In the Proceedings of IASTED International Conference on Artificial Intelligence and Soft Computing , Cancun, Mexico, May 2001

Automatically Improving the Accuracy of User Profiles with Genetic Algorithm*

Yi-Shin Chen
Integrated Media Systems Center
Computer Science Department
University of Southern California
Los Angeles, CA 90089-2564, U.S.A.
email: yishinc@cs.usc.edu

Cyrus Shahabi
Integrated Media Systems Center
Computer Science Department
University of Southern California
Los Angeles, CA 90089-2564, U.S.A.
email: shahabi@usc.edu

ABSTRACT
With information retrieval systems, bridging the gap between the physical characteristics of data with the user perceptions is challenging. In order to address this challenge, employing user profiles to improve the retrieval accuracy becomes essential. However, the system performance may degrade due to inaccuracy of user profiles. Therefore, for an approach to be effective, it should offer a learning mechanism to correct user input errors. Focusing on an image retrieval application, we utilize the users' relevance feedback to improve the profiles automatically using genetic algorithms (GA). Our experimental results indicated that the retrieval accuracy is significantly increased using the GA-based learning mechanism.

KEY WORDS
Image retrieval, fuzzy logic, genetic algorithm, relevance feedback, user profiles

1 Introduction

As the amount of available data is growing rapidly, the task of retrieving accurate information becomes increasingly essential in various domains. Whether we are looking for the latest news on websites, searching for images in virtual museums, downloading music clips in digital libraries, or purchasing products in e-commerce stores, we need accurate search engines or recommendation systems to locate our *desired information*. To achieve this goal, researchers in numerous research communities such as information retrieval (IR) [25, 2], software agents (SA) [14, 16], databases [4, 24] and collaborative filtering [18, 1] are aiming at constructing systems for providing accurate information in an efficient manner to users.

However, due to the subjectivity of human perception, it is challenging to bridge the gap between the physical characteristics of data (e.g., textures of images, lexical fragments of text files) with user perceptions to process requests. In order to address this challenge, researchers in IR and SA communities have obtained the retrieval results utilizing user profiles. These systems select items based on a comparison between the item representative and a user profile; both of these are a set of lexical fragments and their corresponding weights. Hence, these systems can suffer from inaccurate user profiles.

To address this problem, various learning techniques, such as *Bayesian classifiers*, *neural networks*, and *genetic algorithms* (GAs), have been utilized for revising user profiles in several research studies [17, 23, 13, 16] resulting in various levels of improvement. These studies have two major weaknesses. One is *content limitation*, e.g., lexical fragment methods can only be applied to text contents. The other is *over-specialization*, i.e., users can only obtain the information indicated in their profiles and have no chance of exploring new information they might desire. Moreover, because of the complication of user profiles, the learning processes are always time consuming and are not appropriate if user preferences change rapidly and frequently.

In this paper, we propose a collaborative approach that addresses the above mentioned drawbacks . Basically, there are several agents/functions/users (hereafter denoted as evaluators) in our system for providing answers to the user queries. Each user profile only needs to contain the user's personal standards and her confidence values to those evaluators she trusts. At the retrieval time, our system takes answers from all the trusted evaluators into consideration[1].

The advantage of this design is that different types of queries based upon different content can be uniformly supported within our system. Typical examples of these queries are "searching images with similar color/texture/shape", "finding clothes of a specific style", "suggesting articles recommended by famous people " and "locating CDs which are close to my purchasing history". Moreover, because most of the current IR techniques, such as [8, 15, 7, 17, 23, 13, 16], are orthogonal to our study, they can be used by our proposed system as well.

On the other hand, because the system relies heavily upon user profiles for providing accurate query results, the system accuracy may decline if user profiles are inaccurate.

*This research has been funded in part by NSF grants EEC-9529152 (IMSC ERC) and ITR-0082826, NASA/JPL contract nr. 961518, DARPA and USAF under agreement nr. F30602-99-1-0524, and unrestricted cash/equipment gifts from NCR, IBM, Intel and SUN.

[1] This is similar in concept to a popular TV game-show in the US where the players can poll the audience when they do not know the answer to a posed question. In our case, we only poll those members of the audience who we think they "know" the answer and we also provide appropriate weight to each member of audience based on our level of trust.

In practice, obtaining user profiles has been challenging. For example, some users may be uninterested in providing the data or unintentionally input incorrect information. Hence, to address this problem, we utilize the users' relevance feedback to improve the profiles automatically using Genetic Algorithm (GA) [9].

To the best of our knowledge, only very few studies [22, 16] incorporate GA for improving the user profiles. In addition to the drawbacks mentioned above, users must directly get involved in the evolution. That is, these systems need to acquire users' feedback for every generation. This leads into users frustration when using the system. On the contrary, with our design, user involvement is needed for providing the feedback only at the initial step of the evolution.

In this paper, we concentrate our application domain on image retrieval and construct an *adaptive soft query system* (ASQ) for supporting image queries. The results in [20] demonstrate that our proposed soft query model, assuming accurate user profiles, outperforms the conventional methods in the range of 20% to 160% depending upon other system settings. In this paper, we have relaxed our assumptions and focused on correcting user profiles using the GA learning mechanism. Our experimental results indicate significant increase in the system's accuracy after integrating this learning mechanism.

The remainder of this paper is organized as follows. Section 2 describes the concept of genetic algorithms used in our *adaptive* soft query system. In Section 3, we discuss the soft query system and the learning mechanism. Section 4 depicts the results of our evaluations as well as the details of the system implementation and our benchmarking method. Section 5 concludes the paper.

2 Genetic algorithms

Genetic algorithms (GAs), which were introduced by Holland [9], are iterative search techniques based on the spirit of natural evolution. By emulating biological selection and reproduction, GAs can efficiently search through the solution space of complex problems. Basically, a GA operates on a population of candidate solutions called *chromosomes*. A chromosome, which is composed of numerous genes, represents an encoding of the problem and associates it with a fitness value evaluated by the fitness function. This fitness value determines the goodness and the survival ability of the chromosome.

Generally, GA starts by initializing the population and evaluating its corresponding fitness values. Before it terminates, GA produces newer generations iteratively. At each generation, a portion of the chromosomes is selected according to the survival ability for reproducing offspring. The offspring are generated through crossover and mutation processes and are used for replacing some chromosomes in the population with a probability consistent with their fitness values. In other words, with the help of the fitness function to point out the correct direction, GA could construct better and better chromosomes from the best partial genes of past samplings. Please reference [6] for mathematical foundations.

In summary, GA is composed of a fitness function, a population of chromosomes and three operators - selection, crossover and mutation. The parameter settings of the operators can be chosen depending on the applications or remain unchanged even when the applications are varied. However, the fitness function and the coding method are required to be specially designed for each problem. The design of fitness function and encoding method for ASQ will be described in Section 3.2.

3 Adaptive Soft Query System

ASQ consists of two main components: an online soft query system and an offline GA-based learning mechanism. The soft query system, which is described in Section 3.1, provides personalized query results by integrating image information and user profiles with a fuzzy-logic based aggregation technique. Since the system performance is deeply influenced by the accuracy of user profiles, ASQ uses a GA based learning mechanism, which is described in Section 3.2, to improve the user profiles through users' feedback.

Here, we briefly describe the system behavior when employing these two components. After a user submits a query, the soft query system is activated to aggregate image information and the corresponding user profile for retrieving personalized query results. The result is a list of images ranked by their soft membership values for a given query. The user can modify the ratings of images[2] and return them to the system as his/her relevance feedback. This input feedback stimulates the learning mechanism for offline improvement of the user's profile.

3.1 Soft Query System

The soft query system was first introduced by authors in [20]. In order to incorporate user perceptions into the query processing, the system needs to keep two different types of data, which are elucidated in Section 3.1.1, and employs a special mechanism, which is described in Section 3.1.2, for integrating image and user information into the query processing.

We are aware of the high complexity of the soft query system and have addressed it in [19] and [21]. This optimized version is comparable in complexity to the conventional image retrieval techniques. To simplify the discussion, we focus only on the general framework of soft query.

[2] For the real application, the user might simply drag the image and move it up in the list using a couple of mouse clicks.

3.1.1 Soft Query Data Model

Two different types of data are utilized during the query processing. The first is image information such as size, style and color. The other is user information (the user profile). Each image is associated with two categories of attributes. The first category is termed as *features* ($\Im$), whose attributes are user-independent. The other category as defined in Definition 3.1 is called *properties* (T), whose attributes are user-dependent. Every image retains several features and/or properties.

Definition 3.1: Each property T_i partitions the set of all images into k classes, $C_1^{T_i}, C_2^{T_i} \cdots C_k^{T_i}$. The membership of an image, n, to a class $C_j^{T_i}$ is determined by an evaluator (denoted as e) via the following probability function: $P(n \in C_j^{T_i} \mid e)$, whose value could be either a real number within the range of 0 to 1 or words describing user perceptions (e.g., *high* or *low*). ∎

The user profile is composed of two parts: user confidence data and user fuzzy cut value. The formal definition of user confidence data is the following:

Definition 3.2: E denotes a set of evaluator representatives in the database. O represents the set of observers who have assigned reference confidence values to evaluators. π is a confidence value for an observer o to an evaluator e; $\pi \;:\; o \in O,\ e \in E \;\to b$. Note that the value of b is a form of human judgment.∎

Commentary:

In practice, asking people to describe their perceptions with very precise values is almost impossible. Moreover, different people have different interpretations of words. That is, the information describing personalities, physical features, preferences and personnel evaluation is imprecise. In order to handle this uncertainty during the query processing, fuzzy logic (FL) [26] is adapted into our system. The concept of FL was first introduced by Zadeh [26] to problems for which precise formulation is not possible. The original FL has the weakness that uncertainty cannot be used when it computes with words. Therefore, Karnik and Mendel advocated type 2 FL [11, 12] for overcoming this disadvantage. However, for the sake of simplicity, we only consider the original FL in this paper.

With the help of FL, the soft query system can store and employ human's fuzzy perceptions. First, users pick up the words already defined in the system (hereafter denoted as fuzzy sets) to express their opinions. Then, all fuzzy words will be converted to real values customized for the user perceptions according to the user's fuzzy cut value.

Note that using the same model, we can capture conventional image retrieval operations such as finding all images similar to a query image in color, shape or texture. Moreover, the model can utilize the presence of multiple algorithms for the same comparison function. To illustrate, consider the following example.

Example 3.1: The property, CS, partitions the set of all images into k classes, C_1^{CS} to C_k^{CS}. C_j^{CS} is a class of all images similar in color to image j. The memberships of images to these classes are:

$$P(image1 \in C_1^{CS} \mid \text{ QBIC-}\mathcal{A}_0 \text{ Algorithm}) = 1$$
$$P(image1 \in C_2^{CS} \mid H_i\text{Function}) = 0.7$$
$$P(image1 \in C_2^{CS} \mid userA) = Middle$$

This example illustrates the flexibility of the model where evaluators can be algorithms, functions, or humans. It also shows that the membership value can be a fuzzy set or a real value. ∎

3.1.2 Query Processing Method

Each query statement can be decomposed into several atomic clauses. Some clauses access and process feature-values, and the others utilize property classes. Therefore, our soft query model consists of two different query processing methods: feature query and property query. These two query methods compute the image membership separately and combine their results using the standard rules of fuzzy logic. The final query results are ordered according to their membership values.

The feature query method is straightforward and simply involves the Boolean evaluation of the predicates. As opposed to the user-independent results of feature queries, the results of property queries depend upon the user submitting the queries. The probability of an image belonging to the result set is presented by a real number between 0 and 1. The aggregation functions to compute the membership $\lambda_{d,c}$ of image d to a property class c given by user u should satisfy the properties - *Conservation*, *Monotonicity*, *Commuatativity* and *Associativity* [5]. With these properties, the query optimizer can replace the original query with a logically equivalent one and still obtain exactly the same result. The optimized aggregation functions we proposed are stated below. Refer to [21] for more details.

$$\begin{aligned} E_f &= \{e \mid e \in E, \pi_{u,e} = f\} \\ \theta^{*}_{d,c,f} &= f \times \max_{e \in E_f}\{P(d \in c \mid e)\} \\ \lambda_{d,c} &= \max_{f \in F}\{\theta^{*}_{d,c,f}\} \end{aligned} \tag{1}$$

$$\begin{aligned} \theta^{**}_{d,c,f} &= \min(f, \max_{e \in E_{u,f}}\{P(d \in c \mid e)\}) \\ \lambda_{d,c} &= \max_{f \in F}\{\theta^{**}_{d,c,f}\} \end{aligned} \tag{2}$$

3.2 GA-based Learning Mechanism

This learning mechanism is an automatic and off-line process. It employs GA for improving the user profile by decoding the best chromosome to replace existing user profiles in the database after its evolution. User involvement is only needed for providing the relevance feedback as the goal of GA prior to the beginning of evolution. Note: the learning mechanism is only triggered by the user feedback and not needed during querying processing.

Subsequently, we explicate the coding design for GA in our learning mechanism. The chromosomes represent possible user profiles of a specific user and each gene corresponds to a record in the user profiles. Two types of records are involved in the genes. One is user confidence information with k records, where k is the number of user representatives in the system. The value of the ith gene is an integer in $[0, L-1]$, where L is the number of fuzzy terms used in the system, and denotes the user's confidence level to user i. The other is a user fuzzy cut value which is associated with the $(k+1)$th gene. The value of fuzzy cut is $(t+1)/L$, where $t \in [0, L-1]$ is the value of this gene.

For example, suppose that there are 50 user representatives and 8 different fuzzy terms in the system, there will be 51 genes per chromosome where the first 50 genes represent the corresponding confidence values to the user representatives, and the last gene represents the value of the user fuzzy cut. Additionally, after decoding, the value of 0 in gene i indicates that the confidence level to user i is "none" and the value of 6 in gene 51 indicates that the value of fuzzy cut is $(6+1)/8 = 0.875$. Likewise, after encoding, "full' 'confidence level to user i is represented by a number 7 in gene i and the 0.75 fuzzy cut is denoted by a number 5 in gene 51.

This coding method can guarantee a one-to-one mapping of profiles to chromosomes. That is, a chromosome will be decoded to one and only one legal user profile, and a user profile will be encoded to one and only one chromosome. Consequently, the solution space will be equal to the searching space in GA. This implies that our coding method is effective.

Finally, our GA fitness function in the learning mechanism is described. The fitness function heavily utilizes the users' relevance feedback B. It first decodes the chromosome into a user profile. Secondly, it obtains the query result Q according to this user's profile using Equation (1). In other words, this process needs to interact with the database for obtaining the image information. Finally, it generates the fitness value by measuring the degree of similarity between the query result and the users' relevance feedback using the following equation:

$$\text{Similarity Distance}(Q, B) = \frac{\sum_{i=1}^{n} Q_i \times B_i}{\sqrt{\sum_{i=n}^{n} Q_i^2 \times \sum_{i=n}^{n} B_i^2}} \tag{3}$$

Once a user offers his/her user feedback to trigger the learning process, the learning mechanism first encodes the corresponding user profile to a chromosome and randomly generates other chromosomes as the initial population. Subsequently, GA iteratively discover better user profiles until it achieves the terminal condition such as the fitness value of one chromosome being 1 or the generation number being 200. In the end, the learning mechanism decodes the best chromosome to a user profile for replacing the current user profile in the database.

4 Performance Evaluation

In this section, we first describe our experimental setup and benchmarking method in Section 4.1. Subsequently, the details of our experimental results are discussed in Section 4.2.

4.1 Experimental Setup and Benchmarking Method

We developed GA for ASQ using SUGAL [10] for its wide range of operators and datatypes. ASQ is implemented in C and on top of Microsoft Access 2000, which is running on a Pentium II 233MHZ processor with Microsoft NT4.0.

In order to populate data for evaluation purposes, we propose a parametric algorithm. By changing the parameters of the algorithm, we can simulate various database contents. The benchmarking method incorporates two data structures to generate the perfect knowledge, a 3-dimensional matrix $\mathring{A}_0$, about the classification behaviors of all users for providing ideal user feedback. One is the classification information of evaluators and the other is user profiles.

Before populating cube $\mathring{A}_0$, the algorithm first randomly generates the classification information of k evaluators. The classification information of each evaluator is a matrix $\bar{\mathcal{E}}$ with X rows and Z columns, where X represents the number of images and Z is the number of classes. The element $\bar{\mathcal{E}}(i, j)$ represents the membership for image i to class j defined by this evaluator. Subsequently, the system randomly generates a list of confidence values and a fuzzy cut value for each profile $\ddot{\pi}$. Each confidence value is represented by a fuzzy term, which is an integer in the range of $[0, 7]$, where 0 represents "no confidence at all" and 7 represents "full confidence".

After constructing these two data structures, the system then populates the user classifications to $\mathring{A}_0$ according to the query results by aggregating $\ddot{\pi}$ and $\bar{\mathcal{E}}$ using Equation (1). To simulate imperfect or wrong user feedback during GA, the system tunes the classification data retrieved from $\mathring{A}_0$ as relevance feedback by a noisy process according to the noise level. Noise level 0 represents perfect feedback and noise level 10 represents complete chaos. Finally, in order to reproduce inaccurate user profiles, the system regenerates all user profiles by utilizing random values as the confidence values and fuzzy cut values.

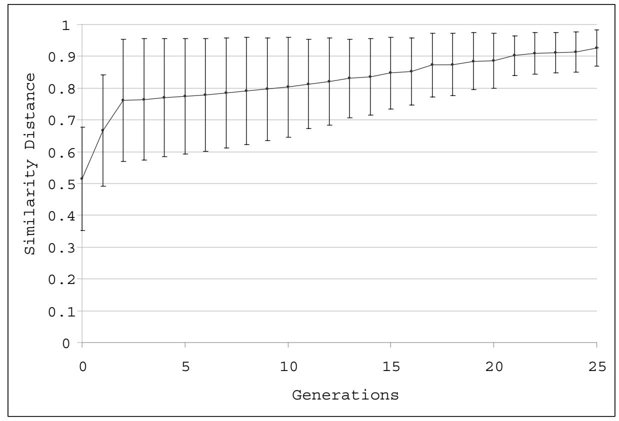
4.2 Experimental Results

We conducted several sets of experiments to verify our system performances and to compare the results of different GA parameter settings. In these experiments, we observed a significant margin of improvement after incorporating GA in matching the user expectations in various settings. It is also shown that the performance improvement of our learning mechanism is independent of the number of users and classes. Moreover, the improvement is linearly increased with the number of images. However, due to the space limitation, we only report on the improvements achieved by applying our learning mechanism.

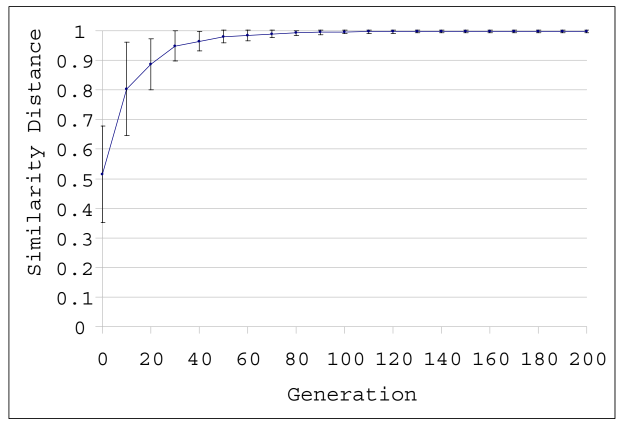
The results shown for each set of experiments are averaged over twenty runs, where each run is executed with different seeds for the random generator functions. The parameter settings of GA operators [10, 6] are: population size=30, two-point crossover, tournament selection, keep the elitism, mutation rate = 0.1, natural simulated annealing replacement method [3] and annealing decay=0.95. The benchmark settings of the following figures, i.e., the number of images, the number of users, the number of classes, and the number of user representatives, are fixed at $X = 500$, $Y = 300$, $Z = 20$, and $k = 50$, respectively.

Figure 1 demonstrates the improvements achieved by our learning mechanism. The X-axis is the generation number and the Y-axis depicts the similarity distance computed by Equation (3). The error bar around each similarity distance value indicates the standard deviation over 20 runs. We introduce no noise condition where users provide perfect feedback to ASQ. In this experiment, Figure 1.a indicates that the performance of our system improves drastically after incorporating the learning mechanism under the same query request and user feedback. As observed, the system performance increases nearly 50% within five generations of the evolution, and the user expectations are matched almost perfectly after 50 generations of learning. Moreover, in the majority of our experiments, GA can acquire ideal user profiles within 100 generations, i.e., the retrieval accuracy is increased up to 100%. These results suggest that our learning mechanism is efficient and can greatly improve user profiles.

In order to compare the system performance when the user relevance feedback are imperfect, we introduce five different noise levels in the experiments of Figure 2, where noise level 0 represents perfect feedback and noise level 10 represents complete chaos. The X-axis is the number of generations and the Y-axis depicts the similarity distance (between query results and perfect user feedback) computed by Equation (3). Additionally, the system decodes the best chromosome for replacing the user profile in the database and acquires different user query and corresponding feedback in every 200 generations. As revealed by Figure 2, the system accuracy remains the same even though the queries and user feedback are altered. It shows that our GA-based learning mechanism can obtain global-optimal profiles for users. Furthermore, although the noise levels affect the system performance, our learning mechanism still improves the quality of user profiles in the range of 40% to 80%. This figure also indicates that our learning mechanism has the ability to tolerate noises during the learning process.



a. 25 generations



b. 200 generations

Figure 1. The system improvement

5 Conclusion

We propose a collaborative approach to bridge the gap between the physical characteristics of data with the user perceptions in information retrieval problems. However, this system heavily relies on user profiles for providing accurate query results. The system accuracy may decline if user profiles are inaccurate. Therefore, we introduced a learning mechanism which utilizes the users' relevance feedback to improve the user profiles automatically using genetic algorithms. The experimental results indicated that the retrieval accuracy is significantly increased up to 100% by our GA-based learning mechanism. It has also demonstrated that our learning mechanism has the ability of tolerating noise during the learning process and improvement is in the range of 40% to 80%.

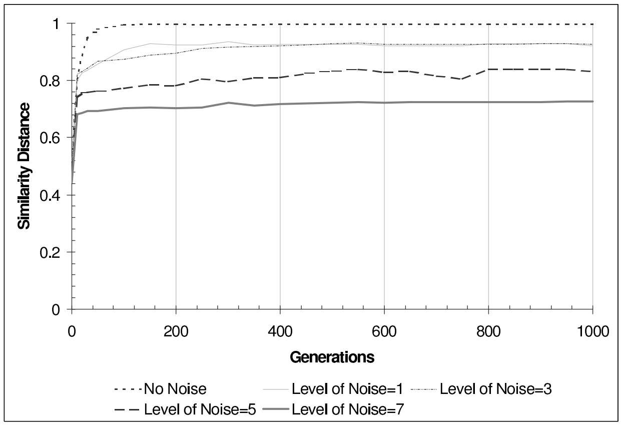


Figure 2. Impacts of noise levels

We intend to extend this study in two ways. First, we intend to apply the same concepts to a different application, E-commerce recommendation systems, in order to alleviate their information overload problems. Second, our aggregation function is defined in the original fuzzy logic domain. However, recently Karnik and Mendel [11, 12] introduced fuzzy type 2 to incorporate uncertainty in computation. Implementing our aggregation function in fuzzy type 2 domain will enhance accuracy of the computations and further improve the results.

References

[1] M. Balabanovi. An adaptive web page recommendation service. In *Proceedings of Autonomous Agents*, pages 378–385, Marina Del Rey, California USA, 1997.

[2] A. Berger and J. Lafferty. Information retrieval as statistical translation. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, Berkeley, CA USA, August 1999.

[3] Y.-S. Chen and J.-S. Chen. A natural genetic algorithm in job shop problem. *Master Thesis, National Centrail University, Taiwan*, 1997.

[4] R. Fagin. Combining fuzzy information from multiple systems. In *Proc. Fifteenth ACM Symp. on Principles of Database Systems*, pages 216–226, 1996.

[5] R. Fagin. Fuzzy queries in multimedia database systems. *ACM SIGACT-SIGMOD-SIGART Symposium on Principle of Database Systems*, 1998.

[6] D. Goldberg. *Genetic Algorithms in Search, Optimisation, and Machine Learning*. Addison-Wesley, Wokingham, England, 1989.

[7] V. Gudivada and V. Raghavan. Design and evaluation of algorithms for image retrieval by spatial similarity. *IEEE Transations on Information Systems*, 13(2):115–144, 1995.

[8] J. Hafner, H. S. Sawhney, W. Equitz, M. Flickner, and W. Niblack. Efficient color histogram indexing for quadratic form distance functions. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 17:729–736, 1995.

[9] J. Holland. *Adaption in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, Michigan, 1975.

[10] A. Hunter. Sugal programming manual. *http://www.trajan-software.demon.co.uk/sugal.htm*, 1995.

[11] N. Karnik and J. Mendel. Introduction to type-2 fuzzy logic systems. In *Proceeding of 1998 IEEE FUZZ Conference*, pages 915–920, Anchorage, AK, May 1998.

[12] N. Karnik and J. Mendel. Operations on type-2 fuzzy sets. *Int'l. J. on Fuzzy Sets and Systems*, 2000.

[13] W. Lam, S. Mukhopadhyay, J. Mostafa, and M. Palakal. Detection of shifts in user interests for personalized information filtering. In *Proceeding of the 19th Int'l ACM-SIGIR Conf on Research and Development in Information Retrieval*, pages 317–325, 1996.

[14] P. Maes. Agents that reduce work and information overload. *Communications of the ACM*, 37(7):30–40, 1994.

[15] M. Mirmehdi and M. Petrou. Segmentation of color textures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:142–159, 2000.

[16] A. Moukas. Amalthea: Information discovery and filtering using a multiagent evolving ecosystem. In *1st Int. Conf. on The Practical Applications of Intelligent Agents and Multi-Agent Technology (PAAM)*, London, 1996.

[17] M. Pazzani and D. Billsus. Learning and revising user profiles: The indentification of interesting web sites. *Machine Learning*, 27:313–331, 1997.

[18] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of ACM Conf. on Computer-Supported Cooperative Work*, pages 175–186, Chapel Hill, NC, 1994.

[19] C. Shahabi and Y.-S. Chen. Efficient support of soft query in image retrieval systems. In *Proceeding of SSGRR 2000 Computer and eBusiness Conference*, Rome, Italy, August 2000.

[20] C. Shahabi and Y.-S. Chen. Soft query in image retrieval systems. In *Proceeding of SPIE Internet Imaging , Electronic Imaging 2000*, volume 3964, pages 57–68, San Jose, CA, 2000.

[21] C. Shahabi and Y.-S. Chen. A unified and efficient framework for customized query in image retrieval systems. In *Proceeding of International Conference on Enterprise Information Systems*, Setubal, Portugal, July 2001.

[22] B. Sheth. Evolving agents for personalized information filtering. In *Proceedings of the Ninth IEEE Conference on Artificial Intelligence for Applications*, 1993.

[23] A. Tan and C. Teo. Learning user profiles for personalized information dissemination. In *Proceeding of Int'l Joint Conf. on Neural Network*, pages 183–188, 1998.

[24] Y. Theodoridis, E. Stefanakis, and T. Sellis. Efficient cost models for spatial queries using r-trees. *IEEE Transactions on Knowledge and Data Engineering*, 12(1):19–32, 2000.

[25] D. Widyantoro, T. Ioerger, and J. Yen. An adaptive algorithm for learning changes in user interests. In *8th Int'l Conf on Information and Knowledge Management*, November 1999.

[26] L. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems 1(1)*, pages 3–28, 1978.