

On Member Labelling in Social Networks

Rafael Corchuelo^(✉), Antonia M. Reina Quintero, and Patricia Jiménez

ETSI Informática, Avda. Reina Mercedes s/n, E-41012 Sevilla, Spain
{corchu,reinaqu,patriciajimenez}@us.es

Abstract. Software agents are increasingly used to search for experts, recommend resources, assess opinions, and other similar tasks in the context of social networks, which requires to have accurate information that describes the features of the members of the network. Unfortunately, many member profiles are incomplete, which has motivated many authors to work on automatic member labelling, that is, on techniques that can infer the null features of a member from his or her neighbourhood. Current proposals are based on local or global approaches; the former compute predictors from local neighbourhoods, whereas the latter analyse social networks as a whole. Their main problem is that they tend to be inefficient and their effectiveness degrades significantly as the percentage of null labels increases. In this paper, we present Katz, which is a novel hybrid proposal to solve the member labelling problem using neural networks. Our experiments prove that it outperforms other proposals in the literature in terms of both effectiveness and efficiency.

Keywords: Social networks · Member labelling · Hybrid approach · Neural networks

1 Introduction

On-line social media have sprouted out during the last decade. They have paved the way for on-line social networks whose members typically interact to share or to retrieve information from one another. Never before has it been easier to find information about individuals, their demographics, their likes, their dislikes, the activities in which they engage, their opinions, their thoughts, and so on. And something that is even more important: their relationships.

Software agents are being used in tasks such as searching for experts regarding a given topic, recommending resources (posts, videos, music, and the like), assessing opinions, targeting advertisements, sociological studies, and so on. For these agents to succeed in producing accurate information, it is very important that the information in a member's profile be as complete as possible. Unfortunately, it is not uncommon that many members do not complete their profiles [11], which makes it very difficult for software agents to work well.

Many authors have paid attention to a problem that is commonly referred to as member labelling (aka. member classification, node classification, link-based classification, or collective classification). Simply put, the idea is to infer the

features of a member of a social network as accurately as possible using solely the features available from members with whom he or she has a relationship [1]. This has proven to work well because social networks have a property that is known as homophily [19], according to which members who have similar features tend to have stronger relationships than members that have very dissimilar features. The current proposals in the literature are based on local or global methods. The former learn a predictor from the features of the members of a social network, including some neighbours; the latter tackle the problem from a global perspective and attempt to analyse social networks as a whole. The main problem with current proposals is that they have proven to be inefficient and ineffective as the size of a social network or the number of null features increases.

This motivated us to work on Katz, which is a novel hybrid proposal to solve the member labelling problem. It is based on neural networks, which are used to infer a predictor for each member feature using the information provided by an unbounded neighbourhood. It starts analysing each member’s profile in isolation, and then explores his or her neighbourhood searching for the relationships and features that contribute the most to producing a better predictor. It is not a local method since it explores an unbounded context and selects the most interesting features and neighbours to learn a predictor; neither is it a global method because it does not attempt to analyse social networks as a whole; that is the reason why we refer to Katz as a hybrid proposal. Our experiments on quite a large real-world social network prove that it outperforms other proposals in the literature in terms of both effectiveness and efficiency.

The rest of the paper is organised as follows: Section 2 describes our proposal; Section 3 reports on the results of our experiments; Section 4 surveys the related work and compares it to ours; finally, Section 5 presents our conclusions.

2 Our Proposal

Katz works on a social network that is represented as a graph in which a node represents all of the features of a member profile and an edge represents a relationship to another member. It analyses the network and returns a map in which each feature is associated with a set of neural networks that can be used to label a new member regarding that feature. Note that each feature is predicted by means of a set of neural networks that are learnt from different partitions of the social network; the goal, which has been confirmed empirically, is to decrease the error rate by using an ensemble-predictor approach instead of the single-predictor approach that is common in the literature. In the following subsections, we first present the main procedure of Katz and then an ancillary procedure that is used to extend a neural network to the most appropriate neighbourhood.

Main Procedure: Figure 1 shows the main procedure of Katz. It works on a graph (N, E) that represents a social network. N is a collection of vectors of the form $(m, f_1, f_2, \dots, f_n)$, where m is the unique identifier of a member of the social network and f_i are the values of its features ($i = 1 \dots n$); features can be

```

1: Katz(N, E)
2:   m =  $\emptyset$ 
3:   for each feature f used in N do
4:     ns =  $\emptyset$ 
5:     repeat  $\beta$  times
6:       t = select nodes in N with a value for f
7:       ts = create a training set with  $\lceil \gamma |t| \rceil$  nodes from t
8:       vs = t \ ts
9:       n = null
10:      do
11:        (n', ts', vs') = expandNeuralNetwork(n, ts, vs, N, E)
12:      exit when n = n'
13:      (n, ts, vs) = (n', ts', vs')
14:    end
15:    w = 1/error(n, vs)
16:    ns = ns  $\cup$  {(n, w)}
17:  end
18:  m = m  $\cup$  {(f, ns)}
19: end
20: return m

```

Fig. 1. Main procedure of Katz

either numeric (e.g., age, salary, or opinion polarity about a topic) or categorical (e.g., nationality, gender, or dislikes). E is a collection of vectors of the form (m_1, m_2, k, w) , where m_1 and m_2 are the identifiers of two members of the social network, k denotes a kind of relationship between them, and w is the weight of that relationship. The relationships include any kind of interaction between any two members of a social network (e.g., replies to posts, post forwards, friendship requests, message exchanges, and so on). Thus, the weight of edge (m_1, m_2, k, w) is computed as the number of actual interactions of type k that have occurred between members m_1 and m_2 .

The result of the main procedure is computed in variable m , which is a map that associates every feature in the social network with a collection of tuples of the form (n, w) , where n is a neural network, which acts as a regressor or a classifier for the corresponding feature, and w is its weight, which is the inverse of the error rate; that is, the smaller the error rate, the more important the neural network and the larger the error rate, the less important the neural network. Katz returns β rules for every feature, where β is a user-provided parameter. To label a new member regarding a given feature, the neural networks are applied one after the other. In the case of numeric features, the values predicted by each rule are weighted according to their normalised error rate and then averaged; in the case of categoric features, the results are weighted according to the normalised error rate and the most voted one is returned.

The main procedure basically iterates over the set of features in the social network; in each iteration, it repeats the following procedure β times: it first selects the subset of nodes that have a value for the feature being analysed and then splits it into a training set and a validation set. The size of the training set is controlled by means of γ , which is a user-provided parameter; the remaining


```

1: expandNeuralNetwork(n, ts, vs, N, E)
2:   if n = null then
3:     n = learn network from ts
4:   else
5:     c = expand the neighbourhood of ts and vs using (N, E)
6:     for each (u, v) in c do
7:       n' = learn a network from u
8:       ts' = u
9:       vs' = v
10:      if error(n', vs') < error(n, vs) then
11:        (n, ts, vs) = (n', ts', vs')
12:      end
13:    end
14:  end
15: return (n, ts, vs)

```

Fig. 2. Procedure to expand a neural network

| X0 = member | age(X0) | gender(X0) | group(X0) | school(X0) |
|-------------|---------|------------|-----------|------------|
| m1 | 23 | male | student | physics |
| m2 | 22 | male | student | arts |
| m3 | 23 | female | lecturer | physics |
| m4 | 24 | female | staff | physics |



| X0 = member | age(X0) | gender(X0) | group(X0) | school(X0) | X1 = send-message(X0) | age(X1) | gender(X1) | group(X1) | school(X1) | weight(X1) |
|-------------|---------|------------|-----------|------------|-----------------------|---------|------------|-----------|------------|------------|
| m1 | 23 | male | student | physics | m5 | 24 | female | lecturer | arts | 12 |
| m1 | 23 | male | student | physics | m6 | 32 | male | student | physics | 18 |
| m2 | 22 | male | student | arts | m7 | 22 | male | student | arts | 2 |
| m2 | 22 | male | student | arts | m8 | 24 | female | lecturer | arts | 1 |
| m2 | 22 | male | student | arts | m9 | 32 | female | lecturer | physics | 23 |
| m3 | 23 | female | lecturer | physics | m4 | 24 | female | staff | physics | 18 |
| m3 | 23 | female | lecturer | physics | m9 | 32 | female | lecturer | physics | 90 |
| m3 | 23 | female | lecturer | physics | m1 | 23 | male | student | physics | 12 |
| m4 | 24 | female | staff | physics | null | null | null | null | null | null |

Fig. 3. Excerpt of an expanded training set

nodes are used for validation purposes. It then initialises a neural network n to a null network that does nothing, and then repeatedly expands it until no further expansion is possible. Expanding a neural network consists of extending it to some neighbours as long as this helps to reduce the error rate. We provide additional details in the following subsection.

Expanding Neural Networks: Figure 2 shows the procedure to expand a neural network. It works on a neural network n , a training set ts , a validation set vs , and a social network (N, E) ; it returns a new neural network, the training set from which it was learnt, and the validation set on which it was validated.

The procedure first checks if the input neural network is null, in which case it simply learns a neural network from the training set and returns it. Otherwise, it first expands the neighbourhood of the training and the validation sets using the information provided by the social network. Expanding the neighbourhood of a dataset means that its vectors are expanded with additional components that represent the features of a kind of neighbour. For instance, Figure 3 shows

Table 1. Experimental results

| Age | NJ | | LG | | MP | | B | | Katz | |
|---------|----|---|----|---|-------|-------|-------|-------|-------|------|
| Nullif. | E | T | E | T | E | T | E | T | E | T |
| 5.00% | | | | | 5.55 | 7.01 | 9.70 | 12.39 | 8.85 | 6.75 |
| 10.00% | | | | | 9.39 | 8.22 | 14.30 | 12.74 | 12.55 | 6.69 |
| 15.00% | | | | | 16.45 | 9.18 | 10.70 | 15.60 | 10.88 | 4.36 |
| 20.00% | | | | | 6.49 | 10.61 | 11.16 | 16.30 | 11.75 | 4.71 |
| 25.00% | | | | | 20.43 | 11.37 | 20.00 | 14.55 | 18.83 | 4.70 |
| 30.00% | | | | | 24.08 | 10.94 | 31.86 | 16.23 | 15.62 | 4.68 |
| 35.00% | | | | | 35.55 | 13.56 | 21.17 | 15.63 | 10.59 | 5.05 |
| 40.00% | | | | | 23.39 | 12.46 | 15.31 | 17.88 | 17.86 | 4.66 |
| 45.00% | | | | | 50.13 | 12.92 | 36.26 | 21.65 | 13.18 | 6.38 |
| 50.00% | | | | | 40.23 | 15.81 | 17.44 | 20.75 | 10.48 | 5.59 |
| Mean | | | | | 23.17 | 11.21 | 18.79 | 16.37 | 13.06 | 5.36 |

| Age | NJ | | LG | | MP | | B | | Katz | |
|---------|-------|------|-------|-------|-------|-------|-------|-------|-------|------|
| Nullif. | E | T | E | T | E | T | E | T | E | T |
| 5.00% | 19.12 | 5.56 | 19.77 | 5.60 | 3.44 | 11.77 | 7.94 | 13.51 | 5.40 | 4.10 |
| 10.00% | 26.85 | 6.79 | 21.28 | 6.95 | 6.81 | 12.27 | 8.33 | 15.06 | 9.09 | 3.76 |
| 15.00% | 27.55 | 7.33 | 22.01 | 7.83 | 9.00 | 12.64 | 14.96 | 18.77 | 13.55 | 3.74 |
| 20.00% | 22.76 | 9.04 | 22.75 | 6.96 | 10.33 | 15.55 | 8.94 | 20.13 | 17.36 | 3.23 |
| 25.00% | 34.96 | 7.25 | 23.56 | 7.39 | 19.24 | 16.57 | 8.59 | 20.62 | 25.13 | 4.40 |
| 30.00% | 48.85 | 7.99 | 24.25 | 7.27 | 7.88 | 12.54 | 17.79 | 22.72 | 12.34 | 5.07 |
| 35.00% | 23.34 | 8.59 | 25.05 | 8.13 | 22.35 | 13.76 | 35.32 | 26.06 | 20.84 | 4.76 |
| 40.00% | 22.85 | 7.19 | 35.82 | 8.14 | 24.67 | 16.75 | 27.80 | 23.97 | 20.46 | 2.44 |
| 45.00% | 54.20 | 7.53 | 26.50 | 8.65 | 19.85 | 16.79 | 19.85 | 23.39 | 21.07 | 3.33 |
| 50.00% | 49.13 | 7.86 | 44.71 | 10.52 | 52.53 | 17.77 | 17.79 | 27.51 | 44.04 | 3.57 |
| Mean | 29.96 | 7.51 | 25.57 | 7.74 | 17.61 | 14.64 | 16.73 | 21.17 | 17.93 | 3.84 |

| Gender | NJ | | LG | | MP | | B | | Katz | |
|---------|-------|-------|-------|------|-------|------|-------|------|-------|------|
| Nullif. | E | T | E | T | E | T | E | T | E | T |
| 5.00% | 5.40 | 6.35 | 10.25 | 6.54 | 5.62 | 5.40 | 9.73 | 4.65 | 8.01 | 4.08 |
| 10.00% | 5.72 | 9.33 | 13.54 | 6.90 | 14.50 | 5.61 | 16.40 | 4.86 | 12.73 | 4.14 |
| 15.00% | 6.52 | 10.71 | 22.70 | 6.04 | 12.84 | 6.42 | 11.45 | 5.97 | 11.17 | 4.44 |
| 20.00% | 14.85 | 12.21 | 26.83 | 6.96 | 24.15 | 6.08 | 17.67 | 6.67 | 15.25 | 4.76 |
| 25.00% | 15.29 | 13.68 | 31.03 | 7.16 | 5.59 | 7.49 | 23.57 | 5.72 | 17.17 | 3.22 |
| 30.00% | 7.71 | 15.19 | 20.18 | 8.41 | 35.44 | 5.93 | 32.64 | 6.49 | 10.59 | 4.14 |
| 35.00% | 25.13 | 16.59 | 21.76 | 9.59 | 6.35 | 4.73 | 23.68 | 6.95 | 16.38 | 3.55 |
| 40.00% | 18.76 | 18.08 | 23.57 | 7.91 | 20.12 | 3.62 | 24.36 | 7.03 | 14.36 | 4.06 |
| 45.00% | 13.36 | 19.68 | 29.06 | 7.91 | 27.63 | 3.21 | 14.86 | 5.82 | 13.36 | 4.09 |
| 50.00% | 28.18 | 20.87 | 26.63 | 9.50 | 27.29 | 2.88 | 42.70 | 6.43 | 15.29 | 3.83 |
| Mean | 12.09 | 14.27 | 22.16 | 7.69 | 17.95 | 5.14 | 21.71 | 6.06 | 13.43 | 4.03 |

| Gender | NJ | | LG | | MP | | B | | Katz | |
|---------|-------|-------|-------|-------|-------|-------|-------|-------|-------|------|
| Nullif. | E | T | E | T | E | T | E | T | E | T |
| 5.00% | 21.75 | 5.50 | 16.52 | 7.54 | 4.06 | 5.81 | 9.85 | 4.55 | 6.46 | 5.41 |
| 10.00% | 23.86 | 6.59 | 18.81 | 9.03 | 11.93 | 5.69 | 15.55 | 4.98 | 5.48 | 4.04 |
| 15.00% | 27.77 | 7.92 | 19.95 | 9.47 | 10.85 | 5.89 | 10.73 | 5.32 | 8.01 | 5.27 |
| 20.00% | 30.22 | 8.19 | 21.11 | 7.34 | 11.14 | 6.30 | 8.69 | 5.33 | 7.12 | 5.24 |
| 25.00% | 35.98 | 9.64 | 22.26 | 7.92 | 8.18 | 6.55 | 22.06 | 5.59 | 21.34 | 5.30 |
| 30.00% | 24.09 | 9.59 | 23.47 | 9.40 | 25.70 | 6.88 | 19.83 | 5.69 | 28.90 | 2.09 |
| 35.00% | 40.91 | 11.79 | 24.63 | 8.97 | 14.68 | 7.94 | 35.23 | 6.84 | 28.30 | 2.07 |
| 40.00% | 53.60 | 9.75 | 25.74 | 9.76 | 21.84 | 9.44 | 24.57 | 7.36 | 25.61 | 1.57 |
| 45.00% | 30.26 | 7.60 | 30.94 | 9.48 | 30.13 | 9.62 | 31.53 | 8.74 | 29.31 | 2.28 |
| 50.00% | 32.96 | 7.78 | 28.06 | 11.67 | 21.03 | 12.00 | 19.98 | 10.05 | 32.13 | 2.29 |
| Mean | 32.14 | 8.43 | 22.75 | 9.06 | 15.95 | 7.61 | 19.80 | 6.44 | 17.27 | 3.56 |

| Group | NJ | | LG | | MP | | B | | Katz | |
|---------|-------|-------|-------|-------|-------|-------|-------|-------|-------|------|
| Nullif. | E | T | E | T | E | T | E | T | E | T |
| 5.00% | 14.36 | 5.01 | 18.56 | 6.07 | 8.53 | 9.52 | 9.17 | 4.13 | 8.33 | 5.40 |
| 10.00% | 23.36 | 9.27 | 23.09 | 9.35 | 14.53 | 9.32 | 13.18 | 9.30 | 11.34 | 5.06 |
| 15.00% | 27.86 | 10.79 | 25.28 | 10.75 | 17.54 | 10.82 | 15.18 | 10.74 | 12.83 | 3.85 |
| 20.00% | 32.36 | 12.26 | 27.57 | 12.30 | 20.53 | 12.23 | 17.17 | 12.27 | 14.33 | 4.26 |
| 25.00% | 36.86 | 13.84 | 29.75 | 13.64 | 23.53 | 13.76 | 19.17 | 13.82 | 15.84 | 4.19 |
| 30.00% | 41.36 | 15.22 | 32.08 | 15.31 | 26.53 | 15.22 | 21.18 | 15.13 | 17.33 | 5.77 |
| 35.00% | 45.86 | 16.70 | 34.43 | 16.54 | 29.53 | 16.56 | 23.17 | 16.60 | 18.84 | 5.14 |
| 40.00% | 50.36 | 18.02 | 36.70 | 18.25 | 32.53 | 18.33 | 25.18 | 18.06 | 20.33 | 2.97 |
| 45.00% | 54.86 | 19.72 | 38.72 | 19.74 | 35.53 | 19.76 | 27.18 | 19.64 | 21.83 | 3.82 |
| 50.00% | 59.36 | 21.27 | 45.05 | 21.01 | 38.53 | 21.32 | 29.17 | 21.35 | 23.33 | 3.41 |
| Mean | 38.66 | 14.21 | 30.72 | 14.29 | 24.73 | 14.68 | 19.98 | 14.10 | 16.43 | 4.39 |

| Lik./Dis. | NJ | | LG | | MP | | B | | Katz | |
|-----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Nullif. | E | T | E | T | E | T | E | T | E | T |
| 5.00% | 19.49 | 5.49 | 13.50 | 7.91 | 10.34 | 5.19 | 15.19 | 10.84 | 10.75 | 6.26 |
| 10.00% | 24.50 | 9.33 | 20.25 | 9.28 | 18.35 | 9.26 | 20.20 | 9.31 | 13.75 | 5.62 |
| 15.00% | 27.00 | 10.78 | 21.12 | 10.80 | 17.84 | 10.83 | 22.70 | 10.83 | 15.25 | 9.25 |
| 20.00% | 29.49 | 12.18 | 21.95 | 12.17 | 20.34 | 12.25 | 25.20 | 12.20 | 16.75 | 7.68 |
| 25.00% | 31.99 | 13.62 | 22.85 | 13.69 | 22.85 | 13.74 | 27.70 | 13.75 | 18.25 | 6.99 |
| 30.00% | 34.49 | 15.10 | 23.63 | 15.12 | 25.35 | 15.06 | 30.19 | 15.14 | 19.75 | 9.72 |
| 35.00% | 37.00 | 16.73 | 24.50 | 16.64 | 27.84 | 16.63 | 32.70 | 16.51 | 21.25 | 9.48 |
| 40.00% | 39.50 | 17.95 | 25.42 | 18.02 | 30.35 | 18.12 | 35.20 | 18.09 | 22.76 | 16.02 |
| 45.00% | 42.00 | 19.64 | 32.22 | 19.50 | 32.85 | 19.51 | 37.69 | 19.46 | 24.26 | 19.39 |
| 50.00% | 44.50 | 21.34 | 37.04 | 20.85 | 35.35 | 20.91 | 40.19 | 21.19 | 25.75 | 13.73 |
| Mean | 33.00 | 14.22 | 22.65 | 14.40 | 24.15 | 14.15 | 28.70 | 14.73 | 18.85 | 10.41 |

an excerpt of an initial training set on the left; on the right, that training set has been expanded with the features of the neighbours regarding the ‘sends-message’ relationship; note that the weight of the relationship is added as an additional feature to the vector.

Then, the procedure iterates through the set of expansions of the training set and learns a new neural network from each one. It returns the expanded neural network that achieves the smallest error rate together with the training set from which it was learnt and validation set on which the error rate was computed.

3 Experimental Results

We conducted a series of experiments to analyse how Katz performs in practice. The experiments were carried out using a Java 1.7 implementation that was run

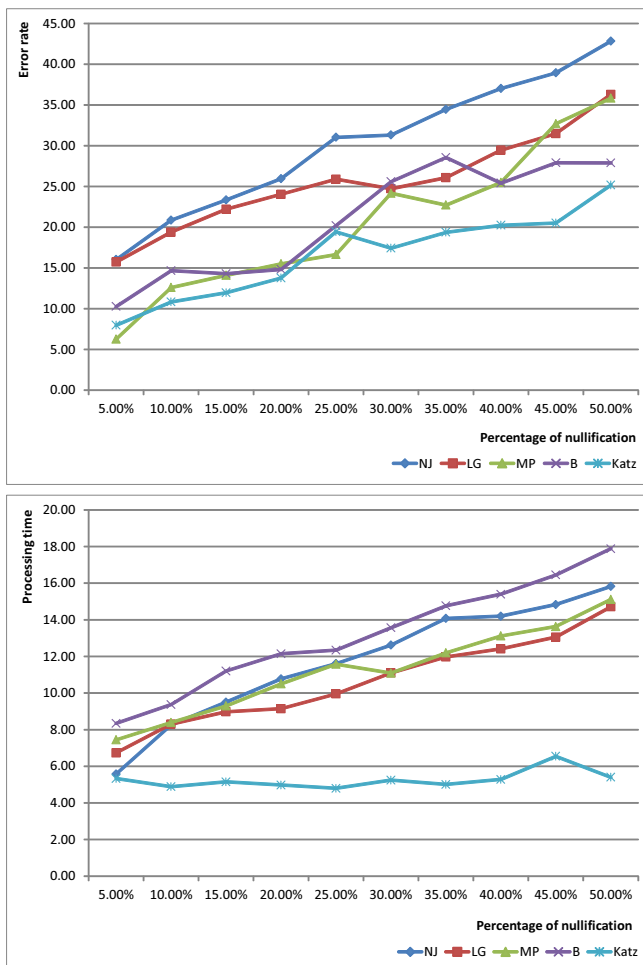


Fig. 4. Graphic summary

on a four-threaded Intel Core i7 computer that ran at 2.93 GHz, had 16 GiB of RAM, Windows 7 Pro 64-bit, Oracle’s Java Development Kit 1.7.9_02, and Weka 3.6.8.

We implemented the general framework by Neville and Jensen [17] (NJ) and then the specific proposals by Lu and Getoor [12] (LG), Macskassy and Provost [13] (MP), and Bhagat et al. [3] (B). Regarding the previous proposals, we considered that a labelling was estabale when no more than 5% of the features changed in an iteration of the method. Regarding Katz, we experimented with several combinations of parameters and kinds of neural networks. We found out that the following values for the parameters work quite well: $\beta = 10$, that is, 10 neural networks learnt for each feature, and $\gamma = 0.25$, that is, 25% of the nodes

available in the social network are used for training purposes and the remaining for validation purposes. Regarding the learning technique, we found out that RBFN networks [4] are the best performing in this context.

The experiments were performed on a dataset that consisted in a dump of our university social network. This network has 56,431 members, each of which is characterised by a profile that includes the following features: age (a natural value), gender (male, female), group (student, lecturer, staff), nationality (Spanish, French, Italian, and so on), school (Computer-Science, Mathematics, Physics, Philology, and so on), likes, and dislikes; other features like name, address, national id or passport were discarded to keep the data anonymous; neither was it very interesting to attempt to predict them. The likes and dislikes are sets of key words that are selected by the members from a list that is computed automatically from the messages post by the members of the network; to deal with them in our experiments, we selected the top 50 key words and created binary features of the form `likes.X` or `dislikes.Y`, where X or Y represents key words. The relationships between the members of the network are the following: posts-to-wall, replies-to-post, forwards-post, sends-message, follows-member, requests-friendship. This dataset was particularly useful because almost every profile has accurate features that are set automatically using the students' registration data or the lecturers' and staff's work contracts, and the likes and dislikes are also selected from sets of pre-computed key words. That is, we had quite a large correctly labelled dataset on which could conduct quite a precise validation.

To evaluate our proposal and compare it to others, we created several datasets from the previous one. They were versions of the original dataset in which we nullified the features of 5% up to 50% nodes that were chosen randomly. This helped us to evaluate how our proposal works and compare it to others in terms of error rate (E) and processing time (T). The error rate was computed as the percentage of wrong predictions; in the case of numeric features a $\pm 10\%$ tolerance threshold was established to consider a prediction wrong. The processing time was measured in CPU plus IO hours, since these timings are far more reliable and stable than user times.

Table 1 shows our results and Figure 4 summarises them using a couple of charts. (The columns that correspond to proposals NJ and LG regarding feature 'age' are empty because these methods cannot be applied to numeric features.) Regarding effectiveness, the first conclusion is that the error rate increases steadily as the percentage of nullification increases, but Katz keeps the smallest global mean in the majority of cases, where global mean refers to the computed mean for each feature for a given nullification percentage, cf. the upper part of Figure 4. To compare the results more precisely, we have computed the tendency lines for each proposal according to the percentage of nullification, which is denoted as N :

| Proposal | Error rate tendency | R² |
|-----------------|----------------------------|----------------------|
| NJ | $2.78N + 14.72$ | 0.99 |
| LG | $1.88N + 15.17$ | 0.93 |
| MP | $2.99N + 4.16$ | 0.95 |
| B | $2.15N + 9.17$ | 0.88 |
| Katz | $1.69N + 7.37$ | 0.92 |

Note that the R^2 coefficient is very good in every case, which means that there is a clear linear tendency in the results. The smallest slope corresponds to Katz, which means that it is the proposal whose error rate increases at the lowest pace as the percentage of nullification increases; it is followed by LG, but note that the error rate of this proposal is roughly double as Katz’s.

Regarding efficiency, the first conclusion is that Katz seems to have a behaviour that is very stable, whereas the other proposals seem to require more processing time as the percentage of nullification increases. To confirm this idea, we have also computed the tendency lines for each proposal, namely:

| Proposal | Processing time tendency | R² |
|-----------------|---------------------------------|----------------------|
| NJ | $1.04N + 5.62$ | 0.98 |
| LG | $0.66N + 6.08$ | 0.98 |
| MP | $0.78N + 6.93$ | 0.97 |
| B | $1.00N + 7.63$ | 0.98 |
| Katz | $0.08N + 4.81$ | 0.95 |

Note that the R^2 coefficient is again very good in every case. The smallest slope corresponds again to Katz, which means that it is the proposal whose processing time increases at a lower pace as the percentage of nullification increases. Note that it is very close to 0.00, which means that the processing time remains almost constant; the reason is that the size of the training sets decrease as the percentage of nullification increases, which makes learning neural networks easier; unfortunately, as the percentage of nullification increases, the number of neighbours that must be explored to keep as a low error rate as possible increases. Katz is followed by the other proposals, which require considerably more processing time since they have to iterate until the labelling is stable enough, which is more and more difficult as the percentage of nullification increases.

4 Related Work

There are two mainstream approaches to the member labelling problem [9], namely: local and global methods. They both work on a graph-based representation of the social network being analysed, where the nodes store member features and the edges keep track of their interactions, but differ in that the former focus on learning local predictors from every member and his or her local neighbourhood, whereas the latter analyse the social network as a whole.

Below, we report on both approaches and discuss on how our proposal improves on them from a conceptual point of view.

Local Methods: These methods can be further classified into instantiations of the Iterative Classification Algorithm by Neville and Jensen [17] or instantiations of the Gibbs Sampling Algorithm by Geman and Geman [8].

The methods that are based on the Iterative Classification Algorithm [17] transform a social network into a dataset of vectors, each of which provides the features of a member’s profile plus some aggregated features that correspond to the members in his or her neighbourhood. They analyse each feature in isolation as follows: they first learn a local predictor from the members whose profiles provide a non-null value for that feature. (Informally, this is commonly referred to as “the member is labelled”.) The predictor is either a regressor or a classifier depending on whether the feature being analysed is numeric or categorical. It is then used to compute the label of the unlabelled members, as long as they have at least a labelled neighbour. Note that labelling a member will likely change the values of the aggregated features in the neighbourhood, so the labelling process needs to be repeated iteratively until the labels do not change dramatically or do not change at all. The previous idea has been instantiated many times in the literature, the difference being the kind of predictor used: Neville and Jensen [17] used Naive-Bayes predictors, Lu and Getoor [12] used logistic regression, Macskassy and Provost [13] used a voting approach, and Bhagat et al. [3] and McDowell et al. [16] used k -nearest neighbours. Recently, Cataltepe et al. [6] have used different types of predictors for member features and neighbourhood features, which are then combined to produce an ensemble predictor.

The methods that are based on the Gibbs Sampling Algorithm [8] work in four phases, namely: bootstrapping, burn-in, collecting, and labelling. In the bootstrapping phase, they learn a predictor in a way that is very similar to the methods that are based on the Iterative Classification Algorithm, and then use it to label the unlabelled members. Then, the burn-in phase is repeated a number of times; in each repetition, the members that were initially unlabelled are randomly ordered and then new labels are computed using a predictor, which can be the same that was used in the bootstrapping phase or a new one [14]. In the sample collection phase, the process is repeated a pre-defined number of times and the count of labels assigned to each member is computed. Finally, in the labelling phase, the members that were initially unlabelled are assigned the most likely label according to the counts that were computed in the previous phase. Both McDowell et al. [15] and Macskassy and Provost [14] have instantiated this idea; the former used Naive-Bayes and k -nearest neighbours and the latter used different combinations of predictors.

Global Methods: The most common methods in this category are based on random walks and optimisation.

A random walk on a graph is a very special case of a Markov chain. The core idea was introduced by Zhu et al. [23]: they rely on a transition matrix P that encodes the probability that a random walk proceeds between any two members of a social network using their relationships. Given an unlabelled member, the method assigns it the most common label out of the members that can be reached from it using random walks. That is, it requires to compute an approximation

to the closure of P ; in practice, the procedure stops when the closure is stable enough, that is, when the probabilities do not change dramatically or do not change at all. This general idea has been instantiated many times in the literature, with some variations, namely: Szummer and Jaakkola [20] start their walks from unlabelled nodes and consider only labelled nodes that can be reached in a pre-defined number of steps; contrarily, Callut et al. [5] consider only walks that start and end in a labelled node regarding a given feature, but do not go through labelled nodes regarding the same feature in the intermediate steps.

The methods that are based on optimisation map the problem into a number of constraints plus an objective function for which the global maximum or the minimum needs to be found. The main problem with this approach is that it typically results in an optimisation problem with a number of variables that very typically exceeds the limits of current solvers [2]. Thus, the authors who have instantiated this idea have focused on approximating the results as efficiently as possible [7, 10, 18, 21, 22].

Discussion: The main difference amongst Katz and the other methods in the literature is that it does not analyse a pre-defined neighbourhood, neither treats it all of the members of a social network, all of the features, and all of their relationships equally. It first learns a predictor using the member’s features only and then tries to improve it by searching the most adequate neighbours and features using the error rate as the only search heuristic. It is then a hybrid approach since it does not focus on a local or a global neighbourhood, but finds the most appropriate for each feature. This, in turn, leads to a method that needs not be applied repeatedly in order to label a social network very well and very efficiently, as our experimental results prove.

A key feature is that Katz does not rely on a single predictor, but on several predictors that are learnt from different parts of the social network in order to adapt better to its peculiarities and reduce the error rate. The methods that are based on the Gibbs Sampling Algorithm also label a member using several predictions, but the difference is that Katz makes predictions using several predictors, not the different predictions that are computed using a single predictor from randomising the order in which it is applied to the members of a network. This has resulted in a method that has proven experimentally to achieve a very low error rate, even in cases in which there are many null labels, which have proven to be very difficult to deal with using other proposals.

Another strong point is that Katz relies on using neural networks, which have been proven to learn good models from complex data like ours [4]. This allows it to be applied to any kind of features, whereas some methods in the literature can only be applied to a kind of features. For instance, the proposals by Neville and Jensen [17] and Lu and Getoor [12] can only be used with categorical features because, unfortunately, the technique on which they rely does not deal with numeric features.

Finally, a common weak feature of the existing methods is that they tend to be inefficient. Iterative methods typically require an unbounded number of iterations for a labelling to become stable; Macskassy and Provost [14] proposed to

limit the number of iterations of their proposal and they experimentally proved that roughly 2,200 iterations was enough to achieve good results, but it is not completely clear if this figure works in general. Global methods typically lead to problems that are not computationally tractable and then can only be approximated. Our experiments prove that Katz is very efficient in practice and more scalable than other proposals in the literature.

5 Conclusions

In this paper, we have introduced Katz, which is a new hybrid proposal to solve the problem of labelling the members of a social network, that is, to infer the values of a member's profile missing features using his or her neighbourhood. It is based on neural-network predictors that are computed from a member's profile and an unbounded neighbourhood, which makes it very effective; furthermore, it does not require to iterate multiple times until the labelling converges, which makes it very efficient. Our experiments on a real-world university social network prove that it outperforms other proposals in the literature.

Acknowledgments. Our work was supported by the European Commission (FED-ER), the Spanish and the Andalusian R&D&I programmes (grants TIN2007-64119, P07-TIC-2602, P08-TIC-4100, TIN2008-04718-E, TIN2010-21744, TIN2010-09809-E, TIN2010-10811-E, TIN2010-09988-E, TIN2011-15497-E, and TIN2013-40848-R). We are grateful to our support staff for their help to obtain a dump of our university social network. We also thank Opileak.com for sharing their social media analysis platform with us.

References

1. Aggarwal, C.C.: Social Network Data Analytics. Springer (2011)
2. Bhagat, S., Cormode, G., Muthukrishnan, S.: Node classification in social networks. In: Social Network Data Analytics, pp. 115–148. Springer (2011)
3. Bhagat, S., Cormode, G., Rozenbaum, I.: Applying Link-Based Classification to Label Blogs. In: Zhang, H., Spiliopoulou, M., Mobasher, B., Giles, C.L., McCallum, A., Nasraoui, O., Srivastava, J., Yen, J. (eds.) WebKDD 2007. LNCS, vol. 5439, pp. 97–117. Springer, Heidelberg (2009)
4. Bianchini, M., Maggini, M., Jain, L.C.: Handbook on Neural Information Processing, vol. 49. Springer (2013)
5. Callut, J., Françoise, K., Saerens, M., Dupont, P.E.: Semi-supervised Classification from Discriminative Random Walks. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part I. LNCS (LNAI), vol. 5211, pp. 162–177. Springer, Heidelberg (2008)
6. Cataltepe, Z., Sonmez, A., Baglioglu, K., Erzan, A.: Collective Classification Using Heterogeneous Classifiers. In: Perner, P. (ed.) MLDM 2011. LNCS, vol. 6871, pp. 155–169. Springer, Heidelberg (2011)

7. Chakrabarti, S., Dom, B., Indyk, P.: Enhanced hypertext categorization using hyperlinks. In: SIGMOD Conference, pp. 307–318 (1998)
8. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.* **6**(6), 721–741 (1984)
9. Kazienko, P., Kajdanowicz, T.: Collective classification, structural features. In: *Encyclopedia of Social Network Analysis and Mining*, pp. 156–168 (2014)
10. Kleinberg, J.M., Tardos, E.: Approximation algorithms for classification problems with pairwise relationships: metric labeling and Markov random fields. *J. ACM* **49**(5), 616–639 (2002)
11. Lenhart, A., Madden, M.: Teens, privacy and online social networks. Tech. rep., Pew Internet (2007)
12. Lu, Q., Getoor, L.: Link-based classification. In: *ICML*, pp. 496–503 (2003)
13. Macskassy, S.A., Provost, F.: A simple relational classifier. In: *Proceedings of the SIGKDD 2003 2nd Workshop on Multi-Relational Data Mining* (2003)
14. Macskassy, S.A., Provost, F.J.: Classification in networked data: a toolkit and a univariate case study. *Journal of Machine Learning Research* **8**, 935–983 (2007)
15. McDowell, L., Gupta, K.M., Aha, D.W.: Cautious inference in collective classification. In: *AAAI*, pp. 596–601 (2007)
16. McDowell, L., Gupta, K.M., Aha, D.W.: Cautious collective classification. *Journal of Machine Learning Research* **10**, 2777–2836 (2009)
17. Neville, J., Jensen, D.: Iterative classification in relational data. In: *AAAI 2000 Workshop on Learning Statistical Models from Relational Data*, pp. 42–49 (2000)
18. Neville, J., Jensen, D.: Dependency networks for relational data. In: *ICDM*, pp. 170–177 (2004)
19. Singla, P., Richardson, M.: Yes, there is a correlation: from social networks to personal behavior on the Web. In: *WWW*, pp. 655–664 (2008)
20. Szummer, M., Jaakkola, T.: Partially labeled classification with Markov random walks. In: *NIPS*, pp. 945–952 (2001)
21. Taskar, B., Abbeel, P., Koller, D.: Discriminative probabilistic models for relational data. In: *UAI*, pp. 485–492 (2002)
22. Yedidia, J.S., Freeman, W.T., Weiss, Y.: Generalized belief propagation. In: *NIPS*, pp. 689–695 (2000)
23. Zhu, X., Ghahramani, Z., Lafferty, J.D.: Semi-supervised learning using Gaussian fields and harmonic functions. In: *ICML*, pp. 912–919 (2003)