RECOMENDAÇÃO DE TAGS SOB DEMANDA

GUILHERME VALE MENEZES

RECOMENDAÇÃO DE TAGS SOB DEMANDA

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

ORIENTADOR: NIVIO ZIVIANI

Belo Horizonte

Fevereiro de 2011

GUILHERME VALE MENEZES

DEMAND DRIVEN TAG RECOMMENDATION

Dissertation presented to the Graduate Program in Computer Science of the Universidade Federal de Minas Gerais in partial fulfillment of the requirements for the degree of Master in Computer Science.

ADVISOR: NIVIO ZIVIANI

Belo Horizonte

February 2011

Menezes, Guilherme Vale

M543d Demand Driven Tag Recommendation / Guilherme Vale Menezes. — Belo Horizonte, 2011.
xviii, 45 f. ; 29cm

Dissertação (mestrado) — Universidade Federal de Minas Gerais. Departamento de Ciência da Computação.

Orientador: Nivio Ziviani.

1. Computação - Teses. 2. Recuperação da informação - Teses. 3. Web 2.0 - Teses. I. Orientador. II. Título.

CDU 519.6*73 (043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Recomendação de tags por demanda

GUILHERME VALE FERREIRA MENEZES

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROF. NIVIO ZIVIANI - Orientador
Departamento de Ciência da Computação - UFMG

PROF. ADRIANO ALONSO VELOSO - Co-orientador
Departamento de Ciência da Computação - UFMG

PROF. ALTIGRAN SOARES DA SILVA
Departamento de Ciência da Computação - UFAM

PROF. ALBERTO HENRIQUE FRADE LAENDER
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 25 de fevereiro de 2011.

Acknowledgments

At first I wish to thank my family for all the support and patience on the weekdays and weekends during the development of this work. I would not have finished it if it were not for the support of my mother, Magda, father, Tarciso, and my two brothers, Murilo and Marcelo.

I would also like to thank very much my girlfriend, Renata, for all the support she gave me during the last seven years. I apologize for the boring computer talk and thank you for enduring it.

I wish to specially thank my adviser, Prof. Nivio Ziviani, for the many things I have learned during the time we have been together: not only the solid technical support I have received in LATIN, but also the values I have absorbed and which I will keep for the rest of my life.

Also many thanks to the other researchers who collaborated with this work, such as Prof. Adriano Veloso, Prof. Edleno Silva de Moura, Prof. Marcos Gonçalves and Prof. Jussara Almeida.

Finally, I wish to thank all the friends from LATIN and LBD for the things I have learned by working with them. Alan, Prof. Alberto, Prof. Altigran, Anisio, Cristiano, Denilson, Fabiano, Marco Modesto, Marco Cristo, Rickson, Thales, Thiago, Thierson, Tinti, Tupy, Wallace and Wladmir, thank you all.

"Knowing is not enough; we must apply. Willing is not enough; we must do."
(Johann Wolfgang von Goethe)

Resumo

Rotulagem colaborativa (*collaborative tagging*) permite que usuários assinalem palavras-chave (ou *tags*) que descrevam o conteúdo de objetos, o que facilita a navegação e melhora algoritmos de busca sem o uso de categorias pré-definidas. Em sistemas de *tagging* de larga escala, sistemas de recomendação de *tags* podem ajudar usuários a assinalar rótulos a objetos e ajudar a consolidar o vocabulário entre diferentes usuários. Uma abordagem promissora para recomendação de *tags* é explorar a co-ocorrência entre elas. Nesse caso, o enorme tamanho do vocabulário de *tags* é um desafio, porque (1) a complexidade computacional pode crescer exponencialmente com o número de *tags* e (2) o peso atribuído a cada *tag* pode ficar distorcido já que diferentes *tags* operam em diferentes escalas e os seus respectivos pesos podem não ser diretamente comparáveis. Neste trabalho nós propomos um método novo de recomendação de *tags* que é baseado em demanda e que faz recomendações a partir de um conjunto inicial de *tags* previamente associado a um objeto. Ele reduz o espaço de possíveis soluções, e, portanto, sua complexidade aumenta polinomialmente com o tamanho do vocabulário de *tags*. Além disso, o peso de cada *tag* é calibrado usando uma abordagem de minimização de entropia que corrige possíveis distorções e provê recomendações mais precisas. Nós conduzimos uma avaliação sistemática de métodos propostos usando três tipos de mídia: áudio, páginas Web e vídeo. Os resultados experimentais mostram que o método proposto é rápido e melhora a qualidade da recomendação em diferentes cenários experimentais. Por exemplo, no caso de um popular site de músicas ele provê melhoras em precisão (p@5) de $6{,}4\%$ a $46{,}7\%$ (dependendo do número de *tags* dadas como entrada), melhorando métodos de recomendação de *tags* baseados em co-ocorrência recentemente propostos.

Palavras-chave: Recomendação, Classificação Multilabel, Web 2.0.

Abstract

Collaborative tagging allows users to assign arbitrary keywords (or tags) describing the content of objects, which facilitates navigation and improves searching without dependence on pre-configured categories. In large-scale tag-based systems, tag recommendation services can assist a user in the assignment of tags to objects and help consolidate the vocabulary of tags across users. A promising approach for tag recommendation is to exploit the co-occurrence of tags. However, these methods are challenged by the huge size of the tag vocabulary, either because (1) the computational complexity may increase exponentially with the number of tags or (2) the score associated with each tag may become distorted since different tags may operate in different scales and the scores are not directly comparable. In this work we propose a novel method that recommends tags on a demand-driven basis according to an initial set of tags applied to an object. It reduces the space of possible solutions, so that its complexity increases polynomially with the size of the tag vocabulary. Further, the score of each tag is calibrated using an entropy minimization approach which corrects possible distortions and provides more precise recommendations. We conducted a systematic evaluation of the proposed method using three types of media: audio, Web pages and video. The experimental results show that the proposed method is fast and boosts recommendation quality on different experimental scenarios. For instance, in the case of a popular music radio Web site it provides improvements in precision (p@5) ranging from 6.4% to 46.7% (depending on the number of tags given as input), outperforming a recently proposed co-occurrence based tag recommendation method.

Palavras-chave: Recommendation, Multi-label Classification, Web 2.0.

Contents

Chapter 1

Introduction

Social interaction has become one of the central aspects around which World Wide Web applications are built in the Web 2.0 era. End-users contribute with content created by themselves, using social applications to collaborate and interact with each other by means of blog posts, comments, forum messages, videos, audio or other media. They work collectively to generate, augment, correct or update the content of digital objects that will be consumed by their peers.

This new paradigm has brought challenges for Information Retrieval (IR) methods. The lack of a centralized publisher allowed the production of astonishing amounts of user-generated data. At the same time, low-quality content composes most of this data, since there are no quality standards in most Web 2.0 applications. Furthermore, a large fraction of the generated content is in the form of multimedia, such as audio, image or video. Current industrial IR technology cannot efficiently deal with multimedia objects directly, and they usually resort to textual meta-data that surrounds each object (e.g., description or comments). However, this textual data is often non-existent or simply too noisy to be used effectively in IR.

In this context, social tagging has emerged as a means to allow users to describe their objects by using a set of descriptive keywords (tags). These keywords may be used to help users in organizing their objects, serving as a personalized, non-hierarchical categorization system. IR methods can take advantage of tags to improve the retrieval of objects through browsing or searching. In fact, recent studies have demonstrated that tags are among the best textual features to be exploited by IR services, such as automatic classification [Figueiredo et al., 2009].

In order to incentive the use of tags, social tagging applications have developed tag recommendation services that aim to aid the user in the act of assigning keywords to objects. There are some main purposes in the implementation of such services: (1) they make the as-

signment of keywords an easier task, motivating users to contribute with a larger set of tags; (2) they provoke a convergence in the use of keywords among different users [Sen et al., 2006], reducing noise and improving the effectiveness of IR processes; (3) they remind users of more rich or specific tags, which are descriptive despite not being among the most frequently used.

This dissertation presents a new algorithm that recommends tags by exploiting tag co-occurrence patterns. Our algorithm extracts co-occurrence patterns based on the tags associated with previous objects in the collection, that is, tags that were associated to objects by system users in the past. Recommendations are performed by using these patterns. An advantage of relying only on tag co-occurrence is that the algorithm does not depend on the content of the objects, i.e., the method is indifferent to the object media type. This feature makes our method suitable to social tagging systems such as *LastFM*[1], *Flickr*[2] and *Youtube*[3], which contains audio, images and videos, respectively.

Typically, an object has several different tags associated with it, and tag recommendation systems are expected to provide users with a small set of tags (3-5) to choose from. There is an unlimited number of ways to describe an object by choosing arbitrary keywords. Our strategy to this problem is to treat each possible tag already existent in the system as a class for the object, modeling the problem of recommending tags as a *multi-label classification* problem.

This approach is challenging since the vocabulary of tags in systems such as *YouTube, Delicious*[4] and *LastFM* is on the thousands. Current automatic classifiers cannot deal well with problems with many thousands of classes. Multi-label classification systems also have to consider possible combinations among these tags, which makes this problem even harder in the tag recommendation context.

1.1 Objectives

In this dissertation we study the problem of recommending tags related to a specific object given an initial set of tags already associated with this object. Formally, an initial set of tags $\mathcal{I}_o$, which is used to describe the object o, is provided to the recommendation method. The method subsequently outputs a set of related tags $\mathcal{C}_o$ ($\mathcal{I}_o \cap \mathcal{C}_o = \emptyset$), which are regarded as appropriate for describing this object.

Our purpose in this dissertation is to design better tag recommendation algorithms by

[1] www.lastfm.com
[2] www.flickr.com
[3] www.youtube.com
[4] www.delicious.com

exploiting more elaborate tag co-occurrence association rules. We are able to generate such rules by taking an on-demand approach, that is, we generate rules on the fly according to each provided object. This is possible because we can project the search space according to the input object, significantly reducing the computation needed to extract rules.

A specific objective is to reduce the distortions created in the calculation of the score of candidate tags. This problem is common in applications that need to rank tags according to some criterion. These distortions are present specially in cases in which the number of tags in an object is small, as shown in our experiments. Another specific objective is to generate datasets that can be used in experiments with tag recommendation, which are datasets crawled from social bookmarking networks such as Delicious.

1.2 Contribution of the Dissertation

We present a Lazy Associative Tag REcommender, referred to as LATRE, which is an algorithm based on the Lazy Associative Classifier (LAC) classifier [Veloso et al., 2006] that has been developed to deal with large-scale problems with thousands of tags (or classes). LATRE exploits co-occurrence of tags by extracting association rules on a demand-driven basis. These rules are the basic components of the classification model produced by LATRE. In this case, rules have the form $\mathcal{X} \rightarrow y$, where $\mathcal{X}$ is a set of tags and y is the predicted tag. Some of the results presented in this dissertation were published in [Menezes et al., 2010].

Rule extraction is a major issue for recommendation methods based on co-occurrence [Heymann et al., 2008; Sigurbjörnsson and van Zwol, 2008], since the number of extracted rules may increase exponentially with the number of tags. LATRE, on the other hand, extracts rules from the training data on the fly, at recommendation time. The algorithm projects the search space for rules according to qualitative information present in each test object, allowing the extraction of more elaborate rules with efficiency.

In other words, LATRE projects the training data according to the tags in $\mathcal{I}_o$ and extracts rules from this projected data. This ensures that only rules that carry information about object o (i.e., a test object) are extracted from the training data, drastically bounding the number of possible rules. In fact, the computational complexity of LATRE is shown to increase polynomially with the number of tags in the vocabulary. This efficiency enables LATRE to explore portions of the rule space that could not be feasibly explored by other methods.

After a set of rules is extracted for object o, LATRE uses them to rank candidate tags that are more likely to be correctly associated with this object. Each extracted rule $\mathcal{X} \xrightarrow{\theta} y$ is interpreted as a vote given for tag y and the weight of the vote is given by θ, which is the conditional probability of object o being associated with tag y given that o contains all tags

in $\mathcal{X}$. Weighted votes for each tag are added and tags that scored higher are placed on the beginning of the ranking. Usually, there are many candidate tags and, thus, properly ranking them is also a difficult issue, since different candidate tags may operate in different scales (i.e., a popular tag may receive a large number of "weak" votes, and this tag is likely to be placed before a specific tag which received a small number of "strong" votes). In order to enforce all tags to operate in the same scale, so that they can be directly compared, we employed an entropy-minimization calibration approach to correct possible distortions in the scores of candidate tags.

Experimental results obtained from collections crawled from Delicious, LastFM and YouTube show that LATRE recommends tags with a significantly higher precision in all collections when compared to a recent co-occurrence based algorithm proposed by Sigurbjörnsson and van Zwol [2008], which we considered our baseline method. The study of the effectiveness of LATRE on three different collections corresponding to different media types (i.e., Web pages, audio and video) is also an important contribution of our work, as most of the methods in the literature are tested only with one collection and one media type. Depending on the number of tags provided as input to LATRE, it obtained gains in precision (p@5) ranging from 10.7% to 23.9% for Delicious, from 6.4% to 46.7% for LastFM, and from 16.2% to 33.1% for YouTube.

1.3 Related Work

Social tagging systems allow users to associate keywords (tags) to any object, such as a Web page, a video or a photo. These system have become popular in the Web 2.0 context, in which a large amount of data is created and there is the necessity of indexing it for later retrieval. Tag recommendation services have emerged in social tagging systems with the objective of helping users to better describe the content they created. For an introductory discussion of social tagging and tag recommendation, please see Section 2.1.

Possible sources of information for tag recommendation could be: (1) tags previously associated with objects in the collection and (2) the textual content of other features (e.g., title, description, user comments) associated with the object for which the recommendation is expected. While in case of (2) there could be more input for the sake of recommendation, problems such as the lack of standardization in content format and the presence of noisy content (e.g., non-existing words [Suchanek et al., 2008]) benefit the use of recommendation methods that exploit solely tag co-occurrence information [Garg and Weber, 2008; Sigurbjörnsson and van Zwol, 2008].

We presented the related works in order of similarity to this dissertation. The first

two presented works ([Sigurbjörnsson and van Zwol, 2008] and [Garg and Weber, 2008]) are the most related to our work in terms of the problem they treated and the methodology they adopted. The next three related works ([Heymann et al., 2008], [Krestel et al., 2009] and [Wu et al., 2009]) also explored tag co-occurrence to solve other tag expansion problems (such as tag recommendation using content and the cold-start problem). The following four related works ([Xu et al., 2006], [Weinberger et al., 2008], [Liu et al., 2009] and [Siersdorfer et al., 2009]) used tag co-occurrence in applications different from tag expansion, such as tag disambiguation. The last related works listed in this section also applied classification in the social tagging domain.

Sigurbjörnsson and van Zwol [2008] explored tag co-occurrence with the objective of recommending tags directly to users based on an initial set of tags already associated with the object. In this scenario, their method first associates a list of related tags with each input tag, and then combines the lists for all input tags in a single final ranking of related tags. The two similarity measures used to compute co-occurrence of tags were the conditional probability and the Jaccard coefficient. They combined the lists of related tags by summing up the co-occurrence measures for each occurrence of a related tag. Finally, they altered the final relatedness scores by promoting tags that are more descriptive of the content of the objects, obtaining significant improvements. The problem they studied is very similar to ours, i.e., output a ranking of tags using only community knowledge (not personal information). For this reason, we chose this method as a baseline in our experiments. The main difference between the method described in [Sigurbjörnsson and van Zwol, 2008] and our's is that we employ a filtering process to reduce the size of the training data according to an input object. This reduction causes an increased efficiency and allows LATRE to extract rules that are more elaborate. The use of calibration is also a difference that improves the performance of our method, specially for objects with a small number of tags.

Another related work is presented by Garg and Weber [2008]. They study the problem of making personal recommendations using the history of all tags a user has applied in the past. Given a set of tags as input, the authors use a Naive Bayes classifier to obtain a ranking of related tags specifically for the user. This model considers the probability that a related tag will generate the set of input tags according to the user past behavior. Additionally, they use collective knowledge to estimate the co-occurrence of tags in the whole system, using a scoring method which is based on TF-IDF. The best results were obtained by combining the personal model and the collective model. They concluded that adding personal history can improve the effectiveness of co-occurrence tag-recommendation. Our method also models the problem as classification, but we do not explore personal history and therefore the methods are not directly comparable.

Heymann et al. [2008] use association rules to expand a set of tags of an object. They

only use rules of size two, i.e., one tag in the antecedent ("if") and one in the consequent ("then") of each rule. The reason is that these simple rules are easier to compute than more complex ones, that is, rules that have more than one tag in the antecedent of consequent. They assessed the effectiveness of their tag expansion method by measuring precision and recall of a tag-based object search engine. They only used single tag queries, that is, they did not need to combine the expansion list of different input tags. The experiments have shown that the expanded set of tags can increase recall by 50% while keeping the same precision. Our work differs from [Heymann et al., 2008] in that we are able to extract rules of larger size to improve efficacy. Furthermore, we correct the final scores of candidate tags using calibration.

Krestel et al. [2009] use Latent Dirichlet Allocation (LDA) to expand the set of tags of objects annotated by only a few users (the cold start problem). They use LDA to uncover the latent topics associated with each object in a dataset obtained from Delicious. Since each object may be annotated by several users from different backgrounds, and each user may regard the object as concurrently belonging to many topics, the discovered topics represent the different semantic views of the object. After a comparison with [Heymann et al., 2008] they concluded that their method is more accurate and yields more specific recommendations, a characteristic that may be useful in some applications. Even though this method also uses tag co-occurrence, it focus on a different problem than ours.

Wu et al. [2009] model tag recommendation in Flickr as a learn-to-rank problem using RankBoost. Their problem is to recommend tags given a set of previously assigned tags and the content of the images. They use as tag similarity measures the conditional probability, the Jaccard coefficient and a tag similarity measure that explores image feature similarity in a Visual Language Model (VLM). These similarity measures are given as features to the learn to rank algorithm. They conclude that content features can help to ease the ambiguity, polysemy and synonymy problems in tag recommendation. The main difference between [Wu et al., 2009] and our method is that they use content features from pictures, combining them with tag co-occurrence information.

Tag co-occurrence has also been used in contexts different from tag expansion. For example, the tag recommendation algorithm described in [Xu et al., 2006] uses co-occurrence information to select a small set of informative tags from the tags collectively used to describe an object. They give higher values to tags that have been used together by the same user (complementary tags) and lower value to different tags that have been used by different users to describe the same object (tags that describe the same concept).

Another example is the identification of ambiguous tags using co-occurrence distributions. The method in [Weinberger et al., 2008] suggests tags that help to disambiguate the set of tags previously assigned to an object. The key observation is that very different distri-

butions of co-occurring tags arise after adding each ambiguous tag. A third example is tag ranking [Liu et al., 2009], in which the authors used a random walk process based on tag co-occurrence information to generate a ranking that is shown to improve image search, tag recommendation and group recommendation. Finally, tag translation using tag co-occurrence is described in [Siersdorfer et al., 2009]. The authors created a tag co-occurrence graph and used network similarity measures to find candidates for translation.

Classification algorithms have been used in tag recommendation in the past. Heymann et al. [2008] use an SVM classifier to predict whether a tag t is associated with an object o based on the textual content of o. Their approach does not scale to many thousands of tags since they need to build a binary classifier for each tag. In their experiments, they use only the top 100 tags of their Delicious collection. A second approach is used by Song et al. [2008b]. They group objects into clusters using a graph partitioning algorithm, and they train a Naive Bayes classifier using the generated clusters as classes. When a new object arrives, their method classifies the object into one of the clusters and use the tags of the cluster to generate a ranking of candidate tags. A third way is to consider each tag as a class and model tag recommendation as a multi-label classification problem. In this case, tags are used as both features and labels, and the classification algorithm must be able to deal with a very large number of classes. This approach is discussed in [Garg and Weber, 2008] and [Song et al., 2008a], and used in this work. While Garg and Weber [2008] use a Naive Bayes classifier and Song et al. [2008a] propose a multi-label sparse Gaussian process classification to model tag recommendation, our work is based on associative classification (see Section 2.3), which can be applied to problems with thousands of classes.

In conclusion, tag co-occurrence is a very recent and active topic of research in Information Retrieval. While many previous publications have focused in tag co-occurrence as a means to solving different problems (such as tag recommendation and the cold start problem), none of them has given focus to an on-demand approach. This dissertation will cover this aspect.

1.4 Organization of the Dissertation

The rest of the dissertation is organized as follows. In Chapter 2 we cover some basic concepts. They will improve the readers' understanding of the following chapters. In Chapter 3 we provide an in-depth description of our proposed method, whereas in Chapter 4 we describe our collections and discuss our experiments and results. Finally, in Chapter 5 we offer conclusions and possible directions for future work.

Chapter 2

Basic Concepts

In this chapter we present some basic definitions and concepts that will help the reader to better understand the context of this thesis. First, we discuss the characteristics of social tagging systems and how they can be used to improve Information Retrieval tasks. Next, we overview multi-label classification, which is used in our method. Finally, we introduce associative classification and show its application in tag recommendation.

2.1 Social Tagging

Social tagging systems allow users to annotate an object, such as a Web page, a video or a photo, with freely chosen keywords (or tags). These systems constitute a new phenomenon, specially when considered in the context of the Web 2.0, which allowed the creation of a large amount of user generated data over the past few years. Some researchers have recently studied the main characteristics of social tagging systems and the corresponding data their users generate. In this section we discuss these characteristics and give examples of how they can be used to improve Information Retrieval (IR) tasks, such as clustering, search and recommendation.

Social tagging systems are now popular due to the widespread adoption of Web 2.0 applications, specially social networks (Facebook[5], MySpace[6], Orkut[7]), online media publishing systems (YouTube, Flickr, LastFM) and social bookmarking systems (Delicious, StumbleUpon[8], Digg[9], Technorati[10]). In fact, a survey published in 2007 has shown that 28% of

[5] http://www.facebook.com/
[6] http://www.myspace.com/
[7] http://www.orkut.com/
[8] http://www.stumbleupon.com/
[9] http://www.digg.com/
[10] http://www.technorati.com/

Web users had already used tags to categorize content such as a photo, a news story or blog posts [Rainie, 2007].

In most social tagging systems, the tagging process is related to personal organization. For example, tags are commonly used in social bookmarking systems, which are systems that allow users to bookmark objects on the Web, i.e., keep them stored for later access. In this scenario, users associate a set of tags to each object they bookmark with the purpose of easily retrieving them in the future. Another example are blogs and other sites in which users generate their own content. Users in this case are publishers and associate tags to their own content in order to make it easier to be retrieved by other users or by themselves.

In such a scenario, users associate keywords that are related to their personal sense of organization, i.e., they do not need to make sense to other users in general. For example, users may associate to a blog post the tags "philosophy" and "psychology", but they also may use the tags "to read" or "i37-2010". In other words, users organize the set of objects they have already tagged in personal classification systems, which are suited specifically for their own context. They adapt their own classification of objects according to their language, region, interest area, intention and personal conventions.

This personalized use of tags is possible because of the lack of a controlled vocabulary to describe the users' objects. This form of classification contrasts to traditional classification systems, such as the Dewey Decimal Classification (DDC). The DDC attempts to classify all the human knowledge into a pre-defined hierarchy of classes (a taxonomy). This hierarchy spans from a set of ten main classes, such as "Religion" and "Science" (see Table 2.1), going down three levels. Other well-known examples of traditional taxonomies are the Linnean classification system, which is used to categorize all living organisms, and the Medical Subject Headings (MeSH), which is used to index journal articles and books in the medical area. The classification systems based on tags has the advantage of having a much lower barrier to entry, which reduces the effort of classification and permits objects to be classified by ordinary Web users.

Despite the fact that social tagging systems serve mainly to the purpose of personal organization, the collective use of tags causes the emergence of a more general classification system. The reason is that users naturally converge in the way they describe objects, regardless of their personal aspects. For example, a popular picture that depicts a cat will probably have the keyword "cat" as one of the most frequently assigned tags. This general classification system is usually referred to as a *folksonomy* [Wal, 2007].

The three main aspects that differentiates folksonomies from traditional taxonomies are (1) the lack of an hierarchical organization, (2) its unconstrained nature, i.e., a user can assign any keyword to any object, and (3) its bottom-up generation process. These characteristics make folksonomies a constantly evolving classification system, since new

Table 2.1. The Dewey Decimal Classification (DDC) main classes.

000 - Computer science, information and general works
100 - Philosophy and psychology
200 - Religion
300 - Social sciences
400 - Language
500 - Science
600 - Technology and applied science
700 - Arts and recreation
800 - Literature
900 - History, geography, and biography

concepts are naturally assimilated. Furthermore, folksonomies are cheaper to construct and maintain than a traditional taxonomy, since no specialized professional is needed. For these reasons, there have been an increasing interest in using this information to aid IR tasks.

However, the use of folksonomies in Information Retrieval has also caused the emergence of new interesting challenges to IR methods. First, many of the tags assigned to objects are noisy, since they may only make sense to the user who created them and do not serve to the community as a whole. IR methods should take advantage of only informative tags, while avoiding the destructive effect of noisy tags. objects with a small number of tags are specially prone to the negative effects of noise, since they do not contain enough information to differentiate useful tags from damaging ones. This effect was observed in our experiments (Chapter 4).

Second, these systems have a huge vocabulary of different tags, since there is a large quantity of possible sets of keywords that may describe an object. This is a challenge to current machine learning techniques. Many classification algorithms, for example, cannot deal efficiently with problems of thousands of classes. Moreover, issues such as synonymy and polysemy (i.e., different keywords with the same meaning and same keywords with a different meaning) makes the problem even more difficult from the point of view of IR algorithms. In this work we propose a multi-label classification algorithm that can extract more elaborate patterns with efficiency, effectively reducing the aforementioned problems.

Third, tags can be applied to objects with very different purposes. For example, users may describe a picture of a cat objectively as “cat”, or more subjectively as “cute”. They can also describe the picture with more specific or general keywords, such as “Siamese cat” or “animal”. Furthermore, they can describe the process of producing the photograph, using a keyword such as “close-up”. These are only some examples in an infinity of other possibilities. Recent research has characterized these application scenarios and proposed methods that better organize folksonomies. For instance, Plangprasopchok and Lerman

[2009] propose a method that generates taxonomies based on tag usage data. Another example of ontology generation from community-based semantics is [Mika, 2007]. In [Heymann and Garcia-Molina, 2006] the authors present a simple algorithm for taxonomy generation from tag data. Semantic relatedness between tags is studied in [Cattuto et al., 2008], which also characterizes the nature of the relationship between tags. Zhang et al. [2006] also study how to statistically infer global semantic information from folksonomies.

An IR task that may benefit from tag data is document clustering. For example, [Lu et al., 2009] proposes a clustering method that is based on a tripartite graph model. This kind of graph modelling is common for methods that use tag information, and essentially models the data as a graph with three different types of nodes: users, objects and tags. Their method simultaneously cluster objects, users and tags by using a modified K-Means method, which calculates the centroid in each step using information from the neighborhood of each node. In [Ramage et al., 2009] another document clustering method based on K-Means is proposed. The authors use an extended vector representation of each document, containing not only the terms in its content but also the tags associated to it. Li et al. [2008] use folksonomies to find communities of users interested in similar topics.

Search is another application that uses tags in IR. Bischoff et al. [2008] study the usefulness of tags in search by comparing tags and the terms used in search queries for the same documents. They concluded that most tags can be used for search, since the tag application usage pattern follows the searching behavior. Carman et al. [2009] extend this analysis by observing that the distribution of tags and queries are very similar, but not identical. Furthermore, queries are usually more similar to the content of Web pages than tags, but queries and tags are more similar to one another than to content. This is a clear indication that tags may be used to improve the effectiveness of search engines. Schenkel et al. [2008] propose a new document ranking method that exploits information from the tripartite graph to significantly improve the search engine effectiveness. The method expands the query with a set of similar tags before performing the search in the document index.

Another possible application for tags in IR is in object recommendation. Konstas et al. [2009] use the tripartite graph of users, objects and tags to design a new collaborative filtering algorithm that recommends objects to users. Their method uses a Random Walk approach to infer relationships between objects. In [Sen et al., 2009] another approach for object recommendation using tag data is presented. The authors use the similarity between users and tags to predict the similarity between users and objects. Finally, Shepitsen et al. [2008] propose another object recommendation algorithm that uses clusters of tags as intermediaries between users and objects.

2.2 Multi-label Classification

As observed in Section 2.1, the collective use of tags causes the emergence of a global classification system, in which each tag associated with an object is considered as a class of that object. From this point of view, the problem of predicting which tags will be applied to an object (tag recommendation) is similar to the problem of predicting which classes will be assigned to that object (multi-label classification). In this section we introduce multi-label classification and discuss the peculiarities of its use in tag recommendation.

Multi-label classification contrasts with the simpler problem of single-label classification, in which each object can be associated with only one label from a set of disjoint labels L, $|L| > 1$. In multi-label classification, examples are associated with a set of labels $\mathcal{Y} \subseteq L$. In this framework, the algorithm associates a set of labels $\mathcal{Y}_i$ to each test object $\mathcal{X}_i$. Multi-label classification methods are required in applications such as medical diagnosis, protein function classification, music categorization, semantic scene classification, and document classification [Tsoumakas et al., 2006]. For example, a band can be classified in many categories simultaneously, such as “alternative”, “rock”, “pop” and “indie”. Similarly, a photograph can be classified in “sunset” and “sea” at the same time.

A common approach to multi-label classification is to train a separate binary classifier for each class. Each classifier is used to independently associate a score to each class, and the top k classes are assigned to the test object. For example, Comité et al. [2003] use decision trees as individual classifiers. Another approach based on SVM is presented in [Elisseeff and Weston, 2005]. These methods are called *problem transformation methods*, since they transform a multi-label problem into several binary classification problems. More formally, in these approaches the original dataset D is transformed into $|L|$ datasets D_l that contain all the examples in the original dataset, each one labelled as l or $\neg l$. A single-label classifier is then trained in each D_l [Tsoumakas et al., 2006].

One of the drawbacks of the binary approach is that they do not exploit correlation among labels, that is, they consider that each label is independent of each other. This is an oversimplifying strategy in some applications. If there is ambiguity in the use of a label, the correlation among classes can help to narrow the meaning of the label. For instance, the label “rock” may refer to a music genre or to a geological formation. If it co-occurs with the label “music”, than we can discard the geology meaning from our candidate labels. A multi-label classifier that do not use correlation would classify the object both as a geological formation and music genre, disregarding any attempt of disambiguation.

Another benefit of considering correlation among labels is that the method can infer new labels that were not explicitly given in the test instance. In the previous example, the label “rock” may be strongly correlated with the label “rock and roll”. Therefore, training

objects that contain the label “rock and roll” can also be used to predict new labels to objects labeled only with “rock”.

In this dissertation, for example, we show that exploring the correlation among tags can significantly improve the quality of the prediction, specially if objects have a relatively large number of tags (see Chapter 4). The reason is that tags are noisy labels, since they can be freely assigned by users. Consequently, correlation in tag recommendation is paramount.

Another major drawback of the binary classifier approach is that it cannot deal with problems of many thousands of labels, since methods must calculate a score for each label during run-time. It means that we need to evaluate each binary classifier for each tag in the vocabulary in order to predict a set of recommended tags. If the vocabulary of tags is big, the binary classifier approach is not feasible. In our experiments, the dataset from Delicious contains about 800,000 different tags. Therefore, we need a multi-label classifier that can deal with a large vocabulary of tags.

A third drawback of the binary classifier approach is that it can not deal naturally with a dynamic set of labels. If a new label appears, a separated binary classifier must be trained for that specific label. There must be enough data to perform this training, otherwise the new binary classifier would introduce noise in the method output. Therefore, there must be a way to decide what labels are worth a classifier, and when to train it. In applications in which new labels are constantly being created, we need a solution that can incorporate these new labels transparently. This is another reason why the binary classifier approach is not suited to tag recommendation.

2.3 Associative Classification

In our dissertation scenario we need a multi-label classification algorithm that can scale, that is able to exploit label correlations, and that can deal with a dynamic set of labels. The solution we exploited was to use an associative classifier [Liu et al., 1998]. Associative classification methods first extract association rules from the training data, and then build a classifier using these rules. These algorithms extract rules of the form $\mathcal{X} \rightarrow y$, in which the antecedent is a set of features and the consequent is a class. The associative classifier uses association rule mining algorithms, such as Apriori [Agrawal and Srikant, 1994] and FPGrowth [Han et al., 2000], to find the appropriate rules in the training data. A minimum support is used to limit the set of extracted rules, that is, the algorithms only extract rules that occurred with a frequency higher than sup_{min}. Furthermore, a minimum confidence value

θ_{min} is also used to filter the rule set. For a rule $\mathcal{X} \rightarrow y$, θ is given by

$$\theta = \frac{sup(\mathcal{X} \cup y)}{sup(\mathcal{X})}$$

The confidence of a rule $\mathcal{X} \rightarrow y$ can be though as the conditional probability that the class y will occur given that $\mathcal{X}$ has occurred.

During the test phase, the algorithm checks if each rule matches the test object. In a single-label variation of the algorithm, one class must be chosen using the set of matching rules $\mathcal{M}$. There are different approaches to make this choice. For example, a greedy approach would be to choose only the top ranked rule according to some criterion, such as information gain. Another strategy is to perform a poll, in which each rule is viewed as a weighted vote. For instance, the score for class y_i in a set of rules $\mathcal{M}$ is given by:

$$s(y_i) = \sum_{\mathcal{X} \rightarrow y_i \in \mathcal{M}} w_i$$

The class with the highest score is the output. In this case, the algorithm is able to deal with conflicting rules [Veloso et al., 2007].

A variation of the algorithm is to consider not only the topmost class in the ranking of classes, but the top-k classes. In this case, we have a multi-label classification method. This is the approach adopted in this dissertation. However, we had to perform some modifications in the algorithm to adapt it to our needs. For example, in tag recommendation the domains of $\mathcal{X}$ and c are the same. Therefore, there is a set of trivial rules that must be ignored in the extraction phase. We further discuss our algorithm in Chapter 3.

Associative classification has many advantages when used in tag recommendation. As stated earlier in Section 2.2, we need a method that exploits correlation among tags, that is, that can identify the correct meaning of ambiguous tags and that can identify different tags with the same meaning. In tag recommendation these two problems are common. Associative classification naturally deal with correlation by extracting rules that have more than one label as an antecedent, i.e., $|\mathcal{X}| > 1$.

Furthermore, associative classification take advantage of a well-studied research field, which is association rule mining. Robust and efficient algorithms, such as Apriori and FP-Growth [Han et al., 2000], can be directly exploited in associative classification. These algorithms are able to deal with problems with many thousands of labels. They are specially suited to tag recommendation problems, that have a large vocabulary of tags.

A third advantage of associative classification in tag recommendation is that it can deal with a dynamic set of tags more easily than binary classifiers. If a new tag appears in the

system, it is naturally incorporated into the method in the form of association rules. If the new tag is not frequent enough, an association rule that contains it will have a low weight, and its influence on the classifier would be small. These characteristics make associative classifier a good option to be used in tag recommendation.

2.4 Summary

In this chapter we presented some basic definitions and concepts that allow to reader to better understand the following chapters. First, we introduce social tagging and tag recommendation. Next, we discuss multi-label classification and associative classification, describing their relation to tag recommendation. In Chapters 3 and 4 we present our approach to tag recommendation and some experiments we performed.

Chapter 3

Proposed Method

In this chapter we present a new demand driven tag recommendation method, which we refer to as Lazy Associative Tag REcommender (LATRE). We first formally model the problem using associative classification. New, we present the lazy aspect of the solution, i.e. the filtering of the training data according to an input object with the objective of increasing the efficiency of the rule extraction process. Later, we describe how we attribute score to candidate tags, and then we present our calibration approach, which reduces distortions in candidate tags' scores.

3.1 Multi-label Classification Model

We have essentially modeled the tag recommendation task as a multi-label classification problem. In this case, we have as input the *training data* (referred to as $\mathcal{D}$), which consists of objects of the form $d = <\mathcal{I}_d, \mathcal{Y}_d>$, where both $\mathcal{I}_d$ and $\mathcal{Y}_d$ are sets of tags, and initially, $\mathcal{I}_d$ contains all tags that are associated with object d, while $\mathcal{Y}_d$ is empty. The *test set* (referred to as $\mathcal{T}$) consists of objects of the form $t = <\mathcal{I}_t, \mathcal{Y}_t>$, where both $\mathcal{I}_t$ and $\mathcal{Y}_t$ are sets of tags associated with object t. However, while tags in $\mathcal{I}_t$ are known in advance, tags in $\mathcal{Y}_t$ are unknown, and functions learned from $\mathcal{D}$ are used to predict (or recommend) tags that are likely to be in $\mathcal{Y}_t$ based on tags in $\mathcal{I}_t$. We developed the LATRE method within this model. Recommendation functions produced by LATRE exploit the co-occurrence of tags in $\mathcal{D}$, which are represented by association rules [Agrawal et al., 1993], as defined below.

Definition 3.1 *An association rule is an implication $\mathcal{X} \overset{\theta}{\rightarrow} y$, where the antecedent $\mathcal{X}$ is a set of tags, and the consequent y is the predicted tag. The domain for $\mathcal{X}$ is denoted as $\mathcal{I}=\{\mathcal{I}_1 \cup \mathcal{I}_2 \cup \ldots \cup \mathcal{I}_m\}^*$ (i.e., $\mathcal{X} \subseteq \mathcal{I}$), where $m=|\mathcal{D}| + |\mathcal{T}|$ and the operator A^* denotes the power set of A. The domain for y is $\mathcal{Y}=\{\mathcal{Y}_1 \cup \mathcal{Y}_2 \cup \ldots \cup \mathcal{Y}_m\}$ (i.e., $y \in \mathcal{Y}$). The size of*

rule $\mathcal{X} \rightarrow y$ is given by the number of tags in the antecedent, that is $|\mathcal{X}|$. The strength of the association between $\mathcal{X}$ and y is given by θ, which is simply the conditional probability of y being in $\mathcal{Y}_o$ given that $\mathcal{X} \subseteq \mathcal{I}_o$.

We denote as $\mathcal{R}$ a rule-set composed of rules $\mathcal{X} \xrightarrow{\theta} y$. Next we present the major steps of LATRE: rule extraction, tag ranking, and calibration.

3.2 Demand-Driven Rule Extraction

The search space for rules is huge. Existing co-occurrence based recommendation methods, such as the one proposed in [Sigurbjörnsson and van Zwol, 2008], impose computational cost restrictions during rule extraction. A typical strategy to restrict the search space for rules is to prune rules that are not sufficiently frequent (i.e., minimum support). This strategy, however, leads to serious problems because the vast majority of the tags are usually not frequent enough. An alternate strategy is to extract only rules $\mathcal{X} \rightarrow y$ such that $|\mathcal{X}| \leq \alpha_{max}$, where α_{max} is a pre-specified threshold which limits the maximum size of the extracted rules. However, in actual application scenarios, methods such as Sigurbjörnsson and van Zwol's are only able to efficiently explore the search space for rules if α_{max}=1. When the value of α_{max} is increased, the number of rules extracted from $\mathcal{D}$ increases at a much faster pace (i.e., there is a combinatorial explosion).

A possible drawback of this approach (i.e., α_{max}=1) is that more complex rules (i.e., rules with $|\mathcal{X}|$>1) will be not included in $\mathcal{R}$. An assumption is that these rules may provide important information for the sake of recommendation. However, in order to test this assumption we need an efficient method that can work using arbitrary values of α_{max}. One possible solution is to extract rules on a demand-driven basis, but before discussing this solution we need to present the definition of useful association rules.

Definition 3.2 *A rule $\{\mathcal{X} \rightarrow y\} \in \mathcal{R}$ is said to be useful for object $t =<\mathcal{I}_t, \mathcal{Y}_t>$ if $\mathcal{X} \subseteq \mathcal{I}_t$. That is, rule $\{\mathcal{X} \rightarrow y\} \in \mathcal{R}$ can only be used to predict tags for object $t \in \mathcal{T}$ if all tags in $\mathcal{X}$ are included in $\mathcal{I}_t$.*

The idea behind demand-driven rule extraction is to extract only those rules that are useful for objects in $\mathcal{T}$. In this case, rule extraction is delayed until an object $t =<\mathcal{I}_t, \mathcal{Y}_t>$ is informed. Then, tags in $\mathcal{I}_t$ are used as a filter which configures $\mathcal{D}$ in a way that only rules that are useful for object t can be extracted. This filtering process produces a projected training data, $\mathcal{D}_t$, which is composed of objects of the form $d^t =< \mathcal{I}_d^t, \mathcal{Y}_d^t >$, where $\mathcal{I}_d^t$=$\{\mathcal{I}_d \cap \mathcal{I}_t\}$ and $\mathcal{Y}_d^t$=$\{\mathcal{I}_d - \mathcal{I}_d^t\}$.

We now illustrate the filtering process. In Table 3.1 there are 5 objects $\{<\mathcal{I}_{d_1}, \mathcal{Y}_{d_1}>; <\mathcal{I}_{d_2}, \mathcal{Y}_{d_2}>; <\mathcal{I}_{d_3}, \mathcal{Y}_{d_3}>; <\mathcal{I}_{d_4}, \mathcal{Y}_{d_4}>; <\mathcal{I}_{d_5}, \mathcal{Y}_{d_5}>\}$ in $\mathcal{D}$, and one object $<\mathcal{I}_{t_1}, \mathcal{Y}_{t_1}>$ in $\mathcal{T}$.

Table 3.2 shows $\mathcal{D}$ after being projected according to $\mathcal{I}_{t_1}$. In this case, $\mathcal{I}_{d_1}^{t_1}$=$\{\mathcal{I}_{t_1} \cap \mathcal{I}_{d_1}\}$=$\{$unicef$\}$, and $\mathcal{Y}_{d_1}^{t_1}$=$\{\mathcal{I}_{d_1} - \mathcal{I}_{d_1}^{t_1}\}$=$\{$children, un, united, nations$\}$. The same procedure is repeated for the remaining objects in $\mathcal{D}$, so that $\mathcal{D}^{t_1}$ is finally obtained. For an arbitrary object $t \in \mathcal{T}$, we denote as $\mathcal{R}_t$ the rule-set extracted from $\mathcal{D}_t$.

Lemma 3.1 states that all rules extracted from the filtered training set are useful for the input object (according to Definition 3.2). In other words, after the filtering we only keep the objects that help us in extracting useful patterns.

Table 3.1. Training data and test set.

		$\mathcal{I}$	$\mathcal{Y}$
	d_1	unicef children un united nations	$\emptyset$
	d_2	un climatechange summit environment	$\emptyset$
$\mathcal{D}$	d_3	climatechange islands environment	$\emptyset$
	d_4	children games education math	$\emptyset$
	d_5	education children unicef job	$\emptyset$
$\mathcal{T}$	t_1	unicef education haiti	?

Table 3.2. Projected training data for object t_1.

		$\mathcal{I}^t$	$\mathcal{Y}^t$
	$d_1^{t_1}$	unicef	children un united nations
$\mathcal{D}_{t_1}$	$d_4^{t_1}$	education	children games math
	$d_5^{t_1}$	unicef education	children job

Lemma 3.1 *All rules in $\mathcal{R}_t$ are useful for object t =$<\mathcal{I}_t, \mathcal{Y}_t>$.*

Proof Let $\mathcal{X} \rightarrow y$ be an arbitrary rule in $\mathcal{R}_t$. In this case, $\mathcal{X} \subseteq \mathcal{I}_t$. Thus, according to Definition 3.2, this rule must be useful for object t. ∎

For instance, any rule extracted from $\mathcal{D}_{t_1}$ (i.e., Table 3.2) is useful for object t_1. Examples of rules extracted from $\mathcal{D}_{t_1}$ include:

- unicef $\xrightarrow{\theta=1.00}$ children
- $\{$unicef$\wedge$education$\}$ $\xrightarrow{\theta=1.00}$ children
- education $\xrightarrow{\theta=0.50}$ math

Since $\{$unicef$\wedge$education$\} \subseteq \mathcal{I}_{t_1}$, all these rules are useful for object t_1. An example of rule that is useless for object t_1 is "climatechange $\xrightarrow{\theta=1.00}$ environment", and it is easy to

see that this rule cannot be extracted from $\mathcal{D}_{t_1}$, since tag "climatechange" is not present in $\mathcal{D}_{t_1}$.

The next theorem states that LATRE efficiently extracts rules from $\mathcal{D}$. The key intuition is that LATRE works only on tags that are known to be associated to each other, drastically narrowing down the search space for rules.

Theorem 3.1 *The complexity of LATRE increases polynomially with the number of tags in the vocabulary.*

Proof Let n be the number of tags in the vocabulary. Obviously, the number of possible association rules that can be extracted from $\mathcal{D}$ is 2^n. Also, let $t = <\mathcal{I}_t, \mathcal{Y}_t>$ be an arbitrary object in $\mathcal{T}$. Since $\mathcal{I}_t$ contains at most k tags (with $k \ll n$), any rule useful for object t can have at most k tags in its antecedent. Therefore, the number of possible rules that are useful for object t is $(n\text{-}k) \times (k + \binom{k}{2} + \ldots + \binom{k}{k}) = O(n^k)$ (since $k \ll n$), and thus, the number of useful rules increases polynomially in n. Since, according to Lemma 1, LATRE extracts only useful rules for objects in $\mathcal{T}$, then the complexity of LATRE also increases polynomially in n. ∎

An important practical aspect of LATRE is that the projection of the dataset (as shown in the examples in Tables 3.1 and 3.2) greatly reduces the size of both n and k, since we only consider candidate tags that co-occur at least once with any tag in the test object (n is reduced), while the size of the test object is small in practice (k is reduced). For instance, in Tables 3.1 and 3.2 we have $k_1 = 1$, $k_2 = 0$, $k_3 = 0$, $k_4 = 1$ and $k_5 = 2$ for the projected dataset. Therefore, the average k per object is $(1 + 0 + 0 + 1 + 2)/5 = 4/5 = 0.8$. This number is much smaller than the upper bound in the number of tags in an object, which is 5 in the example.

3.3 Tag Ranking

In order to select candidate tags that are more likely to be associated with object $t \in \mathcal{T}$, it is necessary to sort tags by combining rules in $\mathcal{R}_t$. In this case, LATRE interprets $\mathcal{R}_t$ as a poll, in which each rule $\mathcal{X} \xrightarrow{\theta} y \in \mathcal{R}_t$ is a vote given by tags in $\mathcal{X}$ for candidate tag y. Votes have different weights, depending on the strength of the association they represent (i.e., θ). The weighted votes for each tag y are summed, giving the score for tag y with regard to object t, as shown in Equation 3.1 (where y_i is the i-th candidate tag, and $\theta(\mathcal{X} \rightarrow y_i)$ is the value θ assumes for rule $\mathcal{X} \rightarrow y_i$):

$$s(t, y_i) = \sum \theta(\mathcal{X} \rightarrow y_i), \text{ where } \mathcal{X} \subseteq \mathcal{I}_t \tag{3.1}$$

Thus, for an object t, the score associated with tag y_i is obtained by summing the θ values of the rules predicting y_i in $\mathcal{R}_t$. The likelihood of t being associated with tag y_i is obtained by normalizing the scores, as expressed by the function $\hat{p}(y_i|t)$, shown in Equation 3.2:

$$\hat{p}(y_i|t) = \frac{s(t, y_i)}{\sum_{j=0}^{n} s(t, y_j)} \tag{3.2}$$

Candidate tags for object t are sorted according to Equation 3.2, and tags appearing first in the ranking are finally recommended.

3.4 Calibration

According to Equation 3.1, the score associated with a tag is impacted by two characteristics: (1) the number of votes it receives and (2) the strength of these votes. While both characteristics are intuitively important to estimate the likelihood of association between tags and objects, it may be difficult to decide which one is more important. In some cases, the scores associated with different tags cannot be directly compared, because they operate in different scales (i.e., the score associated with popular tags are likely to be higher than the scores associated with specific tags, simply because they receive a large number of votes). This means that the same value of score can be considered either high or low, depending on the tag.

An approach for this problem would be to inspect the expected likelihood $\hat{p}(y|t)$, in order to make scores associated with different tags directly comparable. The obvious problem with this approach is that the correct value for $\hat{p}(y|t)$ is not known in advance, since $t \in \mathcal{T}$, and thus we cannot verify if $y \in \mathcal{Y}_t$. An alternative is to use a validation set (denoted as $\mathcal{V}$), which is composed of objects of the form $v = <\mathcal{I}_v, \mathcal{Y}_v>$, where both $\mathcal{I}_v$ and $\mathcal{Y}_v$ are sets of tags associated with object v, and $\{\mathcal{I}_v \cap \mathcal{Y}_v\}$=$\emptyset$. That is, the validation set essentially mimics the test set, in the sense that $\mathcal{Y}_v$ is not used for the sake of producing rules, but only to find possible distortions in the value of $\hat{p}(y|v)$.

The key intuition of our approach is to contrast values of $\hat{p}(y|v)$ for which $y \notin \mathcal{Y}_v$, and values of $\hat{p}(y|v)$ for which $y \in \mathcal{Y}_v$. In an ideal case, for a given tag y, there is a value f_y such that:

- if $\hat{p}(y|v) \leq f_y$, then $y \notin \mathcal{Y}_v$
- if $\hat{p}(y|v) > f_y$, then $y \in \mathcal{Y}_v$

Once f_y is calculated, it can be used to determine whether a certain value of $\hat{p}(y|v)$ is low or high, so that the score associated with different tags can be directly compared. However, more difficult cases exist, for which it is not possible to obtain a perfect separation in the space of values for $\hat{p}(y|v)$. Thus, we propose a more general approach to calculate f_y. The basic idea is that any value for f_y induces two partitions over the space of values for $\hat{p}(y|v)$ (i.e., one partition with values that are lower than f_y, and another partition with values that are higher than f_y). Our approach is to set f_y with the value which minimizes the average entropy of these two partitions. In the following we present the basic definitions in order to detail this approach.

Definition 3.3 *Let y be an arbitrary tag, and let $v =<\mathcal{I}_v, \mathcal{Y}_v>$ be an arbitrary object in $\mathcal{V}$. In this case, let $o(y, v)$ be a binary function such that:*

$$o(y,v) = \begin{cases} 1 & \text{if } y \in \mathcal{Y}_v \\ 0 & \text{otherwise} \end{cases}$$

Definition 3.4 *Consider $\mathcal{O}(y)$ a list of pairs $<o(y,v), \hat{p}(y|v)>$, sorted in increasing order of $\hat{p}(y|v)$. That is, $\mathcal{O}(y)$=$\{\ldots, <o(y,v_i),\ \hat{p}(y|v_i)>, <o(y,v_j), \hat{p}(y|v_j)>, \ldots\}$, such that $\hat{p}(y|v_i) \leq \hat{p}(y|v_j)$. Also, consider c a candidate value for f_y. In this case, $\mathcal{O}_c(y, \leq)$ is a sub-list of $\mathcal{O}(y)$, that is, $\mathcal{O}_c(y, \leq)$=$\{\ldots, <o(y,v), \hat{p}(y|v)>, \ldots\}$, such that for all pairs in $\mathcal{O}_c(y, \leq)$, $\hat{p}(y|v) \leq c$. Similarly, $\mathcal{O}_c(y, >)$=$\{\ldots, <o(y,v),\ \hat{p}(y|v)>, \ldots\}$, such that for all pairs in $\mathcal{O}_c(y, >)$, $\hat{p}(y|v) > c$. In other words, $\mathcal{O}_c(y, \leq)$ and $\mathcal{O}_c(y, >)$ are two partitions of $\mathcal{O}(y)$ induced by c.*

Definition 3.5 *Consider $N_0(\mathcal{O}(y))$ the number of elements in $\mathcal{O}(y)$ for which $o(y,v)$=0. Similarly, consider $N_1(\mathcal{O}(y))$ the number of elements in $\mathcal{O}(y)$ for which $o(y,v)$=1.*

In our entropy-minimization calibration approach we can calculate the entropy of tag y in $\mathcal{O}(y)$ using Equation 3.3.

$$E(\mathcal{O}(y)) = -\left(\frac{\mathrm{N}_0(\mathcal{O}(y))}{|\mathcal{O}(y)|} \times \log \frac{\mathrm{N}_0(\mathcal{O}(y))}{|\mathcal{O}(y)|}\right) - \left(\frac{\mathrm{N}_1(\mathcal{O}(y))}{|\mathcal{O}(y)|} \times \log \frac{\mathrm{N}_1(\mathcal{O}(y))}{|\mathcal{O}(y)|}\right) \quad (3.3)$$

The first step is to calculate the sum of the entropies of tag y in each partition induced by c, according to Equation 3.4.

$$E(\mathcal{O}(y), c) = \frac{|\mathcal{O}_c(y, \leq)|}{|\mathcal{O}(y)|} \times E(\mathcal{O}_c(y, \leq)) + \frac{|\mathcal{O}_c(y, >)|}{|\mathcal{O}(y)|} \times E(\mathcal{O}_c(y, >)) \quad (3.4)$$

The second step is to set f_y to the value of c which minimizes $E(\mathcal{O}(y), c)$. Now, the final step is to calibrate each $\hat{p}(y|t)$ (note that $t \in \mathcal{T}$) using the corresponding f_y (which was obtained using the validation set). The intuition is that f_y separates values of $\hat{p}(y|t)$ that should be considered high (i.e., $\hat{p}(y|t) > f_y$) from those that should be considered low (i.e., $\hat{p}(y|t) \leq f_y$). Thus, a natural way to calibrate $\hat{p}(y|t)$ is to calculate how many times $\hat{p}(y|t)$ is greater than f_y. This can be easily done as shown in Equation 3.5. The values of $\hat{c}(y|t)$ are directly comparable, since the corresponding values of $\hat{p}(y|t)$ were normalized by f_y. Thus, $\hat{c}(y|t)$ is used to sort candidate tags that are more likely to be associated with object t:

$$\hat{c}(y|t) = \frac{\hat{p}(y|t)}{f_y} \tag{3.5}$$

In some cases, calibration may drastically improves recommendation performance, as we will show in the next chapter.

3.5 Summary

In this chapter we have presented our Lazy Tag REcommendation (LATRE) method. We showed how we modelled tag recommendation as an associative classification problem. Next, we discussed its lazy aspect, describing how the filtering of the training data increases the algorithm efficiency. Later, we showed how we attribute scores to candidate tags, and how we correct distortions in the scores by using calibration. In the next chapter we will present our experimental evaluation.

Chapter 4

Experiments

In this chapter we empirically analyze the recommendation performance of LATRE. We employ as the basic evaluation metrics precision at x (p@x), which measures the proportion of relevant tags in the x first positions in the tag ranking, and MRR [Garg and Weber, 2008; Sigurbjörnsson and van Zwol, 2008], which shows the capacity of a method to return relevant tags early in the tag ranking. We first present the baseline and collections employed in the evaluation, and then we discuss the recommendation performance of LATRE on these collections.

4.1 Baseline

The baseline method used for comparison is described in [Sigurbjörnsson and van Zwol, 2008]. It is a co-occurrence method which also employs association rules. It gives less weight to candidate tags that are either too popular or too rare. Furthermore, its algorithm gives more weight to candidate tags that are higher in each candidate tag list with the goal of smoothening the co-occurrence values decay. The main difference between Sigurbjörnsson and van Zwol's and LATRE is that our method employs a filtering process to reduce the size of the training data according to an input object. This reduction causes an increased efficiency and allows LATRE to extract rules that are more elaborate than the baseline.

The reason the work in [Sigurbjörnsson and van Zwol, 2008] was used as baseline is that it is the state-of-the-art in tag recommendation using only tag co-occurrence and collective knowledge. Other related methods in Section 1.3 were not considered as baselines because they use additional information, such as the user history [Garg and Weber, 2008], image features [Wu et al., 2009] and the page content [Heymann et al., 2008; Siersdorfer et al., 2009].

4.2 Collections

Differently from related work that present results restricted to a single collection [Garg and Weber, 2008; Heymann et al., 2008; Sigurbjörnsson and van Zwol, 2008], in this dissertation we experiment with several collections, namely, Delicious Web pages, LastFM artists and YouTube videos. These datasets were chosen because they come from popular Web 2.0 systems and because they represent different media types (Web pages, audio and video). This diversity will help to validate our approach in several real scenarios. The datasets can be obtained by e-mail request to the Laboratory for Treatment of Information (LATIN) at UFMG.

4.2.1 Delicious

Delicious is a popular social bookmarking application that permits users to store, share and discover bookmarks. Users can categorize their bookmarks using tags, which serve as personal indexes so that a user can retrieve its stored pages. The assignment of tags in Delicious is collaborative, i.e., the set of tags associated with a page is generated by many users in collaboration.

For the Delicious crawl we used its "Recent Bookmarks" page, which is a public timeline that shows a subset of the most recently bookmarked objects. We collected unique bookmarked page entries and extracted the set of most frequently used bookmarks for a page (i.e., its "top bookmarks"). Delicious makes available as many as 30 "top bookmarks". Therefore, this is the maximum number of tags per object in our crawled dataset.

The crawl was performed in October 2009. Table 4.1 presents some statistics for the Delicious dataset.

Table 4.1. In this table we show some statistics related to the Delicious dataset. The last three entries refer to the first, second and third quartiles of the distribution of tags per object.

Description	Value
# unique objects	560,033
# unique tags	872,502
mean # tags/object	18.27
minimum # tags/object	1
maximum # tags/object	30
# tags/object: first quartile	8
# tags/object: second quartile	17
# tags/object: third quartile	27

4.2.2 LastFM

LastFM is a Web 2.0 music website and Internet radio. It allows users to collaboratively contribute with tags to categorize and describe the characteristics of artists, such as the music genre. LastFM was crawled using a snowball approach, which collects a set of seed artists and follows links to related artists. The artists used as seeds are the ones associated with the system most popular tags.

The crawl was performed in October 2008. Table 4.2 presents some statistics for the LastFM dataset.

Table 4.2. Some statistics related to the LastFM dataset.

Description	Value
# unique objects	99,161
# unique tags	109,443
mean # tags/object	26.88
minimum # tags/object	1
maximum # tags/object	210
# tags/object: first quartile	7
# tags/object: second quartile	14
# tags/object: third quartile	31

4.2.3 YouTube

YouTube is the largest video sharing application on the Web. Users that upload videos to YouTube can provide a set of tags that describe them for indexing purposes. YouTube is different from Delicious and LastFM in that it is non-collaborative, that is, only the video uploader can provide tags.

YouTube was crawled using a snowball approach, following links between related videos. The all-time most popular videos were used as seeds. Our sample was obtained in July 2008. In Table 4.3 we show some statistics for the YouTube dataset.

4.3 Pre-Processing Steps and Setup

In order to assess the recommendation performance of the evaluated methods, we equally divided the tags associated with test object $t = <\mathcal{I}_t, \mathcal{Y}_t>$: half of the tags is included in $\mathcal{I}_t$, and the other half is included in $\mathcal{Y}_t$ and used to assess the performance. This division is made by shuffling the tags and including the first half in $\mathcal{I}_t$ and the last half in $\mathcal{Y}_t$. A similar approach has been adopted in [Garg and Weber, 2008] and [Sigurbjörnsson and van Zwol, 2008]. We

Table 4.3. Some statistics related to the YouTube dataset.

Description	Value
# unique objects	180,778
# unique tags	132,694
mean # tags/object	9.98
minimum # tags/object	1
maximum # tags/object	91
# tags/object: first quartile	6
# tags/object: second quartile	9
# tags/object: third quartile	14

applied Porter's stemming algorithm [Porter, 1980] to avoid trivial recommendations such as plurals and small variations of the same input word.

We split each collection into three subsets: the first subset is composed of objects with a large number of tags, the second subset is composed of objects with a moderate number of tags, and the third subset is composed of objects with a small number of tags. The range of tags per object was selected in a way that the corresponding subsets have approximately the same number of objects. These subsets are characterized in Table 4.4.

Then, we randomly selected 20,000 objects from each of the subsets. We divided each group of selected objects into 5 partitions of 4,000 objects each, and we used 5-fold cross validation to assess recommendation performance. We use the validation set to find the best parameters for each evaluated method.

Table 4.4. We divided each collection into three subsets according to the number of tags per object. We show the number of objects in each of these subsets and the average number of tags per object (and its standard deviation) in each subset. Note that we excluded all objects associated with a single tag, since we need at least one tag in $\mathcal{I}_t$ and one tag in $\mathcal{Y}_t$.

Collection	Range	# Objects	Avg. # Tags
Delicious	2 to 6 tags/object	188,173	3.94 ± 1.38
	7 to 12 tags/object	167,613	9.50 ± 1.73
	13 to 30 tags/object	170,708	15.92 ± 2.53
LastFM	2 to 6 tags/object	29,622	3.96 ± 1.39
	7 to 16 tags/object	30,215	10.55 ± 2.77
	17 to 152 tags/object	31,492	44.99 ± 25.30
YouTube	2 to 5 tags/object	56,721	3.63 ± 1.09
	6 to 9 tags/object	53,284	7.39 ± 1.11
	10 to 74 tags/object	59,285	13.60 ± 5.02

4.4 Results

All experiments were performed on a Linux PC with an Intel Core 2 Duo 2.20GHz and 4GBytes RAM. In the following sections we discuss the effectiveness and the computational efficiency of LATRE.

4.4.1 Precision

Tables 4.5, 4.6 and 4.7 show the results for p@x for each subset of the three collections. The precision at x was determined by counting how many tags in the top x recommendations were correct according to the $\mathcal{Y}^t$ set. We denote the number of correct tags as c. Next, we computed the proportion of correct tags in the set of x tags, obtaining a value between 0 and 1. More formally, we have the following equation (Equation 4.1).

$$\text{p@x} = \frac{\text{c}}{\text{x}} \tag{4.1}$$

If the number of recommended tags was smaller than x, we still considered a fixed x. For instance, for p@5, if only 3 tags were recommended and only the second was correct, than our precision would be $1/5 = 0.2$, i.e., 5 is fixed.

We varied x from 1 to 5, following the analysis performed in previous work [Garg and Weber, 2008; Sigurbjörnsson and van Zwol, 2008]. The reason is that we are interested in the performance of the methods on the top of the ranking (i.e., the first 5 recommendations), since in tag recommendation the user is not likely to scan a large number of tags before choosing which ones are relevant. Furthermore, it is better to recommend good tags earlier in the ranking (e.g., p@1), so that the user has to scan fewer tags. We executed three algorithms over the subsets: the baseline, LATRE without calibration (referred to as LATNC) and LATRE.

Statistical tests have shown that LATRE performs significantly better ($p < 0.05$) than the baseline in all scenarios we experimented with. LATRE has shown gains in p@5 from 6.4% in LastFM to 23.9% in Delicious if we consider only the lower ranges; considering only the middle ranges, LATRE has shown gains in p@5 from 10.7% in Delicious to 28.9% in YouTube; and considering only the upper ranges, LATRE has shown gains in p@5 from 17.2% in Delicious to 46.7% in LastFM. It is important to note that the absolute precision values shown in this paper are underestimated, since there may be additional tags that are relevant to the user and that were not used by he/she to describe the object (and thus are not in $\mathcal{Y}_t$), as discussed in [Garg and Weber, 2008].

One interesting conclusion we could draw from the experiments is that calibration has its best performance in the lower ranges, indicating that the distortions described in Section

Table 4.5. Delicious: results for p@1, p@3, p@5 and MRR. Statistically significant differences ($p < 0.05$) are shown in (1) bold face and (2) marked with an asterisk (*), representing respectively (1) cases in which LATNC and/or LATRE performs better than the baseline and (2) cases in which LATNC performs worse than the baseline. LATRE relative gains are shown in the last column.

		Baseline	LATNC	LATRE	[% gain]
2-6 tags/object	p@1	.106	.105	**.129**	**21.7**
	p@3	.063	.059*	**.077**	**22.2**
	p@5	.046	.045*	**.057**	**23.9**
	MRR	.158	.154	**.188**	**19.0**
7-12 tags/object	p@1	.307	**.328**	**.330**	**7.5**
	p@3	.196	**.214**	**.218**	**11.2**
	p@5	.150	**.160**	**.166**	**10.7**
	MRR	.417	**.426**	**.435**	**4.3**
13-30 tags/object	p@1	.506	**.544**	**.543**	**7.3**
	p@3	.362	**.414**	**.413**	**14.1**
	p@5	.285	**.335**	**.334**	**17.2**
	MRR	.627	**.659**	**.659**	**5.1**

Table 4.6. LastFM: results for p@1, p@3, p@5 and MRR.

		Baseline	LATNC	LATRE	[% gain]
2-6 tags/object	p@1	.313	.320	**.327**	**4.5**
	p@3	.174	**.180**	**.187**	**7.5**
	p@5	.125	.128	**.133**	**6.4**
	MRR	.403	.409	**.418**	**3.7**
7-16 tags/object	p@1	.539	**.574**	**.575**	**6.7**
	p@3	.366	**.410**	**.411**	**12.3**
	p@5	.279	**.314**	**.316**	**13.3**
	MRR	.646	**.670**	**.672**	**4.0**
17-152 tags/object	p@1	.400	**.564**	**.564**	**41.0**
	p@3	.328	**.476**	**.475**	**44.8**
	p@5	.289	**.425**	**.424**	**46.7**
	MRR	.560	**.695**	**.695**	**24.1**

3 have a more damaging effect in these ranges. It is specially difficult to perform well in the lower ranges since there is very little information to work with, e.g., when there are two tags associated with an object, only one tag can be used as input and only one tag can be considered to be the correct answer. Several applications that use tag co-occurrence could benefit from calibration in these cases, such as in tag expansion for the cold start problem [Krestel et al., 2009; Heymann et al., 2008] (see Section 1.3).

Table 4.7. YouTube: results for p@1, p@3, p@5 and MRR.

		Baseline	LATNC	LATRE	[% gain]
2-5 tags/object	p@1	.288	**.328**	**.350**	**21.5**
	p@3	.153	**.171**	**.184**	**20.3**
	p@5	.105	**.113**	**.122**	**16.2**
	MRR	.356	**.385**	**.411**	**15.5**
6-9 tags/object	p@1	.405	**.485**	**.491**	**21.2**
	p@3	.273	**.356**	**.365**	**33.7**
	p@5	.204	**.254**	**.263**	**28.9**
	MRR	.497	**.556**	**.567**	**14.1**
10-74 tags/object	p@1	.534	**.626**	**.627**	**17.4**
	p@3	.419	**.528**	**.530**	**26.5**
	p@5	.347	**.459**	**.462**	**33.1**
	MRR	.560	**.704**	**.706**	**10.7**

As the number of tags per object increases, the benefit of using more elaborated rules becomes clearer. The gains in precision in the middle and upper ranges are mainly due to LATNC (i.e., LATRE without calibration), and the reason is that there are more opportunities for producing complex rules in these ranges, i.e., there is more information available. Applications that use tag co-occurrence in objects with a large number of tags could benefit from these elaborated rules, such as tag expansion for index enrichment [Sigurbjörnsson and van Zwol, 2008; Heymann et al., 2008], tag ranking [Liu et al., 2009] or tag translation [Siersdorfer et al., 2009].

It is interesting to notice that in the range 17-152 of LastFM, the baseline has achieved a recommendation performance which is lower than its performance in the range 7-16 (p@1=0.40 vs. p@1=0.54). The baseline does not perform well in range 17-152 of LastFM because this subset has a high number of tags per object (see Table 4.4), and the algorithm tends to recommend tags that are too general, such as “music” and “listen”.

Furthermore, Tables 4.5, 4.6 and 4.7 show that the absolute precision values for Delicious are lower than the corresponding values in LastFM and YouTube. The reason is that Delicious has a much more diverse set of objects, since Web pages can contain or refer to any kind of data, information or media. As an example, YouTube and LastFM pages can also be stored as a bookmark in Delicious. In fact, the number of distinct tags in Delicious in the five partitions used in our experiment (20,000 objects) is higher than in YouTube and LastFM. In the lower ranges, Delicious has 13,247 unique tags, while LastFM and YouTube have 5,164 and 6,860, respectively. The same relative proportions were found in the middle and higher ranges.

4.4.2 Mean Reciprocal Rank (MRR)

Tables 4.5, 4.6 and 4.7 also show the values of MRR for all ranges. Statistical significance tests were performed ($p < 0.05$) and results that are statistically different from the baseline are shown in bold face. MRR shows the capacity of a method to return relevant tags early in the tag ranking. The reciprocal rank is the multiplicative inverse of the ranking of the first correct answer of a ranking. The mean reciprocal rank is the average of the reciprocal rank for all test instances. More formally, we have Equation 4.2, in which N is the number of test instances and $rank_i$ is the rank of the first correct answer in the ranking for test instance i:

$$MRR = \frac{1}{|N|} \sum_{i=1}^{|N|} \frac{1}{rank_i} \tag{4.2}$$

We can see that LATRE has results significantly better than the baseline for all ranges in all datasets. MRR measures if the method returns the first relevant tag early in the tag ranking. It is different from p@x since it only considers the first tag of the ranking, which means that relevant tags recommended later are not considered. It is also a complementary metric in the sense that methods that recommend more relevant tags lower in the ranking can have a lower score than methods that recommend fewer relevant tags early in the ranking. In other words, while p@x measures if a method recommends more relevant tags, MRR measures if the method recommends relevant tags earlier in the ranking.

The MRR results show that LATRE also have a better performance in top of the recommendation rankings. In a tag recommendation system the user would have to scan a smaller amount of tags before finding a relevant one. Moreover, Tables 4.5, 4.6 and 4.7 show that LATRE's gain in MRR is always smaller than its gain in precision. The reason is that LATRE is better not only considering the first relevant tag, but also considering all the relevant tags.

4.4.3 Computational Efficiency

We evaluated LATRE performance by measuring the average execution time per object. Table 4.8 shows the results for each subset. For subsets with few tags per object, the average and maximum tagging time are in the order of few milliseconds. As expected, the average tagging time increases with the ratio of tags per object. However, even for objects that are associated with many tags, the average time spent per object is never greater than 1.8 seconds. This makes LATRE specially well-suited for real-time tag recommendation [Song et al., 2008b].

The last set of experiments aims at verifying the increase in the number of extracted rules as a function of α_{max}. According to Theorem 1, the number of rules extracted by

Table 4.8. Tagging time in seconds. We also show the standard deviation of the average tagging time.

Collection	Subset	Avg. Time	Max. Time
Delicious	2-6	0.0023 ± 0.0019	0.016
	7-12	0.067 ± 0.047	0.33
	13-30	0.47 ± 0.24	1.23
LastFM	2-6	0.0062 ± 0.0055	0.039
	7-16	0.20 ± 0.15	1.18
	17-152	1.77 ± 0.34	2.44
YouTube	2-5	0.0023 ± 0.0023	0.037
	6-9	0.027 ± 0.026	0.32
	10-74	0.31 ± 0.27	1.56

Table 4.9. Number of extracted rules for each subset of Delicious.

α_{max}	2-6 tags/object		7-12 tags/object		13-30 tags/object	
	Baseline	LATRE	Baseline	LATRE	Baseline	LATRE
1	2.10^8	1.10^6	3.10^8	1.10^7	3.10^8	7.10^7
2	2.10^{12}	2.10^6	5.10^{12}	2.10^7	6.10^{12}	7.10^7
3	3.10^{16}	3.10^6	8.10^{16}	2.10^7	1.10^{17}	8.10^7

Table 4.10. Number of extracted rules for each subset of LastFM.

α_{max}	2-6 tags/object		7-16 tags/object		17-152 tags/object	
	Baseline	LATRE	Baseline	LATRE	Baseline	LATRE
1	2.10^7	3.10^6	5.10^7	2.10^7	5.10^7	5.10^7
2	1.10^{11}	3.10^6	3.10^{11}	3.10^7	4.10^{11}	6.10^7
3	5.10^{14}	3.10^6	2.10^{15}	4.10^7	3.10^{15}	6.10^7

Table 4.11. Number of extracted rules for each subset of YouTube.

α_{max}	2-5 tags/object		6-9 tags/object		10-74 tags/object	
	Baseline	LATRE	Baseline	LATRE	Baseline	LATRE
1	5.10^7	1.10^6	6.10^7	9.10^6	6.10^7	4.10^7
2	3.10^{11}	2.10^6	4.10^{11}	1.10^7	4.10^{11}	6.10^7
3	2.10^{15}	2.10^6	3.10^{15}	1.10^7	3.10^{15}	7.10^7

LATRE increases polynomially. Tables 4.9, 4.10 and 4.11 contrasts the number of rules extracted by LATRE with the number of rules that would be extracted by the baseline. We show results for Delicious, LastFM and YouTube. Clearly, the number of rules extracted by the baseline increases exponentially, while the number of rules extracted by LATRE increases at a much slower pace, usually remaining in the same order of magnitude.

The number of extracted rules for the baseline increases only with the size of the vocabulary for the range. Since the baseline must pre-compute all associations between tags, for a vocabulary of size V the number of extracted rules is $V^{\alpha_{max}+1}$. In the case of LATRE, the number of extracted rules also depends on the size of the test instance, and is computed by using a counter inside LATRE program.

Table 4.12. We calculated the average extracted rule size for Delicious, LastFM and YouTube. The average rule size is the number of tags in $\mathcal{Y}_t$, which we denote as k.

Collection	Range	# Avg. k
Delicious	2 to 6 tags/object	1.017
	7 to 12 tags/object	1.127
	13 to 30 tags/object	1.317
LastFM	2 to 6 tags/object	1.112
	7 to 16 tags/object	1.355
	17 to 152 tags/object	1.157
YouTube	2 to 5 tags/object	1.049
	6 to 9 tags/object	1.118
	10 to 74 tags/object	1.352

In Theorem 3.1 we have proved that the complexity of LATRE increases polynomially with the number of tags in the vocabulary. If the vocabulary has size n and the maximum size of a rule antecedent is k, the complexity of LATRE is $O(n^k)$. In Table 4.12 we show the average size of k that we obtained in our experiments. It is interesting to notice that the average k is never greater than 1.5, meaning that our method's complexity is almost linear in the number of possible tags to recommend.

Another interesting observation obtained from the data in Table 4.12 is that in LastFM the rule size is smaller for range 17 to 152 tags/object in relation to the previous range. After analyzing the test and training data, we realized that the objects in this range had a great amount of noise and many tags in the test objects were not found in the training dataset. For this reason, LATRE could not extract complex patterns for the specific object, and the rule size became smaller. However, it is interesting to notice that the precision did not suffer any negative influence due to this fact.

4.5 Limitations of LATRE

LATRE does not apply in situations in which the answer time needs to be very low and the recommendations cannot be cached for a long time, such as in real time systems. Furthermore, LATRE does not apply if the object has no input tags, i.e., an object has just been created and the user have not associated any tag yet. Based on our experiments, we can see that LATRE is better suited for tag expansion applications such as index enrichment, since it works better in a large amount of tags is given as input.

4.6 Summary

In this Chapter we show experimental results that aim to evaluate LATRE in terms of precision and efficiency. We used three different datasets to perform experiments, namely, Delicious, LastFM and Youtube. We compared our method with a state-of-the-art baseline that uses only collective knowledge. Our method obtained better precision and MRR scores when compared to the baseline. Furthermore, we have shown that our method is efficient and that it extracts a smaller number of rules due to its on demand approach. Next, we expose our conclusions and plans for future work.

Chapter 5

Conclusions and Future Work

In this dissertation we have introduced LATRE, a novel co-occurrence based tag recommendation method. Tag recommendation is a recent problem that has arisen in the context of social tagging systems, which allow users to associate keywords to objects in Web 2.0 systems. Tag recommendation incentive users to contribute with a bigger and richer set of tags. At the same time, it fosters convergence in the set of used tags, effectively improving the quality of IR tasks.

We presented some background information on social tagging systems, on multi-label classification methods and on associative classification systems in Chapter 2. We modeled the tag recommendation problem as a multi-label classification problem, in which each label is a tag in the system. A set of tags is given as input, and a set of related tags is given as output.

In Chapter 3 we present our method with details. First, we formalize the modeling of the problem as a multi-label classification problem. Next, we present the lazy (on-demand) aspect of our method with details. The on-demand approach reduces the search space for new rules, improving efficiency in rule extraction and allowing the use of more elaborate rules in feasible time. LATRE interprets each extracted rule as a vote for a candidate tag. After all votes are summed, tags are sorted according to their scores. Finally, LATRE calibrates the scores in order to correct possible distortions in the final ranked list of tags.

We present experimental results and discussions in Chapter 4. We used three different datasets, namely, Delicious Web pages, LastFM artists and YouTube videos. These datasets were chosen since they are from popular systems and are representative of a diversity of media types. We compared our method with a state-of-the-art baseline that also uses rules, but do not explore a lazy approach.

Our method obtained significant improvements in precision at x and MRR. The reason is that it is able to extract more elaborate rules in feasible time (the number of extracted rules

is much smaller). While our proposed calibration mechanism has its best performance in subsets with a few number of tags, the use of more elaborate rules improves precision in subsets with a larger number of tags. LATRE achieved improvements in precision (p@5) from 10.7% to 23.9% for Delicious, from 6.4% to 46.7% for LastFM, and from 16.2% to 33.1% for YouTube.

Furthermore, we measured the execution time of LATRE, concluding that it has a feasible execution time and can be used in industrial systems. The average time spent per object is never great than 1.8 second. In the vast majority of objects the execution time is smaller than 1 second. Moreover, we have shown that the number of extracted rules in LATRE grows in a much smaller rate when compared to the baseline. The baseline can only deal efficiently with rules of size 1, while LATRE can exploit rules of larger size.

As future work, we will investigate other textual features of the Web 2.0, and how these features may improve tag recommendation. Between possible features we can cite the object description, its title and comments. In specific domains we can exploit even more information, such as a Web page text and structure. For example, we can use the textual content of items to infer the similarity between tags.

Another future direction is to explore the personalized data in order to improve recommendation to specific users. In social bookmaking systems such as Delicious we have access to user information, that is, we know the sequence of objects a user has tagged in the past. We can consequently explore user information to obtain personalized tag recommendations, that is, recommendations using collective knowledge and the specific user history. For instance, a user that speaks German will most likely find more relevant tags in German.

An interesting study would be to measure the robustness of the methods by introducing noise in the folksonomy. In this way we could measure the impact of noise (such as ambiguous tags) in the effectiveness and performance of the methods. The results would be interesting to the design of new methods that are robust to very noisy environments.

Bibliography

Agrawal, R., Imielinski, T., and Swami, A. N. (1993). Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 207--216, Washington, USA.

Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules in large databases. In *Proceedings of the International Conference on Very Large Data Bases*, pages 487--499, San Francisco, USA.

Bischoff, K., Firan, C. S., Nejdl, W., and Paiu, R. (2008). Can all tags be used for search? In *Proceedings of the International Conference on Information and Knowledge Management*, pages 193--202, Napa Valley, USA.

Carman, M. J., Baillie, M., Gwadera, R., and Crestani, F. (2009). A statistical comparison of tag and query logs. In *Proceedings of the International ACM SIGIR Conference on Research & Development of Information Retrieval*, pages 123--130, Boston, USA.

Cattuto, C., Benz, D., Hotho, A., and Gerd, S. (2008). Semantic grounding of tag relatedness in social bookmarking systems. In *Proceedings of the International Conference on The Semantic Web*, pages 615--631, Karlsruhe, Germany.

Comité, F. D., Gilleron, R., and Tommasi, M. (2003). Learning multi-label alternating decision trees from texts and data. In *Machine Learning and Data Mining in Pattern Recognition*, pages 251--274.

Elisseeff, A. and Weston, J. (2005). A kernel method for multi-labelled classification. In *Proceedings of the International ACM SIGIR Conference on Research & Development of Information Retrieval*, pages 274--281, Salvador, Brazil.

Figueiredo, F., Belém, F., Pinto, H., Almeida, J., Gonçalves, M., Fernandes, D., Moura, E., and Cristo, M. (2009). Evidence of quality of textual features on the web 2.0. In *Proceedings of the International Conference on Information and Knowledge Management*, pages 909--918, Hong Kong, China.

Garg, N. and Weber, I. (2008). Personalized, interactive tag recommendation for flickr. In *Proceedings of the ACM Conference on Recommender Systems*, pages 67--74, Lausanne, Switzerland.

Han, J., Pei, J., and Yin, Y. (2000). Mining frequent patterns without candidate generation. *SIGMOD Record*, 29(2):1--12.

Heymann, P. and Garcia-Molina, H. (2006). Collaborative creation of communal hierarchical taxonomies in social tagging systems. Technical report, Stanford InfoLab, Stanford University, Stanford, USA.

Heymann, P., Ramage, D., and Garcia-Molina, H. (2008). Social tag prediction. In *Proceedings of the International ACM SIGIR Conference on Research & Development of Information Retrieval*, pages 531--538, Singapore, Singapore.

Konstas, I., Stathopoulos, V., and Jose, J. M. (2009). On social networks and collaborative recommendation. In *Proceedings of the International ACM SIGIR Conference on Research & Development of Information Retrieval*, pages 195--202, Boston, USA.

Krestel, R., Fankhauser, P., and Nejdl, W. (2009). Latent dirichlet allocation for tag recommendation. In *Proceedings of the ACM Conference on Recommender Systems*, pages 61--68, New York, USA.

Li, X., Guo, L., and Zhao, Y. E. (2008). Tag-based social interest discovery. In *Proceedings of the International Conference on World Wide Web*, pages 675--684, Beijing, China.

Liu, B., Hsu, W., and Ma, Y. (1998). Integrating classification and association rule mining. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 80--86, New York, USA.

Liu, D., Hua, X.-S., Yang, L., Wang, M., and Zhang, H.-J. (2009). Tag ranking. In *Proceedings of the International Conference on World Wide Web*, pages 351--360, Madrid, Spain.

Lu, C., Chen, X., and Park, E. K. (2009). Exploit the tripartite network of social tagging for web clustering. In *Proceedings of the International Conference on Information and Knowledge Management*, pages 1545--1548, Hong Kong, China.

Menezes, G. V., Almeida, J. M., Belém, F., Gonçalves, M. A., Lacerda, A., de Moura, E. S., Pappa, G. L., Veloso, A., and Ziviani, N. (2010). Demand-driven tag recommendation. In *Proceedings of the European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 402--417, Barcelona, Spain.

Mika, P. (2007). Ontologies are us: A unified model of social networks and semantics. *Journal of Web Semantics*, 5(1):5--15.

Plangprasopchok, A. and Lerman, K. (2009). Constructing folksonomies from user-specified relations on flickr. In *Proceedings of the International Conference on World Wide Web*, pages 781--790, Madrid, Spain.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3):130--137.

Rainie, L. (2007). Tagging play. Article available at `http://pewresearch.org/pubs/402/tagging-play/`.

Ramage, D., Heymann, P., Manning, C. D., and Garcia-Molina, H. (2009). Clustering the tagged web. In *Proceedings of the International Conference on Web Search and Data Mining*, pages 54--63, Barcelona, Spain.

Schenkel, R., Crecelius, T., Kacimi, M., Michel, S., Neumann, T., Parreira, J. X., and Weikum, G. (2008). Efficient top-k querying over social-tagging networks. In *Proceedings of the International ACM SIGIR Conference on Research & Development of Information Retrieval*, pages 523--530, Singapore, Singapore.

Sen, S., Lam, S. K., Rashid, A. M., Cosley, D., Frankowski, D., Osterhouse, J., Harper, F. M., and Riedl, J. (2006). tagging, communities, vocabulary, evolution. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, pages 181--190, New York, USA.

Sen, S., Vig, J., and Riedl, J. (2009). Tagommenders: Connecting users to items through tags. In *Proceedings of the International Conference on World Wide Web*, pages 671--680, Madrid, Spain.

Shepitsen, A., Gemmell, J., Mobasher, B., and Burke, R. (2008). Personalized recommendation in social tagging systems using hierarchical clustering. In *Proceedings of the ACM Conference on Recommender Systems*, pages 259--266, Lausanne, Switzerland.

Siersdorfer, S., Pedro, J. S., and Sanderson, M. (2009). Automatic video tagging using content redundancy. In *Proceedings of the International ACM SIGIR Conference on Research & Development of Information Retrieval*, pages 395--402, Boston, USA.

Sigurbjörnsson, B. and van Zwol, R. (2008). Flickr tag recommendation based on collective knowledge. In *Proceedings of the International Conference on World Wide Web*, pages 327--336, Beijing, China.

Song, Y., Zhang, L., and Giles, C. L. (2008a). A sparse gaussian processes classification framework for fast tag suggestions. In *Proceedings of the International Conference on Information and Knowledge Management*, pages 93--102, Napa Valley, USA.

Song, Y., Zhuang, Z., Li, H., Zhao, Q., Li, J., Lee, W.-C., and Giles, C. L. (2008b). Real-time automatic tag recommendation. In *Proceedings of the International ACM SIGIR Conference on Research & Development of Information Retrieval*, pages 515--522, Singapore, Singapore.

Suchanek, F. M., Vojnovic, M., and Gunawardena, D. (2008). Social tags: Meaning and suggestions. In *Proceedings of the International Conference on Information and Knowledge Management*, pages 223--232, Napa Valley, USA.

Tsoumakas, G., Katakis, I., and Vlahavas, I. (2006). A review of multi-label classification methods. In *Proceedings of the ADBIS Workshop on Data Mining and Knowledge Discovery*, pages 99--109, Thessaloniki, Greece.

Veloso, A., Jr., W. M., Gonçalves, M., and Zaki, M. (2007). Multi-label lazy associative classification. In *Proceedings of the European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 605--612, Warsaw, Poland.

Veloso, A., Jr., W. M., and Zaki, M. J. (2006). Lazy associative classification. In *Proceedings of the International Conference on Data Mining*, pages 645--654, Hong Kong, China.

Wal, T. V. (2007). Folksonomy coinage and definition. Article available at `http://www.vanderwal.net/folksonomy.html`.

Weinberger, K. Q., Slaney, M., and Zwol, R. V. (2008). Resolving tag ambiguity. In *Proceedings of the ACM International Conference on Multimedia*, pages 111--120, Vancouver, Canada.

Wu, L., Yang, L., Yu, N., and Hua, X.-S. (2009). Learning to tag. In *Proceedings of the International Conference on World Wide Web*, pages 361--370, Madrid, Spain.

Xu, Z., Fu, Y., Mao, J., and Su, D. (2006). Towards the semantic web: Collaborative tag suggestions. In *Proceedings of the WWW Collaborative Web Tagging Workshop*, Edinburgh, Scotland.

Zhang, L., Wu, X., and Yu, Y. (2006). Emergent semantics from folksonomies: A quantitative study. In *Journal on Data Semantics VI*, pages 168--186.