Journal of the Royal Statistical Society
SERIES C Applied Statistics

Appl. Statist. (2012)
61, *Part* 3, *pp.* 471–492

Estimating probabilities in recommendation systems

Mingxuan Sun and Guy Lebanon

Georgia Institute of Technology, Atlanta, USA

and Paul Kidwell

Lawrence Livermore National Laboratory, Livermore, USA

[Received December 2010. Final revision October 2011]

Summary. Recommendation systems are emerging as an important business application with significant economic impact. Currently popular systems include Amazon's book recommendations, Netflix's movie recommendations and Pandora's music recommendations. We address the problem of estimating probabilities associated with recommendation system data by using non-parametric kernel smoothing. In our estimation we interpret missing items as randomly censored observations of preference relations and obtain efficient computation schemes by using combinatorial properties of generating functions. We demonstrate our approach with several case-studies involving real world movie recommendation data. The results are comparable with state of the art techniques while also providing probabilistic preference estimates outside the scope of traditional recommender systems.

Keywords: Kernel smoothing; Ranked data; Recommender systems

1. Introduction

Recommendation systems are emerging as an important business application with significant economic impact. The data in such systems are collections of incomplete tied preferences across n items that are associated with m different users. Given an incomplete tied preference associated with an additional $(m+1)$th user, the system recommends unobserved items to that user on the basis of the preference relations of the $m+1$ users. Currently deployed recommendation systems include book recommendations at amazon.com, movie recommendations at netflix.com and music recommendations at pandora.com. Constructing accurate recommendation systems (that recommend to users items that are truly preferred over other items) is important for assisting users as well as increasing business profitability. It is an important topic of on-going research in machine learning and data mining.

In most cases of practical interest the number of items n that are indexed by the system (items may be books, movies, songs, etc.) is relatively high in the 10^3–10^4 range. Perhaps because of the size of n, almost always each user observes only a small subset of the items, typically in the range 10–100. As a result the preference relations that are expressed by the users are over a small subset of the n items.

Formally, we have m users providing incomplete tied preference relations on n items

Address for correspondence: Mingxuan Sun, College of Computing, Georgia Institute of Technology, 266 Ferst Drive, Atlanta, GA 30332, USA.
E-mail: cynthia@cc.gatech.edu

 0035–9254/12/61471

$$\left.\begin{aligned} S_1: &\quad A_{1,1} \prec A_{1,2} \prec \ldots \prec A_{1,k(1)}, \\ S_2: &\quad A_{2,1} \prec A_{2,2} \prec \ldots \prec A_{2,k(2)}, \\ &\quad \vdots \\ S_m: &\quad A_{m,1} \prec A_{m,2} \prec \ldots \prec A_{m,k(m)} \end{aligned}\right\} \tag{1}$$

where $A_{i,j} \subset \{1, \ldots, n\}$ are sets of items (without loss of generality we identify items with integers $1, \ldots, n$) defined by the following interpretation: user i prefers all items in $A_{i,j}$ to all items in $A_{i,j+1}$. The notation $k(i)$ above is the number of such sets provided by user i. The data (1) are incomplete since not all items are necessarily observed by each user, i.e. $\cup_{j=1}^{k(i)} A_{i,j} \subsetneq \{1, \ldots, n\}$, and may contain ties since some items are left uncompared, i.e. $|A_{i,j}| > 1$. Recommendation systems recommend items to a new user, denoted as $m+1$, on the basis of their preference

$$S_{m+1}: A_{m+1,1} \prec A_{m+1,2} \prec \ldots \prec A_{m+1,k(m+1)} \tag{2}$$

and its relationship to the preferences of the m users (1).

As an illustrative example, assuming that $n = 9$ and $m = 3$, the data

$$\begin{aligned} S_1: &\quad 1, 8, 9 \prec 4 \prec 2, 3, 7, \\ S_2: &\quad 4 \prec 2, 3 \prec 8, \\ S_3: &\quad 4, 8 \prec 2, 6, 9 \end{aligned}$$

correspond to $A_{1,1} = \{1, 8, 9\}$, $A_{1,2} = \{4\}$, $A_{1,3} = \{2, 3, 7\}$, $A_{2,1} = \{4\}$, $A_{2,2} = \{2, 3\}$, $A_{2,3} = \{8\}$, $A_{3,1} = \{4, 8\}$ and $A_{3,2} = \{2, 6, 9\}$, and $k(1) = k(2) = 3$ and $k(3) = 2$. From the data we may guess that item 4 is relatively popular across the board whereas some users like item 8 (users 1 and 3) and some hate it (user 2). Given a new $(m+1)$th user issuing the preference $1 \prec 2, 3, 7$ we might observe a similar pattern of preference or taste to that of user 1 and recommend to the user item 8. We may also recommend item 4 which has broad appeal resulting in the augmentation

$$1 \prec 2, 3, 7 \mapsto 1, 4, 8 \prec 2, 3, 7.$$

We note that in some cases the preference relations (1) arise from users providing numeric scores to items. For example, if the users assign 1–5 stars to movies, the set $A_{i,j}$ contains all movies that user i assigned $6 - j$ stars to and $k(i) = 5$ (assuming that some movies were assigned to each of the 1-, 2-, 3-, 4- and 5-star levels). As pointed out by a wide variety of studies in economics and social sciences, e.g. Cliff and Keats (2003), such numeric scores are inconsistent among different users. We therefore proceed to interpret such data as ordinal rather than numeric.

A substantial body of literature in computer science has addressed the problem of constructing recommendation systems as this has been an active research area since the 1990s. We have attempted to outline the most important and successful approaches. The earliest efforts made a prediction for the rating of items based on the similarity of the test user and the training users (Resnick *et al.*, 1994; Breese *et al.*, 1998; Herlocker *et al.*, 1999). Specifically, these attempts used similarity measures such as Pearson correlation (Resnick *et al.*, 1994) and vector cosine similarity (Breese *et al.*, 1998; Herlocker *et al.*, 1999) to evaluate the similarity level between different users.

More recent work includes user and movie clustering (Breese *et al.*, 1998; Ungar and Foster, 1998; Xue *et al.*, 2005), item–item similarities (Sarwar *et al.*, 2001), Bayesian networks (Breese *et al.*, 1998), dependence network (Heckerman *et al.*, 2000) and probabilistic latent variable models (Pennock *et al.*, 2000; Hofmann, 2004; Marlin, 2004).

Most recently, the state of the art methods including the winner of the Netflix competition are based on non-negative matrix factorization of the partially observed user rating matrix. The factorized matrix can be used to fill out the unobserved entries in a way that is similar to latent factor analysis (Goldberg *et al.*, 2001; Rennie and Srebro, 2005; Lawrence and Urtasun, 2009; Koren, 2010).

Each of the above methods focuses exclusively on user ratings. In some cases item information is available (movie genre, actors, directors, etc.) which have led to several approaches that combine voting information with item information (Basu *et al.*, 1998; Popescul *et al.*, 2001; Schein *et al.*, 2002).

Our method differs from the methods above in that it constructs a full probabilistic model on preferences, it can handle heterogeneous preference information (not all users must specify the same number of preference classes) and does not make any parametric assumptions. In contrast with previous approaches it enables clear meaningful statistical estimation procedures for not only the prediction of item ratings, but also the discovery of association rules and the estimation of probabilities of interesting events. Note that non-negative matrix factorization may be considered a probabilistic model assuming exponential family models such as Poisson or normal. Such an approach, however, models the scores as numeric variables rather than the ordering themselves as is the approach of this paper.

In this paper we describe a non-parametric statistical technique for estimating probabilities on preferences based on the data (1). This work extends non-parametric density estimation over rankings (Lebanon and Mao, 2008) to include ranking data of arbitrary incompleteness and tie structure making it amenable to a wide range of real world applications. This technique may be used in recommendation systems in different ways. Its principal usage may be to provide a statistically meaningful estimation framework for issuing recommendations (in conjunction with decision theory). However, it also leads to other important applications including mining association rules, exploratory data analysis and clustering items and users. Two key observations that we make are

(a) incomplete tied preference data may be interpreted as randomly censored permutation data and
(b) using generating functions we can provide a computationally efficient scheme for computing the estimator in the case of triangular smoothing.

We proceed in the next sections to describe notation and our assumptions and estimation procedure, and we follow with case-studies demonstrating our approach on real world recommendation systems data.

2. Definitions and estimation framework

We describe the following notation and conventions for permutations, which are taken from Diaconis (1988) where more detail may be found. We denote a permutation by listing the items from most preferred to least separated by a '$\prec$' or '|' symbol: $\pi^{-1}(1) \prec \pi^{-1}(2) \prec \ldots \prec \pi^{-1}(n)$, e.g. $\pi(1) = 2$, $\pi(2) = 3$ and $\pi(3) = 1$ is $3 \prec 1 \prec 2$. Rankings with ties occur when judges do not provide enough information to construct a total order. In particular, we define tied rankings as a partition of $\{1, \ldots, n\}$ to $k < n$ disjoint subsets $A_1, \ldots, A_k \subset \{1, \ldots, n\}$ such that all items in A_i are preferred to all items in A_{i+1} but no information is provided concerning the relative preference of the items among the sets A_i. We denote such rankings by separating the items in A_i and A_{i+1} with a '$\prec$' or '|' notation. For example, the tied ranking $A_1 = \{3\}, A_2 = \{2\}$ and $A_3 = \{1, 4\}$ (items 1 and 4 are tied for last place) is denoted as $3 \prec 2 \prec 1, 4$ or $3|2|1, 4$.

Rankings with missing items occur when judges omit certain items from their preference information altogether. For example assuming a set of items $\{1,\ldots,4\}$, a judge may report a preference $3 \prec 2 \prec 4$, omitting altogether item 1 which the judge did not observe or experience. This case is very common in situations involving a large number of items n. In this case judges typically provide a preference only for the $l \ll n$ items that they observed or experienced. For example, in movie recommendation systems we may have $n \sim 10^3$ and $l \sim 10^1$.

Rankings can be full (permutations), with ties, with missing items or with both ties and missing items. In either case we denote the rankings by using the '$\prec$' or '|' notation or using the disjoint sets $A_1,\ldots,A_k$ notation. We also represent tied and incomplete rankings by the set of permutations that are consistent with it. For example,

$$3 \prec 2 \prec 1,4 = \{3 \prec 2 \prec 1 \prec 4\} \cup \{3 \prec 2 \prec 4 \prec 1\},$$
$$3 \prec 2 \prec 4 = \{1 \prec 3 \prec 2 \prec 4\} \cup \{3 \prec 1 \prec 2 \prec 4\} \cup \{3 \prec 2 \prec 1 \prec 4\} \cup \{3 \prec 2 \prec 4 \prec 1\}$$

are sets of two and four permutations corresponding to tied and incomplete rankings respectively.

It is difficult to posit directly a coherent probabilistic model on incomplete tied data such as data (1). Different preferences relations are not unrelated to each other: they may subsume one another (e.g. $1 \prec 2 \prec 3$ and $1 \prec 3$), represent disjoint events (e.g. $1 \prec 3$ and $3 \prec 1$) or interact in more complex ways (e.g. $1 \prec 2 \prec 3$ and $1 \prec 4 \prec 3$). A valid probabilistic framework needs to respect the constraints resulting from the axioms of probability, e.g. $p(1 \prec 2 \prec 3) \leqslant p(1 \prec 3)$.

Our approach is to consider the incomplete tied preferences as censored permutations, i.e. we assume a distribution $p(\pi)$ over permutations $\pi \in \mathcal{G}_n$ ($\mathcal{G}_n$ is the symmetric group of permutations of order n) that describes the complete without-ties preferences in the population. The data that are available to the recommender system (1) are sampled by drawing m independent identically distributed (IID) permutations from $p : \pi_1,\ldots,\pi_m \sim^{\text{IID}} p$, followed by censoring to result in the observed preferences $S_1,\ldots,S_m$

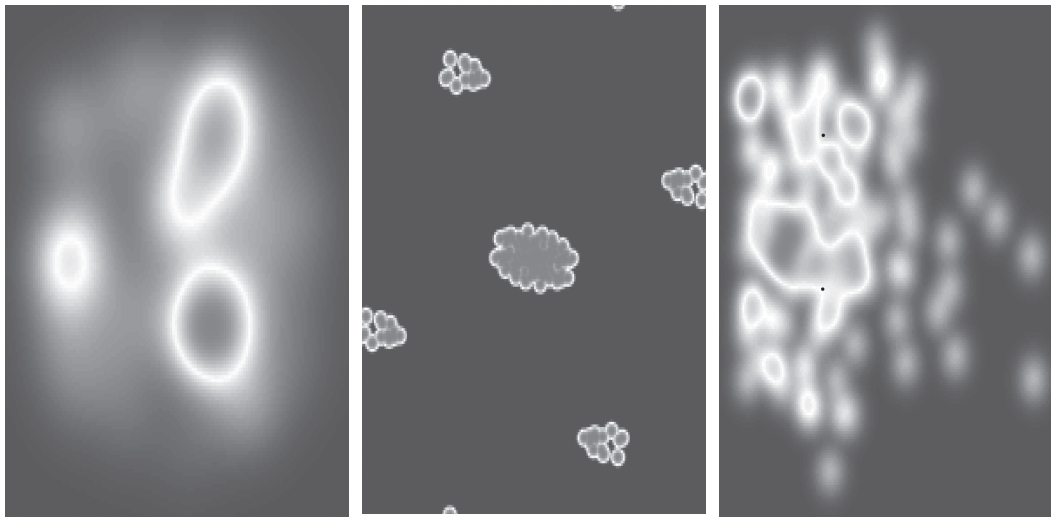
$$\pi_i \sim p(\pi), \qquad S_i \sim p(S|\pi_i), \qquad i = 1,\ldots,m+1, \tag{3}$$

$$p(\pi|S) = \frac{I(\pi \in S)\, p(\pi)}{\sum\limits_{\sigma \in S} p(\sigma)}, \tag{4}$$

$$p(S|\pi) = \frac{p(\pi|S)\, q(S)}{p(\pi)} = \frac{I(\pi \in S)\, p(\pi)\, q(S)}{p(\pi) \sum\limits_{\sigma \in S} p(\sigma)} = \frac{I(\pi \in S)\, q(S)}{\sum\limits_{\sigma \in S} p(\sigma)} \tag{5}$$

where $q(S)$ represents the probability of observing the censoring S consisting of permutations σ or equivalently it describes a random process resulting in a particular censoring (specifically, it is not equal to $\Sigma_{\sigma \in S}\, p(\sigma)$).

Although many approaches for estimating p given $S_1,\ldots,S_m$ are possible, experimental evidence points to the fact that, in recommendation systems with high n, the distribution p does not follow a simple parametric form such as the Mallows, Bradley–Terry or Thurstone models (Marden, 1996). Fig. 1 gives a demonstration how the number of modes and complexity increase with n. In Fig. 1, which appears also in Kidwell *et al.* (2008), we show a density estimate (using kernel smoothing) of rankings embedded in a two-dimensional space by using multi-dimensional scaling. The distance function in this case was the average Kendall's τ-distance over all possible permutations that are consistent with the partial rankings. Fig. 1 shows three different panels corresponding to different data sets of increasing n. (Note that the choice of metric on rankings is not straightforward; here Kendall's τ is a reasonable choice as it displays a high



(a) (b) (c)

Fig. 1. Heat map visualization of the density of ranked data by using multi-dimensional scaling with expected Kendall's τ-distance: (a) APA voting ($n = 5$); (b) Jester ($n = 100$); (c) EachMovie ($n = 1628$)

degree of sensitivity producing separation even when projected into two dimensions. None of these cases show a simple parametric form, and the complexity of the density increases with the number of items n. This motivates the use of non-parametric estimators for modelling preferences over a large number of items.) As n increases, the number of the modes increases and the density surface itself becomes less regular. Intuitively, different probability mass regions correspond to different types of judges. For example in movie preferences probability modes may correspond to genre as fans of drama, action, comedy, etc. having similar preferences.

We therefore propose to estimate the underlying distribution p on permutations extending non-parametric kernel smoothing on rankings (Lebanon and Mao, 2008). The standard kernel smoothing formula applies to the permutation setting as

$$\hat{p}(\pi) = \frac{1}{m} \sum_{i=1}^{m} K_h\{T(\pi, \pi_i)\}$$

where $\pi_1, \ldots, \pi_m \sim^{\text{IID}} p$, T is a distance on permutations such as Kendall's τ-distance and $K_h(r) = h^{-1}K(r/h)$ is a normalized unimodal function. In the case at hand, however, the observed preferences π_i as well as π are replaced with permutations sets $S_1, \ldots, S_m$, R representing incomplete tied preferences

$$\hat{p}(R) = \sum_{\pi \in R} \hat{p}(\pi) = \frac{1}{m} \sum_{i=1}^{m} \sum_{\pi \in R} \sum_{\sigma \in S_i} q(\sigma|S_i) K_h\{T(\pi, \sigma)\} \tag{6}$$

where $q(\sigma|S_i)$ serves as a surrogate for the unknown $p(\sigma|S_i) \propto I(\sigma \in S_i)\ p(\sigma)$ (see equation (4)). For example, a uniform $q(\sigma|S_i)$ indicates that, given a censored ranking corresponding to a user's ratings, the precise permutation of preferences is uniform over the set of compatible permutations.

Selecting $q(\sigma|S_i) = p(\sigma|S_i)$ would lead to consistent estimation of $p(R)$ in the limit $h \to 0$, $m \to \infty$ assuming positive $p(\pi)$ and $p(S)$ by appealing to standard kernel density consistency results found in Wand and Jones (1995). Such a selection, however, is generally impossible since $p(\pi)$ and therefore $p(\sigma|S_i)$ are unknown.

In general the specific choice of the surrogate $q(\sigma|S)$ is important as it may influence the estimated probabilities. Furthermore, it may cause underestimation or overestimation of $\hat{p}(R)$ in the limit of large data. An exception occurs when the sets $S_1, \ldots, S_m$ are either subsets of R or disjoint from R. In this case $\lim_{h\to 0}\{K_h(\pi,\sigma)\} = I(\pi = \sigma)$, resulting in the following limit (with probability 1 by the strong law of large numbers):

$$\lim_{m\to\infty}\lim_{h\to 0}\{\hat{p}(R)\} = \lim_{m\to\infty}\left\{\frac{1}{m}\sum_{i=1}^{m} I(S_i \subset R)\sum_{\sigma\in S_i} q(\sigma|S_i)\right\} \tag{7}$$

$$= \lim_{m\to\infty}\left\{\frac{1}{m}\sum_{i=1}^{m} I(S_i \subset R)\right\} = \lim_{m\to\infty}\left\{\frac{1}{m}\sum_{i=1}^{m} I(\pi_i \in R)\right\} = p(R). \tag{8}$$

Thus, if our data are comprised of preferences S_i that are either disjoint or a subset of R we have consistency *regardless* of the choice of the surrogate q. Such a situation is more realistic when the preference R involves a small number of items and the preferences S_i, $i = 1, \ldots, m$, involve a larger number of items. This is often so for recommendation systems where individuals report preferences over 10–100 items and we are mostly interested in estimating probabilities of preferences over fewer items such as $i \prec j, k$ or $i \prec j, k \prec l$ (see Section 4). Nevertheless, real world recommendation systems data may show sparsity patterns that violate this assumption. In such cases the method proposed may still be used for engineering purposes but the consistency result no longer applies.

The main difficulty with the estimator above is the computation of

$$\sum_{\pi\in R}\sum_{\sigma\in S_i} q(\sigma|S_i) K_h\{T(\pi,\sigma)\}.$$

In the case of high n and only a few observed items k the sets S_i and R grow factorially as $(n-k)!$ making a naive computation of equation (6) intractable for all except the smallest n. In the next section we explore efficient computations of these sums for a triangular kernel K_h and a uniform $q(\pi|S)$.

3. Computationally efficient kernel smoothing

In previous work (Lebanon and Mao, 2008) the estimator (6) is proposed for tied (but complete) rankings. That work derives closed form expressions and efficient computation for estimator (6) by assuming a Mallows kernel (Mallows, 1957):

$$K_h\{T(\pi,\sigma)\} = \exp\left\{-\frac{T(\pi,\sigma)}{h}\right\}\prod_{j=1}^{n}\frac{1-\exp(-1/h)}{1-\exp(-j/h)} \tag{9}$$

where T is Kendall's τ-distance on permutations (below $I(x) = 1$ for $x > 0$ and 0 otherwise)

$$T(\pi,\sigma) = \sum_{i=1}^{n-1}\sum_{l>i} I\{\pi\,\sigma^{-1}(i) - \pi\,\sigma^{-1}(l)\}. \tag{10}$$

Unfortunately these simplifications do not carry over to the case of incomplete rankings where the sets of consistent permutations $S_1, \ldots, S_m$ are not cosets of the symmetric group. As a result the problem of probability estimation in recommendation systems where n is high and many items are missing is particularly challenging. However, as we show below, replacing the Mallows kernel (9) with a triangular kernel leads to efficient computation in the case of ties and incomplete rankings. Specifically, the triangular kernel on permutation is

$$K_h\{T(\pi,\sigma)\} = \{1 - h^{-1}\,T(\pi,\sigma)\}\,I\{h - T(\pi,\sigma)\}/C \tag{11}$$

where the bandwidth parameter h represents both the support (the kernel is 0 for all larger distances) and the inverse slope of the triangle (Fig. 2). As we show below the normalization term C is a function of h and may be efficiently computed by using generating functions. Table 1 displays the linear decay of equation (11) for the simple case of permutations over $n = 3$ items.

3.1. Combinatorial generating function

Generating functions, a tool from enumerative combinatorics, allow efficient computation of estimator (6) by concisely expressing the distribution of distances between permutations. Kendall's τ $T(\pi,\sigma)$ is the total number of discordant pairs or inversions between π and σ (Stanley, 2000) and thus its computation becomes a combinatorial counting problem. We associate the following generating function with the symmetric group of order n permutations:

$$G_n(z) = \prod_{j=1}^{n-1} \sum_{k=0}^{j} z^k. \tag{12}$$

As shown for example in Stanley (2000) the coefficient of z^k of $G_n(z)$, which we denote as $[z^k]G_n(z)$, corresponds to the number of permutations σ for which $T(\sigma,\pi') = k$. For example, the distribution of Kendall's τ $T(\cdot,\pi')$ over all permutations of three items is described by $G_3(z) = (1+z)(1+z+z^2) = 1z^0 + 2z^1 + 2z^2 + 1z^3$, i.e. there is one permutation σ with $T(\sigma,\pi') = 0$, two permutations σ with $T(\sigma,\pi') = 1$, two with $T(\sigma,\pi') = 2$ and one with $T(\sigma,\pi') = 3$. Another important generating function is

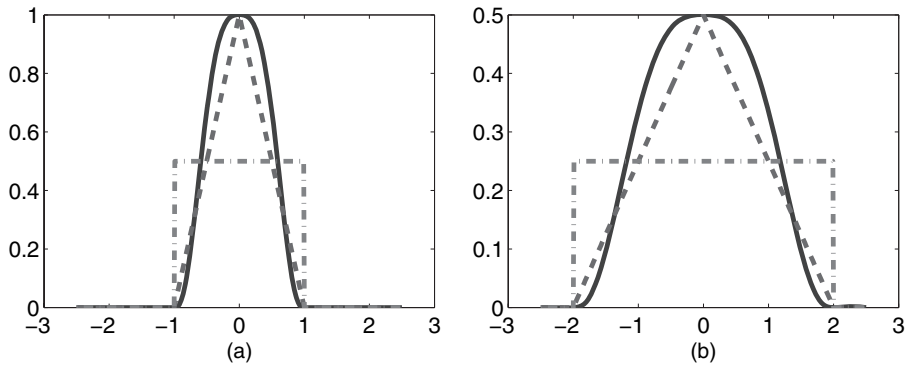


Fig. 2. Tricube (——), triangular (– – –) and uniform (•–•–•) kernels on $\mathbb{R}$ with bandwidth (a) $h = 1$ and (b) $h = 2$

Table 1. Triangular kernel on permutations ($n = 3$)

	$K_3(\cdot, 1 \prec 2 \prec 3)$	$K_5(\cdot, 1 \prec 2 \prec 3)$
$1 \prec 2 \prec 3$	0.50	0.33
$1 \prec 3 \prec 2$	0.25	0.22
$2 \prec 1 \prec 3$	0.25	0.22
$3 \prec 1 \prec 2$	0	0.11
$2 \prec 3 \prec 1$	0	0.11
$3 \prec 2 \prec 1$	0	0

$$H_n(z) = \frac{G_n(z)}{1-z} = (1 + z + z^2 + z^3 + \ldots) G_n(z)$$

where $[z^k] H_n(z)$ represents the number of permutations σ for which $T(\sigma, \pi') \leqslant k$.

Proposition 1. The normalization term $C(h)$ is given by $C(h) = [z^h] H_n(z) - h^{-1} [z^{h-1}] \times G_n'(z)/(1-z)$.

Proof. The proof factors the non-normalized triangular kernel $CK_h(\pi, \sigma)$ to $I\{h - T(\pi, \sigma)\}$ and $h^{-1}\, T(\pi, \sigma)\, I\{h - T(\pi, \sigma)\}$ and makes the following observations. First we note that summing the first factor over all permutations may be counted by $[z^h]\, H_n(z)$. The second observation is that $[z^{k-1}]\, G_n'(z)$ is the number of permutations σ for which $T(\sigma, \pi') = k$, multiplied by k. Since we want to sum over that quantity for all permutations whose distance is less than h we extract the $(h-1)$th coefficient of the generating function $G_n'(z) \Sigma_{k \geqslant 0}\, z^k = G_n'(z)/(1-z)$. We thus have

$$C = \sum_{\sigma : T(\pi', \sigma) \leqslant h} 1 - h^{-1} \sum_{\sigma : T(\pi', \sigma) \leqslant h} T(\pi', \sigma) = [z^h] H_n(z) - h^{-1} [z^{h-1}] \frac{G_n'(z)}{1-z}.$$

Proposition 2. The complexity of computing $C(h)$ is $O(n^4)$.

Proof. We describe a dynamic programming algorithm to compute the coefficients of G_n by recursively computing the coefficients of G_k from the coefficients of $G_{k-1}, k = 1, \ldots, n$. The generating function $G_k(z)$ has $k(k+1)/2$ non-zero coefficients and computing each of them (using the coefficients of G_{k-1}) takes $O(k)$. We thus have $O(k^3)$ to compute G_k from G_{k-1} which implies $O(n^4)$ to compute G_k, $n = 1, \ldots, n$. We conclude the proof by noting that once the coefficients of G_n have been computed the coefficients of $H_n(z)$ and $G_n(z)/(1-z)$ are computable in $O(n^2)$ as these are simply cumulative weighted sums of the coefficients of G_n.

Note that computing $C(h)$ for one or many h-values may be done off line before the arrival of the rankings and the need to compute the estimated probabilities.

Denoting by k the number of items that are ranked in either S or R or both, the computation of $\hat{p}(\pi)$ in equation (6) requires $O(k^2)$ on-line and $O(n^4)$ off-line complexity if either non-zero smoothing is performed over the entire data, i.e. $\max_{\pi \in R} \max_{i=1}^{n} \max_{\sigma \in S_i} \{T(\sigma, \pi)\} < h$ or, alternatively, we use the modified triangular kernel $K_h^*(\pi, \sigma) \propto (1 - h^{-1}) T(\pi, \sigma)$ which is allowed to take negative values for the most distant permutations (normalization still applies though).

Proposition 3. For two sets of permutations S and R corresponding to tied incomplete rankings

$$\frac{1}{|S||R|} \sum_{\pi \in S} \sum_{\sigma \in R} T(\pi, \sigma) = \frac{n(n-1)}{4} - \frac{1}{2} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \{1 - 2\, p_{ij}(S)\} \{1 - 2\, p_{ij}(R)\}, \tag{13}$$

$$p_{ij}(U) = \begin{cases} I\{\tau_U(j) - \tau_U(i)\} & i \text{ and } j \text{ are ranked in } U \text{ with } \tau_U(i) \neq \tau_U(j), \\ 1 - \dfrac{\tau_U(i) + \{\phi_U(i) - 1\}/2}{k+1} & \text{only } i \text{ is ranked in } U, \\ \dfrac{\tau_U(j) + \{\phi_U(j) - 1\}/2}{k+1} & \text{only } j \text{ is ranked in } U, \\ \frac{1}{2} & \text{otherwise} \end{cases}$$

with $\tau_U(i) = \min_{\pi \in U} \{\pi(i)\}$ and $\phi_U(i)$ being the number of items that are tied to i in U.

Proof. We note that equation (13) is an expectation with respect to the uniform measure. We thus start by computing the probability $p_{ij}(U)$ that i is preferred to j for $U = S$ and $U = R$ under the uniform measure. Five scenarios exist for each of $p_{ij}(U)$ corresponding to whether each of i and j are ranked by S and R. Starting with the case that i is not ranked and j is ranked, we note that i is equally likely to be preferred to any item or to be preferred to. Given the uniform distribution over compatible rankings item j is equally likely to appear in positions $\tau_U(j), \ldots, \tau_U(j) + \phi_U(j) - 1$. Thus

$$p_{ij} = \frac{1}{\phi_U(j)} \frac{\tau_U(j)}{k+1} + \ldots + \frac{1}{\phi_U(j)} \frac{\tau_U(j) + \phi_U(j) - 1}{k+1} = \frac{\tau_U(j) + \{\phi_U(j) - 1\}/2}{k+1}. \tag{14}$$

Similarly, if j is unknown and i is known then $p_{ij} + p_{ji} = 1$. If both i and j are unknown either ordering must be equally likely given the uniform distribution, making $p_{ij} = \frac{1}{2}$. Finally, if both i and j are known $p_{ij} = 1, \frac{1}{2}, 0$ depending on their preference. Given p_{ij}, linearity of expectation, and the independence between rankings, the change in the expected number of inversions relative to the uniform expectation $n(n-1)/4$ can be found by considering each pair separately:

$$\begin{aligned} E\{T(i,j)\} &= \tfrac{1}{2} P(i \text{ and } j \text{ disagree}) - \tfrac{1}{2} P(i \text{ and } j \text{ agree}) \\ &= \tfrac{1}{2}[p_{ij}(\sigma)\{1 - p_{ij}(\pi)\} + \{1 - p_{ij}(\sigma)\} p_{ij}(\pi) \\ &\quad - p_{ij}(\sigma)\, p_{ij}(\pi) - \{1 - p_{ij}(\sigma)\}\{1 - p_{ij}(\pi)\}] \\ &= -\tfrac{1}{2}\{1 - 2\, p_{ij}(\sigma)\}\{1 - 2 p_{ij}(\pi)\}. \end{aligned}$$

Summing the $n(n-1)/2$ components yields the desired quantity.

Corollary 1. Denoting the number of items ranked by either S or R or both as k, and assuming either $h > \max_{\pi \in R} \max_{i=1}^{n} \max_{\sigma \in S_i} \{T(\sigma, \pi)\}$ or that the modified triangular kernel $K_h^*(\pi, \sigma) \propto (1 - h^{-1})\, T(\pi, \sigma)$ is used, the complexity of computing $\hat{p}(R)$ in equation (6) (assuming uniform $q(\pi | S_i)$) is $O(mk^2)$ on line and $O(n^4)$ off line.

Proof. The proof follows from noting that equation (6) reduces to $O(n^4)$ off-line computation of the normalization term, $O(k^2)$ on-line computation of the form equation (13) and $O(m)$ computation of the final summation.

4. Applications and case-studies

We divide our experimental study into three parts. In the first we examine the task of predicting probabilities. The remaining two parts use these probabilities for rank prediction and rule discovery. Motivation for the multiple-evaluation paradigms that are used can be found in both the probabilistic nature of the estimated rankings and the widely acknowledged difficulty in the evaluation of recommender systems (Herlocker *et al.*, 2004).

In our experiments we used three data sets. The Movielens data set (`http://www.grouplens.org`) contains 1 million ratings from 6040 users over 3952 movies on a scale of 1–5. The EachMovie data set (`http://www.grouplens.org/node/76`) contains 2.6 million ratings from 74424 users over 1648 movies on a scale of 0–5. The Netflix data set (`http://www.netflixprize.com/community`) contains 100 million movie ratings from 480 189 users over 17 770 movies on a scale of 1–5. In all of these data sets users typically rated only a small number of items. Histograms of the distribution of the number of votes per user, number of votes per item and vote distribution appear in Fig. 3.

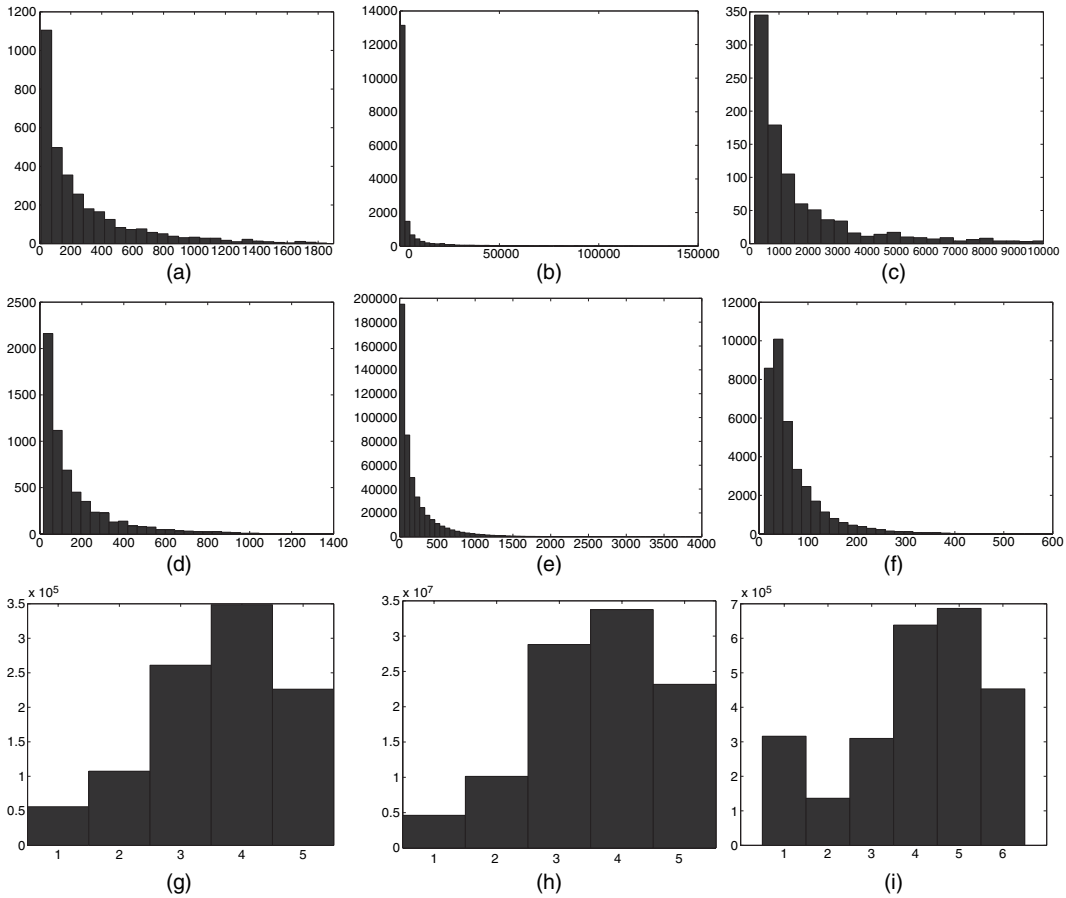


Fig. 3. Histograms of (a)–(c) the number of user votes per movie, (d)–(f) the number of movies ranked per user and (g)–(i) votes aggregated over all users and movies: (a), (d), (g) Movielens; (b), (e), (h) Netflix; (c), (f), (i) EachMovie

4.1. Estimating probabilities

We consider here the task of estimating $\hat{p}(R)$ where R is a set of permutations corresponding to a tied incomplete ranking. Such estimates may be used to compute conditional estimates $\hat{P}(R|S_{m+1})$ which are used to predict which augmentations R of S_{m+1} are highly probable. For example, given an observed preference $3 \prec 2 \prec 5$ we may want to compute $\hat{p}(8 \prec 3 \prec 2 \prec 5 | 3 \prec 2 \prec 5) = \hat{p}(8 \prec 3 \prec 2 \prec 5)/\hat{p}(3 \prec 2 \prec 5)$ to see whether item 8 should be recommended to the user.

For simplicity we focus in this section on probabilities of simple events such as $i \prec j$ or $i \prec j \prec k$. The next section deals with more complex events. In our experiment, we estimate the probability of $i \prec j$ for the $n = 53$ most rated movies in Netflix and $m = 10000$ users who rate most of these movies. The probability matrix of the pairs is shown in Fig. 4 where each cell corresponds to the probability of preference between a pair of movies determined by row j and column i. In Fig. 4(a) the rows and columns are ordered by average probability of a movie being preferred to others $r(i) = \Sigma_j \hat{p}(i \prec j)/n$ with the most preferred movie in the first row and column (Fig. 4(c) indicates the ordering according to $r(i)$). It is interesting to note the high level of correlation between the average probabilities and the pairwise probabilities as indicated by the uniform colour gradient. In Fig. 4(b) the movies were ordered first by popularity of genres and then by

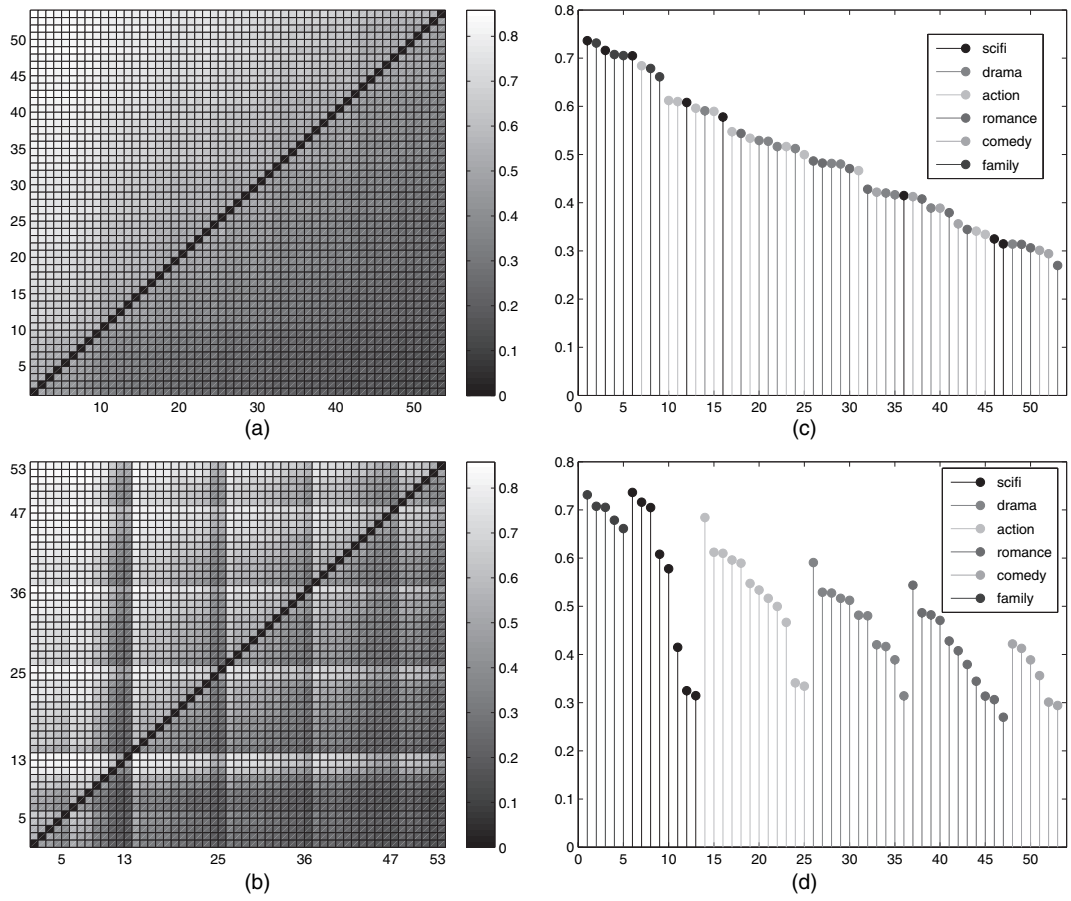


Fig. 4. (a), (b) Estimated probability that movie i is preferred to movie j and (c), (d) plot of $r(i) = \Sigma_j \hat{p}(i \prec j)/n$ for all movies grouped by genre: the movies were ordered by (a), (c) $r(i)$ and (b), (d) first by popularity of genres and then by $r(i)$

$r(i)$. Fig. 4(d) indicates that ordering. The names, genres and both orderings of all 53 movies appear in Table 2.

The three highest movies in terms of $r(i)$ are *Lord of the Rings: the Return of the King*, *Finding Nemo* and *Lord of the Rings: the Two Towers*. The three lowest movies are *Maid in Manhattan*, *Anger Management* and *The Royal Tenenbaums*. Examining the genre (the colours in Figs 4(c) and 4(d)) we see that family and science fiction are generally preferred to others movies whereas comedy and romance generally receive lower preferences. The drama and action genres are somewhere in the middle.

Also interesting is the variance of the movie preferences within specific genres. Family movies are generally preferred to almost all other movies. Science fiction movies, in contrast, enjoy high preference overall but exhibit a larger amount of variability as a few movies are among the least preferred. Similarly, the preference probabilities of action movies are widely spread with some movies being preferred to others and others being less preferred. More specifically (see Fig. 4(b)) we see that the decay of $r(i)$ within genres is linear for family and romance and non-linear for science fiction, action, drama and comedy. In these last three genres there are a few really 'bad'

Table 2. Information on the 53 most rated movies of Netflix†

Movie	*Genre*	*Order1*	*Order2*	*Mean*
Finding Nemo	6	2	1	4.27
Shrek	6	4	2	4.07
The Incredibles	6	5	3	4.06
Monsters, Inc.	6	8	4	4.01
Shrek II	6	9	5	4.13
Lord of the Rings: the Return of the King	1	1	6	4.40
Lord of the Rings: the Two Towers	1	3	7	4.41
Lord of the Rings: the Fellowship of the Ring	1	6	8	4.42
Spider-Man II	1	12	9	3.99
Spider-Man	1	16	10	3.85
The Day After Tomorrow	1	36	11	3.43
Tomb Raider	1	46	12	2.86
Men in Black II	1	47	13	3.04
Pirates of the Caribbean I	3	7	14	4.29
The Last Samurai	3	10	15	3.94
Man on Fire	3	11	16	3.84
The Bourne Identity	3	13	17	3.99
The Bourne Supremacy	3	15	18	3.91
National Treasure	3	17	19	3.53
The Italian Job	3	19	20	3.75
Kill Bill II	3	23	21	3.47
Kill Bill I	3	25	22	3.60
Minority Report	3	31	23	3.61
S.W.A.T.	3	44	24	3.09
The Fast and the Furious	3	45	25	2.84
Ocean's Eleven	2	14	26	3.98
I, Robot	2	20	27	3.72
Mystic River	2	21	28	3.54
Troy	2	22	29	3.61
Catch Me if You Can	2	24	30	3.73
Big Fish	2	28	31	3.35
Collateral	2	29	32	3.60
John Q	2	34	33	3.07
Pearl Harbor	2	35	34	3.23
Swordfish	2	39	35	3.22
Lost in Translation	2	48	36	2.56
50 First Dates	4	18	37	3.76
My Big Fat Greek Wedding	4	26	38	3.60
Something's Gotta Give	4	27	39	3.43
The Terminal	4	30	40	3.47
How to Lose a Guy in 10 Days	4	32	41	3.33
Sweet Home Alabama	4	38	42	3.29
Sideways	4	41	43	2.54
Two Weeks Notice	4	43	44	3.11
Mr. Deeds	4	49	45	2.92
The Wedding Planner	4	50	46	2.71
Maid in Manhattan	4	53	47	2.52
The School of Rock	5	33	48	3.33
Bruce Almighty	5	37	49	3.51
Dodgeball: a True Underdog Story	5	40	50	3.19
Napoleon Dynamite	5	42	51	2.57
The Royal Tenenbaums	5	51	52	2.39
Anger Management	5	52	53	3.03

†Columns are movie titles, genres, order1 (the ordering in the upper row of Fig. 4), order2 (the ordering in the bottom row of Fig. 4) and average ratings. Genres indicated by numbers from 1 to 6 represent science fiction, drama, action, romance, comedy and family. The correlation between the average ratings and the average probabilities of being preferred to others, $r(i) = \Sigma_j \hat{p}(i \prec j)/n$, is 0.93.

movies that are substantially lower than the rest of the curve. Table 2 shows the full information including titles, genres and orderings of the 53 most rated movies in Netflix.

We plot the individual values of $\hat{p}(i \prec j)$ for three movies: *Shrek* (family), *Catch Me if You Can* (drama) and *Napoleon Dynamite* (comedy) (Fig. 5). Comparing the three stem plots we observe that *Shrek* is preferred to almost all other movies, *Napoleon Dynamite* is less preferred than most other movies and *Catch Me if You Can* is preferred to some other movies but less preferred than others. Also interesting is the linear increase of the stem plots for *Catch Me if You Can* and *Napoleon Dynamite* and the non-linear increase of the stem plot for *Shrek*. This is probably a result of the fact that for very popular movies there are only a few comparable movies with the rest being very likely to be less preferred movies ($\hat{p}(i \prec j)$ close to 1).

In a second experiment (Fig. 6) we compare the predictive behaviour of the kernel smoothing estimator with that of a parametric model (Mallows model) and the empirical measure (the frequency that the event occurs in the m samples). We evaluate the predictive performance of a probability estimator by separating the data into two parts: a training set that is used to construct the estimator and a testing set that is used for evaluation via its log-likelihood. A higher test set log-likelihood indicates that the model assigns high probability to events that occurred. Mathematically, this corresponds to approximating the Kullback–Leibler divergence between

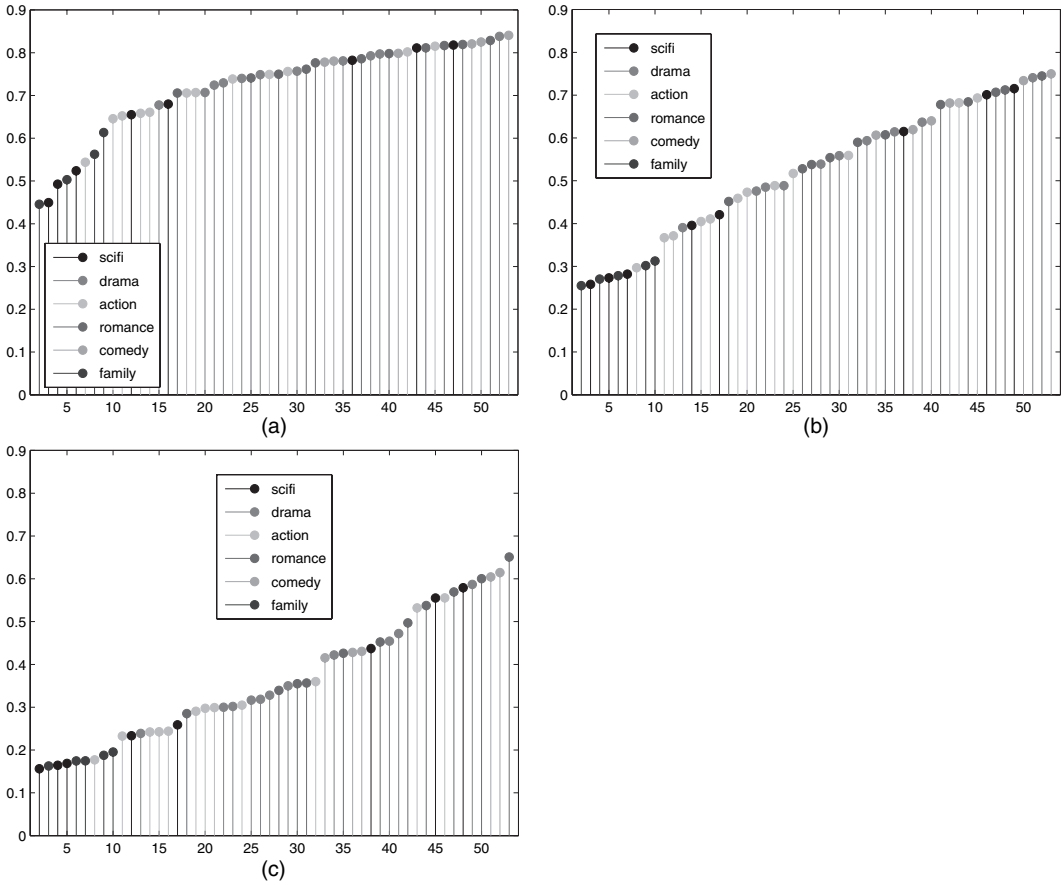


Fig. 5. Value $\hat{p}(i \prec j)$ for all j for (a) *Shrek*, (b) *Catch Me if You Can* and (c) *Napoleon Dynamite*

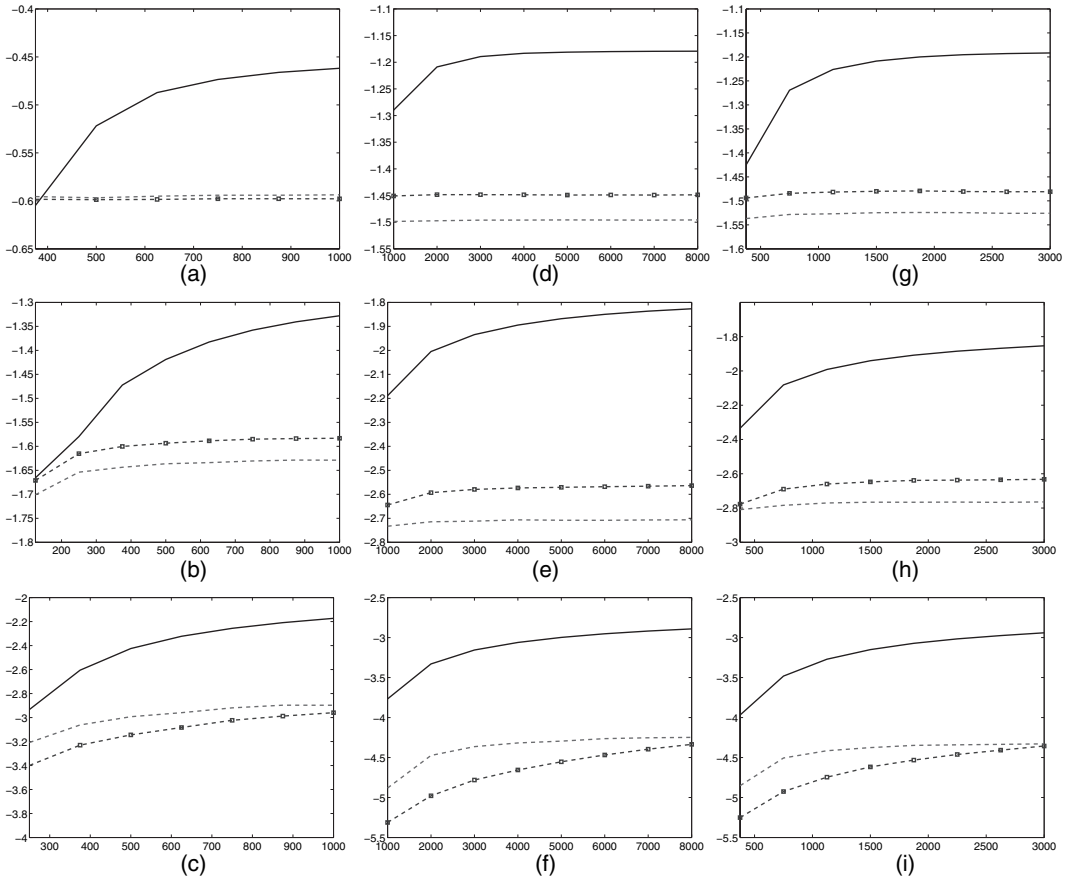


Fig. 6. Test set log-likelihood for kernel smoothing (———), Mallows's model (-------) and the empirical measure (−□−) with respect to training size m (x-axis) for a small number of items (a) $n = 2$, (b) $n = 3$ and (c) $n = 4$ for the Movielens data set, (d) $n = 3$, (e) $n = 4$ and (f) $n = 5$ for the Netflix data set and (g) $n = 3$, (h) $n = 4$ and (i) $n = 5$ for the EachMovie data set: both the Mallows model (which is also intractable for large n, which is why $n \leqslant 5$ in the experiment) and the empirical measure perform worse than the kernel estimator $\hat{p}$

nature and the model. Since the Mallows model is intractable for large n we chose in this experiment small values of n: 3, 4 and 5.

We observe that the kernel estimator consistently achieves higher test set log-likelihoods than the Mallows model and the empirical measure. The former is due to the breakdown of parametric assumptions as indicated by Fig. 1 (note that this happens even for n as low as 3). This is due to the superior statistical performance of the kernel estimator over the empirical measure.

4.2. Rank prediction

Our task here is to predict ranking of new unseen items for users. We follow the standard procedure in collaborative filtering: the set of users is partitioned into two sets: a training set and a testing set. For each of the test set users we further split the observed items into two sets: one set used for estimating preferences (together with the preferences of the training set users) and the second set to evaluate the performance of the prediction (Pennock *et al.*, 2000), i.e. a

probability density is estimated by applying the estimator to the training data and this density is then used to predict rankings for a test user conditional on the test user's observed preferences. Given a loss function $L(i, j)$ which measures the loss of predicting rank i when the true rank is j (rank here refers to the number of sets of equivalent items that are more or less preferred than the current item) we evaluate a prediction rule by the expected loss. We focus on three loss functions: $L_0(i, j) = 0$ if $i = j$ and $L_0(i, j) = 1$ otherwise, $L_1(i, j) = |i - j|$ which reduces to the standard collaborative filtering evaluation technique that was described in Pennock *et al.* (2000) and an asymmetric loss function (rows correspond to the estimated number of stars (0–5) and columns to actual number of stars (0–5):

$$L_{\mathrm{e}} = \begin{pmatrix} 0 & 0 & 0 & 3 & 4 & 5 \\ 0 & 0 & 0 & 2 & 3 & 4 \\ 0 & 0 & 0 & 1 & 2 & 3 \\ 9 & 4 & 1.5 & 0 & 0 & 0 \\ 12 & 6 & 3 & 0 & 0 & 0 \\ 15 & 8 & 4.5 & 0 & 0 & 0 \end{pmatrix}. \tag{15}$$

In contrast with the L_0- and L_1-loss, L_{e} captures the fact that recommending bad movies as good movies is worse than recommending good movies as bad.

For example, consider a test user whose observed preference is $3 \prec 4, 5, 6 \prec 10, 11, 12 \prec 23 \prec 40, 50, 60 \prec 100, 101$. We may withhold the preferences of items 4 and 11 for evaluation purposes. The recommendation systems then predict a rank of 1 for item 4 and a rank of 4 for item 11. Since the true ranking of these items are 2 and 3 the absolute value losses are $|1 - 2| = 1$ and $|3 - 4| = 1$ respectively.

In our experiment, we use the kernel estimator $\hat{p}$ to predict ranks that minimize the posterior loss and thus adapts to customized loss functions such as L_{e}. The prediction in this case is a refinement $\delta(A)$ of the input ranking A which seeks to approximate the true preference B, i.e. the loss function $L\{\delta(A), B\}$ quantifies the adverse effect of recommending according to the rule $A \to \delta(A)$. Specifically, assuming that an appropriate loss function is given we select the prediction rule δ^* that minimizes the posterior loss

$$\delta^*(A) = \arg\min_{Z \in \mathcal{Z}} [E_{\hat{p}(B|A)}\{L(\mathcal{Z}, B)\}] \tag{16}$$

where $\mathcal{Z}$ is a set of potential refinements of A under consideration. This is an advantage of a probabilistic modelling approach over more *ad hoc* rule-based recommendation systems.

Fig. 7 compares the performance of our estimator with several standard baselines in the collaborative filtering literature: two older memory-based methods vector similarity and correlation in Breese *et al.* (1998) and a recent state of the art non-negative matrix factorization (Lawrence and Urtasun, 2009). The kernel smoothing estimate performed similarly to the state of the art estimator but substantially better than the memory-based methods to which it is functionally similar. Fig. 8 shows the kernel bandwidth selection via cross-validation using the L_1-loss.

4.3. Rule discovery

In the third task, we used the estimator $\hat{p}$ to detect noteworthy association rules of the type $i \prec j \Rightarrow k \prec l$ (if i is preferred to j then probably k is preferred to l). Such association rules are important for both business analytics (devising marketing and manufacturing strategies) and recommendation system engineering. Specifically, we used $\hat{p}$ to select sets of four items i, j, k

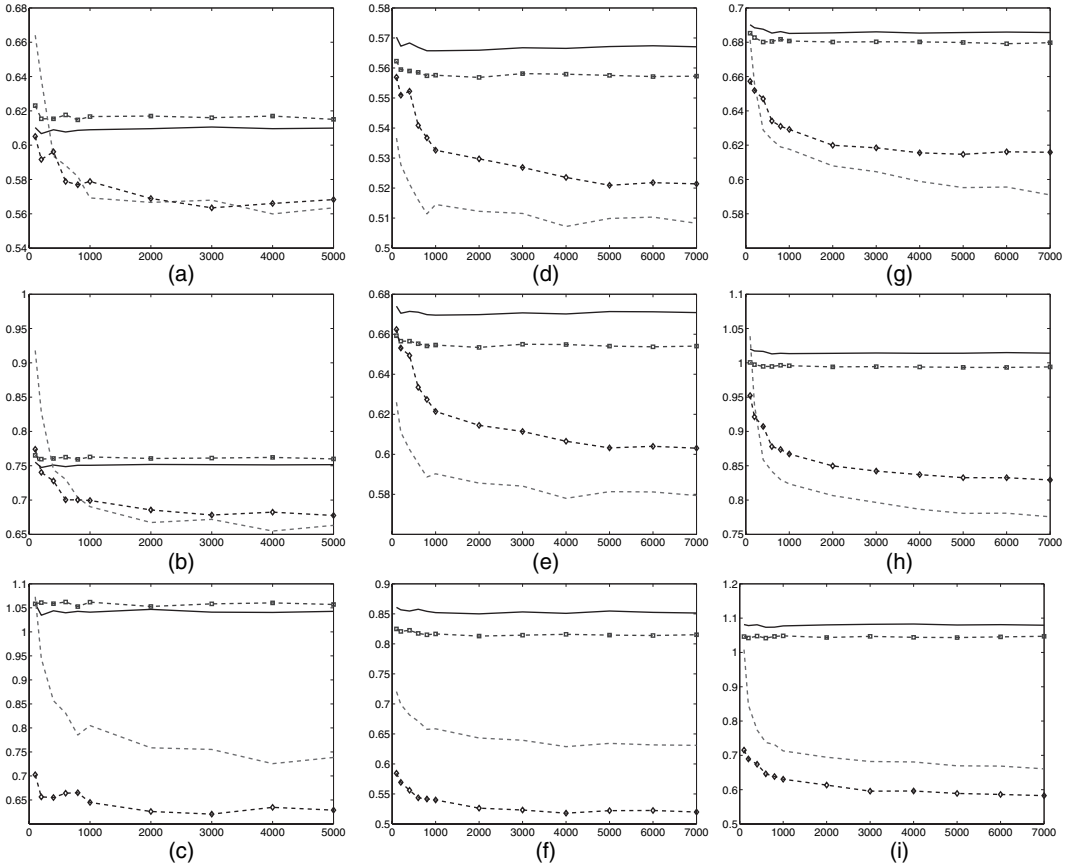


Fig. 7. Prediction loss with respect to training size on (a)–(c) the Movielens data set (6040 users over 3952 movies), (d)–(f) Netflix (10000 users over 800 movies) and (g)–(i) EachMovie (10000 users over 1000 movies) (the kernel smoothing estimate performed similarly to the state of the art non-negative matrix factorization but substantially better than the memory-based methods to which it is functionally similar) (———, vector similarity; ■, correlation; -------, non-negative matrix factorization; ♦, rank): (a), (d), (g) 0–1 loss L_0; (b), (e), (h) L_1-loss; (c), (f), (i) asymmetric loss L_e

and l for which the mutual information $I(i \prec j; k \prec l)$ is maximized. The mutual information is

$$I(i \prec j; k \prec l) = p(i \prec j \cap k \prec l) \log\left\{ \frac{p(i \prec j \cap k \prec l)}{p(i \prec j)p(k \prec l)} \right\} + p(j \prec i \cap k \prec l) \log\left\{ \frac{p(j \prec i \cap k \prec l)}{p(j \prec i)p(k \prec l)} \right\}$$
$$+ p(i \prec j \cap l \prec k) \log\left\{ \frac{p(i \prec j \cap l \prec k)}{p(i \prec j)p(l \prec k)} \right\} + p(j \prec i \cap l \prec k) \log\left\{ \frac{p(j \prec i \cap l \prec k)}{p(j \prec i)p(l \prec k)} \right\}. \tag{17}$$

After these sets have been identified we detected the precise shape of the rule (i.e. $i \prec j \Rightarrow k \prec l$ rather than $j \prec i \Rightarrow k \prec l$ by examining the summands in the mutual information expectation).

Table 3 shows the top 10 rules that were discovered. These rules nicely isolate viewer preferences for genres such as fantasy, romantic comedies, animation and action (note, however, that genre information was not used in the rule discovery). To evaluate the rule discovery process quantitatively we judge a rule $i \prec j \Rightarrow k \prec l$ to be good if i and k are of the same genre and j and l are of the same genre. This quantitative evaluation appears in Fig. 9 where it is contrasted with the same rule discovery process (maximizing mutual information) based on the empirical

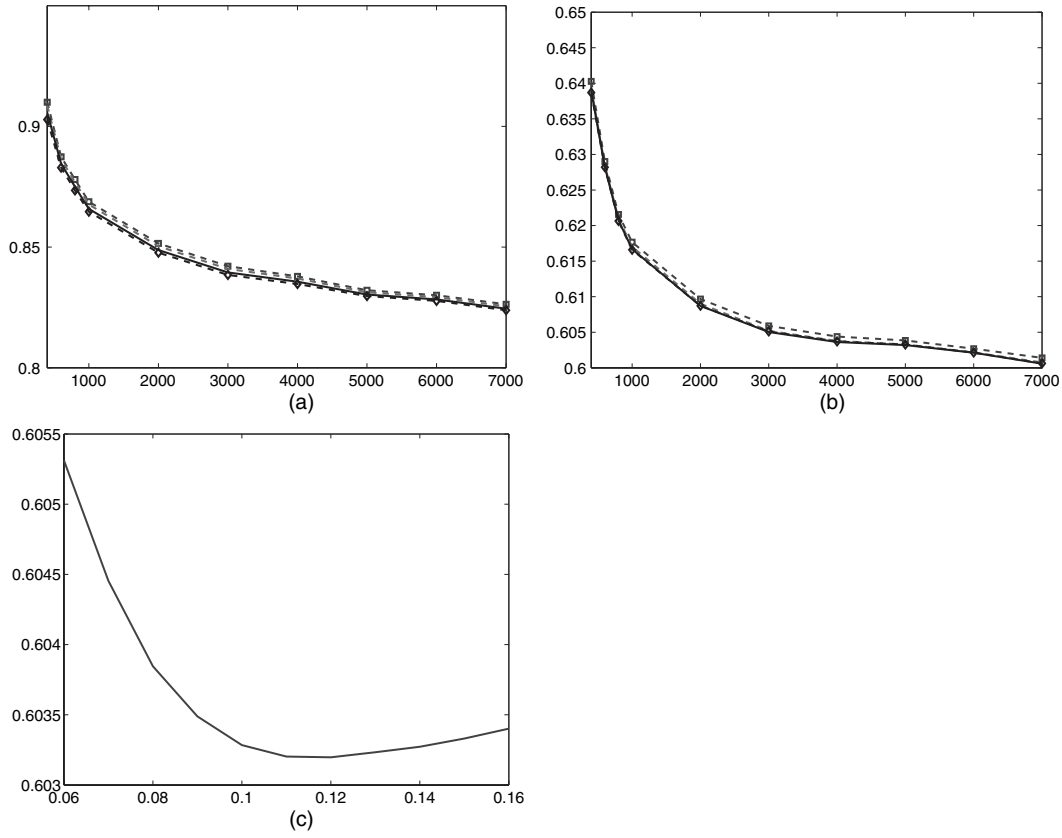


Fig. 8. L_1-loss by using our estimator with various kernel bandwidths h with respect to training size on (a) EachMovie (10000 users over 1000 movies) and (b) Netflix (10000 users over 800 movies) (■, $h = 0.08$; -------, $h = 0.10$; ———, $h = 0.12$; ♦, $h = 0.14$; the x-axis is the training size), and (c) loss by using various kernel bandwidths when the training size is fixed at 5000 for Netflix (the x-axis is the kernel bandwidth)

measure. Although this rule discovery experiment is scored as a success on the basis of the genre of the movies identified, this criterion is only a proxy for movie similarity in the absence of some known measurement. A qualitative examination of these results shows that much more than genre is recovered; for example Table 3, rule 1, states if *Shrek* $\prec$ *Lord of the Rings*: *the Fellowship of the Ring* then *Shrek II* $\prec$ *Lord of the Rings*: *the Return of the King*, i.e. movies of the same series are identified as preferred to other movies in the same series. Table 4 shows the top rules that were discovered within the same genre.

In another rule discovery experiment, we used $\hat{p}$ to detect association rules of the form i ranked highest $\Rightarrow$ j ranked second highest by selecting i and j that maximize the score

$$\frac{p\{\pi(i) = 1, \pi(j) = 2\}}{p\{\pi(i) = 1)\, p(\pi(j) = 2\}}$$

between pairs of movies in the Netflix data. We similarly detected rules of the form i ranked highest $\Rightarrow$ j ranked lowest by maximizing the scores

$$\frac{p\{\pi(i) = 1, \pi(j) = \text{last}\}}{p\{\pi(i) = 1)\, p(\pi(j) = \text{last}\}}$$

between pairs of movies.

Table 3. Top 10 rules discovered by the kernel smoothing estimator on Netflix in terms of maximizing mutual information

Shrek $\prec$ *Lord of the Rings: the Fellowship of the Ring*	$\Rightarrow$	*Shrek II* $\prec$ *Lord of the Rings: the Return of the King*
Shrek $\prec$ *Lord of the Rings: the Fellowship of the Ring*	$\Rightarrow$	*Shrek II* $\prec$ *Lord of the Rings: the Two Towers*
Shrek II $\prec$ *Lord of the Rings: the Fellowship of the Ring*	$\Rightarrow$	*Shrek* $\prec$ *Lord of the Rings: the Return of the King*
Kill Bill II $\prec$ *National Treasure*	$\Rightarrow$	*Kill Bill I* $\prec$ *I, Robot*
Shrek II $\prec$ *Lord of the Rings: the Fellowship of the Ring*	$\Rightarrow$	*Shrek II* $\prec$ *Lord of the Rings: the Two Towers*
Lord of the Rings: the Fellowship of the Ring $\prec$ *Monsters, Inc.*	$\Rightarrow$	*Lord of the Rings: the Two Towers* $\prec$ *Shrek*
National Treasure $\prec$ *Kill Bill II*	$\Rightarrow$	*Pearl Harbor* $\prec$ *Kill Bill I*
Lord of the Rings:the Fellowship of the Ring $\prec$ *Monsters, Inc.*	$\Rightarrow$	*Lord of the Rings: the Return of the King* $\prec$ *Shrek*
How to Lose a Guy in 10 Days $\prec$ *Kill Bill II*	$\Rightarrow$	*50 First Dates* $\prec$ *Kill Bill I*
I, Robot $\prec$ *Kill Bill II*	$\Rightarrow$	*The Day After Tomorrow* $\prec$ *Kill Bill I*

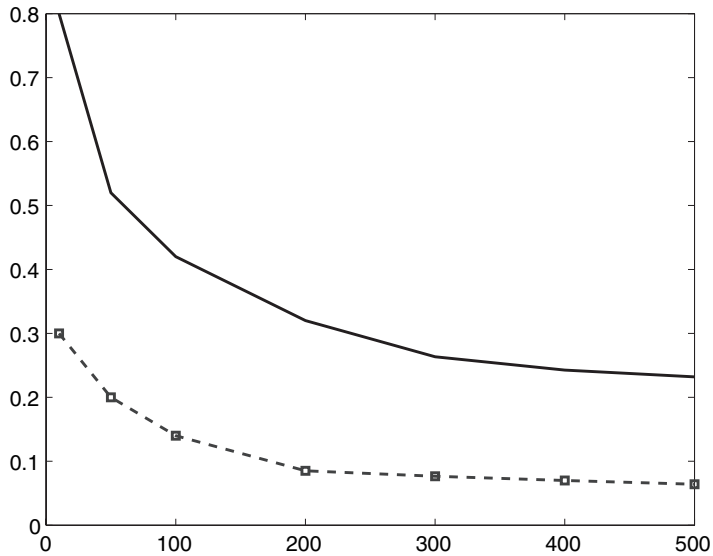


Fig. 9. Quantitative evaluation of the rule discovery (———, kernel; ■, empirical): the x-axis represents the number of rules discovered (i.e. increasing values on the x-axis correspond to selecting additional sets of movies with decreasing mutual information) and the y-axis represents the frequency of good rules in the discovered rule; here a rule $i \prec j \Rightarrow k \prec l$ is considered good if i and k are of the same genre and j and l are of the same genre

Part (a) of Table 5 shows the top nine rules of 100 most rated movies, which nicely represents movie preference of similar type, e.g. romance, comedies and action. Part (b) of Table 5 shows the top nine rules which represent like and dislike of different types of movie, e.g. like of romance leads to dislike of action or thriller. Although the paired movies in part (a) both come from the same genre, examining a specific pair *The Royal Tenenbaums* $\Rightarrow$ *American Beauty* and referring to the Internet movie database descriptions reveals that these movies are described by many common terms such as 'dark humour, depression, deadpan, drugs, husband–wife relationship,...'. To summarize, although the relationships identified are in part judged to be good

Table 4. Rules within the same genre (top three, science fiction; middle three, drama; bottom three, action) discovered by the kernel smoothing estimator on Netflix in terms of maximizing mutual information†

Spider-Man $\prec$ *Lord of the Rings: the Fellowship of the Ring*	$\Rightarrow$	*Spider-Man II* $\prec$ *Lord of the Rings: the Return of the King*
The Day After Tomorrow $\prec$ *Lord of the Rings: the Fellowship of the Ring*	$\Rightarrow$	*Spider-Man II* $\prec$ *Lord of the Rings: the Return of the King*
Men in Black II $\prec$ *Spider-Man*	$\Rightarrow$	*Tomb Raider* $\prec$ *Spider-Man II*
Pearl Harbor $\prec$ *Catch Me if You Can*	$\Rightarrow$	*Troy* $\prec$ *Mystic River*
I, Robot $\prec$ *Catch Me if You Can*	$\Rightarrow$	*Troy* $\prec$ *Ocean's Eleven*
Collateral $\prec$ *I, Robot*	$\Rightarrow$	*Lost in Translation* $\prec$ *Pearl Harbor*
National Treasure $\prec$ *Kill Bill II*	$\Rightarrow$	*S.W.A.T.* $\prec$ *Kill Bill I*
The Fast and the Furious $\prec$ *Kill Bill II*	$\Rightarrow$	*The Italian Job* $\prec$ *Kill Bill I*
The Bourne Supremacy $\prec$ *Man on Fire*	$\Rightarrow$	*The Bourne Identity* $\prec$ *The Last Samurai*

†The rules are in the form of $i \prec j \Rightarrow k \prec l$, where i, j, k and l are of the same genre.

Table 5. Top rules discovered by kernel smoothing estimate on Netflix

(a) Like A $\Rightarrow$ *like B*		
Kill Bill I	$\Rightarrow$	*Kill Bill II*
Maid in Manhattan	$\Rightarrow$	*The Wedding Planner*
Two Weeks Notice	$\Rightarrow$	*Miss Congeniality*
The Royal Tenenbaums	$\Rightarrow$	*Lost in Translation*
The Royal Tenenbaums	$\Rightarrow$	*American Beauty*
The Fast and the Furious	$\Rightarrow$	*Gone in 60 Seconds*
Spider-Man	$\Rightarrow$	*Spider-Man II*
Anger Management	$\Rightarrow$	*Bruce Almighty*
Memento	$\Rightarrow$	*Pulp Fiction*
(b) Like A $\Rightarrow$ *dislike B*		
Maid in Manhattan	$\Rightarrow$	*Pulp Fiction*
Maid in Manhattan	$\Rightarrow$	*Kill Bill I*
How to Lose a Guy in 10 Days	$\Rightarrow$	*Pulp Fiction*
The Royal Tenenbaums	$\Rightarrow$	*Pearl Harbor*
The Wedding Planner	$\Rightarrow$	*The Matrix*
Peal Harbor	$\Rightarrow$	*Memento*
Lost in Translation	$\Rightarrow$	*Pearl Harbor*
The Day After Tomorrow	$\Rightarrow$	*American Beauty*
The Wedding Planner	$\Rightarrow$	*Raiders of the Lost Ark*

or bad by genre, qualitatively these relationships can be seen to be much closer than those obtained by randomly selecting movies from the same genre.

In a third experiment, we used $\hat{p}$ to construct an undirected graph where vertices are items (Netflix movies) and two nodes i and j are connected by an edge if the average score of the rule i ranked highest $\Rightarrow$ j ranked second highest and the rule j ranked highest $\Rightarrow$ i ranked second highest is higher than a certain threshold. Fig. 10 shows the graph for the 100 most rated movies in Netflix (only movies with vertex degree greater than 0 are shown). The clusters in the graph corresponding to vertex colour and numbering were obtained by using a graph partitioning algorithm and the graph is embedded in a two-dimensional plane by using standard graph visualization techniques. Within each of the identified clusters movies are clearly similar with respect to genre, and an even finer separation can be observed when looking at specific clusters. For example, clusters 6 and 9 both contain comedy movies, whereas cluster 6 tends towards slapstick humour and cluster 9 contains romantic comedies.

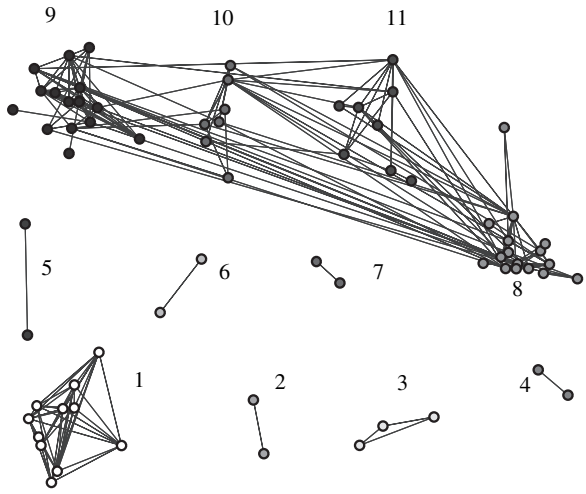


Fig. 10. Graph corresponding to the 100 most rated Netflix movies where edges represent high affinity as determined by the rule discovery process (see the text for more details): 1, *American Beauty*, *Lost in Translation*, *Pulp Fiction*, *Kill Bill I*, *II*, *Memento*, *The Royal Tenenbaums*, *Napoleon Dynamite...*; 2, *Spider-Man*, *Spider-Man II*; 3, *Lord of the Rings I*, *II*, *III*; 4, *The Bourne Identity*, *The Bourne Supremacy*; 5, *Shrek*, *Shrek II*; 6, *Meet the Parents*, *American Pie*; 7, *Indiana Jones and the Last Crusade*, *Raiders of the Lost Ark*; 8, *The Patriot*, *Pearl Harbor*, *Men of Honor*, *John Q*, *The General's Daughter*, *National Treasure*, *Troy*, *The Italian Job...*; 9, *Miss Congeniality*, *Sweet Home Alabama*, *Two Weeks Notice*, *50 First Dates*, *The Wedding Planner*, *Maid in Manhattan*, *Titanic...*; 10, *Men in Black I*, *II*, *Bruce Almighty*, *Anger Management*, *Mr. Deeds*, *Tomb Raider*, *The Fast and the Furious*; 11, *Independence Day*, *Con Air*, *Twister*, *Armageddon*, *The Rock*, *Lethal Weapon IV*, *The Fugitive*, *Air Force One*

5. Summarizing remarks

Estimating distributions from tied and incomplete data is a central task in many applications with perhaps the most obvious being collaborative filtering. An accurate estimator $\hat{p}$ enables going beyond the traditional item–rank prediction task. It can be used to compute probabilities of interest, to make recommendations using loss functions more closely tied to the user experience, to find association rules and to perform a wide range of additional data analysis tasks. We demonstrate the first non-parametric estimator for such data (subject to sampling assumptions in Section 2) that is computationally tractable, i.e. polynomial rather than exponential in n. The computation is made possible by using generating function and dynamic programming techniques.

We examine the behaviour of the estimator $\hat{p}$ in three sets of experiments. The first set of experiments involves estimating probabilities of interest such as $p(i \prec j)$. The second set of experiments involves predicting preferences of held-out items which is directly applicable in recommendation systems. In this task, our estimator outperforms other memory-based methods (to which it is similar functionally) and performs similarly to state of the art methods that are based on non-negative matrix factorization. In the third set of experiments we examined the usage of the estimator in discovering association rules such as $i \prec j \Rightarrow k \prec l$.

From a practical perspective, robustness to departures from the assumptions must be considered, specifically random censoring and consistency. Previous work has demonstrated that the random-censoring assumption may be violated as people tend to rate items that they feel strongly about more frequently than those for which they do not have strong feelings (Marlin and Zemel, 2007). Such a tendency should not have a substantial negative effect on recommendations as attitudes towards polarizing movies will be captured and the use of the notion of compatible sets encourages average ratings for infrequently rated movies. Secondly, although

consistency may not hold in every case, scenarios can be devised which make it very likely, e.g. restricting attention to situations with very large n and only estimating probabilities over a very small number of items. Additionally kernel density estimators tend to flatten peaks and to lift valleys, but the relative values of the probabilities will retain the same ordering probably mitigating the effect on the recommendations. In practice, the empirical performance that is observed over several data sets and in several tasks indicates that any adverse effect based on departures from these assumptions produces an effect that is no larger than that experienced with other state of the art approaches. To summarize, although these assumptions may not always hold, the practical effect is likely to be negligible.

References

Basu, C., Hirsh, H. and Cohen, W. (1998) Recommendation as classification: using social and content-based information in recommendation. In *Proc. 15th Natn. Conf. Artificial Intelligence, Madison*, pp. 714–720. Cambridge: MIT Press.

Breese, J., Heckerman, D. and Kadie, C. (1998) Empirical analysis of predictive algorithms for collaborative filtering. In *Proc. 14th Conf. Uncertainty in Artificial Intelligence, Madison*, pp. 43–52. San Francisco: Morgan Kaufmann.

Cliff, N. and Keats, J. A. (2003) *Ordinal Measurement in the Behavioral Sciences*. Mahwah: Erlbaum.

Diaconis, P. (1988) *Group Representations in Probability and Statistics*. Hayward: Institute of Mathematical Statistics.

Goldberg, K., Roeder, T., Gupta, D. and Perkins, C. (2001) Eigentaste: a constant time collaborative filtering algorithm. *Inform. Retriev.*, **4**, 133–151.

Heckerman, D., Maxwell Chickering, D., Meek, C., Rounthwaite, R. and Kadie, C. (2000) Dependency networks for inference, collaborative filtering, and data visualization. *J. Mach. Learn. Res.*, **1**, 49–75.

Herlocker, J. L., Konstan, J. A., Borchers, A. and Riedl, J. (1999) An algorithmic framework for performing collaborative filtering. In *Proc. 22nd A. Int. Conf. Research and Development in Information Retrieval*, pp. 230–237. New York: Association for Computing Machinery.

Herlocker, J. L., Konstan, J. A., Terveen, L. G. and Riedl, J. T. (2004) Evaluating collaborative filtering recommender systems. *ACM Trans. Inform. Syst.*, **22**, 5–53.

Hofmann, T. (2004) Latent semantic models for collaborative filtering. *ACM Trans. Inform. Syst.*, **22**, 89–115.

Kidwell, P., Lebanon, G. and Cleveland, W. S. (2008) Visualizing incomplete and partially ranked data. *IEEE Trans. Visualizn Comput. Graph.*, **14**, 1356–1363.

Koren, Y. (2010) Factor in the neighbors: scalable and accurate collaborative filtering. *ACM Trans. Knowl. Discov. Data*, **4**, 1–24.

Lawrence, N. D. and Urtasun, R. (2009) Non-linear matrix factorization with gaussian processes. In *Proc. 26th A. Int. Conf. Machine Learning, Montreal*, pp. 601–608. Madison: Omnipress.

Lebanon, G. and Mao, Y. (2008) Non-parametric modeling of partially ranked data. *J. Mach. Learn. Res.*, **9**, 2401–2429.

Mallows, C. L. (1957) Non-null ranking models. *Biometrika*, **44**, 114–130.

Marden, J. I. (1996) *Analyzing and Modeling Rank Data*. London: CRC Press.

Marlin, B. (2004) Modeling user rating profiles for collaborative filtering. In *Advances in Neural Information Processing Systems*, pp. 627–634. Cambridge: MIT Press.

Marlin, B. M. and Zemel, R. S. (2007) Collaborative filtering and the missing at random assumption. In *Proc. 23rd Conf. Uncertainty in Artificial Intelligence, Vancouver*, pp. 50–54. Association for Uncertainty in Artificial Intelligence Press.

Pennock, D. M., Horvitz, E., Lawrence, S. and Giles, C. L. (2000) Collaborative filtering by personality diagnosis: a hybrid memory- and model-based approach. In *Proc. 16th Conf. Uncertainty in Artificial Intelligence, Stanford*, pp. 473–480. San Francisco: Morgan Kaufmann.

Popescul, A., Ungar, L. H., Pennock, D. M. and Lawrence, S. (2001) Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *Proc. 17th Conf. Uncertainty in Artificial Intelligence*.

Rennie, J. D. M. and Srebro, N. (2005) Fast maximum margin matrix factorization for collaborative prediction. In *Proc. 22nd Int. Conf. Machine Learning*, pp. 713–719. New York: Association for Computing Machinery.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J. (1994) Grouplens: an open architecture for collaborative filtering of netnews. In *Proc. Conf. Computer Supported Cooperative Work*, pp. 175–186. New York: Association for Computing Machinery.

Sarwar, B., Karypis, G., Konstan, J. and Reidl, J. (2001) Item-based collaborative filtering recommendation algorithms. In *Proc. Int. Conf. World Wide Web, Hong Kong*, pp. 285–295. New York: Association for Computing Machinery.

Schein, A. I., Popescul, A., Ungar, L. H. and Pennock, D. M. (2002) Methods and metrics for cold-start recommendations. In *Proc. 25th A. Int. Conf. Research and Development in Information Retrieval*, pp. 253–260. New York: Association for Computing Machinery.

Stanley, R. P. (2000) *Enumerative Combinatorics*, vol. 1. Cambridge: Cambridge University Press.

Ungar, L. H. and Foster, D. P. (1998) Clustering methods for collaborative filtering. In *Wrkshp Recommendation Systems, Madison.* American Association for Artificial Intelligence.

Wand, M. P. and Jones, M. C. (1995) *Kernel Smoothing*. London: Chapman and Hall.

Xue, G. R., Lin, C., Yang, Q., Xi, W. S., Zeng, H. J., Yu, Y. and Chen, Z. (2005) Scalable collaborative filtering using cluster-based smoothing. In *Proc. 28th A. Int. Conf. Research and Development in Information Retrieval*, pp. 114–121. New York: Association for Computing Machinery.