DOI 10.1007/s10115-003-0089-6

Knowledge and Information Systems (2003) 5: 201–224

Knowledge and Information Systems

Feature Weighting and Instance Selection for Collaborative Filtering: An Information-Theoretic Approach*

Kai Yu[1,2], Xiaowei Xu[1,3], Martin Ester[2] and Hans-Peter Kriegel[2]

[1]Corporate Technology, Siemens AG, Munich, Germany
[2]Institute for Computer Science, University of Munich, Munich, Germany
[3]Information Science Department, University of Arkansas at Little Rock, Little Rock, Arkansas, USA

Abstract. Collaborative filtering (CF) employing a consumer preference database to make personal product recommendations is achieving widespread success in E-commerce. However, it does not scale well to the ever-growing number of consumers. The quality of the recommendation also needs to be improved in order to gain more trust from consumers. This paper attempts to improve the accuracy and efficiency of collaborative filtering. We present a unified information-theoretic approach to measure the relevance of features and instances. Feature weighting and instance selection methods are proposed for collaborative filtering. The proposed methods are evaluated on the well-known EachMovie data set and the experimental results demonstrate a significant improvement in accuracy and efficiency.

Keywords: Collaborative filtering; Data mining; Feature weighting; Instance-based learning; Instance selection; Recommender systems

1. Introduction

The tremendous growth of information gathered in E-commerce has motivated the use of information filtering and personalization technology. A major problem consumers face is how to find the desired product from the millions of products available. It is crucial for the vendor to find the consumer's preferences for products. Collaborative filtering (CF)-based recommender systems have emerged

* This work was performed in Corporate Technology, Siemens AG.
Received 28 July 2001
Revised 1 February 2002
Accepted 10 May 2002

in response to these problems (Resnick et al., 1994; Shardanand and Maes, 1995; Billsus and Pazzani, 1998; Breese et al., 1998).

CF-based recommender systems accumulate a database of consumer preferences, and use it to predict a particular consumer's preference for target products like music CDs, books, web pages, and movies. The consumer's preference can be recorded through either explicit votes or implicit usage/purchase history. Collaborative filtering can help E-commerce in converting web surfers into buyers by personalization of the web interface. It can also improve cross-sales by suggesting other products in which the consumer might be interested. In a world where an E-commerce site's competitors are only one or two clicks away, gaining consumer loyalty is an essential business strategy. Collaborative filtering can improve loyalty by creating a value-added relationship between supplier and consumer.

Collaborative filtering has been very successful in both research and practice. However, important research issues remain to be addressed in order to overcome two fundamental challenges in collaborative filtering (Sarwar, 2000). (1) Scalability: existing collaborative filtering algorithms can deal with thousands of consumers in a reasonable amount of time, but modern E-commerce systems need to handle millions of consumers efficiently; (2) accuracy: consumers need recommendations they can trust to help them find products they will like. If a consumer trusts a recommender system, purchases a product, but finds he or she does not like the product, the consumer will be unlikely to use the recommender system again.

This paper addresses these two challenges from a novel perspective by studying the problems of feature relevance and instance relevance in a unified information-theoretic framework. In order to improve the accuracy and scalability, a feature relevance measure and an instance relevance measure are applied to weight the features and select relevant instances. Empirical analysis shows that the proposed method is successful.

In Section 2, we briefly review related work in collaborative filtering and instance-based learning (IBL). In Sections 3 and 4, we study feature relevance and instance relevance respectively. Feature weighting and instance selection are integrated in a unified framework to improve the performance of collaborative filtering in Section 5. Section 6 reports an empirical evaluation of the proposed method. The paper ends with a summary and a discussion of some interesting future work.

2. Related Work

In this section, we review related work in collaborative filtering. We focus on instance-based collaborative filtering algorithms which belong to a class of instance-based learning algorithms (IBL). Therefore, we also give some background on IBL including the use of feature weighting, instance weighting and instance selection to improve the performance of IBL.

2.1. Collaborative Filtering

The task in collaborative filtering is to predict the preference of an active consumer for a given product based on a consumer preference database, which is normally

represented as a consumer–product matrix with each entry $v_{u,i}$ indicating the vote of consumer u for product i. There are two general classes of collaborative filtering algorithms: instance-based methods and model-based methods.

The instance-based algorithm (Resnick et al., 1994; Shardanand and Maes, 1995) is the most popular prediction technique in collaborative filtering applications. The basic idea is to compute the active consumer's rating of a product as a similarity-weighted average of the ratings given to that product by other consumers. Specifically, the prediction $P_{a,i}$ of active consumer a's ratings of product i is given by:

$$P_{a,i} = \bar{v}_a + k \sum_{b \in \text{neighborhood}(a,\mathscr{T}_i)} r(a,b)(v_{b,i} - \bar{v}_b) \tag{1}$$

where $\mathscr{T}_i$, the training set for product i, includes all the consumers who have rated product i, and neighborhood$(a, \mathscr{T}_i)$ returns all the neighbors of active consumer a in $\mathscr{T}_i$, where neighbors can be defined as all the consumers in $\mathscr{T}_i$ (Breese et al., 1998), or the results of k-nearest neighbor query (Herlocker et al., 1999) or range query Shardanand and Maes (1995). $\bar{v}_a$ is the mean vote for consumer a, $v_{b,i}$ is consumer b's rating of i, $r(a,b)$ is the similarity measure between consumer a and b, and k is a normalizing factor such that the absolute values of the weights sum to unity. The Pearson correlation coefficient is the most popular similarity measure, which is defined as (Resnick et al., 1994):

$$r(a,b) = \frac{\sum_{j \in \text{overlap}(a,b)} (v_{a,j} - \bar{v}_a)(v_{b,j} - \bar{v}_b)}{\sqrt{\sum_{j \in \text{overlap}(a,b)} (v_{a,j} - \bar{v}_a)^2 \sum_{j \in \text{overlap}(a,b)} (v_{b,j} - \bar{v}_b)^2}} \tag{2}$$

where overlap(a,b) indicates that the similarity between two consumers is computed over the products which they both rated. Shardanand and Maes (1995) claimed better performance by computing similarity using a constrained Pearson correlation coefficient, where the consumer's mean votes are replaced by a constant, the midpoint of the rating scale.

Instance-based methods have the advantages of being able to rapidly incorporate the most up-to-date information and provide relatively accurate predictions (Breese et al., 1998), but they suffer from poor scalability for large numbers of consumers. This is because the search for all similar consumers is slow in large databases.

Model-based collaborative filtering, in contrast, uses the consumer preference database to learn a model, which is then used for predications. The model can be built off-line over several hours or days. The resulting model is very small, very fast, and essentially as accurate as instance-based methods (Breese et al., 1998). Model-based methods may prove practical for environments in which consumer preferences change slowly with respect to the time needed to build the model. Model-based methods, however, are not suitable for environments in which consumer preference models must be updated rapidly or frequently.

2.2. Instance-Based Learning

IBL algorithms (Aha et al., 1991) compute a similarity (distance) between a new instance and stored instances when generalizing. One of the most straightforward

IBL algorithms is the nearest neighbor algorithm (Cover and Hart, 1967; Hart, 1968). During generalization, instance-based learning algorithms use a distance function to determine how close a new instance is to each stored instance, and use the nearest instance or instances to predict the target. Other instance-based machine learning paradigms include instance-based reasoning (Stanfill and Waltz, 1986), exemplar-based generalization (Saltzberg, 1991; Wettschereck et al., 1995), and case-based reasoning (CBR) (Kolodner, 1993).

The prediction accuracy of many IBL algorithms is highly sensitive to the definition of the distance function. Many feature weighting methods have been proposed to reduce this sensitivity by parameterizing the distance function with feature weights. Wettschereck et al. (1995) review and empirically compare some feature weighting methods. Feature weighting and feature selection have also received wide attention in the machine learning community (Blum and Langley, 1997). In applications of vector similarity in information retrieval, word frequencies are typically modified by the inverse document frequency (Saloton and McGill, 1983). The idea is to reduce weights for commonly occurring words, capturing the intuition that they are not useful in identifying the topic of a document, while words that occur less frequently are more indicative of the topic. Breese et al. (1998) applied an analogous transformation to votes in a collaborative filtering database, which is termed inverse user frequency. The idea is that universally liked products are not as useful in capturing similarity as less common products. So inverse user frequency weight is defined as follows:

$$w_j = \log \frac{n}{n_j} \tag{3}$$

where n_j is the number of consumers who have voted for product j, and n is the total number of consumers in the database. Note that if everyone has voted on product j, then the weight of j is zero.

The accuracy of IBL algorithms can be further improved by instance weighing. The idea is to weight each instance based on its ability to reliably predict the target of an unseen instance (Saltzberg, 1990, 1991). The weight of an instance defines an area within the feature space. A reliable instance is assigned to a bigger area. An unreliable instance represents either noise or an 'exception' – thus, it will receive a smaller area. For the instance to be used in prediction, the target instance must fall within its area. Anand et al. (1998) introduced a generalization of exception spaces. The resulting exception spaces are called Knowledge INtensive exception Spaces or KINS. KINS removes the restriction on the geometric shape of exception spaces.

Since IBL algorithms search through all available instances to classify (or predict) a new instance, it is also necessary to decide what instances to store for generalization in order to reduce excessive storage and time complexity, and to possibly even improve accuracy. Therefore instance selection has become an important topic in IBL and data mining (Pradhan and Wu, 1999; Wilson and Martinez, 2000; Liu and Motoda, 2001). Some algorithms seek to select representative instances, which could be border points (Aha et al., 1991) or central points (Zhang, 1992). The intuition behind retaining border points is that *internal* points do not affect the decision boundaries as much as border points, and thus can be removed. However, noisy points are prone to be judged as border points and added to the training set. As for central points, selection should be carefully done since the decision boundary lies halfway between two nearest instances of different classes. Another class of algorithms attempts to remove noisy

points before selecting representative instances (Wilson and Martinez, 2000). For example, DROP3 uses a simple noise-filtering pass: any instance misclassified by its k nearest neighbors is removed (Wilson and Martinez, 2000). For almost all the algorithms mentioned above, classification has to be performed at least once in each step of removing or adding an instance, so it has a rather high computational complexity. Recently, Smyth and Mckenna (1999) proposed an instance selection method for CBR. They introduce the concept of competence groups and show that every case-base is organized into a unique set of competence groups, each of which makes its own contribution to competence. They devise a number of strategies to select a footprint set (a union of a highly competent subsets of cases in each group). Patterson et al. (2002) presented a clustering-based instance selection method for CBR. They use the k-means clustering algorithm to group cases based on their degree of similarity. When a new case is presented, the closest cluster is identified and the generalization is performed only on the selected cluster.

In IBL paradigm, the purpose of feature or instance weighting is to improve the accuracy, while instance selection is used to reduce the storage and speed up the generalization. We propose using feature weighting and instance selection for collaborative filtering. Many studies have investigated feature weighting and instance selection independently. However, these two topics seem closely related. Blum and Langley (1997) pointed out that more studies need to be conducted to increase the understanding of this relationship. Our work is unique in that we study this relationship in a unified information-theoretic framework.

3. Feature Weighting Methods

Collaborative filtering is built on the assumption that a good way to predict the preference of an active consumer for a target product is to find other consumers who have similar preferences and use their votes for that product to make a prediction. The similarity measure is based on preference patterns of consumers. A consumer's votes on the product set not including the target product can be regarded as features of this consumer. The introduction of feature weighting into collaborative filtering may improve the accuracy of prediction since it can enhance the role of relevant products while reducing the impact of irrelevant products. We define the feature-weighted constrained Pearson coefficient as:

$$(a,b) = \frac{\sum_{j\in \text{overlap}(a,b)} W_{i,j}^2 (v_{a,j} - v_0)(v_{b,j} - v_0)}{\sqrt{\sum_{j\in \text{overlap}(a,b)} W_{i,j}^2 (v_{a,j} - v_0)^2 \sum_{j\in \text{overlap}(a,b)} W_{i,j}^2 (v_{b,j} - v_0)^2}} \tag{4}$$

where $W_{i,j}$ represents the weight of product j with respect to the target product i, and v_0 is a constant representing the midpoint of votes. When $W_{i,j} = 1$ equation (4) is equal to the constrained Pearson coefficient. In this paper v_0 is set at 3 since analysis of the database used shows that it is the most frequent rating in the 6-point scale from 0 to 5.

3.1. Feature Relevance

The idea of instance-based CF is to predict the target (vote) based on the knowledge of some other features (votes). So some kind of mutual correlation

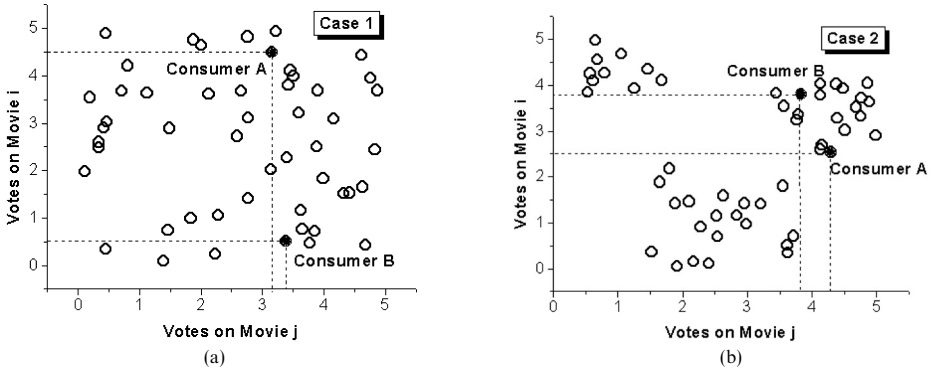

Fig. 1. Distribution of consumer votes on two movies in Example 3.1.

between features and the target should be investigated. If the vote for the target product i is found to be highly dependent on the vote for some product j, clearly a larger weight should be assigned to j. For a better understanding, let us consider the next example.

Example 3.1. As shown in Fig. 1, if 50 consumers give votes for movie i and movie j, let us consider two different situations, case 1 and case 2. In case 1, we find consumers are nearly uniformly distributed in the movie–movie vote space. If A and B are two arbitrary consumers who have similar ratings for movie j, it does not necessarily indicate that they also have similar ratings for movie i. In case 2, however, we find that those consumers who dislike movie j always like movie i, while those consumers who like movie j always rate the other one just above average. This indicates that in case 2 movie j should play an important role in inferring consumer preference for movie i, while in case 1 it is not so useful.

The *dependence of product i on product j* can be formally defined by the following conditional probability:

$$p\left(|v_{A,i} - v_{B,i}| < e \mid |v_{A,j} - v_{B,j}| < e\right) \tag{5}$$

where A and B represent two arbitrary consumers and e is a threshold. If the difference between two votes is less than e, then the two votes are considered *close*. The above conditional probability indicates the probability of two arbitrary consumers having close preference for product i given the condition that the two consumers have close preference for product j.

We develop an information-theoretic measure that is equivalent to the above probabilistic dependence definition in the case of discrete voting. First we introduce the concept of *mutual information*. In information theory, mutual information represents a measure of statistical dependence between two random variables X and Y with associated probability distributions $p(x)$ and $p(y)$ respectively. Following Shannon theory (Shannon, 1948) the mutual information between X and Y is defined as:

$$I(X;Y) = \sum_x \sum_y p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right) \tag{6}$$

Furthermore, mutual information can be equivalently transformed into the

following formulas:

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \\ &= H(X) + H(Y) - H(X,Y) \end{aligned} \tag{7}$$

where $H(X)$ is the entropy of X, $H(X|Y)$ is the conditional entropy of X given Y and $H(X,Y)$ is the joint entropy of two random variables. The definition of the conditional entropy, the joint entropy and the proof of the above equations can be found in Deco and Obradovic (1996). The equations above indicate that mutual information also represents a reduction of entropy (uncertainty) of one variable given information about the other variable. In the following theorem, we will show that when the voting scale is discrete, mutual information is equivalent to the probabilistic definition of dependence.

Theorem 3.1. Let $P(V_i)$, $P(V_j)$, and $P(V_i, V_j)$ be the margin and joint distributions of votes for two products i, j, and $e = 1$ the interval of discrete vote value, $0, 1, \ldots, N$; assume that $P(V_i)$ and $P(V_j)$ are fixed, if A and B are two arbitrary consumers who have voted for both products, then $I(V_i; V_j)$ increases as *dependence* increases, which means the differential of *dependence* defined by equation (5) with respect to the mutual information $I(V_i; V_j)$ is always positive.

$$\frac{d[p(|v_{A,i} - v_{B,i}| < e \mid |v_{A,j} - v_{B,j}| < e)]}{d[I(V_i; V_j)]} > 0 \tag{8}$$

Proof. (See Appendix) □

The above theorem shows that large mutual information between the votes for two products reflects a high dependence between them. Therefore, the analysis encourages us to apply mutual information in computing the weighted similarity measure equation (4) between consumers, where the weight of product j with respect to the target product i is given by the following:

$$W_{i,j} = I(V_i; V_j) \tag{9}$$

If there is a total m products in the data set, the computation results in an $m \times m$ matrix.

3.2. Estimation of Mutual Information

We use the following equation to estimate the mutual information between two products:

$$I(V_i; V_j) = H(V_i) + H(V_j) - H(V_i, V_j) \tag{10}$$

where

$$H(V_i) = -\sum_{k=0}^{N} p(v_i = k) \log_2 p(v_i = k)$$

$$H(V_j) = -\sum_{k=0}^{N} p(v_j = k) \log_2 p(v_j = k)$$

$$H(V_i; V_j) = -\sum_{k=0}^{N} \sum_{l=0}^{N} p(v_j = l, v_i = k) \log_2 p(v_j = l, v_i = k)$$

In the above equations, $H(V_i, V_j)$ is the joint entropy between two products. k and l are possible vote values (in our experiment, k, $l = 0, 1, 2, 3, 4, 5$). Since not all the consumers have voted for the two products, in equation (10) the entropy is calculated using the consumers who rated the corresponding product, while the joint entropy is calculated using the consumers who rated both products. The calculation involves probability estimation, which has been a crucial task in machine learning (Cestnik, 1990). One important characteristic of consumer preference databases is that they contain many missing values. A straightforward approach to probability estimation might be unreliable. When observations of a random event are limited, a Bayesian approach to estimate the unknown probability is m-Estimation (Cestnik, 1990), which has proven effective and been widely used in machine learning (Mitchell, 1997). Suppose that out of n examples the event whose probability we are attempting to estimate occurs r times. Then the m-Estimation is given by

$$p = \frac{r + m \cdot P}{n + m} \tag{11}$$

Here P is our prior estimate of the probability that we wish to determine, and m is a constant, which determines how heavily to weight P relative to the observed data. When the number of observations n is very small, the estimated probability will be close to the prior value. The best value for m can be determined experimentally. However, if the product set is large, too many experiments are required, making this method impractical. In this paper we used a very simplified method, setting $m = \sqrt{n}$ (Cussens, 1993). Therefore the probabilities are estimated as follows:

$$p(v_i = k) = \frac{r_i^k + \sqrt{n_i} \cdot P(v = k)}{n_i + \sqrt{n_i}} \tag{12}$$

$$p(v_j = l) = \frac{r_j^l + \sqrt{n_j} \cdot P(v = k)}{n_j + \sqrt{n_j}} \tag{13}$$

$$p(v_j = l, v_i = k) = \frac{r_{i,j}^{k,l} + \sqrt{n_{i,j}} \cdot p(v_i = k) \cdot p(v_j = l)}{n_{i,j} + \sqrt{n_{i,j}}} \tag{14}$$

where $k, l = 0, 1, \ldots, N$. In equation (12), n_i denotes the number of consumers who rated product i, and r_i^k the number of consumers who rated product i by value k, while the a priori probability $P(v = k)$ is derived from the whole data set regardless of any specific product. In equation (14), $n_{i,j}$ denotes the number of consumers who rated both product i and j, $r_{i,j}^{k,l}$ denotes the number of consumers who rated product i by value k and meanwhile rated product j by value l. We determine the a priori joint probability assuming that the probabilities of votes on two products are independent. If the average number of overlapping consumers between two products is n, and there is a total of m products in the training data set, the computational complexity for calculating the mutual information between all pairs of products is $O(nm^2)$.

4. Selecting Relevant Instances

The collaborative filtering algorithms first compute the correlation coefficient between the active consumer a and all other consumers; then all consumers

whose coefficient is greater than a certain threshold (which is set to 0 in our work) are identified and the weighted average of their votes for the target product is calculated. Obviously the computational complexity is linear to the number of advisory consumers, who cast a vote for the predicted product (size of $\mathscr{T}_i$ in equation (1)). One way to speed up recommendation determination is to reduce the number of advisory consumers. This can be done through random sampling or data focusing techniques (Ester et al., 1995); however, the use of these methods includes the risk of sacrificing quality through information loss. In response to this challenge, we propose a method for reducing the training data set by selecting a highly relevant instance set $\mathscr{S}_i \subseteq \mathscr{T}_i$, and rewriting equation (1) as the following:

$$P_{a,i} = \bar{v}_a + k \sum_{b \in \text{neighborhood}(a,S_i)} \text{r}(a,b)(v_{b,i} - \bar{v}_b) \tag{15}$$

4.1. Relevance of Instances

In this section, we study the relevance of instances (or consumers) in an information-theoretical framework and try to remove the irrelevant ones to improve the quality and salability of collaborative filtering. Our basic idea is that for an advisory consumer with his or her preference records, if the votes on other products cannot provide enough information to support why he or she cast the vote on the target product, then this consumer will not be useful in aiding the learner to search the hypothesis space. In the rest of this section, we suppose we are attempting to predict consumer votes on a target product i, and hence the instance set $\mathscr{T}_i$ we consider only contains the consumers who have voted on product i. Note that in collaborative filtering the target product to be predicted can be any one in the data set. If a consumer $u \in \mathscr{T}_i$, then u is called an *instance with respect to target product i*, and his or her rating of the target product is called the *instance's value*, denoted by $v_{u,i}$, while his or her ratings of the rest of the voted product set $\mathscr{F}_{u,i}$, denoted by $d_{u,i}$, are called the *instance description with respect to target product i*, and $\mathscr{F}_{u,i}$ is called the *instance feature set with respect to target product i*. A consumer preference database always has a large proportion of missing values (e.g. up to 98% in the EachMovie data set) and each consumer rated a unique list of products; therefore in the learning task of collaborative filtering, different instances have different feature sets. In the following, we introduce a measure of instance relevance, and interpret it from Bayesian learning's point of view.

Definition 4.1. (Rationality of instance) Given an instance $u \in \mathscr{T}_i$ represented by its description $d_{u,i}$ over its feature set $\mathscr{F}_{u,i}$ and a target value $v_{u,i}$, the *rationality of instance u with respect to target product i*, denoted by $R_{u,i}$, is the uncertainty reduction of instance value $v_{u,i}$ given knowledge of description $d_{u,i}$, which can be encoded into bits:

$$\begin{aligned} R_{u,i} &= H(v_i = v_{u,i}) - H(v_i = v_{u,i} | v_{u,\mathscr{F}_{u,i}} = d_{u,i}) \\ &= -\log_2 p(v_i = v_{u,i}) + \log_2 p(v_i = v_{u,i} | v_{u,\mathscr{F}_{u,i}} = d_{u,i}) \end{aligned} \tag{16}$$

A typical method for deciding a priori uncertainty $H(v_i = v_{u,i})$ is to assume uniform priors; that is, if the instance value has N possible values we set $H(v_i = v_{u,i}) = -log_2 1/N$. If a large number of instances are given, then a statistical

approach can be applied. For example, given a consumer with a score 4 for the target movie i, we set the a priori uncertainty to be 1 bit if 50% of the consumers who rated the movie give it a score of 4. Furthermore, if it is inferred that the consumer has a probability of 75% to vote 4 for the target movie after we know his or her votes on other movies – the instance description – then according to equation (16) the consumer's rationality with respect to movie i is $-\log_2 0.5 + \log_2 0.75 = 0.59$ bit. From an intuitive perspective, the definition of *rationality* measures the relation between an instance's description and its value. In the following paragraph, we interpret rationality from the perspective of Bayesian learning and show how this relation will play an important role in evaluating an instance's relevance for learning.

In a Bayesian learning scenario for predicting consumer ratings of the target product i, the learner considers some set of candidate hypotheses $\mathscr{H}_i$ and wants to finding a *maximum a posteriori* (MAP) hypothesis $h_i \in \mathscr{H}_i$ given the observed instance set $\mathscr{T}_i$:

$$h_i^{MAP} = \arg\max_{h_i \in \mathscr{H}_i} p(h_i \mid \mathscr{T}_i) = \arg\max_{h_i \in \mathscr{H}_i} \frac{p(\mathscr{T}_i \mid h_i) p(h_i)}{p(\mathscr{T}_i)} \tag{17}$$

Suppose h_i^{real} is the real function that the learner is looking for. The instance selection problem can then be interpreted as finding an optimal subset of instances $\mathscr{S}_i \subseteq \mathscr{T}_i$ to maximize the a posteriori probability of h_i^{real}:

$$\begin{aligned} \mathscr{S}_i^{opt} &= \arg\max_{\mathscr{S}_i \subseteq \mathscr{T}_i} p(h_i^{real} \mid \mathscr{S}_i) \\ &= \arg\min_{\mathscr{S}_i \subseteq \mathscr{T}_i} H(\mathscr{S}_i \mid h_i^{real}) - H(\mathscr{S}_i) + H(h_i^{real}) \\ &= \arg\max_{\mathscr{S}_i \subseteq \mathscr{T}_i} H(\mathscr{S}_i) - H(\mathscr{S}_i \mid h_i^{real}) \end{aligned} \tag{18}$$

It is reasonable to assume that each instance is drawn independently and each instance value is independent of its description when the hypothesis is absent; this give us:

$$\begin{aligned} \mathscr{S}_i^{opt} &= \arg\max_{\mathscr{S}_i \subseteq \mathscr{T}_i} \sum_{u \in \mathscr{S}_i} H[(v_{u,i}, v_{u,\mathscr{F}(u,i)}) - H(v_{u,i}, v_{u,\mathscr{F}(u,i)} \mid h_i^{real})] \\ &= \arg\max_{\mathscr{S}_i \subseteq \mathscr{T}_i} \sum_{u \in S_i} [H(v_{u,i} \mid v_{u,F(u,i)}) - H(v_{u,\mathscr{F}(u,i)}) \\ &\quad - H(v_{u,i} \mid h_i^{real}, v_{u,\mathscr{F}(u,i)}) + H(v_{u,\mathscr{F}(u,i)})] \\ &= \arg\max_{\mathscr{S}_i \subseteq \mathscr{T}_i} \sum_{u \in \mathscr{S}_i} [H(v_{u,i}) - H(v_{u,i} \mid h_i^{real}, v_{u,\mathscr{F}(u,i)})] \end{aligned} \tag{19}$$

h_i^{real} is the underlying function which bridges the gap between the instance description and the instance value, and thus can be dropped from the equations. Then the instance rationality (in Definition 4.1) surprisingly is expressed as:

$$S_i^{opt} = \arg\max_{\mathscr{S}_i \subseteq \mathscr{T}_i} \sum_{u \in \mathscr{S}_i} [H(v_{u,i}) - H(v_{u,i} \mid v_{u,\mathscr{F}_{u,i}})] = \arg\max_{\mathscr{S}_i \in \mathscr{T}_i} \sum_{u \in S_i} R_{u,i} \tag{20}$$

The above equation clearly shows that the instance rationality plays an important role in machine learning. Namely, an instance with higher rationality contributes more to increasing the a posteriori probability of the real hypothesis, and

accordingly decreases other hypotheses' a posteriori probabilities, while an instance with lower or even negative rationality contributes little to or even reduces the a posteriori probability of the real hypothesis, and therefore is identified as irrelevant or noisy instance. The calculation of the instance rationality needs estimates the a posteriori probability of the instance value, since h_i^{real} is unknown in practice. Theoretically any learning approach explicitly addressing probabilities, such as naive Bayesian method (Mitchell, 1997) can be applied. However, collaborative filtering is a special learning task, in which the target product may be any in the product list. For example, in the EachMovie data set there are 1628 movies to be predicted. Considering each vote has $N = 6$ possible values, naive Bayesian method needs to calculate $1628 * 1628 * 6 * 6$ probabilities and maintain them in the memory, which requires almost 380 Mbytes if each probability needs 4 bytes. Furthermore, it is required to run the leave-one-out learning approach for each of the millions of entries in the data set. To avoid excessive computation, we further introduce a weaker definition for the instance rationality and greatly simplify its estimation. An important advantage of the new definition is that it only involves the mutual information introduced in Section 3.1 and thus enables us to treat feature relevance and instance relevance in a unified information-theoretic framework.

Definition 4.2. (General rationality of instance) Given an instance $u \in \mathscr{T}_i$ with its feature (product) set $\mathscr{F}_{u,i}$ and the target product i, if entropy $H(V_i)$ is a priori uncertainty of the votes on the product i, then *general rationality of instance u with respect to target product i*, denoted by $R^*_{u,i}$, is the uncertainty reduction of V_i given knowledge of $V_{\mathscr{F}_{u,i}}$, which are the votes on feature set $\mathscr{F}_{u,i}$. It can be encoded into bits:

$$R^*_{u,i} = H(V_i) - H(V_i \mid V_{\mathscr{F}_{u,i}}) = I(V_i; V_{\mathscr{F}_{u,i}}) \tag{21}$$

General rationality is derived from the rationality Definition 4.1 by removing the specification of vote values. Note that the instance relevance in the new definition only depends on which products the consumer rated, but has nothing to do with the vote values. This point is useful in collaborative filtering since each consumer rated a different set of products. $R^*_{u,i}$ is a generalization of $R_{u,i}$. If $R^*_{u,i}$ is high then $R_{u,i}$ is very likely to be high. Therefore, general rationality can be viewed as a rough approximation of the former one and also provides a quality measure to the instance relevance. The following theorem shows that the computation of general rationality can be greatly simplified under some assumptions.

Theorem 4.1. Given an instance $u \in \mathscr{T}_i$ with its feature (product) set $\mathscr{F}_{u,i}$, if each feature $j \in \mathscr{F}_{u,i}$ is independent of the other features whether given V_i or not, then the following conclusion holds:

$$R^*_{u,i} = \sum_{j \in \mathscr{F}(u,i)} I(V_i; V_j) \tag{22}$$

Proof. (See the Appendix.) □

Theorem 4.1 provides an easy way to calculate the general rationality of an instance under some assumptions. A very interesting point is that Theorem 4.1 shows that instance relevance is intimately related to feature relevance. The mutual information matrix in equation (9) for feature weighting can be used

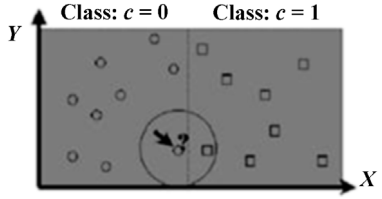

Fig. 2. An NN classifier biased by an irrelevant feature Y.

directly here. The assumption of feature independence given the instance value (or label) has been widely adopted in many literatures, like naive Bayesian classifier (Mitchell, 1997) and expectation maximum (EM) clustering (Witten and Frank, 1999). It has been reported that the naive Bayesian classifier under this assumption outperforms many other learning methods in many applications (Domingos and Pazzani, 1996). However, the assumption that the features are independent without being given the instance value seems to conflict with our work on measuring the relevance between products. In our experiment, we found that the mutual information between products is always close to zero, indicating the relevance between products is rather weak.

Instance-based learning (IBL) methods always suffer from the effect of irrelevant attributes, as does instance-based collaborative filtering. As shown in Fig. 2, let us consider an example of nearest neighbor (NN) classifier, where there are two independent attributes X and Y such that $H(C|X) = 0$ and $H(C|Y) = H(C)$. If only X is applied for classification, the instances can be well classified. But once X and Y are considered together the accuracy will degrade. In both cases the general rationality R_1^* and R_2^* are the same:

$$R_1^* = I(C;X) = H(C), R_2^* = I(C;X,Y) = I(C;X) + I(C;Y) = H(C)$$

This example shows that we should consider other issues besides rationality. Existence of irrelevant features might not decrease the rationality of an instance but still might mislead the distance measure in IBL. Accordingly instance-based collaborative filtering has a similar problem. Suppose two consumers with the same rationality are given; we argue that the one with fewer voted products should be preferred. The reasons are: (1) since each instance can be viewed as a specific rule (Domingos, 1996), we should prefer the shorter one, following Occam's razor (Mitchell, 1997); (2) it is indicated in Theorem 4.1 that each feature contributes a little to the rationality; therefore the instance with more voted products is likely to have more irrelevant features. Here we applied a simple heuristic to penalize the instances that have a complex description. The *rationality strength of an instance* is defined as follows:

$$R_{u,i}^{strength} = \frac{1}{|\mathscr{F}_{u,i}|} R_{u,i}^* \tag{23}$$

where $|\mathscr{F}_{u,i}|$ is the number of features in $\mathscr{F}_{u,i}$. $R_{u,i}^{strength}$ can be interpreted as the average feature relevance of instance u. In this sense it is interesting that a relevant instance is one with many relevant features. Given a pool of instances $\mathscr{T}_i$, we will select a subset $\mathscr{S}_i \subseteq \mathscr{T}_i$ such that each instance $u \in \mathscr{S}_i$ has a high general rationality and a high rationality strength.

5. Feature Weighting and Instance Selection for Collaborative Filtering

5.1. Feature Relevance and Instance Relevance in an Information-Theoretic Framework

Using Sections 3 and 4 as a basis, we can interpret feature relevance and instance relevance in an information-theoretic framework. In particular, we investigate four related issues: feature selection and weighting, as well as instance selection and weighting. We also come to explain why we choose feature weighting and instance selection for collaborative filtering. We still suppose we are attempting to predict consumer votes on a target product i.

Feature relevance. As described in Section 3, the relevance of feature V_i (votes on product j) with respect to V_i (votes on target product i) is the mutual information between them:

$$W_{i,j} = I(V_i; V_j) \tag{24}$$

As described in equation (4), the relevance measure can serve as a feature weighting method and can be applied to feature selection. However, although a preference data set has a long product list, each consumer normally rated a rather small portion of it. For example, each consumer rated an average of about 30 of the 1628 movies in the EachMovie data set. In such a situation further reducing the feature number might lead to poor prediction quality. On the other hand, our investigation on the EachMovie data set showed that there was not a dramatic difference in feature relevance. This indicates that it is difficult to distinguish relevant from irrelevant features and accuracy might decrease if feature selection is performed. Therefore we chose feature weighting to improve the accuracy of collaborative filtering.

Instance relevance. As described in Section 4, the relevance of instance (consumer) u with respect to V_i is described by the general rationality and the rationality strength. Both are in the form of mutual information:

$$R^{*}_{u,i} = \sum_{j \in \mathscr{F}_{u,i}} I(V_i; V_j) \tag{25}$$

$$R^{strength}_{u,i} = \frac{1}{|\mathscr{F}_{u,i}|} \sum_{j \in \mathscr{F}_{u,i}} I(V_i; V_j) \tag{26}$$

Similarly there are two possibilities: instance weighting and instance selection. Instance weighting normally aims at improving the accuracy. In CF, the number of consumers increases explosively while the number of products remains relatively stable and is much lower than that of consumers. For instance, there are 72,916 consumers and 1623 movies in the EachMovie dataset. Therefore we argue that it is more desirable to reduce the number of consumers to improve the scalability and efficiency of collaborative filtering, while maintaining or improving upon a certain level of accuracy.

Interestingly, feature relevance and instance relevance demonstrate a very close relationship: an instance is relevant if its features are relevant. This conclusion is useful in collaborative filtering since the data set is very sparse and each consumer has a unique feature (product) set.

5.2. Proposed Approach to Feature Weighting and Instance Selection in Collaborative Filtering

According to Section 4, we should select consumers with enough general rationality and pick out the consumers with a higher strength from those selected. This approach is complex to apply in practice. Since most consumers in our experiment give some tens of votes, roughly speaking, if a consumer's general rationality is low, he or she cannot be of a high rationality strength. Thus we select consumers based only on the strength. As a result of instance selection, in addition to the original consumer preference database, we maintain an index table of selected consumers for every target product. During the prediction phase, we use feature weighting and instance selection to improve the accuracy, efficiency and scalability of collaborative filtering. In summary, our algorithm proceeds in the following steps:

1. Based on the training database, estimate the mutual information between votes on each pair of products and produce a matrix described by equation (9) or (24).
2. For each target product i, sort all the consumers $u \in \mathscr{T}_i$ in descending order with respect to rationality strength and select the top $min(MIN\_SIZE, \mathscr{T}_i \times r)$ consumers according to a sampling rate r, where MIN_SIZE is set to 150 to avoid over-reduction. This results in an index table of the selected training consumer set $\mathscr{S}_i$.
3. As described in equations (4) and (15), in the prediction phase we calculate the weighted constrained Pearson correlation between query consumer a and every selected consumer $u \in \mathscr{S}_i$, then search a's neighbors whose similarity to a is greater than zero. Finally a weighted average of the votes of similar consumers is calculated.

If we have n consumers and m products in the original training data set, the computational complexity of the training phase (step 1 and 2) is $O(nm^2) + O(nm) + O(nlogn)$. With a sampling rate r, the speed-up factor of prediction is expected to be $1/r$.

6. Empirical Analysis

In this section, we report results of an experimental evaluation of our proposed feature weighting and instance selection techniques for collaborative filtering. We describe the data set used, the experimental methodology, as well as the performance improvement compared with collaborative filtering without feature weighting and instance selection.

6.1. The EachMovie Database

We ran experiments using the well-known EachMovie[1] data set, which was part of a research project at the Systems Research Center of Digital Equipment Corporation. The database contains votes from 72,916 consumers on 1628 movies.

[1] For more information see http://www.research.digital.com/SRC/EachMovie/.

Consumer votes were recorded on a numeric six-point scale (we transfer it to 0, 1, 2, 3, 4, and 5). Although 72,916 consumers are available, we restrict our analysis to 35,527 consumers who gave at least 20 ratings.[2] Moreover, to speed up our experiments, we randomly select 10,000 consumers from 35,527 consumers and divide them into a training set (8000 consumers) and a test set (2000 consumers).

6.2. Metrics and Methodology

As applied in Breese et al. (1998), we also employ two protocols: *All but One*, and *Given K*. In the first protocol, we randomly hide an existing vote for each test consumer, and try to predict its value given all the other votes that the consumer has given. The *All but One* experiments are indicative of what might be expected of the algorithms under steady state usage where the database has accumulated a fair amount of data about a particular consumer. In the second protocol, *Given K*, we randomly select K votes from each test consumer as the observed votes, and then attempt to predict the remaining votes. It allows us to determine the performance when a consumer is new to a particular recommender system.

We use *mean absolute error* (MAE) and *e-accuracy* to evaluate the accuracy of prediction. MAE is the average difference between the actual votes and the predicted votes. This metric has been widely used in previous work (Resnick et al., 1994; Shardanand and Maes, 1995; Breese et al., 1998; Herlocker et al., 1999). *e-accuracy* is the percentage of tests whose absolute error is less than e. We believe it provides more knowledge about the distribution of error. In particular, when e is set to 0.5 the rounded value of the prediction exactly equals the actual vote. In addition, Shardanand and Maes (1995) argue that CF accuracy is most crucial when predicting extreme ratings (very high or very low) for products. Intuitively, since the goal is to provide recommendations, high accuracy on the high-rated and low-rated products is most preferred. Therefore we also investigate the accuracy in predicting extreme votes (*Extremes*), where the actual vote is 0, 1, 2, or 5. (Our study shows more than 50% of votes are 3 or 4.) For efficiency measurement, we use the *average prediction time per vote*, which should be linearly related to the size of the selected instance set. To get a reliable efficiency measurement, each test was repeated 10 times and then the mean calculated. We applied *movie average* and *constrained Pearson* for comparison. In *movie average*, we use the mean vote received by the target movie i as our prediction result. In *constrained Pearson* we set the mean vote to 3. The Pearson correlation coefficient between the active consumer a and all the other consumers in the instance set is calculated. All consumers whose coefficients are above 0 are then identified as neighbor consumers. Finally a weighted average of the votes on movie i is computed. In addition, for our empirical study on feature weighting and instance selection, we applied several other feature weighting and instance selection methods for comparison, whose details are described in the next subsections.

[2] This is because we want to evaluate our methods for the protocol of *Given K* (see Section 6.2) with k in the range of 10–20.

Table 1. Performance of feature weighting methods.

Method (Feature weighting)		All			Extremes		
		MAE	0.5 Accu.	1.0 Accu.	MAE	0.5 Accu.	1.0 Accu.
Movie average		1.10	27.1%	52.5%	1.59	6.51%	24.5%
All but One	Con. Pearson	0.982	31.3%	58.4%	1.40	10.0%	31.6%
	Inv. user freq.	0.994	31.3%	58.8%	1.41	9.93%	32.5%
	Entropy	0.979	31.6%	58.9%	1.39	10.3%	32.6%
	Mutual info.	0.938	34.1%	61.2%	1.30	12.8%	39.7%
Given 10	Con. Pearson	1.02	30.2%	56.0%	1.46	9.10%	28.1%
	Inv. user freq.	1.03	29.6%	56.3%	1.47	7.96%	28.1%
	Entropy	1.02	30.0%	56.3%	1.46	9.00%	28.6%
	Mutual info.	1.01	30.8%	56.8%	1.43	9.93%	30.6%
Given 20	Con. Pearson	1.00	30.8%	57.7%	1.43	9.82%	30.8%
	Inv. user freq.	1.02	30.1%	57.7%	1.44	9.41%	31.1%
	Entropy	1.00	31.3%	57.9%	1.43	10.1%	31.2%
	Mutual info.	0.982	32.6%	59.2%	1.37	12.4%	36.0%

6.3. Performance of Feature Weighting

We tested the proposed feature weighting method introduced in Section 3, as well as two other feature weighting approaches: *inverse user frequency*-based and *entropy*-based weighting. The inverse user frequency method (Breese et al., 1998) is described by equation (3). The idea is that popular movies are not as useful in capturing consumer preference as less popular movies. Here we applied entropy as another weighting method, because a movie receiving very diverse votes should be much more useful in capturing consumer preference than a movie receiving only similar votes. The weight of a movie j is calculated by:

$$W_{i,j} = H(V_j) \tag{27}$$

Our experimental results are shown in Table 1. The mutual information-based weighting method outperforms other methods in terms of accuracy. Compared with the constrained Pearson method, the MAE error in *All but One* protocol was reduced from 0.982 to 0.938 by a factor of 4.5%; the 0.5 accuracy was improved from 31.3% to 34.1% by a factor of 8.9%; while in predicting extreme votes (*Extremes*) the improvement is more impressive: the MAE was reduced by a factor of 7.1%, 0.5 accuracy improved by 28% and 1.0 accuracy improved by 26%. While entropy-based weighting only slightly improved the accuracy, the inverse user frequency method resulted in worse quality than the constrained Pearson method. In the other two protocols, *Given 10* and *Given 20*, we obtained similar results. The improvement of *Given 10* is not as significant as that of the other two, which indicates that consumers with limited available information are hard to predict. A serious problem is that the accuracy achieved in predicting extreme votes (*Extremes*) is still much worse than that achieved in predicting other votes. Further improvement is obviously needed. In Section 6.5, it will be shown that feature weighting combined with instance selection can further improve the accuracy of extreme vote (*Extremes*) prediction (the 1.0 accuracy is improved by 45.2%!).

Table 2. Performance of instance selection methods.

Method (Instance Selection)		All			Extremes			Time
		MAE	0.5 Accu.	1.0 Accu.	MAE	0.5 Accu.	1.0 Accu.	(ms)
Movie average		1.10	27.1%	52.5%	1.59	6.51%	24.5%	
All but One	Con. Pearson	0.982	31.3%	58.4%	1.40	10.0%	31.6%	48.2
	Modified IB2	0.959	33.5%	59.4%	1.35	13.2%	34.5%	31.6
	Rand. $r = 0.0625$	1.02	30.5%	58.1%	1.42	9.33%	31.3%	3.2
	Rand. $r = 0.125$	1.01	31.0%	58.2%	1.41	9.63%	31.5%	6.1
	Rand. $r = 0.25$	0.989	31.2%	58.5%	1.41	9.81%	32.0%	11.8
	Info. $r = 0.0625$	0.960	32.7%	59.5%	1.38	11.4%	34.5%	5.8
	Info. $r = 0.125$	0.959	32.4%	60.1%	1.36	11.7%	35.7%	8.2
	Info. $r = 0.25$	0.962	32.7%	59.9%	1.37	11.4%	35.5%	13.5
Given 10	Con. Pearson	1.02	30.2%	56.0%	1.46	9.10%	28.1%	30.4
	Modified IB2	1.01	31.5%	56.6%	1.42	11.5%	30.1%	21.6
	Rand. $r = 0.0625$	1.05	29.4%	56.0%	1.48	9.03%	27.0%	2.1
	Rand. $r = 0.125$	1.04	30.3%	56.6%	1.47	9.10%	27.9%	4.1
	Rand. $r = 0.25$	1.03	30.5%	56.2%	1.47	9.12%	28.2%	7.9
	Info. $r = 0.0625$	1.02	30.2%	56.8%	1.45	10.0%	29.2%	3.6
	Info. $r = 0.125$	1.01	30.4%	56.9%	1.43	10.6%	31.0%	6.3
	Info. $r = 0.25$	1.01	31.3%	57.5%	1.43	10.6%	30.7%	8.2
Given 20	Con. Pearson	1.00	30.8%	57.7%	1.43	9.82%	30.8%	35.6
	Modified IB2	0.988	32.0%	58.6%	1.38	13.3%	33.5%	23.7
	Rand. $r = 0.0625$	1.04	30.5%	56.8%	1.46	9.42%	29.5%	2.4
	Rand. $r = 0.125$	1.02	30.9%	57.4%	1.46	10.7%	29.9%	5.0
	Rand. $r = 0.25$	1.01	30.7%	57.5%	1.45	9.60%	30.7%	9.3
	Info. $r = 0.0625$	0.987	31.5%	58.6%	1.41	10.8%	32.8%	4.4
	Info. $r = 0.125$	0.985	31.9%	58.9%	1.40	11.4%	33.9%	7.2
	Info. $r = 0.25$	0.987	32.2%	58.9%	1.42	11.0%	33.5%	10.8

6.4. Empirical Analysis of Instance Selection

We investigated three instance selection algorithms including random sampling, modified IB2 algorithm and the proposed information-theoretic instance selection algorithm. The first algorithm randomly samples consumers according to a selection rate r from the entire consumer data set. The prediction is generated by applying the constrained Pearson algorithm to the selected data set. For every selection rate the random sampling was repeated 10 times and the results averaged. IB2 is a well-known instance selection method (Aha et al., 1991), which is used to reduce the storage of nearest neighbor classifiers. The algorithm selects incorrectly classified instances in order to put more strength on border instances and hard instances. We modified it to consumer selection in instance-based collaborative filtering, which is not classification but regression. For a target movie i, modified IB2 randomly selects 150 consumers $\mathscr{S}_i$ from $\mathscr{T}_i$, and incrementally processes the remaining consumers in $\mathscr{T}_i$ following a simple rule: if the absolute prediction error of $v_{u,i}$ is greater than 0.5 by using the current instance set $\mathscr{S}_i$, then consumer u is added into $\mathscr{S}_i$. In the prediction phase constrained Pearson method is then performed on the selected consumer set $\mathscr{T}_i$.

The experimental results are shown in Table 2. We evaluate the algorithms in terms of accuracy and efficiency in the prediction phase. As pointed out in Section 4, the speed-up reflects the reduction of the instance set because the run time is linear to the size of instance set. In summary, random sampling approaches lead to a dramatic increase of efficiency, but at the expense of accuracy. Modified

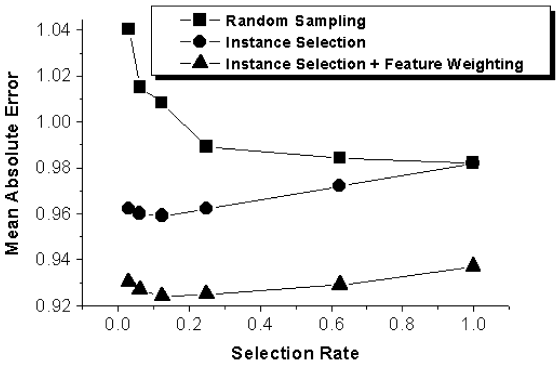

Fig. 3. MAE performance using different selection rates (*All but One*).

IB2 slightly speeds up the run time by a factor of roughly 3/2. This is because the 0.5 accuracy is always about 30% and hence modified IB2 removes 1/3 of instances from the original instance set. Moreover, it results in a significant improvement of accuracy. Our analysis shows that modified IB2 maintains nearly all the instances with extreme votes (*Extremes*) while removing relatively more instances with vote value 3 or 4. This may reduce the bias caused by instances with vote value 3 or 4 who are the consumers without a clear preference. Finally, the proposed instance selection based on an information-theoretic relevance measure achieved the best overall performance in terms of accuracy and efficiency. Its accuracy is comparable to modified IB2 while the efficiency is greatly improved. For example, the overall MAE was reduced from 0.982 to 0.960 by a factor of 2.2%, while the prediction time was reduced from 48.2 to 5.8 by a factor of 8.3 in the case of *All but One* and with $r = 0.0625$. The 1.0 accuracy of predicting extreme votes (*Extremes*) was also improved from 31.6% to 35.5% by a factor of 12.3%. The selection rate of 0.0625 did not result in a speed-up factor of 16. This is because the minimal size of instance set is chosen to avoid the over-reduction of instance set, as described in Section 5.2.

It is of interest to study the accuracy of the proposed instance selection method using different selection rates. As shown in Fig. 3, the MAE continually decreases as the selection rate is decreased until sampling rate reaches 0.125. This result shows that over-reduction of the instance set will degrade the quality. Therefore, an optimal selection rate should be determined. This problem can be resolved through experiment (e.g. cross-validation). Here we attempt to give an automatic solution. Figure 4 shows the case when the target movie is *Dances with Wolves*. In Fig. 4(a) the rationality strengths of consumers are sorted in ascending order (a total of 6474 of 8000 consumers rated this movie). The quality of MAE using different r is given in Fig. 4(b). The optimal selection rate shown in Fig. 4(b) corresponds to the marked cut point in Fig. 4(a) where the consumers with higher strength are selected. It can be seen in Fig. 4(a) that rationality strength begins to dramatically increase at the right side of the cut point. A similar phenomenon is seen when other movies are analyzed, which inspired us to treat the instance selection problem as a classification problem: an instance whose rationality strength is greater than a threshold is classified as a relevant instance and is otherwise classified as an irrelevant instance. To find the cut point, we performed a simple 2-class expectation maximization (EM) clustering (Witten

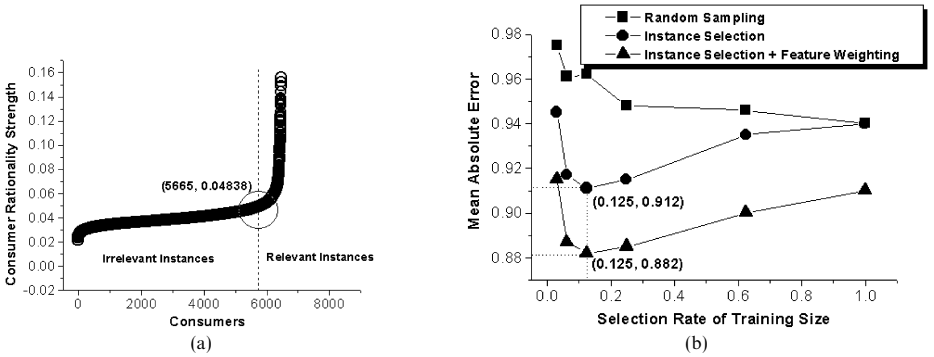

Fig. 4. (a) Consumer rationality strengths sorted in ascending order. (b) Prediction MAE at different selection rates (the target movie is *Dances with Wolves*).

and Frank, 1999) in a one-dimensional space-rationality strength. The algorithm attempts to maximize the likelihood of the clustering model under the assumption that each cluster follows a Gaussian distribution. At first the instances are sorted in ascending order by strength, and the mid point is selected for cutting. So half of the consumers are classified as irrelevant ones, and the others are relevant. Based on the above division the mean and standard deviation of each cluster are calculated. Then an iterative process begins:

1. *Expectation step.* Based on the calculated means and deviations each instance is reclassified into one of the two clusters according to its a posteriori probability of membership.
2. *Maximization step.* The mean and standard deviation of each cluster are calculated based on the classification in step 1.

The iteration continues until the clustering remains unchanged. The algorithm is fast since it is in a one-dimensional space and the convergence is reached in very few steps. Due to space limitations we will skip the details of the EM algorithm, which can be found in Witten and Frank (1999). Another advantage of this automatic determination of selection rate is that the resulting selection rate is optimal for the given target (product). Therefore, an optimal selection rate is determined for each product instead of one selection rate being used for all products. This improvement is confirmed by our experimental results: automatic instance selection performed better than the use of single selection rate $r = 0.125$ for every product. Detail results are given in the next subsection (Table 3). Figures 3 and 4(b) also show that feature weighting further improves the accuracy of collaborative filtering after the instance selection is performed, indicating that the two approaches can be combined together in order to get optimal performance in terms of accuracy and efficiency.

6.5. Combining Feature Weighting and Instance Selection

Finally, as proposed in Section 5.2, we combined the information-theoretic feature weighting and instance selection to reach a maximal level of performance. The empirical results (Table 3) show that the advantages of the two approaches are additive: the prediction accuracy was significantly better than the traditional

Table 3. The performance of different approaches combining feature weighting and instance selection.

Method (Instance Selection)		All			Extremes			Time
		MAE	0.5 Accu.	1.0 Accu.	MAE	0.5 Accu.	1.0 Accu.	(ms)
Movie average		1.10	27.1%	52.5%	1.59	6.51%	24.5%	
All but One	Con. Pearson	0.982	31.3%	58.4%	1.40	10.0%	31.6%	48.2
	Info. $r = 0.0625$	0.927	33.6%	62.3%	1.26	13.4%	44.4%	6.0
	Info. $r = 0.125$	0.924	34.2%	63.2%	1.27	13.5%	44.7%	8.5
	Info. Auto.	0.920	34.0%	63.6%	1.20	14.7%	45.2%	8.1
Given 10	Con. Pearson	1.02	30.2%	56.0%	1.46	9.10%	28.1%	30.4
	Info. $r = 0.0625$	1.00	30.3%	58.1%	1.40	11.0%	34.1%	3.6
	Info. $r = 0.125$	1.00	31.1%	58.4%	1.40	11.6%	34.2%	6.5
	Info. Auto.	1.00	31.5%	58.1%	1.40	11.7%	34.5%	5.2
Given 20	Con. Pearson	1.00	30.8%	57.7%	1.43	9.82%	30.8%	35.6
	Info. $r = 0.0625$	0.970	32.5%	60.0%	1.34	11.9%	39.5%	4.6
	Info. $r = 0.125$	0.967	33.3%	60.2%	1.33	12.8%	39.4%	7.2
	Info. Auto.	0.968	32.5%	60.1%	1.32	13.1%	40.2%	6.0

constrained Pearson method, while the efficiency was also greatly improved. The results also indicate that the combined approach even outperformed the feature weighting approach in terms of accuracy. For example, in the *All but One* protocol, feature weighting combined with instance selection ($r = 0.0625$) reduced the overall MAE from 0.982 to 0.927 by a factor of 5.6%. The quality in predicting extreme votes (*Extremes*) is still more impressive: MAE was reduced from 1.40 to 1.26 by a factor of 10%, 0.5 accuracy improved from 10% to 13.4% by a factor of 34%, and 1.0 accuracy improved from 31.6% to 44.4% by a factor of 40.5%. The run time sped up by a factor of 8.0. Feature weighting caused a small increase in computational cost, but it is negligible compared to the speed-up caused by instance selection. Furthermore the combined approach using automatic instance selection performs very well in terms of accuracy and efficiency. It is even better than the best case hitherto when selection rate of 0.125 is used for all products. This result shows that EM clustering can be used to automatically distinguish relevant instances from irrelevant ones.

7. Conclusion

In this paper, feature relevance and instance relevance for collaborative filtering are studied in a unified information-theoretic framework. Our work shows that the two perspectives are intimately related: from a probabilistic relevance analysis, mutual information was proposed to measure the relevance of features with respect to the target product; the Bayes learning then inspired our definition of instance rationality. After some simplification the general rationality and its strength, both in form of mutual information, were proposed to serve as a measure of instance relevance. It was argued that the combination of feature weighting and instance selection based on relevance analysis can improve the collaborative filtering in terms of accuracy and efficiency. The empirical results have shown that mutual information-based feature selection achieves a good accuracy. Instance selection not only dramatically speed up the prediction but also improved the accuracy. Further experiments showed that the combination of feature weighting

and instance selection reaches an optimal performance. For instance, the accuracy (1.0 accuracy) was improved by a factor of about 40%, while the run time was reduced by a factor 8 (in the *All but One* protocol).

Our experimental results demonstrate that feature weighting and instance selection can be successfully applied to collaborative filtering-based recommender systems. Relevance is an important topic in machine-learning and data-mining research. We believe that more work needs to be done in order to reveal the role of feature relevance and instance relevance in mining large databases. The relationship between feature relevance and instance relevance also needs further study.

Acknowledgements. The authors thank the anonymous reviewers for their constructive and helpful suggestions to improve this paper.

References

Aha DW, Kibler D, Albert MK (1991) Instance-based learning algorithms. Machine Learning 6:37–66

Anand SS, Patterson DW, Hughes JG (1998) Knowledge intensive exception spaces. In Proceedings of the 15th national conference on artificial intelligence and 10th innovative applications of artificial intelligence conference, AAAI/IAAI 98, July, pp 574–579

Billsus D, Pazzani MJ (1998) Learning collaborative information filters. In Proceedings of the 15th international conference on machine learning, pp 46–54

Blum AL, Langley P (1997) Selection of relevant features and examples in machine learning. Artificial Intelligence, pp 245–271

Breese JS, Heckerman D, Kadie C (1998) Empirical analysis of predictive algorithms for collaborative filtering. In Proceedings of the 14th conference on uncertainty in artificial intelligence (UAI-1998), pp 43–52

Cost S, Salzberg S (1993) A weighted nearest neighbor algorithm for learning with symbolic features. Machine Learning 10:57–78

Cover TM, Hart PE (1998) Nearest neighbor pattern classification. Institute of Electrical and Electronics Engineers Transactions on Information 13(1):21–27

Cestnik B (1990) A crucial task in machine learning. In Proceedings of the ninth European conference on artificial intelligence, pp 147–149

Cussens J (1993) Bayes and pseudo-Bayes estimates of conditional probability and their reliability. In European conference on machine learning. Lecture Notes in Artificial Intelligence 667, Springer, pp 136–152

Deco G, Obradovic D (1996) An information-theoretic approach to neural computing. Spinger, New York, pp 10–11

Domingos P (1996) Unifying instance-based and rule-based induction. Machine Learning 24:141–168

Domingos P, Pazzani ML (1997) Beyond independence: conditions for the optimality of the simple Bayesian classifier. In Proceedings of the 13th international conference on machine learning (ICML)

Ester M, Kriegel HP, Xu X (1995) A database interface for clustering in large spatial databases. In Proceedings of the first international conference on knowledge discovery and data mining (KDD95), Montreal, Canada, pp 94–99

Hart PE (1968) The condensed nearest neighbor rule. IEEE Transactions on Information Theory 14:515–516

Herlocker JL, Konstan JA, Borchers A, Riedl J (1999) An algorithmic framework for performing collaborative filtering. In Proceedings of the conference on research and development in information retrieval

Hill W, Stead L, Rosenstein M, Furnas G (1995) Recommending and evaluating choices in a virtual community of use. In Proceedings of ACM CHI95 conference

Kolodner J (1993) Case-based reasoning. Morgan Kaufmann, San Mateo, CA

Liu H, Motoda H (1999) Instance selection and construction for data mining. Kluwer, Dordrecht

Mitchell T (1997) Machine learning. McGraw-Hill, New York, pp 65, 177, 179

Patterson D, Galushka M, Rooney N, Anand SS (2002) Towards dynamic maintenance of retrieval knowledge in CBR. In Proceedings of the 15th international FLAIRS conference, FL

Pradhan S, Wu X (1999) Instance selection in data mining. Technical report, Department of Computer Science, University of Colorado at Boulder

Resnick P, Iacovou N, Sushak M, Bergstrom P, Riedl J (1994) GroupLens: an open architecture for collaborative filtering of netnews. In Proceedings of the 1994 computer supported collaborative work conference
Saloton G, McGill M (1983) Introduction to modern information retrieval. McGraw-Hill, New York
Saltzberg S (1990) Learning with nested generalised examplars. Kluwer, Norwell, MA
Saltzberg S (1991) A nearest hyperrectangle learning method. Machine Learning 6:277–309
Sarwar BM, Karypis G, Konstan JA, Riedl J (2000) Analysis of recommender algorithms for e-commerce. In proceedings of ACM E-commerce 2000 conference
Shannon CE (1948) A mathematical theory of communication. Bell System Technology journal 27
Shardanand U, Maes P (1995) Social information filtering algorithms for automating 'Word of mouth'. In Proceedings of ACM CHI95 conference
Smyth B, McKenna E (1999) Footprint-based retrieval. In Proceedings of third international conference on case-based reasoning, Munich, Germany, pp 343–357
Stanfill C, Waltz D (1986) Toward memory-based reasoning. Communications of ACM 29:213–1228
Wettschereck D, Dietterich TG (1995) An experimental comparison of nearest-neighbor and nearest-hyperectangle algorithms. Machine Learning 19(1):5–28
Wettschereck D, Aha DW, Mohri T (1997) A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. Artificial Intelligence Review 11:273–314
Wilson DR, Martinez TR (2000) Reduction techniques for instance-based learning algorithms. Machine Learning 38(3):257–286
Witten L, Frank E (1999) Data mining: practical machine learning tools and techniques with Java implementations. Morgan Kaufmann, p 221
Zhang J (1992) Selecting typical instances in instance-based learning. In Proceedings of the ninth San Mateo, CA, international conference on machine learning, pp 470–479

Appendix

A.1. Proof of Theorem 3.1

Proof. Since $P(V_i)$ and $P(V_j)$ are fixed, inequation (8) can be rewritten as:

$$\frac{[p(|v_{A,i} - v_{B,i}| < e \mid |v_{A,j} - v_{B,j}| < e)]}{d[H(V_i) + H(V_j) - H(V_i, V_j)]} = \frac{d[p(|v_{A,i} - v_{B,i}| < e \mid |v_{A,j} - v_{B,j}| < e)]}{d[-H(V_i, V_j)]} > 0 \tag{28}$$

Next, we have

$$\begin{aligned} -H(V_i, V_j) &= \sum_{k=0}^{N} \sum_{l=0}^{N} p(v_i = k, v_j = l) \log_2 p(v_i = k, v_j = l) \\ &= \sum_{k=0}^{N} \sum_{l=0}^{N} p_{k,l} \log_2 p_{k,l} \end{aligned} \tag{29}$$

where $p_{k,l} = p(v_i = k, v_j = l)$. Since consumer A and consumer B are drawn independently, we have

$$p(|v_{A,i} - v_{B,i}| < e \mid |v_{A,j} - v_{B,j}| < e) \tag{30}$$

$$= \frac{\sum_{k=0}^{N} \sum_{l=0}^{N} p(v_{A,i} = v_{B,i} = k, v_{A,j} = v_{B,j} = l)}{\sum_{l=0}^{N} p(v_{A,j} = v_{B,j} = l)}$$

$$= \frac{\sum_{k=0}^{N} \sum_{l=0}^{N} p_{k,l}^2}{\sum_{l=0}^{N} p(v_j = l)^2} = \frac{1}{K} \sum_{k=0}^{N} \sum_{l=0}^{N} p_{k,l}^2 \tag{31}$$

where $K = 1/\sum_{l=0}^{N} p(v_j = l)^2$. Consider the conditions that $P(V_i)$ and $P(V_j)$ are fixed and $\sum_{k=0}^{N}\sum_{l=0}^{N} p_{k,l} = 1$, both first terms of equations (29) and (30) can be seen as functions with respect to $N \times N - 2N + 1$ independent variables, $p_{k,l}, k, l \neq N$. Then we perform partial differentiations:

$$\frac{\partial[-H(V_i, V_j)]}{\partial(p_{k,l})} = \log_2 p_{k,l} - \log_2 p_{k,N} \tag{32}$$

$$\frac{\partial[p(|v_{A,i} - v_{B,i}| < e \mid |v_{A,j} - v_{B,j}| < e)]}{\partial(p_{k,l})} = 2K(p_{k,l} - p_{k,N}) \tag{33}$$

where $k, l = 0, 1, \ldots, N-1$. In the above two equations, $p_{k,l} - p_{k,N}$ and $\log_2 p_{k,l} - \log_2 p_{k,N}$ always have the same sign, therefore the inequations (28) and (8) hold. □

A.2. Proof of Theorem 4.1

Proof. According to Definition 4.2, we have :

$$\begin{aligned} R^*_{u,i} &= I(V_i; V_{\mathscr{F}_{u,i}}) \\ &= H(V_i) - H(V_i|V_{\mathscr{F}_{u,i}}) \\ &= H(V_{\mathscr{F}_{u,i}}) - H(V_{\mathscr{F}_{u,i}}|V_i) \end{aligned} \tag{34}$$

Since each product $j \in \mathscr{F}_{u,i}$ is independent of the others whether given V_i or not, then

$$\begin{aligned} R^*_{u,i} &= \sum_{j \in \mathscr{F}_{u,i}} H(V_j) - \sum_{j \in \mathscr{F}_{u,i}} H(V_j|V_i) \\ &= \sum_{j \in \mathscr{F}_{u,i}} I(V_i; V_j) \end{aligned} \tag{35}$$

Therefore the conclusion equation (35) holds. □

Author Biographies

Kai Yu is a PhD student, supported by a Siemens Scholarship, at the Institute for Computer Science, University of Munich, Germany. Currently he is a Guest Scientist at Corporate Technology, Siemens AG. His research interests include speech signal processing, machine learning, data mining and their applications in E-commerce. He received a BS and an MS in 1998 and 2000, respectively, from Nanjing University, China.

Xiaowei Xu is an Associate Professor at the Information Science Department, University of Arkansas at Little Rock. He has been working as a Senior Research Scientist at Corporate Technology, Siemens AG. His research interests include data mining, machine learning, database systems and information retrieval. With his students and colleagues, he has developed several systems, including spatial data mining, web mining for adaptive web interface design, scalable recommender systems for E–commerce, and a database management system supporting protein–protein docking prediction. He received his BS from Nankai University, China, his MS from Shenyang Institute for Computing Technology, Chinese Academy of Sciences, and his PhD from the University of Munich, Germany.

Martin Ester received a PhD in Computer Science from the Swiss Federal Institute of Technology (ETH Zurich) in 1990. He has been working for Swissair developing expert systems before he joined the University of Munich in 1993 as an Assistant Professor in the areas of databases and data mining. He has co-authored the first German text book on Knowledge Discovery in Databases. Since November 2001, he has been an Associate Professor at the School of Computing Science of Simon Fraser University, where he co-directs the database and data mining lab. His current research interests include hypertext mining, mining in biological databases and the integration of data mining with knowledge management.

Hans-Peter Kriegel is a Full Professor of Database Systems in the Institute for Computer Science at the University of Munich. He is considered one of the internationally leading researchers in the areas knowledge discovery, data mining and similarity search in large databases. His research interests are in spatial and multimedia database systems, particularly in query processing, performance issues, similarity search, high-dimensional indexing, and in parallel systems. Data exploration using visualization led him to the area of knowledge discovery and data mining. Kriegel received his MS and PhD in 1973 and 1976, respectively, from the University of Karlsruhe, Germany. Hans-Peter Kriegel has been chairman and program committee member in many international database conferences. He has published over 200 refereed conference and journal papers. In 1997 Hans-Peter Kriegel received the internationally prestigious 'SIGMOD Best Paper Award 1997' for the publication and prototype implementation 'Fast Parallel Similarity Search in Multimedia Databases' together with four members of his research team.

Correspondence and offprint requests to: Xiaowei Xu, Information Science Department, University of Arkansas at Little Rock, 2801 South University, Little Rock, AR 72204-1099, USA. Email: xwxu@ualr.edu