A Maximum Entropy Web Recommendation System: Combining Collaborative and Content Features

Xin Jin, Yanzan Zhou, Bamshad Mobasher
Center for Web Intelligence
School of Computer Science, Telecommunication, and Information Systems
DePaul University, Chicago, Illinois, USA
{xjin,yzhou,mobasher}@cs.depaul.edu

ABSTRACT

Web users display their preferences implicitly by navigating through a sequence of pages or by providing numeric ratings to some items. Web usage mining techniques are used to extract useful knowledge about user interests from such data. The discovered user models are then used for a variety of applications such as personalized recommendations. Web site content or semantic features of objects provide another source of knowledge for deciphering users' needs or interests. We propose a novel Web recommendation system in which collaborative features such as navigation or rating data as well as the content features accessed by the users are seamlessly integrated under the maximum entropy principle. Both the discovered user patterns and the semantic relationships among Web objects are represented as sets of constraints that are integrated to fit the model. In the case of content features, we use a new approach based on Latent Dirichlet Allocation (LDA) to discover the hidden semantic relationships among items and derive constraints used in the model. Experiments on real Web site usage data sets show that this approach can achieve better recommendation accuracy, when compared to systems using only usage information. The integration of semantic information also allows for better interpretation of the generated recommendations.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data Mining*; I.2.6 [**Artificial Intelligence**]: Learning; I.5.1 [**Pattern Recognition**]: Models—*Statistical*

General Terms

Algorithms

Keywords

Maximum Entropy, Recommendation, Web Usage Mining, User Profiling

1. INTRODUCTION

Web recommendation systems have been widely used to help users locate information on the Web. In many cases, Web users' interests or preferences are not directly accessible; instead, they are implicitly captured during users' interaction with the Web site, such as navigating through a sequence of pages, or reflected through numeric ratings to items.

Many data mining and machine learning approaches have been used in building Web recommendation systems. Such systems, generally take the Web users' navigation or rating data as input, and applying data mining or machine learning approaches to discover usage patterns that represent aggregate user models. When a new user comes to the site, his/her activity will be matched against these patterns to find like-minded users and select potential interesting items as recommendations.

Typically, recommendation algorithms rely only on user information (collaborative features) such as navigational history or rating data, and additional information such as content features of the items, which may provide valuable source of complementary knowledge about user's activities, is usually ignored. By incorporating content information with a user's navigation or rating behavior, we may be able to gain a deeper understanding of her underlying interests. For instance, we may find an association pattern "page $A \Rightarrow$ page B" with high support and confidence in navigation data, or a pattern such as "movie C and movie D are always rated similarly" in rating data. The discovery of such patterns, by itself, does not explain the underlying reasons for their existence.

To address this issue considerable work has been done to enhance traditional recommendation systems by integrating data from other sources such as content attributes, linkage structure, and user demographics [13, 8, 14, 17, 5]. In these approaches, content information such as keywords or attributes of items, or user demographic data is collected as well as the usage or rating data. In order to make use of available data sources, different combination methods are tried to make recommendations more effective and interpretable. Generally, an integrated approach is prefered during the mining or model learning phase to avoid subjective or ad hoc ways of combining evidence. This is also one of our main motivations behind the maximum entropy recommendation system introduced in this paper.

Maximum entropy model is a powerful statistical model which has been widely applied in many domains such as statistical language learning [15], information retrieval [4] and text mining [10]. The goal of a maximum entropy model is to find a probability distribution which satisfies all the constraints in the observed data while maintaining maximum entropy. One of the advantages of such a model is that it enables the unification of information from multiple knowledge sources in one framework. Each knowledge source can be considered as a set of constraints in the model. From the intersection of all these constraints, a probability distribution with the highest entropy can be learned. Recently, we have seen applications of maximum entropy in Web recommendation systems [12, 18, 6]. In these systems, only statistics from users' navigation or rating data are used as features to train a maximum entropy model. Therefore, the advantage of integrating multiple knowledge sources is not fully exploited.

In this paper, we propose a novel Web recommendation system, in which users' navigational or rating data and the semantic content features associated with items are seamlessly integrated under the maximum entropy principle. First, we discover statistics from users' navigation data and use them as one set of constraints. Secondly, for content information, we use a new approach based on Latent Dirichlet Allocation (LDA) [1] to discover the hidden semantic relationships among visited or rated items and specify another set of constraints based on these item association patterns. The LDA model is a proven approach that can uncover the hidden relationships among co-occurring objects. An example of such hidden patterns are "topics" connecting documents and words [1], and the author-topic model in [9]. In our framework, the two sets of constraints are combined to fit the maximum entropy model, based on which we generate dynamic recommendations for active users.

2. RECOMMENDATIONS BASED ON MAXIMUM ENTROPY

In this section, we present our maximum entropy recommendation model and its algorithm. The scalability of this system is also discussed.

2.1 Notation

In this paper, we will use the following notations:

- $U = \{u_1, u_2, \ldots, u_m\}$: a set of m users.

- $T = \{t_1, t_2, \ldots, t_n\}$: a set of n pages/items.

- $A = \{a_1, a_2, \ldots, a_l\}$: a set of l attributes for pages/items within a site. In a content-intensive site, these attributes can be keywords from the site. While in a database-driven site, each web page/item represents an object which corresponds to a record in the underlying database. In this case, we extract the attributes of these records. Each page/item then can be represented as an attribute vector $t_j =< a_1^j, a_2^j, \cdots >$.

- For navigation data, each user is represented as a sequence of visited pages as: $u_i =< t_1^i, t_2^i, \cdots >$, where $t_k^i \in T$. We use binary representation here, assuming each page is either visited or not visited. (Our model can be easily extended to handle non-binary data as well.)

- For rating data, each user is represented as a set as: $u_i = \{< t_1^i, r_1^i >, < t_2^i, r_2^i >, \ldots\}$, where $t_k^i \in T$ and r_k^i takes ordinal values, such as from {1,2,3,4,5}.

- $H(u_i)$: recent navigation/rating history of user i, a subset of u_i. Since the most recent activities have greater influence on reflecting a user's present interests, we only use the last several pages to represent a user's history.

2.2 The Maximum Entropy Model

The system consists of two components. The offline component accepts constraints from the navigation/rating data and Web site content information, and estimates the model parameters. The online part reads an active user session and runs the recommendation algorithm to generate recommendations (a set of pages or rating predictions for unrated items) for the user.

We use the following approach to generate predictions or recommendations for active users. In case of navigation data, the conditional probability $Pr(t_d|H(u_i))$ of a page t_d being visited next given a user's recent navigational history $H(u_i)$. In case of rating data, the conditional probability $Pr(< t_d, r_d > |H(u_i))$ of an item t_d receiving rating r_d next given a user's recent rating history $H(u_i)$.

In our model, we use two sources of knowledge about Web users' navigational behavior, namely features based on item-level usage patterns, and features based on item content associations. Features will be presented in Section 3. Based on each feature f_s, we represent a constraint as:

$$\sum_{u_i} \sum_{t_d \in T} Pr(t_d|H(u_i)) f_s(H(u_i), t_d) = \sum_{u_i} f_s(H(u_i), D(H(u_i))) \tag{1}$$

where $D(H(u_i))$ denotes the page following u_i's history in the training data. Each constraint specifies that the expected value of each feature w.r.t. the model distribution should always equal its observation value in the training data. After we have defined a set of features, $F = \{f_1, f_2, \cdots, f_t\}$, and accordingly, generated constraints for each feature, it's guaranteed that, under all the constraints, a unique distribution exists with maximum entropy [3]. This distribution has the form:

$$Pr(t_d|H(u_i)) = \frac{exp(\sum_s \lambda_s f_s(H(u_i), t_d))}{Z(H(u_i))} \tag{2}$$

where $Z(H(u_i)) = \sum_{t_d \in T} Pr(t_d|H(u_i))$ is the normalization constant ensuring that the distribution sums to 1, and λs are the parameters needed to be estimated . Thus, each source of knowledge can be represented as features (and constraints) with associated weights. By using Equation 2, all knowledge sources (represented by various features) are taken into account to make predictions about users' next action given their past navigation.

There have been several algorithms which can be applied to estimate the λs. Here we use the Sequential Conditional Generalized Iterative Scaling (SCGIS) [2], which seems to be very efficient. To boost efficiency techniques such as dimensionality reduction using clustering [2, 11], efficient training algorithms [7], and automatic feature selection methods can be used.

2.3 Algorithm for Generating Recommendations

After we have estimated the parameters associated with each feature, we can use Equation 2 to compute the probability of any unvisited page t_i being visited next given certain user's history, and then pick the pages with highest probabilities as recommendations, i.e. we compute the conditional probability $Pr(t_i|u_a)$ given an active user u_a. Then we sort all pages based on the probabilities and pick the top N pages to get a recommendation set. The algorithm is as follows:

Input: an active user session u_a, parameters λs estimated from the training data.

Output: a list of N pages sorted by probability of being visited next given u_a.

1. Consider the last pages of the active user session, for each page t_i that does not appear in the active session, assume it is the next page to be visited, and evaluate all the features based on their definitions. (In this paper, we use bigram as usage features, so we only look at the last visited page. Apparently, high-order page transitions can be used as features as well, with higher computation complexity.)
2. Using Equation 2 to compute $Pr(t_i|u_a)$.
3. Sort all the pages in descending order of $Pr(t_i|u_a)$ and pick the top N pages to get a recommendation set.

In case of making a rating prediction for a target item t_d, we consider all rated items with their ratings as this user's history $H(u_i)$. Then we consider each possible rating (1-5) for the target item and evaluate all the features. We choose the rating r_d with the highest probability $Pr(< t_d, r_d > |H(u_i))$ as predicted rating for t_d.

3. IDENTIFICATION OF FEATURES

Under maximum entropy principle, each knowledge source is considered as a set of features and constraints imposed on the model. Thereafter, we will introduce our methods of identifying features from both usage and content information.

3.1 Features Based on Usage/Rating Information

Users' interests and preferences are implicitly embedded in their navigation/rating activities on some Web pages/items. Here we discover page/item associations from users' navigation/rating data as features.

- For navigation data, since the order of pages being visited conveys important information, we define features based first-order Markov page transitions. For each page transition $t_a \rightarrow t_b$ where $Pr(t_b|t_a) \geq \mu$ (μ is a pre-defined threshold), we define a feature as:

$$f_{t_a,t_b}(H(u_i), t_d) = \begin{cases} 1 & \text{if page } t_a \text{ is the last page} \\ & \text{in } H(u_i)\text{, and } t_b = t_d, \\ 0 & \text{otherwise} \end{cases}$$

- For rating data, generally the order of ratings is not important. We have to resort to other approaches to select highly correlated item pairs as features. Here we use item rating similarity to measure the associations between items. To offset the difference of rating scales, we adopt the *Adjusted Cosine Similarity* measure:

$$sim(t_a, t_b) = \frac{\sum_{k=1}^{m} (r_a^k - \overline{r_k}) \times (r_b^k - \overline{r_k})}{\sqrt{\sum_{k=1}^{m} (r_a^k - \overline{r_k})^2 \times \sum_{k=1}^{m} (r_b^k - \overline{r_k})^2}}$$

where, r_a^k denotes user k's rating on item a, and $\overline{r_k}$ stands for the average rating of user k.

For each item-pair $< t_a, t_b >$, where $sim(t_a, t_b) \geq \mu$ (μ is the user-specified threshold, used to filter out some less related item pairs), we define a feature as:

$$f_{t_a,r_a,t_b,r_b}(H(u_i), t_b, r_b) = \begin{cases} 1 & \text{if user rated item } t_a \\ & \text{with rating } r_a \text{ and} \\ & \text{item } t_b \text{ with rating } r_b, \\ 0 & \text{otherwise} \end{cases}$$

3.2 Features Based on Content Information

Web site content information provides another kind of information of users' interests and preferences. For example, in a university site, if a user visited several pages which contain frequent words like "admission", "application", we may guess this user was interested in applying for admissions into a program. In a movie site, high ratings for "Indiana Jones" and "Air Force One" may suggest that a user is a Harrison Ford's fan and enjoys Action-Adventure movies. We exploit such content features in our maximum entropy model. In text-oriented Web sites, content features may be terms or phrases extracted from pages. In sites with an underlying relational schema among object (e.g, products or movies), features can be semantic attributes associated with the objects.

Generally, considering all the attributes in a site is not feasible, due to the huge number of attributes and the irrelevancy and redundancy of many features. In this paper, we propose an attribute selection method based on the Latent Dirichlet Allocation (LDA) model [1], first proposed in the context of text mining, used to discover the hidden "topics" underlying a corpus. Here, we assume there exist a set of hidden variables underlying the item-attribute cooccurrence data. The hidden variables can be thought of as "classes" or "types" of these items, with each item being probabilistically associated with multiple classes. The complete probability model is:

$$\begin{aligned} \theta &\sim Dirichlet(\alpha) && (3) \\ z_i|\theta^{t_i} &\sim Discrete(\theta^{t_i}) && (4) \\ \phi &\sim Dirichlet(\beta) && (5) \\ a_j|z_i, \phi^{z_i} &\sim Discrete(\phi^{z_i}) && (6) \end{aligned}$$

Here, z represents a set of hidden "classes". θ^{t_i} denotes item t_i's association with multiple "classes" and ϕ^{z_i} specifies "class" z_i as a distribution over attributes. α and β are hyperparameters of the prior of θ and ϕ.

We use the Variational Bayes technique to estimate each item's association with multiple "classes" (θ), and the associations between "classes" and attributes (β). As shown in [1], these "classes" themselves can be used as derived attributes for text classification task. We observe that most items are only strongly associated with one "class" (based on θ vector for each user), so we assign each item to its dominant "class". Therefore, each "class" comprises of a set of items which are semantically similar. To better interpret these "classes", we can easily identify their prominent attributes based on β.

After assigning each item to a "class", we can define our features. In the case of navigation data, for each item pair $< t_a, t_b >$ where t_a and t_b both belong to the same "class" z, we define a feature function as:

$$f_{t_a,t_b,z}(H(u_i), t_d) = \begin{cases} 1 & \text{if page } t_a \text{ is the last page} \\ & \text{in } H(u_i),\ t_b = t_d, \\ 0 & \text{otherwise} \end{cases}$$

In the case of rating data, similarly, if t_a and t_b both belong to the same "class" z, we define a feature as:

$$f_{t_a,r_a,t_b,r_b,z}(H(u_i), t_b, r_b) = \begin{cases} 1 & \text{if user rated} \\ & \text{item } t_a \text{ with rating } r_a, \\ & \text{item } t_b \text{ with rating } r_b, \\ 0 & \text{otherwise} \end{cases}$$

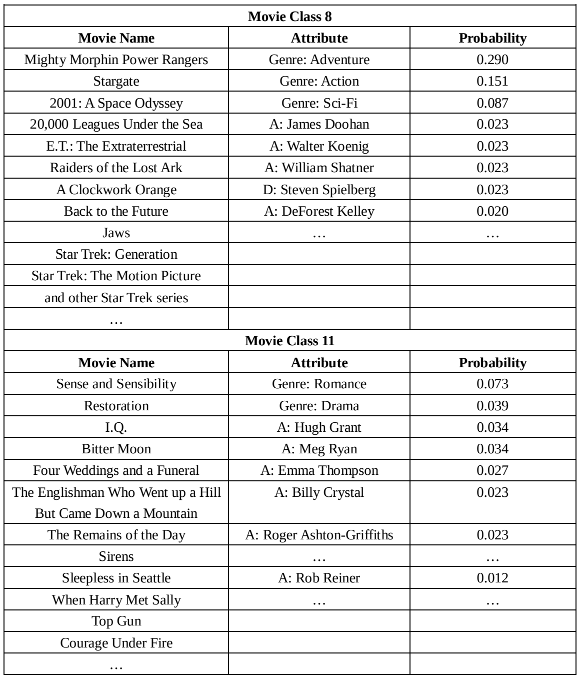
4. EXPERIMENTS

In this section, we report our experiments conducted on two data sets with different characteristics. One data set is usage data associated with a real estate Web site (together with the semantic attributes associated with the real estate properties). The second data set is data set is the EachMovie rating data often used in the context of collaborative filtering (together with content attributes associated with movies).

4.1 Data Sets and Evaluation Metrics

The primary function of the Real Estate site is to allow prospective buyers to visit various pages and information related to some 300 residential properties. The portion of the Web usage data during the period of analysis contained approximately 24,000 user sessions from 3,800 unique users. The data was filtered to limit the final data set to those users that had visited at least 3 properties. We also extracted the content attributes related to each property including price, number of bedrooms, size, school district, etc. We refer to this data set as the "Realty data." The navigation data is randomly divided into 10 training and test sets to be used for cross-validation.

Our evaluation metric for this data set is called *Hit Ratio* and is used in the context of top-N recommendation framework: for each user session in the test set, we take the first K pages as a representation of an active session to generate a top-N recommendations. We then compare the recommendations with page $K + 1$ in the test session, with a match being considered a *hit*. We define the Hit Ratio as the total number of hits divided by the total number of user sessions in the test set. Note that the Hit Ratio increases as the value of N increases. Thus, in our experiments, we pay special attention to smaller number recommendations (between 1 and 10) that result in good hit ratios.



Movie Class 8		
Movie Name	**Attribute**	**Probability**
Mighty Morphin Power Rangers	Genre: Adventure	0.290
Stargate	Genre: Action	0.151
2001: A Space Odyssey	Genre: Sci-Fi	0.087
20,000 Leagues Under the Sea	A: James Doohan	0.023
E.T.: The Extraterrestrial	A: Walter Koenig	0.023
Raiders of the Lost Ark	A: William Shatner	0.023
A Clockwork Orange	D: Steven Spielberg	0.023
Back to the Future	A: DeForest Kelley	0.020
Jaws	…	…
Star Trek: Generation		
Star Trek: The Motion Picture		
and other Star Trek series		
…		
Movie Class 11		
Movie Name	**Attribute**	**Probability**
Sense and Sensibility	Genre: Romance	0.073
Restoration	Genre: Drama	0.039
I.Q.	A: Hugh Grant	0.034
Bitter Moon	A: Meg Ryan	0.034
Four Weddings and a Funeral	A: Emma Thompson	0.027
The Englishman Who Went up a Hill But Came Down a Mountain	A: Billy Crystal	0.023
The Remains of the Day	A: Roger Ashton-Griffiths	0.023
Sirens	…	…
Sleepless in Seattle	A: Rob Reiner	0.012
When Harry Met Sally	…	…
Top Gun		
Courage Under Fire		
…		

Figure 1: Examples of movie classes and their top attributes

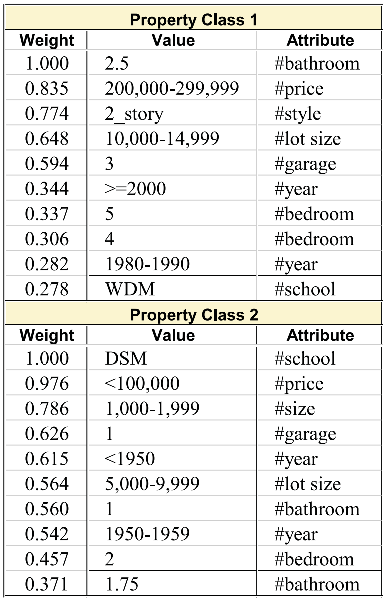
The Movie data set contains a total of 2,811,983 numeric ratings (rating scale 1-6) of 72916 users for 1628 different movies. We extracted movie content information from the Internet Movie Database (http://www.imdb.com), including Movie's genre, director, cast, etc. We kept 900 movies that has complete content information, and chose 5000 users who had rated at least 20 movies. The evaluation metric for this data set is the standard *Mean Absolute Error* (MAE). Specifically, for each user-movie pair in test set, we make a prediction for the target movie and compute the absolute deviation between the actual rating and the predicted rating. MAE is defined as the sum of all the deviations divided by the total number of predictions. Note that lower MAE values represent higher recommendation accuracy.

4.2 Examples of Item Groups

As stated in Section 3, we use LDA to identify item groups from content information. Here we run LDA on property-attribute matrix and movie-attribute matrix respectively and assign each item to one dominant class. To illustrate these item groups, we list the top attributes (based on ϕ) as well as the items of each class.

Figure 1 depicts two movie classes from 20 classes we generate. For each class, we show the movies, top attributes and corresponding probability (ϕ) associated with the class. We can see Class 8 consists of mostly Sci-Fi movies, including the Star Trek series. Naturally attributes like "Adventure", "Action", "Sci-Fi" have dominant probabilities. Actors from Star Trek series and director Steven Spielberg also appear on top of this list. Movie Class 11 seems to be a group of romance comedy movies, acted by Hugh Grant and Meg Ryan. As we can see attributes "Romance", "Drama", "Hugh Grant" and "Meg Ryan" have highest probabilities.

In this example, we see that a user has given high rat-



Property Class 1		
Weight	Value	Attribute
1.000	2.5	#bathroom
0.835	200,000-299,999	#price
0.774	2_story	#style
0.648	10,000-14,999	#lot size
0.594	3	#garage
0.344	>=2000	#year
0.337	5	#bedroom
0.306	4	#bedroom
0.282	1980-1990	#year
0.278	WDM	#school

Property Class 2		
Weight	Value	Attribute
1.000	DSM	#school
0.976	<100,000	#price
0.786	1,000-1,999	#size
0.626	1	#garage
0.615	<1950	#year
0.564	5,000-9,999	#lot size
0.560	1	#bathroom
0.542	1950-1959	#year
0.457	2	#bedroom
0.371	1.75	#bathroom

Figure 2: Examples of real estate property classes and their top attributes

ing (6) to "Sense and Sensibility", and the system correctly predicts the user's rating (6) for "Four Weddings and a Funeral". From the training data, we find that these two movies have relatively high rating similarity (0.65). At the same time, the prediction algorithm notices that the content features involving them also tend to have high weights, which indicate they belong to the same movie group and share some common attributes.

Similarly, Figure 2 depicts the top attributes associates with two of the discovered classes among real estate properties. In each case, the (normalized) weight indicating the importance of the attribute in the class is shown on the left. It can be seen that Class 1 represents relatively new 2-story properties in $200K-$300K range with 4-5 bedrooms all of which are located in the WDM school district. On the other hand, Class 2 represents much smaller and older properties with 1-2 bedrooms. The most prominent attribute in this group is the DSM school district.

4.3 Quantitative Evaluation

For "Realty data", to exploit the ordering information in users' navigation data, we built another recommendation system based on the standard first-order Markov model to predict and recommend which page to visit next. The Markov-based system models each page as a state in a state diagram with each state transition labeled by the conditional probabilities estimated from the actual navigational data from the server log data. It generates recommendations based on the state transition probability of the last page in the active user session.

Figure 3 depicts the comparison of the *Hit Ratio* measures for these two recommender systems in "Realty data". The experiments show that the maximum entropy recommendation system has a clear overall advantage in terms of accuracy over the first-order Markov recommendation system on this data set.

For the movie data, we implement a standard item-based collaborative filtering algorithm (Item-based CF) [16] to generate predictions for users in test data for comparison. For each user we took 50% of the data as training data, and the rest as test data. For Item-based CF, we tried different neighbor sizes and used the best result achieved (with 40 neighbors). For maximum entropy model, we ran LDA to generate 20 movie groups and define features as in Section 3.

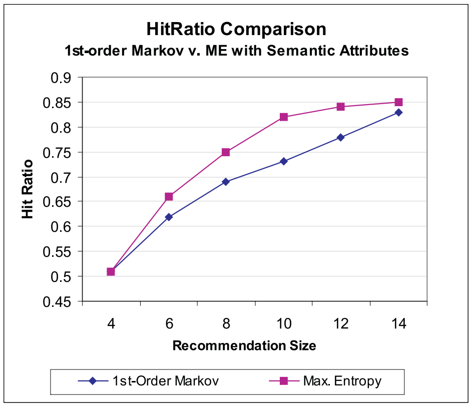
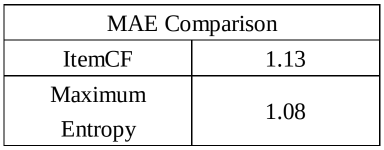


Figure 3: Recommendation accuracy: maximum entropy model vs. Markov model



MAE Comparison	
ItemCF	1.13
Maximum Entropy	1.08

Figure 4: Recommendation accuracy: maximum entropy model vs. Item-Based CF

The results are shown in Figure 4. By actually looking at the cases where maximum entropy system makes better prediction, we find that in most cases, item=based CF algorithm can not find similar item neighbors (highest similarity is only about 0.50-0.60), which makes the predictions based on item neighbors less accurate.

One of the problems associated with some traditional recommendation algorithms emanated from the sparsity of data sets to which they are applied. Data sparsity has a negative impact on the accuracy and predictability of recommendations. This is one area in which, we believe, the integration of semantic knowledge with ratings data can provide significant advantage. Here we created multiple training/test data sets in which the proportion of the training data to the complete ratings data set was changed from 90% to 10% (For each user, we randomly select certain percentage of ratings as training data and the rest as test data). These proportions have a direct correspondence with the level of sparsity in the ratings data.

Figure 5 indicates that the advantage of our system over the item-based CF system tends to be more distinctive when the data gets sparser. One possible explanation is that when the training data gets sparser, we are less likely to find movie neighbors with very high rating similarity and make accurate predictions based on neighbor movies' ratings. At this time, for maximum entropy recommendation system, some features based on movie rating similarity also get lower weights,

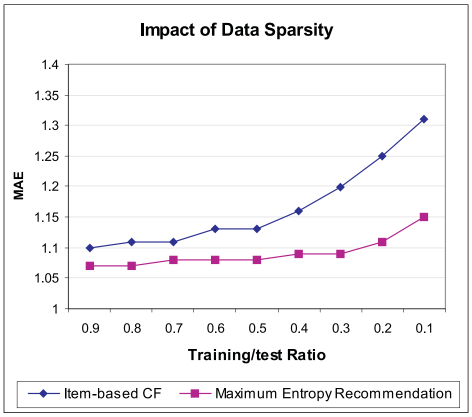


Figure 5: Impact of Data Sparsity on Recommendations

but features based on movie content similarity will be less affected and still contribute remarkable weights to make accurate predictions.

5. CONCLUSION

In this paper, we have proposed a novel Web recommendation system in which users' navigation or rating data as well as the content features accessed by the users are seamlessly integrated under the maximum entropy principle. In the case of content information, we use a new approach based on Latent Dirichlet Allocation (LDA) to discover the hidden semantic relationships among items. The resulting models are used to generate dynamic recommendations for active users. Experiments show that this approach can achieve better recommendation accuracy with small number of recommendations. Furthermore, the proposed framework provides reasonable accuracy in predictions in the face of high data sparsity.

6. REFERENCES

[1] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[2] J. Goodman. Sequential conditional generalized iterative scaling. In *Proceedings of NAACL-2002*, 2002.

[3] F. Jelinek. *Statistical Methods for Speech Recognition.* MIT Press, MA, 1998.

[4] J. Jeon and R. Manmatha. Using maximum entropy for automatic image annotation. In *Proceedings of the International Conference on Image and Video Retrieval (CIVR-2004)*, 2004.

[5] X. Jin, Y. Zhou, and B. Mobasher. A unified approach to personalization based on probabilistic latent semantic models of web usage and content. In *Proceedings of the AAAI 2004 Workshop on Semantic Web Personalization (SWP'04)*, San Jose, CA, 2004.

[6] X. Jin, Y. Zhou, and B. Mobasher. Task-oriented web user modeling for recommendation. In *Proceedings of the 10th International Conference on User Modeling (UM'05)*, UK, April 2005.

[7] R. Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the Sixth Conference on Natural Language Learning(2002)*, 2002.

[8] B. Mobasher, H. Dai, T. Luo, Y. Sun, and J. Zhu. Integrating web usage and content mining for more effective personalization. In *E-Commerce and Web Technologies: Proceedings of the EC-WEB 2000 Conference*, Lecture Notes in Computer Science (LNCS) 1875, pages 165–176. Springer, September 2000.

[9] M.Steyvers, P. Smyth, M. Rosen-Zvi, and T. Griffiths. Probabilistic author-topic models for information discovery. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, 2004.

[10] K. Nigram, J. Lafferty, and A. McCallum. Using maximum entropy for text classification. In *Proceedings of IJCAI-1999*, 1999.

[11] D. Pavlov, E. Manavoglu, D. Pennock, and C. Giles. Collaborative filtering with maximum entropy. *IEEE Intelligent Systems, Special Issue on Mining the Web Actionable Knowledge*, 2004.

[12] D. Pavlov and D. Pennock. A maximum entropy approach to collaborative filtering in dynamic, sparse, high-dimensional domains. In *Proceedings of Neural Information Processing Systems(2002)*, 2002.

[13] M. Pazzani. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13(5-6):393–408, 1999.

[14] A. Popescul, L. Ungar, D. Pennock, and S. Lawrence. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *Proceedings of 17th UAI*, Seattle, WA, 2001.

[15] R. Rosenfeld. Adaptive statistical language modeling: A maximum entropy approach. Phd dissertation, CMU, 1994.

[16] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International WWW Conference*, Hong Kong, May 2001.

[17] K. Yu, A. Schwaighofer, V. Tresp, W. Ma, and H. Zhang. Collaborative ensembling learning: Combining collaborative and content-based information filtering. In *Proceedings of 19th UAI*, 2003.

[18] C. Zitnick and T. Kanade. Maximum entropy for collaborative filtering. In *Proceedings of 20th International Conference on Uncertainty in Artificial Intelligence (UAI'04)*, Banff, Canada, July 2004.