PERCEPTION-BASED DECISION PROCESSING AND ANALYSIS

by

Masoud Nikravesh, Gamil Serag-Eldin
and Soaud Ben-Soafi

Memorandum No. UCB/ERL M03/21

20 June 2003

PERCEPTION-BASED DECISION PROCESSING AND ANALYSIS

by

Masoud Nikravesh, Gamil Serag-Eldin
and Soaud Ben-Soafi

Memorandum No. UCB/ERL M03/21

20 June 2003

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Perception-Based Decision Processing and Analysis

Masoud Nikravesh, Gamil Serag-Eldin and Soaud Ben-Soafi
BISC Program, Computer Sciences Division, EECS Department
University of California, Berkeley, CA 94720, USA
Email: nikravesh@cs.berkeley.edu
Tel: (510) 643-4522
Fax: (510) 642-5775
URL: http://www-bisc.cs.berkeley.edu

Abstract: Searching a database records and ranking the results based on multi-criteria queries is central for many database applications used within organizations in finance, business, industrial and other fields. For Example, the process of ranking (scoring) has been used to make billions of financing decisions each year serving an industry worth hundreds of billion of dollars. To a lesser extent, ranking has also been used to process hundreds of millions of applications by U.S. Universities resulting in over 15 million college admissions in the year 2000 for a total revenue of over $250 billion. College admissions are expected to reach over 17 million by the year 2010 for total revenue of over $280 billion. In this paper, we will introduce fuzzy query and fuzzy aggregation as an alternative for ranking and predicting the risk for credit scoring and university admissions, which currently utilize an imprecise and subjective process. In addition we will introduce the BISC Decision Support System. The main key features of the BISC Decision Support System for the internet applications are 1) to use intelligently the vast amounts of important data in organizations in an optimum way as a decision support system and 2) To share intelligently and securely company's data internally and with business partners and customers that can be process quickly by end users. The model consists of five major parts: the Fuzzy Search Engine (FSE), the Application Templates, the User Interface, the database and the Evolutionary Computing (EC).

1 Introduction

Most of the available systems 'software' are modeled using crisp logic and queries, which results in rigid systems with imprecise and subjective process and re-

sults. In this chapter, we introduce fuzzy querying and ranking as a flexible tool allowing approximation where the selected objects do not need to match exactly the decision criteria resembling natural human behavior.

The model consists of five major parts: the Fuzzy Search Engine (FSE), the Application Templates, the User Interface, the database and the Evolutionary Computing (EC). We developed the software with many essential key features. The system is designed as generic system that can run different application domains. To this end, the Application Template module provides all needed information for a certain application as object attributes and properties, and serve as a guideline structure for building a new application.

The Fuzzy Search Engine (FSE) is the core module of the system. It has been developed to be generic so that it would fit any application with minimal changes. The main FSE components are the membership functions, similarity functions and aggregators. Administrator can also change the membership function to be used to do searches.

Through the user interface a user can enter and save his/her profile, input criteria for a new query, run different queries and display results. The user can manipulate manually the result by eliminating what he/she disproof and the ranking according to his/her preferences.

This process is monitored and learned by the Evolutionary Computing (EC) module recording and saving user preferences to be used as basic queries for that particular user. We present our approach with three important applications: ranking (scoring) which has been used to make financing decisions concerning credit cards, cars and mortgage loans; the process of college admissions where hundreds of thousands of applications are processed yearly by U.S. Universities; and date matching as one of the most popular internet programs. However, the software is generic software for much more diverse applications and to be delivered as stand alone software to both academia and businesses.

Consider walking into a car dealer and leaving with an old used car paying a high interest rate of around 15% to 23% and your colleague leaves the dealer with a luxury car paying only a 1.9% interest rate. Consider walking into a real estate agency and finding yourself ineligible for a loan to buy your dream house. Also consider getting denied admission to your college of choice but your classmate gets accepted to the top school in his dream major. Welcome to the world of ranking, which is used both for deciding college admissions and determining credit risk. In the credit rating world, FICO (Fair Isaac Company) either makes you or breaks you, or can at least prevent you from getting the best rate possible (Fair Isaac). Admissions ranking can either grant you a better educational opportunity or stop you from fulfilling your dream.

When you apply for credit, whether it's a new credit card, a car loan, a student loan, or a mortgage, about 40 pieces of information from your credit card report are fed into a model. That model provides a numerical score designed to predict your risk as a borrower. When you apply for university or college admission, more than 20 pieces of information from your application are fed into the model. That model provides a numerical score designed to predict your success rate and risk as a student. In this paper, we will introduce fuzzy query and fuzzy aggregation as an alternative for ranking and predicting risk in areas which currently utilize an imprecise and subjective process.

The areas we will consider include: credit scoring (***Table 1***), credit card ranking (***Table 2***), and university admissions (***Table 3***). Fuzzy query and ranking is robust, provides better insight and a bigger picture, contains more intelligence about an underlying pattern in data and is capable of flexible querying and intelligent searching (Nikravesh, 2001a). This greater insight makes it easy for users to evaluate the results related to the stated criterion and makes a decision faster with improved confidence. It is also very useful for multiple criteria or when users want to vary each criterion independently with different degrees of confidence or weighting factor (Nikravesh, 2001b).

2 Fuzzy Query and Ranking

In the case of crisp queries, we can make multi-criterion decision and ranking where we use the functions AND and OR to aggregate the predicates. In the extended Boolean model or fuzzy logic, one can interpret the AND as a fuzzy-MIN function and the OR as a fuzzy-MAX function. Fuzzy querying and ranking is a very flexible tool in which linguistic concepts can be used in the queries and ranking in a very natural form. In addition, the selected objects do not need to match the decision criteria exactly, which gives the system a more human-like behavior.

2.1 Measure of Association and Fuzzy Similarity

As in crisp query and ranking, an important concept in fuzzy query and ranking applications is the measure of association or similarity between two objects in consideration. For example, in a fuzzy query application, a measure of similarity between two a query and a document, or between two documents, provides a basis for determining the optimal response from the system. In fuzzy ranking applications, a measure of similarity between a new object and a known preferred (or non-preferred) object can be used to define the relative goodness of the new object. Most of the measures of fuzzy association and similarity are simply extensions from their crisp counterparts. However, because of the use of perception

Table 1. Variables, Granulation and Information used to create the Credit Rating System Model.

AOA: Amount owed on accounts is too high. 01
LDA: Level of Delinquency on accounts. 02
BRA: Too few bank revolving accounts.03
BorNRA: Too many bank or national revolving accounts. 04
RILI: lack of recent installment loan information: 04
ACB: Too many accounts with balances. 05
CFA: Too many Consumer finance accounts. 06
APH: Account payment history too new to rate.07
RI: Too many recent inquiries in the last 12 months.08
AOinL12M: Too many accounts opened in the last 12 months. 09
PBtoCLRI: Proportion of balances to credit limits is too high on revolving accounts. 10
AORI: Amount owed on revolving accounts is too high.11
LRCH: Length of revolving credit history is too short.12
TD: Time since delinquency is too recent or unknown.13
LCH: Length of credit history is too short.14
LRBRI: Lack of recent bank revolving information.15
LRRAI: Lack of recent revolving account information. 16
RNMBI: No recent non-mortgage balance information.17
NAwD: Number of accounts with delinquency.18
ACPasA: Too few accounts currently paid as agreed.19
TDPRorC: Time since derogatory public record or collection.20
APDonA: Amount past due on accounts.21
SDDPRorC: Serious delinquency, derogatory public record, or collection.22
BorNRAwB: Too many bank or national revolving accounts with balances.23
RB: No recent revolving balances.24
LILH: Length of installment loan history 25
NRA: Number of revolving accounts.26
BNRorORA: Number of bank revolving or other revolving accounts.26
ACPasA: Too few accounts currently paid as agreed. 27
NofEA: Number of established accounts.28
DofLI: Date of last inquiry too recent.29
BB: No recent bankcard balances.29
TRAO: Time since most recent account opening too short.30
AwRPI: Too few accounts with recent payment information.31
AOonDA: Amount owed on delinqcent accounts. 31
LofrILI: Lack of recent installment loan information.32
PofLBtoLA: Proportion of loan balances to loan amounts is too high. 33
LTOILE: Length of time open installment loans have been established * 36
NFCAERLFH: Number of finance company accounts established relative to length of finance history 37
SDPRCF: Serious delinquency and public record or collection filed X 38
SD: Serious delinquency X 39
DPRCF: Derogatory public record or collection filed X 40
LRHFALFA: Lack of recent history on finance accounts, or lack of finance accounts * 99
LRIALAL: Lack of recent information on auto loan, or lack of auto loans * 98

AOA={'Too Low'; 'Low'; 'Average'; 'High'; 'Too High';Extremely High'; 'Not Care'};
LDA = {'Too Low'; 'Low'; 'Average'; 'High'; 'Too High';Extremely High'; 'Not Care'};
BRA ={'Too Few'; 'Few'; 'Some'; 'Many'; 'Too Many'; 'Not Care'};
BorNRA = {'Too Few'; 'Few'; 'Some'; 'Many'; 'Too Many'; 'Not Care'};
RILI = {'Lacking'; 'Not Enough'; 'Enough';'Not Care'};
ACB = {'Too Few'; 'Few'; 'Some'; 'Many'; 'Too Many'; 'Not Care'};
CFA = {'Too Few'; 'Few'; 'Some'; 'Many'; 'Too Many'; 'Not Care'};
APH = {'Too New'; 'New'; 'Kind of New'; 'Established'; 'Well Established'; 'Not Care'};
RI = {'Too Few'; 'Few'; 'Some'; 'Many'; 'Too Many'; 'Not Care'};
AOinL12M = {'Too Few'; 'Few'; 'Some'; 'Many'; 'Too Many'; 'Not Care'};
PBtoCLRI= {'Too Low'; 'Low'; 'Average'; 'High'; 'Too High';'Extremely High'; 'Not Care'};
AORI = {'Too Low'; 'Low'; 'Average'; 'High'; 'Too High';'Extremely High'; 'Not Care'};
LRCH ={'Too Short'; 'Short'; 'Average'; 'Long'; 'Too Long'; 'Not Care'};
TD = {'Too Recent'; 'Recent'; 'No Recent';'Unkown'; 'Not Care'};
LCH= {'Too Short'; 'Short'; 'Average'; 'Long'; 'Too Long'; 'Not Care'};
LRBRI = {'Lacking'; 'Not Enough'; 'Enough';'Not Care'};
LRRAI= {'Lacking'; 'Not Enough'; 'Enough';'Not Care'};
RNMBI= {'Too Recent'; 'Recent'; 'No Recent';'Unkown'; 'Not Care'};
NAwD = {'Too Few'; 'Few'; 'Some'; 'Many'; 'Too Many'; 'Not Care'};
ACPasA = {'Too Few'; 'Few'; 'Some'; 'Many'; 'Too Many';'Not Care'};
TDPRorC ={'Too Short'; 'Short'; 'Average'; 'Long'; 'Too Long';'Not Care'};
APDonA = {'Too Low'; 'Low'; 'Average'; 'High'; 'Too High';'Extremely High'; 'Not Care'};
SDDPRorC= {'Not Serous'; 'Serious'; 'Very Serious'; 'Extremely Serious'; 'Not Care'};
BorNRAwB = {'Too Few'; 'Few'; 'Some'; 'Many'; 'Too Many'; 'Not Care'};
RB = {'Too Recent'; 'Recent'; 'No Recent'; 'Not Care'};
LILH= {'Too Short'; 'Short'; 'Average'; 'Long'; 'Too Long';'Not Care'};
NRA = {'Too Low'; 'Low'; 'Average'; 'High'; 'Too High';'Extremely High'; 'Not Care'};
BNRorORA ={'Too Low'; 'Low'; 'Average'; 'High'; 'Too High';'Extremely High'; 'Not Care'};
ACPasA = {'Too Few'; 'Few'; 'Some'; 'Many'; 'Too Many'; 'Not Care'};
NofEA = {'Too Low'; 'Low'; 'Average'; 'High'; 'Too High';'Extremely High'; 'Not Care'};
DofLI = {'Too Recent'; 'Recent'; 'No Recent'; 'Not Care'};
BB = {'Too Recent'; 'Recent'; 'No Recent'; 'Not Care'};
TRAO = {'Too Short'; 'Short'; 'Average'; 'Long'; 'Too Long';'Not Care'};
AwRPI = {'Too Few'; 'Few'; 'Some'; 'Many'; 'Too Many'; 'Not Care'};
AOonDA = {'Too Low'; 'Low'; 'Average'; 'High'; 'Too High';'Extremely High'; 'Not Care'};
LofrILI = {'Lacking'; 'Not Enough'; 'Enough';'Not Care'};
PofLBtoLA = {'Too Low'; 'Low'; 'Average'; 'High'; 'Too High';'Extremely High'; 'Not Care'};
LTOILE = {'Too Short'; 'Short'; 'Average'; 'Long'; 'Too Long';'Not Care'};
NFCAERLFH = {'Too Low'; 'Low'; 'Average'; 'High'; 'Too High';'Extremely High'; 'Not Care'};
SDPRCF = {'Not Serous'; 'Serious'; 'Very Serious'; 'Extremely Serious'; 'Not Care'};
SD = {'Not Serous'; 'Serious'; 'Very Serious'; 'Extremely Serious'; 'Not Care'};
DPRCF = {'Not Serous'; 'Serious'; 'Very Serious'; 'Extremely Serious'; 'Not Care'};
LRHFALFA = {'Lacking'; 'Not Enough'; 'Enough';'Not Care'};
LRIALAL = {'Lacking'; 'Not Enough'; 'Enough';'Not Care'};

Table 2. Variables, Granulation and Information used to create the Credit Card Ranking System Model.

```
% Vis: Vissaa
% VisG: Vissaa Gold
% VisP: Vissaa Platinum
% MSCS: Masters Cards
% MSCSG: Masters Cards Gold
% MSCSP: Masters Cards Platinum
% Amaexs: Americana Experesses
% APR: Annual Percentage Rate
% APRC: Cash Advance APR
% AF: Annual Fee
% GP: Grace Periods
% CAF: Cash Advance Fee
% IIR: Introductory Interest Rate
% RBP: Rebate Programs
% FVR: Fix vs. Variable Rate
% GF: General Fee
% CF: Consumer Feedback
% RI: Reputation of Issuer
% FF: Freqoet Flyer
% CA: Card Acceptability
% RCF: Return Check Fee
% LPF: Late Payment Fee
% SI: Security Interest
% DO: Dispute Option
% CS: Customer Service
% SPP: Special Payment Plan
% PP: Partner Programs
% IYR: Itemize Annual Report

CARDName= {'Vissaa'; 'Vissaa Gold'; 'Vissaa Platinum'; 'Masters Cards'; 'Masters Cards Gold'; ...
    'Masters Cards Platinum'; 'Americana Experesses'; 'Not Care'};
APR={'Very Low'; 'Low'; 'Average'; 'High'; 'Very High';'Extremely High'; 'Not Care'};
APRC={'Very Low'; 'Low'; 'Average'; 'High'; 'Very High';'Extremely High'; 'Not Care'};
AF= {'Very Low'; 'Low'; 'Average'; 'High'; 'Very High';'Extremely High'; 'Not Care'};
GP={'Extremely Short'; 'Very Short'; 'Short'; 'Medium'; 'Long'; 'Very Long'; 'Not Care'};
CAF= {'Very Low'; 'Low'; 'Average'; 'High'; 'Very High';'Extremely High'; 'Not Care'};
IIR= {'Very Low'; 'Low'; 'Average'; 'High'; 'Very High';'Extremely High'; 'Not Care'};
RBP= {'No Rebate'; 'Some Rebate'; 'Good Rebate'; 'Great Rebate'; 'Not Care'};
FVR= {'Fix Rate'; 'Not Quite Fix'; 'Not Quite Vaiable'; 'Variable'; 'Not Care'};
GF= {'Very Low'; 'Low'; 'Average'; 'High'; 'Very High';'Extremely High'; 'Not Care'};
CF= {'Vey Bad'; 'Bad'; 'Not Bad'; 'Average'; 'Good'; 'Great'; 'Not Care'};
RI= {'Vey Bad'; 'Bad'; 'Not Bad'; 'Average'; 'Good'; 'Great'; 'Not Care'};
FF={'No Frequent Flyer'; 'Some Frequent Flyer'; 'Good Frequent Flyer'; 'Great Frequent Flyer'; 'Not Care'};
CA= {'Very Low'; 'Low'; 'Average'; 'High'; 'Very High';'Extremely High'; 'Not Care'};
RCF = {'Very Low'; 'Low'; 'Average'; 'High'; 'Very High';'Extremely High'; 'Not Care'};
LPF = {'Very Low'; 'Low'; 'Average'; 'High'; 'Very High';'Extremely High'; 'Not Care'};
SI = {'Very Low'; 'Low'; 'Average'; 'High'; 'Very High';'Extremely High'; 'Not Care'};
DO={'No Disbute'; 'Some Disbute'; 'Good Disbute'; 'Great Disbute'; 'Not Care'};
CS = {'Vey Bad'; 'Bad'; 'Not Bad'; 'Average'; 'Good'; 'Great'; 'Not Care'};
SPP= {'No Option'; 'Some Option'; 'Good Option'; 'Great Option'; 'Not Care'};
PP={'No Partner'; 'Some Partner'; 'Good Partner'; 'Great Partner'; 'Not Care'};
IYR ={'Yes', 'No'};
```

Table 3. Variables, Granulation and Information used to create the University Admission System Model.

	EthnicName = {'American'; 'Chinese'; 'French'; 'Greek'; 'Indian'; 'Irish'; 'Italian'; 'Japanese'; 'Mediterranean ';'Persian'; 'Spanish'; 'Taiwanese', 'Not Care'};
% AP : Advanced Placement	Residency={'California Resident'; 'US Resident'; 'International', 'NotCare' };
% IBHL : International Bacculaureat Higher Level (IBHL)	Sex={'Male'; 'Female, 'Not Care' };
% HW: Honors and Awards	Minority={'No'; 'Yes'; 'Not Care'};
% GPA: 12th Grade Courses GPA	HW= {'Few'; 'Some'; 'Lot'; 'Not Care'};
% CP: Course pattern	AAA= {'Kind of Active'; 'Active'; 'Exceptional'; 'Not Care'};
% GPAP: Pattern of Grades through time	CP={'Less Than Required'; 'Required'; 'Recommended'; 'Above Recommendation' };
% SAT II	Concern={'Kind of Concern'; 'Concern'; 'Very Concern'; 'Enthusiast'};
% SAT I	Motivation={'Kind of Motivated'; 'Motivated'; 'Highly Motivated'; 'Enthusiast'};
% CAoSI: Creative Achievement or Sustained Intellectual	IMajor={'Kind of Interested'; 'Interested'; ' Very Interested'; 'Enthusiast'};
% AAsO: Academic Achievement and Outreach	AP= {'Very Low'; 'Low'; 'Medium'; 'High'; 'Very High' };
% ClaCV: Contribution to the intellectual and cultural vitality	IBHL= {'Very Low'; 'Low'; 'Medium'; 'High'; 'Very High' };
% DPBaE: Diversity in the Personal Background and Experience	SATI={'Very Low'; 'Low'; 'Medium'; 'High'; 'Very High' };
% Leadership	SATII={'Very Low'; 'Low'; 'Medium'; 'High'; 'Very High' };
% Motivation	GPA= {'Very Low'; 'Low'; 'Medium'; 'High'; 'Very High' };
% Concern: Concern for Community and others	Employment={'Few'; 'Average'; 'Kind High'; 'High'; 'Lot'};
% AAA: Achievements; Art or Athletics	CAoSAI= {' Low'; 'Kind Low'; 'Average'; 'Kind of High'; 'High'; 'Exceptional' };
% Employment	AAsO={' Low'; 'Kind Low'; 'Average'; 'Kind of High'; 'High'; 'Exceptional'};
% IMajor: Interest in the Major	ClaCV={' Low'; 'Kind Low'; 'Average'; 'Kind of High'; 'High'; 'Exceptional'};
	DPBaE={' Low Diversity'; 'Kind Low Diversity'; 'Diverse'; 'Kind of High Diversity'; 'High Diversity'; 'Exceptional'};
	Leadership={' Low'; 'Kind Low'; 'Average'; 'Kind of High'; 'High'; 'Exceptional' };

based and fuzzy information, the computation in the fuzzy domain can be more powerful and more complex. This section gives a brief overview of various measures of fuzzy association and similarity and various types of aggregation operators involved, along with the description of a simple procedure of utilizing these tools in real applications.

Various definitions of similarity exist in the classical, crisp domain, and many of them can be easily extended to the fuzzy domain. However, unlike in the crisp case, in the fuzzy case the similarity is defined on two fuzzy sets. Suppose we have two fuzzy sets A and B with membership functions $\mu_A(x)$ and $\mu_B(x)$, respectively. Table 4 lists a number of commonly used fuzzy similarity measures between A and B. The arithmetic operators involved in the fuzzy similarity measures can be treated using their usual definitions while the union and the intersection operators need to be treated specially. It is important for these operator pairs to have the following properties: (1) conservation, (2) monotonicity, (3) commutativity, and (4) associativity (cf. Table 5 for the definitions of these properties). It can be verified that the triangular norm (T-norm) and triangular co-norm (T-conorm) (Nikravesh, 2001b; Bonissone and Decker, 1986; Mizumoto, 1989; Fagin, 1998 and 1999) conform to these properties and can be applied here. A detailed survey of some commonly used T-norm and T-conorm pairs will be provided shortly along with other aggregation operators.

Table 4. Measures of Association

Simple Matching Coefficient :	$\lvert A \cap B \rvert$
Dice's Coefficient :	$2\dfrac{\lvert A \cap B \rvert}{\lvert A \rvert + \lvert B \rvert}$
Jaccard's Coefficient :	$\dfrac{\lvert A \cap B \rvert}{\lvert A \cup B \rvert}$
Cosine Coefficient :	$\dfrac{\lvert A \cap B \rvert}{\lvert A \rvert^{1/2} \cdot \lvert B \rvert^{1/2}}$
Overlap Coefficient :	$\dfrac{\lvert A \cap B \rvert}{\min(\lvert A \rvert, \lvert B \rvert)}$
Dissimilarity Coefficient :	$\dfrac{\lvert A \Delta B \rvert}{\lvert A \rvert + \lvert B \rvert} =$
1 – Dice's Coefficient :	$\lvert A \Delta B \rvert = \lvert A \cup B \rvert - \lvert A \cap B \rvert$

While any of the five fuzzy similarity measures can be used in an application, they have different properties. The Simple Matching Coefficient essentially generalizes the inner product and is thus sensitive to the vector length. The Cosine Coefficient is a simple extension to the Simple Matching Coefficient but normalized with respect to the vector lengths. The Overlap Coefficient computes the degree of overlap (the size of intersection) normalized to the size of the smaller of the two fuzzy sets. The Jaccard's Coefficient is an extension to the Overlap Coefficient by using a different normalization. The Dice's Coefficient is yet another extension to the Overlap Coefficient, and both the Jaccard's and Dice's Coefficients are frequently used in traditional information retrieval applications.

In the definition of all five similarity metrics, appropriate aggregation operator pairs are substituted in place of the fuzzy intersection ($\cap$) and fuzzy union operators ($\cup$). As discussed previously, a number of different T-norm and T-conorm pairs are good candidates for this application. There exist many different types of T-norm and T-conorm pairs (Mizumoto, 1989), and they are all functions from $[0,1]x[0,1] \rightarrow [0,1]$ and conform to the list of properties in Table 5. Table 6 shows a number of commonly used T-norm and T-conorm pairs that we consider here. Note that each pair of T-norm and T-conorm satisfies the DeMorgan's law: $\sim T(x,y) = S(\sim x, \sim y)$ where "~" is the negation operator defined by $\sim x = 1-x$.

The minimum and the maximum are the simplest T-norm and T-conorm pair. It can be verified that the minimum is the largest T-norm in the sense that $T(x,y) \leq min(x,y)$ for any T-norm operator T. Similarly, the maximum is the smallest T-conorm. Both the minimum and the maximum are idempotent since $min(x,x)=x$ and $max(x,x)=x$ for any x.

Contrary to the minimum the drastic product produces as small a T-norm value as possible without a violation of the properties in Table 5. Similarly, the drastic sum produces as large a T-conorm value as possible. Thus, the value produced by any other T-norm (T-conorm) operator must lie between the minimum (maximum) and the drastic product (drastic sum).

The bounded difference and its dual, the bounded sum, are sometimes referred to as the Lukasiewicz T-norm and T-conorm. It is important to note that they conform to the law of excluded middle of classific bivalent logic, i.e. $T(x,\sim x)=0$ and $S(x,\sim x)=1$.

The algebraic product and algebraic sum have intuitive interpretations in the probabilistic domain as being the probability of the intersection and the union of two independent events, respectively. In addition, they are smooth functions that are continuously differentiable.

Besides the fixed T-norm and T-conorm pairs described above, there are also a number of parametric T-norm and T-conorm pairs that contain a free parameter

for adjusting the behavior (such as softness) of the operators. A commonly used pair due to Hamacher is defined as: $T(x,y) = xy / [p+(1-p)(x+y-xy)]$ and $S(x,y) = [x+y-xy-(1-p)xy]/[1-(1-p)xy]$ where $p \geq 0$ is free parameter. In particular, we obtain the Hamacher product/sum and the Einstein product/sum (cf. Table 6) by setting p to 0 and 2, respectively.

So far we have introduced several different types of fuzzy association/similarity metrics involving a variety T-norm and T-conorm pairs. An appropriate similarity metric can be selected to compute the distance between two objects according to the requirements of a particular application. In most practical applications we may have to consider more than one attribute when comparing two objects. For example, computing the similarity between two students' academic achievements may require separate comparisons for different subjects, e.g. sciences, mathematics, humanities, etc. Thus, it is useful to have a principled manner for aggregating partial similarity scores between two objects computed on individual attributes. We call such a function an aggregation operator (or simply an aggregator) and define it as a function $f: [0,1]x...x[0,1] \rightarrow [0,1]$.

As for the similarity metric, there are a variety of aggregation operators to choose from, depending on the nature of a particular application (Detyniecki M, 2000). Given our discussion of the T-norm and T-conorm operators, it should not be surprising that many T-norms and T-conorms can be used as aggregation operators. In particular, the associative property (cf. Table 5) of T-norms and T-conorms make them applicable in aggregating more than two values. Intuitively, T-norm aggregators have a minimum-like (or conjunctive) behavior while T-conorms have a maximum-like (or disjunctive) behavior, and these behaviors should be taken into account in selecting an appropriate aggregator to use.

Table 5. Properties of aggregation operators for triangular norms and triangular co-norms.

• ***Conservation*** $t(0,0) = 0; t(x,1) = t(1,x) = x$	• ***Conservation*** $s(1,) = 1; s(x,0) = s(0,x) = x$
• ***Monotonicity*** $t(x_1,x_2) \leq t(x'_1,x'_2)$ *if* $x_1 \leq x'_1$ *and* $x_2 \leq x'_2$	• ***Monotonicity*** $s(x_1,x_2) \leq s(x'_1,x'_2)$ *if* $x_1 \leq x'_1$ *and* $x_2 \leq x'_2$
• ***Commutativity*** $t(x_1,x_2) = t(x_2,x_1)$	• ***Commutativity*** $s(x_1,x_2) = s(x_2,x_1)$
• ***Associativity*** $t(t(x_1,x_2),x_3) = t(x_1,t(x_2,x_3))$	• ***Associativity*** $s(s(x_1,x_2),x_3) = s(x_1,s(x_2,x_3))$

Table 6. Triangular norm/triangular co-norm pairs.

Minimum : $t(x_1, x_2) = \min\{x_1, x_2\}$ *Maximum :* $s(x_1, x_2) = \max\{x_1, x_2\}$
Drastic Product : $t(x_1, x_2) = \begin{cases} \min\{x_1, x_2\} & \text{if } \max\{x_1, x_2\} = 1 \\ 0 & \text{otherwise} \end{cases}$
Drastic sum : $s(x_1, x_2) = \begin{cases} \max\{x_1, x_2\} & \text{if } \min\{x_1, x_2\} = 0 \\ 1 & \text{otherwise} \end{cases}$
Bounded difference : $t(x_1, x_2) = \max\{0, x_1 + x_2 - 1\}$ *Boubded sum :* $s(x_1, x_2) = \min\{1, x_1 + x_2\}$
Einstein product : $t(x_1, x_2) = (x_1 \cdot x_2)/(2 - (x_1 + x_2 - x_1 \cdot x_2))$ *Einstein sum :* $s(x_1, x_2) = (x_1 + x_2)/(1 + x_1 \cdot x_2)$
Algebraic product : $t(x_1, x_2) = x_1 \cdot x_2$ *algebraic sum :* $s(x_1, x_2) = x_1 + x_2 - x_1 \cdot x_2$
Hamacher product : $t(x_1, x_2) = (x_1 \cdot x_2)/(x_1 + x_2 - x_1 \cdot x_2)$ *Hamacher sum :* $s(x_1, x_2) = (x_1 + x_2 - 2x_1 \cdot x_2)/(1 - x_1 \cdot x_2)$

Table 7. Fuzzy-Min and Fuzzy-Max Operators.

Conjunction rule : $\mu_{A \wedge B}(x) = \min\{\mu_A(x), \mu_B(x)\}$
Disjunction rule : $\mu_{A \vee B}(x) = \max\{\mu_A(x), \mu_B(x)\}$
Negation rule : $\mu_{\sim A}(x) = 1 - \mu_A(x)$
$\mu_{A \wedge A}(x) = \mu_A(x)$
$\mu_{A \wedge (B \vee C)}(x) = \mu_{(A \wedge B)}(x) \vee \mu_{(A \wedge C)}(x)$
If : $\mu_A(x) \leq \mu_A(x')$ **AND** $\mu_B(x) \leq \mu_B(x')$ **Then:** $\mu_{A \wedge B}(x) \leq \mu_{A \wedge B}(x')$
If Query (A) and Query (B) are equivalent: $\mu_A(x) = \mu_B(x)$

One of the simplest aggregation operators is the arithmetic mean: $f(x_1,...,x_N) = (x_1+...+x_N)/N$. This simple averaging operator is often considered as the most unbiased aggregator when no further information is available about an application. It is also most applicable when different attributes all have relatively even importance or relevance to the overall aggregated result.

A simple extension of the arithmetic mean, the linearly weighted mean, attaches different weights to the attributes, and is defined by: $f(x_1,...,x_N) = (w_1x_1+...+w_Nx_N)/N$ where $w_1,...,w_N \geq 0$ are linear weights assigned to different attributes and the weights add up to one. The weights can be interpreted as the relative importance or relevance of the attributes and can be specified using domain knowledge or from simple linear regression.

Extension to the arithmetic mean also includes the geometric mean: $f(x_1,...,x_N) = (x_1...x_N)^{1/n}$ which is equivalent to taking the arithmetic mean in the logarithmic domain (with an appropriate exponential scaling), and the harmonic mean: $f(x_1,...,x_N) = n/(1/x_1+...+1/x_N)$ which is particularly appropriate when the x_i's are rates (e.g. units/time). Both geometric mean and harmonic mean also have their weighted versions.

Another family of non-linear aggregation operator involves ordering of the aggregated values. This family includes the median, the k-order statistic, and more generally the ordered weighted average. For N values in ascending order the median is taken to be the $(N+1)/2$'th value if N is odd or the average of the $N/2$ and $N/2+1$'th value if N is even. The k-order statistics generalizes the median operator to take the k'th value, thus including median, minimum, and maximum as special cases. The ordered weighted average (OWA), first introduced by Yager (1988), generalizes both the k-order statistic and the arithmetic mean and is defined as: $f(x_1,...,x_N) = w_1x_{\sigma(1)}+...+w_Nx_{\sigma(N)}$ where w's are non-negative and add up to one, and $x_{\sigma(i)}$ denotes the i'th value of x's in ascending order. By using appropriate weights OWA provide a compromise between the conjunctive behavior of the arithmetic mean and the disjunctive behavior of the k-order statistic.

Finally, it is of interest to include in our discussion a family of aggregators based on fuzzy measures and fuzzy integrals since they subsume most of the aggregators described above. The concept of fuzzy measure was originally introduced by Sugeno (Sugeno, 1974) in the early 1970's in order to extend the classical (probability) measure through relaxation of the additivity property. A formal definition of the fuzzy measure is as follows:

Definition 1. Fuzzy measure: Let X be a non-empty finite set and Ω a Boolean algebra (i.e. a family of subsets of X closed under union and complementation, including the empty set) defined on X. A fuzzy measure, g, is a set function $g:\Omega \rightarrow [0,1]$ defined on Ω, which satisfies the following properties: (1) Boundary

conditions: $g(\phi)=0$, $g(X)=1$. (2) Monotonicity: If $A \subseteq B$, then $g(A) \leq g(B)$. (3) Continuity: If $F_n \in \Omega$ for $1 \leq n < \infty$ and the sequence $\{F_n\}$ is monotonic (in the sense of inclusion), then $lim_{n\to\infty}\, g(F_n) = g(lim_{n\to\infty}\, F_n)$. And (X, Ω, g) is said to be a fuzzy measure space.

To aggregate values with respect to specific fuzzy measures a technique based on the concept of the fuzzy integral can be applied. There are actually several forms of fuzzy integral; for brevity let us focus on only the discrete Choquet integral proposed by Murofushi and Sugeno (1989).

Definition 4 (Choquet) Fuzzy integral: Let (X, Ω, g) be a fuzzy measure space, with $X = \{x_1, \cdots, x_N\}$. Let $h: X \to [0,1]$ be a measurable function. Assume without loss of generality that $0 \leq h(x_1) \leq \cdots \leq h(x_N) \leq 1$, and $A_i = \{x_i, x_{i+1}, \cdots, x_N\}$. The Choquet integral of h with respect to the fuzzy measure g is defined by

$$\int_C h \circ g = \sum_{i=1}^{N} [h(x_i) - h(x_{i-1})]\, g(A_i) \qquad (1)$$

where $h(x_0)=0$.

An interesting property of the (Choquet) fuzzy integral is that if g is a probability measure, the fuzzy integral is equivalent to the classical Lebesgue integral and simply computes the expectation of h with respect to g in the usual probability framework. The fuzzy integral is a form of averaging operator in the sense that the value of a fuzzy integral is between the minimum and maximum values of the h function to be integrated. It can be verified that most of the aggregation operators we have described so far, including the minimum, maximum, median, arithmetic mean, weighted average, k-order statistic, ordered-weighted average, are all special cases of the Choquet fuzzy integral. A distinct advantage of the fuzzy integral as a weighted operator is that, using an appropriate fuzzy measure, the weights represent not only the importance or relevance of individual information sources but also the interactions (redundancy and synergy) among any subset of the sources. However, the representational power of fuzzy integrals and fuzzy measures comes at the expense of having a greater number of free parameters to specify. For N attributes a full specification of fuzzy measures requires 2^N-2 numbers. Alternatives such as using a decomposable k-additive fuzzy measure have been proposed to trade off the number of parameters and the representational power (Grabisch, 1996). Further description of these alternatives, as well as techniques for specifying and learning fuzzy measures, are beyond the scope of this paper and interested readers can refer to (Grabisch et al., 2000).

Having introduced a variety of tools that are required to evaluate fuzzy association/similarity between two objects, a simple algorithm in pseudo code is provided

below to illustrate how these machineries can be used in a practical implementation.

Input: two objects A and B
A: N discrete attributes
For the i^{th} attribute, A^i is an array of length M^i, where M^i is the number of possible linguistic values of the i^{th} attribute.
i.e. each A^i_j, i in 1,...,N and j in 1,...,M^i, gives the degree of A's i^{th} attribute having j^{th} linguistic value.
B: similar to A with the same dimensions.

Other parameters:
AggregatorType
SimilarityType
TNormType
OptionalWeights

Output: An aggregated similarity score between A and B

Algorithm:
For each i=1 to N
SAB^i = ComputeSimilarity(A^i, B^i ,SimilarityType, TNormType)
End
Return Aggregate(SAB, AggregatorType, OptionalWeights)

Sub ComputeSimilarity(X, Y, SimilarityType, TNormType)
Switch SimilarityType:
Case SimpleMatchingCoefficient:
Return $|X \cap Y|$

Case CosineCoefficient:
Return $|X \cap Y| / (|X|^{1/2} |Y|^{1/2})$

Case OverlapCoefficient:
Return $|X \cap Y| / \min(|X|, |Y|)$

Case Jaccard's Coefficient:
Return $|X \cap Y| / (|X \cup Y|)$

Case Dice's Coefficient:
Return $2|X \cap Y| / (|X| + |Y|)$
...
End

Sub Aggregate(S, AggregatorType, OptionalWeights)

```
        Switch AggregatorType:
        Case Min:
                Return min(S)
        Case Max:
                Return max(S)
        Case Mean:
                Return mean(S)
        Case Median:
                Return median(S)
        Case WeightedAverage:
                Return WeightedAverage(S, OptionalWeights)
        Case OrderedWeightedAverage:
                Return OrderedWeightedAverage(S, OptionalWeights)
        Case ChoquetIntegral:
                Return ChoquetIntegral(S, OptionalWeights)
        Case SugenoIntegral:
                Return SugenoIntegral(S, OptionalWeights)
        ...
End
```

This algorithm takes as input two objects, each with *N* discrete attributes. Similarity scores between the two objects are first computed with respect to each attribute separately, using a specified similarity metric and T-norm/conorm pair. As described previously, the computation of a similarity score with respect to an attribute involves a pair wise application of the T-norm or T-conorm operators on the possible values of the attribute, followed by other usual arithmetic operation specified in the similarity metric. Finally, an aggregation operator with appropriate weights is used to combine the similarity measures obtained with respect to different attributes.

In many situations, the controlling parameters, including the similarity metric, the type of T-norm/conorm, the type of aggregation operator and associated weights, can all be specified based on the domain knowledge of a particular application. However, in some other cases, it may be difficult to specify a priori an optimal set of parameters. In those cases, various machine learning methods can be employed to automatically "discover" a suitable set of parameters using a supervised or unsupervised approach. For example, the Genetic Algorithm (GA) and DNA-based computing, as described in later sections, can be quite effective.

2.2 Precisions and Recall Measure

Table 8 and ***Figure 1*** show the definition of precision, recall and their relationship. Given a user's criteria, the data provided for modeling, and the strategy de-

fined in *Figure 2*, the recall/precision relationship has been optimized. Therefore, a user will get better precision and recall in fuzzy or imprecise situations.

Table 8. Measures of Precision, Recall and several other relevant attributes.

$Precision: P = \frac{\lvert A \cap B \rvert}{\lvert B \rvert}$
$Recall: R = \frac{\lvert A \cap B \rvert}{\lvert A \rvert}$
$Fallout: F = \frac{\lvert \bar{A} \cap B \rvert}{\lvert \bar{A} \rvert}$
$Generality: G = \frac{\lvert A \rvert}{N}$
$Retrieved/Relevent: A \cap B$
$Retrieved/Non-Relevent: \bar{A} \cap B$
$Not-Retrieved/Relevent: A \cap \bar{B}$
$Not-Retrieved/Not-Relevent: \bar{A} \cap \bar{B}$

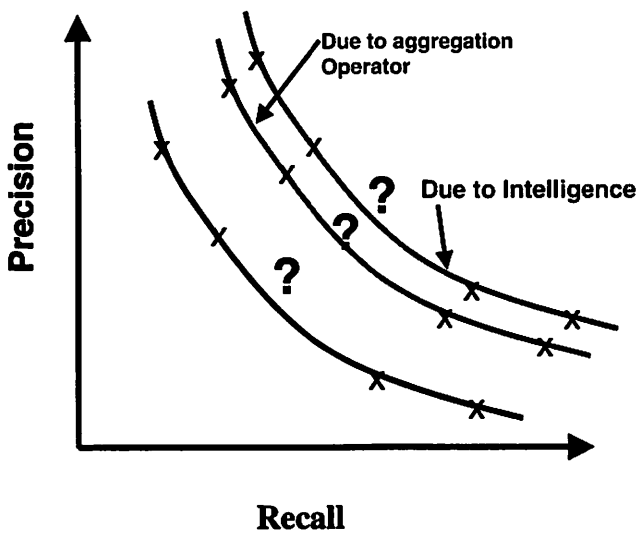


Figure 1. Inverse relationship between Precision and Recall.

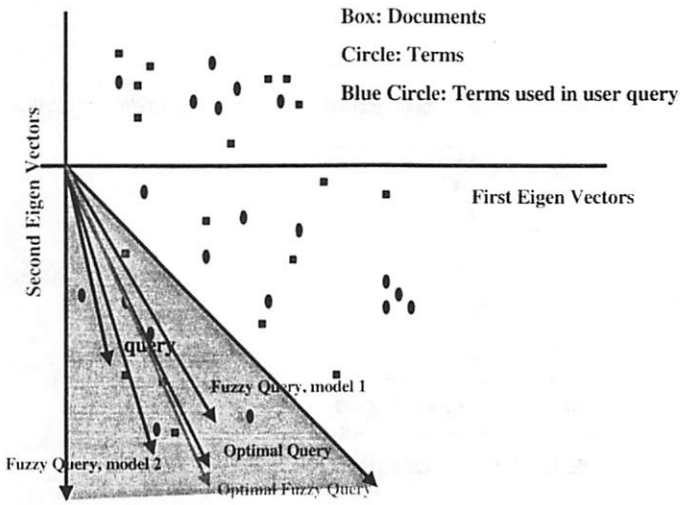


Figure 2. Schematic diagram of the performance of the Fuzzy-Latent Semantic Indexing method.

2.3 Search Strategy

There are several ways to search and query in databases such as *Latent Semantic Indexing (LSI),* full text scanning, inversion, and the use of signature files. While *LSI* has limitations, it is highly rewarding, since it is easy to implement and update; it is fast; it works in a reduced domain; it is scaleable; and it can be used for parallel processing. One solution to its Boolean model is to use an extended Boolean model or fuzzy logic. In this case, one can add a fuzzy quantifier to each term or concept. In addition, one can interpret the AND as a fuzzy-MIN function and the OR as a fuzzy-MAX function respectively.

The most straightforward way to search is ***full text scanning***. The technique is simple to implement; has no space overhead; minimal effort on insertion or update is needed; a finite state automaton can be built to find a given query; and Boolean expressions can be used as query resolution. However, the algorithm is too slow.

The ***inversion*** method is the most suitable techniques followed by almost all commercial systems (if no semantics are needed).It is easy to implement and fast. However, storage overhead is up to 300% and updating the index for dynamic systems and merging of lists are costly actions. In this study, in addition to inversion techniques, ***Fuzzy-Latent Semantic Indexing (FLSI)*** originally developed for text

retrieval has been used (Nikravesh, 2001a and 2001b). ***Figure 2*** shows a schematic diagram of the performance of ***FLSI***. ***Figure 3*** and ***Figure 4*** show the performance of ***FLSI*** for text retrieval purposes. The following briefly describes the **FLSI** technique (Nikravesh, 2001a and 2001b):

Fuzzy-based decompositions are used to approximate the matrix of document vectors.

Terms in the document matrix may be presented using linguistic terms (or fuzzy terms such as most likely, likely, etc) rather than frequency terms or crisp values.

Decompositions are obtained by placing a fuzzy approximation onto the eigensubspace spanned by all the fuzzy vectors.

Empirically, we establish our technique such that the approximation errors of the fuzzy decompositions are close to the best possible; namely, to truncated singular value decompositions.

The followings are the potential applications of the FLSI:

1. *Search Engines:* The recent explosion of online information on the World Wide Web has given rise to a number of query-base search engines. However, this information is useless unless it can be effectively and efficiently searched.

2. *Fuzzy Queries in Multimedia Database Systems:* Even though techniques exist for locating exact matches for traditional database, finding relevant partial matches for Multimedia database systems might be a problem. It may not be also easy to specify query requests precisely and completely - resulting in a situation known as a fuzzy-querying.

3. *Query Based on User Profile:* It employs as combinations of technologies that take the result of the queries and organize them into categories for presentation to the user. The system can then save such document organizations in user profiles, which can then be used to help classify future query results by the same user.

4. *Information Retrievals:* The goal in information retrieval is to find documents that are relevant to a given user query.

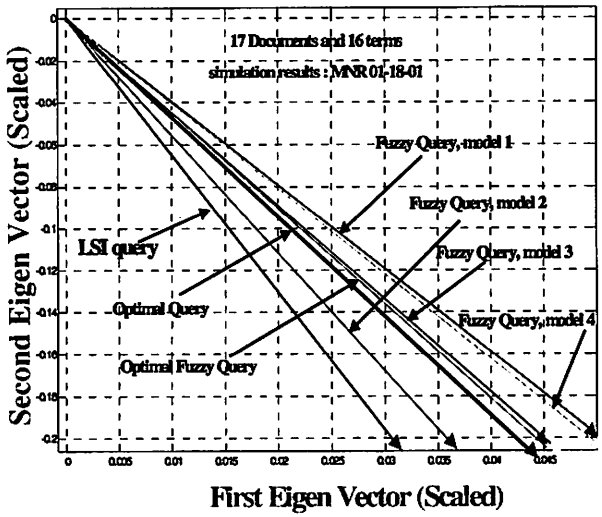


Figure 3. Example 1 of FLSI for text retrieval.

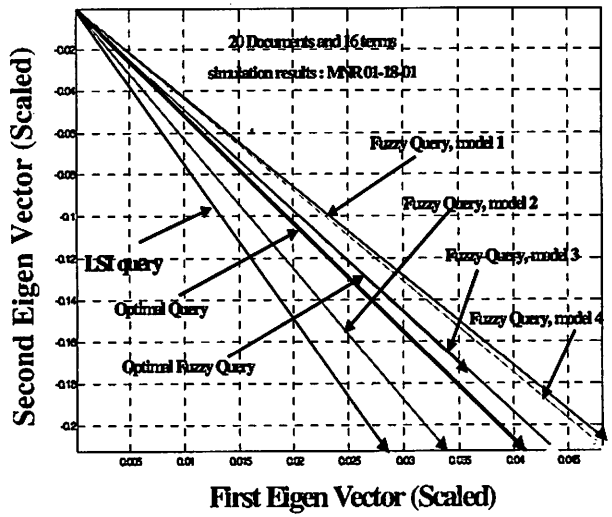


Figure 4. Example 2 of FLSI for text retrieval.

5. *Summary of Documents:* Human-quality text summarization systems are difficult to design, and even more difficult to evaluate, in part because documents can differ along several dimensions, such as length, writing style and lexical usage.

    - Text Summarization-Single Document
    - Text Summarization-Multi Documents

    Multi-document summarization differs from single in that the issues of compression, speed, redundancy and passage selection are critical in the formation of useful summaries.

6. *Information Fusion Such as Medical Records, Research Papers, News, etc:* Two groups of database or News are generated independently of each other, quantified the same n terms in the same m documents. The documents or NEWS form the two groups are similar but not necessarily identical. We are interested in merging documents or NEWS.

7. *File and Folder Organiser:* Organizers operate on data matrix (e.g., terms X file or folder; names or date X file or folder; etc.) to derive similarities, degree of match, clusters, and derive rules.

8. *Matching People*: Matching People operate on data matrix (e.g., Interests X People; Articles X people; etc.) to derive similarities and degree of match.

9. *Association Rule Mining for Terms-Documents:* Association Rule Mining algorithm operates on data matrix (e.g., Terms X Documents) to derive rules.

    i) Documents Similarity; Search Personalization-User Profiling. *Often time it is hard to find the "right" term and even in some cases the term does not exist.* The User Profile is

automatically constructed from text document collection and can be used for Query Refinement and provide suggestions and for ranking the information based on pre-existence user profile.

ii) Terms Similarity; Automated Ontology Generation and Automated Indexing The ontology is automatically constructed from text document collection and can be used for Query Refinement.

10. *E-mail Notification:* E-mail notification whenever new matching documents arrive in the database, with link directly to documents or sort incoming messages in right mailboxes

11. *Modelling Human Memory:* The Technique can be used in some degree to model some of the associative relationships observed in human memory abased on term-term similarities.

12. *Calendar Manager*: automatically schedule meeting times.

13. *Others:* Telephony, Call Center, Workgroup Messages, E-Mail, Web-Mail, Personal Info, Home-Device Automation, etc.

2.4 Intelligent Data Mining: Fuzzy- Evolutionary Computing (Nikravesh 2002, 2003a, and 2003b and Loia et al. 2003)

2.4.1. Pattern Recognition

In the 1960s and 1970s, pattern recognition techniques were used only by statisticians and were based on statistical theories. Due to recent advances in computer systems and technology, artificial neural networks and fuzzy logic models have

been used in many pattern recognition applications ranging from simple character recognition, interpolation, and extrapolation between specific patterns to the most sophisticated robotic applications. To recognize a pattern, one can use the standard multi-layer perceptron with a back-propagation learning algorithm or simpler models such as self-organizing networks (Kohonen, 1997) or fuzzy c-means techniques (Bezdek, 1981; Jang and Gulley, 1995). Self-organizing networks and fuzzy c-means techniques can easily learn to recognize the topology, patterns, and distribution in a specific set of information.

2.4.2 Clustering

Cluster analysis encompasses a number of different classification algorithms that can be used to organize observed data into meaningful structures. For example, k-means is an algorithm to assign a specific number of centers, k, to represent the clustering of N points (k<N). These points are iteratively adjusted so that each point is assigned to one cluster, and the centroid of each cluster is the mean of its assigned points.

In general, the k-means technique will produce exactly k different clusters of the greatest possible distinction. Alternatively, fuzzy techniques can be used as a method for clustering. Fuzzy clustering partitions a data set into fuzzy clusters such that each data point can belong to multiple clusters. Fuzzy c-means (FCM) is a well-known fuzzy clustering technique that generalizes the classical (hard) c-means algorithm and can be used where it is unclear how many clusters there should be for a given set of data. Subtractive clustering is a fast, one-pass algorithm for estimating the number of clusters and the cluster centers in a set of data. The cluster estimates obtained from subtractive clustering can be used to initialize iterative optimization-based clustering methods and model identification methods.

In addition, the self-organizing map technique known as Kohonen's self-organizing feature map (Kohonen, 1997) can be used as an alternative for clustering purposes. This technique converts patterns of arbitrary dimensionality (the pattern space) into the response of one- or two-dimensional arrays of neurons (the feature space). This unsupervised learning model can discover any relationship of interest such as patterns, features, correlations, or regularities in the input data, and translate the discovered relationship into outputs.

2.4.3 Mining and Fusion of Data

In the past, classical data processing tools and physical models solved many real-world complex problems. However, this should not obscure the fact that the world of information processing is changing rapidly. Increasingly we are faced on the one hand with more unpredictable and complex real-world, imprecise, chaotic, multi-dimensional and multi-domain problems with many interacting parameters in situations where small variability in parameters can change the solution completely. On the other hand, we are faced with profusion and complexity of computer-generated data. Unfortunately, making sense of these complex, imprecise and chaotic data which are very common in Engineering and science applications, is beyond the scope of human ability and understanding. What this implies is that the classical data processing tools and physical models that have addressed many complex problems in the past may not be sufficient to deal effectively with present and future needs.

Tables 9 and **10** show the list of the Data Fusion (dominated by Integration process) and Data Mining techniques (Dominated by Interpretation process)

Table 9. Data Mining Techniques (Interpretation)

Deductive Database Client
Inductive Learning
Clustering
Case-based Reasoning
Visualization
Statistical Package

Table 10. Data Fusion Techniques (Integration)

Deterministic

-Transformation based (projection, ...)
-Functional evaluation based (vector quantization, ...)
-Correlation based (pattern match, if/then productions)
-Optimization based (gradient-based, feedback, LDP, ...)
Non-deterministic

-Hypothesis testing (classification, ...)
-Statistical estimation (maximum likelihood, ...)
-Discrimination function (linear aggregation, ...)
-Neural network (supervised learning, clustering, ...)
-Fuzzy Logic (fuzzy c-mean clustering, ...)
-Hybrid (genetic algorithm, Bayesian network, ...)

2.4.4 Intelligent Information Processing

In conventional information processing technique, once all the pertinent data is properly fused, one has to extract the relevant information from the data and draw the necessary conclusions. This can be done either true reliance on human expert or an intelligent system that has the capability to learn and modify its knowledge base as new information become available. In intelligent information processing techniques, the process of information fusion is an integrated part of the information mining. **Table 11** shows the comparison between Conventional and intelligent techniques for information processing.

Table 11. Conventional Vs. Intelligent

Conventional ----------
-Data assumption: a certain probability distribution -Model: weight functions come from varigram trend and probability constraints -Simulation: Stochastic, not optimized
Intelligent ----------
-Data automatic clustering and expert-guided segmentation -Classification of relationship between data and targets -Model: weight functions come from supervised training based on initial known information Simulation: optimized by GA, SA, ANN, and BN

2.4.5 Data Mining

Data Mining or "classification to explore a dataset" is a trend in clustering techniques in which the user has no or little prior assumptions about the data, but wants to explore if data or subset of data falls into "meaningful group" (a term for which the user may not even have a specific definition). Many clustering and data mining algorithm assume a certain type of input such as numerical (in case of k-means) or categorical input. In addition, most techniques either use a prior knowl-

edge to define distance or similarity measure or use probabilistic techniques which break down as the dimensionality of the corresponding feature space increases. It is also require a prior knowledge about the problem domain to fix the number and starting points in which it is clearly not accessible easily where the number of input pararemeters are very large in hyperspace. Finally the clustering problem is an optimization problem and is known to be NP-hard problem.

When data is imprecise and has mix nature (numerical and categorical) and several objectives to be matched at the same time, the optimization problem may be more complex and will fall into Multi-Objective and Multi-Criteria with conflicting objectives which in this case, the conventional techniques could not be applied.

A unified approach based on soft computing will help fill the existing technology gap and is bound to play a key role in solving the above problems. Soft computing is consortium of computing methodologies (Fuzzy Logic (GL), Neuro Computing (NC), Genetic Computing (GC), and Probabilistic Reasoning (PR) including ; Genetic Algorithms (GA), Chaotic Systems (CS), Belief Networks (BN), Learning Theory (LT)) which collectively provide a foundation for the Conception, Design and Deployment of Intelligent Systems. Among main components of soft computing are the artificial neural computing, fuzzy logic computation, and the evolutionary computing.

The intelligent computing techniques will establish a unified framework to solve the above challenges using Soft Computing Techniques (SCT) to utilize the specific strength of each method to address different aspects of the problem. Fuzzy Logic ideal for handling subjective and imprecise information, uncertainty management and knowledge integration. Neural network powerful tool for self-learning and data integration and does not require specification of structural relationships between the input and output data. Evolutionary Computing is effective for handling scale problems, dynamic updating, for pattern extraction, reduce the complexity of the neuro-fuzzy model, and robust optimization along the multidimensional, highly nonlinear and non-convex search hyper-surfaces.

Motivated by current advances in DNA computing which has been showed promises toward solving complex problem including "NP-complete" problems such as Hamiltonian path problem and Satisfiability Problem with ability to pursue an unbounded number of independent computational searches in parallel, we will use Artificial DNA computing to solve the optimization problem.

2.4.6. Genetic Algorithm

Genetic algorithm (GA) is one of the stochastic optimization methods which is simulating the process of natural evolution. GA follows the same principles as those in nature (survival of the fittest, Charles Darwin).

GA first was presented by John Holland as an academic research. However, today GA turns out to be one of the most promising approaches for dealing with complex systems which at first nobody could imagine that from a relative modest technique. GA is applicable to multi-objectives optimization and can handle conflicts among objectives. Therefore, it is robust where multiple solutions exist. In addition, it is highly efficient and it is easy to use.

Another important feature of GA is its capability of extraction of knowledge or fuzzy rules. GA is now widely used and applied to discovery of fuzzy rules. However, when the data sets are very large, it is not easy to extract the rules.

2.4.7 DNA Computing: Intelligent Data Mining Techniques

To overcome such a limitation, a new coding technique is needed. Motivated by current advances in DNA computing which has been showed promises toward solving complex problem including "NP-complete" problems such as Hamiltonian path problem and Satisfiability Problem with ability to pursue an unbounded number of independent computational searches in parallel, we will use a new coding method based on biological DNA and Artificial DNA computing to solve the optimization problem.

The DNA can have many redundant parts which is important for extraction of knowledge. In addition, this technique allows overlapped representation of genes and it has no constraint on crossover points. Also, the same type of mutation can be applied to every locus. In this technique, the length of chromosome is variable and it is easy to insert and/or delete any part of DNA chromosomes. Since the length of the chromosome in artificial DNA coding is variable, it will be very easy to include genetic operations such as virus and enzyme operations. This process and the overlap and redundancy of genes will give the genes the ability to adapt, which increases the chance of survival of genes far beyond the lifetime of individuals.

Artificial DNA algorithm can be used in a hierarchical fuzzy model for pattern extraction and to reduce the complexity of the neuro-fuzzy models. In addition, artificial DNA can be use to extract the number of the membership functions required for each parameter and input variables.

The DNA coding method and the mechanism of development from artificial DNA are suitable for knowledge extraction including fuzzy IF ...THEN from large data set for Data Mining purposes. The rules are extracted from DNA chromosomes as follows. Each artificial amino acid has several meaning. The meaning of genes is determined by the combination of the amino acids. Each amino acid can be translated into an input variable and its membership function. A sequence of amino acids (one genes) corresponds to one fuzzy rule. The Artificial DNA chromosomes having several genes make up a set of fuzzy rules. Each rule represent a subset of data. Therefore, not only data will be mined and clustered but also will be translated into factual knowledge given the linguistic nature of the IF ... THEN rules. This will give a new ability to the user such that the rules based on factual knowledge (data) and knowledge drawn from human experts (inference) will be combined, ranked, and clustered based on the confidence level of human and factual support. This will effectively provide validation of an interpretation, a model, a hypothesis, or alternatively indicate a need for rejection or reevaluation. This will also provide the ability to answer "What if?" questions in order to decrease uncertainty during the process of data Mining and knowledge extraction.

We claim that Fuzzy- artificial DNA can be used for robust optimization along the multidimensional, highly nonlinear and non-convex search hyper-surfaces, generalize its estimation through evolution and manage the uncertainty through fuzzy based technique, even though the environment may partially observable.

The main features of the new methodologies are:

- It uses minimal prior knowledge with respect to the input structure of data and its probability distribution
- Minimal a prior knowledge require about the problem domain to fix the number and starting points
- Can be used to solve optimization problems known as NP-hard problem.
- Can be used when data is imprecise and has mix nature (numerical and categorical)
- Can be used when several objectives to be matched at the same time
- Can be used for Multi-Objective and Multi-Criteria optimization with conflicting objectives
- Scalability/parallel processing
- Can be used for high dimensionality in the feature space with respect to data/problem-space (sparse-data)
- Can extract both the cluster and association rules given certain objective

3 Implementation - Fuzzy Query and Ranking

In this section, we introduce fuzzy query and fuzzy aggregation for credit scoring, credit card ranking, and university admissions.

3.1 Application to Credit Scoring

Credit scoring was first developed in the 1950's and has been used extensively in the last two decades. In the early 1980's, the three major credit bureaus, Equitax, Experian, and TransUnion worked with the Fair Isaac Company to develop generic scoring models that allow each bureau to offer an individual score based on the contents of the credit bureau's data. FICO is used to make billions of financing decisions each year serving a 100 billion dollar industry. Credit scoring is a statistical method to assess an individual's credit worthiness and the likelihood that the individual will repay his/her loans based on their credit history and current credit accounts. The credit report is a snapshot of the credit history and the credit score is a snapshot of the risk at a particular point in time. Since 1995, this scoring system has made its biggest contribution in the world of mortgage lending. Mortgage investors such as Freddie Mac and Fannie Mae, the two main government-chartered companies that purchase billion of dollars of newly originated home loans annually, endorsed the Fair Isaac credit bureau risk, ignored subjective considerations, but agreed that lenders should also focus on other outside factors when making a decision.

When you apply for financing, whether it's a new credit card, car or student loan, or a mortgage, about 40 pieces of information from your credit card report are fed into a model (***Table 1***). This information is categorized into the following five categories with different level of importance (% of the score):

- Past payment history (35%)
- Amount of credit owed (30%)
- Length of time credit established (15%)
- Search for and acquisition of new credit (10%)
- Types of credit established (10%)

When a lender receives your Fair Isaac credit bureau risk score, up to four "score reason codes" are also delivered. These explain the reasons why your score

was not higher. Followings are the most common given score reasons (Fair Isaac);

- Serious delinquency
- Serious delinquency, and public record or collection filed
- Derogatory public record or collection filed
- Time since delinquency is too recent or unknown
- Level of delinquency on accounts
- Number of accounts with delinquency
- Amount owed on accounts
- Proportion of balances to credit limits on revolving accounts is too high
- Length of time accounts have been established
- Too many accounts with balances

By analyzing a large sample of credit file information on people who recently obtained new credit, and given the above information and that contained in Table 1, a statistical model has been built. The model provides a numerical score designed to predict your risk as a borrower. Credit scores used for mortgage lending range from 0 to 900 (usually above 300). The higher your score, the less risk you represent to lenders. Most lenders will be happy if your score is 700 or higher. You may still qualify for a loan with a lower score given all other factors, but it will cost you more. For example, given a score of around 620 and a $25,000 car loan for 60 months, you will pay approximately $4,500 more than with a score of 700. You will pay approximately $6,500 more than if your score is 720. Thus, a $25,000 car loan for 60 months with bad credit will cost you over $10,000 more for the life of the loan than if you have an excellent credit score.

Given the factors presented earlier and the information provided in ***Table 1***, a simulated model has been developed. A series of excellent, very good, good, not good, not bad, bad, and very bad credit scores have been recognized (without including history). Then, fuzzy similarity and ranking have been used to rank the new user and define his/her credit score. ***Figure 5*** shows the simplified flow diagram and flow of information for PNL-Based Fuzzy Query. In the inference engine, the rules based on factual knowledge (data) and knowledge drawn from human experts (inference) are combined, ranked, and clustered based on the confidence level of human and factual support. This information is then used to build the fuzzy query model with associated weights. In the query level, an intelligent knowledge-based search engine provides a means for specific queries. Initially we blend traditional computation with fuzzy reasoning. This effectively provides validation of an interpretation, model, hypothesis, or alternatively, indicates the need to reject or reevaluate. Information must be clustered, ranked, and translated to a format amenable to user interpretation.

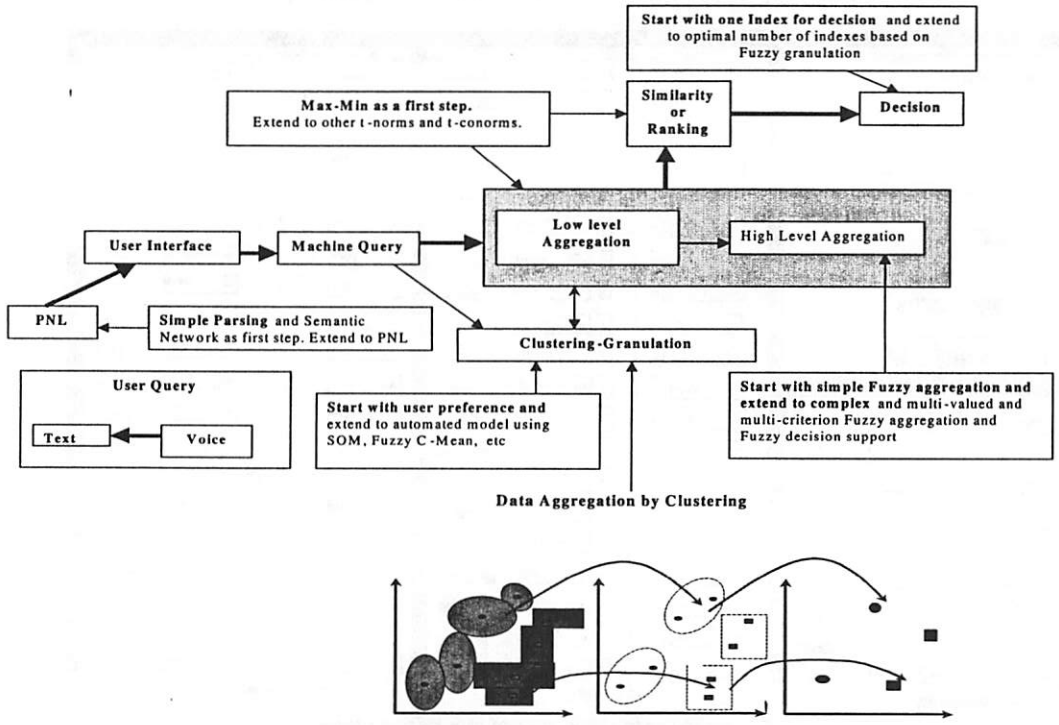


Figure 5. Simplified flow diagram and flow of information for PNL-Based Fuzzy Query.

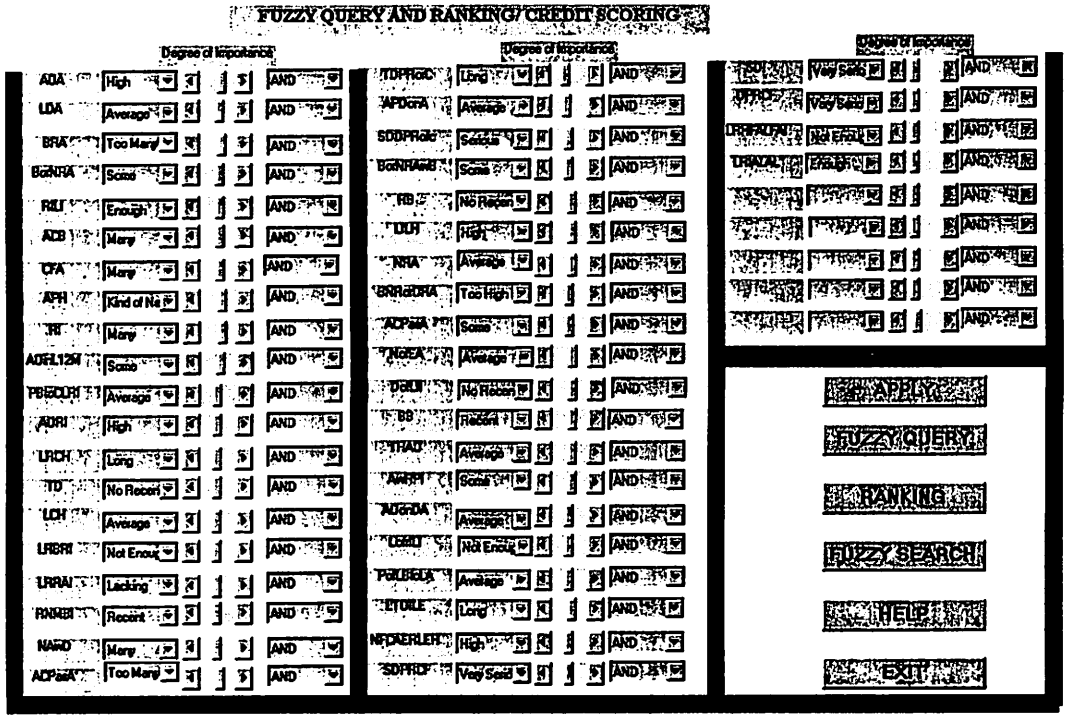


Figure 6. A snapshot of the software developed for credit scoring.

Figure 6 shows a snapshot of the software developed for credit scoring. *Table 1* shows the granulation of the variables that has been used for credit scoring/ranking. To test the performance of the model, a demo version of the software is available at: http://zadeh.cs.berkeley.edu/ (Nikravesh, 2001a). Using this model, it is possible to have dynamic interaction between model and user. This provides the ability to answer "What if?" questions in order to decrease uncertainty, to reduce risk, and to increase the chance to increase a score.

3.2 Application to Credit Card Ranking

Credit ratings that are compiled by the consumer credit organization such as the U.S. Citizens for Fair Credit Card Terms (CFCCT) (U.S Citizens for Fair Credit Card Terms) could simply save you hundreds of dollars in credit card interest or help you receive valuable credit card rebates and rewards including frequent flyer miles (free airline tickets), free gas, and even hundreds of dollars in cash back bonuses.

CFCCT has developed an objective-based method for ranking credit cards in US. In this model, interest rate has the highest weighting in the ranking formula. FCC rates credit cards based on the following criteria (U.S Citizens for Fair Credit Card Terms):

- Purchase APR
- Cash Advance APR
- Annual Fees
- Penalty for cards that begin their grace periods at the time of purchase/posting instead of at the time of billing
- Bonuses for cards that don't have cash advance fees
- Bonuses for cards that limit their total cash advance fees to $10.00
- Bonuses for introductory interest rate offers for purchases and/or balance transfers
- Bonuses for cards that have rebate/perk programs
- Bonuses for cards that have fixed interest rates.

Table 12. Credit cards ranked by the CFCCT.

Classic Cards	Type	Gold Cards	Type	Platinum Cards	Type
Pulaski B& T	V	Pulaski	MC	Capital One	VP
Ark. Natl	MC/V	Capital One	VP	NextCard	VP
Capital One	V	SFNB	V	BofA	VP
NextCard	V	NextCard	V	Simmons	VP
Wachovia	V	BofA	V	G&L Bank	MCP/VP
MCP/VPBlue	AMEX	Wachovia	V	Aria	VP
Helena Natl	MC/V	Blue	AMEX	Ever	VP
Simmons	V	Helena	MC/V	Blue	AMEX
Metro. Natl.	V	Simmons	V	AF	VP
Umbrella	V	Metro.	V	Banco	VP

V=Visa; MC=MasterCard; AMEX=American Express

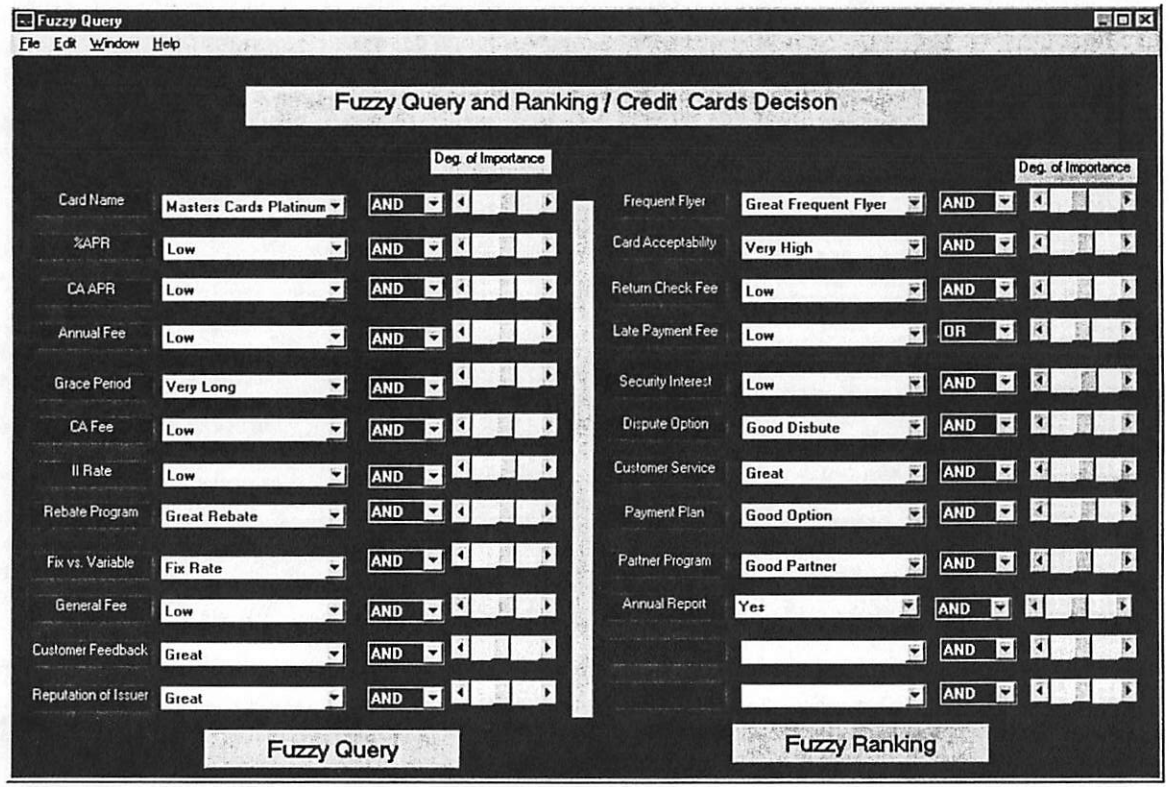


Figure 7. A snapshot of the software developed to rank credit cards.

Table 12 shows the top 10 classic cards, the top 10 gold cards, and the top 10 platinum cards which have been ranked by the CFCCT method (U.S Citizens for Fair Credit Card Terms) as of March 2001. Given the above factors and the information provided in ***Table 8***, a simulated model has been developed. A series of excellent, very good, good, not good, not bad, bad, and very bad credit cards have been recognized for the credit cards listed in ***Table 9***. Then, fuzzy similarity and ranking has been used to rank the cards and define a credit score. ***Figure 7*** shows a snapshot of the software developed to rank credit cards. ***Table 2*** shows the granulation of the variables that has been used for the rankings. To test the performance of the model, a demo version of the software is available at: http://zadeh.cs.berkeley.edu/ (Nikravesh, 2001a).

3.3 University Admissions

Hundreds of millions of applications were processed by U.S. universities resulting in more than 15 million enrollments in the year 2000 for a total revenue of over $250 billion. College admissions are expected to reach over 17 million by the year 2010, for total revenue of over $280 billion. In Fall 2000, UC Berkeley was able to admit about 26% of the 33,244 applicants for freshman admission (University of California-Berkeley). In Fall 2000, Stanford University was only able to offer admission to 1168 men from 9571 applications (768 admitted) and 1257 women from 8792 applications (830 admitted), a general admit rate of 13% (Stanford University Admission).

The UC Berkeley campus admits its freshman class on the basis of an assessment of the applicants' high school academic performance (approximately 50%) and through a comprehensive review of the application including personal achievements of the applicant (approximately 50%) (University of California-Berkeley). For Fall 1999, the average weighted GPA of an admitted freshman was 4.16, with a SAT I verbal score range of 580-710 and a SAT I math score range of 620-730 for the middle 50% of admitted students (University of California-Berkeley). While there is no specific GPA for UC Berkeley applicants that will guarantee admission, a GPA of 2.8 or above is required for California residents and a test score total indicated in the University's Freshman Eligibility Index must be achieved. A minimum 3.4 GPA in A-F courses is required for non-residents. At Stanford University, most of the candidates have an un-weighted GPA between 3.6 and 4.0 and verbal SAT I and math SAT I scores of at least 650 (Stanford University Admission) At UC Berkeley, the academic assessment includes student's academic performance and several measured factors such as:

- College preparatory courses
- Advanced Placement (AP)

- International Baccalaureate Higher Level (IBHL)
- Honors and college courses beyond the UC minimum and degree of achievement in those courses
- Uncapped UC GPA
- Pattern of grades over time
- Scores on the three required SAT II tests and the SAT I (or ACT)
- Scores on AP or IBHL exams
- Honors and awards which reflect extraordinary, sustained intellectual or creative achievement
- Participation in rigorous academic enrichment
- Outreach programs
- Planned twelfth grade courses
- Qualification for UC Eligibility in the Local Context

All freshman applicants must complete courses in the University of California's A-F subject pattern and present scores from SAT I (or ACT) and SAT II tests with the following required subjects:

a. History/Social Science - 2 years required
b. English - 4 years required
c. Mathematics - 3 years required, 4 recommended
d. Laboratory Science - 2 years required, 3 recommended
e. Language Other than English - 2 years required, 3 recommended
f. College Preparatory Electives - 2 years required

At Stanford University, in addition to the academic transcript, close attention is paid to other factors such as student's written application, teacher references, the short responses and one-page essay (carefully read for quality, content, and creativity), and personal qualities.

The information provided in this study is a hypothetical situation and does not reflect the current UC system or Stanford University admissions criteria. However, we use this information to build a model to represent a real admissions problem. For more detailed information regarding University admissions, please refer to the University of California-Berkeley and Stanford University, Office of Undergraduate Admission (University of California-Berkeley; Stanford University Admission).

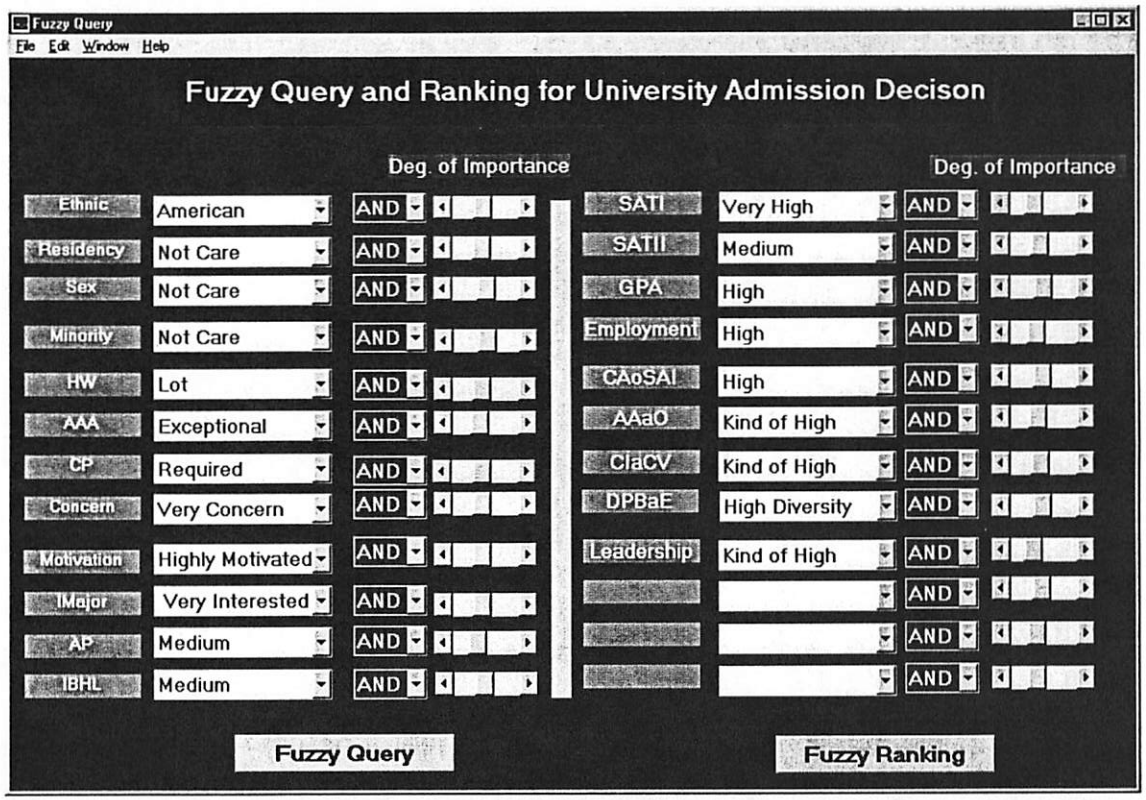


Figure 8. A snapshot of the software for University Admission Decision Making.

Given the factors above and the information contained in ***Table 3***, a simulated-hypothetical model (a Virtual Model) was developed. A series of excellent, very good, good, not good, not bad, bad, and very bad student given the criteria for admission has been recognized. These criteria over time can be modified based on the success rate of students admitted to the university and their performances during the first, second, third and fourth years of their education with different weights and degrees of importance given for each year. Then, fuzzy similarity and ranking can evaluate a new student rating and find it's similarity to a given set of criteria.

Figure 8 shows a snapshot of the software developed for university admissions and the evaluation of student applications. Table 3 shows the granulation of the variables that was used in the model. To test the performance of the model, a demo version of the software is available at: http://zadeh.cs.berkeley.edu/ (Nikravesh, 2001a). Incorporating an electronic intelligent knowledge-based search engine, the results will eventually be in a format to permit a user to interact dynamically with the contained database and to customize and add information to the database. For instance, it will be possible to test an intuitive concept by dynamic interaction between software and the human mind.

This will provide the ability to answer "What if?" questions in order to decrease uncertainty and provide a better risk analysis to improve the chance for "increased success" on student selection or it can be used to select students on the basis of "diversity" criteria. The model can be used as for decision support and for a more uniform, consistent and less subjective and biased way. Finally, the model could learn and provide the mean to include the feedback into the system through time and will be adapted to the new situation for defining better criteria for student selection.

In this study, it has been found that ranking and scoring is a very subjective problem and depends on user perception (***Figure 9 and Figure 10***) and preferences in addition to the techniques used for the aggregation process which will effect the process of the data mining in reduced domain (***Figure 11***). Therefore, user feedback and an interactive model are recommended tools to fine-tune the preferences based on user constraints. This will allow the representation of a multi-objective optimization with a large number of constraints for complex problems such as credit scoring or admissions. To solve such subjective and multi-criteria optimization problems, GA-fuzzy logic and DNA-fuzzy logic models [2] are good candidates.

In the case of the GA-Fuzzy logic model, the fitness function will be defined based on user constraints. For example, in the admissions problem, assume that we would like to select students not only on the basis of their achievements and criteria defined in Table 3, but also on the basis of diversity which includes gender distribution, ethnic background distribution, geophysical location distribution, etc.

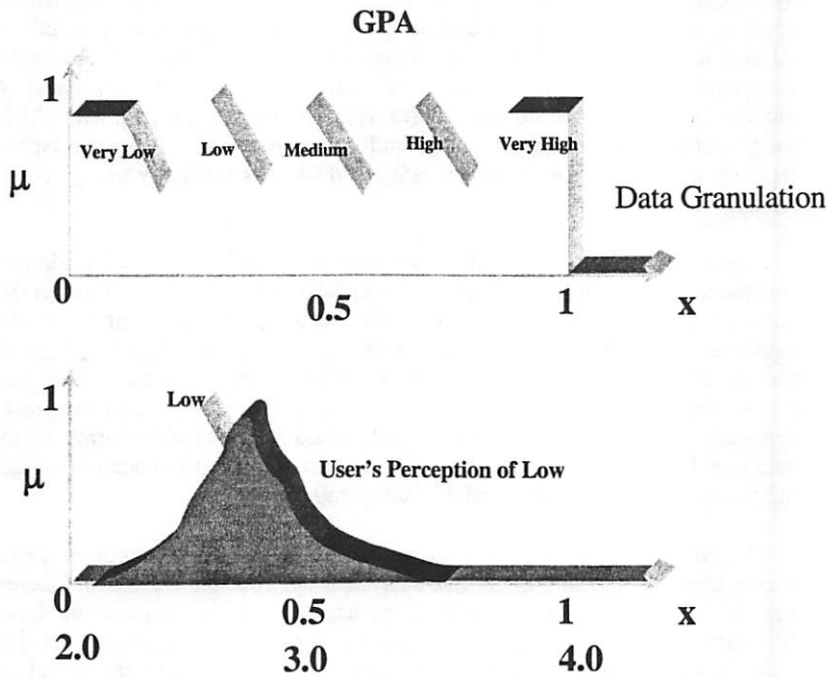


Figure 9. User's perception of "GPA Low"

The question will be "what are the values for the preferences and which criteria should be used to achieve such a goal?" In this case, we will define the genes as the values for the preferences and the fitness function will be defined as the degree by which the distribution of each candidate in each generation match the desired distribution. fuzzy similarity can be used to define the degree of match which can be used for better decision analysis.

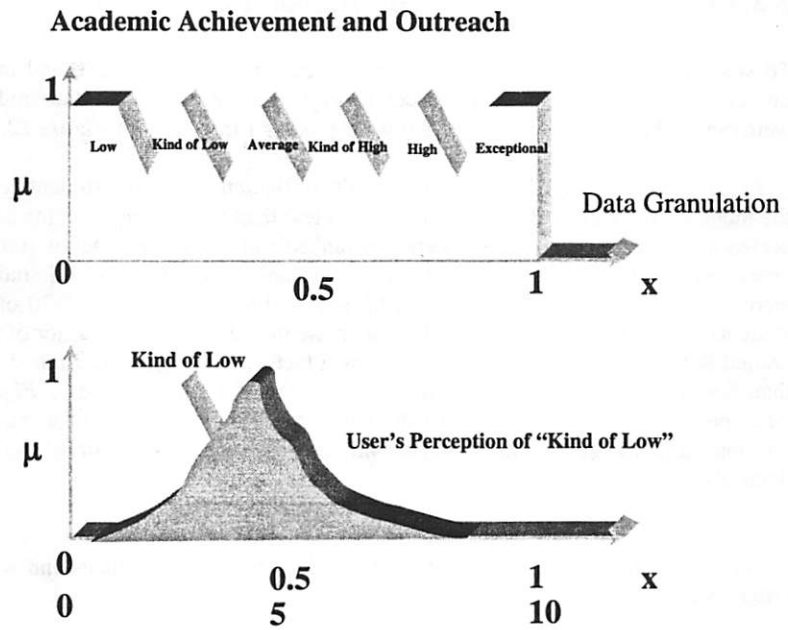


Figure 10. User's perception of Academic

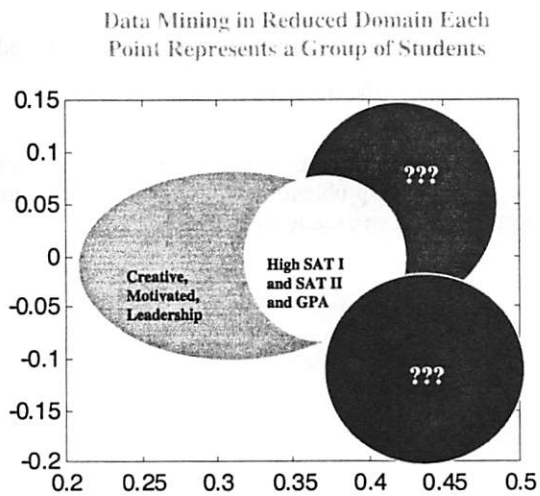


Figure 11. Typical Text and Rule Data Mining based on Techniques described in "Search Strategy and Figure 5.

3.3.1 Effect of Preferences on Ranking of Students

To study the effect of preferences in the process of student selection and in the process of the ranking, the preferences in ***Figure 8*** were changed and students were ranked based on perturbed preferences, models 1 through 5 in ***Figure 12.***

Figures 13.a through ***13.d*** show the results of the ranking of the students given the models 1 through 5. It is shown that given less than %10 changes on the actual preferences, most of the students were mis-ranked and mis-placed. Out of 100 students, less than %50 students or as an average only %41 of the actual students were selected (***Figure 13.a***). ***Figure 13.b*** shows that only less than %70 of the students will be correctly selected if we increase the admission by a factor of two, around %85 if we increase the admission by a factor of 3 (***Figure 13.c***), and less than %90 if we increase the admission by a factor of 4 (***Figure 13.d***). ***Figures 14.a*** through ***14.d*** show typical distribution of the 21 variables used for the Admission model. ***Figures 14.a*** through ***14.d*** show that the distribution of the students also drastically has been changed.

Now, the question will be "what are the values for the preferences and which criteria should be used to achieve such a goal?"

- Given a set of successful students, we would like to adjust the preferences such that the model could reflect this set of students.

- Diversity which includes gender distribution, ethnic background distribution, geophysical location distribution, etc.

To solve such subjective and multi-criteria optimization problems with a large number of constraints for complex problems such as University Admissions, the BISC Decision Support System is an excellent candidate.

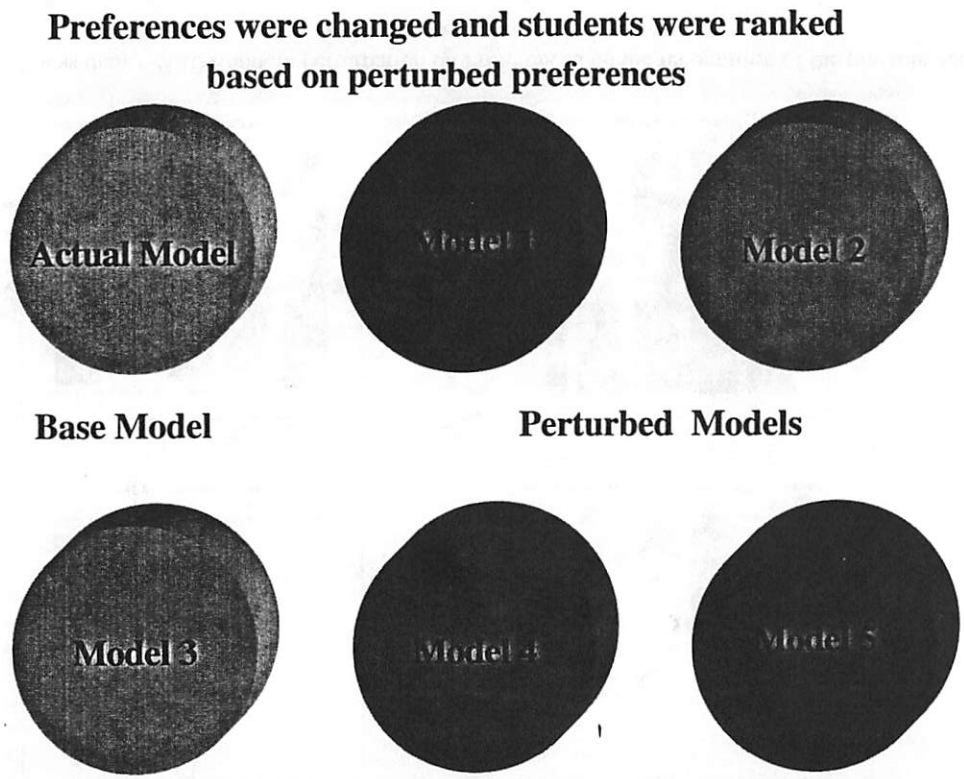


Figure 12. Models 1 through 5 are models based on preferences were perturbed around the actual value.

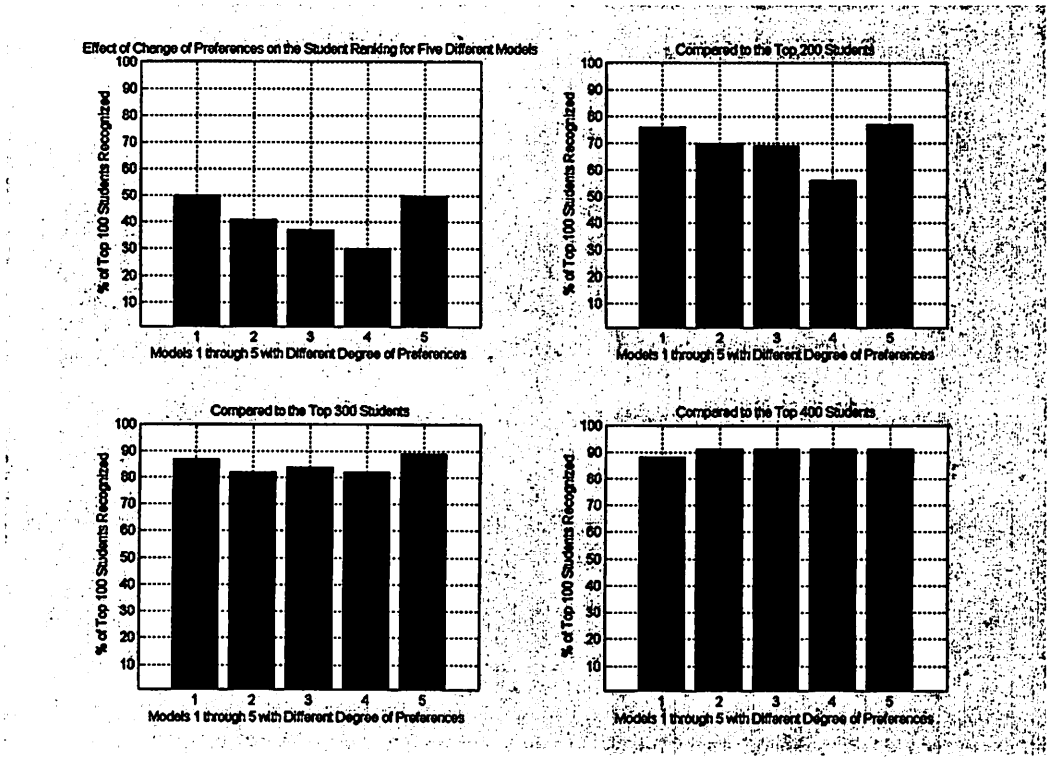


Figure 13. Effect of less than +-%10 Random perturbation on Preferences on the recognition of the pre-selected students given actual model.

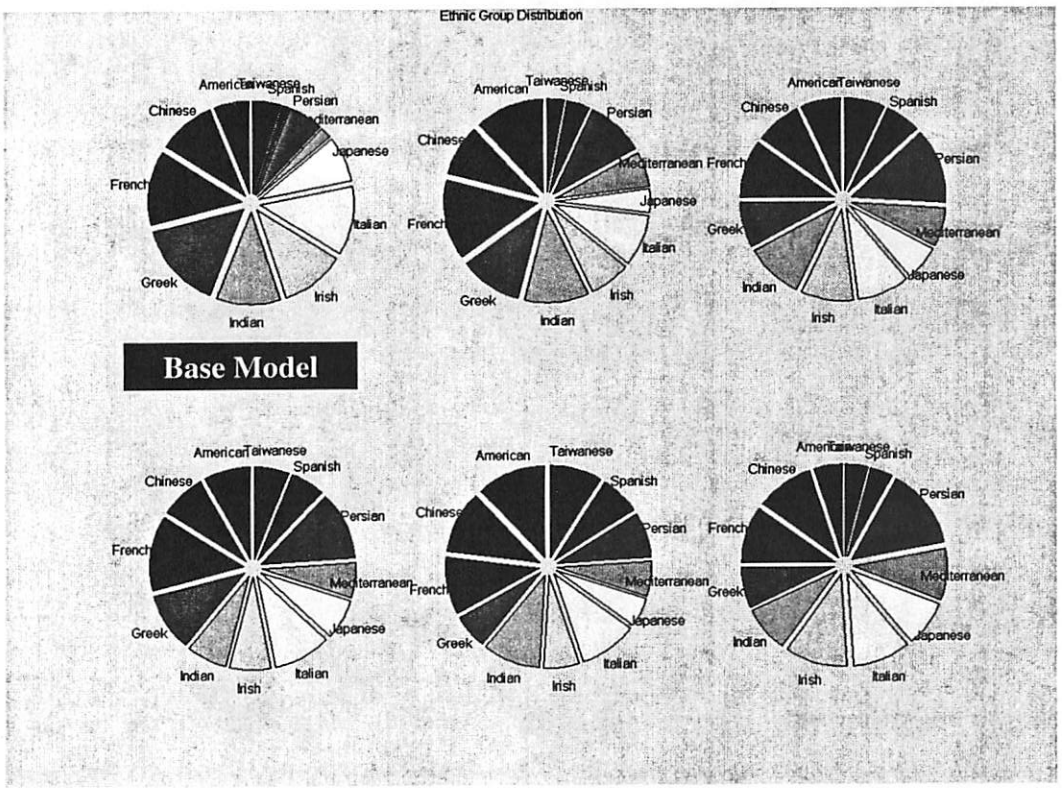


Figure 14.a. Ethnic Group Distribution

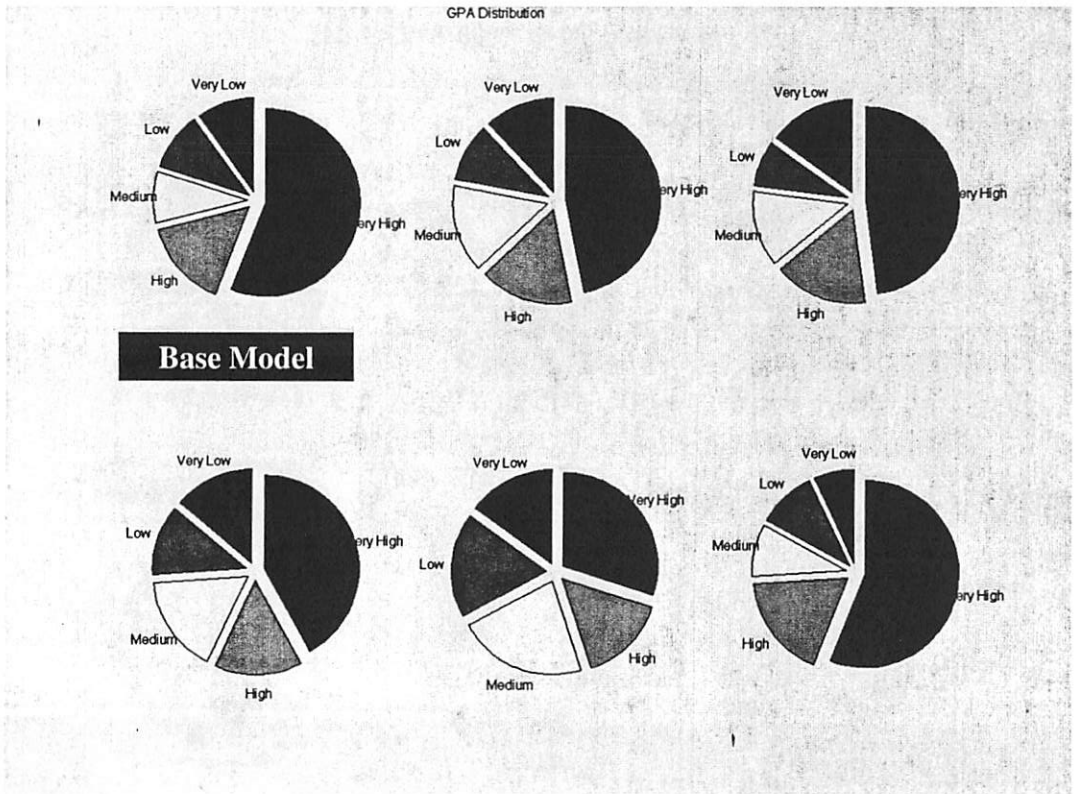


Figure 14.b. GPA Distribution

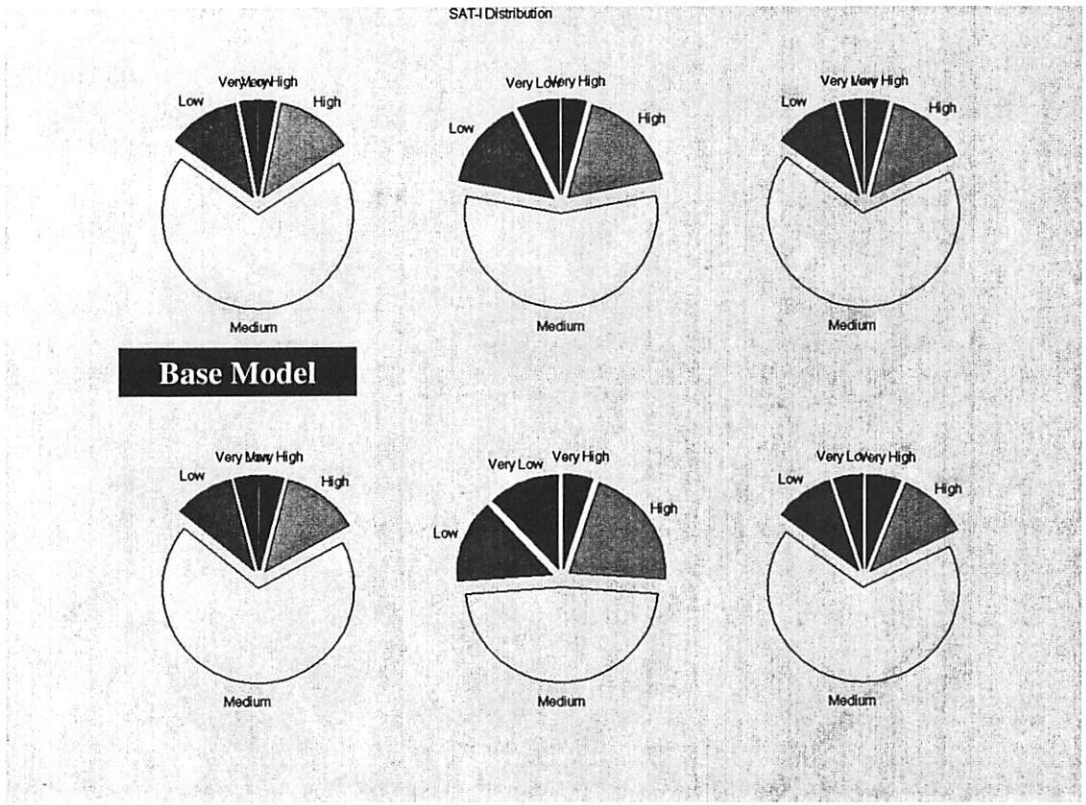


Figure 14.c. SAT-I Distribution

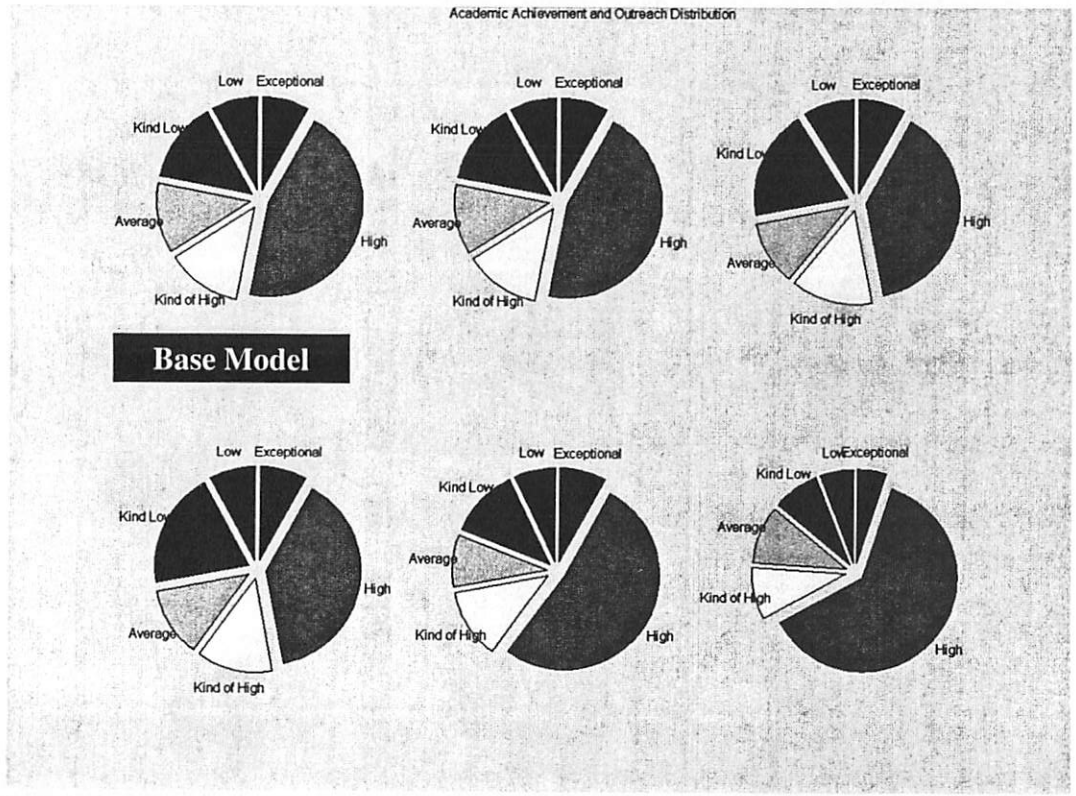


Figure 14.d. Academic Achievement Distribution

4 BISC Decision Support System

Decision Support systems may represented in either of the following forms 1) physical replica of a system, 2) analog or physical model, 3) mathematical (qualitative) model, and 4) mental models. Decision support system is an approach or a philosophy rather than a precise methodology that can be used mainly for

- strategic planning such as resource allocation
- management control such as efficient resources utilization
- operational control for efficient and effective execution of specific tasks

Decision support system is an approach or a strategy rather than a precise methodology, which can be used for 1) use intelligently the vast amounts of important data in organizations in an optimum way as a decision support system and 2) share intelligently and securely company's data internally and with business partners and customers that can be process quickly by end users and more specifically for :

- strategic planning such as resource allocation
- management control such as efficient resources utilization
- operational control for efficient and effective execution of specific tasks

The main key features of the Decision Support System for the internet applications are 1) to use intelligently the vast amounts of important data in organizations in an optimum way as a decision support system and 2) To share intelligently and securely company's data internally and with business partners and customers that can be process quickly by end users. In this section, we describe the use of the BISC Decision Support System as an intelligent real-time decision-making and management model based on two main motivations:

- In recent years, needs for more cost effective strategy and multicriteria and multiattribute optimization in an imprecise and uncertain environment have emphasized the need for risk and uncertainty management in the complex dynamic systems. There exists an ever-increasing need to improve technology that provides a global solution to modeling, understanding, analyzing and managing imprecision and risk in real-time automated decision-making for complex dynamic systems.

- As a result intelligent dynamic systems with growing complexity and technological challenges are currently being developed. This requires new technology in terms of development, engineering design and virtual simulation models. Each of these components adds to the global sum of uncertainty about risk of during decision-making process. While the technological expertise of each component becomes increasingly complex, there is a need for better integration of each component into a global model adequately capturing the uncertainty on key strategic parameters. The uncertainty quantification on such key parameters is required in any type of decision analysis.

The BISC (Berkeley Initiative in Soft Computing) Decision Support System Components include (***Figure 15***):

- Data Management: database(s) which contains relevant data for the decision process
- User Interface
    - users and decision support systems (DSS) communication
- Model Management and Data Mining
    - includes software with quantitative and fuzzy models including aggregation process, query, ranking, and fitness evaluation
- Knowledge Management and Expert System: model representation including
    - linguistic formulation,
    - functional requirements
    - constraints
    - goal and objectives
    - linguistic variables requirements

- Evolutionary Kernel and Learning Process
  - Includes software with quantitative and fuzzy models including, Fuzzy-GA, fuzzy aggregation process, ranking, and fitness evaluation
- Data Visualization: Allows end-users or decision makers can intervene in the decision-making process and see the results of the intervention

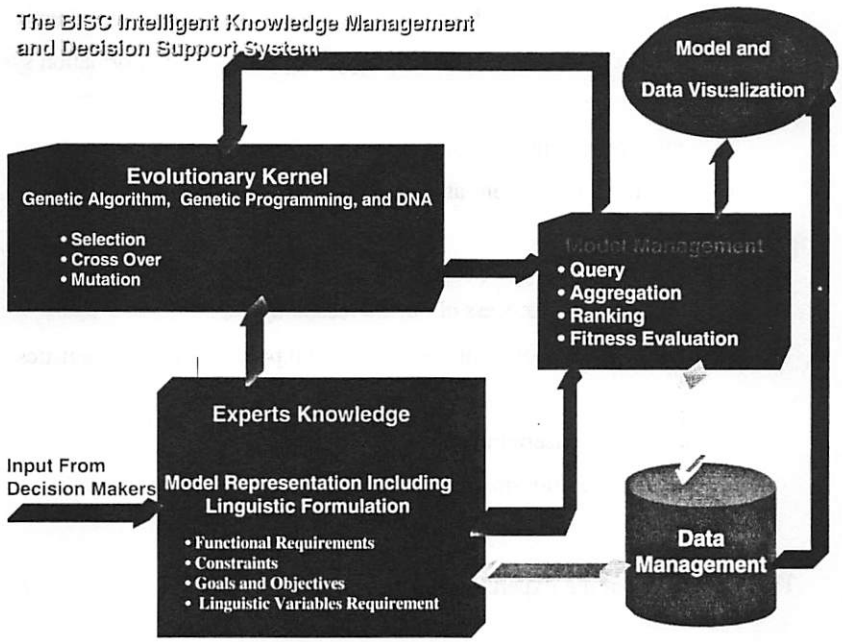


Figure 15. The BISC Decision Support System

Data Visualization and Visual Interactive Decision Making allows end-user or decision makers to recognize trends, patterns, and anomalies that can not be predicted or recognized by standard analysis methods and include the following components:

- Visual interactive modeling (VIM): user can intervene in the decision-making process and see the results of the intervention

- Visual interactive simulation (VIS): users may interact with the simulation and try different decision strategies

The Expert System uses both Fuzzy Logic and Case-Based Reasoning (CBR) for the following reasons:

- Case-Based Reasoning (CBR)
  - solve new problems based on history of given solved old problems
  - Provide a framework for knowledge acquisition and information system development
  - enhance learning capability
  - generate explanations and recommendation to users

- Fuzzy Logic
  - simulating the process of human reasoning
  - framework to computing with word and perception, and linguistics variables.
  - deals with uncertainties
  - creative decision-making process

The components of the Expert System include (***Figure 16***)

- the knowledge base contains engineering knowledge for model representation which provide problem solving environment
- the inference engine provide reasoning, conclusions, and recommendation
- the user interface and knowledge based editor provide dialog environment for questions and answers

- the advisor and translator can translate the machine inference to a human understandable advice, recommendation, and logical explanation

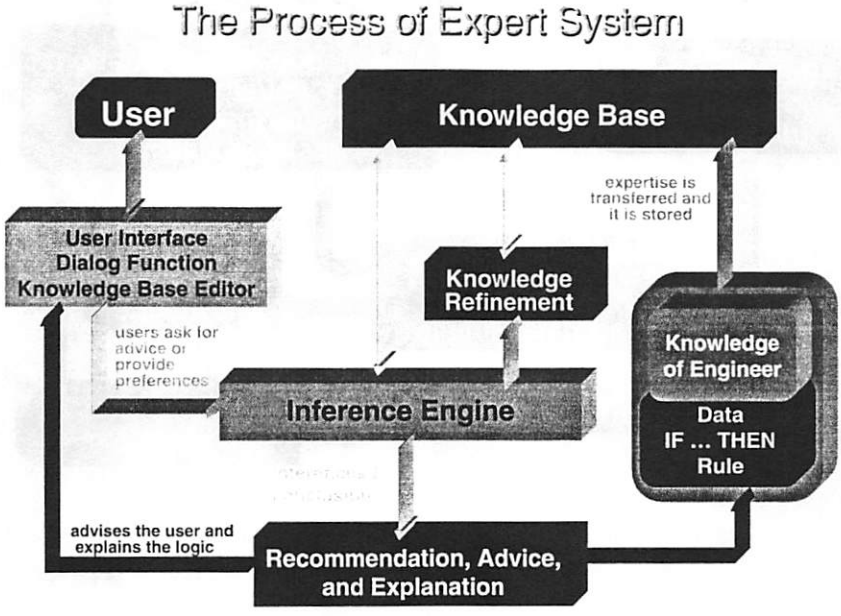


Figure 16. The components of the Expert System

The Data and Knowledge Management model include the following components (***Figure 17***)

- knowledge discovery and data mining- using search engines, databases, data mining, and online analytical processing, the proper knowledge must be found, analyzed, and put into proper context
- organize knowledge bases - it stores organizational knowledge and best practices
- knowledge acquisition - determines what knowledge (information) is critical to decision making

- knowledge representation - target audiences are defined and technologies are put into place to enable knowledge delivery when needed

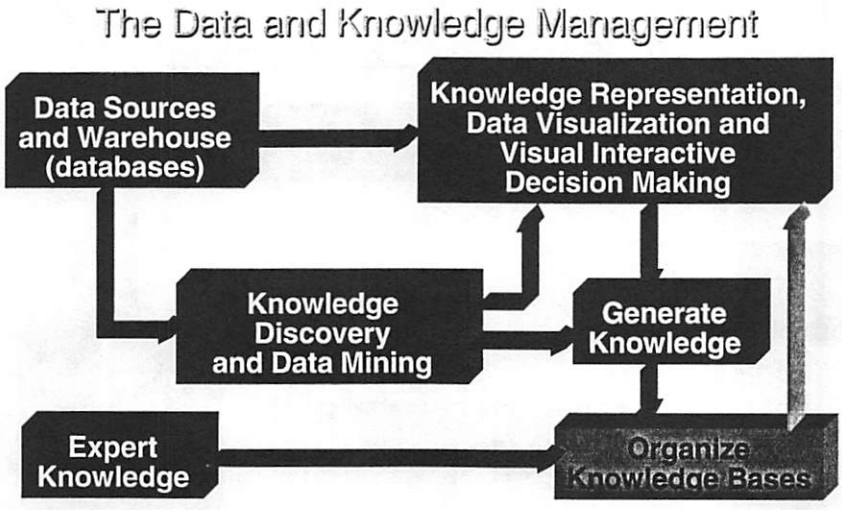


Figure 17. The Data and Knowledge Management Model

4.1 Implementation- BISC Decision Support System

In this section, we will introduce the BISC-DSS system for university admissions. In the case study, we used the GA-Fuzzy logic model for optimization purposes. The fitness function will be defined based on user constraints. For example, in the admissions problem, assume that we would like to select students not only on the basis of their achievements and criteria defined in Table 3 as a successful student, but also on the basis of diversity which includes gender distribution, ethnic background distribution, geophysical location distribution, etc. The question will be "what are the values for the preferences and which criteria should be used to achieve such a goal?" In this case, we will define the genes as the values for the preferences and the fitness function will be defined as the degree by which the distribution of each candidate in each generation match the desired distribution. Fuzzy similarity can be used to define the degree of match, which can be used for better decision analysis.

Figure 18 shows the performance of the conventional GA. The program has been run for 5000 generations and *Figure 18* shows the last 500 GA generations. As it is shown, the GA technique has been approached to a fitness of 80% and no further improvement was expected. Given what has been learned in each generation with respect to trends in the good genes, a series of genes were selected in each generation and has been used to introduce a new initial population to be used for GA. This process has been repeated until it was expected no improvement be achieved. *Figure 19* shows the performance of this interaction. The new model has reached a new fitness value, which is over 95%. *Figure 20* show the results of the ranking of the students given the actual model, predicted model (Model number 1) and models 2 through 4 which has been used to generate the initial population for training the fuzzy-GA model. It is shown that the predicted model ranked and selected most of the predefined students (*Figures 20.a-20.d*) and predefined distributions (*Figures 21.a-21.f*) and properly represented the actual model even though the initial models to generate the initial population for training were far from the actual solution (*Figures 20.a-20.d and 21.a-21.f*). Out of 100 students, more than 90% students of the actual students were selected (*Figure 20.a*). *Figure 20.b* shows that %100 of the students will be correctly selected if we increase the admission by a factor of less than two. In has been concluded for this case study that %100 of students were selected if we increase the student admission by a factor of less than 1.15. *Figures 20.a-20.d* and *21.a-21.f* show that the initial models, model 2 through 5, were far from the actual model. Out of 100 students, less than 3% of the actual students were selected (*Figure 20.a*), around 5% if we increase the admission by a factor of 2 (*Figure 20.b*), around 10% if we increase the admission by a factor of 3 (*Figure 20.c*), and less than 15% if we increase the admission by a factor of 4 (*Fiure. 20.d*). *Figures 21.a-21.f* show typical distribution of the 21 variables used for the admission model. *Figures 21.a* through *21.f* show that the distribution of the student are properly presented by the predicted model and there is an excellent match between the actual model and the predicted model, even though the distributions of the initial populations are far from the actual model.

To show if the new technique is robust, we tested the methodology with different initial populations and different constraints. In addition, we have used the methodology for different problems. It has been concluded that in all cases, we were able to design a model, which represents the actual model given that all the constraints have been defined. *Figure 22* shows the results from data mining in reduced domain using part of a selected dataset as shown on *Figure 11* as a typical representation and techniques and strategy represented in *Figure 5.*

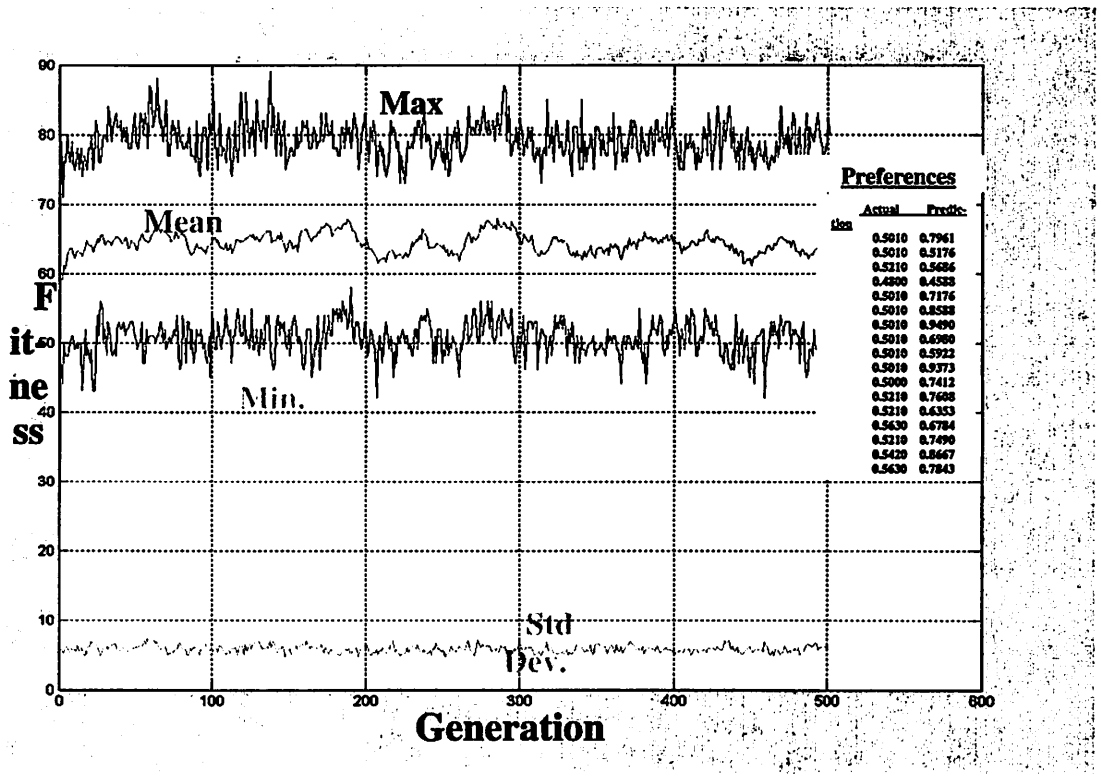


Figure 18. Conventional GA: Multi-Objective Multi-Criteria Optimization for the University Admission

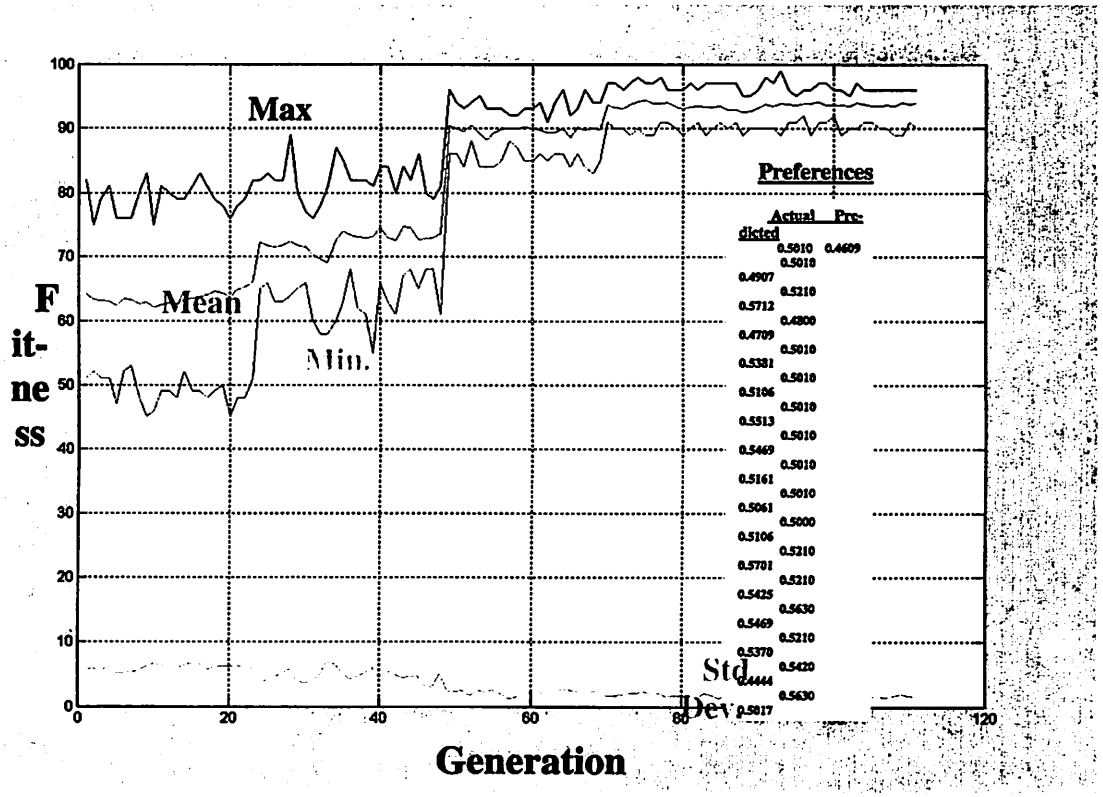


Figure 19. Interactive-GA Multi-Objective Multi-Criteria Optimization for the University Admission

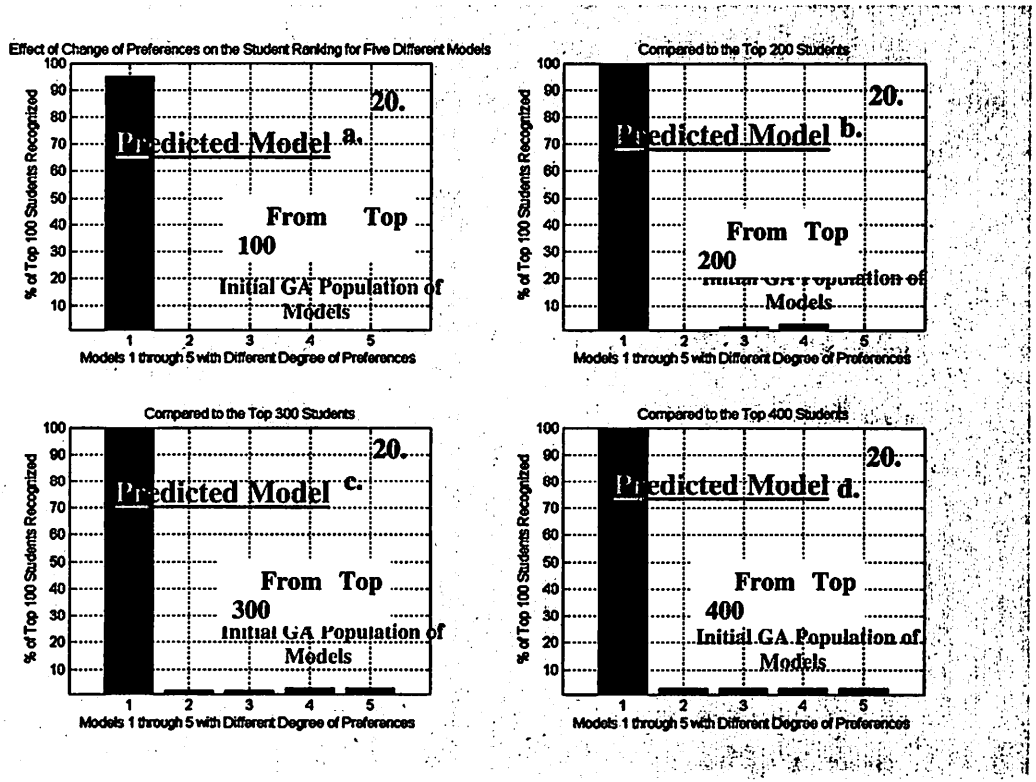


Figure 20. Results of the Ranking of the Students given Predicted Model and initial population for Fuzzy-GA Model

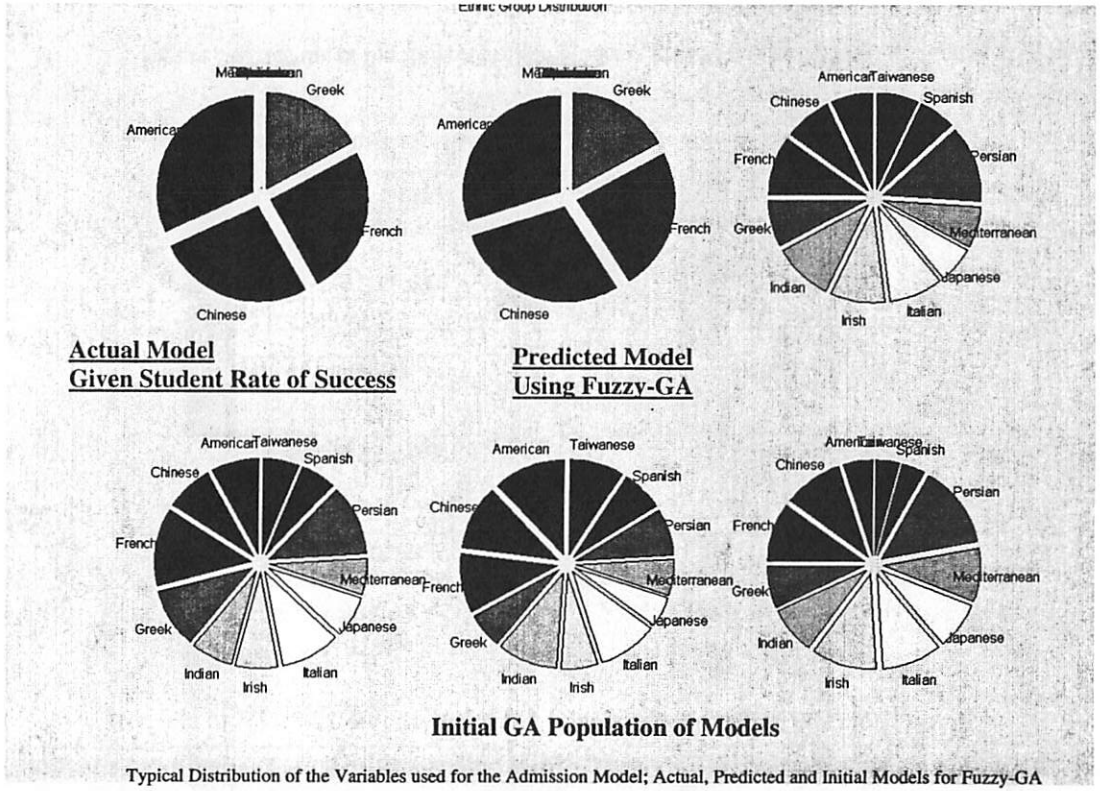


Figure 21.a. Ethnic Group Distribution

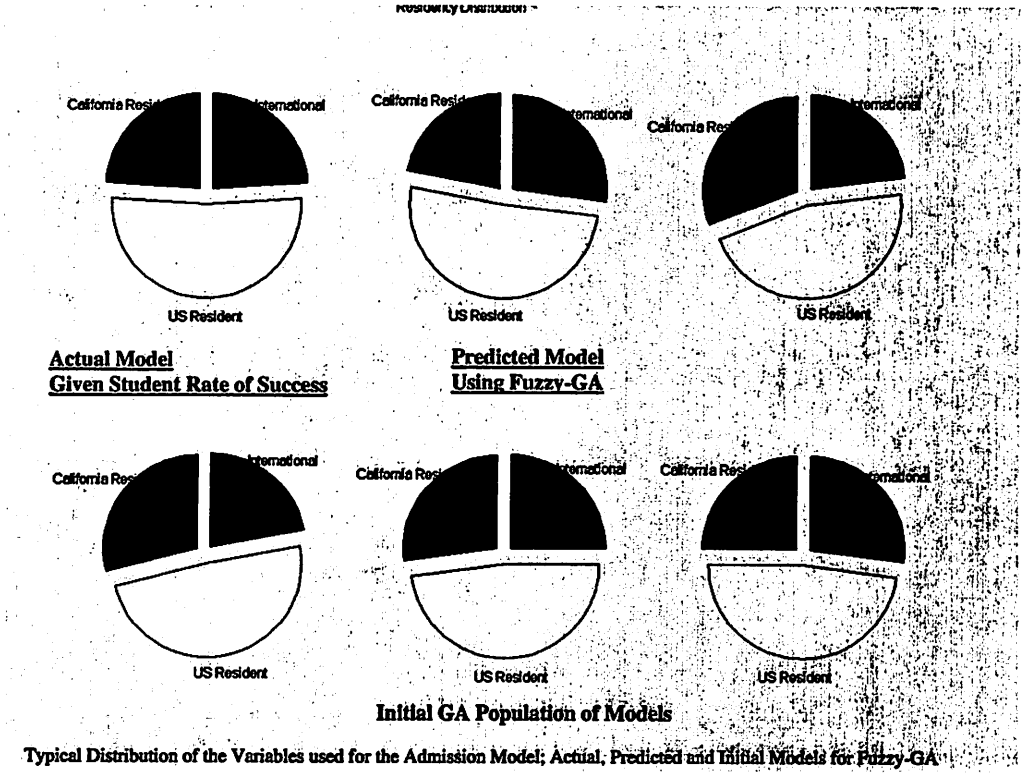


Figure 21.b. Residency Distribution

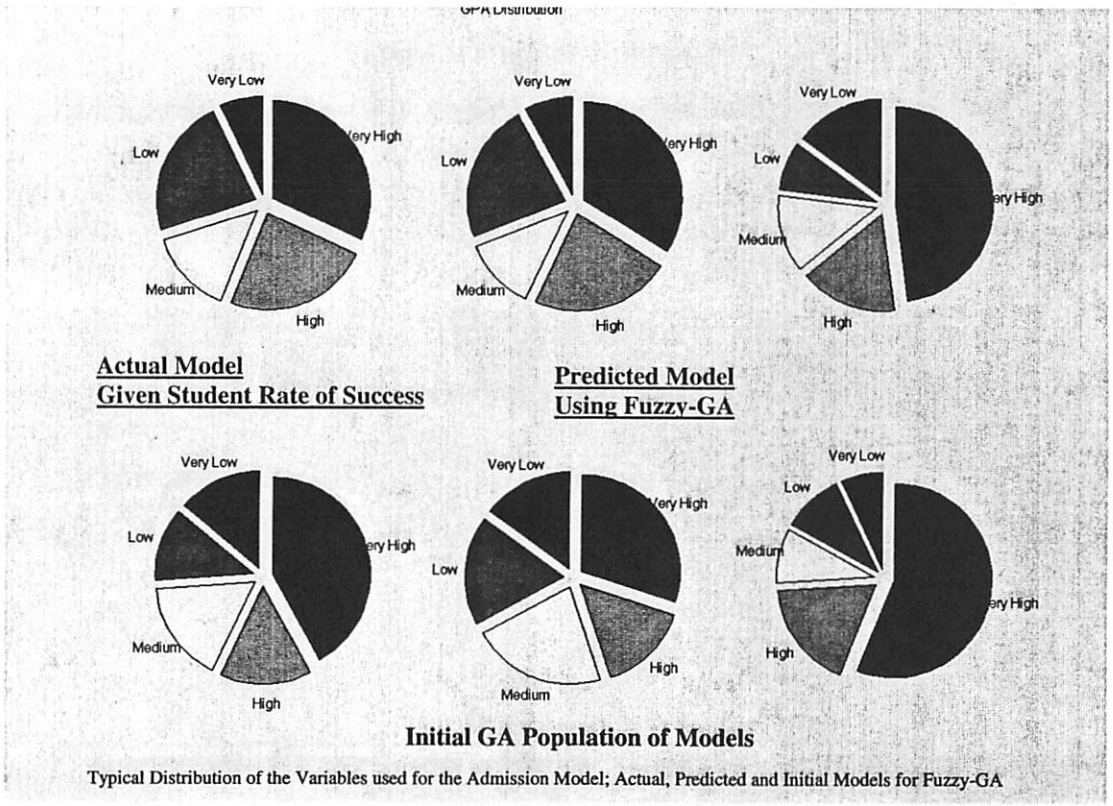


Figure 21.c. GPA Distribution

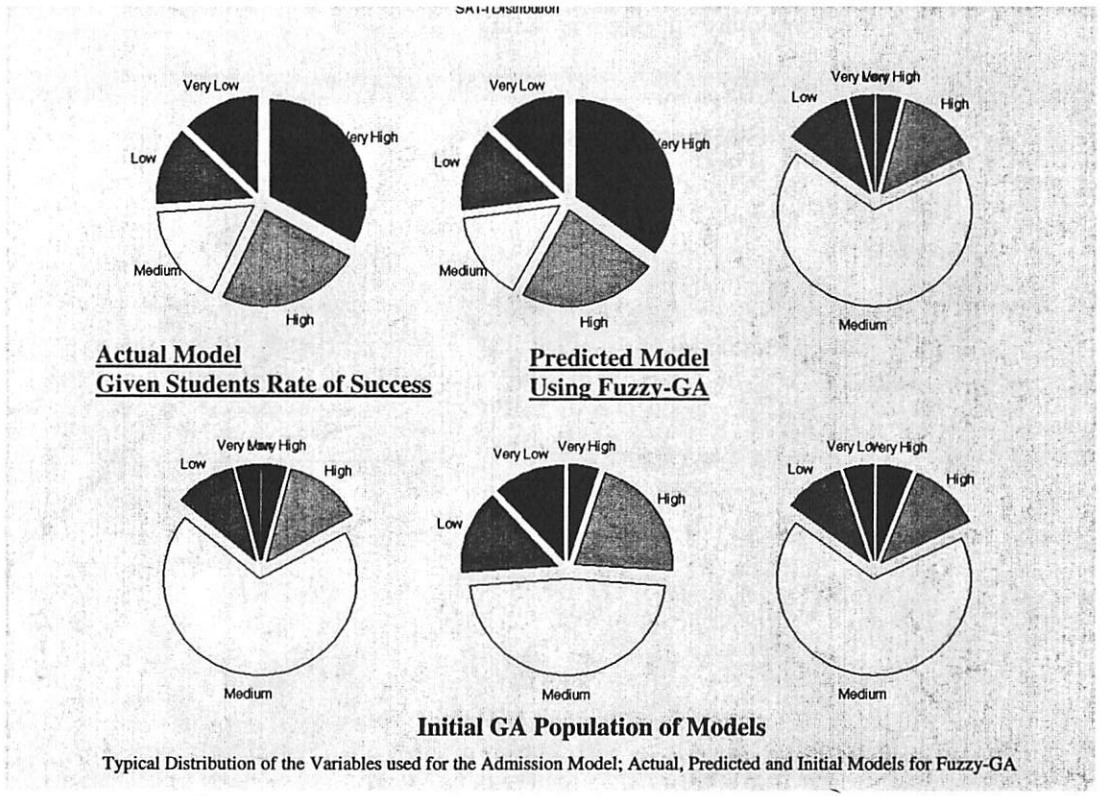
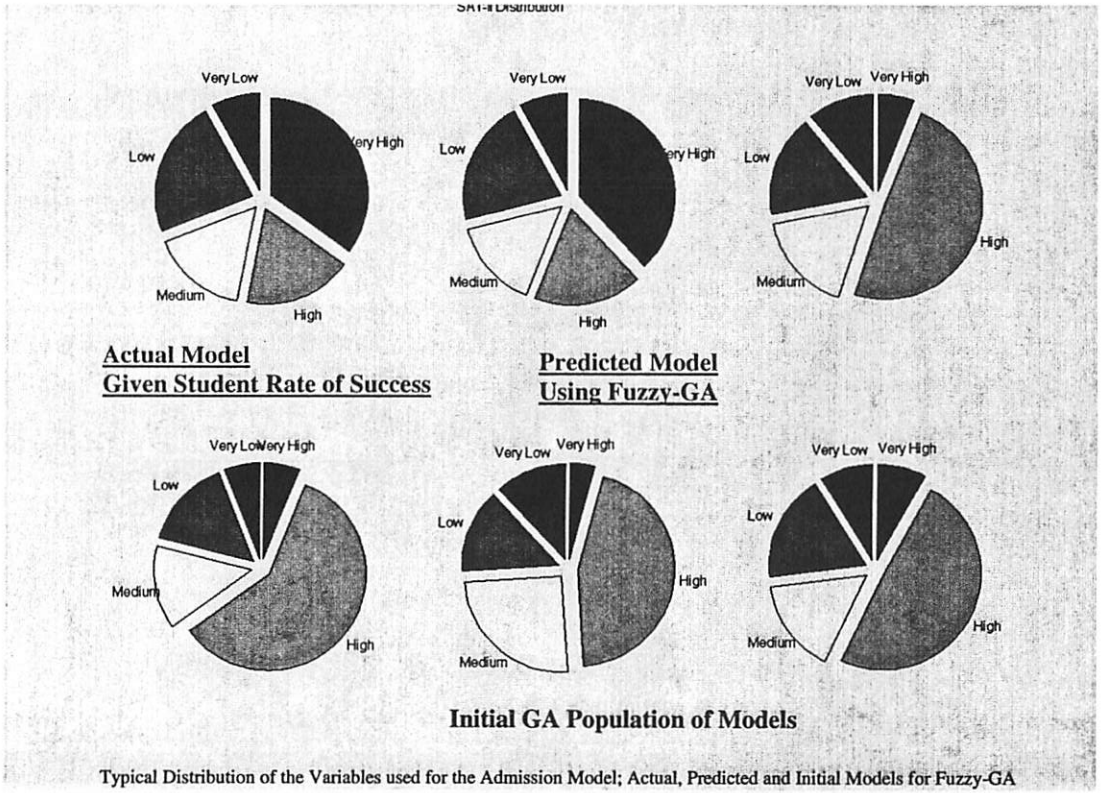


Figure 21.d. SAT-I Distribution



Typical Distribution of the Variables used for the Admission Model; Actual, Predicted and Initial Models for Fuzzy-GA

Figure 21.e. SAT-II Distribution

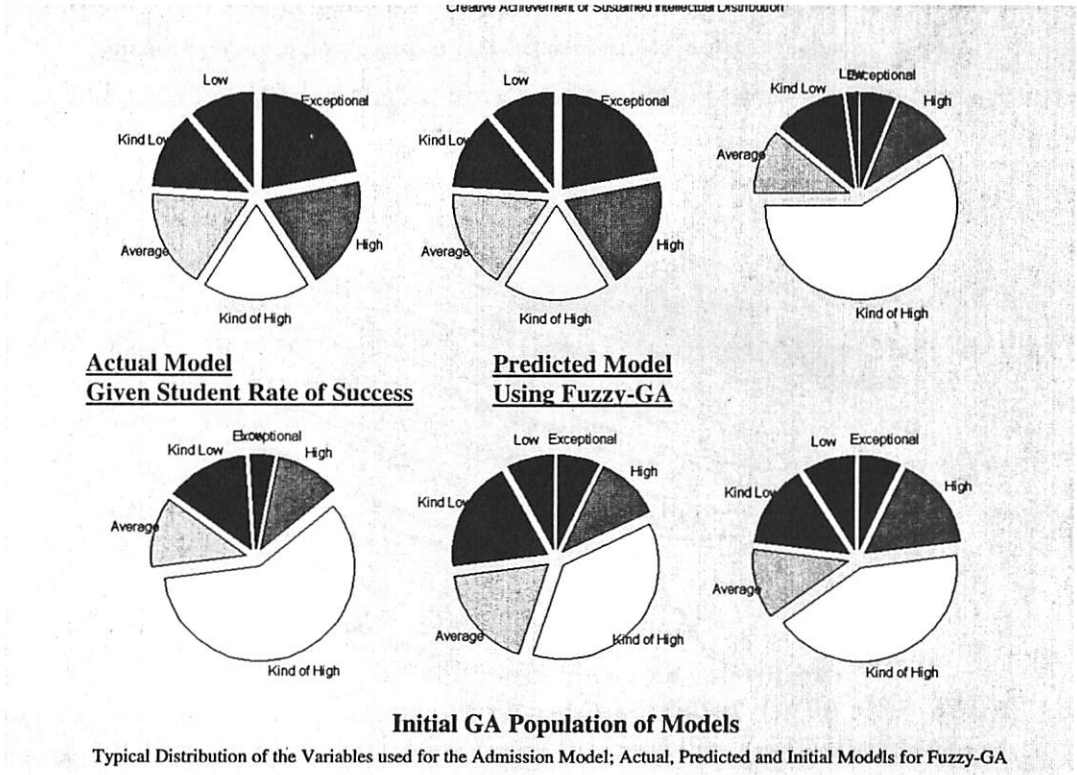


Figure 21.f. Creative Achievement or Sustained Intellectual Distribution

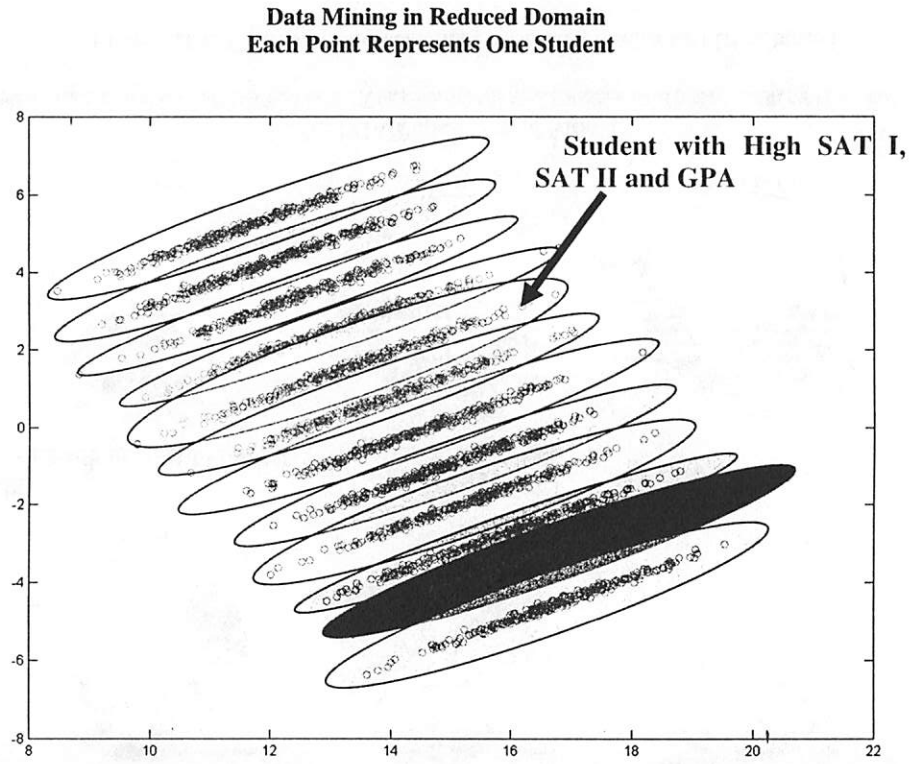


Figure 22. Data Mining based on Techniques described in "Search Strategy" and Fig. 5. on selected dataset

5.3 Date Matching

The main objective of this project was to find the best possible match in the huge space of possible outputs in the databases using the imprecise matching such as fuzzy logic concept, by storing the query attributes and continuously refining the query to update the user's preferences. We have also built a Fuzzy Query system, which is a java application that sits on top of a database.

With traditional SQL queries (relational DBMS), one can select records that match the selection criteria from a database. However, a record will not be selected if any one of the conditions fails. This makes searching for a range of potential candidates difficult. For example, if a company wants to find an employee who is proficient in skill A, B, C and D, they may not get any matching records, only because some candidates are proficient in 3 out of 4 skills and only semi-proficient in the other one. Since traditional SQL queries only perform Boolean matching, some qualities of real life, like "far" or "expensive" or "proficient", which involve matters of degree, are difficult to search for in relational databases. Unlike Boolean logic, fuzzy logic allows the degree of membership for each element to range over an interval. So in a fuzzy query, we can compute how similar a record in the database is to the desired record. This degree of similarity can be used as a ranking for each record in the database. Thus, the aim of the fuzzy query project for date matching is to add the capability of imprecise querying (retrieving similar records) to traditional DBMS. This makes some complex SQL statements unnecessary and also eliminates some repetitious SQL queries (due to empty-matching result sets).

In this program, one can basically retrieve all the records from the database, compare them with the desired record, aggregate the data, compute the ranking, and then output the records in the order of their rankings. Retrieving all the records from the database is a naïve approach because with some preprocessing, some very different records are not needed from the database. However, the main task is to compute the fuzzy rankings of the records so efficiency is not the main concern here.

The major difference between this application and other date matching system is that a user can input his hobbies in a fuzzy sense using a slider instead of choosing crisp terms like "Kind of" or "Love it". These values are stored in the database according to the slider value **(Figures 23 and 24)**.

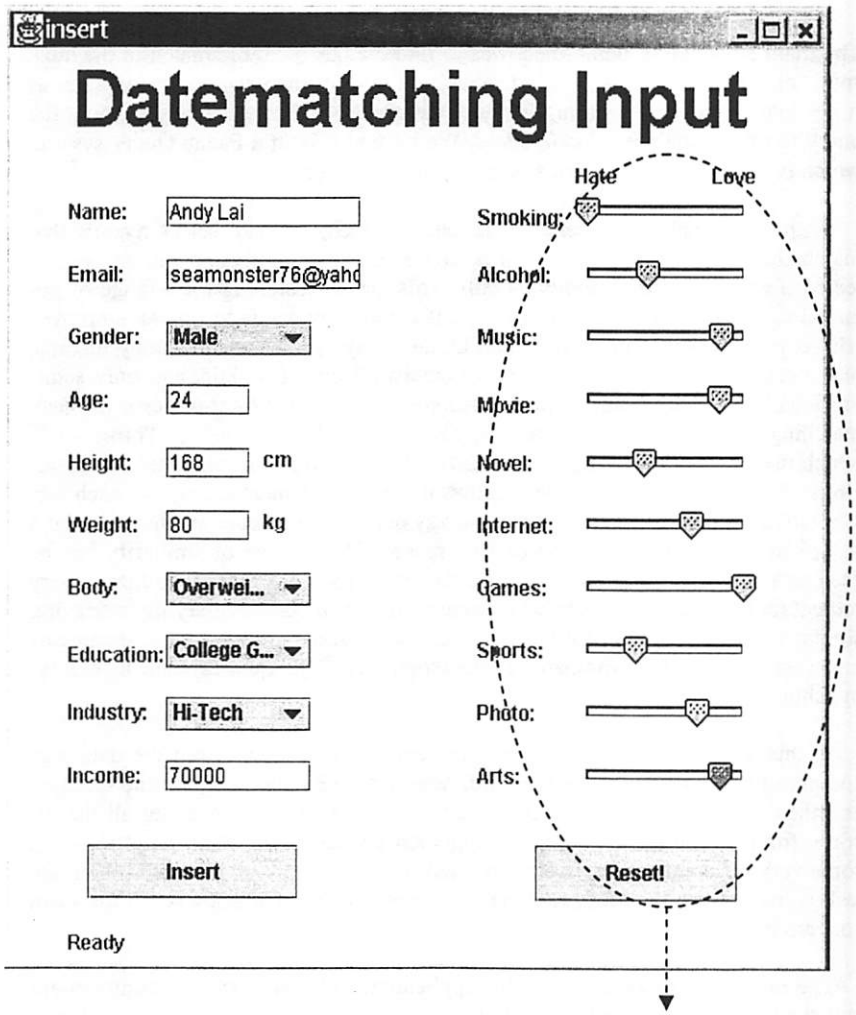


Figure 23. Date matching input form

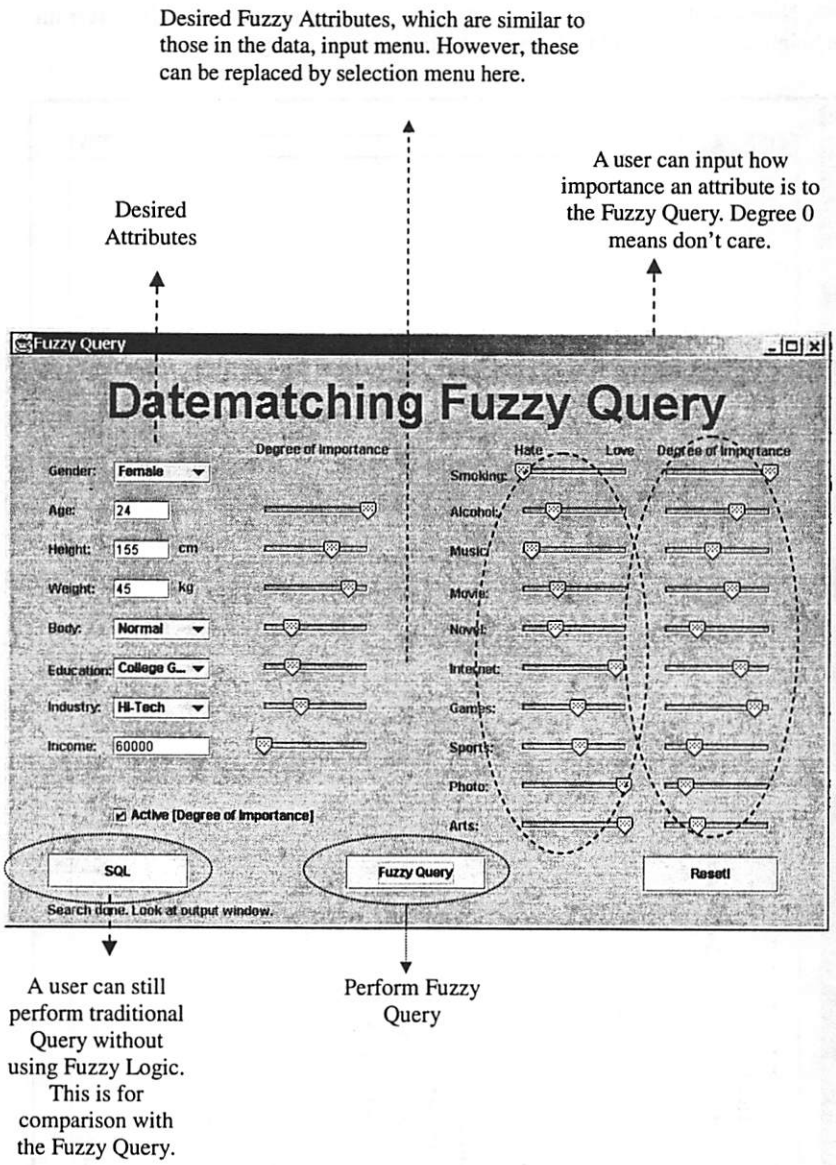


Figure 24. Snapshot of the Date Matching Software

Figure 25 shows the results are obtained from fuzzy query using the search criteria in the previous page. The first record is the one with the highest ranking – 80%. Note that it matches the age field of the search criteria but it's off a bit from the height and weight fields. So one can do imprecise querying.

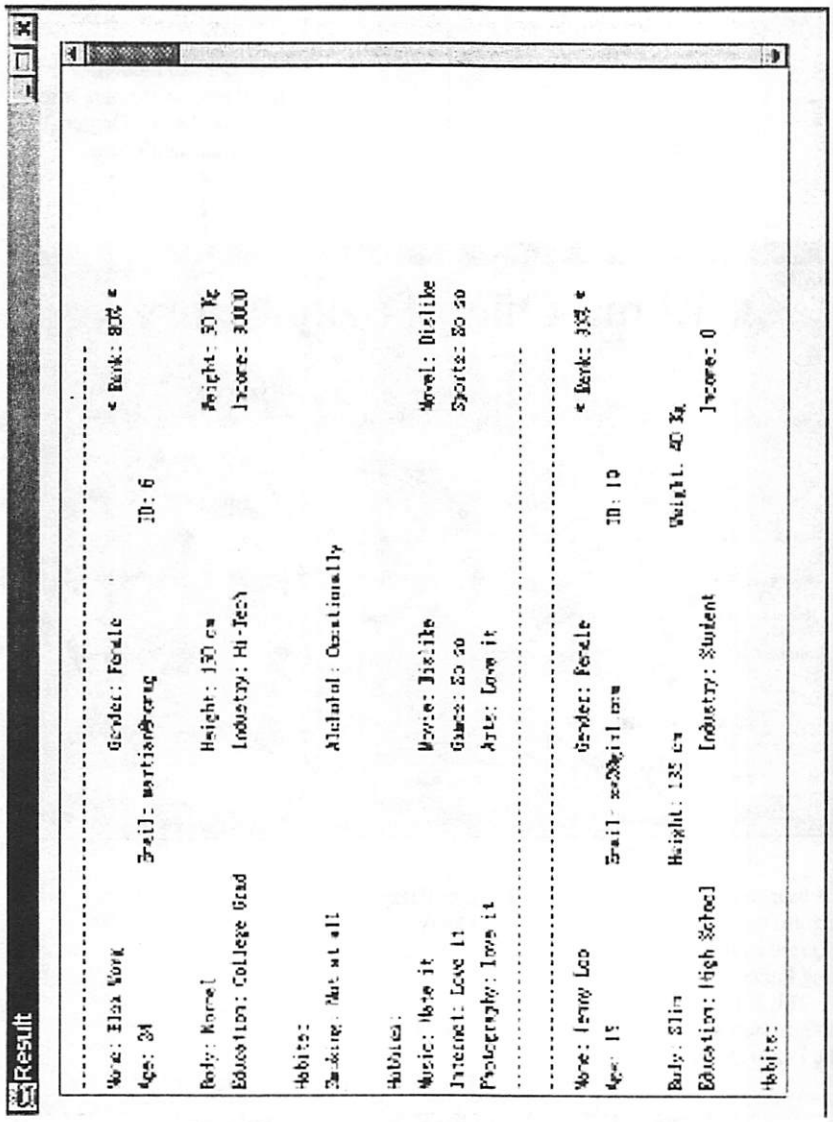


Figure 25. Sample of the output from Date Matching software

The system is modulated into three main modules (**Figure 26**). The core module is the fuzzy engine which accepts input from a GUI module and outputs result to another GUI module. The GUIs can be replaced by other processing modules such that the input can be obtained from other system and the result can be used for further analysis.

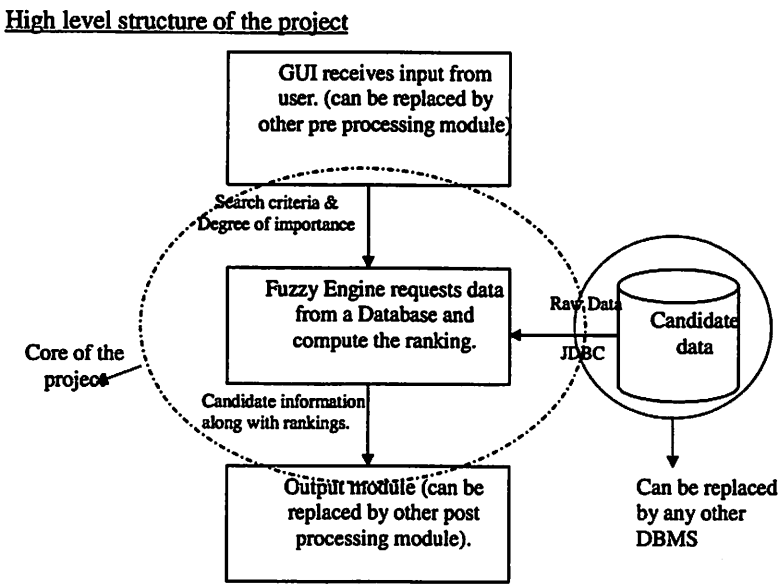


Figure 26. System Structure

The current date matching software can be modified or expanded in several ways:

1. One can build a server/client version of date-matching engine so that we can use a centralized database and all users around the world can do the matching through the web. The ranking part (computation) can still be done on local machine since every search is different. This can also help reduce the server load.
2. The attributes, granulation models and the "meaning" of the data can be tunable so that the system is more configurable and adaptive to changes.

3. User preference capability can be added to the system. (The notion of "overweight" and "tall" can be different to different people.)
4. The GUI needs to be changed to meet real user needs.
5. One can build a library of fuzzy operators and aggregation functions such that one can choose the operator and function that matches the application.
6. One can instead build a generic fuzzy engine framework which is tunable in every way to match clients' needs.
7. The attributes used in the system are not very complete compared to other data matching systems online. However, the attributes can be added or modified with some modification to the program without too much trouble.

Recently, we have added a web interface to the existing software and built the database framework for further analysis in user profiling so that users could find the best match in the huge space of possible outputs. We saved user profiles and used them as basic queries for that particular user. Then, we stored the queries of each user in order to "learn" about this user's preference. In addition, we rewrote the fuzzy search engine to be more generic so that it would fit any system with minimal changes. Administrator can also change the membership function to be used to do searches. Currently, we are working on a new generic software to be developed for a much more diverse applications and to be delivered as stand alone software to both academia and businesses.

5.3 BISC-DSS Potentials

The followings are the potential applications of the BISC Decision Support System:

1. *Physical Stores or E-Store:* A computer system that could instantly track sales and inventory at all of its stores and recognize the customer buying trends and provide suggestion regarding any item that may interest the customer

    - to arrange the products
    - on pricing, promotions, coupons, etc

- for advertising strategy

2. *Profitable Customers:* A computer system that uses customer data that allows the company to recognize good and bad customer by the cost of doing business with them and the profits they return

   - keep the good customers
   - improve the bad customers or decide to drop them
   - identify customers who spend money
   - identify customers who are profitable
   - compare the complex mix of marketing and servicing costs to access to new customers

3. *Internet-Based Advising: :* A computer system that uses the expert knowledge and the customer data (Internet brokers and full-service investment firms) to recognize the good and bad traders and provide intelligent recommendation to which stocks buy or sell

   - reduce the expert needs at service centers
   - increase customer confidence
   - ease-of-use
   - Intelligent coaching on investing through the Internet
   - allow customers access to information more intelligently

4. *Managing Global Business:* A computer system responding to new customers and markets through integrated decision support activities globally using global enterprise data warehouse
   - information delivery in minutes
   - lower inventories
   - intelligent and faster inventory decisions in remote locations

5. *Resource Allocator:* A computer system that intelligently allocate resources given the degree of match between objectives and resources available

   - resource allocation in factories floor
   - for human resource management
   - find resumes of applicants posted on the Web and sort them to match needed skill and can facilitate training and to manage fringe benefits programs
   - evaluate candidates predict employee performance

6. *Intelligent Systems to Support Sales:* A computer system that matching products and services to customers needs and interest based on case-based reasoning and decision support system to improve

   - sale
   - advertising

7. *Enterprise Decision Support:* An interactive computer-based system that facilitates the solution of complex problems by a group of decision makers either by speeding up the process of the decision-making process and improving the quality of the resulting decisions through expert and user (company-customer) collaboration and sharing the information, goals, and objectives.

8. *Fraud Detection:* An Intelligent Computer that can learn the user's behavior through in mining customer databases and predicting customer behaviours (normal and irregularities) to be used to uncover, reduce or prevent fraud.

   - in credit cards
   - stocks

- financial markets
- telecommunication
- insurance

9. *Supply-Chain Management (SCM):* Global optimization of design, manufacturing, supplier , distribution, planning decisions in a distributed environment

10. *BISC-DSS and Autonomous Multi-Agent System:* A key component of any autonomous multi-agent system –especially in an adversarial setting -- is decision module, which should be capable of functioning in an environment of imprecision, uncertainty and imperfect reliability. BISC-DSS will be focused on the development of such system and can be used as a decision-support system for ranking of decision alternatives. BISC-DSS can be used :

    - As global optimizer for planning decisions in a distributed environment
    - To facilitates the solution of complex problems by a group of autonomous agents by speeding up the process of decision-making, collaboration and sharing the information, goals, and objectives
    - To intelligently allocate resources given the degree of match between objectives and resources available
    - Assisting autonomous multi-agent system in assessing the consequences of decision made in an environment of imprecision, uncertainty, and partial truth and providing a systematic risk analysis
    - Assisting multi-agent system answer "What if Questions", examine numerous alternatives very quickly, ranking of decision alternatives, and find the value of the inputs to achieve a desired level of output

11. *BISC-DSS can be integrated into TraS toolbox to develop:* Intelligent Tracking System (ITraS): Given the information about suspicious activities such as phone calls, emails, meetings, credit card information, hotel and airline reservations that are stored in a database containing the originator, recipient, locations, times, etc. we can use BISC-DSS and visual data mining to find suspicious pattern in data using geographical maps. The technology developed can detect unusual patterns, raise alarms based on classification of activities and offer explanations based on automatic learning techniques for why a certain activity is placed in a particular class such as "Safe", "Suspicious", "Dangerous" etc. The underlying techniques can combine expert knowledge and data driven rules to continually improve its classification and adapt to dynamic changes in data and expert knowledge.

12. *BISC-DSS can be integrated into fuzzy conceptual set toolbox to develop TIKManD:* A new Tool for Intelligent Knowledge Management and Discovery (TIKManD). The model can be used to recognize terrorism activities through data fusion & mining and pattern recognition technology given online textual information through Email or homepages and voice information given the wire tapping and/or chat lines or huge number of "tips" received immediately after the attack.

The followings are the potential applications areas of the BISC Decision Support System:

- *Finance:* stock prices and characteristics, credit scoring, credit card ranking
- *Military:* battlefield simulation and decision making
- *Medicine:* diagnosis
- *Marketing:* store and product display and electronic shopping

- *Internet:* provide knowledge and advice to large numbers of user
- *Education:* university admission

5 Web Intelligence: Web-Based BISC Decision Support system

Most of the existing search systems 'software' are modeled using crisp logic and queries. In this chapter we introduce fuzzy querying and ranking as a flexible tool allowing approximation where the selected objects do not need to match exactly the decision criteria resembling natural human behavior. The model consists of five major modules: the Fuzzy Search Engine, the Application Templates, the User Interface, the Database and the Evolutionary Computing. The system is designed in a generic form to accommodate more diverse applications and to be delivered as stand-alone software to academia and businesses.

5.1 Web Intelligence: Introduction

Searching database records and ranking the results based on multi-criteria queries is central for many database applications used within organizations in finance, business, industrial and other fields. Most of the available systems 'software' are modeled using crisp logic and queries, which results in rigid systems with imprecise and subjective process and results. In this chapter we introduce fuzzy querying and ranking as a flexible tool allowing approximation where the selected objects do not need to match exactly the decision criteria resembling natural human behavior.

The model consists of five major modules: the Fuzzy Search Engine (FSE), the Application Templates (AT), the User Interface (UI), the Database (DB) and the Evolutionary Computing (EC). We developed the software with many essential features. It is built as a web-based software system that users can access and use over the Internet. The system is designed to be generic so that it can run different application domains. To this end, the Application Template module provides information of a specific application as attributes and properties, and serves as a guideline structure for building a new application.

The Fuzzy Search Engine (FSE) is the core module of the system. It has been developed to be generic so that it would fit any application. The main FSE component is the query structure, which utilizes membership functions, similarity functions and aggregators.

Through the user interface a user can enter and save his profile, input criteria for a new query, run different queries and display results. The user can manually eliminate the results he disapproves or change the ranking according to his preferences.

The Evolutionary Computing (EC) module monitors ranking preferences of the users' queries. It learns to adjust to the intended meaning of the users' preferences.

5.2 Model framework

The DSS system starts by loading the application template, which consists of various configuration files for a specific application (see section 5.4) and initializing the database for the application (see section 5.6), before handling user's requests, see **Figure 27.**

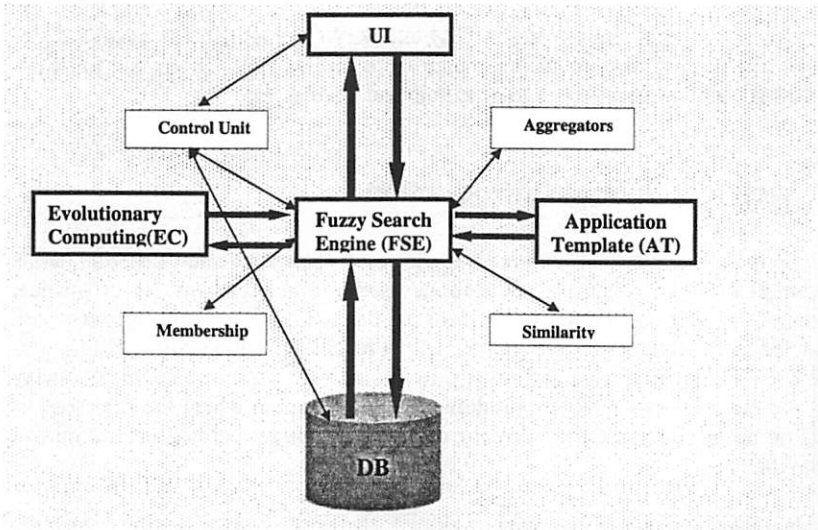


Figure 27. The BISC-DSS general framework

Once the DSS system is initialized, users can enter in the user interface their own profiles or make a search with their preferences. These requests are handled by the control unit of the system. The control unit converts user input into data objects that are recognized by the DSS system then, based on the request types, it forwards them to the appropriate modules.

If the user wants to create a profile, the control unit will send the profile data directly to the database module, which stores the data in the database for the application. If the user wants to query the system, the control unit will direct the user's preferences to the Fuzzy Search Engine, which queries the database (see section 5.3). The query results will be sent back to the control unit and displayed to the users.

5.3 Fuzzy Engine

5.3.1 Fuzzy Query, search and Ranking

To support generic queries, the fuzzy engine has been designed to have a tree structure. There are two types of nodes in the tree, category nodes and attribute nodes, as depicted in **Figure 28.** While multiple category levels are not necessary, they are allowed to allow various refinements of the query through the type of aggregation of the children. The categories can only act to aggregate the lower levels. The attribute nodes contain all the important information about query. They contain the membership functions for the fuzzy comparison as well as use the various aggregation methods to compare two values.

The flow of control in the program when a query is executed is as follows. The root node receives a query formatted as a fuzzy data object and is asked to compare the query fuzzy data to a record from the database also formatted as a fuzzy data object. At each category node, the compare method is called for each child and then aggregated using an aggregator object.

The attribute nodes handle the compare method slightly different than the category nodes. There are two different ways attributes may be compared. The attribute nodes contain a list of membership functions comprising the fuzzy set. The degrees of membership for this set are passed to the similarity comparator object, which currently has a variety of different methods to calculate the similarity between the two membership vectors. In the other method, the membership vector created by having full membership to a single membership function specified in the fuzzy data object, but no membership value for the other functions.

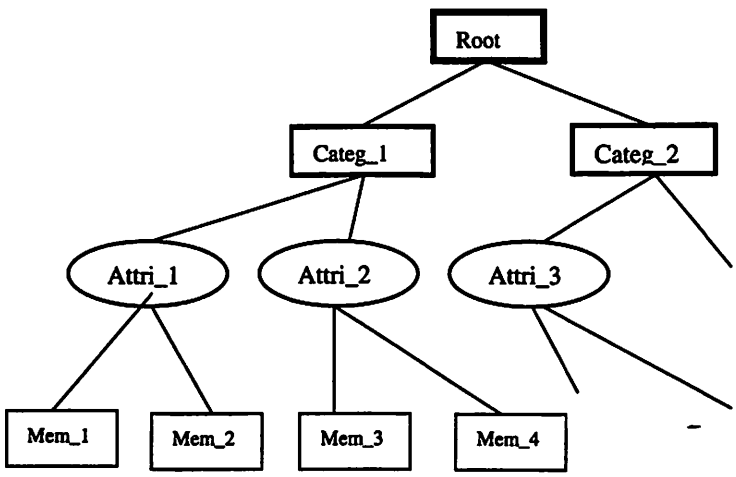


Figure 28. The Fuzzy search engine tree structure.

The resulting comparison value returned from the root node is assigned to the record. The search request is then added to a sorted list ordered by this ranking in descending value. Each of the records from the database is compared to the query and the results are returned. For certain search criteria, it may be desirable to have exact values in the query. For such criteria, the database is used to filter the records for comparison.

5.3.2 Membership function

Currently there are three membership functions implemented for the Fuzzy Engine. A generic interface has been created to allow several different types of membership functions to be added to the system. The three types of membership functions in the system are: Gaussian, Triangular and Trapezoidal. These functions have three main points, for the lower bound, upper bound and the point of maximum membership. For other functions, optional extra points may be used to define the shape (an extra point is required for the trapezoidal form).

5.4 Application Template

The DSS system is designed to work with different application domains. The application template is a format for any new application we build, it contains data of different categories, attributes and membership functions of that application. The

```
#########################################################################
#This is a properties file for membership definition. We should specify
#the following properties for an attribute:
# - A unique identifier for each defined membership function.
# - A type from the following: {Gaussian, Triangle, Trapezoid}
# - Three points: Lowerbound, Upperbound, Maximum
# - Optional point: Auxillary Maximum
# Format:
# <MF_Name>.membershipFunctionName = <MF_Name>
# <MF_Name>.membershipFunctionType = {Gaussian/Triangle/Trapezoid}
# <MF_Name>.lowerBound          = lowerBoundValue
# <MF_Name>.upperBound          = upperBoundValue
# <MF_Name>.maxValue            = maxValue
# <MF_Name>.optionPoint         = pt1, pt2, pt3 ...
#
#########################################################################
```

```
#########################################################################
#
# Gender Membership Functions
#
male.membershipFunctionName = male
male.membershipFunctionType = Triangle
male.lowerbound         = 1
male.upperbound         = 1
male.maxValue           = 1

female.membershipFunctionName = female
female.membershipFunctionType = Triangle
female.lowerbound       = 0
female.upperbound       = 0
female.maxValue         = 0
#
# Age Membership Functions
#
young.membershipFunctionName = young
young.membershipFunctionType = Triangle
young.lowerbound        = 0
young.upperbound        = 35
young.maxValue          = 20

middle.membershipFunctionName = middle
middle.membershipFunctionType = Triangle
middle.lowerbound       = 20
middle.upperbound       = 50
middle.maxValue         = 35

old.membershipFunctionName = old
old.membershipFunctionType = Triangle
old.lowerbound          = 35
old.upperbound          = 100
old.maxValue            = 50
```

Figure 29. Template of the date matching application

application template module consists of two parts the application template data file, and the application template logic. The application template data file specifies

all the membership functions, attributes and categories of an application. We can consider it as a configuration data file for an application. It contains the definition of membership functions, attributes and the relationship between them.

The application template logic parses and caches data from the data file so that other modules in the system can have faster access to definitions of membership functions, attributes and categories. It also creates a tree data structure for the fuzzy search engine to transverse. **Figure 29** shows part of the sample configuration file from the Date Matching application.

5.5 User interface

It is difficult to design a generic user interface that suits different kind of applications for all the fields. For example, we may want to have different layouts for user interfaces for different applications. To make the DSS system generic while preserving the user friendliness of the interfaces for different applications, we developed the user interfaces into two parts.

First, we designed a specific HTML interface for each application we developed. Users can input their own profiles, make queries by specifying preferences for different attributes. Details for the DSS system are encapsulated from the HTML interface so that the HTML interface design would not be constrained by the DSS system.

The second part of our user interface module is a mapping between the parameters in the HTML files and the attributes in the application template module for the application. The input mapping specifies the attribute names each parameter in the HTML interface corresponds to. With this input mapping, user interface designer can use any input method and parameter names freely (**Figure 30**).

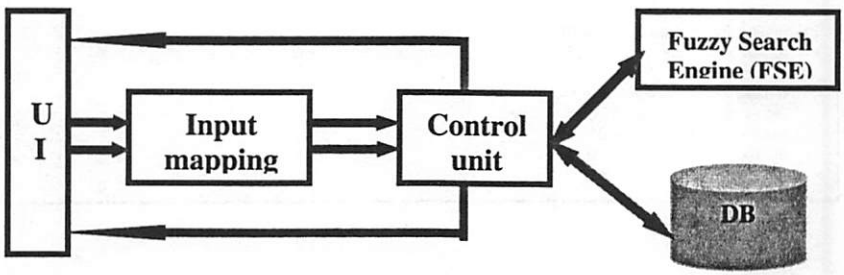


Figure 30. User interface data flow

5.6 Database (DB)

The database module is responsible for all the transactions between the DSS system and the database. This module handles all queries or user profile creations from the Fuzzy Engine and the Control Unit respectively. For queries from the Fuzzy Search Engine, it retrieves data from the database and returns it in a data object form. Usually queries are sets of attribute values and their associated weights. The database module (**Figure 31**) returns the matching records in a format that can be manipulated by the user, as eliminating one or more record or changing their order. For creating user profile, on the other hand, it takes data objects from the Control Unit and stores it in the database. There are three components in the DB module: the DB Manager (DBMgr), the DB Accessor (DBA) and DB Accessor Factory (DBA Factory).

5.6.1 DB Manager

The DB Manager is accountable for two things: setting up database connections and allocating database connections to DB Accessor objects when needed. During the initialization of the DSS system, DB Manager loads the right driver, which is used for the communications between the database and the system. It also supplies information to the database for authentication purposes (e.g. username, password, path to the database etc).

5.6.2 DB Accessor Factory

The DB Accessor Factory creates DB Accessor objects for a specific application. For example, if the system is running the date matching application, DB Accessor Factory will create DB Accessor objects for the date matching application. The existence of this class serves the purpose of using a generic Fuzzy Search Engine.

5.6.3 DB Accessor

DB Accessor is responsible for storing and getting user profiles to and from the database. It also saves queries from users to the database so that other modules in the system can analyze user's preferences. It is the component that queries the database and wrap result from the database into data objects that are recognized by our application framework.

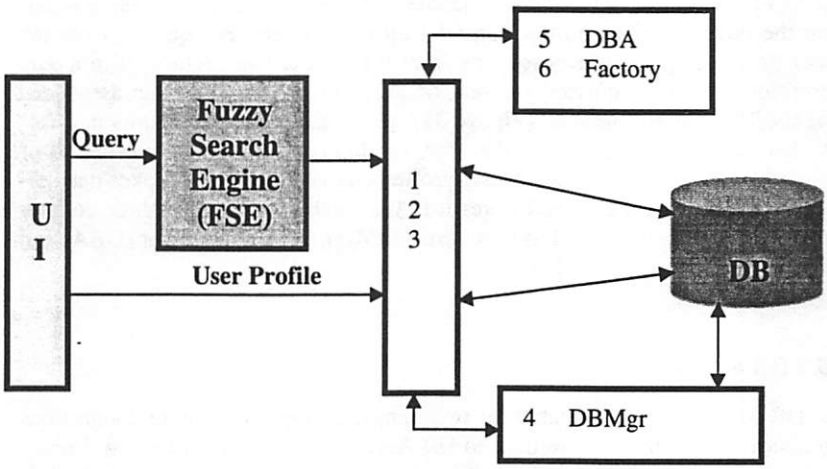


Figure 31. Database module components

5.7 Applications

In this work, we implemented our approach on four important applications: Credit ranking (scoring) (**Figure 32.a. 32.b**), which has been used to make financing decisions concerning credit cards, cars and mortgage loans; the process of college admissions where hundreds of thousands of applications are processed yearly by U.S. Universities (**Figure 33**); and date matching (**Figures 34.a and 34.b**) as one of the most popular internet programs. Even though we implemented three applications, the system is designed in a generic form to accommodate more diverse applications and to be delivered as stand-alone software to academia and businesses.

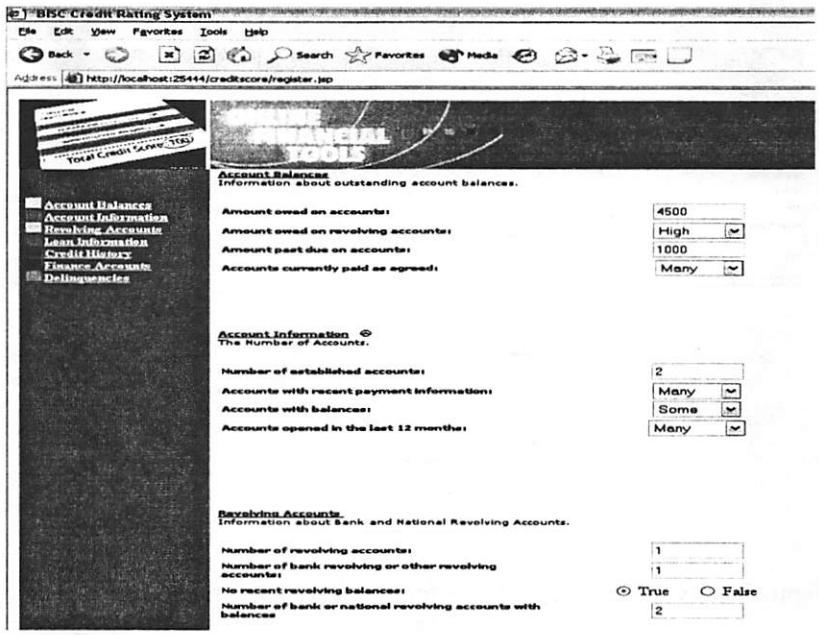


Figure 32.a. A snapshot of the variable input for credit scoring software.

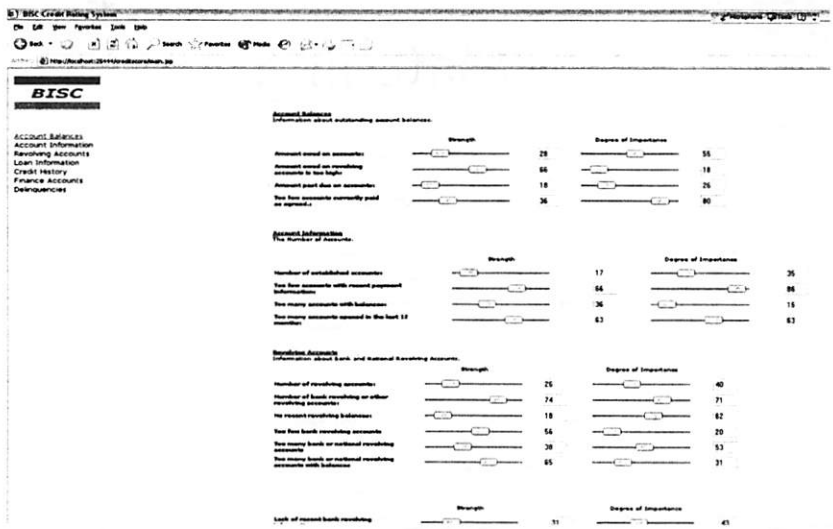


Figure 32.b. A snapshot of the software developed for credit scoring.

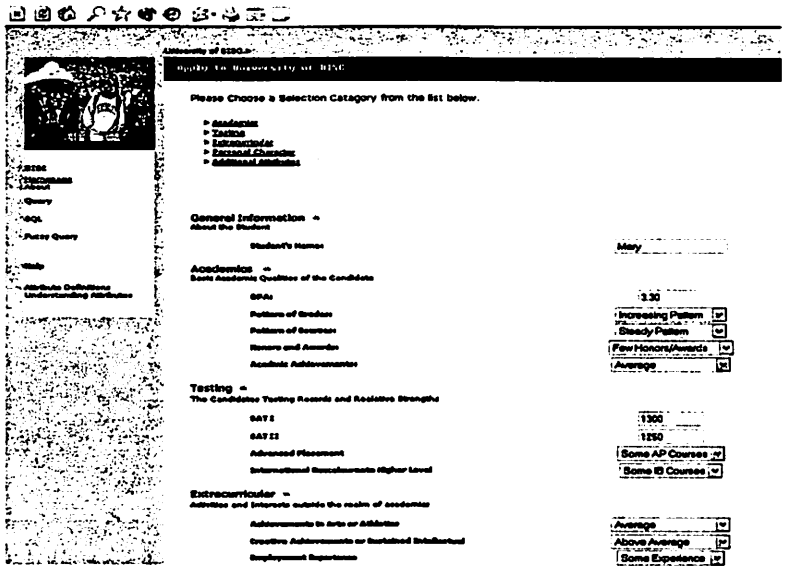


Figure 33. A snapshot of the software for University Admission Decision Making.

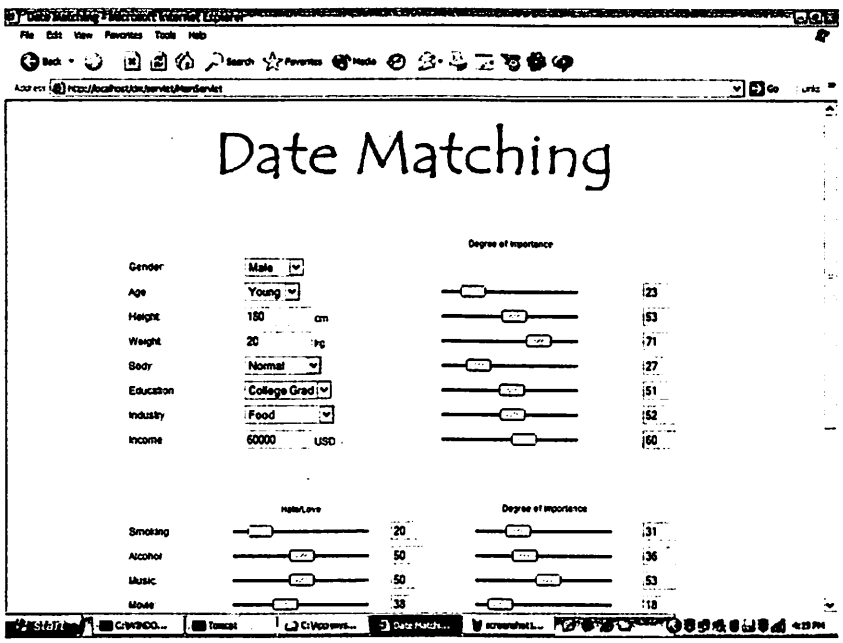


Figure 34.a. Date matching input form

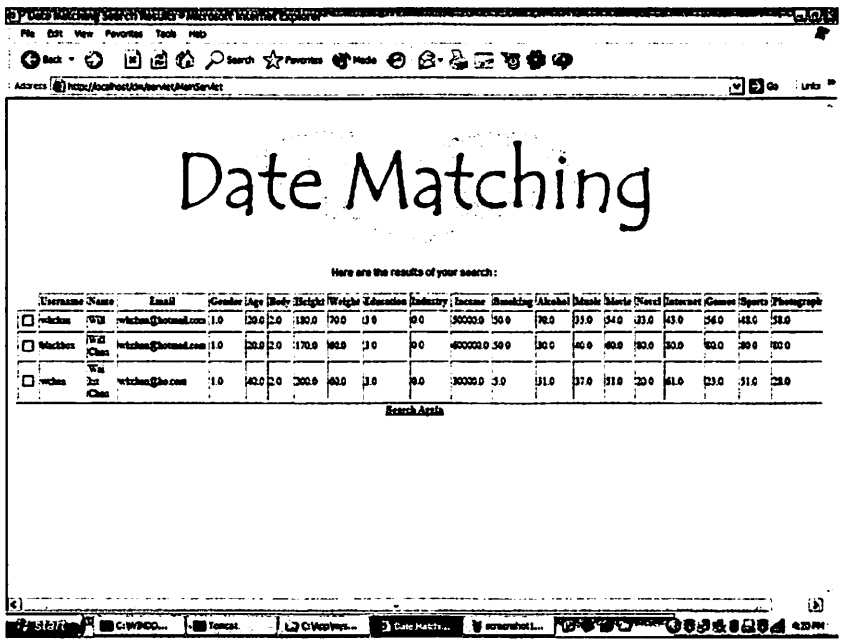


Figure 34.b. shows the results are obtained from fuzzy query using the search criteria in the previous page. The first record is the one with the highest ranking.

5.8 Evolutionary Computing for the BISC Decision Support system (EC-BISC-DSS)

In the Evolutionary Computing (EC) module of the BISC Decision Support System, our purpose is to use an evolutionary method to allow automatic adjusting of the user's preferences. These preferences can be seen as parameters of the fuzzy logic model in form of weighting of the used variables. These preferences are then represented by a weight vector and genetic algorithms will be used to fix them.

In the fuzzy logic model, the variables are combined using aggregation operators. These operators are fixed based on the application expert knowledge. However, we may have to answer to the question: how to aggregate these variables? Indeed, to make decision regarding the choice of the aggregators that have to be used in addition to the preferences the application expert might need help. We propose to automatically select the appropriate aggregators for a given application according to some corresponding training data. Moreover, we propose to combine these selected aggregators in a decision tree. In the Evolutionary Computation ap-

proach, Genetic Programming, which is an extension of Genetic Algorithms, is the closest technique to our purpose. It allows us to learn a tree structure which represents the combination of aggregators. The selection of these aggregators is included to the learning process using the Genetic Programming.

Genetic algorithms and genetic programming will be first introduced in the next section. Then, their adaptation to our decision system will be described.

5.8.1 Genetic algorithms and genetic programming

Introduced by J. Holland (1992), Genetic Algorithms (GAs) constitute a class of stochastic searching methods based on the mechanism of natural selection and genetics. They have recently received much attention in a number of practical problems notably in optimization problems as machine learning processes (Banzhaf et al., 1982).

5.8.1.1 Basic description

To solve an optimization problem, usually we need to define the search method looking for the best solution and to specify a measure of quality that allows to compare possible solutions and to find the best one. In GAs, the search space corresponds to a set of individuals represented by their DNA. These individuals are evaluated by a measure of their quality called fitness function which has to be defined according to the problem itself. The search method consists in a evolutionary process inspired by the Darwinian principle of reproduction and survival of the fittest individual.

This evolutionary process begins with a set of individuals called population. Individuals from one population are selected according to their fitness and used to form a new population with the hope to produce better individuals (offspring). The population is evolved through successive generations using genetic operations until some criterion is satisfied.

The evolution algorithm is resumed in **Figure 35**. It starts by creating randomly a population of individuals which constitute an initial generation. Each individual is evaluated by calculating its fitness. Then, a selection process is performed based on their fitness to choose individuals that participate to the evolution. Genetic operators are applied to these individuals to produce new ones. A new generation is then created by replacing existing individuals in the previous generation by the new ones. The population is evolved by repeating individuals' selection and new generations creation until the end criterion is reached in which case the evolution is stopped.

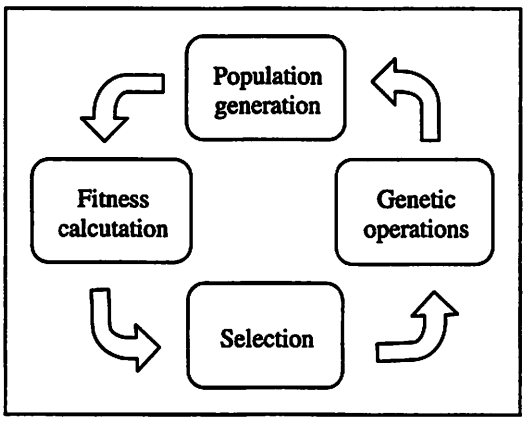


Figure 35. Genetic Algorithm Cycle

The construction of a GA for any problem can be separated into five tasks: choice of the representation of the individuals, design of the genetic operators, determination of the fitness function and the selection process, determination of parameters and variables for controlling the evolution algorithm, and definition of the termination criterion.

In the conventional GAs, individuals' DNA are usually represented by fixed-length character strings. Thus, the DNA encoding requires a selection of the string length and the alphabet size. Binary strings are the most common encoding because its relative simplicity. However, this encoding might be not natural for many problems and sometimes corrections must be made on the strings provided by genetic operations. Direct value encoding can be used in problems where use of binary encoding would be difficult. In the value encoding, an individual's DNA is represented by a sequence of some values. Values can be anything connected to the problem, such as (real) numbers.

5.8.1.2 Genetic operators

The evolution algorithm is based on the reproduction of selected individuals in the current generation breeding a new generation composed of their offspring. New individuals are created using either sexual or asexual reproduction. In sexual reproduction, known as crossover, two parents are selected and DNA from both parents is inherited by the new individual. In asexual reproduction, known as muta-

tion, the selected individual (parent) is simply copied, possibly with random changes.

Crossover operates on selected genes from parent DNA and creates new offspring. This is done by copying sequences alternately from each parent and the points where the copying crosses is chosen at random. For example, the new individual can be bred by copying everything before the crossover point from the first parent and then copy everything after the crossover point from the other parent. This kind of crossover is illustrated in **Figure 36** for the case of binary string encoding. There are other ways to make crossover, for example by choosing more crossover points. Crossover can be quite complicated and depends mainly on the encoding of DNA. Specific crossover made for a specific problem can improve performance of the GA.

Mutation is intended to prevent falling of all solutions in the population into a local optimum of the solved problem. Mutation operation randomly changes the offspring resulted from crossover. In case of binary encoding we can switch a few randomly chosen bits from 1 to 0 or from 0 to 1 (see **Figure 37**.). The technique of mutation (as well as crossover) depends mainly on the encoding of chromosomes. For example when permutations problem encoding, mutation could be performed as an exchange of two genes.

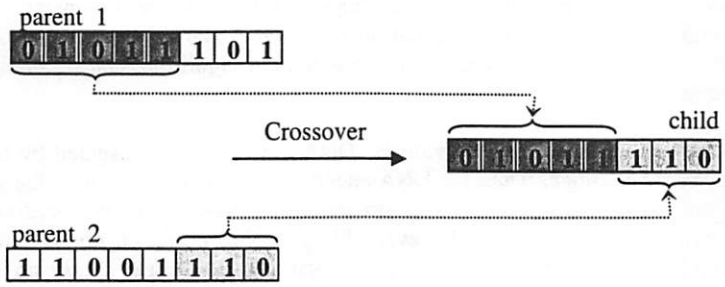


Figure 36. Genetic Algorithm - Crossover

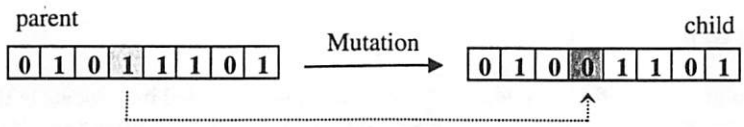


Figure 37. Genetic Algorithm - Mutation

5.8.1.3 Selection process

Individuals that participate to genetic operations are selected according to their fitness. Even that the main idea is to select the better parents in the hope that they will produce better offspring, the problem of how to do this selection remains. This can be done in many ways. We will describe briefly some of them. The (μ,λ) selection, consists in breeding λ offspring from μ parents and then μ offspring will be selected for the next generation. In the Steady-State Selection, in every generation a few good (with higher fitness) individuals are selected for creating new offspring. Then some bad (with lower fitness) individuals are removed and replaced by the new offspring. The rest of population survives to new generation. In the tournament selection, a group of individuals is chosen randomly and the best individual of the group is selected for reproduction. This kind of selection allows to give a chance to some weak individual in the population which could contain good genetic material (genes) to participate to reproduction if it is the best one in its group. Elitism selection aims at preserving the best individuals. So it first copies the best individuals to the new population. The rest of the population is constructed in ways described above. Elitism can rapidly increase the performance of GA, because it prevents a loss of the best found solution.

5.8.1.4 Parameters of GA

The outline of the Basic GA is very general. There are many parameters and settings that can be implemented differently in various problems. One particularly important parameter is the population size. On the one hand, if the population contains too few individuals, GA has few possibilities to perform crossover and only a small part of search space is explored. On the other hand, if there are too many individuals, GA slows down. Another parameter to take into account is the number of generations which can be included in the termination criterion.

For the evolution process of the GA, there are two basic parameters: crossover probability and mutation probability. The crossover probability indicates how often crossover will be performed. If there is no crossover, offspring are exact copies of parents. If there is crossover, offspring are made from parts of both parent's DNA. Crossover is made in hope that new chromosomes will contain good parts of old chromosomes and therefore the new chromosomes will be better. However, it is good to leave some part of old population survives to next generation. The mutation probability indicates how often parts of chromosome will be mutated. If there is no mutation, offspring are generated immediately after crossover (or directly copied) without any change. If mutation is performed, one or more parts of a chromosome are changed.

5.8.1.5 Genetic programming

Genetic programming (GP) is a technique pioneered by J. Koza (1992) which enables computers to solve problems without being explicitly programmed. It is an extension of the conventional GA in which each individual in the population is a computer program. It works by using GAs to automatically generate computer programs that can be represented as linear structures, trees or graphs. Tree encoding is the most used form to represent the programs. Tree structures are composed of primitive functions and terminals appropriate to the problem domain. The functions may be arithmetic operations, programming commands, and mathematical logical or domain-specific functions. To apply GP to a problem, we have to specify the set functions and terminals for the tree construction. Also, besides the parameters of the conventional GA, other parameters which are specific to the individual representation can be considered such as tree size for example.

Genetic operations are defined specifically for the type of encoding used to represent the individuals. In the case of tree encoding, new individuals are produced by removing branches from one tree and inserting them into another. This simple process ensures that the new individual is also a tree and so is also syntactically valid. The crossover and mutation operations are illustrated in **Figures 38 and 39.** The mutation consists in randomly choosing a node in the selected tree, creating a new individual and replacing the sub-tree rooted at the selected node by the created individual. The crossover operation is performed by randomly choosing nodes in the selected individuals (parents) and exchanging the sub-trees rooted at these nodes which produce two new individuals (offspring).

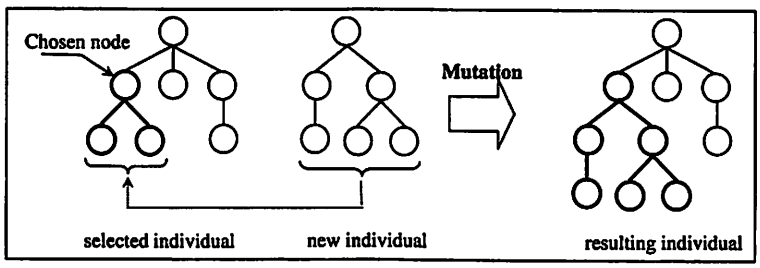


Figure 38. Genetic programming - Tree-encoding individual mutation

5.8.2 Implementation

After having introduced the GA and GP background, now we are going to describe their application to our problem. Our aim is at learning fuzzy-DSS parameters which are the weight vector representing the user preferences associated to the variables that have to be aggregated on the one hand, and the adequate decision

tree representing the combination of the aggregation operators that have to be used, on the other hand.

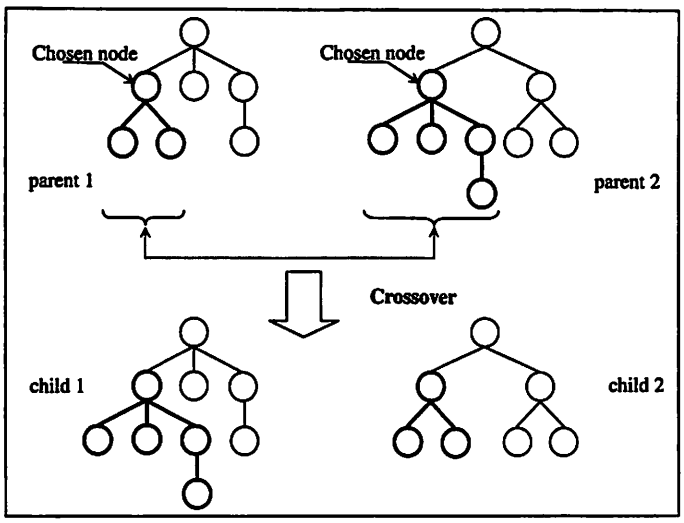


Figure 39. Genetic programming - Tree-encoding individual crossover.

5.8.2.1 Preferences learning using GA

Weight vector being a linear structure, can be represented by a binary string, in which weight values are converted to binary numbers. This binary string corresponds to the individual's DNA in the GA learning process. The goal is to find the optimal weighting of the variables. A general GA module can be used by defining a specific fitness function for each application as shown in **Figure 40.**

Let's see the example of the University Admissions application. The corresponding fitness function is shown **Figure 41.** The fitness is computed based on a training data set composed of vectors $\vec{X}_1,\cdots,\vec{X}_N$ of fuzzy values $(X_{i1},\cdots X_{ik})$ for each $\vec{X}_i$. Each value of a fuzzy variable is constituted of a crisp value between 0 and 1 and a set of membership functions. During the evolution process, for each weighting vector $(W_1W_2,\cdots,W_k)$, the corresponding fitness function is computed. Using these weights, a score is calculated for each vector. Afterward, these scores are ranked and compared with the actual ranking using similarity measure.

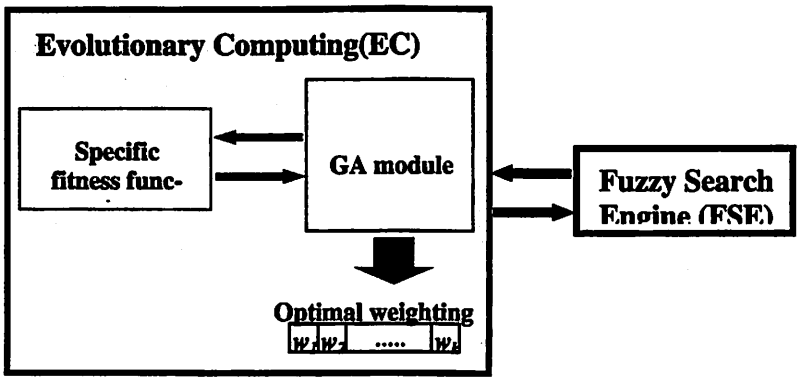


Figure 40. Evolutionary Computing Module: preferences learning.

Let's assume that we have N students and the goal is to select among them n students that will be admitted. Each student is then represented by value vector in the training data set. The similarity measure could the common vectors in the n top ones between the computed and the actual ranking. This intersection has then to be maximized. We can also consider the intersection on a larger number $n_1>n$ of top vectors. This measure can be combined to the first one with different degrees of importance. The Fitness value will be a weighted sum of these two similarity measures.

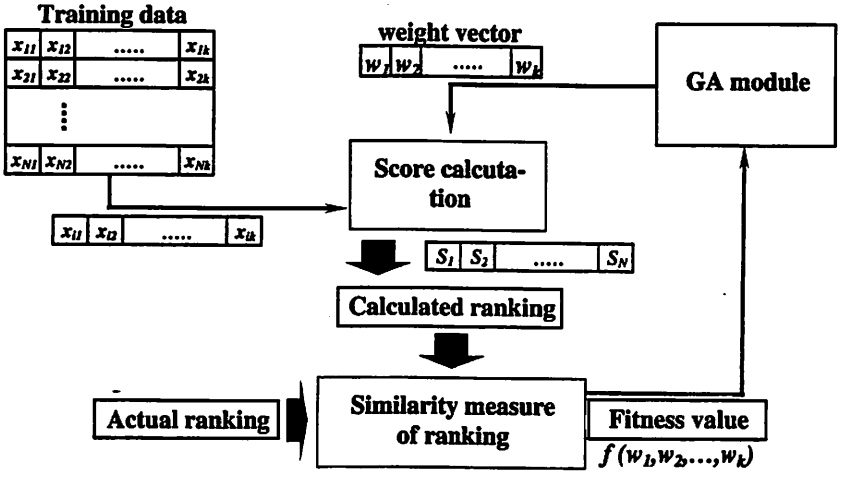


Figure 41. EC Module: Specific fitness function for the "University Admissions Application".

5.8.2.2 Aggregation tree learning using GP

We have seen the learning of the weights representing the user preferences regarding the fuzzy variables. However, the aggregators that are used are fixed in the application or by the user. But it is more interesting to adjust these aggregators automatically. We propose to include this adjustment in the GA learning process.

Aggregators can be combined in form of a tree structure which can be built using a Genetic Programming learning module. It consists in evolving a population individuals represented by tree structures. The evolution principle remains the same as in a conventional GP module but the DNA encoding needs to be defined according to the considered problem. We propose to define an encoding for aggregation trees which is more complex than for classical trees and which is common to all considered applications. As shown in **Figure 42**, we need, in addition to the fitness function specification, to define a specific encoding.

We need to specify the functions (tree nodes) and terminals that are used to build aggregation trees. The functions correspond to aggregation operators and terminals (leaves) are the fuzzy variables that have to be aggregated. Usually, in GP the used functions have a fixed number of arguments. In our case, we prefer not to fix the number of arguments for the aggregators. We might however define some restrictions such as specifying minimal and maximal number of arguments. These numbers can be considered as parameters of the learning process. This encoding property allows a largest search space to solve our problem. Another property which is indispensable specificity is the introduction of weights values in the tree structure. Instead of finding weights only for the fuzzy variables, we have to fix them also at each level of their hierarchical combination. This is done by fixing weight values for each aggregator.

Tree structures are generated randomly as in the conventional GP. But, since these trees are augmented according the properties defined above, the generation process has to be updated. So, we decided to generate randomly the number of arguments when choosing an aggregator as a node in the tree structure. And for the weights, we chose to generate them randomly for each node during its creation.

Concerning the fitness function, it is based on performing the aggregation operation a the root node of the tree that has to be evaluated. For the University Admissions application, the result of the root execution corresponds to the score that has to be computed for each value vector in the training data set. The fitness function, as in the GA learning of the user preferences, consists in simple or combined similarity measures. In addition, we can include to the fitness function a complementary measure that represent the individual's size which has to be minimized in order to avoid huge size trees.

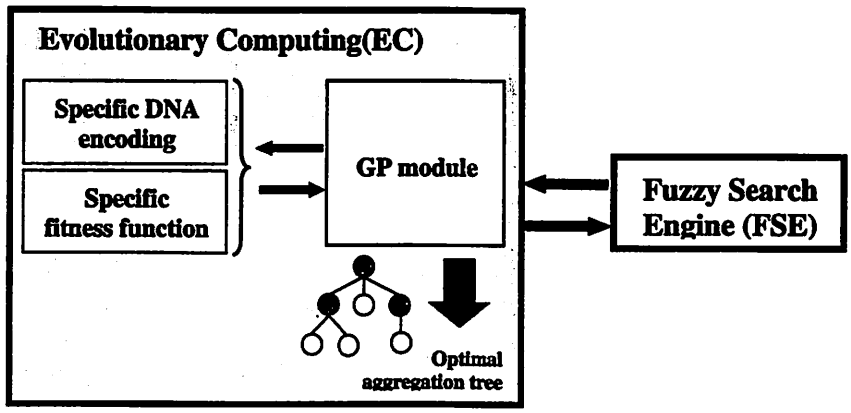


Figure 42. Evolutionary Computing Module: aggregation tree learning.

6 Conclusions

Most of the existing search systems 'software' is modeled using crisp logic and queries. In this paper, we introduced fuzzy querying and ranking as a flexible tool allowing approximation where the selected objects do not need to match exactly the decision criteria resembling natural human behavior. Searching database records and ranking the results based on multi-criteria queries is central for many database applications used within organizations in finance, business, industrial and other fields. The model consists of five major modules: the Fuzzy Search Engine (FSE), the Application Templates (AT), the User Interface (UI), the Database (DB) and the Evolutionary Computing (EC). We developed the software with many essential features. It is built as a web-based software system that users can access and use over the Internet. The system is designed to be generic so that it can run different application domains. To this end, the Application Template module provides information of a specific application as attributes and properties, and serves as a guideline structure for building a new application.

The Fuzzy Search Engine (FSE) is the core module of the system. It has been developed to be generic so that it would fit any application. The main FSE component is the query structure, which utilizes membership functions, similarity functions and aggregators.

Through the user interface a user can enter and save his profile, input criteria for a new query, run different queries and display results. The user can manually eliminate the results he disapproves or change the ranking according to his preferences.

The Evolutionary Computing (EC) module monitors ranking preferences of the users' queries. It learns to adjust to the intended meaning of the users' preferences.

The BISC decision support system key features are 1) intelligent tools to assist decision-makers in assessing the consequences of decision made in an environment of imprecision, uncertainty, and partial truth and providing a systematic risk analysis, 2) intelligent tools to be used to assist decision-makers answer "What if Questions", examine numerous alternatives very quickly and find the value of the inputs to achieve a desired level of output, and 3) intelligent tools to be used with human interaction and feedback to achieve a capability to learn and adapt through time In addition, the following important points have been found in this study 1) no single ranking function works well for all contexts, 2) most similarity measures work about the same regardless of the model, 3) there is little overlap between successful ranking functions, and 4) the same model can be used for other applications such as the design of a more intelligent search engine which includes the user's preferences and profile (Nikravesh, 2001a and 2001b). We have also described the use of evolutionary computation methods for optimization problem in the BISC decision support system. It is an original idea in combining fussy logic, machine learning and evolutionary computation. We gave some implementation precisions for the University Admissions application. We plan also to apply our system to many other applications.

7 Acknowledgement

Funding for this research was provided by the British Telecommunication (BT) and the BISC Program of UC Berkeley. The authors would like to acknowledge the effort by has been done by CS199 (BISC search engine group) and special thanks to Jonathan K. Lee, Wai-Kit Chan, Shuangyu Chang, Harmon Singh, Neema Raphael, Arthur Bobel, Naveen Sridhar, Wai-Lam Chan (Suke), Thanh Thanh Le, Trai Le, Nelly Tanizar, Kaveh Moghbeli, and Kit Hoi Lai (Andy).

8 References

1. Banzhaf, W., P. Nordin, R.E. Keller, F.D. Francone, Genetic Programming : An Introduction On the Automatic Evolution of Computer Programs and Its Applications, dpunkt.verlag and Morgan Kaufmann Publishers, San Francisco, CA, USA, 1998, 470 pages.

2. Bezdek, J.C., 1981, Pattern Recognition with Fuzzy Objective Function Algorithm, Plenum Press, New York.
3. Bonissone P.P., Decker K.S. (1986) Selecting Uncertainty Calculi and Granularity: An Experiment in Trading; Precision and Complexity, in Uncertainty in Artificial Intelligence (L. N. Kanal and J. F. Lemmer, Eds.), Amsterdam.
4. Detyniecki M (2000) Mathematical Aggregation Operators and their Application to Video Querying, Ph.D. thesis, University of Paris VI.
5. Fagin R. (1998) Fuzzy Queries in Multimedia Database Systems, Proc. ACM Symposium on Principles of Database Systems, pp. 1-10.
6. Fagin R. (1999) Combining fuzzy information from multiple systems. J. Computer and System Sciences 58, pp 83-99.
7. Fair, Isaac and Co.: http://www.fairisaac.com/.
8. Grabisch M (1996) K-order additive fuzzy measures. In Proc of 6th intl Conf on Information Processing and Management of Uncertainty in Knowledge-based Systems, Spain, pp 1345-50
9. Grabisch M, Murofushi T, Sugeno M (2000) Fuzzy Measures and Integrals:Theory and Applications, Physica-Verlag, NY
10. Holland, J. H.. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. MIT Press, 1992. First Published by University of Michigan Press 1975.
11. Jang, J.S.R., and N. Gulley, 1995, Fuzzy Logic Toolbox, The Math Works Inc., Natick, MA.
12. Kohonen, T., 1997, Self-Organizing Maps, Second Edition, Springer.Berlin.
13. Kohonen, T., 1987, Self-Organization and Associate Memory, 2nd Edition, Springer Verlag., Berlin.
14. Koza, J. R., Genetic Programming: On the Programming of Computers by Means of Natural Selection, Cambridge, Mass. : MIT Press, USA 1992, 819 pages.
15. Mizumoto M. (1989) Pictorial Representations of Fuzzy Connectives, Part I: Cases of T-norms, T-conorms and Averaging Operators, Fuzzy Sets and Systems 31, pp. 217-242.
16. Murofushi T, Sugeno M (1989) An interpretation of fuzzy measure and the Choquet integral as an integral with respect to a fuzzy measure. Fuzzy Sets and Systems, (29): pp 202-27
17. Nikravesh M. (2001a) Perception-based information processing and retrieval: application to user profiling, 2001 research summary, EECS, ERL, University of California, Berkeley, BT-BISC Project. (http://zadeh.cs.berkeley.edu/ & http://www.cs.berkeley.edu/~nikraves/ & http://www-bisc.cs.berkeley.edu/).
18. Nikravesh M. (2001b) Credit Scoring for Billions of Financing Decisions, Joint 9th IFSA World Congress and 20th NAFIPS International Conference. IFSA/NAFIPS 2001 "Fuzziness and Soft Computing in the New Millenium", Vancouver, Canada, July 25-28, 2001.
19. Masoud Nikravesh, F. Aminzadeh, and Lotfi A. Zadeh, (2003a), Soft Computing and Intelligent Data Analysis in Oil Exploration, Development in Petroleum Science, # 51, Elesevier Science B. V., The Netherlands, 2003.
20. Masoud Nikravesh and Ben Azvine (2002), Fuzzy Queries, Search, and Decision Support System, Journal of Soft Computing, Volume 6 (5), August 2002.
21. Masoud Nikravesh, B. Azvine, R. Yagar, and Lotfi A. Zadeh (2003b) "New Directions in Enhancing the power of the Internet", to be published in the Series

Studies in Fuzziness and Soft Computing, Physica-Verlag, Springer (August 2003)
22. Stanford University Admission, http://www.stanford.edu/home/stanford/facts/undergraduate.html
23. Sugeno M (1974) Theory of fuzzy integrals and its applications. Ph.D. Dissertation, Tokyo Institute of Technology.
24. U.S. Citizens for Fair Credit Card Terms; http://www.cardratings.org/cardrepfr.html.
25. University of California-Berkeley, Office of Undergraduate Admission, http://advising.berkeley.edu/ouars/.
26. Vincenzo Loia, Masoud Nikravesh and Lotfi A. Zadeh (2003), Fuzzy Logic and the Internet", to be published in the Series Studies in Fuzziness and Soft Computing, Physica-Verlag, Springer (August 2003)
27. Yager R (1988), On ordered weighted averaging aggregation operators in multicriteria decision making, IEEE transactions on Systems, Man and Cybernetics (18), 183-190.

Enhancing the Power of the Internet

Fuzz-IEEE 2003
Panel Session FlintPnl: Fuzzy Logic and the Internet
Monday, May 26, 4:30PM-6:00PM
Room: Ballroom: Salon D,
Chair: Masoud Nikravesh
Panelist: Lotfi A. Zadeh, Tomohiro Takagi, and Detlef Nauck

Short Description:

World Wide Web search engines have become the most heavily-used online services, with millions of searches performed each day. Their popularity is due, in part, to their ease of use. The central tasks for the most of the search engines can be summarize as 1) query or user information request- do what I mean and not what I say!, 2) model for the Internet, Web representation-web page collection, documents, text, images, music, etc, and 3) ranking or matching function-degree of relevance, recall, precision, similarity, etc.

Design of any new intelligent search engine should be at least based on two main motivations:

- The web environment is, for the most part, unstructured and imprecise and much of world knowledge consists of perceptions. To deal with information in the web environment what is needed is a logic that supports modes of reasoning which are approximate rather than exact. While searches may retrieve thousands of hits, finding decision-relevant and query-relevant information in an imprecise environment is a challenging problem, which has to be addressed. In addition, dealing with perception-based information is more complex and more effort intensive than dealing with measurement-based information. In this instance, as in many others, complexity is the price that has to be paid to achieve superior performance.

- Another, and less obvious, is deduction in an unstructured and imprecise environment given the huge stream of complex information. For Example; Search engines, with Google at the top, have many remarkable capabilities. But what is not among them is the deduction capability—the capability to synthesize an answer to a query by drawing on bodies of information which are resident in various parts of the knowledge base. It is this capability that differentiates a question-answering system, Q/A system for short, from a search engine. The main thrust of the fuzzy-logic-based approach to question-answering is that to achieve significant question-answering capability it is necessary to develop methods of dealing with the reality that much of world knowledge—and especially knowledge about underlying probabilities is perception-based.

From Search Engines to Question-Answering Systems
The Need for New Tools

Lotfi A. Zadeh
Berkeley Initiative in Soft Computing (**BISC**)
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
CA 94720-1776;
Telephone: 510-642-4959; Fax: 510-642-1712
Zadeh@cs.berkeley.edu

Fuzz-IEEE 2003
FLINT Special Session

Abstract:

Search engines, with Google at the top, have many remarkable capabilities. But what is not among them is the deduction capability - the capability to synthesize an answer to a query by drawing on bodies of information which are resident in various parts of the knowledge base. It is this capability that differentiates a question-answering system, Q/A system for short, from a search engine. Upgrading a search engine to a Q/A system is a complex, effort-intensive, open-ended problem. Semantic Web and related systems for upgrading quality of search may be viewed as steps in this direction. But
what may be argued, as is done in the following, is that existing tools, based as they are on bivalent logic and probability theory, have intrinsic limitations. The principal obstacle is the nature of world knowledge.

The centrality of world knowledge in human cognition, and especially in reasoning and decision-making, has long been recognized in AI. The Cyc system of Douglas Lenat is a repository of world knowledge. The problem is that much of world knowledge consists of perceptions. More specifically, perceptions are f-granular in the sense that (a) the boundaries of perceived classes are fuzzy; and (b) the perceived values of attributes are granular, with a granule being a clump of values drawn together by indistinguishability, similarity, proximity or functionality. What is not widely recognized is that f-granularity of perceptions put them well beyond the reach of computational bivalent-logic-based
theories.

Dealing with world knowledge needs new tools. A new tool which is suggested for this purpose is the fuzzy-logic-based method of computing with words and perceptions (CWP), with the understanding that perceptions are described in a natural language. A concept which plays a key role in CWP is that of Precisiated Natural Language (PNL). It is this language that is the centerpiece of our approach to reasoning and decision-making with world knowledge.

A concept which plays a key role in organization of world knowledge is that of an epistemic (knowledge-directed) lexicon (EL). Basically, an epistemic lexicon is a network of nodes and weighted links, with node i representing an object in the world knowledge database, and a weighted link from node i to node j representing the strength of association between i and j. The name of an object is a word or a composite word, e.g., car, passenger car or Ph.D. degree. An object is described by a relation or relations whose fields are attributes of the object. The values of an attribute may be granulated and associated with granulated probability and possibility

distributions. For example, the values of a granular attribute may be labeled small, medium and large, and their probabilities may be described as low, high and low, respectively. Relations which are associated with an object serve as PNL-based descriptions of the world knowledge about the object. For example, a relation associated with an object labeled Ph.D. degree may contain attributes labeled Eligibility, Length.of.study, Granting.institution, etc. The knowledge associated with an object may be context-dependent. What should be stressed is that the concept of an epistemic lexicon is intended to be employed in representation of world knowledge - which is largely perception-based - rather than Web knowledge, which is not.

In conclusion, the main thrust of the fuzzy-logic-based approach to question-answering which is outlined in this abstract, is that to achieve significant question-answering capability it is necessary to develop methods of dealing with the reality that much of world knowledge is perception-based. Dealing with perception-based information is more complex and more effort-intensive than dealing with measurement-based information. In this instance, as in many others, complexity is the price that has to be paid to achieve superior performance.

Short Bio: Prof. Lotfi A. Zadeh; BISC Director

Prof. Zadeh is a Professor in the Graduate School,Computer Science Division, Department of EECS, University of California, Berkeley. In addition, he is serving as the Director of BISC (Berkeley Initiative in Soft Computing). His earlier work was concerned in the main with systems analysis, decision analysis and information systems. His current research is focused on fuzzy logic, computing with words and soft computing. Lotfi Zadeh is a Fellow of the IEEE, AAAS, ACM, AAAI, and IFSA. He is a member of the National Academy of Engineering and a Foreign Member of the Russian Academy of Natural Sciences. He is a recipient of the IEEE Education Medal, the IEEE Richard W. Hamming Medal, the IEEE Medal of Honor, the ASME Rufus Oldenburger Medal, the B. Bolzano Medal of the Czech Academy of Sciences, the Kampe de Feriet Medal, the AACC Richard E. Bellman Central Heritage Award, the Grigore Moisil Prize, the Honda Prize, the Okawa Prize, the AIM Information Science Award, the IEEE-SMC J. P. Wohl Career Acheivement Award, the SOFT Scientific Contribution Memorial Award of the Japan Society for Fuzzy Theory, the IEEE Millennium Medal, the ACM 2000 Allen Newell Award, and other awards and honorary doctorates.

Concept-Based Information Retrieval and Search Engine

Tomohiro Takagi
Department of Computer Science,
Meiji University
1-1-1 Higashi-Mita, Tama-ku,
Kawasaki-shi, Kanagawa-ken 214-8571 Japan
+81-44-934-7469
Takagi@cs.meiji.ac.jp

Fuzz-IEEE 2003
FLINT Special Session

Abstract

Since a fuzzy set is defined by enumerating its elements and the degree of membership of each element, we can use it to express word ambiguity by enumerating all possible meanings of a word, then estimating the degrees of compatibilities between the word and the meanings.

Based on this approach, we have proposed using conceptual fuzzy sets (CFSs) to represent the various meanings of a concept that change dynamically depending on the context. A CFS (is realized as neural networks in which a node represents a concept and a link represents the strength of the relation between two (connected) concepts. The activation values agreeing with the grades of membership are determined through this associative memory. In a CFS, the meaning of a concept is represented by the distribution of the activation values of the other nodes. The distribution evolves from the activation of the node representing the concept of interest.

This presentation will start with my motivation to propose CFSs and algorithm to generate CFSs. It will describe how it works to represent the context dependent meaning of a word and to measure a conceptual distance between documents. Next, information filtering and image search (Google-Based Search Engine for Multimedia Data) will be introduced as its applications to information retrieval using capability of conceptual matching. Finally we will introduce our approach to enhancing CFSs based on brain architecture.

Short Bio: Prof. Takagi received his B.Sc from Keio University and MSc. (Fuzzy Control and Reasoning) & PhD. (Fuzzy System Identification) in Computer Science from Tokyo Institute of Technology (1979 and 1983). Prof. Takagi currently is the Professor and also Chair of Computer Science Course in graduate school of Science and Technology of Meiji University from 2000 to 2001. From 1988-1998, he was the Manager, central research laboratory and corporate multimedia promotion division at Matsushita Electric Industrial Co., LTD. He was also the deputy director at the Laboratory for International Fuzzy Engineering Research (LIFE), which was a national project supported by the Ministry of International Trade and Industry, from 1991 to 1993. From 1984 to 1988, he was the Director, Development Division, Inter-field Systems Inc. From 1983 to 84, he was the EECS research fellow, Department of Electrical Engineering Computer Science, University of California Berkeley and in 1983 he received his Doctor of engineering degree from the Tokyo Institute of Technology. He Proposed the Takagi-Sugeno model, which is one of the most popular methodologies for developing fuzzy systems in the doctoral dissertation. Prof. Takagi has over 20 years research and industrial experience and worked as consultant to major companies and

funded several key projects in the area of soft computing. He published and presented over 100 articles on diverse topics and served as technical editor and several national and international technical committees and technical chairs including advisory board or technical review committee members for both government agencies and non-government agencies throughout the world. Prof. Takagi is the Member of IEEE, IEEE Computer Society, IEEE Communications Society, IEEE Systems Man & Cybernetics Society, and Association for Computing Machinery and Japan Society for Fuzzy Theory and Systems.

Computational Intelligence to Automate and Enhance the Intelligent Data Analysis Process

Detlef Nauck
BTexact Technologies
Adastral Park, United Kindom
+44.1473.605661
detlef.nauck@bt.com

Fuzz-IEEE 2003
FLINT Special Session

Abstract:

The computerization of all aspects of our daily live and the ever-growing use of the Internet make it ever easier to collect and store data. Nowadays customers expect that businesses cater for their individual needs. In order to personalize services, intelligent data analysis (IDA) and adaptive (learning) systems are required. Simple linear statistical analysis as it is mainly used in today's businesses cannot model complex dynamic dependencies that are hidden in the collected data. IDA goes one step further than today's data mining approaches and also considers the suitability of the created solutions in terms like usability, comprehension, simplicity and cost. The intelligence in IDA comes from the expert knowledge that can be integrated in the analysis process, the knowledge-based methods used for analysis and the new knowledge created and communicated by the analysis process.

In addition to statistical methods, today we also have modern intelligent algorithms based on computational intelligence and machine learning. Computational intelligent methods like neuro-fuzzy systems and probabilistic networks or AI methods like decision trees or inductive logic programming provide new, intelligent ways for analyzing data. All these methods are part of IDA. The advantage of IDA is that it both allows the inclusion of available knowledge and the extraction of new, comprehensible knowledge about the analyzed data.

In this talk I'll describe a platform for IDA that make extensive use of computational intelligence and soft computing to automate and enhance the IDA process. This platform - SPIDA - enables us to quickly derive and implement IDA solutions. Examples of such solutions

Short Bio: Dr. Detlef Nauck is working as a Chief Research Scientist in the Computational Intelligence Group of BTexact's Research Department, where he is currently leading a research program in Intelligent Data Analysis. Before that he worked as a Senior Researcher at the Department of Computer Science of the University of Braunschweig (1990-1996) and as a Senior Research Fellow and Senior Lecturer at the Faculty of Computer Science of the Otto-von-Guericke University of Magdeburg (1996-1999). He holds a Masters degree in Computer Science (1990) and a PhD in Computer Science (1994) both from the University of Braunschweig. He also holds a Venia Legendi in Computer Science (Habilitation) from the Otto-von-Guericke University of Magdeburg (2000). His research interests are in the area of Neural Networks and Fuzzy Systems, especially in their combinations, which are known as Neuro-Fuzzy Systems. He has developed several neuro-fuzzy learning algorithms that are able to derive linguistically interpretable rules from data. Since he has joined BTexact, Dr. Nauck has worked in several Intelligent Data

Analysis projects and in a project about creating autonomous machine learning systems. Dr. Nauck has published seven books and more than 70 papers and he is a regular member of program committees for conferences on computational intelligence. He is a Visiting Senior Lecturer at the University of Magdeburg and member of IEEE and the German Society of Computer Scientists (GI). Dr. Nauck is currently a member of the steering committee of EUNITE - the European Network of Excellence on Intelligent Technologies for Smart Adaptive Systems. EUNITE is funded by the Information Society Technologies Program (IST) within the European Union's Fifth RTD Framework Program. Dr. Nauck is the chairman of the EUNITE Research Committee on Integration of Intelligent Methods.

Web Intelligence: Conceptual Search Engine and Navigation

Masoud Nikravesh
Berkeley Initiative in Soft Computing (**BISC**)
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
CA 94720-1776;
Telephone: 510-643-4522; Fax: 510-642-5775
Nikravesh@cs.berkeley.edu

Fuzz-IEEE 2003
FLINT Special Session

Abstract:

In this presentation, first we will present the role of the fuzzy logic in the Internet. Then we will present an intelligent model that can mine the Internet to conceptually match and rank homepages based on predefined linguistic formulations and rules defined by experts or based on a set of known homepages. The FCM model will be used for intelligent information and knowledge retrieval through conceptual matching of both text and images (here defined as "Concept"). The FCM can also be used for constructing fuzzy ontology or terms related to the context of the query and search to resolve the ambiguity. This model can be used to calculate conceptually the degree of match to the object or query. We will also present the integration of our technology into commercial search engines such as Google ™ and Yahoo! as a framework that can be used to integrate our model into any other commercial search engines, or development of the next generation of search engines.

Short Bio: Dr. Nikravesh is the BISC Associate Director and BTExact technology Senior Fellow in the Computer Science Division, Department of Electrical Engineering and Computer Sciences at the University of California, Berkeley and Research Scientist in the Imaging and Informatics Group at NERSC (National Energy Research Scientific Computing Division, Lawrence Berkeley National Laboratory. His credentials have led to front-page news at Lawrence Berkeley National Laboratory News and headline news at the Electronics Engineering Times. Dr. Nikravesh is the LBNL-NERSC (National Energy Research Scientific Computing Division) representative to the DiMI Executive Committee. Dr. Nikravesh has over 20 years research and industrial experience and worked as consultant to over 15 major companies and funded several key projects in the area of soft computing, data mining and fusion, control, and earth sciences through US government and major oil companies. He published and presented over 100 articles and published several books on diverse topics and served as technical editor and several national and international technical committees and technical chairs including advisory board or technical review committee members for both government agencies and non-government agencies throughout the USA and abroad. He served as member of IEEE, SPE, AICHE, SEG, AGU, and ACS.

Enhancing the Power of the Internet

Fuzz-IEEE 2003

Panel Session FlintPnl: Fuzzy Logic and the Internet
Monday, May 26, 4:30PM-6:00PM, Room: Ballroom: Salon D, Chair: Masoud Nikravesh
Panelist: Lotfi A. Zadeh, Tomohiro Takagi, and Detlef Nauck

Tuesday, May 27, 1:30PM-3:30PM, Room: Ballroom: Salon D, Chair: M. Nikravesh/O. Nasraoui

1:30PM
From search engines to question-answering systems: The need for new tools
Lotfi A. Zadeh

2:10PM
Concept-based web communities for Google search engine
Tomoe Tomiyama, Ryosuke Ohgaya, Akiyoshi Shinmura, Takayuki Kawabata, Tomohiro Takagi, and M. Nikravesh

2:30PM
Intention-aware information-delivery system
K. Hanamura, K. Kawabata, and Tomohiro Takagi

2:50PM
I-miner: A web usage mining framework using hierarchical intelligent systems
Ajith Abraham

3:10PM
An intelligent web recommendation engine based on fuzzy approximate reasoning
Olfa Nasraoui and Chris Petenes

Wednesday, May 28, 1:30PM-3:10PM, Room: Hotel Room: Salon A, Chair: M. Nikravesh/N. Mouaddib

1:30PM
A philosophical study on fuzzy sets and fuzzy applications
Tero T. Joronen

1:50PM
Fuzzy personalized wireless information agents
Yan-Qing Zhang, Wei Fan, and Jiannong Cao

2:10PM
Traffic engineering with MPLS using fuzzy logic for application in IP networks Raulison Alves Resende, Sandro M. Rossi, Akebo Yamakami, Luiz H. Bonani, and Edson Moschim

2:30PM
Improve TCP performance over ATM-UBR with FED+
Yoon-Tze Chin, Shiro Handa, Fumihito Sasamori, and Shinjiro Oshita

2:50PM
A fuzzy linguistic summarization technique for TV recommender systems
M. Gelgon, N. Mouaddib, A. Pigeau, G. Raschia, and R. Saint-Paul

From Search Engines to Question-Answering Systems The Need for New Tools

Lotfi A. Zadeh
Berkeley Initiative in Soft Computing (**BISC**)
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
CA 94720-1776;
Zadeh@eecs.berkeley.edu
Telephone: 510-642-4959; Fax: 510-642-1712

Fuzzy-IEEE 2003
FLINT Special Session

Extended Abstract

Search engines, with Google at the top, have many remarkable capabilities. But what is not among them is the deduction capability—the capability to synthesize an answer to a query by drawing on bodies of information which are resident in various parts of the knowledge base. It is this capability that differentiates a question-answering system, Q/A system for short, from a search engine.

Construction of Q/A systems has a long history in AI. Interest in Q/A systems peaked in the seventies and eighties, and began to decline when it became obvious that the available tools were not adequate for construction of systems having significant question-answering capabilities. However, Q/A systems in the form of domain-restricted expert systems have proved to be of value, and are growing in versatility, visibility and importance.

Search engines as we know them today owe their existence and capabilities to the advent of the Web. A typical search engine is not designed to come up with answers to queries exemplified by "How many Ph.D. degrees in computer science were granted by Princeton University in 1996?" or "What is the name and affiliation of the leading eye surgeon in Boston?" or "What is the age of the oldest son of the President of Finland?" or "What is the fastest way of getting from Paris to London?"

Upgrading a search engine to a Q/A system is a complex, effort-intensive, open-ended problem. Semantic Web and related systems for upgrading quality of search may be viewed as steps in this direction. But what may be argued, as is done in the following, is that existing tools, based as they are on bivalent logic and probability theory, have intrinsic limitations. The principal obstacle is the nature of world knowledge.

The centrality of world knowledge in human cognition, and especially in reasoning and decision-making, has long been recognized in AI. The Cyc system of Douglas Lenat is a repository of world knowledge. The problem is that much of world knowledge consists of perceptions. Reflecting the bounded ability of sensory organs, and ultimately the brain, to resolve detail and store information, perceptions are intrinsically imprecise. More specifically, perceptions are f-granular in the sense that (a) the boundaries of perceived classes are fuzzy; and (b) the perceived values of attributes are granular, with a granule being a clump of values drawn together by indistinguishability, similarity, proximity or functionality. What is not widely recognized is that f-granularity of perceptions put them well beyond the reach of computational bivalent-logic-based theories. For example, the meaning of a simple perception described as "Most Swedes are tall," does not admit representation in predicate logic and/or probability theory.

Dealing with world knowledge needs new tools. A new tool which is suggested for this purpose is the fuzzy-logic-based method of computing with words and perceptions (CWP), with the understanding that perceptions are described in a natural language. A concept which plays a key role in CWP is that of Precisiated Natural Language (PNL). It is this language that is the centerpiece of our approach to reasoning and decision-making with world knowledge.

A concept which plays an essential role in PNL is that of precisiability. More specifically, a proposition, p, in a natural language, NL, is PL precisiable, or simply precisiable, if it is translatable into a mathematically well-defined language termed precisiation language, PL. Examples of precisiation

languages are: the languages of propositional logic; predicate logic; modal logic; etc.; and Prolog; LISP; SQL; etc. These languages are based on bivalent logic. In the case of PNL, the precisiation language is a fuzzy-logic-based language referred to as the Generalized Constraint Language (GCL). By construction, GCL is maximally expressive.

A basic assumption underlying GCL is that, in general, the meaning of a proposition, p, in NL may be represented as a generalized constraint of the form X isr R, where X is the constrained variable; R is the constraining relation, and r is a discrete-valued variable, termed modal variable, whose values define the modality of the constraint, that is, the way in which R constrains X. The principal modalities are; possibilistic (r=blank); probabilistic (r=p); veristic (r=v); usuality (r=u); fuzzy random set (r=rs); fuzzy graph (r=fg); and Pawlak set (r=ps). In general, X, R and r are implicit in p. Thus, precisiation of p, that is, translation of p into GCL, involves explicitation of X, R and r. GCL is generated by (a) combining generalized constraints; and (b) generalized constraint propagation, which is governed by the rules of inference in fuzzy logic. The translation of p expressed as a generalized constraint is referred to as the GC-form of p, GC(p). GC(p) may be viewed as a generalization of the concept of logical form. An abstraction of the GC-form is referred to as a protoform (prototypical form) of p, and is denoted as PF(p). For example, the protoform of p: "Most Swedes are tall" is Q A's are B's, where A and B are labels of fuzzy sets, and Q is a fuzzy quantifier. Two propositions p and q are said to be PF-equivalent if they have identical protoforms. For example, "Most Swedes are tall," and "Not many professors are rich," are PF-equivalent. In effect, a protoform of p is its deep semantic structure. The protoform language, PFL, consists of protoforms of elements of GCL.

With the concepts of GC-form and protoform in place, PNL may be defined as a subset of NL which is equipped with two dictionaries: (a) from NL to GCL; and (b) from GCL to PFL. In addition, PNL is equipped with a multiagent modular deduction database, DDB, which contains rules of deduction in PFL. A simple example of a rule of deduction in PFL which is identical to the compositional rule of inference in fuzzy logic, is: if X is A and (X, Y) is B then Y is A∘B, where A∘B is the composition of A and B, defined by $\mu_B(v) = \sup_u(\mu_A(u) \wedge \mu_B(u,v))$, where μ_A and μ_B are the membership functions of A and B, respectively, and $\wedge$ is min or, more generally, a T-norm. The rules of deduction in DDB are organized into modules and submodules, with each module and submodule associated with an agent who controls execution of rules of deduction and passing results of execution.

In our approach, PNL is employed in the main to represent information in the world knowledge database (WKD). For example, the items:

- If X/Person works in Y/City then it is likely that X lives in or near Y
- If X/Person lives in Y/City then it is likely that X works in or near Y

are translated into GCL as:

Distance (Location (Residence (X/Person), Location (Work (X/Person) isu near,

where isu, read as ezoo, is th e usuality constraint. The corresponding protoform is:

F (A(B(X/C), A(E(X/C)) isu G.

A concept which plays a key role in organization of world knowledge is that of an epistemic (knowledge-directed) lexicon (EL). Basically, an epistemic lexicon is a network of nodes and weighted links, with node i representing an object in the world knowledge database, and a weighted link from node i to node j representing the strength of association between i and j. The name of an object is a word or a composite word, e.g., car, passenger car or Ph.D. degree. An object is described by a relation or relations whose fields are attributes of the object. The values of an attribute may be granulated and associated with granulated probability and possibility distributions. For example, the values of a granular attribute may be labeled small, medium and large, and their probabilities may be described as low, high and low, respectively. Relations which are associated with an object serve as PNL-based descriptions of the world knowledge about the object. For example, a relation associated with an object labeled Ph.D. degree may contain attributes labeled Eligibility, Length.of.study, Granting.institution, etc. The knowledge associated with an object may be context-dependent. What should be stressed is that the concept of an epistemic lexicon is intended to be employed in representation of world knowledge — which is largely perception- based—rather than Web knowledge, which is not.

As a very simple illustration of the use of an epistemic lexicon, consider the query "How many horses received the Ph.D. degree from Princeton University in 1996." No existing search engine would come up with the correct answer, "Zero, since a horse cannot be a recipient of a Ph.D. degree." To

generate the correct answer, the attribute Eligibility in the Ph.D. entry in EL should contain the condition "Human, usually over twenty years of age."

In conclusion, the main thrust of the fuzzy-logic-based approach to question-answering which is outlined in this abstract, is that to achieve significant question-answering capability it is necessary to develop methods of dealing with the reality that much of world knowledge—and especially knowledge about underlying probabilities is perception-based. Dealing with perception-based information is more complex and more effort-intensive than dealing with measurement-based information. In this instance, as in many others, complexity is the price that has to be paid to achieve superior performance.

Acknowledgements

Research supported in part by ONR N00014-00-1-0621, ONR Contract N00014-99-C-0298, NASA Contract NCC2-1006, NASA Grant NAC2-117, ONR Grant N00014-96-1-0556, ONR Grant FDN0014991035, ARO Grant DAAH 04-961-0341 and the BISC Program of UC Berkeley.

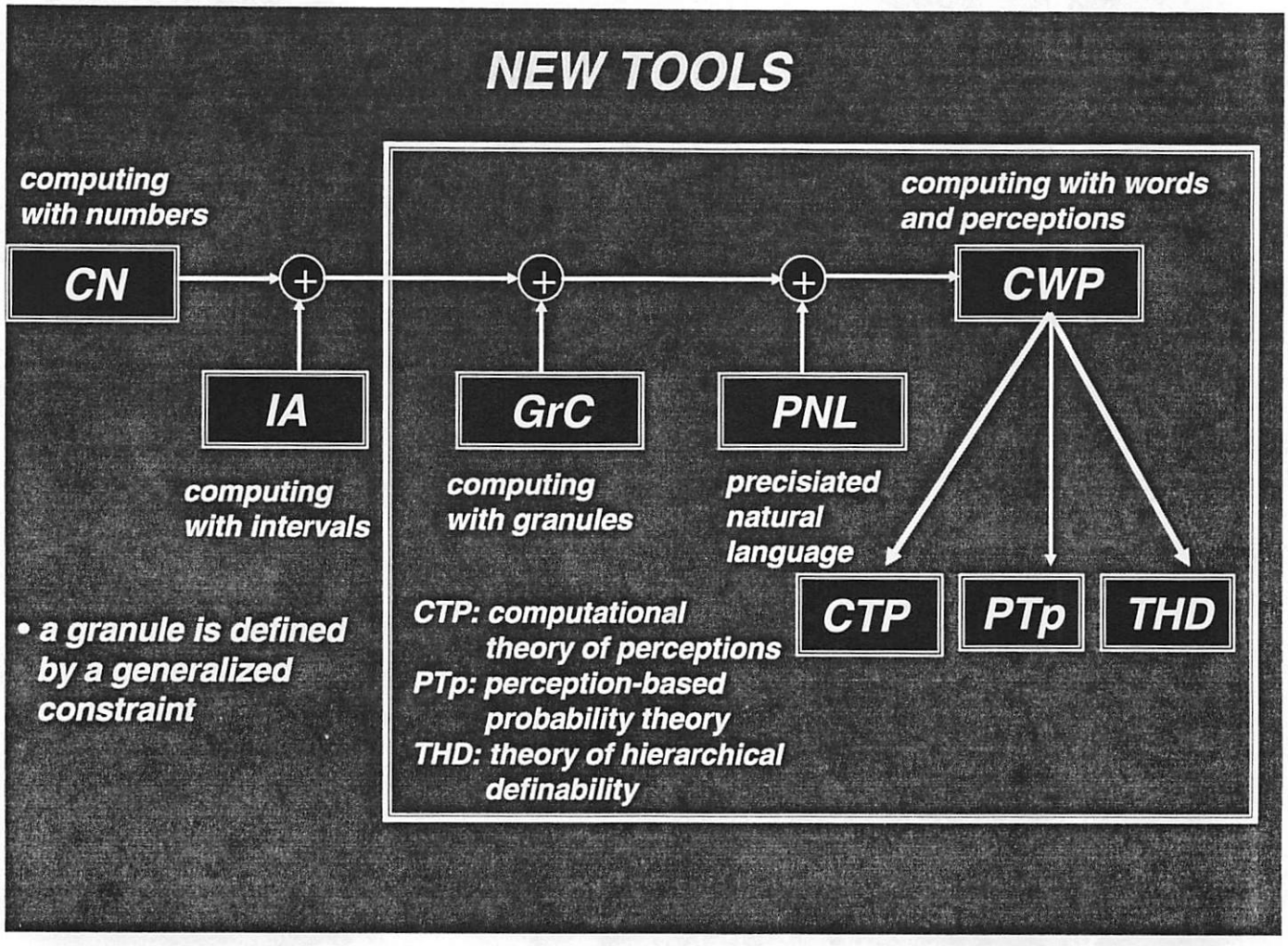
References

1. L. A. Zadeh, From Computing with Numbers to Computing with Words -- From Manipulation of Measurements to Manipulation of Perceptions, IEEE Transactions on Circuits and Systems, 45, 105-119, 1999.
2. L. A. Zadeh, "A new direction in AI: Towards a Computational Theory of Perceptions," *AI magazine*, vol. 22, pp. 73--84, 2001.
3. L.A. Zadeh, Toward a Perception-based Theory of Probabilistic Reasoning with Imprecise Probabilities, *Journal of Statistical Planning and Inference*, 105 233–264, 2002.
4. L. A. Zadeh and M. Nikravesh, Perception-Based Intelligent Decision Systems; Office of Naval Research, Summer 2002 Program Review, Covel Commons, University of California, Los Angeles, July 30th-August 1st, 2002.
5. M. Nikravesh and B. Azvine; New Directions in Enhancing the Power of the Internet, Proc. Of the 2001 BISC Int. Workshop, University of California, Berkeley, Report: UCB/ERL M01/28, August 2001.
6. V. Loia , M. Nikravesh, L. A. Zadeh, *Journal of Soft Computing*, Special Issue; fuzzy Logic and the Internet, Springer Verlag, Vol. 6, No. 5; August 2002.
7. M. Nikravesh, R. Yager and L. A. Zadeh, "Enhancing the Power of the Internet", to be published in the Series Studies in Fuzziness and Soft Computing, Physica-Verlag, Springer (August 2003).

Short Bio: Prof. Lotfi A. Zadeh; BISC Director

Prof. Zadeh is a Professor in the Graduate School,Computer Science Division, Department of EECS, University of California, Berkeley. In addition, he is serving as the Director of BISC (Berkeley Initiative in Soft Computing). His earlier work was concerned in the main with systems analysis, decision analysis and information systems. His current research is focused on fuzzy logic, computing with words and soft computing. Lotfi Zadeh is a Fellow of the IEEE, AAAS, ACM, AAAI, and IFSA. He is a member of the National Academy of Engineering and a Foreign Member of the Russian Academy of Natural Sciences. He is a recipient of the IEEE Education Medal, the IEEE Richard W. Hamming Medal, the IEEE Medal of Honor, the ASME Rufus Oldenburger Medal, the B. Bolzano Medal of the Czech Academy of Sciences, the Kampe de Feriet Medal, the AACC Richard E. Bellman Central Heritage Award, the Grigore Moisil Prize, the Honda Prize, the Okawa Prize, the AIM Information Science Award, the IEEE-SMC J. P. Wohl Career Acheivement Award, the SOFT Scientific Contribution Memorial Award of the Japan Society for Fuzzy Theory, the IEEE Millennium Medal, the ACM 2000 Allen Newell Award, and other awards and honorary doctorates.

BISC Program of the EECS Department-Computer Sciences Division-University of California-Berkeley, is the world-leading center for basic and applied research in soft computing. The principal constituents of soft computing (SC) are fuzzy logic (FL), neural network theory (NN) and probabilistic reasoning (PR), with the latter subsuming belief networks, evolutionary computing including DNA computing, chaos theory and parts of learning theory. Some of the most striking achievements of BISC Program are: fuzzy reasoning (set and logic), new soft computing algorithms making intelligent, semi-unsupervised use of large quantities of complex data, uncertainty analysis, perception-based decision analysis and decision support systems for risk analysis and management, computing with words, computational theory of perception (CTP), and precisiated natural language (PNL).


NEW TOOLS
computing with numbers
CN
IA
computing with intervals
GrC
computing with granules
PNL
precisiated natural language
computing with words and perceptions
CWP
CTP
PTp
THD
• a granule is defined by a generalized constraint
CTP: computational theory of perceptions
PTp: perception-based probability theory
THD: theory of hierarchical definability

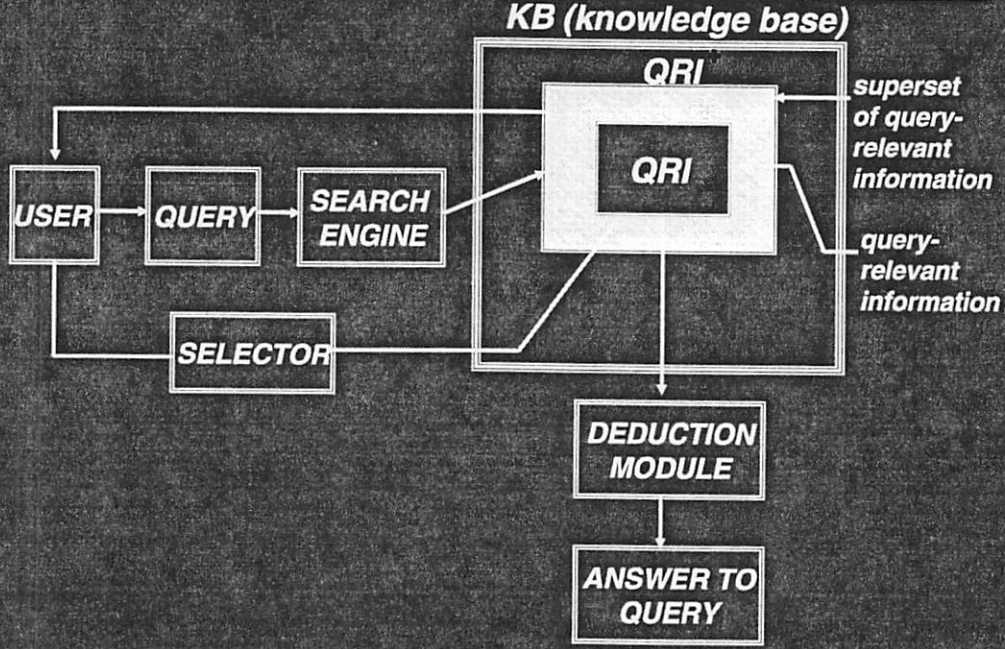
INTELLIGENT DECISION SYSTEM

- INFORMATION-ON-DEMAND MODULE
- INFORMATION PROFERRAL MODULE
- INFORMATION ALERT MODULE

INFORMATION-ON-DEMAND MODULE=Q/A SYSTEM
Q/A SYSTEM=SEARCH ENGINE + DEDUCTION MODULE

BASIC STRUCTURE OF QUESTION-ANSWERING

QUESTION-ANSWERING=SEARCH+DEDUCTION



Conceptual Fuzzy Sets and its Application to Web Information Retrieval

Tomohiro Takagi
Department of Computer Science,
Meiji University
1-1-1 Higashi-Mita, Tama-ku,
Kawasaki-shi, Kanagawa-ken 214-8571 Japan

+81-44-934-7469
Takagi@cs.meiji.ac.jp

Fuzzy-IEEE 2003
FLINT Special Session

Extended Abstract

Since a fuzzy set is defined by enumerating its elements and the degree of membership of each element, we can use it to express word ambiguity by enumerating all possible meanings of a word, then estimating the degrees of compatibilities between the word and the meanings. Based on this approach, we have proposed using conceptual fuzzy sets (CFSs) to represent the various meanings of a concept that change dynamically depending on the context. We applied the conceptual fuzzy set to web information retrieval based on its capability to measure conceptual distance between documents.

Conceptual Fuzzy Sets

Main cause of vagueness is ambiguity in the language. According to the theory of "meaning representation from use" proposed by Wittgenstein the various meanings of a word can be represented by other words, and we can assign grades of activation showing the degree of compatibility between labels. A CFS achieves this by using distributions of activations. In a CFS, the activation values agree with the grades of membership and the meaning of a concept is represented by the distribution of the activation values of the other nodes. Because the distribution changes depending on which labels are activated as a result of the conditions the activations show a context-dependent meaning. When more than two labels are activated, a CFS is generated by the overlapping propagations of their activations. In the CFSs, words may have synonymous, antonymous, hypernymous and hyponymous relation to other words. These relations are too complicated to be represented in a hierarchical structure. In this talk, we introduce RBF-like networks to generate CFSs.

Let's think about Java. If we are talking about computers, "java" will be understood as a programming language. If we are looking at a menu at a cafe, it will be understood as a kind of coffee. Its meaning is thus determined by context generated by the presence of related words, such as FORTRAN and C. Experimental results showed that CFSs provide us deferent meaning representations in deferent contexts.

Web community distillation

We applied CFSs to cluster web pages and distill their communities. The applied system processes web pages along with following steps and distills communities of pages. The system is roughly divided into two parts; filtering part (steps 1-6) and classifying part (step 7).

1. Obtain web pages that are similar or linked to a sample web page using Google™.
2. Analyze each HTML file obtained step 1 and generate a word vector.
   - Nouns and adjectives are extracted from the HTML file
   - TF-IDF values are calculated and attached to the words
3. Input the word vector into CFSs unit. Propagation of activation occurs from input word vector in the CFSs unit. The meanings of the keywords are represented in other expanded words regarding context.
4. Input the expanded word vector into SVM unit. The SVM unit determines whether the word vector matches to a topic or not, and store the URLs being positive into a database.
5. Repeat steps 2-4 for all HTML files resulted in step 1.
6. Repeat steps 1-5 until there are no new pages

found.
7. Classify all web pages in the database and distill communities.

To evaluate this system we selected actual hope pages and simulated community distillation. The results show that conceptual expansion using CFSs has effect to restrain unnecessary words and to emphasize important ones. CFSs also provide us very effective and conceptual measurement performance among text notes.

Image search

We improved Google™ image search capability in two steps as follows.

<Step 1: Relevance feed back>

We developed the system to perform followings relevance feedback circulations. Experimental results show that the relevance feed back improved the image search capability.
- send user's query to Google™
- Google™ sends back retrieved images
- show the images and let the user to select positive examples
- refine query and send it to Google™

<Step 2: Query expansion using CFS>

In the step 1, when popular words are used as a query, such as "cat", relevance feed back did not work well. Popular images are frequently contained in web pages with weak relations with texts in the pages. In this step, CFSs extend query and the results are blended to word vector obtained from analysis of HTML files in the relevance feed back. Experimental results show significantly better results comparing with simple relevance feedback.

Acknowledgements

Partial funding for this research was provided by the British Telecommunication (BT) and the BISC Program of UC Berkeley.

References

1. M. Nikravesh, et al., Web Intelligence, Conceptual-Based Model, Book Chapter in Enhancing the Power of the Internet, edited by Nikravesh et al., Studies in Fuzziness and Soft Computing, Springer-Verlag (To be Published, 2003).
2. T. Takagi, et al., Conceptual Fuzzy Sets as a Meaning Representation and their Inductive Construction, International Journal of Intelligent Systems, Vol. 10, 929-945 (1995).
3. T. Takagi, et al., Multilayered Reasoning by Means of Conceptual Fuzzy Sets, International Journal of Intelligent Systems, Vol. 11, 97-111 (1996).
4. T. Takagi and M.Tajima, Proposal of a Search Engine based on Conceptual Matching of Text Notes, IEEE International Conference on Fuzzy Systems FUZZ-IEEE'2001, S406- (2001)
5. T. Takagi, Ket el., Exposure of Illegal Website using Conceptual Fuzzy Sets based Information Filtering System, the North American Fuzzy Information Processing Society - The Special Interest Group on Fuzzy Logic and the Internet NAFIPS-FLINT 2002, 327-332 (2002)
6. T. Takagi, et al., Conceptual Fuzzy Sets-Based Menu Navigation System for Yahoo!, the North American Fuzzy Information Processing Society - The Special Interest Group on Fuzzy Logic and the Internet NAFIPS-FLINT 2002, 274-279 (2002)
7. V. Loia , M. Nikravesh, L. A. Zadeh, *Journal of Soft Computing*, Special Issue; fuzzy Logic and the Internet, Springer Verlag, Vol. 6, No. 5; August 2002.
8. M. Nikravesh, et. al, "Enhancing the Power of the Internet", to be published in the Series Studies in Fuzziness and Soft Computing, Physica-Verlag, Springer (August 2003).
9. M. Nikravesh, Fuzzy Logic and Internet: Perception Based Information Processing and Retrieval, Berkeley Initiative in Soft Computing, Report No. 2001-2-SI-BT, September 2001a.
10. M. Nikravesh, BISC and The New Millennium, Perception-based Information Processing, Berkeley Initiative in Soft Computing, Report No. 2001-1-SI, September 2001b.
11. M. Nikravesh, et al., Fuzzy logic and the Internet (FLINT), Internet, World Wide Web, and Search Engines, International Journal of Soft Computing-Special Issue in fuzzy logic and the Internet , 2002
12. M. Nikravesh, Fuzzy Conceptual-Based Search Engine using Conceptual Semantic Indexing, NAFIPS-FLINT 2002, June 27-29, New Orleans, LA, USA
13. M. Nikravesh and B. Azvin, Fuzzy Queries, Search, and Decision Support System, International Journal of Soft Computing-Special Issue in fuzzy logic and the Internet , 2002
14. M. Nikravesh, V. Loia, and B. Azvine, Fuzzy logic and the Internet (FLINT), Internet, World Wide Web, and Search Engines, to be appeared in International Journal of Soft Computing-Special Issue in fuzzy logic and the Internet , 2002

15. M. Nikravesh, Fuzzy Conceptual-Based Search Engine using Conceptual Semantic Indexing, NAFIPS-FLINT 2002, June 27-29, New Orleans, LA, USA
16. V. Loia, M. Nikravesh and Lotfi A. Zadeh, "Fuzzy Logic an the Internet", to be published in the Series Studies in Fuzziness and Soft Computing, Physica-Verlag, Springer (August 2003)

Short Bio

Prof. Takagi received his B.Sc from Keio University and MSc. (Fuzzy Control and Reasoning) & PhD. (Fuzzy System Identification) in Computer Science from Tokyo Institute of Technology (1979 and 1983). Prof. Takagi currently is the Professor and also Chair of Computer Science Course in graduate school of Science and Technology of Meiji University from 2000 to 2001. From 1988-1998, he was the Manager, central research laboratory and corporate multimedia promotion division at Matsushita Electric Industrial Co., LTD. He was also the deputy director at the Laboratory for International Fuzzy Engineering Research (LIFE), which was a national project supported by the Ministry of International Trade and Industry, from 1991 to 1993. From 1984 to 1988, he was the Director, Development Division, Inter-field Systems Inc. From 1983 to 84, he was the EECS research fellow, Department of Electrical Engineering Computer Science, University of California Berkeley and in 1983 he received his Doctor of engineering degree from the Tokyo Institute of Technology. He Proposed the Takagi-Sugeno model, which is one of the most popular methodologies for developing fuzzy systems in the doctoral dissertation. Prof. Takagi has over 20 years research and industrial experience and worked as consultant to major companies and funded several key projects in the area of soft computing. He published and presented over 100 articles on diverse topics and served as technical editor and several national and international technical committees and technical chairs including advisory board or technical review committee members for both government agencies and non-government agencies throughout the world. Prof. Takagi is the Member of IEEE, IEEE Computer Society, IEEE Communications Society, IEEE Systems Man & Cybernetics Society, and Association for Computing Machinery and Japan Society for Fuzzy Theory and Systems.

Web Intelligence: Conceptual Search Engine and Navigation

Masoud Nikravesh
Berkeley Initiative in Soft Computing (**BISC**)
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
CA 94720-1776;
Nikravesh@eecs.berkeley.edu
Telephone: 510-643-4522; Fax: 510-642-5775

Fuzzy-IEEE 2003
FLINT Special Session

Extended Abstract

World Wide Web search engines have become the most heavily-used online services, with millions of searches performed each day. Their popularity is due, in part, to their ease of use. The central tasks for the most of the search engines can be summarize as 1) query or user information request-do what I mean and not what I say!, 2) model for the Internet, Web representation-web page collection, documents, text, images, music, etc, and 3) ranking or matching function-degree of relevance, recall, precision, similarity, etc.

Design of any new intelligent search engine should be at least based on two main motivations:

i· The web environment is, for the most part, unstructured and imprecise. To deal with information in the web environment what is needed is a logic that supports modes of reasoning which are approximate rather than exact. While searches may retrieve thousands of hits, finding decision-relevant and query-relevant information in an imprecise environment is a challenging problem, which has to be addressed.

ii· Another, and less obvious, is deduction in an unstructured and imprecise environment given the huge stream of complex information.

One can use clarification dialog, user profile, context, and ontology, into an integrated frame work to design a more intelligent search engine. The model will be used for intelligent information and knowledge retrieval through conceptual matching of text. The selected query doesn't need to match the decision criteria exactly, which gives the system a more human-like behavior. The model can also be used for constructing ontology or terms related to the context of search or query to resolve the ambiguity. The new model can execute conceptual matching dealing with context-dependent word ambiguity and produce results in a format that permits the user to interact dynamically to customize and personalized its search strategy.

It is also possible to automate ontology generation and document indexing using the terms similarity based on Conceptual-Latent Semantic Indexing Technique (CLSI). Often time it is hard to find the "right" term and even in some cases the term does not exist.

The ontology is automatically constructed from text document collection and can be used for query refinement. It is also possible to generate conceptual documents similarity map that can be used for intelligent search engine based on CLSI, personalization and user profiling. The user profile is automatically constructed from text document collection and can be used for query refinement and provide suggestions and for ranking the information based on pre-existence user profile.

Given the ambiguity and imprecision of the "concept" in the Internet, which may be described by both textual and image information, the use of Fuzzy Conceptual Matching (FCM) is a necessity for search engines. In the FCM approach, the "concept" is defined by a series of keywords with different weights depending on the importance of each keyword. Ambiguity in concepts can be defined by a set of imprecise concepts. Each imprecise concept in fact can be defined by a set of fuzzy concepts. The fuzzy concepts can then be related to a set of imprecise words given the context. Imprecise words can then be translated into precise words given the ontology and ambiguity resolution through clarification dialog. By constructing the ontology and fine-tuning the strength of links (weights), we could construct a fuzzy set to integrate piecewise the imprecise concepts and precise words to define the ambiguous concept.

In this presentation, first we will present the role of the fuzzy logic in the Internet. Then we will present an intelligent model that can mine the Internet to conceptually match and rank homepages based on predefined linguistic formulations and rules defined by experts or based on a set of known homepages. The FCM model will be used for intelligent information and knowledge retrieval through conceptual matching of both text and images (here defined as "Concept"). The FCM can also be used for constructing fuzzy ontology or terms related to the context of the query and search to resolve the ambiguity. This model can be used to calculate conceptually the degree of match to the object or query. We will also present the integration of our technology into commercial search engines such as Google ™ and Yahoo! as a framework that can be used to integrate our model into any other commercial search engines, or development of the next generation of search engines.

Challenges and Road Ahead

During the August 2001, BISC program hosted a workshop toward better understanding of the issues related to the Internet (Fuzzy Logic and the Internet-FLINT2001, Toward the Enhancing the Power of the Internet). The main purpose of the Workshop was to draw the attention of the fuzzy logic community as well as the Internet community to the fundamental importance of specific Internet-related problems. This issue is critically significant about problems that center on search and deduction in large, unstructured knowledge bases. The Workshop provided a unique opportunity for the academic and corporate communities to address new challenges, share solutions, and discuss research directions for the future. Following are the areas that were recognized as challenging problems and the new direction toward the next generation of the search engines and Internet. We summarize the challenges and the road ahead into four categories as follows:

- ***Search Engine and Queries:***
    - Deductive Capabilities
    - Customization and Specialization
    - Metadata and Profiling
    - Semantic Web
    - Imprecise-Querying
    - Automatic Parallelism via Database Technology
    - Approximate Reasoning
    - Ontology
    - *Ambiguity Resolution through Clarification Dialog; Definition/Meaning & Specificity*User Friendly
    - Multimedia
    - Databases
    - Interaction
- ***Internet and the Academia:***
    - Ambiguity and Conceptual and Ontology
    - Aggregation and Imprecision Query
    - Meaning and structure Understanding
    - Dynamic Knowledge
    - Perception, Emotion, and Intelligent Behavior
    - Content-Based
    - Escape from Vector SpaceDeductive Capabilities
    - Imprecise-Querying
    - *Ambiguity Resolution through Clarification Dialog*
    - *Precisiated Natural Languages (PNL)*
- ***Internet and the Industry:***
    - XML=>Semantic Web
    - Workflow
    - Mobile E-Commerce
    - CRM
    - Resource Allocation
    - Intent
    - Ambiguity Resolution
    - Interaction
    - Reliability
    - Monitoring
    - Personalization and Navigation
    - Decision Support
    - Document Soul
    - Approximate Reasoning
    - Imprecise QueryContextual Categorization
- ***Fuzzy Logic and Internet; Fundamental Research:***
    - Computing with Words (CW)
    - Computational Theory of Perception (CTP)
    - Precisiated Natural Languages (PNL)

The potential areas and applications of Fuzzy Logic for the Internet include:

- ***Potential Areas:***

- Search Engines
- Retrieving Information
- Database Querying
- Ontology
- Content Management
- Recognition Technology
- Data Mining
- Summarization
- Information Aggregation and Fusion
- E-Commerce
- Intelligent Agents
- Customization and Personalization

- ***Potential Applications:***

  - Search Engines and Web Crawlers
  - Agent Technology (i.e., Web-Based Collaborative and Distributed Agents)
  - Adaptive and Evolutionary techniques for dynamic environment (i.e. Evolutionary search engine and text retrieval, Dynamic learning and adaptation of the Web Databases, etc)
  - Fuzzy Queries in Multimedia Database Systems
  - Query Based on User Profile
  - Information Retrievals
  - Summary of Documents
  - Information Fusion Such as Medical Records, Research Papers, News, etc
  - Files and Folder Organizer
  - Data Management for Mobile Applications and eBusiness Mobile Solutions over the Web
  - Matching People, Interests, Products, etc
  - Association Rule Mining for Terms-Documents and Text Mining
  - E-mail Notification
  - Web-Based Calendar Manager
  - Web-Based Telephony
  - Web-Based Call Centre
  - Workgroup Messages
  - E-Mail and Web-Mail
  - Web-Based Personal Info
  - Internet related issues such as Information overload and load balancing, Wireless Internet-coding and D-coding (Encryption), Security such as Web security and Wireless/Embedded Web Security, Web-based Fraud detection and prediction, Recognition, issues related to E-commerce and E-bussiness, etc.

Conclusions

Intelligent search engines with growing complexity and technological challenges are currently being developed. This requires new technology in terms of understanding, development, engineering design and visualization. While the technological expertise of each component becomes increasingly complex, there is a need for better integration of each component into a global model adequately capturing the imprecision and deduction capabilities. In addition, intelligent models can mine the Internet to conceptually match and rank homepages based on predefined linguistic formulations and rules defined by experts or based on a set of known homepages. The FCM model can be used as a framework for intelligent information and knowledge retrieval through conceptual matching of both text and images (here defined as "Concept"). The FCM can also be used for constructing fuzzy ontology or terms related to the context of the query and search to resolve the ambiguity. This model can be used to calculate conceptually the degree of match to the object or query.

Future Works

TIKManD (Tool for Intelligent Knowledge Management and Discovery)

In the future work, we intent to develop and deploy an intelligent computer system is called ***"TIKManD (Tool for Intelligent Knowledge Management and Discovery)"***.

The system can mine Internet homepages, Emails, Chat Lines, and/or authorized wire tapping information (which may include Multi-Lingual information) to recognize, conceptually match, and rank potential terrorist and criminal activities (both common and unusual) by the type and seriousness of the activities. This will be done automatically or semi-automatically based on predefined linguistic formulations and rules defined by experts or based on a set of known terrorist activities given the information provided through law enforcement databases (text and voices) and huge number of "tips" received immediately after the attack. Conceptual Fuzzy Set (CFS) model will be used for intelligent information and knowledge retrieval through conceptual matching of text, images and

voice (here defined as "Concept"). The CFS can be also used for constructing fuzzy ontology or terms relating the context of the investigation (Terrorism or other criminal activities) to resolve the ambiguity. This model can be used to calculate conceptually the degree of match to the object or query. In addition, the ranking can be used for intelligently allocating resources given the degree of match between objectives and resources available.

Google™ and Yahoo! Concept-Based Search Engine

There are two type of search engine that we are interested and are dominating the Internet. First, the most popular search engines that are mainly for unstructured data such as Google ™ and Teoma which are based on the concept of Authorities and Hubs. Second, search engines that are task spcifics such as 1) Yahoo!: manually-pre-classified, 2) NorthernLight: Classification, 3) Vivisimo: Clustering, 4) Self-organizing Map: Clustering + Visualization and 5) AskJeeves: Natural Languages-Based Search; Human Expert.

Google uses the PageRank and Teoma uses HITS (Ding et al. 2001) for the Ranking. To develop such models, state-of-the-art computational intelligence techniques are needed. These include and are not limited to:

- Latent-Semantic Indexing and SVD for preprocessing,
- Radial-Basis Function Network to develop concepts,

- Support Vector Machine (SVM) for supervised classification,
- fuzzy/neuro-fuzzy clustering for unsupervised classification based on both conventional learning techniques and Genetic and Reinforcement learning,
- non-linear aggregation operators for data/text fusion,
- automatic recognition using fuzzy measures and a fuzzy integral approach
- self organization map and graph theory for building community and clusters,
- both genetic algorithm and reinforcement learning to learn the preferences,
- fuzzy-integration-based aggregation technique and hybrid fuzzy logic-genetic algorithm for decision analysis, resource allocation, multi-criteria decision-making and multi-attribute optimization.
- text analysis: next generation of the Text, Image Retrieval and concept recognition based on soft computing technique and in particular Conceptual Search Model (CSM). This includes
  - Understanding textual content by retrieval of relevant texts or paragraphs using CSM followed by clustering analysis.
  - Hierarchical model for CSM
  - Integration of Text and Images based on CSM
  - CSM Scalability, and
  - The use of CSM for development of
    - Ontology
    - Query Refinement and Ambiguity Resolution
    - Clarification Dialog
    - Personalization-User Profiling

Acknowledgements

Funding for this research was provided by the British Telecommunication (BT) and the BISC Program of UC Berkeley.

References

1. L. A. Zadeh, From Computing with Numbers to Computing with Words -- From Manipulation of Measurements to Manipulation of Perceptions, IEEE Transactions on Circuits and Systems, 45, 105-119, 1999.
2. L. A. Zadeh, "A new direction in AI: Towards a Computational Theory of Perceptions," *AI magazine*, vol. 22, pp. 73--84, 2001.
3. L.A. Zadeh, Toward a Perception-based Theory of Probabilistic Reasoning with Imprecise Probabilities, *Journal of Statistical Planning and Inference*, 105 233–264, 2002.
4. L. A. Zadeh and M. Nikravesh, Perception-Based Intelligent Decision Systems; Office of Naval Research, Summer 2002 Program Review, Covel Commons, University of California, Los Angeles, July 30th-August 1st, 2002.
5. M. Nikravesh and B. Azvine; New Directions in Enhancing the Power of the Internet, Proc. Of the 2001 BISC Int. Workshop, University of California, Berkeley, Report: UCB/ERL M01/28, August 2001.
6. V. Loia , M. Nikravesh, L. A. Zadeh, *Journal of Soft Computing*, Special Issue; fuzzy Logic and the Internet, Springer Verlag, Vol.

6, No. 5; August 2002.
7. M. Nikravesh, et. al, "Enhancing the Power of the Internet", to be published in the Series Studies in Fuzziness and Soft Computing, Physica-Verlag, Springer (August 2003).
8. M. Nikravesh, Fuzzy Logic and Internet: Perception Based Information Processing and Retrieval, Berkeley Initiative in Soft Computing, Report No. 2001-2-SI-BT, September 2001a.
9. M. Nikravesh, BISC and The New Millennium, Perception-based Information Processing, Berkeley Initiative in Soft Computing, Report No. 2001-1-SI, September 2001b.
10. M. Nikravesh, V. Loia,, and B. Azvine, Fuzzy logic and the Internet (FLINT), Internet, World Wide Web, and Search Engines, International Journal of Soft Computing-Special Issue in fuzzy logic and the Internet , 2002
11. M. Nikravesh, Fuzzy Conceptual-Based Search Engine using Conceptual Semantic Indexing, NAFIPS-FLINT 2002, June 27-29, New Orleans, LA, USA
12. M. Nikravesh and B. Azvin, Fuzzy Queries, Search, and Decision Support System, International Journal of Soft Computing-Special Issue in fuzzy logic and the Internet , 2002
13. M. Nikravesh, V. Loia, and B. Azvine, Fuzzy logic and the Internet (FLINT), Internet, World Wide Web, and Search Engines, to be appeared in International Journal of Soft Computing-Special Issue in fuzzy logic and the Internet , 2002
14. M. Nikravesh, Fuzzy Conceptual-Based Search Engine using Conceptual Semantic Indexing, NAFIPS-FLINT 2002, June 27-29, New Orleans, LA, USA
15. V. Loia, M. Nikravesh and Lotfi A. Zadeh, "Fuzzy Logic an the Internet", to be published in the Series Studies in Fuzziness and Soft Computing, Physica-Verlag, Springer (August 2003)

Short Bio: Dr. Masoud Nikravresh; BISC Associate Director

Dr. Masoud Nikravesh received his BS from Abadan Institute of Technology, MS and PhD in Chemical Engineering from the University of South Carolina (August 1993 and Dec.1994) and received full scholarship from University of California, Berkeley for his second Ph.D. in Material Sciences and Mineral Engineering Department for Fall 1994, when he decided to start his scientific carrier as Postdoc researcher in Spring 1995 at University of California-Berkeley and Lawrence Berkeley National Lab as joint appointment. Dr. Nikravesh is the BISC Associate Director, BTExact technology Senior Fellow and BISC Program Manager in the Computer Science Division, Department of Electrical Engineering and Computer Sciences at the University of California, Berkeley and Research Scientist in the Imaging and Informatics Group at NERSC (National Energy Research Scientific Computing Division, Lawrence Berkeley National Laboratory. In addition, he is serving as the Associate Director (Co-founder) of Zadeh Institute for Information Technology (Information Technology) and Chairs of BISC-Earth Sciences, BISC-Fuzzy Logic and Internet, and BISC-Recognition Technology Groups. His credentials have led to front-page news at Lawrence Berkeley National Laboratory News and headline news at the Electronics Engineering Times. Dr. Nikravesh is the LBNL-NERSC (National Energy Research Scientific Computing Division) representative to the DiMI Executive Committee. Dr. Nikravesh has over 20 years research and industrial experience and worked as consultant to over 15 major companies and funded several key projects in the area of soft computing, data mining and fusion, control, and earth sciences through US government and major oil companies. He published and presented over 100 articles and published several books on diverse topics and served as technical editor and several national and international technical committees and technical chairs including advisory board or technical review committee members for both government agencies and non-government agencies throughout the USA and abroad.. He served as member of IEEE, SPE, AICHE, SEG, AGU, and ACS.

BISC Program of the EECS Department-Computer Sciences Division-University of California-Berkeley, is the world-leading center for basic and applied research in soft computing. The principal constituents of soft computing (SC) are fuzzy logic (FL), neural network theory (NN) and probabilistic reasoning (PR), with the latter subsuming belief networks, evolutionary computing including DNA computing, chaos theory and parts of learning theory. Some of the most striking achievements of BISC Program are: fuzzy reasoning (set and logic), new soft computing algorithms making intelligent, semi-unsupervised use of large quantities of complex data, uncertainty analysis, perception-based decision analysis and decision support systems for risk analysis and management, computing with words, computational theory of perception (CTP), and precisiated natural language (PNL).

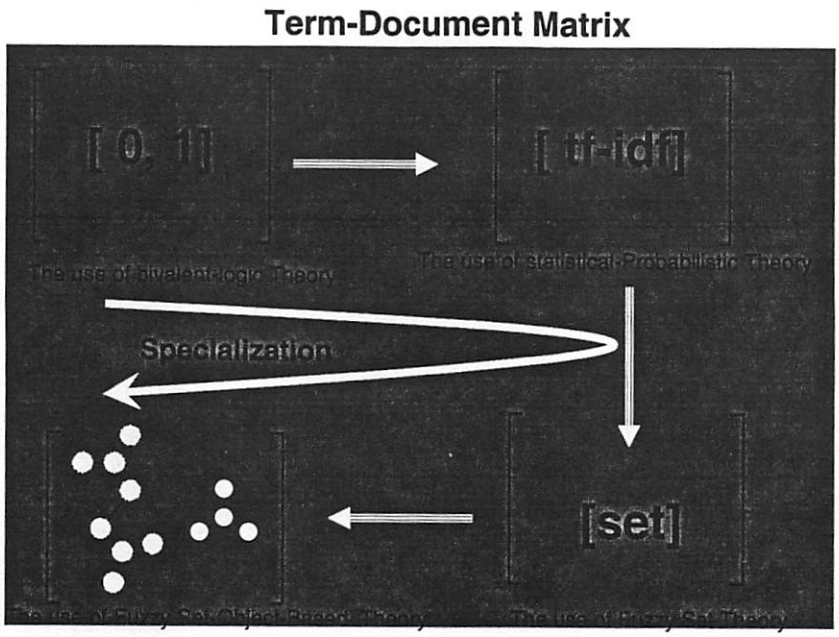


Figure 1. Evolution of Term-Document Matrix representation

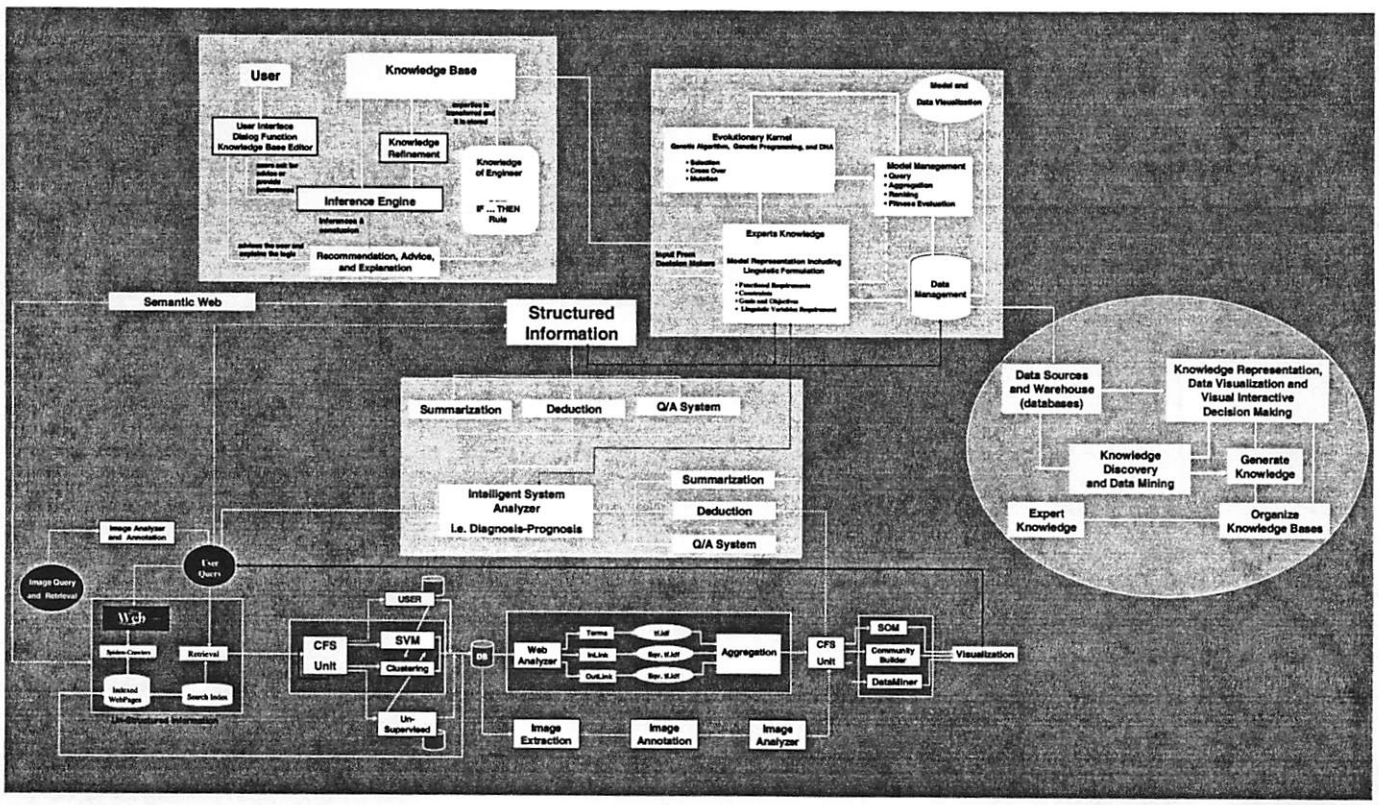


Figure 2. Concept-Based Intelligent Decision Analysis