Smart Advisor and Search Optimizer: Web-based Applications of Fuzzy Rules, Intelligence Systems and Hierarchical Clustering for Relational Decisions

Atul Parvatiyar[1], Sushil K. Prasad[2], Raj Sunderraman[2] and Yanqing Zhang[2]

Georgia State University
Atlanta, GA 30303

Abstract *- Smart Advisor and Search Optimizer are two intelligent Web agents helping users search Web information efficiently and make good decisions effectively. They are related to each other because Smart Advisor uses Search Optimizer as one component. Search Optimizer is a match maker between a user profile and search objectives, which can narrow down the search using a small number of screens of user input and well-defined categories. A new clustering algorithm for personal searches, group searches (peer groups), and all users searches is designed. In addition, this platform maintains member bookmarks with the ability for members to highlight some very frequently used bookmarks. The user has the ability to refine existing views or new searches to the extent possible and to bookmark the current search at any time. The search optimizer has been designed to allow flexibility and future scalability. Smart Advisor is a fuzzy expert agent with hierarchical fuzzy knowledge base. It has a natural interactive interview process for a user (i.e., a user can play a game with the Smart Advisor to gradually find out rational solutions), and produces advisory at different stages. For example, the college selection with relevant advice is a multi-feature-based complex process of assessment and decision for different domains (major, selectivity, college cluster, and individual college, etc.), hierarchical fuzzy assessment trees are designed based on hierarchical relationships among features so as to make a user do detailed assessment on a small group of relevant features on different level. A fuzzy assessment tree consists of many nodes including leaf nodes representing interview features provided by a user and high-level nodes representing abstract features generated by interview features. Such a bottom-up propagation calculation will eventually reach the root node of the fuzzy assessment tree, and generate a final possibility of college admission. The hierarchical advisory expert system is that hybrid experience of many experts is used to make a reliable decision, i.e., a student can virtually talk to many experienced experts in parallel to get robust assessment and stable advice. Importantly, it is a general framework that can be used in other applications such as e-Commerce, expert assessment and self-evaluation systems.*

1. Introduction

Smart Advisor (SA) is an expert system employing fuzzy logic which helps high school students select a few colleges to apply. It has an interactive interview process with animated help, and produces advisory at different stages. The output short list of colleges to apply contains five kinds of colleges (highly selective down to non-selective) with possibilities of acceptance in each college according to rules provided. Output also contains a plan of study. The system is also able to find possibility of acceptance in a user chosen college. All data collected is stored, to minimize questions when a student is a repeat user.

The search optimizer (SO) is a "match maker" between student profile and colleges. This platform employs profile and other information gathered from a student to search a database of colleges and output links to the home pages of the colleges found. SO attempts to narrow down the search using not too many screens of user input, but provide access to the current set of colleges found at each stage and the ability to further refine the search using additional screens of student input to the extent possible. Past searches of a student may be employed in the search process.

A key problem with today's search engines is that they are keyword based and their data is open-ended which causes a single search to generate thousands of hits, wasting users time and, worse, frustrating their search effort. For a well-defined domain, data can be categorized precisely, albeit with some effort, and a category/feature-based search can be performed resulting in only few selected hits for each search. Category-based search also allows users to click his/her way to the target information, as opposed to fumbling with correct descriptions/keywords to be employed. Additionally, individual user's usage pattern can be intelligently tracked, thus enabling users to quickly arrive at their target(s) with only few clicks by boot-strapping off a related past search. Similar tracking is possible of groups of related users for added functionalities. Such category-based search engines provide efficiency for the users on one hand and allow the database portal owner to effectively organize their backend information on the other hand.

[1] Dept. of Marketing, Robinson College of Business
[2] Dept. of Computer Science, College of Arts and Sciences

The Search Optimizer is a sophisticated category-based search engine with smart web-based GUI that allows a community of users to efficiently search and find key information from a large collection of data. SO intelligently manages a system of bookmarks and frequently requested searches at the individual, peer group and community levels. SO is a dynamic system that continually adapts to user requests and can be easily tuned to different levels of adaptabilities. It is designed for scalability on the web, with performance-tuned database access patterns and patent-pending algorithms. A very important feature of SO and SA is its flexibility to be customized for any collection of data on any domain. The core of the Search Optimizer system is domain independent and much of the GUI system is also domain independent. SO was designed with customization in mind.

Under the hood, Smart Advisor and Search Optimizer are two intelligent Web agents helping users search Web information efficiently and make good decisions effectively. They are related to each other because Smart Advisor uses Search Optimizer as one component for finding colleges. Smart Advisor is a fuzzy expert agent with hierarchical fuzzy knowledge base using fuzzy logic [1-5]. It has a natural interactive interview process for a user (i.e., a user can play a game with the Smart Advisor to gradually find out rational solutions), and produces advisory at different stages. For example, the college selection with relevant advice is a multi-feature-based complex process of assessment and decision for different domains (major, selectivity, college cluster, and individual college, etc.), hierarchical fuzzy assessment trees are designed based on hierarchical relationships among features so as to make a user do detailed assessment on a small group of relevant features on different level. A fuzzy assessment tree consists of many nodes including leaf nodes representing interview features provided by a user and high-level nodes representing abstract features generated by interview features. Such a bottom-up propagation calculation will eventually reach the root node of the fuzzy assessment tree, and generate a final possibility of college admission (output of the root). The major advantage of the hierarchical advisory expert system is that hybrid experience of many experts is used to make a reliable decision, i.e., a student can virtually talk to many experienced experts in parallel to get robust assessment and stable advice. Smart Advisor can be used in other applications because it is a general framework. Search Optimizer is a match maker between a user profile and search objectives, which can narrow down the search using a small number of screens of user input and well-defined categories. A new clustering algorithm for personal searches, group searches (peer groups), and all users searches is designed. In addition, this platform maintains member bookmarks with the ability for members to highlight some very frequently used bookmarks. The user has the ability to refine existing views or new searches to the extent possible and to bookmark the current search at any time. Both the search optimizer and the smart advisor has been designed to allow flexibility and future scalability.

This paper has been organized as follows. Section 2 provides technical details of Smart Advisor, going into aspects of user perspective, architecture, tree-based hierarchical clustering, fuzzy rules, etc. Section 3 provides architecture of Search Advisor and briefly provides details of hot views and user profile-based filtering. Section 4 provides conclusion emphasizing the wide applicability of the two platforms due to their framework design and scalability. Section 6 contains some basic fuzzy logic background in the Appendix.

2. Smart Advisor

2.1 User Level Basic Functionalities

Here, we describe how a typical first-time user and a typical repeat user will use the system.

2.1.1 Typical Screen Sequence for a First-Time User

A typical first time user comes to the welcome screen (Welcome, basic Intro, Demo, Interview), and goes to a *guided interview screen (also, the play screen)* (some may choose quick interview mode – form filling). Once in guided interview screen, the user is queried hierarchically, through repeated updated instances of this screen. The purpose of these screens is to collect profile features as well as, based on the context, to give specific advice and to show user's possibilities into five categories of colleges (using initial assessment, detailed assessment, and major-specific assessments). At the same time, user gets familiar with the system and its capabilities, question hierarchy and phases, and the navigation maps. The effect of each change in the input or output feature in any phase at any level of hierarchy is quickly calculated and the changes in the possibilities of admission is promptly displayed in the next screen.

All the while, an animated character, "Jill/Jack" interacts with the user explaining the screen components, input questions, output results, and parceling out context and input/output value based advices. "Jill/Jack" can also be made to sit on the sidelines, and then the corresponding explanations are available to the user as "blurbs" hidden within each components, and advices written out along side with the components on the screen.

At any time in the interview process, the user can then ask the system to find colleges to apply, which takes the user to *selected college screen* which shows a few selected optimum colleges to apply, user's possibilities in each of them, and college specific advices/plan of actions. In addition, this screen also shows additional advices/plan of action based on different college categories. User can subsequently choose some of his/her own colleges to be evaluated against user's profile. The user can go back and

forth between guided interview and selected college screens to change profile and see its effect on college selections and possibilities of acceptances, playing "what if" games.

2.1.2 A typical Usage Pattern for a Repeat User

A repeat user is prompted with calendar and high-school status (freshman, junior, senior, etc.) based advices/plan of actions, as well as specific questions based on the plan of actions given last time. This allows a repeat user to quickly update his/her profile. The user typically then goes to guided interview screen to further update his/her profile and continue with the interview session, or the last "what if" game. (Or the user can edit his/her favorite colleges and see his/her improved chances in selected college screen). Once in guided interview/play screen, a repeat user typically will alter values of one or more input or output features and play "what if" games to determine how his/her chances of getting into different categories of colleges would improve.

To aid in quick navigation through different phases and the hierarchy of these phases, a click-able navigation sub-tree in the play screen is always visible (available to first time user as well). The effect of each change in the input or output feature in any phase at any level of hierarchy is quickly calculated and the changes in the possibilities of admission is promptly displayed in the next screen.

As for a new user, a repeat user would typically go back and forth between play and selected college screens to see how changes to input/out features alter college selections and possibilities, and what specific future actions he/she must take to get into a specific college or a college category.

Users would be able to use SA as a trusted, involved and smart counselor, who remembers their profile into a folder and their past history, advises them through their high school years through all their phases of school, gives them increasingly better and specific nuggets of advice as they make progress through their schools, and guides them toward a selected set of colleges to apply which optimizes their potential. SA would also be a smart tool to play self-guided "what if" games to explore what specific action one should be taking to have maximum impact on the selection of colleges one can get in and on his/her chances of admission.

2.2 Top Level Technologies

The smart advisor platform is a Web-based expert system with multiple hierarchical domain-specific knowledge bases built by many experienced experts. The smart advisor has very unique novel capabilities supported by advanced techniques such as expert system, fuzzy logic, databases, statistics, decision tree, and client-server networking. Appendix in Section 6 contains background on fuzzy logic. Here, for clarity, two top-level key techniques are described.

2.2.1 The Hierarchical Advisory Expert System with Multiple Fuzzy Assessment Trees

Since college selection with relevant advice is a multi-feature-based complex process of assessment and decision for different domains (major, selectivity, college cluster, and individual college, etc.), hierarchical fuzzy assessment trees are designed based on hierarchical relationships among features so as to make a user do detailed assessment on a small group of relevant features on different level. A fuzzy assessment tree consists of many nodes including leaf nodes representing interview features provided by a user and high-level nodes representing abstract features generated by interview features. Each high-level node (parent node) has a fuzzy rule table that is used to calculate possibility (output of the parent node) based on input features from low-level child nodes. Such a bottom-up propagation calculation will eventually reach the root node of the fuzzy assessment tree, and generate a final possibility of college admission (output of the root). There are several types of fuzzy assessment trees:

(1) five initial assessment trees,
(2) five detailed assessment trees,
(3) five major assessment trees,
(4) multiple college cluster assessment trees, and, possibly in the future, multiple individual college assessment trees.

The major advantage of the hierarchical advisory expert system is that hybrid experience of many experts is used to make a reliable decision, i.e., a student can virtually talk to many experienced experts in parallel to get robust assessment and stable advice.

The key results of the smart advisor are

(1) A short list of colleges with acceptable high possibilities of college admission, and
(2) Relevant advisory comments and plan of action.
(3) Finally, a user can not only understand himself (or herself) in possibilities and capabilities of different level college admission, but also know what to do to enhance chances to get into desired colleges in the near future.

2.2.2 Self-Play-Instant-Assessment Advisory Game

A user can freely play with the friendly real-time smart advisor at any time (24 hours a day, but a student has to make an appointment with a human counselor for an interview). The more a user plays, the more reliable assessment, accurate advice and stable colleges. A user can enjoy the smart game either guided by the smart advisor (convenient for first time users) or controlled by himself (or

herself) (flexible for repeat users). In general, a user can play with every node of a particular fuzzy assessment tree to carefully assess his (or her) capabilities and possibilities by changing input child features reasonably, and at the mean time the user can know how to improve himself (or herself) for the relevant child features to achieve a higher level of the output of the parent node. Obviously, such a powerful self-play-instant-assessment merged by the smart advisor's knowledge and the user's self -information is an ideal real-time advisory environment for users (especially young high-school students). High-school students can play with the smart advisor throughout their high-school years.

Thus, the smart advisor is a real-time intelligent agent using latest intelligent techniques, advanced databases and efficient networking functions. The smart advisor (client-server version on the Internet or portable version on a user's local computer) is a short-cut bridge between high school graduation and college admission.

2.3 Implementation Level Architecture

2.3.1 Four Phases of SA Game

The student interview and evaluation process is a four-phase process. In Phase I, a few features (GPA, SAT and student's self-evaluation scores for strength of his/her curriculum and extra-curricular activities) are employed to find student's *initial assessment* in five categories of colleges (most selective, highly selective, selective, less selective and non-selective). These initial assessments result in five PC1 values for the five *college clusters*. In Phase II, which results in a *detailed assessment* of student's chances in the five college clusters, a student is led through an elaborate and engaging interview process to collect all his/her *basic features* (such as SAT-I score, number of extra-curricular activities engaged in, etc). At each juncture through this interview process, calculations are made by the system and the student receives an update on his or her chances (*scores*), using a color-coded star display. The corresponding five possibilities in the five college clusters are collectively referred to as PC2's. In Phase III, student's chances are calculated in his/her desired major (*disciplinePreference*), resulting in five PC3's. If the disciplinePreference for the member is missing/undefined Phase III is skipped. In the last phase, Phase IV, student's *preference profile* is collected and Search Optimizer platform is employed to filter out 100 or fewer colleges. These colleges are then individually matched against student's features to obtain PC4's (only one value per college). The four PCs are then combined to obtain student's chances in each of the filtered colleges (PC). Finally, a short list of 10 to 20 colleges is produces that the student is recommended to apply in. So the generated results are very convenient for a user. If the results are not satisfactory, the user may continue to play with SA.

2.3.2 Rules Trees

Phase I and II employ hierarchical fuzzy rule structures (a partial order on top of the basic features, referred to as trees for convenience, example shown in Figure 1), namely, Initial Assessment Tree (*IA*) and Detailed Assessment Tree (*DA*). The reason these are partial orders is because a node could be input to multiple parent nodes, therefore the resulting graph is a Directed Acyclic Graph. Likewise, for Phase III, each major has a Major Assessment Tree (*MAi*), and, for Phase IV, each college has a College Assessment Tree (*CAj*). Thus, the logical structure of all the four phases combined together is a giant partial order (referred to as the *saTree*) such that SA is the root of four kinds of sub-trees in the following order: IA, DA, MAi, and CAj (i runs through all major trees and j runs through all college trees). The saTree has unique node numbers for all its nodes, including the *leaves*, which are the basic features of the students, and the *internal nodes*, which are *derived features* of students based on the leaf features of the corresponding sub-tree.

IA and DA trees are unique, but there are several MAi's and CAj's. It is envisioned that multiple colleges with admission profile can have the same CAj. Refer to Appendix page SA-19 for a pictorial explanation of the trees.

2.3.3 Rules Table

Each internal node of saTree, say node X with children Y and Z, has a rules table, with input columns corresponding to Y and Z, and five output columns corresponding to the strengths of the derived feature X for the five college clusters. If child Y has five range of values and Z has six range of values, then the rules table has 30 rows, one for each combination of values of Y and Z. Logically, therefore, the rules table is five different tables for five college clusters, one corresponding to each of its output column (Exceptions are tables for CAj's which have just one output column).

Each value range (*value gradation*) for an input column, say Y, has a typical value (*center value, C*), a low value (*L*) and a right, upper value (*R*). Neighboring ranges may not be disjoint, so R for a range may be greater than L for the following range. Overlapping ranges lie at the core of fuzziness. In the above example, therefore, a specific pair of values of X and Y, may correspond to up to four rows in the rules table, and therefore "fire" the corresponding rules. Thus, the value calculated for node X is robust, employing fuzzy logic.

2.3.4 Miscellaneous Aspects

Explanation, Advisory and Jill's Phrases: Each node X with children Y and Z not only has rules table but also carries explanations for features X, Y, and Z and for value gradations of X, Y, and Z. Likewise, there are advisory phrases corresponding to future plan of actions/study (*POS*),

for each value gradation of X, Y, and Z. Advisory phrases are displayed once a new value for a feature has been employed to calculate new scores.

Jill has a sound and graphics URL corresponding to each value gradations of all features X, Y, and Z corresponding to explanation phrases in one screen and to advisories in the next screen.

Base and Session Vectors: For each member, SA keeps a folder of student's current basic and derived features into the *base vector*. During a "what if" game, student may change his/her base feature values. The current "what if" play values of student features are kept track of separately in the *session vector*.

Default Session Vector: For all subtrees of SA, namely IA, DA, MAi (for all discipline trees i), CAj (for all college trees j), a default session vector is initialized manually based on typical values ("Average Joe" values).

Flexible Framework Design: The Smart Advisor system has been designed as a general framework, allowing managers to alter the feature trees, the rules, explanations and advisories at each node of the feature trees, and various other tuning parameters at any time in the database without any change in the source code. This will allow managers to alter the interview order and context as well as to refine the rules with experience.

3. Search Optimizer

The search optimizer platform has very unique capabilities as compared to its competitors.

This platform contains a college database, as well as, simpler databases for financial aid sources, careers and majors, and standardized tests. It contains search algorithms to find colleges based on certain categories of search criteria, producing links to the college sites. One can reorder these search categories, to effect a change in the order these categories are presented to a user. The design is unaffected even when one chooses to add or delete search categories (however such changes entail database updates by the DBA).

Hot Views and Online Clustering of Past Searches: Several hot views (past searches) are captured for various groups of users which may allow a user to quickly narrow down a search for colleges, by boot-strapping off a related past search. Users can also search for career positions, majors, financial aid sources, and browse standardized test pages. Hot views are maintained by employing a new clustering algorithm for personal searches, group searches (peer groups), and all searches. In addition, this platform maintains member bookmarks with the ability for members to highlight some very frequently used bookmarks. Member-profile based views are also maintained. Each view contains the selected categories along with the number of matching colleges and a sample college.

The member has the ability to refine existing views or new searches to the extent possible and to bookmark the current search at any time. Searches can also be further refined by employing a member's profile information, thereby reducing the number of user inputs.

The search optimizer has been designed to allow flexibility and future scalability. The design employs minimum number of database accesses for efficiency of search. Hidden fields are effectively used to minimize database accesses. A compact layout has been proposed for the user interface design. A member can change his or her peer group at any time to see the hot views pertaining to that peer group.

4. Conclusions

Both SA and SO are flexible and scalable frameworks that can be used in other applications such as e-Business applications. In the future, other computational Web intelligence techniques [6,7] can be applied to SA and SO to make them have online learning ability.

References

[1]. http://www-bisc.cs.berkeley.edu/.

[2]. http://www.abo.fi/~rfuller/ifsa.html.

[3]. http://morden.csee.usf.edu/Nafipsf/.

[4] Yen J. and Langari R. (1999). Fuzzy Logic: Intelligence, Control, and Information. Prentice Hall.

[5] Zadeh L.A. (1965). Fuzzy Sets. Information and Control, 8, 338-353.

[6] Zhang Y.-Q., and Lin T.Y. (2002). Computational Web Intelligence (CWI): Synergy of Computational Intelligence and Web Technology. Proc. of FUZZ-IEEE2002 of World Congress of Computational Intelligence 2002: Special Session on Computational Web Intelligence, 1104-1107.

[7] Zhang Y.-Q. and Kandel A. (1998). Compensatory Genetic Fuzzy Neural Networks and Their Applications. Series in Machine Perception Artificial Intelligence, Volume 30, World Scientific.

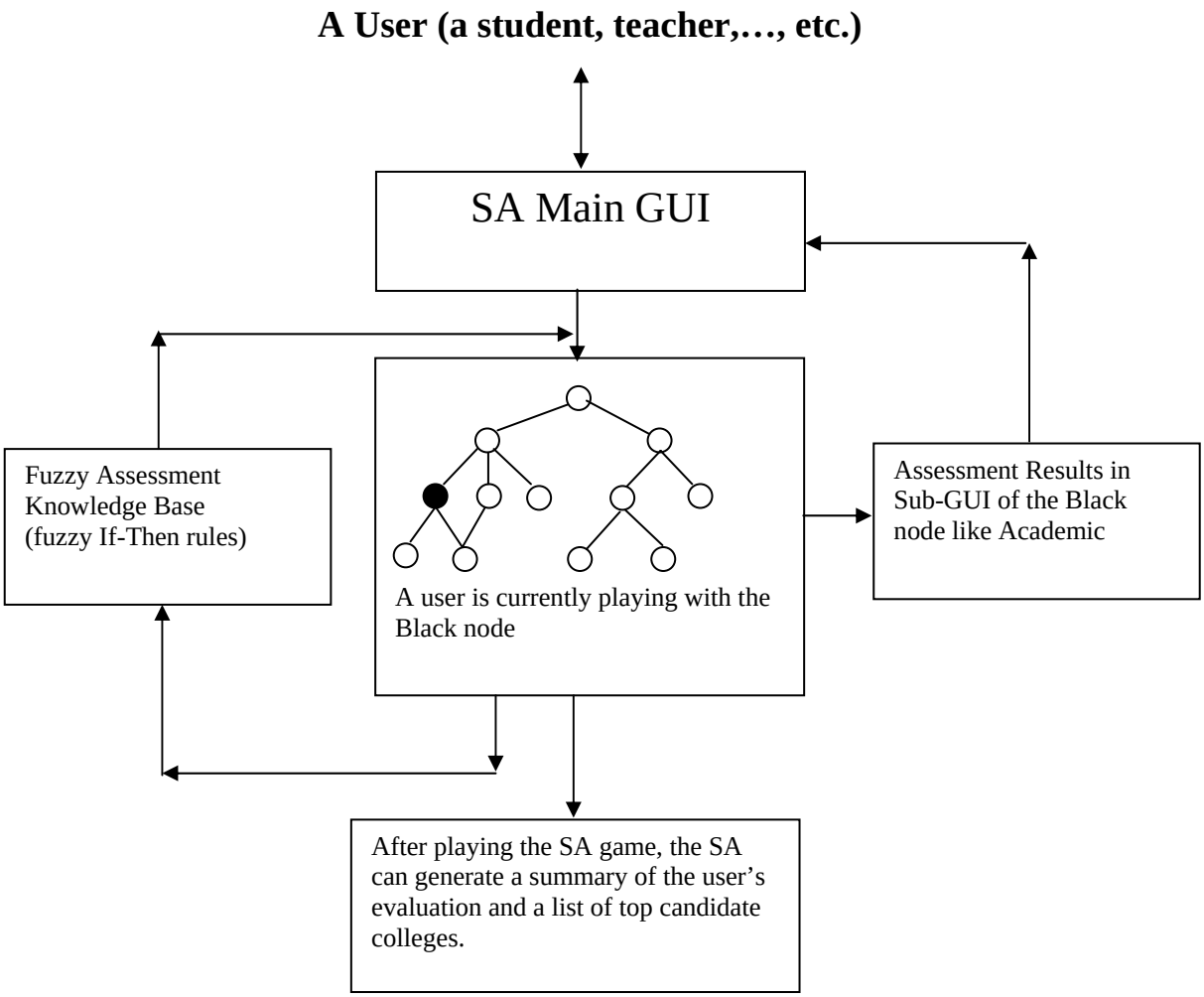

Figure 1 SA-Game using Fuzzy Rules Tree and Hierarchical Clustering

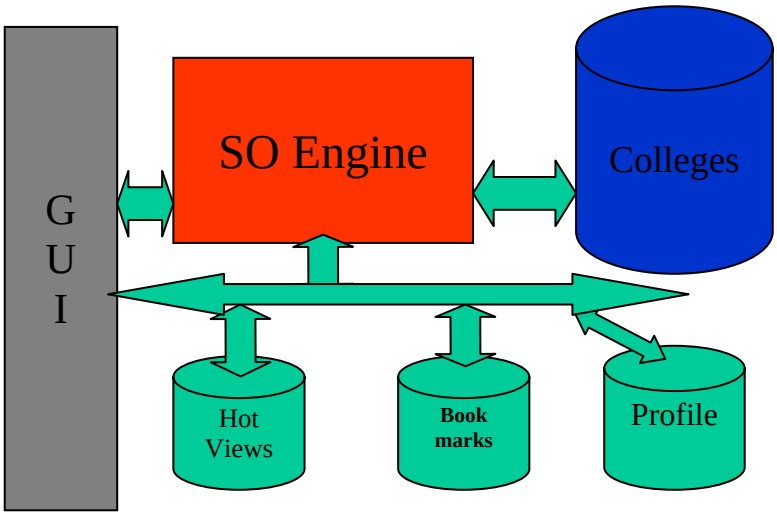

Figure 2 Search Optimizer Architecture