Expert Systems with Applications 36 (2009) 8324–8332

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A collaborative filtering method based on artificial immune network

A. Merve Acilar *, Ahmet Arslan

Selcuk University, Eng.-Arch. Fac., Computer Eng., 42003 Konya, Selcuklu, Turkey

ARTICLE INFO

Keywords:
Recommender systems
Collaborative filtering
Artificial immune system
Sparsity
Scalability
aiNet

ABSTRACT

A system is seriously required for helping users to find their path on the shopping and entertainment web sites where the amounts of on-line information vastly increase. Therefore, recommender systems, new type of internet based software tool, appeared, and became an appealing subject for researchers. Collaborative filtering (CF) technique based on user is the one of the method widely used by recommender systems but they have some problems for waiting to be developed solutions that are more efficient. One of these mainly problems is data sparsity. While the number of products is increase, the ratio of common rated products is decrease so calculating the computations of neighbourhood become difficult. The other one is scalability which is the performance problem of the existing algorithms on the datasets has large amounts of information.

In this article, we tackle these two questions: (1) how the data sparsity can be reduced ? (2) How to make recommendation algorithms more scalable? We present an approach to addressing the both of these problems at the same time by using a new CF model, constructed based on the Artificial Immune Network Algorithm (aiNet). It is chosen because aiNet is capable of reducing sparsity and providing the scalability of dataset via describing data structure, including their spatial distribution and cluster interrelations. The new user-item ratings dataset reduced by applying aiNet (aiNetDS) given more stable results and produced predictions more quickly than the raw user-item ratings dataset (rawDS). Besides, the effects of using clustering for forming the neighbourhoods to the system performance are investigated. For this, both of these dataset are clustered by using *k*-means algorithm and then these cluster partitions are used as neighbourhoods. As a result, it has been shown that the clustered aiNetDS is given more accurate and quick results than the others are.

1. Introduction

The amount of information in the world is growing so quickly with the widespread and easy usage of internet. Therefore, acquired data from the sites of e-commerce, entrainment etc. is increasing far more rapidly than our ability to process it. All of them such as Amazon, Yahoo and CDNow suggest a lot of chooses to their potential customers everyday which makes difficult to find the true products that best meet user's needs and preferences. For overcoming this problem, recommender systems appeared and became an appealing subject for researchers. It is a kind of personal software assistant learning the evolving interests of their users by applying the information-processing algorithms to the mass of this information.

These recommendations base on the top overall sellers on a site, on the demographics of the consumer, or an analysis of the past buying behaviour of the consumer as a prediction for future buying behaviour. The forms of them include suggesting products to the consumer, providing personalized product information, summarizing community opinion, and providing community critiques. Broadly, these recommendation techniques are part of personalization on a site because they help the site adapt itself to each customer. Under this broader definition, recommender systems serve to support a customization of the consumer experience in the presentation of the products sold on a Web site. In a sense, recommender systems enable the creation of a new store personally designed for each consumer. Recommender systems enhance E-commerce sales in three ways: helping customers find products they wish to purchase; converting browsers into buyers; improving cross-sell by suggesting additional products for the customer to purchase; improving loyalty by creating a value-added relationship between the site and the customer (Schafer, Konstan, & Riedl, 2001).

Mainly two distinct techniques are used by today recommendation systems: content-based methods and collaborative methods. Content-based methods analyze the content of information sources (e.g. the HTML source of web pages) that have been rated to create a

* Corresponding author. Tel.: +90 332 2233722; fax: +90 332 241 0635.
E-mail addresses: msakiroglu@selcuk.edu.tr, msakiroglu@hotmail.com (A. Merve Acilar).

0957-4174/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2008.10.029

profile of the user's interests in terms of regularities in the content of the information that was rated highly. This profile may be used to rate other unseen information sources or to construct a query of a search engine (Pazzini, 1999). In contrast to content-based techniques, collaborative methods do not need any information about item's content. 'The task in collaborative filtering is to predict the utility of items to a particular user (the active user), based on a database of user votes from a sample or population of other users (the user database)' (Breese, Heckerman, & Kadie, 1998). Both approaches share the common aim of assisting in the user's search for items of interest.

1.1. Related work

In this article, we focus on collaborative filtering techniques. Some of the research literature related to collaborative filtering is presented continuation of this section.

Tapestry (Goldberg, Nichols, Oki, & Terry, 1992) is one of the earliest implementations of collaborative filtering-based recommender systems. This system relied on the explicit opinions of people from a close-knit community, such as an office workgroup. In 1994, Resnick et al. proposed a recommender system based on collaborative filtering for recommend news to newsgroup's users. The pearson correlation coefficient was firstly used for computing the similarity between two users in this article. In 1998, Breese et al. described CF techniques based on correlation coefficient, vector similarity, and statistical Bayesian methods and compared the predictive accuracy of the various methods in a set of representative problem domains (Breese et al., 1998).

Later, several ratings-based automated recommender systems were developed (Billsus & Pazzini, 1998; Claypool, Gokhale, & Miranda, 1999; Herlocker, Konstan, & Riedl, 2000; Pennock, Horvitz, Lawrence, & Lee Giles, 2000; Sarwar, Karypis, Kontsan, & Riedl, 2000). Although there are achieving widespread success on the web, still CF algorithms have some problems for waiting to be developed solutions that are more efficient. One of these mainly problems is data sparsity. While the number of products is increase, the ratio of common rated products is decrease so calculating the computations of neighbourhood become difficult. The other one is the scalability problem which is a performance problem of the existing algorithms on the datasets has large amounts of information. For solving these problems, a lot of study has been made until 2000. Goldberg et al. proposed a collaborative filtering algorithm called Eigentaste that use *universal queries* to elicit real-valued user ratings on a common set of items and applies principal component analysis (PCA) to the resulting dense subset of the ratings matrix. PCA facilitated dimensionality reduction for offline clustering of users and rapid computation of recommendations (Goldberg, Roeder, Gupta, & Perkins, 2000). Sarwar explored item-based collaborative filtering techniques to address the sparsity and scalability. Item based techniques first analyze the user-item matrix to identify relationships between different items, and then use these relationships to indirectly compute recommendations for users (Sarwar, 2001). Li & Kim applied clustering techniques to the item-based collaborative filtering framework to solve the cold start problem also it suggested a way to integrate the content information into collaborative filtering (Li & Kim, 2003). Huang et al. applied an associative retrieval framework and related spreading activation algorithms to explore transitive associations among consumers through their past transactions and feedback. They thought transitive associations were a valuable source of information to help infer consumer interests and could be explored to deal with the sparsity problem (Huang, Chen, & Zeng, 2004). Cheung et al. used the latent class model (LCM) to alleviate the sparsity problem. Firstly, they studied how the LCM can be extended to handle customers and products outside the training set. In addition, they proposed the use of a pair of LCMs (called dual latent class model – DLCM), instead of a single LCM, to model customers' likes and dislikes separately for enhancing the prediction accuracy (Cheung, Tsui, & Liu, 2004). Li et al. described a collaborative music recommender system (CMRS) based on their proposed item-based probabilistic model, where items are classified into groups and predictions are made for users considering the Gaussian distribution of user ratings. In addition, that model has been extended for improved recommendation performance by utilizing audio features that help alleviate three well-known problems associated with data sparseness in collaborative recommender systems: user bias, non-association, and cold start problems in capturing accurate similarities among items (Li, Myaeng, & Kim, 2007). Ahn presented a new heuristic similarity measure that focuses on improving recommendation performance under cold-start conditions where only a small number of ratings are available for similarity calculation for each user (Ahn, 2008).

In addition to them, data/web mining are used for improved the quality of CF-based recommender systems frequently. For example, Cho et al. proposed a recommendation methodology based on Web usage mining, and product taxonomy to enhance the recommendation quality and the system performance of current CF-based recommender systems (Cho & Kim, 2004). In other example, a new methodology for enhancing the quality of CF recommendation that uses customer purchase sequences. The proposed methodology was applied to a large department store in Korea and compared to existing CF techniques (Cho, Cho, & Kim, 2005). Another work proposed a knowledge map platform to provide an effective knowledge support for utilizing composite e-services. A data mining approach was applied to extract knowledge patterns from the usage records of composite e-services. Based on the mining result, topic maps were employed to construct the knowledge map. Meanwhile, the proposed knowledge map was integrated with recommendation capability to generate recommendations for composite e-services via data mining and collaborative filtering techniques (Liu, Ke, Lee, & Lee, 2008).

Another group of researchers studied on recommender systems benefited from several kinds of artificial intelligence techniques in their studies. Artificial immune system (AIS) inspired by theoretical immunology and observed immune functions, principles and models is the one of these techniques. Cayzer and Aickelin applied the AIS to the task of film recommendation by collaborative filtering. They explored to identify a sub-set of good matches on which recommendation could be based using the highly distributed, adaptive and self-organising nature of AIS (Cayzer & Aickelin, 2005). Chen et al. investigated the effect of different affinity measure algorithms for the AIS. Two different affinity measures, Kendall's Tau and Weighted Kappa, were used to calculate the correlation coefficients for the movie recommender (Aickelin & Chen, 2004). Morrison built an Artificial Immune System that found a group of users in the database who were similar to the target user in their web site preferences. The idiotypic effects would ensure that this group was as diverse as possible and created an ideal base for predicting and recommending web sites (Morrison, 2003). Mihaljevic et al. proposed in their paper addresses construction of a web portal news article recommender based on artificial immune system combined with Danger theory. System knowledge represented learned user preferences using implicit tracking of user actions (Mihaljevic, Cvitas, & Zagar, 2006). Sobecki and Szczepański presented application of AIS collaborative filtering in the system Reporter that was based on Wiki-news and recommends both articles and interface layouts. Wiki-based information systems were gaining its popularity among many different users so it was becoming necessary to apply recommendation for most effective information delivery according to them (Sobecki & Szczepański, 2007).

1.2. Contributions

In this study, using a new collaborative filtering method based on artificial immune network is proposed as a solution for sparsity and scalability problems. To addressing the both of these problems at the same time by using a new CF model, constructed based on the Artificial Immune Network Algorithm (aiNet) is chosen because aiNet is capable of reducing sparsity and providing the scalability of dataset via describing data structure, including their spatial distribution and cluster inter-relations. Besides, the effects of using clustering for forming the neighbourhoods to the system performance investigation, datasets are clustered by using k-means algorithm and then these cluster partitions are used as neighbourhoods.

1.3. Organization

This paper is organized as follow. Basic principles and features of Collaborative Filtering and Artificial Immune Network Model-aiNet are explained in Sections 2 and 3 respectively. In Section 4, a new collaborative filtering approach based on artificial immune network has been introduced. The experimental results and the discussion of them have been given in Section 5. Finally, in Section 6, we present our conclusions and perspectives.

2. Collaborative filtering

Recommender systems apply data analysis techniques to problem of helping users find the items they would like to purchase at e-commerce or entertainment sites by producing a predict likeliness score or a list of top-N recommended items for a given user. Collaborative filtering (CF) is the most important personalized recommendation method widely in recommender systems (Li, Lu, & Xuefeng, 2005). The basic idea of a CF system is to generate recommendations based on the experiences of past similar users. Formally, in CF recommenders, there are two kind of sets $U = \{u_1, u_2, \ldots, u_m\}$ and $I = \{I_1, I_2, \ldots, I_n\}$,which are represent users and items (such as books, movies and etc.), respectively. Here, each user u_i, $i = 1,2,\ldots,m$ has rated a subset of items I. The rating of user u_i for item s_j, $j = 1,2,\ldots,n$ is denoted by $r_{i,j}$. All the available ratings are collected in a $m \times n$ user-item matrix denoted by R. Then the similarities among the users are computed. One common approach to compute the similarity of preferences among users is the Pearson correlation coefficient. Computing of Pearson correlation coefficient is given in Eq. (1). Based on these similarity values, CF system computes neighborhoods of the active user a and users are ranked by their similarity measures in relation to the target user a. The k most similar (highest ranked) users are selected as the k-nearest neighbours of user a. Then, system predicts the ratings that user a would probably give to the other not yet-rated items via the help of these neighbourhoods using Eq. (2). Finally, the CF system will output several items with the best predicted ratings as the recommendation list (Sakiroglu, 2005).

The other output of the CF system is the top-N recommendations. For the generation of this list, firstly users are ranked by their similarity measures in relation to the target user a. The k most similar (highest ranked) users are selected as the k-nearest neighbours of user a. The frequency count of item is calculated by scanning the rated item of the k-nearest neighbours. The items then are sorted based on frequency count. The N most frequent items that have not yet been rated by target user u are selected as the top-N recommendations (Liu & Shih, 2005).

2.1. Computing of Pearson correlation coefficient

$$Corr(a,k) = \frac{\sum_{i=1}^{n}(a_i - \bar{a})(k_i - \bar{k})}{\sqrt{\sum_{i=1}^{n}(a_i - \bar{a})^2 \sum_{i=1}^{n}(k_i - \bar{k})^2}} \tag{1}$$

The notations $\bar{a}$ and $\bar{k}$ denote the average rating of the items rated by the users a and k, respectively. Moreover, the variable n denotes the set of common rated products. Additionally, the parameters a_i and k_i indicates the rating value given to item i by user a and k, respectively.

2.2. Computing prediction

The prediction score, $P_{a,j,}$ on item j for target user a is computed as follows (Resnick, Lacovou, Suchak, Bergstrom, & Riedl, 1994)

$$P_{a,j} = \bar{a} + \frac{\sum_{(k \in N) \wedge (k\_rates\_j)} (k_j - \bar{k}) * similarity(a,k)}{\sum_{(k \in N) \wedge (k\_rates\_j)} |similarity(a,k)|} \tag{2}$$

Here, $\bar{a}$ and $\bar{k}$ are the average rating of user a and k, respectively, k_j is the rating given by user k to item j. Similarity (a,k) is the similarity among users a and k, computed using Pearson correlation given in Eq. (1). The summation is calculated for only those neighbours who have rated item j.

2.3. Sparsity & scalability problems of CF

When the number of users and items in an e-commerce or entertainment site grow rapidly, two major issues must be addressed (Sarwar, 2001).

The first issue is related to sparsity. In a large ecommerce site such as Amazon.com, there are millions of products and so customers may rate only a very small portion of those products. Most similarity measures used in CF work properly only when there exists an acceptable level of ratings across customers in common. Such sparsity in ratings makes the formation of neighbourhood inaccurate, thereby resulting in poor recommendation. Many approaches have been proposed to overcome the sparsity problem. These approaches can be classified into three categories: implicit ratings, hybrid filtering and product-to product correlation. The implicit ratings approaches attempt to increase the number of ratings through observing customers' behaviour. The hybrid filtering approaches combine content-based filtering and CF for augmenting sparse preference ratings. These approaches learn to predict which products a given customer will like by matching properties associated with each product to those associated with products that he/she has liked in the past, and then use such a content-based prediction to convert a sparse customer profile into a dense one. Instead of identifying the neighborhood of similar customers, the product-to-product correlation approach analyzes the customer profile to identify relationships between different products and then uses these relations to compute the prediction score for a given customer–product pair (Cho & Kim, 2004).

The second issue is related to scalability. Recommender systems for large e-commerce sites have to deal with millions of customers and products. Because these systems usually handle very high dimensional profiles to form the neighbourhood, the nearest neighbour algorithm is often very time-consuming and scales poorly in practice. To address the scalability problems in CF-based recommender systems, a variety of approaches have been developed. These approaches can be classified into two main categories: dimensionality reduction techniques and model-based approaches (Sarwar, 2001). Latent Semantic Index (LSI) is a widely used dimensionality reduction technique. It uses singular value decomposition (SVD) to factor the original rating space into three matrices and performs the dimensionality reduction by reducing the singular matrix. In model-based approaches, a model is first built based on the rating matrix and then the model is used in making recommendations. Usually, the model is expensive to build, but rapid to execute. Several data mining techniques such as Bayesian network, clustering and association rule mining have been applied to building the model (Cho & Kim, 2004).

In this paper, we used a hybrid model for solving the scalability problem. We used mechanism of network suppression of artificial immune network model-aiNet for dimensions reduction, and then clustered the reduced data set using *k*-means algorithm as a model based approach. This clustered dataset via reduced by aiNet is used for producing predicts to users. Sparsity problem is solved by using implicit rating. These ratings are produced using the hypermutation mechanism of aiNet.

3. Artificial immune network model – aiNet algorithm

Immunology can be defined as the study of the defence mechanism that confers resistance against diseases. The system whose main function is to protect our bodies against the constant attack of external micro organisms is called the immune system. The immune system consists of a complex set of cells and molecules that protect our bodies against infection. This system is a natural, rapid, and effective defence mechanism for a given host against infections (de Castro & Timmis, 2002). Besides, it possess the cells which are capable of pattern recognition, diversity, autonomy, noise tolerance, self-organization, learning, gaining memory, fault detection, optimization etc. For benefiting these characteristics of immune system, a new research field is emerged called artificial immune system. It can be defined as computational systems inspired by theoretical immunology and observed immune functions, principles, and models, which are applied to problem solving. In this section firstly, network theory of immune system will be described then the aiNet algorithm developed based on this theory will be explained.

The immune network theory was formally proposed by N.K. Jerne in 1974 (Jerne, 1974), is a conceptually different theory of how the components of the immune systems interact with each other and with the environment (antigens). This theory is rooted in the demonstration that animals can be stimulated to make antibodies capable of recognizing parts of antibody molecules produced by other animals of the same species or strain. According to Jerne (1974), this made reasonable the assumption that, within the immune system of one given individual, any antibody molecule could be recognized by a set of other antibody molecules (de Castro & Timmis, 2002).

The immune network learning algorithm proposed by de Castro and Von Zuben (2001) named aiNet (Artificial Immune NETwork), model will consist of a set of cells, named antibodies, interconnected by links with associated connection strengths. The aiNet antibodies are supposed to represent the network internal images of the pathogens (input patterns) contained in the environment to which it is exposed. The connections between the antibodies will determine their interrelations, providing a degree of similarity (in a given metric space) among them: the closer the antibodies, the more similar they are. Based upon a set of unlabeled patterns $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_M\}$, where each pattern (object, or sample) $\boldsymbol{x}_i$, $i = 1, \ldots, M$ is described by L variables (attributes or characteristics) (de Castro & Von Zuben, 2001). The aiNet learning algorithm can be summarized as follows:

1. *Initialization:* create an initial random population of network antibodies;
2. *Antigenic presentation:* for each antigenic pattern, do
   a. *Clonal selection and expansion:* for each network element, determine its affinity with the antigen presented. Select a number of high affinity elements and reproduce (clone) them proportionally to their affinity;
   b. *Affinity maturation:* mutate each clone inversely proportional to affinity. Re-select a number of highest affinity clones and place them into a clonal memory set;
   c. *Metadynamic:* eliminate all memory clones whose affinity with the antigen is less than pre-defined threshold;
   d. *Clonal interactions:* determine the network interactions (affinity) among all the elements of the clonal memory set;
   e. *Clonal suppression:* eliminate those memory clones whose affinity with each other is less than a pre-specified threshold;
   f. *Network construction:* incorporate the remaining clones of the clonal memory with all network antibodies;
3. *Network interactions:* determine the similarity between each pair of network antibodies;
4. *Network suppression:* eliminate all network antibodies whose affinity is less than a pre-specified threshold;
5. *Diversity:* introduce a number of new randomly generated antibodies into the network;
6. *Cycle:* repeat Steps 2 to 5 until a pre-specified number of iterations is reach (de Castro & Timmis, 2002).

aiNet, attempts to reduce redundancy by eliminating similar antibodies, based upon their degree of similarity (affinity) with other network antibodies. This has the effect of controlling the population size (de Castro & Timmis, 2002).

4. A collaborative filtering method based on artificial immune network

In this section, a new collaborative filtering method based on aiNet has been introduced in details. aiNet will consist of a set of cells, named antibodies, interconnected by links with associated connection strengths. It aims at building a memory set that recognizes and represents the data structural organization. This network will be constructed to answer questions like: (1) Is there a great amount of redundancy within the data set and, if there is, how can we reduce it? (2) Is there any group or subgroup intrinsic to the data? (3) How many groups are there within the dataset? (4) What is the structure or spatial distribution of these data (groups)? (de Castro & Von Zuben, 2001). All answers of these questions are useful for our purpose. So, aiNet was applied to the rating matrix. Then, the acquired new rating matrix was clustered using *k*-means for generating more quickly recommendations. Block diagram of the proposed work is given in Fig. 1.

Continue of this section, how the aiNet is applied to the collaborative filtering will be explained. Table 1 presents the mapping between components of the collaborative filtering and immune system. Here, antigens (Ag) represent the user of database and recognized by antibodies. The antibodies (Ab) has been randomly generated initially then they have maturated and the selected ones have constituted the memory $\text{Ab}_{\{\text{m}\}}$. Generally, the distance measurements are used such as Euclid, Hamilton etc for affinity computation but we have used the Pearson correlation instead of them as we want to find the similarities not differences between the users. A vector f_j holds the affinity values computed using Pearson correlation in this work.

The steps of the aiNet learning algorithm and their short explanations for collaborative filtering are given in Table 2. As shown from the table there are two kinds of suppression mechanism: clonal and network. These mechanisms have made attractive the aiNet for solving the scalability problem. The antibodies that could not recognize any antigen have been eliminated via clonal suppres-

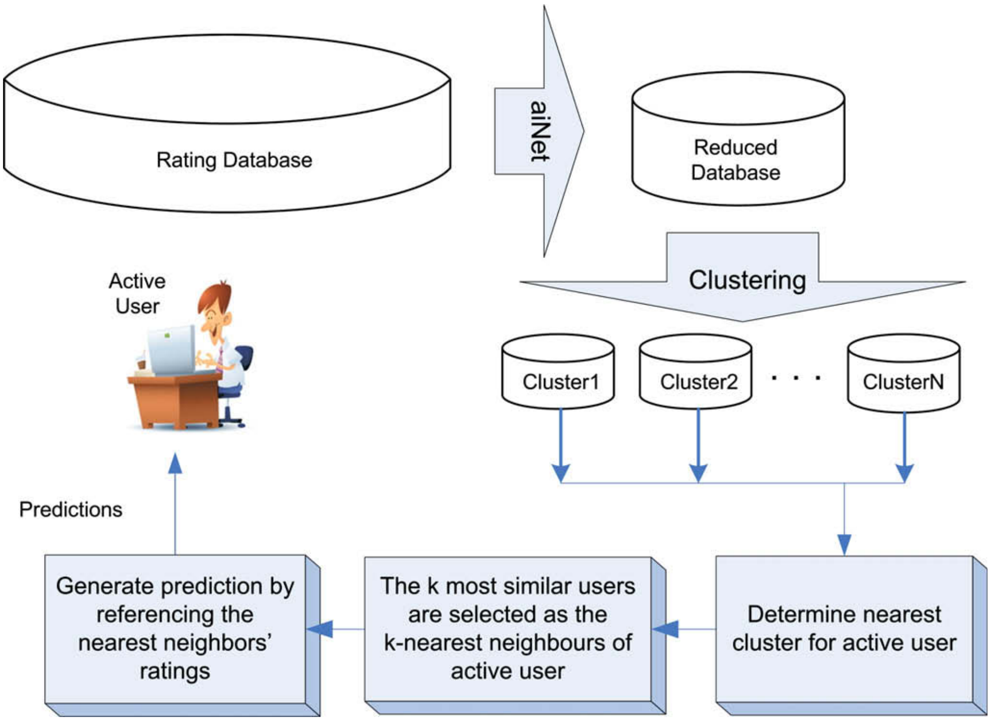

Rating Database
aiNet
Reduced Database
Clustering
Cluster1
Cluster2
. . .
ClusterN
Active User
Predictions
Generate prediction by referencing the nearest neighbors' ratings
The k most similar users are selected as the k-nearest neighbours of active user
Determine nearest cluster for active user

Fig. 1. Block diagram of the proposed work.

Table 1
Mapping between the components of the collaborative filtering and immune system.

Immune system	Collaborative filtering
Ag	Users of training set
Ab	Randomly generated users for recognized the user of training set
$Ab_{\{m\}}$	Users who constitute the memory matrix
f_j	A vector that holds the affinity values computed using Pearson correlation

Table 2
Detailed algorithm of the proposed method.

Pseudo code	Explanations
Input: $\mathbf{Ab}$,$\mathbf{Ag}$,gen,n,d,β,σ Output: $\mathbf{Ab}_{\{m\}}$	
for t = 1 to gen;	
for j = 1 to M;	// For each antigenic pattern $\mathbf{Ag}_j \in \mathbf{Ag}$, j = 1,.,M, do;
$\mathbf{f}$(j,:): = affinity($\mathbf{Ab},\mathbf{Ag}$(j,:));	// Determine affinity vector f using Pearson correlation
$\mathbf{Ab_n}$: = select($\mathbf{Ab},\mathbf{f}$(j,:),n);	// n highest affinity antibodies is selected. Ab_n is formed
$\mathbf{C}$(j,:): = clone($\mathbf{Ab_n}$,β,$\mathbf{f}$(j,:));	// β% of Ab_n cloned to proportionally to affinity values,acquired set C
$\mathbf{C}^*$(j,:): = hypermut($\mathbf{C},\mathbf{f}$(j,:));	// Each Ab of C mutated inversely proportional to their affinities
$\mathbf{f}^*$(j,:): = affinity($\mathbf{C}^*$(j,:),$\mathbf{Ag}$(j,:));	// Determine the affinity among $\mathbf{Ag}_j$ and C^*
$\mathbf{M_j}$: = select($\mathbf{C}^*$,$\mathbf{f}^*$(j,:),γ);	// From $\mathbf{C}^*$, re-select γ% of the antibodies with highest $\mathbf{f}^*$(j,:)
$\mathbf{M_j}$: = apoptosis($\mathbf{M_j},\mathbf{f}^*$(j,:),d);	// Eliminate memory clones from $\mathbf{M_j}$ whose affinity f^*(j,:) <d
$\mathbf{S}$: = affinityMemory($\mathbf{M_j}$, $\mathbf{M_j}$);	// Determine the affinity among the memory clones
$\mathbf{M_j}^*$: = suppression($\mathbf{M_j},\mathbf{S},\sigma$);	// *Clonal* suppression: eliminate memory clones from $\mathbf{M_j}$ whose S> σ
$\mathbf{Ab}^*$: = insert($\mathbf{Ab}^*$, $\mathbf{M_j}^*$);	// $\mathbf{Ab}^* \leftarrow [\mathbf{Ab}^*;M_j^*]$ for $\mathbf{Ag_j}$
end;	
$\mathbf{S}$: = affinityMemory($\mathbf{Ab}^*$, $\mathbf{Ab}^*$);	// Determine the affinity among all the memory antibodies from $\mathbf{Ab}^*$
$\mathbf{Ab}_{\{m\}}$: = suppression($\mathbf{Ab}^*$,$\mathbf{S}$,σ);	// *Network* suppression: eliminate all the antibodies from Ab^* whose S> σ
end;	

sion. The selected antibodies for each antigen have been put into memory matrix. On the other hand, similar antibodies may be existed in the memory matrix at the same time and the Ag recognized by the one of Ab has been also recognized by the other similar one but it is not necessary. In this instance, network suppression has eliminated the antibodies whose similarity among them is above the predefined suppression threshold value. The suppression threshold (σ) controls the specificity level of the antibodies, the clustering accuracy, and network plasticity. Using this internal memory obtained at the results of suppression processes has seen as a solution of scalability problem. It has been a positive affect on the performance as facilitating the determination and selection of neighbourhoods. Besides, the hypermutation mechanism is utilized as a solution to sparsity problem in our work. The sparsity has been decreased at each iteration via mutation. As the dataset has approximated to completeness, finding similar users has become easier. As a result, it can be said that aiNet is capable of reducing sparsity and providing the scalability of dataset at the same time.

5. Experimental evaluation

5.1. Dataset

In order to illustrate the performance of our proposal, we focused on the MovieLens dataset (<http://www.movielens.umn.edu>) which has been widely used as a benchmark problem for evaluating recently proposed approaches. It consists of 100.000 ratings, which were assigned by 943 users on 1682 movies. Users should have stated their opinions for at least 20 movies in order to be included. Ratings follow the 1(bad)–5(excellent) numerical scales. Starting from the initial data set, five data were generated (u1.base, u2.base, u3.base, u4.base, u5.base and u1.test, u2.test, u3.test, u4.test, u5.test). For each data split, 80%

of the original set was included in the training and 20% of it was included in the test data. The test sets in all cases were disjoint. Our experimental works are performed on these five distinct splits.

5.2. Experimental setup

To evaluate our approach, the steps showed in Fig. 2 were followed. As mentioned above, five data group were generated from the movie lens data set. For each data group split, 80% of the original set was included in the training and 20% of it was included in the test data. All experiments were realized on these five data groups separately and their averages were used as a result in the graphics.

In experiment 1, the aiNet algorithm was executed on the training set and memory matrix that called *aiNetDS* was acquired. *aiNetDS* was representing the internal image of the training set. Then, the CF algorithm was executed on the new reduced training set for twenty five users selected randomly from test set and was computed predictions for films which they have not voted. To make comparison, the CF algorithm was also executed directly on the training data set called *rawDS* for forming a benchmark.

Experiment 2 followed the same procedure as experiment 1. However, for investigating the effects of using clustering for forming the neighbourhoods to the system performance the *aiNetDS* was clustered by using *k*-means algorithm, obtained *clustered-aiNetDS,* and then these cluster partitions were used as neighbourhoods. Similar to experiment 1 for forming a benchmark, firstly the training data set was clustered by *k*-means that called *clusterded-rawDS* then CF applied to it.

All of them were implemented by Matlab 7.1 R14 and conducted on a PC with Intel Pentium M Processor 1.70 Ghz and 512MB RAM. They were compared in terms of prediction accuracy and the system performance. The accuracy of the predictions was evaluated by MAE and the system performance was assessed by the Response Time in both of these experiments. Also sparsity and compression rates are used to for observing the variations of sparsity and scalability of the datasets.

5.3. Experimental metrics

Mean absolute error (MAE) and response time metrics are used to evaluate the effectiveness of the proposed method in this study. The MAE is a statistical accuracy metric and evaluates our methodology in terms of quality. It measures the average absolute deviation between a predicted rating and the true rating. It is expressed as in the definition 3 where N is the number of all items, and p_i and r_i represent the predicted rating and the true rating, respectively:

$$MAE = \frac{\sum_{i=1}^{N} |p_i - r_i|}{N} \tag{3}$$

The other metric, response time used as performance evaluation metric in addition to the quality evaluation metric MAE. It defines the amount of time required to compute all the recommendations for the training set, employed to measure the system performance (Cho & Kim, 2004).

We also used two ratios for observing the variations of sparsity and scalability of the dataset: sparsity and compression rates. They are expressed in the definition 4 and 5, respectively. The lower values of all metrics used in this work accepted as satisfactory results.

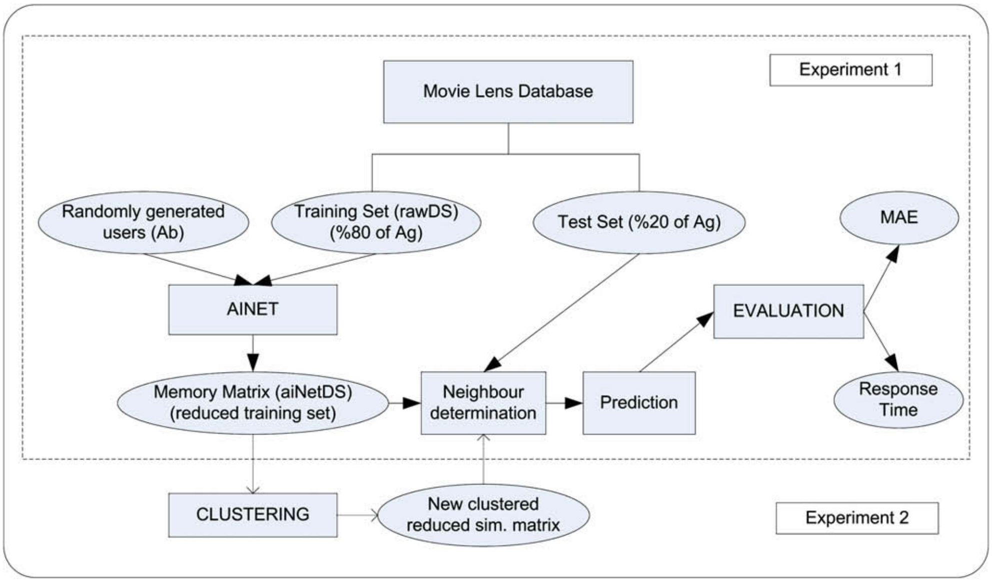
$$Sparsity\ Rate = 1 - \frac{Not\ Zero\ Ratings}{All\ Ratings} \tag{4}$$

$$Compression\ Rate = \frac{(All\ Users - Memory\ Users)}{All\ Users} \tag{5}$$

5.4. Experimental results

Firstly we want to investigate the results of the experiment 1 regarding to the sparsity, compression rates and dimensions of the training sets before and after applied aiNet. They calculated for each data group called split and the results are given in Table 3.

As shown in table, the sparsity rate has decreased from 94.96% to 13.13% as average on the training sets via the hyper mutation mechanism in the aiNet. Furthermore, the dimension of training set has decreased from 943 × 1681.4 to 278 × 1681.4 by the use of clonal and network suppressions. So the average compression rate has been calculated as 70.51% from these dimension values. While the comparison rate value has increased, the data set has become more scalable. To see how the alteration of these rates affected the performance and quality of the recommender system, we examine the graphics where the MAE and response time evaluation metrics are depicted.



Experiment 1
Movie Lens Database
Randomly generated users (Ab)
Training Set (rawDS) (%80 of Ag)
Test Set (%20 of Ag)
MAE
AINET
EVALUATION
Memory Matrix (aiNetDS) (reduced training set)
Neighbour determination
Prediction
Response Time
CLUSTERING
New clustered reduced sim. matrix
Experiment 2

Fig. 2. Description of the experimental process: aiNet (Exp. 1) & clustered aiNet based CF systems (Exp. 2).

The alteration of the average MAE and Response Time values of the distinct five training sets that applied aiNet (aiNetDS) and not applied aiNet (rawDS) according to the neighbourhood number n are shown in Fig. 3. While the n has increased, MAE has decreased for both of them but aiNetDS has given more accurate results than the rawDS as its sparsity was less. The similar thought was valid for the response time metric. When the n has increased, the response time has increased too, for both of them. But as seen from Fig. 3 aiNetDS has produced predictions more quickly then the rawDS. Because the number of users where the search for forming the neighbours has reduced from 943 to 278 as averagely.

In experiment 2, we want to explore the effects of using clustering for forming the neighbourhoods to the system performance. So the *aiNetDS* was clustered by using k-means algorithm, obtained *clustered-aiNetDS,* and then these cluster partitions were used as neighbourhoods. Similarly, the rawDS was clustered by k-means too and acquired *clustered-rawDS* then CF applied to it. Before giving the results of the experiment 2, the k-means algorithm was summarized.

Table 3
The sparsity & compression rates and dimensions of the training sets before and after applied aiNet.

	Sparsity rate		Dimensions of training set		Compression rate
	Not applied aiNet	Applied aiNet	Not applied aiNet	Applied aiNet	
Split-1	94.96%	13.82%	943 × 1682	263 × 1682	72.11%
Split-2	94.96%	12.95%	943 × 1682	280 × 1682	70.31%
Split-3	94.96%	12.87%	943 × 1682	280 × 1682	70.31%
Split-4	94.96%	12.95%	943 × 1682	291 × 1682	69.14%
Split-5	94.95%	13.04%	943 × 1679	276 × 1679	70.73%
Average	94.96%	13.13%	943 × 1681.4	278 × 1681.4	70.51%

k-means clustering (Han & Kamber, 2001) is a method commonly used to partition a set of data into groups. This scheme proceeds by selecting m initial cluster centres and then iteratively refining them. (1) Each instance d_i is assigned to its closest cluster centre; (2) each cluster centre C_j is updated to the mean of its constituent instances. The algorithm has converged when the assignment of instances to clusters no longer changes.

For analyzing the effects of the k parameter of the k-means algorithm on the clustering and comprehending its relationship with number n, we explore the graphics depicted in Fig. 4. It shows the alteration of the average MAE values of the distinct five training sets that (a) not applied aiNet (rawDS) and (b) applied aiNet (aiNetDS) according to k parameter and the neighbourhood number n.

When the plots showed in Fig. 4 are compared, it has seen that the MAE has changed in the higher error interval for rawDS than the aiNetDS. So we can say that, the aiNetDS has produced more accurate results than the rawDS. Also as seen form the figure, the best results of the MAE has obtained for the higher value of n and lower value of k but the actual parameter that effect the MAE is n for the both of graphics. Because the similar MAE values has been acquired for the different values of the parameter k at the

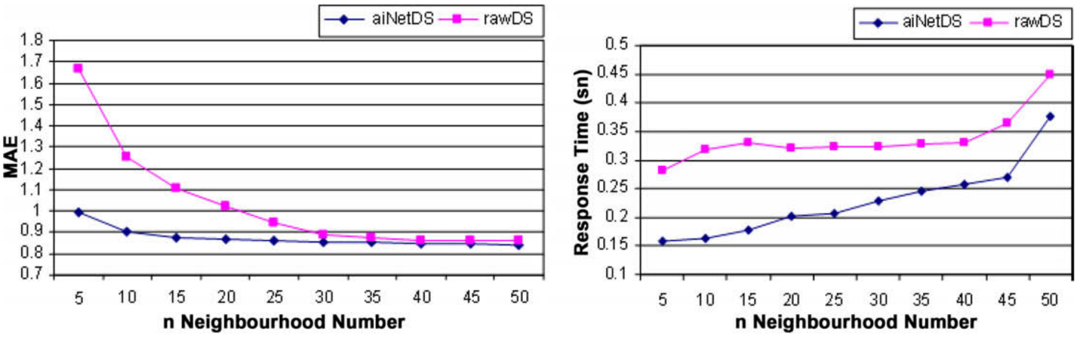


Fig. 3. Alteration of the average MAE (left plot) and response time (right plot) values of the distinct five training sets that applied aiNet (aiNetDS) and not applied aiNet (rawDS) according to the neighbourhood number n.

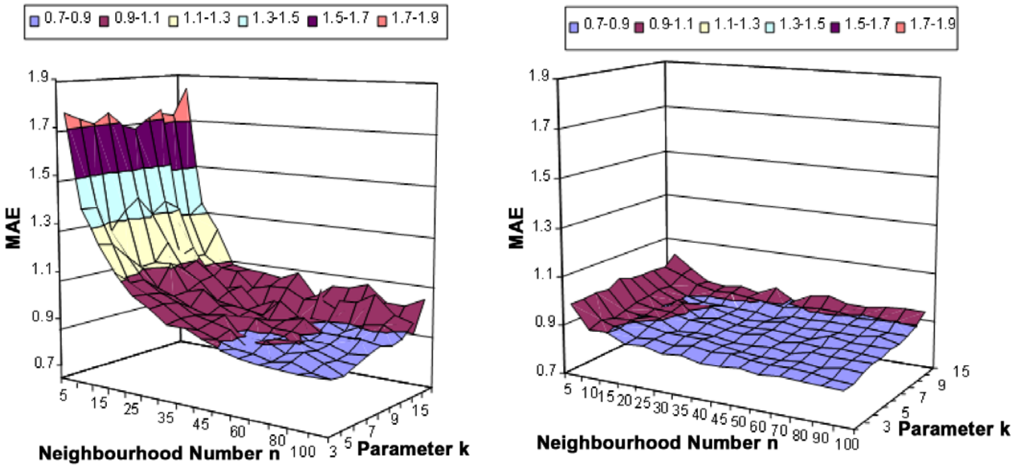


Fig. 4. The Alteration of the average MAE values of the distinct five training sets that not applied aiNet (rawDS) (left plot) and applied aiNet (aiNetDS) (right plot) according to k parameter and the neighbourhood number n.

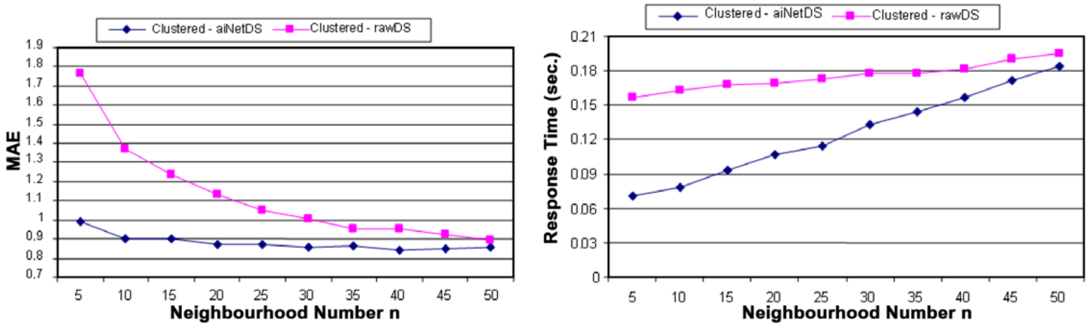


Fig. 5. The Alteration of the average MAE (left plot) and response time (right plot) values of the clustered aiNetDS and clustered rawDS according to the neighbourhood number n while $k = 3$.

same numbers of n. Consequently, we took the parameter k value as 3 while surveying the changing of the average MAE according to neighbourhood number n for the following analysis.

Fig. 5 shows the alteration of the of the average MAE and response time values of the distinct five training sets which are clustered aiNetDS and clustered rawDS according to the neighbourhood number n while $k = 3$. For both of them, while the neighbourhood number n has increased the response time has increased too, contrarily MAE has decreased. However the averages of the aiNetDS's MAE are always lower than the rawDs's. As for the response time, it has seen that the aiNetDS produced more rapid predictions.

The alterations of the average time passing for the clustering of aiNetDS and rawDS with k-means according to parameter k have also investigated and the results have given in Fig. 6. While the parameter k value are enhancing, the clustering process are getting more longer time for both datasets. But always aiNetDS needs rather short time than rawDS.

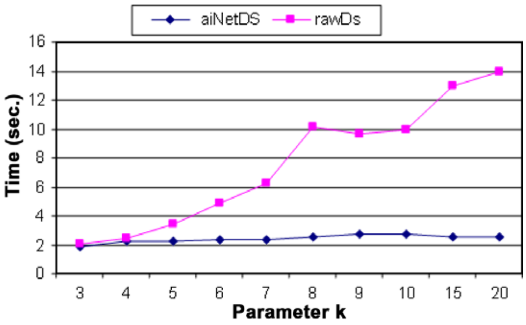


Fig. 6. The alterations of the average time passing for the clustering of aiNetDS and rawDS with k-means according to parameter k.

For now the datasets of aiNetDS and rawDS have been compared and seen that aiNetDS has given always more better results regarding to average MAE and response time metrics. Therefore continue of our experiments, we are comparing results of obtained from aiNetDS and clustered-aiNetDS. These results are depicted in Fig. 7.

As seen from the figure, the average MAE value has decreased when the n has increased and for the both of them the MAE values are closed to each other. But when we look at to the response time, it is realized that the clustered-aiNetDS's performance is better than the other. As a result we can say that, clustering affected the system performance as positive.

6. Conclusion

Data overload has increasingly become a significant issue in the use of information systems. Systems that assist with that issue, such as collaborative filtering-based recommender systems, have been successfully used in a number of applications, but suffer several limitations such as data sparsity and scalability emerging from the high dimension of dataset. In this study, we tackle these two questions: (1) how the data sparsity can be reduced? (2) How to make recommendation algorithms more scalable? We present an approach to addressing the both of these problems at the same time by using a new method. Our proposed method can be described as a filtering algorithm which utilizes the Artificial Immune Network Algorithm (aiNet). It is chosen because aiNet is capable of reducing sparsity and providing the scalability of dataset via describing data structure, including their spatial distribution and cluster inter-relations.

According to experimental results, the sparsity rate has decreased via the hyper mutation mechanism and the dimension of training set has decreased too by the use of clonal and network suppressions in the aiNet. Because of providing 70.51% data reduc-

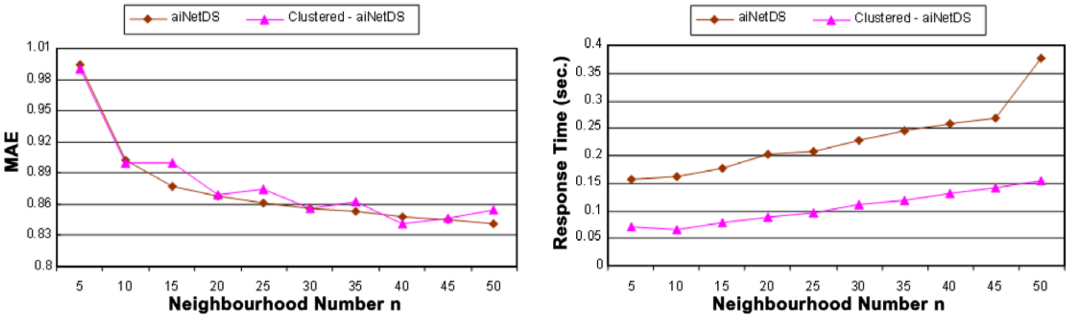


Fig. 7. The changes of the average MAE (left plot) and Response Time (right plot) values of the aiNetDS and clustered-aiNetDS according to the neighbourhood number n while $k = 3$.

tion, we can produce high quality predictions more quickly. Also we want to explore the effects of using clustering for forming the neighbourhoods to the system performance. For this, the datasets that not applied aiNet (rawDS) and applied aiNet (aiNetDS) are clustered by using *k*-means algorithm and then these cluster partitions are used as neighbourhoods. The clustered datasets of aiNetDS and rawDS have been compared and seen that aiNetDS has given always more better results regarding to average MAE and response time metric. Then we are comparing results of obtained from aiNetDS and clustered-aiNetDS and seen that their MAE values are closed to each other. But when we look at to the response time, it is realized that the clustered-aiNetDS's performance is better than the other. As a result we can say that, clustering affects the system performance as positive. As the conclusion, the quality of predictions and system performance are improved and the advantage of clustered aiNet based CF system is showed at the end of empirical studies. Our future work will include investigation of applying the proposed method to more real-world datasets and using different similarity measures.

Acknowledgements

The authors acknowledge the support of this study provided by Selçuk University Scientific Research Projects and would like to thank the GroupLens Research Project Team from University of Minnesota for making the MovieLens dataset (http://www.movielens.umn.edu) available for researchers. The authors have also thanked to TUBITAK for their support of this study.

References

Ahn, H. J. (2008). A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. *Information Sciences, 178*, 37–51.

Aickelin, U., & Chen, Q. (2004). On affinity measures for artificial immune system movie recommenders. In *Proceedings of the 5th International Conference on Recent Advances in Soft Computing*, Nottingham, UK.

Billsus, D., & Pazzini, M. J. (1998). Learning collaborative information filters. In *Proceeding of 15th international conference on machine learning*, Morgan Kaufmann Publishers Inc. (pp. 46–54, 93).

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

Cayzer, S., & Aickelin, U. (2005). A recommender system based on idiotypic artificial immune networks. *Journal of Mathematical Modelling and Algorithms, 4*(2), 181–198.

Cheung, K., Tsui, K., & Liu, J. (2004). Extended latent class models for collaborative filtering. *IEEE Transactions on Systems Man and Cybernetics – Part A: Systems and Humans, 34*(1), 143–148.

Cho, Y. B., Cho, Y. H., & Kim, S. H. (2005). Mining changes in customer buying behaviour for collaborative recommendations. *Expert System with Applications, 28*, 359–369.

Cho, Y. H., & Kim, J. K. (2004). Application of web usage mining and product taxonomy to collaborative recommendations in e-commerce. *Expert System with Applications, 26*, 233–246.

Claypool, M., Gokhale, A., & Miranda, T. (1999). Combining content-based and collaborative filters in an online newspaper. In *Proceedings of the SIGIR-99 workshop on recommender systems: Algorithms and evaluation*.

de Castro, L. N., & Timmis, J. (2002). *Artificial immune systems: A new computational intelligence approach*. London: Springer-Verlag. 357 p.

de Castro, L. N., & Von Zuben, F. J. (2001). aiNet: An artificial immune network for data analysis. In Hussein A. Abbass, Ruhul A. Sarker, & Charles S. Newton (Eds.), *Data mining: a heuristic approach* (pp. 231–259). USA: Idea Group Publishing [Chapter III].

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

Goldberg, K., Roeder, T., Gupta, D., & Perkins, C. (2000). Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval, 4*(2), 133–151.

Han, J., & Kamber, M. (2001). *Data mining: Concepts and techniques*. Morgan Kaufman Publishers. 550 p.

Herlocker, J., Konstan, J. A., & Riedl, J. (2000). Explaining collaborative filtering recommendations. In *ACM 2000 conference on computer-supported collaborative work* (pp. 241–250). Philadelphia, PA.

Huang, Z., Chen, H., & Zeng, D. (2004). Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Transactions on Information Systems, 22*(1), 116–142.

Jerne, N. K. (1974). Clonal selection in a lymphocyte network. In G. M. Edelman (Ed.), *Cellular selection and regulation in the immune response* (pp. 39). NY: Raven Press.

Li, Q., & Kim, M. K. (2003). Clustering approach for hybrid recommender system, In *Proceedings of the IEEE/WIC international conference on web intelligence (WI'03).*

Li, Y., Lu, L., & Xuefeng, L. (2005). A hybrid collaborative filtering method for multiple-interests and multiple-content recommendation in E-Commerce. *Expert System with Applications, 28*, 67–77.

Li, Q., Myaeng, S. H., & Kim, M. K. (2007). A probabilistic music recommender considering user opinions and audio features. *Information Processing and Management, 43*, 473–487.

Liu, D., Ke, C., Lee, J., & Lee, C. (2008). Knowledge maps for composite e-services: A mining-based system platform coupling with recommendations. *Expert Systems with Applications, 34*, 700–716.

Liu, D., & Shih, Y. (2005). Integrating AHP and data mining for product recommendation based on customer lifetime value. *Information & Management, 42*, 387–400.

Mihaljevic, B., Cvitas, A., & Zagar, M. (2006). Recommender system model based on artificial immune system. In *28th international conference on information technology interfaces (ITI 2006)* (pp. 367–372). Cavtat, Croatia.

Morrison, T. (2003). Similarity measure building for website recommendation within an artificial immune system. Ph.D. Thesis, University of Nottingham.

Pazzini, M. J. (1999). A framework for collaborative, content based and demografic filtering. *Artificial Intelligence Review, 13*(5–6), 393–408.

Pennock, D. M., Horvitz, E., Lawrence, S., & Lee Giles, C. (2000). Collaborative filtering by personality diagnosis: a hybrid memory- and model-based approach. In *Proceedings of the 16 conference on uncertainty in artificial intelligence* (pp. 473–480).

Resnick P., Lacovou N., Suchak M., Bergstrom P., & Riedl J. (1994). GroupLens: An open architecture for collaborative filtering of netnews. In *Proceedings of CSCW94* (pp. 175–186).

Sakiroglu, A. M. (2005). A recommender system based on artificial immune for web sites. Master thesis, University of Selcuk.

Sarwar, B. (2001). Sparsity, scalability, and distribution in recommender systems. PhD thesis, University of Minnesota.

Sarwar, B. M., Karypis, G., Kontsan, J. A. & Riedl, J. T. (2000). Application of dimensionality reduction in recommender system – A case study. In ACM WebKDD 2000 web mining for e-commerce workshop (pp. 82–90). ACM Press.

Schafer, J. B., Konstan, J. A., & Riedl, J. (2001). *E-Commerce recommendation applications*. University of Minnesota.

Sobecki, J., & Szczepański, L. (2007). Wiki-news interface agent based on AIS methods. *Lecture Notes in Computer Science, 4496/2007*, 258–266.