Association for Information Systems
AIS Electronic Library (AISeL)

PACIS 2003 Proceedings

Pacific Asia Conference on Information Systems (PACIS)

December 2003

A Cluster-indexing CBR Model for Collaborative Filtering Recommendation

Tae Roh
Korea Advanced Institute of Science and Technology

Kyong Oh
Hansung University

Ingoo Han
Korea Advanced Institute of Science and Technology

A Cluster-indexing CBR Model for Collaborative Filtering Recommendation

Tae Hyup Roh[a], Kyong Joo Oh [b] and Ingoo Han [c]

[a] Graduate School of Management Korea Advanced Institute of Science and Technology,
207-43 Cheongryangri-Dong, Dongdaemun-Gu, Seoul, Korea, 130-012.
rohth@kgsm.kaist.ac.kr

[b] Department of Business Administration, Hansung University
389, 3-Ga, Samsun-Dong, Sungbuk-Gu, Seoul Korea, 136-792
johanoh@hansung.ac.kr

[c a] Graduate School of Management Korea Advanced Institute of Science and Technology,
207-43 Cheongryangri-Dong, Dongdaemun-Gu, Seoul, Korea, 130-012.
ighan@kgsm.kaist.ac.kr

Abstract

Collaborative filtering (CF) recommendation is a knowledge sharing technology for distribution of opinions and facilitating contacts in network society between people with similar interests. The main concerns of the CF algorithm are about prediction accuracy, speed of response time, problem of data sparsity, and scalability. In general the efforts of improving prediction algorithms and lessening response time are decoupled. We propose a three-step CF recommendation model which is composed of profiling, inferring and a final prediction step, while considering prediction accuracy and computing speed simultaneously. This model combines a CF algorithm with two machine learning processes, SOM (Self-Organizing Map) and CBR (Case Based Reasoning), by changing an unsupervised clustering problem into a supervised user preference reasoning problem, which is a novel approach for the CF recommendation field. This paper demonstrates the utility of the CF recommendation based on SOM cluster-indexing CBR, with validation against control algorithms through an open dataset of user preference.

Keywords

Collaborative filtering, recommendation system, self-organizing map, case-based reasoning.

1. Introduction

The advent of the network world induced by the rapid development of the Internet and the accompanying adoption of the Web has promoted the chances to create greater business opportunities and to reach customers more easily. This 24 7 on-line accessibility has resulted in the enlargement of choices, but customers are now faced with information overload. Their own arduous efforts are required to retrieve information that matches their preferences (Good, Schafer, Konstan, Borchers, Sarwar, Herlocker, & Riedl, 1999; Goldberg, Roeder, Gupta, & Perkins, 2001; Cho, Kim, & Kim, 2002). What is needed is an automated and sophisticated decision support system to suggest personalized information in a brief form without going through an annoying process.

Collaborative Filtering (CF) recommendation is a knowledge sharing technology for distribution of opinions and facilitating contacts in network society between people with similar interests. The CF recommendation is the process of multiple users sharing information on the preferences and actions of an affinity group tracked by a system which, then, tries to make useful recommendations to individual users based on the patterns it predicts (Kumar, Raghavan, Rajagopalan, & Tomkins, 1998; Aggarwal, Wolf, Wu, & Yu, 1999; Herlocker, Konstan, & Riedl, 2000). CF recommendation also provides a complementary tool for information retrieval systems that facilitates users' navigation in a meaningful and personalized way. Most content retrieval methodologies use some type of a similarity score to match a query describing the content with key words, the individual titles or items, and then present the user with a ranked list of suggestions. However, conventional CF does not use any actual content (e.g. words, description, URL, etc.) of the items. It is rather based on preference ratings information to match users with similar interests together and to predict a user's rating for an unseen item by examining his/her cyber community's rating for that item. The CF recommendation systems are built on the assumption that a good way to find interesting content is to find other people who have similar interests and then recommend items that those similar users like (Breese, Heckerman, & Kadie, 1998).

Our focus is on technical system development research**,** especially, the design and analysis of an algorithm for CF recommendation. As the number of users and items increases and the contents of each user's preference to the items changes, typical CF recommendation needs exponentially growing computation time for finding an affinity group and predicting each user's unknown preferences (Claypool, Gokhale, Miranda, Murnikov, Netes, & Sartin, 1999; Sarwar, Karypis, Konstan, & Riedl, 2000; Cho, Kim, & Kim, 2002). We find the potential for simultaneous improving prediction accuracy and efficiency by separating on and off-line steps using recent clustering and reasoning machine learning techniques. This study presents a three-step CF model, SOM cluster-indexing CBR CF predictor, which is composed of profiling, inferring and a prediction step. The profiling step is operated off-line while the inferring and prediction steps are run in the on-line system. The SOM network has been studied as one of the most popular unsupervised neural network models for clustering and visualization in a number of real-world problems (Kohonen, Hynninen, Kangas, & Laaksonen, 1996). The CBR is well known as it benefits from the case-specific knowledge of past problems to find solutions to the new problems (Kim & Han, 2001). These two outperforming machine learning methods can be combined for CF and increase the accuracy and efficiency in the recommendation process.

The rest of this paper is organized as follows. Section 2 provides a brief overview of CF models for recommendation, several recommendation techniques and issues with an emphasis on algorithmic features shown by previous research. Details of the proposed SOM Cluster-indexing CBR CF model are provided in Section 3. Section 4 describes the data set summary, evaluation metrics and experimental design of suggested model and comparative methods. Experiments are run on an open dataset associated with the MovieLens preference rating dataset, 6 experimental protocols, and 2 evaluation metrics for the various algorithms. Results are shown in Section 5 and the overall summary and conclusion are presented in the final section, 6.

2. Background

2.1. Collaborative filtering recommendation

CF recommendation works by collecting human judgments (known as ratings) for items in a given domain and matching together people who share the same information needs or the same taste. Users of a CF recommendation system share their analytical judgments and opinions regarding each item that they consume so that other users of the system can better decide which items to consume. CF systems can support predictions or recommendations for large-scale communities of users, and due to the large number of users, anonymity can be provided. In return, the CF system provides useful personalized recommendations for interesting items (Herlocker, Konstan, Borchers, & Riedl, 1999).

The fundamental function of CF is to predict the preferences of one user, referred to as the "*active user*," based on the preferences of a group of users. The problem space can be formulated as a matrix of users versus items, with each cell representing a user's rating on a specific item. Let I be the whole set of items, $I_h \quad I$ be the subset that have been rated by the "*active user*", U_a, and $I_r = (I \quad I_h{}^C)$ be the subset that have not been rated by the U_a. CF systems estimate U_a's preferences for items in I_r based on the overlap between his/her preference ratings for items in I_h and those of the other users.

One key advantage of CF is that it does not consider the content of the items being recommended, but human determine the relevance, quality, and interest of an items in the information stream. As a result, filtering can be performed on items that are hard to analyse with computers, such as movies, ideas, feelings, people and so on. Rather than mapping users to items through "content attributes" or "demographics," CF treats each item and user individually. At the same time, CF's dependence on human ratings can be a drawback. For a CF system to work well, several users must evaluate each item; even then, new items cannot be recommended until some users have taken the time to evaluate them. These limitations, often referred to as "*data sparsity*" and "*cold start problem*", cause trouble for users seeking obscure items (since nobody may have rated them) or advice on new items (since nobody has had a chance to evaluate them) (Good et al., 1999; Herlocker et al., 1999).

CF related research starts from the Tapestry system, out of Xerox (Goldberg, Nichols, Oki, & Terry, 1992) which coined the term "collaborative filtering" in the context of a system for filtering email using binary category flags. GroupLens is a pioneering and ongoing effort in CF (Resnick, Iacovou, Sushak, Bergstrom, & Riedl, 1994; Konstan, Miller, Maltz, Herlocker, Gordon, & Riedl, 1997; Good et al., 1999; Herlocker et al., 1999; Schafer, Konstan, & Riedl, 2001). The GroupLens team initially implemented a neighbourhood-based CF system for rating Usenet articles. Several similar systems were developed around the same time as the GroupLens Usenet system, including the Ringo music recommender which used a number of measures of distance between users, including Pearson correlation, constrained Pearson correlation, vector cosine (Shardanand & Maes, 1995), and the Bellcore Video Recommender (Hill, Stead, Rosenstein, & Furnas, 1995). These three research systems used what have come to be called neighbourhood-based prediction algorithms. Due to their speed, flexibility, and understandability, neighbourhood-based prediction algorithms are currently one of the most effective ways to compute predictions in CF. Breese et al. (1998) identify two major classes of CF prediction algorithms; memory-based CF and model-based CF. Memory-based algorithms operate over the entire user database to make predictions. The most common memory-based models are based on the notion of nearest neighbours, using a variety of distance measures. Model-based systems are based on a compact model inferred from the data. In this framework, our SOM cluster-indexing CBR CF predictor model would be considered model-based CF.

More recently, a number of machine learning techniques and hybrid filtering techniques have been challenged. Hybrid filtering models combine recommendations from multiple sources which include the content of the item or page, the ratings of users, content-based filtering, and demographic information and so on. Balabanovic and Shoham (1997) apply 'Selection agent,' which decides the recommendation algorithm between content-based filtering and CF. Pazzani (1999) shows the hybrid approach for recommendation that uses more of the available information and consequently has more precise recommendations. The strengths of the different approaches can be complementary. Basu, Hirsh, and Cohen (1998) present an inductive learning approach to recommendation that is able to use both ratings information and other forms of information about each item in predicting user preferences. Aggarwal et al., (1999) apply graph theory and Delgado and Ishii (1999) suggest a weighted-majority rating approach. Pennock, Horvitz, Lawrence, and Giles (2000) suggest a hybrid memory and model based approach for personality diagnosis that computes the preference probability with the same personality grouping.

In this paper, we introduce a computational machine learning model to CF with empirical tests on non-binary explicit numeric data. Since we adapt the dense user-item matrix using the reference data set induced by SOM clusters' centroid value of each item, the correlation matrix is directly computed and then the active user's preference to item is predicted.

2.2. Self-organizing map

The SOM network, known as competitive learning or self-organization, has been shown as one of the most popular unsupervised competitive neural network learning models, for clustering and visualization in a number of real-world problems (Kohonen et al., 1996; Sheng-Tun, 2002). SOM has been adapted as an analytical tool in various marketing domains including database marketing (Ha & Park, 1998), segmentation of on-line markets (Vellido, Lisboa, & Meehan, 1999) and automatic labelling of customer clusters (Yuan & Chang, 2001).

Besides the SOM network is capable of mapping high-dimensional similar input data into clusters close to each other. It has two-layer, fully connected networks with a weight matrix. Sometimes, SOM called "topology-preserving maps," assumes a topological structure among the cluster units. A topological map is simply a mapping that preserves neighbourhood relations and performs a topology-preserving projection from the data space onto a regular two-dimensional grid. This property is observed in the brain, but is not found in other Artificial Neural Networks (ANNs), such as back propagation neural networks. The resulting maps provide users an intuitive and familiar way of correlating and illustrating input data sets. The SOM network has visualization capabilities in providing informative pictures of the data space and can compare data vectors or whole data sets with each other. Furthermore, SOM can be used for clustering, classification, and modelling. The versatile properties of SOM make it a valuable tool in data mining. Relating to the clustering capability of SOM, Mangiameli, Chen, and West (1996) demonstrate that it is a better clustering algorithm than hierarchical clustering with overlapped dispersion, irrelevant variables, outliers or different sized populations. Their sensitivity analysis of SOM proves its insensitivity to learning rates which vary in the self-organizing process.

A conventional feature network of the classical SOM-algorithm is given in Figure 1 and pseudo code description is as follows (Günter & Bunke, 2002).

In spite of several excellent applications, SOM has some limitations that hinder its performance. The typical limitations and the settlements are due to the vulnerability of convergence along a number of cluster and weight initialization, network size, and stopping rule conditions (Kim & Han, 2001). To determine the number of cluster, we adapt the visualization techniques by use of Principle Component Analysis (PCA). PCA was first introduced in 1901 by Karl Pearson and Hotelling generalized it to random variables in 1933. The idea is to keep only the "principle" eigenvectors (components). The number of eigenvectors to retain depends on the variances (eigenvalues) but is typically small. If v eigenvectors are retained, data is projected along the first n principal eigenvectors. In this study, PCA facilitates dimensionality reduction for offline clustering of user and rapid online cluster assignment, and also users are projected onto the "eigen-plane" in 2 or 3 dimension scatter plot for visualization.

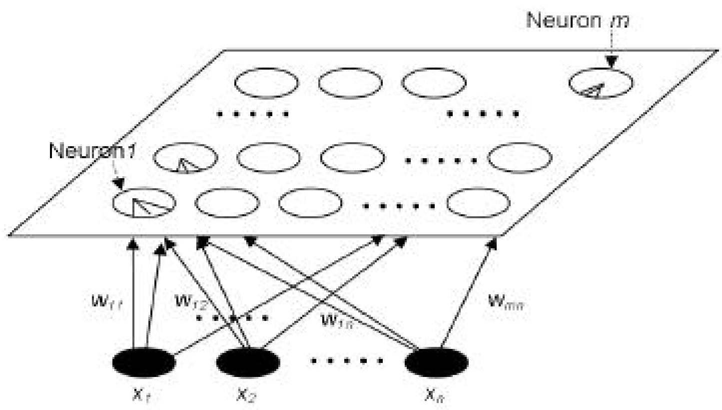

Figure 1. Self-Organizing Map – conventional feature mapping network. The inputs are fully connected to each output neuron. However, only a few connections are shown.

2.3. Case based reasoning

CBR is a methodology for building an analogy process, one way of human reasoning, which is the inference that a certain resemblance implies further similarity. CBR allows for combining a cognitive model describing how people use and reason from past experience with a technology for finding and presenting such experience (Choy, Lee, & Lo, 2002; Chiu, 2002). Its main advantages over other techniques are as follows: In the CBR system, most knowledge is acquired in the case base and so it reduces the knowledge acquisition effort. That is, it makes use of existing case data, e.g. in database, so it requires less general knowledge which is very difficult to get. Second, it requires less maintenance effort. Since rule bases or models should consider many dependencies between rules, often difficult to understand for non experts, and effects of changes of the rule base are hard to predict, rule bases or models are difficult to maintain. However, case bases are easier to maintain, because cases are independent from each other, domain experts and novices understand cases quite easily and maintenance of the CBR system (partially) can be done by adding/deleting cases.

CBR algorithms have been used for marketing decision making processes. Hui, Fong, and Jha (2001) present the hybrid CBR–ANN approach that integrates ANN with the CBR cycle to extract knowledge from service records for the web customer service. Choy et al. (2002) apply CBR to integrate customer relationship management (CRM) and supplier relationship management (SRM) for facilitating supply chain management of supplier selection. Chiu (2002) suggests a case-based customer classification approach for direct marketing, which combines Genetic Algorithm and the CBR process.

The traditional process involved in CBR can be represented by a schematic cycle as shown in Figure 2. Aamodt and Plaza (1994) and Bradley (1994) describe CBR as a cyclical process.

- Representation: a description of the current problem is input into the system.
- Retrieval: the system retrieves the closest-matching cases stored in a case base.
- Reuse: the system uses the current problem and closest-matching cases to generate a solution to the current problem.
- Revision: the solution is revised through feedback from the user or the environment.
- Retainment: if appropriate, the validated solution is added to the case base for use in future problem-solving.

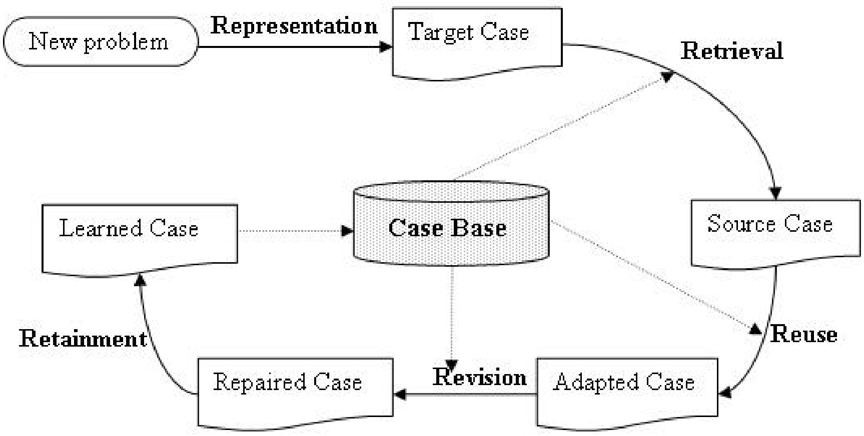

Figure 2. CBR process as a schematic cycle comprising the five "Re"s.

The SOM cluster-indexing CBR CF recommendation model can shorten real time computational complexity and provide additional generalized knowledge derived from the users' explicit preference patterns through the CBR process. Generalized knowledge can be acquired by the centroid values of clusters obtained using clustering techniques, which are added to the case-base as representative cases and then used as a case indexing scheme in order for retrieving more relevant cases. The cluster-indexing approach assumes that there are some different subgroups (clusters) in each rated group. The centroid values of clusters are new artificial cases that extract the information from the whole case-base and represent each clustered case.

3. SOM cluster-indexing CBR CF recommendation

This study utilizes the outperforming SOM as a clustering tool and the strength of CBR as assistance to index and retrieve like-minded users. The point of this study is to support the usage of: the SOM to find clusters of users; and, CBR for indexing and retrieval in the CF recommendation process. The centroid values of clusters are the values of weight vectors that are the interim results from SOM learning processes, and these are standardized due to difference of rating scale. The standardized centroid values have the same representation scheme as raw users' rating value in spite of being learned artificial cases. These values represent clustered users of the entire user-base, thus they are used as an indexing tool for each user. These standardized centroid values are consistent if learning is implemented again with the same parameters, although the addition of new users modify the standardized centroid values just a little.

The definite process of the cluster-indexing method is composed of three steps: profiling, inferring and prediction steps. In the profiling step, PCA and preliminary SOM testing is performed to fit the stable cluster condition. Clusters are derived from the dense subset of user-item rating DB, and all the training users are indexed by the SOM process in accordance with the similarity to the centroid values of each cluster. In the inferring step, CBR, compares an active user with the centroid values of each cluster. The most similar cluster is inferred from reference users indexed within the selected cluster. After the inferring step, preference prediction is performed with correlation based CF between an active user and reference users of a selected cluster. Figure 3 depicts the proposed model architecture.

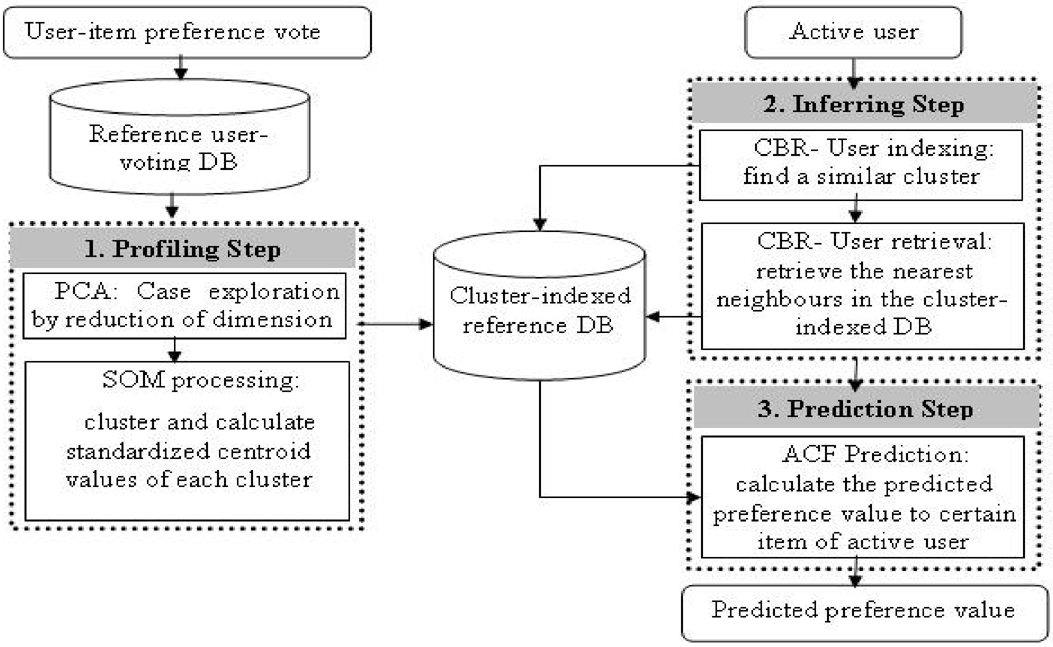

Figure 3. Model architecture of SOM cluster-indexing CBR CF recommendation

In the profiling step, PCA is used for visualizing users' patterns and reducing the dimension of input items before SOM clustering. Clusters are then derived from the SOM process using reference user DB with dense preference rating data. The reference users are indexed by the cluster and comprise the cluster-indexing reference DB.

Step 1. *User clustering with the centroid values of clusters by the SOM.*

1.1. Explore users' distribution by PCA.

1.2. Determine number of clusters using PCA factors.

1.3. Initialize weight vectors of the SOM.

1.4. Find clusters and the standardized centroid values of clusters.

As a classification part of the CBR process, an active user is compared with the standardized centroid values of each cluster. After that, the most similar cluster is determined and retrieved by the indexing and retrieval process of CBR.

Step 2. *Active user indexing and retrieval with CBR.*

2.1. Index an active user with the centroid values of clusters having minimum distance calculated by the k-Nearest Neighbour method, which is:

$$Min\_D = \sqrt{\sum_{m=1}^{n} \left| \frac{v_{a,m}}{S} - C_{ref,m} \right|^2}$$

where m is the given item, $v_{a,m}$ is the active user's rating value to item m, S is the standardizing factor for vote scale, and C_{ref}, m represents the centroid values to item m of the fixed clusters.

2.2. Retrieve the neighbours that were indexed in the same cluster.

Step 3. *Prediction of active user's predicted preference value.*

3.1. Calculate Pearson correlation between the active user and the most similar neighbours that were indexed in the same cluster.

$$w(a,i) = \sum_{j} \frac{(v_{a,j} - m_a)}{s_a} \frac{(v_{i,j} - m_i)}{s_i}$$

where $w(a,i)$ is the Pearson correlation coefficient to compute the weight for each user's contribution that is indexed in the same cluster.

3.2. Compute the prediction $P_{a,,t}$ of active user U_a on target item It.

$$P_{a,t} = m_a + k \sum_{i \neq a} w(a,i)(v_{i,t} - m_i)$$

where the sum is same cluster-indexing users in the reference DB, $v_{i,t}$ is the vote cast by the other user i on item t, and m_a is active user U_a 's mean vote. The constant k in front of the sum is an appropriate normalization factor.

4. Experiments

Experiments are run on open datasets, 6 different experimental predictors, and 2 evaluation metrics for the various algorithms. Results are compared with base-line models and other comparative models in terms of the level of data sparsity and machine learning techniques.

4.1. Data: MovieLens dataset

MovieLens data sets were collected by the GroupLens Research Project at the University of Minnesota (http://www.cs.umn.edu/Research/GroupLens/data/). The dataset consists of 100,000 ratings from 943 users on 1,682 movies with every user having at least 20 ratings and simple demographic information for the users (age, gender, occupation, zip code) is included. The ratings are on a numeric five-point scale with 1 and 2 representing negative ratings, 4 and 5 representing positive ratings, and 3 indicating ambivalence.

We sample a reference user set that has enough rating information to discover similar user patterns. The number of users that are extracted as the reference sample is 251. Among 100,000 rating records, 6,093 were generated from those 251 users rating data to 33 items out of 10 movie genres.

4.2. Evaluation metrics

Recommender systems researchers use several different measures for the quality of recommendations produced: statistical accuracy, decision-support metrics, coverage measures. Statistical accuracy is the accuracy of a filtering system measured by comparing the numerical predictions values against user ratings for the items that have both predictions and ratings. Mean Absolute Error (MAE) has been widely used to measure the performance of a prediction engine (Breese et al., 1998; Herlocker et al., 1999; Good et al., 1999) and Golberg et al., (2000) use Normalized MAE (NMAE). Decision-support metrics is how effectively predictions help a user select high-quality items from the item set, e.g. Receiver Operating Characteristic (ROC) sensitivity, precision/recall performance, reversal rates (Good et al., 1999). Coverage measures the percentage of items for which a recommendation system can provide predictions. Low coverage is usually due to small size of neighbourhood and sampling of users to find neighbours.

We apply two metrics to our evaluation – NMAE and the ROC curve including the area under ROC curve.

- *NMAE (Normalized Mean Absolute Error):* We look at the average absolute deviation of the predicted vote to the actual vote on items the users in the test set have actually voted on. That is, if the number of predicted votes in the test set is in the active case, ma, then the average absolute deviation for an active user is:

$$MAE = \frac{1}{m_a} \sum_{j=1}^{m_a} | P_{a,j} - v_{a,j} |$$

Since our numerical rating scale gives ratings over the range [1, 5], we normalize to express errors as percentages of full scale: Normalized Mean Absolute Error is:

$$NMAE = \frac{MAE}{r_{\max} - r_{\min}}$$

- *ROC curve and area under ROC curve:* This procedure is a useful way to evaluate the performance of classification schemes in which there is one variable with two categories by which subjects are classified. ROC sensitivity is a signal processing measure of the decision making power of a filtering system. Operationally, it is the area under the ROC – a curve that plots the sensitivity vs. 1- specificity of the test (Swets, 1988). Sensitivity refers to the probability of a randomly selected good item being accepted by the filter. Specificity is the probability of a randomly selected bad item being rejected by the filter. Points on the ROC curve represent trade-offs supported by the filter. The ROC sensitivity ranges from 0 to 1 where 1 is perfect and 0.5 is random.

To test the difference of 6 protocols' performance, we use one-way ANOVA with a post hoc test - Bonferroni procedure in the equal variance assumed for multiple comparisons statistics of MAE (Breese et. al, 1998). In this study, the post hoc test procedure is used for investigating the differences between specific experimental protocols in conjunction with ANOVA – comparing each protocol's MAE difference (Hair, Anderson, Tatham, & Black, 1995).

4.3. Experimental setup

At first, we build the dataset into three experimental sets for testing the effect of the available information level (the number of items on which active users have voted). In the first set, named *Allbut1*, we withhold one selected item for each user in the test set, and try to predict its value given all the other votes the user has voted on. In the second and third set of experiments, we select 5 and 10 votes from each test user as the observed votes, and then attempt to predict those preference levels. We call these *Given5*, and *Given10*. The Allbut1 experiments measure the algorithms' performance when given as much data as possible from each test user. The Given experiments look at users with less data available, and examine the performance of the algorithms when there is relatively little known about an active user.

Secondly, we present metrics derived from empirical analysis of the proposed SOM cluster-indexing CBR CF model, hereafter referred to as SCP, compared to baseline models and comparative 3-step models.

<table>
<tr><th colspan="3">Protocol</th><th>Description</th></tr>
<tr><td colspan="2">Proposed model</td><td>SCP</td><td>♦ SOM cluster-indexing CBR CF predictor</td></tr>
<tr><td rowspan="5">Comparative model</td><td rowspan="3">Baseline model</td><td>UAP</td><td>♦ by-user-average CF predictor</td></tr>
<tr><td>IAP</td><td>♦ by-item-average CF predictor</td></tr>
<tr><td>SPP</td><td>♦ simple Pearson CF predictor</td></tr>
<tr><td rowspan="2">3-step model</td><td>SIP</td><td>♦ SOM cluster induction CF predictor</td></tr>
<tr><td>SNP</td><td>♦ SOM cluster neural network CF predictor</td></tr>
</table>

Table 1. List of experiment protocols

4.3.1. SOM cluster-indexing CBR CF (SCP) model

On the belief that an affinity group can be clustered according to the distribution of their rating values, a couple of clustering and visualizing methods are performed to find the number of clusters. In this study, clustering techniques involve two distinct works: (1) the determination of the number of clusters present in the reference-base; and (2) the assignment of reference users to one cluster. The number of clusters, which is the number of nodes in the output layer, depends on the expected number of clusters, but there is currently no apparent practical or theoretical way of determining the optimal size of the output layer (Nour & Madey, 1996). There is a possible instability due to the randomness of clusters, so it requires a policy for initial cluster selection. To reduce this possibility, the SCP model contains the PCA process and preliminary SOM clustering. At first, each reference user's distribution is summarized by PCA using 6,093 rating records in the 33 movies among 10 genres before SOM clustering. In this experiment, we chose 3 components by PCA, and data is projected onto the "eigen-plane" for pre-visualization shown as Figures, 4 and 5.

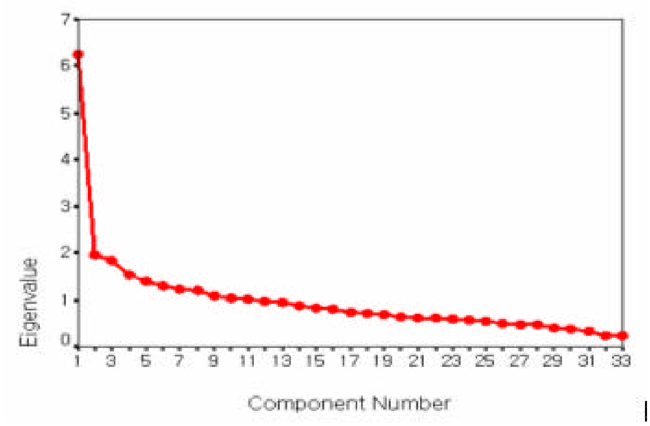

Figure 4. Scree curve of eigenvalue explained by components. The largest amount of eigenvalue is explained by the first component. The first three components can be the representative factors of the population dataset.

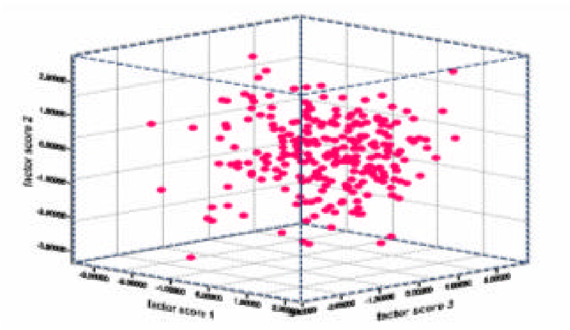

Figure 5. Scatter plot of reference users projected onto the 3D "eigen-plane" for visualization.

In preliminary SOM clustering, the number of clusters is set as 2 to 10. When deciding the optimal number of clusters, the lowest cluster number is selected so that each cluster can have as many indexed reference users as possible. If a cluster has no or only a few users, this cluster does not have sufficient user cases to find a more similar user within indexed users. The total number of neurons in the output layer is decided as being 4 clusters according to the results of the preliminary SOM that was performed to find the number of clusters.

Based on the SOM learning, each user in reference DB is indexed into a cluster. In addition to indexing, the centroid values of each item are deduced and its average values of each genre cluster suggest each cluster's inferable genre preference characteristics shown in Table 2. It appears as if each cluster is grouped by rating level, and a quick glance seems to suggest that the average rating is going up, i.e. C4 (0.7810) is relatively more liberal than C1 (0.5400).

	Genre	Children's	Drama	Adventure	Sci-Fiction	Crime	Thriller	War	Romance	Comedy	Horror	Average
Cluster	C1	0.4845	*0.4775*	0.5432	0.5435	0.6473	0.5477	0.6090	0.4980	0.5540	**0.6870**	0.5400
	C2	0.5595	0.6050	0.6756	0.6525	**0.8367**	0.6693	0.6570	0.5590	*0.5540*	0.5790	0.6490
	C3	0.6910	0.5820	0.7218	**0.7488**	0.6340	0.6857	0.7090	0.5970	0.6313	*0.5450*	0.6860
	C4	0.7180	0.6965	0.8336	0.8124	0.8210	0.7910	**0.8820**	*0.6785*	0.6897	0.7230	0.7810

a. Genre average centroid values of SOM cluster

Genre		**Children's**		**Drama**		**Adventure**					**Science-Fiction**	
Item		Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7	Item 8	Item 9	Item 10	Item 11
Cluster	C1	0.407	0.562	0.488	0.467	0.431	0.690	0.543	0.690	0.362	0.705	0.676
	C2	0.520	0.599	0.611	0.599	0.666	0.790	0.637	0.752	0.533	0.822	0.676
	C3	0.739	0.643	0.581	0.583	0.675	0.841	0.741	0.754	0.598	0.942	0.857
	C4	0.751	0.685	0.766	0.627	0.759	0.966	0.828	0.913	0.702	0.974	0.941

Genre											**Crime**	
Item		Item 12	Item 13	Item 14	Item 15	Item 16	Item 17	Item 18	Item 19	Item 20	Item 21	Item 22
Cluster	C1	0.597	0.516	0.364	0.542	0.570	0.523	0.503	0.556	0.427	0.593	0.670
	C2	0.689	0.622	0.489	0.607	0.682	0.685	0.673	0.699	0.534	0.875	0.783
	C3	0.884	0.705	0.606	0. 766	0.647	0.599	0. 681	0.758	0.741	0.592	0.666
	C4	0.891	0.705	0.663	0.839	0. 790	0.783	0.797	0.850	0.681	0.902	0.784

Genre			**Thriller**			**War**	**Romance**		**Comedy**			**Horror**
Item		Item 23	Item 24	Item 25	Item 26	Item 27	Item 28	Item 29	Item 30	Item 31	Item 32	Item 33
Cluster	C1	0.679	0.466	0.467	0.710	0.609	0.569	0.427	0.655	0.601	0.406	0.687
	C2	0.852	0.628	0.560	0.820	0.657	0.642	0.476	0.631	0.585	0.446	0.579
	C3	0.644	0.733	0.588	0.736	0.709	0.645	0.549	0.800	0.519	0.575	0.545
	C4	0.777	0.835	0.615	0.923	0.882	0.779	0.578	0.815	0.630	0.624	0.723

b. Centroid values of SOM cluster by item

Table 2. Centroid values of SOM clusters - Bold numbers mean the most preferred genre in each cluster; and, italicized numbers are the worst.

However, a closer look shows that each cluster has different genre and movie preferences. For example, C1 is a comparatively negative group, but this group prefers tough genres such as horror, crime, and war movies to soft ones such as drama, romance films. On the other hand, C3 rates higher than C1. This affinity group likes science-fiction and adventure rather than horror, which is the most preferred genre for C1.

4.3.2. Comparative models

Previous research on prediction algorithms for CF has tended to compare the performance of algorithms exclusively within that research, so making comparisons of algorithm performance from paper to paper is difficult. We show three baseline predictors, *by-user-average CF predictor (UAP)*, *by-item-average CF predictor (IAP)* and *simple Pearson CF predictor (SPP)*, to provide benchmarks against which any predictor could be compared. The baseline algorithms are simple, efficient and return reasonable results. The UAP returns the average of the votes the given user has already entered. The IAP gives the average vote for the given movie of all users that have voted for that movie (Fisher, Hildrum, Hong, Newman, Thomas, & Vuduc, 2000). The SPP returns the Pearson correlation based neighbourhood prediction.

To demonstrate the utility of the SOM cluster-indexing CBR model, we change the CBR process into neural network and induction technique as a classification method withholding the 1st SOM profiling step and the 3rd Pearson correlation-based prediction step. The *SOM neural network CF Predictor (SNP)* uses the well known back propagation neural network algorithm in the classification step and all decision coefficients are controlled to achieve the best prediction accuracy. The *SOM induction CF Predictor (SIP)* uses decision tree technique known as induction, especially SEE 5.0 algorithm which is an upgraded version of Quinlan's (1993) decision tree classifier C4.5.

5. Results

To validate the effectiveness of SCP, prediction accuracy is compared with the comparative experimental algorithms in terms of NMAE and the area under the ROC curve. Table 3 tabulates the accuracy results of each protocol on the MovieLens dataset.

Metric / Protocol	NMAE			ROC Area
	Allbut1	Given5	Given10	
UAP	0.1600	0.1863	0.1901	0.7127
IAP	0.1714	0.1930	0.1951	0.6071
SPP	0.1497	0.1719	0.1734	0.8006
SNP	**0.1413**	0.1557	0.1638	0.8166
SIP	0.1687	0.1892	0.1933	0.7430
SCP	0.1475	**0.1524**	**0.1583**	**0.8461**
Average	0.1564	0.1747	0.1790	

(a) Prediction accuracy and area under ROC curve of each protocol – Lower value of NMAE metric indicates better performance, and for ROC area, it is vice versa.

	IAP	SPP	SNP	SIP	SCP
UAP	-0.0202	0.0665 **	0.1052 ***	-0.0131	0.1271 ***
IAP	--	0.0867 **	0.1253 ***	0.0070	0.1472 ***
SPP	--	--	0.0387 *	-0.0797 **	0.0605 **
SNP	--	--	--	-0.1183 ***	0.0219
SIP	--	--	--	--	0.1402 ***

(b) Statistical significance test - one-way ANOVA with the post hoc test - Bonferroni procedure based on MAE: Mean differences and significance level at ***1%, **5% and *10% level for the pair-wise comparison of performance between protocols.

Table 3. Performance results.

SPP reflects better users' preference than simple average predictors (UAP, IAP) in all the experiment datasets, Allbut1, Given5 and Given10 under the 5% significance level among the baseline model, which is the same result as the ROC area metric. This result supports that using a CF algorithm is more accurate than simple average predictors that have not considered the like-minded affinity group to predict active users' preference. Comparing between UAP and IAP, when not enough rating data is available to apply a CF algorithm, user average value has more predictive power than item average value.

SCP and SNP, which use a clustering-classification method, yield superior results to SPP, which is a type of memory-based model. However, the SIP model shows far worse performance than all other experimental protocols. So model-based CF methods, especially machine learning technique-based CF models, have promising potential to improve, but their success depends on methodology suitability. Especially, the SCP model shows an outperforming result (0.1583 NMAE on Given 10 and an ROC area of 0.8461) compared to other comparative modelling by NMAE and ROC area metrics. It dominates UAP, IAP, SPP, SIP models under the 5% and 1% significance levels and yields better performance than SNP. According to our experiment, the SCP model can alleviate prediction error about 4%.

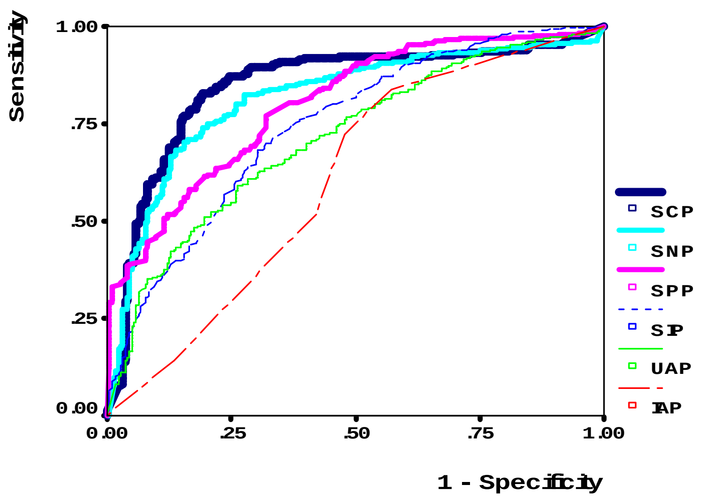

Figure 6. ROC curve comparison of CF prediction protocols. Upper line indicates more prediction accuracy.

From the viewpoint of data sparsity, Given10's average NMAE (0.1790), which has less preference information, reports a higher error rate than Allbut1 (0.1564) and Given5 (0.1747). This implies that as explicit information, about preference rating to items, becomes sparse, the prediction accuracy decreases.

Overall, among the ROC curves illustrated in Figure 6, the SCP model is stably dominant to other clustering-classification CF models and memory-based models. This implies that when the item recommendation criterion is changed, the SCP model can be flexibly applied. For example, even if the recommendable item criterion is changed from 5 stars to 4 stars in the movie recommendation, the SCP model does still works.

6. Conclusion

In this paper, we propose an SCP model, a new approach in the CF recommendation field, which applies two machine learning techniques, SOM and CBR to the consecutive CF prediction process. This model takes profiling, inferring and predicting steps that are operated in both on- and off-line parts. The SOM facilitates affinity user grouping and extraction of representative centroid values of each cluster's items for assisting case indexing and retrieval of CBR. This study shows that cluster-indexing CBR is an effective user indexing method: The performance of our model yields superior results compared to memory-based CF techniques and other previous hybrid CF models. The NMAE values induced from our model indicate that predicted rating values will be within roughly 15% of the true rating values. So items with predicted ratings well above the mean for a new user in many cases will correspond to desirable items for that user. These accuracies are comparable with those reported for a completely different data set (jokes); the algorithms in Goldberg et al., (2000) show NMAE from 0.187 to 0.237 in the 20 unit rating scale [-10, +10]. Herlocker et al., (1999) report MAE from 0.768 to 0.828. When these are normalized to the 4 unit rating scale [1, 5], they yield NMAE from 0.192 to 0.207 in the same MovieLens data set. According to Goldberg et al., (2000), if user ratings are distributed uniformly or normally, random predictions yield NMAE of 33% and 28% respectively. This suggests that there is room for improved accuracy for all current CF algorithms. In fact, our model yields superior performance when compared to other traditional memory-based CF algorithms and also other NN and induction based CF prediction algorithms, while at the same time alleviating the burden of computational complexity.

As the size of the user pool and available items are usually very large, system efficiency is also a major concern like accuracy. For SOM clustering of our study, most of the computational cost is involved in the training process, and this process is done off-line in the profiling step. After that, the CBR process and CF prediction in the selected cluster are a compact representation of the raw ratings information, and thus the time and space complexities on making recommendations are quite low. With the arrival of more preference ratings and/or new items, this CF recommendation model will be re-trained and updated periodically.

We are experimenting with a number of variations, such as k-means clustering and hybrid approaches with adaptive online weighting to further improve accuracy without altering online computation time. In fact, this study compares several computational approaches to CF recommendation, considering prediction accuracy and response speed simultaneously. Especially, numerous data mining techniques, such as SOM, NN, induction, and CBR have shown the potential for improving. The promising potential for CF systems can be investigated by integrating with product/customer-specific information profiling, implicit information analysis such as web-page navigation history and retrieval technology. In the future research, we will suggest a hybrid recommendation algorithm and try to apply our model into a real-world personalized recommendation site.

References

Aamodt, A., & Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *Artificial Intelligence Communications,* 7(1), 39–59.

Aggarwal, C.C., Wolf, J.L., Wu, K., & Yu, P.S. (1999). Horting hatches an egg: a new graph-theoretic approach to collaborative filtering. *Proceedings of the ACM KDD'99 Conference* (pp. 201-212). San Diego, CA.

Balabanovíc, M., & Shoham, Y. (1997). Fab: content-based, collaborative recommendation. *Communications of the ACM, 40 (3),* 66-72.

Basu, C., Hirsh, H., & Cohen, W. (1998). Recommendation as classification: using social and content base information in recommendation. *Proceedings of the 1998 Workshop on Recommender Systems* (pp.11-15).

Bradley, P.S. (1994). Case-based reasoning: business applications. *Communication of the ACM, 37(3),* 40–43.

Breese, J.S., Heckerman, D., & Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98)* (pp. 43-52).

Chiu, C. (2002). A case-based customer classification approach for direct marketing. *Expert Systems with Applications, 22(2),* 163-168.

Cho, Y. H., Kim, J. K., & Kim, S. H. (2002). A personalized recommender system based on web usage mining and decision tree induction. *Expert Systems with Applications, 23(3),* 329-342.

Choy, K. L., Lee, W. B., & Lo, V. (2002). Development of a case based intelligent customer–supplier relationship management system. *Expert Systems with Applications, 23(3),* 281-297.

Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., & Sartin, M. (1999). Combining content-based and collaborative filters in an online newspaper. *ACM SIGIR '99 Workshop on Recommender Systems*, Berkely, CA.

Delgado, P.J., & Ishii, N. (1999). Memory-based weighted majority prediction for recommender systems. *SIGIR Workshop on Recommender Systems.*

Fisher, D., Hildrum, K., Hong, J., Newman, M., Thomas, M., & Vuduc, R. (2000). SWAMI: A framework for collaborative filtering algorithm development and evaluation. *SIGIR 2000,* Athens, Greece.

Goldberg, D., Nichols, D., Oki, B.M., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM, 35(12),* 61-70.

Goldberg, K.., Roeder, R., Gupta, D., & Perkins, C. (2001). Eigentaste: A Constant Time Collaborative Filtering Algorithm. *Information Retrieval Journal, 4(2),* 133-151.

Good, N., Schafer, J.B., Konstan, J., Borchers, A., Sarwar, B., Herlocker, J., & Riedl, J. (1999). Combining collaborative filtering with personal agents for better recommendations. *Proceedings of the 1999 Conference of the American Association of Artificial Intelligence (AAAI-99).*

Günter, S., & Bunke, H. (2002). Self-organizing map for clustering in the graph domain, *Pattern Recognition Letters, 23(4),* 405-417.

Ha, S. H., & Park, S. C. (1998). Application for data mining tool to hotel data mart on the Internet for database marketing. *Expert Systems with Applications,* 15(1), 1-31.

Hair, J.F., Anderson, R.E., Tatham, R.L., & Black, W.C. (1995). *Multivariate data analysis with readings*, Englewood Cliffs, NJ: Prentice Hall.

Herlocker, J., Konstan, J., & Riedl, J. (2000). Explaining collaborative filtering recommendations, *Proceedings of the ACM 2000 Conference on Computer Supported Cooperative Work* (pp.2-6).

Herlocker, J., Konstan, J., Borchers, A., & Riedl, J. (1999). An algorithmic framework for performing collaborative filtering. *Proceedings of the 1999 Conference on Research and Development in Information Retrieval.*

Hill, W., Stead, L., Rosenstein, M. and Furnas, G. (1995). Recommending and evaluating choices in a virtual community of use. *CHI 95* (pp. 194-201). Denver, CO, ACM Press.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology, (24),* 417-441.

Hui, S.C., Fong, A.C.M., & Jha, G. (2001). A web-based intelligent fault diagnosis system for customer service support. *Engineering Applications of Artificial Intelligence, 14(4),* 537-548.

Kim, K.S., & Han. I.G. (2001). The cluster-indexing method for case-based reasoning using self-organizing maps and learning vector quantization for bond rating cases. *Expert Systems with Applications, 21(3)*, 147-156.

Kohonen, T., Hynninen, J., Kangas, J., & Laaksonen, J. (1996). SOM_PAK: The self-organizing map program package. *Technical Report A31*, Helsinki University of Technology, Laboratory of Computer and Information Science, FIN-02150.

Kolodner. J.L. (1993). *Case-based reasoning*, Morgan Kaufmann: Los Altos, CA.

Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J. L., Gordon, L. R. and Riedl, J. (1997). GroupLens: Applying collaborative filtering to Usenet news. *Communications of the ACM, 40(3),* 77-87.

Mangiameli, P., Chen, S.K., & West, D. (1996). A comparison of SOM neural network and hierarchical clustering methods. *European Journal of Operation Research, 93*, 402-417.

Nour, A.N., & Madey, G.R. (1996). Heuristic and optimization approaches to extending the Kohonen self organizing algorithm. *European Journal of Operation Research*, *93,* 428-448.

Pazzani, M.J. (1999). A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review, 13(5/6),* 393-408.

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Phil. Mag,(*2), 559-572.

Pennock, D. M., Horvitz, E., Lawrence, S., & Giles, C. L. (2000). Collaborative filtering by personality diagnosis: A hybrid memory- and model-based approach, *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI-2000)* (pp. 473-480). Stanford, CA.

Quinlan, J.R. (1993). *C4.5: Programs for Machine Learning*, Morgan Kaufmann.

Resnick, P., Iacovou, N., Sushak, M., Bergstrom, P., & Riedl, J. (1994). GroupLens: An open architecture for collaborative filtering of netnews. *Proceedings of the 1994 Computer Supported Collaborative Work Conference.*

Sarwar, B.M., Karypis, G., Konstan, J., & Riedl, J. (2000). Analysis of recommendation algorithms for e-commerce. *Proceedings of ACM E-Commerce 2000 Conference,* (pp.158-167).

Schafer, J.B., Konstan, J.A., & Riedl J. (2001). E-commerce recommendation applications. *Data Mining and Knowledge Discovery, 5(1-2),* 115-153.

Shardanand, U., & Maes, P. (1995). Social information filtering: algorithms for automating "Word of Mouth." *Proceedings of ACM CHI '95* (pp. 210-217). Denver, CO.

Sheng-Tun, L. (2002). A web-aware interoperable data mining system. *Expert Systems with Applications, 22(2),* 135-146.

Swets, J. A. (1988). Measuring the accuracy of diagnostic systems. *Science, 240,* 1285-1289.

Vellido, A., Lisboa, P.J.G., & Meehan, K. (1999). Segmentation of the on-line market using neural networks. *Expert Systems with Applications*, 17(4), 303–314.

Watson, I. (1997). Applying case-based reasoning: Techniques for enterprise systems, Morgan Kaufmann, CA, USA.

Yuan, S. & Chang, W. (2001). Mixed-initiative synthesized learning approach for web-based CRM. *Expert Systems with Applications, 20(2),* 187–200.