Rule Mining and Missing-Value Prediction in the Presence of Data Ambiguities

Kasun Wickramaratna, Miroslav Kubat, Kamal Premaratne, Thanuka Wickramarathne
Department of Electrical and Computer Engineering, University of Miami
{k.wickramaratna@umiami.edu, mkubat@miami.edu, kamal@miami.edu, t.wickramarathne@umiami.edu}

Abstract

The success of knowledge discovery in real-world domains often depends on our ability to handle data imperfections. Here we study this problem in the framework of association mining, seeking to identify frequent itemsets in transactional databases where the presence of some items in a given transaction is unknown. We want to use the frequent itemsets to predict "missing items": based on the partial contents of a shopping cart, predict what else will be added. We describe a technique that addresses this task, and report experiments illustrating its behavior.

Introduction

The task of association mining is to detect frequently co-occurring groups of items in transactional databases. These frequent *itemsets* can be exploited in predictions. Suppose a set of transactions frequently contains the itemset $\{i_1, i_2, i_3\}$. Observation of the items $\{i_1, i_2\}$ in the partial contents of a shopping cart may lead us to expect that the customer will also buy i_3. Although association mining has usually been cast in the department-store paradigm, many other domains can be converted to the same scenario.

Suppose we have a database of customer ratings of a line of products. If a group of users tend to rate products $\{p_1, p_2, p_3\}$ in a similar way, $\{\langle p_1, \mathfrak{r}_1\rangle, \langle p_2, \mathfrak{r}_2\rangle, \langle p_3, \mathfrak{r}_3\rangle\}$, then a new user's ratings $\{\langle p_1, \mathfrak{r}_1\rangle, \langle p_2, \mathfrak{r}_2\rangle\}$ lead us to expect that this user, too, will rate p_3 as $\mathfrak{r}_3$. By considering a $\langle product, rating\rangle$ pair as an item, we can use the department-store paradigm. But the scope of applications is broader than it seems. Thus in medical diagnosis, a patient's symptoms are rarely due to a single cause—multiple diseases tend to conspire. Having identified one, the physician wants to anticipate the others, if only to suggest additional lab tests. But knowing the presence or absence of a disease is not enough. What matters is also the severity as quantified, say, by a value from $\Theta = \{Critical, Medium, Normal\}$. When assigning the value, a physician relies on his/her experience and/or the experience of colleagues. Such ratings are inevitably ambiguous and not easily expressed in terms of probabilities—for instance, it would be mistaken to assume that the statement, "the symptom is $Critical$ with a 70% confidence" implies a 30% confidence in the complement of $Critical$. The lack of mechanisms to accommodate such subjectivity often necessitates various unwarranted "interpolations." Their inadequacy motivates our research: knowing some (ambiguous) ratings a user has given, we want to predict his or her other ratings.

Recent work studied this issue in the framework of classifier induction: for class-label ambiguities (Subasingha *et al.* 2008) as well as for attribute-value imperfections (Hewawasam, Premaratne, & Shyu 2007). But the problem studied here is more general. Whereas classification usually seeks to predict a single preselected class attribute, we are concerned here with the case where *any* attribute can be the "class label." We want to predict all unknown *items* based on the partial knowledge of the presence of other items (note that classification is only a special case of this task). A collaborative-filtering-based approach to this task has been recently proposed by (Wickramarathne 2008), but to use association mining to this end is new. We propose a novel technique, DS-ARM—Dempster-Shafer based Association Rule Mining and report experiments illustrating its behavior.

Problem Statement

We use $p_j,\ j = \overline{1, N_p}$, to denote products (or attributes) in the dataset. Let $\Theta = \{\theta_1, \ldots, \theta_K\}$ be the set of mutually exclusive and exhaustive ratings. Rating values that can be assigned to a product are thus drawn from the power set 2^{Θ} of Θ and $\mathfrak{r}_\ell,\ \ell = \overline{1, N_\mathfrak{r}}$, where $N_\mathfrak{r} = |2^{\Theta}|$, are used to denote user assigned ratings. We refer to each pair $\langle product, rating\rangle$ or $\langle attribute, value\rangle$ as an *item* and the item vector of a single user as a *transaction*. More formally, let $I = \{i_{j\ell} | j = \overline{1, N_p}, \ell = \overline{1, N_\mathfrak{r}}\}$ be a set of distinct items where $i_{j\ell} = \langle p_j, \mathfrak{r}_\ell\rangle$. Let a database consist of N transactions, $T_1, \ldots, T_N$, such that $T_k \subseteq I$, $\forall k$. An *itemset, X*, is a group of items, i.e., $X \subseteq I$. The *support* of itemset X is the number, or the percentage, of transactions that subsume X. An itemset that satisfies a user-specified minimum support value is called a *frequent itemset* or a *high support itemset*.

Let us assume that an association mining program has already discovered all high support itemsets. For each such itemset, X, any pair of subsets, $r^{(a)}$ and $r^{(c)}$, such that $r^{(a)} \cup r^{(c)} = X$ and $r^{(a)} \cap r^{(c)} = \emptyset$, we can define an association rule: $r : r^{(a)} \Rightarrow r^{(c)}$; $r^{(a)}$ is the rule's *antecedent* and $r^{(c)}$ is the *consequent*. The rule reads: if all items from $r^{(a)}$ are present in a transaction, then all items from $r^{(c)}$ are also present in the same transaction. The rule does not have to be absolutely reliable. The probabilistic *confidence* in the rule $r^{(a)} \Rightarrow r^{(c)}$ can be defined with the help of the support (relative frequency) of the antecedent and consequent as the percentage of transactions that contain $r^{(c)}$ among those transactions that contain $r^{(a)}$:

$$conf = support\,(r^{(a)} \cup r^{(c)})/support\,(r^{(a)}). \qquad (1)$$

The number of rules implied by X grows exponentially in the number of items; it is thus practical to consider only high-confidence rules derived from high-support itemsets.

Given an itemset s in a transaction, we want to predict the remaining items of this transaction. The association rules we generate for this purpose must satisfy the following: (1) The rule antecedents should be sufficiently similar to s. (2) The rule consequent is limited to *any* single item $\langle p_j, \bullet \rangle \notin s$.

In summary: Given the itemset $s \subseteq I$, find the matching rules of the form $r^{(a)} \Rightarrow i_{j\ell}$, such that $r^{(a)}$ is "close" (we will formalize this later) to s and $\langle p_j, \bullet \rangle \notin s$, and exceed the user-set minimum support, θ_s, and minimum confidence, θ_c. Find a method to combine rules with mutually contradicting consequents. Ultimately, we are predicting the $\langle product, rating \rangle$ values of unrated products.

Representation of Imperfect Data

Preliminaries: DS Theory

Let us define the *frame of discernment (FoD)* as a set of mutually exclusive and exhaustive propositions, $\Theta = \{\theta_1, \ldots, \theta_K\}$. A proposition, θ_i, referred to as a *singleton*, represents the lowest level of discernible information. We assume that all products have the same FoD. In our context, θ_i states that the "rating of given product is equal to θ_i." Elements in 2^Θ, the power set of Θ, form all propositions of interest. Any proposition that is not a singleton, e.g., (θ_1, θ_2), is referred to as *composite*. In our context, composite propositions represent ambiguous ratings.

The mapping $m : 2^\Theta \longmapsto [0, 1]$ is a basic belief assignment (BBA) for the FoD Θ if (Shafer 1976);

$$m(\emptyset) = 0; \quad \sum_{A \subseteq \Theta} m(A) = 1. \qquad (2)$$

The BBA of a proposition $A \subseteq \Theta$ is free to move into its individual singletons. This is how DS theory models *ignorance*. Any proposition A that possesses a non-zero mass, i.e., $m(A) > 0$, is called a *focal element;* the set of focal elements, $\mathfrak{F}$, is referred to as the *core.* The triple $\{\Theta, \mathfrak{F}, m(\bullet)\}$ is called the *body of evidence (BoE).*

An indication of the evidence one has towards all propositions that may themselves imply a given proposition $A \subseteq \Theta$ is quantified via the *belief,* $Bel(A) \in [0, 1]$, defined as

$$Bel(A) = \sum_{B \subseteq A} m(B). \qquad (3)$$

$Bel(A)$ represents the total support that can move into A without any ambiguity. Note that $Bel(A) = m(A)$ if A is a singleton. *Plausibility* of A is defined as $Pl(A) = 1 - Bel(\overline{A})$; it represents the extent to which one finds A plausible.

A probability distribution $Pr(\cdot)$ satisfying $Bel(A) \leq Pr(A) \leq Pl(A)$, $\forall A \subseteq \Theta$, is said to be compatible with the underlying BBA $m(\bullet)$. An example of such a probability distribution is the *pignistic probability distribution* $BetP(\bullet)$ defined for each singleton $\theta_i \in \Theta$ as follows (Smets 1999):

$$BetP(\theta_i) = \sum_{\theta_i \in A \subseteq \Theta} m(A)/|A|. \qquad (4)$$

Here $|A|$ denotes the cardinality of set A.

The Dempster's rule of combination (DRC) makes it possible to arrive at a new BoE by fusing the information from several BoEs that span the same FoD. Consider the two BoEs, $\{\Theta, \mathfrak{F}_1, m_1(\bullet)\}$ and $\{\Theta, \mathfrak{F}_2, m_2(\bullet)\}$. Then,

$$K_{12} = \sum_{B_i \cap C_j = \emptyset} m_1(B_i)\, m_2(C_j) \qquad (5)$$

indicates the *conflict* between the evidence of the two BoEs. If $K_{12} < 1$, then the two BoEs are compatible, and the two BoEs can be combined to obtain the overall BoE $\{\Theta, \mathfrak{F}, m(\bullet)\}$ as follows: for all $A \subseteq \Theta$,

$$m(A) \equiv (m_1 \oplus m_2)(A) = \frac{\sum\limits_{B_i \cap C_j = A} m_1(B_i)\, m_2(C_j)}{(1 - K_{12})}. \qquad (6)$$

A variation of the DRC that can be used to address the reliability of the evidence provided by each contributing BoE is to incorporate a *discounting factor* d_i, $d_i \leq 1$, to each BoE (Shafer 1976). The BBA thus generated is

$$m(A) = (\hat{m}_1 \oplus \hat{m}_2)(A), \text{ where, for } i = 1, 2,$$
$$\hat{m}_i(A) = \begin{cases} d_i m_i(A), & \text{for } A \subset \Theta; \\ (1 - d_i) + d_i m_i(\Theta), & \text{for } A = \Theta. \end{cases} \qquad (7)$$

Attribute Value Ambiguities

The FoD of rating of product p_j, is taken to be finite and is denoted by Θ_{pref}. For instance, in a "five-star" rating system $\Theta_{pref} = \{1, 2, 3, 4, 5\}$;. The number of possible singleton values a product rating may assume is $|\Theta_{pref}|$ and $\mathfrak{r}_\ell \in 2^{\Theta_{pref}}$. The "intra-attribute BBA" or the BBA of rating of product p_j is a BBA $m_j : 2^{\Theta_{pref}} \mapsto [0, 1]$ defined on the FoD Θ_{pref}; $\{\Theta_{pref}; \mathfrak{F}_j; m_j\}$ is the corresponding intra-attribute BoE (*intra-BoE*) (Hewawasam, Premaratne, & Shyu 2007). Note that, $\mathfrak{F}_j = \emptyset$ denotes that the product p_j is "not rated". We assume that an $\langle product, rating \rangle$ vector whose ratings are all "not rated" is non-existent (i.e., in our context, each user has rated at least one product).

The intra-attribute BBA captures the uncertainty among the ratings each product may take and it allows several types of common data imperfections to be conveniently modeled. This "inter-attribute BBA" can capture the interrelationships among different attributes (Hewawasam, Premaratne, & Shyu 2007). Clearly, the inter-FoD Θ_T of each attribute vector T is the cross-product of the intra-FoD of each attribute. The inter-BBA of a given record is referred to as Data Record BBA (DR-BBA).

Table 1 shows a toy domain with four distinct products and the ratings (some ambiguous, others "crisp") given by two users. An empty field indicates the user has not rated the product. DR-BBAs generated from Table 1 is shown in Table 2. The user rating vector u_1 is converted into four data-records in the cross-product space (that grows exponentially in the number of ambiguous ratings). So, the proposed method becomes expensive in highly ambiguous domains. Our toy domain can be seen as a transaction database where each $\langle product, rating\rangle$ represents an item and each row represents a transaction. This database is then used for frequent-itemset detection and for association rule generation.

Table 1: Intra-BBAs of Two Data Records

product	p_1		p_2		p_3		p_4	
user	$\mathfrak{F}_1$	m_1	$\mathfrak{F}_2$	m_2	$\mathfrak{F}_3$	m_3	$\mathfrak{F}_4$	m_4
u_1	4	1.0	4	0.8	3	0.6		
			4,5	0.2	3,4	0.4		
u_2	1	1.0	5	1.0			3	0.8
							2,3	0.2

Table 2: DR-BBAs of the Data Records in Table 1

Data Rec.	DR-BBA	Itemset
$u_1^{(1)}$	0.48	$\langle p_1, 4\rangle, \langle p_2, 4\rangle, \langle p_3, 3\rangle$
$u_1^{(2)}$	0.32	$\langle p_1, 4\rangle, \langle p_2, 4\rangle, \langle p_3, (3, 4)\rangle$
$u_1^{(3)}$	0.12	$\langle p_1, 4\rangle, \langle p_2, (4, 5)\rangle, \langle p_3, 3\rangle$
$u_1^{(4)}$	0.08	$\langle p_1, 4\rangle, \langle p_2, (4, 5)\rangle, \langle p_3, (3, 4)\rangle$
$u_2^{(1)}$	0.80	$\langle p_1, 1\rangle, \langle p_2, 5\rangle, \langle p_4, 3\rangle$
$u_2^{(2)}$	0.20	$\langle p_1, 1\rangle, \langle p_2, 5\rangle, \langle p_4, (2, 3)\rangle$

Making Predictions

Given a user's ratings with ambiguities, and asked to predict unrated products, the first step is to get the cross product of intra-BBAs and find the DR-BBAs of the given rating vector. For instance, if the given user is u_2 (Table 1), and we are asked to predict the rating for product p_3, we get two "data records" $u_2^{(1)}$, $u_2^{(2)}$ (Table 2). The matching rule set is generated for each of the records, and the prediction is made by the combination of the rules. Each prediction is discounted based on the corresponding DR-BBA; the discounted BBAs are then combined in making the final prediction.

For a given itemset $s \subseteq I$, we want to find all rules of the form $r^{(a)} \Rightarrow i_{j\ell}$, where $r^{(a)}$ "matches" s and $\langle p_j, \bullet\rangle \notin s$, that exceed minimum support and minimum confidence. Note that the consequent $i_{j\ell}$ is a single item, i.e., $\langle p_j, \mathfrak{r}_\ell\rangle$. For each unrated product p_j, the corresponding ruleset—all the matching rules having a consequent of the form $\langle p_j, \mathfrak{r}_\ell\rangle; \mathfrak{r}_\ell \subseteq 2^{\Theta_{pref}}$—is selected and a DS theoretic approach is used to combine the rules. This prediction is given as a DS theoretic mass structure over the set of singletons or the frame of discernment. If no rule consequent in the generated ruleset has $\langle p_j, \bullet\rangle$, no prediction is made for p_j.

Distance Metric

We define a rule $r^{(a)} \Rightarrow i_{j\ell}$ and given itemset s is "matching" iff (a) $\forall\langle p_j, \bullet\rangle \in r^{(a)} \rightarrow \langle p_j, \bullet\rangle \in s$, and (b) $\forall\langle p_j, \mathfrak{r}_j^{(r^{(a)})}\rangle \in r^{(a)}$ and $\langle p_j, \mathfrak{r}_j^{(s)}\rangle \in s$: $d_j \equiv |\mathfrak{r}_j^{(r^{(a)})} - \mathfrak{r}_j^{(s)}| \leq d_t$; where d_t is a user-set distance threshold. $\mathfrak{r}_j^{(r^{(a)})}$ and $\mathfrak{r}_j^{(s)}$ are the ratings given for the product p_j in the rule antecedent $r^{(a)}$ and given itemset s respectively. If the rating $\mathfrak{r}_j$ is not a singleton we take the mean value to calculate the distance.

Distance between a matching rule antecedent and incoming itemset s is denoted by $\mathfrak{d}_{s,r^{(a)}}$; where $\mathfrak{d}_{s,r^{(a)}} = \sum_j d_j/|r^{(a)}|$.

Rule Generation

To expedite rule generation, we rearrange the database by the use of the *flagged IT-tree* developed by (Li & Kubat 2006). Our goal is not to generate all association rules, but to build a predictor from a set of 'effective' association rules. The rule generation algorithm takes an incoming itemset as the input and returns a graph that defines the association rules entailed by the incoming itemset.

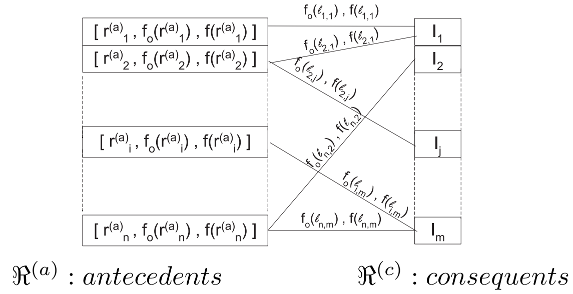


Figure 1: The Rule Graph, G. $f(r_i^{(a)})$= frequency count of antecedent, $f(\ell_{i,j})$=support count of rule $r_i^{(a)} \Rightarrow I_j$

The graph consists of two lists; the antecedents list $\Re^{(a)}$ and the consequents list $\Re^{(c)}$. In each node, $r_i^{(a)}$, the antecedents list keeps the corresponding frequency count $f(r_i^{(a)})$. As shown in Fig. 1, the line $\ell_{i,j}$ between the two lists links an antecedent $r_i^{(a)}$ with a consequent I_j. The cardinality of the link, $f(\ell_{i,j})$, represents the support of the rule $r_i^{(a)} \Rightarrow I_j$. The frequency counts denoted by $f_o(\bullet)$ are used when building the graph. If T_i is a transaction, then $f_o(r^{(a)})$ records the sum of DR-BBAs of all T_is, where $r^{(a)}$ is the largest "matching" itemset in T_i. All frequency counts are initialized to zero at the beginning and are updated as we traverse the IT-tree according to the rule generation algorithm from (Wickramaratna, Kubat, & Premaratne 2009).

After completing the rule graph, we select rules that exceed the minimum support and minimum confidence in the rule-combination step. To account for data skewness and to avoid loosing important rules, we use the modified support value described in the next section.

Basic Belief Assignment

In association mining, a user-set threshold decides which rules have 'high support.' The rules that pass this test are all treated equally, regardless of their supports, and decisions are based solely on the rules' confidence values. Since an intuitive approach would give more weight to rules with higher support, we propose a method to assign the rule-masses based on both confidence and support values (though the supports should have a smaller impact).

In many applications, the training sets are skewed. Thus in a medical domain, the percentage of patients with the "critical" rating for renal failure might be only 2%. Therefore, the rules suggesting "critical" rating for renal failure will have very low support, and rules suggesting the complement will have higher supports. A predictor built from a skewed training set often tends to favor the "majority" classes at the expense of "minority" classes. To mitigate this problem, we adopt the following modified support value:

Definition 1 (Partitioned-Support) *The partitioned-support p_supp of the rule, $r^{(a)} \Rightarrow r^{(c)}$ is the percentage of transactions that contain $r^{(a)}$ among those transactions that contain $r^{(c)}$, i.e.,*

$$p\_supp = support(r^{(a)} \cup r^{(c)})/support(r^{(c)}).$$

With Definition 1 in place, we take inspiration from the traditional F_α-measure (van Rijsbergen 1979) and use the weighted harmonic mean of support and confidence to assign the following BBA to the rule $r^{(a)} \Rightarrow \langle p_j, \mathfrak{r}_\ell \rangle$:

$$m(\langle p_j, \mathfrak{r}\rangle | r^{(a)}) = \begin{cases} \beta, & \text{for } p_j\text{'s rating } \mathfrak{r} = \mathfrak{r}_\ell; \\ 1-\beta, & \text{for } p_j\text{'s rating } \mathfrak{r} = \Theta_{pref}; \\ 0, & \text{otherwise,} \end{cases} \tag{8}$$

where $\beta = \dfrac{(1+\alpha^2) \times conf \times p\_supp}{\alpha^2 \times p\_supp + conf}$ with $\alpha \geq 1$. Note that, as α increases, the emphasis placed upon the partitioned-support measure in $m(\bullet)$ decreases.

With this mass allocation, the effectiveness of a rule is tied to both its confidence and partitioned-support. Moreover, just as the F_α-measure enables one to 'trade' precision and recall, the mass allocation above allows us to trade the effectiveness of the confidence and partitioned-support of a rule. Parameter α can quantify the user's willingness to trade increased confidence for lower partitioned-support.

Discounting Factor

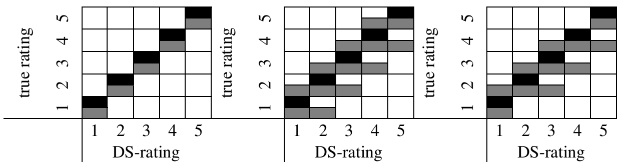
The reliability of the evidence provided by each contributing BoE is addressed by incorporating the following discounting factor (Hewawasam, Premaratne, & Shyu 2007):

$$d = [1 + Ent]^{-1}[1 + \ln(N_p - |r^{(a)}|)]^{-1},$$
$$\text{with } Ent = -\sum_{i_j \subseteq \Theta} m(i_{j\ell}|r^{(a)}) \ln[m(i_{j\ell}|r^{(a)})]. \tag{9}$$

Recall that N_p denotes the number of products in the database. The term $1/(1 + Ent)$ accounts for the *uncertainty* of the rule about its consequent. The term $1/(1 + \ln[N_p - |r^{(a)}|])$ accounts for the *non-specificity* in the rule antecedent. Note that, d increases as Ent decreases and length of rule antecedent increases. As dictated by (7), the BBA then gets accordingly modified. The DRC is then used on the modified BoEs to combine the evidence.

Experiments

We experimented with Movielens, a movie recommendation domain widely used for benchmarking (Research 2007). The dataset consists of 100,000 ratings provided by 943 users for 1682 movies. The ratings are integers from the interval [1,5], with 5 being best. To demonstrate our technique's full functionality, we needed soft ratings that were not available in Movielens. We thus created the dataset DS-Movielens by artificially introducing soft ratings: we relied on different user profiles obtained from "partial probability models," a widely used methodology to convert data with diverse types of imperfections into the DS theoretic framework (Blackman & Popoli 1999), (Hewawasam, Premaratne, & Shyu 2007). To be more specific, we used—as in (Wickramarathne 2008)—three user profiles: zero tolerance, ±1 tolerance, and end-weighted ±1 tolerance. The partial probability models for each profile are shown in Figure 2. The horizontal axis (lighter shading) always represents the user rating as it appears in the DS-Movielens dataset; the vertical axis (dark shading) represents the true rating a movie should receive. A power-set approach enables us to account for user rating imperfections without resorting to various "assumptions" and "interpolations" that may be hard to justify.



(a) 0 tolerance (b) ±1 tol. (c) ±1 tol. end-weight

Figure 2: Partial probability models of user profiles.

We then built DS-Movielens by the following steps: (a) Select a user rating that has been rated as $\mathfrak{r}_k$. (b) Randomly, with the probabilities $\{p, (1-p)/2, (1-p)/2\}$, select one user profile from Figs 2(a), 2(b), and 2(c), respectively. (c) Obtain the corresponding feasible true ratings and DS theoretic basic probability assignment (BPA) $\mathfrak{r}_k^{(DS)}$ via the procedure in (Blackman & Popoli 1999). (d) Replace $\mathfrak{r}_k$ with $\mathfrak{r}_k^{(DS)}$. (e) Repeat for all rated entries in Movielens dataset.

Performance Criteria

Datasets with Hard Ratings: The mean absolute error (MAE) is the most popular performance criterion to evaluate user ratings (Herlocker *et al.* 2004). Since our algorithm presents the prediction as a mass structure over the FoD $\Theta_{pref} = \{1, 2, 3, 4, 5\}$, to compute the MAE, these DS theoretic predictions are converted to hard predictions via the pignistic transformation. Pignistic transformation converts the DS-theoretic "soft" decision to a hard decision. Note that even though we did this transformation to directly compare our results with other available methods, this approach is some what unfair to the proposed DS-ARM whose strength lies in its ability to generate soft decisions. In addition to MAE, other standard performance metrics—such as precision, recall, and F_1—were used in the results.

Datasets with Soft Ratings: When the user preference ratings are soft, we must determine how well the predicted BoE's ($\hat{\gamma}_j$) approximate the ground truths (γ_j). $BetP_{\gamma_j}$ denotes the pignistic probabilities drawn from the BoE γ_j. Taking inspiration from (Jousselme, Grenier, & Bosse 2001), we evaluate the soft result via the metric

$$DS\_PE = \sum_{j=1}^{\mathfrak{N}_p} \frac{1}{\sqrt{2}} ||BetP_{\hat{\gamma}_j} - BetP_{\gamma_j}|| \div \mathfrak{N}_p, \qquad (10)$$

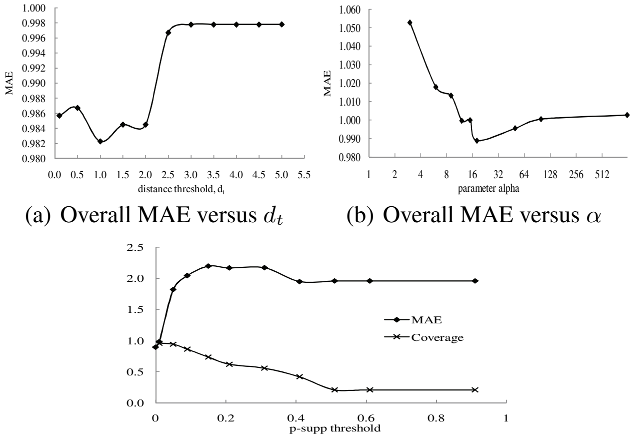
where $\mathfrak{N}_p$ is the number of predictions made and $|| \bullet ||$ denotes the Euclidean norm. Note that DS_PE takes values from $[0, 1]$: $DS\_PE = 0$ means pignistic probabilities of the prediction is exactly same as that of the ground truth. We could also have used the KL-divergence instead of the Euclidean norm, but the error then would not be bounded by the closed interval $[0, 1]$. Moreover, KL-divergence requires the pignistic distributions corresponding to the true and predicted BPAs to have identical supports.

Experiment Setup

For consistency with previous work, we followed the methodology from (Herlocker *et al.* 1999): We randomly selected 10% of users and, for each of them, we withheld 5 randomly selected ratings, i.e., we "hid" 5 non-empty fields in the ratings matrix and prevented them from being used for training. We then used these withheld ratings as an independent testing set. The remaining ratings represented the training set. We repeated this process for 10 different random splits into training and testing sets. Results shown here are the average results obtained from the 10 splits.

Experiment 1. DS-ARM Performance: Let us first investigate DS-ARM's behavior under diverse parameter settings. The technique performance is likely to depend on the distance threshold d_t, the minimum p_supp threshold, and the parameter α in (8). For the time being, let us focus on mean absolute error, MAE. Throughout the experiments, we will keep two parameters fixed at "baseline values," while varying the third parameter. The baseline values are $p\_supp = 0.01, \alpha = 10, d_t = 1.5$.

Figure 3(a) shows how the performance varies with growing d_t, with the other parameters fixed. The the minimum error was achieved when $d_t = 1.0$. With high d_t, the error increases due to the contributions from too "dissimilar" rules. When the distance threshold is tight, few rules are involved and the lack of diverse opinions seems to cause errors. Figure 3(b) shows how MAE varies with changing α. Minimum is reached around $\alpha = 20$; which supports our use of the partitioned-support value in mass allocation. Figure 3(c) shows how MAE varies with minimum partitioned-support threshold of selecting rules. Best performance us obtained by keeping the threshold very low. We have to remember that the computational costs of rule combination are high if many rules are combined. We observe that as the p_supp threshold increases, the coverage decreases (i.e., no rules are selected to predict certain preferences).



(a) Overall MAE versus d_t

(b) Overall MAE versus α

(c) MAE and Coverage Vs p_supp

Figure 3: Behavior of DS-ARM

Experiment 2. DS-ARM Performance on Hard Data: Although the main strength of DS-ARM is its ability to deal with ambiguous ratings, we still wanted to see how it compares to older techniques when "crisp" values are used. To this end, we compared DS-ARM with one of the most widely used mechanisms of Automated Collaborative Filtering (ACF): the correlation analysis based approach from (Herlocker *et al.* 1999). We will refer to our re-implementation of this algorithm by the acronym CORR.

The parameters of both systems were set to maintain at least 95% level of "coverage," calculated as the percentage of predictions made by the predictor out of the total number of predictions. Predictors sometimes fail to achieve 100% coverage due to lack of evidence. The parameter settings are: for DS-ARM, $d_t = 1.0$, $\alpha = 20$, $p\_supp\ thres = 0.01$; and for CORR, similarity threshold = 0.1.

Table 3 summarizes the results, with boldface values indicating the best performance. Although the difference is on average only marginal, our method consistently out performs CORR in predicting high user ratings "3-5". For ratings "1-2", CORR is better. Note that the frequencies of ratings "1-2" are low (the ratings distribution is "1": 6%, "2": 11%, "3": 27%, "4": 34%, "5": 21%). Based on the results, we conclude that the two methods provide comparable performance even in the case of "crisp" data.

Experiment 3. DS-ARM Performance on Soft Data: As we said, we are not aware of any other system that can predict ratings based on the "soft" data such as those in DS-Movielens. Still, we felt that some comparison with previous work is needed. This is why we decided to use the CORR approach we worked with in Experiment 2 and to interpret the hard decisions made by other predictors as soft

Table 3: Performance Comparison: Hard Decisions

algo.	Metric	True Rating					mean
		1	2	3	4	5	MAE
DS-ARM	MAE	2.31	1.62	**0.68**	**0.39**	**1.06**	0.89
	Pr	0.38	0.13	**0.38**	**0.39**	**0.36**	
	Re	0.08	0.04	**0.39**	**0.63**	**0.23**	
	F_1	0.14	0.06	**0.39**	**0.48**	**0.27**	
CORR	MAE	**1.80**	**1.38**	0.71	0.57	1.18	0.91
	Pr	**0.40**	**0.19**	0.33	0.38	0.32	
	Re	**0.16**	**0.16**	0.38	0.51	0.19	
	F_1	**0.23**	**0.18**	0.36	0.44	0.24	

decisions. The comparison of CORR and DS-ARM is made simpler by the fact that it is in the nature of correlation analysis that the predictions of CORR are not necessarily integer-valued. To be able to interpret a CORR prediction, $\hat{\mathfrak{r}}_k$, as soft, we relied on the following DS-theoretic BPA:

$$\hat{m}_k(A) = \begin{cases} \lceil \hat{\mathfrak{r}}_k - \hat{\mathfrak{r}}_k, & \text{for } A = \lfloor \hat{\mathfrak{r}}_k \text{ when } \hat{\mathfrak{r}}_k \notin \Theta; \\ \hat{\mathfrak{r}}_k - \lfloor \hat{\mathfrak{r}}_k, & \text{for } A = \lceil \hat{\mathfrak{r}}_k \text{ when } \hat{\mathfrak{r}}_k \notin \Theta; \\ 1, & \text{for } A = \hat{\mathfrak{r}}_k \text{ when } \hat{\mathfrak{r}}_k \in \Theta; \\ 0, & \text{otherwise,} \end{cases} \tag{11}$$

where $\lceil \hat{\mathfrak{r}}_k$ and $\lfloor \hat{\mathfrak{r}}_k$ denote the lowest integer rating that does not fall below and the highest integer rating that does not exceed the CORR prediction $\hat{\mathfrak{r}}_k$, respectively. For instance, with $\Theta = \{1, 2, 3, 4, 5\}$, the CORR prediction 3.3 is interpreted as the Bayesian statement, "The rating is 3 with 70% confidence, and 4 with 30% confidence"; (11) corresponds well with this interpretation. In addition, the known DS-Movielens user ratings could also be ambiguous. Therefore, when working with CORR, the pignistic transformation was used on the known ambiguous ratings.

Table 4 compares the results of CORR and DS-ARM, using the performance metric DS_PE defined by (10). The probability of selecting the "zero tolerance user" varies from 1 (no ambiguity) to 0.8 (20% ambiguity). The results indicate that DS-ARM indeed outperforms CORR on these data.

Table 4: Performance Comparison: Soft Data

algo.	Zero tolerance user selection probability, p				
	1.00	0.95	0.90	0.85	0.80
DS-ARM	**0.60**	**0.58**	**0.57**	**0.55**	**0.53**
CORR	0.61	0.61	0.60	0.59	0.58

Conclusion

The newly proposed technique DS-ARM for recommender systems with ambiguous ratings is based on our recently developed algorithm (Wickramaratna, Kubat, & Premaratne 2009). Although the technique was originally developed for association mining, we have shown that it can be used also for ratings-predictions. Surprisingly, it compares favorably with correlation analysis even when the data are "crisp." For data with ambiguities, we modified the correlation-based approach accordingly.

The reader will have noted that although we worked with real-world data, we had to add ambiguities "artificially"; this might leave the (false) impression that we targeted a problem that in reality does not exist. Our answer is that we are facing a kind of chicken-and-egg problem: since induction techniques are rarely capable of handling ambiguities, data providers tend to "sanitize" the data by removing the ambiguities; consequently, the data mining community lacks an incentive to investigate the related issues in the depth they deserve. Indeed, one of the reasons we embarked on this research was to break this vicious circle by drawing the attention of other scientists to these unfairly neglected problems.

Acknowledgment

The work was supported by NSF Grant IIS-0513702.

References

Blackman, S., and Popoli, R. 1999. *Design and Analysis of Modern Tracking Systems*. Norwood, MA: Artech House.

Herlocker, J. L.; Konstan, J. A.; Borchers, A.; and Riedl, J. T. 1999. An algorithmic framework for performing collaborative filtering. In *Ann. Int. ACM SIGIR Conf. on Res. and Dev. in Inf. Retrieval*, 230–237.

Herlocker, J. L.; Konstan, J. A.; Terveen, L. G.; and Riedl, J. T. 2004. Evaluating collaborative filtering recommender systems. *ACM Trans. on Inf. Syst.* 22(1):5–53.

Hewawasam, K. K. R. G. K.; Premaratne, K.; and Shyu, M.-L. 2007. Rule mining and classification in a situation assessment application: A belief theoretic approach for handling data imperfections. *IEEE Trans. Syst., Man, Cyber., Pt B* 37(6):1446–1459.

Jousselme, A.-L.; Grenier, D.; and Bosse, E. 2001. A new distance between two bodies of evidence. *Information Fusion* 2(2):91 – 101.

Li, Y., and Kubat, M. 2006. Searching for high-support itemsets in itemset trees. *Intell. Data Anal.* 10(2):105–120.

Research, G. 2007. Movielens data sets. Dept. of Comp. Sc. and Eng., Univ. of Minnesota, Twin Cities, MN [Online]http://www.grouplens.org/taxonomy/term/14.

Shafer, G. 1976. *A Mathematical Theory of Evidence*. Princeton Univ. Press.

Smets, P. 1999. Practical uses of belief functions. In Laskey, K. B., and Prade, H., eds., *Proc. Conf. Uncertainty in Art. Intell. (UAI'99)*, 612–621. Morgan Kaufmann.

Subasingha, S. P.; et al. 2008. *Data Mining: Foundations and Practice*, Springer. chapter Using Association Rules for Classification from Databases Having Class Label Ambiguities: Belief Theoretic Method, 539–562.

van Rijsbergen, C. J. 1979. *Information Retrieval*. London, UK: Butterworths.

Wickramarathne, T. L. 2008. A belief theoretic approach for automated collaborative filtering. Master's thesis, Dept. of Elect. and Comp. Eng., Univ. Miami, Coral Gables, FL.

Wickramaratna, K.; Kubat, M.; and Premaratne, K. 2009. Predicting missing items in shopping carts. *IEEE Trans. on Know. and Data Eng. (to appear, DOI:10.1109/TKDE.2008.229).*