LinearTag Models:
Recommendations Using Linear User Profiles Based on Tags

Claudia Becerra[1], Fabio A. Gonzalez[1], Alexander Gelbukh[2]

[1] Universidad Nacional de Colombia, Bogotá D.C.,
Colombia

[2] CIC, Instituto Politécnico Nacional, Mexico City,
Mexico

cjbecerrac@unal.edu.co, fagonzalezo@unal.edu.co, www.gelbukh.com

Abstract. Recommender systems allow the exploration of large collections of products, the discovery of patterns in the products, and the guidance of users towards products that match their interests. Collaborative tagging systems allow users to label products in a collection using a free vocabulary. The aggregation of these tags, also called a Folksonomy, can be used to build a collective characterization of the products in a simple and recognizable vocabulary. In this paper, we propose a family of methods called LinearTag recommenders, which infer users preferences for tags to formulate recommendations for them. We dubbed these inferred user profiles as TagProfiles. We present experiments using them as an interaction artifact that allows users to receive new recommendations as they delete, add or reorder tags in their profiles. Additional experiments using the Movielens dataset, show that the proposed methods generate recommendations with an error margin similar, or even lower than the results reported by methods based on latent factors. Next, we compared TagProfiles against KeywordProfiles, which are profiles based on keywords extracted automatically from textual descriptions of products. This comparison showed that TagProfiles are not only more precise in their predictions, but they are also more understandable by users. At last, we developed a user interface of a movie recommender based on TagProfiles, which we tested with 25 users. This experience showed that TagProfiles are easier to understand and modify by users, allowing them to discover new movies as they interact with their profiles.

Keywords. Collaborative tagging systems, recommender systems, tagging.

1 Introduction

The rise of the Internet has allowed vast collections of data to be available online, democratizing both the generation of online content and the access to information by users. For example, in February 2016, the catalog of books at Amazon.com[1] had about 31 million books of which 135.998 were added during the previous 30 days. The exploration of this type of collections requires intelligent tools to guide users towards products of their interest while minimizing frustration due to the purchase of products that are not of interest to the users and spending valuable time exploring a collection without finding something interesting. Recommender systems are the primary of these tools.

Collaborative filtering algorithms have made great strides by introducing user and product profiling algorithms based on product evaluations provided by users. The patterns found in users evaluations can spread the good experiences of some users in the system, to others. In fact, these systems have demonstrated good ability to forecast the evaluations that users would give to products not yet known to them.

However, user profiles generated by these systems are usually vectors indexed by an abstract space, also named a latent space, devoid of interpretability. Therefore, users cannot modify

---

[1] http://www.amazon.com/b/?node=283155

their profiles to adjust the recommendations received, so users may feel enclosed in "the bubble" described by Pariser [15]. Moreover, some studies have shown that users feel more positive about the recommendations received when they have the opportunity to control them [8, 31].

At the moment, the typical way to escape "the bubble" is to evaluate repeatedly different products that have been searched manually. In this way, the system reacts to the change in the user preferences and refreshes the given recommendations. Unfortunately, this cycle it is not always successful due to limitations of users and systems. For example, the success of manual searches depends heavily on the knowledge that each user has about the product domain. In most cases, users face a blank search box in which they must formulate a query without knowing the vocabulary that characterizes products. For example, in the domain of movie films, for effective searching, it might be necessary to have in mind some names of directors, or actors.

This problem is known as the vocabulary problem [4]. Collaborative Tagging Systems [3, 6, 24], here referred as CTS, have made significant progress in the collective characterization of products using a free language. CTS allow users to tag, using their own words, the features they consider most relevant to the products of the collection. In addition, CTS also enable users to express their agreement with the tags that other users apply to the products. With these two resources ("tags" and "likes") and using vocabulary selection strategies such as those described in [23, 25] it is possible to build depurated vocabularies, also called folksonomies [18], that characterize products in a language that is natural and easy for most users in the system.

In this paper, we propose a family of recommendation models, called *LinearTag* models, which formulates its recommendations by inferring the affinity of users for tags by using product evaluations. Also, we experiment with these inferred profiles, or *TagProfiles*, using them as a tool of interaction that enables users to receive new tailored recommendations as they modify their profiles.

To test the performance of the proposed LinearTag methods, we experimented with them in three different ways. First, we compared our methods against those based on latent factors. Second, we compared our inferred user profiles (TagProfiles) against profiles generated using keywords (KeywordProfiles) extracted from product descriptions instead of tags. Third, we designed a user experience to let 25 users interact with their TagProfiles as a way to obtain new recommendations in a movie recommender system.

2 Preliminaries

2.1 Notation

To expose our models, we will use the notation used by Koren in many of his works [9, 10]. According to this notation the index variables i and j are reserved to refer to a particular user, and a product, respectively. The rating, that a user u gives of a product i, is denoted as r_{ui}. Those ratings vary between 1 and 5, where 1 means total disgust, and 5 total satisfaction. The set of tuples (user,product) for which the evaluations r_{ui} are known, are stored in the set K. Thus, $K = \{(u,i)\,|\,r_{ui}\,\text{is known}\}$. The aim of the recommendation methods is to estimate ratings, $\hat{r}_{ui}$, that minimize the prediction error in the training set:

$$\min \sum_{(u,i)\in \mathrm{K}} (r_{ui} - \hat{r}_{ui})^2 .$$

To prevent overfitting, it is necessary to regularize the models with constants that penalize the norm of the estimated variables. These regularization constants are denoted as: $\lambda_1, \lambda_2, \ldots$ and their optimal values are determined by cross validation.

Also, we assume that there is a tag vocabulary set Γ in which tags, referred as g, are stored. In the set Ψ, we store tuples (i, g) that indicate whether the tag g was applied to the movie i. Therefore, $\Psi = \{(i,g)\,|\,g\,\text{is applied to}\,i\}$.

2.2 Product Characterization Methodologies

In the Information Retrieval field, it is common to use a Vector Space Model (VSM) representation for the documents [22, 29]. In VSM, documents are represented as vectors indexed by terms. Similarly we need to represent products as vectors q_i, but instead of terms we use tags, so $q_i \in \mathbb{R}^{|\Gamma|}$, where individual q_{ig} coefficients, indicate the relevance of the tag g to the movie i:

$$q_i = \langle q_{i1}, q_{i2}, \ldots, q_{i|\Gamma|} \rangle.$$

In this work, we will use three methodologies to assign these coefficients: the hard-q_i methodology; a softened version denoted as soft-q_i and the supervised product characterization proposed by Vig et al. in the *TagGenome* work [31].

The hard-q_i methodology assigns 1 to the coefficient q_{ig} if the tag g was applied to the product i and 0 otherwise. Thus:

$$q_{ig(\text{hard})} = \begin{cases} 1 & \text{if } (i,g) \in \Psi, \\ 0 & \text{otherwise.} \end{cases}$$

The soft-q_i methodology assigns 0.7 and 0.1 depending on whether or not the tag g was applied to i:

$$q_{ig(\text{soft})} = \begin{cases} 0.7 & \text{if } (i,g) \in \Psi, \\ 0.1 & \text{otherwise.} \end{cases}$$

The *TagGenome* methodology, use a multilevel logistic regression model that estimates the probability that a tag g is relevant to the product i (a movie). As a ground truth, they use a survey, in which users assess the relevance of the tags to the movies. The TagGenome model estimates the value of the q_{ig} coefficients as follows:

$$\begin{aligned} q_{ig(\text{TG})} &= P(g \text{ is relevant to } i) \\ &= \frac{1}{1 + e^{-X_{gi}^T \cdot \beta_g}}, \end{aligned}$$

where coefficients β_g are the multilevel coefficients inferred by the logistic regression, that are specific for each tag. And X_{gi} is a vector, of seven features extracted for each pair movie-tag (i, g). Three of this features are similarity measures, between the movie i and the tag g, calculated using different user contributed resources such as tags, ratings,

and keywords extracted from text-based reviews. Other features are count variables, or averages such as the number of times the tag is applied to the item, number of times the tag appears in the product textual reviews, or the average rating of the product. The last variable is a meta-feature that seeks to incorporate to the survey data, those item-tag pairs selected for the TagGenome, on the assumption that for those pairs, the relevance measure is close to 1. As future work we plan to consider soft similarity in a sense presented in [30], when the similarity of features is taken into account.

3 Proposed LinearTag Models

In this work, we propose a family of tag-based recommender models called the LinearTag models. These models are the tag-based versions, of the Single Value Decomposition (SVD) models, denoted as SVD models (or latent factor models), described by Funk [2] and Paterek [16] on the occasion of the Netflix Prize [1]. Therefore, we will first introduce the Latent Factor Models, in which we inspired, and next, we will present the proposed LinearTag models.

3.1 Latent Factors Models

SVD typical models assume that a rating, $\hat{r}_{ui}$ is a similarity measure between a product u and a user i, both represented as vectors in an abstract (or latent) space of size f. Thus:

$$r_{ui} \approx \hat{q}_i^T \cdot \hat{p}_u \text{ where } \hat{q}_i, \hat{p}_u \in \mathbb{R}^f,$$

where $\hat{q}_i$ is the vectorial characterization of the product i, and $\hat{p}_u$ is the characterization of the user u. Usually, the dot product operation is used to calculate the similarity between pairs of those vectors. In this work we will use, two particular SVD models as baselines: the typical RSVD model, and the RSVD+B model (+B stands for "plus bias"). RSVD models, are named after the acronym of Regularized SVD models, in order to emphasize that the norm of the parameters estimated is controlled using a regularization constant λ.

Table 1 shows the equations that Typical RSVD and RSVD+B models use to predict their ratings.

Table 1. RSVD baseline methods

Model name	Rating model
RSVD [2]	$r_{ui} \approx \hat{q}_i^T \cdot \hat{p}_u;\ \hat{q}_i, \hat{p}_u \in \mathbb{R}^f$
RSVD+B [16]	$r_{ui} \approx \hat{q}_i^T \cdot \hat{p}_u + \mu + \hat{b}_i + \hat{b}_u$ $\hat{q}_i, \hat{p}_u \in \mathbb{R}^f,\ \hat{b}_u \in \mathbb{R}$

RSVD+B models [14] integrate, to the prediction, the user and the product bias, denoted respectively by $\hat{b}_i$ and $\hat{b}_u$. These bias estimates indicate the deviations, of users and product ratings, from the overall average rating in the system, denoted by μ.

3.2 Linear Tag Models

In this work, we propose the tag-based versions of the latent models shown in Table 1. Therefore in the proposed LinearTag models, we also characterize products and users as vectors, but instead of using a latent space, we use an interpretable space of tags of dimensionality $|\Gamma|$. Thus:

$$\hat{r}_{ui} \approx q_i^T \cdot \hat{p}_u \text{ where } \hat{q}_i, \hat{p}_u \in \mathbb{R}^{|\Gamma|}.$$

Here, we denote the characterization of the product as q_i, instead of $\hat{q}_i$, to emphasize that in LinearTag models the characterization of the product is known, whereas in latent factor models the product characterization $\hat{q}_i$ has to be estimated.

Table 2. Proposed family of LinearTag models

Model name	Rating model
LinearTag [12, 25]	$\hat{r}_{ui} = q_i^T \cdot \hat{p}_u;\ q_i, \hat{p}_u \in \mathbb{R}^{|\Gamma|}$
LinearTag+B	$\hat{r}_{ui} = q_i^T \cdot \hat{p}_u + \mu + \hat{b}_i + \hat{b}_u$ $\hat{p}_u \in \mathbb{R}^{|\Gamma|};\ \hat{b}_i, \hat{b}_u \in \mathbb{R}$
Adjst. LinearTag+B	$r_{ui} = \sum_{g \in \Gamma} q_{ig} \hat{c}_{ig} \hat{p}_{ug} +$ $\mu + \hat{b}_i + \hat{b}_u$ $q_i, \hat{c}_i, \hat{p}_u \in \mathbb{R}^{|\Gamma|};\ \hat{b}_i, \hat{b}_u \in \mathbb{R}$

With this basic idea, we formulate the family of models presented in Table 2. As it can be observed, Typical LinearTag and LinearTag+B models are the tag-based versions of RSVD and RSVD+B. The Adjusted LinearTag+B method is different because it adjusts the characterization

coefficients of the product, q_{ig}, by estimating the adjustment coefficients $\hat{c}_{ig}$. Therefore, in the Adjusted LinearTag+B model, $\hat{q}_{ig} = q_{ig} \times \hat{c}_{ig}$.

We are aware that different authors have proposed the Typical LinearTag model before us [12, 25]. However, we include it in the family LinearTag because it is the starting point of the family and its behavior can be significantly improved by either using the soft-q_i, or the TagGenome product characterization.

The estimation of the set of unknown parameters $X = \hat{p}_u, \dots$ is calculated, in the same way that RSVD models do [2, 16], by minimizing the regularized error of prediction. The regularization is controlled by adding a term that penalizes the norm of the estimates, using the λ constant. Thus:

$$\min_{X=\{\hat{p}_u, \hat{b}_u \dots\}} F(X) = \sum_{(u,i) \in K} (r_{ui} - \hat{r}_{ui})^2 + \lambda \left(\|\hat{p}_u\|^2 + \left\|\hat{b}_u\right\|^2 + \dots \right).$$

The minimization of the error function is achieved by applying the gradient descent method in successive epochs. In each epoch, the estimation of the parameters is decremented by the gradient of the function to minimize, $F(X)$, multiplied by the learning rate ξ. Therefore, if we represent the parameter to be estimated by x_j, where $x_j \in X$, the value of parameter x_j is modified using the following update rule:

$$x_j \mathrel{+}= -\xi \frac{dF(X)}{dx_j}.$$

In the following subsections, we describe each of the three proposed methods, indicating for each method the error function to minimize, the meaning of the parameters, and the necessary update rules to estimate the values of parameters. To shorten the expressions, we will denote by ϵ_{ui}, the difference between a known evaluation and its estimated value:

$$\epsilon_{ui} = r_{ui} - \hat{r}_{ui}.$$

3.2.1 The Typical LinearTag Model

The typical LinearTag model is the simplest of the LinearTag models proposed. As shown in Table 2, In this model, the user profile, $\hat{p}_u$, is the only unknown variable and its value is found by minimizing the function:

$$\min_{X=\{\hat{p}_u,...\}} F(X) = \sum_{(u,i)\in K} \left(r_{ui} - q_i^T \cdot \hat{p}_u\right)^2 + \lambda_1 \|\hat{p}_u\|^2.$$

The assignation of the q_{ig} coefficients is made using either the hard-q_i, the soft-q_i or the TagGenome methodologies explained in section 2.2 Next, we find the coefficients $\hat{p}_{ug}$ using the gradient descent method. Therefore, by obtaining the derivative of $F(X)$ respect to p_{ug}, we get the update rule:

$$\hat{p}_{ug} \mathrel{+}= \xi\left(q_{ig}\left(\epsilon_{ui}\right) - \lambda_1\hat{p}_{ug}\right).$$

3.2.2 Proposed LinearTag+B Model

As stated before, the LinearTag+B method includes the user and product bias, denoted respectively as $\hat{b}_u$ and $\hat{b}_i$. User biases can be considered the inherent tendency of users to give higher or lower ratings than the overall average rating in the system μ. Likewise, the product bias $\hat{b}_i$, indicates the tendency of certain products to receive higher or lower ratings than the average. For example, the high budget movie Titanic, tends to receive higher rating than similar movies. As in the previous model, unknown parameters $X = \left\{p_u, \hat{b}_u, \hat{b}_i\right\}$ are estimated by minimizing the regularized expression of the prediction error:

$$\min_{\substack{\hat{p}_u \in \mathbb{R}^{|\Gamma|}, \\ \hat{b}_u, \hat{b}_i \in \mathbb{R}}} \left[\sum_{(u,i)\in K} \left(r_{ui} - \left(q_i^T \cdot \hat{p}_u + \mu + \hat{b}_i + \hat{b}_u\right)\right)^2 + \right.$$
$$\left. \lambda_2 \left(\|\hat{p}_u\|^2 + \left\|\hat{b}_i\right\|^2 + \left\|\hat{b}_u\right\|^2\right)\right].$$

Its parameters are estimated using the following update rules:

$$\hat{p}_{ug} \mathrel{+}= \xi(q_{ig}\epsilon_{ui} - \lambda_2\hat{p}_{ug}),$$
$$\hat{b}_i \mathrel{+}= \xi(\epsilon_{ui} - \lambda_2\hat{b}_i),$$
$$\hat{b}_u \mathrel{+}= \xi(\epsilon_{ui} - \lambda_2\hat{b}_u).$$

3.2.3 Proposed Adjusted LinearTag+B Model

The Adjusted LinearTag+B model tries to adjust the initial characterization of the movies, in a collaborative way, by introducing in the expression to minimize, an adjustment vector denoted by $\hat{c}_i$. Therefore the new product characterization coefficients, denoted by $\hat{q}_{ig}$, are adjusted by multiplying its initial value by the adjustment coefficients $\hat{c}_{ig}$. Thus, $\hat{q}_{ig} = q_{ig}\hat{c}_{ig}$, and therefore, the expression to minimize changes to:

$$\min_{\substack{\hat{p}_u \in \mathbb{R}^{|\Gamma|}, \\ \hat{b}_u, \hat{b}_i \in \mathbb{R}}} \left[\sum_{(u,i)\in K} \left(r_{ui} - \sum_{g\in\Gamma}\left(q_{ig}\hat{c}_{ig}\hat{p}_{ug} + \mu + \hat{b}_i + \hat{b}_u\right)\right)^2 \right.$$
$$\left. + \lambda_3 \left(\|\hat{p}_u\|^2 + \|\hat{c}_i\|^2 + \left\|\hat{b}_i\right\|^2 + \left\|\hat{b}_u\right\|^2\right)\right].$$

It has the following update rules:

$$\hat{p}_{ug} \mathrel{+}= \xi(q_{ig}\hat{c}_{ig}(\epsilon_{ui}) - \lambda_3\hat{p}_{ug}),$$
$$\hat{c}_{ig} \mathrel{+}= \xi(q_{ig}\hat{p}_{ug}(\epsilon_{ui}) - \lambda_3\hat{c}_{ig}),$$
$$\hat{b}_i \mathrel{+}= \xi(\epsilon_{ui} - \lambda_3\hat{b}_i),$$
$$\hat{b}_u \mathrel{+}= \xi(\epsilon_{ui} - \lambda_3\hat{b}_u).$$

4 Experiments

To experimentally validate the proposed models, we conducted experiments in three different ways. First, we compared the performance of the LinearTag methods against RSVD methods already described in Table 1. Second, we applied the LinearTag methods to a different, but interpretable space, in order to compare the inferred user profiles. Therefore, instead of using a space of tags, we represented products and users in a space of keywords automatically extracted from movie descriptions. The extracted keywords were weighted using a technique similar to *tf-idf*. Third, and final, we ran an experiment with 25 users to investigate if users can understand and interact with their inferred user profiles as an artifact to control the recommendations that they receive.

4.1 Experiment 1: Comparing LinearTag Methods against RSVD methods

4.1.1 Data

To measure the prediction error of the proposed models, we use the Movielens 1M dataset [7]. This dataset has 1,000,209 ratings of 3,900 films, given by 6,400 users. This data was collected with the movie recommender system www.movielens.org. The used tag vocabulary was the 1,129 tags of the TagGenome work [31], which were selected using the quality measure proposed by Sen et al. [23]. The characterization of products, as vectors q_i, in the space of tags, was made by using the three methodologies for product characterization explained in section 2.2: the hard-q_{ig}, the soft-q_{ig} and the TagGenome methodologies. Recall that, unlike hard q_{ig} and soft q_{ig} methodologies that only use tag applications, the TagGenome approach also requires textual reviews of the movies. For our experimentation, we just use the TagGenome relevance values publicly available at http://files.grouplens.org/datasets/tag-genome. Table 3 shows an example of the ten most relevant tags applied to the movies Toy Story and Rambo, and the relevance values assigned by the logistic regression.

4.1.2 Evaluation Measure

The metric used to measure the prediction error was the root-mean-square error (RMSE) calculated as follows:

$$RSME = \sqrt{\frac{\sum_{(u,i)\in TestSet}(r_{ui}-\hat{r}_{ui})^2}{|TestSet|}}.$$

All values of RMSE are averages over five randomly selected test folds. On each fold, 80% of the dataset was used for training and 20% for testing.

Table 3. Example of the TagGenome folksonomy

Toy Story (1995)		Rambo (2008)	
Tag	Relev.	Tag	Relev.
toys	0.999	world war ii	0.998
computer animation	0.998	wwii	0.987
pixar animation	0.995	best war films	0.973
kids and family	0.995	wartime	0.965
animation	0.987	war	0.951
animated	0.979	true story	0.944
children	0.973	gunfight	0.923
cartoon	0.947	story	0.863
pixar	0.941	germany	0.862
disney anim. feat.	0.929	forceful	0.856
imdb top 250	0.924	biographical	0.832
kids	0.922	tense	0.825
story	0.913	courage	0.822
fun	0.900	history	0.793
childhood	0.898	historical	0.770
adventure	0.894	runaway	0.761
light	0.884	war movie	0.754
original	0.880	talky	0.708
great	0.872	dramatic	0.700
great movie	0.859	sacrifice	0.697

4.1.3 Baseline Method and Initial Values for Parameters

We use the RSVD and RSVD+B methods, described in Table 1, as baselines. We fixed the initial parameters as follows: epochs number: 100, learning rate $\xi = 0.001$; regularization constants $\lambda_1 = \lambda_2 = \lambda_3 = 0.011$, dimensions of the latent space $f = 1,129$ (the same number of tags in the TagGenome folksonomy $|\Gamma|$).

4.1.4 Results and Discussion

Table 4 shows the results obtained for each methodology of product characterization. Since the dawn of collaborative filtering methods [5, 20, 28], *content-based filtering* systems have been considered their counterpart. Therefore, the success of collaborative latent factor models [1, 11], has brought harsh criticisms over content-based recommenders, mainly due to their poor performance [13, 17, 19]. Our results might provide evidence that supports these

criticisms when comparing the poor $RMSE = 2.20035$ achieved by the Typical LinearTag method against the $RMSE = 0.86446$ attained by the Typical RSVD method. Fortunately, our results also gives evidence against those criticisms when the $RMSE = 0.83525$ reached by the Adjusted-LinearTag+B method is observed. As can be seen, content-based recommender systems can be enriched with some features that make them similar or even better than the methods based on latent factors.

Therefore, first, we can observe that Typical LinearTag $RMSE$ is reduced from 2.20035 to 0.96330, and 0.89355 when products are characterized using either the soft-q_i or the TagGenome methodology instead of the hard-q_i. The reason is that hard-q_i characterization prevents the algorithm, to use not applied tags to estimate their ratings, whereas the soft-q_i and the TagGenome permit to do it. Then we can conclude that using a hard-q_i characterization that only works with the tags explicitly applied to the movies has a negative impact in the RSME achieved.

Besides, LinearTag + B methods generate a considerable reduction of the RMSE in every method tested. Even the $RMSE = 0.85265$ reached by the LinearTag+B method using the TagGenome product characterization is better than the $RSME = 0.86420$ attained by the Baseline RSVD+B. So, we can say that the incorporation of product and user biases into the rating prediction rules make the algorithms smarter as they can distinguish better products among others with similar characteristics, and consequently, they can formulate better recommendations. Furthermore, the Adjusted LinearTag+B method decreases the RMSE in all sets of features by adjusting the characterization of the products in a collaborative way. This last improvement, produced a reduction of 3.3% outperforming the RMSE attained by the baseline method RSVD+B.

4.2 Experiment 2: Comparing LinearTag versus LinearKeyword profiles

In this section, we compare the inferred user tag profiles generated against user profiles defined in other interpretable spaces. For this purpose,

Table 4. Rating prediction RMSE results obtained using Movielens1M+TagGenome folksonomy

Model	Features	RSME
Baseline Typical RSVD	Latent factors	0.86446
Baseline RSVD+B	Latent factors	**0.86420**
Typical LinearTag	hard-$q_i$tags	2.20035
	soft-$q_i$tags	0.96330
	TagGenome	0.89355
Proposed LinearTag+B	hard-$q_i$tags	0.90123
	soft-$q_i$tags	088835
	TagGenome	0.85265
Adjusted LinearTag+B	hard-$q_i$tags	0.88629
	soft-$q_i$tags	0.87565
	TagGenome	**0.83525**

we extracted the words in the movie descriptions and characterized movies in a vector space of keywords weighting them using a tf-idf like methodology. We call this a *keyword space*.

4.2.1 Data

We tested our proposed methods over the same Movielens 1M dataset used in the previous experiment, but we added the characterization of movies in the keyword space.

For this purpose, we first gathered movies synopses and casts descriptions from Netflix's web pages. Next, we characterized movies as bags of words d_i. And finally, we preprocessed the aggregated vocabulary Σ to reduce its dimensionality.

We preprocess synopses by concatenating people names and last names (e.g. TomCruise); removing numbers and stop words; and deleting words that occurred whether in less than ten movies or in more than 95% of the movies. After the preprocessing stage, we obtained a vocabulary Σ with a dimensionality of 6.155 words.len

Next, we used the Okapi BM25 [21] weighting scheme as the tf-idf like methodology. Thus:

$$q_{iw(BM25)} = \log\left(\frac{M-df(w)}{M}\right)\frac{(k_1+1)tf(w,d_i)}{K+tf(w,d_i)},$$
$$K = k_1\left((1-b)+b\frac{len(d_i)}{avdl}\right),$$

where coefficients q_{iw} indicate the relevance of the word w to the movie i, $df(w)$ is the number

of movies in which appeared the word w, M is the number of movies, $tf(w, d_i)$ is the number of occurrences of word w in the movie bag-of-words d_i, $\text{len}(d_i)$ is the number of words in d_i, and $avdl = 33$ is the average of words on each document d_i. According to [21], constants k_1 and b where assigned to 1.2 y 0.75. Table 6 shows movies "Bewitched" and "Rocky V" characterized in a keyword space using the Okapi BM25 weighting scheme.

Table 5. Family of LinearTag models

Model name	Rating model
LinearKeyword	$\hat{r}_{ui} = q_i^T \cdot \hat{p}_u;\ q_i, \hat{p}_u \in \mathbb{R}^{|\Gamma|}$
LinearKeyword+B	$\hat{r}_{ui} = q_i^T \cdot \hat{p}_u + \mu + \hat{b}_i + \hat{b}_u$ $\hat{p}_u \in \mathbb{R}^{|\Gamma|};\ \hat{b}_i, \hat{b}_u \in \mathbb{R}$
Adjst. LinearKeyw.+B	$r_{ui} = \sum_{g \in \Gamma} q_{ig} \hat{c}_{ig} \hat{p}_{ug} + \mu + \hat{b}_i + \hat{b}_u$ $q_i, \hat{c}_i, \hat{p}_u \in \mathbb{R}^{|\Gamma|};\ \hat{b}_i, \hat{b}_u \in \mathbb{R}$

Table 6. Example of movies characterized using Okapi BM25

Bewitched (2005)		Rocky V (1990)	
Tag	BM25	Tag	BM25
WillFerrell	0.237	BurtYoung	0.249
Jack	0.147	TaliaShire	0.242
update	0.142	broke	0.15
Samantha	0.131	upandcoming	0.15
sitcom	0.131	shots	0.15
witch	0.119	boxer	0.15
NicoleKidman	0.119	crooked	0.142
convinced	0.116	trainer	0.136
MichaelCaine	0.114	glory	0.136
right	0.107	accountant	0.131
hoping	0.105	ended	0.131
know	0.103	lifetime	0.128
career	0.099	memory	0.124
perfect	0.098	training	0.124
doesn't	0.097	rocky	0.121
actor	0.092	inspired	0.107
make	0.068	taking	0.101
film	0.045	career	0.099

4.2.2 Results and Discussion

Table 7 shows the results obtained with the LinearKeyword methods. For ease of comparison, we transcribed some results obtained by the LinearTag methods. From these results, we can draw three conclusions. i) LinearTag methods attain lower RMSEs than LinearKeyword methods, ii) in the same way that LinearTag methods do, LinearKeyword methods reach an important reduction in their RMSEs when user and product bias estimators are incorporated in the rating prediction, lowering the RMSE from 3.73389 to 0.90912, and iii) the method Adjusted LinearTag+B does not produce improvement when it is used with a product characterization based on a tf-idf like methodology.

Table 7. RMSE in Movielens1M+Netflix Keywords

Model	Weights	RSME
Typical LinearKeyword	BM25	3.73389
Typical LinearTag	soft-q_i	0.96330
Proposed LinearKeyword+B	BM25	0.90912
Proposed LinearTag+B	soft-q_i	0.88835
Adjusted LinearKeyword+B	BM25	0.90914
Adjusted LinearTag+B	soft-q_i	**0.87565**

Table 8 and Table 9 show respectively the TagProfile and KeywordProfile of a particular user from the Movielens system. If we observe the 'Liked' 'column in Table 8, we can see that the user likes horror, sci-fi and actions movies. Whereas, if we read his (or her) KeywordProfile we have to do a little research of the proper names to understand how both profiles describe the preferences of the same user. From this research, we learned that Angela Bassett has had a recurring role on horror series, as well as Michael Rooker, who is famous for his participation in "The Walking Dead" and "Henry: Portrait of a Serial Killer". Following the comparison, keywords such as such as "thriller" and "killer" seem informative, but we also find non-informative keywords such as "crash", "city" and "school" that, at first sight, does not seem to give any clue about user's preferences.

Same happens when we compare the columns labeled "Disliked" of the inferred user profiles. Again, with a little research we learned that

Table 8. An example of the ten most liked and disliked tags in a TagProfile of a random user

	TAGS			
rank	Liked	p_{ug}	Disliked	p_{ug}
1	true story	0.182	light	0.238
2	vampire	0.179	talky	0.185
3	crime	0.160	queer	0.182
4	creepy	0.151	world war ii	0.164
5	Coming of age	0.149	cross-dressing	0.149
6	rape	0.145	oscar (best cinem	0.142
7	vampires	0.140	humorous	0.139
8	action	0.137	muppets	0.138
9	sci-fi	0.136	africa	0.127
10	suspense	0.134	multiple story lines	0.122

Table 9. An example of the ten most liked and disliked keywords in a KeywordProfile of the same user from Table 8

	KEYWORDS			
rank	Liked	p_{ug}	Disliked	p_{ug}
1	thriller	0.016	best	0.011
2	MichaelRooker	0.015	Won	0.009
3	killer	0.015	WoodyAllen	0.008
4	crash	0.013	Madonna	0.006
5	TomSizemore	0.013	SpikeLee	0.006
6	AngelaBassett	0.012	comedy	0.005
7	City	0.012	NeilJordan	0.005
8	marty	0.012	BobHoskins	0.004
9	scott	0.012	JackWarden	0.003
10	School	0.012	FrankOz	0.003

keyword "FrankOz" in the KeywordProfile can be aligned to the tag "muppets" in the TagProfile because Frank Oz performed the Muppet characters of Miss Piggy and Fozzie Bear. Also, actor Jack Warden seems to have participated in several comedies. Therefore, with these observations, we conclude that when compared against keywords, tags vocabularies are more suitable to express people preferences, or at least they require less domain knowledge to be understood.

4.3 Experiment 3: Using TagProfiles as an Artifact of Interaction

To test the TagProfiles generated, we developed an movie-recommender-system interface based on TagProfiles and tested it with 25 users. This interface builds users' TagProfiles as users rate movies and then formulates new recommendations as users modify their TagProfiles. Individuals were undergraduate and graduate computer science students at the Universidad Nacional de Colombia.

4.3.1 Survey Questions

The purpose of providing users with their explicit TagProfiles was manifold. Particularly we wanted to find answers to the following questions:

1. Do users consider that their TagProfiles properly express their preferences?

2. Might users learn new tags that better define their preferences, by modifying their profiles?

3. Do users understand the TagProfile interaction concept?

4.3.2 An Interface based on TagProfiles

We developed a prototype of a movie recommender system based on LinearTag models. The collection of movies used is the one of the Movie-lens 1MB dataset, with the same tag folksonomy used in the experiment 1. We characterize movies using the soft-q_i characterization methodology, which only needs the information of the set of tags applied to each movie.

Figure 1 and Figure 2 show the two phases of the TagProfiles interface developed. In the first phase, users rate movies as they observe how their TagProfiles are inferred and displayed. In the second phase, users obtain new recommendations as they criticize their profiles by editing them. Figure 1 shows an example of interaction during the first phase of the TagProfiles interface. In the example shown in the screen-shot, a hypothetical user rated several movies starring Tom Hanks with five starts and several movies starring Jackie Chan with one star. Then, it can be seen that the system inferred that user's most liked tags

are "drama", "greatActing", "basedOnaBook" and so on. The most disliked tags are "martialArts", "kungFu" and so on. Figure 2 shows an example of the interaction on phase two. Now, the same user inserts into his (or her) profile, tags related with Japanese animes such as "Miyazaki", "Studio Ghibli", "animated" and so on. Then, the system recommends Miyazaki's most awarded movies such as "Spirited Away" and "Howl´s Moving Castle".
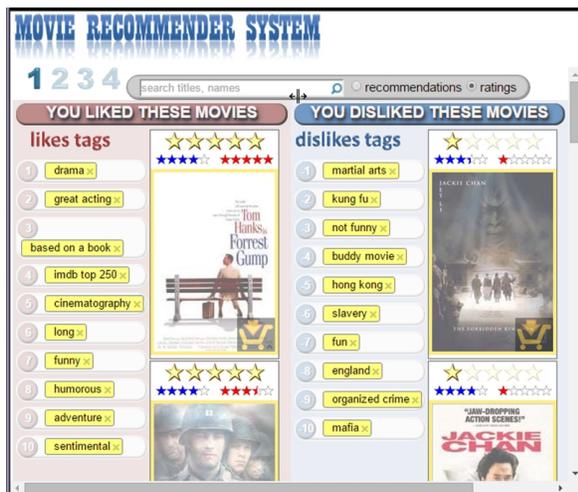
Fig. 1. Screen-shoot of the first phase of the TagProfiles Interface

4.3.3 Users Experience

We introduced the system to participants showing them a brief tutorial on the system and then asked them to rate at least 20 movies before they change to phase 2. Then, in phase 2 users were invited to refine their profiles by reordering and deleting the tags they do not agree with and adding new tags until their profile reflected their preferences. As users modified their profiles, they received new recommendation; so we asked users to add to their wish-lists those movies they considered interesting. The experience finished when users completed five movies in their wish-list. Finally, we asked them to fill a little survey with the following questions:

Fig. 2. Screen-shoot of the second phase of the TagProfiles interface

1. Do you consider that your preferences are properly expressed by your TagProfile?

2. Might you enumerate three new tags that you discovered from your profile?

3. Did you discover new movies using the TagProfile interface?

4.3.4 Results and Discussion

We got the following results:

1. 90% of users consider their profiles properly capture their preferences.

2. 100% of users found at least three new tags and enumerate them.

3. 100% of users filled their wish-lists with at least five movies.

From this experience we draw three conclusions. First, collaborative tags definitively are a suitable vocabulary to express user preferences. Second, TagProfiles, used as artifacts of exploration, help users to elicit their preferences relieving the vocabulary problem. Third, TagProfiles empower users to explore actively the data collection in a very personalized way.

5 Related Work

The research for models based on tags and the comparison of them against models based on latent models is not new. In fact, some hybrid models reach similar RMSE rates as the model TagiCofi proposed by Zhen et al. [32], or the model Hybrid-Tag proposed by Sen et al. [25]. These hybrid models produce rating estimations that can be viewed as a weighted average of the predictions given by tag-based models and latent-factor models. In spite of their precision, we did not compare our proposed models against them because we are interested only in fully interpretable models to generate linear profiles (or TagProfiles) that let users explore a collection in a personalized way by modifying this profile. For this reason, we focus our attention on the improvement of linear models based on tags.

Regarding those models, in the work *Tagommenders* [26]*,* Sen et al. propose the models Implicit-Tag, Implicit-Tag-pop and Regress-Tag. Similarity, Nguyen et al. [14] proposed variations with reduced dimensionality of the Regress-Tag model named Implicit-Tag, Implicit-Tag-Pop. Among these models, Regress-Tag is the only one that produces numerical estimations of ratings using the following linear regression:

$$\hat{r}_{ui} \quad = \quad \sum_{g \in \Gamma} \hat{c}_{ug} \cdot p_{ug} + \hat{b}_u;$$

$$p_u \in \mathbb{R}^{|\Gamma|},$$

where coefficients $\hat{c}_{ug}$ are the weights assigned to the user affinity coefficients p_{ug}, in order to minimize the prediction error. The coefficients p_{ug} are not estimated in the linear regression because they are calculated in a previous phase. The coefficients p_{ug} are calculated using a Bayesian generative model that believes that ratings r_{ug}, that users give to movies with an specific tag g, are observed variables produced by a hidden variable p_{ug}, which in turn is generated by a normal distribution whose media and variance are estimated by maximizing the likelihood of the ratings r_{ug} observed. According to the reported results, Regress-Tag was the best method but could not attain the error margin of RSVD baseline, producing a prediction error 3.57% superior.

On the other hand, the tag-based recommendation method proposed by Nguyen et al. [14] reached the prediction error attained by the latent factor methods. This model predicts ratings using the same regression model used by the Regress-Tag method. In addition, this model produces low-dimensional TagProfiles with only those tags that can predict more of the 50% of the rating using a metric based on mutual information [27]. The main issue of this approach is that predictions can only be formulated and presented to users with this kind of tags. For other users, it is necessary to lower the percentage of prediction, and when this happens, the prediction error increases in a considerable way. For example, when the percentage of prediction is lowered from 50% to 40% the error measure increases by a 46%, surpassing the RMSE produced by latent factor models by 41%.

From this review, we conclude that at the moment, there are not approaches that reach the same RMSE errors attained by our methods.

6 Conclusion

In this work, we experimented with building user profiles based on tags as a resource to generate recommendations and also as a tool for providing a personalized browsing experience of a data collection for the users. For this purpose, we proposed a set of methods (LinearTag) that generates interpretable linear user profiles achieving similar or even lower prediction errors than the ones produced by the methods based on latent factors in the state-of-the-art.

Next, we compared profiles based on tags (TagProfiles) against those based on keywords automatically excerpted from product descriptions. Unlike tags, keywords were more difficult to understand requiring, in some cases, domain knowledge for interpretation and in other cases, they cannot be interpreted in the context of user preferences.

Finally, we experienced with 25 users who interacted with their TagProfiles with the aim of getting new recommendations by modifying their profiles. This experience showed that: i) users interacted with their TagProfiles in a

natural way; ii) users easily improved their tag vocabulary as they interacted with their profiles; and iii) users participated actively in the process of adjusting their profiles as they received new recommendations better adjusted to their preferences.

Acknowledgements

References

1. **Bennet, J. & Lanning, S. (2007).** The netflix prize. *KDD Cup and Workshop*, volume 2007, pp. 35.

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

3. **Furnas, G., Fake, C., von Ahn, L., Schachter, J., Golder, S., Fox, K., Davis, M., Marlow, C., & Naaman, M. (2006).** Why do tagging systems work? *CHI'06 extended abstracts on Human factors in computing systems*, ACM, pp. 36–39.

4. **Furnas, G. W., Landauer, T. K., Gomez, L. M., & Dumais, S. T. (1987).** The vocabulary problem in human-system communication. *Commun. ACM*, Vol. 30, No. 11, pp. 964–971.

5. **Goldberg, D., Nichols, D., Oki, B., & Terry, D. (1992).** Using collaborative filtering to weave an information tapestry. *Commun*, Vol. 35, No. 12, pp. 61–70.

6. **Golder, S. & Huberman, B. (2005).** The structure of collaborative tagging systems. Arvix preprint cs.DL/0508082.

7. **Harper, F. & Konstan, J. (2015).** The movielens datasets: History and context. *ACM Trans. Interact. Syst.*, Vol. 5, No. 4, pp. 1–19.

8. **Harper, F., Xu, F., Kaur, H., Condiff, K., Chang, S., & Terveen, L. (2015).** Putting users in control of their recommendations. *Proceedings of the 9th ACM Conference on Recommender Systems*, ACM, pp. 3–10.

9. **Koren, Y. (2008).** Factorization meets the neighbor-hood: A multifaceted collaborative filtering model. *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, pp. 426–434.

10. **Koren, Y. (2010).** Collaborative filtering with temporal dynamics. *Communications of ACM*, Vol. 53, No. 4, pp. 89–97.

11. **Koren, Y., Bell, R., & Volinsky, C. (2009).** Matrix factorization techniques for recommender systems. *Computer*, Vol. 42, No. 8, pp. 30–37.

12. **Lewis, D., Schapire, R., Callan, J., & Papka, R. (1996).** Training algorithms for linear text classifiers. *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, pp. 298–306.

13. **Lops, P., Gemmis, M., & Semeraro, G. (2011).** *Recommender Systems Handbook*, chapter Content-based Recommender Systems: State of the Art and Trends. Springer US, pp. 73–105.

14. **Nguyen, T. & Riedl, J. (2013).** *UMAP 2013*, chapter Predicting Users Preference from Tag Relevance. Springer Berlin Heidelberg, pp. 272–280.

15. **Pariser, E. (2011).** *The filter bubble: What the Internet is hiding from you*. Penguin UK.

16. **Paterek, A. (2007).** Improving regularized singular value decomposition for collaborative filtering. *Proc. KDD Cup Workshop at SIGKDD' 07 13th ACM Int. Conf. on Knowledge Discovery and Data Mining*, ACM, pp. 39–42.

17. **Pazzani, M. & Billsus, D. (2007).** *The Adaptive Web*, chapter Content-Based Recommendation Systems. Springer Berlin Heidelberg, pp. 325–341.

18. **Peters, I. (2009).** *Folksonomies. Indexing and Retrieval in Web 2.0*, volume 1. Walter de Gruyter & Co., Hawthorne, NJ, USA.

19. **Pilászy, I. & Tikk, D. (2009).** Recommending new movies: Even a few ratings are more valuable than metadata. *Proceedings of the Third ACM Conference on Recommender Systems*, ACM, pp. 93–100.

20. **Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994).** Grouplens: An open architecture for collaborative filtering of netnews. *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, ACM, pp. 175–186.

21. **Robertson, S. (2005).** *TREC: Experiment in Information Retrieval*, chapter How Okapi Came to TREC. MIT Press, pp. 287–300.

22. **Salton, G., Wong, A., & Yang, C. (1975).** A vector space model for automatic indexing. *Commun. ACM*, pp. 613–620.

23. **Sen, S., Harper, F., LaPitz, A., & Riedl, J. (2007).** The quest for quality tags. *Proceedings of the 2007 International ACM Conference on Supporting Group Work*, ACM, pp. 361–370.

24. **Sen, S., Lam, S., Rashid, A., Cosley, D., Frankowski, D., Osterhouse, J., Harper, F., & Riedl, J. (2006).** Tagging, communities, vocabulary, evolution. *Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work*, ACM, pp. 181–190.

25. **Sen, S., Vig, J., , & Riedl, J. (2009).** Learning to recognize valuable tags. *Proceedings of the 14th international conference on Intelligent user interfaces*, ACM, pp. 87–96.

26. **Sen, S., Vig, J., & Riedl, J. (2009).** Tagommenders: connecting users to items through tags. *Proceedings of the 18th international conference on World wide web*, ACM, pp. 671–680.

27. **Shannon, C. (2001).** A mathematical theory of communication. *SIGMOBILE Mob. Comput. Commun. Rev.*, Vol. 1, pp. 3–55.

28. **Shardanand, U. & Maes, P. (1995).** Social information filtering: Algorithms for automating "word of mouth". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM Press, Addison-Wesley Publishing Co., pp. 210–217.

29. **Sidorov, G. (2013).** *Non-linear construction of n-grams in computational linguistics: syntactic, filtered, and generalized n-grams [in Spanish]*.

30. **Sidorov, G., Gelbukh, A. F., Gómez-Adorno, H., & Pinto, D. (2014).** Soft similarity and soft cosine measure: Similarity of features in vector space model. *Computación y Sistemas*, Vol. 18, No. 3, pp. 491–504.

31. **Vig, J., Sen, S., & Riedl, J. (2012).** The tag genome: Encoding community knowledge to support novel interaction. *ACM Transactions on Interactive Intelligent Systems*, Vol. 2, No. 3, pp. 13.

32. **Zhen, Y., Li, W.-J., & Yeung, D.-Y. (2009).** Tagicofi: Tag informed collaborative filtering. *Proceedings of the Third ACM Conference on Recommender Systems*, ACL, pp. 69–76.