2**
Demon AC	1.031e-2	1.118e-2	1.648e-2	4.290e-2	3.277e-2	3.890e-2	9.589e-2	4.968e-2	5.841e-2
Demon Density AC	2.276e+0	1.290e-1	1.307e-1	1.170e-1	2.574e+0	1.199e+0	2.305e+0	2.173e+0	1.191e+0
EG1h AC	**1.197e-3**	**4.395e-3**	**3.207e-3**	2.884e-2	2.386e-2	3.926e-2	3.152e-2	**1.451e-2**	1.730e-2
EG2h AC	3.354e-3	5.484e-3	6.124e-3	2.926e-2	**2.045e-2**	**2.587e-2**	1.195e-2	1.522e-2	2.246e-2
LBQ1h	1.990e-1	2.523e-1	2.333e-1	3.132e-1	2.750e-1	2.763e-1	3.634e-1	3.251e-1	2.733e-1
LBQ2h	1.927e-1	2.478e-1	2.313e-1	3.132e-1	2.772e-1	2.531e-1	2.969e-1	3.273e-1	2.972e-1
Baseline	7.450e-3	1.512e-2	3.126e-2	5.285e-2	4.789e-2	5.890e-2	6.427e-2	7.447e-2	7.789e-2
wvRN	2.773e-2	1.684e-2	2.349e-2	1.353e-1	6.965e-2	4.214e-2	1.159e+0	7.229e-2	7.475e-2

TABLE I: CoRA: mean of KLD of predicted and actual quantification for all quantifiers.

	IMDb			IMDb Bottom			IMDb Top		
Method	10	20	30	10	20	30	10	20	30
Infohiermap	7.948e-3	1.918e-2	2.124e-2	1.075e-1	1.384e-1	2.013e-1	8.496e-3	6.752e-3	3.047e-3
Demon	1.321e-1	1.599e-1	1.759e-1	2.570e-1	4.254e-1	5.762e-1	5.177e-3	1.617e-2	6.159e-2
Demon Density	5.451e-2	6.009e-2	2.171e-2	2.777e-1	4.067e-1	4.882e-1	4.615e-1	5.638e-1	4.365e-1
EG1h	1.716e-1	1.916e-1	1.065e-1	2.856e-1	4.215e-1	5.029e-1	5.155e-1	5.761e-1	5.162e-1
EG2h	7.248e-1	7.684e-1	5.248e-1	1.681e+0	2.482e+0	2.694e+0	1.638e+0	1.463e+0	1.407e+0
Infohiermap AC	7.830e-4	4.996e-3	2.188e-4	1.024e-1	1.353e-1	1.991e-1	3.328e-3	**3.661e-3**	**8.415e-4**
Demon AC	1.248e-1	1.457e-1	1.549e-1	2.519e-1	4.224e-1	5.740e-1	**8.824e-6**	1.308e-2	5.938e-2
Demon Density AC	1.267e-2	1.202e-2	1.108e-2	**2.908e-3**	**3.852e-4**	**7.043e-3**	2.040e-2	4.241e-2	6.925e-2
EG1h AC	**4.525e-4**	1.490e-4	**5.560e-5**	7.915e-1	6.056e-1	3.593e+0	2.060e-2	1.211e-2	4.608e+0
EG2h AC	1.359e+0	5.452e-1	3.711e+0	7.915e-1	6.056e-1	3.593e+0	2.465e+0	1.915e+0	1.795e+0
LBQ1h	6.883e-4	**7.872e-5**	5.674e-3	2.314e-1	1.955e-1	1.793e-1	1.076e-1	1.176e-1	1.010e-1
LBQ2h	1.052e-2	3.967e-3	8.692e-3	1.323e-1	7.828e-2	3.484e-2	2.069e+0	6.573e-2	6.638e-2
Baseline	8.461e-3	1.427e-2	2.154e-3	2.750e-1	4.360e-1	4.872e-1	8.468e-2	2.104e-1	2.568e-1
wvRN	3.196e-3	6.541e-3	7.053e-4	4.486e-1	5.672e-1	5.368e-1	1.432e-1	2.257e-1	3.309e-1

TABLE II: IMDb: mean of KLD of predicted and actual quantification for all quantifiers.

w.r.t. the degree distribution of the network may introduce a drift on the label frequency: this is reflected by the results of the baseline and LBQ which, increasing the size of the sample worsen their KLD. Conversely, our approaches tend to increase their performances as the size of V_u increases: this is due to the greater connectivity that can be exploited to assign labels. In CoRA and Google+ EG again registers the best KLD values, while on IMDb a community-based method (DEMON) obtains the best performances.

Top-k sampling: Similarly to the *Bottom-k*, the *Top-k* node sampling introduces a distribution drift due to the unequal probability for each node to be part of the V_u set. Contrary to the previous sampling strategy, the real challenge here is to correctly discriminate the information given by the high

	Google+ School			Google+ School Bottom			Google+ School Top		
Method	10	20	30	10	20	30	10	20	30
Infohiermap	5.723e-3	6.247e-3	6.126e-3	7.281e-3	5.620e-3	3.475e-3	4.862e-4	**5.511e-4**	2.011e-3
Demon	8.370e-3	1.309e-2	1.517e-2	2.375e-2	3.563e-2	4.276e-2	1.936e-3	1.638e-3	**3.452e-4**
Demon Density	8.710e-3	1.393e-2	1.661e-2	2.375e-2	3.563e-2	4.251e-2	2.159e-3	1.350e-3	3.875e-3
EG1h	1.366e-3	1.685e-3	1.823e-3	6.584e-3	5.154e-3	6.316e-3	8.580e-4	5.801e-3	1.719e-2
EG2h	2.255e-3	5.116e-3	5.017e-3	9.139e-3	7.984e-3	6.816e-3	**3.752e-4**	1.042e-3	1.159e-3
Infohiermap AC	8.770e-3	2.744e-3	2.641e-3	6.004e-1	5.385e-1	4.822e-1	9.563e-1	1.061e+0	9.135e-2
Demon AC	3.047e-1	3.092e-1	3.150e-1	2.912e-1	2.840e-1	2.816e-1	2.910e-1	2.152e-1	1.887e-1
Demon Density AC	3.038e-1	3.081e-1	3.138e-1	2.900e-1	2.830e-1	2.810e-1	2.927e-1	2.202e-1	1.989e-1
EG1h AC	**4.333e-4**	**1.160e-3**	**1.354e-3**	4.573e-3	3.424e-3	2.508e-3	1.411e-2	1.860e-3	2.531e-3
EG2h AC	7.465e-4	3.504e-3	4.177e-3	**2.913e-3**	**2.531e-3**	**1.795e-3**	3.110e-2	1.407e-2	1.167e-2
LBQ1h	2.867e-1	2.590e-1	2.623e-1	2.622e-1	2.512e-1	2.472e-1	4.284e-1	3.555e-1	3.163e-1
LBQ2h	2.887e-1	2.581e-1	2.612e-1	2.543e-1	2.416e-1	2.376e-1	4.273e-1	3.555e-1	3.146e-1
LBQ3H	2.830e-1	2.538e-1	2.555e-1	2.518e-1	2.390e-1	2.360e-1	4.221e-1	3.522e-1	3.138e-1
Baseline	1.772e-3	3.396e-3	5.385e-3	2.375e-2	3.563e-2	4.261e-2	1.753e-1	1.180e-1	8.617e-2
wvRN	2.206e-3	7.085e-3	4.391e-3	1.294e-2	1.657e-2	1.338e-2	1.881e-1	1.069e+0	4.037e-1

TABLE III: Google+: mean of KLD of predicted and actual quantification for all quantifiers.

connectivity of the unlabeled nodes. The selected nodes are hubs: their high degree increases the probability of being connected with nodes that do not share common labels. We can observe how Baseline as well as LBQ and wvRN are not able to record the best performances: community-based approaches on CoRA, IMDb and Google+ show the overall better accuracy.

4) Homophily estimation: Label Assortativity: To justify the discussed results we have analyzed the degree of homophily of our three datasets: to do so, we have used a network measure called *assortativity*. Assortativity measures the preference of a node to attach to others that are similar in some way, for this reason it can used as a proxy to estimate the overall homophily level within a network w.r.t. a specific feature (i.e. node degree). Due to the problem addressed and to the three proposed test set construction strategies, we will focus on a specific instantiation of this measure: Label Assortativity which measures how much nodes tend to being connected with similar labeled ones. To compute assortativity is commonly used the Pearson correlation coefficient, that lies in $[-1, 1]$: a positive value indicates a correlation between nodes with similar labels, while a negative one denotes relationships between nodes with different labels. When the correlation is equal to 1, the network has a perfect assortative mixing patterns, when it is equal to 0 it is non-assortative, while when it is equal to -1 the network is completely disassortative. CoRA and Google+ have high Label Assortativity (0.6233 and 0.8912, respectively): this reflects positively on the results of the EG quantifiers, which (almost always) outperform the other approaches, exploiting directly the homophily property. Instead, on IMDb which has a lower Label Assortativity (0.2787), i.e. lower homophily, community-based approaches, which make use of broader information for assigning class labels, outperform the other quantifiers.

5) Overlapping Community Discovery: Comparing the proposed community-based algorithms on the datasets with lower values of Label assortativity, we can note a significant predominance of Infohiermap KLD scores over the DEMON's ones. Given the high KLD values of the latter we may conjecture that overlaps can be a cause of misclassification: this result is supported by the fact that avoiding the merging phase to reduce the community's size (as well as their overlaps) we improve DEMON's performances on all the datasets.

VII. Conclusion

In this work we have proposed two approaches for performing quantification in complex networks. Our quantification methods exploit the homophily effect observed in many social networks. The first method, based on community discovery, estimates class prevalence in a population by considering the characteristics of the communities composing the entire network, while the second approach infers the class of each node in the network by observing the class distribution in its neighborhood. The thorough experimental evaluation that we have carried out shows that our methods outperform state-of the-art quantifiers. Given the ever-growing availability of social networks and social media, solving the quantification problem on networks opens up new avenues for the estimation of social indicators based on big data, provided that we can rely on relatively small surveys of labeled data. Moreover, in the experimental section we have seen that the proposed approaches are stable w.r.t. variations on the criteria used to build the test set. Such is an highly valuable property because quantification major impact emerge when analyzing dynamic networks: in such scenarios make assumption on the class distribution of novel nodes is not an easy task and being able to continuously providing a valid estimate can be crucial to support decision processes.

Acknowledgement

This work was partially funded by the European Community's H2020 Program under the funding scheme "FETPROACT-1-2014: Global Systems Science (GSS)", grant agreement #641191 CIMPLEX "Bringing CItizens, Models and Data together in Participatory, Interactive SocialL EXploratories", https://www.cimplex-project.eu.

References

[1] A. Bella, C. Ferri, J. Hernández-Orallo, and M. J. Ramírez-Quintana, "Quantification via probability estimators," in *ICDM 2010*.

[2] M. Coscia, F. Giannotti, and D. Pedreschi, "A classification for community discovery methods in complex networks," *Statistical Analysis and Data Mining*, vol. 4, no. 5, pp. 512–546, 2011.

[3] M. Coscia, G. Rossetti, F. Giannotti, and D. Pedreschi, "Demon: a local-first discovery method for overlapping communities," in *KDD*, 2012, pp. 615–623.

[4] E. Deaux and J. W. Callaghan, "Key informant versus self-report estimates of health-risk behavior," *ERX*, vol. 9, no. 3, 1985.

[5] A. Esuli and F. Sebastiani, "Machines that learn how to code open-ended survey data," *IJMR*, vol. 52, no. 6, 2010.

[6] A. Esuli and F. Sebastiani, "Sentiment quantification," *IEEE Intelligent Systems*, vol. 25, no. 4, 2010.

[7] G. Forman, "Quantifying trends accurately despite classifier error and class imbalance," in *KDD 2006*.

[8] G. Forman, "Counting positives accurately despite inaccurate classification," in *ECML*, 2005.

[9] G. Forman, "Quantifying counts and costs via classification," *DMKD*, vol. 17, no. 2, 2008.

[10] N. Z. Gong, W. Xu, L. Huang, P. Mittal, E. Stefanov, V. Sekar, and D. Song, "Evolution of social-attribute networks: Measurements, modeling, and implications using google+," *CoRR*, vol. abs/1209.0835, 2012.

[11] L. A. Goodman, "Snowball sampling," *The Annals of Mathematical Statistics*, vol. 32, no. 1, 1961.

[12] D. D. Heckathorn, "Respondent-driven sampling: a new approach to the study of hidden populations," *Social problems*, 1997.

[13] D. J. Hopkins and G. King, "A method of automated nonparametric content analysis for social science," *AJPS*, vol. 54, no. 1, 2010.

[14] D. D. Lewis, "Evaluating and optmizing autonomous text classification systems," in *SIGIR 1995*.

[15] S. A. Macskassy and F. Provost, "Classification in networked data: A toolkit and a univariate case study," *JMLR*, vol. 8, Dec. 2007.

[16] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annual Review of Sociology*, vol. 27, no. 1, 2001.

[17] L. Milli, A. Monreale, G. Rossetti, F. Giannotti, D. Pedreschi, and F. Sebastiani, "Quantification trees," *2013 IEEE ICDM*, 2013.

[18] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Physical Review E*, vol. 76, no. 3, p. 036106, 2007.

[19] E. Rahme and L. Joseph, "Estimating the prevalence of a rare disease: Adjusted maximum likelihood," *The Statistician*, vol. 47, pp. 149–158, 1998.

[20] M. Rosvall and C. T. Bergstrom, "Multilevel compression of random walks on networks reveals hierarchical organization in large integrated systems," *PLoS ONE*, vol. 6(4), 2011.

[21] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad, "Collective classification in network data," *AI Magazine*, vol. 29, no. 3, pp. 93–106, 2008.

[22] C. R. Shalizi and A. C. Thomas, "Homophily and contagion are generically confounded in observational social network studies," Tech. Rep. arXiv:1004.4704, Apr 2010.

[23] A. S. Sinan Aral, Lev Muchnika, "Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks," 2009.

[24] M. Spreen and R. Zwaagstra, "Personal network sampling, outdegree analysis and multilevel analysis: Introducing the network concept in studies of hidden populations," *International Sociology*, vol. 9, no. 4, 1994.

[25] L. Tang, H. Gao, and H. Liu, "Network quantification despite biased labels," in *MLG 2010*.

[26] S. L. Warner, "Randomized response: A survey technique for eliminating evasive answer bias," *Journal of the American Statistical Association*, vol. 60, no. 309, pp. 63–69, 1965.

[27] J. K. Watters and P. Biernacki, "Targeted sampling: options for the study of hidden populations," *Social Problems*, 1989.

[28] J. Xue and G. Weiss, "Quantification and semi-supervised classification methods for handling changes in class distribution," in *KDD 2009*.

[29] X.-H. Zhou, D. K. McClish, and N. A. Obuchowski, *Statistical Methods in Diagnostic Medicine*. New York, US: Wiley, 2002.