

DEVELOPING A FRAMEWORK FOR INSURANCE UNDERWRITING EXPERT SYSTEM

Shlomo Berkovsky¹, Yaniv Eytani¹ Edward Furman²

{slavax, ieytani}@cs.haifa.ac.il efurman@stat.haifa.ac.il

¹University of Haifa, Computer Science Department, 31905, Haifa, Israel

²University of Haifa, Statistics Department, 31905, Haifa, Israel

Key words: Expert Systems, Recommendation Systems, Insurance Underwriting

ABSTRACT

The process of insurance underwriting determines whether insurance company will accept an application for insurance. It is a complex decision-making process and due to information overload problems, it becomes harder for the insurance companies to efficiently estimate risks in a reasonable time.

In this paper we present a framework of an expert system for insurance premium assessment. It combines various Artificial Intelligence techniques, both supervised and unsupervised learning. The proposed framework certainly does not pretend to replace a human underwriter by an electronic one. Rather, it aims at producing viable estimations regarding the clients risk levels, allowing to speed-up the underwriting process and to utilize new potentials.

1. INTRODUCTION

A critical skill for success in insurance risk management is underwriting [1]. The term underwriting is interpreted as the process by which an insurance company determines whether or not and on what basis it will accept an application for insurance. Underwriting is a very complex decision-making process, which involves the manipulation of many types of information on multiple levels. The actuarial design of the product, abnormalities that increase the likelihood of loss and the quality of the available data must be weighted and considered to resolve underwriting problems. This must be done in the context of a competitive market environment while balancing the expected claims with the expected investment income from the use of the pooled funds.

In the past, the expense of capturing underwriting data outweighed the perceived benefit, and companies could define products with simple actuarial design and minimal underwriting and do well. Although this policy becomes impossible in today's competitive, investment-oriented marketplace, in many insurance companies underwriting process still remains simple and manually-managed procedure.

Nowadays, insurance companies collect and share more and more personal data. It becomes harder and harder for them to efficiently estimate risk assessment in a reasonable time due to the information overload problem [2, 3]. Thus, overcoming the information overload is important issue, since by doing so insurance companies will be able to provide better insurance premium estimation to the customers in a faster and easier manner. It will hold down expenses and increase the profits of the insurance companies.

Luckily, the cost of hardware and software has fallen to the point that enables most of the insurance companies to move towards the electronic capture and manipulation of underwriting data. Ways to enhance the underwriting process have evolved with the use of new technologies, especially information technology. The tools of the later will play a much more important role in insurance and risk assessment as the amount and complexity of available information increases.

An expert system is a computer system which emulates the decision-making ability of a human expert [4]. In simple words, such systems contain knowledge derived from an expert in some narrow domain. Expert systems are meant to solve real problems which normally would require a specialised human expert (in this case an actuarial specialist). The narrow domain is mentioned since it is difficult to encode enough knowledge into a system so that it may solve a variety of problems.

In recent years, a new category of expert systems, named Recommendation Systems (RS) has emerged [4, 5, 6, 7, 8]. RS are a powerful new technology for extracting additional value for a business from its customer databases. In the general case, these systems help customers find products they want to buy from a business. RS benefit customers by enabling them to find products they like. Conversely, they help the business by generating more sales. Widespread implementations of RS are based on Collaborative Filtering (CF) [9, 10]. CF addresses the information overload problem by incorporating the opinions of human users into an information filtering system. It takes into account human judgments only, and based on those judgments it groups people of similar opinions to create a "virtual community" of users.

In the case of actuarial expert system a "virtual community" of costumers is formed basing on the assumption that similarity implies similar risk level. The system will group the users according to a dynamically learned similarity measure. Our novel approach assigns each of the underwriting properties a different weight basing on the relative importance of the property. Finally, the system suggests an insurance premium similar to the premium values of the users in a close "virtual community" of the given user.

Note that the proposed framework certainly does not pretend to replace a human underwriter by an electronic one. Rather, it offers wise utilization of the existing expert systems and artificial intelligence algorithms, and introduces new potentials unrealized till now. Our intent is the development of an expert system that brings the data investigation process to levels human beings can not reach. Such a system will significantly speed-up the underwriting process, improve its accuracy, and hold down the expenses of the insurance companies.

The rest of the paper is organized as follows, in section 2 we discuss the information overload problems. In section 3 we describe a framework for creating an expert system. In section 4 we describe various machine learning techniques to facilitate the describe architecture. In section 5 we give a detailed description of an expert system for insurance underwriting. In section 6 we give a detailed user scenario. Finally we conclude and plan our future research.

2. Background

Overcoming the information overload and achieving better future prediction of an arbitrary policyholder risk measure is important to insurance companies. This way they are able to enhance their services, to fit better their offerings to a specific client and thus to increase the possibilities for higher profits.

Many existing information systems deal with customers, searching for products. Most of the recommendation systems today can be defined as systems that recommend a customer items they "believe" the customer would be interested in. Although our recommendation system does not fit in such a definition, the main methodological tools used for its formation are similar to those utilized in the classical RS context.

RS usually base on algorithms from two adjacent fields: information retrieval and collaborative filtering. Good et al. [11] state three technologies that are commonly used to address information overload challenge. These technologies are:

- Information Retrieval: focusing on tasks involving the fulfillment of transient interest queries;
- Information Filtering: classifying streams of new context into categories;
- Collaborative Filtering: answering two questions: "Which item should I view?", and "How much will I like the item?"

Information retrieval [12, 13, 14] focuses on an analysis of an item's content and the development of a personal customer interest profile. The advantage of information retrieval is that it does not depend on having other customers in the system, let alone others with similar tastes.

Information filtering systems [15, 16, 17] sort through large volumes of dynamically generated information and present to

the user those which are likely to satisfy his or her information requirement. If one considers information retrieval from a very general "information selection" viewpoint, information filtering is a special case in which the information space is very dynamic.

Collaborative filtering [18, 19, 20] is mostly used in recommendation systems, and not as a search technique par-excellence. The idea behind collaborative filtering is recommending a customer what to buy by looking at her neighbors. Mainly, neighbors of a customer are (top-N) customers that have the greatest similarity to the customer. This similarity can be history-wise (customers with similar history facts, i.e. past claims, medical history etc.) or profile-wise (customers with similar properties, i.e. age, marital status, demographic details etc.).

As opposed to information retrieval, for collaborative filtering to be effective, it requires that several customers evaluate each item; even then, new items cannot be recommended until some customers have taken the time to evaluate them. These problems are better known as the sparsity and first-rater problem.

We developed a novel approach for an expert system for the insurance underwriting process based on risk assessment of the costumers. The detailed architecture and design of the system will be elaborated in the following sections.

3. Expert system architecture

We briefly describe the main approaches dealing with the study of how to create self-organizing applications that learn from past experience and provide a prediction about future. In our case the procedure of estimating a client risk level can be described as a two-level process.

At the first level, called the Learning Stage, the system statically partitions the available (training) data about past clients and measures the relative influence (weight) of each of the properties in the costumers' profiles. The Learning Stage consists of two sub-stages: clustering of similar clients and determining the weights of profile properties.

- **Clustering** [21] is an algorithm that takes a data set of inputs and divides them into equivalence classes, so that every input in a class is "similar" in some way. Thus, a

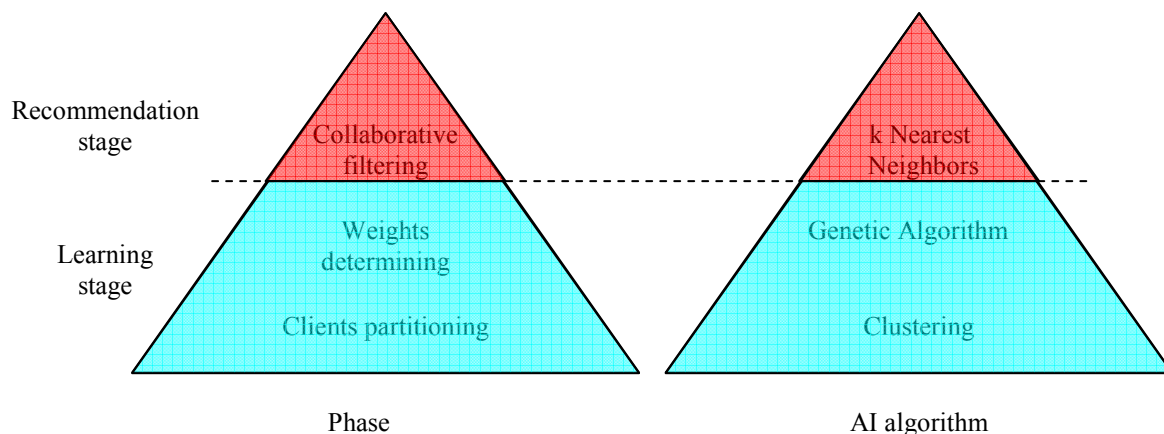


Figure1: System architecture

cluster is a collection of objects which are “similar” between them and are “dissimilar” to the objects belonging to other clusters.

- **Case-based reasoning (CBR)** [22] focuses on an analysis of an item’s content and the development of a personal user interest profile. The advantage of this method is that it does not depend on having other users in the system since each user is treated individually.

At the second level, called the Recommendation Stage, the system generates a recommendation using the collaborative filtering mechanism that was discussed earlier.

- **Collaborative filtering with k -Nearest Neighbors (kNN)**. Classical recommendation systems are systems that give recommendations, correlated with his/her personal profile to a customer. Those systems usually employ algorithms based on collaborative filtering. The idea behind collaborative filtering is to give a recommendation based on the data known about customer’s “neighbors”. Mainly, neighbors of a customer are “top- k ” other customers having the greatest similarity to the one. These k neighbors can be identified using the k -Nearest Neighbors algorithm [23, 24]. This similarity can be both history-wise (customers with similar history facts, i.e., past claims, medical history etc...), and profile-wise (customers with similar properties, i.e. age, marital status, demographic details etc...).

This architecture can be depicted as a pyramid structure, where each stage bases on the results of the previous stage. Finding the nearest neighbors in the Recommendation Stage uses properties weights that were obtained in the Learning Stage. Obtaining consistent results by the GA requires it to be operated on homogeneous groups acquired from the clustering process.

4. Machine Learning (ML) approaches

Machine learning studies the design of computer systems able to induce patterns, regularities, or rules from past experiences. Learner (a computer system) processes data representing past experiences and tries to either develop an appropriate response to future data, or describe the data seen in some meaningful way.

For example, in medical domain a learner deals with a set of patient cases (patient records) with corresponding diagnoses. It can either try to predict the presence of a disease for future patients, or to characterize the dependencies between diseases and symptoms. There are three different types of learning (see [25, 26, 27]):

- Unsupervised learning (understanding the relationships between data components).
- Supervised learning (learning a mapping between an input v and a desired output y).
- Reinforcement learning (learning to act in the environment based on the delayed rewards).

In this section we will describe machine-learning techniques of these three types. In the next section we will discuss how to

create an underwriting expert system by modularly combining them together.

4.1. Unsupervised clustering with K-means

At the initial stage of the learning process the system partitions the customers’ pool to clusters of similar customers according to their risk levels. We use this measure as this is the primary assessment of customers’ similarity. Clustering can be viewed as the process of organizing objects into groups whose members are similar in some way.

Clustering uses a similarity criterion based on distance. Two or more objects belong to the same cluster if they are “close” according to a given distance (in this case risk level). This is called distance-based clustering. Figure 2 shows an example of employing a clustering algorithm according to the geometrical distance criteria on a set of two-dimensional points

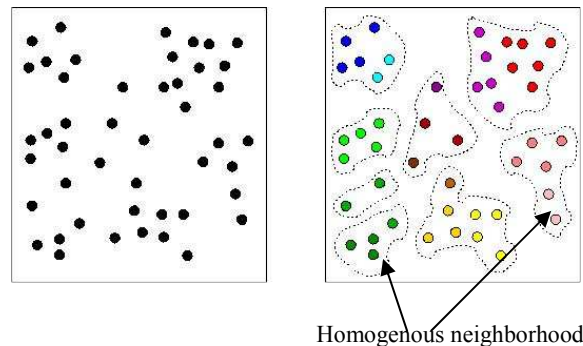


Figure 2: Homogenous neighborhood formation with clustering

One of the most widespread unsupervised learning algorithms that solve the clustering problem is K-means [28]. Consider the pseudo-code of K-means algorithms:

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2. Assign each object to the group that has the closest centroid.
3. When all objects have been assigned, recalculate the positions of the K centroids.
4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

The algorithm classifies a given data set through a certain number of clusters (assume k) fixed a priori. The main idea is to define k centroids, one for each cluster. The next steps iteratively recalculate k new centroids as the centers of the clusters resulting from the previous step. This loop changes the locations of k centroids, until no more changes are done.

This algorithm aims at minimizing an objective function (in this case a squared error function):

$$J = \sum_{j=1}^k \sum_{i=1}^n \|x_i^j - c_j\|^2,$$

where $\|x_i^j - c_j\|^2$ is a chosen distance measure between a data point x_i^j and the cluster center c_j , is an indicator of the distance of the n data points from their respective cluster centers

4.2. Genetic case-based reasoning

We would like to estimate the influence of the different underwriting properties on the insurance premium. In order to do so each property is assigned to be of a different weight. These weights will be used in the recommendation stage when calculating an estimation of the insurance premium. A genetic algorithm is used to determine an optimal set of property weights for each of the clusters found by the clustering algorithm.

Genetic algorithms (GAs) [29, 30] search for optimal solution by sampling the search space at random and creating a set of individuals representing possible solutions (called chromosomes). Each of these solution candidates has an assigned fitness value. These solutions undergo either recombination, mutation or survive and are chosen with a probability that depends on their fitness level to evolve into a new generation of solutions.

Recombination provides a mechanism for mixing genetic material within the population. Mutations introduce new genetic material thereby preventing the search from stagnating. The next population of solutions is chosen from the parent and the offspring generations in accordance with a survival strategy that favors fit individual but does not preclude the survival of the less fit.

Consider the following pseudo-code of typical genetic algorithm:

1. Initialize a random population of individuals
2. Evaluate fitness of all initial individuals of population
3. While true
 - 3.1. Select a sub-population for offspring production
 - 3.2. Recombine the "genes" of selected parents
 - 3.3. Perturb the mated population stochastically
 - 3.4. Evaluate its new fitness
 - 3.5. Select the survivors from actual fitness
4. Exit on termination criterion

The process is well described in Figure 3. In principle, the application provides an unlimited number of sample patterns $[v,y]$. Measurement vector v goes into estimating function $d(v)$, which generates d . An evaluating system compares d and y . Based on the remaining error, parameters of estimating function are modified until no further improvements can be achieved.

4.3. Collaborative filtering with kNN

As we already mentioned, one of the approaches widely used in recommendation systems is collaborative filtering. When using collaborative filtering, a mechanism to generate the recommendation is needed. The algorithm of nearest neighbors finding is agreed to be one of the most effective mechanisms for recommendations generation based on a set of customer profiles [31, 32].

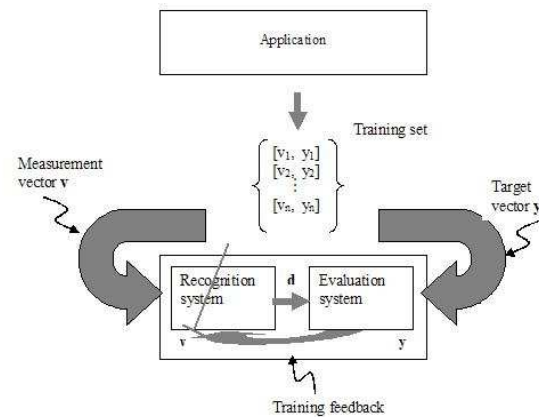


Figure 3. Learning from examples

Applying proximity (similarity) measures between two customers profiles constitutes the neighborhood. In current RS nearest neighborhood is formed by various similarity techniques [33, 34, 35], such as the Pearson correlation, cosine similarity, or mean squared differences. In this research we propose to apply the last one.

The mean squared difference is a measure that evaluates the degree of dissimilarity between two customers. For every element in the customers profile vectors we take the square of the difference value. The average of these values is then the mean squared difference.

Let us denote P_{x_n} to be customer x_n 's profile vector consisting of $|P_{x_n}|$ components, P_{x_m} and $|P_{x_m}|$ by analogy profile vector and number of components for customer x_m . Then the mean squared differences between customer x_n 's profile P_{x_n} and customer x_m 's profile P_{x_m} is given by the following:

$$D_{x_n x_m} = \frac{(P_{x_n} P_{x_m})^2}{|P_{x_n}|} = \frac{\sum_{j=1}^{|P_{x_n}|} (P_{x_n,j} - P_{x_m,j})^2}{|P_{x_n}|}$$

The lower the mean squared difference, the greater the similarity.

Another issue that arises after computing the similarity between the customers is how to form the neighborhood. There are two schemes for doing so:

- **Center-based neighborhood** [7]. Forming a neighborhood of size one for a given customer x_n , by simply selecting the nearest (the most similar) neighbors, or choosing all the neighbors, whose similarity value is higher than a certain threshold.
- **Aggregate neighborhood** [20]. Forming a neighborhood of size one, for a customer x_n , by first picking the closest neighbor to x_n , and then looking at the point where there are j neighbors in the neighborhood N . The algorithm computes the centroid of the neighborhood N . This centroid is defined as:

$$C = \frac{1}{j} \sum_{v \in N} v$$

where v represents the profile vector for each customer in the neighborhood. A customer N_i , such that $N_i \in N$, is selected as the $j+1$ st neighbor only if N_i is the closest

one to the centroid. In the next iteration the centroid is recomputed for $j+1$ neighbors and the process continues until $|N|=1$.

5. Underwriting system description

Consider an insurance company with n customers, we denote $P_x=(P_{x1}, P_{x2}, \dots, P_{xm})^T$ to be the profile matrix of all customers (every element of P_x is a vector of c various components such as age, education level and history-based, such as number of claims, claims amounts etc.). In addition to the given matrix P_x , we denote a set Y to be a set of all possible values customers risk measure can take. We call some categorical vector $y=(y_1, y_2, \dots, y_n)^T$ to be the target vector and its general element y_i , s.t. $1 \leq i \leq n$ signifies the risk measure of customer i , certainly $y_i \in Y$, for each i . In other words, we may refer to the pair (P_x, y) as all known data before the beginning of the learning process and we call this pair a pattern.

Our architecture describes an RS that will determine risk measure $y_{n+1} \in Y$ for any arbitrary future customer x_{n+1} . To implement the system we use the following means:

1. Case-based reasoning algorithm. This method deduces the risk measure of a customer from his/her own-history and the profile of the one. We use the pattern (P_x, y) to determine the vector of weights $\kappa=(\kappa_1, \kappa_2, \dots, \kappa_c)^T$. This vector implies the relative contribution of each of components (age, level of education, number of claims and more...) to the total customer's risk measure y_i and it is valid for any customer x_i . Consequently, the function $d(\kappa^T \cdot P_{x_{n+1}})$ gives us the risk measure y_{n+1}^j for a new customer x_{n+1} . The real world pattern (P_x, y) may be considered as a training set for the algorithm. Let us notice, that we may further improve our predictions by dividing the profile matrix of all customers P_x into m partitions $P_{x1}, P_{x2}, \dots, P_{xm}$, using any well-known clustering method. In this case the pair (P_{xj}, y) , s.t. $1 \leq j \leq m$ provides the necessary training set for evaluating the weights vector κ_j per each partition of P_x . Hence, a future customer's x_{n+1} risk measure y_{n+1}^j is defined this time by the function $d(\kappa_j^T \cdot P_{x_{n+1}j})$ dependent on the partition, that specific customer x_{n+1} belongs to.

2. Neighborhood by history and profile. Here, we use the history of a customer to find other customers that are similar to him, history-wise and preference-profile of the customer to find other customers that are similar to him, profile-wise. Then we apply their history and profile weights to deduce the risk measure. In our project we propose to use so called weighted distance measures. Certainly, we may easily generalize, for instance, the results of distance computation by adding a supplementary parameter κ_j , implying the contribution of any component j . Then, the new mean squared difference $D'x_n x_m$ is as follows:

$$D'x_n x_m = \sum_{j=1}^c k_j (P_{x_{nj}} - P_{x_{mj}})^2$$

The vector of the coefficients $\kappa=(\kappa_1, \kappa_2, \dots, \kappa_c)^T$ is evaluated by means of the CBR algorithms. In the case when $\kappa_j=1/c$ one returns to the original distance computation. Obviously, the sum

$$\sum_{j=1}^c k_j = 1$$

and $\kappa_j \geq 0$. The risk measure y_{n+1}^j of any future customer x_{n+1} is then easily determined, by means of a resemblance principle. The usage of neighborhoods is not new, and is commonly applied in many recommendation systems ([6],[9],[13],[19] and [23]). Similarly to those systems, our system also employs the technique of neighborhood formation. However, unlike most of the systems available today, we try to combine two different neighborhood formations: one is based on the history-profile of the customers and the other is based on the preference-profile of the customers. We try to evaluate both formations and infer if such a complex approach helps to provide more accurate customer's risk measure estimation.

3. A hybrid of CBR and Neighborhood by History and Profile. We propose to combine both of the above methods in order to implement a complex recommendation system, whose output will provide more accurate estimation of arbitrary customer risk measure. Applying case-based reasoning algorithms, we propose to derive a credibility factor α , such that the final recommendation is

$$y_{n+1} = \alpha y_{n+1}^1 + (1 - \alpha) y_{n+1}^2$$

6. Example scenario

In this section we will describe the typical scenario of risk assessment process. It is based on the architecture discussed in the previous sections. As discussed before, the process of estimating a client risk level is done as a two level process containing both a learning stage a recommendation stage.

In learning stage the systems partitions past customers by clustering them according to their risk level. We use this measure first, as this is our primary assessment of customers' similarity. This is a rough form of clustering as clusters are formed by a heterogeneous group of costumers. We would like to estimate the risk levels basing on homogenous groups. We achieve this by further clustering the elements of each "risk cluster" basing on similar profiles.

Using a genetic algorithm and real-life risk levels (determined by the insurance company) allows us to estimate the influence of each property in the insurance premium calculation by giving them different weights. Note, that the set of properties is a closed set limited by information given by the underwriting company. We associated determine these for each cluster separately as they different homogeneous groups.

In recommendation stage the systems estimates the costumer risk level. The system first identify to which cluster this customer belongs by measuring the distance between the user profiles to the center of each of the cluster center. As each cluster represents a highly homogeneous

7. Conclusions and further research

Overcoming the information overload and achieving better future prediction of an arbitrary policyholder risk measure is important to insurance companies. A framework for an expert

system that emulates the decision-making ability of a human expert was developed.

This expert system is based on a powerful new technology called Recommendation Systems that has emerged lately. Our Recommendation System is implemented using Collaborative Filtering. It addresses the information overload problem by grouping similar people for the purpose of creating viable recommendations.

In the case of risk level assessment the system will group the users according to a dynamically learned similarity measure. Each of the underwriting properties is assigned a different weight basing on the relative importance of this property. Finally, the system suggests an insurance premium that is similar to premium values of the users in a close proximity "virtual community" of the given user.

As future work we plan to develop a distributed version of our framework. We conceive this as important progress avenue of expert systems. It corresponded to real life scenarios where data is divided among many, often rivalry, different vendors. Thus, a protocol for sharing and merging semi-heterogeneous data should be devised.

We believe that flexible nature of the develop framework can facilitate not only usage in automated insurance underwriting but also in wide range of automated expert system ranging from part of speech tagging to E-commerce applications.

REFERENCES

1. J.Cookson, "Underwriting Gain and Loss Cycles", Chapter 33, p.769-779 in *Group Insurance*, 3rd Edition, Winsted, ACTEX Publications, 2000.
2. P.Maes, "Agents that Reduce Work and Information Overload", Communication of the ACM July 1994, Vol. 37, No. 7, 31-40.
3. J.Buchanan, N.Kock, "Information overload: A decision making perspective", p.49-58. Springer Verlag, Berlin, 2001.
4. J.Giarratano, G.Riley, "Expert Systems Principles and Programming", 3rd Edition, PWS-KENT Publishing Co., Boston, 1998.
5. J.S.Breese, D.Heckerman, C.Kadie, "Empirical analysis of predictive algorithms for collaborative filtering", In Proceedings of the 14th conference on Uncertainty in Artificial Intelligence, pages 43-52, 1998.
6. G.Salton, "Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer", Addison-Wesley, 1989.
7. B.Sarwar, G.Karypis, J.Konstan, J.Riedl, "Analysis of recommendation algorithms for e-commerce", In Proceedings of ACM E-Commerce, 2000.
8. B.Kitts, D.Freed, M.Vrieze, "Cross-sell: A fast promotion-tunable customer-item recommendation method based on conditional independent probabilities", In Proceedings of ACM SIGKDD International Conference, p. 437-446, 2000.
9. P.Resnick, N.Iacovou, M.Suchak, P.Bergstrom, J.Riedl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews", In Proceedings of CSCW '94, Chapel Hill, NC.
10. U.Shardanand, P.Maes, "Social Information Filtering: Algorithms for Automating 'Word of Mouth'", In Proceedings of CHI '95, Denver, CO.
11. N.Good, J.B.Schafer, J.A.Konstan, A.Borchers, B.Sarwar, J.Herlocker, J.Riedl, "Combining Collaborative Filtering with Personal Agents for Better Recommendations", In Proceedings of the 1999 Conference of the American Association of Artificial Intelligence (AAAI-99), p.439-446.
12. J.B.Schafer, J.A.Konstan, J.Riedl, "E-Commerce Recommendation Applications", Journal of Data Mining and Knowledge Discovery, vol.5 nos.1/2, p.115-152, 2001.
13. L.Chen, K.Sycara, "WebMate: A Personal Agent for Browsing and Searching", In Proceeding of the 2nd International Conference on Autonomous and Multi-Agent Systems, 1998, p.132-139.
14. L.Goffinet, M.Noirhomme-Fraiture, "Automatic Hypertext Link Generation based on Similarity Measures between Documents", Research Paper, RP-96-034, Institut d'Informatique, FUNDP, Namur, 1995.
15. U.Shardanand, "Social Information Filtering for Music Recommendation", A Master Thesis, MIT, September 1994.
16. N.J.Belkin, B.W.Croft, "Information filtering and information retrieval: Two sides of the same coin?", Communications of the ACM, 35(12), p.29-38, December 1992.
17. B.Sheth, "A Learning Approach to Personalized Information Filtering, A Master Thesis, MIT, 1994.
18. C.Aggrawal, J.L.Wolf, K.Wu, P.S.Yu, "Horting Hatches an Egg: A New Graph-Theoretic Approach to Collaborative Filtering", In Proceedings of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, San Diego, CA, 1999.
19. B.M.Sarwar, G.Karypis, J.A.Konstan, J.T.Riedl, "Application of Dimensionality Reduction in Recommender System - A Case Study", Full length paper at ACM WebKDD 2000 Web Mining for E-Commerce Workshop.
20. B.Sarwar, G.Karypis, J.Konstan, J.Riedl, "Item-Based Collaborative Filtering Recommendation Algorithms", The tenth international World Wide Web conference, October 2001, Orlando, FL, p.285-295.
21. A.Jain, M.N.Murty, P.Flynn, "Data clustering: A review", ACM Computing Surveys, 31(3) p.264-323, September 1999.
22. F.Marir, M.Watson, "Case-Based Reasoning: A review", The knowledge Engineering Review, vol. 9, No. 4, 1994.
23. T. Cover, P. Hart, "Nearest Neighbor pattern classification", IEEE Trans. Information Theory, 13:21-27, 1967.
24. R.L.Morin, D.E.Raeseide, "A Reappraisal of Distance-Weighted k-Nearest Neighbor Classification for Pattern Recognition with Missing Data", IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-11 (3), p. 241-243, 1981.
25. T.Hastie, R.Tibshirani, J.Friedman, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction", Springer Verlag Publishers, 2001.
26. D.Koller, M.Sahami, "Toward optimal feature selection", In Proceedings of ICML '96, p.284-292, 1996.

27. L.P.Kaelbling, M.L.Littman, A.W.Moore, "Reinforcement Learning: A Survey", Volume 4, p. 237-285, 1996.
28. J. B. MacQueen "Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability", Berkeley, University of California Press, 1:281-297, 1967.
29. D.E.Goldberg, "Genetic algorithms in search optimization and machine learning", Reading, MA: Addison-Wesley, 1989.
30. D.Thierens, D.E.Goldberg, „Mixing in genetic algorithms“, Proceedings of the Fifth International Conference On Genetic Algorithms, p. 38-45, 1993.
31. H.Lieberman, N.W.Van Dyke, A.S. Vivacqua, "Let's browse: A collaborative browsing agent", Knowledge-Based Systems, 12(8):427-431, 1999.
32. S.Siegel, "Nonparametric Statistics for the Behavioral Sciences", New York: McGraw-Hill, 1956,p. 75-83.
33. B.Limthamaphon, Z.Zhang, Y.Zhang, "Querying and Negotiating Structured E-Commerce Activities", Workshop on Agent Mediated Electronic Commerce, Bologna, Italy, 2002.
34. M.J.Pazzani, D.Billsus, "Adaptive Web Site Agents", Autonomous Agents and Multi-Agent Systems 5(2), p. 205-218, 2002.
35. J.E.Ratsaby, "Meeting the Challenges of E-Business by Distributed Artificial Intelligence", Bar-Ilan International Symposium on the Foundations of Artificial Intelligence, 2001.