Collaborative Tag Recommendation System based on Logistic Regression [⋆]

E. Montañés[1], J. R. Quevedo[2], I. Díaz[1], and J. Ranilla[1]
{montaneselena,quevedo,sirene,ranilla}@uniovi.es

[1] Computer Science Department, University of Oviedo (Spain)
[2] Artificial Intelligence Center, University of Oviedo (Spain)

Abstract. This work proposes an approach to collaborative tag recommendation based on a machine learning system for probabilistic regression. The goal of the method is to support users of current social network systems by providing a rank of new meaningful tags for a resource. This system provides a ranked tag set and it feeds on different posts depending on the resource for which the recommendation is requested and on the user who requests the recommendation. Different kinds of collaboration among users and resources are introduced. That collaboration adds to the training set additional posts carefully selected according to the interaction among users and/or resources. Furthermore, a selection of post using scoring measures is also proposed including a penalization of oldest post. The performance of these approaches is tested according to F_1 but just considering at most the first five tags of the ranking, which is the evaluation measure proposed in ECML PKDD Discovery Challenge 2009. The experiments were carried out over two different kind of data sets of Bibsonomy folksonomy, core and no core, reaching a performance of 26.25% for the former and 6.98% for the latter.

1 Introduction

Recently, tag recommendation has been gained popularity as a result of the interest of social networks. This task can be defined as the process of providing promising keywords to the users of a social network in the presence of resources of the network itself. These keywords are called tags and the users can assign them to the resources [12]. Tagging resources present several advantages: they facilitate other users a later search and browsing, they consolidate the vocabulary of the users, they provide annotated resources and they build user profiles. An option to perform such task could be to provide tags manually to each user, but this time-consuming and tedious task could be avoided using a Tag Recommender System (TRS).

Folksonomies are examples of large-scale systems that take advantage of a TRS. A Folksonomy [9] is a set of posts included by a user who has attached

[⋆] This research has been partially supported by the MICINN grants TIN2007-61273 and TIN2008-06247.

a resource through a tag. Generally, each resource is specific to the user who added it to the system, as `Flickr`, which shares photos, or `BibSonomy`, which shares bookmarks and bibtex entries. However, for some types of networks identical resources can be added to the system by different users, as is the case of `Del.icio.us` which shares bookmarks.

This paper proposes an approach to collaborative tag recommendation based on a logistic regression learning process. The work starts from the hypothesis that a learning process improves the performance of the recommendation task. It explores several information the learner feeds on. In this sense, the training set depends on each test post and it is specifically built for each of them. In addition, a set of additional posts carefully selected are added to the training set according to the collaboration among users and/or resources.

The remainder of the paper is structured as follows. Section 2 presents background information about tag recommendation in social networks. Our approach is put in context in Section 3 while the proposed method is provided in Sections 4, 5 and 6. Section 7 describes the performance evaluation metric. The results conducted on public data sets are presented and analyzed in Section 8. Finally, Section 9 draws conclusions and points out some possible challenges to address in the near future.

2 Related Work

Different approaches have been proposed to support the users during the tagging process depending on the purpose they were built for. Some of them makes recommendations by analyzing content [1], analyzing tag co-occurrences [23] or studying graph-based approaches [10].

Brooks et al. [4] analyze the effectiveness of tags for classifying blog entries by measuring the similarity of all articles that share a tag. Jäschke et al. [10] adapt a user-based collaborative filtering as well as a graph-based recommender built on top of FolkRank. TagAssist [24] recommends tags of blog posts relying upon tags previously attached to similar resources.

Lee and Chun [14] propose an approach based on a hybrid artificial neural network. ConTag [1] is an approach based on Semantic Web ontologies and Web 2.0 services. CoolRank [2] utilizes the quantitative value of the tags that users provide for ranking bookmarked web resources. Vojnovic et al. [27] keep in view collaborative tagging systems where users can attach tags to information objects.

Basile et al.[3] propose a smart TRS able to learn from past user interaction as well as from the content of the resources to annotate. Krestel and Chen [13] raise TRP-Rank (Tag-Resource Pair Rank), an algorithm to measure the quality of tags by manually assessing a seed set and propagating the quality through a graph. Zhao et al. [29] propose a collaborative filtering approach based on the semantic distance among tags assigned by different users to improve the effectiveness of neighbor selection.

Katakis et al. [12] model the automated tag suggestion problem as a multilabel text classification task. If the item to tag exists in the training set, then it suggests the most popular tags for the item. Tatu et al. [25] use textual content associated with bookmarks to model users and documents.

Sigurbjornsson et al. [23] present the results by means of a tag characterization focusing on how users tags photos of `Flickr` and what information is contained in the tagging.

Most of these systems require information associated with the content of the resource itself [3]. Others simply suggest a set of tags as a consequence of a classification rather than providing a ranking of them [12]. Some of them require a large quantity of supporting data [23]. The purpose of this work is to avoid these drawbacks using a novel approach which establishes a tag ranking through a machine learning approach based on logistic regression.

3 Tag Recommender Systems (TRS)

A folksonomy is a tuple $F := (\mathcal{U}, \mathcal{T}, \mathcal{R}, \mathcal{Y})$ where $\mathcal{U}$, $\mathcal{T}$ and $\mathcal{R}$ are finite sets, whose elements are respectively called users, tags and resources, and $\mathcal{Y}$ is a ternary relation between them, i. e., $\mathcal{Y} \subseteq \mathcal{U} \times \mathcal{T} \times \mathcal{R}$, whose elements are tag assignments (posts). When a user adds a new or existing resource to a folksonomy, it could be helpful to recommend him/her some relevant tags.

TRS usually take the users, resources and the ratings of tags into account to suggest a list of tags to the user. According to [15], a TRS can briefly be formulated as a system that takes a given user $u \in \mathcal{U}$ and a resource $r \in \mathcal{R}$ as input and produces a set $\mathcal{T}(u, r) \subset \mathcal{T}$ of tags as output.

Jäschke et al. in [10] define a post of a folksonomy as a user, a resource and all tags that this user has assigned to that resource. This work slightly modifies this definition in the sense that it restricts the set of tags to the tags used simultaneously by a user in order to tag a resource.

There are some simple but frequently used TRS [10] based on providing a list of ranked tags extracted from the set of posts connected with the current annotation.

- MPT (Most Popular Tags): For each tag t_i, the posts with t_i are counted and the top tags (ranked by occurrence count) are utilized as recommendations.
- MPTR (Most Popular Tags by Resource): The number of posts in which a tag occurs together with r_i is counted for each tag. The tags occurring most often together with r_i are then proposed as recommendations.
- MPTU (Most Popular Tags by User): The number of posts in which a tag occurs together with u_i is counted for each tag. The tags occurring most often together with u_i are then proposed as recommendations.
- MPTRU (Most Popular Tags by Resource or User): The number of posts in which a tag occurs together either with r_i or u_i is counted for each tag. The tags occurring most often together with either r_i or u_i are taken as recommendations.

Our hypothesis is that the introduction of a learning system is expected to improve their performance of these systems. These are the key points of the system:

- The training set depends on each test post and it is specifically built for each of them. Section 4 explains the way of building the initial training set and the example representation.
- Several training sets are built according to different kinds of collaboration among users and resources, performing post selection adapting several scoring measures and penalizing oldest posts. Afterwards all of them are compared and evaluated. This approaches are detailed in Section 5.
- The learning system adopted was LIBLINEAR [5], which provides a probabilistic distribution before the classification. This probability distribution is exerted to rank the tags taking as the most suitable tag the one with highest probability value. The tags of the ranking will be all that appear in the categories of each training set. This entails that some positive tags of a test post might not be ranked. This issue is exposed in depth in Section 6.

4 Test and Training Data Representation

This section depicts the whole procedure followed in order to provide a user and a resource with a set of ranked tags. These recommendations are based on a learning process that learns how the users have previously tagged the resources. The core of the method is a supervised learning algorithm based on logistic regression [5].

The traditional approach splits the data into training and test sets at the beginning. Afterwards, a model is inferred using the training set and it is validated thanks to the test set [12]. In this paper, the methodology used is quite different in the sense that the training and test sets are not fixed. The test set is randomly selected and afterwards an *ad hoc* training set is provided for each test post. This paper studies different training sets built according to the resource and the user for whom the recommendations are provided.

4.1 Definition of the Test Set

According to the definition of a folksonomy in Section 3, it is composed by a set of posts. Each post is formed by a user, a resource and a set of tags, i.e.,

$$p_i = (u_i, r_i, \{t_{i_1}, \ldots, t_{i_k}\})$$

Each post of a folksonomy is candidate to become a test post. Each test post is then turned into as many examples as tags used to label the resource of this post. Therefore, post p_i is split into k test examples

$$\begin{aligned} e_1 &= (u_i, r_i, t_{i_1}) \\ &\vdots \\ e_k &= (u_i, r_i, t_{i_k}) \end{aligned} \tag{1}$$

Example 1 Let the following folksonomy be

post	*date*	*User*	*Resource*	*Tags*
p_1	d_1	u_1	r_1	t_1
p_2	d_2	u_1	r_2	t_2
p_3	d_3	u_2	r_1	t_1
p_4	d_4	u_3	r_1	t_3
p_5	d_5	u_2	r_2	t_4
p_6	d_6	u_2	r_1	t_2, t_3
p_7	d_7	u_2	r_2	t_2, t_5
p_8	d_8	u_3	r_2	t_1

(2)

Let $p_7 = (u_2, r_2, \{t_2, t_5\})$ be a randomly selected test post at instant d_7. Therefore the test set is formed by

example	*date*	*User*	*Resource*	*Tags*
e_1	d_7	u_2	r_2	t_2
e_2	d_7	u_2	r_2	t_5

(3)

4.2 Definition of the Initial Training Set

Whichever learning system strongly depends on the training set used to learn. In fact, in order to guarantee a better learning, it would be ideal for the distribution of the categories in both training and test sets to be as similar as possible. Therefore, the selection of an adequate training set is not a trivial task that must be carefully carried out.

Once the test set is randomly selected, an *ad hoc* training set is dynamically chosen from the posts posted *before* the test post.

The point of departure for building the training set is the set of posts concerning with the resource or the user for which the recommendations are demanded. Once the posts are converted into examples, those examples whose tags have been previously assigned to the resource by the user to whom the recommendations are provided are removed because it has no sense to recommend a user the tags he/she had previously used to label the resource. This section deals with the way of building the initial training set. Next section will explain in depth the way of selecting promising posts through a collaborative approach, using relevance measures for post selection and penalizing oldest posts.

Let $p_i = (u_i, r_i, \{t_{i_1}, \ldots, t_{i_k}\})$ be a test post.

Let $\mathcal{R}_{r_i}$ be the subset of posts associated to a resource r_i and

$$\mathcal{R}^t_{r_i} = \{p_i / p_i \in \mathcal{R}_{r_i} \ and \ it \ was \ posted \ before \ t\}$$

Let $\mathcal{P}_{u_i}$ be the personomy (the subset of posts posted by a user constitutes the so-called personomy) associated to a user u_i and

$$\mathcal{P}^t_{u_i} = \{p_i / p_i \in \mathcal{P}_{u_i} \ and \ it \ was \ posted \ before \ t\}$$

Therefore, the training set associated to p_i is formed by

$$UR^{d_i}_{u_i,r_i} = \{\mathcal{P}^{d}_{u_i} \cup \mathcal{R}^{d}_{r_i}\}\backslash\{p_j/p_j = (u_i, r_i, \{t_{i_1}, ..., t_{i_n}\})\}$$

Example 2 Let us show an example of each training set for the test set of Example 1.

In this case the training set is computed as follows.

$$UR^{d_7}_{u_2,r_2} =$$

$$\{\mathcal{P}^{d_7}_{u_2} \cup \mathcal{R}^{d_7}_{r_2}\}\backslash\{p_j/p_j = (u_i, r_i, \{t_{i_1}, ..., t_{i_n}\})\} =$$

$$\{\{p_3, p_5, p_6\} \cup \{p_2, p_5\}\}\backslash\{p_5\} =$$

$$\{p_2, p_3, p_6\}$$

Therefore the training set is defined as follows.

$example$	$date$	$User$	$Resource$	$Tags$
e_2	d_2	u_1	r_2	t_2
e_3	d_3	u_2	r_1	t_1
e_{6_1}	d_6	u_2	r_1	t_2
e_{6_2}	d_6	u_2	r_1	t_3

(4)

4.3 Example Representation

Now we will explain the way of transforming the post into a computable form understandable for a machine learning system. Therefore, we have to define the features which characterize the examples as well as the class of each example.

The features which characterize the examples are the tags previously used to tag the resource in the folksonomy. Hence, each example will be represented by a vector V of size M (the number of tags of the folksonomy) where $v_j \geq 1$ if and only if t_j was used to tag the resource before and 0 otherwise, where $j \in 1, \ldots, M$. The class of an example will be the tag the user has tagged the resource with at this moment.

Let us represent the training set of Example 2.

Example 3 As an illustration of how to represent a example, let us represent example e_{6_1} of Example 2. The class of e_{6_1} is t_2, which is its corresponding tag. The features are t_1 and t_3, since the resource r_1 of e_{6_1} was also tagged before by t_1 in p_1 and p_3 and by t_3 in p_4. The representation of example e_{6_1} is then $\{1, 0, 1, 0, 0\}$.

4.4 Simple Feature Selection

An additional proposal to improve the representation is also adopted, since removing redundant or non-useful features which add noise to the system is usually helpful to increase both the effectiveness and efficiency of the classifiers. The example representation based on tags as features makes possible a simple feature selection in the training set. This selection consists of keeping just those tags which represent the test set.

Obviously, this is possible just in case the information about the resource of the test post is considered for building the training set, which is the case here. This approach is based on the fact that in a linear system, as the one adopted here, the weights of the features that neither represent the test post nor contribute to obtain the ranking for this post. Therefore, they could be considered as irrelevant features beforehand. This fact can be assumed only for a particular test post. Thus, this is another advantage of building a training set particularly for each test post.

Let us consider the test post of Example 1 and the training set of Example 2.

Example 4 The features for the test post are t_2 and t_4, hence, the training set of Approach 3 in Example 2 will be reduced to be represented at most with these two tags. Originally, that training set has the following representation:

example	*date*	*resource*	*features*	*category*
e_2	d_2	r_2	$\emptyset$	t_2
e_3	d_3	r_1	t_1	t_1
e_{6_1}	d_6	r_1	t_1, t_3	t_2
e_{6_2}	d_6	r_1	t_1, t_2, t_3	t_3

(5)

In the folksonomy represented in Example 1, resource r_2 does not have any tag assigned before instant d_2, then its representation is an empty set of features. Analogously, resource r_1 has only been tagged before instant d_3 with t_1, particularly in instant d_1 by user u_1, then it is represented only by feature t_1. The instant d_6 in which the resource r_1 was tagged deserves special attention. Since this resource has been tagged before d_6 with t_1 and t_3, then both tags are included in its representation. Besides, in example e_{6_1} when the category is t_2, the tag t_3 is also added because it is a tag assigned in the same instant. In the same way, in example e_{6_2} when the category is t_3, the tag t_2 is included, since it is a tag assigned in the same instant.

Reducing such representation to the tags of the test post, the results of this new approach is

example	*date*	*resource*	*features*	*category*
e_2	d_2	r_2	$\emptyset$	t_2
e_3	d_3	r_1	$\emptyset$	t_1
e_{6_1}	d_6	r_1	$\emptyset$	t_2
e_{6_2}	d_6	r_1	t_2	t_3

(6)

5 Post Selection

This section copes with the way of selecting promising posts to be included as examples for the machine learning system. The selection of such posts is carefully carried out taking into account several issues. Let us expose the outline of the process now that will be discussed in depth later. Firstly, only posts that satisfy certain collaborative conditions will be the candidates to add to the initial training set. Secondly, every candidate is scored according to certain measure of relevance. Thirdly, such relevance is penalized depending on the time that the post was posted with regard to the test post. Finally, once a ranking of the candidates is established according to such scoring measure with the correspondent penalization, the most relevance ones will be the posts that will form the final training set. Therefore, the choice of the training set for a given test post is reduced to define the criteria the posts must satisfy to be included in the training set.

5.1 Collaborative conditions

Several approaches are proposed to introduce collaboration among users and resources. The effect over the training set is the presence of additional posts carefully selected according to the collaboration among users and/or resources. The collaborative conditions can be the following:

- Collaboration using resources
  - Take the tags in the posts (contained in the training set described in Section 4.2) that were assigned to the **resource** r_i of the test post p_i. Let be this set T_{r_i}.
  - Take the posts (contained in the training set described in Section 4.2) that contain the tags of T_{r_i}.
  - Add such posts to the training set described in Section 4.2 ($UR^{d_i}_{u_i,r_i}$). Hence, the training set is formed by the posts of $UR^{d_i}_{u_i,r_i} \cup T_{r_i}$.
- Collaboration using users
  - Take the tags in the posts (contained in the training set described in Section 4.2) that were assigned by the **user** u_i of the test post p_i. Let be this set T_{u_i}.
  - Take the posts (contained in the training set described in Section 4.2) that contain the tags T_{u_i}.
  - Add such posts to the training set described in Section 4.2 ($UR^{d_i}_{u_i,r_i}$). Hence, the training set is formed by the posts of $UR^{d_i}_{u_i,r_i} \cup T_{u_i}$.
- Collaboration using both resources and users by union
  - Take the tags in the posts (contained in the training set described in Section 4.2) that were assigned to the **resource** r_i of the test post p_i, that is the set T_{r_i}, and that were assigned by the **user** u_i of the test post p_i, that is the set T_{u_i}.
  - Take the posts (contained in the training set described in Section 4.2) that contain the tags of $T_{r_i} \cup T_{u_i}$

    - Add such posts to the training set described in Section 4.2 ($UR^{d_i}_{u_i,r_i}$). Hence, the training set is formed by the posts of $UR^{d_i}_{u_i,r_i} \cup T_{r_i} \cup T_{u_i}$.
- Collaboration using both resources and users by intersection
    - Take the tags in the posts (contained in the training set described in Section 4.2) that were assigned to the **resource** r_i of the test post p_i, that is the set T_{r_i}, and that were assigned by the **user** u_i of the test post p_i, that is the set T_{u_i}.
    - Take the posts (contained in the training set described in Section 4.2) that contain the tags of $T_{r_i} \cap T_{u_i}$
    - Add such posts to the training set described in Section 4.2 ($UR^{d_i}_{u_i,r_i}$). Hence, the training set is formed by the posts of $UR^{d_i}_{u_i,r_i} \cup (T_{r_i} \cap T_{u_i})$.

5.2 Relevance measures

Once the set of candidates is obtained, they will be scored according to several measures in order to select the most relevant ones. The following scoring measures have been applied before to perform feature selection. Here, they will be adapted to select posts, that, in fact, they are examples instead of features. All of them depend on two parameters, which will be defined before presenting them. The parameter a will be the number of tags that certain post p_j shares with the post of test p_i (in its representation through the resource of the post as described in Section 4.3) and the parameter b will be the number of tags that certain post p_j has (again in its representation through the resource of the post as described in Section 4.3), but the post of test p_i does not.

- From Information Retrieval (IR), *document frequency df* [21] and F_1 [22].
- A family of measures coming from Information Theory (IT). These measures consider the distribution of the words over the categories. One of the most widely adopted [18] is the *information gain* (IG), which takes into account either the presence of the word in a category or its absence, whereas others are the *expected cross entropy for text* (CET) or χ^2 [17]. They are all defined in terms or probabilities which, in turn, are defined from the parameters mentioned above.
- Those which quantify the importance of a feature f in a category c by means of evaluating the quality of the rule $f \to c$, assuming that it has been induced by a Machine Learning (ML) algorithm [18] (in this paper changing feature by example/post). Some of these measures are based on the percentage of successes and failures of the applications of the rules as, for instance, the *Laplace* measure (L) which slightly modifies the percentage of success and the *difference* (D). Other measures that deal with the number of examples of the category in which the feature occurs and the distribution of the examples over the categories are, for example, the *impurity level* (IL) [20]. Some other variants of the foresaid measures studying the absence of the feature in the rest of the categories have also been adopted [18], leading respectively to the L_{ir}, D_{ir} and IL_{ir} measures.

5.3 Recent posts and most relevant posts

A TRS that provides the most on-fashion folksonomy tags would be desirable. This suggest emphasizing the most recent posts more than the oldest ones. For this purpose, a penalizing function is applied to the score granted by a measure. However, some measures reaches negative values, then an increasing function that guaranties a positive value should be applied before the penalizing function in order to keep the ranking the measure gives. The option adopted will be to use the arc tangent and to apply a translation of $\frac{\pi}{2}$. Then, the penalizing functions will be of the form $\frac{1}{(1+\frac{t}{d})^e}$, where t is the time the post was posted, d is the time unit and e is a parameter that controls the penalizing degree. Therefore, if m is the score granted by a measure, the final score granted to each post will be

$$\left(\arctan(m) + \frac{\pi}{2}\right) \cdot \frac{1}{(1+\frac{t}{d})^e}$$

Once the ranking of the posts is established, it is necessary to define a cutoff for selecting the most relevant ones. Some statistics in a folksonomy show that for a given test post its training set might contain either too few posts or too many ones. Both extreme situations are detrimental for the machine learning systems. Applying a percentage of posts to select the most relevant ones avoids neither having too few posts nor to many ones. The alternative used in this paper consists of applying an heuristic able to considerably reduce the posts selected when initially there are too many of them and also able to slightly reduce the posts selected when initially there are too few of them. If n is the original number of posts, such heuristic is defined as follows

$$floor(2 \cdot n)^{0.75}$$

6 Learning to Recommend

The key point of this paper is to provide a ranked set of tags adapted to a user and a resource. Therefore, it could be beneficial to have a learning system able to rank the tags and to indicate the user which tag is the best and which one is the worst for the resource. Taking into account this fact, a preference learning system can not be applied since that kind of methods yield a ranking of the examples (posts) rather than a ranking of categories (tags)[11].

As the input data are multi-category, a system of this kind is expected to be used. However, these systems do not provide a ranking. They can be adapted to produce a partial ranking in the following way: It is possible to take the labels they return and to place them first as a whole and to place the rest of the labels also as a whole afterwards. Obviously, this approach does not establish an order among the labels they recommend but it orders all those labels it returns as a whole with regard to the labels it does not provide.

The system we need must provide a global ranking of labels. Therefore a multi-label system could be used, but again they need an adaptation to deal with ranking problems. In fact, some multi-label classification systems perform a ranking and then they obtain the multi-label classification [26]. Hence, it is possible to obtain a ranking directly from them.

Elisseeff and Weston [6] propose a multi-label system based on Support Vector Machines (SVM), which generates a ranking of categories. The drawback is that the complexity is cubic and although they perform an optimization to reduce the order to be quadratic, they admit that such complexity is too high to apply to real data sets.

Platt [19] uses SVM to obtain a probabilistic output, but just for a binary classification and not for multi-category. A priori one might think about performing as many binary classification problems as the number of tags (categories) that appear in the training set. The problem would turn them into decide if a post is tagged with certain tag or not. But this becomes unfeasible since we are talking of about hundreds of thousands of tags.

With regard to the problem of tag recommendation, Godbole and Sarawagi in [8] present an evolution of SVM based on extending the original data set with extra features containing the predictions of each binary classifier and on modifying the margin of SVMs in multi-label classification problems. The main drawback is that they perform a classification rather than a ranking.

In this framework, LIBLINEAR ([5] and [7]) is an open source library [3] which is a recent alternative able to accomplish multi-category classification through logistic regression, providing a probabilistic distribution before the classification.

This paper proposes to use this probability distribution to rank the tags, taking as most suitable tag the one with the highest probability value. In the same sense the most discordant tag will be the one with the lowest probability.

This work uses the default LIBLINEAR configuration after a slight modification of the output. The evaluation in this case takes place when a resource is presented to the user. Then, a ranking of tags (the tags of the ranking will be all which appear in the categories of the training set) is provided by the learning model.

If such resource has not been previously tagged, the ranking is generated according to a priori probability distribution. It consists of ranking the tags of the user according to the frequency this user has used them before. Therefore, no learning process is performed in this particular case.

7 Performance Evaluation

So far, no consensus about an adequate metric to evaluate a recommender has been reached [10]. Some works do not include quantitative evaluation [28] or they include it partially [16]. However, the so called LeavePostOut or LeaveTagsOut proposed in [15] and [10] sheds light on this issue. They pick up a random post for each user and they provide a set of tags for this post based on the whole folksonomy except such post. Then, they compute the precision and recall [12]

[3] available at http://www.csie.ntu.edu.tw/∼cjlin/liblinear/

as follows

$$precision(T) = \frac{1}{|D|} \sum_{(u,r)\in D} \frac{|\mathcal{T}^+(u,r) \cap \mathcal{T}(u,r)|}{|\mathcal{T}(u,r)|} \tag{7}$$

$$recall(T) = \frac{1}{|D|} \sum_{(u,r)\in D} \frac{|\mathcal{T}^+(u,r) \cap \mathcal{T}(u,r)|}{|\mathcal{T}^+(u,r)|} \tag{8}$$

where D is the test set, $\mathcal{T}^+(u,r)$ are the set of tags user u has assigned to resource r (positive tags) and $\mathcal{T}(u,r)$ are the set of tags the system has recommended to user u to assign to resource r. The F_1 measure could be computed from them as

$$F_1 = \frac{1}{|D|} \sum_{(u,r)\in D} \frac{2|\mathcal{T}^+(u,r) \cap \mathcal{T}(u,r)|}{|\mathcal{T}(u,r)| + |\mathcal{T}^+(u,r)|} \tag{9}$$

The evaluation adopted in this paper consists of computing the F_1, but just considering at most the first five tags of the ranking. Notice that such kind of evaluation quantifies the quality of a classification rather than the quality of a ranking.

8 Experiments

8.1 Data Sets

The experiments were carried out over the ECML PKDD Dicovery Challenge 2009 datasets [4]. This work studies the Task 1: Content-Based Tag Recommendations and Task 2: Graph-Based Recommendations of the 2009 Challenge. The test dataset of the former contains posts, whose user, resource or tags are not contained in the post-core at level 2 of the training data whereas the latter assures that the user, resource, and tags of each post in the test data are all contained in the training data's post-core at level 2.

The post-core at level 2 is got through cleaning dump and removing all users, tags, and resources which appear in only one post. This process is repeated until convergence and got a core in which each user, tag, and resource occurs in at least two posts.

The tags were cleaned by removing all characters which are neither numbers nor letters from tags. Afterwards,those tags which were empty after cleaning or matched one of the tags imported, public, systemimported, nn, systemunfiled were removed.

The cleaned dump contains all public bookmarks and publication posts of BibSonomy [5] until (but not including) 2009-01-01. Posts from the user dblp (a mirror of the DBLP Computer Science Bibliography) as well as all posts from users which have been flagged as spammers have been excluded.

[4] http://www.kde.cs.uni-kassel.de/ws/dc09/dataset

[5] http://www.bibsonomy.org

To make the experiments, the datasets of the Tasks 1 and 2 were split into 2 different datasets. The former is made up by bookmark posts whereas the latter by bibtex posts. These sets will be respectively called bm09 no core and bt09 no core for Task 1 and bm09 core and bt09 core for Task 2.

8.2 Discussion of results

This section deals with the experiments carried out. A binary representation of the examples was empirically chosen. Hence, the value of a feature will be 1 if this feature appears in the example and 0 otherwise. For each one, several tag sets are provided depending on the parameters described before:

- The four ways of collaboration: resource, user, union and intersection.
- The twelve measures for selecting relevant posts.
- The penalizing degree of the oldest posts. Several values were checked. Those are 0, 0.0625, 0.25 and 1.

Data	Kind of Collaboration	Measure	Penalizing degree	F_1
'bm09 no core’	Intersection	IL_{ir}	0	28.54%
'bt09 no core’	Intersection	IL_{ir}	0.0625	28.56%
'bm09 core’	Intersection	IG	0	30.90%
'bt09 core’	Intersection	IG	0.0625	37.07%

Table 1. The parameters and performance of the best settings in training data for all post collections

Table 1 shows the best setting parameters together with their F_1 computed when at most 5 tags are returned, obtained using training sets. The parameters of Table 1 were established to classify the test datasets, obtaining the results shown in Table 2.

	Data	F_1
	'bm09 no core’	7.28%
	'bt09 no core’	6.75%
	'bm09 core’	24.21%
	'bt09 core’	28.76%
Task 1	'bm09 and bt09 no core’	6.98%
Task 2	'bm09 and bt09 core’	26.25%

Table 2. The performance of test data for all post collections

The performance corresponding to Tasks 1 and 2 can be seen in the two last rows of Table 2.

Table 3 shows the effect of including collaboration, post selection and a penalization of the oldest posts.

Data	F_1 no Collaboration	F_1 no post selection	F_1 no penalization
'bm09 no core'	28.02%	27.25%	28.54%
'bt09 no core'	28.53%	27.30%	28.51%
'bm09 core'	29.32%	26.32%	30.90%
'bt09 core'	36.10%	34.74%	36.84%

Table 3. The effect of collaboration, post selection and a penalization of the oldest posts

All the experiments carried out allow to conclude that the collaboration slightly improves the performance of the recommender. Particularly, the collaboration using resources and using both resources and users by intersection offer the best results with regard to the collaboration using users and using both resources and users by union. Furthermore, collaboration by intersection grants the best results. Including measures to select promising posts improves the recommender. Although the behavior among them is quite similar, measures coming from the Information Theory field together with those based on the impurity level provide the best results. The former seem to be more adequate to the core data whereas the latter improve the results of the no core data. The effect of time differs from one collection to another. The best results are reached without taking into account the time (parameter $e = 0$) for the bookmark collections, either core or no core versions. However, it seems that penalizing oldest posts improves the performance for the bibtex collections, either core or no core versions.

9 Conclusions

This work proposes a TRS based on a novel approach which learns to rank tags from previous posts in a folksonomy using a logistic regression based system. The TRS includes several ways of collaboration among users and resources. It also includes a selection of promising posts using scoring measures and penalizing the oldest ones.

The collaboration using intersection of tags that both users assign and resources have improves the performance of the recommender with regard to other types of collaboration. Selecting posts using scoring measures makes the recommender provide best tags, although in general the behavior of all of them is quite similar. However, the Information Theory measures offers best results for the core data and the impurity level measures do it for the no core data. Finally, penalizing oldest posts improves the results for the bibtex collections, but it does not obtain satisfactory results for bookmarks collections.

References

1. B. Adrian, L. Sauermann, and T. Roth-Berghofer. Contag: A semantic tag recommendation system. In *Proceedings of I-Semantics' 07*, pages 297–304, 2007.
2. H.S. Al-Khalifa. Coolrank: A social solution for ranking bookmarked web resources. In *Innovations in Information Technology, 2007. Innovations '07. 4th International Conference on*, pages 208–212, 2007.
3. Pierpaolo Basile, Domenico Gendarmi, Filippo Lanubile, and Giovanni Semeraro. Recommending smart tags in a social bookmarking system. In *Bridging the Gep between Semantic Web and Web 2.0 (SemNet 2007)*, pages 22–29, 2007.
4. Christopher H. Brooks and Nancy Montanez. Improved annotation of the blogosphere via autotagging and hierarchical clustering. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 625–632, New York, NY, USA, 2006. ACM.
5. S.S. Keerthi C. J. Lin, R. C. Weng. Trust region newton method for logistic regression. *Journal of Machine Learning Research*, (9):627–650, 2008.
6. Andre Elisseeff and Jason Weston. A kernel method for multi-labelled classification. In *In Advances in Neural Information Processing Systems 14*, pages 681–687. MIT Press, 2001.
7. Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, August 2008.
8. Shantanu Godbole and Sunita Sarawagi. Discriminative methods for multi-labeled classification. In Honghua Dai, Ramakrishnan Srikant, and Chengqi Zhang, editors, *PAKDD*, volume 3056 of *Lecture Notes in Computer Science*, pages 22–30. Springer, 2004.
9. Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Trend detection in folksonomies. In Yannis S. Avrithis, Yiannis Kompatsiaris, Steffen Staab, and Noel E. O'Connor, editors, *Proceedings of the First International Conference on Semantics And Digital Media Technology (SAMT)*, volume 4306 of *LNCS*, pages 56–70, Heidelberg, 12 2006. Springer.
10. Robert Jäschke, Leandro Marinho, Andreas Hotho, Lars Schmidt-Thieme, and Gerd Stumme. Tag recommendations in folksonomies. In Alexander Hinneburg, editor, *Workshop Proceedings of Lernen - Wissensentdeckung - Adaptivit (LWA 2007)*, pages 13–20, sep 2007.
11. Thorsten Joachims. Optimizing search engines using clickthrough data. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142, New York, NY, USA, 2002. ACM.
12. Ioannis Katakis, Grigorios Tsoumakas, and Ioannis Vlahavas. Multilabel text classification for automated tag suggestion. In *Proceedings of ECML PKDD Discovery Challenge (RSDC08)*, pages 75–83, 2008.
13. Ralf Krestel and Ling Chen. The art of tagging: Measuring the quality of tags. pages 257–271. 2008.
14. Sigma On Kee Lee and Andy Hon Wai Chun. Automatic tag recommendation for the web 2.0 blogosphere using collaborative tagging and hybrid ann semantic structures. In *ACOS'07: Proceedings of the 6th Conference on WSEAS International Conference on Applied Computer Science*, pages 88–93, Stevens Point, Wisconsin, USA, 2007. World Scientific and Engineering Academy and Society (WSEAS).
15. Leandro Balby Marinho and Lars Schmidt-Thieme. Collaborative tag recommendations. In Springer Berlin Heidelberg, editor, *Studies in Classification, Data Analysis, and Knowledge Organization*, pages 533–540, 2008.

16. Gilad Mishne. Autotag: a collaborative approach to automated tag assignment for weblog posts. In *WWW '06: Proceedings of the 15th International Conference on World Wide Web*, pages 953–954, New York, NY, USA, 2006. ACM Press. paper presented at the poster track.
17. D. Mladenic and M. Grobelnik. Feature selection for unbalanced class distribution and naive bayes. In *Proc. 16th International Conference on Machine Learning ICML-99*, pages 258–267, Bled, SL, 1999.
18. E. Montañés, I. Díaz, J. Ranilla, E.F. Combarro, and J. Fernández. Scoring and selecting terms for text categorization. *IEEE Intelligent Systems*, 20(3):40–47, May/June 2005.
19. John C. Platt and John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 1999.
20. J. Ranilla and A. Bahamonde. Fan: Finding accurate inductions. *International Journal of Human Computer Studies*, 56(4):445–474, 2002.
21. G. Salton and M. J. McGill. *An introduction to modern information retrieval.* McGraw-Hill, 1983.
22. F. Sebastiani. Machine learning in automated text categorisation. *ACM Computing Survey*, 34(1), 2002.
23. Börkur Sigurbjörnsson and Roelof van Zwol. Flickr tag recommendation based on collective knowledge. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 327–336, New York, NY, USA, 2008. ACM.
24. Sanjay Sood, Sara Owsley, Kristian Hammond, and Larry Birnbaum. Tagassist: Automatic tag suggestion for blog posts. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM 2007)*, 2007.
25. M. Tatu, M. Srikanth, and T. D'Silva. Rsdc'08: Tag recommendations using bookmark content. In *Proceedings of ECML PKDD Discovery Challenge (RSDC08)*, pages 96–107, 2008.
26. G. Tsoumakas and I. Katakis. Multi label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3):1–13, 2007.
27. M. Vojnovic, J. Cruise, D. Gunawardena, and P. Marbach. Ranking and suggesting tags in collaborative tagging applications. Technical Report MSR-TR-2007-06, Microsoft Research, 2007.
28. Zhichen Xu, Yun Fu, Jianchang Mao, and Difu Su. Towards the semantic web: Collaborative tag suggestions. In *WWW2006: Proceedings of the Collaborative Web Tagging Workshop*, Edinburgh, Scotland, 2006.
29. Shiwan Zhao, Nan Du, Andreas Nauerz, Xiatian Zhang, Quan Yuan, and Rongyao Fu. Improved Recommendation based on Collaborative Tagging Behaviors. In *Proceedings of the International ACM Conference on Intelligent User Interfaces (IUI2008)*, Canary Islands, Spain, 2008.