Learning to Recommend Related Entities to Search Users

Bin Bi*
UCLA
bbi@cs.ucla.edu

Hao Ma
Microsoft Research
haoma@microsoft.com

Bo-June (Paul) Hsu
Microsoft Research
paulhsu@microsoft.com

Wei Chu
Microsoft
chu.wei@microsoft.com

Kuansan Wang
Microsoft Research
kuansan.wang@microsoft.com

Junghoo Cho
UCLA
cho@cs.ucla.edu

ABSTRACT

Over the past few years, major web search engines have introduced *knowledge bases* to offer popular facts about people, places, and things on the entity pane next to regular search results. In addition to information about the entity searched by the user, the entity pane often provides a ranked list of related entities. To keep users engaged, it is important to develop a recommendation model that tailors the related entities to individual user interests.

We propose a probabilistic Three-way Entity Model (TEM) that provides personalized recommendation of related entities using three data sources: *knowledge base*, *search click log*, and *entity pane log*. Specifically, TEM is capable of extracting hidden structures and capturing underlying correlations among *users*, *main entities*, and *related entities*. Moreover, the TEM model can also exploit the click signals derived from the entity pane log. We further provide an inference technique to learn the parameters in TEM, and propose a principled preference learning method specifically designed for ranking related entities. Extensive experiments with two real-world datasets show that TEM with our probabilistic framework significantly outperforms a state of the art baseline, confirming the effectiveness of TEM and our probabilistic framework in related entity recommendation.

Categories and Subject Descriptors: H.3.3 [Information Search and Retrieval]

General Terms: Algorithm, Experimentation

1. INTRODUCTION

Traditionally, web search engines have led users toward web pages chosen by lexical matches against the search string. However with the introduction of *knowledge bases* over the past few years, commercial search engines are moving towards retrieval based on a semantic understanding of the user query. The knowledge base is being used to provide popular facts about people, places, and things alongside traditional search results. It allows search to evolve from returning pages that match query terms to finding *entities* that the words describe.

A knowledge base is a centralized repository of content about entities, their attributes and mutual relationships. Well-known examples of knowledge bases include Freebase, YAGO, Microsoft Satori, and Google Knowledge Graph. For instance, Freebase consists of a large set of metadata about movies, music, books, well-known people, and things. A subset of the entities and their relations in Freebase is depicted in Figure 1. It includes entities corresponding to four people, two movies and their genres. The links between the entities represent their relationships, such as "*Adam McKay* is the director of *Anchorman*" and "*Kristen Wiig* is an actor appearing in *Anchorman 2*". In this example, "director", "actor" and "genre" are attributes of the entity *Anchorman*.

With the introduction of a knowledge base, a web search engine enables users to search for things – movies, celebrities, landmarks and more – and instantly get rich information relevant to the queries. Figure 2 shows an example search result for the query "pacific rim" together with its entity pane provided by a commercial search engine. The search engine recognizes that "pacific rim" is the title of a movie corresponding to an entity in the knowledge base. We refer to the entity a user searches for as the *main entity*. An entity pane that presents information about the main entity shows up to the right of the regular search results. On the entity pane, in addition to the description of the movie *Pacific Rim*, a list of movies related to *Pacific Rim* is also visually presented below. We refer to an entity related to the search as a *related entity*. The provided related entities allow users to quickly access other relevant entities and offer the ability to explore more information within the same search session. In order to keep users engaged, it is important to develop a recommendation model that generates related entities closely matched with their interests.

Currently, major search engines recommend related entities based on their similarities to the main entity that the user searched for. There are various measures of the similarity between a main entity and an entity to recommend. A common measure is the frequency of the two entities being co-clicked in the same session across all search users. A related entity is recommended if and only if it is frequently co-clicked with the main entity. This co-click based approach essentially maximizes the likelihood that people agree on the relatedness irrespective of any individual user. Such a global

*This work was done when the first author was on a summer internship at Microsoft Research.

WSDM'15, February 2–6, 2015, Shanghai, China.

http://dx.doi.org/10.1145/2684822.2685304.

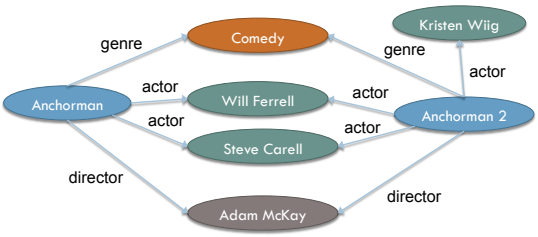

Figure 1: Example of the entities and their relations taken from Freebase

recommendation method brings the same list of related entities to every user who searches for the same main entity, as user-specific information is completely ignored. But the same recommendation cannot satisfy users with distinct interests. For example, given a movie as the main entity, one user may be interested in viewing the other movie entities with the same director, while another user may want to view the movie entities from the same genre.

To the best of our knowledge, no work has been done on developing a recommendation model for a search engine to tailor related entities to an individual user's unique taste and preference. To personalize recommendations, we need to build user-specific profiles from their interactions with the search engine. In this paper, the users' interactions are collected in the *search click log* and the *entity pane log*. The search click log stores history of user clicks on URLs, while the entity pane log stores clicks on the entity pane. In this work, we aim to build a probabilistic recommendation model that can customize the suggested entities, which are related to a given main entity, based on the user's past history stored in the usage logs.

Despite considerable research on the *search click log* over the last decade, little is known about the emerging *entity pane log*. This work also represents the first study exploiting the entity pane's implicit user feedback for entity recommendation. Our empirical studies find that the entity pane click-through rates (CTR) play important roles in enhancing recommendation quality of related entities. Therefore, we include these strong CTR signals in the recommendation model.

In addition to CTRs, our recommendation model involves three important dimensions: *user*, *main entity*, and *related entity*. Without the user dimension, the model would degenerate to a global recommendation method which fails to personalize suggested entities, as discussed above. On the other hand, if recommendations were based purely on the user dimension, while totally ignoring main entities, then the suggested entities would be utterly unrelated to the searches. The interactive feedback in the usage logs reveals the three-way correlations among these three dimensions. The recommendation model aims to discover and exploit their ternary relationships. We refer to our probabilistic recommendation model as Three-way Entity Model, abbreviated as TEM.

To determine the parameters in TEM, we propose learning their optimal values from a training set of observations constructed from the entity pane log. As mentioned above, this log contains user feedback on the relatedness of recommended entities. Positive observations can be readily derived from click feedback by interpreting a user click as a vote in favor of relatedness. Nevertheless, it is nontrivial to derive negative observations, since a non-click may not indicate the absence of relatedness. We propose a principled

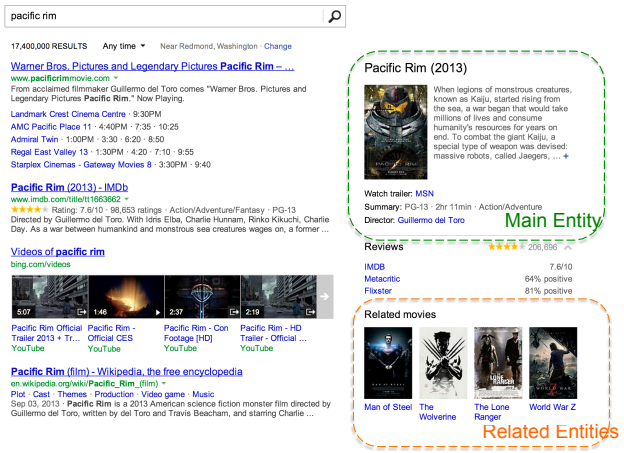

Figure 2: Example of search results with the entity pane taken from a commercial web search engine

solution to this issue, specialized for the problem of ranking related entities.

The major contributions of our work are summarized as follows:

1. This paper presents the first solution – the probabilistic model TEM– for a search engine to personalize its recommendation of related entities. The recommended entities are customized to be not only related to the given main entity, but also tailored to the user's interest and preference.

2. The TEM model leverages three data sources: *knowledge base*, *search click log*, and *entity pane log*. This is the first work to utilize the entity pane log to recommend related entities. Specifically, the CTRs derived from the entity pane log turn out to be strong signals for entity recommendation.

3. The TEM model uncovers the underlying three-way relationships among *user*, *main entity*, and *related entity*. Jointly modeling all three dimensions prevents TEM from making static or irrelevant recommendations. An inference technique is introduced to learn the parameters of TEM.

4. We propose a principled method for training set construction to work around the problem of missing negative samples. The proposed method is specifically designed for ranking related entities.

5. We conducted extensive experiments with two real-world datasets of different domains collected from a commercial web search engine. The experimental results demonstrate that TEM with our probabilistic framework significantly outperforms the state of the art used by a commercial search engine. It confirms the effectiveness of TEM and our probabilistic framework in entity recommendation and the efficacy of personalization.

The rest of this paper is organized as follows. In Section 2, we describe the prior work related to ours. The problem statement is given in Section 3. Section 4 introduces our TEM model. In Section 5, we present the experimental results. Finally, we conclude the paper in Section 6.

User ID	Time	Main entity	Related entity	Rank	Click
32	7/9/2013 10:32:26	Pacific Rim	Man of Steel	1	0
32	7/9/2013 10:32:26	Pacific Rim	The Wolverine	2	0
32	7/9/2013 10:32:26	Pacific Rim	The Lone Ranger	3	1
32	7/9/2013 10:32:26	Pacific Rim	World War Z	4	0
498	6/16/2013 15:16:41	Leonardo DiCaprio	Kate Winslet	1	0
498	6/16/2013 15:16:41	Leonardo DiCaprio	Baz Luhrmann	2	0
498	6/16/2013 15:16:41	Leonardo DiCaprio	Johnny Depp	3	1

Figure 3: Sample records taken from an entity pane log

2. RELATED WORK

A research topic related to our work is personalized web services, including web search and entity/news recommendation, although the tasks are quite different. Micarelli et al. [12] provided a summary of research works on this topic. Personalized search exploits user search histories to deliver more relevant results than those provided by traditional search engines [15]. Unlike the task addressed in this paper, Blanco et al. [4] worked on a fundamentally different entity recommendation task. The goal of their work was to recommend possible future queries related to the user's current search query based on a knowledge base. In their paper, the future queries were referred as to related entities, as opposed to the related entities in our context. Chu and Park [6] proposed a feature-based bilinear regression framework for personalized recommendation on news content. This approach greatly alleviated the cold-start issue of recommending for new users by leveraging interest patterns in user profiles recognized from regression over historical interactive feedback. Sun et al. [16] introduced CubeSVD to perform three-way data analysis for personalized search. Similar to personalized search, our work exploits prior user actions to model their interests for personalized recommendation on related entities.

Over recent decades, a few studies have been conducted on three-way data analysis. Acar and Yener [1] gave an overview of multiway models, algorithms as well as their applications in diverse disciplines. These studies commonly represented observational data as a third-order tensor, which is a higher-order generalization of a vector and a matrix. A three-way model was then constructed for extracting hidden structures and capturing underlying correlations between variables in the third-order tensor. A well-known three-way model, called Tucker3, was introduced by Tucker [17, 7]. It is an extension of singular value decomposition to third-order tensors. Tucker3 has been successful in many applications [16, 18].

Three-way data analysis has been widely performed in the context of multiverse recommendation. Karatzoglou et al. [10] introduced a collaborative filtering method based on third-order tensor decomposition to provide context-aware recommendations. Rendle and Schmidt-Thieme [14] used the tensor decomposition technique in recommending to users tags for annotating specific items in social tagging systems. Despite its success in recommender systems, tensor decomposition does not apply to related entity recommendation. In particular, tensor decomposition suffers from the cold-start problem, as it represents each object in the system with a unique ID. Given the knowledge base and the usage logs, tensor decomposition cannot utilize the valuable information derived from the various nature of data sources. In order to do so, we developed TEM, a new probabilistic model for three-way data analysis.

Table 1: Notations used throughout this paper

Notation	Description
X	Feature matrix for users
Y	Feature matrix for main entities
Z	Feature matrix for related entities
u	User identity
m	Main entity
r	Related entity
$\mathbf{x}_u$	Feature vector for user u
$\mathbf{y}_m$	Feature vector for main entity m
$\mathbf{z}_r$	Feature vector for related entity r
U	Number of unique users
M	Number of main entities
R	Number of related entities
I	Number of features for each user
J	Number of features for each main entity
K	Number of features for each related entity
Θ	Model parameter
η, β	Weight coefficients
o	Preference relation

3. PROBLEM STATEMENT

In a nutshell, the objective of this paper is to recommend to the user a ranked list of entities relevant to the main entity by leveraging three pieces of information: *knowledge base*, *search click log*, and *entity pane log*.

Figure 3 presents a few sample records from the entity pane click log of a real search engine. Each row represents an *instance* indicating whether user u clicked related entity r given main entity m. There are two *page impressions* in the table, each of which indicates the list of related entities recommended for a given (u, m) pair. The *Rank* column gives the rank of each related entity in recommendation lists, and the *Click* column indicates whether related entities were clicked or not (1 for click, 0 for no click).

For notational convenience, let U denote the total number of unique users in the log, M denote the total number of main entities, and R denote the total number of related entities. The notations used throughout this paper are given in Table 1. Some of the notations will be explained in later sections.

As discussed above, a click event is naturally associated with three salient dimensions: $User \times Main\ entity \times Related\ entity$.

[User dimension]

The user dimension targets user interest patterns, building search profiles by logging user interactions with the search engine. In this paper, a user profile maps a user to a vector of entities and attributes representing the user's interests. In order to model user interest as accurately as possible, we collect click history from two sources: *search click log* and *entity pane log*. The former records user click history on URLs, while the latter reflects user click history on entities.

Since the *search click log* reports user clicks on URLs, but we are looking for user interest in entities, we need a mapping from URLs to entities. Fortunately, with the help of the open source Freebase[1] knowledge base, it is easy to map users to the entities they are interested in. An illustration is shown in Figure 4.

Each entity in Freebase is linked to some URLs that are related to this entity. For example, for the movie *Avatar*[2], by utilizing the relationships "/common/topic/official_website"

[1] `http://www.freebase.com/`
[2] `http://www.freebase.com/m/0bth54`

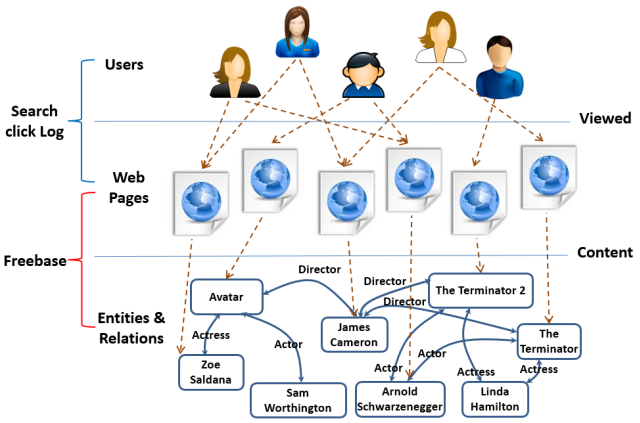

Figure 4: Illustration on joining search click log with Freebase knowledge base

and "/common/topic/topic_equivalent_webpage", we can obtain this movie's official site, IMDb pages as well as Wikipedia pages, etc. Moreover, other information related to this movie is available to us, including actors, directors, genres, producers, etc. With the help of this data, user-clicked URLs in *search click log* can be mapped to corresponding entities in Freebase, as demonstrated in Figure 4. The attributes of these entities can also be obtained from the Freebase knowledge base.

For the *entity pane log*, as shown in Figure 3, the clicked entities are already known, so we can simply extract corresponding attributes from Freebase to represent user interests. By combining the above signals, entities and their attributes can be used to model users in a naturally way. For instance, a group of users may often view entities about action movies, while another group may prefer basketball players. If users are modeled appropriately through their usage patterns, these preferences and interests should help the search engine recommend related entities more accurately. Some sample features we extract for users are shown in Table 3 in Section 5.

Formally, each user is represented as a vector of features, denoted by $\mathbf{x}$, where $\mathbf{x} \in \mathbb{R}^I$ and I is the number of user features.

[Main entity]

In nature, the main entity reflects the search user's current search interest. In addition to user profiling by modeling his or her interest pattern based on the usage log, it is important to capture a user's current search intent expressed by the main entity, which provides valuable context. Ignoring main entities will compromise the performance of a recommendation model. In particular, if related entities are obtained based purely on the user's past preferences, while neglecting to model his or her current interest, then the recommended entities will be completely independent of what the user is searching for, leading to dissatisfaction.

The feature space for main entities is spanned by their attributes extracted from the knowledge base. Each main entity is represented as a vector of features, denoted by $\mathbf{y}$, where $\mathbf{y} \in \mathbb{R}^J$ and J is the dimensionality of the feature space for main entities.

[Related entity]

A user may click a related entity when it is aligned with both the user's interest pattern and current intent. User clicks on related entities are the interactive feedback used to relate patterns in user features to main entities. For example, suppose there is a fan of the film director *Steven Spielberg*. (Such a user can be identified from his or her past usage pattern.) Given a movie as a main entity, a good recommender should recommend the other movies related not just to the given entity, but also directed by *Steven Spielberg*, instead of recommending movies related in other ways, such as sharing the same actors.

Each related entity is represented as a column vector of features, denoted by $\mathbf{z}$, where $\mathbf{z} \in \mathbb{R}^K$ and K is the number of features for each related entity.

As we have argued, all three dimensions, $User \times Main\ entity \times Related\ entity$, can improve entity recommendation, which motivates our joint modeling of these factors. The joint model is intended to capture structural dependencies of the three dimensions, revealing the underlying ternary relations.

Problem Statement *Given the feature representations for users* $\mathbf{x}$*, main entities* $\mathbf{y}$ *and related entities* $\mathbf{z}$*, we aim to develop a recommendation model that uncovers the three-way correlations among them to recommend a ranked list of entities related to a given main entity for any user.*

4. THREE-WAY ENTITY MODEL

In this section, we present a three-way probabilistic model, TEM, designed to uncover the pattern correlations among $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ for recommending related entities. More specifically, we first define a real-valued function $\Psi_{umr}(\Theta)$ of the model parameter Θ which captures the ternary relationship among the three dimensions. A likelihood function is then employed to relate the values of $\Psi_{umr}(\Theta)$ to observed actions on related entities. Finally, the parameter Θ is obtained by performing inference on TEM. The effect of the three-way interactions will be analyzed in this section.

4.1 Trilinear function

To jointly model users, main entities, and related entities, we define a trilinear function Φ_{umr} of $\mathbf{x}_u$, $\mathbf{y}_m$, and $\mathbf{z}_r$ as follows:

$$\Phi_{umr}(\eta) = \sum_{i=0}^{I}\sum_{j=0}^{J}\sum_{k=0}^{K} \eta_{ijk} \cdot x_{ui} \cdot y_{mj} \cdot z_{rk}, \qquad (1)$$

where $\mathbf{x}_u$ denotes the feature vector for user u, $\mathbf{y}_m$ denotes the feature vector for main entity m, and $\mathbf{z}_r$ denotes the feature vector for related entity r. x_{ui} is the i-th feature of $\mathbf{x}_u$, y_{mj} is the j-th feature of $\mathbf{y}_m$, and z_{rk} is the k-th feature of $\mathbf{z}_r$. η consists of a set of weight coefficients, which is introduced to capture the associations among the three objects $\mathbf{x}_u$, $\mathbf{y}_m$, and $\mathbf{z}_r$. The weight η_{ijk} quantifies the affinity of three features x_{ui}, y_{mj}, and z_{rk}. Note that η can be represented as a third-order tensor, where the value of each entry η_{ijk} will be learned from historical logs.

In order for the trilinear function to capture the pairwise associations between the three dimensions, we prepend a 1 at the beginning of each feature vector. As a result, the users, main entities, and related entities are represented as:

$$\begin{aligned} \mathbf{x}_u &= [1, x_{u1}, x_{u2}, \cdots, x_{uI}]^T, \\ \mathbf{y}_m &= [1, y_{m1}, y_{m2}, \cdots, y_{mJ}]^T, \\ \mathbf{z}_r &= [1, z_{r1}, z_{r2}, \cdots, z_{rK}]^T. \end{aligned}$$

Notice that when there is a large number of features for each dimension, $\eta \in \mathbb{R}^{(I+1)\times(J+1)\times(K+1)}$ becomes a huge

tensor for which inference is intractable. To overcome this problem, we need to reduce the dimensionality of each feature vector. Given massive training data, we resort to random projections [9] for dimensionality reduction. Random projections essentially project each feature space onto a random lower-dimensional subspace, which yield results comparable to conventional dimensionality reduction approaches such as Principal Component Analysis (PCA). However, random projections are significantly less computationally expensive than PCA. We study the effect of random projections on entity recommendation in the experiment section.

4.2 CTR incorporation

The three-way associations, which are systematically modeled by the trilinear function $\Phi_{umr}(\eta)$, contribute an important indicator to entity recommendation, especially for the rare/new entities for which we have zero or insufficient click data. To further enhance the recommendation quality of popular entities, we derive CTR features from the interactive feedback collected in the entity pane log. The CTRs have been shown to be strong signals for various recommendation tasks [11, 8]. CTR is defined as the ratio of the number of clicks on a certain related entity and the number of page impressions in which the related entity is presented. We extract three sets of CTRs from the entity pane log:

1. $CTR(r)$: CTRs on related entities
2. $CTR(m, r)$: CTRs on main entities and related entities
3. $CTR(u, m, r)$: CTRs on users, main entities, and related entities

Following hybrid approaches proposed for personalized search and recommendation [5, 2, 6], we integrate the trilinear function $\Phi_{umr}(\eta)$ with the CTR features, and define a real-valued function $\Psi_{umr}(\Theta)$ as:

$$\begin{aligned}\Psi_{umr}(\Theta) &= \Phi_{umr}(\eta) + \beta^T \mathbf{c}^{umr} \\ &= \sum_{i=0}^{I}\sum_{j=0}^{J}\sum_{k=0}^{K} \eta_{ijk} \cdot x_{ui} \cdot y_{mj} \cdot z_{rk} + \beta^T \mathbf{c}^{umr}, \quad (2)\end{aligned}$$

where $\mathbf{c}^{umr}$ is a vector of CTR features specific to user u, main entity m and related entity r. β is a vector of weight coefficients. $\Theta = (\eta, \beta)$ consists of all the parameters to be learned from historical logs.

4.3 Likelihood function

In this subsection, we introduce a likelihood function to relate the values of $\Psi_{umr}(\Theta)$ to the click log collected from the entity pane. The click log provides user preferences for related entities by keeping track of clicks as implicit feedback. One important fact about the click log is that only positive observations are available - each click can be considered as positive feedback for the corresponding triple (u, m, r) indicating that user u is interested in viewing entity r, which is related to main entity m. However, the non-clicked triples (u, m, r) (i.e., given main entity m, user u did not click recommended entity r on the entity pane), do not provide such clear conclusions. There are at least two different interpretations for any non-clicked triple. One possibility is negative feedback, meaning that the user was not interested in the recommended entity. Another possibility is that the user did not even see the entity, in which case the user's interested in the entity is unknown.

User ID	Time	Main entity	Related entity	Rank	Click
32	7/9/2013 10:32:26	Pacific Rim	Man of Steel	1	0
32	7/9/2013 10:32:26	Pacific Rim	The Wolverine	2	0
32	7/9/2013 10:32:26	Pacific Rim	The Lone Ranger	3	1
32	7/9/2013 10:32:26	Pacific Rim	World War Z	4	0
498	6/16/2013 15:16:41	Leonardo DiCaprio	Kate Winslet	1	0
498	6/16/2013 15:16:41	Leonardo DiCaprio	Baz Luhrmann	2	0
498	6/16/2013 15:16:41	Leonardo DiCaprio	Johnny Depp	3	1

Figure 5: Preference relations induced by the click feedback in the entity pane log

If we simply ignore all non-clicked triples, typical machine learning algorithms are not able to learn anything from the positive observations alone. One may opt to consider the non-clicked triples as negative feedback. More specifically, training data is created by assigning positive class labels to clicked triples, and negative class labels to non-clicked triples. The problem with this approach is that all non-clicked triples the algorithm predicts in the future are presented to the learning algorithm as negative observations. This approach misinterprets non-clicked triples, which are actually missing values.

To address this problem, we use triple pairs as training data instead of individual triples. As opposed to replacing non-clicked triples with negative observations, we assume that users prefer the related entities they clicked over all other non-clicked ones on the same page impression. More specifically, given two triples (u, m, r_i) and (u, m, r_j) in the same page impression, user u prefers entity r_i over entity r_j if and only if r_i was clicked by u while r_j was not, which is denoted by $r_i^{u,m} \succ r_j^{u,m}$. Note that this assumption reasonably disregards click position bias, given the fact that only several related entities are presented in each page impression. This is different from the long lists of web search results, in which users are prone to click top ranked pages.

We create training data $\mathcal{D}$ by including all preference relations induced, as follows:

$$\mathcal{D} = \{(u, m, r_i, r_j) | r_i^{u,m} \succ r_j^{u,m} \vee r_j^{u,m} \succ r_i^{u,m}\}, \quad (3)$$

where each preference relation $o = (u, m, r_i, r_j)$ is considered as a training sample. For the entities that are both clicked by a user, we cannot infer any preference. The same is true for two entities either of which a user did not click. The running example in Figure 5 shows the preference relations induced by the click feedback in the entity pane log. In the first page impression, as the user clicked the related entity *The Lone Ranger*, we infer that he or she prefers *The Lone Ranger* over the other three recommended movies, indicated by the arrows in the figure. Similarly, it can be inferred that, for the second page impression, the user is more interested in *Johnny Depp* than the others.

A logistic function $\mathcal{F}(\cdot)$ as the likelihood function is then employed to relate the values of $\Psi_{umr}(\Theta)$ to the pairwise preference, as follows:

$$\begin{aligned} & p(r_i^{u,m} \succ r_j^{u,m} | \Psi_{umr_i}(\Theta), \Psi_{umr_j}(\Theta)) \\ = & \frac{1}{1 + e^{-g_{r_i,r_j}^{u,m}(\Psi_{umr_i}(\Theta) - \Psi_{umr_j}(\Theta))}} \\ = & \mathcal{F}(g_{r_i,r_j}^{u,m}(\Psi_{umr_i}(\Theta) - \Psi_{umr_j}(\Theta))), \quad (4)\end{aligned}$$

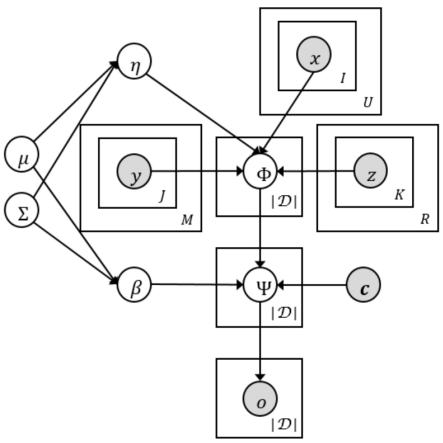

Figure 6: Graphical representation of Three-way Entity Model

where $g^{u,m}_{r_i,r_j} \in \{-1, 1\}$ denotes whether user u clicks r_i or r_j for main entity m:

$$g^{u,m}_{r_i,r_j} = \begin{cases} 1 & \text{if } u \text{ clicks } r_i \text{ given } m, \\ -1 & \text{if } u \text{ clicks } r_j \text{ given } m. \end{cases}$$

The probability $p(r^{u,m}_i \succ r^{u,m}_j | \Psi_{umr_i}(\Theta), \Psi_{umr_j}(\Theta))$ gives the likelihood that user u prefers entity r_i over entity r_j, both related to main entity m. Given the inferred parameter Θ, the likelihood of observing all preference relations in training data is then given by:

$$\begin{aligned} p(\mathcal{D}|\Theta) &= \prod_{(u,m,r_i,r_j)\in\mathcal{D}} p(r^{u,m}_i \succ r^{u,m}_j | \Psi_{umr_i}(\Theta), \Psi_{umr_j}(\Theta)) \\ &= \prod_{(u,m,r_i,r_j)\in\mathcal{D}} \frac{1}{1 + e^{-g^{u,m}_{r_i,r_j}(\Psi_{umr_i}(\Theta) - \Psi_{umr_j}(\Theta))}} \\ &= \prod_{(u,m,r_i,r_j)\in\mathcal{D}} \mathcal{F}(g^{u,m}_{r_i,r_j}(\Psi_{umr_i}(\Theta) - \Psi_{umr_j}(\Theta))). \end{aligned} \tag{5}$$

4.4 TEM & Inference

As discussed above, we need to learn the parameter Θ (i.e., η and β) from observed preference relations induced by user clicks on the entity pane, so that related entities can be recommended in the future.

For notational clarity, we define $\bar{\Theta}$ as a vector concatenating all the entries in η and β. The $\bar{\Theta}$ is considered as a random variable, and assumed to follow a Gaussian distribution:

$$\bar{\Theta} \sim \text{Gaussian}(\mu, \Sigma). \tag{6}$$

We impose a zero-mean isotropic Gaussian prior on the variable $\bar{\Theta}$, i.e.,

$$p(\bar{\Theta}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\sum_i \bar{\theta}_i^2}{2\sigma^2}}. \tag{7}$$

The graphical representation of the probabilistic model TEM is given in Figure 6. First, η is sampled from a Gaussian distribution. Given the η as well as features $\mathbf{x}_u$, $\mathbf{y}_m$, and $\mathbf{z}_r$, by Equation (1) we obtain the value of function $\Phi_{umr}(\eta)$ for each triple (u, m, r) in the training data $\mathcal{D}$. Incorporating $\Phi_{umr}(\eta)$ into the features of click-through rates weighted by β, which is drawn from a Gaussian distribution, gives the value of function $\Psi_{umr}(\Theta)$, using Equation (2). With the value of $\Psi_{umr}(\Theta)$, each preference relation o in $\mathcal{D}$ can be obtained by the likelihood function defined as in Equation (4).

We learn the parameter $\bar{\Theta}$, consisting of η and β by fitting the probabilistic model TEM to the training data $\mathcal{D}$. Specifically, we obtain the posterior distribution of the parameter Θ given all observations in training data $\mathcal{D}$, according to the Bayes' Rule:

$$p(\bar{\Theta}|\mathcal{D}) = \frac{p(\bar{\Theta})p(\mathcal{D}|\bar{\Theta})}{p(\mathcal{D})} \propto p(\bar{\Theta})p(\mathcal{D}|\bar{\Theta}), \tag{8}$$

where $p(\bar{\Theta})$ is the prior distribution defined as in Equation (7), and $p(\mathcal{D}|\bar{\Theta})$ is the likelihood of observing all preference relations defined as in Equation (5). *Maximum a posteriori* (MAP) estimation is then conducted to infer the parameter $\bar{\Theta}$. That is, we find a $\bar{\Theta}$ such that the posterior probability $p(\bar{\Theta}|\mathcal{D})$ is maximized, i.e.,

$$\begin{aligned} &\arg\max_{\bar{\Theta}} p(\bar{\Theta}|\mathcal{D}) \\ &= \arg\max_{\bar{\Theta}} p(\bar{\Theta})p(\mathcal{D}|\bar{\Theta}) \\ &= \arg\max_{\bar{\Theta}} \{\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\sum_i \bar{\theta}_i^2}{2\sigma^2}} \\ &\quad \times \prod_{(u,m,r_i,r_j)\in\mathcal{D}} \frac{1}{1 + e^{-g^{u,m}_{r_i,r_j}(\Psi_{umr_i}(\Theta) - \Psi_{umr_j}(\Theta))}}\}. \end{aligned} \tag{9}$$

We can equivalently transform this optimization problem into maximizing the logarithm of the posterior probability $p(\bar{\Theta}|\mathcal{D})$ as follows:

$$\begin{aligned} &\arg\max_{\bar{\Theta}} \mathcal{L}(\bar{\Theta}) \\ &= \arg\max_{\bar{\Theta}} \log p(\bar{\Theta}|\mathcal{D}) \\ &= \arg\max_{\bar{\Theta}} \{-\frac{\sum_i \bar{\theta}_i^2}{2\sigma^2} \\ &\quad + \sum_{(u,m,r_i,r_j)\in\mathcal{D}} \log \frac{1}{1 + e^{-g^{u,m}_{r_i,r_j}(\Psi_{umr_i}(\Theta) - \Psi_{umr_j}(\Theta))}}\}. \end{aligned} \tag{10}$$

Equation (10) is an unconstrained convex optimization problem, which has a unique maximum. We use the *Limited-memory BFGS* algorithm [13] to solve the optimization problem and to estimate the parameters η and β. This involves computation of the gradients $\nabla_\eta \mathcal{L}(\bar{\Theta})$ and $\nabla_\beta \mathcal{L}(\bar{\Theta})$, i.e.:

$$\begin{aligned} \frac{\partial \mathcal{L}(\bar{\Theta})}{\partial \eta_{ijk}} &= \sum_{(u,m,r_a,r_b)\in\mathcal{D}} \{\frac{g^{u,m}_{r_a,r_b}}{1 + e^{g^{u,m}_{r_a,r_b}(\Psi_{umr_a}(\Theta) - \Psi_{umr_b}(\Theta))}} \\ &\quad \times x_{ui} y_{mj} (z_{r_a k} - z_{r_b k}) - \frac{\eta_{ijk}}{\sigma^2}\}, \end{aligned} \tag{11}$$

$$\begin{aligned} \frac{\partial \mathcal{L}(\bar{\Theta})}{\partial \beta_i} &= \sum_{(u,m,r_a,r_b)\in\mathcal{D}} \{\frac{g^{u,m}_{r_a,r_b}}{1 + e^{g^{u,m}_{r_a,r_b}(\Psi_{umr_a}(\Theta) - \Psi_{umr_b}(\Theta))}} \\ &\quad \times (c_i^{umr_a} - c_i^{umr_b}) - \frac{\beta_i}{\sigma^2}\}. \end{aligned} \tag{12}$$

With the parameter estimate $\hat{\Theta} = (\hat{\eta}, \hat{\beta})$, we can recommend a ranked list of entities r related to the main entity m searched by any user u. More specifically, given any triple (u, m, r), we compute the value of function $\Psi_{umr}(\hat{\Theta})$ by:

$$\Psi_{umr}(\hat{\Theta}) = \sum_{i=0}^{I}\sum_{j=0}^{J}\sum_{k=0}^{K} \hat{\eta}_{ijk} \cdot x_{ui} \cdot y_{mj} \cdot z_{rk} + \hat{\beta}^T \mathbf{c}^{umr}. \tag{13}$$

Table 2: Statistics of experimental datasets

Dataset	# users	# entities	# instances
Movie	36,641	15,409	224,567
Celebrity	26,371	2,016	1,450,609

Related entities are then ranked in descending order of the $\Psi_{umr}(\hat{\Theta})$ scores. Entities with the highest scores will be recommended to the user u.

4.5 Three-way Interaction Effect

The TEM model raises the important question: How significant is the effect of the three-way correlations to modeling user clicks? To answer this question, one may test the statistical significance of the interaction effect with a t-test on the weights η which quantify the correlations among $\mathbf{x}_u$, $\mathbf{y}_m$, and $\mathbf{z}_r$. This practice, however, misinterprets the weight coefficients η in the nonlinear TEM model [3]. The correct measure of the three-way interaction effect for TEM should be a third partial derivative of the likelihood function $\mathcal{F}(\cdot)$ instead.

Let Δ denote either the derivative or the difference operator, depending on whether the corresponding feature values are discrete or continuous. The three-way interaction effect is then estimated by $\hat{\mu}_{xyz} = \frac{\Delta^3 \mathcal{F}}{\Delta x \Delta y \Delta z}$. When x, y and z are discrete features, the interaction effect can be derived as:

$$\begin{aligned}\hat{\mu}_{xyz} &= \frac{\Delta^3 \mathcal{F}}{\Delta x \Delta y \Delta z} \\ &= \mathcal{F}(\eta_x + \eta_y + \eta_z + \eta_{xy} + \eta_{xz} + \eta_{yz} + \eta_{xyz} + \tilde{\mathbf{c}}) \\ &\quad -\mathcal{F}(\eta_x + \eta_y + \eta_{xy} + \tilde{\mathbf{c}}) - \mathcal{F}(\eta_x + \eta_z + \eta_{xz} + \tilde{\mathbf{c}}) \\ &\quad -\mathcal{F}(\eta_y + \eta_z + \eta_{yz} + \tilde{\mathbf{c}}) + \mathcal{F}(\eta_z + \tilde{\mathbf{c}}) \\ &\quad +\mathcal{F}(\eta_y + \tilde{\mathbf{c}}) + \mathcal{F}(\eta_x + \tilde{\mathbf{c}}) - \mathcal{F}(\tilde{\mathbf{c}}) \end{aligned} \quad (14)$$

where the η terms denote the weights of the features specified by the respective subscripts, and $\tilde{\mathbf{c}}$ represents the linear combination of all remaining features and weight coefficients. When some or all of x, y and z are continuous features, we can derive similar equations for the interaction effect, which are omitted due to the lack of space.

The standard error of the interaction effect estimate $\hat{\mu}_{xyz}$ is obtained by the Delta method:

$$\hat{\mu}_{xyz} \sim \text{Gaussian}\left(\mu_{xyz}, \frac{\partial}{\partial \eta}\left[\frac{\Delta^3 \mathcal{F}}{\Delta x \Delta y \Delta z}\right] \Omega_\eta \frac{\partial}{\partial \eta}\left[\frac{\Delta^3 \mathcal{F}}{\Delta x \Delta y \Delta z}\right]\right), \quad (15)$$

which gives the estimate of the asymptotic variance of $\hat{\mu}_{xyz}$:

$$\hat{\sigma}^2_{xyz} = \frac{\partial}{\partial \eta}\left[\frac{\Delta^3 \mathcal{F}}{\Delta x \Delta y \Delta z}\right] \hat{\Omega}_\eta \frac{\partial}{\partial \eta}\left[\frac{\Delta^3 \mathcal{F}}{\Delta x \Delta y \Delta z}\right], \quad (16)$$

where $\hat{\Omega}_\eta$ is a consistent covariance estimator of η.

For the t-test, we define the t statistic as $t = \frac{\hat{\mu}_{xyz}}{\hat{\sigma}_{xyz}}$. With the statistic, we test the null hypothesis that the overall effects of the three-way interactions equal zero for given training data, which gives p-value < 0.05. So we reject the null hypothesis, which indicates the fact that the three-way interaction effect is statistically significant to modeling user clicks on related entities.

5. EMPIRICAL EVALUATION

In this section, we report the experimental results of TEM on real-world data collected by a commercial search engine.

Table 3: Features for movie & celebrity recommendation

Movie recommendation	
User dimension	*Main & related movie*
Viewed entities	
Types of viewed entities	Actors
Viewed movie's actors	Directors
Viewed movie's directors	Genres
Viewed movie's genres	Country of origin
Viewed movie's country	Language
Viewed movie's language	Producers
Viewed movie's producers	Series
Viewed movie's series	Story
Viewed movie's story	Subject
Viewed movie's subject	Music
Viewed movie's music	.....
.....	

Celebrity recommendation	
User dimension	*Main & related celebrity*
Viewed entities	Profession
Types of viewed entities	Movie acted
Attributes of viewed entities	Movie directed
Viewed pop singers	Book written
Viewed business leaders	Music genre
Viewed writers	Organization
Viewed musicians	Spouse
Viewed actors	Nationality
Viewed film directors	Language
.....	Types
	.....

We compare the results of TEM against those of several competitors. Analysis and discussion of the experimental results are presented in this section.

5.1 Data

Although TEM is a generic probabilistic model which is applicable to recommending various kinds of entities, we take two specific types of recommendation tasks as case studies for empirical evaluation: *movie recommendation* and *celebrity recommendation*. The movie recommendation task is to recommend a ranked list of movies that are related to the movie searched by the user. For celebrity recommendation, we aim to present to the user other celebrities related to the one he or she searched for.

We collected the entity pane log data for March 2013 through July 2013 from a commercial search engine. For the two recommendation tasks, movies and celebrities were extracted by aligning the entities in the log with those in Freebase. Freebase is a collaborative knowledge base of more than twenty million entities, including well-known people, places, movies and things. The basic statistics of the movie dataset and the celebrity dataset are given in Table 2.

Table 3 lists some features we used for the two recommendation tasks (Due to the space limitation, we do not list all the features in this paper). Specifically, to develop user profiles, by joining *search click log*, *entity pane log* and the Freebase *knowledge base* (See Section 3 for details), we collected the popular entities the users had viewed together with their attributes/types as features, such as popular movies, pop stars and well-known writers. Each of the features was represented as the frequency of its occurrence in the logs for each user. In addition, for movie recommendation, with the help of the knowledge base we included the attributes (i.e., genres, countries, languages, etc.) of the movies viewed into the user dimension. This enables the model to learn user characteristics from the various aspects of their viewed

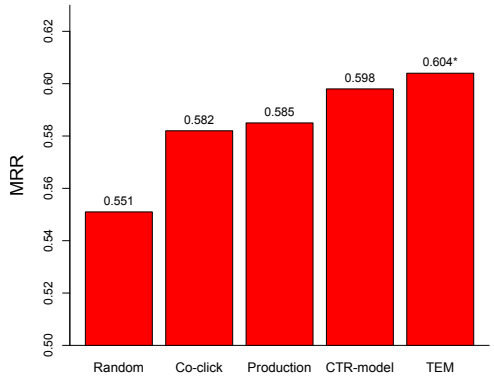

Figure 7: MRR for movie recommendation

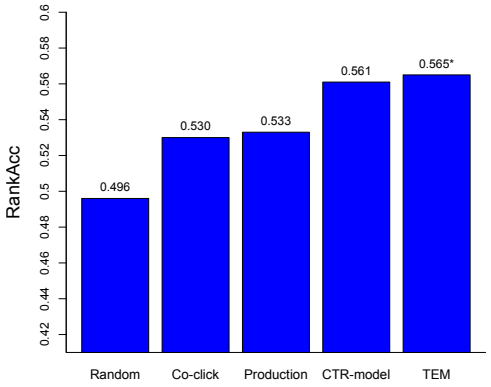

Figure 8: RankAcc for movie recommendation

movies, and thus to recommend related movies based on their preferences. As for celebrity recommendation, we included popular celebrities the users had viewed in the user-specific feature vectors, such as business leaders, musicians, actors and directors. For main entities and related entities, we constructed two different feature sets for the movie task and the celebrity task. For movie recommendation, we extracted the attributes of the movies as features, such as actors, directors, genres, languages, and subjects, whereas the celebrity recommendation model selected the celebrity-related features, such as the movies directed by the directors, the books written by the writers, and their spouses. With the domain-specific feature sets, TEM is able to discover the correlations among the three dimensions for recommending related entities.

In addition to the features listed in Table 3, we obtained the features of click-through rates (CTR) which are considered very strong signals for recommendation. More specifically, based on the entity pane log, we collected r-specific CTRs: $CTR(r)$, (m,r)-specific CTRs: $CTR(m,r)$, and (u,m,r)-specific CTRs: $CTR(u,m,r)$. We also collected from the search log the frequency of entities viewed in the same session. As a result, for movie recommendation there were a total of 1653 features for each user, and 419 features for each main entity and related entity. For celebrity recommendation, there were a total of 1938 features for each user, and 562 features for each main entity and related entity.

5.2 Evaluation strategy

To evaluate the quality of entity recommendation, we split both the movie dataset and the celebrity dataset into a training set and a test set. The test set consisted of the latest page impression for each user, and the training set contained the rest. That is, TEM was used to rank a list of entities related to the last main entity searched by each user.

Let Q be a set of tuples (u,m). For each tuple $(u,m) \in Q$, a recommendation algorithm returns a ranked list of related entities with respect to user u and main entity m. To analyze the recommendation results, we used two evaluation metrics. The first metric was the standard Mean Reciprocal Rank (MRR). The Reciprocal Rank of a ranked list is the multiplicative inverse of the rank of the first hit in the list. The MRR score of a recommendation algorithm is the average reciprocal rank obtained by the ranked lists given by the algorithm with respect to the set Q. Formally,

$$MRR = \frac{1}{|Q|}\sum_{n=1}^{|Q|}\frac{1}{rank^{(n)}}, \tag{17}$$

where $rank^{(n)}$ is the rank of the first clicked entity in the ranked list for the n-th tuple. The other metric used for evaluation was called RankAcc. RankAcc was introduced to measure what fraction of preference orders $r_i \succ r_j$ is captured by a ranked list of recommended entities. Formally, we define RankAcc of a recommendation algorithm as:

$$RankAcc = \frac{1}{|Q|}\sum_{n=1}^{|Q|}\frac{|\{(r_i^{(n)}, r_j^{(n)})|i<j \wedge r_i^{(n)} \succ r_j^{(n)}\}|}{|\{(r_i^{(n)}, r_j^{(n)})|i<j\}|}, \tag{18}$$

where i and j are the ranks of related entities r_i and r_j in the ranked list, respectively. So $i<j$ suggests that entity r_i is ranked higher than entity r_j. Therefore, the fraction in Equation (18) gives the number of preference orders $r_i \succ r_j$ consistent with the rank orders out of the total number of pairs (r_i, r_j) induced by the rank.

5.3 Recommendation accuracy

We first evaluated the recommendation quality of the compared algorithms, *Random*, *Co-click*, *Production*, *CTR-model*, and *TEM* on the two real-world datasets. The *Random* approach is a naive algorithm which randomly ranks the related entities in each page impression[3]. *Co-click* exploits the valuable signal that an entity should be recommended for another given entity if and only if the two entities are frequently co-clicked. Specifically, given a main entity m, *Co-click* estimates $p(r|m)$, the conditional probability of recommending related entity r, based on the number of their co-occurrences in the click log. *Co-click* then ranks the entities r by the conditional probabilities $p(r|m)$. The co-click signal by itself has been shown to be very effective and the strongest baseline method for entity recommendation [19]. *Production* represents the recommendation approach currently employed by a commercial search engine. It reflects the state of the art in the specific application by major search engines. *CTR-model* is a simplified version of *TEM*. It builds up the recommendation model in a way similar to *TEM*, except that *CTR-model* only utilizes the CTR features without incorporating the trilinear function $\Phi_{umr}(\eta)$. In essence, *CTR-model* and *TEM* build upon the same probabilistic framework, while different in feature sets used for training. We introduced the *CTR-model* in the interests of investigating the power of the CTR features derived from the entity pane log as well as the power of our probabilistic

[3]The number of related entities presented in each page impression is greatly limited by a user's screen size. It is normally ranging from 3 to 5. As a result, an algorithm which always provides the worst rankings would produce the MRR score approximately 0.25.

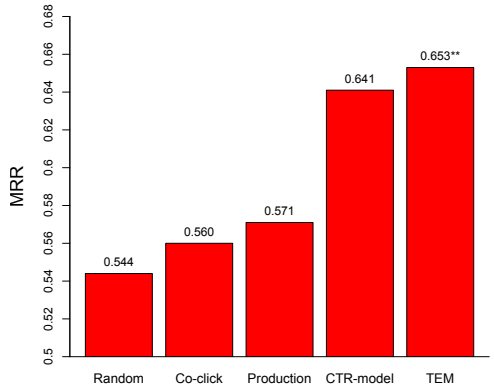

Figure 9: MRR for celebrity recommendation

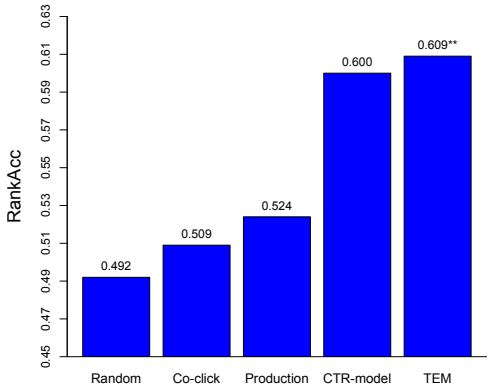

Figure 10: RankAcc for celebrity recommendation

framework. We set $\sigma^2 = 5$ for the Gaussian prior in the probabilistic framework. For the *TEM* model, we set the number of random projection dimensions as 20, since that produced the best recommendations.

Figure 7 shows the MRR score of each algorithm for movie recommendation. From this figure, we observe that the other four methods are clearly superior to the *Random* approach. *Co-click* and *Production* give similar MRR results, as current search engines recommend related entities based on the co-click signal. It is interesting to see that *CTR-model* produces a high MRR result, even better than the state-of-the-art baseline *Production*. This shows the great potential of the click feedback in the entity pane log. Also, it suggests the ability of our probabilistic framework to leverage CTR signals for entity recommendation. To further compare *CTR-model* and *TEM*, we performed a paired t-test which had p-value < 0.05, denoted by $*$. It indicated that the improvement of *TEM* over *CTR-model* is statistically significant. Figure 8 depicts the RankAcc scores of all compared algorithms for movie recommendation. From this figure, we observe the pattern similar to that of Figure 7.

For celebrity recommendation, the MRR and the RankAcc are shown in Figure 9 and Figure 10, respectively. Again, it is observed that *CTR-model* produces much higher MRR and RankAcc than those of both baselines *Co-click* and *Production*, and that *TEM* consistently outperforms all the other methods. To further measure the improvement of *TEM* over *CTR-model*, we performed a paired t-test between the two approaches. The $**$ in both figures indicate p-value < 0.01, which show that *TEM* significantly improves over *CTR-model*.

5.4 Efficacy of personalization

Our *TEM* model personalizes recommendation results by taking the user dimension into consideration. The user dimension captures a user's past interactions with the search engine, such as the various types of entities he or she has viewed and their characteristics. *TEM* analyzes the underlying associations between induced user profiles and their actions on the entity pane to recommend the related entities tailored to their interests.

We took movie recommendation as a case study to investigate the personalization efficacy of *TEM*. Table 4 depicts an example of ranking the movie entities related to the movie *The Great Gatsby* by the four approaches *Co-click*, *Production*, *CTR-model*, and *TEM*. The particular search user was a fan of actor *Leonardo DiCaprio*, who starred in *The Great Gatsby*. Among the four related movies, the user jumped

Table 4: Related movies recommended for a fan of actor Leonardo DiCaprio by *Co-click*, *Production*, *CTR-model*, and *TEM*

User		
A fan of actor Leonardo DiCaprio		
Main movie entity		
The Great Gatsby		
Related movie entities		
Co-click / Production	*CTR-model*	*TEM*
Iron Man 3	Iron Man 3	Django Unchained
Man of Steel	Star Trek (2013)	Iron Man 3
Star Trek (2013)	Django Unchained	Star Trek (2013)
Django Unchained	Man of Steel	Man of Steel

to *Django Unchained* to explore, which is the only movie starring *Leonardo DiCaprio*. Since the user had viewed the entity of *Leonardo DiCaprio*, by analyzing her historical logs *TEM* recognized her interest and thus put *Django Unchained* at the top of the ranked list. On the other hand, the other three approaches failed to customize the ranking of the related movies based on the user's interest.

To further study the personalization efficacy of *TEM*, we conducted a quantitative evaluation. In particular, we first split the movie test set into five subsets based on the numbers of movie entities viewed by the users in the past. The number of users in each test subset is given in Figure 11. It is seen that 42% of users have viewed at least one movie entity in our log. The methods *Co-click*, *Production*, *CTR-model*, and *TEM* were used to recommend related movies for the users in each test set to evaluate their efficacy of personalization. Figure 12 shows the MRR scores of each algorithm on each test set. From this figure, it is observed that the three methods *Co-click*, *Production* and *CTR-model* produce consistent MRR results across the different test sets in spite of the drop for the "7∼9" set[4]. This suggests that the unique preference of an individual user has little effect on the three methods for customizing the recommendation results. Our *TEM* model, however, increases the MRR as users have viewed an increasing number of movie entities. This confirms *TEM*'s ability to personalize recommended entities. Enriching user profiles will potentially improve the quality of recommendation of *TEM* for users.

5.5 Effect of random projections

In this section, we investigate the effect of random projections on the quality of recommendation for *TEM*. In par-

[4]The MRR for the test set "7∼9" is not statistically reliable given the small number of users in the set.

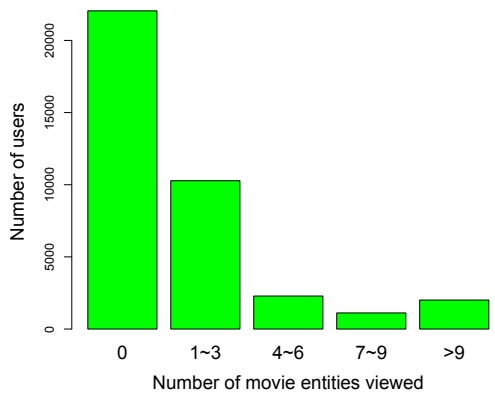

Figure 11: User distribution over the numbers of movie entities viewed in the past

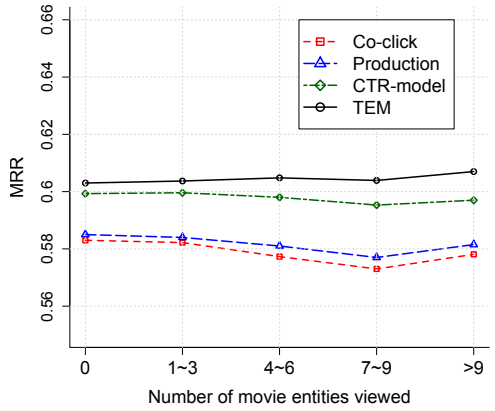

Figure 12: MRR for varying numbers of movie entities viewed in the past

ticular, we applied *TEM* to the training data of varying-dimensional feature space produced by random projections, and computed MRR for each random projection dimension. Figure 13 plots the MRR scores of the two recommendation tasks for different random projection dimensions. We observe that as more dimensions were used, *TEM* produced better recommendation results for both tasks. This is not surprising because increasing the number of random projection dimensions increases the capacity of the *TEM* model by giving it more tunable parameters, and also preserves more information about the original data.

6. CONCLUSION

This paper addresses the problem of recommending entities related to the main entity returned to a user by a web search engine. We propose the probabilistic model TEM, which leverages the three data sources, *knowledge base*, *search click log*, and *entity pane log*, for personalized recommendation of related entities. The TEM model not only utilizes the CTR signals derived from the entity pane log, but also exploits the three-way relationships among *user*, *main entity*, and *related entity*. Experimental results on movie recommendation and celebrity recommendation show that TEM with our probabilistic framework significantly improves over the state of the art technique employed by a major search engine. This confirms the effectiveness of TEM and the probabilistic framework on related entity recommendation.

7. REFERENCES

[1] E. Acar and B. Yener. Unsupervised multiway data analysis: A literature survey. *IEEE TKDE*, 2009.

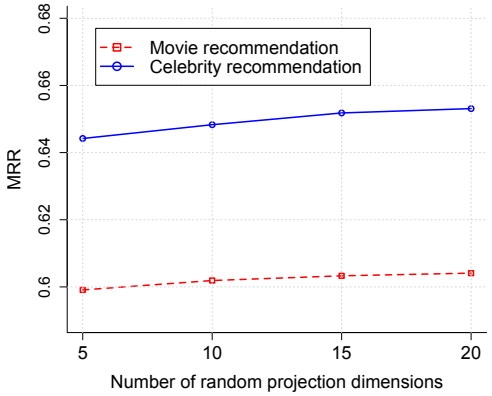

Figure 13: MRR for varying random projection dimensions

[2] J.-w. Ahn, P. Brusilovsky, J. Grady, D. He, and S. Y. Syn. Open user profiles for adaptive news systems: Help or harm? In *Proc. of WWW '07*, pages 11–20, Canada, 2007.

[3] C. Ai and E. C. Norton. Interaction terms in logit and probit models. *Economics Letters*, 80(1):123 – 129, 2003.

[4] R. Blanco, B. B. Cambazoglu, P. Mika, and N. Torzec. Entity recommendations in web search. In *International Semantic Web Conference (2)*, pages 33–48. Springer, 2013.

[5] R. Burke. Hybrid systems for personalized recommendations. In *Proc. of ITWP '03*, pages 133–152, Acapulco, Mexico, 2005.

[6] W. Chu and S.-T. Park. Personalized recommendation on dynamic content using predictive bilinear models. In *Pro. of WWW '09*, pages 691–700, Madrid, Spain, 2009.

[7] R. Coppi and S. Bolasco, editors. *Multiway data analysis*. North-Holland Publishing Co., Amsterdam, 1989.

[8] Y. Inagaki, N. Sadagopan, G. Dupret, A. Dong, C. Liao, Y. Chang, and Z. Zheng. Session based click features for recency ranking. In *Proc. of AAAI '10*. AAAI Press.

[9] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proc. of STOC '98*, pages 604–613, Dallas, Texas, USA, 1998.

[10] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver. Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering. In *Proc. of RecSys '10*, pages 79–86, Barcelona, Spain, 2010.

[11] A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani. Adwords and generalized online matching. *Journal of the ACM*, 54(5), Oct. 2007.

[12] A. Micarelli, F. Gasparetti, F. Sciarrone, and S. Gauch. The adaptive web. chapter Personalized Search on the World Wide Web, pages 195–230. Berlin, Heidelberg, 2007.

[13] J. Nocedal and S. Wright. *Numerical Optimization*. Springer series in operations research and financial engineering. Springer, New York, NY, 2nd edition, 2006.

[14] S. Rendle and L. Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *Proc. of WSDM '10*, pages 81–90, New York, USA, 2010.

[15] M. Speretta and S. Gauch. Personalized search based on user search histories. In *Proc. of WI '05*, pages 622–628, Washington, DC, USA, 2005.

[16] J.-T. Sun, H.-J. Zeng, H. Liu, Y. Lu, and Z. Chen. Cubesvd: A novel approach to personalized web search. In *Proc. of WWW '05*, pages 382–390, Chiba, Japan, 2005.

[17] L. R. Tucker. The extension of factor analysis to three-dimensional matrices. In *Contributions to mathematical psychology.*, pages 110–127. New York, 1964.

[18] M. A. O. Vasilescu and D. Terzopoulos. Multilinear image analysis for facial recognition. In *Proc. of ICPR '02*, pages 511–514, 2002.

[19] X. Yu, H. Ma, B.-J. P. Hsu, and J. Han. On building entity recommender systems using user click log and freebase knowledge. In *Proc. of WSDM '14*, pages 263–272, New York, NY, USA, 2014.