Predicting Users' Preference from Tag Relevance

Tien T. Nguyen and John Riedl

GroupLens Research, Computer Science and Engineering,
University of Minnesota, Minneapolis, MN 55455
nguy1749@umn.edu, riedl@cs.umn.edu
http://www.grouplens.org/

Abstract. Tagging has become a powerful means for users to find, organize, understand and express their ideas about online entities. However, tags present great challenges when researchers try to incorporate them into the *prediction task* of recommender systems. In this paper, we propose a novel approach to infer user preference from tag relevance, an indication of how strong each tag applies to each item in recommender systems. We also present a methodology to choose tags that tell most about each user's preference. Our preliminary results show that at certain levels, some of our algorithms perform better than previous work.

Keywords: algorithms, recommender system, mutual information, tag relevance.

1 Introduction

Tagging is, today, a popular means for users to organize, and express their opinions about any item of their interests. It also serves as a means to retrieve information. Originally, tags were used mainly in retrieval systems. Subsequently, they are incorporated into other domains such as recommender systems, since tags are considered as what users think about items [5]. Sen et al. propose that tags are also bridges connecting users and items [3]. These bridges help users find and make decision about items. Sen et al. credit Paul Lamere and his colleagues with coining the term "tagomendations" to refer to any recommendation systems that incorporate tags into the *prediction task* (predicting users' preferences for some items in the collections).

However, incorporating tags into the *prediction task* is challenging due to two characteristics of the traditional tagging model. First, tags are ***binary.*** When a user applies a tag to a item, he can only indicate a binary relationship between the tag and the movie [7]. For example, a movie can be tagged as *romantic*, but this tag alone cannot describe how romantic the movie is. Second, tags are ***sparse.*** Not all users tag all the movies they watched [7]. Moreover, it is unlikely that they apply the same sets of tags to the same set of movies.

To over come the two limitations, some researchers, such as Sen et al. [3], have successfully inferred user preference from a tag via tag-clicks, or tag-search, and then incorporate these preferences into the *prediction task*. Sen et al. introduce a tag-based recommender system that infers user preferences from tags. In order to capture these preference signals, they propose several algorithms, such as: 1) users have higher preference for tags if they themselves apply these tags (**tag-applied**) , or 2) if they themselves searched for these tags (**tag-searched**), or 3) if the tags are of high quality (**tag-quality**)[4]. Their analysis shows that prediction accuracy can be improved if these preference signals are taken into account. However, inferring user preference via users' actions on tags like Sen et al. [3] requires extra efforts such as tagging, searching, or clicking from users. Furthermore, Sen et al. [3] propose constant numbers of tags selected for their algorithms. Nevertheless, different users have different preference, leading to different numbers of tags required to express individual preference.

To avoid these troubles for users, we need to be able to asses user preference directly via the applied tags themselves. Vig et al.[7] define tag-genome for tag relevance, an information space that indicates how strong each tag applies to each item. To calculate the relevance for each pair of tag t and movie m, Vig et al.[7] use several techniques such as *Rating-sim* (the correlation between item preference and tag preference), and *Tag-lsi-sim* (which helps to capture missing signals when tags were not applied to movies even if they are relevant to these movies; where lsi stands for *latent semantic indexing*, a mathematical technique that expresses the relationships between tags and movies they are applied). We think that Vig et al.'s approach has great potentials for the inference of user preference directly from the applied tags. However, Vig et al. only use tag relevance to navigate through movie collections, not to predict user preference.

In this paper, we aim to overcome the issues with Sen et al.'s approach by incorporating tag relevance as proposed by Vig et al. into the *prediction task*. To the best of our knowledge, no recommender system has ever done this. Furthermore, we use the mutual information framework in information theory [6] to avoid the constant thresholds applied to all users as in Sen et al. Next, we present how we use this framework to choose tags for each user.

2 Selecting Tags for Each User Using Mutual Information

According to Information Theory [6], the *mutual information* of two random variables is the amount of information one variable can tell about the other. Given two random variables X and Y, their mutual information is defined as follow:

$$I(X;Y) = \sum_{x\in X}\sum_{y\in Y} P(x,y)\log_2 \frac{P(x,y)}{P(x)P(y)} \tag{1}$$

Our objective is to find a subset of quality tags $T' \subset T$ that can explain the most about users' preferences via their rating history and tag relevance. Hence, given user u and tag t, the two random variables are the user's rating history R_u and the relevances L_t of tag t to all movies s/he rated.

In our approach, dealing with continuous values of tag relevance is less desirable, since user rating behavior is discrete, i.e. users give either 1 star, or 2 star etc. for any given movie. We bucket the relevance of tag t to movie m into 8 buckets with the width of 0.2.

Deriving from equation 1, the mutual information of tag t and the preference of user u is defined as:

$$I_{u,t}(R_u; L_t) = \sum_{r \in R_u} \sum_{l \in L_t} P_{u,t}(r, l) \log_2 \frac{P_{u,t}(r, l)}{P_u(r) P_{u,t}(l)} \tag{2}$$

where: $\forall$ rating $r \in R_u$, $P_u(r)$ is the probability that user u gave a rating r for all movies s/he rated; $\forall$ bucketed relevance $l \in L_t$, $P_{u,t}(l)$ is the probability that tag t has the bucketed relevance l for all movies that user u rated; $P_{u,t}(r, l)$ is the joint-probability of the two random variables, indicating the probability that user u gives a rating r for all movies that have tag t with bucketed relevance l.

The mutual information $I_{u,t}(R_u; L_t)$ ranges from 0 to the minimum of the expected values of the information contained in R_u and L_t. To select tags that have sufficient predictive power, we need to set a threshold, which is the minimum amount of information about the preference of user u (R_u) a tag can tell. Since user preference varies from user to user, the amount of information in their ratings also varies. Hence the threshold, if any, may be appropriate for one user, but not for another. To deal with this problem, we define *normalized mutual information*, $NI_{u,t}(R, L)$, the percentage of the information about user u's preference that tag t can tell:

$$NI_{u,t}(R, L) = \frac{I_{u,t}(R; L)}{H_u(R)} \times 100\% \tag{3}$$

Therefore, we set a threshold α for a tag to be selected if it can tell at least a certain percentage of user preference. In this paper, we consider five α thresholds from 10% to 50% with 10% increment. For example: at $\alpha = 50\%$, a selected tag t must explain at least 50% of the information about the preference of user u.

3 Case Study: Predicting Ratings in MovieLens

To evaluate our approach, we use the MovieLens dataset[1], a tagging-supported movie recommender system. MovieLens has been in continuous use since 1997. As of 10/19/2012, there are 209,844 unique users who provide approximate 19 million movie ratings , and 28,421 tags for 19,515 movies. In this study, we calculate the relevances of 1,633 quality tags for 9,063 movies. We only consider users who rated more than 99 movies.

3.1 Tag-Based Recommender Algorithms

Since each user has a different number of selected tags, we prefer to use some machine-learning methods that are robust in making predictions for an individual. Among many available methods, we choose to use the followings:

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

Linear Regression: The linear regression model expresses ratings as a linear combination of tag relevances. For each user u, and movie m associated with the selected set of quality tags T' by a vector of relevances $\boldsymbol{L_m}$, his rating for movie m can be represented as: $r_{u,m} = Intercept_u + \beta_u \boldsymbol{L_m} + \epsilon_{u,m}$, where β_u is the vector of coefficients learnt for user u, and $\epsilon_{u,m}$ is the residual error. We choose the linear approach for its simplicity. We build our linear regression model by using the lm command in R with 5 fold-cross-validation to learn the best β_u.

Support Vector Machine: We also study a non-linear approach to build our model. We choose support vector machine due to its robustness in gaining better generalization performance. We use Chang et al.'s svm library [2] with their suggested radial basis function kernel (RBF). We also optimize the SVM by using grid-search algorithm proposed by Chang et al [2] to find the best parameter for C and γ parameters of the kernel.

In order to evaluate our proposed algorithms, we compare our results with the results reported by Sen et al's [3] for their three algorithms *funk-svd*, *regress-tag* and *cosine-tag*. We choose these results for three reasons: 1) the *funk-svd* achieves the best results among tag-based as well as non-tag-based recommender systems, 2) the *regress-tag* achieves the best results, and 3) the *cosine-tag* achieves the worst results among the tag-based recommender systems. However, with the mutual information approach at high α thresholds mentioned above, we are required to have a large pool of users. Sen et al. study 1,315 users in their pruned set, which is not sufficient for our analysis. Therefore, we do not analyze on the same dataset as Sen et al. does. Furthermore, Sen et al. use different filtering methods to select tags and users. Hence, the comparisons we make in this paper are relative. To make our results comparable to that of Sen et al.'s work [3], we report the same metric they report: mean absolute error (MAE) and top-5. MAE is the average absolute difference between the predicted ratings and the observed ratings, reflecting the performance of an algorithm for the prediction task. Top-5 is the percentage of top predicted movies that users rate at certain stars or higher. Like Sen et al., we choose 4 stars as the cutoff point.

4 Results and Evaluation

Linear-regression algorithm with threshold α = k% is denoted **k-LR**. SVM algorithm with threshold α is denoted **k-SVM**. Optimized SVM algorithm with the threshold α = k% is denoted **opt-k-SVM**. Due to limited space, we only report results that are competitive to that of Sen et al.

Our Performance Compared to Sen et al. (2009) Results

Table 1 shows how our proposed algorithms performs against the results reported in Sen et al. [3]. In MAE term, our 50-LR algorithm achieves lower MAE (0.54) than all Sen et al.'s algorithms. Furthermore, the MAEs reported in Sen et al. (except that of *funk-svd*) are higher, and not in the 95% confidence interval of our 50-LR algorithm. The optimized SVM algorithms also achieve good performances (avg. MAE:0.64), approximately the same as *cosine-tag* algorithm

Table 1. The MAE and top-5 precision performances of our explicit algorithms. Lower MAE corresponds to better performance. Higher Top-5 precision corresponds to better performance. Sen et al.'s performances for Funk-svd: MAE=0.56, Top-5 precision=0.80; Regress-tag: MAE=0.58, Top-5 precision=0.83, Cosine-tag: MAE=0.64, Top-5 precision=0.62.[2]

	MAE			Top-5 Precision		
Algorithm	Avg.	95% CI	STD (σ)	Avg.	95% CI	STD (σ)
40-LR	0.79	[0.77,0.80]	0.93	0.70	[0.70,0.71]	0.25
50-LR	0.54	[0.53,0.56]	0.30	0.70	[0.69,0.70]	0.25
10-SVM	0.73	[0.72,0.73]	0.25	0.53	[0.53,0.53]	0.28
20-SVM	0.71	[0.70,0.71]	0.24	0.55	[0.54,0.55]	0.28
30-SVM	0.69	[0.68,0.69]	0.24	0.56	[0.55,0.56]	0.28
40-SVM	0.68	[0.67,0.68]	0.25	0.56	[0.56,0.56]	0.27
50-SVM	0.67	[0.66,0.67]	0.25	0.55	[0.55,0.56]	0.28
opt-30-SVM	0.64	[0.64,0.65]	0.21	0.61	[0.61,0.62]	0.27
opt-40-SVM	0.64	[0.64,0.65]	0.23	0.63	[0.62,0.63]	0.28
opt-50-SVM	0.63	[0.62,0.64]	0.25	0.61	[0.60,0.62]	0.28

(0.64). Some of our top-5-precision performances do not perform as good as Sen et al.'s algorithms do. We suspect that predicting a movie to be top-rated is harder than predicting across the rating-scale. We leave this for future work.

Mutual Information as a Way to Select Predictive Tags
As shown in table 1, although some of our algorithms do not achieve high performances, the others achieve very low MAEs, comparable to the results in Sen et al. This suggests that the mutual information approach is able to select the subsets of quality tags T' that have predictive power for each user. For example, at $\alpha = 30\%$, the top 10 tags that have predictive powers for at least 75% users are: {*masterpiece,good acting, great acting,imdb top 250, excellent script, interesting, great movie, drama, classic,oscar (best directing)* } respectively. On the other hand, the bottom 10 are {*tim robbins, united nations, vampire, william h. macy, matthew mcconaughey, jim carrey, michael keaton, jena malone,olympics,john turturro*}. The top-10 are tags that describe the quality of the movies. However, the bottom-10 are mostly only about the components of the movies. Furthermore, the number of selected tags is different for each user. For example, 230 users have 17 selected tags, but 400 users have only 8 selected tags. With the same number of selected tags, the tags are different, depending on the taste of each user. 5 tags {*weird, action packed, chase, fast paced, tense* }, which characterize action movies, are chosen for user 73782. The preference of user 129098 is described mostly by 5 tags {*chick flick, romantic, happy, ending, big budget, romance, girlie movie*}, which characterize emotional movies.

[2] The top-5 precision for cosine-tag was not reported in Sen et al.'s paper. We contacted the lead author to confirm our estimate from the reported figure.

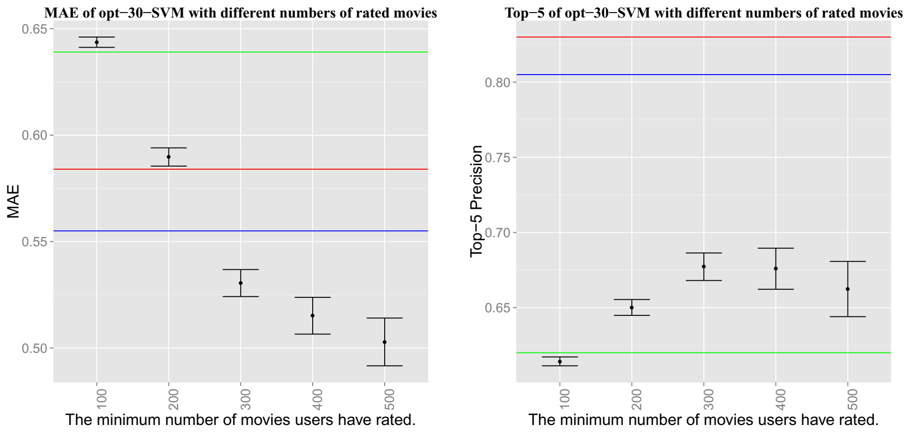

Fig. 1. Our performances compared to Sen et al.'s in term of MAE (left figure) and Top-5 precision (right figure) for the opt-30-SVM with the 95 % confidence intervals. Blue-line: Sen et al.'s MAE (0.56) and Top-5 precision (0.80) for Funk-svd. Red-line: Sen et al.'s MAE (0.58) and Top-5 precision (0.83) for Regress-tag. Green-line: Sen et al.'s MAE (0.64) and Top-5 precision(0.62) for Cosine-tag.

In addition, we observe that the higher the α threshold, the better the performance our SVM and linear regression (LR) can achieve. However, the higher α, the fewer users we can make predictions for. For example: with SVM algorithms, at $\alpha = 10\%$, we could make predictions for 48609 users. At $\alpha = 40\%$, only we could do it for 20705 users who rated at least 99 movies. As For 50-LR, due to the linear regression assumptions, we can only make prediction for 8400 users. Therefore, the higher the α, the wider the confident interval.

The Effect of the Number of Rated Movies on the Proposed Algorithms We also consider the number of movies a user has rated, because the probabilities in equation 2 vary with this number. We choose algorithm *opt-30-SVM* to analyze the effect due to two reasons. First, with this high α threshold, we still have a large pool of users (34309 users who rated at least 99 movies) to make our analysis reliable. Second, this threshold seems reasonable in real-life applications. As shown in figure 1 (right), as the minimum number of rated movies for our users increased, the MAEs *opt-30-SVM* decrease. For example, with the minimum of 100 movies, the achieved MAE is 0.64 (95% CI : [0.64,0.65]). With the minimum of 500 movies, the achieved MAE is 0.51 (95% CI of [0.49,0.51]). Nevertheless, like the α threshold, the higher the minimum number, the fewer users we can make predictions for.

5 Conclusion

In this paper, we introduce and evaluate two algorithms for recommender systems that take into account information from tags. Although, our approach has some limitations such as high MAEs or low top-5 precision, some of our proposed algorithms achieved better MAEs than, or at least equal to, the three well-known algorithms in Sen et al [3], with some default, and optimized configurations. We also present a methodology to select only tags that have predictive powers to improve the predictions for each user. However, our approach is sensitive to the number of rated movies and the α thresholds. In particular, for this paper, we only consider users who rated more than 99 movies. Furthermore, as α thresholds increase, the number of users we can make predictions for decreases. In future work, we hope to address these sensitivities.

Acknowledgement. This work has been supported by the National Science Foundation under grants IIS 08-08692, IIS 09-68483. The authors thank Daniel Kluver for information theory discussions. The authors also thank Duyen (Mary) Nguyen for proof-readings and suggestions.

References

1. Bell, R., Koren, Y.: Lessons from the Netflix prize challenge. ACM SIGKDD Explorations Newsletter Special issue on Visual Analytics, 75–79 (2007)
2. Chang, C., Lin, C.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2(3), 27:1–27:27 (2011)
3. Sen, S., Vig, J., Riedl, J.: Tagommenders: connecting users to items through tags. In: ACM International Conference on World Wide Web, WWW, pp. 221–230 (2009)
4. Sen, S., Vig, J., Riedl, J.: Learning to recognize valuable tags. In: Proceedings of the 14th International Conference on Intelligent user Interfaces, IUI, pp. 87–96 (2009)
5. Shirky, C.: Ontology is overrated (2005), `http://www.shirky.com/writings/ontology_overrated.html` (retrieved on December 28, 2012)
6. Shannon, C.: A mathematical theory of communication. ACM SIGMOBILE Mobile Computing and Communications Review 5(1), 3–5 (2001)
7. Vig, J., Sen, S., Riedl, J.: Encoding Community Knowledge to Support Novel Interaction. ACM Transactions on Interactive Intelligent Systems, TiiS 2(3) (2012)