Nurturing Tagging Communities

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Shilad Wieland Sen

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor Of Philosophy

March, 2009

Nurturing Tagging Communities

by Shilad Wieland Sen

ABSTRACT

Member contributions power many online communities. Users have uploaded billions of images to flickr, bookmarked millions pages on del.icio.us, and authored millions of encyclopedia articles at Wikipedia. Tags — member contributed words or phrases that describe items — have emerged as a powerful method for searching, organizing, and making sense of, these vast corpora.

In this thesis we explore the dynamics, challenges, and possibilities of tagging systems. We study the way in which factors influencing an individual user's choice of tags can affect the evolution of community tags as a whole. Like other community-maintained systems, tagging systems can suffer from low quality contributions. We study interfaces and algorithms that can differentiate between low quality and high quality tags. Finally, we explore tagommenders, tag-based recommendation algorithms that combine the flexibility of tags with the automation of recommender systems.

We base our explorations on tagging activity in the MovieLens movie recommendation system. We analyze tagging behavior, user studies, and surveys, of 97,000 tags and 3,600 users. Our results provide insight into the dynamics of existing tagging communities, and suggest mechanisms that address challenges of, and provide extensions to, tagging systems.

Acknowledgements

I would like to thank:

- John Riedl, for his youthful exuberance and sage advice.
- Grouplens, for their energy, intellect, and comraderie.
- Max Harper and Dan Frankowski for their friendship and intellectual curiosity.
- My family, for believing in me.
- Katy, for her patience, encouragement, and companionship.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 The Adult Annotator

The MovieLens[1] website recommends movies to users based on their movie ratings. Until recently, a small group of graduate students maintained the movie database with the help of a part-time volunteer [Cosley, 2006]. MovieLens could not afford to pay a team of expert editors, as Amazon[2] and All Movie Guide did.[3] As users repeatedly pointed out, MovieLens database content was limited by the abilities and time constraints of its maintainers.

In January 2006 MovieLens opened up its database by empowering users to apply tags - descriptive words or phrases - to movies.[4] Two weeks after tagging was launched, a user applied the tag *nude brook shields*, to the movie "The Blue Lagoon". Nudity tags remained relatively rare until six months later, when a user applied the tag *notable nudity* to the movie "National Lampoon's Vacation." The MovieLens community adopted the tag immediately. Over the next three months, *notable nudity* was used 30 times by 20 different users. Only four other nudity-related tags were used during the same time period.

Several months later, on September 9th, a single dedicated tagger whom we will refer to as "Adult Annotator" choose to refine the *notable nudity* annotations. Between

[1] http://movielens.org
[2] http://amazon.com
[3] http://allmovieguide.com
[4] The implementation of tagging in MovieLens is described in detail in Chapter 3.

one and two o'clock in the morning, Adult Annotator developed a nine-category classification for movie nudity ranging from *nudity (topless - brief)* to *nudity (full frontal - notable)*, resulting in 279 applications of the tags. The MovieLens community immediately switched its preferred nudity vocabulary to match Adult Annotator's. In the three yeas since they were introduced, Adult Annotator's tags have been used 1,370 times by 137 different users. During the same time period, users have only used *notable nudity* ten times.

Tags provide a mechansim for users to adapt the information space of MovieLens in ways that are meaningful to them. While the adult annotator used this power to introduce a carefully crafted set of tags, there is no guarantee that other users will do the same. There may be ways that MovieLens can nurture more thoughtful taggers like the Adult Annotator. In this thesis, we explore the dynamics of tagging systems. Based on our insights into tagging dynamics, we suggest mechanisms that overcome the challenges of quality control and information overload inherent in tagging systems.

1.2 What are Tags?

Tags are words or phrases that capture some facet of an entity. Users may use tags to describe movies they have seen, websites they have visited, or photographs they have uploaded. Tags have exploded in popularity due to their flexiblity [Shirky, 2005], their ability to harness the power of ordinary users, and their social appeal [Millen et al., 2005]. Three examples of popular real-world tagging sites are:

- **Delicious**,[5] a social bookmarking site, enables users to label web pages with tags. Since Delicious was launched in 2003, 5.3 million users have tagged over 180 million distinct URLs.[6]

- **LibraryThing**[7] enables users to catalog their personal book collection. As users add books to their collection, they describe them with tags. LibraryThing users have contributed over 43 million tags describing 4 million books.[8]

[5] http://del.icio.us
[6] http://blog.delicious.com/blog/2008/11/delicious-is-5.html
[7] http://librarything.com
[8] http://www.librarything.com/zeitgeist

- **Flickr**[9] enables users to share their photos online. Since image classification algorithms are not yet reliable enough to accurately identify an image's topical contents, Flickr relies on users to apply tags to the images they upload. Flickr users have uploaded and tagged over 3 billion photographs.[10]

Tags have become a ubiquituous feature among the social web applications dominating the Internet for two reasons [O'Reilly, 2007]. First, tagging systems scale by turning content creation over to the members of an online community. MovieLens maintainers have limited resources, but MovieLens users want more content. Tags empower MovieLens users to create content themselves.

A second reason for tag's popularity lies in their flexibility. Tagging systems place few restrictions on tag creation.[11] Clay Shirky argues that tags evolve organically to match the concepts important to a community precisely because the tags are created by the commmunity itself[Shirky, 2005]. Moreover, Shirky suggests that although particular individuals may create low quality or idiosyncratic tags, the aggregate collection of tags in a community reflects the overall characteristics of the community.

Tags draw their scalability from contributions by ordinary users. However, a user's motivation for tagging may seem puzzling. Why did MovieLens's Adult Annotator single-handedly invent a classification for nudity? Ames et al. finds that a user's motivations for tagging fall along two key dimensions: personal versus social and organizational versus communicative:

- **Personal / Organizational -** A user tags so they can later find items with a tag. For example, when Adult Annotator wants to watch a good movie with *nudity (rear)*, he can use MovieLens to search for highly recommended movies with the tag.

- **Personal / Communicative -** A user tags to remember characteristics of a particular item. For example, a user on LibraryThing may tag a book with *paris* to remind her that it was set in Paris. While the personal / organizational dimension

[9] flickr.com

[10] http://mashable.com/2008/11/03/flickr-3-billion-photos-uploaded/

[11] Systems often place lexical restrictions on tags. For example, some systems do not allow spaces in tags.

helps a user to search for an item she tagged, the communicative dimension helps the user make sense of an item once she has found it.

- **Social / Organizational-** A user may tag to help other users locate items. For example, when a user uploads an photograph to Flickr they may use the tag *saxophone* to help users searching for saxophones find their photograph.

- **Social / Communicative -** Users tag to communicate some facet of an item to other users. For example, four users on MovieLens used a tag to express their opinion that "The Princess Bride" was *funny*. While the social / organizational dimension helps other users to search for an item a user tagged, the communicative dimension helps a user make sense of an item once she has found it.

1.3 Why Study Tags?

We study tags for two main reasons. First, we hope to understand and overcome the challenge of quality control inherent in tagging systems. Second, we use tags as a motivation for studying tags is to use them as a vehicle for studying member contributions more broadly.

Quality Control. Tagging systems may lack the quality control of expert-maintained content, and they may contribute to a user's overall information overload. In MovieLens, for example, users consider more than half of all tags to be of low quality.[12] In a direct rebuttal of Shirky, Peter Merholz argues that tagging systems encourage tag synonyms (*nude* versus *nudity*) that increase the difficulty of finding items[13]. Should a user search for *nude*, *nudity*, or *notable nudity*?

Furthermore, tag evolution may be volatile. Even similar systems can result in very different vocabularies of tags. For example Amazon.com[14] users have applied 705 distinct tags to the book Liberal Fascism: The Secret History of the American Left by Jonah Goldberg. LibraryThing users have applied 107 tags to the same book. However, the nature of the tags for Goldberg's book differ between the two systems. The top three tags on LibraryThing are factual in nature: *politics* (applied by 42 people), *history*

[12] Based on a user survey we conduct in Chapter 5.
[13] http://www.peterme.com/archives/000558.html
[14] http://www.amazon.com

(30), and *non-fiction* (16). In contrast, the top three tags on Amazon are editorial tags apparently intended to provide comedic value: *wingnut welfare* (373), *propaganda* (290), and *editor promised cake* (221).

By understanding tag creation, evolution, and quality we hope to to address challenges faced by tagging systems. Due to the popularity of tags, improvements in tagging systems can potentially impact millions of users. The three example sites we described support millions of taggers who have created billions of tags.

Relationship to other types of member contributions. Member contributions power many online communities. Users have uploaded billions of images to Flickr and edited millions of articles on Wikipedia. We refer to sites such as Flickr and Wikipedia, where the members of the community create valuable content, *member-maintained* communities [Cosley, 2006].

Our second motivation for studying tags is to use them as a vehicle for studying member contributions more broadly. Tags' structural simplicity allows us to easily analyze their lexical and social characteristics. Conducting similar analyses of Wikipedia articles might be difficult due to its sheer size. On YouTube, similar analyses of uploaded videos may be impossible due to the lack of algorithms that reliably classify video content.

Despite their simplicity, tags capture many challenges of more complex forms of member contributions. Like Wikipedia articles and YouTube videos, the quality of tags varies. As we show in Chapter 4, taggers are influenced by community norms. We hope that by studying tagging systems we can understand these phenomena from the perspective of more general community contributions.

1.4 An Overview of Our Work

Chapters 2 and 3 provide background information central to our analyses of tagging systems. In Chapter 2 we present a taxonomy for design choices in tagging systems, along with existing research related to each design component. Using our taxonomy, Chapter 3 describes the tagging design of the MovieLens recommender system that serves as an experimental platform for our research.

MovieLens's Adult Annotator single-handedly shifted community norms for a group

of tags in MovieLens. We also saw that two similar tagging systems (Amazon and LibraryThing) produced very different tag vocabularies for the same item (Goldberg's "Liberal Faciscm"). In Chapter 4 we try to understand the evolution of tag vocabularies. We analyze the motivations for taggers such as MovieLens's Adult Annotator, and study the factors that determine their choice of tags. We also investigate the effects of user interfaces on tag choice.

Tags shift editorial power from a few content experts to the online masses. This shift risks reducing quality. In Chapters 5 and 6 we explore mechanisms for detecting and displaying high quality tags while supressing low quality ones. In Chapter 5 we explore interfaces for determining a tag's quality. We compare designs for thumb-based rating feedback about tag quality using a controlled experiment. In Chapter 6 we propose and evaluate algorithms that automatically infer a tag's quality using the ratings from Chapter 5 along with other tag characteristics.

In the final chapters we shift focus towards understanding a user's personal preferences towards tags, and we also explore algorithms that use those tag preferences to infer a user's preference for an item. As a concrete example, based on Adult Annotator's extensive knowledge of nudity, it is plausibile that she enjoys movies tagged with *nudity (topless)*. In our terminology, Adult Annotator has a high *tag preference* for *nudity (topless)*. We begin in Chapter 7 by carefully examining the connection between tag quality and tag preference. We propose and analyze algorithms that infer users' tag preferences based on their interactions with a tagging site. Finally, in Chapter 8 we explore *tagommenders*, algorithms that predict a user's preferences for items (e.g. movies) based on her preference for tags.

Chapter 2

Design of Tagging Systems

While all tagging systems follow the same high-level principle of allowing people to apply free form textual labels (tags) to items in the system (e.g., web log entries, bookmarks, or pictures), there are several important choices that define a design space for tagging systems. For example, on Flickr, users primarily tag their own uploaded photos. In contrast, Amazon's products are not created by the users of the site — they are created by product manufacturers such as book publishers. Because of this, Amazon users apply their own individual set of tags to a product.

In this chapter we explore the design space for tagging systems. We ground our analyses through examples from real-world tagging systems. As we traverse the design space, we describe the landscape of related research.

2.1 Tags and Tag Applications

Tagging systems provide information about *items* such as photos, books, and web pages. Tagging systems impose few restrictions on tags. As a result, users apply tags for a wide variety of reasons, and tags support a wide variety of functions.

A *tag application* is a relationship between a user, a tag and an item. Marlow et al. describes two models for tag applications: the *set* model, and the *bag* model [Marlow et al., 2006]. In the set model, a tag can only be added to an item once. Tagging systems such as YouTube and Flickr that support the set model do not generally identify the user that created the tag. Thus, they effectively model tag applications as

Figure 2.1: Amazon's interface for *Snow Crash*. All screen captures in this chapter were taken in December 2008 unless otherwise specified.

an $< item, tag >$ tuple[1] .

In the bag model, different users can add the same tag to a particular item, and their tags are stored separately. Tagging systems that support the bag model, such as Delicious, Amazon, and LibraryThing, model tag aplications as a $< user, item, tag >$ triple. These systems generally display the number of users who applied a particular tag to an item. For example, 166 users on Amazon have tagged Neal Stephenson's *Snow Crash* with *cyberpunk* (Figure 2.1).[2]

Since the bag model allows multiple users to apply the same tag to an item, systems that support the bag model can treat the number of users who apply a particular tag to an item as a measure of agreement on the tag application. This can be advantageous in systems with a high degree of tagging activity around a shared set of item. Systems that are unlikely to have tagging activity by multiple users around the same item are unlikley to benefit from the bag model.

2.2 Creating Tags

Tagging permission. The permission model controls who can tag items. Marlow et al. define three permission models for tags: *self-tagging*, *permission-based*, and *free-for-all* [Marlow et al., 2006]. In the self-tagging permission model, only the owner of an

[1] Vander also describe the set vs. bag dimension in a blog post, but use the terminology of broad (bag) or narrow(set). Although Vander defined the dimension earlier, we prefer Marlow's terminology. Vander's blog post can be found at http://personalinfocloud.com/2005/02/explaining_and_.html

[2] The distinction between the set model and bag model describes the external user interactions with a tagging system. Even systems using the set model internally store information about the user who applied a specific tag.

item (e.g. the person who uploaded a video to YouTube) can tag the item. In systems with permission-based tags, such Flickr, the owner of an item may grant permission for other users to tag the item. Finally, systems with shared items, such as Amazon and Delicious, support free-for-all tagging in which any user can tag any item.

Different tagging systems support varying degrees of item ownership. Users of YouTube "own" the videos they post. On the other hand, books, movies, and albums in systems such as Amazon are not created by individual members of the community. Sites supporting item ownership (YouTube) generally use self-tagging or permission-based models. Since there are no item owners, sites with shared ownership (Amazon) support a free-for-all model.

Systems with self-tagging and permission-based tagging often support tag applications following the set model. Since the tagging system only allows a few users to tag an item, there is little benefit in providing each user with his or her own set of tags. The self-tagging model is most effective when the creator of the item is likely to tag the item, but other users are not.

Tag creation interface. Most tagging systems allows users to type multiple tags in a single input text box. However, many systems offer mechanisms to assist users in choosing tags. This feature, called *tag suggestion* is most commonly implemented as a drop-down menu that automatically suggest tags that a user types. Other systems enable users to click on suggested tags to add them to the tag input box. Figure 2.2 shows Delicious' interface for creating tags.

Tag suggestion algorithm. The exact algorithm for tag suggestion differs from site to site. Flickr suggests tags the user has applied in the past. Delicious suggests tags chosen from the intersection of tags applied to the item by other users, and the tags applied by the user in the past[Rader and Wash, 2008]. Other systems, such as Slashdot, offer no tag suggestions.

In addition to basic tag suggestion algorithms such as those used by Delicious and Flickr, researchers have proposed more complex algorithms. Mishne et al. [Mishne, 2006] and Sood et al. [Sood et al., 2007] both investigate algorithms for inferring tags that should be applied to blog posts by analyzing the textual content of the post. Jaschke et al. studies tag suggestion algorithms based upon tagging histories [Jaschke et al., 2007]. These tag suggestion algorithms predict the tags users apply more reliably than existing

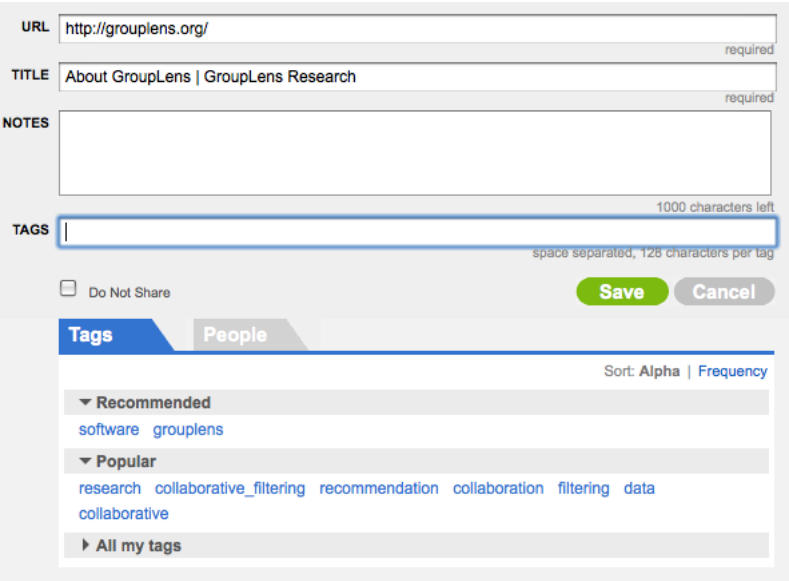


Figure 2.2: Tag creation on Delicious

algorithms.

Incorporating improved algorithms in tagging system might ease the process of tag creation and encourage users to apply more tags. In addition, designers may be able to suggest tags that influence the tag vocabulary in positive ways.

2.3 Displaying an Item's Tags

Tag sharing. Tag sharing describes the extent to which a user's tags are shown to other users of the system. At one extreme, fully private systems such as Gmail only make a tag application visible to the person who applied it. At the other extreme are fully shared systems such as LibraryThing, where all tag applications are visible to all users. Systems in the middle, such as Delicious, often make tags publicly visible by default, but enable users to mark certain tag applications they create as private.

Millen et al. point to the social aspects of tagging as one of the key reasons for tags' popularity [Millen et al., 2005]. Because of this, the decision to keep tags private may have a large impact on a tagging system. Most systems with private tags choose to keep them private for data privacy (e.g. Gmail).

Tag selection. Systems that allow tag sharing may not be able to display every tag applied to every item because of the sheer number of tag applications that may exist (for example, Delicious users have applied 10,688 distinct tags to describe the website Digg[3]).

The most common tag selection algorithms orders an item's tags according to the number of users who have applied the tag to the item. Since this algorithm relies on multiple applications of the same tag to an item, it only pertains to tagging systems that use the bag model (Figure 2.1). Tag selection algorithm serve as the lens through which users view tags. Because of this, they play an important role in the success of tagging systems.

Website purpose. Websites that primarily focus on tagging, such as Delicious, may face different design constraints than websites with other primary purposes (such as Amazon). Sites dedicated to tagging may allocate more screen space to tagging features. On the other hand, sites whose primary purpose is not tagging may provide a rich databases of item information (such as Amazon). These systems may find that users apply fewer informational tags than on sites without item information, such as Flickr.

2.4 Visualizing and Navigating Tags

Tag search. Tagging systems traditionally allow users to locate items annotated with a particular tag by typing the tag into a textual input field, or by clicking on a tag's hyperlink. Figure 2.3 shows the search results page for *saxophone* on LibraryThing, which displays a list of books tagged with *saxophone*, ordered by the number of users who applied the tag to the book. The number of users who apply the tag to the item captures both popularity and relevance. More users apply tags to popular items than unpopular ones. More users apply a particular tag to a relevant item than an irrelevant one. LibraryThing's implementation of tag search is typical of other tagging systems.

Tag clouds. In addition to describing facets of an item, tags enable users to visualize and navigate through items and tags. One common visualization for tags is the *tag cloud*, which lists the most popular tags for an item or search in alphabetical order. Tag clouds

[3] As of January 2007.

Figure 2.3: Search for the tag *saxophone* on LibraryThing.

convey the popularity of a tag by changing the font size for a tag; tags applied more frequently are shown in larger font. Figure 2.4 shows the tagcloud for Neal Stephenson's *Snow Crash* on LibraryThing.

Tag clouds have been a popular topic among researchers. Hassan et al. cluster related tags within tag clouds [Hassan-Montero and Herrero-Solana, 2006]. Rivadeneira et al. evaluate a variety of tag clouds effectiveness in supporting specific user tasks [Rivadeneira et al., 2007]. Halvey et al. study the effect of a tag cloud's sort order and font size [Halvey and Keane, 2007]. These researchers found that tag clouds effectively support searching, browsing, and impression formation. Moreover, small changes in font, ordering, and layout can lead to significant improvements in effectiveness.

Social navigation. Tagging systems often exhibit characteristics common among "web 2.0" systems [O'Reilly, 2007]. Many tagging systems support social navigatation by displaying the users that applied a particular tag, and the tags a particular user applied. For example, when searching for a tag, LibraryThing also displays a list of users who have applied the tag most often. LibraryThing's user profile pages also summarize the tags a user applies most often. Because tags relate users to concepts (tags) and items, they enable navigational social pivots between users and items, tags,

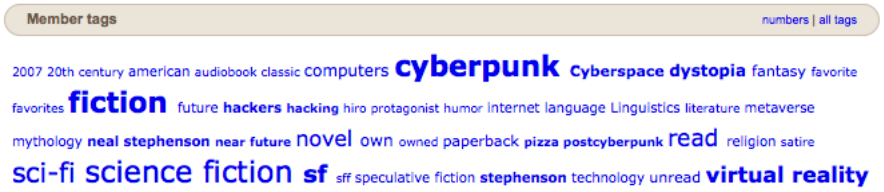


Figure 2.4: LibraryThing's tag cloud for *Snow Crash*.

or other users.

Other visualizations. Industry practitioners and researchers have investigated other visualizations for tags. Flickr offers tag clusters that disambiguate between multiple meanings for the same tag. For example, a search for *turkey* displays separate clusters for photos related to the country Turkey, Thanksgiving, and nature scenes. Dubinko et al.'s TagLines system displays the evolution of popular tags within Flickr, along with example photos using each tag [Dubinko et al., 2007]. Begelman et al. propose algorithms for navigating sets of related tags in Delicious[Begelman et al., 2006].

When viewed independently, tag applications express structurally simple relationships. However, in aggregate, tags contain powerful signals of community behavior. Both TagLines and Flickr's tag clusters analyze relationships between tags in order to offer insights into a tagging community. Future researchers may create even more compelling applications drawing on new analses of intra-tag relationships.

2.5 Summary

In this chapter we presented a taxonomy for design decisions faced by tagging system designers. We discussed choices for tagging data models, tag creation, tag display, and tag visualizations. We illustrated design choices using real-world tagging systems, and discussed existing research related to each design choice.

Our research extends existing research related to the design dimensions we discussed in this chapter. Chapter 4 describes our work on tag evolution. We build on early work that describes tagging communities by conducting the first controlled study of tag creation. We analyze the role of tag suggestion and tag selection algorithms in the evolution of a tagging community's vocabulary of tags. In Chapters 5 and 6 we further

on the descriptions and anecdotal observations of earlier researchers on tag selection algorithms by conducting the first empirical study of them. We propose algorithms that display high quality tags while supressing low quality ones.

Although researchers have not investigated interfaces for tag creation, different interfaces may affect the quantity of tag contribution. In addition, designers may be able to leverage novel interfaces to control the tag vocabulary in positive ways.

Chapter 3

MovieLens Platform

As a platform for our experiments, we incorporated tagging features into the MovieLens movie recommendation system[1] . MovieLens was created in 1997 by members of the Grouplens research group at the University of Minnesota. MovieLens primarily serves as a movie recommendation system: users rate movies and receive movie recommendations in return. More details of the MovieLens system can be found in [Dahlen et al., 1998]. In this chapter, we describe and motivate the design of tagging in MovieLens.

3.1 MovieLens Tagging Design Decisions

The choice of U.S. Theatrical Movies as an item domain influenced some design decisions. MovieLens users do not own the movies they tag. As we discussed in section 2.2, sites without item ownership generally support free-for-all tagging permissions since there is no clearly preferred tagger for an item. MovieLens thus supports a free-for-all tag permission model.

MovieLens supports the bag model for tag applications for two reasons. First, the bag model is generally more popular among sites with free-for-all models such as Delicious, LibraryThing, and Amazon. Second, we were interested in measuring tagger agreement supported by the bag model. Thus, MovieLens' data model for tag applications consists of a set of triples composed of a user, a movie, and a tag.

[1] http://www.movielens.org

MovieLens tags are delimited by commas to give users the flexibility to apply multi-word tags. Some users used this flexibility to create tags that functioned as mini movie reviews. For example, one user applied the following tag to the movie "The Secret of Roan Inish" (1994)

> "All I remember about this movie is that Jamie was a seal. And something about the mother being a unicorn... unless I have my Irish movies messed up."

MovieLens' tag selection algorithm varied depending on the current tagging experimental conditions. Generally, tags were publicly visible and ordered by popularity.

3.2 Pages Offering Tagging Features

MovieLens displays tags througout the website. In order to encourage tagging activity, during the year after tagging was introduced all new users were shown a screen that described the tagging features and encouraged them to apply tags (Figure 3.1).

The MovieLens home page displayed a list of the ten most recently applied tags (Figure 3.2). This helped increase user awareness of tagging activity. For each movie returned by a search, MovieLens displays up to three user tags (tag applications created by the user), and three comunity tags (tag applications not created by the user) (Figure 3.4). In addition to faceted searches such as title an genre searches, the search results screen is used to display lists of recommended and rated movies. A full list of tags applied to a movie was available on a movie's details page (Figure 3.3).

Figure 3.5 shows the interface for applying tags, which was available on both movie lists and movie details pages. To apply a tag, a user clicks the "add tag" link, which opens a text box. MovieLens dynamically generates an *auto-completion* list of tags matching what has been typed thus far. The user may select a tag from this list, or she may ignore it and continue typing her tags. A user can click the plus sign next to a tag applied by another community member to apply the tag to the item themselves.

Users can navigate through tag space by clicking a tag hyperlink, or by using a tag search box with the auto-completion feature. Figure 3.6 shows a search for the tag *martial arts*. A list of movies tagged with *martial arts* is displayed ordered by the

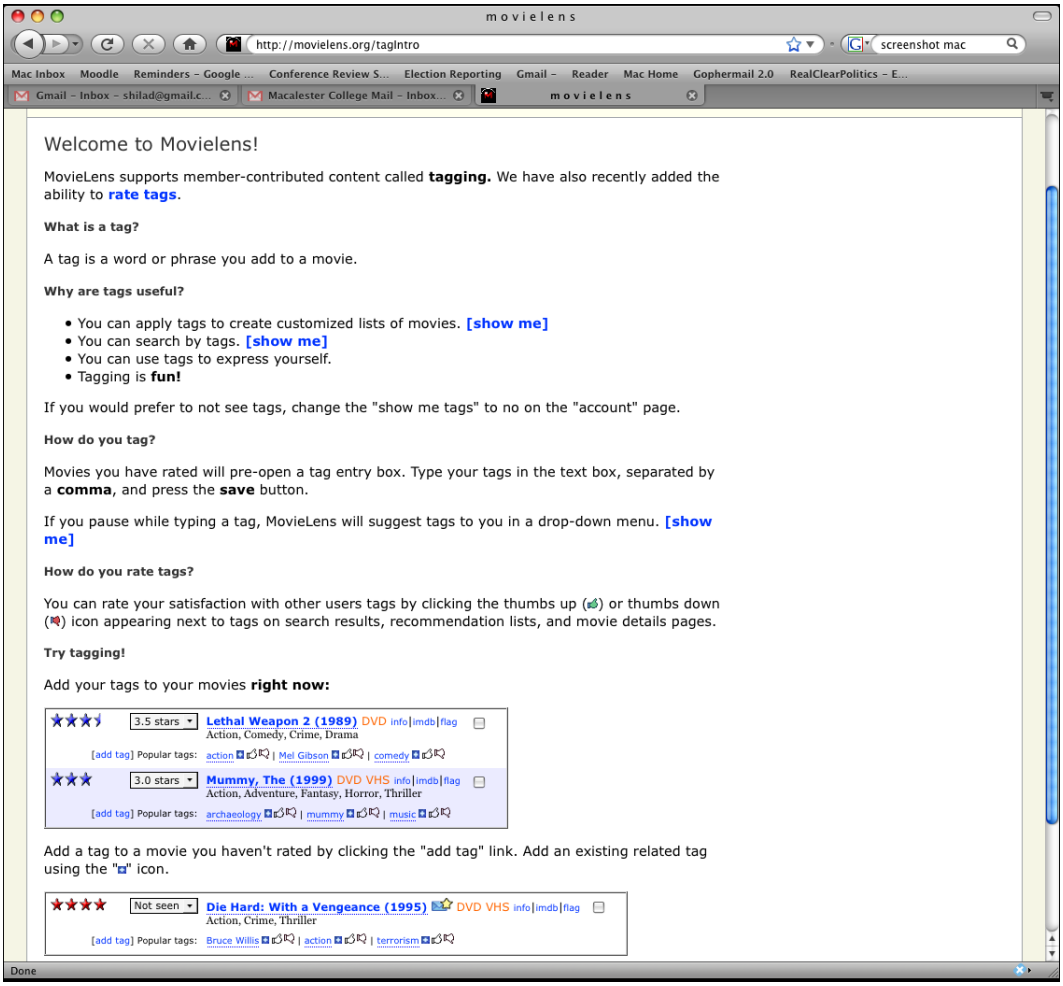


Figure 3.1: MovieLens tagging introductory page.

Figure 3.2: Recent tags displayed on the MovieLens home page.

Figure 3.3: Movie details page tags.

Figure 3.4: Movie list with tags.

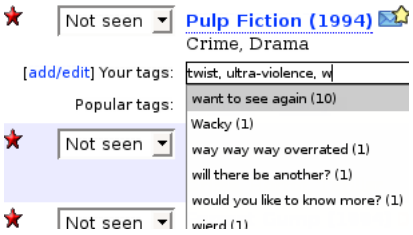


Figure 3.5: Adding tags with auto-complete.

Figure 3.6: Tag search results page.

number of users who applied the tag to the movie. A set of related tags is displayed in a tag cloud format. We also created a "Your Tags" page that lists all the tags that a user has applied along with a sampling of tags applied to each movie (Figure 3.7).

MovieLens attempts to make tagging as easy as possible. Tagging features are implemented using *AJAX* based functionality provided by the script.aculo.us[2] library. AJAX enables lightweight interaction between users and the tagging system, reducing the time needed to create tags. For example, when a user chooses to apply a tag to a movie on the search results page, she types the new tags into a popup dialog window instead of leaving the search results page .

[2] http://script.aculo.us/

Figure 3.7: The MovieLens "Your Tags" page.

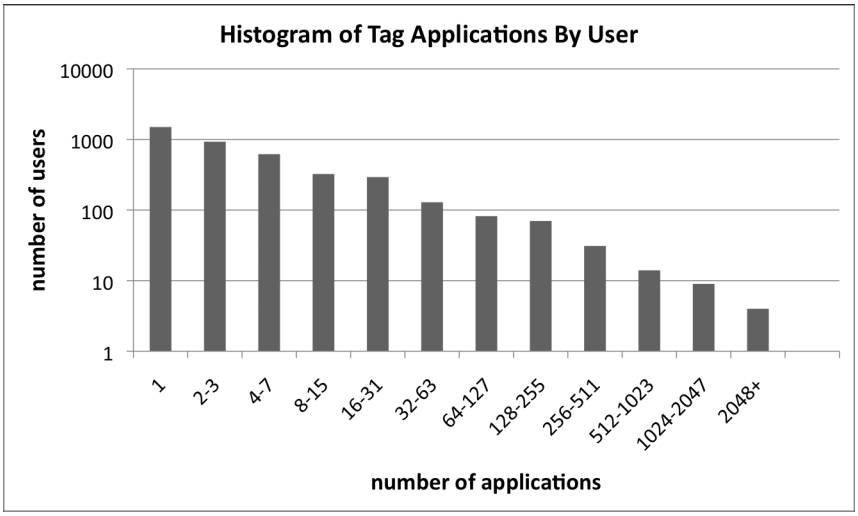


Figure 3.8: Histogram of tag applications grouped by user. The y-axis (number of users) is on a logarithmic scale. The disribution fits an exponential distribution ($y = 3006e^{-0.524x}, R^2 = 0.99$).

3.3 Basic Usage Statistics

The MovieLens site averages approximately 1390 active users per week[3] . Since MovieLens was launched, 131,346 users have rated an average of 122 movies each. Detailed statistcs about MovieLens movie ratings can be found in section [ref: tagommenders].

Between the launch of the MovieLens tagging system in January 2006 and January 2009, 3,629 users have generated 86,082 tag applications for 7,144 movies. Approximately 10% of MovieLens users apply at least one tag. The distribution of tag applications per tagger generally follows an exponential distribution ($y = 3006e^{-0.524x}, R^2 = 0.99$). Ignoring non-taggers, the mean number of tag applications is approximately 23, while the median is 2. 148 users have applied more than 100 tags, and 13 users have applied more than 1000 tags. The most prolific user has created 5,811 tag applications.

The tag search interface accounts for approximately 4% of all MovieLens searches - more than any other advanced search field.

Figure 3.9 displays the growth of tags, taggers, and tag applications over time in MovieLens. The number of distinct tags appears to grow at a proportionally but not

[3] 10 week average ending November 30, 2008

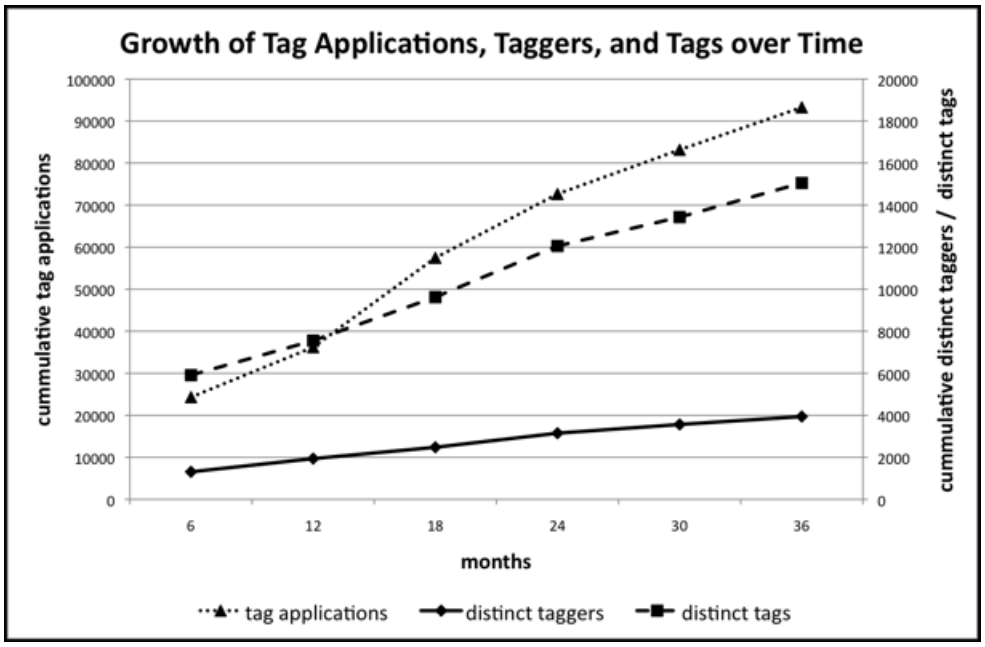


Figure 3.9: Growth of taggers, tags, and tag applications in MovieLens.

asymptotically slower rate that the number of tag applications. This finding contradicts the tag evolution on Delicious described by Cattuto et al. [Cattuto et al., 2006] and Golder et al. [Golder and Huberman, 2006], where tags grow at an asymptotically slower rate than tag applications. We hypothesize that our tagging community does not yet have enough tagging activity to exhibit asymptotic differences.

In the upcoming chapters we describe experiments that manipulate the MovieLens tagging interface to provide insight into users' tagging behavior. We also conduct offline analyses on data generated by MovieLens users.

Chapter 4

Vocabulary Evolution

4.1 Introduction

A critical characteristic of tagging systems that promote social navigation is their *vocabulary*, the set of tags used by members of the community. Instead of imposing controlled vocabularies or categories, tagging systems' vocabularies emerge organically from the tags chosen by individual members. This free approach contrasts with traditional expert-curated ontologies used in fields such as biology and library science [Uschold and Gruninger, 1996]. One valuable aspect of evolving vocabularies is that users invent personally meaningful tags, easing tasks such as organizing and re-finding items.

Individual invention, however, may not be best for the group as a whole. Social navigation may be more powerful in communities that share a common vocabulary. As an extreme example, people who speak different languages will find little value in each others' tags. User goals will also affect the value of others' tags. "Owned" is useful for remembering which books are in one's library, but not so helpful for others looking to discover new books to read. Even people trying to communicate the same idea often disagree how to describe it. Is your flavored carbonated drink a soda, a soft drink, a pop, a coke, or a tonic [Coye, 1994]?[1] The ESP Game [von Ahn and Dabbish, 2004] demonstrates how difficult it is for two people to agree on even simple descriptive words for a picture. In [Guy and Tonkin, 2006], Guy et al. suggest that correcting "sloppy

[1] See also http://www.popvssoda.com/.

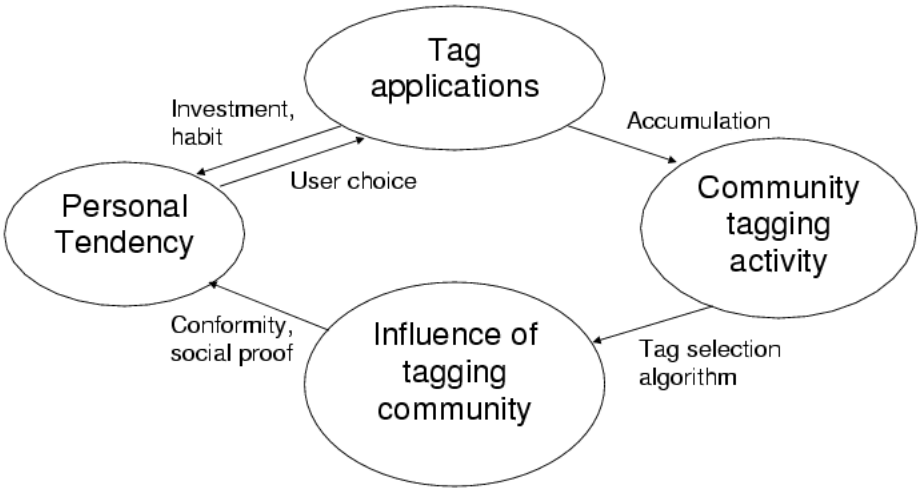


Figure 4.1: Relationship between community influence and user tendency. (From top, clockwise) tag applications accumulate to create a community's tags. Tagging systems use tag selection algorithms to select the tags shown to users. Based on these displayed tags, users form perceptions of community tagging norms. These norms influence the tags users apply. The tags users apply create an investment in a particular vocabulary.

tags" in a vocabulary can improve a tagging system's effectiveness.

In this chapter we examine factors that influence both the way people choose tags, and ultimately, the degree to which community members share a vocabulary. Figure 4.1 shows tag applications, and three factors that are likely to influence how people apply tags: people's *personal tendency* to apply tags based on their past tagging behaviors, *community influence* of the tagging behavior of other members, and the *tag selection algorithm* that chooses which tags to display.

Personal tendency. People choose tags based on their *personal tendency*, their preferences and beliefs about the tags they apply. New users have an initial personal tendency based on their experiences with other tagging systems, their comfort with technology, their interests and knowledge [Golder and Huberman, 2006], and so on. Personal tendency evolves as people interact with the tagging system.

Figure 4.1 indicates how users' own tagging behavior influences their future behavior through creating investment and forming habits. The tags one has applied are an investment in a personal vocabulary for organizing items. Changing vocabularies midstream is costly. For someone who has labeled Pepsi, Coke, and Sprite as "pop", it

would make little sense to label RC and Mountain Dew as "soda". Further, people are creatures of habit, prone to repeating behaviors they have performed frequently in the past [Ouellette and Wood, 1998]. Both habit and investment argue that people will tend to apply tags in the future much as they have applied them in the past. There are also other factors that might influence a user's personal tendency to apply tags. For example, they may become more knowledgeable about the items they tag. We do not model these factors in this chapter.

Community influence. Figure 4.1 suggests that the community influences tag selection by changing a user's personal tendency. Golder and Huberman find that the relative proportions of tags applied to a given item in del.icio.us appears to stabilize over time [Golder and Huberman, 2006]. They hypothesize that the set of people who bookmark an item stabilize on a set of terms in large part because people are influenced by the tagging behavior of other community members. Similarly, Cattuto examines whether the tags most recently applied to an item affect the user's tag application for the item [Cattuto et al., 2006].

The theory of social proof supports the idea that seeing tags influences behavior. Social proof states that people act in ways they observe others acting because they come to believe it is the correct way for people to act [Cialdini, 2001]. For example, Asch found that people conform to others' behavior even against the evidence of their own senses [Asch, 1951]. Cosley et al. found that a recommender system can induce conforming behavior, influencing people to rate movies in ways skewed toward a predicted rating the system displays, regardless of the prediction accuracy [Cosley et al., 2003].

Research questions. Unlike previous work that focuses on how vocabulary emerges around items [Golder and Huberman, 2006], we focus on factors affecting the way individual *users* apply tags across the domain of tagged items. Our first two research questions address the strength of the two factors we believe most affect the evolution of individuals' vocabularies:

RQ1: How strongly do investment and habit affect personal tagging behavior?

RQ2: How strongly does community influence affect personal tagging behavior?

To the extent that the community influences individual taggers, system designers have the power to shape the way the community's vocabulary evolves by choosing which tags to display. In the extreme case, a system might never show others' tags, thus eliminating community influence entirely. Even systems that do make others' tags visible will often have too many tags to practically display. Figure 4.1 shows the tag selection algorithm acts as a filter on the influence of the community. We ask two research questions about the effect of choosing tags to present:

RQ3: How does the tag selection algorithm influence the evolution of the community's vocabulary?

RQ4: How does the tag selection algorithm affect users' satisfaction with the system?

Finally, we examine whether communities converge on the *classes* of tags they use (e.g., factual versus subjective), rather than on individual tags. We explore whether these different classes of tags are more or less valuable to users of tagging systems:

RQ5: Do people find certain tag classes more or less useful for particular user tasks?

Our work differs from prior tag-related research in a number of ways. First, we focus on people rather than items. Second, we study a new tagging system rather than a relatively mature one. Third, we compare behavior across several variations of the same system rather than looking at a single example. Fourth, we study tagging as a secondary feature, rather than as the community's primary focus.

We believe that our perspective and questions will give fresh insight into the mechanisms that affect the evolution and utility of tagging communities. We use this insight to provide designers with tools and guidelines they can use to shape the behavior of their own systems.

The rest of this chapter is organized as follows. Section 4.2 presents our experimental manipulations and metrics within this tagging system. Sections 4.3, 4.4, and 4.5 address our first three research questions related to personal tendency, community influence, and tag selection algorithm. Section 4.6 covers research questions four and five, which explore the value of a vocabulary to the community. We conclude in section 4.7 with

a discussion of our findings, limitations, design recommendations, and ideas for future research in tagging systems.

4.2 Experimental Setup

Each user was provided with the common tagging elements described in Section 3.1. We now describe the experimental manipulations we performed to gain insight into our research questions.

We randomly assigned users who logged in to MovieLens during the experiment to one of four experimental groups. Each group's tags were maintained independently (i.e. members of one group could not see another group's tags).

Each group used a different tag selection algorithm that chose which tags to display, if any, that had been applied by other members of their group. We used these algorithms to manipulate the dimensions of tag sharing and tag visibility.

The **unshared** group was not shown any community tags, corresponding to a private system where no tags are shared between members.

The **shared** group saw tags applied by other members of their group to a given movie. If there were more tags available than a widget supported (i.e. three tags on the movie list, seven tags on the auto-complete list), the system *randomly selected* which tags to display.

The **shared-pop** group interface was similar to that of the shared group. However, when there were more tags available than a widget supported, the system displayed the *most popular* tags, i.e., those applied by the greatest number of people. Both the details page and the auto-complete drop-down displayed the number of times a tag was applied in parentheses. We expected this group to exhibit increased community influence compared to the shared group because, since everyone would see the most popular items, people would tend to share the same view of the community's behavior.

The **shared-rec group** interface used a *recommendation algorithm* to choose which tags to display for particular movies. When displaying tags for a target movie, the system selected the tags most commonly applied to both the target movie and to the most similar movies to the target movie. Similarity between a pair of movies was defined as the cosine similarity of the ratings provided by MovieLens users. Note that this means

Table 4.1: Overall tag usage statistics by experimental group. Note that the tags column overall total is smaller than the sum of the groups, because two groups might independently use the same tag.

group	users	taggers	tags	tag applications
unshared	830	108	601	1,546
shared	832	162	809	1,685
shared-pop	877	154	1,697	4,535
shared-rec	827	211	1,007	3,677
overall	3,366	635	3,263	11,443

that a tag that was never actually applied to a movie could appear as being associated with that movie–and further, that tags could be displayed for a movie that had never had a tag applied to it.

We collected usage data from January 12, 2006 through February 13, 2006. Table 4.1 lists basic usage statistics overall and by experimental group. During the experiment, 3,366 users logged into MovieLens, 635 of whom applied at least one tag. A total of 3,263 tags were used across 11,443 tag applications.

4.2.1 Metrics

As shown in Table 4.1, basic usage metrics differed widely between experimental groups. However, these differences are not statistically significant due to effects from "power taggers." Most of our research questions are not about differences in quantity, but rather, about how the tags people apply and view influence their future decisions on which tags to apply. In most cases, we study this influence at the level of categories of tags, which we call *tag classes*. Golder et al. present seven detailed classes of tags[Golder and Huberman, 2006]. We collapse Golder's seven classes into three more general classes that are related to specific user tasks that tags could support in the MovieLens community. We list short descriptions of Golder's tag classes that were folded into each of our tag classes in parentheses.

1. **Factual tags** identify "facts" about a movie such as people, places, or concepts. We operationally define factual tags as tags that most people would agree apply to a given movie. Factual tags help to describe movies and also help to find related movies (Golder's classes: item topics, kinds of item, category refinements).

Table 4.2: Ten most popular tags in each tag class, and how often each tag was applied.

Factual	**Subjective**	**Personal**
action (134)	classic (235)	bibliothek (253)
drama (104)	chick flick (61)	in netflix queue (177)
disney (86)	funny (60)	settled (148)
comedy (86)	overrated (54)	dvd (122)
teen (64)	girlie movie (51)	my dvds (110)
james bond (62)	quirky (39)	netflixq (87)
super-hero (57)	special (29)	get (58)
japan (56)	funny as hell (25)	ohsoso (48)
true story (55)	funniest movies (23)	buy (35)
crime (54)	must see! (22)	(s)vcd (32)

2. **Subjective tags** express user opinions related to a movie. They can be used to help evaluate a movie recommendation (item qualities).
3. **Personal tags** have an intended audience of the tag applier themselves. They are most often used to organize a user's movies (item ownership, self-reference, task organization).

In order to assign tags to classes, we manually coded the 3,263 distinct tags into one of the three classes. If tags were incomprehensible, or did not fit in a class, the tag was coded as class **other**. Each tag was coded by two people. Coders agreed on 87% of tags. When coders differed, the coders discussed the tag and reached a consensus.

The final distribution of tags across tag classes was 63% factual, 29% subjective, 3% personal, and 5% other. Unless mentioned otherwise, we ignore the class "other" when performing tag-class-based analyses. For each tag class, Table 4.2 shows the ten tags of that class applied most often, across groups.

We often define influence in terms of the cosine similarity between tag class distributions. By tag class distribution we mean the proportion across these three tag classes of a group of tags, tag applications, or tag views. Cosine similarity is useful because it normalizes for the size of the distributions.

For example, suppose we wish to talk about the community influence on a specific tag application by a user. We can treat the tags the user saw before applying that tag as a distribution across the three tag classes. Suppose that 62% of the tag views were of factual tags, 25% were subjective, and 13% were personal. Likewise, we can look

at the class of the tag applied and think of it as a tag class distribution. If the tag is subjective, the distribution would be 0% factual, 100% subjective, and 0% personal. We can encode these as vectors: $x = [0, 1, 0]$ $y = [0.62, 0.35, 0.13]$. We then compute cosine similarity of x and y as $\frac{x \cdot y}{\|x\|\|y\|}$, or $\frac{0*0.62+1*0.25+0*0.13}{\sqrt{0^2+1^2+0^2}\sqrt{0.62^2+0.25^2+0.13^2}} \approx 0.37$. If the tag applied had been a factual tag, then the similarity would have been about 0.91.

One disadvantage of using cosine similarity it that it can be hard to understand how to interpret differences between two similarity values. As a frame of reference, the similarity between the uniform tag class distribution $[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$ and any tag application is $1/\sqrt{3} \approx 0.58$.

Finally, upon completion of the tagging experiment we conducted a survey of all MovieLens users. A detailed description of the survey is presented in section 7. We include results from the survey as they are relevant.

4.3 Personal Tendency

We are now ready to explore our first research question:

RQ1: How strongly do investment and habit affect personal tagging behavior?

In the model in Figure 4.1, a user's personal tendency determines the types of tags they apply. In this section, we examine how strongly investment and habit affect user choices. We measure the strength of this association by comparing the tags a user has applied in the past to the tags they apply in the future.

The solid line in Figure 4.2 shows the average cosine similarity between the tag class distribution across the users first $n - 1$ tags and the tag class of their nth tag. We smoothed lines exponentially with weight 0.7. The horizontal line graphically displays the similarity of any tag application to the uniform tag class distribution. We will discuss the third line in section 4.4.

Once a user has applied three or more tags, the average cosine similarity for the nth tag application is more than 0.83. Moreover, similarity of a tag application to the user's past tags continues to rise as users add more tags.

As well as reusing tag classes, users also reuse individual tags from their vocabulary. Figure 4.3 shows that as users apply more tags, the chance that an applied tag will

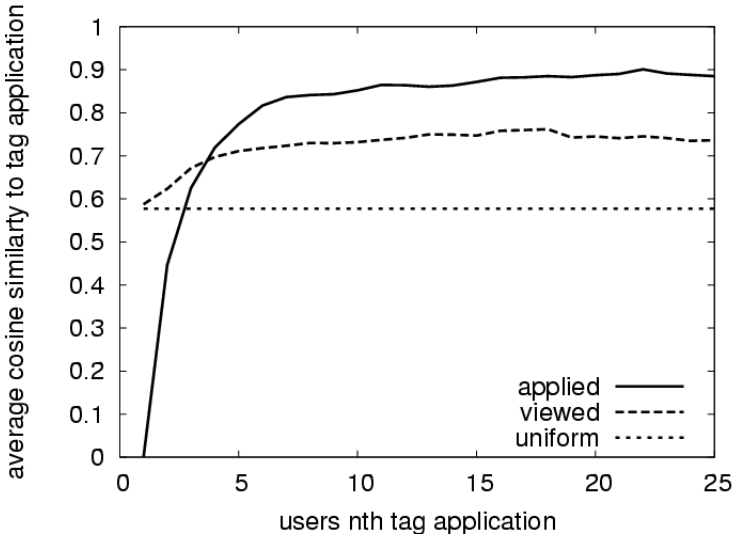


Figure 4.2: Similarity of tag class of the nth tag applied by a user to tag class distributions of other tags applied by the user before the nth tag (applied), of tags viewed by the user (viewed), and of the uniform tag class distribution (uniform). Both habit/investment and tags viewed appear to influence the class of applied tags.

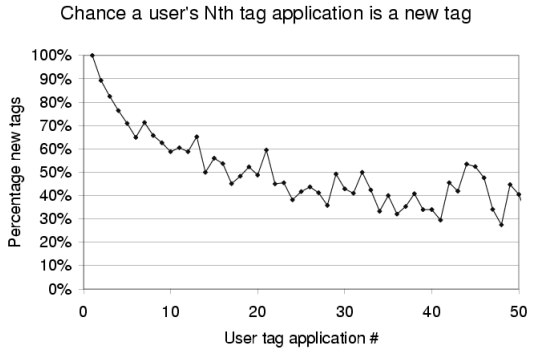


Figure 4.3: Chance a user's Nth tag application is a new tag.

be new for them drops. In total, 51% of all tag applications are tags that the tagger has previously applied (experimental groups are grouped together to increase statistical power). As a baseline, we determined through simulation that users randomly selecting tags without any tendency to repeat tags would have about 27% tag reuse.

Clearly habit and investment influence tagging behavior. We wanted to determine if these factors are entirely responsible for user behavior. If habit and investment were the only factors determining personal tendency, the law of large numbers implies that tag class distributions for the four experimental groups should converge as more users enter the system. Table 4.3 shows that this is not the case; at the end of the experiment, the tag class distribution for the four groups was very different. In Section 4.5, we showed that the ending tag class distribution in groups vary significantly, both from a user-centric, and tag-centric viewpoint.

We should note that our study examines only a portion of the feedback loop between personal tendency and tag applications. It may be possible to directly measure personal tendency through more invasive iterative surveying techniques. We leave such studies for future work.

We conclude by restating our findings related to a user's personal tendency:

1. Habit and investment influence user's tag applications.
2. Habit and investment influence grows stronger as users apply more tags.
3. Habit and investment cannot be the only factors that contribute to vocabulary evolution.

4.4 Influence of Tag Views

We now turn our attention to our second research question:

RQ2: How does the tagging community influence personal vocabulary?

Our model from Figure 4.1 shows that viewing community tags indirectly impacts a user's tag applications by changing a user's personal tendency. We measure this by comparing the tags a users saw before each of their tag applications. Note that while [Cattuto et al., 2006] analyzes community influence in del.icio.us from an item-centric (i.e. web-page-centric) point of view, we consider a user-centric analysis. Because we

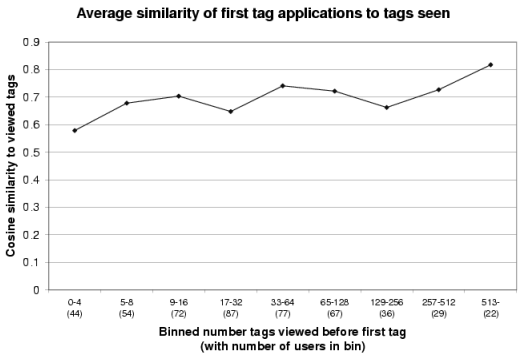


Figure 4.4: Average cosine similarity of the class of a user's first tag application to class distribution of tags viewed before the user applied the first tag. Results are grouped by number of tags viewed before the first application. Bins are logarithmic in an effort to put roughly an equal number of people in each bin.

control our tagging system, we can record the tags each user is shown while using the system (this is not possible with external analyses of del.icio.us).

During our experiments, we recorded all tags a user had viewed on the home page, search results, and "your tags" pages. Additionally, we recorded all tags displayed in auto-complete lists.[2]

As in our analysis of personal tendency, we measure the cosine similarity between a tag application's tag class and the tag class distribution the user *has seen* up until that point. This average similarity over user's nth tag applications is shown by the dotted curved line in Figure 4.2. Although the similarity between tag views and tag applications is weaker than the similarity between a user's personal tendency and their tag applications, it is stronger than the uniform tag distribution baseline.

We also examine how the number of tags viewed before a user's first tag application influences the choice of tags to apply. Figure 4.4 shows the average cosine similarity between a user's first tag class and the class distribution they saw before applying their first tag. A gentle upward trend is apparent; users who view more tags before their first tag application are more likely to have their first tag influenced by the community.

Based on our analysis, community influence plays an important role in vocabulary. In particular:

[2] Due to an implementation error, we failed to log tag views on movie details pages. However, we estimate that movie details pages account for less than 5% of total tag views.

Table 4.3: Final tag application class distribution by experimental group. The dominant tag class for each group is **bolded**. (Each row sums to 100%.)

Group	Subjective	Factual	Personal
Unshared	24%	38%	**39%**
Shared	**60%**	37%	3%
Shared-pop	9%	**82%**	9%
Shared-rec	20%	**67%**	12%

1. Community influence affects a user's personal vocabulary.
2. Community influence on a user's first tag is stronger for users who have seen more tags.

4.5 Choosing Tags to Display

We have shown that users are influenced by the community tags that they see. In our tagging model (Figure 4.1), the algorithm for choosing tags to display serves as the user's lens into community tagging activity. We explore this relationship in our third research question:

RQ3: How does the tag selection algorithm affect a user's personal vocabulary?

We examine algorithm influence using two approaches. First we explore the relationship between tag selection algorithms and resulting tag class distributions. Second, we examine the distribution of the actual tag phrases themselves.

4.5.1 Tag Class Distributions

We begin by looking at how tag display algorithms influence the distribution of tag classes (subjective, factual, and personal). We measure this influence by comparing the tag classes distributions between experimental groups, each of which had a different display algorithm. We consider both the final distribution, and the distribution as it varied across time during the experiment.

The final distributions have very large differences across our experimental groups. Table 4.3 shows the shared-rec and shared-pop groups are dominated by factual tags,

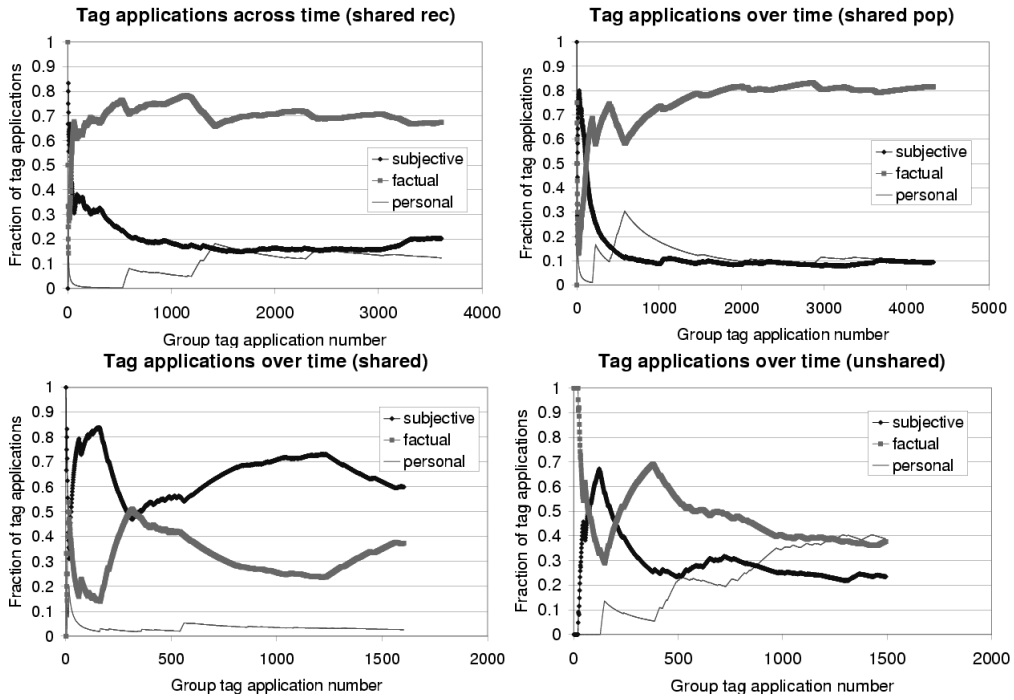


Figure 4.5: Tag class distribution over time for each experimental group: shared rec, shared pop, shared, and unshared, respectively. The first two appear to converge more strongly, to factual tags.

the shared group by subjective tags, and the unshared group is divided more evenly.

We compared the proportion of tag applications in each tag class across groups, and found the difference between groups to be significant ($\chi^2(6, N = 11073) = 2730$, $p < .001$). We wanted to know if particular power users were skewing tag-class distribution, so we also compared user-centric distributions. We compared the dominant tag class (i.e. the class with the most tag applications by the user) for all users with five or more tags across the experimental groups. The proportions of dominant tag classes between groups is again significant ($\chi^2(6, N = 168) = 123$, $p < .001$).

Differences in tag class convergence may be due to our experimental manipulation, community influence, or evolved personal tendency (though probably not initial personal tendency, since we randomly assigned users to groups). There are multiple possible explanations.

In addition to final tag class distributions, we looked at whether tag class distributions converged quickly, slowly, or not at all by examining per-group plots of tag class distribution over time (Figure 4.5). In all graphs, the X axis is the tag application number for the group, and the Y axis is the fraction of tag applications of the given class. The tag application number represents the number of tag applications by users in the experimental group since the beginning of the experiment (this can be roughly thought of as time).

Visually, it looks as if the shared-pop and shared-rec groups converged. In each, factual tags rapidly became and remained the dominant class by a large margin. By contrast, the shared and unshared groups have less visual evidence of convergence. In the shared group, subjective tags were often the dominant class, although there were more factual tags during tag applications 300-331, and there is more drifting in general. In the unshared group personal tags rise continuously until they become the dominant class at the end.

Likely the shared-rec and shared-pop display algorithms both favor tags applied by many different people, and those tend to be factual in nature (80% of tags applied by 5 or more people are factual). Perhaps this contributes to the greater number of factual tags in these groups. It is also perhaps interesting that these groups were the more convergent ones, and also had greater numbers of tags (see Table 4.1). This suggests that perhaps the interface strengthened convergence.

Finally, we note that these graphs represent a tag-centric view of tag class distribution, and power taggers may disproportionately influence these graphs. We also considered graphs that use a user-centric view of tag class distribution, where every user gets equal weight. Personal tag class proportions in the user-centric graphs are tempered due to the fact that, on average, personal tags are applied many more times (14.9) than factual (3.5) or subjective (2.6) tags. Since the user-centric view weighed each user equally, the actions of the few users who applied a personal tag many times were tempered by the many users who did not apply the tag at all. For example, while personal tags are most common in the tag-centric view of the unshared group, they are least common in the user-centric view. Other graphs show similar behaviors.

In summary, experimental groups exhibit different final tag class distributions and rates of tag class convergence. While we cannot definitively attribute these differences to

tag selection algorithms, we hypothesize that the shared-rec and shared-pop algorithms may encourage vocabularies to converge on factual tags, while the unshared selection algorithm encourages personal tag use by eliminating any motivation to create tags that are good for the community.

4.5.2 Tag Reuse

In additional to looking at tag class convergence, we would like to know if the convergence of actual tag phrases differs across groups. As a measure of tag convergence, we look at the average number of users who apply a tag. We chose this metric because it is more robust to power taggers than, for example, average applications per tag. Since every tag is applied by at least one user, the minimum value for this metric is 1.0. As a baseline, the unshared group averages 1.10 users per tag. The shared group follows with 1.27 users per tag. Next, the shared-pop group averages 1.31 users per tag. Finally, the shared-rec group, which exposed users to the largest number of tags during their use of MovieLens, yields 1.73 users per tag. Clearly the user interface has some effect on tag convergence.

Figure 4.6 breaks down origination of user tags (the first application of a tag by a particular user) based on the original creator of the tag. If the user is the first person to use a tag, we say they *invented* the tag. If somebody else in the experimental group used the tag, but the user has not seen the tag, we say that the user *reinvented* the tag. Finally, if the user saw the tag before applying it for the first time, we say the user *borrowed* the tag. For example, because the unshared group doesn't see other users it has no borrowed tags, but does have invented and reinvented tags. The origination results match our tag reuse metric: the shared-rec group uses more borrowed tags while the unshared group invents and reinvents more tags.

4.6 Value of Tags to the Community

In the previous three sections, we analyzed factors that contribute to vocabulary evolution in tagging communities. We now turn our attention towards exploring the value of a vocabulary to the community. We frame our exploration using our last two research questions:

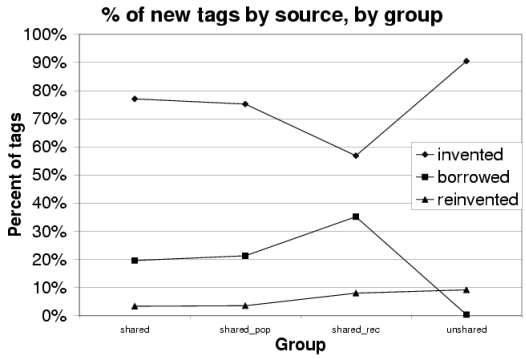


Figure 4.6: Origination of user tags, per experimental group. Re-invented tags are tags that are not seen by the user prior to creation, but were previously applied by other group members. The shared_rec group has more borrowed tags, and the unshared group has more invented tags.

RQ4: How does the tag selection algorithm affect users' satisfaction with the tagging system?

RQ5: Do people find certain tag classes more or less useful for particular user tasks?

We base our answers to these questions on a survey that we administered to MovieLens users. Before turning to the two research questions, we describe this survey in detail.

4.6.1 Survey Description

At the conclusion of our tagging study, we emailed 1,596 MovieLens users and asked them to complete a survey about their tagging experiences. The selected users comprised all taggers and non-taggers who had seen at least one tag, opted in to receive MovieLens emails, and had not received another MovieLens email in the past three months.

We divided the tagging survey into two main sections. In the first section, we asked general questions about a user's tagging experience, such as why they created tags and how much they liked the MovieLens tagging features. 365 users (23% of emailed users) completed this section of the survey.

In the second section, we asked users about specific tag applications. For each tag

application, the user was presented with the tag, the movie it was applied to, and a series of questions about the application. All users were asked about five tag applications created by other people:

- One tag application from each of the unshared, shared, and shared-pop groups.
- One actual tag application from the shared-rec group, where a user actually applied the tag to the movie.
- One inferred tag application from the shared-rec group where the shared-rec algorithm inferred a tag that had not ever been applied to the movie.

In addition to selecting one tag from each of the five groups, we ensured that the five applications spanned the three tag classes. Finally, for those users who were taggers, we asked questions about up to four of their own tag applications, again including at least one tag from each of the three tag classes.

327 users answered questions about at least five tags. After users answered their first set of questions about tag applications, they were given the option to continue answering questions about tag applications. 173 users answered ten or more sets of questions. In total, users answered questions about 3960 tag applications.

4.6.2 Mapping Tag Classes to User Tasks

We now skip forward to our fifth research question, which relates to some of our high-level survey results:

RQ5: Do people find that different tag classes are more and less useful for supporting various user tasks?

Hammond et al. suggest that reasons for tagging are generally application-specific [Hammond et al., 2005]. Based on prior experience with MovieLens users, we selected five user tasks related to tagging. In the tagging survey, we asked users whether they agree that tags are helpful for each of the user tasks. Below we list the user tasks as they were described in our survey, and show the percentage of taggers and the percentage of overall users that agreed that tags were helpful for the task.

1. *Self-expression* - I think the tagging features help me express my opinions. (50% of taggers agree / 30% of all users agree)

task	factual	subjective	personal
self-expression	39%/NA	**80%/NA**	20%/NA
organizing	62%/NA	61%/NA	**87%/NA**
learning	**60%/49%**	46%/36%	10%/7%
finding	**59%/48%**	35%/27%	12%/8%
decision support	41%/33%	**45%/35%**	13%/8%
overall	**56%/44%**	44%/31%	13%/9%

Figure 4.7: Usefulness of tag classes for user tasks. Percentages list agreement of responses by (only users who applied a at least one tag / all users). For each user task, the most highly rated tag class is bolded. The bottom row lists lists overall user satisfaction for each tag class.

2. *Organizing* - I think the tagging features help me organize my movies. (44% / 23%)
3. *Learning* - I think the tagging features help me know more about the movies to which they are applied. (37% / 27%)
4. *Finding* - I think the tagging features help me find movies I am looking for. (27% / 19%)
5. *Decision support* - I think the tagging features help me decide whether or not to watch the movie to which they are applied. (21% / 14%)

In addition to asking whether tagging supports the five user tasks in general, we asked whether each tag application supported the five tasks. The questions about learning, finding, decision support, and overall usefulness were asked about both tags the user applied and tags the users did not apply. We only asked questions about self-expression and organizing for tags a user had actually applied, since these tasks are most relevant to the tagger herself. Figure 4.7 details our results per tag class.

Figure 4.7 indicates that different tag classes are useful for different tasks. Factual tags are useful for learning about and finding movies. Subjective tags are useful for self-expression. Personal tags are useful for organization. Both factual and subjective tags are moderately useful in decision support.

The final row in Figure 4.7 gives results per tag class for overall user satisfaction with the tag. Users generally prefer factual tags and dislike personal tags. Additionally, users said they would prefer *not* to see 67% of personal tags they were asked about

(compare this to 27% for factual tags and 37% for subjective tags).

4.6.3 Differences by Choice of Tag Display

In section 4.5 we demonstrated that different tag display algorithms appear to lead to different tag class distributions for a community's vocabulary. We return to our fourth research question, which examines user satisfaction resulting from different tag display algorithms:

RQ4: How does the algorithm for choosing tags to display affect user satisfaction with the tagging system?

Users complained that the shared-rec tag selection algorithm resulted in an overly invasive tagging interface. Our choice of user interface may be partly to blame. In order to encourage tagging, we designed the tag input box to automatically pop open on movies users had rated. Furthermore, the auto-completion list automatically appeared, suggesting tags inferred by the algorithm. MovieLens users did not like these design decisions, perhaps because they interfered with other common user tasks such as rating movies.

Secondly, users did not like the tag inference algorithm itself. While users said they would like to see 36.5% of the actual tag applications in the shared-rec group, they only wanted to see 18.0% of the tag applications that were inferred using our algorithm. Users were confused by some of the inferred tags, and understandably so, because they were not informed that the displayed tags may not actually have been applied to the movie. For example, one user comments about the tag "small town" which was inferred for the movie "Swiss Family Robinson":

> I'm confused - I thought it was about people on a deserted island???

In addition, the algorithm led to a far higher number of tags being displayed in the interface. It generated 5,855,393 tag views, compared to 710,313 for shared-pop, 379,313 for shared, and 12,495 for unshared. 64% of surveyed users in the shared-rec said they would like to be able to hide tagging features - more than any other experimental group. However, it appears that the pervasive presence of tags had some effect in converting users to taggers. 25% of the users in the shared-rec group applied at least one tag,

compared to 19%, 17%, and 13% of users in the shared, shared-pop, and unshared groups respectively. A chi square analysis indicates that this difference is significant $\chi^2(3, N = 3,357) = 43.7, p < 0.001$.

In contrast, the unshared group had a relatively unobtrusive tag display. While 36.3% of users in the shared-rec group disliked tags overall, only 13% of users in the unshared group disliked the tagging features (along with 25% of users in the shared and shared-pop groups).

4.7 Discussion

4.7.1 Vocabulary Evolution

It may be desirable to "steer" a user community toward certain types of tags that are beneficial for the system or its users in some way. To this end, designers may wish to take advantage of our finding that pre-existing tags affect future tagging behavior. For instance, a new tagging system might be seeded by its designers with a large set of tags of the preferred type. Our results suggest that users would tend to follow the pre-seeded tag distribution. At the extreme, a site owner could seed a tag system with a nearly complete vocabulary of useful tags.

We point out several areas for further research related to tagging vocabulary. First, differences in users' attitudes towards different tag classes suggest that it may be valuable for tagging systems to classify tags. Researchers should investigate both automatic techniques to infer tag classes, and user interface designs that support manual classification of tags by the community.

Second, deriving relationships and structure from the tags that are applied may provide additional guidance in how to display tags in ways that aid search and navigation. Perhaps automated tools can be developed to help guide the emerging vocabulary. For instance, users could be steered to prefer the tag "soda" rather than "pop", if soda is being used heavily by other users. Perhaps the system could conflate the terms transparently, so that users could use either term effectively.

4.7.2 Other Issues

Our results point towards several guidelines for designers of tagging systems. First, some popular systems such as flickr do not support the notion of private tags. Hammond et al. argue in support of solely public tags [Hammond et al., 2005]:

> Social bookmarking tools, as with the Web at large, usually pay users back many times over in utility for whatever privacy they may have surrendered.

While users cite organizing their movies as one of the most important reasons for creating tags, they overwhelmingly dislike seeing others' personal tags:

> There should potentially be a private/public tag option. I don't really need to see how many people have a movie on their NetFlix list.

Therefore, we suggest that designers create affordances for hiding a user's personal tags from other users. In some tagging systems, other design dimensions may reduce the need for explicit personal tags. For instance, in del.icio.us, tags are public, but the most popular tags are chosen for display, so personal tags are unlikely to appear. Moreover, in common uses of del.icio.us, such as viewing one's own saved pages, or viewing an acquaintance's saved pages, tags from the rest of the community do not appear at all.

Recall that our user feedback suggests that tagging features should not be overly intrusive. In MovieLens, there are a subset of users who do not value tags, and would prefer to hide them entirely:

> Tagging is very heavy on the movielens user interface, and it would be good to be able to hide it. I can see their use when searching for movies, but most of the time I just look up a known movie to see its expected score...

One key difference between MovieLens and other tagging systems is that MovieLens is not primarily a tagging system. MovieLens exists to make recommendations, and users sometimes found the tagging features interfered with their primary goals. MovieLens has existed for over eight years, and adding new highly-visible features such as tagging was not welcomed by some long-standing users. Indeed, our survey results show that new MovieLens users were significantly less likely to want to hide the tagging features than users who existed before the features were introduced ($\chi^2(1, N = 248) = 7.6$, $p < .01$).

As tagging is increasingly added to existing systems, designers should consider the full range of use cases of their system.

One reason some users did not tag is because they could not think of any tags. This problem was cited by fully 68% of non-taggers in the unshared group, but only 40% of non-taggers in other groups. Offering tag suggestions is one way for designers to encourage more people to use tags.

Chapter 5

Tag Quality Interfaces

5.1 Introduction

MovieLens users only find 21% of tags to be worthy of general display.[1] Low quality tags cluttering an interface may be useless, or worse, they may be misleading, inappropriate, or offensive. Good tags, however, can tie entities together to enhance browsing or search, and they may serve as a source of descriptive information.

The lack of quality control on displayed tags is particularly dangerous given the self-reinforcing nature of tagging vocabularies. Conformity theory predicts that the tags that users see from other users will influence the tags that they in turn assign [Asch, 1956]. Conformity has been observed in practice. In the previous chapter, we saw that users tend to create tags resembling other tags they see in the community 4. Golder and Huberman [Golder and Huberman, 2006] and Cattuto [Cattuto et al., 2006] independently show that tagging vocabularies reach a stable equilbrium: once a tag becomes popular it remains popular. Systems that can select good tags not only improve the experience of the user who sees the tags, they also encourage those users to create good tags in return.

In order to design algorithms that automatically distinguish between high and low quality tags, we need to identify examples of high and low quality tags so that we can better understand the differences between them. In this chapter we explore interfaces

[1] 21% of tags in the tag quality survey we describe in Chapter 6 have an average rating of three stars or more.

that collect user feedback about the quality of tags. We explore several lightweight interfaces for collecting member feedback, and examine which interfaces lead to the richest data for understanding the quality of individual tags. We call direct feedback from users about a tag's quality *explicit* feedback. Rating interfaces that evaluate tag quality based on explicit ratings can only be effective for those tags that have been rated. Our first research question examines the relationship between rating interface and rating quantity:

RQ1: Which rating interfaces lead to the most ratings?

If users rate the same tag consistently, a few tag ratings may be enough for a tagging system to form an impression of the quality of a tag. Our second research question explores the agreement between ratings of a tag's quality:

RQ2: Do users agree on tag quality ratings?

In Chapter 6, we explore tag selection algorithms that use data collected from the interfaces we study in this chapter.

5.2 Methods

In order to study explicit tag feedback, we introduced tag ratings to the MovieLens community. Our design of a tag rating system was based on two guiding principles: users should be able to rate tags with a single click, and the ratings interface should require minimal screen space. Since a star-based rating system requires too much space, we selected a thumbs up / thumbs down rating system, similar to that used in many commercial applications such as Amazon[2] , TiVo[3] , and reddit[4] .

While many commercial applications incorporate both thumbs up and thumbs down ratings, several only employ one or the other. For example, BoardGameGeek originally employed thumbs up and down moderation, but shifted to only thumbs up moderation to "make it harder for people to gang up" and "reduce hurt feelings."[5] In sites such as

[2] Amazon.com uses thumb ratings for meta-reviewing.

[3] The TiVo digital video recorder collects user feedback through a thumb-based interface

[4] The news aggregation service reddit.com allows users to click an up-arrow or a down-arrow for each article.

[5] http://www.boardgamegeek.com/thread/156510

Figure 5.1: Tags as they appear on the MovieLens search results screen, next to the experimental thumbs up and thumbs down ratings widgets.

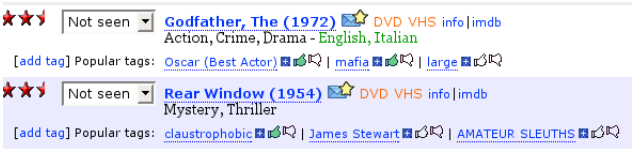
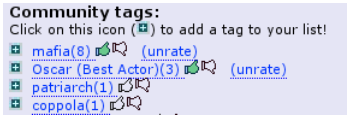


Figure 5.2: Tags and the experimental ratings widgets as they appear on the movie details screen.



YouTube, users provide positive feedback about items by marking items as "favorites." Other sites allow solely negative feedback. Users of Google Video, for example, may mark tags as "spam" but have no means of providing positive feedback. Even within the same domain, designers of commercial systems are divided on the issue. The social news site reddit[6] has both up and down arrows, whereas the social news site digg[7] has a "digg" button in a highly visible place on the interface, with a less visible "bury" button elsewhere.

To investigate the utility of different rating interfaces, we randomly split users into four experimental groups representing possible combinations of positive (thumbs up) and negative (thumbs down) ratings widgets:

1. Control group **C** was not shown any tag rating widgets.
2. Group **U** was shown only the thumbs up tag rating widget.
3. Group **D** was shown only the thumbs down tag rating widget.
4. Group **UD** was shown both the thumbs up and thumbs down tag rating widgets.

The tag rating interface appeared alongside all tags displayed on the MovieLens search results page (Figure 5.1) and movie details page (Figure 5.2). Search results pages displayed up to three tags per movie, while the movie details page displayed up

[6] www.reddit.com
[7] www.digg.com

Table 5.1: Statistics for each experimental group. The group with up and down ratings generated more positive ratings than the group with only up ratings. The number of raters in each group also varied significantly.

Group	**Num Users**	**Num Raters**	**Thumbs Up**	**Thumbs Down**	**Total**
control (**C**)	1494	0 (0.0%)	0	0	0
up only (**U**)	1576	80 (5.1%)	1325	0	1325
down only (**D**)	1581	153 (9.7%)	0	11903	11903
up and down (**UD**)	1600	227 (14.2%)	4027	9814	13841
total	6251	460 (7.3%)	5352	21717	27069

to twenty tags. MovieLens randomly selected and ordered tags for display from among the tags applied to a movie.

To help motivate users to provide tag ratings, we implemented simple user interface responses to rating actions. Tags shift to the front of a movie's tag list in response to a positive rating, and tags move to the end of the movie details page list and are hidden from the search results page in response to a negative rating. We incorporated AJAX javascript controls to enable fast, lightweight rating interactions. We enabled the tag rating features on January 21, 2007 and collected data for one hundred days.

5.3 Effects of the Rating Interface

Methods for selecting tags to display depend on data – either implicit data about user behavior, or data collected explicitly from users. Many explicit ratings-based systems find collecting sufficient data a challenge. Thus, a key question is which interfaces attract the most ratings. Our first research question is:

RQ1: Which rating interfaces lead to the most ratings?

Table 5.1 shows a summary of up/down ratings applied during the experimental period by users in the different groups. In total, 460 users (7.3% of active users during the time period, displayed in the Num Users column of the table) generated a total of 27,069 tag ratings. 72% of tag ratings occured from the search results page, while 28% occured on the movie details page. A small number of users supplied the majority of tag ratings. For example, the top rater provided 10.4% of all tag ratings (2,823), and the top 20% of raters provided 93.5% of all tag ratings (25,322).[8] 51.5% of raters

[8] This distribution is common in member-maintained communities. For instance, in Wikipedia the

applied 3 or fewer ratings.

5.3.1 Results:

The presence of different ratings interfaces leads to significant differences in ratings contributions. The descriptive statistics from Table 5.1 give an intuitive feel for the results. Users in Group UD rated more times (13,841) than users in Group D (11,903) or in Group U (1,325). Also, more users in Group UD rated one or more times (14.2%) as compared with users in Group D (9.7%) or Group U (5.1%). These differences are statistically significant. We also find that more users from Group UD contributed one or more tag ratings than from either of the other experimental groups.

Although on average, users in group D generated more negative ratings per-user as compared with users in group UD (means 7.53 vs. 6.13), this difference is not statistically significant. The difference in the means might be attributed to the presence of the most prolific rater in Group D, who singlehandedly rated 2,823 times.

Interestingly, we do find that users are more likely to rate tags positively in the presence of a thumbs-down rating widget. This is demonstrated by the fact that users in Group UD gave a thumbs up to an average of 2.5 tag applications, while users in Group U gave a thumbs up to just 0.8 tag applications.

We thought the additional up ratings in the UD group might be due to tag "churn" introduced by negative tag ratings (negatively rated tags disappear and the user is presented with additional tags to rate). To test this hypothesis, we measured the tag-specific probabilities that a displayed tag would be rated positively across both Groups U and UD. We then calculated each group's expected number of up ratings based on their displayed tags. We find that the number of up ratings in Group UD is 1.5 times the expected number, while the U group is half the expected number. Therefore, we cannot attribute the extra positive ratings in the Group UD to tag churn. Apparently there is something about the presence of both ratings in the interface that leads to more up ratings.

Overall, we find that the interface containing both up and down ratings widgets led to the greatest levels of contributions. We later return to the impact of these contributions on tag selection methods. However, the general message is that more contributions leads

most profilic 10% of users generate 80% of all edits [Voss, 2005].

to greater coverage, and therefore more successful interfaces for displaying high quality tags.

5.3.2 Analysis of Statistical Significance

Our first finding is that the presence of different ratings interfaces leads to significant differences in ratings contributions. Because the distribution of work per-user is strongly skewed, we must apply non-parametric statistical tests to determine differences. To measure the differences in per-user ratings between groups, we examine the ratings of all users who log in to the system during the experimental period. We test for differences using a one-way Wilcoxon test, and report significance based on the p-value resulting from a Chi-Square approximation. Users in Group UD rated more than users in Group D ($n = 3181$, means 8.65 versus 7.53, $ChiSquare = 14.64$, $DF = 1$, $p < 0.001$), and they also rated more than users in Group U ($n = 3176$, means 8.65 vs. 0.84, $ChiSquare = 75.85$, $DF = 1$, $p < 0.001$). Users in Group D rated more than users in Group U ($n = 3157$, means 7.53 vs. 0.84, $ChiSquare = 25.04$, $DF = 1$, $p < 0.001$).

We also find that more users from Group UD contributed one or more tag ratings than from either of the other experimental groups. To test for significance, we conduct a likelihood ratio Chi-Square test. We find that users in Group UD were more likely to rate one or more tags than users in Group D (14.19% vs. 9.68%, $ChiSquare = 15.47$, $p < 0.001$), and they were also more likely to rate than users in Group U (14.19% vs. 5.08%, $ChiSquare = 78.45$, $p < 0.001$). Users in Group D were more likely to rate one or more tags than users in Group U (9.68% vs. 5.08%, $ChiSquare = 24.84$, $p < 0.001$).

The difference in negative ratings per user between groups D (6.13) and UD (7.53) is not statistically significant using a Wilcoxon test ($n = 3181$, $ChiSquare = 0.05$, $df = 1$, $p = 0.82$). The difference in positive ratings per user between groups U (0.8) and UD (2.5) is significant using a Wilcoxon test ($n = 3176$, $ChiSquare = 36.24$, $df = 1$, $p < 0.001$).

5.4 Identifying Controversial Tags

Users obviously do not agree on all tags. We hoped to identify controversial tags by finding tags whose ratings had the highest entropies. Entropy measures the amount of

Table 5.2: Top 10 most controversial tags based on thumb ratings as measured by expected entropy.

tag	**expected entropy**	**up**	**down**
comedy	0.987	28	30
classic	0.986	25	24
stylized	0.983	20	21
nudity (full frontal)	0.980	18	20
romance	0.980	18	17
quirky	0.977	25	20
magic	0.974	18	15
animation	0.974	26	20
steven spielberg	0.973	12	12
sci-fi	0.972	14	17

uncertainty associated with a random variable, and is calculated by summing over all possible outcomes $x_1 ... x_n$:

$$H(X) = -\sum_{i=1}^{n} p(x_i) log_2 p(x_i). \tag{5.1}$$

In our application, we wish to measure the amount of disagreement in the thumb ratings for a particular tag. Thus, suppose, the tag *foo* has 2 positive votes and 3 negative votes. The entropy of the ratings for *foo* is

$$-0.4 \cdot log(0.4) - 0.6 \cdot log(0.6) = 0.97 \tag{5.2}$$

Now suppose that the tag *bar* has 20 positive ratings and 30 negative ratings. Since the ratio of positive to negative ratings is the same, the entropy will be the same.

Assume that we could collect an infinite number of ratings (evidence) for each tag. Based on the small amount of evidence for *foo* in our initial observation, we would probably not be surprised if the "true" ratio of the inifinite collection of ratings for *foo* turned out to be 0.2, 0.5, or 0.7. On the other hand, we may be more surprised if the ratio for *bar* jumped to 0.7. Thus, after our initial observation of 5 and 50 ratings, we may be more inclined to believe the level of disagreement in *bar* than *foo*.

A Bayesian approach to entropy calculation treats the up to down ratio itself as a random variable. If we assume that all up/down ratios for tags are equally likely (this is

not far from actual reality), then, given u up ratings and d down ratings, the probability of a particular ratio q being f is:

$$p(q = f|u, d) = \frac{f^u(1-f)^d}{\beta(u+1, d+1)}. \tag{5.3}$$

Where β represents Euler's standard Beta function.

Based on this probability calculation, we can calculate the expected entropy of the ratings by combining a "weighted average" of the entropies for all possible ratios f weighted by the probability of each f as calculated in equation 5.3:

$$\int_{f=0}^{1} p(q = f|u, d)\,(-f \cdot log\,(f) - (1.0 - f) \cdot log\,(f)) \tag{5.4}$$

Using this formulation, we get an expected entropy of 0.84 for the example with five votes and 0.96 for the example with fifty votes.

Table 5.2 lists the most controversial tags as measured by expected entropy. Controversial tags appeared to contain information that is already displayed in MovieLens (*comedy*, *sci-fi*, *steven spielberg)*, subjective (*classic*, *stylized*, *quirky)*, or about a controversial topic (*nudity* - *full frontal*).

5.5 User Agreement for Tag Quality Ratings

5.5.1 Intra-User Agreement

In MovieLens, users rate specific applications of tags to movies. For instance, Sally may rate the tag *zombies* on the movie "28 Days Later" positively, and she may later rate *zombies* on "Dawn of the Dead." If Sally rates applications of *zombies* consistently, her second rating of zombie may be wasting her valuable effort. To reduce Sally's effort a system might assume that she will rate the tag *zombies* positively for all movies.

We found that users generally rate the same tag consistently, regardless of the item it was applied to. 91% of thumb ratings for the same tag, by the same user, but for different items were identical. A second measure of intra-user agreement that we examined was the average Bayesian expected entropy within a user's rating for a tag (thumbs-up = +1, thumbs-down=-1). The mean expected entropy for these sets, was 0.53, significantly less than the overall mean expected entropy for all thumb ratings of

0.79 ($n = 10451$, $p < 0.01$). As a point of reference, an entropy of 0.53 corresponds to a split of 12% up ratings and 88% down ratings (or vice versa). Based on these findings, although systems may want to allow users to rate individual tag applications, they should interpret a rating for a tag application as strong evidence of a user's general feeling like for a tag.

5.5.2 Inter-User Agreement

Sally is not the only MovieLens user who likes the tag *zombies.* 81% of all thumb ratings for *zombies* in MovieLens are thumbs up ratings. If several raters agree on a tag's quality, a system may be able to conclude that most users have similar opinions of the tag, increasing the tag selection method's coverage for *all* users. In this section we explore the level to which different users agree on their ratings for a particular tag's quality.

To measure the rating agreement between users, we grouped all thumb ratings for a particular tag together, and took the mean Bayesian expected entropy across all sets of thumb ratings. The mean expected entropy was 0.71, more than the per-user expected entropy of 0.53, but less than the background entropy of 0.79 for all ratings ($n = 7718$, $p < 0.01$). As a point of reference, an entropy of 0.53 corresponds to a split of 19% up ratings and 71% down ratings (or vice versa).

We conclude that different users show some agreement about a tag's quality, but not as much agreement as within the same user.

Perhaps Sally provided MovieLens's fifth positive rating for *zombies* and no users had rated the tag negatively. This high level of initial agreement offers a promising signal for tag quality that can be easily implemented by system designers.

The previous four ratings for *zombie* may have all come from the same user (we know from the previous section that a user will generally rate a tag consistently). To be sure that the initial consecutive ratings are independent confirmation, a designer may want to require that they come from different users.

Based on these scenarios, we examine two agreement-based heuristics for determining tag quality. **Consec-apps** ranks tags based on the number of initial identical ratings. **Consec-users** is similar, but it requires that the ratings come from different users.

Figure 5.3: Percent of remaining ratings that, after an initial number of identical ratings, remain positive or negative.

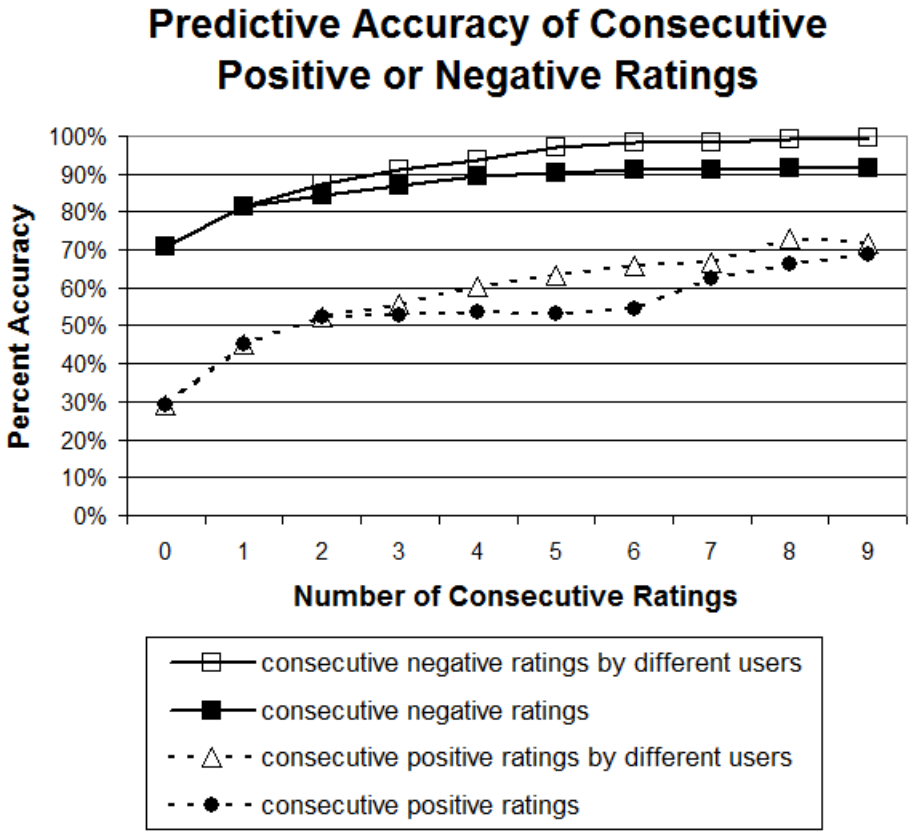


Figure 5.3 shows the percent of ratings that, after a certain number of initial consecutive positive or negative ratings, remain positive or negative. The graph presents both the count consec-apps and consec-users metrics. Both metrics for negative consecutive ratings serve as accurate predictors. After a tag receives four consecutive thumbs down ratings (regardless of user), 90% of the remaining ratings will be thumbs down. On the other hand, even after a tag receives strictly thumbs up ratings by 9 different users, only 71% of the remaining tags are positive.

5.6 Summary

In this chapter we analyzed the relationship between tag quality rating interfaces and the number of ratings elicited by the interface. We found that users generate more positive ratings when they could also rate negatively. We also found that users were more likely to rate at least one tag when presented with both positive and negative rating options.

We analyzed the level of agreement in ratings for tag quality, and found that a user shows strong agreement in her own ratings for a tag, and weaker agreement with other user's ratings for the tag. We introduced a new measure of disagreement (Bayesian expected entropy), which takes into account the amount of evidence supporting a tag's quality.

Based on our findings, tag systems choosing to collect feedback about tag quality should collect both positive and negative ratings. We also suggest that tagging systems interpret a user's rating for the quality of a tag application broadly as a signal of the quality of the tag in general.

Chapter 6

Tag Quality Algorithms

6.1 Introduction

In the previous chapter we showed that carefully designed interfaces encourage users to provide feedback about the quality of a tag. In this chapter, we explore methods for identifying high quality tags that should be displayed, while supressing low quality ones. These methods draw on the tag quality ratings we described in the previous chapter along with other signals of quality, such as searches for a particular tag.

Sites can control the quality of tags they choose to display by ordering the tags applied to a particular item by quality, and only displaying those with highest quality. Many tagging sites, including Amazon, Delicious, and LibraryThing use a simple method to order an item's tags: select the tags applied to the item most frequently. This intuitive method, which we call *num-item-apps*, is based on the assumption that the number of people who have added a tag to an item is a good estimate for how much other people will like to see that tag in the future. We call the method a tagging system uses to select and order the tags it displays its *tag selection algorithm*.

Despite the popularity of num-item-apps, other tag selection algorithms may also be effective. For example, if people search for high quality tags more often than low quality ones, systems that take into account the number of searches for a tag may improve upon num-item-apps. Moreover, num-item-apps does not apply to all domains. While a tag is a particular word or phrase, we define a *tag application* as an association between a tag and an item. Sites such as Amazon, Delicious, and LibraryThing that support

individual tag applications enable users to add their own copy of a tag to an item. On the other hand, sites such as Flickr and YouTube that support shared tag applications prevent users from re-adding a tag that was already applied to an item by somebody else. Since each tag is applied to an item only once, a site that supports shared tag applications cannot use num-item-apps.

The design of tag selection algorithms is important for two reasons. First, tagging systems can often only display a small fraction of all the tags applied by users due to limited screen space. Users who are shown good tags are likely to be better informed and more satisfied. For example, Delicious users have applied over 1,000 distinct tags describing Amazon.com. Should Amazon be *shopping*, *myfav*, or *GRDE226*. Second, as we saw in the Amazon example, tag selection algorithms can promote community norms. In the introduction we contrasted LibraryThing and Amazon's tags for Jonah Goldberg's book "Liberal Faciscm." Amazon's tag selection algorithm featured and therefore promoted subjective tags. Although Amazon's example encourages poor tagging behavior, system designers should see this as an opportunity. Carefully engineered tag selection algorithms can create and promote positive community norms.

In this chapter we explore novel tag selection algorithms for displaying high quality tags and hiding low quality tags. The same tag may have different quality for different users. For example, *my netflix queue* may be high quality to users who applied the tag (it helps them organize their movies), but low quality for everyone else. We define a tag as high quality for a particular user and item if the user wants to see the tag alongside the item.

We structure our exploration of tag selection algorithms around four research questions:

RQ1: Which metrics should researchers use when evaluating tag selection algorithms?

In order for us to empirically compare the performance of various tag selection algorithms, we need *metrics* that quantify algorithm performance. We evaluate several metrics that simulate the behavior of tag selection algorithms on offline data.

RQ2: How well do tag selection algorithms based on implicit user behavior perform?

Ideally, tag selection algorithms would draw on natural behavior already present in tagging communities such as tag application, tag browsing, and tag searching. We explore the effectiveness of tag selection algorithms drawing on these *implict* signals of tag quality.

RQ3: How well do tag selection algorithms based on explicit behavior perform?

Many community-maintained sites such as YouTube and Digg enable users to moderate contributions through explicit ratings. Inspired by these sites, we evaluate explicit moderation of tags using thumbs up and down ratings. Although these algorithms require more user effort than implicit algorithms, they may lead to improvements in tag selection algorithms that justify the extra user effort. We examine whether explicit ratings-based moderation mechanisms improve tag selection algorithms.

RQ4: How does the performance of each class of tag selection algorithms change as tag density increases?

MovieLens users generate far less tagging activity than a site such as Delicious, where users generate 400 tag applications in a matter of minutes.[1] Implicit tag selection algorithms rely on user tagging activity. Higher levels of tagging activity should lead to implicit signals based on more evidence, and therefore less noise. We explore the relationship between the performance of tag selection algorithms and the tag density of a site.

Finally, we deploy the best offline performer from each class of tag selection algorithm live on MovieLens. We evaluate users' reactions to each tag selection algorithm to see if the offline results reflect actual user preference in the live system.

6.2 Related Work

Several researchers have studied moderation in online communities. Cosley et al. find that "Wiki-like" systems that immediately display user contributions lead to more contribution than systems that require members to review contributions before they are displayed [Cosley et al., 2005]. In other work, Cosley et al. show that intelligent task

[1] Based on a recent examination of timestamps in Delicious's tag stream.

routing can be used to help users find tasks they might complete to improve the system [Cosley et al., 2006]. Lampe and Resnick analyze the moderation system utilized on the online forum slashdot[2] [Lampe and Resnick,]. They find that although the community perceives that forum moderations are generally fair, comments that are assigned low scores, or posted late in a conversation are often overlooked by moderators. Both Arnt and Zilberstein and Weimer et al. explore machine learning techniques for predicting moderation scores in online forums [Arnt and Zilberstein, 2003] [Markus Weimer, 2007]. Our research differs from the general work on moderation of contributions in that we focus on a type of contribution (tags) and test our results using a user-study.

6.3 Offline Evaluation

6.3.1 Experimental Methods

Although we believe that many users apply thumb ratings to indicate tag value, we wanted to collected a "gold-standard" set of tag value ratings for use in our analyses. The gold-standard allows us to gather unambiguous feedback about tag quality, and it enables us to analyze the relationship between the thumb ratings and actual tag quality. To collect this dataset, we emailed 2,531 active MovieLens users and asked them to complete an online survey in which they provide feedback on tag quality. Users were asked to rate up to twenty tags applied to five movies on a five star scale. We selected movies with a reasonably high-level of tagging activity to ensure that the implicit signals of quality were not to noisy. We wanted to select some movies the user had seen and others they had not. We also wanted to collect feedback from multiple users about a tag so that we could measure agreement. Based on these goals, we selected five movies:

- Two movies that users most frequently rated and tagged (*The Usual Suspects* and *Star Wars Episode IV - A New Hope*).
- Two randomly-selected, frequently-tagged movies that the user had rated.
- One randomly-selected, frequently-tagged movie that the user had not rated.

Figure 6.1 shows an example screen from the survey. As a point of reference, users were instructed that MovieLens would only display tags rated 3,4, and 5 stars. 577 users

[2] http://www.slashdot.org

Figure 6.1: We asked users to rate tags for five movies on a one-to-five scale. We instructed them that MovieLens would only choose to show them tags rated 3, 4, or 5 stars.

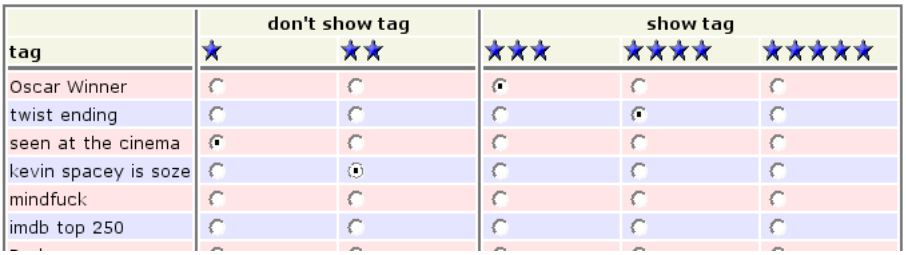


Table 6.1: Distribution of one-to-five tag ratings by rating value. The average overall rating is 2.17

tag rating value	1	2	3	4	5
percentage of ratings	46%	14%	21%	10%	7%

responded to the survey (22.8% response rate) and rated at least one tag application. 546 users rated tags for all five movies. We gave users the option of continuing to rate tags after they completed rating tags for their first five movies.

Users provided 74,987 one-to-five ratings. Two users provided more than 1,000 ratings, while 253 users provided 100 or more ratings. The distribution of the one-to-five ratings is shown in Table 6.1. Users deemed 38% of rated tag applications worthy of display. The average tag rating was 2.17.

We considered a variety of machine learning classifiers as tag selection algorithms. Each classifier used features of a particular tag, user, and movie as input (e.g. the number of users who have applied a tag), and outputted the predicted probability that a tag is high-value. We experimented with a number of different classification methods, including decision trees, logistic regression, and neural networks using the WEKA machine learning toolkit. We found that support vector machines (SVMs) outperformed other classification methods.

Because SVMs are known to have difficulty with features with differing numerical scales, we log-transformed the features that best fit a log-normal distribution.[3] Next, we

[3] We ran a chi-squared test for a normal distribution before and after log transforming each feature. We transformed the seven features whose χ^2 improved five times or more after the transformation: tag-length, apps-per-movie, num-item-apps, num-apps, num-searches, num-users, and num-search-users.

z-score normalized all features and clipped values more than three standard deviations from the mean.[4] We experimented with a vareity of different SVM kernels (linear, RBF, polynomial) and implementations, and chose svm_perf [Joachims, 2006] based on its linear kernel optimizations and diverse optimization criteria. We found that ROC area and top-n% optimization criteria outperformed the standard zero-one loss function. We use five-fold cross validation in all offline analyses.

6.3.2 Metrics

We experimented with three metrics that distill a tag selection algorithm's performance to a single score. Each metric evaluates whether a tag selection algorithm correctly classifies a survey rating as low value (two stars or less), or high value (three stars or more). This mapping from stars to value was marked on the key presented to users in the survey. In this section we explore:

RQ1: Which metrics should researchers use when evaluating tag selection algorithms?

[Herlocker et al., 2004] that metrics for recommender systems fall into one of two categories: accuracy-based metrics, and precision-based metrics. Inspired by this work, we explore the effectiveness of these metrics for tagging systems. *Classification accuracy* metrics examine the fraction of responses correctly classified as low or high value. *Precision* metrics examine the survey responses ranked highest by an algorithm, and measure the fraction that have high value. We consider one classification accuracy metric and two precision metrics:

Classification Accuracy (*class-acc*) measures the percentage of survey ratings that a tag selection algorithm classifies correctly as hide or display. Class-acc weighs each tag as equally important, regardless of whether the tag would be displayed by the selection algorithm or not.

Simulated Precision of an Item's Top-n (*item-top-n*) focuses on those tags a tag selection algorithm is most likely to display. Item-top-n groups survey responses by item and user, and then selects the most highly predicted n tags for each (user, item)

[4] Less than 1% of values were clipped for most features. The highest level of clipping for any feature affected 2.5% of its values. We clipped and normalized features to overcome the sensitivity of SVMs on inconsistent numerical ranges [Joachims, 2006].

pair. Item-top-n's score measures the percentage of the top n tags that are high quality. We set n to be three based on the number of tags MovieLens displays for a movie on the search result screen.[5]

Precision of the Top-n% (*overall-top-n%*) is a relaxation of item-top-n. We wondered whether a simplified version of item-top-n that might be easier to use in practice would yield similar results. Instead of grouping results by item, overall-top-n% orders all survey ratings by their predicted value regardless of item, and measures the percentage of the top-ranked n% that have high-value. Based on the percentage of an item's tags typically displayed in MovieLens[6] , we set n to be 25%.

For each of the 21 tag selection algorithms presented in the next section of this paper we calculate results using all three quality metrics. To compare metrics, we evaluated the Pearson correlation (similarity of numerical scores) and Spearman correlation (similarity of rankings) for the 21 scores between each pair of metrics. Although all three metrics ranked performance similarly, the actual numerical scores produced by the metrics differed significantly (Fisher's z-test $p \leq 0.05$). Overall-top-25% and item-top-3 yielded a Pearson correlation of 0.98 while the two other pairs yielded 0.89 and 0.85. While the Spearman rank correlation between overall-top-25% and item-top3 was highest (0.99), the difference between it and the other metric pairs were less noticeable (0.96 and 0.97).

Since item-top-3 closely simulates actual tagging system behavior, we found its numerical values to be easily interpretable. An item-top-3 score of 0.65 indicates that roughly two out of the top three tags displayed for a movie are deemed displayable. We can similarly interpret overall-top-25%'s numerical scores based on its strong correlation with item-top-3. On the other hand, class-acc scores were quite difficult to interpret due to its weightings of all tags equally. A naive predictor that classifies all responses as low-value yields a classification accuracy of 61%, while the best tag selection algorithm we consider in this paper yields 70% classification accuracy. It seems difficult for system designers to translate classification accuracy into expected system behavior.

Both overall-top-25% and item-top-3 have disadvantages. Item-top-3 requires slightly

[5] Search results account for 95% of all tag views.
[6] About 25% of the possible tag applications are displayed.

	rankings	**scores**	**tuning**	**implementation**
class-acc	high	low	low	low
overall-top-25%	high	high	medium	low
item-top-3	high	high	low	medium

Table 6.2: Qualitative Goodness of Different Metrics. The columns list the metric's quality of relative rankings for tag selection algorithms, the quality of numerical scores for algorithms, the difficulty of parameter tuning, and the difficulty of implementing the metric.

more implementation effort due to its increased complexity. The choice of n in overall-top-n% can be difficult. We set it to be the fraction of all tag applications that would be displayed. Although this worked well for us, it is difficult to know whether this approach will succeed in other domains. Table 6.2 summarizes the strengths and weaknesses of each metric. Despite its increased implementation difficulty, we favor item-top-n due to its interpretability and ease of parameter selection.

In the following section describing our offline analyses we report item-top-3 results. We do not report overall-top-25% due to its similarity to item-top-3, nor do we report class-acc due to the difficulty in interpreting numerical scores.

6.3.3 Tag Selection Features

Tag selection algorithms determine which tags to show based on certain criteria. For example, num-item-apps orders tags by the number of times they have been applied to an item. Although this simple algorithm serves as a de facto industry standard, tag selection algorithms may order tags based on other criteria such as the number of times the tag has been applied across all items (num-apps) or the number of users who have searched for a tag (num-search-users). We call these criteria *features*, and in this section we describe the features we evaluate in this paper. In support of RQ2 and RQ3, we group features by the "class" of tag selection algorithm they correspond to: implicit or explicit.

When describing each feature, we label it with its specificity. Some features such as num-search-users are broadly applicable to all instances of a tag, returning identical results regardless of the item to which they are applied. We describe this broad level of specificity as *per-tag*. Other features, such as the num-item-apps, return different results

for each (tag, item) pair. We describe the specificity of these narrower features as *per-item-tag*. Finally, we annotate features that are user-specific, such as a particular user's thumb rating as *per-tag-user* or *per-item-tag-user*. Narrow features capture a more specific usage of a particular tag (e.g. for a particular user, item or both), potentially offering a more precise signal of tag quality to tag selection algorithms. However, narrow features require a more specific behavior (e.g. a rating by a particular user instead of a rating by any user), and consequently need more activity to achieve the same level of coverage as broader features.

6.3.4 Implicit Features

Implicit features may improve the quality of displayed tags without additional user effort or interface modifications. Table 6.3 describes all 11 implicit tag selection features. We include two baselines: the de facto industry standard num-item-apps algorithm, and a random tag selection algorithm that arbitrarily orders tags. When presenting formulas for the features in Table 6.3 we use the following terminology: M is the collection of all movies, U is the collection of all users, and T is the collection of all tags. A is the collection of all tag applications. $user(ta)$, $tag(ta)$, and $movie(ta)$ denote the user, tag, and movie associated with a tag application. As shorthand notation, we denote the subset of A applied to a particular movie $\{ta \in A : movie(ta) = m\}$ as A_m. We similarly note applications applied by a particular user, and for a particular tag as A_u and A_t. We describe the set of all tag searches as S with similar subset notation. Below, we describe some additional details of the features.

The *tagshare* of a particular tag for a movie is the fraction of a movie's tag applications that are for the particular tag. For example, in MovieLens the movie "The Visitor" has been tagged with *immigrants* (by 2 users), *New York City* (1), *PG13* (1), and *R* (1). Since there are a total of 5 tag applications, *immigrants'* tagshare would be $2/5 = 0.4$, while *R* would have a tagshare of 0.2. Additionally, we smoothed each tagshare value by adding a constant of .0465 to the numerator and 5.0 to the denominator. We determined these constants by using an empirical Bayes methodology[7] [Carlin and Louis, 1997]. After smoothing, tagshare for *immigrants* and *R* would be

[7] We treated the fraction of a movie's tag applications that are a particular tag as a Bernoulli variable with a beta conjugate prior.

Table 6.3: Description of implicit features for tag selection.

feature	**specificity**	**motivating hypothesis**	**description**	**formula**	**top 5 results**
random	n/a	n/a	Randomly orders tags.	$random()$	n/a
num-item-apps	per-item-tag	Tags applied to a particular item by more users are more relevant.	Orders tags by the number of times they have been applied to a particular item.	$\|A_{m,t}\|$	*prison* (The Shawshank Redemption), *quentin tarantino* (Pulp Fiction), *anime* (Spirited Away), *time travel* (Twelve Monkeys), *holocaust* (Schindler's List)
num-apps	per-tag	Tags applied more times overall across items are more relevant.	Orders tags by the number of times they have been applied across all items.	$\|A_t\|$	*classic, tumey's dvds, less than 300 ratings, 70mm, r*
num-users	per-tag	Tags applied overall across items by more users are more relevant.	Orders tags by the number of users who have applied the tag across all items.	$\|\{user(ta) : ta \in A_t\}\|$	*classic, comedy, action, sci-fi, fantasy, funny*
num-searches	per-tag	Tags searched for more times are more relevant.	Orders tags by the number of searches for the tag.	$\|S_t\|$	*comedy, classic, oscar (best picture), seen more than once, action*
num-search-users	per-tag	Tags searched for by more users are more relevant.	Orders tags by the number of users who searched for the tag.	$\|\{user(ts) : ts \in S_t\}\|$	*comedy, oscar, classic, nudity (topless), sci-fi.*
tag-share	per-item-tag	Tags that account for a larger fraction of an item's tag applications are more relevant.	Orders tags by the fraction of the item's tags applications that are for the tag.	$\frac{\|A_{t,m}\|+0.0465}{\|A_m\|+5.0}$	*anime* (Perfect Blue), *why the terrorists hate us* (Bratz: The Movie), *serial killer* (Copycat), *jane austen* (Persuasion), *tom hanks* (The Money Pit)
avg-fraction-items-tagged	per-tag	Tags whose creators apply the tag many times are more likely to be list-making tags, and less relevant for the community as a whole.	Orders by the average, across all users, of the fraction of all the items tagged by the user that have the tag.	$mean_{u \in U} \frac{\|A_{t,u}\|+0.6}{\|A_u\|+5.0}$	*dvdhyllssa, dvd, tumey's dvds own, oppl, sven's to see list*
apps-per-item	per-tag	Tags applied more often to the items to which they are applied are more relevant.	Orders tags by the average number of times they are applied to their items.	$\frac{\|A_t\|}{\|\{item(ta):ta \in A_t\}\|+k}$	*pixar, mozart, johnny cash, quentin tarantino, monty python*
num-tag-words	per-tag	Tags with many words are less desirable.	Orders tags by the number of words in them.	$num\_words(t)$	45% of all tags are one word, 38% are two words, 11% are three words, and 13% are four or more words. Mini-reviews or comments account for many of the longest tags.
tag-length	per-tag	Tags with very few letters are less desirable.	Orders tags by the number of letters in them.	$num\_characters(t)$	The five most used tags with at most three letters: *r, dvd, own, war, vhs*

2.465/10 = 0.2465 and 1.465/10 = 0.1465 respectively. This smoothing reflects the possibility that the two initial applications of immigrants may be due to chance.

Avg-fraction-movies-tagged measures the average, across all users, of the fraction of all the movies tagged by the user that have the tag. For example, if a user created tag applications *dog* (applied to movies 1 and 2), *cat* (movies 2 and 3), and *mouse* (movie 4), the fraction of the user's tagged movies that are tagged with *dog* are 2/4 = 0.5. We smoothed this value by adding a constant of 0.6 to the numerator, and 5.0 to the denominator (we determined these constants using an empirical Bayes methodology [Carlin and Louis, 1997]). After smoothing the value of avg-fraction-movies-tagged for *dog* for our hypothetical user is (2+0.6) / (4+5.0) = 0.29.

Figure 6.2 compares the item-top-3 performance of tag selection algorithms based on each of the implicit features. 95% Confidence intervals are ±.0036 (n=74987, p ≤ 0.05)[8] . Differences between all neighboring pairs are statistically significant at the 0.05 level except for num-item-apps and tag-share, which achieve equal scores. All features outperform the random baseline (0.41). Num-item-apps, the most common feature among real-world systems, performs well, outperforming all features except for apps-per-movie.

Some features were skewed by a small number of users. For example, since num-apps orders tags by the number of applications, it predicts equal value for a tag that is used 3 times by 10 users and one that is used 30 times by one user. Because a few users can heavily influence num-apps, two personal list-making tags rank among the 10 most often applied tags: *tumey's dvds* and *less than 300 ratings*. These personal tags are generally disliked by people other than their creator[Sen et al., 2006]. A system can reduce the number of personal tags by normalizing each user's influence over num-apps. A simple method for normalization is to count the number of users who apply a tag instead of the number of applications. Due to this normalization, num-users and num-search-users outperform their non-normalized counterparts num-apps and num-searches .537 to .418 and .498 to .475 respectively according to item-top-3.

Figure 6.2 also includes a tag selection algorithm based on a combination of all implicit features (all-implicit). We constructed the all-implicit algorithm identically to

[8] 95% confidence intervals range from ±.0036 for scores near .5, ±.0034 for those scores furthest away from .5

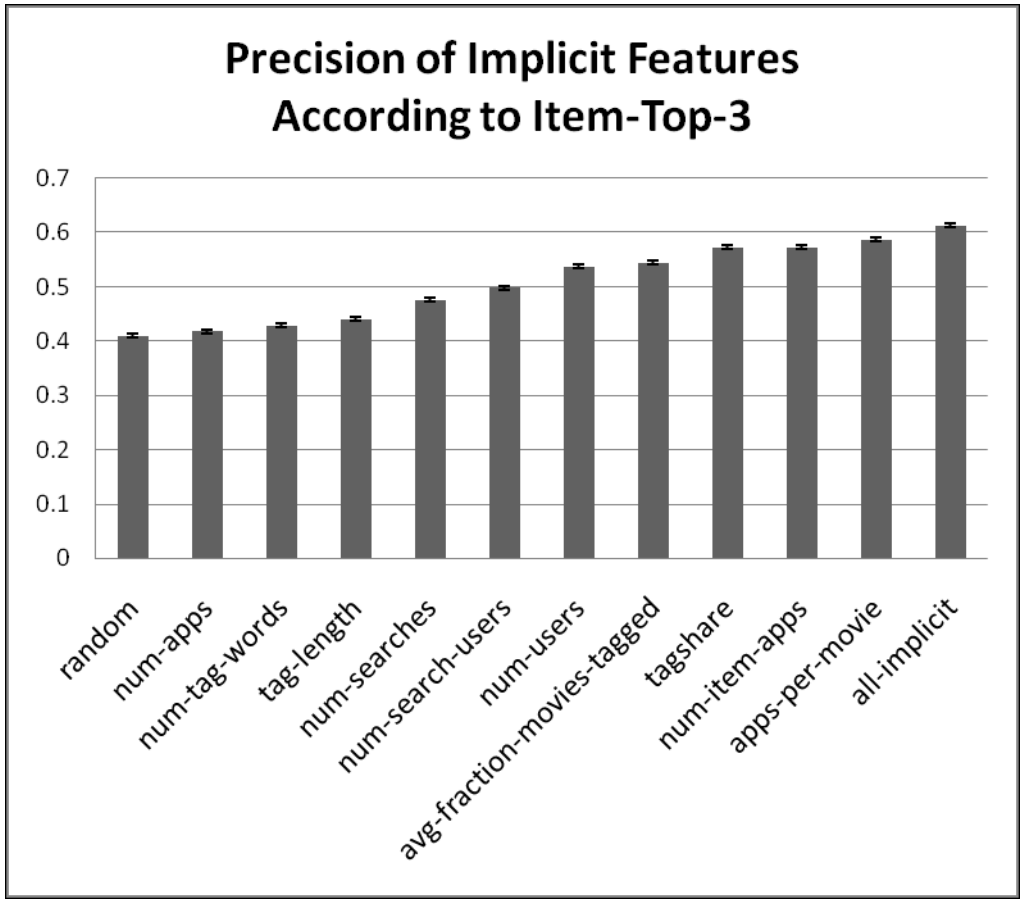


Figure 6.2: The performance of implicit features according to item-top-3. Differences between all neighboring pairs are statistically significant at the 0.05 level except for num-item-apps and tag-share, which achieve equal scores.

the single feature tag selection algorithms, but used multiple features as input into the SVM classifier. We found that a combination of five features (num-tag-users, num-item-apps, tag-words, tag-length, average-fraction-movies-tagged, apps-per-item) outperformed any single feature, achieving an item-top-3 score of 0.613. We determined the five top features by first selecting the top performing single feature, and iteratively adding the feature that most improved the current set of best performers. Adding more than five features did not provide a significant improvement.

We can now answer:

RQ2: How well do tag selection algorithms based on implicit user behavior perform?

To summarize our findings: num-item-apps, the feature most popular among real-world system designers, performs well among the implicit features. Apps-per-movie performs best among all implicit features, with a slight but significant edge over num-item-apps. The top performing tag selection algorithm is all-implicit, the combination of implicit features. All-implicit outperforms num-item-apps by 3%, which translates to one more high quality tag for every 33 tags displayed. We will return to RQ2 in the following chapter to study our algorithms in a live community.

6.3.5 Explicit Features

We now shift our focus to explore explicit features based on the thumbs up and thumbs down ratings. When presenting formulas for the explicit features we extend the terminology from the previous section. A single thumb rating is a data point containing four pieces of information: $< user, item, tag, \{\pm 1\} >$. R represents the collection of all thumb ratings for tags. As shorthand, we denote the subset of ratings applied to a particular movie $\{tr \in R : movie(tr) = m\}$ as R_m. We similarly note applications applied by a particular user, and for a particular tag as R_u and R_t.

We use $bayes\_frac\_up()$ to refer to a bayesian estimate of the fraction of a collection of ratings that are thumbs up. We found it important to use a bayesian estimate to adjust for collections of tags with few ratings. Ideally, instead of using a bayesian approach we would collect a large collection of ratings for a particular entity (i.e. the tag *funny*). We call the result we would if we collected a very large number of ratings

a tag's "true" fraction of thumbs up ratings. Unfortunately, collecting many ratings for *funny* is difficult with a limited population of users. Suppose instead that we have only received one thumb rating for *funny* and it was positive. We could naively estimate that the true fraction of thumbs up was 100%. However, based on our experience in MovieLens, that seems very unlikely. The bayesian paradigm provides a method for analytically encoding this uncertainty. In order to calculate the bayesian estimate, we treat the thumbs up and down ratings as a Bernoulli random variable with a beta conjugate prior [Gelman et al., 2003]. Using a gamma distribution for hyperparameters[9] we selected the beta conjugate prior most likely to produce the data. We repeated this procedure for four groupings of thumb data: by tag ($\alpha = 0.54$, $\beta = 1.74$), by user ($\alpha = 0.76$, $\beta = 2.15$), by user-tag ($\alpha = 0.27$, $\beta = 0.96$), and by movie-tag ($\alpha = 0.41$, $\beta = 1.42$). As an example, suppose a particular user has rated the tag *zombies* thumbs up three times and thumbs down once. We use the user-tag parameters ($\alpha = 0.27$, $\beta = 0.96$) to calculate a Bayesian estimate of the true fraction of thumbs up ratings: $\frac{3+0.27}{3+1+0.96} = 0.66$.

Table 6.4 describes all 7 explicit tag selection features. We now give details for several of the explicit features.

Normalized-global-avg is a user-normalized version of *global-avg*. We average each user's *user-avg*, the bayesian estimates for a user's average rating for a tag. Averaging in this way ensures that we limit a single prolific rater's influence over a tag's average.

Reputation estimates the value of a tag based on the reputation of the users who have applied the tag. The reputation of a user is computed as the average value of all the tags the user has applied. We use normalized-global-avg[10] as a measure of a tag's value.

We also experimented with three composite algorithms composed of multiple features. The first composite algorithm was a heuristic combining explicit features that is easily implementable by system designers called *hierarchical-value*. The underlying

[9] We chose a gamma distribution because its domain (all positive real numbers) covers the possible values of α and β for a beta distribution. We selected a shape parameter of 1.15 and scale parameter of 15.0 by visually tuning the distribution's probability density functions based on our domain intuition. Methods for choosing bayesian hyperparameters has been shown to have little impact over final solutions in other domains [Besag and Green, 1993].

[10] We chose normalized-global-average because it performed best among the features that were not composite.

Table 6.4: Description of explicit features for tag selection.

feature	**specificity**	**motivating hypothesis**	**description**	**formula**	**top 5 results**
global-avg	per-tag	Tags that are rated more highly have higher value.	Orders tags by the bayesian estimate of the fraction of thumbs up ratings.	$bayes\_frac\_up(R_t)$	*addiction*, *submarine*, *golden palm*, *superhero*, *james bond*
global-app-avg	per-tag-item	Specific applications of tags to a movie that are rated highly have higher value.	Orders tag applications by the bayesian estimate of the fraction of thumbs up ratings.	$bayes\_frac\_up(R_{t,m})$	*mafia* (The Godfather), *pixar* (Toy Story), *james bond* (Casino Royale), *time travel* (Back to the Future), *pixar* (Ratatouille)
user-average	per-tag-user	Users that rate a tag highly value that tag.	Orders tags by the bayesian estimate of the user's fraction of thumbs up ratings for the tag.	$bayes\_frac\_up(R_{t,u})$	not applicable (user specific)
user-rating	per-tag-item-user	Users that rate a tag application thumbs up value that tag application.	Orders tag applications by the user's rating for that specific application.	$R_{t,u,m}$	not applicable (user specific)
normalized-global-avg	per-tag	Similar to global-avg, but weights each user's ratings equally.	See description in text.	See description in text.	*superhero, studio ghibli, submarine, road trip, martial arts*
reputation	per-tag	Tags applied by users who create highly rated tags will have high value.	See description in text.	See description in text.	*dinosaurs, serial killer, martial arts, space, archeology*
hierarchical-value	per-item-tag	See description in text.	See description in text.	See description in text.	*rome* (Gladiator), *dinosaur* (Jurassic Park), *adolf hitler* (Downfall - Der untergana), *genocide* (Hotel Rwanda), *excellent script* (Juno)

principle of hierarchical-value is to begin with broad features early in a tag's lifecycle when little data is available about the tag. As more data is available about a tag, we shift to more specific features. The value of hierarchical-value for a tag application ta follows:

- If the user rated the tag, return $user\_avg(ta)$
- Otherwise, return a linear combination of $global\_app\_avg$, $normalized\_global\_avg$, and $reputation$ as follows:

$$\frac{\begin{array}{l}4.0 \cdot reputation(ta) + \\ 2.0 \cdot |R_{t,m}| \cdot global\_app\_avg(ta) + \\ 1.0 \cdot |R_t| \cdot normalized\_global\_avg(ta)\end{array}}{4.0 + 2.0 \cdot |R_{t,m}| + 1.0 \cdot |R_t|}$$

Hierarchical-value weights each of the three component features according to the amount of evidence for each of them. Since our evidence is thumb ratings, the weighting is a constant multiplied by the number of thumb ratings for the tag or item-tag. [11]

We also included two tag-selection algorithms that use SVMs. All-explicit is a combination of all simple explicit features. All-implicit-explicit is a combination of all implicit and explicit features. As opposed to the expert-tuned heuristic nature of hierarchical value, all-explicit and all-implicit-explicit use an SVM to automatically optimize a weighting for features. However, since we used linear kernels, the SVM cannot express relationships structures as complex as hierarchical-value. Thus, the SVM composites choose automatic linear optimization for many features over hand-engineered non-linear complexity.

Figure 6.3 shows the performance of all explicit features according to item-top-3. We also include the best implicit performers (num-item-apps, apps-per-movie, and all-implicit) and the random baseline. In general, explicit features significantly outperformed implicit features.[12] In fact two individual explicit features (normalized-global-avg, reputation) significantly outperformed the best combination of implicit features (all-implicit). As with implicit features, user normalized features (normalized-global-avg) outperform the non-normalized features (global-avg).

[11] We experimented with a number of different constants for each of the three components and found most to perform similarly. The chosen constants yield the best results among those combinations we tested.

[12] At the 0.05 level.

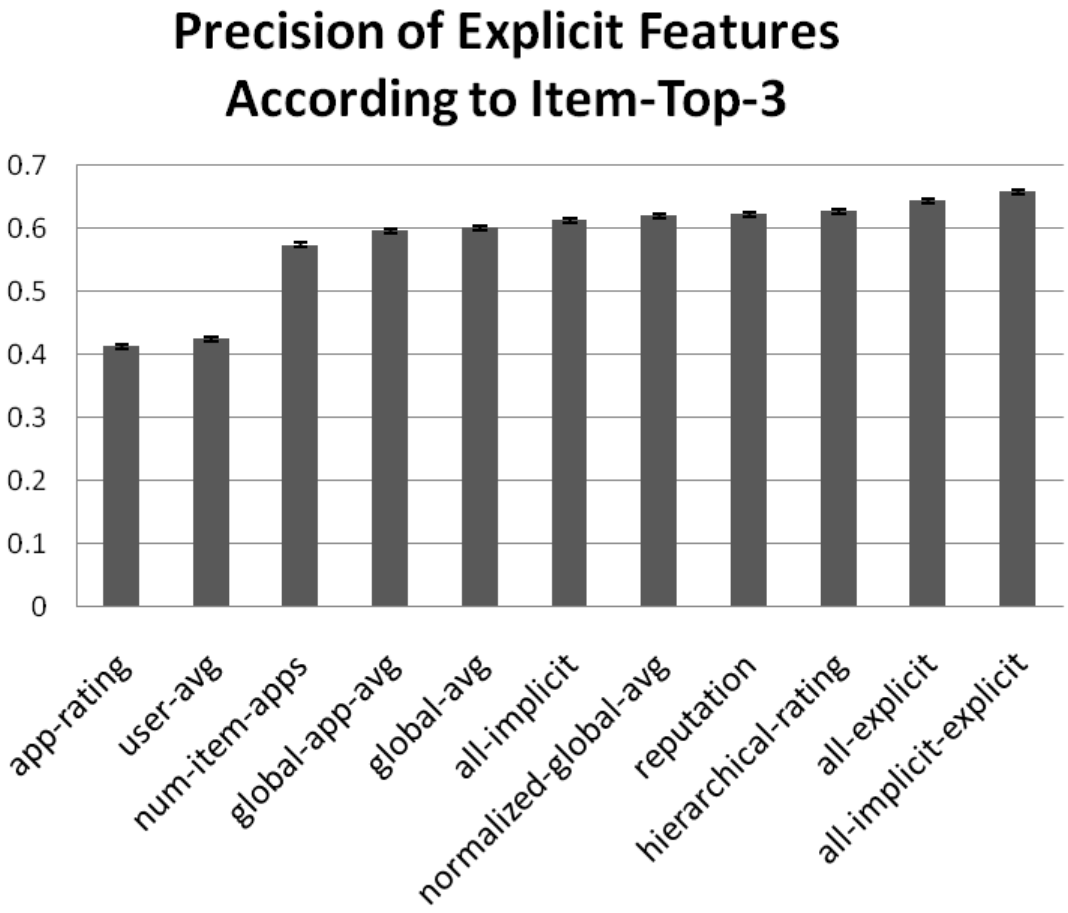


Figure 6.3: The performance of explicit features according to item-top-3. Note that the ordering of results is consistent between metrics. Differences between all neighboring pairs are statistically significant at the 0.05 level. App-rating and user-avg perform poorly due to their low coverage. Three explicit features (normalized-global-avg, reputation, and hierarchical-value) outperformed the best combination of implicit features (all implicit).

App-rating and user-avg performed poorly due to their low coverage. App-rating only generated a prediction for 0.8% of the survey responses, while user-rating only generated 4.3%. However, app-rating and user-avg did perform well when they generated predictions. 90.0% of the survey results associated with a positive app-rating value were high value tag applications. 80.0% of the survey results associated with a positive user-avg value were high value applications. Thus, the values of these features are important when they exist.

Reputation slightly but significantly outperformed normalized-global-avg. While both features performed similar on tags with a moderate to high number of ratings, reputation outperformed normalized-global-avg on those tags with few ratings. As an example, we calculated the reputation of every tag the moment it was introduced to the system, before it had any ratings. The top five tags were *serial killer, space, superhero, martial arts*, and *dinosaurs* while the bottom five were *dvd-ram, clv, cav, rent*, and *2.5*. 80% of *serial killer's* 261 eventual ratings were thumbs-up, while all 66 of *rent's* ratings were thumbs down. If MovieLens had incorporated reputation when tagging features were first introduced, it would have been able to identify *serial killer* as a high value tag and *rent* as a low value tag the moment a user first entered them into the system.

We now address:

RQ3: How well do tag selection algorithms based on explicit behavior perform?

As we mentioned earlier, several of the explicit features (normalized-global-avg, reputation, and hierarchical-value) significantly outperformed the best combination of implicit features (all-implicit). The best single explicit feature, hierarchical-value, outperformed the best combination of explicit features 0.626 to 0.613 for item-top-3. We also found that a tag selection algorithm combining all implicit and explicit features performed even better, with a score of 0.658 for item-top-3. Six features were necessary to achieve these results: hierarchical-value, reputation, user-normalized-global-tag-average, tag-share, user-tag-average and tag-length. We chose these features in an iterative greedy fashion, similarly to how we chose the features for all-implicit. We conclude that tag selection algorithms based on implicit and explicit features outperform those based on just implicit features using offline data. In the next section, we test our

offline results with a user study.

6.4 User Study

Although our offline analyses provide evidence for the performance of various tag selection algorithms, we had to make simplifying assumptions. For example, although we carefully designed our tag value survey, we cannot be certain that users' feelings for tag value would be similar in our survey and a live online community. Although the metrics exhibit consistency, we cannot be certain of the extent to which they reflect a user's experience. In this section, we explore the performance of the tag selection algorithms by deploying them live on MovieLens and observing user behavior.

We chose to deploy three tag selections algorithms for three months to ensure that there was enough data for our statistical analyses. We include num-item-apps to compare the algorithms to the most popular algorithm used by real tagging systems. We include implicit-only because designers of existing tagging systems can adopt it without adding any user interface elements. We include all-implicit-explicit for designers of systems who are willing to add thumb-based tag moderation. We do not include all-explicit because we assume that system designers would use the implicit signals at their disposal. We evaluate each tag selection algorithm by comparing the number of thumbs up and down received for tags displayed by the algorithm.

6.4.1 Methodology

We deployed each tag selection algorithm live on MovieLens, taking care to dynamically update each feature as users generated new data. For example, when a user gave a thumb rating to a tag we updated the explicit features, and when a user performed a tag search we updated num-searches.

We chose an experimental setup based on three requirements:

- *Control for users.* As we have seen, a few power users can significantly skew results. We avoided user-specific differences by showing a particular user different algorithms for different movies.
- *Control for movies.* Individual movies' characteristics such as popularity may

affect tag selection algorithms. We avoided movie-specific differences by using different algorithms for different users for a particular movie.

- *Ensure ordering consistency for a particular user and movie.* We want a particular user to see the same ordering of tags for a particular movie, regardless of when they visit the movie. We accomplish this by deterministically selecting experimental conditions based on a user and movie.

Based on these requirements, we decided to show each user a different algorithm for each movie. Our approach ensured that each user saw roughly the same number of examples from each experimental condition, each user consistently saw the same algorithm for a particular movie, and different users saw different conditions for the same movie. We chose a tag selection algorithm for a user id and movie id using a pseudo-random hashing function[13] that deterministically assigned a particular user id and movie id to an experimental group.

We ran the experiment for three months and two days starting in April 2008 (we explain the significance of the two days in the next section). During this time 5,695 users logged into MovieLens, and 592 users (10.4% of active users) applied 18,271 thumb ratings to tag applications. Two users rated 1,000 or more tags, while 366 users rated five or fewer. The num-item-apps experimental condition received 6,448 ratings, all-implicit received 6,190 ratings, and all-implicit-explicit received 5,633 ratings.

We use *thumbs up precision*, the fraction of thumbs ratings that are thumbs up, as the measure of a tag selection algorithm's effectiveness. We chose this metric based on its ease of interpretability, its similarity to our offline metrics, and its reflection of actual tagging system behavior.

6.4.2 False Start

Two days after launching the survey, we discovered an unexpected trend. The thumbs up precision was 26% for num-item-apps, 17% for all-implicit, and 14% for all-implicit-explicit. These results were ordered precisely opposite to what our offline analyses predicted. We carefully re-examined our algorithms, but found no errors.

[13] We used John Von Neumann's middle square method for generating random numbers. We chose this method because we need the numbers to be deterministic given user and item ids, and not dependent on ordering [Metropolis et al., 1953].

The cause of the backwards results lay in the user interface itself. When we introduced tagging in MovieLens, we displayed the number of users who applied the tag to the item alongside each tag. This design is similar to those seen on sites such as Amazon and LibraryThing. Num-item-apps, one of our experimental conditions, displayed the number of users who had applied a particular tag to an item. Thus, tags ordered by num-item-apps were consistent with the number displayed alongside the tags. We hypothesize that users interpreted the number shown alongside each tag as a signal of quality, and rated tags applied more times more highly. As a result, num-item-apps performed best. The implicit-only algorithm only slightly reordered the results, and performed second best. The implicit and explicit algorithm had the greatest divergence from num-item-apps, and performed worst. After two days, we realized the source of the backwards results and removed the numbers displayed alongside each tag.[14]

6.4.3 Results

After correcting the user interface, the thumb ratings from the user study were consistent with the offline results. 42.0% of ratings applied to tags displayed by num-item-apps received a thumbs up rating compared to 44.1% for all-implicit and 49.1% for all-implicit-explicit. All pairwise differences are statistically significantly ($p \leq 0.05$, n = 6277, 5996, 5433).

6.4.4 Effects of Different Levels of Tagging Activity

One key question is how the performance gap between implicit and explicit changes with the volume of tagging activity. Are sites with substantially more tagging activity likely to see different results than those on MovieLens? Our last research question explores this relationship:

RQ4: How does the performance of each class of tag selection algorithms change as tag density increases?

It seems plausible that tagging sites with an abundance of tagging activity, and therefore an abundance of implicit data, would be able to rely solely on implicit tag

[14] Although we don't include these two days in our analyses in the results for the following section, doing so would have little effect on them. The false start results account for only 5% of all thumb ratings during the experimental time period.

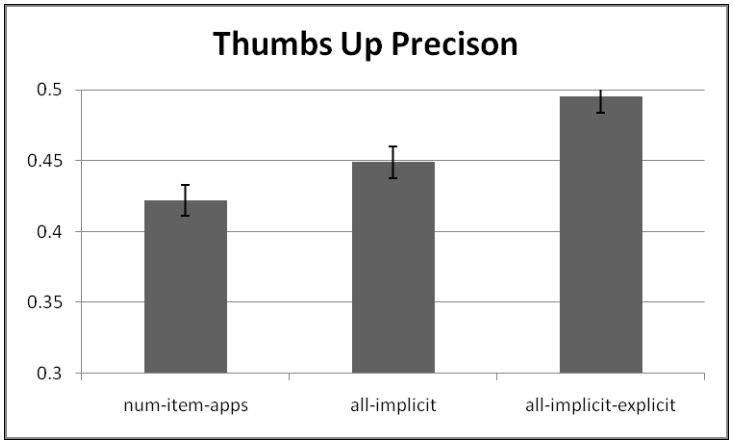


Figure 6.4: The fraction of thumb ratings received by tags displayed in each experimental group. These results support the offline results. Tag selection algorithms based on implicit and explicit data outperform those based on just implicit data. Algorithms based on implicit data outperform the industry standard. Differences are statistically significant ($p \leq 0.05$).

selection algorithms. To study the effects of increased activity on tag selection algorithm performance, we grouped our tag selection algorithm results by the number of tags applied to a movie. Movies with more tags overall simulate sites with more tagging activity. Movies with fewer tags overall simulate sites with less tagging activity.

Figure 6.5 shows the thumbs up precision of each class of tag selection algorithm grouped by the number of tag applications per movie. We used equal-count binning to choose the tag application ranges for the X axis. 95% confidence intervals range from 0.021 to 0.026. We generally find that performance increases with tagging activity. However, the fundamental results hold: implicit and explicit algorithm outperforms the implicit only algorithm (all differences are significant at the .05 level except for 26-42 apps). The implicit-only algorithm outperforms the industry standard algorithm (only the 77+ bin is significant).

The performance gap between the implicit and explicit algorithm and the other two algorithms seems to increase as tagging activity increases. This indicates that even sites with moderately high tagging activity (79-200 tag applications per item[15]) can achieve significant improvements by deploying thumb-based tag moderation. The performance

[15] 10 items in our dataset had more than 200 tag applications

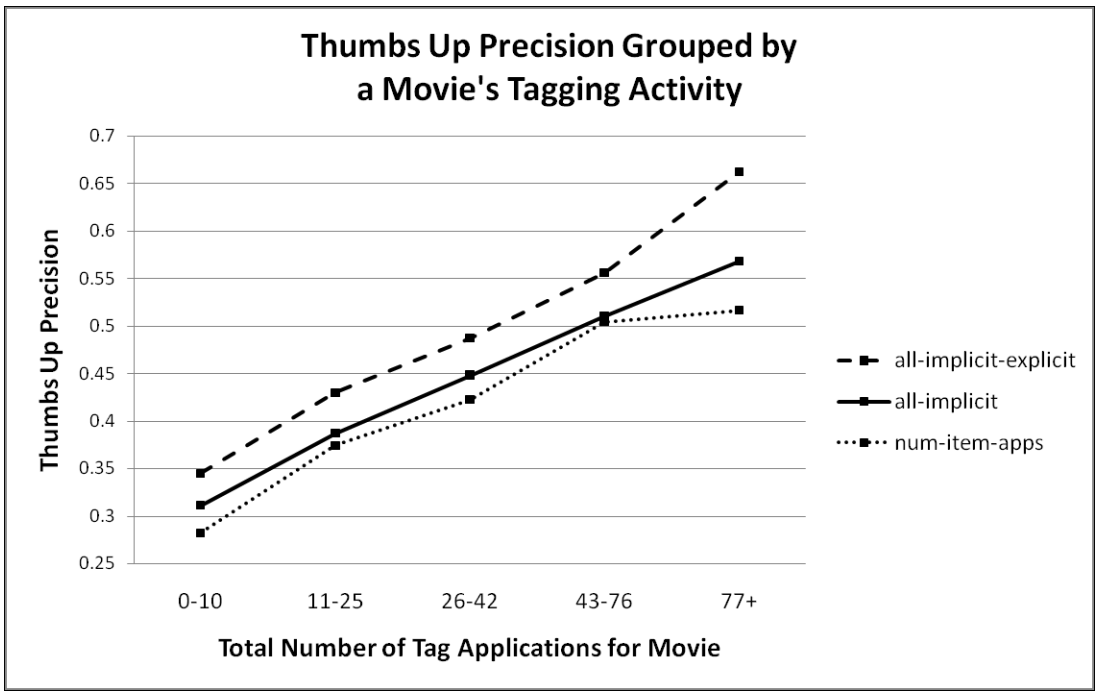


Figure 6.5: Thumbs up precision for different tag selection algorithms, grouped by the total number of tag applications for the movie. We used equal-count binning. The relative performance of the implicit and explicit tag selection algorithms seems to increase with tagging activity. This indicates that explicit tag moderation is valuable even for sites with a great deal of tagging activity.

of the algorithms beyond 200 tags per item is inconclusive.

6.5 Summary and Discussion

In this chapter, we analyzed tag selection algorithms that display high quality tags while supressing low quality ones. We presented a metric (item-top-3) for offline analysis of these algorithms that simulates real-world tagging system behavior. Using this metric, we showed that tag selection algorithms based on explicit ratings data outperform those based on implicit behavior, such as tag searches. We also showed that algorithms based on both implicit and explicit behavior perform best in offline analyses. Finally, we validated the results of our analyses through a user study of 5,695 MovieLens users.

In Chapter 4, we found that users create tags that resemble the tags they view. Thus, in addition to directly improving the quality of tags shown to users, our tag selection algorithms promise to change user tagging behavior in positive ways.

We offered two main contributions for designers. First, based on our exploration of metrics we recommend that system designers use the *item-top-n* metric when evaluating tag selection algorithms. Second, we have demonstrated in both offline and online analyses that tag selection algorithms combining a variety of implicit and explicit signals of tag value perform best. We encourage designers to include explicit mechanisms for users to provide feedback about a tag's value.

Our results indicate that even sites with relatively high tagging activity may benefit from using all-implicit-explicit algorithm. Although we hypothesize that the all-implicit-explicit algorithm continues to add benefit for items with more than 200 tag applications, more research is needed to be certain. Regardless, active sites should benefit from using the algorithm for less popular items. Even Delicious, one of the most popular tagging sites, has fewer than 200 tag applications for most items[16]

Our analyses focused on systems with personal tag applications. Although some of the features we explore require individual tag applications (apps-per-movie, tagshare) many others apply to sites with shared applications. To study the performance of our techniques in shared systems, we simulated a shared system by only retaining the

[16] We randomly selected 500 recently tagged URLs and found that 233 (53%) had fewer than 200 tag applications.

first tag application of a tag for an item. The item-top-3 value for all-implicit-explicit, our best performer only dropped by 0.003 to 0.655. These results suggest that machine learning-based tag selection algorithms can be adapted for sites with shared applications. Future researchers should further investigate the performance of tag selection algorithms in domains with the set model for tag applications.

Although our user study highlighted statistically significant differences between algorithms, we wonder about how users perceive these differences. In the user study all-implicit beat num-item-apps by about 3%. This difference translates to one extra good tag per 33 tags displayed. All-implicit-explicit beats num-item-apps by 8%, translating to an extra good tag per 12 tags displayed. While we suspect that this larger improvement is noticeable, more research is necessary.

A few features that had a small (but significant) impact on performance may still be noticeable to users. User-average, for example, was quite accurate when it was available. Moreover, it provides immediate feedback in response to a user's rating for tag. For these reasons, we believe that designers should incorporate user-specific features such as user-average. Based on these results, we believe that users may benefit from more complex personalized algorithms, such as algorithms that leverage patterns in tag value among subgroups of users.

During the "false start" in our user study, we found that a tag's context influences users' perceptions of the tag's quality. Tags applied more often were deemed higher quality. The interpretability of num-item-apps standard makes it easy for designers to add context consistent with its ranking of tags (the number of times a tag has been applied). It seems more difficult to add explanations for composite features such as all-implicit-explicit. One option designers might consider is to display a score alongside each tag. When users click on a score, systems could display an explanation of the individual features contributing to the score.

Chapter 7

Tag Preference

7.1 Introduction

According to the tag selection algorithms from Chapter 6, the highest quality tag application in MovieLens is *serial killer*, applied to the movie "Copycat" by five MovieLens users. One hundred twenty one users have rated *serial killer* as a high quality tag. Despite users' general agreement that *serial killer* is a high quality tag, not all users like movies about serial killers. For example, Alice (a pseudonym for a real MovieLens user), believes that *serial killer* is a high quality tag, and rated the tag quality thumbs up. However, Alice dislikes movies about serial killers.[1] Alice's dislike for the topic *serial killer* may explain her one star rating for Hannibal.

In this section we explore *tag preference*, which we define as a user's liking for items with a particular tag. It seems plausible that tag quality affects tag preference. For example, it may be difficult for a user to judge her preference for a low quality tag: how should one feel about *Tumey's DVDS*? We begin by examining the relationship between tag preference and tag quality:

RQ1: How are tag value and tag preference related?

We then explore algorithms that infer users preference for tags based on their interaction with a tagging system:

[1] Alice rated her liking for movies about serial killers one out of five stars in a survey we describe in the following section.

RQ2: Can we infer a user's preferences for tags?

We consider *tag preference inference* algorithms that analyze a user's interactions with tags or movies. We refer to a user's interactions with tags or items as *signals* of her interest in the tag or item. For instance, Alice's application of the tag *shipwrecked* may suggest that she is interested in *swashbucklers* (a signal of tag interest). In addition, Alice's rating of 4.5 stars for "The Mask of Zorro" and her click on a hyperlink leading to the movie "The Pirates of Penzance" (signals of item interest) may also have been a result of her liking for *swashbucklers*.

We evaluate eleven tag preference inference algorithms using 118,017 tag preference ratings collected in a user survey on the MovieLens movie recommender website.

We believe this work to be important for two reasons. First, successful tag preference inference algorithms may describe a user's interests through tags that are meaningful to other users. These interest profiles may help users understand each others interests, and may help them find like-minded individuals. Second, a user's tag preferences may be used to infer her preferences for items. As we discuss in Chapter 8, these tag-based recommendation algorithms may lead to more flexible recommendation systems based on the dimensions of items that users find most important. Much of the research we present in Chapter 7 and Chapter 8 appears in [Sen et al., 2009].

7.2 Experimental Datasets

We conduct our analyses using five sets of data collected from the MovieLens website. Table 7.1 presents statistics for each dataset. We now describe the data contained in each dataset along with details not specified in the table.

Movie Ratings: MovieLens users rate movies on a one to five star scale.

Movie Clicks: We logged clicks on links to detailed information about a particular movie for approximately 17 months starting in December 2006.

Tag Applications: Tags applied by users as described in section 3.3.

Tag Searches: Tag searches are textual searches for tags, or clicks on tag hyperlinks. 1,000 users have searched for at least five distinct tags. 107 users have searched for at least 50 distinct tags.

In our model for tag-based recommendation, we first infer users' preferences for tags.

Table 7.1: Size of different datasets we use in this chapter. Count is the number of the entities the dataset contains. Num-users is the number of users that generated those entities. For example, the first two columns in the third row indicate that 84,155 tags have been applied by 3,582 users. The last two columns indicate the same numbers after the pruning we apply for our analyses in the second half of this paper.

	before pruning		**after pruning**	
dataset	**count**	**num-users**	**count**	**num-users**
movie ratings	15,395,368	162,556	1,720,390	5,637
movie clicks	552,078	11,997	343,711	5,637
tag apps	84,155	3,582	65,229	2,105
tag searches	48,031	3,314	31,148	1,968
tag pref ratings	118,017	995	n/a	n/a

In order to evaluate our tag preference inference algorithms, we conducted a survey of tag preferences for MovieLens users. We emailed survey invitations to 8,361 active users. 995 users responded (an 11.9% response rate).

In the survey, we showed each user a collection of tags, and asked them to "estimate how much" they "would like movies with each tag" using a one to five star scale, or unsure. We asked each user to complete at least 60 tag ratings, but gave them the option to complete more if they wished. In total, users supplied 118,017 ratings for 9,889 distinct tags (mean 117 ratings per user, median 78). 800 users completed the requested 60 ratings, while seven provided more than 1,000 ratings.

The breakdown of the survey responses by star ratings is as follows: 6% (7,641) were 5 stars, 17% (20,597) were 4 stars, 22% (26,499) were 3 stars, 13% (15,135) were 2 stars, 13% (15,155) were 1 star, and 28% (32,990) were unsure. The average tag rating was 2.89. The unsure rating was used differently by different users. Among the 25% of users who used the unsure rating most often, unsure ratings accounted for 43% of ratings. Among the 50% of users who used the unsure rating least often, unsure ratings accounted for only 18.3% of ratings.

Pruning: Although we draw on all data when analyzing tag preference inference algorithms in Section 7.4, we pruned the data sets for our analyses of tag-based recommendations in Section 7.5. In order to reduce the computational requirements of our analyses, we focused on a set of movies with a minimum threshold of tags, and a set of users with a rich profile of MovieLens behavior. We began pruning the tag-recommendation dataset by selecting movies that had been tagged with at least five distinct tags. We wanted to focus on tags that represented concepts applicable to

multiple movies, so we required that each tag be applied to at least five movies. We iteratively repeated th is pruning until we reached a stable set of movies and tags. After movie ratings, movie clicks are the most abundant source of behavioral information we have for MovieLens users. Since we wanted to explore the effectiveness of tag-based recommendations for domains without tag ratings, we only included users that had clicked five or more movies.

After pruning, 1,720,390 ratings remained from 5,637 users for 2,636 movies. Since applying a tag may indicate interest in a tag, we also track tag applications created by users in the pruned set. In total, 1,315 users in the pruned set applied 50,060 tags (mean of 38 tags per user, median of 2). More statistics are shown in Table 7.1

7.3 Preference And Quality

In this section, we address our first research question:

RQ1: How are tag quality and tag preference related?

In order to explore the relationship between tag quality and preference, we constructed a co-occurence matrix for user responses along tag preference and tag quality. For each tag preference survey response we look to see if the user who provided the response also provided a thumb rating for the tag's quality. We treat tag preference ratings of one and two stars as a low rating, and those rated four and five stars as a high rating. We do not include three star ratings in the analysis. If the user has provided multiple thumb ratings for the tag, we use the mean thumb rating.

Table 7.2 shows the co-occurence matrix for tag preference and quality. Tag preference strongly correlates with tag quality. After ignoring three star and "unknown" preferences, 80% of the tags rated high in preference (4 or 5 stars) are also rated high in quality (thumbs up). The converse is also true. 83% of the tags rated low in preference (1 or 2 stars) are also rated low in quality (thumbs down). Tag preference responses labeled "unknown" also showed a skewed quality distribution. 77% of the "unknown" tag preference ratings had low quality ratings.

We were surprised that users associated more low quality tags with high preference, than high quality tags with low preference. We expected that users would identify some high quality factual tags capturing an import facet of a movie that they didn't enjoy.

Table 7.2: Co-occurence matrix between tag preference and tag quality.

	tag preference star rating			
	1 or 2	**3**	**4 or 5**	**not sure**
thumbs down	459 (19%)	291 (12%)	153 (6%)	492 (20%)
thumbs up	90 (4%)	195 (8%)	601 (25%)	145 (6%)

Table 7.3: Breakdown of tag preference survey responses that indicated high preference by the user (star rating), but low quality (thumb ratings).

category	**percent**	**description**
inaccurate	13%	an inaccurate tag (*aliens* for "Alladin").
redundant	26%	a tag that is redundant with the existing movie information (*tarantino*)
personal	4%	a personal tag (*seen more than once*)
subjective	16%	a subjective tag (*funny*)
other	41%	no explanation for the preference / quality inconsistancy.

For example, Alice agrees *serial killer* is a high quality tag for "Hannibal", but she generally dislike movies with violence. In order to better explore this relationship, we coded the 153 survey responses with high preference and low quality.

Table 7.3 shows our taxonomy for tag preference responses labeled high preference, but low quality. Some of the inconsistencies seem to result from specific relationships between a tag and a movie:

- **Inaccurate tags:** 13% of the high preference / low quality responses correspond to tags applied "inaccurately" such as *aliens* for the Disney movie "Alladin" and *global warming* for the ancient Roman film "Gladiator."
- **Redundant tags:** 26% of the the high preference / low quality tags are redundant — they provide information already displayed by MovieLens. In previous surveys, some users noted that they dislike tags that contain information already displayed in MovieLens, such as actors, genres, and directors. Users may rate *Quentin Tarantino* thumbs down because it is redundant, but may provide a high preference rating since they like his movies.
- **Subjective tags:** 16% of the high preference / low quality responses are subjective tags. Users may rate the subjective tag *funny* as low quality if they don't find a particular movie funny, but they may give the tag a high preference rating if they generally like comedies.

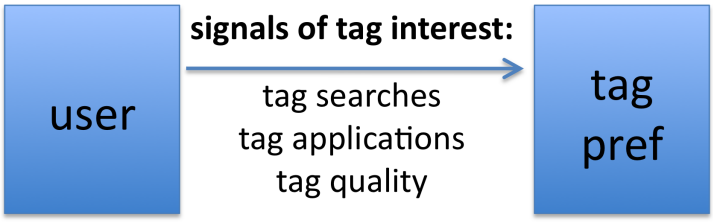


Figure 7.1: Inferring a user's preference for a tag based on her direct interactions with a tag such as her searches for a tag and her applications of a tag.

- **Personal tags:** 4% of the high preference / low quality responses are personal tags. Users may dislike others' personal tags, but may like movies they have *seen more than once*.

In total 59% of the low quality / high preference responses may result from tag quality responses being movie specific, while preference responses apply generally to a tag.

In summary, we find a high correlation betwen tag quality and tag preference. Surprisingly, we found that users claimed that they liked concepts related to many low quality tags. We found that this inconsistancy was primarly due to a descrepancy in the scopes of quality and preference. A user might disagree with the application of a particular tag to a movie (quality), but they may generally like movies with the tag (preference).

Based on the correlation between tag quality and tag preference, we include the output of our top performing tag selection algorithm as a feature for predicting a user's preference for tags.

7.4 Inferring Tag Preference

In this section we address RQ2,

RQ2: Can systems infer a user's preferences for tags?

We consider two approaches to inferring tag preference. First, algorithms can directly infer a user's preference for a tag based on her direct interactions with the tag (Figure 7.1). For example, if Alice searches for *animation*, she is probably interested in

it. Second, an algorithm may indirectly infer a user's preference for a tag based on her interactions with items having the tag (Figure 7.2). For example, Alice has assigned five-star ratings to three movies tagged with *animation*: "Shrek", "Pinnocchio", and "Toy Story". Based on these movie ratings, we may infer that she would enjoy other movies tagged with *animation*.

7.4.1 Inferring Preference using Tag Signals

We consider three algorithms based on direct signals of a user's interest in a tag (Figure 7.1). Users may be more interested in tags they themselves apply. **Tag-applied** infers higher preference for those tags a user has applied. Users may also be interested in the tags for which they have searched. **Tag-searched** infers higher preference for tags for which a user has searched. Both tag-applied and tag-searched use a simple 0 or 1 numeric coding.

We also use a third implicit tag signal: a tag's quality (**tag-quality**). As we mentioned in the introduction, a user's preference towards a tag may be correlated with the tag's quality. In order to examine this relationship, we include the output of the "all-implicit" tag selection algorithm as a feature. In order to make our results more generalizable to other sites, we do not draw on the tag quality thumb ratings unique to MovieLens.

All tag preference algorithms translate between a score (i.e. 0 or 1 for tag-applied) and a one-to-five star inferred tag preference according to a simple linear relationship. This relationship is estimated by performing a least-squares regression between the algorithm scores and a user's actual preference for tags as reported in the survey.

7.4.2 Weightings: Translating Item Signals to Tag Signals

In the previous section we proposed tag preference inference algorithms that draw on signals of a user's interest in tags. In the following section (7.4.3) we explore algorithms that infer tag preferences based on signals of a user's interest in items (Figure 7.2). In this section we explore item $\rightarrow$ tag *weightings* that numerically express a relationship between a tag and the item to which it is applied. These weightings are used by the tag preference inference algorithms in Section 7.4.3.

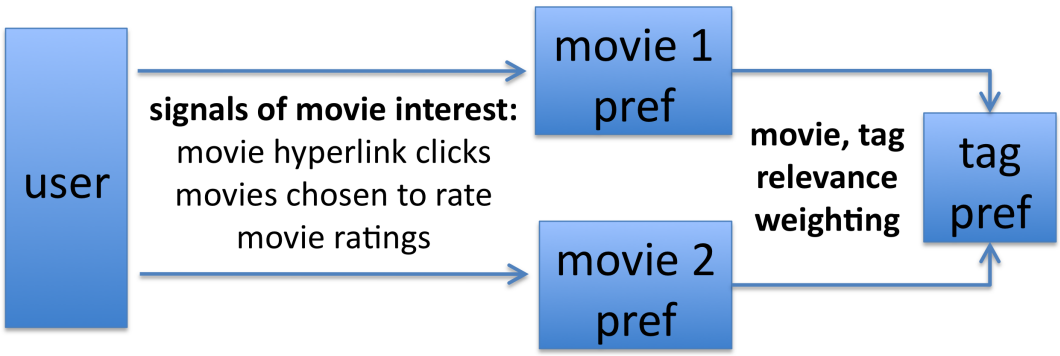


Figure 7.2: Inferring a user's preference for a tag indirectly based on her interactions with items having a tag such as her rating of items with the tag.

We begin with an example motivating tag-item weightings. Consider a simple tag preference inference algorithm based on a user's movie ratings: a user's preference for tags is their average rating for items with the tag. When calculating a particular user's preference for the tag *cars*, a system may rely on the user's ratings for two movies tagged with *cars*: "Gone in 60 Seconds," and "Cast Away." *Cars* accounts for 9 of the 38 tag applications for "Gone in 60 Seconds", but only 1 of the 36 tag applications for "Cast Away." Based on these counts, it seems plausible that the user's rating for "Gone in 60 Seconds" may reflect the user's preference for *cars* more strongly than "Cast Away."

We consider five algorithms for calculating a numeric weight describing the relationship between a tag and an item. The following terminology is used when presenting formulas for weightings: A is the set of all tag applications, M is the set of all movies, R is the set of all ratings, C is the set of all movie clicks. As shorthand, we define A_m to be the subset of tag applications for movie m, and A_t to be the subset of applications for tag t. We use similar notation for R, C, and M.

Constant weighting is a baseline that returns a constant value for all of a movie's tags.

Tagshare hypothesizes that user agreement is a signal of relevance. Tagshare is defined as the is the fraction of a movie's tag applications that are accounted for by a particular tag.[2] We smooth tagshare by assuming that 20 "fake" tags have been

[2] The term *tagshare* was first used by Tim Spalding from Librarything in a blog post on February 20, 2007: http://www.librarything.com/thingology/2007/02/when-tags-works-and-when-they-dont.php

applied to the movie. In practice this adds 20 to the denominator for all weightings, and ensures that a tag that has been applied 1 out of 2 times has lower tagshare than a tag that has been applied 50 out of 100 times. If $A_{t,m}$ is the set of applications of tag t to movie m, the formula for the tagshare is:

$$\text{tagshare}(m,t) = \frac{|A_{t,m}|}{|A_m|}.$$

TF/IDF (term frequency / inverse document frequency) hypothesizes that rare tags may be more relevant than common tags. TF/IDF is a popular heuristic weighting in the natural language processing community [Salton and Buckley, 1987]. In our application of TF/IDF to tagging systems, we treat tags as terms and movies as documents. TF/IDF favors tags that have been applied many times to a particular movie, but to few movies overall. The formula for the TF/IDF score of a particular tag t assigned to movie m is:

$$\text{tfidf}(m,t) = \frac{|A_{t,m}|}{|A_m|} \cdot log\left(\frac{|M|}{|M_t|}\right).$$

Note that TF/IDF simply multiplies tagshare by a weighting term that varies inversely with the number of movies to which a tag is applied.

Emission probability, a probabilistic version of tagshare, estimates the probability that a particular movie will "emit" a particular tag. We treat a movie's tags as a multinomial random variable with a dirichlet conjugate prior [Gelman et al., 2003] The prior is the probability of a particular tag t occurring across all movies:

$$\text{prior}(t) = \frac{|A_t|}{|A|}.$$

The movie-specific probability of a movie m emitting tag t is the fraction of the movie's tag applications that are for tag t smoothed to the prior:

$$\text{emission-probability}(m,t) = \frac{|A_{m,t}| + \alpha \cdot prior(t)}{|A_m| + \alpha}.$$

This smoothing is equivalent to a bayesian estimate of the movie-specific multinomial distribution given the overall prior. Since emission probability smooths to the background probability for a tag, no tag will have an emission probability of 0 (even if the tag is not applied to the item).

Probability informed, a probabilistic version of TF/IDF, assumes a generative model of tag creation. Figure 7.3 provides an overview of the model. The model assumes

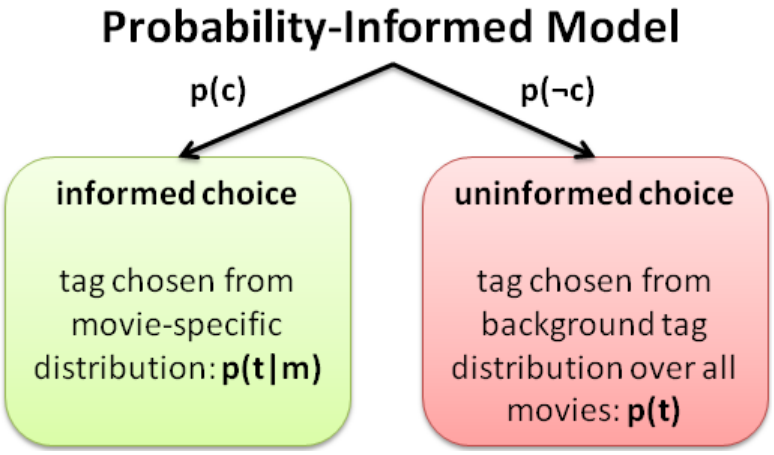


Figure 7.3: The probability-informed generative model of tag creation. Tags are either an informed choice that contain movie-specific information, or background noise. Each option has a prior probability.

that each tag was generated by either a movie-specific distribution for tags (*informed*), or the background distribution of all tags (*not-informed*).

The value output by prob-informed is the probability that a particular tag was generated by the movie-specific distribution. We use bayes rule to calculate this probability. In the following section, c (informed **c**hoice) indicates that probability that a tag was a result of movie-specific distribution and $\neg c$ indicates that it was not. We define the probability that a tag t applied to movie m was an informed choice as follows:

$$\begin{aligned} \text{prob-informed}(m,t) &= p(c|m,t) \\ &= \frac{p(t|c,m)\cdot p(c)}{p(t|m)} \\ &= \frac{p(t|m,c)\cdot p(c)}{p(t|\neg c,m)\cdot p(\neg c)+p(t|c,m)\cdot p(c)} \end{aligned}$$

We treat the background not-informed distribution as a multinomial variable based on the overall distribution of tags. Thus, $p(t|m,\neg c)$ is the probability of randomly drawing tag t from among all tags applied, which we named prior(t) in emission probability. The movie-specific probability $p(t|m,c)$ is simply the emission probability for t and m.

We experimented with different values for the priors $p(c)$ and $p(\neg c)$. Since $p(c)+p(\neg c)=1.0$, only one of the two parameters is free. We found little difference among a wide range of parameter choices. Indeed, different choices for prior parameters did

not change the rankings for a movie's tags - just the absolute values of the weights associated with those tags. We selected 0.5 for $p(c)$ and 0.5 for $p(\neg c)$ based on our estimate of the amount of low-value tags in MovieLens.[3]

As an example of the weighting, consider the tag *johnny cash* applied to the movie "Walk the Line" by 10 different people. The tag *johnny cash* is a fairly rare tag overall; therefore, the probability the 10 tag applications were based on the background distribution over all tags ($9.54 * 10^{-5}$) is much lower than the the probability it was generated by the movie-specific distribution ($1.09 * 10^{-3}$). After including prior information using equation 7.1, the probability *johnny cash* was generated by the informed movie-specific distribution is 0.96.

7.4.3 Inferring Preference using Item Signals

Now that we have investigated weightings for translating signals of item interest into signals of tag interest, we explore algorithms that infer a user's preferences for tags based on item signals (Figure 7.2).

We explore six different algorithms for calculating a user's preference for a tag based on her interactions with items having the tag. The six algorithms can be grouped according to the type of signal of movie interest that they use. The first two algorithms (movie-clicks, movie-log-odds-clicks) use clicks on movie hyperlinks as a signal of a user's interest in a movie. The third and fourth algorithms (movie-r-clicks, movie-r-log-odds-clicks), analyze the specific movies a user chooses to rate. The last two algorithms (movie-ratings, movie-bayes) draw on a user's numeric ratings for movies.

When presenting formulas for the algorithms we extend the terminology from the previous section. $w_{m,t}$ represents the weighting between a movie and tag as calculated by one of the five weighting functions.

Movie-clicks: The movie clicks algorithm is based on the hypothesis that users clicks on movies with tags they like more often. This algorithm estimates a user's preference for a tag based on the fraction of clicked movies that have the tag. Instead of weighting each movie equally, we weight movies using the five weightings we described

[3] For each tag application, we checked whether the tag had more thumbs up or thumbs down ratings. Approximately 50% had more thumbs up ratings.

in section 7.4.1. If movies(C_u) is the set of movies clicked by user u, then:

$$\text{movie-clicks}(u,t) = \frac{\sum_{m \in \text{movies}(C_u)} w_{m,t}}{|\text{movies}(C_u)|}.$$

The bottom denominator is the number of movies clicked because the sum of the weights for each movie is normalized to 1.

Movie-log-odds-clicks: Similar to movie-clicks, movie-log-odds-clicks assumes that users click movies with tags they like more often, but it adjusts for overall tag popularity. Movie-log-odds-clicks uses the log odds metric to compare the movie-specific tag frequency to the overall tag frequency.

$$\begin{aligned} \text{logit}(p) = &\quad log\left(\frac{p}{1-p}\right) \\ \text{movie-log-odds-clicks}(u,t) = &\quad \text{logit}(\text{movie-clicks}(u,t)) - \text{logit}(\frac{\sum_{m \in M_t} w_{m,t}}{|M|}) \end{aligned}$$

Movie-r-clicks: Users' movie viewing decisions may correlate with their tag preferences. For instance, a user may choose to watch a movie because it contains *violence*. We assume that users have watched the movies they have rated. Based on this, we consider a version of the movie-clicks algorithm that substitutes the movies a user has rated for the movies they have clicked.

Movie-r-log-odds-clicks: Similar to movie-r-clicks, movie-r-log-odds-clicks uses the movie-log-odds-clicks algorithm, but substitutes the movies a user has rated for the movies they have clicked.

Movie-ratings: Perhaps users rate movies with a particular tag consistently. For example, Alice consistently rated three *animated* movies five stars. Movie-ratings draws on this signal by predicting that a user's preference for a tag is the user's average rating for movies with the tag. As with the previous inference algorithms, we consider four different weightings. If $r_{u,m}$ is user u's rating for movie m:

$$\text{movie-ratings}(u,t) = \frac{\sum_{m \in M_t} w_{m,t} \cdot r_{u,m}}{\sum_{m \in M_t} w_{m,t}}.$$

The sums in both the numerator and denominator ignore movies the user has not rated.

Movie-bayes: Movie-bayes is a bayesian generative model for how users rate movies with a particular tag. Figure 7.4 describes the model. For every user u and tag t we select

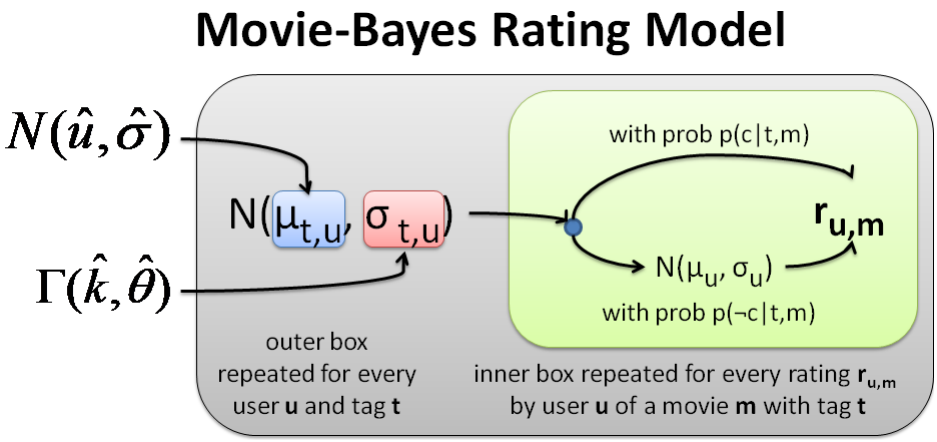


Figure 7.4: Movie-bayes is a generative model for how users rate movies with a particular tag. For every user u and tag t we select a distribution $N(\mu_{\mathrm{t,u}}, \sigma_{\mathrm{t,u}})$ specific to a user, tag pair from a single hyper-distribution. For each rating r by u for movie m with tag t, the tag may be the result of an informed or uninformed choice. If t is the result of an informed choice, the rating is chosen from the user-tag-specific distribution. If t was not the result of an informed choice, the rating is chosen from the user's background ratings distribution $N(\mu_u, \sigma_u)$. We estimate the hyperparameters for hyperdistributions $N(\hat{\mu}, \hat{\sigma})$ and $\Gamma(\hat{k}, \hat{\theta})$ using the empirical bayes methodology. We calculate the expected parameters for a particular user and tag, $\mu_{\mathrm{t,u}}$ and $\sigma_{\mathrm{t,u}}$, using numerical approximation.

a distribution $N(\mu_{\text{t,u}}, \sigma_{\text{t,u}})$ specific to a user, tag pair from a single hyper-distribution. For each rating r by user u for movie m with tag t, the tag may be the result of an informed or uninformed choice. If t is the result of an informed choice, the rating is chosen from the user-tag-specific distribution. If t was not the result of an informed choice, the rating is chosen from the user's background ratings distribution $N(\mu_u, \sigma_u)$.

We adopt the bayesian paradigm of considering all possible user-tag-specific normal distributions. For each distribution, we calculate the probability that that distribution generated the user's ratings. We then take the expectation of the mean for a particular tag and user ($\mu_{t,u}$) over all possible distributions by applying bayes rule based on the user's ratings. In the following formulas $R_{u,t}$ is the set of ratings by user u for movies tagged with t, and $E(X)$ denotes the expectation of random variable X. $\Gamma(\hat{k}, \hat{\theta})$ and $N(\hat{\mu}, \hat{\sigma})$ specify the gamma and normal hyperdistributions for the standard deviation and mean respectively.

$$
\begin{aligned}
&\text{movie-bayes}(u,t) = E(\mu_{t,u}|R_{u,t}) \\
&= \int_{\mu}\int_{\sigma} \mu \cdot p\Big(\mu, \sigma | R_{u,t}, N(\hat{\mu}, \hat{\sigma}), \Gamma(\hat{k}, \hat{\theta})\Big) \\
&= \int_{\mu}\int_{\sigma} \mu \cdot p(R_{u,t}|\mu, \sigma) \cdot p\Big(\mu, \sigma | N(\hat{\mu}, \hat{\sigma}), \Gamma(\hat{k}, \hat{\theta})\Big) \\
&= \int_{\mu}\int_{\sigma} \mu \cdot p(R_{u,t}|\mu, \sigma) \cdot p\Big(\mu | N(\hat{\mu}, \hat{\sigma})\Big) \cdot p\Big(\sigma | \Gamma(\hat{k}, \hat{\theta})\Big) \qquad (7.1)
\end{aligned}
$$

In equation 7.1 the second term, $p(R_{u,t}|\mu, \sigma)$, is the probability of the user's ratings for movies with a tag based on a user-tag-specific ratings distribution. To calculate this probability, we treat the ratings as independent events. As described earlier, each rating may be the result of the user's background ratings distribution, or a rating may be the result of the user's tag specific distribution. The user's background distribution, $N(\mu_u, \sigma_u)$, is fit to all of the user's ratings. The user-tag-specific distribution is chosen according to prob-informed(m, t). If $pi(m, t)$ is prob-informed(m, t), then:

$$
\begin{aligned}
p(R_{u,t,}|\mu, \sigma) &= \prod_{r \in R_{u,t}} p(r|\mu, \sigma) \\
&= \prod_{r \in R_{u,t}} \Big[p(r|N(\mu, \sigma))pi(m,t) + p(r|N(\mu_u, \sigma_u))(1 - pi(m,t)) \Big]
\end{aligned}
$$

The third term in equation 7.1, $p(\mu|N(\hat{\mu}, \hat{\sigma}))$, is the prior probability of a mean for the user-tag-specific normal distribution. We assume the user-tag-specific mean is drawn

from a normal distribution with hyper-parameters chosen using the empirical bayes methodology [Gelman et al., 2003]:

$$p(\mu_{|}N(\hat{\mu},\hat{\sigma})) = p(\mu|N(2.885,1.0)).$$

The fourth term in equation 7.1, $p\Big(\sigma|\Gamma(\hat{k},\hat{\theta})\Big)$, is the prior probability a of a standard deviation for the user-tag-specific normal distribution. We assume the deviation is drawn from a gamma distribution with hyper-parameters chosen using the empirical bayes methodology:

$$p(\sigma|\Gamma(\hat{k},\hat{\theta})) = p(\sigma|\Gamma(2.0,1.0)).$$

The choice of a gamma and normal distributions as hyper-distributions for a normal distribution is common in the Bayesian literature [Gelman et al., 2003]. We evaluate the complete integral using numerical approximation [Press et al., 1986].

7.4.4 Tag Preference Inference Results

As an evaluation metric, we calculate the Pearson correlation between each one to five star survey preference response and the inferred value from a particular algorithm (e.g. log-odds-clicks).

We begin by comparing the five different weighting algorithms we presented in section 7.4.1. Weightings did not perform consistently across algorithms. Tagshare and prob-informed performed best for four of six algorithms (movie-ratings, movie-bayes, movie-clicks, movie-r-clicks). All weightings performed identically for the movie-log-odds-clicks algorithm. The emit weighting performed erratically. It performed significantly worse ($p \leq 0.05$) than other weightings for movie-bayes, movie-clicks, and movie-r-clicks, but significantly better than other weightings for movie-log-odds-r-clicks. We suspect that this erratic behavior is due to the large variance in weightings that it outputs. In the following sections we use the best performing weighting for each algorithm.

Figure 7.5 shows pearson correlations for all tag preference inference algorithms. 95% confidence intervals are displayed on the graph. All pairwise differences are significant. The two algorithms based on explicit item rating signals (movie-ratings, movie-bayes) outperformed all implicit measures. Both the click-based and ratings-based movie-log-odds-clicks algorithms performed poorly. The two tag-based algorithms (tag-searched

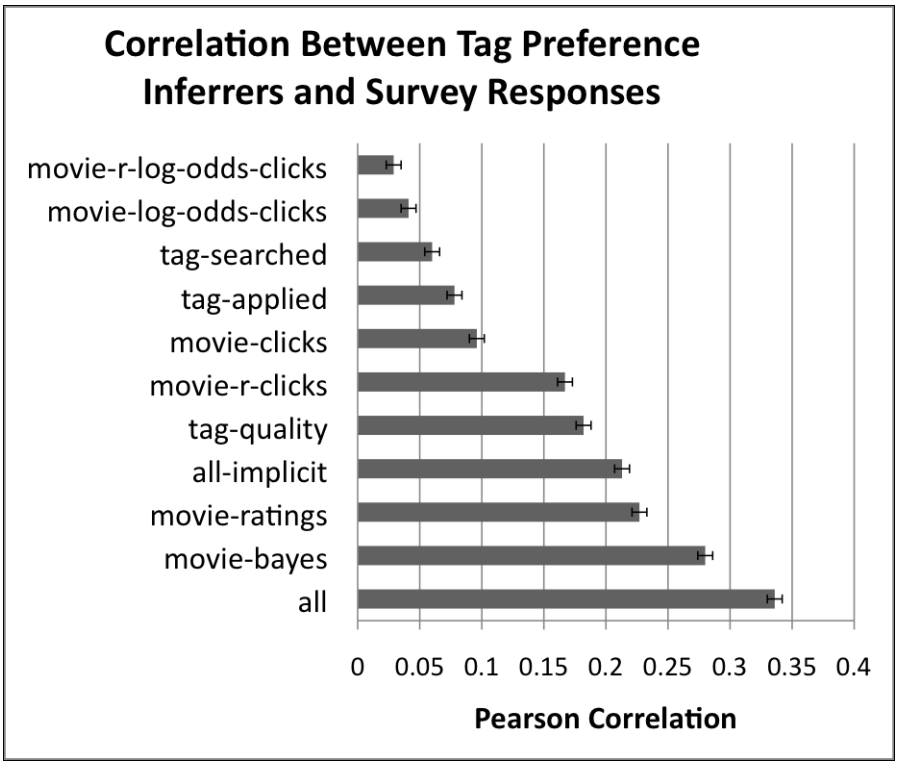


Figure 7.5: Pearson correlation between inferred tag preference, and actual tag preference. All pairwise differences are significant at the 0.05 level according to a two-tailed t-test. Algorithms based on explicit item signals performed best, followed by those based on implicit item signals, followed by those based on tag signals. *All*, a linear combination of all algorithms, performed best.

and tag-applied) did not perform as well as the movie-clicks algorithm. We suspect that this is due to the relatively small amount of tagging activity on MovieLens. For example, only 2.8% of the survey rating responses were associated with a tag the user had applied. Similarly, only 0.6% of the survey responses where associated with a tag the user had searched for. Tag-quality performed best among the algorithms using tag signals. Tag quality and tag preference are clearly correlated.

We also evaluated linear combinations of tag preference algorithms. We combined algorithms using a least square regression between tag preference responses and algorithm outputs.[4] **All-implicit** is the best combination of all tag preference inference algorithms that do not use movie ratings. **All** is the best combination of all tag preference algorithms. "All-implicit" outperformed each individual implicit feature. "All" outperformed all other algorithms.

7.5 Summary

In this chapter we explored tag preference, a user's general liking for items with a particular tag. We found tag preference to be strongly correlated with tag quality. We also explored tag preference inference algorithms that infer a user's liking for a tag. We found that the best tag preference inference algorithms relying on movie ratings outperformed the best algorithms that did not.

Our results can directly inform the design of tagging communities. Sites such as LibraryThing display interest profiles based on the tags a user has applied. Tags applied more often by a user are displayed more prominently (Figure 7.6). In this chapter we developed algorithms that infer a user's interest in tags. Our algorithms perform significantly better than the simple algorithms used by sites such as LibraryThing.

We use our tag preference inference algorithms as a building block in the next chapter, where we infer a user's preference for items based on their preference for tags.

Finally, we believe there are a number of fundamental issues surrounding the relationship between preference and quality. It may be challenging to design interfaces that collect ratings along both the quality and preference dimension without confusing users. Perhaps because of this, most systems such as YouTube and Netflix only collect ratings

[4] We tried using support vector machines as well, but they yielded no significant improvement

Figure 7.6: A user's tag profile on LibraryThing.

along the preference dimension. Future research might explore interfaces for differentiating between quality and preference and examine the role quality and preference play in different domains.

Chapter 8

Tagommenders

8.1 Introduction

Recommender systems enable users to navigate vast collections of items. Amazon suggests products users may like based on their ratings, clicked items, and purchased items [Linden et al., 2003]. Users of Digg receive news articles based on other articles they find interesting [Rose, 2008]. Members of Netflix receive movie recommendations based on their movie ratings [Bennett and Lanning, 2007]. In each of these scenarios, recommender systems choose a few items a user will like most from among thousands, or even millions, of possibilities. This task, which we call *recommend*, is one of two tasks supported by nearly all recommender systems [Schafer et al., 2007]. For the second common task, *predict*, a recommender system predict s which rating a user will assign to a particular item. For example, a user of "Rate Your Music" [1] might receive a predicted rating of 4.2 out of 5 stars for the album "White Blood Cells" by the White Stripes based on a five star rating for "In Rainbows" by Radiohead. For both the recommend and predict tasks, recommender systems help users understand an unknown relationship between themselves and an item by comparing a user's behavior (e.g. album clicks and ratings) to patterns of behavior in other users.

Tagging systems offer users an alternate way to address the recommend and predict tasks. Shirky suggests that since tags are created by users, they represent concepts meaningful to them [Shirky, 2005]. Because tags are easily comprehended by users, tags

[1] http://www.rateyourmusic.com

serve as a bridge enabling users to better understand an unknown relationship between an item and themselves. In previous work, we validated this relationship by finding that certain types of tags help users to find and make decisions about items [Sen et al., 2006]. For example, Alice[2] is a real user in the MovieLens movie recommendation community we study. She enjoys animated movies, and has assigned five star ratings to "Shrek," "Pinnochio," and "Toy Story". If Alice visits the web page for the movie "Ratatouille" she would see that 5 users have applied the tag *animated* to it. Based on these tags, she might decide she would enjoy the movie (the predict task). Alice might then click on the tag *pixar* to discover the related movie "The Incredibles" (the recommend task).

Recommender algorithms that incorporate tagging information promise to combine the best elements of both types of systems. Lamere refers to these tag-based recommendations as "tagomendations" [Lamere, 2007]. We similary refer to tag-based recommender systems as *tagommenders*. Tagommenders offer the automation of traditional recommender systems, but retain the flexibility of tagging systems. Schafer et al. found that users enjoy specifying feedback about items along a variety of dimensions [Schafer et al., 2002]. Tagommenders enable recommenders to use the dimensions of items that users consider most important.

In this chapter, we design tagommenders inspired by the way in which humans use tags to evaluate items. In Chapter 7 we inferred users' preferences for tags based on their interactions with tags and movies. In this chapter, we use a user's inferred preference for tags to predict her preference for items.

We separate our algorithms by the type of signals they use: *implicit* only, or both implicit and *explicit*. Implicit signals such as clicks and searches occur during users' natural interactions with tags and items. Tagommenders for sites that do not support item ratings, such as the online bookmarking site Delicious[3] , must generate recommendations based on these implicit signals. Other systems with tags, such as Amazon, support explicit signals of interest in the form of item ratings. Our research questions explore the performance of tagommenders in both types of systems.

RQ1: How well do tagommenders perform in systems without ratings?

[2] Alice is a pseudonym to protect the user's privacy
[3] http://del.icio.us

RQ2: How well do tagommenders perform in systems with ratings?

We evaluate RQ2 and RQ2 using movie ratings and tags created by MovieLens users. We believe this work to be important for two reasons. First, we hope that sites with an abundance of tagging activity, such as Delicious or flickr[4], can use our algorithms to improve item recommendations. Second, we believe that tagommenders offer an equally accurate, but more flexible alternative to traditional recommender systems.

8.2 Tagommender Algorithms

In the previous chapter we evaluated methods for inferring users' preferences for tags based on signals of interest in tags and items (Figure 8.1, upper left). We now shift our focus to using those inferred tag preferences to predict ratings for movies (Figure 8.1, upper right). We present five tag-based recommendation algorithms — two based on implicit data, and three based on explicit data. We then describe our methodology including our evaluation metrics and baseline recommender algorithms. Finally, we present results for all algorithms as they relate to our research questions.

8.2.1 Implicit Tag-Based Algorithms

We first consider two tag-based recommendation algorithms that use implicit data in order to support sites without item ratings. As input, these algorithms use tag preferences inferred by all-implicit, the top performing implicit tag preference algorithm. As output, these algorithms produce a score suitable for ranking items in a recommendation list. Recommender systems that do not collect ratings generally do not predict ratings; therefore, the values output by the implicit tag recommendation algorithms are suitable for the recommend task but not the predict task.

In the previous chapter we saw that the average tag preference differed by tag. For example, users generally preferred a high quality tag to a low quality one. During our analyses of tag-based recommenders, we found that these per-tag differences skewed results. We accounted for these per-tag differences by normalizing each tag's inferred preference to have mean 0 and standard deviation 1. In addition, we found that more

[4] http://www.flickr.com

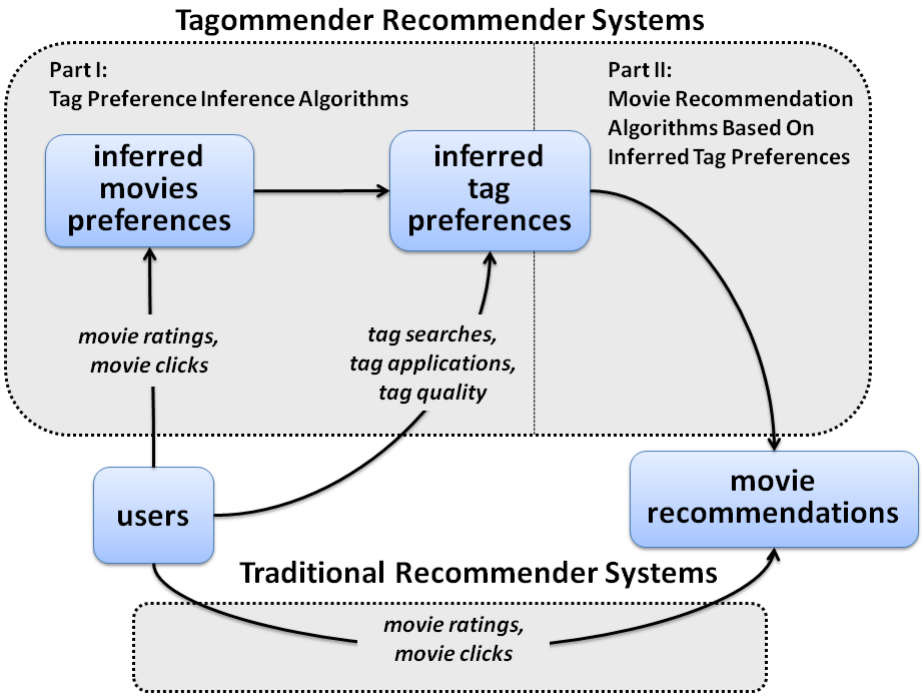


Figure 8.1: Our model of movie tagommenders. Traditional recommender systems (bottom) generate predictions for movies based on movie ratings or clicks. Tagommenders (top) first infer users' preferences for tags (upper left). Based on these inferred preferences for tags, tagommenders generate movie recommendations (upper right). Users' preference for tags can be inferred based on signals of interest in tags (tag applications, searches), or signals of interest in items (movie ratings, clicks). In order to use item signals to infer users' preferences for tags, they must be translated to tag signals (upper left).

active users generally had higher tag preferences than less active ones. To neutralize this effect, we subtracted the average tag preference for each user. We use these normalized tag preference values throughout this section.

We now describe the two implicit tag-based recommender algorithms:

Implicit-tag: The implicit-tag algorithm is inspired by algorithms from information retrieval that calculate the similarity between a user's profile vector and a document's term vector [Raghavan and Wong, 1986]. In information retrieval, the columns in each vector correspond to words. In the implicit-tag algorithm, the columns correspond to tags. To generate a prediction for a movie m, implicit-tag calculates the dot product between users' preferences for movie m's tags and the weighting $w(t, m)$ between tag t and movie m. We use prob-informed as a weighting based on its strong performance in section 4.4. If $ntp(u, t)$ is user u's normalized inferred tag preference for tag t, then user u's predicted score for movie m is:

$$\text{implicit-tag}(u, m) = \sum_{t \in T_m} ntp(u, t) \cdot w(m, t).$$

Implicit-tag-pop: Implicit-tag ignores the overall popularity of a particular movie, an important signal of a users's liking for a movie. We next consider implicit-tag-pop, a version of the algorithm that adds $\text{pop}(m)$, a term estimating a movie's popularity:

$$\text{implicit-tag-pop}(u, m) = \text{implicit-tag}(u, m) + \text{pop}(m).^{5}$$

We experimented with a variety of signals of popularity for a movie based on the number of clicks, tags, clickers, and taggers for a movie. For each possible signal, we fit a function between the signal value for each movie (e.g. num clicks) and the average rating for the movie. We evaluated each signal of popularity based on how well implicit-tag-pop performed using the signal. We found that tags outperformed clicks, counting users (clickers) outperformed counting events (clicks), and log transforming signals improved results. The best overall estimate was a linear estimate based on the log of the number of users who tagged a movie ($R^2 = 0.97$):

$$\text{pop}(m) = 0.31 \cdot \log(|\text{users}(A_m)|) + 3.16.^{6}$$

[5] We experimented with weightings for $\text{pop}(m)$, but found that a weight of 1.0 performed best.

[6] Recall that we use the implicit tag-based algorithms for recommendation but not for prediction. Although we report the the intercept value (3.16), the choice of intercept does not affect the relative ordering. Therefore, the intercept is unnecessary - we could simply use 0.0.

8.2.2 Explicit Tag-Based Algorithms

The final three tag-based algorithms are intended for sites with item ratings; as a result, they rely on both implicit and explicit data. As input these algorithms use tag preferences inferred by *all*, the top performing tag preference algorithm using both implicit and explicit signals. Since these algorithms output a value between 1.0 and 5.0 corresponding to a star rating for a movie, they support both the predict and recommend tasks. We choose three algorithms that model increasingly complex relationships between tag preferences and movie ratings.

Cosine-tag: The success of the traditional item-based rating models that use cosine similarities inspired us to create a similar model based on tags. Cosine-tag predicts that a user's rating for a movie is a weighted average of the user's preferences for the movie's tags. Cosine-tag weights a particular tag according to the adjusted cosine similarity between ratings for a movie and inferred preferences for a tag. We refer to user u's mean movie rating as $\bar{r_u}$, and U_m is the collection of users who rated movie m. The adjusted cosine similarity between movie m and tag t is:

$$\text{sim}(m,t) = \frac{\sum_{u \in U_m} (r_{m,u} - \bar{r_u}) \cdot \text{ntp}(t,u)}{\sqrt{\sum_{u \in U_m} (r_{m,u} - \bar{r_u})^2} \sqrt{\sum_{u \in U_m} \text{ntp}(u)^2}}.$$

Note that $ntp(t,u)$ is already average adjusted, so there is no additional adjusting performed. Given this definition of similarity, cosine-tag constructs a prediction for a movie as the average of the user's preferences for its tags, weighted by the tags similarities to the movie. If T_m is the collection of all tags applied to movie m:

$$\text{cosine-tag}(u,m) = \frac{\sum_{t \in T_m} \text{sim}(m,t) \cdot \text{ntp}(t,u)}{\sum_{t \in T_m} \text{sim}(m,t)} + \bar{r_u}.$$

One choice in this algorithms is the tags over which the average should be calculated. We found that the algorithm performed best when averaging over the 10 most similar tags.

Linear-tag: Since cosine-tag predicts a weighted average of a user's inferred tag preferences, the variability of movie predictions it outputs depend on the variability of

its inputs (inferred tag preferences). Linear-tag models a more complex relationship between an inferred tag preference and a predicted movie rating. For each tag t applied to movie m, linear tag estimates a least-squares fit $y_{t,m}(u)$ between users' inferred tag preferences for t and their ratings for m:

$$\mathrm{y}_{t,m}(u) = \alpha_{t,m}\mathrm{ntp}(t,u) + \beta_{t,m} + \epsilon_{t,m}.$$

In the linear equation above, $\alpha_{t,m}$ is the coefficient between tag t and movie m, $\beta_{t,m}$ is the intercept, and $\epsilon_{t,m}$ is the residual error term.

Linear-tag generates user u's prediction for movie m by averaging the values predicted by each of the linear fits $y_{t,m}(u)$. We found that weighting by inverse residual improved performance because it gave greater importance to more accurate fits.

$$\text{linear-tag}(u,m) = \frac{\sum_{t \in tags(A_m)} \Big(y_{t,m}(u)/\epsilon_{t,m} \Big)}{\sum_{t \in tags(A_m)} \Big(1.0/\epsilon_{t,m} \Big)} + \bar{r_u}.$$

As with cosine-tag, we experimented with averaging over different sets of tags. Averaging over the 5 tags with smallest residual performed best.

Regress-tag: The linear-tag model treats each linear fit between a tag and a movie as independent. This may not be optimal. For example, both *animated* and *animation* have been applied to "Toy Story." It seems plausible that users' inferred preferences for these two tags would correlate. Algorithms aware of relationships between tags may perform better than those that do not.

Regress-tag constructs a linear equation for each movie m. The input variables are all users' inferred tag preferences for tags applied to m. The output is each user's rating for m. If m has tags $t_1, \ldots, t_n$ then:

$$\text{regress-tag}(u,m) = h_0 + h_1\mathrm{ntp}(u,t_1) + \ldots + h_n\mathrm{ntp}(u,t_n).$$

We experimented with three methods for choosing the coefficients h_i: simple least-squares multiple regression, regularized multiple regression, and regression support vector machines. We found that the least-squares and regularized multiple regressions overfit movies with few ratings. For example, several movies had the same number of tags and ratings applied to them. In this case, multiple regression can build an equation

that perfectly fits the input data. These fits often lead to large values h_i that seemed intuitively incorrect and performed poorly. SVMs performed best due to their robustness to overfitting. We used the libsvm library based on its java implementation and efficient performance for linear kernels [Chang and Lin, 2001]. We found that libsvm performed best when c, the tradeoff between the margin and error penalty, was set to 0.005.

8.3 Methodology

We compare the tag-based recommendation algorithms to three naive baselines:

Overall-avg: Overall-avg generates a prediction equal to the overall average rating (3.55):

$$\text{overall-avg}(m, u) = 3.55.$$

In the recommend task, movies are ordered by predicted rating. Since overall-avg returns the same value for every movie, we randomly order recommendation lists.

User-avg: User-avg predicts a user's average for all of his or her movies:

$$\text{user-avg}(m, u) = \frac{\sum_{r \in R_u} r_{m,u}}{|R_u|}.$$

As with overall-avg, given a particular user, user-avg returns the same value for every movie. Thus, we randomly order recommendation lists.

User-movie-avg: User-movie-avg begins by average adjusting all of a user's ratings. The prediction for a movie is the average of all users' adjusted ratings for the movie. If U_m is the collection of users who rated movie m, then:

$$\text{user-movie-avg}(m, u) = \frac{\sum_{u' \in U_m} \left(r_{m,u'} - \text{user-avg}(u') \right)}{|U_m|} + \text{user-avg}(u).$$

While these three naive baselines provide insight into algorithm performance, we ultimately compare our tag-based algorithms to top-performing traditional CF algorithms. We consider three traditional algorithms:

Explicit-item: We include the item-based algorithm introduced by Sarwar et al. based on its accuracy and popularity in real-world systems such as Amazon [Linden et al., 2003]. The explicit item-based model calculates similarities between the ratings for each pair

of movies. In order to predict for a particular movie m, the item model constructs a weighted average of the user's ratings for the movies most similar to m. The rating weights used for the weighted average are based on the similarities to m.

Implicit-item: We compare our implicit tag-based algorithms to Karypis et al.'s item-based algorithm for unary data (such as click and transaction data) [Karypis, 2001]. We selected this algorithm based on its accuracy and popularity. The item-based model calculates similarities between each pair of movies based on the number of times movies co-occur in user baskets. In order to predict for a particular movie m, the movie model sums the similarities between m and the movies in the user's basket that are most similar to m.

Funk-svd: We include Simon Funk's Singular Value Decomposition algorithm due to its strong performance in the Netflix competition [Funk, 2006]. The Funk SVD approximates the full users $\times$ movies rating matrix using a matrix of lower dimension, and uses regularization to manage the sparsity of the ratings matrix.

We used five-fold cross validation in our analyses. For each each of the five test / train splits, we hide 30% of user ratings in the test set, and evaluate the performance of an algorithm by comparing the ordering of a recommendation list to the hidden ratings. Herlocker et al. find two important classes of evaluation metrics: those that evaluate an algorithm's performance on the predict task, and those that focus on the recommend task [Herlocker et al., 2004]. We choose one evaluation metric from each class:

Top-5: As an evaluation metric for the recommendation task, we use top-5, the fraction of the top five recommended movies for a user that are rated four stars or higher by the user[7] . We only consider elements the user has rated when selecting the top 5 movies. The 95% confidence intervals for top-5 was $\pm 0.57\%$ ($n = 28,185$).

MAE: In addition to top-5 we report mean absolute error (MAE), the average absolute difference between the value predicted by a recommender system and the user's actual rating value. MAE reflects an algorithm's performance on the predict task. We examined the distribution of MAE values produced in our analyses and found the 95% confidence intervals for MAE was ± 0.001 ($\mu = 0.577, \sigma = 0.491, n = 516,441$). As we discussed in Section 5.1, the implicit algorithms support only the recommend task.

[7] We choose 4 stars as the cutoff for a "good" rating since it is the first star rating higher than the overall average rating of 3.55 stars. We experimented with other values for n in top-n: 1, 3, 5, 10, 20. We found the results to generally be consistent regardless of choice of n.

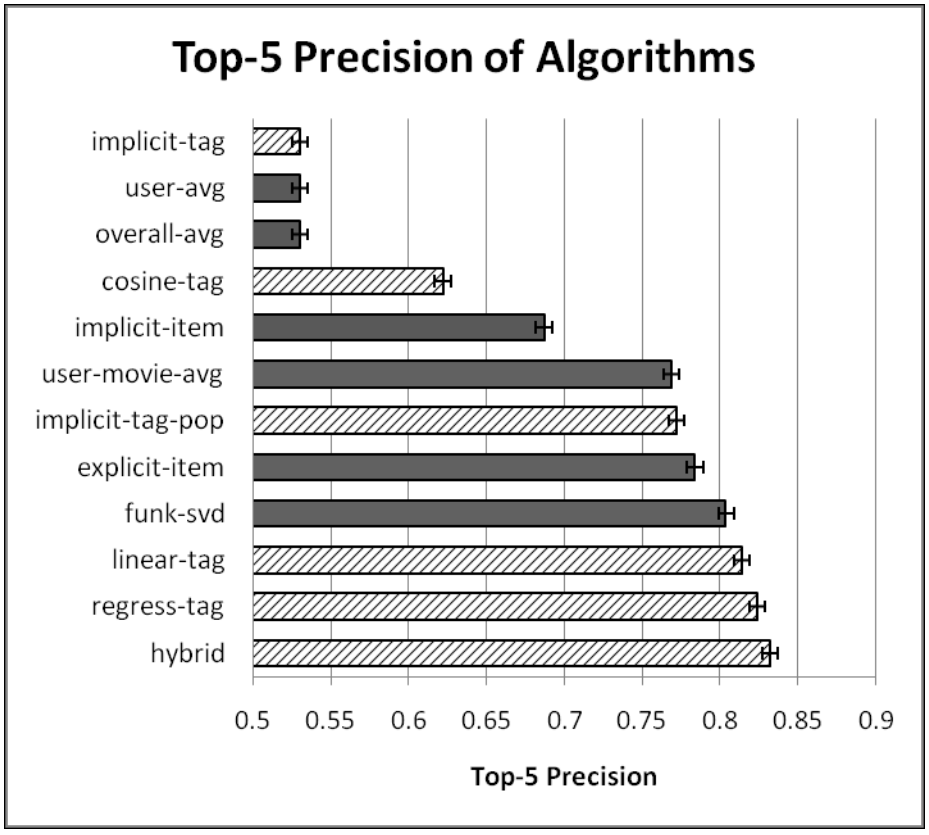


Figure 8.2: Top-5 precision for recommender algorithms. 95% confidence intervals are displayed for each algorithm. Higher top-5 values correspond to better performance. CF algorithms are displayed in solid bars and tag-based algorithms are displayed in striped bars. The best of the tag-based algorithms perform better than the best CF algorithms.

Therefore, we only report MAE for explicit algorithms.

8.4 Tagommenders Results and Discussion

Figure 8.2 shows the top-5 precision for the five tag-based algorithms, the three naive baselines, and the three collaborative filtering baselines. Higher top-5 values correspond to better performance. The traditional CF algorithms are displayed in solid bars and The tag-based algorithms are displayed in striped bars. We also include **hybrid**, a simple linear combination of the best performing tag-based algorithm (regress-tag) and

traditional algorithm (funk-svd) [8] . 95% confidence intervals are displayed for each algorithm.

Implicit-tag, user-tag, and overall-tag all achieve a top-5 of 53%, the same as randomly ordering a recommendation list. Differences between other pairs are significant ($p \leq 0.05$) except for those between user-movie-avg and implicit-tag-pop, and between regress-tag and hybrid. The tag-based algorithms generally perform well. Implicit-tag-pop, the best implicit algorithm, achieves a top-5 of 77%. Regress-tag, the best performing explicit algorithm achieves a top-5 of 82%.

Figure 8.3 shows the MAE for all explicit algorithms. The traditional CF algorithms are displayed in solid bars. Lower MAE values correspond to better performance. The tag-based algorithms are displayed in striped bars. All pairwise differences are significant ($p \leq 0.05$) except hybrid and funk-svd. As discussed in Section 8.2, the implicit algorithms only support the recommend task. Therefore, we only report MAE for explicit algorithms. In general, the tag-based algorithms outperformed the naive baselines, and the traditional CF algorithms outperformed the tag-based algorithms. Regress-tag performed best among the tag algorithms, achieving an mae of 0.584. As with top-5, cosine-tag performed poorly, achieving an mae of 0.639. Among the CF algorithms, funk-svd performs best, achieving an mae of 0.555.

Given these results, we return to our second research question:

RQ2: How well do tagommenders perform in systems without ratings?

As shown in Figure 8.2, implicit-tag-pop (77%) performs significantly better than the popular implicit-item algorithm (69%) according to top-5. We wondered if the strong performance of implicit-tag-pop compared to implicit-item was due to its inclusion of popularity. To test this possibility, we experimented with different methods for building popularity into the implicit-item algorithm. None of them significantly improved its performance. We conclude that the tagommender algorithm performs better than traditional CF algorithms in our evaluation without ratings.

Finally, we address our last research question:

RQ3: How well do tagommenders perform in systems with ratings?

[8] We experimented with all mixtures between 0 and 100 in increments of ten, and found that a 50-50 average performed best

Among the explicit tag algorithms, regress-tag performs best in both top-5 (82%) and MAE (0.584). Among the traditional CF algorithms, funk-svd performs best in both top-5 (80%) and MAE (0.555). Both these differences are significant ($p < 0.05$). We conclude that tagommenders appear to perform better than traditional CF algorithms for the recommend task, but worse for the predict task. However, the recommend seems to be more prevalent in real world systems. Among all popular recommender systems we investigated, only three (Netflix, MovieLens, Rate Your Music) offer predicted ratings. Most recommender systems now follow Amazon's model. Amazon does support the recommend task. However, instead of supporting the predict task through CF, they offer users rich product data, user reviews, and average user ratings. Thus, tagommenders perform better than traditional CF algorithms in the task most important to real world recommender systems.

We conclude by noting that hybrid, the linear combination of funk-svd and regress-tag, offers the best of both tag-based and CF algorithms. Hybrid achieves a top-5 of 83%, which is slightly (but not significantly) better than the top performing tag-based algorithm, and significantly better than traditional CF algorithms. In addition, hybrid's MAE equals that of funk-svd, the best performing traditional CF algorithm. Thus, a simple hybrid algorithm performs better than any CF algorithm on the recommend task and it matches the best CF algorithm on the predict task.

8.5 Conclusion

In this chapter we introduced and evaluated tagommenders, recommender algorithms that make use of tags. We constructed implicit and explicit tag-based recommendation algorithms based on users' inferred tag preferences. These tagommenders outperformed existing CF algorithms in the recommend task most critical to real world recommender systems. Finally, we showed that a hybrid tag and CF algorithm combines the strong predict performance of CF algorithms with the strong recommend performance of tag based algorithms.

We believe that tagommenders may lead to novel interfaces for recommender systems. Since tagommenders use tags as an intermediary entity, their recommendations can be explained based on users' preferences for tags. MovieLens users often ask for the

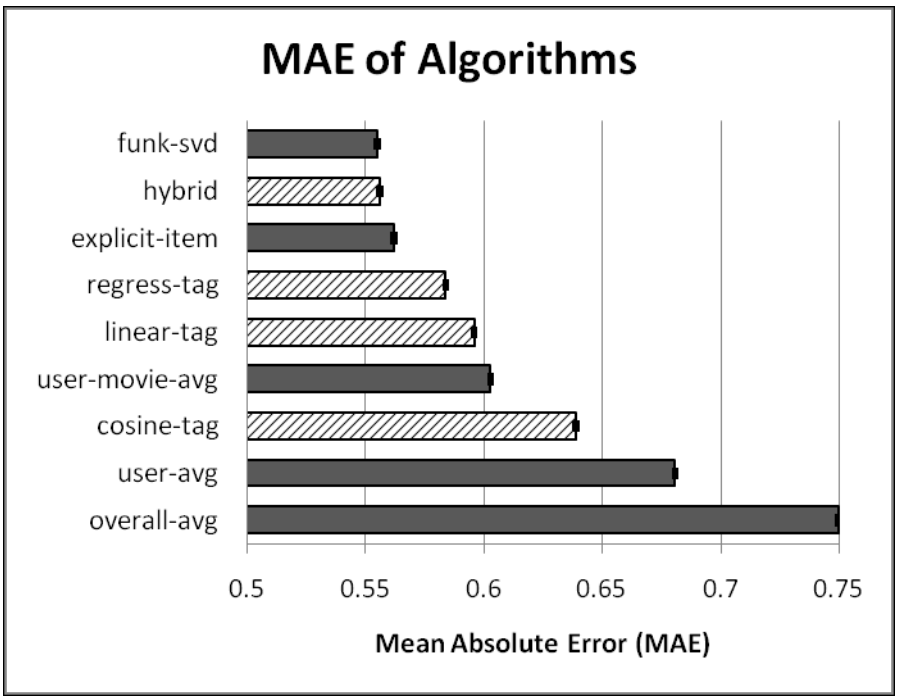


Figure 8.3: MAE for explicit algorithms. We do not include implicit algorithms since they do not support the predict task. 95% confidence intervals are displayed for each algorithm. Lower MAE values correspond to better performance. CF algorithms are displayed in solid bars and tag-based algorithms are displayed in striped bars. In general, tag-based algorithms perform better than naive baselines but worse than their CF counterparts.

Table 8.1: Top 10 inferred tags not applied to movies based on the similarity between tag preferences and movie ratings. We do not include movies in a series (e.g. trilogies), and only include the top entry for tags or movies that appear more than once.

movie	**tag**	**cosine sim**
Pearl Harbor (2001)	*disaster*	0.47
Runaway Bride (1999)	*girlie movie*	0.45
Beauty and the Beast (1991)	*talking animals*	0.42
Air Force One (1997)	*disaster*	0.41
Armageddon (1998)	*will smith*	0.41
Cinderella (1950)	*cartoon*	0.40
Inconvenient Truth (2006)	*documentary*	0.40
The Little Mermaid (1989)	*musical*	0.40
Gone in 60 Seconds (2000)	*exciting*	0.39
My Best Friend's Wedding (1997)	*chick flick*	0.39

opportunity to rate movies on a more diverse set of dimensions. Tagommenders might prove a way to meet that desire.

Relationships between ratings and tags may also be used to infer the tags that should be applied to a movie. Although previous researchers have investigated the tag inference problem [Jaschke et al., 2007], they have not made use of patterns in users' ratings of items. For each movie, the cosine tag recommender calculates the similarity between between users inferred preferences for each tag and their ratings for the movie. For example the most similar tags for the movie "Last of the Mohicans" starring Daniel Day-Lewis are the tags *tribal*, *sword fight*, *cavalry charge*, *historical*, and *stirring*. The only one of these tags that users actually applied to the movie is *tribal.* Figure 8.5 lists 10 <tag, movie> pairs with highest similarity scores.

One important question related to our findings is how tagommenders will perform in domains other than MovieLens. While we cannot be certain, the high tag density of a system such as Delicious might lead to more accurate recommendations.

Chapter 9

Conclusion

In the four years since we began our research on tagging systems, tags have transitioned from an internet novelty to a ubiquitous characteristic of Web 2.0 systems [O'Reilly, 2007]. In Chapter 4, we provided survey evidence supporting two reasons for this rise in popularity:

- Tagging systems offer users a flexible, fast mechanism for augmenting the information space of websites.
- Tagging systems enable users to quickly share information about items relevant to them.

We have shown that community influence significantly affects tag evolution – similar tagging systems can result in significantly different tag vocabularies. In an effort to control this volatility, we explored algorithms and interfaces that can detect high quality tags, and by doing so encourage positive tagging norms. Finally we explored tagommenders, tag-based recommenders that combine the flexibility of tagging systems with the automation of recommenders.

Of course, our work is not an exhaustive analysis of tagging systems. A vast array of open research directions remain.

- We conduct our research using MovieLens. Our results could be validated in other domains.

- Tagommenders generate recommendations based on a user's tag interest profile. Tagommender systems could enable users to view and change their tag interest profiles to create flexible and interactive recommenders.
- We primarily study tags as a method for capturing important characteristics of an item. Novel tagging systems could alternatively use tags as a way of describing relationships between entites.
- Researchers may be able to build tools that help the community to curate its vocabulary of tags.

Tagging systems face many challenges. Because of these challenges, many pundits argue that tags are a passing fad. For example, in a blog post David Brake writes

> Of course tagging is in its infancy and doubtless it will grow in popularity. But does this mean it will become mainstream? I have my doubts. And even if it does I suspect most content creation and tagging will continue to be done by a passionate (or geeky) few, like myself.[1]

Tagging *applications* may change. Social bookmarking sites may decline in popularity. Better image classification algorithms may render image tagging irrelevant. However, tags will continue to offer users a simple method for adapting the information space of their communities. For this reason, we believe tags will remain a fundamental software component. Our research has offered insight into the dynamics, challenges, and possibilities of tagging systems. We hope that these insights will, in a small way, inform tagging system designers and, in turn, improve tagging communities.

[1] http://groupblog.workasone.net/archives/00219.html

Chapter 10

References

[Arnt and Zilberstein, 2003] Arnt, A. and Zilberstein, S. (2003). Learning to perform moderation in online forums. *Web Intelligence, 2003. WI 2003. Proceedings. IEEE/WIC International Conference on*, pages 637–641.

[Asch, 1956] Asch, S. (1956). Studies of independence and conformity: A minority of one against a unanimous majority. *Psychological Monographs*, 70.

[Asch, 1951] Asch, S. E. (1951). Effects of group pressure upon the modification and distortion of judgements. *Groups, Leadership, and Men*, pages 177–190.

[Begelman et al., 2006] Begelman, G., Keller, P., and Smadja, F. (2006). Automated tag clustering: Improving search and exploration in the tag space. In *Collaborative Web Tagging Workshop at WWW2006, Edinburgh, Scotland*.

[Bennett and Lanning, 2007] Bennett, J. and Lanning, S. (2007). The Netflix Prize. In *Proceedings of KDD Cup and Workshop*.

[Besag and Green, 1993] Besag, J. and Green, P. J. (1993). Spatial statistics and Bayesian computation. *Journal of the Royal Statistical Society. Series B (Methodological)*, 55(1):25–37.

[Carlin and Louis, 1997] Carlin, B. P. and Louis, T. A. (1997). Bayes and empirical Bayes methods for data analysis. *Statistics and Computing*, 7(2):153–154.

[Cattuto et al., 2006] Cattuto, C., Loreto, V., and Pietronero, L. (2006). Semiotic dynamics in online social communities. In *The European Physical Journal C*. Springer-Verlag.

[Chang and Lin, 2001] Chang, C.-C. and Lin, C.-J. (2001). *LIBSVM: a library for support vector machines*. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[Cialdini, 2001] Cialdini, R. B. (2001). *Influence Science and Practice*. Allyn and Bacon, MA, USA.

[Cosley, 2006] Cosley, D. (2006). *Helping Hands: Design for Member-Maintained Online Communities*. PhD thesis, University of Minnesota.

[Cosley et al., 2005] Cosley, D., Frankowski, D., Kiesler, S., Terveen, L., and Riedl, J. (2005). How oversight improves member-maintained communities. In *Proceedings of CHI 2005*, pages 11–20, New York, NY. ACM Press.

[Cosley et al., 2006] Cosley, D., Frankowski, D., Terveen, L., and Riedl, J. (2006). Using intelligent task routing and contribution review to help communities build artifacts of lasting value. In *Proceedings of ACM CHI*, Montreal, CA.

[Cosley et al., 2003] Cosley, D., Lam, S. K., Albert, I., Konstan, J., and Riedl, J. (2003). Is seeing believing? How recommender system interfaces affect users' opinions. In *CHI*.

[Coye, 1994] Coye, D. (1994). A linguistic survey of college freshmen: Keeping up with standard American English. *American Speech*, 69(3):260–284.

[Dahlen et al., 1998] Dahlen, B., Konstan, J., Herlocker, J., Good, N., Borchers, A., and Riedl, J. (1998). Jump-starting MovieLens: user benefits of starting a collaborative filtering system with "dead data". *University of Minnesota TR*, pages 98–017.

[Dubinko et al., 2007] Dubinko, M., Kumar, R., Magnani, J., Novak, J., Raghavan, P., and Tomkins, A. (2007). Visualizing tags over time. *ACM Trans. Web*, 1(2):7.

[Funk, 2006] Funk, S. (2006). Netflix update: Try this at home. *sifter. org/ simon/journal/20061211. html*.

[Gelman et al., 2003] Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2003). *Bayesian data analysis, Second Edition*. Chapman & Hall/CRC.

[Golder and Huberman, 2006] Golder, S. and Huberman, B. A. (2006). The structure of collaborative tagging systems. *Journal of Information Science*.

[Guy and Tonkin, 2006] Guy, M. and Tonkin, E. (2006). Tidying up tags? *D-Lib Magazine*, 12(1):1082–9873.

[Halvey and Keane, 2007] Halvey, M. and Keane, M. (2007). An assessment of tag presentation techniques. In *Proceedings of the 16th international conference on World Wide Web*, pages 1313–1314. ACM Press New York, NY, USA.

[Hammond et al., 2005] Hammond, T., Hannay, T., Lund, B., and Scott, J. (2005). Social bookmarking tools: A general review. *D-Lib Magazine*, 11(4).

[Hassan-Montero and Herrero-Solana, 2006] Hassan-Montero, Y. and Herrero-Solana, V. (2006). Improving tag-clouds as visual information retrieval interfaces. In *International Conference on Multidisciplinary Information Sciences and Technologies*.

[Herlocker et al., 2004] Herlocker, J., Konstan, J., Terveen, L., and Riedl, J. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1):5–53.

[Jaschke et al., 2007] Jaschke, R., Marinho, L., Hotho, A., Schmidt-Thieme, L., and Stumme, G. (2007). Tag recommendations in folksonomies. *Lecture Notes in Computer Science*, 4702:506.

[Joachims, 2006] Joachims, T. (2006). Training linear SVMs in linear time. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 217–226, New York, NY, USA. ACM.

[Karypis, 2001] Karypis, G. (2001). Evaluation of item-based top-n recommendation algorithms. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 247–254. ACM New York, NY, USA.

[Lamere, 2007] Lamere, P. (2007). Tagomendations - making recommendations transparent. http://blogs.sun.com/plamere/entry/tagomendations_making_recommedations_transparent. Retrieved on October 30, 2008.

[Lampe and Resnick,] Lampe, C. and Resnick, P. Slash (dot) and burn: Distributed moderation in a large online conversation space. *Proceedings of SIGCHI*, pages 543–550.

[Linden et al., 2003] Linden, G., Smith, B., and York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80.

[Markus Weimer, 2007] Markus Weimer, Iryna Gurevych, M. M. (2007). Automatically assesing the post quality in online discussions on software. In *Proceedings of the Association for Computational Linguistics*.

[Marlow et al., 2006] Marlow, C., Naaman, M., Boyd, D., and Davis, M. (2006). Position paper, tagging, taxonomy, flickr, article, toread. In *Collaborative web tagging workshop at WWW2006, Edinburgh, Scotland*.

[Metropolis et al., 1953] Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087.

[Millen et al., 2005] Millen, D., Feinberg, J., and Kerr, B. (2005). Social bookmarking in the enterprise. *ACM Queue*, 3(9):28–35.

[Mishne, 2006] Mishne, G. (2006). AutoTag: A collaborative approach to automated tag assignment for weblog posts. In *Proceedings of The 15th International Conference on the World Wide Web*, pages 953–954. ACM New York, NY, USA.

[O'Reilly, 2007] O'Reilly, T. (2007). What is Web 2.0: Design patterns and business models for the next generation of software. *Communications & Strategies, No. 1, p. 17, First Quarter 2007*.

[Ouellette and Wood, 1998] Ouellette, J. A. and Wood, W. (1998). Habit and intention in everyday life: The multiple processes by which past behavior predicts future behavior. *Psychological Review*, 127:54–74.

[Press et al., 1986] Press, W., Flannery, B., Teukolsky, S., and Vetterling, W. (1986). *Numerical recipes*. Cambridge University Press New York.

[Rader and Wash, 2008] Rader, E. and Wash, R. (2008). Influences on tag choices in Del.icio.us. In *Proceedings of CSCW 2008*.

[Raghavan and Wong, 1986] Raghavan, V. and Wong, S. (1986). A critical analysis of vector space model for information retrieval. *Journal of the American Society for Information Science*, 37(5):279–287.

[Rivadeneira et al., 2007] Rivadeneira, A. W., Gruen, D. M., Muller, M. J., and Millen, D. R. (2007). Getting our head in the clouds: Toward evaluation studies of tagclouds. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 995–998, New York, NY, USA. ACM.

[Rose, 2008] Rose, K. (2008). Digg: Recommendation engine rolling out this week. http://http://blog.digg.com/?p=127. Retrieved on October 30, 2008.

[Salton and Buckley, 1987] Salton, G. and Buckley, C. (1987). Term weighting approaches in automatic text retrieval. Technical report, Ithaca, NY, USA.

[Schafer et al., 2007] Schafer, J., Frankowski, D., Herlocker, J., and Sen, S. (2007). Collaborative filtering recommender systems. *Lecture Notes in Computer Science*, 4321:291.

[Schafer et al., 2002] Schafer, J. B., Konstan, J., and Riedl, J. (2002). Meta-recommendation systems: User-controlled integration of diverse recommendations. In *Proceedings of the ACM Conference on Information and Knowledge Management*, pages 43–51, MacLean, VA.

[Sen et al., 2006] Sen, S., Lam, S. K., Rashid, A. M., Cosley, D., Frankowski, D., Osterhouse, J., Harper, F. M., and Riedl, J. (2006). tagging, communities, vocabulary,

evolution. In *Proceedings of the ACM 2006 Conference on CSCW*, Banff, Alberta, Canada.

[Sen et al., 2009] Sen, S., Vig, J., and Riedl, J. (2009). Tagommenders: connecting users to items through tags. In *Proceedings of the 18th International Conference on World Wide Web*. ACM Press New York, NY, USA.

[Shirky, 2005] Shirky, C. (2005). Ontology is overrated. http://www.shirky.com/writings/ontology_overrated.html. Retrieved on May 26, 2007.

[Sood et al., 2007] Sood, S., Hammond, K., Owsley, S., and Birnbaum, L. (2007). TagAssist: Automatic tag suggestion for blog posts. In *International Conference on Weblogs and Social Media*.

[Uschold and Gruninger, 1996] Uschold, M. and Gruninger, M. (1996). Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, 11(2).

[von Ahn and Dabbish, 2004] von Ahn, L. and Dabbish, L. (2004). Labeling images with a computer game. In *Proceedings of SIGCHI*, pages 319–326.

[Voss, 2005] Voss, J. (2005). Measuring Wikipedia. In *Proceedings of the International Conference of the International Society for Scientometrics and Informetrics*, Stockholm, Sweden.