# RELEVANCE, NOVELTY, DIVERSITY AND PERSONALIZATION IN TAG RECOMMENDATION

FABIANO MUNIZ BELÉM

# RELEVANCE, NOVELTY, DIVERSITY AND PERSONALIZATION IN TAG RECOMMENDATION

Tese 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 Doutor em Ciência da Computação.

Orientador: Jussara Marques Almeida Ph.D.
Co-orientador: Marcos André Gonçalves Ph.D.

Belo Horizonte
Fevereiro de 2018

FABIANO MUNIZ BELÉM

# RELEVANCE, NOVELTY, DIVERSITY AND PERSONALIZATION IN TAG RECOMMENDATION

Thesis 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 Doctor in Computer Science.

Advisor: Jussara Marques Almeida Ph.D.
Co-advisor: Marcos André Gonçalves Ph.D.

Belo Horizonte
February 2018

Belém, Fabiano Muniz
D1234p Relevance, Novelty, Diversity and Personalization in Tag Recommendation / Fabiano Muniz Belém. — Belo Horizonte, 2018
xxi, 141 f. : il. ; 29cm

Tese (doutorado) — Universidade Federal de Minas Gerais
Orientador: Jussara Marques Almeida Ph.D.
Co-orientador: Marcos André Gonçalves Ph.D.

1. Tag Recommendation. 2. Personalization. 3. Relevance. 4. Novelty. 5. Diversity. I. Título.

CDU 519.6*82.10

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

# FOLHA DE APROVAÇÃO

Relevance, novelty, diversity and personalization in tag recommendation

## FABIANO MUNIZ BELÉM

Tese defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROFA. JUSSARA MARQUES DE ALMEIDA GONÇALVES - Orientadora
Departamento de Ciência da Computação - UFMG

PROF. MARCOS ANDRÉ GONÇALVES - Coorientador
Departamento de Ciência da Computação - UFMG

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

PROFA. GISELE LOBO PAPPA
Departamento de Ciência da Computação - UFMG

PROF. LEANDRO BALBY MARINHO
Departamento de Sistemas e Computação - UFCG

PROF. MARCO ANTÔNIO PINHEIRO DE CRISTO
Instituto de Computação - UFAM

PROF. RODRYGO LUIS TEODORO SANTOS
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 6 de março de 2018.

# Resumo

O projeto e a avaliação de métodos de recomendação de *tags* tem focado, historicamente, em maximizar a relevância das *tags* sugeridas para um dado objeto (e.g., filme, música). A relevância de uma *tag* pode ser definida sob duas perspectivas. Em uma perspectiva *centrada no objeto*, uma *tag* é relevante se ela descreve corretamente o conteúdo do objeto alvo, enquanto em uma perspectiva *personalizada* uma *tag* relevante deve não apenas descrever bem o conteúdo do objeto, mas também satisfazer os interesses do usuário alvo. Entretanto, mesmo utilizando personalização, relevância pode não ser suficiente para garantir a eficácia e utilidade das recomendações, quando considerada isoladamente. Promover novidade e diversidade em recomendação de *tags* não apenas aumenta as chances de que o usuário selecionará pelo menos algumas das *tags* recomendadas, mas também ajuda a promover informação (i.e., *tags*) complementar, cobrindo os múltiplos aspectos ou tópicos relacionados ao objeto alvo. Mesmo assim, nenhum trabalho anterior considerou aspectos de novidade e diversidade no contexto específico de recomendação de *tags*. Nesta tese, temos como objetivo propor novas soluções que considerem múltiplos aspectos relacionados ao problema de recomendação de *tags*, em particular, relevância, novidade, diversidade e personalização. Para isso, primeiramente investigamos a eficácia do uso combinado de vários atributos de qualidade de *tags*, bem como de técnicas de *learning-to-rank* (L2R) em recomendação de *tags* com o objetivo de melhorar a relevância das *tags* recomendadas. Também propomos novos atributos sintáticos e técnicas baseadas na vizinhança do objeto alvo para tratar um cenário específico de *cold start*. Em seguida, ampliamos nosso foco, estendendo nossos melhores métodos para tratar aspectos relacionados a personalização, novidade (especificidade da *tag*) e diversidade (cobertura de tópicos). Nossos métodos foram avaliados utilizando dados reais de cinco aplicações da Web 2.0, a saber, Bibsonomy, LastFM, MovieLens, YahooVideo e YouTube. Nossos resultados experimentais demonstram a eficácia de nossos novos métodos quando comparados ao estado-da-arte e confirmam a viabilidade de melhorar novidade e diversidade com impactos desprezíveis em relevância. Também verificamos que os atributos sintáticos

propostos são responsáveis por ganhos significativos (de até 17% em precisão) sobre nosso melhor método no cenário de *cold start*. Além disso, atestamos os benefícios da personalização para prover melhores descrições para o objeto alvo, que apresentou ganhos de 15% em precisão (em média) sobre o melhor método centrado no objeto.

# Abstract

The design and evaluation of tag recommendation methods have historically focused on maximizing the relevance of the suggested tags for a given object, such as a movie or a song. Tag relevance can be defined in two perspectives. In an *object-centered* perspective, a tag is relevant if it correctly describes the content of the target object, while in a *personalized* perspective, a relevant tag not only describes well the content of the target object, but also matches the interests of the target user. However, even enriched by a personalized perspective, relevance by itself may not be enough to guarantee recommendation usefulness. Promoting novelty and diversity in tag recommendation not only increases the chances that the user will select some of the recommended tags, but also promotes complementary information (i.e., tags), which helps cover multiple aspects or topics related to the target object. Yet, no prior work has tackled novelty and diversity in the specific context of tag recommendation. In this thesis, we aim at proposing novel solutions that effectively address multiple aspects related to the tag recommendation problem, notably, relevance, novelty, diversity and personalization of the suggested tags. Towards that goal, we first investigate the effectiveness of combining various tag quality attributes by means of heuristics and learning-to-rank (L2R) techniques focusing on improving the relevance of recommended tags. We also propose new syntactic attributes and nearest neighbor techniques that are suitable for a cold start scenario in tag recommendation. Then, we expand our focus extending our best methods to address personalization, novelty (tag's specificity) and diversity (topic coverage). We evaluate our strategies using real data from five Web 2.0 applications, namely, Bibsonomy, LastFM, MovieLens, YahooVideo and YouTube. Our experimental results demonstrate the effectiveness of our new methods over state-of-the-art approaches, and attest the viability to effectively increase novelty and diversity with only a slight impact (if any) on relevance. We also found that our proposed syntactic attributes are responsible for significant improvements (up to 17% in precision) over the best relevance-driven method in a cold start scenario. In addition, we assessed the benefits of personalization to provide better descriptions of the target object, with

average gains of 15% in relevance over the best object-centered approach.

# List of Figures

# List of Tables

# Contents

# Chapter 1

# Introduction

Web 2.0 applications have become a rich source of user-generated content. Each page on the Web 2.0 often comprises a main *object* (e.g., a video, image, audio or text) and several sources of data associated with it, referred to as its *features*. The *textual features* of an object are well-defined blocks of text such as title, tags, description and user comments, used to describe the object's content, facilitating the content organization and findability [Belém et al., 2011]. Among all textual features, tags have gained special importance as they offer an effective data source for information retrieval (IR) services such as search [Li et al., 2008], classification [Figueiredo et al., 2012] and item recommendation [Ifada and Nayak, 2016], and may capture user interests reasonably well [Li et al., 2008].

In this context, there is a large interest in developing strategies to recommend tags to users, providing *relevant* and *useful* tag suggestions for a target object, and indirectly improving the quality of the IR services that rely on tags as a data source. This thesis is focused on this problem and aims at proposing novel solutions that effectively address multiple aspects related to it, notably, relevance, novelty, diversity and personalization of the recommended tags.

The tag recommendation scenario we address in this thesis can be described as follows. At the time a given target object $o$ is being created or visualized by a target user $u$, generate and rank a list of candidate tags $C$, sorting it according to relevance, novelty and diversity aspects (as we will define below), and recommend the top $k$ candidates of this ranking to the pair $< u, o >$. The target object $o$ may present an initial set $I_o$ of previously assigned tags, and we are interested in recommending new tags for this object, that is, $I_o \cap C = \emptyset$. The set $I_o$ may be empty, a scenario we denote here as cold start, as defined by Martins et al. [2016].

## 1.1 Motivation

Tag recommendation methods have historically focused on maximizing the *relevance* of the recommended tags [Belém et al., 2011; Lipczak and Milios, 2011; Wu et al., 2009]. Tag relevance can be defined in two perspectives. In an *object-centered* perspective, a tag is relevant if it correctly describes the content of the target object. In a *personalized* perspective, a relevant tag not only describes well the content of the target object, but also matches the interests of the target user [Rendle and Schmidt-Thie, 2010].

Personalization is motivated by the fact that users typically have different interests, levels of expertise or vocabulary biases, and may also have different purposes when choosing tags for a target object (e.g., content organization or content description). Moreover, even users with similar purposes may perceive the object's content differently, particularly in case of multimedia objects (an effect known as the *semantic gap*). All these factors ultimately impact the user's tag choices. Thus, personalized tag recommendation aims at suggesting tags that not only are related to the object's content but also capture the user interests, profile and background, and thus might help services such as content organization. Furthermore, personalized tag recommendations may also provide, either in isolation or collectively (i.e., all personalized recommendations provided to all users who tagged an object) better and more complete descriptions of the object's content, compared to object-centered recommendations.

However, even enriched by personalization, relevance may not be enough, in isolation, to guarantee recommendation usefulness [Vargas and Castells, 2011]. For example, a list of synonyms that well describe the object's content is arguably relevant, but also redundant and less useful than a more *diversified* list covering more aspects related to the object. Indeed, the utility of a recommended item (or tag, specifically) depends on the other items in the list of recommendations [Clarke et al., 2011; Vargas and Castells, 2011], due to the possible redundancy among them. Recommending tags that bring *novel* and *diverse* information with regards to previously ranked tags may promote more complementary information, improving the coverage of the multiple aspects or topics related to the target object and, indirectly, improving results of tag-based information retrieval (IR) services.

*Diversity* is particularly important because multimedia objects on the Web 2.0 may be *multifaceted*, that is, they may be related to various aspects and topics. Take for instance the movie "Sister Act", starring Whoopi Goldberg. Its main genre is Comedy, but it also presents elements from the Action and Musical genres. Arguably, it would be appropriate to recommend tags related to all these genres for this movie. In fact,

**Table 1.1.** Example of tag recommendations for a MovieLens object.

| *Title* | *Relevance Only Recommender* | *Relevance + Diversity Recommender* | *Relevance + Diversity + Novelty Recommender* |
|---|---|---|---|
| X-Men: The Last Stand | dvd, comics, ummarti2006, super-hero, based | dvd, genetics, biology, comics, mckellen | genetics, dvd, biology, mckellen, marvel |

we observed that a large fraction of Web 2.0 objects present multiple categories. For example, 84% of the artists in our LastFM dataset and 63% of the movies in our MovieLens dataset (see Chapter 6) are associated with two or more categories (style or genre). *Novelty*, on the other hand, can increase serendipity, coverage and recall of services that use more "specific" (yet relevant) recommended tags.

Thus, in this thesis, we define novelty as the capacity of recommending *long tail* [Celma and Herrera, 2008] tags, that is, more rare tags. The idea is that a term used as tag many times tends to be a more "obvious" recommendation (if relevant at all), thus being of little use (if any) to improve the description of the target object provided by its tag set. We note that this concept is closely related to tag *specificity*, since rare words tend to be more specific (less general) [Baeza-Yates and Ribeiro-Neto, 1999; Choi, 2015]. Diversity, in turn, refers to the *exhaustivity* [Baeza-Yates and Ribeiro-Neto, 1999; Choi, 2015] of the set of recommended tags, which is defined as the coverage they provide for the topics of the target object. We note that novelty and diversity concepts vary according to the research comunity context (e.g., information retrieval and general recommendation, as we will see in more details in Chapter 3).

To further illustrate the benefits of novelty and diversity in tag recommendation, Table 1.1 shows an example of recommendations produced for a MovieLens object (i.e., movie) by three recommenders: one focused on relevance only, a second one that directly incorporates diversity and a third one that, besides diversity, also considers novelty aspects[1].

The relevance-driven recommender suggested the relevant tags "comics", "super-hero" and (though more vague) "based", possibly referring to the fact that the movie is based on the Marvel's comics X-Men. But it also suggested the general tag "dvd". Notice also that, despite being driven by relevance, the recommender suggested an apparently irrelevant tag (as far as we can tell), "ummarti2006". The second recommender, in turn, which incorporates diversity aspects, also suggested the tags "dvd" and "comics", but together with "genetics" and "biology", which may be seen as other important subjects of the movie plot (a fiction related to genetic evolution). Those two tags cover other topics related to the movie, increasing the diversity of the recom-

[1]These are real recommendations produced by some of our proposed methods and baselines, which will be presented in Chapter 5.

mendations. The tag "mckellen", also suggested by the second recommender, can also be considered relevant, as it refers to one of the main actors of the film, Ian McKellen. We also note that all recommended tags are, to some extent, relevant to the movie, which illustrates a "good side effect" of promoting diversity: ensuring that at least one relevant tag for each topic related to an object will be suggested may demote too general or noisy tags, improving the relevance of the recommendations. In fact, our experimental results corroborate this hypothesis, as we shall see in Chapter 7. Finally, the third recommender, which fully exploits all three aspects, brought one more novel and specific tag, "marvel", which represents well the creators of the movie's universe, not to mention that it is related to the "comics" topic in a more specific way. While this example illustrates that *diversity* and *novelty* are important aspects for tag recommendation, to our knowledge, no previous work has addressed aspects other than relevance and personalization in the specific context of tag recommendation.

Another issue that has been mostly neglected in tag recommendation is the cold start problem, which refers to an insufficient amount of previous information about items or users (e.g., when new items or users are introduced in the system), making it difficult to provide effective recommendations [Saveski and Mantrach, 2014; Schein et al., 2002]. As aforementioned, in the specific tag recommendation domain, cold start has been defined as the absence of an initial set of tags associated with the target object [Martins et al., 2016]. Such scenario correspond to the case of a user who has uploaded a new object to the application and filled some of its textual features, particularly title and description, and needs suggestions of relevant terms to provide as tags[2].

Many state-of-the art tag recommendation methods exploit co-occurrence patterns with the initial tag set, recommending to a target object $o$, associated with an initial set of tags $I_o$, tags that frequently co-occur with tags in $I_o$ in a training collection [Garg and Weber, 2008; Heymann et al., 2008; Krestel and Fankhauser, 2012; Menezes et al., 2010; Sigurbjörnsson and Zwol, 2008; Wu et al., 2009]. Yet, as shown by Martins et al. [2016], the effectiveness of these methods greatly suffers in a cold start scenario in which those initial tags are absent, due to the absence of such co-occurrence information.

In order to address this issue, previous work has exploited other textual features (e.g., title, description), extracting candidate tags directly from the text associated with the target object [Lipczak et al., 2009; Lipczak and Milios, 2011; Ribeiro et al., 2015], or from neighbors (similar/related objects) [Graham and Caverlee, 2008; Lin et al.,

[2]Other scenarios of cold start are also possible, for example, when the user is new in the application (user cold start) Schein et al. [2002].

2012; Martins et al., 2016]. However, these previous efforts focus only on statistical properties of the occurrence of words, such as term frequency (TF) and inverse document frequency (IDF) [Baeza-Yates and Ribeiro-Neto, 1999]. These properties in isolation may fail to identify the most relevant candidate tags, specially from the typically small and possibly low quality texts associated with Web 2.0 objects [Figueiredo et al., 2012]. Thus, it is necessary to propose alternative tag quality attributes to distinguish relevant from non-relevant candidate tags, as well as alternative sources to generate candidate tags, which is one of the topics we tackle in this thesis, as we will discuss in the following section.

In sum, our thesis hypothesis is that we can improve various aspects of the quality of recommended tags, not only relevance, but also diversity, novelty and personalization, by proposing and combining different tag quality attributes to address scenarios with and without cold start. More specifically, by automatically combining various tag quality attributes (some of them are proposed in this thesis), using learning-to-rank techniques, we can improve the relevance of the recommended tags. Novelty and diversity aspects can also be captured by tag quality attributes, and further improved by re-ranking strategies. Finally, by designing suitable attributes to deal with the cold start and personalization issues, it is also possible to improve the effectiveness of the recommendations.

## 1.2 Objectives

Our main goal in this thesis is to propose new tag recommendation strategies that tackle all four aspects of the problem, namely: relevance, diversity, novelty and personalization. In order to improve tag recommendation effectiveness, we explore improvements in each of these aspects individually and conjointly. This is not an easy task as some of these aspects may be contradictory. For example, focusing too much on relevance may generate redundant tag recommendations that cover only some of the topics of an object. Random recommendations tend to be highly novel for a user, but they are probably very irrelevant and impersonal.

This general objective can be narrowed down into four specific goals, driven by the following research questions:

- *Research Question 1 (RQ1): How can we improve the relevance of the recommended tags by means of a combination of tag quality attributes?*

  We note that most existing tag recommendation strategies treat the problem as a multiple candidate tag ranking problem, sorting candidate tags according to some

attributes of relevance and recommending tags that are in the top positions of the generated ranking [Belém et al., 2011; Lipczak and Milios, 2011; Wu et al., 2009]. This modeling of the problem motivates the use of Learning-to-Rank (L2R) based strategies to automatically learn good tag ranking functions.

Thus, in order to improve relevance, we have worked in two fronts: (1) tag attribute engineering, that is, the design of new tag quality attributes to distinguish relevant from non-relevant candidate tags and (2) the automatic combination of these tag quality attributes by means of learning-to-rank (L2R) techniques. Some of these attributes and L2R techniques have already been proposed and evaluated in our previous work [Belém et al., 2011]. Other attributes, such as the topic coverage of a tag, which captures not only relevance, but also diversity (addressed in *RQ4*), are novel contributions of this thesis. Regarding the L2R techniques, only RankSVM [Cao et al., 2009], GP [Belém et al., 2011] and RankBoost [Wu et al., 2009] were previously applied for tag recommendation tasks. In this thesis, we evaluate other five techniques, namely, Random Forests (RF), Multiple Additive Regression Trees (MART), $\lambda$-MART, ListNet and AdaRank, which have demonstrated to be effective in other contexts [Faria et al., 2010; Gomes et al., 2013; Mohan et al., 2011].

- *Research Question 2 (RQ2): How can we generate and rank candidate tags in a cold start scenario in which there are no previously available tags?*

  Our hypothesis is that new tag quality attributes, particularly attributes that exploit the syntactic structure of the associated text, can better distinguish relevant from non relevant candidate tags, improving tag recommendation effectiveness in this scenario. Moreover, new sources of candidate tags deserve special attention in this scenario. Keeping the focus on the relevance aspect of the problem to tackle cold start in tag recommendation, we analyze new tag quality attributes, as well as alternative sources to generate candidate tags from the neighborhood of the target object (i.e., similar objects).

- *Research Question 3 (RQ3): How can we extend the proposed methods to provide personalized recommendations?*

  As we mentioned above, personalization may better satisfy the user interests, profile and background. Moreover, it may also provide better and more complete descriptions of the object's content, compared to object-centered recommendations. Towards answering *RQ3*, we propose new methods, particularly extending the best strategies developed to tackle *RQ1* to address personalization. We also

provide a quantitative assessment of the benefits of personalization when applied to describe the content of Web 2.0 objects.

- *Research Question 4 (RQ4): How can we improve novelty and diversity of tag recommendation, while keeping the same levels of relevance?*

  In order to answer this question, we propose new, complementary tag recommendation strategies to address novelty and diversity, exploiting the inherent tradeoffs that exist among relevance, novelty and diversity. Particularly, we extend the best strategies found in *RQ1* to include new attributes as well as new objective functions that capture novelty and diversity.

## 1.3 Contributions

Towards achieving the proposed goals, we have accomplished the following contributions:

1. Proposal of new tag relevance attributes, grouped into two categories: (1) syntactic attributes, which exploit patterns of the structure of the text associated with the target object, and (2) topic-based attributes, which capture the coverage of the topics (e.g., categories) associated with the target object, thus also being related to diversity.

2. A comparative study of various L2R techniques applied to tag recommendation with a focus on maximizing the relevance of the recommended tags. We compare eight L2R techniques, including the state-of-the-art *GP*, *RankSVM* and *Rankboost* based methods as well as five techniques that have not been previously exploited for tag recommendation. These techniques are *Random Forest* (*RF*), *MART*, $\lambda$-*MART*, *ListNet* and *AdaRank*. Our results indicate that L2R techniques provide significant gains over a state-of-the-art unsupervised heuristic. Among the L2R methods, we found a winning group of methods (*RF*, *MART* and $\lambda$-*MART*), with a slight advantage of two methods (*RF* and $\lambda$-*MART*) over the others. Furthermore, we find that the L2R approach presents a very low additional recommendation time when compared with unsupervised heuristics. Besides the promising results, the flexibility of the L2R framework in terms of the incorporation of new attributes and ability to maximize different target measures makes it an attractive solution for the tag recommendation problem.

3. Analysis of various syntactic patterns (e.g., part-of-speech labels, syntactic dependencies between words in a sentence) of the text associated with Web 2.0 objects that can be exploited to identify and recommend tags. We verified that the texts in each studied Web 2.0 application present patterns that provide good evidence of which words are good candidate tags. For example, in LastFM, various tags correspond to the music genre of artists and appear in sentences with the structure "X is a Y band" where "Y" is a tag.

4. Proposal of three new tag recommendation methods to tackle cold start. The first method, called $RF_{synt}$, extends the aforementioned approach based on $RF$ to include the new tag quality attributes related to the identified syntactic patterns. The second method, $KNN_{synt}$, rely on the initial set of recommendations provided by $RF_{synt}$ to recommend tags from the neighborhood of the target object. Finally, the third method, $RF_{synt}$+$KNN_{synt}$, is an aggregation of the ranking provided by the other two methods. Our experiments showed that our proposed syntactic attributes are responsible for significant improvements ($RF_{synt}$ outperforms $RF$ with gains of up to 17% in precision considering the cold start scenario). $KNN_{synt} + RF_{synt}$, in turn, provides precision gains of up to 21% over $RF$.

5. Proposal of four new tag recommendation methods that exploit novelty and diversity in addition to relevance. Our first method, called $GP_{rnd}$, is a Genetic Programming based tag recommender that extends the relevance-driven method $GP$ to include novelty and diversity metrics at both attribute and objective function levels. $GP$ was chosen due to its flexibility and ease to incorporate new aspects to its objective function. The second method, called $RF_t$, extends the aforementioned relevance-driven approach based on $RF$, which already incorporates some novelty aspects at the "attribute level", to include new tag attributes that capture the extent to which a candidate tag is related to the topics (e.g., categories) of the target object. This solution indirectly captures topic diversity while trying to maximize relevance in its objective function. Unlike $RF_t$, our third method, Explicit Tag Recommendation Diversifier ($xTReD$), *directly* exploits topic diversity by *re-ranking* the recommendations provided by any tag recommender. Finally, our fourth proposal, called Explicit Tag Recommendation Diversifier with Novelty Promotion ($xTReND$), generalizes $xTReD$, to fully exploit relevance, novelty and topic diversity. Although independent, our solutions build upon each other to provide further improvements as we shall see in our experimental evaluation. Although relevance, novelty and diversity of recommendations may seem to be

conflicting objectives, our results show that it is possible to effectively increase novelty and diversity with only a slight impact on relevance.

6. Advances in personalized tag recommendation. We applied our best L2R method (*RF*) to the personalized tag recommendation task, producing results that are significantly superior to the results of a state-of-the-art personalized tag recommender based on *Pairwise Interaction Tensor Factorization* (*PITF*) [Rendle and Schmidt-Thie, 2010]. We also provide a quantitative assessment of the benefits of personalized tag recommendation to provide better descriptions of the target object. Comparing our best personalized and object-centered tag recommenders, both based on the *RF* technique, we find that the former outperforms the latter, with average gains of 15% in relevance.

7. A comprehensive experimental evaluation which explores the tradeoffs between relevance, diversity and novelty for tag recommendation, and demonstrates the effectiveness of our new methods over state-of-the-art approaches. We evaluate our strategies using real data from five Web 2.0 applications, namely, Bibsonomy, LastFM, MovieLens, YahooVideo and YouTube.

## 1.4 Outline

The rest of this thesis is organized as follows. Chapter 2 discusses related work, while Chapter 3 states our target problem. Chapter 4 presents the metrics exploited as attributes by the tag recommenders, which in turn are introduced in Chapter 5. Chapters 6 and 7 describe our experimental methodology and results, respectively. Finally, Chapter 8 presents the conclusions and directions for future work.

# Chapter 2

# Related Work

In this chapter, we review related efforts, starting by presenting existing relevance-driven tag recommendation methods in Section 2.1. Next, we present related work on novelty and diversity in the general context of recommendation and search (Section 2.2). Finally, in Section 2.3, we review general studies on tag analysis.

## 2.1 Relevance-Driven Tag Recommendation

We start by presenting previous tag recommendation methods, which focus only on a single aspect of the problem (relevance). To summarize and organize these methods, we here propose a taxonomy that group them according to multiple criteria. Our taxonomy, depicted in Figure 2.1, is presented considering two levels: The first level contains the four classification criteria we used, namely: (1) the *target* of the recommendations, (2) their *objectives*, (3) the *data sources* the methods exploit, and (4) the underlying *techniques* they employ. All tag recommendation methods can be grouped according to each of the four proposed criteria. The second level, or the "leaves" of the trees, corresponds to the existing classes that arise from each criterion. We note that, as we will mention below, many tag recommendation methods can be associated with more than one of these classes. For another overview of taging systems and recommendation techniques, we refer to [Marinho et al., 2012].

Regarding the first criterion (target of recommendations), previous tag recommendation methods can be divided into two categories: the *object-centered* methods take the object as main target, aiming at providing tags that properly describe its content. They provide the same recommendations regardless of the target user. *Personalized* methods, in turn, take the pair user-object as target, aiming at providing recommendations that not only describe well the target object, but also satisfy the

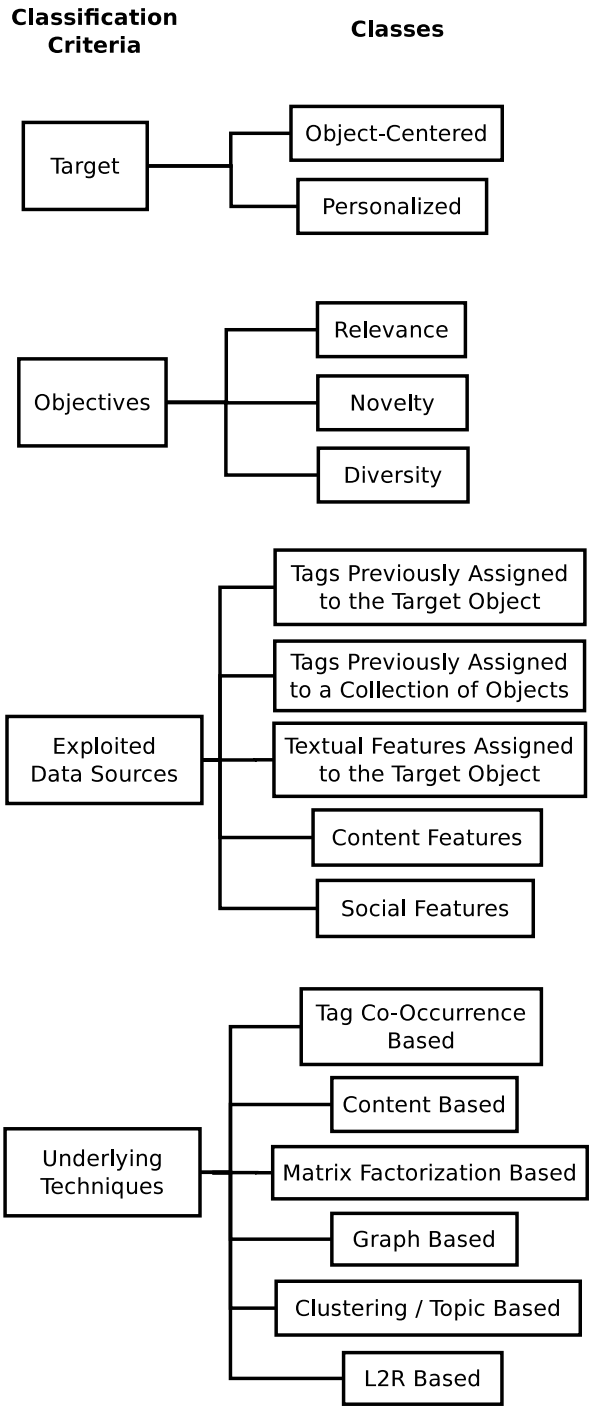



**Figure 2.1.** A taxonomy for previous tag recommendation methods.

target user's interests.

Considering the second criterion (objectives), previous tag recommendation methods have explicitly addressed only relevance as objective. Unlike these previous efforts, in this thesis, we aim at maximizing a combination of relevance, novelty and diversity.

Regarding the third classification criterion (data sources), previous methods have exploited: (1) tags previously assigned to the target object, (2) tags previously assigned to a training collection of objects, (3) textual features (other than tags), such as title, description and user comments, (4) rich media content, that is, image, audio or video, and (5) social features, such as friendship links in social networks and other interactions among users.

Finally, regarding the fourth classification criterion (underlying techniques), tag recommendation methods can be divided into six groups. *Tag co-occurrence based methods* are based on association rules, exploring tags previously assigned to a training collection of objects. They estimate the relevance of a candidate tag by the frequency at which they co-occur with tags previously assigned to the target object. In other words, given the initial set of tags $I_o$ of the target object $o$, tags that are often used jointly with tags in $I_o$ are considered good candidates to be recommended to $o$.

*Content based methods*, in turn, extract candidate tags from the target object (for example, tags related to visual features extracted from an image) or from its features (for example, textual features). Their assumption is that the most relevant information (i.e., tags) are contained in the content of the target object itself or in its associated features.

*Matrix factorization methods* model tag assignments as a matrix and apply dimensionality reduction methods on that matrix. Their goal is to recommend tags by predicting relationships between users, tags and objects from a smaller and possibly less noisy representation of the tagging data.

*Graph-based* methods model the tagging system as a graph (for example, objects are nodes and there is an edge between two nodes if they are similar). Their goal is to extract new candidate tags by exploiting the neighborhood of the target object and/or target user.

*Clustering based methods*, in turn, apply clustering techniques to group objects and tags and recommend the most representative tags of the target object's cluster. They assume that it is possible to extract relevant tags from the tags that describe the main topics of the target object and/or user.

Finally, *learning-to-rank* (L2R) based methods are supervised methods that automatically learn a recommendation function based on training examples of ranked candidate tags. Each candidate tag is associated with a vector of tag quality attributes. These attributes may be generated from results of any of the aforementioned techniques, even other L2R-based methods. The objective of L2R techniques is to automatically combine different pieces of evidence (i.e., attributes or features) of tag quality, generating a model (function) that maps these attributes into a score or rank

position, given the recommendation objective (relevance, novelty, diversity).

We note that, as we will see in the following sections, these groups are not completely disjoint, since most tag recommendation methods employ multiple techniques, as well as exploit several data sources. We here refer to these methods as *hybrid* tag recommendation strategies. For example, some methods employ both tag co-occurrence and content-based techniques, combining them by means of a L2R-based technique.

Next, we present an overview of previous tag recommendation methods, grouping them according to their underlying technique (Sections 2.1.1-2.1.6). In Table 2.1, we classify these methods according to the taxonomy proposed here.

### 2.1.1 Tag Co-occurrence Based Methods

Tag co-occurrence based methods exploit tags previously assigned to a collection of objects to extract *tag co-occurrence* patterns. In particular, many of them exploit these patterns to expand an initial set of tags $I_o$ associated with an object $o$ [Garg and Weber, 2008; Heymann et al., 2008; Krestel et al., 2009; Menezes et al., 2010; Sigurbjörnsson and Zwol, 2008; Wu et al., 2009]. For this purpose, Heymann et al. [2008] used association rules, i.e., implications of the form $X \rightarrow y$, where the antecedent $X$ is a set of tags, and the consequent $y$ is a candidate tag for recommendation, restricting the rules by a confidence threshold. However, the authors did not provide a ranking of the recommended tags. Sigurbjörnsson and Zwol [2008], on the contrary, exploited simple global metrics of tag co-occurrence (e.g., confidence), applying them over all tags in the initial set to produce a final ranking of candidate tags. They also exploited some metrics related to tag frequency to capture the "relevance" of each candidate.

Due to efficiency issues, most of these strategies usually compute co-occurrences between only two tags (i.e., $X$ contains only one tag), possibly missing important co-occurrence relationships. To address this problem, Menezes et al. [2010] proposed LATRE - Lazy Associative Tag Recommendation, which computes association rules in an on-demand manner [Veloso et al., 2006], allowing an efficient generation of more complex and potentially better rules. LATRE produced superior results in comparison with the best method proposed by Sigurbjörnsson and Zwol [2008].

Some studies employed tag co-occurrence techniques to address the personalized tag recommendation problem [Garg and Weber, 2008; Rae et al., 2010]. Garg and Weber [2008] proposed an interactive method. While a user enters/selects new tags for an object, the system suggests related tags to her, based on tags she or other people have already used in the past along with (some of) the tags already en-

**Table 2.1.** Classification of tag recommendation methods.

| | | Exploited Data Sources | | | | | Underlying Techniques | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Method** | Tags Previously Assigned to the Target Object | Tags Previously Assigned to Other Objects | Textual Features (Other than Tags) | Rich Media Content | Social Features | Tag Co-occurrence Based | Content-Based | Matrix Factorization | Graph-Based (including collaborative filtering) | Clustering/Topic Based | L2R-Based |
| **Object-Centered Methods** | Heymann et al. [2008] | ✓ | ✓ | | | | ✓ | | | | | |
| | Sigurbjörnsson and Zwol [2008] | ✓ | ✓ | | | | ✓ | | | | | |
| | LATRE Menezes et al. [2010] | ✓ | ✓ | | | | ✓ | | | | | |
| | Wu et al. [2009] | ✓ | ✓ | | ✓ | | ✓ | ✓ | | | | ✓ |
| | Pedro et al. [2011] | | ✓ | | ✓ | | | | | ✓ | | |
| | Lin et al. [2012] | | ✓ | | ✓ | | ✓ | ✓ | | ✓ | | |
| | Zhu et al. [2014a] | | ✓ | | ✓ | | | ✓ | | ✓ | | |
| | Lipczak and Milios [2011] | | ✓ | ✓ | | | ✓ | ✓ | | | | |
| | Wang et al. [2009] | | ✓ | ✓ | | | ✓ | ✓ | | | | |
| | Lu et al. [2009] | | ✓ | ✓ | | | | | | ✓ | | |
| | Zhang et al. [2009] | | ✓ | ✓ | | | | ✓ | | ✓ | | |
| | Cao et al. [2009] | | | ✓ | | | | | | | | ✓ |
| | Ribeiro et al. [2015] | | | ✓ | | | | ✓ | | | | ✓ |
| | Martins et al. [2013, 2016] | | ✓ | ✓ | | | ✓ | ✓ | | | | ✓ |
| | Song et al. [2011, 2008] | ✓ | ✓ | ✓ | | | | | | ✓ | ✓ | |
| | Krestel et al. [2009] | ✓ | ✓ | | | | | | | | ✓ | |
| | Belém et al. [2011] | ✓ | ✓ | ✓ | | | ✓ | ✓ | | | | ✓ |
| | *This thesis* | ✓ | ✓ | ✓ | | | ✓ | ✓ | | ✓ | | ✓ |
| **Personalized Methods** | Chen and Shin [2013] | | ✓ | ✓ | | ✓ | | ✓ | | ✓ | | |
| | Yin et al. [2013] | | ✓ | ✓ | | ✓ | | | | | | |
| | Rae et al. [2010] | ✓ | ✓ | | | ✓ | ✓ | | | | | |
| | Liu et al. [2010] | | ✓ | | | ✓ | | | | ✓ | | |
| | Garg and Weber [2008] | ✓ | ✓ | | | | ✓ | | | | | |
| | Hu et al. [2010] | ✓ | ✓ | | | | ✓ | | | | | |
| | FolkRank (Jäschke et al. [2007]) | | ✓ | | | | | | | ✓ | | |
| | Guan et al. [2009] | | ✓ | | | | | | | ✓ | | |
| | Feng and Wang [2012] | | ✓ | ✓ | | | | | | ✓ | | |
| | Lops et al. [2013] | | ✓ | ✓ | | | | ✓ | | ✓ | | |
| | PITF (Rendle et al. [2010]) | | ✓ | | | | | | ✓ | | | |
| | He and Chua [2017] | | ✓ | | | | | | ✓ | | | ✓ |
| | Yuan et al. [2017] | | ✓ | | | | | | ✓ | | | ✓ |
| | Nguyen et al. [2017] | | ✓ | | ✓ | | | ✓ | ✓ | | | |
| | Krestel and Fankhauser [2012] | ✓ | ✓ | | | | | | | | ✓ | |
| | *This thesis* | ✓ | ✓ | ✓ | | | ✓ | ✓ | | ✓ | | ✓ |

tered. The suggested tags are dynamically updated with every additional entered tag. Rae et al. [2010] extended the strategy proposed in [Sigurbjörnsson and Zwol, 2008] to address personalized tag recommendation, exploiting tag co-occurrences in different contexts: (1) the whole data collection, (2) the objects of a specific user, (3) the social

contacts of the user, and (4) the groups in which the target user is included.

The advantage of tag co-occurrence methods are three-fold: (1) tag co-occurrences are simple to compute, (2) these methods exploit one of the strongest evidence of tag relevance, and (3) their main data source, the history of tag assignments, is commonly available. However, this group of methods may be seriously affected by the cold start problem, that is, the absence of previous information about the target object (previously assigned tags, in the case studied by Martins et al. [2016]). The vast majority of these methods rely on an initial set of tags associated with the target object which may not be available, for instance, for a newly inserted object. Another problem related to cold start is when the tagging service is new and the history of tag assignments is still very sparse. One common solution to tackle cold start in tag recommendation as well as in the general (item) recommendation context, is to exploit features of the content of the recommendation target (object or user). We describe content-based methods next.

### 2.1.2 Content Based Methods

This group of methods extracts tags from the content of the target object and its associated features, or from features of the target user's profile. A commonly exploited group of object features is the set of the object's textual features such as title, description and user comments. For example, Wang et al. [2009] used extracted candidate terms from the object's textual features, and the traditional TFIDF metric to rank these terms by their relevance to the target object.

Lipczak et al. [2009] and Lipczak and Milios [2011] proposed a hybrid method that extracts terms from the title and description of the target object (a content-based technique) and then expands the set of candidate tags by exploiting tag co-occurrences. They also measured the relevance of the extracted terms by their usage as tags in a training set.

In addition to tag candidates extracted from the textual features, we [Belém et al., 2011] exploited tags that co-occur with tags previously available in the target object and combined various metrics that estimate tag relevance using heuristics and learning-to-rank techniques. Among the metrics we exploited, we found that the term spread (TS), which is defined as the number of *different* textual features that contain the candidate tag, performs better than the traditional term frequency (TF), which counts all repetitions of the same term (candidate tag) in the same textual feature. We also found that further gains can be obtained when considering different weights for different textual features, since some of them, such as the title, usually

present higher descriptive capabilities.

Another content-based tag recommendation method was exploited to produce tag clouds that describe academic experts [Ribeiro et al., 2015]. The authors generated tag candidates from various textual features associated with the publications of the target expert, such as title, abstract and keywords. The candidate tags are ranked according to relevance metrics such as term frequency and coverage, which are combined by L2R techniques. The authors found that traditional content-based tag recommenders perform well at identifying expertise-oriented tags, with article keywords being a particularly effective source of evidence across profiles in different knowledge areas and with various levels of sparsity. Moreover, the L2R approach provided further improvements for expertise profiling.

A feature of the target user commonly exploited by content-based tag recommendation methods is the user's history of tag assignments, which is a good evidence of her interests [Belém et al., 2014]. Lipczak et al. [2009], as well as our personalized tag recommendation methods, extract tag candidates from the target user's history.

Other methods exploit the rich media content associated with the target object [Lin et al., 2012; Pedro et al., 2011; Siersdorfer et al., 2009; Wu et al., 2009; Zhu et al., 2014a]. For example, Wu et al. [2009] computed co-occurrences between tags and visual features extracted from images and exploited a L2R technique called Rankboost [Freund et al., 2003b] to generate the final ranking function. Lin et al. [2012] performed a random walk process over the graph of images with similar visual content. In this graph, the nodes are objects (images in this case) and there is an edge connecting two objects if they present similar visual features. Similarly, Pedro et al. [2011]; Siersdorfer et al. [2009] created a graph of videos based on content similarity and produced recommendations by propagating tags through its edges[1]. However, the *semantic gap* is still a challenge to generate accurate tags exploiting rich media, because the visual similarity between images or videos may not reflect the strength of their relationship [Zhu et al., 2014a]. To mitigate this limitation, Zhu et al. [2014a] proposed a random walk model with adaptive teleportation probabilities.

In addition to textual features associated with the content of the target object, Chen and Shin [2013] exploited social features to recommend tags. They considered tags that frequently appear in objects that are marked as favorite by the target user as candidates for recommendation.

Content-based techniques are commonly exploited to mitigate the cold start prob-

[1]Note that when exploiting rich media content, since the tag candidates are not extracted directly from the content, other techniques such as graph-based are jointly exploited. These techniques are described in the following sections.

lem (absence of initial tags). For example, Martins et al. [2013, 2016] evaluated the impact of the cold start on a family of state-of-the-art tag recommendation methods. They showed that the effectiveness of these methods suffers when they cannot rely on previously assigned tags in the target object. Moreover, exploiting other sources of tag candidates by means of automatic filtering strategies yields limited gains. Thus, the authors proposed a new strategy that exploits both positive and negative relevance feedback from the users to iteratively select input tags to these tag recommendation methods. The proposed strategy generated significant gains (up to 45% in precision) over the best considered baseline. It was also found robust to the lack of user cooperation. However, the drawback of exploiting user feedback (both positive and negative) is that they represent an additional user effort in the recommendation process, and may be impacted by the lack of user cooperation.

The main issue with content-based techniques is the possible lack of novelty: recommending terms that are already assigned to the content (even if they still do not appear as tags) may be less useful than generating more complementary and diversified tags from other sources. Similarly, tags in which the user has previously showed interest are probably accurate and represent well the user's (past) interests, but they may not capture new interests. Besides that, these methods have to deal with a large amount of noise in textual and content features [Figueiredo et al., 2012].

### 2.1.3 Matrix Factorization Based Methods

The most representative method of this group is PITF (Pairwise Interactions Tensor Factorization), a winning method in the *2009 PKDD Discovery Challenge* competition [Rendle and Schmidt-Thie, 2010]. In this method, the tensor (i.e., a tridimensional matrix) that models the pairwise interactions among users, items and tags (i.e., the ranking preferences of the tags for each pair user-object) is factorized in lower dimensional matrices to reduce noise. The PITF model is learned from an adaption of the Bayesian personalized ranking (BPR) criterion.

The advantage of this method is the dimensionality reduction, which may reduce noise and the complexity of posterior computations. However, the cost of matrix factorization operations exacerbates scalability problems. Moreover, data sparsity is a major issue for these techniques. Although matrix factorization has been originally applied to denser data (i.e., data in which unpopular tags, users and objects were filtered out) [Rendle and Schmidt-Thie, 2010], it was greatly outperformed by the hybrid methods proposed in this thesis, which do not assume such kind of filtering.

More recently, He and Chua [2017] propose Neural Factorization Machine (NFM),

a method that combines the linearity of factorization machines (FM) in modelling second-order attribute interactions and the non-linearity of neural networks in modelling higher-order attribute interactions. Similarly, Yuan et al. [2017] propose BoostFM, which integrates boosting into factorization models for general item recommendation. Nguyen et al. [2017] exploit not only the tagging history, but also visual features of images, such as the objects appearing in the image, colors, shapes or other visual aspects, into FM models. We do not compare our methods with these approaches because their drawbacks are similar to the PITF-based technique, which we indeed evaluate and compare to our methods.

### 2.1.4 Graph Based Methods

Graph based tag recommendation methods extract candidates from the neighborhood of the target object and/or user. The nodes of the graph correspond to objects or users, and there is an edge between two objects (or two users) if they are similar. The main sources to compute similarity are the textual features of the objects, including the tags and the folksonomy, that is, the tagging history of the users. Alternatively, visual features extracted from image and video objects can be used to estimate content similarity.

Collaborative filtering-based techniques fall in this category, since they exploit the history of users that are similar to the target user (for example, they share tags in common). For instance, Jäschke et al. [2007] built a similarity graph in which the vertices are users and each edge connects two users that share tags in common, being weighted by some similarity measure such as Jaccard coefficient [Baeza-Yates and Ribeiro-Neto, 1999]. This method recommends to a user $u$ the tags assigned by the $k$ most similar users to $u$. Another method proposed by the same authors, named *FolkRank*, is based on the well known *PageRank* algorithm [Brin and Page, 1998]. The rationale of this method is that an object that receives relevant tags from important users also becomes important, that is, a good source for recommendation. Symmetrically, a tag is relevant if it was associated with important objects by important users. Thus, a mutual reinforcement graph is built, allowing the scoring and recommendation of tags. Hu et al. [2010] proposed a probabilistic method that exploits the vocabulary of the target user and object, as well as the vocabulary of similar users. The authors proposed to use the Kullback-Leibler divergence to estimate the similarity between two users.

Lu et al. [2009] as well as Zhang et al. [2009] proposed to propagate tags between objects with similar textual content, while Feng and Wang [2012] modeled the folksonomy as a heterogeneous graph containing tags, users and objects as nodes.

They employed an optimization strategy, OptRank, to learn the weights of the edges that connect these nodes. Lops et al. [2013] exploited both collaborative filtering and content-based tag recommendation techniques. The former exploits the user and community tagging behavior to produce recommendations, while the latter exploits some heuristics to extract tags directly from the object's textual content. Liu et al. [2010] enhanced their graph-based tag recommender by exploiting explicit social links between users. Guan et al. [2009] modeled personalized tag recommendation as a "query and ranking" problem and proposed a graph-based ranking algorithm for interrelated multi-type entities, namely, tags, users and documents. When a user issues a tagging request, both the document and the user are treated as a part of the query. Tags are then ranked by the graph-based ranking algorithm which takes into consideration both document relevance and user preference.

More recently, Yin et al. [2013] addressed not only the problem of recommending tags, but also of predicting different kinds of relationships (such as relations among users, comments and items, and social links between users). By exploiting a generalized latent factor model and Bayesian inference, the authors found that connecting comments and tags within the same model allows mutual reinforcement and improves prediction accuracy.

This group of methods is relatively less affected by the cold start problem than the tag co-occurrence based methods. However, they usually deal with more noise (originating from other textual or content features) when compared to content-based techniques.

### 2.1.5 Clustering Based Methods

Another group of methods recommend tags based on *clusters* or *topics* of objects. For example, Song et al. [2011, 2008] proposed two clustering based methods. The first method represents the tagged data in two bipartite graphs: a document-tag graph and a document-word graph. In a document-tag graph, tags and objects (documents) are nodes and there is an edge between a document $d$ and a tag $t$ if $t$ was assigned to $d$ by at least one user (Figure 2 in Section 2 illustrates some of these edges). This method finds document topics by leveraging graph partitioning algorithms. The second method aims at finding the most representative documents within the data collections and advocates a sparse multiclass Gaussian process classifier for efficient document classification. For both methods, recommendations are performed by first classifying a new document into one or more clusters, and then selecting the most relevant tags from those clusters as recommended tags.

Krestel et al. [2009] used Latent Dirichlet Allocation (LDA) [Blei, 2012] to assign multiple topics to objects and tags, and recommend tags to an object based on its topics. LDA is a probabilistic model based on the assumption that a document can be represented as a mixture of different topics [Blei, 2012], whereas a topic is defined as a distribution of words from a fixed vocabulary. Krestel and Fankhauser [2012] presented a personalized version of this method.

Clustering may be an interesting strategy to reduce the dimensionality of the problem (exploiting relationships among clusters instead of among entities of the tagging application), and to generate complementary candidate tags that would not be extracted directly from the content of the target object or from similar objects. However, these candidates may be too general (low novelty/specificity), thus being of little use to describe the specific content of the target object or discriminate it from others.

### 2.1.6 Learning-to-Rank Based Methods

Since recommendation is usually modeled as a ranking problem (i.e., we want to recommend the best items first), learning-to-rank techniques constitute a natural approach to tackle it. L2R-based methods are supervised approaches that automatically learn a ranking function based on training examples. Such training examples consist of candidate tags represented as vectors of tag quality attributes to which relevance labels (indicating the tag's relevance level) are assigned (either manually or by exploiting previous tag assignments as ground truth). The objective of this kind of approach is to generate a model (function) that maps the tag quality attributes into a relevance score or rank.

Regarding the application of L2R techniques to the tag recommendation problem, we are aware of a few prior efforts (last column of Table 2.1). Cao et al. [2009] and Wu et al. [2009] exploited RankSVM [Joachims, 2006] and RankBoost [Freund et al., 2003b] as L2R techniques, respectively. In [Belém et al., 2011], we applied both RankSVM and Genetic Programming [Poli et al., 2008] to the tag recommendation problem. Here, we expand our focus comparing eight L2R approaches, extending the main approaches to tackle the personalized tag recommendation problem, and addressing other aspects of the problem, namely, novelty and diversity. Ribeiro et al. [2015] compared the effectiveness of various L2R algorithms, such as Random Forest (RF), Multiple Additive Regression Trees (MART), $\lambda$-MART, AdaRank, ListNet, Ranknet, Coordinate Ascent, Rankboost, RankSVM and Genetic Programming (GP). They found RF, MART and $\lambda$-MART to be the best performing strategies for the tag recommendation problem.

Liu [2009] reviewed existing L2R algorithms in the context of document ranking, categorizing them into three approaches: pointwise, pairwise and listwise. The pointwise approach assumes that each query-document pair in the training data has a numerical score, and thus the L2R problem can be approximated by a regression problem. Pairwise approaches are approximated by binary classification — given a pair of documents, it is necessary to predict which one is the best, while the listwise approach considers the effectiveness of the whole ranking list, typically optimizing a given evaluation measure. The authors analyzed the advantages and disadvantages of each approach, and discussed the relationships between the objective functions used in these approaches and IR evaluation measures. Moreover, experiments using the datasets of the LETOR benchmark indicated that the listwise approach is the most effective among the three approaches.

The advantages of exploiting L2R methods are threefold: (1) they can effectively exploit many attributes in the generation of ranking functions, (2) they can be easily extended to include more attributes and objective functions, and (3) there is a strong theoretical background on learning methods, which has been recently extended for ranking problems [Qin et al., 2010]. A small disadvantage of these methods is the training time necessary to learn the tag recommendation models. However, this step can be performed offline. Moreover, the attribute extraction and the application of the learned models in the online recommendation step usually represents a very small additional cost compared to the recommendation time of unsupervised techniques, as we verified in our experiments.

## 2.2 Novelty and Diversity

Besides relevance, other aspects such as novelty and diversity may also be important to evaluate the *quality* and *utility* of a tag. In fact, result diversification is a problem that has been addressed in other contexts, particularly Web search [Clarke et al., 2012]. In this context, two main families of diversification approaches have emerged to tackle query ambiguity [Santos et al., 2015]. *Implicit* approaches seek to promote diversity by scoring a given search result proportionally to its difference to the results ranked ahead of it, e.g., in terms of these results' textual dissimilarity [Carbonell and Goldstein, 1998] or the divergence of their language models [Zhai et al., 2003]. In contrast, *explicit* approaches seek to diversify the search results on the basis of their coverage of some property of the user's query, such as multiple query categories [Agrawal et al., 2009] or multiple query reformulations [Santos et al., 2010]. Consider-

ing that categories may be absent or noisy (e.g., vague or with non-uniform granularity) in some applications, Yu et al. [2014] proposed the use of latent topics generated by Latent Dirichlet Allocation (LDA) as an alternative to categories and query intents in the problem of query result diversification in e-commerce sites.

In a different direction, Liang et al. [2014] exploit traditional methods of data fusion (i.e., rank aggregation) to improve the diversity of search results. They propose DDF (Diversity Data Fusion), a method which combines data fusion with latent topic diversification. Zhu et al. [2014b], on the other hand, address search result diversification as a learning-to-rank problem, where the scoring function is a combination of relevance and diversity. Finally, Rabinovich et al. [2014] combine rank aggregation with relevance feedback from users. The feedback is exploited with two purposes: (1) ranking documents in intermediate lists, and (2) estimating the effectiveness of each intermediate list in order to improve list recombination.

In association rule mining, similar novelty and diversity concepts have been proposed as measures to evaluate the rule interestingness. According to Geng and Hamilton [2006], a pattern (e.g., association rule) is diverse if its elements differ significantly from each other, while a set of patterns is diverse if the patterns in the set differ significantly from each other. For the same authors, a pattern is novel to a person if he or she did not know it before and is not able to infer it from other known patterns. Thus, novelty in association rules is a subjective measure, depending on the evaluating user.

In the general context of (item) recommendation, previous work mostly focused on implicit approaches to promote novelty and diversity. Celma and Herrera [2008] as well as Vargas and Castells [2011] evaluate novelty and diversity in terms of popularity and dissimilarity of items, based on the idea that novel and diverse items must be different from all items that have been already seen or consumed. Novelty, particularly, was estimated under two perspectives: by the inverse of the popularity of the items (popularity-based perspective) and by the average distance (dissimilarity) of an item to other items in a given context (the application as a whole or a specific user, for example), referred to as distance-based novelty. Diversity, in turn, was estimated as the average *pairwise* distance between recommended items. Note that distance-based novelty and diversity, as previously defined, are closely related but different concepts: the former is taken from the perspective of all other items in a given context, whereas the latter is evaluated within the list of recommended items.

Zhang et al. [2012b] introduce Auralist, a music recommendation framework that promotes diversity, novelty and serendipity (a concept similar to the distance-based novelty from Vargas and Castells [2011]). They show that, although the inclusion of

novelty, diversity and serendipity may slightly impact relevance, it does improve user satisfaction. Lathia et al. [2010], in turn, define novelty and diversity under a temporal perspective, that is, novel/diverse items should be different from what was seen or recommended in the past. Instead of aggregating relevance, novelty and diversity as a single objective, Ribeiro et al. [2012] exploit a multi-objective Pareto optimization algorithm to jointly address these three recommendation quality criteria. The solution in this case is a set of "non dominated" recommendation functions instead of a single function. However, chosing the best solution among the returned set of functions is another non-trivial issue and thus we leave the Pareto approach as future work.

Küçüktunç et al. [2013] and Shi [2013] address the problem of diversified and novel recommendations on graphs. Küçüktunç et al. [2013] model the problem as returning a set of items that extend the history of interests of a user in some items. The only data they assume as available is the graph itself (a social network or a product co-purchasing graph, for example), not relying on pre-defined topics. Their proposed diversity metric, referred as expanded relevance, penalizes recommended items that are close to each other in the graph, and thus present expanded sets (sets of neighbors in the graph) with high intersection and low coverage of the relevant results. Finally, they present a greedy diversification algorithm called BestCoverage, which optimizes the expanded relevance of the result set. In a different direction, Shi [2013] proposes a method based on a first order Markovian graph with transition probabilities between user-item pairs. The author defines a "cost flow" concept, such that items with lower costs are recommended to a user.

Szpektor et al. [2013] address the problem of diversifying question recommendations in Question&Answer applications. According to the authors, showing the users only the main topics in which they had previously expressed interest is not the best strategy to encourage user participation in answering questions. Based on a large-scale online experiment in production in Yahoo! Answers, they find that diversity and novelty promotion allows significant improvements in the number of answers, the daily session length, as well as other activities such as voting.

Despite these previous studies tackling diversity and novelty in item recommendation in general, to our knowledge, there is no previous attempt to explore such aspects in the specific context of tag recommendation.

## 2.3 Tagging Analysis

A related body of previous work focuses on *characterizing* tagging systems, thus producing useful knowledge for the design of tag recommendation systems [Almeida et al., 2010; Figueiredo et al., 2012; Li et al., 2008; Lipczak and Milios, 2010; Rader and Wash, 2008]. For example, Lipczak and Milios [2010] found that the title and the personal tagging history of a user are the main factors that impact tagging decisions, whereas Rader and Wash [2008] showed that personal organization has a stronger impact on tagging decisions than social influences. In another direction, Almeida et al. [2010] and Figueiredo et al. [2012] proposed several metrics to assess the quality of different textual features commonly associated with objects in Web 2.0 applications. They find that the title is the textual feature with the best capacity of *describing* the object's content, followed by tags. These previous studies motivated us to exploit multiple textual features and the user's tagging history as data sources to extract candidate tags and to compute relevance metrics as we describe in Chapter 4.

Other studies address the quality and semantics of tags. For example, Li et al. [2008] found that user-generated tags are consistent with the web textual content with which they are associated, and that they capture the user's interests. In a more recent study, Choi [2015] assessed the quality of tags as subject indexers. The author found that the sets of tags provided by different users are more consistent (similar) among each other than the different indexes generated by professionals. He also found that subject categories showing higher indexing consistency present a more complete set of tags. Finally, they verified a high correlation between the discriminative power of a term and its semantic relatedness to documents. In another recent work regarding tag semantics, Zhang et al. [2012a] studied geo-spatial and temporal relationships between tags, but only apply them to cluster and visualize tags, and *not* to recommend tags.

## 2.4 Summary

In this chapter, we presented the main references related to this thesis. We started discussing previous tag recommendation efforts, classifying them according to their data sources, target of recommendations and employed techniques. In common, they focus only on relevance and personalization issues, while we also address novelty and diversity aspects in this thesis. We also presented related work on novelty and diversity in other domains, namely, search and general (item) recommendation. We concluded this chapter with previous studies on tagging analysis, which we applied in the proposal of our metrics and methods. In the next chapter, we contextualize and

state the problem we address in this thesis.

# Chapter 3

# Contextualization and Problem Statement

In this chapter, we contextualize the tag recommendation problem. We start by defining the target object of the recommendation and its features (Section 3.1). In Section 3.2, we describe the basic elements of a tag recommendation system. We explain the concepts of relevance (from both object-centered and personalized perspectives), novelty and diversity for tag recommendation in Section 3.3. Finally, we formally define the problem we address in this thesis in Section 3.4.

## 3.1 Tags and Objects on the Web 2.0

Each page in a Web 2.0 application is composed by a main *object*[1] (e.g., a textual document, an audio, a video, an image) and various sources of information related to the object, here referred to as its *features*. These features can be classified as content features, textual features, user profile features and social features [Figueiredo et al., 2012].

*Content features* can be extracted from the main object, such as the color histogram of an image. *Textual features*, in turn, comprise the self-contained textual blocks that are associated with an object, usually with a well defined functionality [Figueiredo et al., 2012][2]. Examples of textual features commonly found in different applications are title, description, categories, tags and user comments. In particular,

[1]Some references (e.g., [Lipczak and Milios, 2011; Rendle and Schmidt-Thie, 2010]) use the term *resource* instead of *object*.

[2]Note that these two sets of features may not be disjoint (e.g., when the main object is a textual document).

*tags* are keywords freely assigned by users to succinctly describe the content of the object. As tags are freely created by users, they are not necessarily unigrams (unless the application automatically split them by whitespaces). Thus, compound words can be used as tags, either separated by spaces, hifenized or joined.

Figure 3.1 shows a MovieLens page containing textual features assigned to an object (here represented as a picture of the movie).

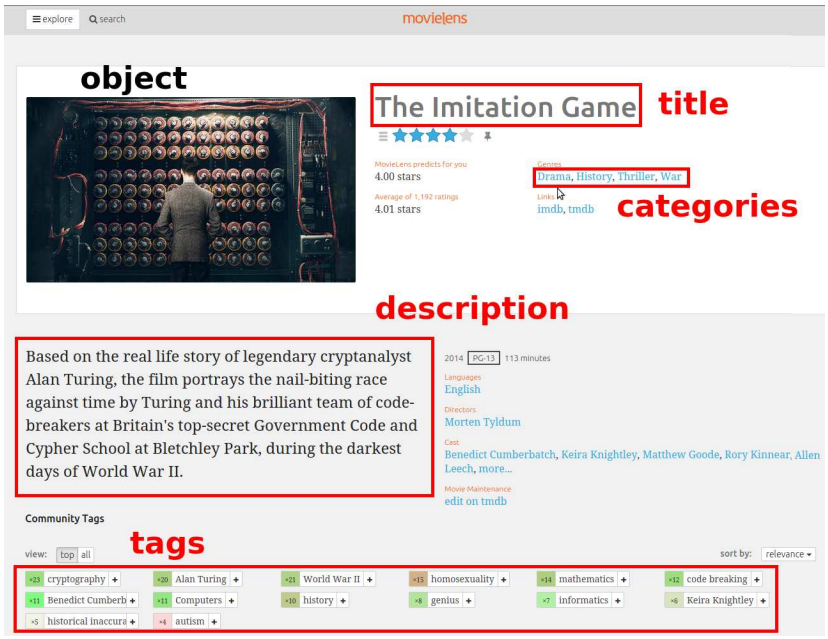



**Figure 3.1.** Web 2.0 page and some of its textual features.

*User profile features* may refer to characteristics of the user who created and uploaded the content or who assigned tags to it, or her interactions with the application. Finally, *social features* refer to interactions among users (e.g., explicit friendship links, subscriptions, upvotes, etc). In particular, while friendship links are explicit indicators of the social connections among users, subscriptions (connections established among users that show interests in one another's content), and endorsements (e.g., "upvotes") are more implicit indicators of the social relationship among these users. Examples of these features may be visualized in Figure 3.2.

The Web 2.0 tags, objects and users form the basic structure of the *folksonomies*. A fusion of the words *folks* ("people") and taxonomy (*taxis* means "classification", while *nomos* or *nomia* means "management"), folksonomy refers to the categorization of objects using freely chosen keywords by users. Unlike a taxonomy, which provides a hierarchical categorization with well-defined classes, a folksonomy establishes categories (each tag may be considered a category) without stipulating or necessarily deriving a hierarchical structure of tags [Quintarelli, 2005; Spiteri, 2007].

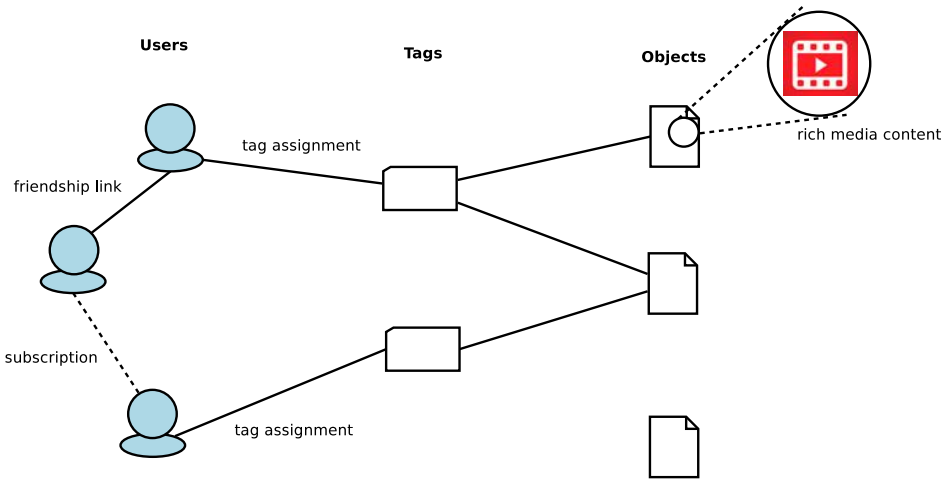



**Figure 3.2.** Examples of features commonly found in Web 2.0 applications. Friendship and subscription links established through the application are examples of *social features*. The set of tags a user assigned to objects in the applications may be considered one of the *user profile features*. Features extracted from the content of the main object (e.g., color histogram) are *content features*.

Formally, a folksonomy is defined as a relational structure $F = (U, T, O, \mathcal{P})$, where $U$, $T$ e $O$ are finite sets composed by users, tags and objects, respectively, and $\mathcal{P}$, the set of postings, is a ternary relation between these elements, that is, $\mathcal{P} \subseteq U \times T \times O$, as defined by Jäschke et al. [2007]. Thus, each element $(u, t, o) \in \mathcal{P}$ represents the assignment of a tag $t$ to an object $o$ by a user $u$ (illustraded as the edges connecting users, tags and objects in Figure 3.2). Wal [2005] identified two types of folksonomies: broad and narrow. A broad folksonomy arises when multiple users can apply the same tag to an object, providing information about which tags are the most popular ones. Examples of broad folksonomies include the online radio station LastFM[3] and the publication sharing application Bibsonomy[4]. A narrow folksonomy occurs when only one user (typically the target object's creator) can tag a given object. The photo sharing site Flickr[5] is an example of narrow folksonomy. While both broad and narrow folksonomies allow the content organization and findability, a broad folksonomy enables to rank the assigned tags by their popularity, as well as tracking of emerging trends in tag usage and developing vocabularies. Commonly, tag popularity in broad folksonomies is visualized in *tag clouds* [Venetis et al., 2011], which also provide an easy way to navigate the tags, objects, and users of a folksonomy.

[3]http://www.last.fm
[4]http://www.bibsonomy.org
[5]http://www.flickr.com/

## 3.2 Tag Recommendation Systems

There are various kinds of recommendation systems, each focused on a different kind of item, such as products in a e-commerce site, books in a digital library, and users in a social network. In this thesis, we are interested in recommending tags. A tag recommendation system usually assists users, providing a list of keywords that ideally describe the content of the object. Thus, the target of tag recommendations is a pair user-object, although it may be personalized or not, that is, it may consider only the target object or it may jointly consider both the object and the user.

As data sources for our tag recommendation strategies, among the various features described in Section 3.1, we focus on three dimensions: (1) tags previously assigned to the target object (when available), (2) other textual features, namely, *title*, *description* and *categories* and (3) the target user's tag assignment history, for personalized methods. The first dimension is based on the hypothesis that tags which frequently co-occur with the tags of an object $o$ are good candidates to be recommended to $o$. The second data source, multiple textual features associated with an object, in turn, may contain various relevant terms to describe it. Finally, the third dimension is motivated by the hypothesis that tags previously assigned by a user are a strong evidence of their interests. Social and rich media content features are left to be exploited in future work, because they did not provide promising results in preliminary experiments.

## 3.3 Relevance, Novelty and Diversity Concepts

Tag relevance can be defined in two perspectives. In an *object-centered* perspective, a tag is relevant if it correctly describes the content of the target object. In a *personalized* perspective, a relevant tag not only describes well the content of the target object, but also matches the interests of the target user [Rendle and Schmidt-Thie, 2010]. Note that, by this definition, the relevance of a tag in a recommendation list does not depend on the other tags provided in the list. Given that a recommendation satisfies the user's need, the usefulness of similar recommendations is arguable according to Vargas and Castells [2011]. Thus, the novelty and diversity concepts should be considered in addition to relevance.

Novelty and diversity definitions may vary according to the contexts they are employed, namely, Web search and item recommendation. Following, we will present the existing definitions of these concepts in each context, adapting them to our specific tag recommendation context.

In information retrieval, similarly to recommendation, promoting relevance alone

may not result in an optimal effectiveness, particularly in search scenarios that are permeated with ambiguous queries and redundant information items [Santos et al., 2015]. A relevance-oriented ranking assumes that the relevance of a document can be estimated with certainty and independently of the estimated relevance of the other retrieved documents. While the first assumption is challenged by ambiguity (multiple interpretations or intents) in the user's query, the second assumption is challenged by redundancy (unnecessary documents covering the same information need) in the search results. Thus, for IR, *novelty* and *diversity* have been defined as a means to tackle redundancy and ambiguity, respectively. That is, novelty in search results ensures that each document brings "new information" with relation to previously ranked documents, while diversity ensures a high *coverage* of the multiple interpretations of the query. This increases the chance that at least one document will satisfy the information need of the user.

As discussed in Chapter 2, in the general recommendation context, the *novelty* of a recommended item refers to how different this item is with respect to what has been previously seen or consumed by a specific user, or by a community as a whole [Vargas and Castells, 2011]. In this context, novelty is commonly associated with item rarity or serendipity [Castells et al., 2011; Celma and Herrera, 2008; Zhang et al., 2012b]. While this concept is suitable for recommendation, because its purpose in general is to expose the user to "novel" experiences, this kind of novelty does not apply for Web search.

In this thesis, we also define the *novelty* of a tag from the perspective of its popularity in the application. That is, we estimate the novelty of a tag by the inverse of the frequency at which the tag is used in the collection. A term used as tag a large number of times tends to be a more "obvious" recommendation (if relevant at all), thus being of little use (if any) to improve the description of the target object provided by its tag set. We note that, according to this definition, noisy terms such as typos may be considered highly novel. However, our methods jointly exploit novelty, relevance and diversity, thus minimizing the chance of recommending noise.

We also note that this definition of novelty is closely related to tag *specificity*, since rare words tend to be more specific (less general). According to Baeza-Yates and Ribeiro-Neto [1999] as well as Choi [2015], specificity is a property of the term semantics, i.e., a term or tag is more or less specific depending on its meaning. For example, "feline" is less specific than "cat" or "persian". One would expect that the most general term "feline" would be used to describe a larger number of objects than the more specific terms. This interpretation of specificity is based on the accuracy of the term as a descriptor of an object's topic. As an alternative, specificity can be inter-

preted as a statistical property of the term use, being estimated as an inverse function of the number of objects in which a term occurs, which is exactly our tag novelty concept. However, we chose the term "novelty" instead of "specificity" to keep consistency with the general recommendation literature [Celma and Herrera, 2008; Vargas and Castells, 2011; Zhang et al., 2012b], which estimates novelty similarly. A related property of object descriptions is the *exhaustivity*, which is defined as the coverage they provide for the main topics of the object [Baeza-Yates and Ribeiro-Neto, 1999; Choi, 2015]. This fits exactly in our tag diversity concept, as we will discuss below.

The *diversity* of a list of recommended items refers to the capacity of the list of recommended items to cover the multiple topics the target user is interested in. Two types of diversification approaches have been exploited in the general context of information retrieval: an implicit and an explicit approach. The former exploits properties of the recommended item [Vargas and Castells, 2011] (or the retrieved document in search), while the latter exploits properties of the target of the recommendation or from the query (e.g., the multiple topics a user is interested in, the multiple interpretations of an ambiguous query).

Based on these ideas, we also propose two diversification approaches for tag recommendation. In our first tag diversification effort, we tackle diversity *implicitly* as the average pairwise semantic distance between the top recommended tags, such that a list of synonyms or semantically related words present low diversity. This definition is exploited by one of our new methods, $GP_{rnd}$, presented in Section 5.5.1.

We also consider an *explicit* diversification approach, which is employed in the design of three of our proposals: the re-ranking strategies *xTReD* and *xTReND* and the Random Forest based method with topic-related attributes, $RF_t$, which are presented in Section 5.5.2. The idea is that a diversified list of tags must cover as many topics related to the target object as possible, and as early in the ranking as possible.

**Table 3.1.** Novelty and Diversity Definitions

| **Context** | **Novelty** | **Diversity** |
|---|---|---|
| **IR** | A means to tackle redundancy | A means to tackle ambiguity |
| **Recommendation** | Unexpectedness / Capacity of bringing items that are different from other items in a given context | Capacity of covering the different topics the target user is interested in |
| **Tag Recommendation** | Specificity | Exhaustivity |

Table 3.1 summarizes the above discussion by presenting the alternative definitions of novelty and diversity in different contexts, including our own. Next, we formally

define our target recommendation problem. In this definition and throughout the rest of this thesis, we use the term *novelty* to refer to the aforementioned popularity-based perspective of novelty. For diversity, we use the terms *explicit diversity* and *implicit diversity* to refer to the topic-related and dissimilarity-based concepts, respectively, or simply *diversity* when both concepts are suitable.

## 3.4 Problem Statement

Let $U$, $O$ and $T$ be the sets of users, objects and tags of a Web 2.0 application, respectively. The proposed tag recommendation strategies are based on the following sources of information:

(1) the set of tag assignments or folksonomy $\mathcal{P} \subseteq U \times O \times T$, represented by a set of triples defined as:

$$\mathcal{P} = \{\langle u, o, t\rangle | \text{ user } u \text{ assigned tag } t \text{ to object } o\},$$

and

(2) for each object $o \in O$, a set of textual features (other than tags) $F_o = \{F_o^1, F_o^2, ..., F_o^n\}$, where each element $F_o^i$ is the set of terms in textual feature $i$ associated with object $o$.

(3) for each object $o \in O$, the set of associated categories (or latent topics) $Z_o$.

Let $I_o$ be the set of tags previously assigned to the target object $o$, and $I_{o,u}$ the set of tags assigned to the target object $o$ by the target user $u$, that is,

$$I_o = \{t | \exists u \in U \text{ such that } \langle u, o, t\rangle \in \mathcal{P}\}$$

$$I_{o,u} = \{t | \langle u, o, t\rangle \in \mathcal{P}\}$$

Thus, we define two tag recommendation tasks:

**Object-Centered Tag Recommendation** Given a set of input tags $I_o$, a set of textual features $F_o$, associated with the target object $o$, and the folksonomy $\mathcal{P}$, generate a list of candidates $C_o$ ($C_o \cap I_o = \emptyset$), sorted according to their joint

relevance (to object $o$), novelty and diversity objectives, and recommend the $k$ candidates in the top positions of $C_o$.[6]

**Personalized Tag Recommendation** Given a set of input tags $I_o$, a set of textual features $F_o$, associated with the target object $o$, and the folksonomy $\mathcal{P}$, generate a list of candidates $C_{o,u}$ ($C_{o,u} \cap I_o = \emptyset$) sorted according to their joint relevance (to both user $u$ and object $o$), novelty and diversity aspects, recommending the $k$ candidates in the top positions of $C_{o,u}$.

More specifically, the relevance aspect is defined as a function of the number of top-recommended tags that are indeed related to the target object $o$ (and target user $u$ for personalized recommendation). Novelty, in turn, is defined as the average specificity (which is an inverse function of popularity) of the top-recommended tags. Finally, in order to measure diversity, we have, for each tag $c \in C_o$ (or $C_{o,u}$)), and each category or latent topic $z$ of the application, the estimated topic proportion $\Pr(z|c)$. The diversity is defined as a function of the number of categories/topics of the target object $o$ that are covered by the top-recommended tags, that is, has non-negligible values for $\Pr(z_o|c)$, for each topic/category $z_o \in Z_o$. The details about the definitions of these metrics will be provided in the following chapters.

We note that, for the object-centered recommendation task, the same tags are provided regardless of the target user. The primary goal of this kind of recommendation is improving the quality of the tags in these objects, thus, improving the effectiveness of services, such as searching, indexing and classification, that use them as data source. On the other hand, the personalized tag recommendation takes the target user into account: the goal is to suggest relevant tags for the object that match the interests, profile and background of the target user. Personalized tag recommenders might provide different answers to different users (or users with different profiles). One important service that can benefit from personalized tag recommendations is personal content organization. However, we argue that other services that rely on good descriptions of the object's content, such as content recommendation and search, might also benefit from personalized tag recommendations.

One observation that supports our argument is that different users may use very different tags to describe the same object, depending on their backgrounds and interests, and how they perceive the object's content. Moreover, objects shared on Web

[6] Note that we refer to the *task* of recommending tags for an object aiming at improving the quality of its tags (but not necessarily matching the interests of any particular user) as object-centered tag recommendation, even though some of the attributes exploited by the methods (see the description of all metrics in Chapter 4), such as the tag co-occurrence metrics, are related to other tags of the object, and, in a sense, could be considered tag related attributes.

2.0 applications are often multifaceted, being related to various topics, and different users may relate to such topics differently. Thus, in applications where multiple users can assign tags to the same object, such as the popular Last.FM, a personalized tag recommender is not only useful for the individual user (e.g., for content organization) but also in a collective sense, as jointly the tags recommended to different users may provide a more complete description of the object, which indirectly helps search and recommendation services. In other words, a set of personalized tags for the same object produced for different users, with different backgrounds and interests, contribute to covering multiple facets or interpretations of the same object, thus helping with the semantic gap.

Table 3.2 summarizes the notation we use to describe the two tag recommendation tasks defined above. Having motivated the tag recommendation tasks addressed here, we now formally define our target scenarios. For both recommendation tasks, our main focus relies on cases in which there are some available tags in the target object (i.e., $I_o \neq \emptyset$) and we want to recommend new (different) tags to it. We note, however, that all of our methods are also able to recommend relevant tags to an object with no initial set of tags by exploiting other textual features and metrics of relevance. Nonetheless, we will exploit specific solutions for this scenario in Section 5.3.2.

Many tag recommendation strategies, and in particular the ones proposed here, exploit co-occurrence patterns by mining relations among tags assigned to the same object (or additionally by the same user) in an object collection. The process of learning such patterns is defined as follows.

For object-centered tag recommendation, following the methodology proposed by Menezes et al. [2010], we define a training set $\mathcal{D} = \{\langle I_d, F_d \rangle\}$, where $I_d$ ($I_d \neq \emptyset$) contains all tags assigned to object $d$, and $F_d$ contains the term sets of the other textual features associated with $d$. There is also a test set $\mathcal{O}$, which is a collection of tuples $\{\langle I_o, F_o, Y_o \rangle\}$, where both $I_o$ and $Y_o$ are sets of tags associated with object $o$. Tags in $I_o$ are known and given as input to the recommender. On the other hand, tags in $Y_o$ may be assumed unknown and to be taken as the relevant recommendations to the target object $o$ (i.e., the *expected answer*). Splitting the tags of each test object into these two subsets facilitates an automatic assessment of the recommendations, as performed by Garg and Weber [2008]; Guan et al. [2009]; Heymann et al. [2008]; Lipczak and Milios [2011]; Menezes et al. [2010]; Rendle and Schmidt-Thie [2010] and further discussed in Chapter 6. In case of a manual evaluation by volunteers, $I_o$ consists of all tags associated with $o$ and $Y_o$ are the tags assigned to $o$ by the volunteers. Similarly, there might also be a validation set $\mathcal{V}$ used for tuning parameters and "learning" recommendation functions (see Chapter 6). Thus, each object $v$ in $\mathcal{V}$ also presents input tags ($I_v$) and

expected answer ($Y_v$).

For personalized tag recommendation, we exploit two different kinds of tag co-occurrences: (1) between tags assigned to the same object by various users (as we do in the object-centered recommendation task) and (2) between tags assigned by the same user to the same object. Thus, there are two variants of the training set for personalized recommendation: (1) $\mathcal{D} = \{\langle I_d, F_d\rangle\}$, where $I_d$ contains all tags assigned to object $d$ (by any user), and (2) $\mathcal{D}' = \{\langle I_{d,u_d}, F_d\rangle\}$, where $I_{d,u_d}$ contains all tags assigned to an object $d$ by each user $u_d \in U_d$, where $U_d$ is the set of users who assigned at least one tag to object $d$. In both cases, $F_d$ contains the term sets of the other textual features associated with $d$, as defined above. The elements of the test object collection $\mathcal{O}$ are tuples $\langle I_o, F_o, Y_{o,u_o}\rangle$, where $I_o$ is a set of input tags, assigned by any user, including the target user, and $Y_{o,u_o}$ (expected answer) is a set of tags assigned by each user $u_o$ who assigned tags to object $o$. Similarly to the object-centered tag recommendation, in case of a manual evaluation, $I_o$ consists of all tags assigned to the object $o$, while $Y_{o,u_o}$ are tags assigned to $o$ by each volunteer $u_o$. Similarly, each element of validation set $\mathcal{V}$ also contains input tags ($I_v$) and expected answer ($Y_{v,u_v}$).

**Table 3.2.** Tag Recommendation: Problem Definition

| | **Tag Recommendation** | |
|---|---|---|
| | **Object-Centered** | **Personalized** |
| **Input** | $I_o$: set of tags previoulsy assigned to object $o$ | |
| | $F_o$: Set of textual features (other than tags) associated with object $o$ | |
| | $\mathcal{P}$: folksonomy (history of tag assignments) | |
| **Output** | $C_o$, $C_o \cap I_o = \emptyset$ sorted by relevance (to $o$), novelty and diversity | $C_{o,u}$, $C_{o,u} \cap I_o = \emptyset$ sorted by relevance (to $u$ and $o$), novelty and diversity |

## 3.5 Summary

In this chapter, we contextualized the tag recommendation problem, defining its possible targets, data sources and objectives. We defined the different relevance, novelty and diversity concepts. Finally, we formally stated the problem, which can be divided into two sub-tasks: the object-centered and personalized tag recommendation problems, in two scenarios, with and without cold start. In the next chapter, we describe the various relevance, novelty and diversity attributes that our proposed methods exploit.

# Chapter 4

# Tag Quality Attributes

In this chapter, we introduce the tag quality attributes we use to estimate relevance, novelty, diversity and personalization aspects of a candidate tag. Some of the relevance and novelty attributes presented in Sections 4.1 and 4.2, respectively, have been previously proposed [Belém et al., 2011; Lipczak and Milios, 2011; Vargas and Castells, 2011]. The group of syntactic relevance attributes, as well as all three diversity metrics introduced in Section 4.3 are novel contributions of this work in the tag recommendation domain.

## 4.1 Relevance Attributes

The tag relevance attributes that have been proposed in previous work are introduced in Sections 4.1.1-4.1.5. They can be grouped into the following five categories, based on the aspect they try to capture regarding the tag recommendation task:

- *Tag Co-occurrence Attributes* (Section 4.1.1): a key aspect in tag recommendation systems that estimates how relevant a candidate tag $c$ is given a set of input tags that often co-occur with $c$ in the dataset.

- *Descriptive Power Attributes* (Section 4.1.2): estimate how accurately a candidate describes the object's content, which is important for information services that exploit object's semantics.

- *Discriminative Power Attributes* (Section 4.1.3): estimate the capability of a candidate to distinguish the target object from others, which is important for tasks such as separating the objects into semantic classes or into levels of relevance regarding a query.

- *Term Predictability* (Section 4.1.4): indicates the likelihood that a candidate will be predicted as a tag.

- *User Interest* (Section 4.1.5): used for personalization, these attributes estimate the interest of a target user in certain tags.

- *Syntactic Attributes*: (Section 4.1.6): estimate the relevance of candidate tags based on the syntactic structure of the associated text.

### 4.1.1 Tag Co-occurrence

Co-occurrence based tag recommendation approaches usually exploit association rules, that is, implications of type $X \rightarrow y$, where the antecedent $X$ is a set of tags and the consequent $y$ is a candidate tag for recommendation. The importance of an association rule is estimated based on **support** ($\sigma$), which is the number of co-occurrences of $X$ and $y$ in the training set, and **confidence** ($\theta$), which is the conditional probability that $y$ is assigned as a tag to an element $d \in \mathcal{D}$ given that all tags in $X$ are also associated with $d$. As the number of rules mined from the training set $\mathcal{D}$ can be very large and some of them may not be useful for recommendation, minimum support and confidence thresholds ($\sigma_{min}$ and $\theta_{min}$, respectively) are used as lower bounds to select only the most frequent and/or reliable rules. This selection can improve both effectiveness and efficiency of the recommender.

At recommendation time, we select the rules whose antecedents are included in the previously assigned set of tags $I_o$. For each term $c$ appearing as consequent of any of the selected rules, we estimate its relevance as a tag for the object (and for the user in the personalized case), given the initial tag set $I_o$, as the sum of the confidences of all rules containing $c$, i.e.:

$$Sum(c, I_o, \ell) = \sum_{X \subseteq I_o} \theta(X \rightarrow c), \quad (X \rightarrow c) \in \mathcal{R}, |X| \leq \ell \tag{4.1}$$

where $\mathcal{R}$ is a set of association rules computed offline over the training set $\mathcal{D}$, given thresholds $\sigma_{min}$ and $\theta_{min}$, and $\ell$ is the size limit for the association rules' antecedents. $Sum$ was proposed by Sigurbjörnsson and Zwol [2008], which also proposed several other attributes related to tag co-occurrences, including $Vote$ and $Vote^+$, which are also considered here. $Vote$ estimates the relevance of a candidate tag $c$ by the number of association rules whose antecedents are tags in $I_o$ and whose consequent is the

candidate $c$. That is:

$$Vote(c, I_o) = \sum_{x \in I_o} j, \text{ where } j = \begin{cases} 1, & \text{if}(x \rightarrow c) \in \mathcal{R} \\ 0, & otherwise \end{cases} \tag{4.2}$$

$Vote^+$ is built from $Vote$ as follows:

$$Vote^+(c, I_o, k_x, k_c, k_r) = \sum_{x \in I_o} j \times Stab(x, k_x) \times Stab(c, k_c) \times Rank(c, x, k_r),$$
$$\text{where } j = \begin{cases} 1, & if \quad x \rightarrow c \in \mathcal{R} \\ 0, & otherwise \end{cases} \tag{4.3}$$

and $k_x$, $k_c$ and $k_r$ are tuning parameters. $Rank(c, x, k_r)$ is equal to $k_r/(k_r + p(c, x))$, where $p(c, x)$ is the position of $c$ in the ranking of candidates according to the confidence of the corresponding association rule (whose antecedent is $x$). This factor is employed to make confidence values decay smoother. $Stab$ is used to reduce the relative importance of terms that occur either too often or very rarely in the training set, and thus may represent poor recommendations. This attribute, defined in Section 4.1.3, is used, as part of $Vote^+$, to weight the confidence values of the tags in the antecedent and in the consequent of the association rules. A similar extension of $Sum$, called $Sum^+$, is also presented by Sigurbjörnsson and Zwol [2008], being reported as the attribute that produces the best tag recommendations out of all attributes proposed in that study. It is defined as:

$$Sum^+(c, I_o, k_x, k_c, k_r) = \sum_{x \in I_o} \theta(x \rightarrow c) \times Stab(x, k_x) \times Stab(c, k_c) \times Rank(c, x, k_r), \tag{4.4}$$

Regarding the co-occurrence attributes for the specific task of personalized tag recommendation, we distinguish two types of co-occurrence patterns: (1) between all tags assigned to an object by different users, which has been previously exploited in the literature [Garg and Weber, 2008; Lipczak et al., 2009; Sigurbjörnsson and Zwol, 2008] and is here adopted for object-centered recommendation as well, and (2) between all tags assigned by the same user to an object, which we propose in this thesis. While the first strategy benefits from a larger amount of data, the second strategy may generate less noise. As we will show in Chapter 7, these two strategies provide quite different results, and the best strategy depends on the complexity of the exploited association rules, given by parameter $\ell$, and can also be influenced by some characteristics of the dataset. For all tag co-occurrence attributes, we use a subscript $u$ to indicate that the

second type of co-occurrence is used. When there is no such subscript, we refer to the first strategy to generate co-occurrence patterns. For example, $Sum_u(c, I_o, \ell)$ indicates that the set of rules $\mathcal{R}$ exploited to compute $Sum$ was generated from sets of tags assigned by the same user to an object (that is, training set $\mathcal{D}'$, defined in Section 3.4). $Sum(c, I_o, \ell)$ refers to the original attribute, which exploits co-occurrences between all tags assigned to an object by different users.

### 4.1.2 Descriptive Power

We exploit four attributes that try to capture, to some extent, the *descriptive power* of a candidate $c$. In [Belém et al., 2011], we exploited them for object-centered tag recommendation, while here we also apply them to the personalized tag recommendation task. This is a novel aspect of this work.

We start by defining the *Term Spread* of a candidate $c$ in an object $o$, $TS(c, o)$, as the number of textual features (except tags, in the present context)[1] of $o$ that contain $c$ [Figueiredo et al., 2012]:

$$TS(c, o) = \sum_{F_o^i \in F_o} j, \text{ where } j = \begin{cases} 1 & \text{if } c \in F_o^i \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

The assumption behind $TS(c, o)$ is that the larger the number of features of $o$ containing $c$, the more related $c$ is to $o$'s content. For example, if the term "Sting" appears in all features of a video, there is a high chance that the video is related to the famous singer. The maximum $TS$ is given by the number of textual features, other than tags, considered. As we exploit title and description, $TS \leq 2$.

The *Term Frequency* of $c$ in object $o$, $TF(c, o)$, is:

$$TF(c, o) = \sum_{F_o^i \in F_o} tf(c, F_o^i), \tag{4.6}$$

where $tf(c, F_o^i)$ is the number of occurrences of $c$ in feature $F_o^i$ of object $o$. Thus, $TF$ considers all textual features of $o$ as a single bag of words, counting all occurrences of $c$ in it. In contrast, $TS$ considers the structure of an object, composed by textual features, which are well-defined blocks of text, counting the number of blocks containing $c$.

Although both $TS$ and $TF$ try to capture how accurately a term describes an

[1] We do not include tags to compute any of the descriptive power attributes, as it does not make sense to use tags previously assigned to the target object as candidates for recommendation.

object's content, neither of them considers that different features may present, in general, different descriptive capacities. For example, the title may describe an object's content more accurately than other textual features [Figueiredo et al., 2012]. Thus, we proposed in [Belém et al., 2011] two other attributes, built on $TF$ and $TS$, that weight a term based on the average descriptive powers of the textual features in which it appears.

The average descriptive power of a textual feature $F^i$ is assessed by the Average Feature Spread (AFS) heuristic [Figueiredo et al., 2012]. Let the *Feature Instance Spread* of a feature $F_o^i$ associated with object $o$, $FIS(F_o^i)$, be the average $TS$ over all terms in $F_o^i$. We define $AFS(F^i)$ as the average $FIS(F_o^i)$ over all instances of $F^i$ associated with objects in the training set $\mathcal{D}$. We then define weighted $TS$ and $TF$ as:

$$wTS(c,o) = \sum_{F_o^i \in F_o} j, \text{ where } j = \begin{cases} AFS(F^i) & \text{if } c \in F_o^i \\ 0 & \text{otherwise} \end{cases} \tag{4.7}$$

$$wTF(c,o) = \sum_{F_o^i \in F_o} tf(c, F_o^i) \times AFS(F^i) \tag{4.8}$$

### 4.1.3 Discriminative Power

One may argue that recommending more infrequent terms (provided that they are not too rare) may be desirable, since they may better *discriminate* objects into different categories, topics, or levels of relevance, particularly considering that several services (e.g., classification, searching) often perform IR on multimedia content by using the associated tags as data sources. This aspect can be heuristically captured by the *Inverse Feature Frequency* ($IFF$) attribute [Figueiredo et al., 2012], an adaptation of the traditional *Inverse Document Frequency* ($IDF$) that considers the term frequency in a specific textual feature (in our case, tags). Given the number of elements in the training set $|\mathcal{D}|$, the $IFF$ of a candidate $c$ is defined as:

$$IFF(c) = \log \frac{|\mathcal{D}| + 1}{f_c^{tag} + 1} \tag{4.9}$$

where $f_c^{tag}$ is the number of elements (objects for object-centered recommendation, or object-user pairs for personalized recommendation) in $\mathcal{D}$ that are tagged with $c$. Note that $c$ may be extracted from other textual features. The value 1 is added to both numerator and denominator to deal with new terms that do not appear as tags in the training data. We note that this attribute may privilege terms from other textual features that do not appear as tags in the training data. Nevertheless, this

attribute will be combined with the other attributes into a function, using learning-to-rank algorithms. Thus, its relative weight can be adjusted.

Along the same lines, one may consider that terms that are very common, such as "video" in a YouTube object collection, are too general and broad, whereas very rare terms may be too specific or may represent noise (e.g., misspellings, neologisms and unknown words). In either case, such terms represent poor recommendations as they have very poor *discriminative power*. Sigurbjörnsson and Zwol [2008] propose the Stability (*Stab*) attribute, which gives more importance to terms with intermediate frequency values:

$$Stab(c, k_s) = \frac{k_s}{k_s + |k_s - \log(f_c^{tag})|} \tag{4.10}$$

where $k_s$ represents the "ideal frequency" of a term and must be adjusted to the data collection. We also use *Stab* to assess the relevance of a candidate tag, but, unlike [Sigurbjörnsson and Zwol, 2008], we apply it not only to tags but also to terms extracted from all textual features $F_o$ associated with target object $o$.

### 4.1.4 Term Predictability

Another important aspect for tag recommendation is term predictability. Heymann et al. [2008] measure this characteristic through the term's *entropy*. The entropy of a candidate $c$ in the tags feature, $H^{tags}(c)$, is defined as:

$$H^{tags}(c) = -\sum_{(c \to i) \in \mathcal{R}} \theta(c \to i) \log \theta(c \to i) \tag{4.11}$$

If a term occurs consistently with certain tags, it is more predictable, thus having lower entropy. Terms that occur indiscriminately with many other tags are less predictable, thus having higher entropy. In other words, $H^{tags}(c)$ measures the concentration of confidence values of all association rules whose antecedent is $c$. If a term is absent in the training set, it receives the maximum entropy value (highest uncertainty about its relevance). Term entropy can be useful particularly for breaking ties, as it is better to recommend more "consistent" or less "confusing" terms. Whereas term entropy was used by Heymann et al. [2008] only to evaluate recommendations, we apply it as an input to the recommendation functions.

Inspired by the method proposed by Lipczak et al. [2009], described in Section 5.1, we propose an attribute called *Predictability* (*Pred*), which measures the probability that a term is used as a tag in an object given that it was used in another textual feature of the same object. Unlike the attribute proposed in [Lipczak et al.,

2009], which computes such co-occurrences separately for each textual feature, $Pred$ is computed by aggregating all textual features of the object. In other words, the *Predictability* of a candidate tag $c$, $Pred(c)$, is defined as:

$$Pred(c) = \frac{f_c^{tag,F}}{f_c^F}, \tag{4.12}$$

where $f_c^{tag,F}$ is the number of objects in the training set in which $c$ appears both as a tag and as a term in *any* other textual feature, and $f_c^F$ is the number of objects in which $c$ is a term associated with any of its textual features (except tags).

### 4.1.5 User Frequency

In order to estimate the relevance of a candidate for a target user and thus provide personalized recommendations, we propose here an attribute called *User Frequency*[2], or $UF$, which is the frequency at which the target user assigns a candidate tag to an object. In other words, given a candidate $c$ and a target user $u$, $UF(c, u)$ is defined as:

$$UF(c, u) = \frac{N_{c,u}}{N_u}, \tag{4.13}$$

where $N_{c,u}$ is the number of times that user $u$ tagged an object with $c$ in the training set $\mathcal{D}$, and $N_u$ is the total number of times user $u$ submitted a tag. Thus, the rationale behind $UF$ is: the more frequently a user $u$ assigns a candidate tag $c$ to other objects in the application, the more relevant $c$ is for $u$. This attribute is computed for all tags used by the target user $u$ in the training set.

### 4.1.6 Syntactic Attributes

In the following, we describe 11 new syntactic attributes. They are based on the conditional probability (estimated with training data) that a word is used as tag given that it was labeled with a given (syntactic) property $\pi$ (e.g., PoS). In order to extract these attributes, we rely on the description's text associated to the target object.

Each sentence of the description can be represented by a *syntactic dependency tree*. Each (whitespace separated) *token* is a node in the tree, and there is an edge between two tokens if one of them is grammatically dependent on the other. Edges can be labeled with the type of syntactic dependency (e.g., object of a preposition, nominal

[2]This attribute is similar to one proposed in [Lipczak et al., 2009]. However, the authors in [Lipczak et al., 2009] make use of the timestamp of the tagging event. As this information is not available in our datasets, we adapted this attribute to consider all tag assignments of user $u$ in the training set.

subject). Additionally, each node can be associated with a part-of-speech (PoS) label. The parent of a node in the tree is called *head* of this token in the sentence. The node with no parents is called *root* of the sentence. In this thesis, we use the natural language processing tool `spaCy`[3] to automatically identify PoS and syntactic dependencies. We focus on the English language, but our work can be adapted for other languages, if a syntactic analysis tool is available for the target language. Figure 4.1 illustrates the syntactic dependency tree of a sentence. Following the root of the sentence (verb "makes"), note the main parts of the sentence, namely, the nominal subject (centered at the proper name "Alliance") and the direct object (centered at the noun "move").

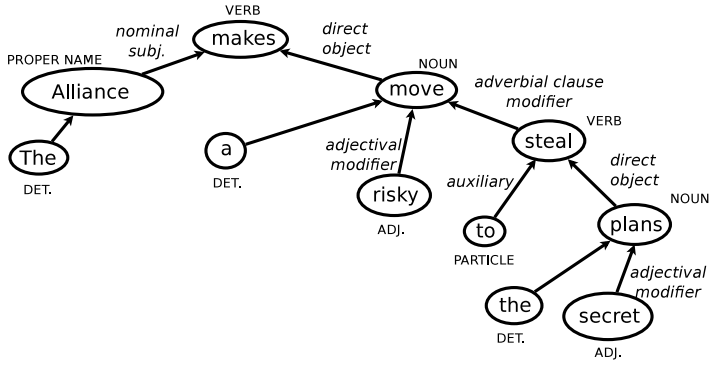



**Figure 4.1.** Syntactic dependency tree of a sentence. PoS labels in capital letters and syntactic functions in italic.

**Table 4.1.** Investigated properties ($\pi$) of candidate tags and words syntactically connected to them.

| Group | Name |
|---|---|
| Candidate tag | Token |
| | PoS |
| | Syntactic function |
| Connected words | Token's head |
| | PoS of the token's head |
| | Syntactic function of the token's head |
| | Root of the sentence |
| | Sequence of tokens between candidate tag and the root of the sentence |
| | Sequence of PoS labels between candidate tag and the root of the sentence |
| | Sequence of syntactic functions between candidate tag and the root of the sentence |
| | Sequence of PoS and syntactic functions between candidate tag and the root of the sentence |

Let $\pi(x, T)$ be a given property associated with a word $x$ in a description $T$. The general formula of our probability-based metrics for a candidate tag $c$ in an object $o$

[3] `https://spacy.io/`

associated with a description text $T_o$ is:

$$P_\pi(c, T_o) = \frac{\sum_{d \in \mathcal{D}} |\{x \in T_d | (x \in \mathcal{I}_d \; and \; \pi(x, T_d) = \pi(c, T_o))\}|}{\sum_{d \in \mathcal{D}} |\{x \in T_d | \pi(x, T_d) = \pi(c, T_o)\}|} \tag{4.14}$$

where $\mathcal{D}$ is the set of training objects, $\mathcal{I}_d$ is the set of tags associated with a training object $d$, and $\pi$ is one the properties listed in Table 4.1. If a word appears multiple times in $T_o$ (note that $T_o$ may be composed by various sentences) with different values for $\pi(c, T_o)$, we choose the maximum value of $P_\pi(c, T_o)$ as the attribute value for the property $\pi$. Take as an example the property $\pi$=PoS. Suppose that a candidate tag $c$ received the PoS label "adverb" in a description $T_o$. Then, its $P_{PoS}(c, T_o)$ value is the ratio between the number of times adverbs extracted from descriptions $T_d$ in the training dataset were used as tags, and the total number of times that any adverb appeared in training object's descriptions.

In order to avoid overinflated estimations of $P_\pi$ values, we disregard properties that occur less than a number *minfreq* of times in $\mathcal{D}$, assigning the average probability value for the corresponding property instead. Take the aforementioned example with the property $\pi$=PoS. Suppose that a word with PoS value "adverb" occurred only once in $\mathcal{D}$, and that it was also used as a tag. This would result in $P_{PoS}$=1 for adverbs. Setting $minfreq > 1$, we take the average probability (over *all other* PoS values that occur more than *minfreq* times), for candidate tags that are adverbs in this example.

The studied properties are divided into two groups in Table 4.1. The first group contains properties related to a single word (candidate tag), while the second group consists of properties related to one or more words connected to the candidate tag in the corresponding syntactic tree.

Among the properties in the first group, $Token$ is the word itself, that is, $Token(x) = x$. The idea of this attribute is that some words have a much higher chance to be used as a tag than others (e.g., words that indicate the object's category, such as "drama" and "comedy" in MovieLens dataset). The idea of the other properties directly related to the candidate tag (PoS and syntactic function) is similar, however they capture the probability of more generic types of tags (nouns, adjectives, direct objects, etc).

As we will see in Chapter 7, looking solely at the syntactic properties of a single word (the candidate tag) may not be enough to discriminate "good" from "bad" candidates. Thus, we also include properties of words connected to the candidate tag in the syntactic dependency tree (second block of Table 4.1). Among them, recall that the "head" of a word $w$ is the parent of $w$ in the syntactic dependency tree, as we described in Section 3. We also include an attribute related to the root of the sentence

that contains the candidate tag.

Finally, the other four properties in the second block of Table 4.1 (whose names start with "Sequence") are related to the whole sequence of words in the path that connects a candidate tag to the root of the corresponding sentence. We include not only the tokens that form the path as properties, but also their PoS and syntactic functions. Note that some of our properties may present overlapping parts of the syntactic tree. However, as we will see in Section 7.1.3, we select the most important attributes by performing a feature importance analysis.

Note that an alternative set of attributes could be defined by considering the various syntactic properties as categorical attributes, and creating a binary attribute for each possible value of each syntactic property (for example, an attribute "adjective" would be valued 1 if the given candidate tag is an adjective, and 0, otherwise). This is equivalent to what Hulth [2003] performed for PoS related attributes. However, due to the large number of possible values (not only for PoS but for all properties we are considering), we opted for a smaller set of attributes that summarizes each desired property $\pi$ by using its $P_\pi$ value.

## 4.2 Novelty Attribute

Vargas and Castells [2011] proposed to estimate the novelty of an item in a list of recommendations as the probability that it has not been previously observed. Thus, the lower the popularity of an item, the more novel it is. Bringing this definition to the context of tag recommendation, we note that the $IFF$ attribute (Equation (4.9) in Section 4.1.3) does capture exactly the aspect proposed by Vargas and Castells [2011], as it favors candidates that occur less frequently in the training set. Thus, although we [Belém et al., 2011] previously employed $IFF$ to recommend tags that can better discriminate an object from the others, an aspect that is related to the relevance of the tag to the target object, we use the same attribute to increase the novelty of the recommendations, that is, to recommend possibly relevant tags that, because they occur very rarely in the training set, they would hardly be recommended by traditional methods.

## 4.3 Diversity Attributes

Recall from Section 3.3, that diversity can be addressed in two perspectives, implicit and explicit. Thus, our diversity attributes are also divided into these two

perspectives. Regarding the implicit notion of diversity [Vargas and Castells, 2011], we estimate the diversity of a candidate tag $c$ with respect to a list $C_o$ of candidates for recommendation for target object $o$ as the average semantic distance between $c$ and each other candidate tag in $C_o$. Thus, we define the *Average Distance to other Candidates* (*ADC*) as:

$$ADC(c, C_o) = \frac{1}{|C_o|} \sum_{t \in C_o, t \neq c} dist(c, t) \tag{4.15}$$

where $dist(c, t)$ measures the dissimilarity between candidate tags $c$ and $t$. There are various ways of estimating the dissimilarity between two terms. In this thesis, we estimate the dissimilarity between terms $t_1$ and $t_2$ by the relative difference between the sets of objects $O_1$ and $O_2$ in which they appear as tag, i.e., $dist(t_1, t_2) = \frac{|O_1 - O_2|}{|O_1 \cup O_2|}$. If both sets are empty, we set $dist(t_1, t_2)$ equal to the maximum value, i.e., 1. Note that by measuring the dissimilarity between two terms in this way, we are basically using the set of objects in which each term appears as tag to represent its possible meanings. Thus, terms that appear in very different sets of objects most probably have very different meanings.

Considering that users often associate tags to web content with organization and categorization purposes [Gupta et al., 2010], tags that are more related to the topics (e.g., categories) of the target object are good candidates for recommendation. The explicit diversity attributes we propose in this thesis, *topic coverage* and *topic similarity*, exploit this idea. Before introducing them, we first estimate the probability that a tag $t$ is associated with a topic $z$, $\Pr(z|t)$, as $\Pr(z|t) = f(t, z)/f(t)$, where $f(t, z)$ is the number of objects in which $z$ appears as a topic and $t$ appears as a tag, and $f(t)$ is the number of objects containing tag $t$, both in the training set $\mathcal{D}$. We also estimate the probability that a topic $z$ is associated with an object $o$, $\Pr(z|o)$, as either $1/n_o$, where $n_o$ is the number of categories associated with object $o$, when categories are available, or alternatively as the result produced by the LDA algorithm (described in Section 6.2.1), when such information is not available.

Let $Z_o$ be the set of topics associated with the object $o$. We define the *topic coverage* of a candidate tag $c$ for an object $o$, $TC(c, o)$, as the fraction of topics of $o$ covered by $c$, that is:

$$TC(c, o) = \frac{1}{|Z_o|} \sum_{z \in Z_o} J(c, z) \text{ where } J(c, z) = \begin{cases} 1, & \text{if } \Pr(z|c) > \Pr(z) \\ 0, & \text{otherwise} \end{cases} \tag{4.16}$$

where $|Z_o|$ is the number of topics associated with object $o$, and $\Pr(z)$ is the prior probability of topic $z$, that is, the fraction of all objects in the training set $\mathcal{D}$ that have topic $z$ associated with them. We consider that candidate $c$ "covers" a topic $z$ ($c$ is highly related to $z$) if the probability of topic $z$ given $c$ is higher than the (prior) probability of $z$.

Multiple topics may be associated with a given object or tag, while the strength of the semantic association between them may vary across different topics. Yet, the topic coverage attribute does not capture such variability. Thus, we propose the topic similarity attribute, which measures the cosine similarity between the distribution of topics of the candidate tag and the distribution of topics of the target object, and thus, takes the strength of the semantic association between topic and object (or tag) into account. We estimate the strength of the association between topic $z$ and object $o$ by the probability of the topic given the object $\Pr(z|o)$. Similarly, the strength of the association between $z$ and candidate tag $c$ is estimated by $\Pr(z|c)$. The *topic similarity* of a candidate tag $c$ with relation to a target object $o$, $TSim(c, o)$, is then defined as:

$$TSim(c,o) = \frac{\sum_{z \in Z_o} \Pr(z|o) \times \Pr(z|c)}{\sqrt{\sum_{z \in Z_o} (\Pr(z|o))^2} \times \sqrt{\sum_{z \in Z_o} (\Pr(z|c))^2}} \tag{4.17}$$

## 4.4 Summary

In this chapter, we presented the attributes that capture relevance, novelty and diversity aspects of tag recommendation. A subset of our relevance attributes capture the personalization aspect. The other relevance attributes capture the descriptive and discriminative power of candidate tags, as well as their predictability and degree of relationship with other tags (tag co-occurrences). The novelty attribute is based on the popularity of a tag, while our diversity attributes can be implicit (exploiting dissimilarity among tags) or explicit (exploiting the topics of the target object). The diversity and syntactic attributes are novel contributions of this work.

The attributes presented in this chapter are exploited by our tag recommendation methods, which in turn are presented in the next chapter.

# Chapter 5

# Tag Recommendation Methods

In this chapter, we describe the tag recommendation methods analyzed in this work. In Sections 5.1 and 5.2, we describe with more detail the state-of-the-art baselines for the object-centered and personalized tag recommendation problems, respectively. Section 5.3 presents our new approaches for the object-centered tag recommendation problem, while Section 5.4 discuss the extension of these methods to address personalization. Finally, Section 5.5 presents our new methods that also consider the other aspects of the problem (novelty and diversity).

## 5.1 State-of-the-art Object-Centered Baselines

The following sections briefly describe the baseline methods used for evaluating our new object-centered tag recommendation methods. We first present general methods (heuristics and L2R-based approaches) that have been mostly evaluated in non cold start scenarios (Sections 5.1.1 and 5.1.2). Later, in Section 5.1.3, we describe baselines for the specific cold start scenario.

### 5.1.1 Unsupervised Heuristics

Our first baseline is $Sum^+$, the best function proposed in [Sigurbjörnsson and Zwol, 2008], which exploits both tag co-occurrences and attributes of tag relevance. We defined $Sum^+$ in Section 4.1.1, Eq. (4.4). $Sum^+$ extends the $Sum$ attribute (Eq. 4.1) similarly to how $Vote^+$ extends $Vote$, that is, by weighting the confidence values by the *Stability* of the terms in the antecedent and consequent of the corresponding association rules. For this method, we use the *Apriori* algorithm [Agrawal and Srikant, 1994] to generate the association rules.

$Sum^+$, as most co-occurrence based strategies [Garg and Weber, 2008; Sigurbjörnsson and Zwol, 2008], restricts the size of the association rules to only one tag in the antecedent (i.e., $\ell$=1) due to efficiency issues. In contrast, $LATRE$ - Lazy Associative Tag Recommender [Menezes et al., 2010], our second baseline, is able to efficiently generate larger association rules by doing it *on demand*. This is in contrast to other strategies (e.g., *Apriori*), which compute all rules from the training set beforehand (i.e., offline), possibly including rules that might not be useful when recommending for objects in the test set. $LATRE$ ranks each candidate $c$ by the sum of the confidences of all rules containing $c$. That is, it uses the $Sum$ metric (Eq. 4.1) with $\ell \geq 1$, thus exploiting solely co-occurrence patterns.

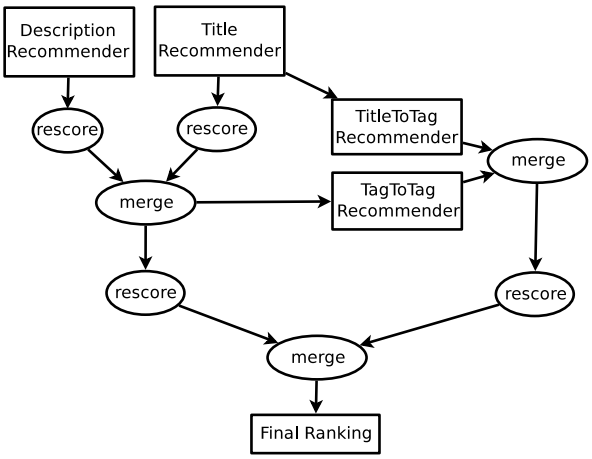



**Figure 5.1.** CTTR algorithm [Lipczak et al., 2009].

Our third baseline is called *Co-occurrence and Text based Tag Recommender* (CTTR). It exploits terms extracted from other textual features, but does not consider tags previously assigned to the *target object*. CTTR is an adaptation of the winner of the ECML Discovery Challenge 2009 [Lipczak et al., 2009], which, in addition to the two aforementioned aspects, also takes the user's tag assignment history into account. We here do not include such user statistics in CTTR, because they include the time instants when the tag assignments were done by each user, and this information is not available in our datasets. Thus, we use CTTR as a baseline for object-centered recommendation only[1]. Like our methods, CTTR also exploits multiple textual features. Thus, the comparison of our methods against this baseline allows us to assess the benefits of applying our relevance attributes to such terms and to exploit co-occurrence of previously assigned tags.

[1]The lack of the user historical information required by the method prevented us from using it as baseline for personalized tag recommendation.

The basic structure of CTTR is depicted in Figure 5.1. As described below, CTTR distinguishes two types of co-occurrences: (1) between tags, in which the antecedents are tags in the objects of the training set, and (2) between terms in the title of an object and its tags, in which the antecedents are terms in the titles of such objects. At recommendation time, the sets of rules related to the extracted terms are combined using corresponding scores. As a final step, the scores obtained from the association rules and from the title and description of the target objects are rescored once again, and merged, resulting in the final ranking.

The first step is the extraction of potential candidates from other textual features associated with the target object, namely its title and description. Each term extracted from the title (or description) is scored according to its usage in previous tagging posts (training set). The score $p_x^i$ is the ratio of the number of times the term $x$ was used simultaneously in $F_d^i$ and as a tag to the total number of objects in which $x$ is associated with the textual feature $F_d^i$, where $i \in \{title, description\}$[2].

Next, the candidate sets generated by title and description are merged. As observed by Lipczak et al. [2009], titles tend to provide more precise recommendations than other textual features, which should be reflected in the merging step. Towards that goal, the authors propose to use a *leading precision rescorer* for weighting the different candidate sources (textual features). This rescorer sets the average precision at the first position of the ranking, $avgP@1$ (calculated over training data) as a new score for the top candidate, and modifies the scores $s_i$ of the following terms proportionally. Let $s_1$ be the old score of the top candidate. The new score $s'_i$ of the $i^{th}$ candidate tag is given by:

$$s'_i = \frac{avgP@1 \times s_i}{s_1} \tag{5.1}$$

After re-scoring, the new scores should be merged in a probabilistic sum. Let $S_t = \{s_t^1, s_t^2, ..., s_t^n\}$ be a set of different scores for candidate tag $t$. The merging function is given by:

$$merge(S_t) = 1 - \prod_{s_t^i \in S_t} (1 - s_t^i) \tag{5.2}$$

The terms extracted in the first step are then expanded through association rules. However, unlike [Sigurbjörnsson and Zwol, 2008], CTTR does not consider any tag that had been previously assigned to the target object. Towards the purpose of generating term candidates by co-occurrences with terms in the target object, Lipczak et al. [2009];

[2]The score $p_x^i$ inspired us to build the *Pred* metric, defined in Chapter 4.

Lipczak and Milios [2011] distinguish two types of co-occurrence relationships: (1) between tags ($\mathcal{R}_{TagToTag}$), and (2) between terms in the title of an object and its tags ($\mathcal{R}_{TitleToTag}$). In other words, while the antecedents of $\mathcal{R}_{TagToTag}$ rules are tags of the training set, the antecedents of $\mathcal{R}_{TitleToTag}$ are terms in the titles of objects in it.

In the online recommendation step, the rule sets related to the extracted terms are combined. Title terms are used as antecedent in the following equation to find title-related tags:

$$S_{TitleToTag}(t,o) = 1 - \prod_{x \in F_o^{title}} (1 - \theta(x \rightarrow t) \times p_x^{title}), \tag{5.3}$$

where $(x \rightarrow t) \in \mathcal{R}_{TitleToTag}$, and $p_x^{title}$ is the usage of the title term $x$ as a tag, as defined above. In the same way, the resulting terms of the title-description merge are taken as antecedent in:

$$S_{TagToTag}(t,o) = 1 - \prod_{x \in \bigcup_i F_o^i \cup I_o} (1 - \theta(x \rightarrow t) \times s_x), \tag{5.4}$$

where $(x \rightarrow t) \in \mathcal{R}_{TagToTag}$, and $s_x = merge(\{p_x^{title}, p_x^{description}\})$ is the score of $x$ achieved after the aforementioned title-description merging step. We note that $s_x$ may be interpreted as a relevance metric since it is similar to $TS$, that is, it captures the importance of a term in the textual features of an object.

At the final step, scores obtained from association rules ($S_{TitleToTag}$ and $S_{TagToTag}$) and from the title and description of the target object are re-scored and merged (with Equations (5.1) and (5.2)), resulting in the final ranking.

In [Belém et al., 2011], we proposed heuristics for object-centered tag recommendation which extend the $Sum^+$ and $LATRE$ baselines to also include one of the four attributes of descriptive power, i.e., $TF$, $TS$, $wTF$ or $wTS$ (Section 4.1.2). We thus proposed eight new ranking functions composed by a weighted linear combination of the output of $Sum^+$ (or $LATRE$) and one of the four attributes. Let $DP$ be the selected *descriptive power* metric (i.e., $TS$, $TF$, $wTS$ or $wTF$), and $c$ be a candidate tag for a target object $o$ associated with a set of previously assigned tags $I_o$. Our previously proposed heuristics have the following general structures:

$$Sum^+DP(c,o,k_x,k_c,k_r,\alpha) = \alpha Sum^+(c,I_o,k_x,k_c,k_r) + (1-\alpha)DP(c,o) \tag{5.5}$$

$$LATRE + DP(c,o,\ell,\alpha) = \alpha Sum(c,I_o,\ell) + (1-\alpha)DP(c,o) \tag{5.6}$$

Parameter $\alpha$ ($0 \leq \alpha \leq 1$) is used as a weighting factor. Note that $Sum^+$ and $Sum$ are computed only over candidates generated from the association rules, whereas $DP$ is computed for terms extracted from other textual features of target object $o$.

### 5.1.2 L2R-Based Object-Centered Tag Recommendation Methods

The basic idea of the L2R approaches for tag recommendation is to use such algorithms to learn a *good ranking function* based on a list $L_{attr}$ of attributes of tag relevance. Three of these approaches, namely RankSVM (Section 5.1.2.1), Genetic Programming (GP) framework (Section 5.1.2.2), and RankBoost (Section 5.1.2.3) were previously exploited in tag recommendation [Belém et al., 2011; Cao et al., 2009; Wu et al., 2009]. The evaluation of the other five techniques, described in Section 5.3.1, is a novel contribution of this work. We chose these L2R methods since they represent different learning paradigms that have been successfully applied to other IR tasks such as classification, search/ranking and image retrieval [Faria et al., 2010; Gomes et al., 2013; Yeh et al., 2007].

**Table 5.1.** List of tag quality attributes exploited by L2R-based methods (non cold start scenario).

| Attribute Category | Object-Centered Recommendation | | Personalized Recommendation | |
|---|---|---|---|---|
| | **Attribute** | **Reference** | **Attribute** | **Reference** |
| Tag Co-occurrence | $Sum(\ell = 1)$ | Eq. (4.1) | $Sum_u(\ell = 1)$ | Eq. (4.1) |
| | $Sum(\ell = 3)$ | Eq. (4.1) | $Sum(\ell = 3)$* | Eq. (4.1) |
| | $Sum^+$ | Eq. (4.4) | $Sum_u^+$ | Eq. (4.4) |
| | $Vote$ | Eq. (4.2) | $Vote_u$ | Eq. (4.2) |
| | $Vote^+$ | Eq. (4.3) | $Vote_u^+$ | Eq. (4.3) |
| Descriptive Power | $TS$ | Eq. (4.5) | $TS$ | Eq. (4.5) |
| | $wTS$ | Eq. (4.7) | $wTS$ | Eq. (4.7) |
| | $TF$ | Eq. (4.6) | $TF$ | Eq. (4.6) |
| | $wTF$ | Eq. (4.8) | $wTF$ | Eq. (4.8) |
| Discriminative Power | $IFF$ | Eq. (4.9) | $IFF$ | Eq. (4.9) |
| | $Stab$ | Eq. (4.10) | $Stab$ | Eq. (4.10) |
| Predictability | $Entropy$ | Eq. (4.11) | $Entropy$ | Eq. (4.11) |
| | $Pred$ | Eq. (4.12) | $Pred$ | Eq. (4.12) |
| User Interests | - | - | $UF$ | Eq. (4.13) |

*For the MovieLens dataset, we replaced $Sum(\ell = 3)$ by $Sum_u(\ell = 3)$, since it produced significantly better results for this dataset.

We start by focusing on how we apply these techniques to the object-centered tag recommendation task, discussing extensions to address personalization in Section 5.4. Moreover, we discuss further extensions of our methods to address the cold start problem in Section 5.3.2. Although these cold start solutions can be applied to the personalized tag recommendation task, we focused on the object-centered task, since we mostly exploit object-related attributes. For object-centered tag recommendation

(without the cold start scenario), the list of attributes $L_{attr}$ exploited by all three L2R methods, includes: $Sum$, $Vote$, $Vote^{+}$, $IFF$, $Stab$, $TS$, $TF$, $wTS$, $wTF$, $H^{tags}$, $Pred$ and $Sum^{+}$, defined in Eqs. 4.1-4.4 (Section 4.1). In particular, we include $Sum$ with both $\ell$=1 and $\ell$=3, thus generating two attribute values for it.

Moreover, the set of candidate tags $C_o$ for each object $o$ includes all terms generated by $LATRE$ and all terms extracted from other textual features. For each candidate $c \in C_o$, for each object $o$, we compute all attributes in $L_{attr}$ using the training set $\mathcal{D}$ (e.g., for $Stab$, $IFF$) and the textual features associated with $o$. Each candidate $c$ is then represented by a vector of attribute values $M_c \in \mathbb{R}^m$, where $m$ is the number of considered attributes ($m = 13$ for object-centered tag recommendation). We also assign a binary label $r_c$ to each candidate $c$ for each object $v$ in validation set $V$ (part of the training set), indicating whether $c$ is a relevant recommendation for $v$ ($r_c$=1) or not ($r_c$=0), based on the contents of $Y_v$.

For the cold start scenario, as we will see in Section 5.3.2, we have a different set of candidate tags. Similarly to the non cold start scenario, terms in the other textual features of the target object (e.g., title, description) are extracted as candidate tags. However, unlike the non cold start scenario, there are no candidate tags generated by $LATRE$, since we can't exploit co-occurrences with the initial (empty) tag set. Moreover, to compensate the lack of co-occurrences, as performed by Martins et al. [2016], candidate tags generated by the CTTR method, as well as candidates originated from the neighborhood of the target object (i.e., similar objects) are included, as will be discussed in Section 5.3.2.

#### 5.1.2.1 RankSVM Based Strategy

RankSVM is based on the state-of-the-art Support Vector Machine (SVM) classification method [Joachims, 2006]. We use the SVM-rank tool[3] to learn a function $f(M_c)$=$f(W, M_c)$, where $W = < w_1, w_2, ...w_m >$ is a vector of weights associated with the considered attributes (i.e., $W \in \mathbb{R}^m$). $W$ is learned by a maximum-margin optimization method that tries to find a hyperplane, defined by $W$, that best separates the "closest" candidate tags (represented by their attribute vectors in $\mathbb{R}^m$) belonging to two different *levels of relevance* (i.e., relevant and irrelevant) assigned to each object-candidate pair in the training. They are employed to produce pairwise ranking statements (i.e., relevant tags must precede irrelevant ones), which in turn are used as input to the RankSVM learning process. At recommendation time, $f(M_c)$ is used to rank all candidates for target object $o$ according to their relative distances to the

[3] `http://www.cs.cornell.edu/People/tj/svm_light/svm_rank.html`

separating hyperplane. RankSVM has 2 key parameters: the type of kernel function, which indicates the structure of the solution function, and cost $j$, which controls the penalty to classification errors in the training process.

#### 5.1.2.2 GP Based Strategy

Genetic Programming (GP) is a framework inspired by the biological mechanisms of genetic inheritance and evolution of individuals in a population [Banzhaf et al., 1998]. GP implements a global search mechanism by evolving a population of individuals over multiple generations. Each individual, representing a possible solution for the target problem (a tag ranking function), is modeled as a tree composed of terminals (leaves) and operators (inner nodes), related to the target problem. In our case, terminals are constants (uniformly distributed between 0 and 1) and attributes (attributes presented in Section 4.1), while the inner nodes are operators sum, subtraction, multiplication as well as protected division and logarithm (so that they return the default value 0 if their inputs are out of their domains). In each generation, each individual is evaluated by a *fitness* function, defined based on quality attributes related to the problem at hand. Only individuals with the highest fitness values are selected, according to some selection method (we adopt the tournament selection, i.e., selecting the best individual among $k$ randomly chosen individuals), to evolve the population.

An initial randomly generated population is evolved in a number of generations, through *crossover* and *mutation* operations. The crossover operation, performed on two selected individuals with probability $p_c$, is implemented by randomly choosing one node of each tree representing a selected individual and exchanging the subtrees below them. It aims at combining good solutions towards a more promising one. The mutation operation, on the other hand, adds new individuals (solutions) to the population, thus increasing the diversity in it. This is useful, for instance, to avoid being trapped in local optima. With probability $p_m$, the mutation of a selected individual is done by first randomly choosing one node of its tree, and then replacing the subtree rooted at it by a new randomly generated subtree, without exceeding a maximum tree depth $d$. Note that population size $n_p$ is kept fixed through all generations. This process continues until a target fitness value $f^t$ or a maximum number of generations $n_g$ is reached. The individual with the best fitness value, usually part of the last generation, is chosen as the final solution for the problem.

GP is a non-linear method that has been applied to various IR tasks. We were the only to use it for recommending tags [Belém et al., 2011], having obtained competitive (or superior) results over RankSVM. GP directly optimizes a target (fitness)

function (e.g., precision) and allows for easy extensions to include more problem-related attributes (terminals) and to address other aspects of the target problem, as we do in this thesis by adding novelty and diversity to the objective function.

The fitness of an individual in this context represents the quality of the recommendations produced by the corresponding ranking function, which we assessed in terms of the *Normalized Discounted Cumulative Gain* (NDCG) in the top-$k$ terms in the ranking of recommended terms, averaged over all recommendations of the training examples[4].

Let $Y$ be the set of relevant tags for object $o$ ($Y = Y_o$), and $C$ be the sorted set of recommendations produced by the ranking function being evaluated. We define the discounted cumulative gain in the first $k$ recommendations, $DCG@k$, as:

$$DCG@k(C,Y) = \sum_{i=1}^{k} \frac{rel(i)}{\log_2(i+1)}, \tag{5.7}$$

where $rel(i)$ is equal to 1 if the $i^{th}$ candidate returned in $C$ is relevant (i.e, it is in $Y$), and 0 otherwise. From this definition, we can define the normalized discounted cumulative gain in the first $k$ recommendations, $NDCG@k$, as:

$$NDCG@k(C,Y) = \frac{DCG@k(C,Y)}{IdealDCG@k}, \tag{5.8}$$

where $IdealDCG$ is the value obtained for $DCG@k$ when there are only relevant candidates at the top-k (or fewer) positions.

#### 5.1.2.3 RankBoost Based Strategy

RankBoost [Freund et al., 2003a] adopts a *boosting* ensemble technique. Boosting is an iterative process that produces and combines different *weak learners*. In each of $i$ iterations, it increases emphasis on training instances which were not well modeled in the previous iteration (i.e., candidate tags in the training set which were not correctly ranked by the model built in the previous iteration, in our case). Increasing the weight given to these "harder" training examples, it potentially improves the final model, which is a linear combination of the weak learners, with weights defined by the learning rate $lr$, a tuning parameter.

RankBoost learns a linear combination of weak rankers as the final ranking function. Each weak ranker consists of a single attribute (one of the attributes in Section 4.1) and a threshold that best distinguishes between relevant and non-relevant can-

[4]We also experimented with other fitness functions (e.g., average precision) obtaining similar results.

didate tags. However, similarly to RankSVM, RankBoost operates on pairs of tags. After each iteration, the tag pairs are re-weighted: it decreases the weight of correctly ranked pairs and increases the weight of wrongly ranked pairs. As a result, the learning at the next iteration will be focused on dealing with pairs which are more difficult to rank. The total number of iterations $i$ is a tuning parameter.

### 5.1.3 State-of-the-art Tag Recommendation Methods for the Cold Start Scenario

In this section, we briefly describe the tag recommendation methods we adopt as baselines in the cold start scenario. The first baseline is CTTR (already described in Section 5.1.1), and thus is evaluated here in both cold start and non cold start scenarios. The second baseline, *K-Nearest Neighbors based Tag Recommender*, or simply *KNN* [Graham and Caverlee, 2008], extracts candidate tags from the $K$ most textually similar objects to the target object, and rank them according to *TermScore* measure, which we define in the following.

To compute *TermScore*, each object $d \in \mathcal{D}$ is first modeled as a bag of terms extracted from all its textual features (including $d$'s tags). The similarity between each object $d \in \mathcal{D}$ and $o$ is then computed using the cosine metric [Baeza-Yates and Ribeiro-Neto, 1999]:

$$Sim(d,o) = \frac{\vec{d} \cdot \vec{o}}{|d| \times |o|} = \frac{\sum_{i=1}^{|V|} w_{i,d} \times w_{i,o}}{\sqrt{\sum_{i=1}^{|V|} w_{i,d}^2} \times \sqrt{\sum_{i=1}^{|V|} w_{i,o}^2}} \tag{5.9}$$

where $|V|$ is the size of the term vocabulary in $\mathcal{D}$, and weight $w_{i,d}$ is a variant of the standard TFIDF metric. Specifically, $w_{i_d}$ is defined as $\sqrt{freq(t_i,d)} \times (1 + log(\frac{|\mathcal{D}|}{df(t_i)+1}))$, where $freq(t_i,d)$ is the frequency of $i^{th}$ term in object $d$ and $df(t_i)$ is the number of objects in $\mathcal{D}$ containing this term.

For each tag $t$ contained in one of the top-$K$ objects with the highest similarities with $o$, we assign the following score:

$$TermScore(t,o) = \sum_{i=1}^{K} Sim(d_i,o)^4 \times freq_{tag}(t,d_i) \tag{5.10}$$

where $freq_{tag}(t,d_i)$ is the number of times $t$ was applied as tag to object $d_i \in \mathcal{D}$[5].

We chose KNN as baseline because this technique is one of the basis of two of our

[5]Note that, in some applications, the same tag may be assigned multiple times to the same object by different users.

proposed tag recommenders for the cold start scenario, as we will see in Section 5.3.2.

The third baseline is referred to as *PoS+TFIDF* [Hulth, 2003]. It extracts candidate tags from the target object's description. In order to rank these candidates, PoS+TFIDF jointly exploits: (1) TFIDF values, (2) the relative position of the first occurrence of a word in the object's description, and (3) the PoS of the candidate tags. Originally, these tag quality attributes are combined using a rule-based classification algorithm, although the author claims the method does not depend on any specific learning technique. Thus, we here use the best performing L2R technique (RF), for a fair comparison with the other methods. We chose *PoS+TFIDF* as baseline because it is the only previous tag recommendation approach that exploits a syntactic attribute (PoS) to rank candidate tags.

## 5.2 State-of-the-art Personalized Baselines

One of the state-of-the-art personalized tag recommendation methods analyzed in this thesis is called Pairwise Interactions Tensor Factorization (PITF) [Rendle and Schmidt-Thie, 2010]. It was the winner of the graph-based personalized tag recommendation task in the PKDD Discovery Challenge 2009. PITF exploits the vocabulary of the target user expressed by the tags assigned by her to other objects as a representation of her interests and as the main evidence to support personalization.

Briefly, this approach explicitly models the two-way interactions between users, tags and objects by factorizing each of the three as a tensor product. From the set of tag assignments $\mathcal{P}$, their approach first infers pairwise ranking constraints. The idea is that, for a given $\langle$user $u$, object $o\rangle$ pair, one can assume that a tag $t_a$ is preferred over another tag $t_b$ if and only if $\langle u, o, t_a\rangle \in \mathcal{P}$ and $\langle u, o, t_b\rangle \notin \mathcal{P}$. These ranking constraints are then used as training data for a learning algorithm, based on a Bayesian Personalized Ranking (BPR) optimization criterion [Rendle et al., 2009b]. This learning method is based on stochastic gradient descent [Ruder, 2016] with bootstrap sampling [Efron and Tibshirani, 1993]. In other words, the pairwise constraints are sampled from the training data.

PITF has the following parameters: the dimension of factorization $\delta$, the number of interactions $\tau$, the learning rate for BPR $\lambda$, and the number of pair samples drawn for each training tuple $s$.

In previous work [Rendle and Schmidt-Thie, 2010], PITF was only evaluated in denser datasets, that is, datasets in which unpopular users, objects and tags were filtered out. In this thesis, all strategies are evaluated in more realistic scenarios,

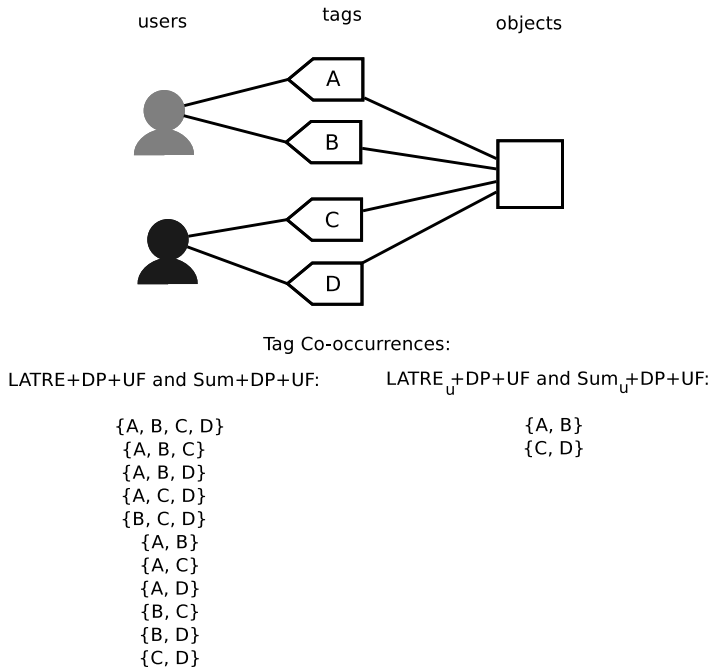



**Figure 5.2.** Tag co-occurrence patterns considered by the personalized tag recommendation methods.

without this kind of filtering. As we will see in Section 7.1.4, our strategies outperform PITF because we exploit several sources of evidence not exploited by it[6].

We also adopt as baselines the heuristics we proposed in [Belém, 2011]. They extend the $Sum^+DP$ and $LATRE{+}DP$ (Equations (5.5) and (5.6)) to also include the user related attribute $UF$ (Chapter 4, Section 4.1). We thus proposed eight new ranking functions composed by a weighted linear combination of the output of $Sum^+DP$ (or $LATRE{+}DP$) and the value of $UF$. Let $c$ be a candidate tag for target pair $\langle u, o\rangle$. The proposed heuristics have the following general structures:

$$Sum^+DP+UF(c,o,u,k_x,k_c,k_r,\alpha,\beta) = \beta Sum^+DP(c,o,I_o,k_x,k_c,k_r,\alpha) + (1-\beta)UF(c,u) \tag{5.11}$$

$$LATRE{+}DP{+}UF(c,o,u,\ell,\alpha,\beta) = \beta LATRE+DP(c,o,I_o,\ell,\alpha) + (1-\beta)UF(c,u) \tag{5.12}$$

Parameter $\beta$ ($0 \leq \beta \leq 1$) is used as a weighting factor. Note that $Sum^+DP$ and $LATRE{+}DP$ are computed only over candidates generated from the association

[6]We have also experimented comparing all strategies in denser versions of our datasets (with p-core=5, that is, filtering out users, tags and objects that appear less than 5 times in the folksonomy), and our methods still outperform PITF in this scenario, with gains ranging from 61% to 174% in precision

rules and terms extracted from other textual features of target object $o$, while $UF$ is computed for terms which were assigned as tags by user $u$ in the training set. We note that a candidate tag $c$ generated by co-occurrences or extracted from textual features may not be included in the tag assignment history of the target user (personomy). In this case, we set $UF(c, u) = 0$. Similarly, a candidate tag extracted from the user's personomy may not be in any textual feature of the target object $o$, presenting value 0 for its descriptive power metrics ($DP$).

In this thesis we propose variants of these two sets of heuristics, defined by the same Equations 5.11-5.12, but differing in the training set used. While $Sum^{+}DP{+}UF$ and $LATRE{+}DP{+}UF$ exploit co-occurrences between tags assigned to objects by different users (training set $\mathcal{D}$ defined in Section 3.4), the two variants, referred to as $Sum_u^{+}DP{+}UF$ and $LATRE_u{+}DP{+}UF$, exploit co-occurrences between tags assigned to the same object by only one user (training set $\mathcal{D}'$, defined in Section 3.4). Figure 5.2 illustrates the differences between the tag co-occurrence patterns of these methods. While $Sum^{+}DP$ and $LATRE{+}DP$ consider all combinations of two or more tags that are assigned to the same object by any user, $Sum_u^{+}DP{+}UF$ and $LATRE_u{+}DP{+}UF$ focus on the co-occurring tags that were assigned by the same user.

## 5.3 New Object-Centered Tag Recommendation Strategies

Our new tag recommendation methods are based on learning-to-rank (L2R) techniques. Some of these techniques have already been exploited for tag recommendation and were described in Section 5.1.2. In this section, we present alternative L2R-based methods that were not exploited in previous tag recommendation studies.

### 5.3.1 New Evaluated L2R Techniques

#### 5.3.1.1 Random Forest Based Strategy

The Random Forest (RF) algorithm [Breiman, 2001] is an ensemble method that combines a collection of decision trees. The learning of each decision tree in the ensemble happens in a recursive way: first, the most discriminative attribute (according to some measure, such as Information Gain) is selected as a decision node. The selected candidate tags are split according to a split value (e.g., average attribute value), and the process repeats in a top-down fashion to form a tree with $l$ terminal nodes, where $l$

is a tuning parameter. Once the decision tree is built, it can assign a real-valued score as output for an unseen (test) candidate tag.

The *RF* method exploits the *bagging* ensemble technique, i.e., each tree within the forest is built with a different bootstrap sample of size $n_b$ drawn from the original set of pairs $(M_c, r_c)$ that represents each candidate tag $c$ for the considered objects in our dataset, where, as discussed before, $M_c \in \mathbb{R}^m$ and $r_c \in \{0, 1\}$. The attribute selection for each split in a tree is conducted on a randomly selected subset of attributes, instead of on the full attribute set, as usually done in traditional decision tree algorithms. Each leaf in each tree corresponds to an output score to be assigned to a candidate tag. Once the forest is built, for each tag candidate $c$ in a target object $o$, the scores given by each tree to $c$ are averaged and used to produce the final ranking of candidates.

Besides $l$, the number $T$ of trees to grow per bootstrap sample and the number of attributes $m$ to consider when splitting each node are tuning parameters in RF. We note that, although each decision tree may suffer from overfitting, the aggregation of a larger number of low-correlated trees can mitigate this problem. The generalization error of a RF depends on both the correlation between trees in the forest and the strength of each individual tree. The more correlated each tree is, the higher the error rate becomes. The stronger each individual tree is (high accuracy), the lower the error rate becomes. By increasing $m$ or lowering $l$, both the correlation and the strength of each tree increases. By lowering $m$ or increasing $l$, each tree becomes more independent (less correlated), but also becomes weaker at the same time. Thus, there exists some optimal values of $m$ and $l$ that provide the optimal balance between the correlation and the strength to get the minimum generalization error. We set those parameters using the validation set, as described in Chapter 6. The implementation of RF and the next four approaches were provided by the RankLib learning to rank tool[7].

RF has been shown consistently effective and competitive in several real world benchmarks [Mohan et al., 2011]. Some of its strengths are its insensitivity to parameter choices, resilience to overfitting, and high degree of parallelization due the fact that single decision trees are built independently from others, thus making RFs inherently parallel.

#### 5.3.1.2 MART Based Strategy

Multiple Additive Regression Trees [Friedman, 2000] combines multiple decision trees by means of a *boosting* ensemble technique. The learning of each decision tree in the ensemble happens in a recursive way: first, the most discriminative attribute

[7] `http://people.cs.umass.edu/~vdang/ranklib.html`

(according to some measure, such as Information Gain) is selected as a decision node. The selected candidate tags are split according to a split value (e.g., average attribute value), and the process repeats in a top-down fashion to form a tree with $l$ terminal nodes, where $l$ is a tuning parameter. In each of its $i$ iterations, MART increases emphasis on training instances which were not well modeled in the previous iteration (i.e., candidate tags in the training set which were not correctly ranked by the model built in the previous iteration, in our case). Increasing the weight given to these "harder" training examples, it potentially improves the final model, which is a linear combination of the outputs of each decision tree, with weights defined by the learning rate $lr$, a tuning parameter.

#### 5.3.1.3 $\lambda$-MART Based Strategy

$\lambda$-MART [Wu et al., 2010], the winning approach at the Yahoo! Learning to Rank Challenge [Chapelle and Chang, 2011], is a combination of MART and the $\lambda$-Rank ranking model, which tries to directly optimize the value of an evaluation metric (listwise approach). The main difference between $\lambda$-MART and MART is that $\lambda$-MART learns which candidate in a pair of candidate tags must appear first in the ranking. In order to learn these pairwise preferences, $\lambda$-MART uses the gain in NDCG obtained from swapping the rank positions of candidates in any given pair of candidate tags for the same object $o$. Thus, unlike MART, the training is made by considering that each candidate tag is in the set of candidate tags $C_o$ for an object $o$. Similarly to MART, the learning rate $lr$, the number of terminal nodes $l$ and the number of iterations $i$ must be specified.

#### 5.3.1.4 ListNet Based Strategy

The goal of ListNet [Cao et al., 2007] is minimizing ranking errors, rather than minimizing errors in classification of pairs of tags or building regression models. ListNet is based on comparing the probability distribution of permutations of lists of candidate tags. Specifically, for a set of candidate tags associated with an object $o$, ListNet first defines a permutation probability distribution based on the scores produced by a ranking function for each candidate tag. It then defines another distribution based on the ground truth relevance labels, and measures the inconsistency between these two distributions. The ranking function is defined as a neural network model. Thus, it is possible to iteratively adjust this ranking function according to the measured inconsistency. This adjustment is done at a predefined learning rate $lr$ during $i$ iterations.

#### 5.3.1.5 AdaRank Based Strategy

The basic idea of AdaRank [Xu and Li, 2007] is to plug a selected evaluation metric into the *boosting* framework and directly optimize this metric. Specifically, it repeatedly builds weak rankers on the basis of re-weighted training of sets of candidate tags, i.e., each set of candidate tags $C_o$ for an object $o$ receives a weight. The weak rankers are linearly combined to make ranking predictions. Each attribute in Section 4.1, in isolation, is used as a weak ranker. The selected weak ranker in each iteration of the algorithm corresponds to the attribute that leads to the best performance over the weighted objects, measured by a given evaluation metric (NDCG, in our case). AdaRank runs for $i$ iterations. At each iteration, AdaRank maintains a distribution of weights over the objects in the training data. Initially, AdaRank sets equal weights to the objects. At each iteration, it increases the weights of those objects whose tags are not ranked well by the model created so far. As a result, the learning at the next iteration will be focused on the creation of a weak ranker that can work on the tag ranking of those "hard" objects.

Table 5.3 shows a summary of the characteristics of all analyzed L2R techniques.

**Table 5.2.** List of relevance-driven tag recommendation methods.

| | **Object-Centered Recommendation** | | **Personalized Recommendation** | |
|---|---|---|---|---|
| | **Method** | **Reference** | **Method** | **Reference** |
| **Baselines** | $Sum^+$ | Section 5.1.1 | | |
| | $LATRE$ | Section 5.1.1 | $PITF$ | Section 5.2 |
| | $CTTR$ | Section 5.1.1 | | |
| | $KNN$ | Section 5.1.3 | | |
| | $PoS + TFIDF$ | Section 5.1.3 | | |
| **Our Previous Heuristics** | $Sum^+DP$ | Section 5.1 | $Sum^+DP{+}UF$ | Section 5.2 |
| | $LATRE{+}DP$ | Section 5.1 | $LATRE{+}DP{+}UF$ | Section 5.2 |
| **Our New Heuristics** | | | $Sum_u^+DP{+}UF$ | Section 5.2 |
| | | | $LATRE_u{+}DP{+}UF$ | Section 5.2 |
| **L2R-based methods** | $RankSVM$ | Section 5.1.2.1 | $RankSVM$ | Section 5.4 |
| | $GP$ | Section 5.1.2.2 | $GP$ | Section 5.4 |
| | $RankBoost$ | Section 5.1.2.3 | $RankBoost$ | Section 5.4 |
| | $MART$ | Section 5.3.1.2 | $MART$ | Section 5.4 |
| | $\lambda$-$MART$ | Section 5.3.1.3 | $\lambda$-$MART$ | Section 5.4 |
| | $RF$ | Section 5.3.1.1 | $RF$ | Section 5.4 |
| | $ListNet$ | Section 5.3.1.4 | $ListNet$ | Section 5.4 |
| | $AdaRank$ | Section 5.3.1.5 | $AdaRank$ | Section 5.4 |
| **Cold start treatment** | $RF_{synt}$ | Section 5.3.2.1 | | |
| | $KNN_{synt}$ | Section 5.3.2.2 | | |
| | $RF_{synt} + KNN_{synt}$ | Section 5.3.2.2 | | |

### 5.3.2 Addressing Cold Start in Tag Recommendation

In this Section, we describe our proposed solutions to the specific cold start scenario. These solutions exploit the same L2R techniques as described in Sections 5.1.2 and 5.3.1, but include other tag quality attributes and an additional tag candidate

**Table 5.3.** Characteristics of our L2R-based strategies.

| Technique | Type of Approach | Ensemble | Generated model |
|---|---|---|---|
| *RankSVM* | Pairwise | - | Hyperplane |
| *RF* | Pointwise | Bagging | Set of "randomized" decision trees |
| *MART* | Pointwise | Boosting | Set of boosted decision trees |
| *λ-MART* | Listwise | Boosting | Set of boosted decision trees |
| *ListNet* | Listwise | - | Neural network |
| *AdaRank* | Listwise | Boosting | Sets of weighted weak rankers |
| *RankBoost* | Pairwise | Boosting | Sets of weighted weak rankers |
| *GP* | Listwise | - | Any function formed by a given set of operators and attributes |

source (the neighborhood of the target object) to compensate the lack of previously assigned tags in the target object. Because of the aforementioned lack of information, some co-occurrence tag quality attributes (e.g., $Sum$, $Sum^+$, $Vote$, $Vote^+$) cannot be exploited effectively. Besides that, the text in Web 2.0 applications is usually small and may present low quality [Figueiredo et al., 2012], thus statistical properties of the occurrence of candidate tags in the text such as our descriptive power attributes (e.g., $TF$, $wTS$, defined in Section 4.1.2) may not be enough to distinguish relevant from non relevant candidates.

Thus, we propose methods that exploit additional evidence of the relevance of candidate tags, in particular syntactic properties of words that occur in the target object description (Section 5.3.2.1). Later, we expand the set of recommendations by exploiting the neighborhood of the target object (Section 5.3.2.2).

#### 5.3.2.1 Including Syntactic Attributes

**Table 5.4.** List of tag quality attributes exploited by L2R-based methods (cold start scenario).

| Category | Attribute | Reference |
|---|---|---|
| Tag Co-occurrence | $S_{TitleToTag}$ | Eq. (5.3) |
| | $S_{TagToTag}$ | Eq. (5.4) |
| Descriptive Power | $TS$ | Eq. (4.5) |
| | $wTS$ | Eq. (4.7) |
| | $TF$ | Eq. (4.6) |
| | $wTF$ | Eq. (4.8) |
| Discriminative Power | $IFF$ | Eq. (4.9) |
| | $Stab$ | Eq. (4.10) |
| Predictability | $Entropy$ | Eq. (4.11) |
| | $Pred$ | Eq. (4.12) |
| Syntactic Properties | $P_\pi$* | Eq. (4.14) |
| Neighborhood-based | $TermScore$ | Eq. (5.10) |

*They consist of 11 attributes, one for each property $\pi$ listed in Table 4.1.

In order to address cold start in tag recommendation and evaluate our new proposed attributes for this scenario, we first extend both RF and RankSVM, including the 11 syntactic attributes described in Section 4.1.6 in their list of attributes, totaliz-

ing 22 attributes for each candidate tag (see the list of attributes used for the cold start scenario in Table 5.4). We will refer to these extensions as $RF_{synt}$ and $RankSVM_{synt}$, respectively, while the corresponding methods without the syntactic attributes will be referred to as simply $RF$ and $RankSVM$[8].

Following, we propose a strategy to provide further and potentially complementary tag recommendations based on the neighborhood of the target object, calculated based on the initial recommendations, instead of based on TFIDF weights only.

#### 5.3.2.2 Neighborhood Expansion

We also analyze the extent to which we can improve tag recommendation by further exploiting the neighborhood of the target object. The neighborhood of the target object has been exploited (though preliminarly) by previous methods [Graham and Caverlee, 2008; Martins et al., 2016], that we refer to as k-Nearest Neighbors based Tag Recommender, or simply KNN.

However, we can filter out noisy terms when computing the neighborhood, using our new proposed *TermScore′* which extends TermScore (Section 5.1.3) using a set of initial tag recommendations as representation for the target object. That is, instead of using all terms weighted by TFIDF, we exploit the scores given to the top-$r$ recommended tags by a recommender $rec$. The objective is to make a new, potentially better, vector representation of the target object $o$. Our assumption is that this new representation contains less noise, because top recommended candidate tags present higher chances to be relevant than the whole set of terms in the object's description and title. This is particularly useful for the target object, which has no available initial tags in a cold start scenario.

More specifically, given an initial tag recommender $rec$, and the top-$r$ recommendations it provides, the new *TermScore'* is calculated as shown in Eq. (5.10), but replacing $Sim(d_i, o)$ by a similarity metric $Sim'(d_i, o, rec, r)$ that considers the weights given by the initial tag recommender $rec$. That is, instead of $w_{i,o}$ in Eq. (5.9), we have $w^{rec}_{i,o}$, which is the score given by $rec$ to the candidate tag $i$ in object $o$, if $i$ is among the top-$r$ candidates, and 0 otherwise.

In our experiments, we used $RF_{synt}$ as the initial recommender $rec$, since it was the best performing method among the others in the considered scenario. Thus, we call the new version of the method as k-Nearest Neighbors with Syntactic Attributes, or simply $KNN_{synt}$.

[8]Note the reuse in the names of the methods both in non cold start and cold start scenarios, although they exploit different sets of attributes (see Tables 5.1 and 5.4)

Finally, to combine the strengths of both new methods, we perform a linear combination of their scores, naming it as $RF_{synt} + KNN_{synt}$. Specifically, the new method produces as score $a \times RF_{synt} + (1 - a) \times KNN_{synt}$, where the weighting factor $a$ is a tuning parameter. Figure 5.3 summarizes the interactions of these new methods. As depicted in Figure 5.3, $RF_{synt}$ combines various tag quality attributes, including syntactic attributes. The recommendations provided by $RF_{synt}$ are further expanded by $KNN_{synt}$. Finally, the scores of these two methods are combined with $RF_{synt} + KNN_{synt}$.

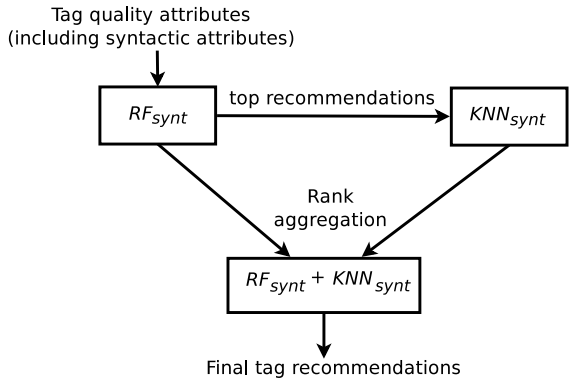



**Figure 5.3.** Basic operation of the proposed methods to address cold start and their combinations.

## 5.4 Extensions of L2R-based Strategies for Personalization

The L2R-based strategies described in Sections 5.1.2.1-5.1.2.2 can be easily extended to include new relevance attributes. In particular, in order to extend them to provide personalized recommendations, we included the attribute $UF$ in the list $L_{attr}$ of attributes. The complete list of attributes is shown in Table 5.1 ($3^{rd}$ column). Recall that, for personalized tag recommendations, each co-occurrence attribute presents two variations, depending on whether the training data used is separated per user or not. Thus, we adopted the best performing version when they are used as heuristics[9]. Besides that, all tags assigned by the target user to objects in the training set $\mathcal{D}$ were included as candidates. That is, the candidate set $C_{o,u}$ for a given object-user pair

[9]For example, $LATRE{+}wTS{+}UF$ performs better than $LATRE_u{+}wTS{+}UF$ (except for MovieLens dataset), while $Sum_u^+wTS{+}UF$ is better than $Sum^+wTS{+}UF$. Thus, we include $Sum(l = 3)$ (component of $LATRE{+}wTS{+}UF$) and $Sum_u(l = 1)$ (component of $Sum^+wTS{+}UF$) as attributes in $L_{attr}$ (except for MovieLens dataset which performed better with $Sum_u(l = 3)$ instead of $Sum(l = 3)$). An alternative would be to include both variations as attributes, but we opted for a smaller set of less redundant attributes.

$\langle o, u \rangle$ includes all terms generated by LATRE, all terms extracted from other textual features in $o$, and all terms assigned as tags by user $u$ in the training set $\mathcal{D}$. For each candidate $c \in C_{o,u}$, we compute the values of all attributes in $L_{attr}$ using $\mathcal{D}$ and the textual features associated with $o$.

Thus, the algorithms for personalized tag recommendation are the same as those described in the previous sections, except for the additional candidates and slightly different set of attributes. We argue that these methods perform well for both personalized and object-centered recommendation because they are flexible and robust strategies to generate relevant recommendation to the target object and to the target object-user pair, as we will discuss in Section 7.1.4. In particular, our methods can provide relevant recommendations to a user even when she does not have a history of tag assignments. In that case, the extraction of candidates from tag co-occurrences and multiple textual features provide more general recommendations to the considered object, which may be relevant to any user. As the user becomes more active, however, our methods can provide a higher level of personalization, thanks to the use of the UF attribute.

Table 5.2 lists all analyzed tag recommendation methods, while Table 5.3 summarizes key characteristics of the different techniques employed in our L2R-based methods.

## 5.5 Adding Novelty and Diversity

In this section, we describe our new tag recommendation methods that address other objectives than relevance, namely, novelty and diversity. They can be classified into implicit and explicit methods, according to the diversification approach they exploit (recall from Section 3.3).

### 5.5.1 Implicit Method

Our implicit strategy, called $GP_{rnd}$, extends the $GP$-based solution described in Section 5.1.2.2. The Genetic Programming approach was chosen due to its flexibility and easiness to incorporate new aspects to its objective function. $GP_{rnd}$ exploits the same set of candidate terms of our relevance driven strategies, including $GP$. However, it introduces new attributes in the list $L_{attr}$ and as part of the objective function. Specifically, we include *Average Distance to other Candidates* ($ADC$), defined in Section 4.3, in $L_{attr}$ and (indirectly) in the objective function. Moreover, unlike in $GP$, which exploits $IFF$ only as a relevance attribute in $L_{attr}$, in $GP_{rnd}$ we also have it as part

of the objective to be optimized, which changes the search space for recommendation functions.

In order to add the novelty of a list of recommended terms $C$ to the objective function of $GP_{rnd}$, we employed the metric *Average Inverse Popularity* over the top $k$ positions of the ranking, $AIP@k$, adapting it from [Vargas and Castells, 2011] to our context. We define $AIP@k$ as a normalized average of the *IFF* values of the first $k$ recommended terms. Let $disc(i) = 1/\log(1+i)$ be a rank discount function that provides a weight for the $i^{th}$ position of the ranking. $AIP@k$ of list $C$ is defined as:

$$AIP@k(C) = \frac{1}{K}\sum_{i=1}^{k} disc(i) \times IFF(c_i), \tag{5.13}$$

where $c_i$ is the $i^{th}$ term in $C$ and $K = \sum_{i=1}^{k} disc(i) \times IFF_{max}$ is the normalization constant.

We introduce diversity to the objective function by using the *Average IntraList Distance* in the top $k$ positions of the list of recommended terms $C$ (AILD@k) [Vargas and Castells, 2011], defined as

$$AILD@k(C) = \frac{1}{K'}\sum_{i=1}^{k}\sum_{j=i+1}^{k} dist(c_i, c_j), \tag{5.14}$$

where $K' = (k^2 - k)/2$ is a normalization constant, and $dist(c_i, c_j)$ is as defined in Section 4.3.

Finally, we define the new objective function (*fitness*) as a convex linear combination of the three aspects (relevance, novelty and diversity) as

$$Fit(C) = \alpha AIP@k(C) + \beta AILD@k(C) + (1-\alpha-\beta)NDCG@k(C), \tag{5.15}$$

where $0 \leq \alpha \leq 1$ and $0 \leq \beta \leq 1$ are tuning parameters to weight the evaluation metrics.

### 5.5.2 Explicit Methods

We propose here three new, complementary methods to address relevance, novelty and explicit diversity.

Our first method, called *Explicit Tag Recommendation Diversifier*, or *xTReD*, seeks to directly maximize the set of categories covered by the recommended tags. In its general form, maximizing topic coverage is an NP-hard problem [Agrawal et al., 2009].

Fortunately, there is a well-known greedy algorithm for this problem, which achieves an approximation factor of $(1-1/e) \approx 0.632$ of the optimal solution [Hochbaum, 1997]. This is also the best possible polynomial-time approximation for the problem, unless NP $\subseteq$ DTIME($n^{O(\log \log n)}$), where $n$ is the number of items to be diversified [Feige, 1998; Khuller et al., 1999]. This greedy approach is described in Algorithm 1.

*xTReD* takes as input an object $o$ and a diversification cutoff $\tau$. In its first step, *xTReD* calls a tag recommendation method *rec* to produce an initial ranking $C_o$ of recommended tags, generated with a relevance-focused objective (line 1). Any relevance-driven tag recommender could be used in this step. We exploit *RF*, which produced the best results among our relevance-driven methods.

**xTReD**$(o, \tau)$
1: $C_o^\tau \leftarrow rec(o, \tau)$ // relevance-driven recommendations
2: $C_o^S \leftarrow \emptyset$
3: **while** $|C_o^S| < min(\tau, |C_o|)$ **do**
4: $\quad t^* \leftarrow argmax_{t \in C_o^\tau} f(o, t, C_o^S)$
5: $\quad C_o^\tau \leftarrow C_o^\tau \setminus \{t^*\}$
6: $\quad C_o^S \leftarrow C_o^S \cup \{t^*\}$
7: **end while**
8: **return** $C_o^S$

Algorithm 1: The xTReD algorithm.

Let $C_o^\tau$ be the top $\tau$ recommendations in $C_o$. The goal is to produce a permutation of $C_o^\tau$ so as to raise the diversity in the top positions of the ranking of recommended tags, given that those tags are often the ones that the user looks at. A complete permutation of $C_o$ ($\tau$=$|C_o|$) could be produced. However, we can reduce $\tau$ for efficiency reasons and as a means to restrict the search for more diverse tags among the most relevant ones, avoiding severe relevance penalties.

The permutation $C_o^S$ is initialized as an empty set (line 2), and is iteratively constructed (lines 3-7). The objective function $f(o, t, C_o^S)$ scores each yet unselected tag $t \in C_o^\tau \setminus C_o^S$ in light of the object $o$ and the tags already in $C_o^S$, selected in the previous iterations of the algorithm (line 4). The highest scored tag, $t^*$, is then removed from $C_o^\tau$ (line 5) and added to $C_o^S$ (line 6). Finally, the produced diverse ranking $C_o^S$ is returned (line 8).

To instantiate the objective function $f(o, t, C_o^S)$ in Algorithm 1, *xTReD* builds upon a state-of-the-art framework for diversifying search results, called xQuAD [Santos et al., 2010]. The xQuAD framework instantiates the aforementioned function in order to score the documents retrieved for a given query proportionally to these documents' coverage and novelty in light of the multiple possible information

needs underlying this query [Santos et al., 2010; Santos and Ounis, 2011]. In the context of *xTReD*, instead of a ranking of documents for a query, we seek to diversify a ranking of tags for a given object. More precisely, *xTReD* includes a new instantiation of the objective function $f(o, t, C_o^S)$, such that:

$$f(o,t,C_o^S) = (1-\lambda) \times \Pr(t|o) + \lambda \times \sum_{z \in Z_o} \Pr(z|o) \Pr(t|o,z) \prod_{t' \in C_o^S} (1 - \Pr(t'|o,z)), \tag{5.16}$$

where $Z_o$ is a set of topics associated with the object $o$ and $0 \leq \lambda \leq 1$ is a tuning parameter used to balance the trade-off between promoting relevance or diversity. The greater the value of $\lambda$, the more importance is given to diversity. The idea is to promote tags that are simultaneously highly related to at least one of the topics of the target object and little related to the topics of the tags already selected as recommendation (captured by the product over $t' \in C_o^S$), hence increasing the coverage of topics in the top positions of the list of recommendations.

When $\lambda = 0$, Equation (5.16) reduces to $\Pr(t|o)$, which results in a pure relevance-driven tag recommendation, as produced by a non-diversification baseline. In our experiments in Chapter 7, we define $\Pr(t|o) = 1/r_t$, where $r_t$ is the position of the tag $t$ in the ranking produced by the initial ranker *rec*. In order to estimate the second half of Equation (5.16), we infer the distribution $\Pr(z|o)$ of topics $z \in Z_o$ for an object $o$ from the available training data or using the LDA algorithm, as discussed in Section 4.3. Finally, to estimate how much a given tag $t$ covers the topic $z$ of the object $o$, we approximate the probability $\Pr(t|o,z)$ as $\Pr(t|o,z) \approx \Pr(t|o) \times \Pr(z|t)$, where $\Pr(z|t)$ is an estimate of the probability that tag $t$ is related to topic $z$, already defined in Section 4.3.

Our second method extends *RF* to include two new metrics, defined in Equations (4.16) and (4.17), that capture explicit diversity as tag attributes. Moreover, this method also includes *IFF* as an attribute, capturing aspects related to both relevance (i.e., discriminative capacity) and novelty (i.e., rarity). Like *RF*, our new method, referred to as $RF_t$, still has the objective of maximizing relevance of the recommendations, capturing novelty and diversity indirectly at the attribute level.

Our third approach, called *explicit Tag Recommendation Diversifier with Novelty Promotion* or simply *xTReND*, builds upon *xTReD* and $RF_t$. Although it uses the same general algorithm described above (Algorithm 1), it differs from *xTReD* in two core components. First, it employs $RF_t$ as the basic recommender[10] (line 1), and thus

[10]We also tested a different method as initial ranker, the GP-based tag recommender. The gains

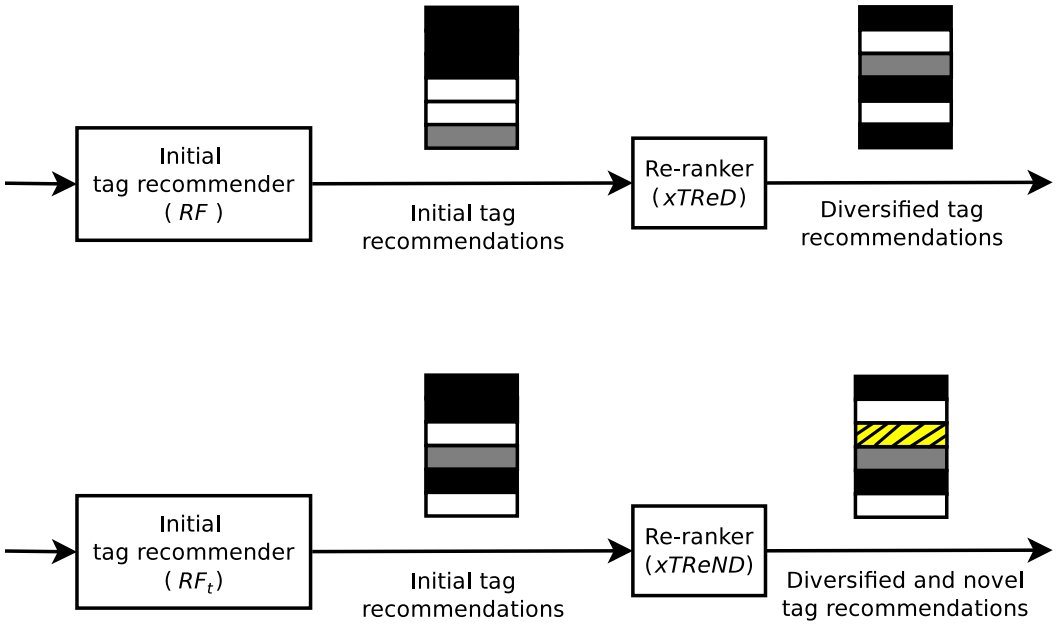



**Figure 5.4.** Illustration of $xTReD$ and $xTReND$: general structure and expected results. The rectangles represent the ranked list of recommended tags, and each color represents a different topic related to the target object.

already captures relevance, novelty and diversity at the attribute level. Second, it uses a new instantiation of the objective function that also captures the same three aspects. The new objective function is defined as:

$$\begin{aligned} f(o,t,C_o^S) = \quad & (1-\alpha-\beta)\times \Pr(t|o) + \alpha \times IFF(t) + \\ & +\beta \times \textstyle\sum_{z\in Z_o} \Pr(z|o)\Pr(t|o,z)\prod_{t'\in C_o^S}(1-\Pr(t'|o,z)), \end{aligned} \tag{5.17}$$

where $IFF$ is the novelty metric defined in Eq. (4.9). The tuning parameters $\alpha$ and $\beta$ ($0 \leq \alpha, \beta \leq 1$) are used to balance the trade-off between promoting relevance or novelty or diversity. The higher the values of $\alpha$ and $\beta$, the more weight is given to novelty and diversity, respectively.

Thus $xTReND$ captures relevance, (popularity-based) novelty and explicit diversity at both attribute and objective levels. Its design is motivated by the absence of a previous approach that directly includes popularity-based novelty, in addition to explicit topic diversity and relevance, as part of the goal to be maximized.

To better distinguish $xTReND$ from $xTReD$, Figure 5.4 illustrates the general structure of these methods and their expected results. In the figure, the rectangles

of the explicit diversification and novelty promotion of $GP$ were similar to the gains of the explicit diversification of $RF$. Thus, we focus our evaluation using the Random Forests based methods in Chapter 7, since it produced the best results.

represent the ranked list of recommended tags. We use different colors to represent different topics related to the target object[11]. Focusing first on *xTReD* (top diagram in Figure 5.4), the initial tag recommender, *RF*, which is driven only by relevance, prioritizes one topic (represented by the black color) over the others, while *xTReD* rearranges the results so as to allow tags related to different topics to appear earlier in the ranking. In contrast, *xTReND* (bottom diagram) uses $RF_t$ as initial recommender, which already introduces some diversification and novelty to the results, compared to *RF*. Like *xTReD*, the *xTReND* re-ranker also promotes tags related to different topics to earlier positions of the ranking, Additionally, *xTReND* is also able to bring a tag related to a novel topic (represented by the yellow rectangle with diagonal lines) to the object's top recommendations.

We further illustrate the re-ranking step performed by *xTReND* by focusing on the real example mentioned in Section 1, namely the recommendations for the movie "X-Men: The Last Stand". For simplicity, consider that only the top $\tau$=10 candidate tags will be re-ranked. Table 5.5 shows, for each of the top 10 most relevant tag candidates, their estimated values of relevance, novelty, and how much they are related to the three topics (i.e., genres) of the movie. Note that the first column of the table presents the candidate tags sorted by relevance. Table 5.6 shows the $f(o, t, C_o^S)$ scores calculated in each iteration of the methods, while Table 5.7 shows the estimated utility of each topic in each iteration. We calculate this topic utility as $u_z(C_o^S) = 1 - \frac{1}{K}\sum_{t \in C_o^S} \Pr(z|t)$, where $K = \sum_{z \in Z_o} u_z(C_o^S)$, and we set $u_z(C_o^S) = 1$ for all topics when $C_o^S = \emptyset$, that is, before the first iteration. Entries containing "-" indicate tags that were already selected in previous iterations. Note that the first column of Table 5.6 presents the list of candidate tags in the order they were selected by *xTReND*, that is, the list of candidates *after* re-ranking.

In the first iteration of the re-ranking, no tags have been selected yet ($C_o^S = \emptyset$). All topics present the same utility (Iteration "0" in Table 5.7). Tag "genetics" has the highest score, probably due the fact that it presents the highest probability to be related to one of the topics of the movie (Sci-Fi), and also presents good relevance and novelty estimates. Since no movie genre of the considered object has been covered yet, all genres are equally good choices to be covered first[12]. Thus, the algorithm appends tag "genetics" to the new, re-ranked list of tag recommendations. Next, in the second iteration, tag "dvd" is selected, despite being little related to any of the three

[11]For the sake of simplicity, we assume in this example that a single topic (color) is associated to each tag (rectangle). In reality, multiple topics may be associated to the same tag $t$, and the strength of the semantic association between them is given by $\Pr(z|t)$.

[12]Recall that we are assuming a uniform distribution of the topics (i.e., genres) related to the movie.

**Table 5.5.** Example of the re-ranking step of *xTReND* for the movie "X-Men: The Last Stand": statistics of top candidate tags (candidates are sorted by relevance).

| **Candidate tag** ($t$) | **Relevance** $\Pr(t\|o)$ | **Novelty** $IFF(t)$ | **Topic probability**: $\Pr(z\|t)$ | | |
|---|---|---|---|---|---|
| | | | *Fantasy* | *Thriller* | *Sci-Fi* |
| dvd | 1.00 | 1.72 | 0.04 | 0.10 | 0.04 |
| genetics | 0.50 | 6.07 | 0.00 | 0.17 | 0.30 |
| biology | 0.33 | 6.07 | 0.00 | 0.23 | 0.27 |
| comics | 0.25 | 4.21 | 0.09 | 0.10 | 0.11 |
| mckellen | 0.20 | 5.87 | 0.09 | 0.09 | 0.09 |
| marvel | 0.17 | 5.56 | 0.09 | 0.16 | 0.16 |
| mutant | 0.14 | 6.19 | 0.00 | 0.09 | 0.23 |
| super-hero | 0.13 | 5.01 | 0.07 | 0.17 | 0.17 |
| based | 0.11 | 2.26 | 0.05 | 0.08 | 0.07 |
| ummarti2006 | 0.10 | 4.69 | 0.05 | 0.13 | 0.05 |

**Table 5.6.** Example of the re-ranking step of *xTReND* for the movie "X-Men: The Last Stand": $f(o, t, C_o^S)$ scores for each candidate tag in each iteration (candidates are shown in the order they are selected by the method).

| **Candidate tag** | **Iteration** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| genetics | 0.177 | - | - | - | - | - | - | - | - | - |
| dvd | 0.163 | 0.158 | - | - | - | - | - | - | - | - |
| biology | 0.141 | 0.135 | 0.131 | - | - | - | - | - | - | - |
| mckellen | 0.093 | 0.092 | 0.091 | 0.090 | - | - | - | - | - | - |
| marvel | 0.091 | 0.089 | 0.088 | 0.087 | 0.087 | - | - | - | - | - |
| mutant | 0.088 | 0.087 | 0.086 | 0.085 | 0.085 | 0.084 | - | - | - | - |
| comics | 0.087 | 0.085 | 0.084 | 0.083 | 0.083 | 0.082 | 0.082 | - | - | - |
| super-hero | 0.077 | 0.075 | 0.074 | 0.073 | 0.073 | 0.073 | 0.073 | 0.072 | - | - |
| ummarti2006 | 0.063 | 0.063 | 0.062 | 0.062 | 0.062 | 0.061 | 0.061 | 0.061 | 0.061 | - |
| based | 0.039 | 0.039 | 0.038 | 0.038 | 0.038 | 0.038 | 0.038 | 0.038 | 0.037 | 0.037 |

**Table 5.7.** Updated marginal utility of each topic in each iteration.

| **Iteration** | *Fantasy* | *Thriller* | *Sci-Fi* |
|---|---|---|---|
| 0 | 1.00 | 1.00 | 1.00 |
| 1 | 1.00 | 0.64 | 0.36 |
| 2 | 0.94 | 0.58 | 0.48 |
| 3 | 0.97 | 0.57 | 0.47 |
| 4 | 0.91 | 0.58 | 0.51 |
| 5 | 0.88 | 0.59 | 0.53 |
| 6 | 0.90 | 0.61 | 0.49 |
| 7 | 0.87 | 0.62 | 0.51 |
| 8 | 0.87 | 0.61 | 0.52 |
| 9 | 0.86 | 0.60 | 0.54 |
| 10 | 0.85 | 0.60 | 0.55 |

genres. This choice is due the high relevance estimate given by the initial recommender, $RF_t$ (see Table 5.5). Following, tag "biology" is selected in the third iteration. This tag is relatively well connected to topic Thriller, which was not yet well covered by the previously selected candidates, according to the statistics of tag occurrences in genres of our MovieLens dataset. At this point, tags related to topics "Thriller" and "Sci-Fi" have been recommended, and thus, the utility of these topics decreased in Iteration "1". Tag "mckellen", referring to one of the main actors, is the next one

selected. This tag is somewhat related to all three genres of this movie, since the actor starred in other movies of these genres (such as other X-Men movies and "The Lord of the Rings"), not to mention that it is also highly novel and specific. This tag is related to the topic "Fantasy" (probably because the referred actor starred in other movies of this genre, such as "The Lord of the Rings" trilogy), which at this point had the maximum utility, since the other selected tags were not related to it. Next, tag "marvel" is appended. In comparison with "comics", which appeared first in the relevance-driven ranking (see Table 5.5), "marvel" is more novel and specific as well as more strongly related to the topics of the considered movie (according to the topic probability estimates). Thus, "marvel" is ranked higher than "comics" after the re-ranking. The other tags are appended similarly, considering the best trade-off between relevance, novelty and topic diversity.

The motivation of promoting novelty and diversity by re-ranking an initial recommendation list, as performed by both *xTReD* and *xTReND*, is that it is an intuitive solution to provide recommendations related to the different topics assigned to the target object, since the contribution of an item to the diversity of the list depends on the other items previously ranked in the list. Thus, an iterative solution that chooses the next recommendation considering the previsouly selected items is more natural than a solution that sorts the whole list in a single step, by the values of a given objective function, as performed by $GP_{rnd}$.

## 5.6 Summary

In this chapter, we presented the tag recommendation methods analyzed in this thesis. First, we described with more detail the relevance-driven baselines mentioned in Chapter 2, for both object-centered and personalized tag recommendation tasks. Next, we presented our new proposals, starting with five object-centered, relevance-driven strategies based on L2R techniques and their extensions to address personalization and cold start. Our contributions lie in the combination of tag quality attributes (some of them are proposed in this thesis, particularly the syntactic and topic related attributes), by means of L2R techniques that were not previously applied to the tag recommendation problem. Then, we presented our four new methods that address novelty and diversity aspects in addition to relevance, namely, $GP_{rnd}$, $RF_t$, *xTReD* and *xTReND*. The novel aspect of $GP_{rnd}$ refers to the new attributes and objective function that capture the different aspects of the problem. $RF_t$, in turn, brings a novel tag quality attribute related to the relationship between a tag and its topics. Finally,

*xTReD* and *xTReND* are re-rankers that explicitly address combinations of relevance, diversity and novelty. In the next chapters, we present the methodology we adopted to evaluate these methods, and the experimental results we obtained so far.

# Chapter 6

# Experimental Methodology

This chapter describes the methodology used in our experimental evaluation of the tag recommendation methods, including datasets (Section 6.1), evaluation protocol (Section 6.2) and parameterization of each method (Section 6.3).

## 6.1 Datasets

We evaluate the tag recommendation methods on five datasets, each containing the *title*, *tags* and *description* associated with real objects from Bibsonomy, LastFM, MovieLens, YouTube and YahooVideo. The Bibsonomy, LastFM and YouTube datasets also include the set of tag assignments ($\mathcal{P}$)[1], thus allowing the evaluation of object-centered and personalized tag recommendation methods. The YahooVideo dataset, in contrast, does not identify the user who assigned each tag, and thus is here used only in the evaluation of object-centered methods.

The Bibsonomy dataset is a snapshot of the system, obtained on January $1^{st}$ 2012, comprising 543,872 objects (bibtex records of publications). It is publicly available[2] and has been used in several previous efforts [Guan et al., 2009; Lipczak et al., 2009; Lipczak and Milios, 2011; Rendle and Schmidt-Thie, 2010]. The MovieLens dataset, also publicly available[3] contains 100,000 tags applied to 10,000 movies. The LastFM and YouTube datasets[4] were collected in August 2009, following a *snowball sampling* [Goodman, 1961]. That is, starting from a set of users (the most popular users) selected as seeds, the crawler recursively collects the objects posted by them and follows their social links to other users, collecting the objects posted by them. Our datasets

[1] On YouTube, only the video owner can assign tags to it.
[2] `http://www.kde.cs.uni-kassel.de/bibsonomy/dumps`.
[3] `http://www.grouplens.org/taxonomy/term/14`
[4] Visit `http://vod.dcc.ufmg.br/recc/` for information on data availability.

include the textual features and tag assignments associated with 2,758,992 LastFM artists and with more than nine million YouTube videos. The YahooVideo dataset was also gathered by snowball sampling, but using the most popular objects as seeds and following links of related videos[5]. It was gathered in October 2008, and contains the features of 160,228 objects.

**Table 6.1.** Datasets statistics.

| Dataset | Number of objects | Sample size | Categorized sample size | Avg. #tags per object ± Standard dev. |
|---|---|---|---|---|
| Bibsonomy | 543,872 | 150,000 | - | 4.9 ± 4.4 |
| LastFM | 2,758,992 | 150,000 | 35,975 | 13.5 ± 24.2 |
| MovieLens | 10,000 | 6,500 | 6,500 | 13.4 ± 15.5 |
| YahooVideo | 160,228 | 140,000 | - | 8.9 ± 7.3 |
| YouTube | 9,000,000 | 150,000 | 150,000 | 10.7 ± 5.7 |

We considered only objects with textual features in English, removed stopwords, and used the Porter Stemming algorithm[6] to remove the affixes of each word in each collected feature. Stemming was performed to avoid trivial recommendations such as plurals and other simple variations of the same word.

In order to evaluate diversity, we used different sources of category information for our datasets. Specifically, we used the pre-assigned categories for YouTube videos as well as the genres associated with each movie in MovieLens. We also collected the musical styles associated with the artists in the LastFM dataset from the AllMusic site[7], and used them as artist categories. The Bibsonomy and YahooVideo datasets do not contain categories and thus were evaluated using latent topics only. To evaluate the cold start scenario and the syntactic attributes, we focus on Bibsonomy, LastFM and MovieLens datasets only, because the complete sentences of the descriptions were not available in the other datasets.

Table 6.1 shows the total number of objects in our datasets and the number of objects in the evaluated samples.

## 6.2 Evaluation Methodology

We adopted a fully automatic evaluation methodology that has been used by most prior studies on tag recommendation [Gemmell et al., 2010; He and Chua, 2017;

[5]We adopted a slightly different sampling strategy for LastFM and YouTube, exploiting users and social links as opposed to videos and related video links, as we use the collected datasets to evaluate personalized recommendation strategies. As YahooVideo does not publish per-user information on tag assignment, we chose not to crawl that application again, thus relying on our previously gathered dataset and evaluating it only for object-centered recommendation.

[6]`http://tartarus.org/~martin/PorterStemmer/`

[7]`http://www.allmusic.com`

Heymann et al., 2008; Lipczak et al., 2009; Lipczak and Milios, 2011; Menezes et al., 2010; Rendle et al., 2009a; Yuan et al., 2017], including personalized tag recommendation [Garg and Weber, 2008; Guan et al., 2009; Rendle and Schmidt-Thie, 2010], as well as content recommendation in general [Guy et al., 2010; Zhang et al., 2012b]. It consists of using a subset of the object's pre-assigned tags as an *expected answer*, that is, as the relevant tags for that object. For personalized tag recommendation, specifically, a subset of the tags assigned by the target user to the target object is used as expected answer. We evaluate our cold start scenario in the object-centered task, using all tags previously assigned to the target object as its expected answer, since no tags are provided as input in this scenario.

Following the proposed methodology, for object-centered recommendation, in the non cold start scenario, for each object $o$ in the test and validation sets, we randomly select half of its tags to be included in $I_o$. The other half are included in $Y_o$, the expected answer for $o$. For the cold start scenario, $I_o$ is empty and all tags are included in $Y_o$. Similarly, for personalized tag recommendation, for each object $o$ in the test and validation sets, half of the tags assigned by the target user $u$ to the object $o$ are included in $I_o$ and the other half in $Y_{o,u}$. Tags assigned by other users to object $o$ (i.e., $I_{o,u'}$ for $u' \neq u$) are also used as input, being included in $I_o$. In all scenarios, we use title and description as textual features in $F_o$. Each object is thus represented by tuple $\langle I_o, F_o, Y_o \rangle$ for object-centered recommendation, or $\langle I_o, F_o, Y_{o,u} \rangle$ for personalized recommendation.

We note that the tags in the expected answer for an object $o$ are not exploited, in any way, to produce the recommendations for $o$ (i.e., they are not used neither for metric computation nor for learning the recommendation function). Thus, from the perspective of the evaluation being performed, these tags are effectively *new*. This methodology allows us to simulate a scenario where these tags have not been assigned to the object yet and, thus, are potential candidates for new recommendations. Moreover, these tags can be considered relevant as we know that one or multiple users actually used them to annotate the object.

We note that this methodology has some limitations, since some of the recommended tags, although not in the expected answer, might still be considered relevant to the given object (or object-user pair). Thus, results obtained according to the adopted methodology represent lower bounds in terms of precision and upper bounds in terms of recall.

Alternative evaluation methodologies would rely on manual assessment of the tag recommendations by either: (1) real users of the system under study, who created the objects for which tags are recommended and/or have already added some tags to them,

or (2) external volunteers. Whereas the former would be desirable, it is extremely hard to perform, particularly when covering different systems and a large number of different methods, as we do here[8].

In fact, the only effort we are aware of that evaluated tag recommendation in an online setting with real users of the application (real targets of the recommendations) was pursued by Jäschke et al. [2009]. They proposed an evaluation framework that relies on stored user clicks on recommended tags. This framework was used in the online tag recommendation task of the 2009 PKDD Discovery Challenge, with an evaluation focused on the Bibsonomy application.

In contrast, while the vast majority of prior studies adopted the automatic approach we used here, some prior efforts [Bi and Cho, 2013; Prokofyev et al., 2012; Siersdorfer et al., 2009; Sigurbjörnsson and Zwol, 2008; Wu et al., 2009] used external volunteers to evaluate the recommendations. However, we argue that this approach is not necessarily better than the automatic one. Indeed, in the case of personalized tag recommendations, this approach may not be adequate at all, as the external evaluations might introduce significant biases and inaccuracies to the evaluation which would be very hard to isolate[9], possibly invalidating the analyses.

Thus, we have adopted the automatic strategy, which is a well-established and widely adopted evaluation protocol in the area, in favor of a more extensive quantitative evaluation. This choice allowed us to cover a large number of methods and datasets, enabling us to draw solid conclusions from statistically significant results. We have performed manual evaluation with volunteers in a small sample of two datasets. Although our new strategy *xTReND*, in this manual evaluation methodology, achieved the higher average results for all considered evaluation metrics, we note that the number of evaluations was not enough to produce statistically solid results in this scenario. This probably occurred because the external volunteers were less familiar with the shown content than application users. We noted, for example, that application users provided tags which are, on average, more specific than those provided by external evaluators. Thus, all results reported in this thesis refer to the automatic evaluation methodology.

To apply the selected methodology, we performed a five-fold cross validation. That is, the objects were randomly distributed into five equal-sized portions. Three portions were treated as training set ($\mathcal{D}$), which was used for extracting association rules

[8] We here compare 18 object-centered and personalized tag recommendation methods proposed by us as well as 4 baselines.

[9] A tag that could be extremely meaningful to a particular user could be considered completely irrelevant by an evaluation with a different perception of the object's content.

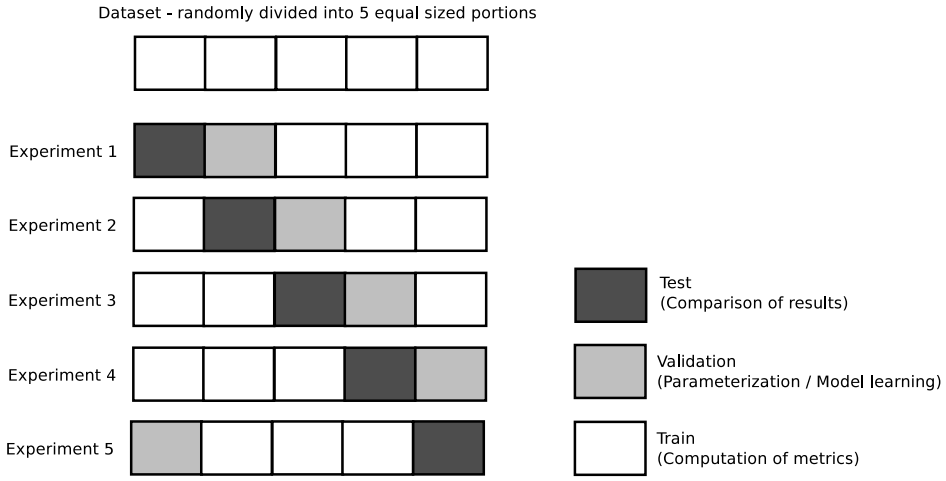



**Figure 6.1.** Illustration of the 5-fold cross-validation procedure.

and computing all metrics. A fourth portion was used as validation set $\mathcal{V}$, which, in turn, is part of the training set, being used to "learn" the solutions (e.g., to compute the Fitness function in the GP evolutionary process as well as learn vector $W$ in RankSVM and the forest of regression trees in RF), and to tune parameters of all recommendation methods, using inner cross-validation in $\mathcal{V}$. The last portion was used for testing. We repeat this procedure 5 times, alternating the roles of each portion of the dataset, as illustrated in Figure 6.1.

As discussed in Section 4.3, we estimate how related a tag $t$ is to a topic, which is necessary to evaluate topic diversity, by the probability of a topic given a tag. We experimented with two sources of topics for objects: (1) an explicit taxonomy represented by categories, obtained from our datasets (see Section 6.1), and (2) implicit topics generated by an unsupervised clustering technique. Specifically, we used Latent Dirichlet Allocation (LDA) [Blei, 2012], a probabilistic approach to generate and assign topics for each object based on terms (tags) contained in it. The use of LDA allows us to evaluate our approach in collections that do not contain explicit categories and to compare results across scenarios with different levels of generalization of categories, as we will see in Chapter 7. Next, we further describe the LDA method (Section 6.2.1) and introduce our main evaluation metrics (Section 6.2.2).

### 6.2.1 Latent Dirichlet Allocation

Latent Dirichlet Allocation, or LDA, is a probabilistic model that is based on the assumption that a document can be represented as a mixture of different topics [Blei, 2012], whereas a topic is defined as a distribution of words from a fixed vocabulary.

Given a number of topics and the distribution of words for each topic, LDA can be described by a generative process that explains how the content of a given document arises. Specifically, the method "generates" words for a given document as follows:

1. Randomly choose a topic distribution;
2. For each word to be generated in the document:
   (a) Randomly draw a topic from the distribution in (1);
   (b) Randomly draw a word from the word distribution corresponding to the selected topic.

This process assumes that each document exhibits topics in different proportions (step 1); each topic associated with a document is drawn from a per-document distribution (step 2a); and each word in the document is drawn from one of its topics (step 2b). In our case, documents refer to objects in our collections, and words refer to tags in $\mathcal{I}_o$ (i.e., previously assigned tags that were not included in the gold standard)[10]. In general, the goal of LDA is to exploit the *observed* terms (tags) in objects to infer their *hidden* topic structure (distribution). This can be thought of as "reversing" the aforementioned generative process.

Formally, given $\Pr(z|o)$, the probability distribution of topics for object $o$, and $\Pr(t|z_i)$, the distribution of tags for a latent topic $z_i$, the probability $\Pr(t_i|o)$ of a tag $t_i$ appearing in an object $o$ is defined as:

$$\Pr(t_i|o) = \sum_{j=1}^{n_Z} \Pr(t_i|z_i = j) \Pr(z_i = j|o), \tag{6.1}$$

where $\Pr(t_i|z_i = j)$ is the probability of tag $t_i$ appearing in topic $j$, and $\Pr(z_i = j|o)$ is the probability of topic $j$ being associated with object $o$. The number of latent topics $n_Z$ is a parameter that allows us to adjust the level of generalization/specificity of topics. The larger the number of topics, the more specific the generated topics.

LDA estimates the distribution of tags in topics $\Pr(t|z)$ and the distribution of topics in an object $\Pr(z|o)$ from a set of unlabeled objects (training set) assuming a *prior* Dirichlet distribution and a fixed number $n_Z$ of topics. A possible approach to infer these probabilities is to use *Gibbs sampling* [Blei, 2012], a sampling method performed in $m$ iterations of the two-step method described above. This is the method adopted in *pLDA*[11], which is the implementation of LDA [Liu et al., 2011] we used to

[10]Initial experiments showed that using terms extracted from other textual features of the object did not improve results, but we intend to further exploit this direction in the future.

[11]http://code.google.com/p/plda

generate topics for each object in our datasets. We discuss how we set the values of $n_Z$ and $m$ in Section 6.3.

### 6.2.2 Evaluation Metrics

We now present the metrics used to evaluate the quality of the recommendations produced by all considered tag recommendation methods. They are also used by the GP framework, whose search process tries to directly maximize the considered evaluation metric, as described in Section 5.1.2.2.

In order to evaluate the relevance of recommended tags, we measured *precision*, *recall* and the *Normalized Discounted Cumulative Gain* (NDCG) [Baeza-Yates and Ribeiro-Neto, 1999], all in the first $k = 5$ recommendations[12]. Precision is the fraction of the set of recommended tags that is relevant, while recall is the fraction of the set of relevant tags for an object that were indeed recommended. NDCG, already defined in Eq. (5.8), considers the order in which tags are recommended, emphasizing ranking relevant tags higher [Baeza-Yates and Ribeiro-Neto, 1999].

Specifically, let $Y$ be the set of relevant tags for object $o$ ($Y = Y_o$), or, in the case of personalized recommendations, for the object-user pair $\langle o, u\rangle$ ($Y = Y_{o,u}$). Let $C$ be the sorted set of recommendations generated by the method being evaluated, $C^k$ the top $k$ elements in $C$, and $C_i$ the $i^{th}$ element in $C$.

The precision in the first $k$ positions of the ranking, $P@k$, is defined as:

$$P@k(C, Y) = \frac{|C^k \cap Y|}{|C^k|} \tag{6.2}$$

The recall, in turn, is defined as:

$$Recall@k(C, Y) = \frac{|C^k \cap Y|}{|Y|} \tag{6.3}$$

To assess the diversity of a list of recommended tags, we use three metrics traditionally used for evaluating search result diversification methods [Clarke et al., 2011; Dang and Croft, 2012; Santos et al., 2010; Vargas et al., 2012]. Two of them – $\alpha$-NDCG and ERR-IA – are the primary evaluation metrics used in the diversity task of the TREC Web track [Clarke et al., 2012]. They are cascade metrics that penalize redundancy (and thus also capture the topic-related novelty discussed in Chapter 3) by modeling the behavior of a user who stops inspecting the ranking once a relevant tag is observed [Vargas and Castells, 2011]. While $\alpha$-NDCG incorporates a notion of

[12]We note that qualitatively similar results of precision, recall and NDCG were also obtained for larger values of $k$.

the *expected gain* attained by each ranked tag, ERR-IA measures the *expected retrieval performance* with respect to multiple topics.

Specifically, in order to define $\alpha$-NDCG@k, we first define $\alpha$-DCG@k as:

$$\alpha\text{-}DCG@k(C,o) = \sum_{i=1}^{k} disc(i) \times \sum_{z \in Z_o} J(C_i, z)(1-\alpha)^{r(i,z,C)}, \tag{6.4}$$

where *disc* is the same discount function used by *DCG* in Eq. (5.8), and $J(C_i, z)$ is equal to 1 if the $i^{th}$ candidate returned in $C$ is related to topic $z$, and 0 otherwise. The tolerance to redundancy is determined by parameter $\alpha$, a value in the [0,1] range. In this thesis, $\alpha$ is set to 0.5, as in many other studies [Clarke et al., 2012; Dang and Croft, 2012; Vargas et al., 2012]. Function $r(i, z, C)$ outputs the number of candidates in $C$ recommended before the $i^{th}$ position that are related to topic $z$, that is:

$$r(i, z, C) = \sum_{j=1}^{i-1} J(C_j, z) \tag{6.5}$$

The normalized $\alpha$-DCG@k, $\alpha$-NDCG@k, is defined as:

$$\alpha\text{-}NDCG@k(C,o) = \frac{\alpha\text{-}DCG@k(C,o)}{IADCG@k}, \tag{6.6}$$

where IADCG@k is the value obtained for $\alpha$-DCG@k when there is no redundancy, that is, all topics associated with the object appear only once in the ranking.

ERR-IA@k, as implemented for the task of the TREC Web track [Clarke et al., 2012], is defined similarly to $\alpha$-NDCG@k. The only difference is that the discount function $disc(i)$ is replaced by $disc_{ERR}(i) = 1/i$.

In addition to $\alpha$-NDCG and ERR-IA, we also assess the diversity of the recommended tags using (sub)topic recall—*S-Recall* [Zhai et al., 2003], which quantifies the fraction of unique topics associated with the object that are covered by the top ranked tags.

To assess novelty, we use the Average Inverse Popularity in the top $k$ recommendations, defined in Eq. (5.13) in Section 5.5.1.

All diversity metrics use the probability of a topic $z$ given a tag $t$, $\Pr(z|t)$, to estimate whether a recommended tag is related to a given topic of the object. We consider that a tag $t$ is related to topic $z$ if $\Pr(z|t) > \Pr(z)$ (recall Section 4.3). All metrics are computed over the top $k$ tags in the recommendation list, with $k$=5 as in [Belém et al., 2011].

We evaluate novelty and diversity orthogonally to relevance. Thus, a tag con-

sidered irrelevant might contribute with higher novelty or diversity. Alternatively, we could embed relevance in the diversity/novelty metric such that only relevant tags could contribute to raise these aspects. We opted for an orthogonal assessment of novelty/diversity because, unlike in previous diversification efforts in other contexts [Clarke et al., 2012], we lack per-topic relevance judgements for tags. Instead, we estimate how related the tags are to a topic using training data, and use this estimation in the diversity metrics.

We make a final note regarding our experimental setup. One might argue that the diversity and novelty improvements obtained by our methods over the baselines are expected because: (1) the diversifier exploits the same source of topics used to evaluate diversity, and (2) both novelty evaluation and attributes exploit the tag popularity in the dataset (estimated using training data). However, we argue that this is a valid approach because topic information is commonly available in objects in the form of categories or can be automatically generated by clustering strategies, such as LDA. In both ways it is possible to identify which topics are relevant for each object. Popularity information can be also computed, and it is, indeed, correlated with novelty and discriminative power. The surprising aspect is the possibility of obtaining large gains in diversity and novelty with little loss (if any) in relevance, as we will discuss in Chapter 7.

Besides that, one might argue that tags considered irrelevant[13] should not contribute to raise diversity, despite being related to a topic of the object. We note however that we tuned all methods to maximize the average diversity across all objects, *without harming relevance* (on average). After this tuning, we observed that a tag considered irrelevant contributed to amplify novelty or diversity for only a small fraction of the objects (less than 4%). Thus, we did indeed filter out the vast majority of such cases.

## 6.3 Parameterization

Our evaluation starts with a series of experiments with the validation set $\mathcal{V}$ to determine the best parameter values for each method in each dataset. For the relevance-driven methods, the best choice was defined as the one that maximizes NDCG@5 in the validation set. Similar results are obtained if any of the other considered relevance evaluation metrics are maximized. For the other methods, we considered the trade-off among relevance, novelty and diversity, choosing the parameter values that allowed the higher gains in novelty and diversity while keeping a small impact (if any) in relevance.

[13]Note that a tag may be relevant even if it is not in the expected answer.

Following, we summarize the parameterization of all tag recommendation strategies.

### 6.3.1 Relevance-Driven Strategies

#### Object-Centered Tag Recommendation Methods

We found that, for both $Sum^+$ and $Sum^+DP$ (for $DP$ equal to $TS$, $TF$, $wTS$ and $wTF$), the best parameter values are $k_r$=$k_x$=$k_c$=5. We also set $k_s$, parameter of the *Stab* metric, equal to 5. We tested these parameters sequentially for values equal to 1, 5, 10, 20 and 50. Best results for $\alpha$ varied between 0.8 and 0.99, depending on the dataset. For both $LATRE$ and $LATRE$+$DP$ (as well as for the L2R-based strategies), we set $\ell$=3, as in [Menezes et al., 2010]. Parameters $\sigma_{min}$ and $\theta_{min}$ directly impact the number of association rules generated, thus affecting the processing time of the recommender. We searched for a good tradeoff between processing time and recommendation precision. The lower $\sigma_{min}$ (or $\theta_{min}$), the larger the number of rules, thus, the longer the processing time. In general, precision decreases as $\sigma_{min}$ and $\theta_{min}$ increase. Thus, we chose $\sigma_{min}$ and $\theta_{min}$ so that the precision loss, with respect to results for $\sigma_{min}$=$\theta_{min}$=0, is under 3%.

We now turn to the parameterization of the L2R object-centered tag recommendation methods. We found our $RF$-based tag recommender to be very insensitive to parameterization. For both cold start and non cold start scenarios, the results obtained in our cross-validation experiments using different numbers of trees per bag ($T$=1, 10, 100) are statistically tied (with 95% confidence) for all datasets. We thus set $T$=1, due to the lower cost. Different sizes for the bootstrap sample $n_b$ also led to the same results, and we set $n_b$=300. We also fixed the number $\phi$ of all attributes selected in each split of the tree according to the default value originally suggested in [Breiman, 2001], i.e $\phi = \lfloor log_2(M+1)+0,5 \rfloor$, where $M$ is the total number of attributes. Despite the fact that this default value has been reported to work well in practice [Liaw and Wiener, 2002], we verified that other values ranging from $0.25M$ to $0.75M$ do not significantly impact our results. The only parameter that (slightly) impacts results is the number of terminal nodes $l$. We used cross validation to determine the best $l$ among values from 10 to 1000, finding that the best choice is $l$=1000 for all datasets.

The number of leaves $l$ also impacts $MART$ and $\lambda$-$MART$, the other tree-based approaches. We experimented with $l$ between 2 and 20, finding $l$=5 as the best choice for all datasets. Since the results obtained with different number of iterations ($i$ = 1500, 3000, 6000) are statistically tied in all $MART$ and $\lambda$-$MART$ experiments, we set $i$=1500 in all experiments due the lower cost. We also varied learning rate $lr$ between 0.0001 and 0.2, finding that the best choice was 0.1 for all datasets. Greater values for

both $l$ and $lr$ do not improve effectiveness, and make these methods very inefficient.

For *ListNet*, our results were not very sensitive to the $lr$ parameter. We tested values ranging from $10^{-7}$ to $10^{-1}$, finding that $lr = 10^{-5}$ always led to the best results. We also varied the number of iterations $i$ between 10 and $10^3$, finding, as best choices, $i = 160$ for MovieLens, $i = 10$ for YahooVideo, and $i = 40$ for the other datasets. Similarly, we tested ten values of $i$ between $10^2$ and $10^3$ for *RankBoost* and *AdaRank*. For *AdaRank*, $i = 300$ was the best choice in most datasets, except for YouTube, where $i = 100$ was the best value. For *RankBoost*, the best values varied according to the dataset: it was set to 500 in Bibsonomy, 700 in LastFM and 300 in the other datasets.

Regarding $LATRE{+}wTS$, $RankSVM$ and $GP$, we adopted the same best parameter values reported in [Belém et al., 2011] for LastFM, YouTube and YahooVideo, since the datasets are the same. For MovieLens and Bibsonomy, which were not included in that work, we follow the same methodology. Using cross-validation in $\mathcal{V}$, we found $j$=100 as the best cost for $RankSVM$, and we used the linear kernel. For $GP$, we set $n$=200 and $g$=200 (as in [Belém et al., 2011]), $k = 2$, $d = 7$, $p_c = 0.6$ and $p_m = 0.1$, as usually done in the literature [Banzhaf et al., 1998]. Finally, for $LATRE{+}wTS$, the best parameter values are $\alpha$=0.9 for Bibsonomy and $\alpha = 0.95$ for MovieLens. We set $\ell$=3 for the metric *Sum* (for all methods), as in [Belém et al., 2011].

In the cold start scenario, we tested both *KNN* and $KNN_{synt}$ using the following values for the number $K$ of nearest neighbors: 1, 5, 10, 20, 100, 1000. For both methods, precision reached its maximum value at $k$=100, and did not improve for higher values. Thus, we used $k$=100 for all neighborhood based methods. For $KNN_{synt}$, we also varied the number of initial candidate tags $r$ in $\{1, 5, 10, 20, 100, 1000\}$, finding that the best value is $r$=5 for all datasets. In Section 7, we will provide a more detailed analysis of this parameter. For the threshold *minfreq* used to filter out possibly noisy syntactic patterns, we set *minfreq*=10, after noticing that a more aggressive filter (i.e., *minfreq*=100) reduces tag recommendation effectiveness. For the ranking aggregation strategy, $KNN_{synt} + RF_{synt}$, we varied the weighting parameter $a$ in the set $\{0, 0.1, 0.2, ..., 1\}$. The best choice was $a = 0.8$ for MovieLens and Bibsonomy datasets, and $a = 0.6$ for the LastFM dataset.

We summarize our parameterization of all relevance-driven object-centered methods in Table 6.2.

### Personalized Tag Recommendation Methods

For the personalized tag recommendation strategies, the parameters in common with the object-centered methods (both heuristic and L2R-based methods) were set

with the same best values discussed in the previous section (shown in Table 6.2). Moreover, we set the descriptive power metric $DP{=}wTS$ in the experiments with heuristics $Sum^{+}DP{+}UF$ and $LATRE{+}DP{+}UF$, and their variants $Sum^{+}_{u}DP{+}UF$ and $LATRE_{u}{+}DP{+}UF$, since $wTS$ was the most promising descriptive power metric according to our findings.

**Table 6.2.** Best parameter values for the object-centered relevance-driven tag recommendation methods.

| Method | Parameter | Bibsonomy | LastFM | MovieLens | YahooVideo | YouTube |
|---|---|---|---|---|---|---|
| $Sum^{+}$ and extensions | $k_r$, $k_x$, $k_c$, $k_s$ | 5 | 5 | 5 | 5 | 5 |
| | $\ell$ | 1 | 1 | 1 | 1 | 1 |
| | $\sigma_{min}$ | 1 | 2 | 1 | 2 | 1 |
| | $\theta_{min}$ | 0.1 | 0.2 | 0.1 | 0.2 | 0.1 |
| $LATRE$ and extensions | $\ell$ | 3 | 3 | 3 | 3 | 3 |
| | $\sigma_{min}$ | 1 | 2 | 1 | 2 | 1 |
| | $\theta_{min}$ | 0.1 | 0.2 | 0.1 | 0.2 | 0.1 |
| $Sum^{+}DP$ | $\alpha$ | 0.9 | 0.95 | 0.9 | 0.8 | 0.8 |
| $LATRE{+}DP$ | $\alpha$ | 0.9 | 0.99 | 0.95 | 0.9 | 0.9 |
| $GP$ | $n$ | 200 | 200 | 200 | 200 | 200 |
| | $g$ | 200 | 200 | 200 | 200 | 200 |
| | $k$ | 2 | 2 | 2 | 2 | 2 |
| | $d$ | 7 | 7 | 7 | 7 | 7 |
| | $p_c$ | 0.6 | 0.6 | 0.6 | 0.6 | 0.6 |
| | $p_m$ | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| | $s$ | 500 | 500 | 500 | 500 | 500 |
| $RankSVM$ | kernel | linear | linear | linear | linear | linear |
| | $j$ | 100 | 100 | 100 | 100 | 100 |
| $RF$ and $RF_t$ | $T$ | 1 | 1 | 1 | 1 | 1 |
| | $\phi$ | 4 | 4 | 4 | 4 | 4 |
| | $l$ | 1000 | 1000 | 1000 | 1000 | 1000 |
| | $n_b$ | 300 | 300 | 300 | 300 | 300 |
| $RankBoost$ | $i$ | 500 | 700 | 300 | 300 | 300 |
| $MART$ | $l$ | 5 | 5 | 5 | 5 | 5 |
| | $lr$ | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| | $i$ | 1500 | 1500 | 1500 | 1500 | 1500 |
| $\lambda$-$MART$ | $l$ | 5 | 5 | 5 | 5 | 5 |
| | $lr$ | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| | $i$ | 1500 | 1500 | 1500 | 1500 | 1500 |
| $AdaRank$ | $i$ | 300 | 300 | 300 | 300 | 100 |
| $ListNet$ | $i$ | 40 | 40 | 160 | 10 | 40 |
| | $lr$ | $10^{-5}$ | $10^{-5}$ | $10^{-5}$ | $10^{-5}$ | $10^{-5}$ |
| $KNN$, $KNN_{synt}$ | $K$ | 100 | 100 | 100 | - | - |
| $KNN$ | $r$ | 100 | 100 | 100 | - | - |
| $KNN_{synt}$ | $r$ | 5 | 5 | 5 | - | - |
| $RF_{synt}{+}KNN_{synt}$ | $a$ | 0.8 | 0.6 | 0.8 | - | - |

The best values of parameter $\beta$, used by heuristics $Sum^{+}DP{+}UF$ and $LATRE{+}DP{+}UF$, and their variants, are shown in Table 6.3. These values allow us to compare the contribution of the $UF$ metric for personalized recommendation purposes. For example, considering $Sum^{+}_{u}wTS{+}UF$ strategy, we found that setting $\beta$ according to Table 6.3 leads to improvements in $P@5$ of up to 10% in LastFM and up to 7% in Bibsonomy and YouTube, with respect to results obtained with $\beta{=}1$ (that is, the weight assigned to $UF$ equal to 0). The improvements are larger in LastFM

**Table 6.3.** Best parameter values for personalized relevance-driven tag recommendation methods. Other parameters are fixed as in Table 6.2.

| Method | Parameter | Bibsonomy | LastFM | MovieLens | YouTube |
|---|---|---|---|---|---|
| $Sum^{+}DP{+}UF$ | $\beta$ | 0.3 | 0.1 | 0.2 | 0.4 |
| $Sum_u^{+}DP{+}UF$ | $\beta$ | 0.3 | 0.1 | 0.3 | 0.4 |
| $LATRE{+}DP{+}UF$ | $\beta$ | 0.5 | 0.9 | 0.5 | 0.7 |
| $LATRE_u{+}DP{+}UF$ | $\beta$ | 0.5 | 0.7 | 0.5 | 0.7 |
| $PITF$ | $\delta$ | 64 | 64 | 64 | 64 |
| | $s$ | 100 | 100 | 100 | 100 |
| | $\lambda$ | 0.01 | 0.01 | 0.01 | 0.01 |
| | $\tau$ | 50 | 50 | 50 | 50 |

probably due to the higher collaborative nature of tags in this application. That is, in LastFM any user has permission to assign tags to an object, whereas in YouTube, only the content publisher has this permission. In Bibsonomy, tags are also collaboratively created, but there is a lower level of activity in this application when compared to LastFM. Thus, LastFM presents a richer tag assignment history, which benefits all personalized recommendation methods. This fact reflects also on the best choices for $\beta$, whose values for LastFM and Bibsonomy are smaller than for YouTube in several cases. Indeed, the importance given to the $UF$ metric in LastFM is slightly higher than in the other two applications, particularly when $Sum^{+}$ is used.

The parameters of the PITF baseline were set as following. Similarly to [Rendle and Schmidt-Thie, 2010], we set the factorization dimension $\delta$=64, and the sample size $s$=100. We tested two different values for the learning rate $\lambda$, namely, 0.05 and 0.01, obtaining the best results for the smaller value. Moreover, as the algorithm converges before 50 iterations in our experiments, we set this value for $\tau$. These parameter values are also shown in Table 6.3.

### 6.3.2 Novelty/Diversity Promotion Strategies

For $GP_{rnd}$, we set $\alpha$=$\beta$, varying both at the same time[14], in the [0,0.6] interval. These parameters capture the tradeoff between relevance and the combination of novelty and diversity. Larger values of $\alpha$ (or $\beta$) lead to great losses in relevance. The value that lead to the best trade-off among relevance, diversity and novelty[15] is 0.1 for LastFM and MovieLens and 0.25 for Bibsonomy, YahooVideo and YouTube datasets.

For our re-ranking strategies, $xTReD$ and $xTReND$, we set the number of positions of the ranking to be diversified $\tau$=25, for efficiency reasons and because the tags in the top positions are much more likely to be selected (and visualized by the

[14]The results obtained following this approach are not worse than the best results when we set $\alpha$=0 (thus removing the novelty component $AIP$) and varied only $\beta$.

[15]The best results were chosen in terms of $\alpha$-$NDCG$ for diversity, but the best parameter values are the same for the other metrics.

user) than lower ranked tags. Our objective with *xTReD* is also to maximize diversity without harming relevance, while *xTReND* aims to maximize both novelty and diversity without diminishing relevance. Thus, we performed a grid search to find the best values for $\lambda$, $\alpha$ and $\beta$ (the tuning parameters) as well as for the number of topics $n_Z$ generated by LDA, such that diversity and novelty (in the case of *xTReND*) are maximized without hurting relevance by more than a factor of $\epsilon$%. We varied $\lambda$ and $\beta$ in 0, 0.05, 0.1, 0.2, ..., 0.9, 0.95, $\alpha$ in 0, 0.001, 0.005, 0.01 and 0.1[16]. For each dataset, we also experimented with the following values of $n_Z$: 5, 10, 100, and the number of predefined categories present in the dataset. We selected the best parameter values by setting $\epsilon$=4%. Finally, for LDA, we set the number of iterations of the Gibbs sampling at $m$=150, as suggested by the pLDA tool. The parameter configuration of LDA and all methods that promote novelty/diversity is shown in Table 6.4.

**Table 6.4.** Best parameter values for each novelty/diversity promotion tag recommendation method.

| **Method** | **Param.** | **Dataset** | | | | |
|---|---|---|---|---|---|---|
| | | **Bibsonomy** | **LastFM** | **MovieLens** | **YahooVideo** | **YouTube** |
| $GP_{rnd}$ | $\alpha$, $\beta$ | 0.25 | 0.1 | 0.1 | 0.25 | 0.25 |
| *xTReD* (w/ categories) | $\lambda$ | - | 1 | 1 | - | 0.9 |
| *xTReD* (w/ latent topics) | $\lambda$ | 0.7 | 0.9 | 0.8 | 0.8 | 0.7 |
| *xTReND* (w/ categories) | $\alpha$ | - | 0.001 | 0.01 | - | 0.01 |
| | $\beta$ | - | 0.95 | 0.9 | - | 0.8 |
| *xTReND* | $\alpha$ | 0.005 | 0.001 | 0.005 | 0.005 | 0.001 |
| (w/ latent topics) | $\beta$ | 0.7 | 0.9 | 0.8 | 0.8 | 0.7 |
| *LDA* | $n_Z$ | 10 | 10 | 19 | 100 | 5 |
| | $m$ | 150 | 150 | 150 | 150 | 150 |
| *xTReD* and *xTReND* (w/ categories and latent topics) | $\tau$ | 25 | 25 | 25 | 25 | 25 |

## 6.4 Summary

In this chapter, we presented the methodology we adopted to evaluate our tag recommendation methods. We described the datasets obtained from five Web 2.0 applications, the evaluation measures and the parameter setup of our methods. We also described the LDA technique exploited here to generate topics to be employed the same way as categories, particularly by the strategies that consider topic diversity. In the next chapter, we present our experimental results.

[16]Values larger than 0.1 were very detrimental to relevance.

# Chapter 7

# Experimental Results

In this chapter, we present the experimental results obtained from the evaluation of our tag recommendation strategies. Recall from Chapter 1 the main research questions that drive this study:
*RQ1: Can we improve the relevance of the recommended tags by means of a combination of tag quality attributes?*
*RQ2: How can we generate and rank candidate tags in a cold start scenario in which there are no previously available tags?*
*RQ3: How can we extend the proposed methods to provide personalized recommendations?*
*RQ4: Can we improve novelty and diversity of tag recommendation, while keeping the same levels of relevance?*

We start focusing on the evaluation of the relevance of tag recommendations provided by relevance-driven methods (Section 7.1), showing the effectiveness of L2R-based strategies in non-cold start (*RQ1*) and cold start scenarios (*RQ2*). Next, keeping the focus on relevance, we evaluate personalized tag recommendation strategies (*RQ3*). Finally, to answer *RQ4*, we evaluate all three aspects of the methods that consider novelty and diversity (Section 7.2), comparing them with the best relevance-driven strategy (*RQ1*). All results presented here were obtained in the test sets, using the best parameter values found in the validation set, as explained in Section 6.3.

## 7.1 Relevance Driven Methods

In this section, we address the topics of our research represented by *RQ1* - *RQ3*. Regarding *RQ1*, we compare different L2R approaches and heuristics for the object-centered (Section 7.1.1) and personalized (Section 7.1.4) tag recommendation tasks,

while we address the specific scenario of *RQ2* in Section 7.1.3. Regarding *RQ3*, we compare our new personalized tag recommendation methods with a state-of-the-art baseline (Section 7.1.4), and we show the benefits of personalization in tag recommendation (Section 7.1.5).

### 7.1.1 Object-Centered Tag Recommendation Results

We discuss the most relevant results of our 16 object-centered tag recommendation methods (8 heuristics and 8 L2R-based strategies), comparing them against the 3 baselines. Table 7.1 shows average P@5 results for all methods and datasets. Average Recall@5 and NDCG@5 are shown in Tables 7.2 and 7.3, respectively.

**Table 7.1.** Object-centered tag recommendation: average P@5 results and 95% confidence intervals (best results within each block - baselines, heuristics, and L2R-based strategies - in shaded entries; best overall results in bold).

| Strategy | Bibsonomy | LastFM | MovieLens | YahooVideo | YouTube |
|---|---|---|---|---|---|
| $Sum^+$ | 0.346 ± 0.003 | 0.411 ± 0.001 | 0.308 ± 0.011 | 0.484 ± 0.003 | 0.245 ± 0.002 |
| *LATRE* | 0.375 ± 0.003 | 0.405 ± 0.001 | 0.299 ± 0.009 | 0.608 ± 0.003 | 0.285 ± 0.004 |
| *CTTR* | 0.307 ± 0.003 | 0.288 ± 0.002 | 0.167 ± 0.006 | 0.467 ± 0.004 | 0.435 ± 0.002 |
| $Sum^+TF$ | 0.427 ± 0.002 | 0.404 ± 0.002 | 0.328 ± 0.004 | 0.643 ± 0.003 | 0.462 ± 0.001 |
| $Sum^+TS$ | 0.426 ± 0.002 | 0.418 ± 0.002 | 0.326 ± 0.003 | 0.673 ± 0.004 | 0.471 ± 0.002 |
| $Sum^+wTF$ | 0.431 ± 0.002 | 0.404 ± 0.002 | 0.328 ± 0.005 | 0.666 ± 0.002 | 0.490 ± 0.002 |
| $Sum^+wTS$ | 0.430 ± 0.003 | 0.417 ± 0.002 | 0.326 ± 0.004 | 0.707 ± 0.002 | 0.502 ± 0.003 |
| *LATRE+TF* | 0.433 ± 0.003 | 0.412 ± 0.001 | 0.309 ± 0.010 | 0.688 ± 0.002 | 0.465 ± 0.003 |
| *LATRE+TS* | 0.435 ± 0.002 | 0.398 ± 0.002 | 0.315 ± 0.010 | 0.716 ± 0.003 | 0.467 ± 0.003 |
| *LATRE+wTF* | 0.440 ± 0.003 | 0.408 ± 0.001 | 0.308 ± 0.010 | 0.718 ± 0.002 | 0.494 ± 0.003 |
| *LATRE+wTS* | 0.438 ± 0.002 | 0.401 ± 0.002 | 0.314 ± 0.008 | 0.733 ± 0.003 | 0.489 ± 0.003 |
| *RankSVM* | 0.456 ± 0.003 | 0.419 ± 0.002 | 0.346 ± 0.006 | 0.754 ± 0.002 | 0.517 ± 0.003 |
| *GP* | 0.441 ± 0.009 | 0.450 ± 0.006 | 0.363 ± 0.004 | 0.755 ± 0.005 | 0.520 ± 0.002 |
| *RankBoost* | 0.451 ± 0.003 | 0.424 ± 0.002 | 0.366 ± 0.002 | 0.763 ± 0.003 | 0.517 ± 0.002 |
| *RF* | **0.500 ± 0.003** | **0.494 ± 0.001** | **0.386 ± 0.006** | **0.797 ± 0.002** | **0.543 ± 0.002** |
| *MART* | 0.495 ± 0.003 | 0.489 ± 0.001 | 0.385 ± 0.002 | 0.792 ± 0.002 | 0.541 ± 0.001 |
| *λ-MART* | **0.500 ± 0.003** | **0.493 ± 0.002** | **0.385 ± 0.003** | **0.797 ± 0.002** | **0.546 ± 0.001** |
| *AdaRank* | 0.454 ± 0.003 | 0.134 ± 0.063 | 0.180 ± 0.149 | 0.712 ± 0.010 | 0.440 ± 0.038 |
| *ListNet* | 0.437 ± 0.006 | 0.398 ± 0.008 | 0.316 ± 0.010 | 0.661 ± 0.003 | 0.499 ± 0.003 |

All reported results are averages over 5 folds (test sets). For the GP-based and RF-based strategies, which are stochastic, each experiment was repeated 5 times. Thus, results are averages over 25 runs (5 folds, 5 seeds). Tables 7.1-7.3 also show 95% confidence intervals, indicating that, with that confidence, results do not deviate from the reported means by more than 3%. For each dataset, the tables are broken into 3 blocks: baselines, heuristics and L2R-based methods. Best results and statistical ties (according to a *2-sided t-test*[1] with $p$-value $< 0.05$) within each block are shown as shaded entries. Best overall results (and statistical ties) are shown in bold.

[1]We also applied the *t-test* with Bonferroni correction [Abdi, 2007] to control for the family-wise error rate.

**Table 7.2.** Object-centered tag recommendation: average Recall@5 results and 95% confidence intervals (best results within each block - baselines, heuristics, and L2R-based strategies - in shaded entries; best overall results in bold).

| Strategy | Bibsonomy | LastFM | MovieLens | YahooVideo | YouTube |
|---|---|---|---|---|---|
| $Sum^+$ | 0.337 ± 0.003 | 0.383 ± 0.001 | 0.253 ± 0.012 | 0.404 ± 0.003 | 0.213 ± 0.002 |
| *LATRE* | 0.366 ± 0.003 | 0.377 ± 0.002 | 0.238 ± 0.013 | 0.512 ± 0.002 | 0.251 ± 0.003 |
| *CTTR* | 0.300 ± 0.002 | 0.268 ± 0.002 | 0.121 ± 0.004 | 0.396 ± 0.003 | 0.405 ± 0.002 |
| $Sum^+TF$ | 0.418 ± 0.002 | 0.375 ± 0.002 | 0.255 ± 0.009 | 0.557 ± 0.003 | 0.424 ± 0.001 |
| $Sum^+TS$ | 0.417 ± 0.002 | 0.389 ± 0.002 | 0.253 ± 0.009 | 0.582 ± 0.004 | 0.432 ± 0.001 |
| $Sum^+wTF$ | 0.422 ± 0.002 | 0.376 ± 0.002 | 0.255 ± 0.010 | 0.579 ± 0.003 | 0.450 ± 0.002 |
| $Sum^+wTS$ | 0.422 ± 0.002 | 0.389 ± 0.002 | 0.253 ± 0.010 | 0.613 ± 0.002 | 0.461 ± 0.002 |
| *LATRE+TF* | 0.424 ± 0.003 | 0.385 ± 0.002 | 0.248 ± 0.013 | 0.593 ± 0.002 | 0.427 ± 0.002 |
| *LATRE+TS* | 0.426 ± 0.002 | 0.375 ± 0.002 | 0.252 ± 0.016 | 0.621 ± 0.002 | 0.431 ± 0.002 |
| *LATRE+wTF* | 0.432 ± 0.003 | 0.381 ± 0.002 | 0.247 ± 0.014 | 0.623 ± 0.002 | 0.455 ± 0.002 |
| *LATRE+wTS* | 0.430 ± 0.002 | 0.377 ± 0.002 | 0.250 ± 0.012 | 0.637 ± 0.002 | 0.451 ± 0.002 |
| *RankSVM* | 0.446 ± 0.003 | 0.390 ± 0.003 | 0.275 ± 0.008 | 0.651 ± 0.001 | 0.474 ± 0.003 |
| *GP* | 0.431 ± 0.009 | 0.415 ± 0.010 | 0.280 ± 0.005 | 0.654 ± 0.004 | 0.478 ± 0.002 |
| *RankBoost* | 0.441 ± 0.002 | 0.394 ± 0.002 | 0.287 ± 0.007 | 0.657 ± 0.003 | 0.474 ± 0.002 |
| *RF* | **0.489 ± 0.003** | **0.460 ± 0.002** | **0.301 ± 0.011** | **0.689 ± 0.001** | **0.498 ± 0.002** |
| *MART* | 0.484 ± 0.002 | 0.455 ± 0.002 | 0.300 ± 0.007 | 0.685 ± 0.002 | 0.496 ± 0.001 |
| *λ-MART* | **0.489 ± 0.002** | **0.459 ± 0.002** | **0.298 ± 0.011** | **0.690 ± 0.002** | **0.502 ± 0.001** |
| *AdaRank* | 0.443 ± 0.004 | 0.152 ± 0.120 | 0.194 ± 0.105 | 0.618 ± 0.009 | 0.408 ± 0.030 |
| *ListNet* | 0.428 ± 0.006 | 0.374 ± 0.006 | 0.259 ± 0.012 | 0.575 ± 0.003 | 0.459 ± 0.003 |

**Table 7.3.** Object-centered tag recommendation: average NDCG@5 results and 95% confidence intervals (best results within each block - baselines, heuristics, and L2R-based strategies - in shaded entries; best overall results in bold).

| Strategy | Bibsonomy | LastFM | MovieLens | YahooVideo | YouTube |
|---|---|---|---|---|---|
| $Sum^+$ | 0.326 ± 0.002 | 0.405 ± 0.001 | 0.299 ± 0.013 | 0.521 ± 0.003 | 0.257 ± 0.002 |
| *LATRE* | 0.349 ± 0.002 | 0.398 ± 0.001 | 0.314 ± 0.010 | 0.637 ± 0.002 | 0.298 ± 0.004 |
| *CTTR* | 0.263 ± 0.003 | 0.265 ± 0.001 | 0.166 ± 0.006 | 0.496 ± 0.004 | 0.450 ± 0.002 |
| $Sum^+TF$ | 0.379 ± 0.002 | 0.394 ± 0.001 | 0.337 ± 0.005 | 0.670 ± 0.003 | 0.455 ± 0.002 |
| $Sum^+TS$ | 0.378 ± 0.002 | 0.411 ± 0.001 | 0.336 ± 0.004 | 0.695 ± 0.004 | 0.469 ± 0.002 |
| $Sum^+wTF$ | 0.381 ± 0.002 | 0.395 ± 0.001 | 0.336 ± 0.005 | 0.691 ± 0.003 | 0.488 ± 0.002 |
| $Sum^+wTS$ | 0.380 ± 0.002 | 0.411 ± 0.001 | 0.336 ± 0.005 | 0.730 ± 0.003 | 0.506 ± 0.003 |
| *LATRE+TF* | 0.386 ± 0.002 | 0.403 ± 0.001 | 0.316 ± 0.010 | 0.708 ± 0.002 | 0.460 ± 0.003 |
| *LATRE+TS* | 0.397 ± 0.002 | 0.387 ± 0.003 | 0.322 ± 0.012 | 0.732 ± 0.002 | 0.467 ± 0.002 |
| *LATRE+wTF* | 0.395 ± 0.002 | 0.399 ± 0.001 | 0.315 ± 0.010 | 0.731 ± 0.003 | 0.488 ± 0.003 |
| *LATRE+wTS* | 0.397 ± 0.001 | 0.388 ± 0.003 | 0.321 ± 0.009 | 0.744 ± 0.002 | 0.489 ± 0.002 |
| *RankSVM* | 0.412 ± 0.002 | 0.407 ± 0.002 | 0.354 ± 0.007 | 0.765 ± 0.001 | 0.515 ± 0.003 |
| *GP* | 0.406 ± 0.006 | 0.440 ± 0.008 | 0.388 ± 0.002 | 0.770 ± 0.004 | 0.530 ± 0.002 |
| *RankBoost* | 0.444 ± 0.002 | 0.402 ± 0.002 | 0.307 ± 0.005 | 0.696 ± 0.003 | 0.488 ± 0.002 |
| *RF* | **0.455 ± 0.003** | **0.469 ± 0.002** | **0.415 ± 0.005** | **0.809 ± 0.001** | **0.553 ± 0.002** |
| *MART* | 0.449 ± 0.003 | 0.463 ± 0.001 | 0.411 ± 0.006 | 0.794 ± 0.002 | 0.547 ± 0.001 |
| *λ-MART* | **0.455 ± 0.002** | **0.468 ± 0.002** | **0.409 ± 0.010** | **0.802 ± 0.003** | **0.551 ± 0.002** |
| *AdaRank* | 0.446 ± 0.004 | 0.160 ± 0.128 | 0.206 ± 0.110 | 0.653 ± 0.010 | 0.419 ± 0.032 |
| *ListNet* | 0.431 ± 0.006 | 0.380 ± 0.007 | 0.273 ± 0.012 | 0.607 ± 0.003 | 0.472 ± 0.003 |

We start with two general findings: (1) the improvements obtained with our methods over the baselines are much more modest in the LastFM dataset, and (2) in general, the absolute values of the results are lower in the MovieLens dataset. The former observation can be explained by two factors: (1) there tends to be less overlap between the contents of title, description and tags associated with the same object on LastFM [Figueiredo et al., 2012], which leads to a greater concentration of *TS* (and

*wTS*) around small values, making it difficult to distinguish "good" from "bad" terms using these metrics; and (2) the number of tags per object tends to be smaller in our LastFM and Bibsonomy datasets (e.g., 48% and 73% of our YahooVideo and YouTube objects have fewer than 10 tags, against 94% of Bibsonomy objects, 88% of LastFM objects and 76% of MovieLens objects). These factors limit the benefits from using *TS* and *wTS* and from exploiting co-occurrence patterns among pre-assigned tags in that dataset. Regarding the second observation, we note that the MovieLens dataset is much smaller than the others (6,500 objects against at least 140,000 objects in the other 4 datasets), and thus provides a smaller training set to compute tag co-occurrences and other tag quality attributes. Besides that, the text of the MovieLens descriptions comprises movie synopsis, which are short and tend to hide part of the movie plot. This also makes it difficult to distinguish relevant from non relevant candidates based on statistics such as TF, similarly to what occurs in LastFM dataset.

Next, we turn our attention to the relative performance of specific methods, starting with the baselines. Consistently with [Menezes et al., 2010], we find that *LATRE* outperforms $Sum^+$ in most datasets. The improvements in P@5 reach 26%, whereas the gains in Recall@5 and NDCG@5 reach 27% and 22%, respectively. LastFM and MovieLens datasets are exceptions, but the difference between the two methods in these datasets is under 6% for all evaluation metrics. Moreover, *CTTR* appears as a good alternative to *LATRE* in the YouTube dataset, with improvements of 53%, 61% and 51% and 48% in average P@5, Recall@5 and NDCG@5, respectively. This occurs because *CTTR* exploits the terms of other textual features, while $Sum^+$ and *LATRE* are purely based on tag co-occurrences. Next, we discuss the results of our heuristics and L2R-based strategies.

### Unsupervised Heuristics

We find that our best heuristic in each dataset produces gains over the best baseline of 15% in P@5, considering average results across all datasets. Similarly, the average gains in Recall@5 and NDCG@5 are 15% and 13%, respectively. However, taking the best results in any dataset, the improvements reach 20% in *P*@5, 25% in Recall@5 and 17% in NDCG@5. Thus, introducing an attribute of descriptive power can greatly improve tag recommendation effectiveness.

In comparison with *CTTR*, the improvements in P@5, Recall@5 and NDCG@5 produced by our heuristics reach 88%, 106% and 93%, respectively, and remain quite impressive even if averaged across all datasets. For example, the corresponding gains produced by *LATRE*+*wTS*, which is one of our best performing heuristics (see below),

over *CTTR*, averaged across all five datasets, are 60%, 66% and 62%, respectively. These results illustrate the benefits of using our descriptive power metrics as well as exploiting pre-assigned tags. Moreover, the strategy adopted by *CTTR* to combine the different dimensions exploited for tag recommendation (i.e., co-occurrences and terms extracted from textual attributes), which is based on the precision that each dimension provides separately, may not be the best choice [Lipczak and Milios, 2011]. Indeed, the same authors later analyzed the potential benefits of introducing a tuning parameter to combine the different dimensions [Lipczak and Milios, 2011]. However, they did not propose an explicit recommendation method that uses this parameter. Our new heuristics use the $\alpha$ parameter that can be adjusted to the dataset (i.e., learned in a training set), producing better results. Moreover, as we show in Section 7.1.1, our L2R-based strategies produce further improvements by learning the weights applied to the different dimensions exploited by our methods and by using a larger set of relevance metrics.

Among the new heuristics, the most promising ones are *LATRE+wTS* and *LATRE+wTF*, as they yield the best results in most cases. To reach this conclusion, we make two observations. First, for any given descriptive power metric *DP* (i.e., *TS*, *TF*, *wTS* or *wTF*), *LATRE+DP* slightly outperforms $Sum^{+}DP$ in most cases (up to 4% in *P*@5, 4% in Recall@5 and 2% in NDCG@5). Thus there is still (modest) benefits when we exploit more complex association rules, but the inclusion of the textual features mitigates the difference in effectiveness among our heuristics. In the few cases where $Sum^{+}DP$ outperforms *LATRE+DP*, the gains in *P*@5 are under 4%.

Our second observation is that, comparing all four descriptive power metrics, *wTS* tends to yield the best results, followed by *wTF*, *TS* and *TF*. In particular, the use of *wTS* in *LATRE+wTS*, as opposed to the traditional *TF* metric, leads to gains of up to 7% in P@5, 6% in Recall@5 and 7% in NDCG@5. This is mainly because *wTS* considers that objects are composed of different features which, in turn, may have different descriptive capacities. *TF*, in contrast, tends to favor very frequent terms, even if they appear in a single feature. Such terms are often less relevant than those appearing across multiple features.

#### Learning-to-Rank based Strategies

We start by noting that the best L2R-based strategies (i.e., *RF* and $\lambda$-*MART*) outperform the best unsupervised heuristic (*LATRE+wTS*) by up to 29%, 23% and 22% in NDCG, precision and recall, respectively. Thus, although the best heuristic already captures the strongest tag quality evidence, we found that it is possible

to achieve significant gains over the heuristic by including other attributes. Moreover, these gains are even higher than the gains achieved by previously evaluated L2R methods [Belém et al., 2011]. They confirm the benefits of exploiting supervised L2R methods for tag recommendation, allowing an automatic search for a solution that combines a larger number of attributes when compared to unsupervised heuristics such as *LATRE+wTS*.

We now turn our attention to the comparison of the eight L2R-based strategies. Unlike existing comparisons of different L2R techniques in other domains such as document ranking [Gomes et al., 2013], there is a clear winning group of methods (*RF*, *MART* and *λ-MART*) in all 5 datasets, with a slight advantage of two of them (*RF* and *λ-MART*). The gains in NDCG of the winner methods over the best of the remaining L2R techniques considered (i.e., either *GP*, *RankSVM* or *RankBoost*) range from 4% to 12%. The corresponding gains in precision and recall reach 10% and 11%, respectively. These results confirm the effectiveness of methods based on an ensemble of decision trees, which are non-linear L2R strategies that have been shown to be effective and competitive in other studies [Friedman, 2000; Mohan et al., 2011].

The second group of methods is formed by the L2R strategies previously exploited in tag recommendation: *GP*, *RankSVM* and *RankBoost*. We conjecture that the results are explained by the following characteristrics of these techniques. *GP* is the most flexible strategy, allowing a wider range of types of recommendation functions (any function formed by the considered operators and attributes). However, this can be also a disadvantage because the search space is larger when compared to the search space of other methods, making it more difficult to find the best function. On the other hand, the shape of functions produced by *RankSVM* is pre-defined by the kernel function, which was set linear here (as this led to the best results), and thus all *RankSVM*-produced functions consist of linear combinations of the attributes. Although *RankBoost* is composed by simpler weak rankers (defined by single attributes), it achieved results similar to *RankSVM*, since its ensemble strategy also produces a linear combination of the attributes. We conjecture that the superiority of the decision tree based methods is due to their better capability to distinguish candidate tags which are non linearly separable. Besides that, *RF* is a robust method due to the ensemble technique it exploits (bagging) and due to the higher variability of the generated decision trees (produced with random sampling of training data and attributes), which makes it more robust against overfitting.

Comparing the general results across different datasets, we note that the best results are for YahooVideo, while MovieLens (which is considerably smaller than the other datasets) and Bibsonomy present the worst results. The observed differences are

possibly due to the number of tags per object, which tends to be smaller in MovieLens and Bibsonomy than in YahooVideo objects, mas we mentioned above. Moreover, these results may also be due to differences in tagging behavior: on YahooVideo and YouTube, tags tend to appear in other textual features of the same object more often than on LastFM [Belém et al., 2011; Figueiredo et al., 2012] and the other datasets, which facilitates the recommendation of relevant tags exploiting some of the considered tag relevance metrics (e.g., *wTS*).

Regarding efficiency, we found that recommendation time, despite some variation across methods, is under 1.3 seconds[2], on average, for all L2R techniques, in a worst case scenario in which no precomputed data is available in cache. Thus, this recommendation time is reasonably short for an interactive task. Moreover, in terms of total recommendation time (which includes the attribute extraction cost), the use of several of the analyzed L2R strategies only incur a small additional cost (under 3%) in comparison to the best heuristic, *LATRE+wTS*. We note that, in practical terms, the difference in the results between the L2R methods and the best heuristic-based methods is of about one tag, considering the top $k$=5 recommendations. However, we argue that the use of L2R is worth it considering the low additional cost in recommendation time.

In sum, recalling the first topic of our investigation (*RQ1*), we found that (1) L2R based strategies are feasible and can significantly outperform state-of-the-art unsupervised heuristics, and (2) *RF*, *λ-MART* and *MART* are the best L2R strategies out of the eight analyzed techniques, providing further gains over previously evaluated L2R-based strategies.

### 7.1.2 Analysis of Our New Syntactic Attributes

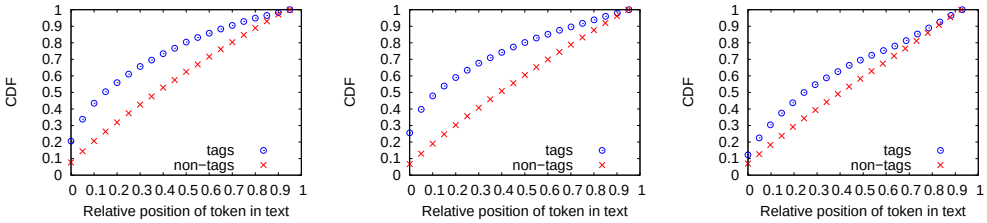
Our goal in this section is to explore several structural properties of texts associated with Web 2.0 objects, such as the relative position of words that are used as tags and various syntactic properties. We aim at identifying those properties that can better distinguish between words that have been assigned as tags to the objects from other words (here referred to as *non-tags*), thus identifying new evidence of potentially good candidate tags.

Specifically, we focus on the object's *description*, using the term "tag" to refer to a word, extracted from the object's description, which has also been assigned as a tag to it. We measure the relative position of tags in the description, and we analyze syntactic patterns of each sentence separately. For each sentence, we first build a corre-

[2]All experiments were performed on a 16-core 2.40GHz Intel(R) Xeon processor, with 50GB RAM.

sponding syntactic dependency tree, labeling each token with their PoS and syntactic functions. We note that statistics for trivially irrelevant tokens, such as punctuation and stopwords, were disregarded from this analysis.

Only some of the aforementioned characteristics have been exploited in keyword extraction/tag recommendation (i.e., PoS and relative position [Hulth, 2003]). The other properties, such as the syntactic function and the path (in the syntactic tree) between a word and the root of the corresponding sentence, have not been analyzed or exploited in tag recommendation yet.




(a) Bibsonomy (b) LastFM (c) MovieLens

**Figure 7.1.** Distribution of the relative position of tags and non-tags in the description's text.

**Table 7.4.** PoS label of tags and non-tags (%).

| | **Bibsonomy** | | **LastFM** | | **MovieLens** | |
|---|---|---|---|---|---|---|
| **PoS** | tags | non-tags | tags | non-tags | tags | non-tags |
| *noun* | 78.5 | 41.7 | 70.3 | 28.2 | 73.5 | 32.0 |
| *adjective* | 15.3 | 12.7 | 20.3 | 9.0 | 16.2 | 12.3 |
| *verb* | 5.5 | 13.9 | 6.4 | 14.7 | 8.5 | 18.0 |
| *other* | 0.7 | 31.7 | 3.0 | 48.1 | 1.8 | 37.7 |

**Table 7.5.** Syntactic function of tags and non-tags (%).

| | **Bibsonomy** | | **LastFM** | | **MovieLens** | |
|---|---|---|---|---|---|---|
| **Syntactic function** | tags | non-tags | tags | non-tags | tags | non-tags |
| *object of a preposition* | 27.2 | 14.6 | 15.6 | 17.3 | 27.0 | 14.5 |
| *compound nouns* | 24.8 | 15.2 | 32.3 | 16.9 | 19.5 | 16.8 |
| *adjectival modifier* | 16.2 | 12.6 | 20.6 | 8.0 | 16.3 | 12.2 |
| *direct object* | 8.5 | 5.8 | 6.1 | 5.6 | 10.6 | 7.2 |
| *nominal subject* | 6.2 | 5.3 | 2.9 | 7.0 | 7.6 | 7.0 |
| *conjunction* | 5.5 | 5.6 | 6.8 | 7.8 | 3.4 | 5.3 |
| *nominal passive subj.* | 2.8 | 1.6 | 0.8 | 1.0 | 0.6 | 0.7 |
| *other* | 8.8 | 39.3 | 14.9 | 36.4 | 14.9 | 36.3 |

Figure 7.1 shows the cumulative probability distribution of the relative position of the first occurrence of a token (tag or non-tag) in LastFM descriptions. The relative position of a token $w$ in a description $T$ is defined as $B(w,T)/|T|$, where $B(w,T)$ is the

number of tokens that appear before $w$ in $T$ and $|T|$ is the total number of tokens in $T$. If a token appears multiple times in $T$, we consider only the relative position of its first occurrence. In all three datasets, tags have a notably higher tendency to appear in the beginning of the object's description when compared to non-tags. For example, 48% of the LastFM tags that appear in the object's descriptions are located in the 10% initial part of the text, while only 19% of the non-tags are located within the same interval. This occurs because the most important words in a text (potentially good tags) tend to be introduced earlier (e.g., in the introductory sentences or paragraphs).

Now we turn our attention to the classes of words (PoS labels) that are more often used as tags. Table 7.4 shows the percentages of tags and non-tags that are classified as each PoS label. We note that the vast majority of tags (ranging from 70% up to 79%) are composed by *nouns*, followed by *adjectives* (15-20%) and *verbs* (5-9%). The distribution among non-tags is quite different, with *other* PoS labels covering a large fraction of the words (almost half in LastFM). Despite such differences, we found that the probability of a word being used as tag given its PoS label is relatively small, falling between 5% and 9% in our datasets. Thus, despite the differences in the PoS label distributions, the use of this probability solely may not be a very strong evidence to identify tags among all words. Thus, it is necessary to consider other features as well.

With that in mind, we also analyze the syntactic function (e.g., direct object, nominal subject) of tags and non-tags. Table 7.5 shows the percentages of tags and non-tags with each syntactic function. We note that the syntactic function of tags is concentrated in a few functions (although this concentration is smoother when compared to the PoS distribution). The most common functions are *object of a preposition*, *compound noun*, *adjectival modifier* and *direct object*. These syntactic functions are consistent with the aforementioned PoS labels (e.g., nouns for objects and nominal subjects, adjective for adjectival modifiers). We note that, for all datasets, tags are more often part of the sentence's object than they are part of the nominal subject. This is particularly noticeable in LastFM, where many nominal subjects are artist names, which do not carry any new information about the artist, and thus are not very useful tags.

Finally, in Table 7.6, we also analyze frequent patterns of syntactic dependencies between tags and the root of the sentences where they appear. We aim at looking not only at the syntactic properties of isolated words, but also at the "connections among words" in the syntactic tree. Specifically, we compute the probability of a word being used as tag, given the path from the given word to the root of the dependence tree where it appears. Table 3 shows examples of frequent patterns and their probabilities

**Table 7.6.** Examples of frequent paths between a tag and the root of the sentence.

| **Bibsonomy** | | **LastFM** | | **MovieLens** | |
|---|---|---|---|---|---|
| *path* | Pr(*tag*\|*path*) | *path* | Pr(*tag*\|*path*) | *path* | Pr(*tag*\|*path*) |
| programming - used | 1.00 | jazz - pianist - is | 1.00 | adaptation - directs | 0.29 |
| programming - applied | 1.00 | blues - singer - was | 1.00 | based | 0.23 |
| neuroscience - literature | 1.00 | death - metal - band - is | 0.96 | comedy - in - stars | 0.20 |
| games - are | 1.00 | heavy - band - was | 0.93 | adapted | 0.18 |
| diploma - thesis - from | 1.00 | rock - singer - is | 0.92 | drama - follows | 0.14 |
| analysis - on - papers | 1.00 | rock - musician - is | 0.92 | way - find | 0.10 |
| book - constitutes | 0.96 | heavy - metal - band - is | 0.91 | thriller - stars | 0.10 |

in all datasets.

We note that the most frequent patterns in LastFM correspond to sentences in the form "X is/was/are/were a Y band/artist/etc", where Y is a tag that characterizes the given artist, usually defining its music genre or style. Such phrases generate paths in the form "Y - band/artist/etc - is/was/are/were" in the dependency trees. In MovieLens, the most frequent patterns are usually related to the movie genre, or specifies that the movie was based on a book/novel/etc. In Bibsonomy, they specify the type of publication (book, diploma thesis, paper, etc) and its main subject (neuroscience, programming, etc). The high values of probabilities Pr($tag|path$), when compared to the other analyzed characteristics (e.g., PoS labels and syntactic functions) reveals a good potential for these probabilities to be used as new attributes for tag recommendation, specially for small texts, in which statistical properties of words such as TF may not be discriminative enough.

### 7.1.3 Cold Start Scenario Evaluation

In this section, we evaluate our solutions in the specific cold start scenario. Our goal is to evaluate the benefits of the inclusion of our syntactic attributes and of exploiting the neighborhood of the target object in this scenario.

We have already shown, in Section 7.1.2, that indeed, the distributions of the investigated syntactic properties are essentially different for tags and non-tags. Next, we compare the effectiveness of tag recommendation methods in the cold start scenario, starting from the baselines in Section 7.1.3. After that, we compare results of state-of-the-art tag recommenders with and without the inclusion of our new syntactic attributes 7.1.3. We estimate the relative importance of all tag quality attributes in Section 7.1.3. Finally, in Section 7.1.3, we show results of our neighborhood expansion approach.

Before showing the results for the cold start scenario, it is worth mentioning that we also tested the effectiveness of the syntactic attributes in a non cold start scenario (that is, when some initial tags are available). Towards this goal, we compared results

**Table 7.7.** Average P@5, R@5 and NDCG@5 results and 95% confidence intervals. Best results and statistical ties in bold.

| | **MovieLens** | **LastFM** | **Bibsonomy** |
|---|---|---|---|
| | | P@5 | |
| *CTTR* | 0.170 ± 0.005 | 0.282 ± 0.003 | 0.299 ± 0.002 |
| *KNN* | 0.189 ± 0.009 | 0.407 ± 0.006 | 0.358 ± 0.002 |
| *PoS + TFIDF* | 0.226 ± 0.006 | 0.292 ± 0.005 | 0.246 ± 0.002 |
| *RankSVM* | 0.176 ± 0.012 | 0.299 ± 0.048 | 0.367 ± 0.003 |
| *RF* | 0.260 ± 0.006 | 0.400 ± 0.003 | 0.379 ± 0.002 |
| $RankSVM_{synt}$ | 0.241 ± 0.010 | 0.331 ± 0.002 | 0.369 ± 0.003 |
| $RF_{synt}$ | 0.303 ± 0.006 | 0.413 ± 0.004 | 0.380 ± 0.002 |
| $KNN_{synt}$ | 0.283 ± 0.007 | 0.420 ± 0.005 | 0.367 ± 0.003 |
| $RF_{synt} + KNN_{synt}$ | **0.314 ± 0.006** | **0.430 ± 0.005** | **0.389 ± 0.003** |
| | | Recall@5 | |
| *CTTR* | 0.123 ± 0.004 | 0.225 ± 0.003 | 0.292 ± 0.002 |
| *KNN* | 0.095 ± 0.006 | 0.232 ± 0.002 | 0.308 ± 0.002 |
| *PoS + TFIDF* | 0.112 ± 0.003 | 0.163 ± 0.003 | 0.214 ± 0.001 |
| *RankSVM* | 0.085 ± 0.008 | 0.168 ± 0.030 | 0.318 ± 0.002 |
| *RF* | 0.129 ± 0.003 | 0.232 ± 0.002 | 0.328 ± 0.002 |
| $RankSVM_{synt}$ | 0.118 ± 0.007 | 0.182 ± 0.001 | 0.319 ± 0.002 |
| $RF_{synt}$ | 0.155 ± 0.004 | 0.239 ± 0.002 | 0.329 ± 0.002 |
| $KNN_{synt}$ | 0.147 ± 0.008 | 0.245 ± 0.003 | 0.317 ± 0.002 |
| $RF_{synt} + KNN_{synt}$ | **0.162 ± 0.007** | **0.250 ± 0.003** | **0.336 ± 0.002** |
| | | NDCG@5 | |
| *CTTR* | 0.167 ± 0.004 | 0.278 ± 0.004 | 0.259 ± 0.002 |
| *KNN* | 0.197 ± 0.011 | 0.443 ± 0.006 | 0.363 ± 0.003 |
| *PoS + TFIDF* | 0.236 ± 0.007 | 0.335 ± 0.006 | 0.261 ± 0.002 |
| *RankSVM* | 0.182 ± 0.017 | 0.322 ± 0.049 | 0.367 ± 0.003 |
| *RF* | 0.273 ± 0.008 | 0.440 ± 0.003 | 0.379 ± 0.002 |
| $RankSVM_{synt}$ | 0.249 ± 0.011 | 0.358 ± 0.003 | 0.369 ± 0.003 |
| $RF_{synt}$ | 0.320 ± 0.006 | 0.454 ± 0.004 | 0.381 ± 0.003 |
| $KNN_{synt}$ | 0.303 ± 0.007 | 0.449 ± 0.006 | 0.365 ± 0.003 |
| $RF_{synt} + KNN_{synt}$ | **0.329 ± 0.007** | **0.464 ± 0.005** | **0.387 ± 0.003** |

of the best L2R strategy with the whole set of attributes, except the syntactic ones ($RF$) with the same strategy including these attributes ($RF_{synt}$), both exploiting co-occurrences with the available tags. We found that results for $RF$ and $RF_{synt}$ in this specific scenario are statistically tied, for all datasets and evaluation metrics. This occurs because tag co-occurrences with the initial tags provide strong candidate tags and tag quality evidence, reducing the need for complementary evidence such as our syntactic attributes to distinguish relevant from non relevant candidates. However, for the cold start scenario, when these co-occurrences cannot be exploited, our syntactic attributes provide clear benefits, as we will present next.

### Effectiveness of the Baselines in Cold Start

Table 7.7 shows average P@5, Recall@5 and NDCG@5 for all methods in the three datasets, along with corresponding 95% confidence intervals. We start our analysis comparing the results of the baselines (first five rows in Table 7.7, for each dataset and evaluation metric). First, we turn our attention to the non-supervised approaches, each one focused on different evidence of tag quality, namely, CTTR and *KNN*. In general, *KNN* is the strongest non-supervised method, with gains of up to 44% in P@5 over CTTR. This is probably because *KNN* takes tags (originated from similar objects) as candidates, as opposed to the other strategy, which exploit other textual features of the target object, which are noisier (carry a larger number of irrelevant terms) than tags [Figueiredo et al., 2012], besides co-occurrences of these words with tags.

Out of all baselines, the strategy with the best overall performance is the *RF* based approach, with gains of up to 15% in P@5, 5% in Recall@5 and 15% in NDCG@5 over the second best baseline, which varies according to each dataset (*KNN* in MovieLens, PoS+TFIDF in LastFM and RankSVM in Bibsonomy). In comparison with PoS+TFIDF, *RF* results present 58%, 56% and 49% higher P@5, Recall@5 and NDCG@5, respectively. PoS+TFIDF is limited to candidate tags extracted from the target object's description, and exploit only word frequency and PoS labels. In comparison with CTTR, *RF* produces gains of up to 55%, 15% and 66% in P@5, Recall@5 and NDCG@5, respectively, because *RF* exploits a larger set of candidate tags (not only extracted from co-occurrences and from the textual features of the target object, as performed by CTTR, but also from similar objects). Moreover, *RF* exploits more tag quality attributes, and automatically combines them using an L2R technique. CTTR, instead, focuses only on frequency statistics of words extracted from the target object and co-occurrences between tags and these words, and does not exploit L2R.

### Adding new Attributes Related to Syntactic Properties

Now we compare $RF_{synt}$ and $RankSVM_{synt}$ results with those produced by the baselines, in order to assess the benefits of including our new syntactic attributes. We note that $RF_{synt}$ outperforms *RF* in two of our datasets (MovieLens and LastFM), with gains of up to 16% in P@5, 20% in Recall@5 and 17% in NDCG@5. The same conclusions hold for the comparison between $RankSVM_{synt}$ and *RankSVM*. The former outperforms the latter with gains of up to 37% in P@5, 38% in Recall@5 and 37% in NDCG@5.

This attests the capacity of our new syntactic structure tag quality attributes to improve tag recommendation, specially in a cold start scenario in which tag co-

occurrences with previously assigned tags (in the target object) cannot be exploited. Another characteristic of the studied datasets, particularly MovieLens and LastFM, is that the (user-generated) descriptions are usually short and may present low quality, making it difficult to rank candidates solely by statistical properties of words such as TF and IDF, as performed by *RF* or *RankSVM*. On the other hand, even short and low quality texts may present some syntactic properties that can be used as evidence to generate and rank candidate tags, favoring $RF_{synt}$ and $RankSVM_{synt}$.

In the Bibsonomy dataset, $RF_{synt}$ and *RF* results are statistically tied (as well as $RankSVM_{synt}$ and RankSVM), probably because the descriptions (abstracts of publications) tend to present higher quality (compared to MovieLens and LastFM objects). These descriptions usually present an adequate size, and the most important keywords of the text tend to re-appear in the different textual features (title, abstract) [Figueiredo et al., 2012]. Thus, word statistics such as TF and wTF are effective in this dataset, making the new proposed attributes less essential to discriminate tags from other words. Comparing the results for MovieLens and LastFM datasets, gains in MovieLens are considerably higher, mainly due to the fact that MovieLens descriptions are usually shorter than LastFM's, and consist of movie synopsis, which tend to hide part of the plot. These characteristics make the use of syntactic properties more important to identify and rank tags in MovieLens.

Comparing our best method against PoS+TFIDF (which is the only baseline that exploits a syntactic attribute (word's PoS), $RF_{synt}$ greatly outperforms PoS+TFIDF, with gains of up to 55% in P@5, 57% in Recall@5 and 46% in NDCG@5, in all datasets. This is due to two main factors: (1) $RF_{synt}$ exploits other 11 syntactic patterns that are not exploited by PoS+TFIDF, and (2) $RF_{synt}$ extracts and rank candidate tags not only from the target object's description (as performed by PoS+TFIDF), but also from similar objects and from tags that co-occur (in training data) with words in the other textual features of the target object, making $RF_{synt}$ a more robust and complete method.

Thus, we found that our syntactic attributes are responsible for significant improvements in at least two datasets, in which statistical properties of the candidate tags, in isolation, cannot discriminate relevant from non-relevant candidates.

### Attribute Importance Analysis

In this section, we estimate the importance of all tag quality attributes exploited in the cold start scenario, by $RF_{synt}$ and $RankSVM_{synt}$. Our goal is to compare the usefulness of our new proposed attributes with relation to the other attributes, as well

as to determine a smaller set of the best, non redundant attributes.

We performed attribute importance analysis in two different ways: calculating the Information Gain (IG) [Baeza-Yates and Ribeiro-Neto, 1999] of each attribute, and the absolute values of their corresponding weights (averaged over the 5 folds) in the model generated by $RankSVM_{synt}$. The top-10 most discriminative attributes according to these measures are respectively listed in Tables 7.8 and 7.9, normalized so that they sum up 1.

**Table 7.8.** Top-10 tag quality attributes ranked according to Information Gain.

| **MovieLens** | | **LastFM** | | **Bibsonomy** | |
|---|---|---|---|---|---|
| $S_{WordToTag}$ | 0.27 | $S_{WordToTag}$ | 0.30 | *TermScore* | 0.38 |
| *TermScore* | 0.22 | *TermScore* | 0.26 | *Entropy* | 0.10 |
| *Entropy* | 0.10 | *Pred* | 0.18 | *wTF* | 0.09 |
| *Relative position* | 0.10 | *Entropy* | 0.06 | *Pred* | 0.08 |
| *Pred* | 0.09 | *wTF* | 0.05 | $S_{WordToTag}$ | 0.08 |
| *IFF* | 0.06 | *Relative position* | 0.04 | $S_{TitleToTag}$ | 0.08 |
| *Stability* | 0.05 | *IFF* | 0.04 | *IFF* | 0.05 |
| *Sentence root* | 0.05 | *Stability* | 0.03 | *Stability* | 0.05 |
| *Seq. synt. funct.* | 0.03 | *Token's head* | 0.02 | *Relative position* | 0.05 |
| *Token's head* | 0.03 | *Seq. synt. funct.* | 0.02 | *Seq. synt. funct.* | 0.03 |

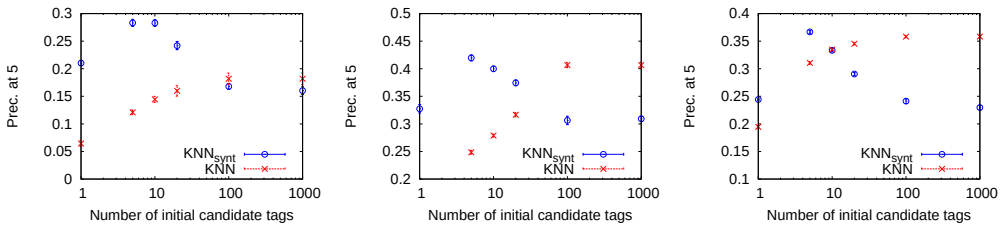
**Table 7.9.** Top-10 tag quality attributes ranked according to SVM weights.

| **MovieLens** | | **LastFM** | | **Bibsonomy** | |
|---|---|---|---|---|---|
| *TermScore* | 0.19 | *TermScore* | 0.36 | $S_{TitleToTag}$ | 0.14 |
| $S_{TitleToTag}$ | 0.15 | *PoS* | 0.14 | *PoS* | 0.12 |
| *Seq. of tokens* | 0.14 | $S_{TitleToTag}$ | 0.12 | *TermScore* | 0.11 |
| *Synt. funct.* | 0.13 | *TS* | 0.05 | *Synt. funct.* | 0.11 |
| *PoS* | 0.11 | *wTS* | 0.05 | $S_{WordToTag}$ | 0.09 |
| *PoS of token's head* | 0.06 | $S_{WordToTag}$ | 0.05 | *PoS of token's head* | 0.08 |
| *Synt. funct. of token's head* | 0.05 | *Token's head* | 0.04 | *Token* | 0.07 |
| $S_{WordToTag}$ | 0.03 | *Seq. synt. funct.* | 0.03 | *Seq. synt. funct.* | 0.07 |
| *TS* | 0.03 | *PoS of token's head* | 0.03 | *Token's head* | 0.05 |
| *wTS* | 0.03 | *Token* | 0.03 | *Synt. funct. of token's head* | 0.04 |

According to IG values (Table 7.8) , we note that at least 1 (up to 3) syntactic structure related attributes appear among the top 10 list of all applications, confirming that they are indeed useful for tag recommendation purposes. Moreover, traditional, word frequency based attributes such as TF do not appear in Table 7.8 as they were ranked below our new attributes. This occurs mainly due to the fact that, as discussed above, the user-generated texts in Web 2.0 applications are usually short and may present low quality. This issue is more noticeable in LastFM (where the title is simply an artist/band name) and MovieLens (where the movie title and description may be vague and hide part of the movie plot). In Bibsonomy, the descriptions (abstracts) tend to present a higher quality, as discussed above. Although the exact order of attributes differ when considering SVM weights, we can obtain similar conclusions from Table

7.9. Even a larger number of our syntactic related attributes appear among the top attributes with larger SVM weights. These results are consistent with those shown in Section 7.1.3.

In all applications, the attributes with highest IG values are the ones related to a graph-based tag recommendation approach (*TermScore*) and word co-occurrence with tags ($S_{WordToTag}$), because they represent key attributes for tag recommendation in cold start. Although these attributes present the highest IG values, they may not be discriminative enough in isolation, as noted from the various baseline results. Thus, clearly some new syntactic related attributes do bring significant improvements for recommendation. Among the syntactic structure related attributes, the most discriminative attributes, according to the IG metric, are: (1) the sequence of syntactic functions that form a path between the candidate tag and the root in the syntactic dependence tree ("Seq. synt. funct."), (2) the root of the sentence that contains the candidate tag ("Sentence root"), and (3) the token that is the head of the candidate tag in the tree ("Token's head").




(a) Movielens (b) LastFM (c) Bibsonomy

**Figure 7.2.** P@5 results for *KNN* and $KNN_{synt}$ as a function of the number of top initial recommendations exploited by these methods.

## Neighborhood Expansion

Now we analyze to which extent we can further improve tag recommendation using a k-nearest neighbors based technique. Unlike the traditional *KNN* baseline, which computes the neighborhood of the target object using TFIDF weights of the words in its textual features, $KNN_{synt}$ exploits the scores provided by $RF_{synt}$ (the best tag recommender analyzed above) as weights.

Comparing $KNN_{synt}$ with the traditional *KNN*, we note that $KNN_{synt}$ produces large improvements in MovieLens (50%, 55% and 54% in P@5, Recall@5 and NDCG@5, respectively). For LastFM, the gains are modest (3%, 5% and 1.4% in P@5, Recall@5 and NDCG@5, respectively). For Bibsonomy, there are even more modest gains (or statistical ties). To investigate this difference in $KNN_{synt}$ gains among the datasets, we

measured the effectiveness of a simple tag recommender based on TFIDF only, which is the measure used by *KNN* to represent objects and compute their similarity. We note that TFIDF recommendations for MovieLens present considerably lower precision than the recommendations obtained for the other two datasets (57% and 375% lower precision with relation to LastFM and Bibsonomy, respectively), because of the differences in the characteristics of the descriptions across datasets, as aforementioned. Thus, the generation of a better object representation with stronger initial candidate tags is much more necessary in MovieLens than in the other two datasets. In spite of these cases of lower gains and statistical ties, as we shall see below, the combination of $KNN_{synt}$ with $RF_{synt}$ is robust enough to provide gains in all datasets.

Now we analyze the impact of the tuning parameters in the effectiveness of the neighborhood based tag recommenders. We have analyzed the impact of the number of neighbors ($K$) in Section 6.3, which affected both $KNN_{synt}$ and *KNN* similarly. Another important parameter for the nearest neighbors based tag recommenders is the number of initial terms to consider in the target object representation. As we mentioned in Section 5.3.2, the content of the textual features of an object may contain noise, and thus it is useful to consider only the most relevant terms (here referred to as initial candidate tags) to represent objects and compute their similarity. Here, we evaluate two alternative approaches: the traditional, based on TFIDF (*KNN*), and our new approach, based on the score provided by $RF_{synt}$ ($KNN_{synt}$). Figure 7.2 shows P@5 results for different values of the number of initial recommended candidate tags ($r$) for both methods and three datasets. As expected, when $r$ is too low ($r$=1), both methods present their lowest precision. However, as we increase $r$, *KNN* and $KNN_{synt}$ present different behaviours: for $KNN_{synt}$, the best results occur when $r$=5 and their precision decreases as we increase $r$. However, the opposite occurs for *KNN*, increasing precision as we increase $r$, until convergence (using all initial candidate tags as object representation). This occurs because TFIDF, in isolation, could not rank all representative terms of the target object among the top positions, and thus it was necessary to use all terms to reach better results. On the other hand, $KNN_{synt}$ is able to better select the most representative candidates, achieving good results using only a few initial tag candidates. However, including too many initial tag candidates in $KNN_{synt}$ starts to be detrimental to its effectiveness, probably because the initial candidates provided by $RF_{synt}$ are not restricted to the target object, and thus may bring some noise. In spite of it, $KNN_{synt}$ reaches better results than *KNN*, specially for the MovieLens dataset, as aforementioned.

Now, we turn our attention to our final, most complete approach to address cold start, $KNN_{synt} + RF_{synt}$. We find that it consistently produces the best results in

this scenario. It outperforms the best baseline with gains of up to 21% in P@5, 26% in Recall@5 and 21% in NDCG@5 (e.g., in MovieLens dataset). For the other two datasets, the gains are more modest (around 3% in all considered evaluation metrics), due to the aforementioned reasons. Although the gains are modest in two datasets, we note that this method presents little additional cost with relation to $RF_{synt}$: it consists in adding up the results of $KNN_{synt}$, which is a straightforward neighborhood based approach, and $RF_{synt}$. In turn, $RF_{synt}$ offers a low additional cost with relation to the baseline $RF$, because the computational cost of our new syntactic attributes is inferior to the cost to compute various of the attributes exploited by $RF$, such as tag co-occurrence and neighborhood based attributes.

Thus, we found that it is worth exploiting the neighborhood of the target object to achieve further (though modest) gains over the best strategy found in the previous section ($RF_{synt}$).

### 7.1.4 Personalized Tag Recommendation Results

We now discuss the most relevant results of our new personalized tag recommendation methods (4 heuristics and 3 L2R-based strategies), comparing them against the $PITF$ baseline. Table 7.10 shows $P$@5 results for all personalized methods and datasets. Recall@5 and NDCG@5 results are shown in Tables 7.11 and 7.12, respectively.

Once again, all reported results are averages over 5 folds (test sets), whereas the results of GP and RF are averages over 25 runs (5 folds, 5 random generator seeds). Tables 7.10-7.12 show 95% confidence intervals, indicating that, with that confidence, most results deviate from the means by less than 2%. For each dataset, the tables are broken into 3 blocks: baseline, new heuristics and L2R-based methods. Best results and statistical ties (according to a 2-sided t-test[3] with $p$-value $< 0.05$) within each block are shown as shaded entries. Best overall results (and statistical ties) are shown in bold. Recall that we do not show results of the personalized methods for YahooVideo as our dataset of this application does not identify the user who assigned each tag.

#### Unsupervised Heuristics

We start by comparing our new heuristics against the baseline PITF. We found that our best heuristic considering overall results ($LATRE$+$wTS$+$UF$) produces gains in P@5 ranging from 48% to 251%, and in Recall@5 and NDCG@5 of up to 255% and

[3]Like for object-centered tag recommendation, we also applied the Bonferroni correction on the results of personalized tag recommendation.

**Table 7.10.** Personalized tag recommendation: average P@5 results and 95% confidence intervals (best results within each block - baselines, heuristics, and L2R-based strategies - in shaded entries; best overall results in bold).

| Strategy | Bibsonomy | LastFM | MovieLens | YouTube |
|---|---|---|---|---|
| *PITF* | 0.332 ± 0.003 | 0.528 ± 0.002 | 0.424 ± 0.003 | 0.145 ± 0.002 |
| $Sum_u^+ wTS{+}UF$ | 0.525 ± 0.001 | 0.633 ± 0.003 | 0.545 ± 0.010 | 0.525 ± 0.002 |
| $Sum^+ wTS{+}UF$ | 0.523 ± 0.002 | 0.488 ± 0.003 | 0.439 ± 0.009 | 0.525 ± 0.002 |
| $LATRE_u{+}wTS{+}UF$ | 0.536 ± 0.002 | 0.633 ± 0.004 | 0.562 ± 0.008 | 0.507 ± 0.002 |
| $LATRE{+}wTS{+}UF$ | 0.548 ± 0.002 | 0.781 ± 0.003 | 0.374 ± 0.011 | 0.507 ± 0.002 |
| *GP-based* | 0.542 ± 0.006 | 0.812 ± 0.004 | 0.563 ± 0.009 | 0.535 ± 0.004 |
| *RankSVM-based* | 0.559 ± 0.001 | 0.707 ± 0.003 | 0.557 ± 0.005 | 0.544 ± 0.001 |
| *RF-based* | **0.601 ± 0.001** | **0.840 ± 0.001** | **0.588 ± 0.010** | **0.572 ± 0.002** |

**Table 7.11.** Personalized tag recommendation: average Recall@5 results and 95% confidence intervals (best results within each block - baselines, heuristics, and L2R-based strategies - in shaded entries; best overall results in bold).

| Strategy | Bibsonomy | LastFM | MovieLens | YouTube |
|---|---|---|---|---|
| *PITF* | 0.329 ± 0.003 | 0.520 ± 0.002 | 0.423 ± 0.003 | 0.132 ± 0.002 |
| $Sum_u^+ wTS{+}UF$ | 0.519 ± 0.001 | 0.617 ± 0.003 | 0.542 ± 0.010 | 0.483 ± 0.001 |
| $Sum^+ wTS{+}UF$ | 0.517 ± 0.002 | 0.473 ± 0.003 | 0.438 ± 0.009 | 0.483 ± 0.001 |
| $LATRE_u{+}wTS{+}UF$ | 0.530 ± 0.003 | 0.616 ± 0.004 | 0.560 ± 0.008 | 0.468 ± 0.002 |
| $LATRE{+}wTS{+}UF$ | 0.542 ± 0.002 | 0.760 ± 0.002 | 0.372 ± 0.011 | 0.468 ± 0.002 |
| *GP-based* | 0.534 ± 0.006 | 0.789 ± 0.004 | 0.561 ± 0.009 | 0.491 ± 0.003 |
| *RankSVM-based* | 0.552 ± 0.001 | 0.688 ± 0.003 | 0.554 ± 0.005 | 0.499 ± 0.001 |
| *RF-based* | **0.592 ± 0.002** | **0.816 ± 0.001** | **0.585 ± 0.010** | **0.525 ± 0.001** |

295%, respectively. Average gains across all datasets are 121% (P@5), 122% (Recall@5) and 157% (NDCG@5). Thus, using a combination of tag co-occurrences, multiple textual features and metrics of relevance, including a metric that captures the tagging history of the user ($UF$), can greatly outperform recommendation methods that are based only on the interrelationships between users, objects and tags, like PITF.

The effectiveness of PITF is particularly poor in YouTube due to the non-collaborative nature of the application, where the user who uploaded the object is the only who can assign tags to it. This characteristic makes training data sparser, limiting the benefits of PITF, which depends on a sufficient amount of postings involving a user $u$ and an object $o$ to recommend relevant tags for the pair $\langle$u, o$\rangle$. Nevertheless, we note that even in collaborative tagging applications, such as Bibsonomy, LastFM and MovieLens the gains of our heuristics over PITF are very large. For example, in Bibsonomy, $LATRE{+}wTS{+}UF$ outperforms PITF by as much as 65% in P@5 (see Table 7.10).

We note that, like PITF, our methods also exploit the vocabulary of the target user, expressed by the tags assigned by her to other objects, as a representation of her interests and main evidence to support personalization. We argue that it is not unlikely that the same user may assign tags to similar objects as these objects better match the user interests and vocabulary. Thus, as our results confirm, it may be interesting

to recommend tags that the user had already assigned to other objects.

**Table 7.12.** Personalized tag recommendation: average NDCG@5 results and 95% confidence intervals (best results within each block - baselines, heuristics, and L2R-based strategies - in shaded entries; best overall results in bold).

| Strategy | Bibsonomy | LastFM | MovieLens | YouTube |
|---|---|---|---|---|
| *PITF* | 0.256 ± 0.002 | 0.412 ± 0.002 | 0.298 ± 0.003 | 0.127 ± 0.001 |
| $Sum_u^+wTS{+}UF$ | 0.471 ± 0.002 | 0.603 ± 0.004 | 0.503 ± 0.009 | 0.525 ± 0.002 |
| $Sum^+wTS{+}UF$ | 0.469 ± 0.002 | 0.464 ± 0.004 | 0.384 ± 0.007 | 0.525 ± 0.002 |
| $LATRE_u{+}wTS{+}UF$ | 0.487 ± 0.002 | 0.605 ± 0.004 | 0.520 ± 0.008 | 0.503 ± 0.002 |
| $LATRE{+}wTS{+}UF$ | 0.501 ± 0.001 | 0.747 ± 0.003 | 0.308 ± 0.010 | 0.503 ± 0.002 |
| *GP-based* | 0.506 ± 0.004 | 0.809 ± 0.004 | 0.521 ± 0.009 | 0.541 ± 0.003 |
| *RankSVM-based* | 0.512 ± 0.002 | 0.665 ± 0.004 | 0.525 ± 0.004 | 0.538 ± 0.001 |
| *RF-based* | **0.553 ± 0.002** | **0.828 ± 0.002** | **0.555 ± 0.011** | **0.578 ± 0.002** |

Next, we compare our four proposed heuristics, focusing first on the two different types of tag co-occurrence patterns exploited by them: (1) between tags assigned to the same object by various users (exploited by $Sum^+wTS{+}UF$ and $LATRE{+}wTS{+}UF$), and (2) between tags assigned by the same user to the same object (used by $Sum_u^+wTS{+}UF$ and $LATRE_u{+}wTS{+}UF$). In YouTube, these two kinds of co-occurrence patterns lead to the same results, since only one user can assign tags to an object. In the other three applications, interestingly, the most effective type of co-occurrence pattern depends on the co-occurrence based method exploited by the recommendation strategy ($Sum^+$ or $LATRE$). On the one hand, if the recommendation is based on $Sum^+$, which exploits relationships between only 2 tags, type (2) is preferred as $Sum_u^+wTS{+}UF$ produces results that are, if not statistically tied, much better than those produced by $Sum^+wTS{+}UF$. For example, the improvements in P@5 reach 30% in the LastFM dataset. This occurs due to the larger amount of noise generated when co-occurrences between all tags in an object are considered. On the other hand, exploiting co-occurrences between tags assigned to the same object by various users may benefit $LATRE$, which exploits more complex association rules (i.e., co-occurrences between more than 2 tags), being more resilient to noise. For example, the improvements in P@5 of $LATRE{+}wTS{+}UF$ over $LATRE_u{+}wTS{+}UF$ vary from 2% up to 23% in Bibsonomy and LastFM datasets. The exception is the MovieLens dataset, where the best alternative is $LATRE_u{+}wTS{+}UF$, which presents 34% higher precision than $LATRE{+}wTS{+}UF$. This exception is possibly due to (1) the much smaller size of MovieLens dataset (and thus smaller training data for association rule mining) and (2) the higher divergence of tags that different users apply to the same object in MovieLens dataset may cause the generation of noisier co-occurrence patterns when considering co-occurrences between tags posted by different users. Considering all objects annotated by at least two users, only 25% of the tags were assigned

by more than 1 user to the same object in MovieLens, against 34% in LastFM and 28% in Bibsonomy. The same conclusions hold for the other three evaluation metrics considered[4].

Consistently with the results of the object-centered recommendation methods that they extend, we find that $LATRE{+}wTS{+}UF$ (or $LATRE_u{+}wTS{+}UF$ in MovieLens case) outperforms $Sum_u^+wTS{+}UF$ in all datasets but YouTube. For example, $LATRE{+}wTS{+}UF$ outperforms $Sum_u^+wTS{+}UF$ with gains of 3% in P@5, on average, in both LastFM and Bibsonomy, while experiencing only a small loss (less than 1%) in YouTube. In MovieLens, $LATRE_u{+}wTS{+}UF$ also outperforms $Sum_u^+wTS{+}UF$ by 3% in P@5. Similarly, the average gains produced by $LATRE{+}wTS{+}UF$ on LastFM and Bibsonomy (and by $LATRE_u{+}wTS{+}UF$ in MovieLens) are 4% in Recall@5 and NDCG@5, respectively, whereas the losses in YouTube do not exceed 1.1%. Thus, $LATRE{+}wTS{+}UF$ and $LATRE_u{+}wTS{+}UF$ are the best heuristic for personalized tag recommendation.

### Learning-to-Rank based Strategies

Like observed for object-centered tag recommendation, all three evaluated L2R-based methods provide further improvements over the heuristics for personalized tag recommendation, in all datasets, although the RF-based strategy is clearly the best performer. For instance, the improvements in P@5 achieved with the RF-based strategy over the best heuristic (that is, $LATRE_u{+}wTS{+}UF$ for MovieLens and $LATRE{+}wTS{+}UF$ for the other datasets) are 9%, on average, across all datasets. Similarly, average gains in Recall@5 and NDCG@5 are 8.4% and 11%, respectively. Moreover, the RF-based strategy consistently outperforms the best of the other two L2R-based strategies in around 5%, on average, in any of the considered metrics. These results confirm the benefits of exploiting Random Forest as an L2R approach for tag recommendation, and the resilience of our methods when applied to both object-centered and personalized tag recommendation tasks, as discussed in Section 7.1.4.

Overall, an important factor that explains the success of our personalized methods (both heuristics and L2R-based methods) is that, as previously mentioned, they can provide relevant recommendations for a user even if she does not present a history of tag assignments. In that case, the extraction of candidates from tag co-occurrences and multiple textual features provide more general recommendations for the considered object, which can be relevant to any user. If the user is more active, however, our

[4]We also compared the purely co-occurrence based methods $Sum^+$ and $LATRE$ with $Sum_u^+$ and $LATRE_u$, respectively, obtaining the same conclusions.

methods can provide a higher level of personalization, due to the use of the $UF$ metric. In other words, our methods are flexible and robust to deal with both object-centered and personalized tag recommendation tasks. In particular, the $RF$-based strategy has shown to be the the most effective solution for both tag recommendation tasks.

### 7.1.5 Benefits of Personalization in Tag Recommendation

In this section, we quantitatively compare our best object-centered and personalized methods under similar conditions, in order to attest if personalized tag recommendations might provide better descriptions of the object when compared to object-centered recommendations, thus improving services that rely on those descriptions, such as search and content recommendation. In other words, we intend to assess whether our user-related tag quality attributes promote globally relevant tags, i.e., tags that can be relevant to some user (not necessarily the target one).

Specifically, we compare the results produced by the RF-based object-centered and personalized tag recommendation methods for each user against the same expected answer. In other words, for each target object-user pair $\langle o, u \rangle$, we use the same input tags $I_o$ to feed both methods, and compare their results against the same expected answer $Y_o$. To guarantee a fair comparison of both methods, we build these two tag sets such that each one contains half of the tags posted by each user who assigned tags to $o$ (randomly selected). Note that this setup is different from the ones used in Sections 7.1.1 and 7.1.4. In the former, the tags of the object were randomly split into $I_o$ and $Y_o$, with no consideration to the user(s) who posted them. In the latter, $I_o$ consisted of half of the tags posted by the target user $u$ and all tags posted by any other user, and the recommended tags were compared against the other tags posted by $u$ ($Y_{o,u}$). Thus, the results presented here are different from those discussed in the two previous sections. In particular, unlike in the experiments discussed in Section 7.1.4, we here compare the tags recommended by the personalized method for a user $u$ with *all* tags that were not used as input (i.e., all tags in $Y_o$), and not only those posted by $u$ ($Y_{o,u}$). This is because, unlike in the previous section, our goal here is to assess the relevance of the suggested tags to the target object only (regardless of their relevance to the target user). Note also that, for a given object $o$, the object-centered method produces the same results to all users.

Precision, recall and NDCG of both methods are shown in Table 7.13 for the Bibsonomy, LastFM, MovieLens and YouTube datasets. Note that the personalized strategy produces results that significantly outperform the object-centered method. The average gain in p@5 is 15% across the four datasets, while corresponding gains

in recall@5 and NDCG@5 are 17% and 16%, respectively. That is, having fixed the expected answer, the personalized recommendations match this expected answer more closely than the object-centered recommendations. These results are in alignment with observations in [Rendle et al., 2009a; Rendle and Schmidt-Thie, 2010], which showed that their personalized tag recommenders outperform even the theoretical upper-bound for any non-personalized tag recommender.

**Table 7.13.** Relevance of our RF-based object-centered and personalized tag recommendations to the target object: average results and 95% confidence intervals (best results for each datast in bold).

| Application | Method | P@5 | recall@5 | NDCG@5 |
|---|---|---|---|---|
| **Bibsonomy** | Object-centered | 0.550 ± 0.002 | 0.535 ± 0.002 | 0.506 ± 0.003 |
| | Personalized | **0.576 ± 0.001** | **0.562 ± 0.001** | **0.533 ± 0.002** |
| **LastFM** | Object-centered | 0.595 ± 0.002 | 0.211 ± 0.001 | 0.602 ± 0.001 |
| | Personalized | **0.713 ± 0.002** | **0.285 ± 0.002** | **0.728 ± 0.002** |
| **MovieLens** | Object-centered | 0.392 ± 0.005 | 0.221 ± 0.004 | 0.414 ± 0.004 |
| | Personalized | **0.505 ± 0.007** | **0.270 ± 0.003** | **0.545 ± 0.008** |
| **YouTube** | Object-centered | 0.543 ± 0.002 | 0.498 ± 0.002 | 0.553 ± 0.002 |
| | Personalized | **0.572 ± 0.002** | **0.525 ± 0.001** | **0.578 ± 0.002** |

Thus, these results are evidence that personalization may improve the quality of the tag recommendations, providing tags that not only might be more important to the target user, and thus to other users with similar interests and profiles, but also that cover the different facets of the object, allowing a more complete description of the content than object-centered recommendations.

## 7.2 Relevance, Novelty and Diversity Driven Methods

In this section, we present results of our proposed methods that address relevance, novelty and diversity aspects. Our main research question we aim at answering here is:

*RQ4: Can we improve novelty and diversity of tag recommendation, while keeping the same levels of relevance?*

First we analyze results of our implicit method $GP_{rnd}$ (Section 7.2.1), then we discuss the results of the three explicit methods, namely, $RF_t$, *xTReD* and *xTReND* (Section 7.2.2).

**Table 7.14.** Average results and 95% confidence intervals. Best results and statistical ties in bold.

| *Collection* | *Method* | *NDCG@5* | *AIP@5* | *AILD@5* | *α-NDCG@5* | *S-Recall@5* |
|---|---|---|---|---|---|---|
| Bibsonomy | *GP* | **0.406 ± 0.006** | 0.539 ± 0.007 | 0.954 ± 0.004 | **0.589 ± 0.011** | **0.782 ± 0.007** |
| | $GP_{rnd}$ | 0.404 ± 0.005 | **0.663 ± 0.011** | **0.976 ± 0.002** | 0.493 ± 0.008 | 0.714 ± 0.007 |
| LastFM | *GP* | **0.440 ± 0.008** | 0.282 ± 0.005 | 0.845 ± 0.004 | 0.362 ± 0.008 | 0.503 ± 0.008 |
| | $GP_{rnd}$ | **0.442 ± 0.005** | **0.345 ± 0.016** | **0.882 ± 0.007** | **0.386 ± 0.009** | **0.537 ± 0.013** |
| MovieLens | *GP* | **0.388 ± 0.002** | **0.504 ± 0.007** | **0.942 ± 0.001** | **0.282 ± 0.009** | **0.439 ± 0.013** |
| | $GP_{rnd}$ | 0.363 ± 0.005 | **0.517 ± 0.018** | **0.943 ± 0.005** | 0.258 ± 0.008 | 0.404 ± 0.015 |
| YahooVideo | *GP* | **0.770 ± 0.004** | 0.434 ± 0.004 | 0.903 ± 0.004 | **0.496 ± 0.008** | **0.561 ± 0.005** |
| | $GP_{rnd}$ | 0.759 ± 0.006 | **0.483 ± 0.007** | **0.926 ± 0.003** | **0.501 ± 0.008** | 0.554 ± 0.005 |
| YouTube | *GP* | **0.530 ± 0.002** | 0.608 ± 0.002 | **0.974 ± 0.001** | **0.743 ± 0.002** | **0.951 ± 0.001** |
| | $GP_{rnd}$ | 0.520 ± 0.003 | **0.659 ± 0.004** | **0.975 ± 0.001** | 0.711 ± 0.003 | 0.938 ± 0.002 |

### 7.2.1 Implicit Method

In this Section, we aim to answer the following question: How does our new solution $GP_{rnd}$ perform compared to the state-of-the-art relevance-driven method *GP*?

Table 7.14 shows results for relevance (*NDCG*), novelty (*AIP*), implicit diversity (*AILD*) and explicit diversity (*α-NDCG* and *S-Recall*[5]), all evaluated on the top $k$=5 positions of the ranking. The explicit diversity results for LastFM, MovieLens and YouTube datasets are based on explicit categories, while Bibsonomy and YahooVideo results are based on latent topics produced by LDA technique. We also evaluated LastFM, MovieLens and YouTube datasets using latent topics, obtaining similar results.

Comparing our new strategy $GP_{rnd}$ with the method it extends (*GP*), we obtained gains in *AIP* (novelty) of 23% in Bibsonomy, 22% in LastFM, 2.5% in MovieLens, 8.5% in YouTube and 11% in YahooVideo, losing at most 2% in *NDCG* in most datasets (except in MovieLens, which presented a 6% loss in NDCG). Thus, it is possible to obtain novel recommendations while maintaining similar levels of relevance with the new proposed objective function, although relevance and novelty may be conflicting objectives. However, it is more difficult to improve implicit diversity, since the AILD results are already very high in *GP*. In fact, our gains are below 4.5%. This happens because the data is sparse, making the values of distance between tags typically large, with small differences between them, given that there is little information about tag co-occurrences (the source of information for *AILD* to estimate the semantic differences between tags).

We find that the $RF_{rnd}$ results for the explicit diversity evaluation metric vary from modest gains of 7%, 6% and 7% in *α-NDCG*, *ERR-IA* and *S-Recall*, respectively (in LastFM dataset) to losses of up to 16%, 17% and 9% in these metrics. One possible

[5] *ERR-IA* results are similar to the other explicit diversity metrics, thus we omitted them to improve readability.

reason for the inexistence of improvements in explicit diversity in most datasets is that *AILD* metric promotes infrequent tags, which tend to be more dissimilar to any tag since there is little information about their co-occurrence. On the other hand, these infrequent tags do not carry information about the (more general) topics they are related, thus providing low explicit diversity improvements, if any.

In sum, we found that $RF_{rnd}$ provides reasonable gains in novelty without significantly harming relevance, but the gains in diversity (both implicit and explicit) are modest or inexistent. In the next section, we analyze methods that exploit diversity explicitly, maximizing the topic coverage of the tag recommendations.

### 7.2.2 Explicit Methods

Now we turn our attention to our new methods that exploit explicit (topic) diversity, namely, *xTReD*, $RF_t$ and *xTReND*, comparing them against the best relevance-driven alternative found in the previous sections (*RF*). More specifically, we aim at answering the following questions, derived from *RQ4*:

*RQ4.1: Do our new topic related attributes contribute to produce better tag recommenders?*

*RQ4.2: How do our new solutions $RF_t$, xTReD and xTReND perform compared to each other and to the best relevance-driven method (i.e., RF)?*

*RQ4.3: Is the use of latent topics a viable alternative to our solutions when the target application does not possess an explicit category system to organize content?*

*RQ4.4: To which extent can we effectively promote novelty and explicit diversity without harming relevance in tag recommendation?*

We address *RQ*4.1 by comparing our new $RF_t$ method, which incorporates diversity and novelty at the attribute level, with the relevance-driven *RF* method. We tackle *RQ*4.2 by comparing our new methods $RF_t$, which captures all three aspects in the attribute level, while focusing on relevance in its objective, *xTReD*, that captures explicit diversity and relevance, but do not try to optimize novelty in its objective function, and *xTReND*, which fully captures all three aspects at both attribute and objective levels. As *RQ*4.3 covers an orthogonal/transversal aspect concerning all previous questions, we tackle it in the context of each individual comparison, analyzing results for all previous questions with explicit categories *and* latent topics. All these comparisons, which cover various datasets, are presented in Sections 7.2.2.1-7.2.2.3. We then tackle *RQ*4.4 by exploring the trade-off among relevance, novelty and diversity in Section 7.2.3.

Tables 7.15 and 7.16 show average NDCG (relevance), AIP (novelty), $\alpha$-NDCG,

*ERR-IA* and *S-Recall* (diversity) results for all methods and datasets for two evaluation scenarios: (1) using the predefined categories available in the datasets as topics, and (2) exploiting latent topics. These results were computed over the top $k$=5 recommended tags, and produced with all methods parameterized according to the best parameter values obtained in the *validation* set (as shown in Tables 6.2-6.4). Note that Table 7.15 shows results only for the three datasets where predefined categories are available (namely, LastFM, MovieLens and YouTube).

**Table 7.15.** Relevance, novelty and diversity of the top $k$=5 recommended tags by all methods (best average results and statistical ties according to a two-sided t-test with $p < 0.05$ are shown in bold). Evaluation scenario: Using pre-defined categories as topics.

| | Method | NDCG | AIP | $\alpha$-NDCG | ERR-IA | S-Recall |
|---|---|---|---|---|---|---|
| LastFM | *RF* | 0.483 ± 0.003 | 0.325 ± 0.003 | 0.404 ± 0.006 | 0.365 ± 0.005 | 0.583 ± 0.009 |
| | $RF_t$ | **0.508 ± 0.005** | 0.328 ± 0.003 | 0.546 ± 0.011 | 0.492 ± 0.009 | 0.738 ± 0.014 |
| | *xTReD* | 0.472 ± 0.003 | 0.353 ± 0.001 | 0.579 ± 0.008 | **0.532 ± 0.006** | 0.729 ± 0.011 |
| | *xTReND* | 0.504 ± 0.004 | **0.365 ± 0.003** | **0.591 ± 0.011** | **0.530 ± 0.009** | **0.780 ± 0.014** |
| MovieLens | *RF* | 0.415 ± 0.005 | 0.515 ± 0.005 | 0.272 ± 0.010 | 0.220 ± 0.009 | 0.446 ± 0.011 |
| | $RF_t$ | **0.428 ± 0.004** | 0.509 ± 0.004 | 0.354 ± 0.019 | 0.285 ± 0.016 | 0.559 ± 0.017 |
| | *xTReD* | 0.409 ± 0.005 | 0.523 ± 0.004 | 0.383 ± 0.006 | 0.316 ± 0.007 | 0.575 ± 0.014 |
| | *xTReND* | 0.415 ± 0.005 | **0.593 ± 0.005** | **0.437 ± 0.010** | **0.352 ± 0.010** | **0.664 ± 0.013** |
| YouTube | *RF* | **0.553 ± 0.002** | 0.610 ± 0.001 | 0.749 ± 0.003 | 0.717 ± 0.003 | 0.949 ± 0.001 |
| | $RF_t$ | **0.555 ± 0.002** | 0.610 ± 0.001 | 0.798 ± 0.002 | 0.761 ± 0.002 | 0.973 ± 0.001 |
| | *xTReD* | 0.535 ± 0.002 | 0.607 ± 0.001 | **0.838 ± 0.002** | **0.813 ± 0.003** | 0.980 ± 0.001 |
| | *xTReND* | 0.536 ± 0.002 | **0.651 ± 0.001** | **0.837 ± 0.002** | 0.807 ± 0.002 | **0.985 ± 0.001** |

In the following, we discuss these results by focusing first on how our new $RF_t$ method compares against our best relevance-driven method *RF*. These two methods have the same relevance-driven objective function and differ at the attribute level: $RF_t$ adds new topic-related attributes capturing explicit diversity. This discussion, which tackles $RQ4.1$, is presented in Section 7.2.2.1. We then approach $RQ4.2$ by comparing our new methods *xTReND* and *xTReD* in Section 7.2.2.2, and comparing *xTReND* and $RF_t$, in Section 7.2.2.3. The treatment of $RQ4.3$ perpasses all those analyses.

#### 7.2.2.1 Do Topic Related Attributes Contribute to Producing Better Tag Recommendations?

We start by comparing the RF-based strategies, whose objective functions are focused on relevance only[6]. Considering the use of categories as source of topics, Table 7.15 shows that our new $RF_t$ strategy greatly outperforms *RF* strategy in terms of all three diversity metrics in all datasets. The improvements in $\alpha$-NDCG, *ERR-IA* and

[6]Although $RF_t$ also exploits diversity attributes, its objective function is based only on relevance.

**Table 7.16.** Relevance, novelty and diversity of the top $k$=5 recommended tags by all methods (best average results and statistical ties according to a two-sided t-test with $p < 0.05$ are shown in bold). Evaluation scenario: Using latent topics (LDA).

| | Method | NDCG | AIP | $\alpha$-NDCG | ERR-IA | S-Recall |
|---|---|---|---|---|---|---|
| Bibsonomy | *RF* | **0.455 ± 0.003** | 0.554 ± 0.001 | 0.574 ± 0.004 | 0.479 ± 0.004 | 0.781 ± 0.003 |
| | $RF_t$ | **0.455 ± 0.003** | 0.555 ± 0.001 | 0.580 ± 0.005 | 0.482 ± 0.004 | 0.789 ± 0.004 |
| | *xTReD* | 0.443 ± 0.001 | 0.553 ± 0.001 | 0.668 ± 0.003 | **0.561 ± 0.003** | 0.878 ± 0.002 |
| | *xTReND* | 0.444 ± 0.001 | **0.569 ± 0.001** | **0.673 ± 0.004** | **0.564 ± 0.004** | **0.883 ± 0.003** |
| LastFM | *RF* | 0.483 ± 0.001 | 0.325 ± 0.001 | 0.570 ± 0.007 | 0.535 ± 0.008 | 0.807 ± 0.006 |
| | $RF_t$ | **0.490 ± 0.001** | 0.324 ± 0.001 | 0.587 ± 0.008 | 0.552 ± 0.008 | 0.824 ± 0.007 |
| | *xTReD* | 0.468 ± 0.001 | 0.326 ± 0.001 | 0.716 ± 0.006 | 0.678 ± 0.006 | 0.956 ± 0.003 |
| | *xTReND* | 0.473 ± 0.002 | **0.333 ± 0.001** | **0.725 ± 0.007** | **0.688 ± 0.007** | **0.960 ± 0.003** |
| MovieLens | *RF* | 0.415 ± 0.002 | 0.515 ± 0.002 | 0.452 ± 0.009 | 0.378 ± 0.009 | 0.656 ± 0.010 |
| | $RF_t$ | **0.426 ± 0.003** | 0.506 ± 0.001 | 0.431 ± 0.012 | 0.381 ± 0.010 | 0.653 ± 0.016 |
| | *xTReD* | 0.411 ± 0.002 | 0.515 ± 0.001 | **0.570 ± 0.007** | **0.472 ± 0.007** | **0.816 ± 0.008** |
| | *xTReND* | 0.418 ± 0.003 | **0.531 ± 0.001** | 0.542 ± 0.014 | **0.482 ± 0.012** | 0.799 ± 0.014 |
| YahooVideo | *RF* | **0.809 ± 0.001** | 0.433 ± 0.001 | 0.509 ± 0.001 | 0.341 ± 0.002 | 0.586 ± 0.001 |
| | $RF_t$ | **0.810 ± 0.001** | 0.433 ± 0.001 | 0.507 ± 0.001 | 0.339 ± 0.002 | 0.585 ± 0.002 |
| | *xTReD* | 0.788 ± 0.001 | 0.439 ± 0.001 | 0.561 ± 0.001 | **0.382 ± 0.002** | 0.623 ± 0.001 |
| | *xTReND* | 0.779 ± 0.002 | **0.463 ± 0.001** | **0.568 ± 0.005** | **0.385 ± 0.006** | **0.628 ± 0.004** |
| YouTube | *RF* | **0.553 ± 0.002** | **0.610 ± 0.001** | 0.556 ± 0.005 | 0.507 ± 0.005 | 0.834 ± 0.004 |
| | $RF_t$ | **0.553 ± 0.002** | **0.610 ± 0.001** | 0.556 ± 0.005 | 0.507 ± 0.005 | 0.834 ± 0.004 |
| | *xTReD* | 0.539 ± 0.001 | 0.608 ± 0.001 | **0.691 ± 0.002** | **0.645 ± 0.003** | **0.955 ± 0.001** |
| | *xTReND* | 0.540 ± 0.001 | **0.609 ± 0.001** | 0.684 ± 0.004 | 0.638 ± 0.004 | **0.952 ± 0.002** |

*S-Recall* reach up to 35%, 35%, and 28%, respectively. Corresponding *average* gains, computed across all datasets, are 24%, 23% and 18%, respectively. We note that the increases in all three diversity metrics are smaller on YouTube, because objects in this dataset (videos) are associated with only one category, which reduces the room for improvements from the use of topic related attributes.

Although both strategies have relevance as the only objective to be maximized, $RF_t$ obtains such great improvements in diversity over *RF* by exploiting attributes that promote tags that are highly related to the topics of the target object, and thus have higher chances to cover these topics. Such gains in diversity are accompanied by some (more modest) improvements also in terms of relevance of the recommendations: the average NDCG of $RF_t$ is up to 5% higher. We note that these gains come with no significant additional cost since the new topic related attributes are easy to compute. Indeed, all probabilities required to compute these attributes can be calculated offline.

Regarding novelty of the recommendations, the average AIP results of both *RF* and $RF_t$ are statistically tied, except in the MovieLens dataset, although the difference is under 2% in this case. One possible explanation for the slightly smaller average AIP obtained with $RF_t$ in this dataset is that MovieLens genres are semantically broader

than the categories of the other datasets. As a consequence, tags in this dataset that are more related to the topics (and thus are promoted by $RF_t$) tend to be more general and occur more often, thus having lower AIP, if compared to tags in the other datasets.

If LDA topics are used (Table 7.16), the gains of $RF_t$ in relevance and diversity over RF are much more modest (if any), probably because the topics are generated in an unsupervised way, exploiting only the previously assigned tags. Some of these tags might be too general (such as "seen" and "based") or even too noisy (i.e., unrelated to the object's content), and thus might not be very appropriate for topic inference. Yet, we do observe some statistically significant improvements in average NDCG (e.g., in MovieLens) as well as in each diversity metric (e.g., LastFM and Bibsonomy). Such improvements reach 3% in average NDCG and in average $\alpha$-NDCG.

#### 7.2.2.2 Is *xTReND* effective when compared to *xTReD*?

We now compare *xTReND*, our new diversifier with novelty promotion, with our *xTReD* diversifier. In common, they address relevance and diversity at the objective function level, although only *xTReND* directly exploits popularity based novelty. Moreover, only *xTReND* includes the new topic related attributes.

Table 7.15 shows that, when categories are used as topics, *xTReND* outperforms *xTReD* with gains in AIP (novelty) of 8% on average, and maximum gains of 13%. The corresponding gains when LDA topics are used are 3% and 6%, respectively, according to Table 7.16. These gains are due to the promotion of tags with higher *IFF*. The surprising aspect is that such gains are achieved with no harm to diversity or relevance in most cases. Indeed, our results show that, for most datasets and scenarios, *xTReND* produces at least the same diversity as *xTReD*, while in some cases there are large improvements.

Note, for instance, the increase in 14% of average $\alpha$-NDCG if categories are used, and in 2% of average ERR-IA when LDA topics are used, both in the MovieLens dataset. Indeed, if categories are used as source of topics, the diversity of the results produced by *xTReND* is at least as good as that of *xTReD*, although often better, *in all cases*, which indicates that it is possible to promote more specific tags that also are highly related to the topics of the object. The few exceptions when the novelty promoted by *xTReND* hurts diversity occur when LDA topics are used (e.g., $\alpha$-NDCG and S-Recall on MovieLens). However, the differences in such cases are under 5%, and are probably due to the higher focus given by *xTReD* to diversity when compared to *xTReND*, which also promotes novelty.

Similarly, we note that the improvements in average AIP (novelty) obtained with

*xTReND* over *xTReD* also come with no detrimental impact on relevance. Instead, we do observe some significant improvements in average NDCG, with gains reaching 7% (e.g., LastFM when categories are used).

#### 7.2.2.3 How does *xTReND* compare to $RF_t$?

We now compare $RF_t$, which captures relevance, novelty and diversity aspects at the attribute level only, with *xTReND*, which addresses all three aspects at both attribute and objective function levels.

Table 7.15 shows that, if categories are used as the source of topics, *xTReND* outperforms $RF_t$ in terms of both diversity and novelty with gains of 11% in average AIP, 12% in average $\alpha$-NDCG, 12% in average *ERR-IA* and 9% in average *S-Recall*, all computed on average across all datasets. The maximum improvements on these metrics on any dataset reach 16%, 24%, 24% and 19%, respectively.

According to Table 7.16, the results are similar if LDA topics are used: the gains in average AIP, $\alpha$-NDCG, *ERR-IA*, and *S-Recall* are, on average, 3.4%, 20%, 22%, 14%, respectively, reaching 7%, 26%, 26% and 22% (also respectively). We note that such gains come with only a small impact (if any) on relevance: compared to $RF_t$, the average NDCG results produced by *xTReND* is at most 4% lower. Thus, it is possible to provide further gains in diversity and novelty by exploiting these aspects at the objective function level.

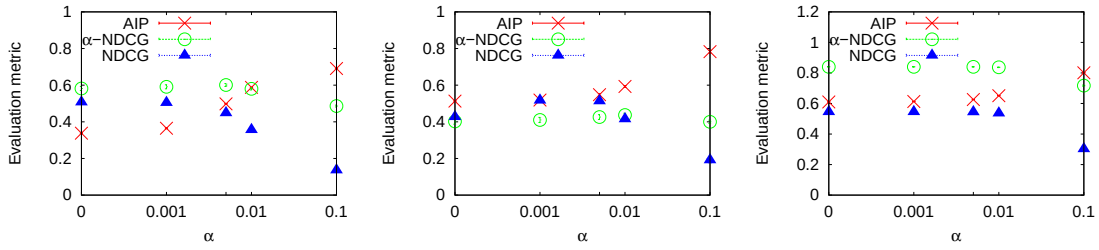
### 7.2.3 Trade-offs Among Relevance, Novelty and Diversity

Finally, we tackle research question *RQ*4.4, and analyze the trade-offs among relevance, novelty and diversity by quantifying how each aspect is affected as we favor one over the others. Ultimately, we want to assess the extent to which one can improve novelty and/or diversity without significantly hurting relevance. To this end, we focus on our best method, *xTReND*, which explicitly captures all three aspects, and analyze its sensitivity to parameters $\alpha$ and $\beta$, the weights given to novelty and diversity, respectively.

We vary $\alpha$ and $\beta$ in the same ranges of values used for parameterizing the method (see Section 6.3), and evaluate the relevance, diversity and novelty of the recommendations produced by *xTReND* in the *test sets*. We perform experiments in each evaluation scenario and, when using latent topics, we also analyze the impact of varying the number of topics $n_Z$ (see discussion below). As we vary $\alpha$ (or $\beta$) we compare the results produced by *xTReND* with: (1) the results produced by *xTReND* when the parameter being varied is set to 0 but all other parameters are fixed at their best values (as shown

in Table 6.2), and (2) the results obtained when $\alpha$=$\beta$=0, that is, the results produced by $RF_t$. The first comparison allows us to assess whether favoring one factor impacts the other compared to the case when the latter is maximized (i.e., corresponding weight is set at the best value). The second comparison allows us to assess the extent to which relevance is degraded as we favor novelty or diversity since, as shown in Tables 7.15 and 7.16, $RF_t$ produces the best results in terms of relevance in all datasets.

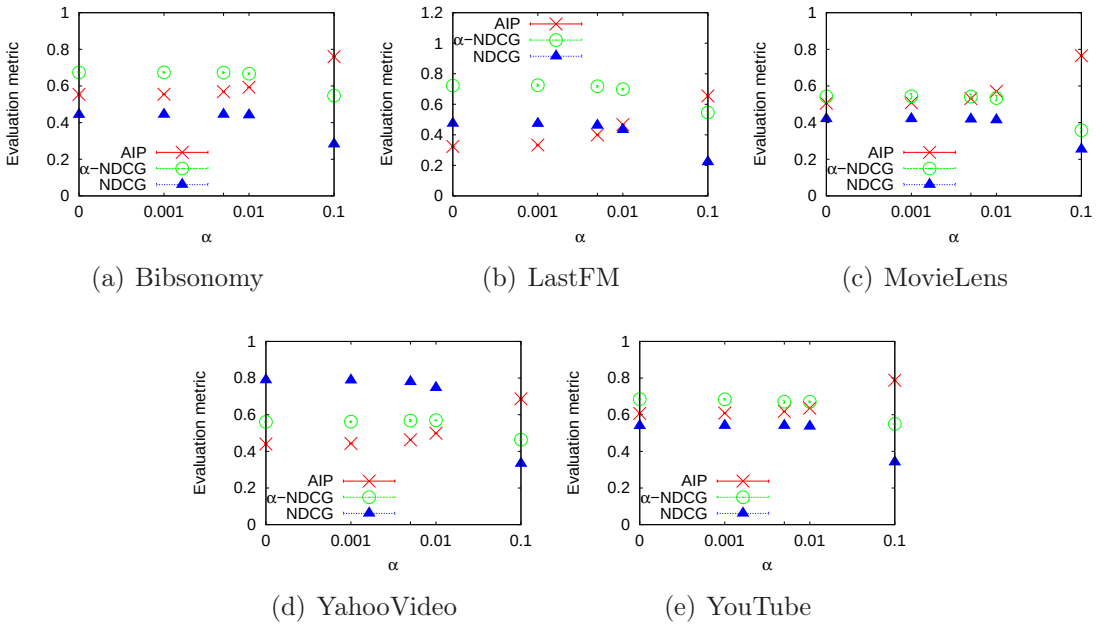
Figures 7.3-7.4 show the impact of parameter $\alpha$ on the average AIP (novelty), NDCG (relevance) and $\alpha$-NDCG (diversity) results in both evaluation scenarios, while Figures 7.5-7.6 show the impact of parameter $\beta$ on the same metrics. All figures show the impact of one parameter when all other parameters are kept fixed at their best values. Results for the other diversity evaluation metrics are similar to those of $\alpha$-NDCG, and thus are omitted. We note that Figures 7.3 and 7.5 show results only for the three datasets where predefined categories are available (namely, LastFM, MovieLens and YouTube). We also note that all figures report average results computed over the top $k$=5 recommended tags, along with corresponding 95% confidence intervals, although some intervals are not visible as they are smaller than the symbols used.




(a) LastFM (b) MovieLens (c) YouTube

**Figure 7.3.** Impact of varying parameter $\alpha$ on average NDCG (relevance), AIP (novelty) and $\alpha$-NDCG (diversity), computed over the top $k$=5 recommended tags. Evaluation scenario: Using pre-defined categories as topics.

Focusing first on the impact of $\alpha$, Figures 7.3-7.4 show that average AIP results always increase as we increase the values of $\alpha$, which is expected. However, values of $\alpha$ beyond a certain threshold, which depends on the dataset, are harmful to both relevance and diversity. Such large values of $\alpha$ lead to recommending very rare tags, which may be noisy and usually present low information about the topics they belong to. For example, setting $\alpha$ to the maximum value tested ($\alpha = 0.1$) causes an increase in average AIP of as much as 105%. However, such improvements come at the cost of a decrease in average NDCG, compared to the initial scenario of $\alpha$=0 (and $\beta$ set to the best choice for each dataset), which varies from 36% to 73%. Similarly, the drop in average $\alpha$-NDCG (diversity) varies from 3% to 35%. Compared to the results produced

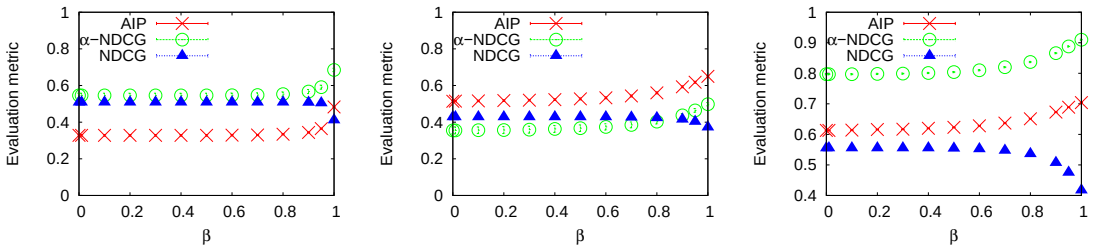



(a) Bibsonomy (b) LastFM (c) MovieLens

(d) YahooVideo (e) YouTube

**Figure 7.4.** Impact of varying parameter $\alpha$ on average NDCG (relevance), AIP (novelty) and $\alpha$-NDCG (diversity), computed over the top $k$=5 recommended tags. Evaluation scenario: Using latent topics (LDA).

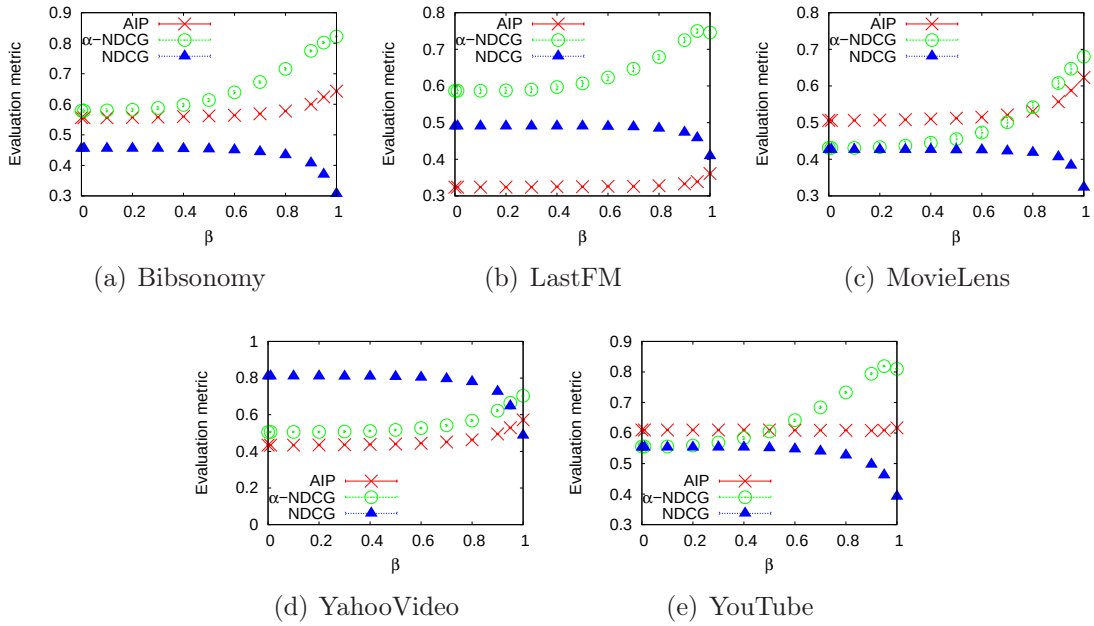
by $RF_t$ (i.e., $\alpha = \beta = 0$), the increase in average AIP is even higher (up to 111%) but so is the decrease in average NDCG, which varies from 52% to 74%. Similarly, the reduction in average $\alpha$-NDCG varies from 1% to 17%.




(a) LastFM (b) MovieLens (c) YouTube

**Figure 7.5.** Impact of varying parameter $\beta$ on average NDCG (relevance), AIP (novelty) and $\alpha$-NDCG (diversity), computed over the top $k$=5 recommended tags. Evaluation scenario: Using pre-defined categories as topics.

We now turn to the impact of $\beta$ on the results. Figures 7.5-7.6 show that, as expected, the average $\alpha$-NDCG results (diversity) always increase with $\beta$ but so do the average AIP results (novelty). This indicates that tags that are highly related to the topics of the target object (diversity) also present a good level of specificity (novelty). However, very large values of $\beta$ may hurt relevance by promoting tags related to the



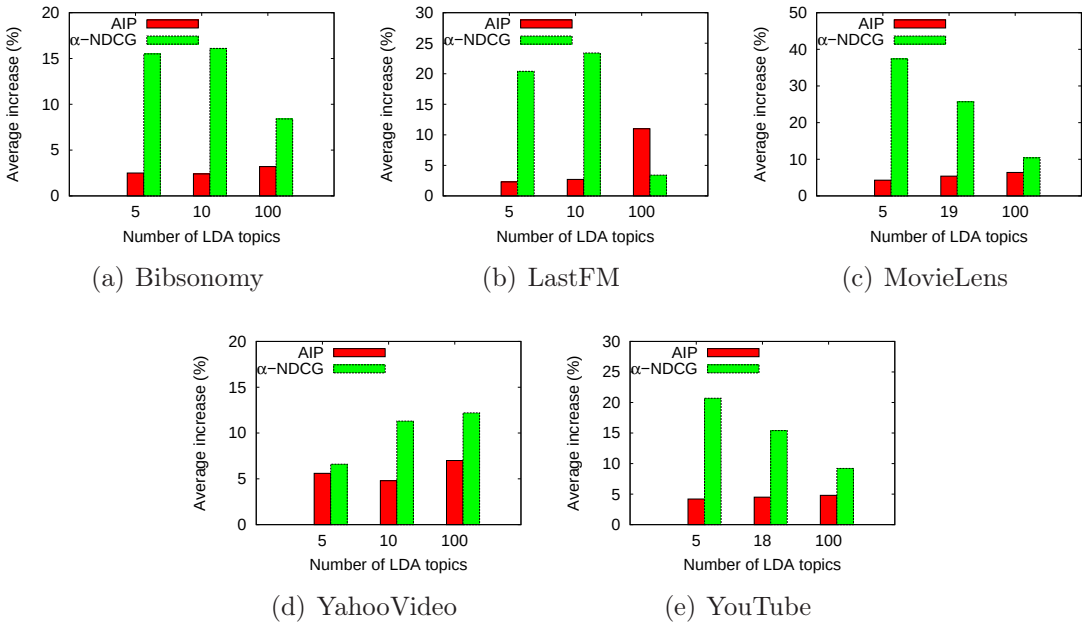

(a) Bibsonomy (b) LastFM (c) MovieLens

(d) YahooVideo (e) YouTube

**Figure 7.6.** Impact of varying parameter $\beta$ on average NDCG (relevance), AIP (novelty) and $\alpha$-NDCG (diversity), computed over the top $k$=5 recommended tags. Evaluation scenario: Using latent topics (LDA).

topics of the target object but less related to the object in particular. For example, compared to the case when $\beta = 0$ and the other parameters are set at their best values, increasing $\beta$ to 1 leads to improvements in average $\alpha$-NDCG and *AIP* of as much as 40% and 47%, respectively. But it also causes a quite dramatic decrease in average NDCG (from 13% to 25%). The differences are even more striking when we compare these results against those produced by $RF_t$: whereas the improvements in average $\alpha$-NDCG and AIP reach 58% and 47%, respectively, the impact in relevance may be quite detrimental, with losses of as much as 40% in average NDCG.

Yet, as discussed in Section 7.2.2.3, it is possible to obtain improvements in both diversity and novelty if we allow a small degradation in relevance (up to $\epsilon$=4%). To further analyze the trade-offs among novelty, diversity and relevance, we here consider a more restrictive scenario when the maximum degradation in average NDCG allowed is only $\epsilon$=1%. Results indicate that *xTReND* is still capable of producing gains in novelty and diversity under such constraints. When categories are exploited, the gains in $\alpha$-NDCG (diversity) reach 8%, 14% and 2% for the LastFM, MovieLens and YouTube datasets, respectively, when compared to results produced by $RF_t$. The corresponding gains in average AIP (novelty) reach 11%, 7% and 3%, respectively. When latent topics are exploited, there are gains of 10%, 10%, 16%, 4% and 13% in average $\alpha$-NDCG for the Bibsonomy, LastFM, MovieLens, YahooVideo and YouTube datasets.




(a) Bibsonomy (b) LastFM (c) MovieLens

(d) YahooVideo (e) YouTube

**Figure 7.7.** Impact of varying the number of LDA topics on average AIP (novelty) and $\alpha$-NDCG (diversity) computed over top $k$=5 recommended tags: average increase over no diversification and novelty promotion.

The corresponding gains in average AIP are 2%, 8%, 3.4%, 3% and 1%. Thus, even if we severely restrict any possible degradation in relevance, *xTReND* can still achieve substantial improvements in diversity and novelty, particularly considering that simultaneously maximizing these three (often conflicting) objectives is quite challenging.

The aforementioned results using LDA topics were obtained by setting the number of latent topics $n_Z$ at the best values obtained in the validation set for each dataset (Table 6.2). The larger the value of $n_Z$, the more specific the generated topics are, which may impact the diversity and the novelty of the recommended tags. Thus, as a final set of experiments, we evaluate how the novelty, diversity and relevance of the results produced by *xTReND* using latent topics is affected as we vary the number of topics used in 5, 10, 100. In these experiments, $\alpha$ and $\beta$ are set at their best values. Figure 7.7 shows improvements in average $\alpha$-NDCG (diversity) and average AIP (novelty)[7] obtained over the initial recommendations produced by $RF_t$ for different numbers of LDA topics. For every dataset, we find that the number of topics does not impact the relevance of the results. That is, average NDCG results are statistically tied across all values of $n_Z$ tested, being thus omitted from the graphs. In contrast, the improvements in average AIP, compared to $RF_t$, slightly increase as

[7]We here analyze diversity and novelty gains, instead of absolute values of the evaluation metrics, because the absolute values are not directly comparable for different values of $n_Z$.

a larger number of topics is used (the improvements increase by as much as 11%). Such increase in average AIP occurs because in order to cover more specific topics the diversifier promotes tags that are probably more specific as well, and thus, with higher $IFF$ values. However, $\alpha$-NDCG gains tend to be higher for a smaller number of topics, as they are are easier to cover compared to when a larger number of more specific topics are used. The exception is YahooVideo, in which the improvements in $\alpha$-NDCG are larger for the higher value of $n_Z$. We conjecture that this might be due to the larger number of collaboratively created tags present in that application, allowing a higher variability of tags and thus latent topics.

### 7.2.4 Summary of the Results of the Explicit Methods

We here summarize our main findings with respect to research questions $RQ4.1$-$RQ4.4$. Our experimental results revealed that:

(1) the use of our new topic related metrics at the attribute level by an L2R-based tag recommendation approach such as RF does contribute to produce better tag recommendations, particularly if predefined categories are used as topics, allowing substantial gains in diversity as well as some (modest) gains in relevance ($RQ4.1$);

(3) *xTReND* provides a better trade-off among the three objectives (relevance, novelty and diversity), being the best alternative between our three new explicit solutions ($RQ4.2$);

(4) though more modest, the improvements of our new methods over the baselines are still significant if LDA topics are used, implying that such unsupervised topic inference strategy can be used to extend the applicability of our solutions to scenarios where predefined categories are not available ($RQ4.3$); and

(5) although relevance, novelty and diversity of recommendations may seem conflicting objectives, it is possible to effectively increase novelty and diversity with only a slight impact on relevance ($RQ4.4$).

## 7.3 Chapter Summary

In this chapter, we presented the experimental results we have obtained to answer the research questions proposed in this thesis. The four main fronts related to the tag recommendation problem we have evaluated are: (1) the combination of tag quality attributes (some of them proposed here) by means of L2R techniques, with focus on maximizing the relevance of the recommended tags; (2) the new tag quality attributes (based on syntactic properties) and techniques (neighborhood expansion) developed to tackle cold start in tag recommendation; (3) the personalization of the proposed methods; and (4) the improvements in novelty and diversity. In the next chapter, we provide a summary of the results in all of these fronts, as well as directions for future work.

# Chapter 8

# Conclusions and Future Work

In this chapter we present a summary of the results of this thesis (Section 8.1) and provide directions for future work (Section 8.2).

## 8.1 Summary of Results

Recall from Chapter 1 the main research questions that drive this study:
*RQ1: How can we improve the relevance of the recommended tags by means of a combination of tag quality attributes?*
*RQ2: How can we generate and rank candidate tags in a cold start scenario in which there are no previously available tags?*
*RQ3: How can we extend the proposed methods to provide personalized recommendations?*
*RQ4: How can we improve novelty and diversity of tag recommendation, while keeping the same levels of relevance?*

The main results obtained in each of these topics are discussed in Sections 8.1.1 to 8.1.4.

### 8.1.1 RQ1 - Combination of Tag Quality Attributes

We combined a number of tag quality attributes by means of heuristics and L2R-based techniques. Some of these attributes and techniques have already been proposed and evaluated in our previous work [Belém et al., 2011]. Other attributes, namely, the topic-related attributes (addressed in RQ4), and syntactic attributes (addressed in RQ2) are novel contributions of this thesis.

The best analyzed L2R-based strategy outperforms the state-of-the-art heuristic, producing gains of up to 29% in average NDCG. Among the L2R based strategies, there is a clear winner group of methods: Random Forests (*RF*), *MART* and $\lambda$-*MART*, which produces gains ranging from 4% to 12% in average NDCG over the best of the remaining L2R-based methods (i.e., the previously proposed methods *GP*, *RankSVM* and *Rankboost*). Furthermore, we found that the L2R approach presents a very low additional recommendation time (under 3%) when compared with the best unsupervised heuristic (LATRE). Besides the promising results, the flexibility of the L2R framework in terms of the incorporation of new attributes and ability to maximize different target measures (as we do here, when adding personalization, novelty and diversity aspects, as well as addressing cold start) makes it an attractive solution for the tag recommendation problem.

It is also worth mentioning that, besides combining tag quality attributes, we also tested combinations of L2R methods by means of a stacking technique, and some straighforward strategies such as summing up the scores given by each method. However, for the same set of attributes, these strategies did not outperform the best L2R based method (RF) in isolation. In general, we note that the greatest gains are achieved when we combine different (and complementary) sources of candidate tags and tag quality attributes.

### 8.1.2 RQ2 - Addressing Cold Start with Syntactic Attributes and Neighborhood Expansion

In this front of work, we proposed syntactic related attributes and nearest neighbor techniques to extend and improve tag recommendation methods in a cold start scenario. We note that these techniques provide much higher gains in this particular scenario than in a scenario in which there are some tags available in the target object, since tag co-occurrences with these initial tags are strong evidence of the quality of a candidate tag, reducing the need for additional attributes.

First, we investigated syntactic patterns of the text associated with Web 2.0 objects that can be exploited to identify and recommend tags. We also proposed new tag quality attributes based on these syntactic patterns, exploiting them to further improve our proposed L2R-based tag recommenders in the given scenario. Our experiments showed that our proposed syntactic attributes are responsible for significant improvements (up to 17% in precision over the best relevance-driven method). A feature importance analysis confirmed that our new attributes are among the most discriminative for the problem in hand, in particular the sequence of syntactic depen-

dencies between the candidate tag and the root of the sentence, the token connected to the candidate tag in the syntactic tree, and the root of the sentence.

Moreover, we also analyzed to which extent we can further improve tag recommendations by exploiting the neighborhood of the target object (i.e., similar objects). We used the L2R-based tag recommender with syntactic attributes to compute a new, complementary neighborhood. Recommendations based on this new neighborhood outperformed those generated from traditional nearest neighbors approaches, which exploit only TFIDF as weights for terms in an object. Finally, $KNN_{synt} + RF_{synt}$, our combination of both neighborhood and L2R-based tag recommenders, consistently produced the best results, with gains of up to 21% over $RF$.

### 8.1.3 RQ3 - Personalization of Tag Recommendation

We proposed four heuristics and evaluated three new L2R-based methods ($RF$, *RankSVM* and $GP$) to address the personalized tag recommendation problem. Furthermore, we have provided a quantitative assessment of the benefits of personalized tag recommendation to provide better descriptions of the target object.

We found that our heuristics produced gains of up to 157% in average NDCG over a state-of-the-art personalized tag recommendation method ($PITF$). Our best L2R method, $RF$, provided average relevance gains of 9% over our best heuristic and gains of 5% over our previous L2R-based strategies (i.e., *RankSVM* and $GP$). Comparing our best personalized and object-centered tag recommendation methods, both based on the RF technique, we found that the former outperforms the latter, with average gains of 15% in relevance. Thus, we found that personalization brings benefits when applied to provide better descriptions of the target object.

### 8.1.4 RQ4 - Improving Novelty and Diversity of Tag Recommendation

We have proposed four new tag recommendation methods aiming at exploiting novelty and diversity, in different levels. Our first method, called $GP_{rnd}$, extends the relevance-driven method $GP$, which already incorporates some novelty aspects at the attribute level, to include novelty and diversity metrics at both attribute and objective function levels. The second method, called $RF_t$, extends the relevance-driven approach based on $RF$ to include new tag attributes that capture the extent to which a candidate tag is related to the topics (e.g., categories) of the target object. This solution indirectly captures topic diversity while trying to maximize relevance in its objective function.

Unlike $RF_t$, our third method, Explicit Tag Recommendation Diversifier (*xTReD*), *directly* exploits topic diversity, by *re-ranking* the recommendations provided by any tag recommender. Finally, our fourth proposal, called Explicit Tag Recommendation Diversifier with Novelty Promotion (*xTReND*), generalizes *xTReD*, to fully exploit relevance, novelty and topic diversity.

Our evaluation showed that $GP_{rnd}$ provides reasonable gains in novelty without significantly harming relevance when compared to $GP$, but the gains in diversity (both implicit and explicit) are modest (at best). We also found that the use of our new topic related metrics at the attribute level (as performed by $RF_t$) does contribute to produce better tag recommendations, particularly if predefined categories are used as topics, allowing substantial gains in diversity (up to 35%) as well as some (modest) gains in relevance (up to 5% in average NDCG), when compared to $RF$. Overall, our new method, *xTReND*, is the best out of the four new methods, considering the trade-offs among relevance, novelty and diversity. Though more modest, the improvements of our new methods over the baselines are still significant if LDA topics are used, implying that such unsupervised topic inference strategy can be used to extend the applicability of our solutions to applications where predefined categories are not available. Finally, although relevance, novelty and diversity of recommendations may seem conflicting objectives, it is possible to effectively increase novelty and diversity with only a slight impact on relevance.

## 8.2 Future Work

In this thesis, we have adopted a fully automatic evaluation methodology to measure relevance, novelty and diversity of our results. As discussed in Section 6.2, regardless of its limitations, this is a well-established and widely adopted evaluation protocol in the area, and allowed us to cover a large number of methods and datasets. In our preliminary experiments with volunteers, we did not obtain a sufficient amount of evaluations to get statistically significant results. Thus, we leave the manual evaluation methodology as future work, performing it by either experiments with external volunteers or with real users of a running tag recommendation system (if possible).

Another possible research direction would be evaluating novelty and diversity in a personalized perspective. For example, we can define novelty in the context of specific users, such that a tag is novel for a given user $u$ if it was not previously assigned by him to other objects. However, preliminary experiments with approaches that exploit user-related novelty produced no significant improvements on novelty without harm-

ing relevance. This occurs because the user-related novelty and the relevance metric that estimates the user interests are conflicting pieces of evidence of the quality of a tag. Regardless of this result, further investigations could be performed, for example, identifying the topics of interest of the target user and diversifying personalized recommendations considering these topics.

Another interesting line of future work is in improvements in the learning techniques (as opposed to our current focus on tag quality attribute engineering). This can be performed, for example, by including *boosting* to the methods that originally do not exploit this technique (e.g., GP and RF). Another interesting topic related to machine learning that is worth studying in our problem is the *transfer learning* from models learned in a given dataset and applied to another.

# Bibliography


Abdi, H. (2007). *Bonferroni and Sidak corrections for multiple comparisons*. Sage, Thousand Oaks, CA.

Agrawal, R., Gollapudi, S., Halverson, A., and Ieong, S. (2009). Diversifying Search Results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 5--14.

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.

Almeida, J., Gonçalves, M., Figueiredo, F., Belém, F., and Pinto, H. (2010). On the quality of information for web 2.0 services. *IEEE Internet Computing*, 14(6):47--55.

Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*.

Banzhaf, W., Nordin, P., Keller, R. E., and Francone, F. D. (1998). *Genetic Programming – An Introduction on the Automatic Evolution of Computer Programs and its Applications*. Morgan Kaufmann.

Belém, F. (2011). Recomendação Associativa de Tags Considerando Múltiplos Atributos Textuais. Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Minas Gerais.

Belém, F., Martins, E., Pontes, T., Almeida, J., and Gonçalves, M. (2011). Associative Tag Recommendation Exploiting Multiple Textual Features. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1033--1042.

Belém, F. M., Martins, E. F., Almeida, J. M., and Gonçalves, M. A. (2014). Personalized and object-centered tag recommendation methods for web 2.0 applications. *Information Processing & Management*, 50(4):524--553.

Bi, B. and Cho, J. (2013). Automatically Generating Descriptions for Resources by Tag Modeling. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*, pages 2387--2392.

Blei, D. M. (2012). Probabilistic topic models. *Communications ACM*, 55(4):77--84.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5--32.

Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the Seventh International Conference on World Wide Web*, pages 107--117.

Cao, H., Xie, M., Xue, L., Liu, C., Teng, F., and Huang, Y. (2009). Social tag prediction based on supervised ranking model. In *Proceedings of the ECML/PKDD Discovery Challenge Workshop*, pages 35--48.

Cao, Z., Qin, T., Liu, T.-Y., Tsai, M.-F., and Li, H. (2007). Learning to rank: From pairwise approach to listwise approach. In *Proceedings of the 24th International Conference on Machine Learning*, pages 129--136.

Carbonell, J. and Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 335--336.

Castells, P., Vargas, S., and Wang, J. (2011). Novelty and diversity metrics for recommender systems: Choice, discovery and relevance. pages 29--37.

Celma, O. and Herrera, P. (2008). A new approach to evaluating novel recommendations. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, pages 179--186.

Chapelle, O. and Chang, Y. (2011). Yahoo! learning to rank challenge overview. *Journal of Machine Learning Research*, 14:1–24.

Chen, X. and Shin, H. (2013). Tag recommendation by machine learning with textual and social features. *Journal of Intelligent Information Systems*, 40(2):261–282.

Choi, Y. (2015). A complete assessment of tagging quality: A consolidated methodology. *Journal of the Association for Information Science and Technology*, 66:798--817.

Clarke, C., Craswell, N., Soboroff, I., and Ashkan, A. (2011). A comparative analysis of cascade measures for novelty and diversity. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, pages 75--84.

Clarke, C. L. A., Craswell, N., and Voorhees, E. M. (2012). Overview of the TREC 2012 Web track. In *Available at: trec.nist.gov/pubs/trec21/papers/WEB12.overview.pdf.*

Dang, V. and Croft, W. (2012). Diversity by proportionality: an election-based approach to search result diversification. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 65--74.

Efron, B. and Tibshirani, R. J. (1993). *An Introduction to the Bootstrap*. Chapman & Hall.

Faria, F., Veloso, A., Almeida, H., Valle, E., Torres, R., Gonçalves, M., and Meira, W. (2010). Learning to Rank for Content-based Image Retrieval. In *Proceedings of the International Conference on Multimedia Information Retrieval*, pages 285--294.

Feige, U. (1998). A threshold of $\ln(n)$ for approximating set cover. *Journal of the ACM*, 45:634--652.

Feng, W. and Wang, J. (2012). Incorporating Heterogeneous Information for Personalized Tag Recommendation in Social Tagging Systems. In *Proceedings of the 18th ACM International Conference on Knowledge Discovery and Data Mining*, pages 1276--1284.

Figueiredo, F., Belém, F., Pinto, H., Almeida, J., and Gonçalves, M. (2012). Assessing the quality of textual features in social media. *Information Processing & Management*, 49:222--247.

Freund, Y., Iyer, R., Schapire, R., and Singer, Y. (2003a). An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Results*, 4:933--969.

Freund, Y., Iyer, R., Schapire, R. E., and Singer, Y. (2003b). An efficient boosting algorithm for combining preferences. *Journal Of Machine Learning Research*, 4:933--969.

Friedman, J. (2000). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189--1232.

Garg, N. and Weber, I. (2008). Personalized, Interactive Tag Recommendation for Flickr. In *Proceedings of the ACM Conference on Recommender Systems*, pages 67--74.

Gemmell, J., Schimoler, T., Mobasher, B., and Burke, R. (2010). Hybrid Tag Recommendation for Social Annotation Systems. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pages 829--838.

Geng, L. and Hamilton, H. J. (2006). Interestingness measures for data mining: A survey. *ACM Computing Surveys*, 38(3).

Gomes, G. C., de Oliveira, V. C., de Almeida, J. M., and Gonçalves, M. A. (2013). Is learning to rank worth it? a statistical analysis of learning to rank methods. *Journal of Information and Data Management*, 4:193--200.

Goodman, L. A. (1961). Snowball Sampling. *Annals of Mathematics and Statistics*, 32(1):148--170.

Graham, R. and Caverlee, J. (2008). Exploring feedback models in interactive tagging. In *Proceedings of the International Conference on Web Intelligence and Intelligent Agent Technology*, pages 141--147.

Guan, Z., Bu, J., Mei, Q., Chen, C., and Wang, C. (2009). Personalized Tag Recommendation Using Graph-Based Ranking On Multi-Type Interrelated Objects. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 540--547.

Gupta, M., Li, R., Yin, Z., and Han, J. (2010). Survey on social tagging techniques. *SIGKDD Explorations*, 12(1):58--72.

Guy, I., Zwerdling, N., Ronen, I., Carmel, D., and Uziel, E. (2010). Social Media Recommendation Based on People and Tags. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 194--201.

He, X. and Chua, T.-S. (2017). Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 355--364.

Heymann, P., Ramage, D., and Garcia-Molina, H. (2008). Social tag prediction. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 531--538.

Hochbaum, D. S., editor (1997). *Approximation algorithms for NP-hard problems*. PWS Publishing Co.

Hu, M., Lim, E.-P., and Jiang, J. (2010). A probabilistic approach to personalized tag recommendation. In *Proceedings of the 2010 IEEE Second International Conference on Social Computing*, pages 33--40.

Hulth, A. (2003). Improved automatic keyword extraction given more linguistic knowledge. In *Conference on Empirical Methods in Natural Language Processing*.

Ifada, N. and Nayak, R. (2016). How Relevant is the Irrelevant Data: Leveraging the Tagging Data for a Learning-to-Rank Model. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 23--32.

Jäschke, R., Eisterlehner, F., Hotho, A., and Stumme, G. (2009). Testing and evaluating tag recommenders in a live system. In *Proceedings of the Third ACM Conference on Recommender Systems*, pages 369--372.

Jäschke, R., Marinho, L., Hotho, A., Schmidt-Thie, L., and Stum, G. (2007). Tag recommendations in folksonomies. In *Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 506--514.

Joachims, T. (2006). Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 217--226.

Khuller, S., Moss, A., and Naor, J. S. (1999). The budgeted maximum coverage problem. *Information Processing Letters*, 70:39--45.

Krestel, R. and Fankhauser, P. (2012). Personalized topic-based tag recommendation. *Neurocomputing*, 76(1):61--70.

Krestel, R., Fankhauser, P., and Nejdl, W. (2009). Latent dirichlet allocation for tag recommendation. In *Proceedings of the 3rd ACM Conference on Recommender Systems*, pages 61--68.

Küçüktunç, O., Saule, E., Kaya, K., and Çatalyürek, U. V. (2013). Diversified recommendation on graphs: Pitfalls, measures, and algorithms. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 715--726.

Lathia, N., Hailes, S., Capra, L., and Amatriain, X. (2010). Temporal diversity in recommender systems. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 210--217.

Li, X., Guo, L., and Zhao, Y. E. (2008). Tag-based social interest discovery. In *Proceedings of the 17th International Conference on World Wide Web*, pages 675--684.

Liang, S., Ren, Z., and de Rijke, M. (2014). Fusion helps diversification. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 303--312.

Liaw, A. and Wiener, M. (2002). Classification and regression by random forest. *R News*, 2(3):18–22.

Lin, Z., Ding, G., Hu, M., Wang, J., and Sun, J. (2012). Automatic Image Annotation Using Tag-Related Random Search Over Visual Neighbors. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 1784--1788.

Lipczak, M., Hu, Y., Kollet, Y., and Milios, E. (2009). Tag sources for recommendation in collaborative tagging systems. In *Proceedings of the ECML/PKDD Discovery Challenge Workshop*.

Lipczak, M. and Milios, E. (2010). The impact of resource title on tags in collaborative tagging systems. In *Proceedings of the 21st ACM Conference on Hypertext and Hypermedia*, pages 179--188.

Lipczak, M. and Milios, E. (2011). Efficient tag recommendation for real-life data. *ACM Transactions on Intelligent Systems Technology*, 3(1):2:1--2:21.

Liu, K., Fang, B., and Zhang, W. (2010). Speak the same language with your friends: Augmenting tag recommenders with social relations. In *Proceedings of the 21st ACM Conference on Hypertext and Hypermedia*, pages 45--50.

Liu, T.-Y. (2009). Learning to Rank for Information Retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225--331.

Liu, Z., Zhang, Y., Chang, E. Y., and Sun, M. (2011). PLDA+: Parallel Latent Dirichlet Allocation with Data Placement and Pipeline Processing. *ACM Transactions on Intelligent Systems Technology*, 2(3):26:1--26:18.

Lops, P., de Gemmis, M., Semeraro, G., Musto, C., and Narducci, F. (2013). Content-based and collaborative techniques for tag recommendation: an empirical evaluation. *Journal of Intelligent Information Systems*, 40(1):41--61.

Lu, Y.-T., Yu, S.-I., Chang, T.-C., and Hsu, J. Y.-j. (2009). A content-based method to enhance tag recommendation. In *Proceedings of the 21st International jont Conference on Artifical intelligence*, pages 2064--2069.

Marinho, L. B., Hotho, A., Jschke, R., Nanopoulos, A., Rendle, S., Schmidt-Thieme, L., Stumme, G., and Symeonidis, P. (2012). *Recommender Systems for Social Tagging Systems*.

Martins, E. F., Belém, F. M., Almeida, J. M., and Gonçalves, M. A. (2013). Measuring and Addressing the Impact of Cold Start on Associative Tag Recommenders. In *Simpósio Brasileiro de Sistemas Multimídia e Web*, pages 325--332.

Martins, E. F., Belém, F. M., Almeida, J. M., and Gonçalves, M. A. (2016). On cold start for associative tag recommendation. *Journal of the Association for Information Science and Technology*, 67(1):83--105.

Menezes, G., Almeida, J., Belém, F., Gonçalves, M., Lacerda, A., Moura, E., Pappa, G., Veloso, A., and Ziviani, N. (2010). Demand-driven tag recommendation. In *Proceedings of the ECML/PKDD Discovery Challenge Workshop*.

Mohan, A., Chen, Z., and Weinberger, K. (2011). Web-search ranking with initialized gradient boosted regression trees. *JMLR: Proceedings of the Yahoo! Learning to Rank Challenge*, 14:77–89.

Nguyen, H. T. H., Wistuba, M., and Schmidt-Thieme, L. (2017). Personalized tag recommendation for images using deep transfer learning. In Ceci, M., Hollmén, J., Todorovski, L., Vens, C., and Džeroski, S., editors, *Machine Learning and Knowledge Discovery in Databases*, pages 705--720.

Pedro, J., Siersdorfer, S., and Sanderson, M. (2011). Content Redundancy in YouTube and its Application to Video Tagging. *ACM Transactions on Information Systems*, 29:13:1--13:31.

Poli, R., Langdon, W., and Mcphee, N. (2008). *A Field Guide to Genetic Programming*. Lulu Enterprises, UK Ltd.

Prokofyev, R., Boyarsky, A., Ruchayskiy, O., Aberer, K., Demartini, G., and Cudré-Mauroux, P. (2012). Tag recommendation for large-scale ontology-based information systems. In *Proceedings of the 11th International Conference on The Semantic Web - Volume Part II*, pages 325--336.

Qin, T., Liu, T.-Y., and Li, H. (2010). A general approximation framework for direct optimization of information retrieval measures. *Information Retrieval*, 13(4).

Quintarelli, E. (2005). Folksonomies: Power to the people. Retrieved on October 8th, 2015. *http://www.iskoi.org/doc/folksonomies.htm*.

Rabinovich, E., Rom, O., and Kurland, O. (2014). Utilizing relevance feedback in fusion-based retrieval. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 313--322.

Rader, E. and Wash, R. (2008). Influences on tag choices in del.icio.us. In *Proceedings of the 2008 ACM Conference on Computer supported cooperative work*, pages 239--248.

Rae, A., Sigurbjörnsson, B., and van Zwol, R. (2010). Improving tag recommendation using social networks. In *Adaptivity, Personalization and Fusion of Heterogeneous Information*, pages 92--99.

Rendle, S., Balby Marinho, L., Nanopoulos, A., and Schmidt-Thieme, L. (2009a). Learning optimal ranking with tensor factorization for tag recommendation. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 727--736.

Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. (2009b). Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, pages 452--461.

Rendle, S. and Schmidt-Thie, L. (2010). Pairwise interaction tensor factorization for personalized tag recommendation. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, pages 81--90.

Ribeiro, I. S., Santos, R. L., Gonçalves, M. A., and Laender, A. H. (2015). On tag recommendation for expertise profiling: A case study in the scientific domain. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 189--198.

Ribeiro, M. T., Lacerda, A., Veloso, A., and Ziviani, N. (2012). Pareto-efficient hybridization for multi-objective recommender systems. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, pages 19--26.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *Computing Research Repository*, abs/1609.04747.

Santos, R., Macdonald, C., and Ounis, I. (2010). Exploiting query reformulations for web search result diversification. In *Proceedings of the 19th International Conference on World Wide Web*, pages 881--890.

Santos, R., Macdonald, C., and Ounis, I. (2015). Search result diversification. *Foundations and Trends in Information Retrieval*, 9(1):1--90.

Santos, R. L. T. and Ounis, I. (2011). Diversifying for multiple information needs. In *In Proceedings of the 1st International Workshop on Diversity in Document Retrieval*, pages 37--41.

Saveski, M. and Mantrach, A. (2014). Item cold-start recommendations: Learning local collective embeddings. In *Proceedings of the ACM Conference on Recommender Systems*, pages 89--96.

Schein, A. I., Popescul, A., Ungar, L. H., and Pennock, D. M. (2002). Methods and metrics for cold-start recommendations. In *Proceedings of the 25th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 253--260.

Shi, L. (2013). Trading-off among accuracy, similarity, diversity, and long-tail: A graph-based recommendation approach. In *Proceedings of the 7th ACM Conference on Recommender Systems*, pages 57--64.

Siersdorfer, S., San Pedro, J., and Sanderson, M. (2009). Automatic video tagging using content redundancy. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 395--402.

Sigurbjörnsson, B. and Zwol, R. (2008). Flickr tag recommendation based on collective knowledge. In *Proceedings of the 17th International Conference on World Wide Web*, pages 327--336.

Song, Y., Zhang, L., and Giles, C. L. (2011). Automatic tag recommendation algorithms for social recommender systems. *ACM Transactions on the Web*, 5:1--31.

Song, Y., Zhuang, Z., Li, H., Zhao, Q., Li, J., Lee, W.-C., and Giles, C. L. (2008). Real-time automatic tag recommendation. In *Proceedings of the 31st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 515--522.

Spiteri, L. F. (2007). Structure and form of folksonomy tags: The road to the public library catalogue. *Webology*, 4(2).

Szpektor, I., Maarek, Y., and Pelleg, D. (2013). When relevance is not enough: Promoting diversity and freshness in personalized question recommendation. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 1249--1260.

Vargas, S. and Castells, P. (2011). Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, pages 109--116.

Vargas, S., Castells, P., and Vallet, D. (2012). Explicit relevance models in intent-oriented information retrieval diversification. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 75--84.

Veloso, A., Jr., W. M., Cristo, M., Gonçalves, M. A., and Zaki, M. J. (2006). Multi-evidence, multi-criteria, lazy associative document classification. In *Proceedings of the 2006 International Conference on Information and Knowledge Management, Arlington, Virginia, USA, November 6-11, 2006*, pages 218--227.

Venetis, P., Koutrika, G., and Garcia-Molina, H. (2011). On the selection of tags for tag clouds. In *Proceedings of the fourth ACM International Conference on Web search and Data Mining*, pages 835--844.

Wal, V. (2005). Explaining and Showing Broad and Narrow Folksonomies. Retrieved on October 8th, 2015. *http://www.vanderwal.net/random/entrysel.php?blog=1635*.

Wang, J., Hong, L., and Davison, B. D. (2009). Tag recommendation using keywords and association rules. In *Proceedings of the ECML/PKDD Discovery Challenge Workshop*.

Wu, L., Yang, L., Yu, N., and Hua, X. (2009). Learning to tag. In *Proceedings of the 18th International Conference on World Wide Web*, pages 361--370.

Wu, Q., Burges, C., Svore, K., and Gao, J. (2010). Adapting boosting for information retrieval measures. *Information Retrieval*, 13(3):254--270.

Xu, J. and Li, H. (2007). Adarank: a boosting algorithm for information retrieval. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 391--398.

Yeh, J., Lin, J., Ke, H., and Yang, W. (2007). Learning To Rank For Information Retrieval Using Genetic Programming. In *SIGIR 2007 Workshop: Learning To Rank For Information Retrieval*.

Yin, D., Guo, S., Chidlovskii, B., Davison, B., Archambeau, C., and Bouchard, G. (2013). Connecting Comments and Tags: Improved Modeling of Social Tagging Systems. In *Proceedings of the 6th ACM International Conference on Web Search and Data Mining*, pages 547--556.

Yu, J., Mohan, S., Putthividhya, D. P., and Wong, W.-K. (2014). Latent dirichlet allocation based diversified retrieval for e-commerce search. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, pages 463--472.

Yuan, F., Guo, G., Jose, J. M., Chen, L., Yu, H., and Zhang, W. (2017). Boostfm: Boosted factorization machines for top-n feature-based recommendation. In *Proceedings of the 22nd Conference on Intelligent User Interfaces*, pages 45--54.

Zhai, C., Cohen, W. W., and Lafferty, J. (2003). Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 10--17.

Zhang, H., Korayem, M., You, E., and Crandall, D. J. (2012a). Beyond Co-occurrence: Discovering and Visualizing Tag Relationships from Geo-Spatial and Temporal Similarities. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining*, pages 33--42.

Zhang, N., Zhang, Y., and Tang, J. (2009). A tag recommendation system based on contents. In *Proceedings of the ECML/PKDD Discovery Challenge Workshop*.

Zhang, Y., Séaghdha, D., Quercia, D., and Jambor, T. (2012b). Auralist: Introducing Serendipity into Music Recommendation. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining*, pages 13--22.

Zhu, X., Nejdl, W., and Georgescu, M. (2014a). An adaptive teleportation random walk model for learning social tag relevance. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 223--232.

Zhu, Y., Lan, Y., Guo, J., Cheng, X., and Niu, S. (2014b). Learning for search result diversification. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 293--302.