A model for ambiguation and an algorithm for disambiguation in social networks*

Janaína Gomide, Hugo Kling, and Daniel Figueiredo

Systems Engineering and Computer Science Program (PESC)
Federal University of Rio de Janeiro (UFRJ), Brazil
{janaina,hugo,daniel}@land.ufrj.br

Abstract. A common assumption when collecting network data is that objects can be uniquely identified. However, in many scenarios objects do not have a unique label giving rise to ambiguities since the mapping between observed labels and objects is not known. In this paper we consider the ambiguity problem that emerges when objects appear with more than one label in the context of social networks. We first propose a probabilistic model to introduce ambiguity in a network by duplicating vertices and adding and removing edges. Second, we propose an simple label-free algorithm to remove ambiguities by identifying duplicate vertices based only in structural features. We evaluate the performance of the algorithm under two classical random network models. Results indicate that network structure can indeed be used to identify ambiguities, yielding very high precision when local structure is preserved.

Keywords: network ambiguity, social networks, network structure, disambiguation

1 Introduction

During the past decade, networks have increasingly been used to encode relationships between objects, from interactions among proteins, to friendship among people, to hyperlinks between webpages. Underlying this abstraction is the premise that objects can be uniquely identified when observing relationships among them. For example, user accounts in Facebook have a unique number identifier that is used when crawling the friendship graph.

However, in many scenarios objects do not exhibit a unique identifier when relationships among them are observed. In particular, a single object may have different labels that appear in reference to the object, or alternatively, a single label may appear in reference to different objects. For example, in the context of social networks, a person (object) may be known by various names (labels), or a single name (label) may be given to different people (objects). Thus, when observing relationships among labels of objects we are faced with ambiguity, since

* This research received financial support through grants from FAPERJ and CNPq (Brazil).

the mapping between observed labels and objects may not be known a priori. In a nutshell, network disambiguation refers to the problem of removing ambiguities among nodes of network that is constructed by observing relationships among ambiguous labels. A more precise formulation is given in Section 3.

In this work we are interested in understanding ambiguity arising when a single object can appear with different labels in the context of social networks. We call this the "Brazilian Ambiguity Problem" (BAP) in allusion to the fact that Brazilians tend to have many first and last names which then appear in many different forms and combinations. Towards this direction, we make the following contributions:

1. Ambiguity model for BAP: based on intuition and empirical observations of real data, we propose a probabilistic model that introduces ambiguity in a social network. The model has three intuitive parameters used for tuning the desired amount and structure of ambiguity and can operate over any original social network. This model is presented in Section 4.
2. Disambiguation algorithm for BAP: again, based on intuition and empirical observations of real data, we propose a simple and efficient label-free algorithm for removing ambiguity in the context of BAP. Our algorithm uses only the structure of the network of observed labels but not the labels themselves to identify nodes (labels) that refer to the same person. We present an extensive analysis of the performance (precision and recall) of algorithm when applying the proposed ambiguity model to random graph models. The algorithm and its evaluation are presented in Sections 5 and 6, respectively.

Identifying ambiguities among nodes of a network of observed labels is an important problem, as one is usually interested in the network of objects. In particular, the network of objects and not labels is the one that is used to characterize and make statements about relationships or other phenomena that depends on the structure. Nevertheless, the problem of name disambiguation has been studied for more two decades, as discussed in Section 2. Our contributions as enumerated above indicates that structure alone in the network of observed labels can contribute to addressing the BAP.

2 Related Work

The problem of network disambiguation is considered a difficult and relatively open problem [**?**,**?**]. Author name disambiguation was initially studied in the Information Sciences using manual and intuitive methods [**?**], but also in Computer Science using sophisticated algorithms [**?**,**?**].

Most approaches found in literature consider label and textual information as main features to remove ambiguities in the network, which might not be available in several contexts. We believe that structural features are fundamental to solve ambiguity in networks in agreement with other recent works [**?**,**?**,**?**].

The problem of more than two people being represented in one node (appear with the same name) has been addressed using a supervised classification algorithm (SVM) considering as features the structural information of the network

[**?**], and also using an unsupervised learning algorithm [**?**]. The BAP (one person appearing with multiple names) has also been addressed using a machine learning approach with structure and textual features [**?**]. Our work contribution is an ambiguity model for the BAP and a disambiguation algorithm that does not use machine learning.

3 Problem Statement

In this section we formalize the network ambiguity problem. Consider a graph $G = (O, E)$ where the vertex set $O = \{o_1, \ldots, o_n\}$ represents objects and the edge set E represents pairwise relationships among the objects. Lets assume that objects have labels and in particular, let $L_i = \{l_{i,1}, \ldots, l_{i,s_i}\}$ denote the set of labels that can be assigned to object o_i. Note that objects have one or more label that are not necessarily unique. Thus, labels of different objects can be identical.

Consider an observation process of relationships among objects that reveals object labels. Thus, a relationship $(o_i, o_j) \in E$ is observed as (l_i, l_j) where $l_i \in L_i$ and $l_j \in L_j$. Let $L = \bigcup_{i=1}^{n} L_i$ denote the set of all different labels. The observation process applied to many (possibly all) relationships $(o_i, o_j) \in E$ will then yield a graph $G' = (L', E')$ where the vertex set $L' \subset L$ represents all observed labels and the edge set E' represents all observed relationships among labels. Note that a given $l \in L'$ can refer to two or more objects while a given $l_1, l_2 \in L'$ can refer to the same object.

The network disambiguation problem is to recover G (network of objects) having observed G' (network of labels). In the context of the "Brazilian Ambiguity Problem" (BAP) studied in this paper, labels of different objects are different, thus, $l_i \neq l_j$ for any $l_i \in L_i$ and $l_j \in L_j$ and for any $i \neq j$. However, we also assume there is no information on the labels themselves (i.e., labels are random numbers), and no information on the number of labels assigned to each object.

4 Ambiguation Model

In this section we present a novel probabilistic model that introduces ambiguity in a network. The model is mostly tailored for social networks and its workings are based on intuition and empirical observations. The idea is to duplicate nodes and add and remove edges to neighbours of the original node. A duplicated node represents a second label for the original node. Therefore, one object (node) of the original network can be represented by two nodes (labels) in the ambiguous network and relationships among the original object (node) can be copied to its duplicate and removed from itself.

Consider a network represented as a graph $G = (V, E)$ in which V is the vertices set (e.g. people), and E is the set of edges (e.g. friendship relationship). In this graph, each vertex uniquely identifies an object in the network. The proposed model has three phases, each with a parameter:

1. **Vertex duplication:** with probability p a vertex is duplicated;
2. **Edge addition:** with probability q an edge between a neighbour of the original vertex and the duplicated vertex is created;
3. **Edge removal:** with probability r an original edge that was copied to a duplicated vertex is removed.

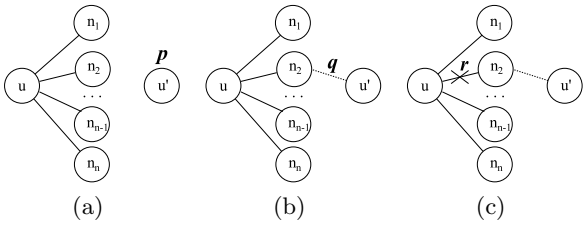

Fig. 1. Parameters of the probabilistic model for create ambiguity in a network. In (a) vertex duplication phase, (b) edge addition phase, and (c) edge removal phase.

In the vertex duplication phase, the vertices are duplicated creating ambiguity. Each vertex $u \in V$, sampled with probability p independently, to generate another graph with a duplicate vertex, u', as shown in Figure 1(a). Note that p controls the amount of ambiguity introduced in the network, so that with $p = 1$ all vertices will have a duplicate in a network.

In the edge duplication phase, the neighbours from the original vertex are copied to the duplicated vertex. For each neighbour $v \in N_u$(neighbours of u) of an original vertex u that has been duplicated, with probability q independently, an edge $e = (u', v)$ is created as illustrated in Figure 1(b). Note that with $q = 1$ all neighbours from u will become neighbours of u'.

In the edge removal phase, edges between an original vertex u and a neighbour v, that has become a neighbour of u' is removed with probability r, independently, as shown in Figure 1(c). Note that for $r = 1$ all edges between the original vertex u and its neighbours that became neighbours of the duplicate vertex u' will be removed. The algorithm for this ambiguity model is described in Algorithm 1.

5 Algorithm for removing ambiguities

In this section we present a simple algorithm to identify ambiguities in the context of BAP in a social network. In particular, we consider just the case where a single object, due to ambiguities, can be represented in the observed label network by more than one vertex. Our algorithm will identify network nodes that represent the same entity without resorting to label information - thus, only structure information will be used.

We develop several structure-based heuristics to identify nodes in the label network that might represent the same entity. For example, we consider that two nodes might refer to the same entity if they are at distance 2, since it is unlikely

Algorithm 1: Model to introduce ambiguity with parameters: p, q, r.

Data: $G = (V, E)$, p, q, r
Result: $G' = (V', E')$
$E' \longleftarrow E$; $V_d \longleftarrow \emptyset$; $E_d \longleftarrow \emptyset$
for *v in V* **do**
  with probability p, duplicate v into v' and $V_d \longleftarrow V_d \cup v'$
for *v' in* V_d **do**
  $v \longleftarrow$ original(v')
  $N \longleftarrow$ neighbours(v)
  for *u in N* **do**
    with probability q, create $e' = (v', u)$ and $E_d \longleftarrow E_d \cup e'$
    if e' *in* E_d **then**
      with probability r, remove $e = (v, u)$ from E'
$V' \longleftarrow V \cup V_d$; $E' \longleftarrow E' \cup E_d$

that a node will have a relationship with itself using two different labels. Moreover, the same is considered if the common neighbourhood between two vertices strongly overlaps, and is contained in one another. We aim in developing a conservative approach to merge nodes, in order to minimize false-positives, allowing greater applicability of the algorithm. The proposed algorithm is described in Algorithm 2.

Algorithm 2: Algorithm - Remove ambiguity

Data: $G = (V, E)$, α
for *v in V* **do**
  $P \longleftarrow \emptyset$; $N_v \longleftarrow$ neighbours(v) ; $D_v^2 \longleftarrow \{u | distance(u, v) = 2\}$
  for *u in* D_v^2 **do**
    if $degree(v) \geq \alpha$ *and* $degree(v) \leq$ *degree(u)* **then**
      $N_u \longleftarrow$ neighbours(u)
      if $N_v \subseteq N_u$ **then**
        $P \longleftarrow P \cup u$
  if *sizeOf(P) = 1* **then** `/* Ambiguity found! Unify` v `and P.first() */`
    merge(v,P.first())

6 Evaluation

In this section we present an extensive evaluation of the performance of the proposed algorithm to remove ambiguities when applied to networks generated by the ambiguity model.

The steps evaluation has the following steps: (i) generate the networks, (ii) introduce ambiguity using the model proposed in Section 4, (iii) apply the algorithm to remove ambiguity proposed in Section 5 and (iv) measure the precision and recall of the algorithm.

In order to generate the networks, we use two models, Erdos-Renyi model, that generates graphs connecting nodes randomly, and Watt-Strograts model, that generates graphs with small-world properties [**?**]. Both networks were generated with $n = 100,000$ vertices and average degree of eight (rewiring probability of two percent was used in the Watts-Strogats model).

Next, we introduce ambiguity into the two networks created. We apply the probabilistic model with different values for the parameters p, q and r aiming to evaluate how these parameters affect the identification of duplicated vertices. The values used for each parameter are 0.1, 0.3, 0.5, 0.7 and 0.9. We apply the algorithm to remove the duplicated vertices, with parameter $\alpha = 0$, and we evaluate the performance by measuring the precision and recall of the algorithm. For each parameter configuration, we perform thirty independent runs and report the sample average of performance metrics. The algorithm performance in the Erdos-Renyi and in the Watts-Strogatz network models with ambiguity are shown in Figures 2 and 3 respectively for all combinations of model parameters.

The precision and recall for the Erdos-Renyi model are shown in Figures 2(a) and 2(b), respectively. Note that the parameter p is not critical to the algorithm, when ten or ninety percent of the vertices are duplicated the performance of the algorithm remains roughly the same. This occurs because in the Erdos-Renyi network model lacks local structure and, therefore, any duplication of vertices and edges creates a local structure that is detected by the algorithm. In these Figures the lines are grouped by the parameter r, so that with smaller values of r we get around 100% of precision and 50% of recall.

In Figures 2(c) and 2(d) we observe an inflexion point with respect to parameter q, with precision and recall growing and then o decrease. This occurs because the number of edges that are removed from the original grows with q. However, for lower values of q the duplicated vertex has a small degree and thus there are many vertices that are candidates to be its original and the algorithm fails to make a decision yielding a lower precision and recall. The inflexion point changes with the value of r because the expected number of removed edges is $d_u qr$ where d_u is the degree of the node u.

Figures 2(e) and 2(f) shows the precision and the recall as a function of parameter r, respectively. Clearly, r is the most sensitive parameter for the performance of the algorithm. Note that precision is more than 90% for values of r lower than 0.5, independent of the other parameters p and q. As r grows the precision and the recall decrease as more original edges are removed and the algorithm fails to find the original vertex that corresponds to the duplicated one.

Results under the Watts-Strogatz network model is shown in Figure 3. In general, results have the same qualitative trends as for the Erdos-Renyi model, with a higher sensitivity in the parameter r. For example Figures 3(e) and 3(f) illustrate that performance degrades quickly as r increses. This occurs due to the local structure present in the Watts-Strogatz model, which makes the algorithm fail if few edges are removed.

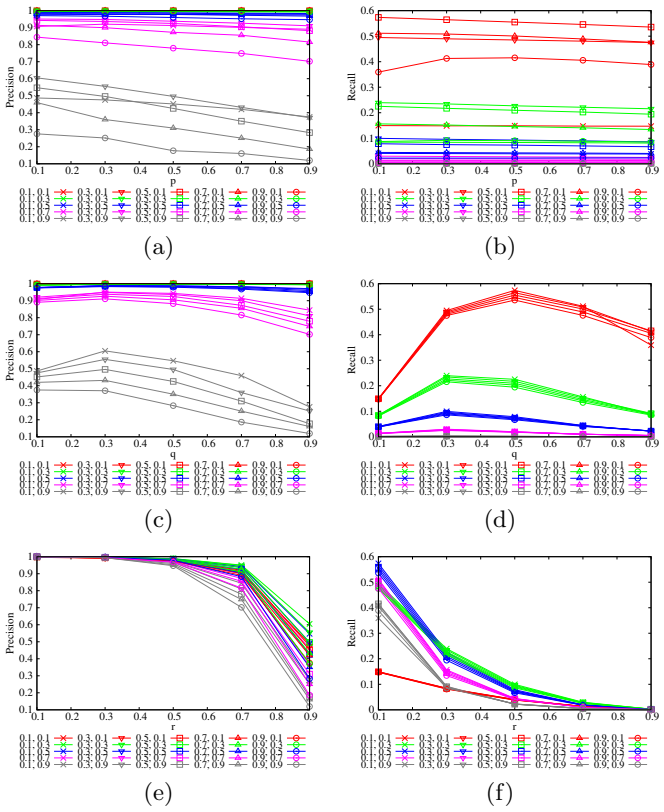

Fig. 2. Evaluation in Erdos-Renyi network with ambiguity. In (a,c,e) precision and in (b,d,f) recall. The pair of values in the legend correspond to p, q, r with the exception of the value appearing in x-axis.

7 Conclusion

In this work we addressed the problem of disambiguation in networks when different labels (vertices) can represent the same object. We proposed a probabilistic model that introduces ambiguity in the context of social networks using three parameters for tuning the desired amount of structural ambiguity. We also propose a simple disambiguation algorithm that uses only structure to identify duplicate nodes. Through simultaneous, we extensively evaluate the performance of the algorithm using random graphs subject to ambiguation introduced by the proposed ambiguity model. Results indicate that the structure of a network can successfully be used to identify ambiguities and does not strongly depend on the amount (fraction) of objects with double identity (duplicated nodes), but on the local structure between the main and the alternative labels. In particular, local network features such as absence of direct edge and common neighbourhood play a key role in disambiguation of social networks.

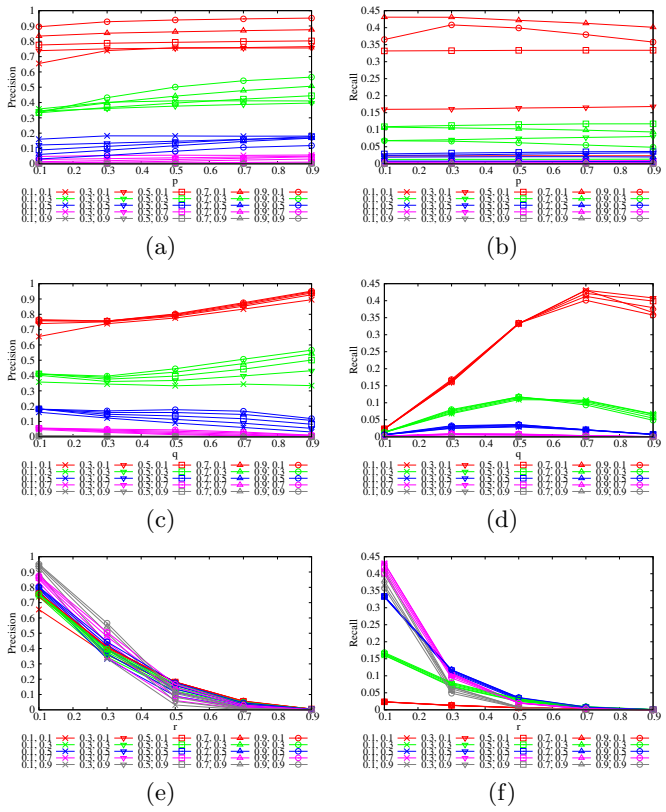

Fig. 3. Evaluation in Watts-Strogatz network with ambiguity. In (a,c,e) precision and in (b,d,f) recall. The pair of values in the legend correspond to p, q, r with the exception of the value appearing in x-axis.

References

1. D. Amancio, O. OliveiraJr., and L. Costa. On the use of topological features and hierarchical charac. for disambiguating names in collab. networks. *EPL*, 2012.
2. S. Elliot. Survey of author name disambiguation: 2004 to 2010. *Library Philosophy and Practice*, 473, 2010.
3. X. Fan, J. Wang, X. Pu, L. Zhou, and B. Lv. On graph-based name disambiguation. *J. Data and Information Quality*, 2(2):10:1–10:23, Feb. 2011.
4. A. A. Ferreira, M. A. Gonçalves, and A. H. Laender. A brief survey of automatic methods for author name disambiguation. *SIGMOD Rec.*, 41(2):15–26, Aug. 2012.
5. L. Hermansson, T. Kerola, F. Johansson, V. Jethava, and D. Dubhashi. Entity disambiguation in anonymized graphs using graph kernels. In *CIKM*, 2013.
6. M. Newman. *Networks: An Introduction*. Oxford University Press, 2010.
7. B. Zhang, T. K. Saha, and M. A. Hasan. Name disambiguation from link data in a collaboration graph. In *ASONAM*, 2014.