Multilabel Text Classification for Automated Tag Suggestion

Ioannis Katakis, Grigorios Tsoumakas, and Ioannis Vlahavas

Department of Informatics,
Aristotle University of Thessaloniki,
54124 Thessaloniki, Greece
{katak,greg,vlahavas}@csd.auth.gr

Abstract. The increased popularity of tagging during the last few years can be mainly attributed to its embracing by most of the recently thriving user-centric content publishing and management Web 2.0 applications. However, tagging systems have some limitations that have led researchers to develop methods that assist users in the tagging process, by automatically suggesting an appropriate set of tags. We have tried to model the automated tag suggestion problem as a multilabel text classification task in order to participate in the ECML/PKDD 2008 Discovery Challenge.

1 Introduction

Tagging can be defined as the process of assigning short textual descriptions or key-words (called tags) to information objects. It is a simple approach to information organization that was regularly practiced over the last decades. Scientific publications for example, are often accompanied by a list of keywords that are either freely entered or selected from an ontology (e.g. ACM Computing Classification) by their authors.

The increased popularity of tagging during the last few years can be mainly attributed to its embracing by most of the recently thriving user-centric content publishing and management Web applications (also known as Web 2.0 applications), such as wikis, web logs (blogs), and resource sharing systems, as one of the main means for the organization of their content.

Within most of these Web 2.0 applications, tagging is characterized by an additional social dimension, as the tagging process involves multiple users attaching freely selected tags to shared content (collaborative tagging).

The simplicity and popularity of collaborative tagging as an information organization approach comes at the expense of several limitations [1]. Firstly, people choose tags based on their personal opinions, their knowledge background and their preferences. Furthermore, users may be describing the same object based on different granularity. This creates a noisy tag space and thus makes it harder to find material tagged by other users. Secondly, people may use polysemous words (a word that has many related senses) in order to tag the web resources. The lack of semantic distinction in tags can lead to inappropriate connections

between items. Another problem is that different tags, which are either synonymous or have closely related meaning increase data redundancy, leading to reduced recall of information. Last, but not least, people tend to assign a very small number of tags to an object.

All these limitations have led researchers to develop methods that assist users in the tagging process, by automatically suggesting an appropriate rich set of tags, in order to avoid the aforementioned obstacles. Related work in the field involve collaborative filtering [2], graph based [2] and text mining [3, 4] approaches. In this paper we view this problem from a different perspective, modeling it as a multilabel text classification task.

The rest of the paper is structured as follows. In the next section we provide some background information on the problem of multilabel classification. After that, we briefly describe the task of the discovery challenge that we have participated in. In Section 4 we present the datasets and comment on some of their main characteristics. In Section 5 we describe the proposed recommender that we evaluate in Section 6. Finally, Section 7 concludes this work.

2 Multilabel Classification

Traditional *single-label* classification is concerned with learning from a set of examples that are associated with a single label λ from a set of disjoint labels L, $|L| > 1$. If $|L| = 2$, then the learning task is called *binary* classification (or *filtering* in the case of textual and web data), while if $|L| > 2$, then it is called *multi-class* classification. In *multilabel* classification, the examples are associated with a set of labels $Y \subseteq L$.

Multilabel classification is a challenging research problem that emerges in several modern applications such as music categorization [5, 6], protein function classification [7–10] and semantic classification of images [11, 12]. In the past, multilabel classification has mainly engaged the attention of researchers working on text categorization [13–15], as each member of a document collection usually belongs to more than one semantic category.

Multilabel classification methods can be categorized into two different groups [16]: i) *problem transformation* methods, and ii) *algorithm adaptation* methods. The first group of methods are algorithm independent. They transform the multilabel classification task into one or more single-label classification, regression or label ranking tasks. The second group of methods extend specific learning algorithms in order to handle multilabel data directly.

The most widely-used problem transformation method, called Binary Relevance (BR Learning), considers the prediction of each label as an independent binary classification task. It learns one binary classifier $h_\lambda : X \to \{\neg\lambda, \lambda\}$ for each different label $\lambda \in L$. It transforms the original data set into $|L|$ data sets D_λ that contain all examples of the original data set, labeled as λ if the labels of the original example contained λ and as $\neg\lambda$ otherwise. It is the same solution used in order to deal with a multi-class problem using a binary classifier, commonly referred to as one-against-all or one-versus-rest.

3 Task Description

We have participated in the second task "Tag Recommendation in Social Bookmark Systems". Bibsonomy[1] is a social bookmarking and publication-sharing system. A user may store and organize Bookmarks (web pages) and BibTeX entries. The main tool provided for content management in BibSonomy is tagging. Users can freely assign tags to Bookmark or BibTeX items when they submit them to the system. This task of the competition requires the development of recommender system for BibSonomy. The recommender should efficiently propose a relevant set of tags to the user when he/she submits a new item (Bookmark or BibTeX) into the system. The organizers of the competition made available a training set including examples of users assigning tags to Bookmark and BibTeX items. A new, unseen, test set will be provided in order to evaluate candidate recommenders. The decisions of each system will be compared with the true tags and the average f-measure will be calculated.

Let D be an evaluation data set, consisting of $|D|$ examples (x_i, Y_i), $i = 1..|D|, Y_i \subseteq L$. Let h be a recommender and $Z_i = h(x_i)$ be the set of labels predicted by h for example x_i. The Precision, Recall and F-measure for the recommender h on test dataset D is calculated as follows.

$$\mathit{Precision}(h, D) = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i \cap Z_i|}{|Z_i|} \qquad \mathit{Recall}(h, D) = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i \cap Z_i|}{|Y_i|}$$

$$F(h, D) = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{2|Y_i \cap Z_i|}{|Z_i| + |Y_i|}$$

4 Data Analysis and Preprocessing

Three training files were provided for the tag recommendation task namely `tas`, `bookmark` and `bibtex`.

- `tas` file: contains the tags that a particular user has assigned to a particular item.
- `bookmark` file: contains metadata for bookmark items like the URL of the web page, a description of the web page, etc.
- `bibtex` file: contains metadata for the bibtex items like the title of the paper, the authors, etc.

In Table 1 you can see the attributes of all three training files.

[1] BibSonomy - http://www.bibsonomy.org

Table 1. Attributes of the three files

File	Attributes
tas	`user, tag, content id, content type, date`
bookmark	`content id, url hash, url, description, extended description, date`
bibtex	`content id, journal volume, chapter, edition, month, day, booktitle, howPublished, institution, organization, publisher, address, school, series, bibtexKey, url, type, description, annote, note, pages, bKey, number, crossref, misc, bibtexAbstract, simhash0, simhash1, simhash2, entrytype, title, author, editor year`

Note that in `bookmark` and `bibtex` files the same resource (i.e. web page or BibTeX entry) may appear several times, one for every user submitted the web page or BibTeX item. Different users might add different meta-data and, of course, different tags into a resource. A BibteX item is identified by its unique `simhash1` attribute and a Bookmark item by its unique `url_hash` attribute. The `content_id` field links the three tables and is unique for a <user,resource> pair.

In order to evaluate the proposed approach we have divided the available files into train and test. We have kept the 80% of the `tas` file for training and the rest for testing. The corresponding `bookmark` and `bibtex` train and test files were created based on the `tas` file using the `content_id` identifier.

Some interesting statistics that we obtained from the data and exploited in our method are presented below:

- In the initial `tas` file there are 816197 records, corresponding to single tags assigned by a specific user into a resource.
- There are 268692 posts in the `tas` file (tag-*set* assignments from a particular user to a specific resource).
- There are 176141 bookmark posts.
- 156054 unique bookmark resources (web pages) in the `bookmark` file distinguished by the `url_hash` attribute.
- There are 92544 bibtex posts.
- 71704 unique bibtex items in the `bibtex` file distinguished by the `simhash1` attribute.
- Only 18192 of the above bibtex items contained abstract descriptions.

After we split the original data into training and test files the following statistics were calculated.

- Only 8.55% of the bookmark items in the test set also exist in the training set.
- Only 9.77% of the bibtex items in the test set also exist in the training set.
- 65.69% of the bookmark users in the test set also exist in the training set.
- 21.89% of the bibtex users in the test set also exist in the training set.
- The average number of tags assigned by a user to a single bookmark item in the test set is 2.76.

– The average number of tags assigned by a user to a single bibtex item in the test set is 3.25.

5 Proposed Recommender

Recommendations are required for every S_i <user,item> pair in the `TestTasFile`. In other words, we want to predict what tags a particular user would assign to this particular item. Therefore, it is important to note that the recommendations should be personalized. Another important observation that arises from the statistics mentioned in the previous section, is that items will probably not appear in the test set but there is an important possibility that the users may reappear. Hence, the tag recommender should be able to exploit prior knowledge about the item or the user but simultaneously be able to make recommendation for unseen users and items. We tried to fulfill these requirements with our tag recommender.

Our recommender works as follows (see Figure 1). The system checks if the item (Bookmark or Bibtex) exists in the training set. If this is the case then the (N) most popular tags for the item are suggested. If the item appears for the first time then the system examines if the user has appeared before. If the user is found, then the most popular tags for the user is the output of the recommender. If neither the item nor the user have appeared before then the multilabel text classifier is called to assign a relevant set of tags.

The classifier is taking into consideration the text representation of the item. This can be the content and the title of the web page or the title and the abstract of the bibtex item. The classifier as implemented in our framework takes three parameters (see Figure 2) in order to classify an item. The first parameter and main input is the text representation of the object. For the bookmark items we obtained the `description`, `extended description` and content of the web page. For the bibtex items, we kept the `journal`, `booktitle`, `bitexAbstract` and `title` attributes. The second parameter is the maximum number of recommendations (M) that the classifier will produce. However, the third parameter (θ) will force the classifier to only recommend labels (tags) that is confident enough.

We have used the Binary Relevance (BR) classifier from the Mulan[2] package. We have selected the BR classifier basically because it is a simple classifier that scales linearly with the number of classes in a multilabel classification dataset. The base learner used with BR was a naive Bayes classifier. We have set up one classifier for the Bookmark items and one for the Bibtex Items.

In order to train the classifiers we had to convert the original data into ARFF (Weka [17]) format. However, in order to decrease the dimensionality of the problem, we kept only words with a minimum frequency $f_{w(min)}$ and tags with minimum frequency of appearence $f_{t(min)}$. Therefore, in order to produce datasets for the classifiers of reasonable sizes we have set $f^1_{w(min)} = 3000$ and

[2] Mulan - **Mu**lti **La**bel Classificatio**n**, (http://mlkd.csd.auth.gr/multilabel.html)

```
Data: The training set including a TasTrainFile, a BookTrainFile and a
      BibTrainFile.
Input: The post S_i(< user, item >) pair from the TasTestFile
Output: The prediction P = {t_1, t_2, ..., t_n} of the system, t_i ϵ T, where T is
        the set of all available tags
1  initialize N_1,N_2;
2  initialize θ_1,θ_2;
3  initialize M_1,M_2;
4  for All S_i in TestTASFile do
5      if S_i.item is Bookmark then
6          if S_i.item appears in BookmarkTrainFile then
7              P ← N_1 most popular tags for S_i.item;
8              if P = ∅ then
9                  P ← bookClassifier(S_i.item.getText(),θ_1,M_1);
10         else
11             if S_i.user appears in TasTrainFile then
12                 P ← N_1 most popular tags for S_i.user;
13                 if P = ∅ then
14                     P ← bookClassifier(S_i.item.getText(),θ_1,M_1);
15             else
16                 P ← bookClassifier(S_i.item.getText(),θ_1,M_1);
17                 if P = ∅ then
18                     P ← N_1 most popular tags in BookmarkTrainFile
19     if S_i.item is Bibtex then
20         if S_i.item appears in BibtexTrainFile then
21             P ← N_2 most popular tags for S_i.item;
22             if P = ∅ then
23                 P ← bibClassifier(S_i.item.getText(),θ_2,M_2);
24         else
25             if S_i.user appears in TasTrainFile then
26                 P ← N_2 most popular tags for S_i.user;
27                 if P = ∅ then
28                     P ← bibClassifier(S_i.item.getText(),θ_2,M_2);
29             else
30                 P ← bibClassifier(S_i.item.getText(),θ_2,M_2);
31                 if P = ∅ then
32                     P ← N_2 most popular tags in BibtexTrainFile
```

Fig. 1. Pseudocode of the proposed tag recommender

Input: S_i: item to be classified, M: number of max recommendations, θ: confidence threshold
Output: The prediction $P = \{t_1, t_2, \ldots, t_n\}$ of the system, $t_i \epsilon T$, where T is the set of all available tags
$P \leftarrow \emptyset$;
C $\leftarrow$ Classifier.getConfidences(T, S_i.item.getText());
R $\leftarrow$ rank C in descending order;
for *(*$i = 0$; $i < M$; i++*)* **do**
  if $R_i > \theta$ **then**
    $P = P \bigcup R_i$
return P

Fig. 2. Multilabel text classification at the proposed recommender

$f^1_{t(min)} = 300$ for the bookmark file and $f^2_{w(min)} = 100$ and $f^2_{t(min)} = 50$ for the bibtex file. These setting led to a bookmark arff file of 208 tags and 2150 words and a bibtex file of 159 tags and 1836. Both datasets are available on-line at: http://mlkd.csd.auth.gr/multilabel.html.

6 Evaluation

We used the f-measure as discussed in section 3 in order to evaluate the framework and tune the parameters. Although we have tried various alternative settings, we have not conducted an exhaustive study for parameter settings. Some of the results obtained are presented in Table 2.

Table 2. F-measure values obtained for various parameter settings.

Parameters			**F-measure**		
θ	**M**	**N**	**All**	**Book**	**Bib**
0.0	10	10	0.0716	0.0782	0.0633
0.0	5	5	0.0848	0.0940	0.0736
0.0	1	1	0.0700	0.0904	0.0453
0.9	10	10	0.0713	0.0752	0.066
0.9	3	3	0.0847	0.0940	0.0734
0.9	10	3	**0.0852**	**0.0942**	**0.0740**

We observe that the best overall results are achieved when $\theta = 0.9$, $M = 10$, $N = 3$ [3]. Note that this is a setting providing 3 recommendations which is close to the average number of tags assigned by the users, as observed in

[3] In order to simplify the selection of parameter values we set $\theta = \theta_1 = \theta_2$, $M = M_1 = M_2$ and $N = N_1 = N_2$.

section 4. There was a slight improvement to these results when we used the classifier to make predictions when the most popular tag set was empty (see Figure 1), for example because of the removal of some tags from the training set. A further small improvement was achieved when we used the most popular tags in bookmarks and bibtex respectively when the classifier predictions where empty. The final f-measures achieved were 0.0856, 0.0942, 0.0751 respectively.

7 Conclusions

We have tried to utilize a multilabel classification algorithm in order to build an automated tag recommender for bibsonomy. Results show that tag recommendation is indeed a challenging and interesting problem for the data mining and machine learning community. Having more time we would like to test more multilabel classification algorithms and apply multilabel feature selection.

Acknowledgements

This work was partially supported by a PENED program (EPAN M.8.3.1, No. 03Δ73), jointly funded by the European Union and the Greek Government (General Secretariat of Research and Technology/GSRT).

References

1. Marchetti, A., Tesconi, M., Ronzano, F.: Semkey: A semantic collaborative tagging system. (2007)
2. Jäschke, R., Marinho, L.B., Hotho, A., Schmidt-Thieme, L., Stumme, G.: Tag recommendations in folksonomies. In Kok, J.N., Koronacki, J., de Ma'ntaras, R.L., Matwin, S., Mladenic, D., Skowron, A., eds.: Knowledge Discovery in Databases: PKDD 2007, 11th European Conference on Principles and Practice of Knowledge Discovery in Databases. Volume 4702 of Lecture Notes in Computer Science., Berlin, Heidelberg, Springer (2007) 506–514
3. Chirita, P.A., Costache, S., Nejdl, W., Handschuh, S.: P-tag: large scale automatic generation of personalized annotation tags for the web. In: WWW '07: Proceedings of the 16th international conference on World Wide Web, New York, NY, USA, ACM (2007) 845–854
4. Sood, S., Hammond, K., Owsley, S., Birnbaum, L.: TagAssist: Automatic Tag Suggestion for Blog Posts. In: Proceedings of the International Conference on Weblogs and Social Media (ICWSM 2007). (2007)
5. Li, T., Ogihara, M.: Detecting emotion in music. In: Proceedings of the International Symposium on Music Information Retrieval, Washington D.C., USA (2003) 239–240
6. Trohidis, K., Tsoumakas, G., Kalliris, G., Vlahavas, I.: Multilabel classification of music into emotions. In: Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR). (2008)
7. Clare, A., King, R.: Knowledge discovery in multi-label phenotype data. In: Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD 2001), Freiburg, Germany (2001) 42–53

8. Diplaris, S., Tsoumakas, G., Mitkas, P., Vlahavas, I.: Protein classification with multiple algorithms. In: Proceedings of the 10th Panhellenic Conference on Informatics (PCI 2005), Volos, Greece (November 2005) 448–456
9. Roth, V., Fischer, B.: Improved functional prediction of proteins by learning kernel combinations in multilabel settings. In: Proceeding of 2006 Workshop on Probabilistic Modeling and Machine Learning in Structural and Systems Biology (PMSB 2006), Tuusula, Finland (2006)
10. Zhang, M.L., Zhou, Z.H.: Multi-label neural networks with applications to functional genomics and text categorization. IEEE Transactions on Knowledge and Data Engineering **18**(10) (2006) 1338–1351
11. Boutell, M., Luo, J., Shen, X., Brown, C.: Learning multi-label scene classification. Pattern Recognition **37**(9) (2004) 1757–1771
12. Kang, F., Jin, R., Sukthankar, R.: Correlated label propagation with application to multi-label learning. In: CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York City, NY, USA, IEEE Computer Society (2006) 1719–1726
13. Yang, Y.: An evaluation of statistical approaches to text categorization. Journal of Information Retrieval **1** (1999) 67–88
14. McCallum, A.: Multi-label text classification with a mixture model trained by em. In: Proceedings of the AAAI' 99 Workshop on Text Learning. (1999)
15. Schapire, R.E. Singer, Y.: Boostexter: a boosting-based system for text categorization. Machine Learning **39**(2/3) (2000) 135–168
16. Tsoumakas, G., Katakis, I.: Multi-label classification: An overview. International Journal of Data Warehousing and Mining **3**(3) (2007) 1–13
17. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems). Morgan Kaufmann (June 2005)