Relational Classification for Personalized Tag Recommendation

Leandro Balby Marinho, Christine Preisach, and Lars Schmidt-Thieme

Information Systems and Machine Learning Lab (ISMLL)
Samelsonplatz 1, University of Hildesheim, D-31141 Hildesheim, Germany
{marinho,preisach,schmidt-thieme}@ismll.uni-hildesheim.de
http://www.ismll.uni-hildesheim.com/

Abstract. Folksonomy data is relational by nature, and therefore methods that directly exploit these relations are prominent for the tag recommendation problem. Relational methods have been successfully applied to areas in which entities are linked in an explicit manner, like hypertext documents and scientific publications. For approaching the graph-based tag recommendation task of the ECML PKDD Discovery Challenge 2009, we propose to turn the folksonomy graph into a homogeneous post graph and use relational classification techniques for predicting tags. Our approach features adherence to multiple kinds of relations, semi-supervised learning and fast predictions.

1 Introduction

One might want tag recommendations for several reasons, as for example: simplifying the tagging process for the user, exposing different facets of a resource and helping the tag vocabulary to converge. Given that users are free to tag, i.e., the same resource can be tagged differently by different people, it is important to personalize the recommended tags for an individual user.

Tagging data forms a ternary relation between users, resources and tags, differently from typical recommender systems in which the relation is usually binary between users and resources. The best methods presented so far explore this ternary relation to compute tag predictions, either by means of tensor factorization [8] or PageRank [3], on the hypergraph induced by the ternary relational data. We, on the other hand, propose to explore the underlying relational graph between posts by means of relational classification.

In this paper we describe our approaches for addressing the graph-based tag recommendation task of the ECML PKDD Discovery Challenge 2009. We present two basic algorithms: *PWA* (probabilistic weighted average)*, an iterative relational classification algorithm enhanced with relaxation labelling, and *WA* (weighted average)*, an iterative relational classification method without relaxation labelling. These methods feature: adherence to multiple kinds of relations, training free, fast predictions, and semi-supervised classification. Semi-supervised classification is particularly appealing because it allows us to evtl. benefit from the information contained in the test dataset. Furthermore, we propose to combine these methods through unweighted voting.

The paper is organized as follows. Section 2 presents the notation used throughout the paper. In Section 3 we show how we turned the folksonomy into a post relational graph. Section 4 introduces the individual classifiers and the ensemble technique we used. In Section 5 we elaborate on the evaluation and experiments conducted for tuning the parameters of our models, and report the results obtained on the test dataset released for the challenge. The paper closes with conclusions and directions for future work.

2 Notation

Foksonomy data usually comprises a set of users U, a set of resources R, a set of tags T, and a set Y of ternary relations between them, i.e., $Y \subseteq U \times R \times T$.

Let

$$X := \{(u,r) \,|\, \exists t \in T : (u,r,t) \in Y\}$$

be the set of all unique user/resources combinations in the data, where each pair is called a *post*. For convenience, let $T(x = (u,r)) := \{t \in T \mid (u,r,t) \in Y\}$ be the set of all tags assigned to a given post $x \in X$. We consider train/test splits based on posts, i.e., $X_{\text{train}}, X_{\text{test}} \subset X$ disjoint and covering all of X:

$$X_{\text{train}} \dot{\cup} X_{\text{test}} = X$$

For training, the learner has access to the set X_{train} of training posts and their true tags $T|_{X_{\text{train}}}$. The tag recommendation task is then to predict, for a given $x \in X_{\text{test}}$, a set $\hat{T}(x) \subseteq T$ of tags that are most likely to be used by the resp. user for the resp. resource.

3 Relation Engineering

We propose to represent folksonomy data as a homogeneous, undirected relational graph over the post set, i.e., $G := (X, E)$ in which edges are annotated with a weight $w : X \times X \rightarrow \mathbb{R}$ denoting the strength of the relation. Besides making the input data more compact – we have only a binary relation $\mathcal{R} \subseteq X \times X$ between objects of the same type – this representation will allow us to trivially cast the problem of personalized tag recommendations as a relational classification problem.

Relational classifiers usually consider, additionally to the typical attribute-value data of objects, relational information. A scientific paper, for example, can be connected to another paper that has been written by the same author or because they share common citations. It has been shown in many classification problems that relational classifiers perform better than purely attribute-based classifiers [1, 4, 6].

In our case, we assume that posts are related to each other if they share the same user: $\mathcal{R}_{\text{user}} := \{(x,x') \in X \times X \,|\, user(x) = user(x')\}$, the same resource: $\mathcal{R}_{\text{res}} := \{(x,x') \in X \times X | res(x) = res(x')\}$, or either share the same user or resource: $\mathcal{R}_{\text{user}}^{\text{res}} := \mathcal{R}_{\text{user}} \cup \mathcal{R}_{\text{res}}$ (see Figure 1). For convenience, let user(x) and res(x) denote the user and resource of post x respectively. Thus, each post is connected to each other either in terms of other users that tagged the same resource, or the resources tagged by the same user. Weights are discussed in Section 4.

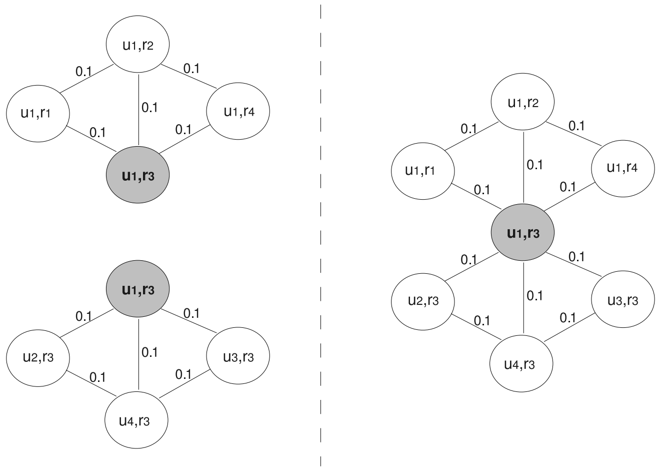

Fig. 1. $\mathcal{R}_{\text{user}}$ (top left), $\mathcal{R}_{\text{res}}$ (bottom left) and $\mathcal{R}_{\text{user}}^{\text{res}}$ (right) of a given test post (nodes in grey)

Note that it may happen that some of the related posts belong themselves to the test dataset, allowing us to evtl. profit from the unlabeled information of test nodes through, e.g., collective inference (see Section 4). Thus, differently from other approaches (e.g., [3, 8]) that are only restricted to X_{train}, we can also exploit the set X_{test} of test posts, but of course not their associated true tags.

Now, for a given $x \in X_{\text{test}}$, one can use the tagging information of related instances to estimate $\hat{T}(x)$. A simple way to do that is, e.g., through tag frequencies of related posts:

$$P(t|x) := \frac{|\{x' \in N_x | t \in T(x')\}|}{|N_x|}, \quad x \in X, \, t \in T \tag{1}$$

while N_x is the neighborhood of x:

$$N_x := \{x' \in X \,|\, (x, x') \in \mathcal{R}, \, T(x) \neq \emptyset\} \tag{2}$$

In section 4 we will present the actual relational classifiers we have used to approach the challenge.

4 Relational Classification for Tag Recommendation

We extract the relational information by adapting simple statistical relational methods, usually used for classification of hypertext documents, scientific publications or movies, to the tag recommendation scenario. The aim is to recommend tags to users by using the neighborhood encoded in the homogeneous graph $G(X, E)$. Therefore we described a very simple method in eq. (1), where the probability for a tag $t \in T$ given a node x (post) is computed by counting the frequency of neighboring posts $x' \in N_x$ that have used the same tag t. In this case the strength of the relations is not taken into account, i.e., all considered neighbors of x have the same influence on the probability of tag t

given x. But this is not an optimal solution, the more similar posts are to each other the higher the weight of this edge should be.

Hence, a more suitable relational method for tag recommendation is the *WeightedAverage (WA)* which sums up all the weights of posts $x' \in N_x$ that share the same tag $t \in T$ and normalizes this by the sum over all weights in the neighborhood.

$$P(t|x) = \frac{\sum_{x' \in N_x | t \in T(x')} w(x, x')}{\sum_{x' \in N_x} w(x, x')} \tag{3}$$

Thus, *WA* does only consider neighbors that belong to the training set.

A more sophisticated relational method that takes probabilities into account is the *probabilistic weighted average (PWA)*, it calculates the probability of t given x by building the weighted average of the tag probabilities of neighbor nodes $x' \in N_x$:

$$P(t|x) = \frac{\sum_{x' \in N_x} w(x, x')P(t|x')}{\sum_{x' \in N_x} w(x, x')} \tag{4}$$

Where $P(t|x') = 1$ for $x' \in X_{train}$, i.e., we are only exploiting nodes contained in the training set (see eq. (2)). We will see in the next paragraph how one can exploit these probabilities in a more clever way. Both approaches have been introduced in [5] and applied to relational datasets.

Since we want to recommend more than one tag we need to cast the tag recommendation problem as a multilabel classification problem, i.e., assign one or more classes to a test node. We accomplish the multilabel problem by sorting the calculated probabilities $P(t|x)$ for all $x \in X_{\text{test}}$ and recommend the top n tags with highest probabilities.

The proposed relational methods could either be applied on $\mathcal{R}^{\text{res}}_{\text{user}}$, i.e., the union of the user and resource relation or on each relation $\mathcal{R}_{\text{user}}$, $\mathcal{R}_{\text{res}}$ individually. If applied on each relation the results could be combined by using ensemble techniques.

4.1 Semi-Supervised Learning

As mentioned before, we would like additionally, to exploit unlabeled information contained in the graph and use the test nodes that have not been tagged yet, but are related to other nodes. This can be achieved by applying collective inference methods, being iterative procedures, which classify related nodes simultaneously and exploit relational autocorrelation and unlabeled data. Relational autocorrelation is the correlation among a variable of an entity to the same variable (here the class) of a related entity, i.e., connected entities are likely to have the same classes assigned. Collective Classification is semi-supervised by nature, since one exploits the unlabeled part of the data. One of this semi-supervised methods is relaxation labeling [1], it can be formally expressed as:

$$P(t|x)^{(i+1)} = M(P(t|x')^{(i)}_{x' \in N_x}) \tag{5}$$

We first initialize the unlabeled nodes with the prior probability calculated using the train set, then compute the probability of tag t given x iteratively using a relational classification method M based on the neighborhood N_x in the inner loop. The procedure stops when the algorithm converges (i.e., the difference of the tag probability between iteration i and $i + 1$ is less than a very small ϵ) or a certain number of iterations is reached.

We used eq. (4) as relational method inside the loop, where we do not require that the neighbors x' are in the training set, but are using the probabilities of unlabeled nodes. For *PWA* this means that in each iteration we use the probabilities of the neighborhood estimated in the previous iteration collectively. *PWA* combined with collective inference is denoted as *PWA** in the following.

For *WeightedAverage* we did not use relaxation labeling but applied a so called *one-shot-estimation* [5, 7]. We did only use the neighbors with known classes, i.e., in the first iteration we exploit only nodes from the training set, while in the next iteration we used also test nodes that have been classified in the previous iterations. The procedure stops when all test nodes could be classified or a specific number of iterations is reached. Hence, the tag probabilities are not being re-estimated like for the relaxation labeling but only estimated once. Thus, *WA* combined with the one-shot-estimation procedure is denoted as *WA**.

4.2 Ensemble

Ensemble classification may lead to significant improvement on classification accuracy, since uncorrelated errors made by the individual classifiers are removed by the combination of different classifiers [2, 6]. Furthermore, ensemble classification reduces variance and bias.

We have decided to combine *WA** and *PWA** through a simple unweighted voting, since voting performs particularly well when the results of individual classifiers are similar; as we will see in Section 5, *WA** and *PWA** yielded very similar results in our holdout set.

After performing the individual classifiers, we receive probability distributions for each classifier K_l as output and build the arithmetic mean of the tag-assignment probabilities for each test post and tag:

$$P(t|x) = \frac{1}{L} \cdot \sum_{l=1}^{L} P_l(t|x), \quad L := |K_l | P_l(t|x) \neq 0, \ t \in T| \tag{6}$$

4.3 Weighting Schemes

The weight w in eq. (3) and (4) is an important factor in the estimation of tag probabilities, since it describes the strength of the relation between x and x'. There are several ways to estimate these weights:

1. For two nodes $(x, x') \in \mathcal{R}_{\text{res}}$, compute their similarity by representing x and x' as user-tag profile vectors. Each component of the profile vector corresponds to the count of co-occurrences between users and tags:

$$\phi^{\text{user-tag}} := (|Y \cap (\{\text{user}(x)\} \times R \times \{t\})|)_{t \in T}$$

2. Similarly to 1, for two nodes $(x, x') \in \mathcal{R}_{\text{user}}$, the node similarity is computed by representing x and x' as resource-tag profile vectors:

$$\phi^{\text{res-tag}} := (|Y \cap (U \times \{\text{res}(x)\} \times \{t\})|)_{t \in T}$$

3. Similar to 2, but x and x' are represented as resource-user profile vectors where each component corresponds to the count of co-occurrences between resources and users:

$$\phi^{\text{res-user}} := (|Y \cap (\{u\} \times \{\text{res}(x)\} \times T)|)_{u \in U}$$

4. The same as in 1, but the node similarity is computed w.r.t. to user-resource profile vectors:

$$\phi^{\text{user-res}} := (|Y \cap (\{\text{user}(x)\} \times \{r\} \times T)|)_{r \in R}$$

The edge weight is finally computed by applying the cosine similarity over the desired profile vectors:

$$\text{sim}(\phi(x), \phi(x')) := \frac{\langle \phi(x), \phi(x') \rangle}{\|\phi(x)\| \|\phi(x')\|} \tag{7}$$

In our experiments we basically used the scheme 1, since there is no new user in the data and therefore we can always build user-tag profile vectors.

5 Evaluation

All the results presented in this section are reported in terms of F1-score, the official measure used by the graph-based tag recommendation task of the ECML PKDD Discovery Challenge 2009. For a given $x \in X_{\text{test}}$ the F1-Score is computed as follows:

$$\text{F1-score}\left(\hat{T}(x)\right) = \frac{2 \cdot \text{Recall}\left(\hat{T}(x)\right) \cdot \text{Precision}\left(\hat{T}(x)\right)}{\text{Recall}\left(\hat{T}(x)\right) + \text{Precision}\left(\hat{T}(x)\right)} \tag{8}$$

Although the methods presented in Section 4 usually do not have free parameters, we realized that $\mathcal{R}_{\text{user}}$ and $\mathcal{R}_{\text{res}}$ can have a different impact in the recommendation quality (cf. Figures 2 and 3), and thereby we introduced a parameter to reward the best relations in $\mathcal{R}^{\text{res}}_{\text{user}}$ by a factor $c \in \mathbb{N}$: if $\mathcal{R}_{\text{res}}$ yields better recommendations than $\mathcal{R}_{\text{user}}$ for example, all edge weights in $\mathcal{R}^{\text{res}}_{\text{user}}$ that refer to $\mathcal{R}_{\text{res}}$ are multiplied by c.

For searching the best c value we performed a greedy search over the factor range $\{1, ..., 4\}$ on a holdout set created by randomly selecting 800 posts from the training data. Tables 1 and 2 show the characteristics of the training and test/holdout datasets respectively. Figure 2 presents the results of *WA*-Full*[1], i.e., *WA** performed over $\mathcal{R}^{\text{res}}_{\text{user}}$, for different c values on the holdout set according to the F1-score. We also plot the results of *WA*-Res* and *WA*-Usr* (i.e., *WA** on $\mathcal{R}_{\text{res}}$ and $\mathcal{R}_{\text{user}}$ resp.).

After finding the best c value on the holdout set, we applied *WA*-Full*, *PWA*-Full*, and the ensemble (c.f. eq. 6) to the challenge test dataset (see Figure 3). Note that the

[1] Since the results of *PWA** and *WA** are very similar, we just report on *WA**.

dataset	$\|U\|$	$\|R\|$	$\|T\|$	$\|Y\|$	$\|X\|$
BibSonomy	1,185	22,389	13,276	253,615	64,406

Table 1. Characteristics of 2-core BibSonomy.

dataset	$\|U\|$	$\|R\|$	$\|X_{\text{test}}\|$
Holdout	292	788	800
Challenge test	136	667	778

Table 2. Characteristics of the holdout set and the challenge test dataset.

results on the challenge test dataset are much lower than those on the holdout set. It may indicate that either our holdout set was not a good representative of the population or that the challenge test dataset represents a concept drift. We plan to further investigate the reasons underlying this large deviation.

According to the rules of the challenge, the F1-score is measured over the Top-5 recommended tags, even though one is not forced to always recommend 5 tags. This is an important remark because if one recommends more tags than the true number of tags attached to a particular test post, one can lower precision. Therefore, we estimate the number of tags to be recommended to each test post by taking the average number of tags used by each test user to his resources. If a given test user has tagged his resources with 3 tags in average, for example, we recommend the Top-3 tags delivered by our algorithms for all test posts in which this user appears.

6 Conclusions

In this paper we proposed to approach the graph-based tag recommendation task of the ECML PKDD Discovery Challenge 2009 by means of relational classification. We first turned the usual tripartite graph of social tagging systems into a homogeneous post graph, whereby simple statistical relational methods can be easily applied. Our methods are training free and the prediction runtime only depends on the number of neighbors and tags, which is fast since the training data is sparse. The models we presented also incorporate a semi-supervised component that can evtl. benefit from test data. We presented two relational classification methods, namely *WA** and *PWA**, and one ensemble based on unweighted voting over the tag probabilities delivered by these methods.

We also introduced a parameter in order to reward more informative relations, which was learned through a greedy search in a holdout set.

In future work we want to investigate new kinds of relations between the posts (e.g. content-based), other ensemble techniques, and new methods for automatically learning more informative weights.

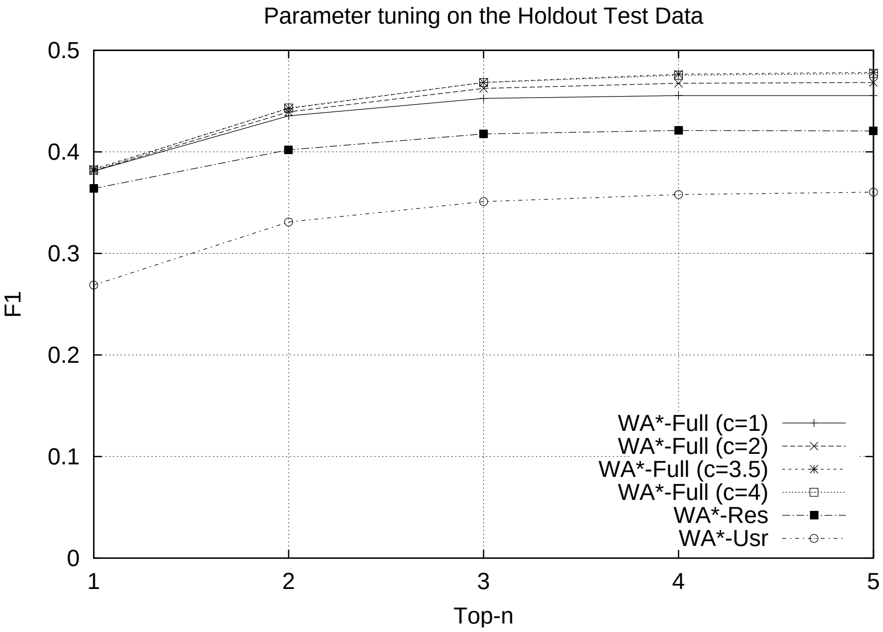

Fig. 2. Parameter search of *WA*-Full* in a holdout set. Best c value found equals 3.5

7 Acknowledgements

This work is supported by CNPq, an institution of Brazilian Government for scientific and technologic development, and the X-Media project (www.x-media-project.org) sponsored by the European Commission as part of the Information Society Technologies (IST) programme under EC grant number IST-FP6-026978. The authors also gratefully acknowledge the partial co-funding of their work through the European Commission FP7 project MyMedia (www.mymediaproject.org) under the grant agreement no. 215006. For your inquiries please contact info@mymediaproject.org.

References

1. Soumen Chakrabarti, Byron E. Dom, and Piotr Indyk. Enhanced hypertext categorization using hyperlinks. In Laura M. Haas and Ashutosh Tiwary, editors, *Proceedings of SIGMOD-98*, pages 307–318. ACM Press, New York, US, 1998.
2. Thomas G. Dietterich. Machine-learning research: Four current directions. *The AI Magazine*, 18(4):97–136, 1998.
3. Robert Jaeschke, Leandro Marinho, Andreas Hotho, Lars Schmidt-Thieme, and Gerd Stumme. Tag recommendations in social bookmarking systems. *AI Communications*, pages 231–247, 2008.
4. Qing Lu and Lise Getoor. Link-based classification using labeled and unlabeled data. icml 2003 workshop on the continuum from labeled to unlabeled data. In *in Machine Learning and Data Mining*, 2003.

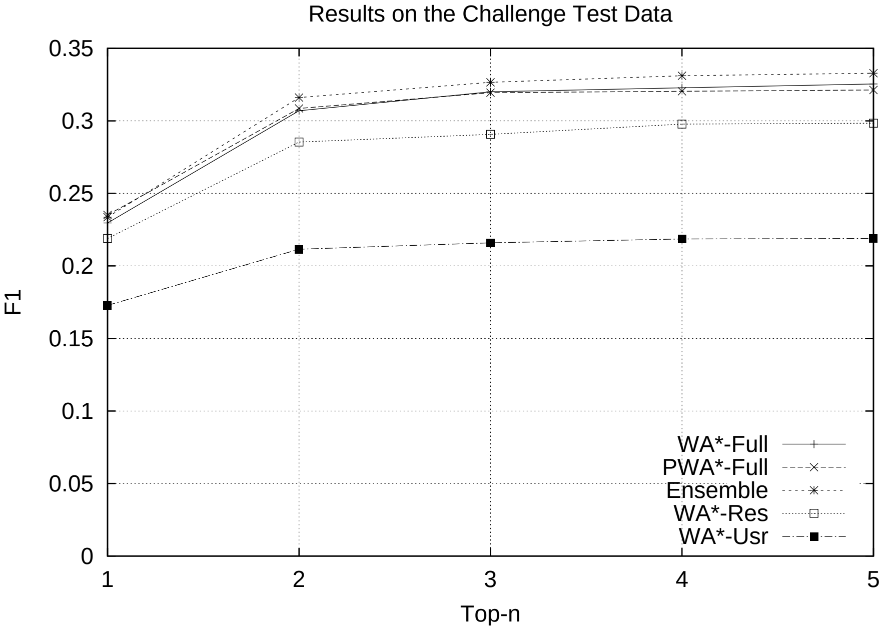

Fig. 3. Results in the challenge test datatest.

5. S. Macskassy and F. Provost. A simple relational classifier. In *Proceedings of the 2nd Workshop on Multi-Relational Data Mining, KDD2003*, pages 64–76, 2003.
6. Christine Preisach and Lars Schmidt-Thieme. Relational ensemble classification. In *ICDM '06: Proceedings of the Sixth International Conference on Data Mining*, pages 499–509, Washington, DC, USA, 2006. IEEE Computer Society.
7. Christine Preisach and Lars Schmidt-Thieme. Ensembles of relational classifiers. *Knowledge and Information Systems*, 14:249–272, 2007.
8. Steffen Rendle, Leandro B. Marinho, Alexandros Nanopoulos, and Lars Schimdt-Thieme. Learning optimal ranking with tensor factorization for tag recommendation. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 727–736. ACM, 2009.