Fusion-Based Recommender System

Keshu Zhang
Applied Research Center
Motorola, Inc.
Tempe, AZ, U.S.A.
keshu.zhang@motorola.com

Haifeng Li
Applied Research Center
Motorola, Inc.
Tempe, AZ, U.S.A.
A00050@motorola.com

Abstract – *Recommender systems have become a common way to help people navigate among increasingly more complex selection options. Various recommendation algorithms have been proposed and a lot of business solutions have been built for applications such as selecting books, cinema, video/TV program, restaurants, etc. However, there is no simply best approach due to the high complexity and uncertainty of the problem. Therefore, a recommender system usually includes multiple different recommenders and aggregates their recommendation results through a combiner. In this paper, we discussed combining recommenders in the framework of information fusion theory including rank fusion, decision fusion, Dempster-Shafer fusion and estimation fusion. Experiments results on the benchmark MovieLens dataset [10] show that our proposed methods with fusion techniques leaded to significant performance improvements over the baselines models.*

Keywords: recommender, rank fusion, decison fusion, D-S fusion, estimation fusion, rating

1 Introduction

Recommender systems investigate user preferences to provide suggestions on items for purchase or consume in order to increase sells or to improve user experience. Because they are of great effects on reducing information overload and provide personalized services to users, an abundant of practical applications have been developed, such as Amazon.com recommendation system of books, CDs, and other products [5], movies recommendation service by MovieLens [6], and news filter service at VERSIFI Technologies (formerly AdaptiveInfo.com) [7]. Due to their great difficulties, recommender systems constitutes a problem-rich research area and have attracted many researchers from both industry and academia since the appearance of the first papers on collaborative filtering in the mid-1990s [2], [3], [4].

In most common formulation, the recommendation problem is reduced to the problem of estimating ratings [8] for the items that have not been seen by a user. Intuitively, this estimation is usually based on the ratings given by this user to other items and on some other information. Once we can estimate ratings for the yet unrated items, we can recommend to the user the item(s) with the highest estimated rating(s). In general, the recommendation problem can be formulated as follows: Let C be the set of all users and let S be the set of all possible items that can be recommended, such as books, movies, or restaurants. The space S of possible items can be very large, ranging in hundreds of thousands or even millions of items in some applications. Similarly, the user space can also be very large. Let u be a utility function that measures the usefulness of item s to user c, i.e., $u : C \times S \to R$, where R is a totally ordered set (e.g., nonnegative integers or real numbers within a certain range). Then, for each user $c \in C$, we want to choose such item $x \in S$ that maximizes the user's utility. More formally:

$$\forall c \in C, \quad L_c(x) = \underset{x \in S}{\arg\max} \quad u(c, x) \tag{1}$$

In recommender systems, the utility of an item $u(c, x)$ is usually represented by a rating $r^c(x)$, which indicates how a particular user liked a particular item, e.g., John gave the movie "Harry Potter" the rating of 4 (out of 5). In general, the utility is initially defined only on the items previously rated by the users. Therefore, the recommendation engine should be able to estimate (predict) the ratings of the non-rated movie/user combinations and issue appropriate recommendations based on these predictions. Once the unknown ratings are estimated, actual recommendations of an item to a user are made by selecting the highest rating among all the estimated ratings for that user, according to (1). Alternatively, we can recommend the N best items to a user or a set of users to an item. The new ratings of the not-yet-rated items can be estimated in many different ways using methods from machine learning, approximation theory, and various heuristics.

A variety of techniques have been proposed for performing recommendation, which can be classified into three categories: content-based recommender, knowledge-based recommender, and collaborative filter (for details, see Section 2). Due to the high complexity and uncertainty of the problem, there is no simply best approach. Different recommendation techniques have strengths as well as weakness in practice.

In an attempt to achieve the better (or even best) recommendation results, we would like to make a recommendation by combining the recommendations from a variety of recommenders. By carefully designing the recommendation fusion algorithm, we believe that the hybrid recommender may be able to overcome the shortcoming of each individual recommender and improve the overall system performance. However, recommendation fusion is extremely difficult. For example, recommendation scores from different techniques may have totally different meaning. Some are probability or posteriori probability measure, some are real values without explicit meaning, and some are simply ranks. A direct fusion of those numbers is meaningless. Besides, top recommendation candidate set from different recommenders may be poorly overlapped, which also introduces challenges to build a large and highly confident recommendation set. In this paper, we will discuss how to handle these challenges. And we will also discuss several potential ways (e.g. rank fusion, decision fusion, Dempster-Shafer fusion and estimation fusion) to fuse recommenders, which may employ different recommendation techniques, or the same technique with different feature sets or parameter settings, etc.

The rest of the paper is organized as follows. Section 2 will give a brief review of popular recommendation techniques. In Section 3, 4, 5 and 6 we will discuss the rank fusion, decision fusion, Dempster-Shafer fusion and estimation fusion strategies for recommender system separately. In each section, we will also show for each strategy how we convert or align different recommendation scores to a common baseline where the meaningful comparison or fusion can be performed. The performance comparison on the benchmark datasets MovieLens are studied in Section 7. Section 8 concludes the paper with some directions for further research.

2 Recommendation Techniques

A variety of recommendation techniques have been proposed by researchers and practitioners. In its most common formulation, the recommendation problem is reduced to the problem of estimating ratings for the items that have not been seen by a user. According to the approaches to estimate ratings, recommenders have the following categories in general [9]:

- Content-based recommendations learn the user's taste based on the items that have been visited by the user. The user will be recommended items similar to the ones the user preferred in the past;
- Collaborative recommendations aggregate ratings, recognize commonalities between users or items on the basis of their ratings, and generate new recommendations based on inter-user or inter-item comparisons.
- Knowledge-based recommendations employ the knowledge about the user and the domain. For example, the user could specify explicit information about her preference.

In fact, each recommendation technique has its strengths as well as weakness. Collaborative filters depend on overlap in ratings across users and have difficulty when the space of ratings is sparse. Content-based techniques also have a start-up problem in that they must accumulate enough ratings to build a reliable classifier. They are also limited by the features that are explicitly associated with the objects that they recommend. Knowledge-based recommendation techniques require ahead of time knowledge acquisition by the knowledge engineer. To avoid certain limitations of these methods, it is preferred to combine the basic recommenders in a hybrid approach to gain better performance with fewer of the drawbacks of any individual one. Most commonly, researchers focus on combining collaborative and contend-based methods such that [8]:

- Implementing collaborative and content-based recommenders separately and combining their ratings into a final one,
- Incorporating some content-based characteristics into a collaborative approach,
- Incorporating some collaborative characteristics into a content-based approach, and
- Constructing a general unifying model that incorporates both content-based and collaborative characteristics.

In this paper, we propose several methods to combine recommendation ratings following but going beyond the first aforementioned approach. Instead of just combining collaborative and content-based recommenders, our methods will fuse any types of recommenders. Basically, we have four different fusion strategies of combining ratings: rank fusion, decision fusion, Dempster-Shafer fusion and estimation fusion. In the following sections, we will discuss them in details.

3 Rank Fusion based Recommender System

3.1 Rank and Normalization

Ranking a set of alternatives based on possibly conflicting preferences is a central problem in the areas of voting and social choice theory. Given a set of rankings (a ranking is a linear ordering of a set of items), the task of rank fusion is the problem of combining these lists in such a way to optimize the performance of the combination [11]. The rank fusion problem is encountered in many settings, one prominent of which is metasearch. It deals with the problem of combining the result lists returned by multiple search engines in response to a given query, where each item in a result list is ordered with respect to a search engine and query dependent relevance score [1]. Several ranking fusion methods have been proposed in the literatures. They can be classified based on whether: (i) they rely on the rank; (ii) they rely on the score; and (iii) they require training data or not.

Specifically, let Ω is a set of items, called universe. Given the universe Ω, a rank list τ w.r.t. Ω is an ordering of a subset $S \subseteq \Omega$, i.e., $\tau = [x_1 \geq x_2 \geq \ldots \geq x_k]$, with each $x_i \in S$, and $\geq$ some ordering relation on S. If $x_i \in \Omega$ is present in τ, written $x_i \in \tau$, let $\tau(i)$ denote the position or rank of x_i.

For a given rank list τ, with $s^\tau(i)$ we will denote the real-valued score assigned to item $i \in U$ in τ. Without loss of generality, we can assume that the higher the score the better the rank $\tau(i)$ of item i. More precisely, for $i, j \in S$ such that $i \neq j$, if $s^\tau(i) \geq s^\tau(j)$ then $\tau(i) \leq \tau(j)$. A rank list for which scores are provided will be called scored rank list.

Normalization is often applied before rank fusion in order to uniform score distributions. It captures different meaning of ranking lists to a unique framework for rank fusion. Given a rank list τ and an item $i \in \tau$, with $w^\tau(i)$ we will indicate the normalized weight of item $x_i \in \tau$, where $w^\tau(i)$ is computed according to either score normalization (2), (3) or rank normalization (4),(5).

- **(Score normalization)** The normalized score weight, $w^\tau(i)$, of an item $i \in \tau$ is defined as

$$w^\tau(i) = \frac{s^\tau(i) - \min_{j \in \tau} s^\tau(j)}{\max_{j \in \tau} s^\tau(j) - \min_{j \in \tau} s^\tau(j)} \qquad (2)$$

- **(Z-score normalization)** The normalized z-score weight, $w^\tau(i)$, of an item $i \in \tau$ is defined as

$$w^\tau(i) = \frac{s^\tau(i) - \mu_{s^\tau}}{\sigma_{s^\tau}} \qquad (3)$$

  where μ_{s^τ} is the mean of the scores in τ and σ_{s^τ} is their standard deviation, i.e., $\mu_{s^\tau} = \frac{1}{|\tau|} \cdot \sum_{j \in \tau} s^\tau(j)$ and $\sigma_{s^\tau} = \sqrt{\frac{1}{|\tau|} \cdot \sum_{j \in \tau} (s^\tau(j) - \mu_{s^\tau})^2}$

- **(Rank normalization)** The normalized rank weight, $w^\tau(i)$, of an item $i \in \tau$ is defined as

$$w^\tau(i) = 1 - \frac{\tau(i) - 1}{|\tau|} \qquad (4)$$

- **(Borda rank normalization)** The normalized Borda rank weight, $w^\tau(i)$, of an item $i \in \Omega$ w.r.t. R is defined as

$$w^\tau(i) = \begin{cases} 1 - \frac{\tau(i)-1}{|\Omega|} & \text{if } i \in \tau \\ \frac{1}{2} + \frac{|\tau|-1}{2 \cdot |\Omega|} & \text{otherwise} \end{cases} \qquad (5)$$

3.2 Rank Fusion

For an item $x_i \in \Omega$ in the recommender system, the recommendation from a recommender τ is usually represented by a rating $r^\tau(i)$, which indicates the relevance between an item and users' choice. Suppose the item set in the recommendation system is

$$\Omega = \{x_1, x_2, \ldots, x_k\} \qquad (6)$$

the output of each basic recommender τ will be a list of ratings

$$R^\tau = \{r^\tau(1), r^\tau(2), \ldots, r^\tau(k)\} \qquad (7)$$

Recommendation fusion (recommender combiner) is to integrate the local recommendations $\tau_1, \ldots, \tau n$, i.e., the local ratings R^τ, into a single recommendation $\hat{\tau}$, i.e., a single list of ratings $\hat{R} = \{\hat{r}(1), \hat{r}(2), \ldots, \hat{r}(k)\}$ through information fusion techniques. Then the system will recommend to the user the items with the highest estimated ratings.

Although any local ratings R^τ indicates the relevance between items and user's choice. They may have different formats according to different local recommendation techniques. Some are probabilities or posteriori probabilities; some are real valued numbers with possibly different range, and some are simply ranks. In order to capture the different meanings of ratings R^τ into a unique framework for most of fusion algorithms, we had better cover the ratings to the same format using one of the techniques (2)-(5) introduced in rank fusion above. In the following, without specific mention, the local ratings R^τ are already covered to the same format if they are originally different.

Rank fusion considers a set n rankings $R = \{\tau_1, \ldots, \tau_n\}$. Let Ω denote the union of the items in $\tau_1, \ldots, \tau_n$, i.e., $\Omega = \cup_{\tau \in R, x_i \in \tau} \{i\}$. Let $\hat{\tau}$ denote the fused rank list, and $s^{\hat{\tau}}(i)$ the fused score for each item $x_i \in \Omega$, as $\hat{\tau}$ will be ordered according to decreasing values of $s^{\hat{\tau}}$.

The most popular rank fusion method is the linear combination of the normalized weight, i.e., for an item $i \in \Omega$, the fused score $s^{\hat{\tau}}(i)$ is defined as

$$s^{\hat{\tau}}(i) = h(i, R)^y \cdot \sum_{\tau \in R} \alpha_\tau \cdot w^\tau(i) \qquad (8)$$

where (i) all the rank lists $\tau \in R$ have been normalized according to the same normalization method; (ii)$h(i, R) = |\{\tau \in R : x_i \in \tau\}|$ is defined as the *rank hits* of i *w.r.t.* a set of rankings R, which illustrates that the more partial rank lists rank the same item, the higher this item should be ranked in the fused list. (iii)$y \in \{0, 1\}$ indicates whether hits are counted or not; and (iv) $\sum_{\tau \in R} \alpha_\tau = 1$ and $\alpha_\tau \geq 0$ indicates the priority assigned to the rank list τ.

In fact, recommendation fusion can be directly formulated to the rank fusion problem by define the scores or the ranks from the ratings in the recommendation lists. We can either directly treat rating as the score in the rank fusion, i.e., $s^\tau(i) = r^\tau(i)$; or take the list of ratings in the recommendation and converted it to the rank list according to its rating values, i.e., if $r^\tau(i) \leq r^\tau(j)$, then $\tau(i) \leq \tau(j)$ for $i, j \in \Omega$. Also, the items set of the recommendation Ω can be either the union of the items in a list of base recommendations, i.e., $\Omega = \cup_{\tau \in R, x_i \in \tau} \{i\}$; or the top rated items in the list, i.e., $\Omega = \cup_{\tau \in R, x_i \in \tau, r^\tau(i) > \alpha} \{i\}$.

For recommendation fusion, we firstly apply score normalization (2), (3) or rank normalization (4), (5), then use rank fusion method (8) to obtain the fused score $s^{\hat{\tau}}(i)$. The scores can now be used to generate the final recommendation list, i.e., the higher the score the higher rank an item should appear in the final recommendation list.

4 Decision Fusion based Recommender System

For an item $x_i \in \Omega$ to be recommended, basically, we need to evaluate its relevance to user's choice which could be used to generate multiple decisions (classes, hypotheses), H_0, H_1, ...,H_m. Here, H could be binary values (1–like, 0–dislike) or multiple values (user's rating 1,2,...,5).

Each local recommender τ employs the mapping rule

$$u_\tau = \gamma_\tau(r^\tau(i)) = \begin{cases} 0 & \text{if } H_0 \text{ is declared} \\ 1 & \text{if } H_1 \text{ is declared} \\ \vdots & \vdots \\ m & \text{if } H_m \text{ is declared} \end{cases}$$

and passes the quantized information u_τ to the fusion center. Then, the recommendation fusion is the distributed decision fusion problem which try to reach a global decision $u_0 = \gamma_0(\mathbf{u})$ that favors $H_j, j = 1, 2, \ldots, m$ based on the received local decisions $\mathbf{u} = (u_1, u_2, \ldots, u_n)$.

In fact, Bayesian decision (classifier) is to minimize the Bayesian cost function [12]

$$\Re = \sum_{i,j} P(u = i|H = j)c_{i,j} \tag{9}$$

And usually, the cost parameter $c_{ij} = \begin{cases} 1 & i \neq j \\ 0 & i = j \end{cases}$. So, the Bayesian decision is

$$\hat{H} = \arg\max_{H_i}\{P(u|H_i) \cdot P(H_i)\}$$

For distributed Bayesian decision, given the local decisions $\mathbf{u} = (u_1, u_2, \ldots, u_n)$, the fused decision is

$$\begin{aligned} u_0 &= \arg\max_{H_i}\{P(u_1, u_2, \ldots, u_n|H_i) \cdot P(H_i)\} \quad (10) \\ &= \arg\max_{H_i}\{P(u_1|H_i) \cdot P(u_1|H_i) \cdot \ldots \cdot P(u_n|H_i) \end{aligned}$$

Specifically, for binary decision, Chair and Varshney [13] presented an optimum data fusion structure which combines the decisions from the individual detectors while minimizing the overall probability of error. Basically, it weights Individual decisions according to their reliability, and then a threshold comparison is performed to obtain the global decision.

$$\sum_{j=1}^{n} \left[u_j \log \frac{1-P_{M_j}}{P_{F_j}} + (1-u_j) \log \frac{P_{M_j}}{1-P_{F_j}} \right] \underset{\hat{H}=0}{\overset{\hat{H}=1}{\lessgtr}} \log \frac{P_0}{P_1}$$

where $P_0 = P(H_0)$, $P_1 = P(H_1)$ are prior probabilities of the two hypotheses, and P_F, P_M denote the probabilities of false alarm and miss of detection respectively.

All of the probability functions could be obtained by empirical probability obtained at training phase. Based on the definition above, the fused decision $u_0 = j, j = 1, 2, \ldots, m$ reflects the item relevant to the user's preference and could be directly used to generate the final recommendation list.

5 Dempster-Shafer Fusion based Recommender System

5.1 Dempster-Shafer Theory

In the last section, we formulate the recommendation fusion as the decision fusion problem. And the fusion algorithms have the strongest foundation, Bayes theory. This theory is based on the classical ideas of probability, and has at its deposal all of the usual machinery of statistics. Compare with Bayes thoery, Dempster-Shafer theory [14] [15] attempts to allow more interpretation of what uncertainty is all about. The D-S theory deals with measures of "belief" as opposed to probability. It can be interpreted as a generalization of probability theory where probabilities are assigned to sets as opposed to mutually exclusive singletons. In D-S theory, evidence can be associated with multiple possible events, e.g., sets of events. As a result, evidence in D-S theory can be meaningful at a higher level of abstraction without having to resort to assumptions about the events within the evidential set.

There are three important functions in D-S theory: the basic probability mass function (*m*), the Belief function (*Bel*), and the Plausibility function (*Pl*). Formally, the description of *m* on the power set $\psi(S)$ of S can be represented with

1. $m(\phi) = 0$,
2. Any $E \in \psi(S)$, $m(E) \geq 0$, and $\sum_{E \in \psi(S)} m(E) = 1$.

Also, for all sets $E \in \psi(S)$,

$$\begin{aligned} Bel(E) &= \sum_{B|B \subseteq E} m(B) \\ Pl(E) &= \sum_{B|B \cap E \neq \phi} m(E). \end{aligned}$$

The two measures, *Belief* and *Plausibility* are non-additive. This can be interpreted as is not required for the sum of all the Belief measures to be 1 and similarly for the sum of the Plausibility measures. In fact, the precise probability of an event lies within the lower and upper bounds of *Belief* and *Plausibility*, respectively, i.e., $Bel(E) \leq P(E) \leq Pl(E)$.

5.2 The Dempster Rule of Combination

The Dempster rule of combination is critical to the original conception of Dempster-Shafer theory. The measures of *Belief* and *Plausibility* are derived from the combined basic probability mass function. Specifically, the combination

m is calculated from the aggregation of multiple probability mass function $m_1, m_2, \ldots, m_n$ in the following manner:

$$m(E) = C^{-1} \cdot \sum_{\substack{E_i \in \psi(S) \\ \bigcap_{i=1}^{n} E_i = E}} \prod_{k=1}^{n} m_k(E_i) \quad (11)$$

where $C = \sum_{\substack{E_i \in \psi(S) \\ \bigcap_{i=1}^{n} E_i \neq \phi}} \prod_{k=1}^{n} m_k(E_i)$ is a normalization factor.

5.3 Dempster-Shafer Fusion for Recommender Combiner

To employ Dempster-Shafer theory on recommendation system, the key is to define the probability mass function m_τ for each local recommender τ. Here, Ω is the set of items to be recommended, and $\psi(\Omega)$ is power set of Ω. For an item $x_i \in \Omega$, $i = 1, \ldots, k$, we assign the mass function on the power set $\psi(\Omega)$ as

$$\begin{aligned} m_\tau(\{x_i\}) &= r^\tau(i) / \sum_{j=1}^{k} r^\tau(j) \\ m_\tau(A) &= 0 \quad \text{for } A \in \psi(\Omega) \text{ and } A \neq \{x_i\} \end{aligned} \quad (12)$$

where $r^\tau(i)$ is the rating output from each recommender τ.

With the Dempster rule of combination (11), the local mass functions from recommenders $\{m_1, m_2, \ldots, m_n\}$ are fused as m. And the evaluation of an item $x_i \in \Omega$ by the fused mass function $m(\{x_i\})$ can be used to generate the final recommendation list.

6 Estimation Fusion based Recommender System

Estimation fusion is the problem of how to best utilize useful information contained in multiple sets of data for the purpose of estimating an unknown quantity. For the recommender system, the goal is to estimate user's preference rating $r(i)$ to an item $x_i \in \Omega$. We treat the output of each local recommender $r^\tau(i)$ as the local estimate $\hat{r}_i^\tau$ of the user's preference to an item $x_i \in \Omega$. With the unified observation model defined in [16],

$$\hat{r}_i^\tau = r(i) + (\hat{r_i^\tau} - r(i)) = r(i) + (-\tilde{r_i^\tau}) \quad (13)$$

where the estimation error $-\tilde{r_i^\tau}$ acts as the observation noise. And the noise variance is $cov(-\tilde{r}_i^\tau) = P_i^\tau$.

The recommendation fusion is to integrating the local estimates $Y = [\hat{r}_i^1, \hat{r}_i^2, \ldots, \hat{r}_i^k]$ to the fused estimation $\hat{r}_i$ of the relevance an item $x_i \in \Omega$ and user's choice. This is obviously an estimation fusion problem.

At the fusion center, only linear unbiased estimation fusion is considered; that is, we consider the most commonly used linear estimation method: best linear unbiased estimation (BLUE). It is also known as linear minimum mean-square error (LMMSE), linear minimum variance (LMS), or linear unbiased minimum variance (LUMS) estimation [16]. It is defined by, for the available information Y at the fusion center,

$$\begin{aligned} \hat{r}_i &= E^*[r(i)|Y] = \bar{r}(i) + C_{r(i)Y} C_Y^{-1}(Y - \bar{Y}) \\ &= \vec{r}(i) + C_{r(i)Y} C_Y^{-1} \tilde{Y} \\ \text{MSE}(\hat{r_i}) &= E[(r(i) - \hat{r_i})(r(i) - \hat{r_i})'] \\ &= C_{r(i)} - C_{r(i)Y} C_Y^{-1} C'_{r(i)Y} \end{aligned} \quad (14)$$

where $\bar{r(i)} = E(r(i))$, $C_{r(i)}$ =cov$(r(i))$, $C_{r(i)Y}$ =cov$(r(i), Y)$ and C_Y =cov(Y) are prior information.

For BLUE, when $cov(-\tilde{r}_i^{\tau_1}, -\tilde{r}_i^{\tau_2}) = 0$ and $cov(r(i), -\tilde{r}_i^{\tau_1}) = 0$, the optimal LMMSE estimation fusion is

$$\begin{aligned} \hat{r}_i &= \left(\sum_{k=1}^{n} \frac{r_i^k}{P_i^k} \right) / \sum_{k=1}^{n} \frac{1}{P_i^k} \\ \text{MSE}(\hat{r}_i) &= 1 / \sum_{k=1}^{n} \frac{1}{P_i^k} \end{aligned} \quad (15)$$

Now the fused user's preference rating estimation $\hat{r}_i$ could be used to generate the final recommendation list directly.

7 Experiments

To evaluate the potential benefit of information fusion based recommender systems, we performed a series of experiments using the benchmark datasets MovieLens [6].

The MovieLens dataset contains 1 million ratings on the scale of 1 to 5 from 6040 customers on 3900 movies. The variables we used to predict unknown ratings include user identification number $c \in C$, movie identification number $x_i \in \Omega$, rating $r(i)$, and genre g. Here, g is a nominal value with 18 options, such as, 'Action', 'Adventure', 'Comedy', ..., etc., which can identify a group of related movies. On average, each user rated about 130 movies. And totally around 25% of movies have at least 3 genres. In the study, we only consider the movies with 3 genres, i.e., $G = [g1, g2, g3]$, and select the first 100 users for testing. For each user, we randomly select 2/3 of rated movies for training and another 1/3 rated movies for testing.

In order to evaluate various fusion strategies for the recommender system, we perform the rating estimation based on each genre under MMSE criteria, i.e.,

$$\hat{r}_i^m = E(r(i)|g_m) \quad m = 1, 2, 3 \text{ and } x_i \in \Omega \quad (16)$$

Therefore, for each user, we obtained 3 lists of rating estimations $R^m = [\hat{r}_1^m, \hat{r}_2^m, \ldots, \hat{r}_k^m]$. We test our recommender combiner strategies by combine the local estimated ratings R^1, R^2 and R^3 to the final rating R. The final combined ratings are either used to create the top rated movie lists, or compared with movies' true ratings.

Since the true rating value $r(i)$ is among $\{1, 2, 3, 4, 5\}$, multiple different programs may share one rating value. For

Table 1: The performance comparison on MoiveLens dataset.

Top N valued List		Single genre based rating estimation			Rank fusion (score norm.)	Rank fusion (Z-score norm.)	Rank fusion (Borda norm.)	Decision fusion	D-S fusion	Estimation fusion
		Genre 1	Genre 2	Genre 3						
$\alpha = 5$	FAR	0.2941	0.4146	0.4268	0.2549	0.2551	0.2497	0.2538	0.2617	0.2611
	MR	0.4812	0.5958	0.5998	0.4464	0.3806	0.4549	0.4456	0.4470	0.4480
$\alpha = 4$	FAR	0.1300	0.1496	0.1757	0.0973	0.1029	0.0981	0.0970	0.1013	0.0958
	MR	0.2083	0.2680	0.2881	0.1756	0.1469	0.1768	0.1735	0.1781	0.1751

performance evaluation, we set a value α to generate the top rated item sets S in the recommendation list, i.e.,

$$S = \{\ x_i : x_i \in \Omega, \text{and } r(i) \geq \alpha\}$$

and $N = |S|$ represent the number of items in top rated list S.

For performance comparison, we use the following two criteria

- (FAR): false_alarm_rate = the number of false positive items in the top N rated list / the length of predicted list.
- (MR): missing_rate = the number of missing items in the top N rated list / the length of true list.

Table 1 presents the performance comparison of rank fusion, decision fusion, D-S fusion and estimation fusion strategies for recommender system by choosing $\alpha = 5$ and $\alpha = 4$ respectively. Particularly, we compared 3 normalization methods (2), (3) and (5) for rank fusion; and considered binary hypotheses for an item $x_i \in \Omega$ by define H_1: in the top N rated list, H_0: not in the top N rated list, for decision fusion.

The results did show that all fused recommendations greatly eliminate the false positive and missing items in the predicted top list. There is no clear winner among the propose fusion strategies for recommendation fusion in this study. But user can always choose the comparibly better stragegies given a particular recommendation problem.

8 Conclusions

Recommender systems have great potential in many applications. It also brings researchers many challenges. Although many techniques have been proposed for the problem, no single method can work well in all situations. Therefore, combining recommendation ratings from multiple recommenders is a widely used approach to improve the system performance. In this paper, we systemically apply various information fusion methods to combine recommenders. Our experiments on benchmark dataset demonstrate that information fusion theory is able to significantly improve the recommendation system performance. In the future, we plan to explore more information fusion techniques in the area.

References

[1] C. Dwork, R. Kumar, M. Noar, and D. Sivakumar, ”Rank aggregation methods for the web,” *10th International Conference on the World Wide Web*, ACM Press and Addison Wesley, 2001, pp. 613–622.

[2] W. Hill, L. Stead, M. Rosenstein, and G. Furnas, ”Recommending and evaluating choices in a virtual community of use,” *Proc. Conf. Human Factors in Computing Systems*, 1995.

[3] P. Resnick, N. Iakovou, M. Sushak, P. Bergstrom, and J. Riedl, ”GroupLens: an open architecture for collaborative filtering of netnews,” *Proc. 1994 Computer Supported Cooperative Work Conf.*, 1994.

[4] U. Shardanand and P. Maes, ”Social information filtering: algorithms for automating Word of Mouth,” *Proc. Conf. Human Factors in Computing Systems*, 1995.

[5] G. Linden, B. Smith, and J. York, ”Amazon.com recommendations: item-to-item collaborative filtering, *IEEE Internet Computing*, Jan./Feb., 2003.

[6] B.N. Miller, I. Albert, S.K. Lam, J.A. Konstan, and J. Riedl, ”MovieLens unplugged: experiences with an occasionally connected recommender system,” *Proc. Int'l Conf. Intelligent User Interfaces*, 2003.

[7] D. Billsus, C.A. Brunk, C. Evans, B. Gladish, and M. Pazzani, ”Adaptive interfaces for ubiquitous web access,” *Comm. ACM*, Vol 45, No. 5, pp. 34–38, 2002.

[8] G. Adomavicius, and A. Tuzhilin, ”Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions,” *IEEE Trans. on knowledge and data engineering*, Vol 17, No. 6, pp. 734–749, June. 2005.

[9] M. Balabanovic and Y. Shoham, ”Fab: Content-Based, Collaborative Recommendation,” *Comm. ACM*, Vol. 40, No. 3, pp. 66–72, 1997.

[10] ”http://movielens.umn.edu”.

[11] M. E.Renda, and U. Straccia, ”Web metasearch: rank vs. score based rank aggregation methods,” *SAC*, pp. 841–846, 2003.

[12] P.R. Tenney, and N.R. Sandell, ”Detection with distributed sensors,” *IEEE Trans. on aerospace and electronic systems*, Vol. 17, No. 4, pp. 501–509, July 1981.

[13] Z. Chair, and P.K. Varshney, ”Optimal data fusion in multiple sensor detection systems,” *IEEE Trans. on aerospace and electronic systems*, Vol. AES-22, No. 1, pp. 98–101, Jan. 1986.

[14] A.P. Dempster, ”A generalization of Bayesian inference,” *J.R.Stat. Soc.*, B30, pp 205–247, 1968.

[15] G. Shafer, "A mathematical thoery of evidence,"*(Princeton, NJ: Univeristy Press)*, 1976.

[16] X. R. Li, Y. M. Zhu, and C. Z. Han, "Unified optimal linear estimation fusionł part I: unified models and fusion rules,"*Proceedings of international conference on information fusion*, July, 2000.