

6. Applications of Soft CBR at General Electric

Abstract

General Electric has used soft computing techniques in a variety of fielded case-based reasoning systems. In doing so we were able to leverage the tolerance for imprecision and uncertainty which is intrinsic to soft computing techniques. The payoff of this conjunctive use of soft computing and case-based reasoning techniques is a more accurate and robust solution than a solution derived from the use of any single technique alone. The fielded systems for medical equipment diagnostics and residential property valuation used fuzzy membership functions for greater selection accuracy through their noise tolerance and for the ability to determine a confidence in a result. A plastic color matching system used fuzzy logic to combine several selection criteria which allowed its users to detect potential problems during the case selection phase while previously these problems were not detected until the color match was complete. Finally, adaptive fuzzy clusters were used for fault classification of aircraft engine sensor data. The advantages of this system were its ability to deal with extremely noisy sensors and to adjust to unpredictable slow drift.

6.1 Introduction

General Electric (GE) has fielded a set of case-based reasoning (CBR) [1][2][9][23] applications that use techniques from soft computing (SC)[22][28]. The SC techniques allow reasoning with uncertainty and imprecision that is beneficial to the CBR process. This chapter will show those benefits for each of the applications from the field of medical equipment diagnosis, plastics color matching, residential property valuation, and aircraft engine monitoring. We will show improving of case selection accuracy, handling of multiple selection criteria, producing a confidence value for a specific result, and dealing with non-stationary cases. In particular, the use of membership functions allows a greater accuracy and smoothness in the selection phase. Handling multiple selection criteria enables the systems to detect potential problems during the selection phase. The ability to determine a confidence in a result makes the systems more useful to the end users.

Finally, cases of non-stationary systems are tracked and updated automatically. Each application has been under development for multiple years, used in the field for over a year, and most are still in use. Figure 6.1 shows the duration of time when each application was developed with a solid line and the time the application has been in use with a dotted line. The following sections will briefly describe an application then go into detail on how soft computing techniques were integrated with the traditional CBR process.

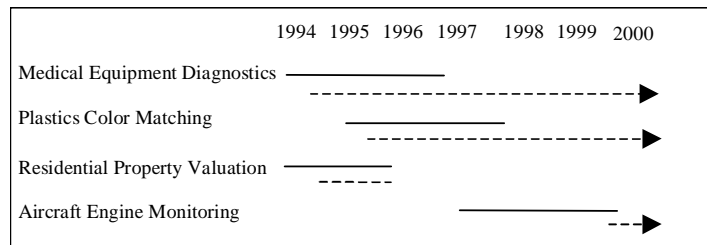


Figure 6. 1: Tool Development and Usage

6.2 Medical Equipment Diagnosis

This section examines the use of SC techniques to boost the performance and quantify the confidence of answers generated by a case-based diagnostic system built for GE Medical Systems (GEMS).

6.2.1 Introduction

ELSI (Error Log Similarity Index) [1] [12] [13] [14] was first conceived in the early 1990's to solve a diagnostic challenge with Computer Tomography (CT) scanners. GEMS had built an infrastructure to allow service to dial in to imaging equipment and diagnose failures while still on the phone with the customer. Once the service engineer had taken a look at the equipment remotely, the customers' issues could often be resolved over the phone. If parts and repairs were needed, the service engineer could send out a field engineer with the correct parts already in hand.

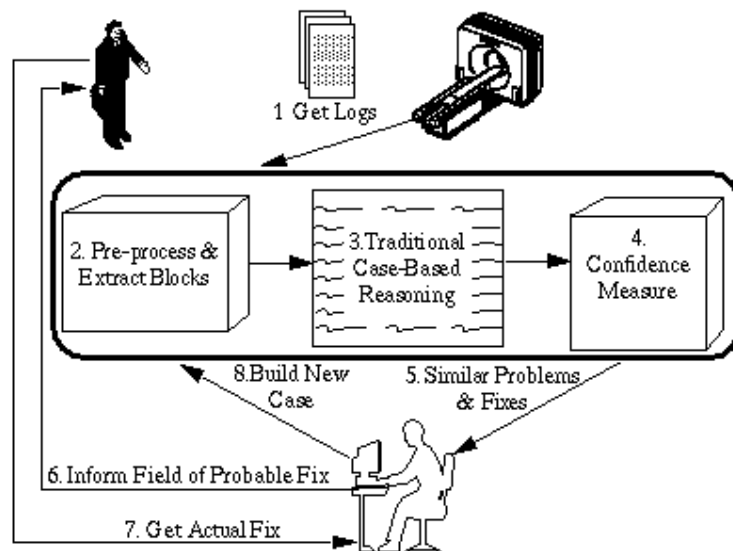


Figure 6. 2: High-Level ELSI Architecture

Some of the most promising diagnostic information on the machines was in error logs generated by the computer processes controlling the machines. The error logs, however, had been primarily designed as debugging tools for developers, not as diagnostic tools for service technicians. The logs were not well formatted, and contained a mix of normal status messages and error messages. The messages were documented one-by-one, yet equipment failures were found to generate more

complex cascades of errors.

Recognizing these combinations of error messages was purely an art form practiced by the more experienced service engineers. An automated system was needed to find these patterns and catalog them, thus creating a common experience base across service engineers. ELSI set out to meet this challenge.

ELSI's primarily case-based approach also provided platform independence (the system could be used on different models and types of equipment), little need for knowledge collection from the equipment designers, and the ability to keep up with design changes as new cases arrived.

Figure 6.2 describes the ELSI process. After error logs have been obtained in step 1, they are pre-processed in step 2. Here, diagnostically extraneous information such as dates and scan numbers is removed. In rare cases useful numeric values are rounded to make exact matches with other logs more likely. In step 3 the logs are compared to other logs with the same fix, and the largest possible blocks of completely matching lines are extracted. When a new case comes in to be diagnosed, its logs are searched for all known blocks, and its similarity to known cases is

$$\text{similarity}(\text{new}, N) = \sqrt{\frac{\sum \text{weights}(\text{matching})}{\sum \text{weights}(\text{new})} \times \frac{\sum \text{weights}(\text{matching})}{\sum \text{weights}(N)}}$$

determined through the formula:

This translates to simply reporting which past cases share the most symptoms (after weighting) with the new case. The cases with the highest similarity were shown to the service engineers, who could look at the matching blocks and the notes from matching cases to make the final determination of how to fix the machine.

In step 4, the blocks of error log are then given weights inversely proportional to the number of different kinds of problems have generated the block. Blocks that appear only for one fix are given the highest weight. Those that appear for many different fixes are thrown out altogether. Steps 5-8 are concerned with updating and reporting tasks.

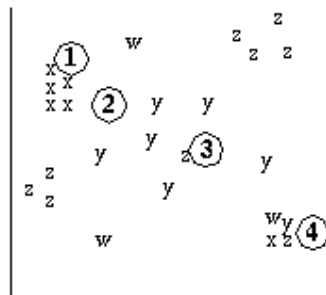
6.2.2 Noise Handling

Although this system was successful when first fielded, it did not effectively handle the high level of noise in the case base. Figure 6.3 shows a simplified view of the types of noise encountered in the case bases. Each letter represents a different fix. The numbers 1 to 4 represent new cases.

Notice these types of noise:

1. fixes like 'x' are tightly clustered, while others like 'y' are not.
2. Fixes like 'z' have multiple clusters; They have different failure modes.
3. Some groups of failures like 'w' 'x' 'y' 'z' generate nearly identical logs.

Given this noise, cases like (1) match fine. The other cases do not. Case (2) is equally close to 'x' and 'y' cases, but the 'y' cluster is much looser. Therefore the case is probably a better match to 'y'. Case (3) is most similar to a 'z', but that 'z' is stranded among 'y' cases. Again 'y' seems like the answer we should be most



confident in.

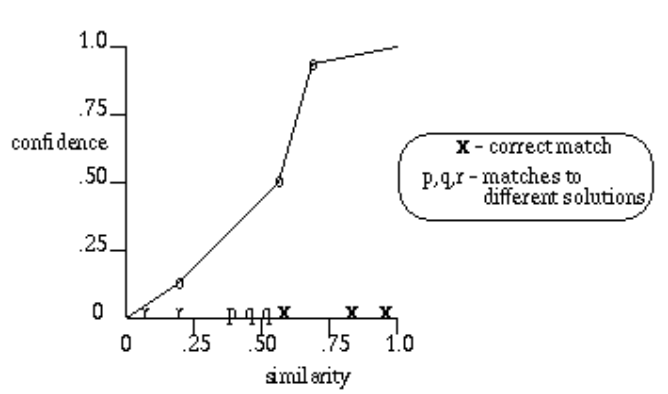


Figure 6. 4 Confidence Membership Function

It was case (4) that raised the red flags first. In some examples, ELSI would report perfect matches to many different cases, all with different fixes. This is because they all happened to have the same routine status messages or minimally helpful error messages and no really useful blocks. The total symptoms matched perfectly however.

Initial attempts at setting a high threshold for weights of blocks (throwing out all but the best ones) had a very bad impact on the system. Trying to find a good "K" for K-nearest neighbors was frustrated by the widely varying frequencies at which different problems occurred. Early efforts to represent case base noise in a formal

statistical way indicated that the calculations needed would be too complex to implement in a practical manner.

As a result, the "confidence measure" was devised as a way to use soft computing methods to automatically determine how confident the system should be in each match. It is method of building membership functions which are modeled after the procedure the ELSI system developers had used to diagnose problems with the system.

6.2.2.1 Representing Confidence

Before we can proceed, we will introduce a graphical notation for confidence membership functions. An example is shown in Figure 6.4.

This graph represents similarity versus confidence for one case stored in the case base. Note the symbols along the x axis. These represent the case's neighbors. An 'x' represents the similarity of a case with the same fix as our target case. Other symbols 'p', 'q', 'r', etc. represent cases with other fixes (i.e., wrong answers).

6.2.2.2 Confidence Based on Fraction of Better Matches

An initial attempt at a confidence measure was based on counting all the matches higher than a certain point, and determining what fraction of them are "right". Figure 6.5 shows two examples of such a confidence measure.

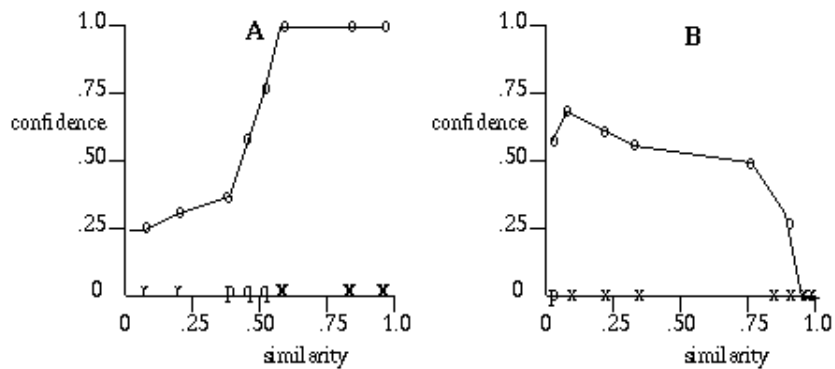


Figure 6. 5 Using Fraction of Closer Cases with Correct Fix

In these examples, confidence was calculated at each similarity where a match occurred. The confidence was calculated as the fraction of cases with similarity greater than or equal to the current level which matched the same solution. The points were then connected to form a membership function.

Example A turned out as one might expect. The confidence rises as the match becomes stronger. Example B, however, has obvious problems. The confidence drops as similarity rises. Clearly, this approach needs to be revised.

6.2.2.3 Number of Solutions with Better Matches

An alternative confidence measure is based on counting all the matches higher than a certain point, and determining how many different solutions are represented. Confidence would be $1 / \langle \text{number_of_solutions} \rangle$ as long as one of the solutions is correct. Otherwise, confidence remains level. Again, confidence is calculated at each similarity where a match exists, and the points are connected into a line. Figure 6.6 shows two examples of such a confidence measure.

With this technique, example A changed very little. Example B has been transformed into a straight-forward and sensible function. Namely, since there are two solutions (x and r) which occur above .05 similarity, the confidence is around 50% for this entire range.

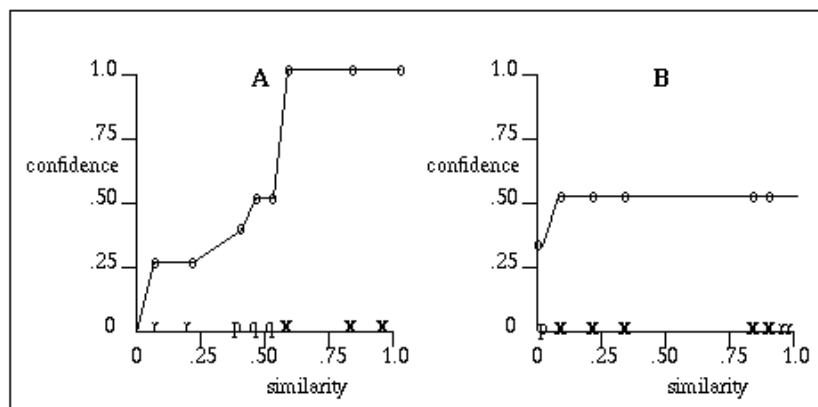


Figure 6. 6 Using Number of Fixes in Closer Cases

Figure 6.7 shows one area in which this approach remains to be improved.

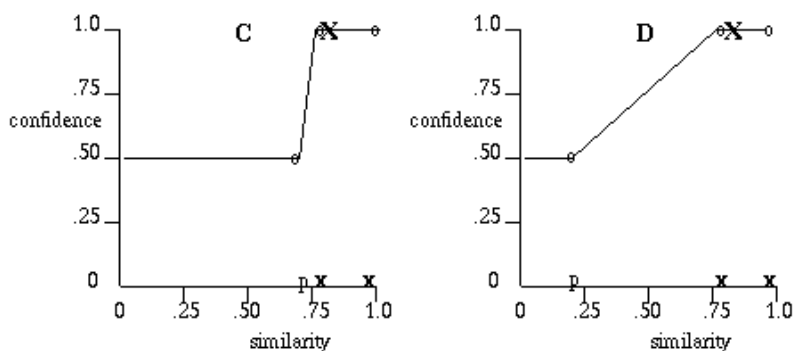


Figure 6. 7 Problem Remaining in this Approach

A .75 match to each of the cases C and D yields a confidence of 1.0. However, case C has an incorrect match at .74, while case D has no incorrect matches anywhere near .75. Clearly, we should be less confident in our .75 match to case C than we are in a .75 match to case D.

Another weakness in this approach is the propensity toward large plateaus at 1.0, 1/2, 1/3, 1/4, etc. This produces more confidence value ties than is necessary.

6.2.2.4 Final Confidence Measure

The final solution we implemented was based on the same formula as above, but we calculate a confidence at fewer points. The formula, again, is $1 / \langle \text{number of solutions with better matches} \rangle$. Moving from best match to worst match (right to left) we start at the first correct match, and continue by calculating a value only at the best match for each solution. Fill in the rest of the function with a distance-weighted average between points (connecting the dots), connecting to the origin at the left, and maintaining a constant value (horizontal line) on the right. Figure 6.8 shows the resulting functions.

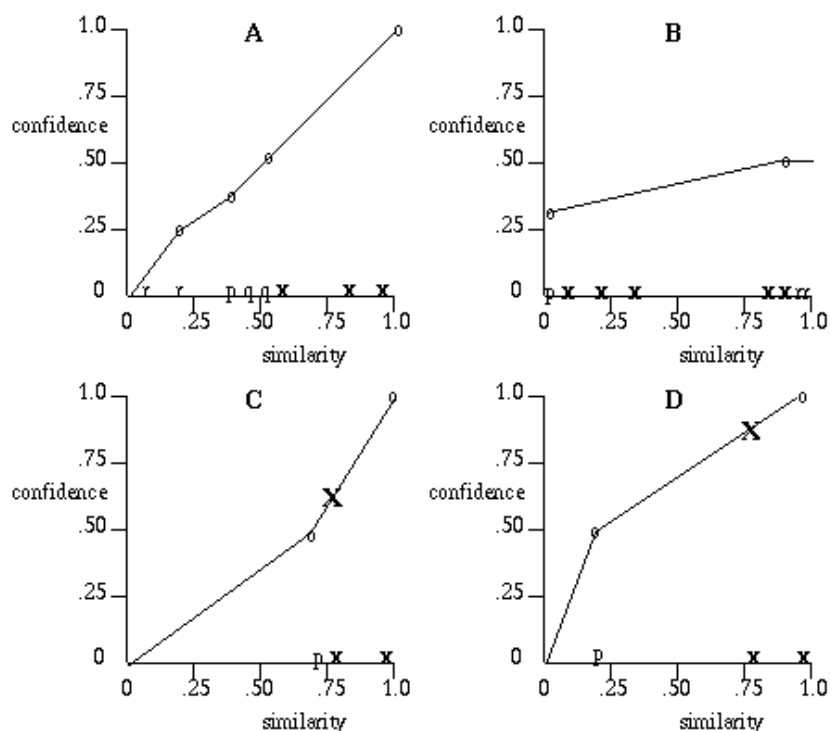


Figure 6. 8 Final Confidence Functions

This final algorithm gives satisfactory membership functions in all the cases

examined so far.

In case A, confidence starts at 0 and passes .5 when it passes the best matching incorrect answer. From that point it rises steadily to confidence of 1 at similarity of 1.

For cases like B where the best matches are to cases with a different solution, confidence never rises above .5. This accurately reflects the fact that no case can ever be so similar to case B that there aren't at least two likely solutions.

This approach smoothes out the plateaus in all four cases, and improves the distinction between the .75 matches to case C and D properly. That is, the confidence is lower at .75 in case C since there is an incorrect match at .74, while it is higher in case D.

6.2.3 Results

Using soft-computing techniques to build a "confidence" member function for ELSI increased the accuracy and lowered the maintenance requirements of the system.

Applying this confidence measure to two noisy case bases on top of three different traditional similarity measures yielded fairly standard improvements of 13% in first-guess accuracy. That means that using leave-one-out testing, 13% more cases had best confidence matches with the same fix than had best similarity matches with the same fix.

Equally notable is the fact that while similarities above .9 were only correct 59% of the time (averaged across all techniques), confidences above .9 were correct 92% of the time. All similarity measures tested showed marked improvements in this area. This increased end-user's satisfaction with the system.

Although the designers of diagnostic systems love to dwell on the accuracy of the system, the real measure of the quality of a system may be in the users' ability to maintain it. This is where the confidence function played its most important role.

The confidence function is a formal way of handling noise in the case base. This translates directly into tolerating the addition of cases which do not have enough information in the logs to narrow down the fix to one part, or cases where the fix is entered incorrectly.

This tolerance of noise allowed for a much simpler case base maintenance procedure. Without it, cases would have to go through a very complicated verification procedure to determine not only that the fix was correct, but that there was enough information in the error log to diagnose it. With the confidence measure, all available cases could be added. Only periodically would a maintainer need to look through the case base for a build-up of cases which did not diagnose correctly, and remove them.

The confidence measure is played a key role in enabling ELSI to be handed off to users with limited computing skills, where they added hundreds of new cases and ran diagnostics on thousands of problems over several years without requiring major help from the system developers.

6.3 Property Valuation

A division of GE Mortgage purchases mortgage packages on the secondary market as investments. These packages can contain up to 1000 mortgages. Property valuations are needed to evaluate the current value of mortgage packages that may be purchased. However, it is not cost effective to have humans appraise these mortgage packages. To automate the valuation process we have developed the Property Financial Information Technology (PROFIT) system [5][8][19], which uses fuzzy logic to enhance case-based reasoning .

6.3.1 Introduction

Residential property valuation [3] is the process of determining a dollar estimate of the property value for given market conditions. Residential property is restricted to a single family residence designed or intended for owner-occupancy. The value of a property changes with market conditions, so any estimate of its value must be periodically updated to reflect those market changes. Any valuation must also be supported by current evidence of market conditions, e.g. recent real estate transactions.

The current manual process for valuing properties usually requires an on-site visit by a human appraiser, takes several days, and costs about \$500 per subject property. This process is too slow and expensive for batch applications such as those used by banks for updating their loan and insurance portfolios, verifying risk profiles of servicing rights, or evaluating default risks for securitized packages of mortgages. The appraisal process for these batch applications is currently estimated, to a lesser degree of accuracy, by sampling techniques. Verification of property value on individual transactions may also be required by secondary buyers. Thus, this work is motivated by a broad spectrum of application areas.

In most cases, the most credible method used by appraisers is the sales comparison approach. This method consists of finding comparables (i.e., recent sales that are comparable to the subject property using sales records); contrasting the subject property with the comparables; adjusting the comparables' sales price to reflect their differences from the subject property (using heuristics and personal experience); and reconciling the comparables adjusted sales prices to derive an estimate for the subject property (using any reasonable averaging method). This process assumes that the item's market value can be derived by the prices demanded by similar items in the same market.

The PROFIT CBR process also consists of selecting relevant cases, adapting them, and aggregating those adapted cases into a single estimate of the property value. This process is shown in Figure 6.9.

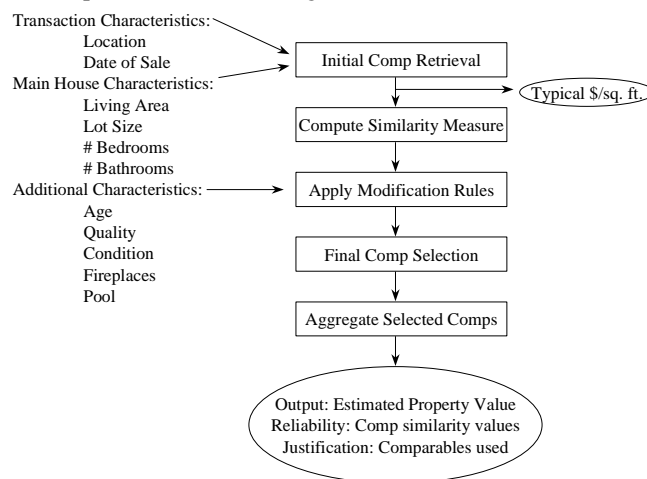


Figure 6.9 Property Valuation Process

Upon entering the subject property attributes, PROFIT retrieves potentially similar comparables from the case-base. This initial selection uses six attributes: address, date of sale, living area, lot area, number of bathrooms, and bedrooms. The comparables are rated and ranked on a similarity scale to identify the most similar ones to the subject property. This rating is obtained from a weighted aggregation of the decision making preferences, expressed as fuzzy membership distributions and relations. Each property's sales price is adjusted to better reflect the subject's value. These adjustments are performed by a rule set that uses additional property attributes, such as construction quality, conditions, pools, fireplaces, etc. The best 4 to 8 comparables are then selected. Finally, the adjusted sales price and similarity of the selected properties are combined to produce an estimate of the value of the subject, a reliability in that estimate, and a justification for the estimate.

6.3.2 Fuzzy Case Retrieval

The initial retrieval extracts a set of potential comparables using standard SQL queries for efficiency purpose. The selection is performed by comparing specific attributes of the subject with the corresponding attribute of each comparable. All the comparables in the retrieved set have values within the allowable deviations. **If the size of the retrieved set is too small, e.g., less than 10, the allowable deviations could be relaxed to increase its size at the expense of retrieval quality.**

This initial retrieval stage uses the following attributes and their corresponding maximum allowable deviations (written after each attribute):

- Date of sale (within 12 months)
- Distance (within 1 mile)
- living area (+ / - 25%)
- lot size (+ 100% / - 50%)
- Number of bedrooms (+/- 3)
- Number of bathrooms (+/- 3)

Figure 6.10 describes our preference criteria for the first four features. The trapezoidal membership distributions representing these criteria have a natural preference interpretation. For each feature, the support of the distribution represents the range of tolerable values and corresponds to the interval-value used in the initial retrieval query. The core represents the most desirable range of values and establishes our top preference. By definition, a feature value falling inside the core will receive a preference value of 1. As the feature value moves away from the most desirable range, its associated preference value will decrease from 1 to 0. At the end of this evaluation, each comparable will have a preference vector, with each element taking values in the (0,1] interval. These values represent the partial degree of membership of each feature value in the fuzzy sets and fuzzy relations representing our preference criteria.

For example, by using the preference distributions shown in Figure 6.10 we can see that the preference value for the attribute date-of-sale of a comparable that was sold within 3 months of today's date is 1. If the date was 6 months ago, its preference value would be 2/3. Any comparable with a date of sale of more than 12 months would be given a preference value of zero.

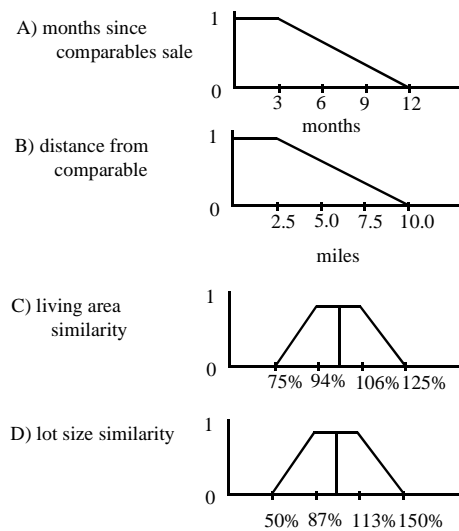


Figure 6. 10 Similarity Membership Functions

The remaining two features, “Number of Bedrooms” and “Number of Bathrooms”, are evaluated in a similar fashion. Their preference functions are represented by two reflexive asymmetric fuzzy relations, illustrated in Figure 6.11

and Figure 6.12, respectively.

Table 6. 1 Fuzzy Relation for Bedrooms

Comparable's		1	2	3	4	5	6+
# Bedrooms							
	1	1.00	0.50	0.05	0.00	0.00	0.00
Subject's	2	0.20	1.00	0.50	0.05	0.00	0.00
# Bedrooms	3	0.05	0.30	1.00	0.60	0.05	0.00
	4	0.00	0.05	0.50	1.00	0.60	0.20
	5	0.00	0.00	0.05	0.60	1.00	0.80
	6+	0.00	0.00	0.00	0.20	0.80	1.00

For instance, using Table 6.1, we can observe that for a subject with 5 bedrooms the preferred comparable would also have 5 bedroom (preference =1), while a 6 bedroom comparable would meet that preference criterion to a degree of 0.8. Similarly, using Table 6.2, we can observe that, for a subject with 2 bathrooms, the preferred comparable would also have 2 bathrooms (preference value =1), while a

Table 6. 2 Fuzzy Relation for Bathrooms

Subject	Comparable								
	1	1.5	2	2.5	3	3.5	4	4.5	5+
1	1.00	0.75	0.20	0.05	0.01	0.00	0.00	0.00	0.00
1.5	0.60	1.00	0.60	0.25	0.10	0.05	0.00	0.00	0.00
2	0.10	0.70	1.00	0.70	0.25	0.05	0.00	0.00	0.00
2.5	0.05	0.20	0.75	1.00	0.75	0.20	0.05	0.00	0.00
3	0.01	0.10	0.40	0.80	1.00	0.80	0.40	0.10	0.05
3.5	0.00	0.05	0.15	0.45	0.85	1.00	0.85	0.45	0.30
4	0.00	0.00	0.05	0.20	0.50	0.90	1.00	0.90	0.70
4.5	0.00	0.00	0.00	0.10	0.30	0.70	0.95	1.00	0.95
5+	0.00	0.00	0.00	0.05	0.15	0.35	0.75	0.95	1.00

6.3.3 Fuzzy Confidence

The users will make critical decisions based on the estimates generated. Therefore, we need to tell them when the system produces an accurate, reliable solution. We achieve this goal by attaching a reliability measure to each estimate. Ideally we would like to have subjects with the highest reliability exhibiting the lowest errors. At the same time we would like to assign high reliability values to as many subjects as possible.

The reliability value is calculated from the following five quantitative characteristics of the case-based reasoning process:

- Number of cases found in the initial retrieval

- Average of the similarity values for the best four cases
- Typicality of problem with respect to the case-base (i.e., if the attributes of the subject fall within typical ranges for the subjects five digit zip code region)
- Span of adjusted sales prices of highest reliability solutions (i.e., the highest adjusted sale price minus the lowest adjusted sale price among the selected comparables)
- Distribution of adjusted sales prices of highest reliability solutions (i.e., average percentage deviation of the adjusted sales price of the comparables from the estimated value of the subject)

These characteristics are evaluated using the fuzzy membership functions illustrated in Figure 6.11. These functions map the numerical value of each parameter into a standard numerical reliability, which ranges from 0 to 1. These standardized reliability values are then aggregated into a final reliability value. Given the conjunctive nature of this aggregation, we decided to use the *minimum* of the standardized reliability values.

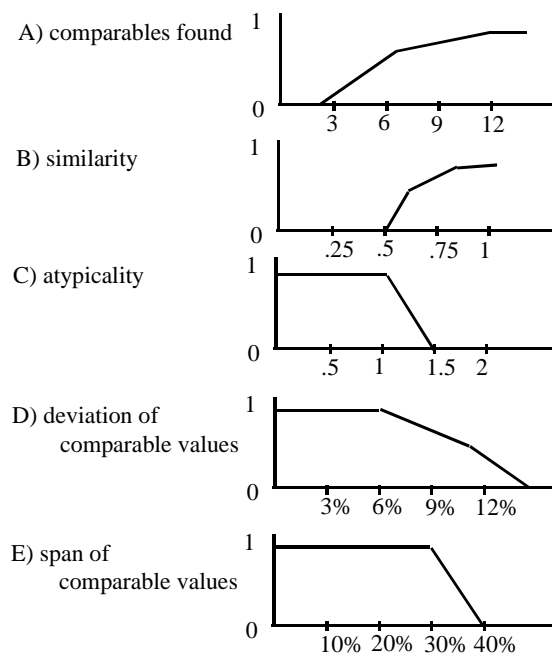


Figure 6. 11 Confidence Membership Functions

Figure 6.11 a) shows that if two or less comparables are found then the standardized reliability for comparables found is 0. If between two and seven comparables are found, the reliability is $((n - 2) * 0.15)$, i.e., the reliability increases 0.15 for each comparable over two to reach 0.75 when there are seven comparables. Between seven and twelve comparables, the reliability is $((n - 7) * 0.05) + 0.75$, i.e.,

the reliability increases 0.05 for each comparable over seven and reaches 1.0 with twelve comparables. Since the aggregation method is minimum operator, a low reliability in any of the characteristics will cause a low reliability in the result regardless of other excellent reliability values for the other characteristics. The other figures show similar membership functions for the other reliability measures.

Figure 6.11 b) shows that we have no reliability in an estimate whose similarity is lower than 0.5. Otherwise the reliability rises to 1 at a similarity of 0.8, and is 1 for anything over 0.8. Figure 6.11 c) is used to determine our reliability based on the subject's atypicality. Atypicality is computed as a normalized deviation in the region's average lot size, living area, number of bedroom, bathrooms, etc. We have no reliability in subjects with atypicality greater than 1.5. Our reliability rises linearly as the atypicality decreases from 1.5 to 1.0 and is one when atypicality is less than 1.0. Figure 6.11 d) shows our reliability for the average deviations in the values of the comparables. We have zero reliability in an estimate if the average comparable deviates from the estimated price by more than 15%. Finally, Figure 6.11 e) is used to determine our reliability based on the size of the span of the adjusted values of the comparables. If the span is greater than 40% of the value of the subject then we considered it too scattered and have no reliability in the estimate.

To create the membership functions for the five characteristics, which are illustrated in Figure 6.11, we ran our system on 7,293 properties from Contra Costa county in California. The predicted sales price of each property was calculated and compared with its actual sales price to derive the estimate's error. The percentage error and its five reliability characteristics were calculated each subject. Table 6.3 shows the values calculated for a random sample of ten of the 7,293 subjects. Each row is a different subject. The columns show the estimate error, the five characteristics calculated along with the estimate, and the reliability value obtained by taking the minimum of the evaluation of the membership functions of Figure 6.11 using the estimate's five characteristics.

Table 6. 3 Confidence Calculation

Error	Comps Found	Simil.	Atyp.	Comps Dev.	Comps Span	Conf. Value
-9.8	3	0.63	1.42	2.02	6.32	0.15
-2	35	0.94	0.38	2.24	8.57	1.00
17.3	11	0.71	0.94	5.67	19	0.70
0.5	24	0.85	0.66	2.05	7.24	1.00
-1.6	14	0.95	0.29	2.89	9.33	1.00
5.2	15	0.90	0.73	3.24	12	1.00
5.2	12	0.74	0.17	4.5	18	0.80
3.1	19	0.74	0.81	2.83	8.11	0.80
-13.9	12	0.82	1.97	3.85	15	0.00
7.8	11	0.77	1.34	4.24	13	0.32

Then we analyzed the conditional distributions of the estimate error, given each of its five reliability characteristics, and try to predict the error. We used C4.5 [16], a decision tree tool, to create rules predicting the error from the system's

characteristics. Then we validated these rules via data visualization. Finally, the rules were manually transformed into the membership functions illustrated in Figure 6.11. The estimate's reliability value is the conjunctive evaluation of all the rules.

6.3.4 Results

At our customer's request, the reliability value generated by the rules was subdivided into three groupings (*good*, *fair*, and *poor*). The reliability measure should then produce the largest good set with the lowest error. Of the 7,293 subjects, we could classify our reliability in 63% as *good*. The *good* set has a medium absolute error of 5.4%, an error which is satisfactory for the intended application. Of the remaining subjects, 24% were classified as *fair*, and 13% as *poor*. The fair set has a medium error of 7.7%, and the poor set has a median error of 11.8%. The PROFIT system was technical success, but shortly after it was completed GE got out of the business of appraising residential property.

6.4 Plastics Color Matching

GE Plastics (GEP) currently provides a color matching service to customers. Customers give GEP a physical sample of the color plastic they want and GEP either finds a close match from their color library or formulates a new color to meet the customer's needs. GEP currently has over 30,000 previously matched colors on file and performs approximately 4,000 color matches per year. When GEP does do a custom color match and formula development, there is a significant cost to GEP and the turnaround for the customer averaged 2 weeks. The goal was to reduce this cost and shorten the turnaround time.

6.4.1 Introduction

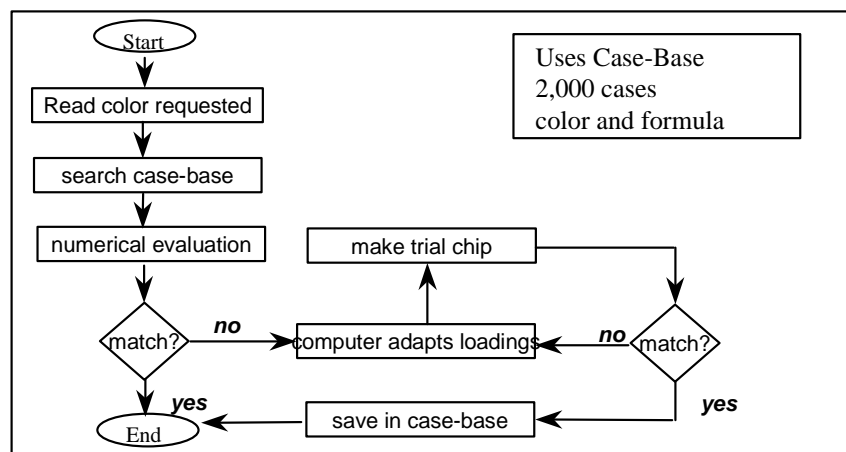


Figure 6. 12 Color Matching Process

Selecting the colorants and loading levels for a color formula was previously accomplished using a combination of human working experience and computationally expensive computer programs. The color matching process that was in place starts with a color matcher inspecting the color request for the type of plastic, physical color standard supplied by customer, and special properties requested. The matcher would then compare the customers color standard with previous color chips that were stored in a filing cabinet. The filing cabinet held about 2,000 plastic chips that were about 2" by 3" by 1/8". These chips were sorted by color. The matcher would select the most similar color from the filing cabinet. Each chip was labeled and another filing cabinet held a formula card for each chip. The matcher would then inspect the physical chip selected from the filing cabinet to determine if it matched the color and special properties requested by the customer. If it did match, then the formula that was associated with the selected chip would be

used for the customer and the match is finished. If the best chip from the filing cabinet was not a satisfactory match, then the matchers used experience along with commercially available computer programs to adapt the colorant loadings. The new loadings would be used to create a small chip containing the adapted loadings. This chip would be compared with the standard. If it was acceptable the adapted formula would be used for the customer and the chip would be placed in the filing cabinet for future reference. If the color was unacceptable then the formula would be adapted repeatedly until an acceptable formula was obtained.

6.4.2 Automated CBR Process

The current process was already a case-based approach where the filing cabinets acted as the case-base and the color matcher searching through the filing cabinet was the case selection. Creating a tool to automate this involved creating a machine readable version of the information in the filing cabinets and then creating the CBR software that performed the search. A numerical representation of the color of the plastic chips and their formulas was stored in a database that acted as the case-base. The automated color matching process developed is shown in Figure 6.12. Figure 6.12 show that the color matcher places the physical color standard in a spectrophotometer and reads the spectrum of the color standard into the color matching system. Next, the color matcher enters key information such as the resin and grade of material in which to generate the match. FormTool then searches its case-base of previous matches for the “best” previous match and adjusts those previous matches to produce a match for the new standard. There are multiple criteria which the color match must satisfy: the color of the plastic must match the standard under multiple lighting conditions, there must be enough pigments to hide the color of the plastic, the cost of colorant formula should be as low as possible, only a limited amount of light can be transmitted through the plastic (optical density), and the color should not change when the plastic is molded at different temperatures.

6.4.3 Fuzzy Case Retrieval

This section describes a method to evaluate the quality of a specific color formula [4]. A selection process that uses this method to evaluate a formula can be used to find the formula that will reproduce a specified color and meet all desired attributes for the application of the specified color. A nearest-neighbor retrieval is used. However, the nearest neighbor must be determined by evaluating the degree of match in all of the attributes described above. This evaluation needs to provide a consistent meaning of an attribute similarity throughout all attributes. The consistency is achieved through the use of fuzzy linguistic terms, such as Excellent, Good, Fair, and Poor, which are associated with measured differences in an attribute. Any number of linguistic terms can be used. A fuzzy preference function is used to calculate the similarity of a single attribute of a case with the corresponding attribute of the subject, see Figure 6.13. In this figure, a difference of 1 unit in the values of that attribute for the subject and comparable would be considered excellent, a difference of 2 would be good, 3 would be fair, and 4 would be poor. This rating is

then transformed into the fuzzy preference function in Figure 6.13.

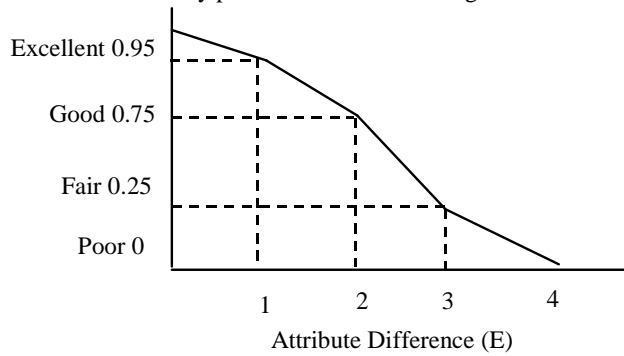


Figure 6.13: Example Membership Function

The result of using fuzzy preference functions is a vector, called the fuzzy preference vector. The vector contains a fuzzy preference value for each attribute. The values in this vector can be combined, through weighted aggregation, to produce a robust similarity value. The use of fuzzy preference functions allows for smooth changes in the result when an attribute is changed unlike the large changes that are possible when step functions are used.

A fuzzy preference function is used to transform a quantifiable value for each attribute into a qualitative description of the attribute that can be compared with the qualitative description of other attributes. A fuzzy preference function allows a comparison of properties that are based on entirely different scales such as cost measured in cents per pound and spectral curve match measured in reflection units. Based on discussions with experts and work to classify previous matches into various sets of linguistic terms we found that there was enough precision in our evaluation of the similarity of the attributes to have four linguistic terms. Table 6.4 shows the linguistic terms and the numeric similarity score that corresponds to each term.

Table 6. 4 Linguistic Terms and Similarity

Fuzzy Rating	Maximum Score	Minimum Score
Excellent	1	0.95
Good	0.94	0.75
Average	0.74	0.25
Poor	0.24	0

Fuzzy preference function were created for each of the following attributes of the color match

- color similarity,
- total colorant load,
- cost of colorant formula,
- optical density of color, and
- color shift when molded under normal and abusive conditions.

The remainder of this section describes how the fuzzy preference functions were constructed for each attribute.

6.4.3.1 Color Similarity

Two different ways of rating the quality of a color match are the spectral color curve match and metamerism of the color. Matching the spectral curve is the best way to match a color for all possible lighting conditions. Minimizing metamerism reduces the color difference under the most common lighting conditions. Minimizing metamerism is the traditional way a color match was done before there was a spectrophotometer that could read the reflectance of a color. Both of these methods are useful in matching a color.

The spectral color curve match is a rating of how closely the color of the formula created matches the color of the standard. A spectral curve is a representation of the amount of light that is reflected from an object at each wavelength of the visible spectrum. Comparing spectral curves of objects is the best way to compare their color, because if the two objects have the same spectral curve, then their colors will match under all lighting conditions. Other color matching techniques only match colors under one lighting condition, so the colors can look quite different under other lighting conditions.

The spectral curve match is characterized by the sums of the squared differences in the reflection values at 31 wavelengths from 400 to 700nm at a 10nm interval. Table 6.5 shows the value of that sum of squares that is needed for an Excellent, Good, Fair, or Poor match. These values are determined by having an expert rate the curve matches in the case-base and then finding the minimum value for each of the ratings, excluding a few outliers.

Table 6. 5 Fuzzy Curve Match

Fuzzy Rating	Maximum Sum of Squares Difference
Excellent	0.000124
Good	0.000496
Fair	0.001984
Poor	0.007936

For example, a sum of square difference of 0.000124 is the maximum difference for an excellent rating, from Table 6.5. The score corresponding to this would be 0.95, the minimum score for excellent from Table 6.5. Sum of square values between the minimum and maximum values have scores that are linearly interpolated from the minimum and maximum values for that rating. Figure 6.14 shows the fuzzy preference function for curve match.

A second measure of the curve match is the metamerism index. Metamerism takes place when two objects are the same color under one lighting condition, but different under another lighting condition. Two color chips can look the same under one lighting condition (sunlight) and different under another (indoor light). This is the effect of metamerism. There are specific lighting conditions that are more common than others. If the spectral curve match is good, there can still be metamerism among the primary lighting conditions. This metamerism should be as small as possible.

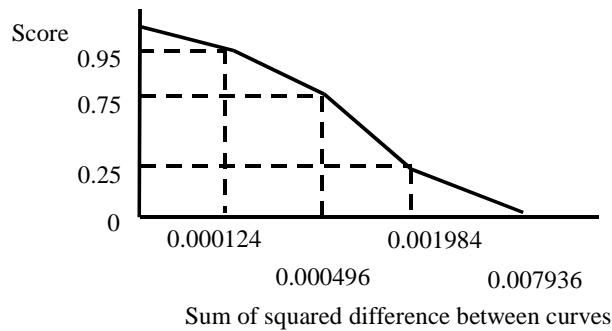


Figure 6. 14 Color Match Membership Function

The metamerism index is measured in dE^* units using the International Commission on Illuminations $L^*a^*b^*$ color scale [81]. In order to calculate dE^* , the stimulus color is determined from the multiple of the spectral power curve of a light source and the reflectance of the object in question. The $L^*a^*b^*$ scale is a three-dimensional space for representing color with L (lightness) on the up-down axis, a (red-greenness) on the left to right axis, and b (yellow-blueness) on the front to back axis. This color space represents color under one lighting condition (spectral power curve), unlike the spectral curve that represents color under all lighting conditions. The difference in $L^*a^*b^*$ values is computed by determining the difference in the L^* , a^* , and b^* values of the two colors. These differences are called dL^* , da^* , and db^* respectively. Then, dE^* is calculated using the formula $E = \sqrt{(L_1 - L_2)^2 + (a_1 - a_2)^2 + (b_1 - b_2)^2}$

In order to use the $L^*a^*b^*$ color scale an illuminant needs to be selected. The standard illuminants that we selected are C (overcast-sky daylight), D65 (normal daylight), and F7 (florescent light). These represent the most common lighting conditions and give visual emphasis on the high, middle, and low wavelengths of the spectral curve respectively. dE^* is the sums of the squared differences in the L^* , a^* , b^* values of two colors.

The metamerism index is the sum of the three dE^* using each of the three illuminants. The mapping of the metamerism to the fuzzy preference scale is presented in Table 6.6. This table and all other similar ones are generated from an analysis of the case-base. The fuzzy preference function generated from Table 6.6 is created easily from the table. These two properties (i.e. curve match and metamerism) specify how closely the color produced by a formula matches the color of a standard.

Table 6. 6 Linguistic Terms for Metamerism

Fuzzy Rating	Maximum dE*
Excellent	0.05
Good	0.15
Fair	0.4
Poor	1.0

6.4.3.2 Total Loading

The total colorant load is total volume of all colorants used for a set volume of plastic. It is best to use the least volume of colorants that makes an acceptable match, for reasons relating to the manufacturing of the plastic.

The quality of the remaining properties depends on the color that is being matched. For example, a cost that is good for a red color might be poor for a white because reds are much more expensive. In order to use fuzzy preference functions for these attributes the case-base must be subdivided into portions that have consistent values for the properties. We have divided the case-base into eleven classes. For each attribute, the fuzzy ratings needed to be calculated separately for each subclass.

An attribute that uses these subclasses is the total loading of colorant in the formula. The total colorant loading of the formula can be characterized in parts per hundred (pph) parts of base material such as plastic. The total colorant loading is dependent upon the color to be made. Table 6.7 shows the fuzzy ratings for total colorant loading for the white and green color subclasses. The rest of the subclasses are similar to the ones presented. Historically, whites tend to require much more colorant than a green color. This is because it takes more of the white colorant to hide the color of the plastic. The difference in typical loadings is accounted for by using separate tables for separate colors. A fuzzy preference function can be easily constructed for each subclass.

Table 6. 7 Linguistic Terms for Total Loading

Fuzzy Rating	White pph	Green pph
Excellent	3	0.4
Good	5	0.7
Fair	7	1.1
Poor	11	3.2

6.4.3.4 Cost

The cost of the colorants in the formula should be kept to a minimum to maximize the profitability of the manufacturing process. The attribute cost is measured in units of cents per pound. The fuzzy ratings for this attribute are specific for particular color subclasses, as illustrated for the red and blue subclasses in Table 6.8. There is a difference in the mapping for the red and blue color families because the cost to make a red tend to be more expensive than the colorants used to make a blue.

Table 6. 8 Linguistic Terms for Cost

Fuzzy Rating	Red Cost (cents/lb)	Blue Cost (cents/lb)
Excellent	4.5	2
Good	9	3.5
Fair	25	10
Poor	72	28

6.4.3.4 Optical Density

The optical density of plastic is the thickness of plastic that is required to stop all light from radiating through the plastic. A specific optical density is required for many applications. For the majority of color formulas, it is desired to make the material opaque to prevent light from transmitting through the material. Optical density can be used to characterize how much light is transmitted through a sample. The type of colorants used in a formula and the loading level of the colorants determine the optical density of the material. The qualitative values of optical density are color dependent. For example, it is easier to obtain the needed optical density in an opaque grey color formula than in a red color.

Table 6.9 shows fuzzy rating of optical density for a grey and a red color.

Table 6. 9 Terms for Optical Density

Fuzzy Rating	Grey dE*	Red dE*
Excellent	5.9	5.9
Good	5.8	5
Fair	5.5	2
Poor	4	1

6.4.3.5 Hide Color Shift

The color shift when molding under normal and abusive conditions comes from the fact that the plastic can be molded at low and high temperatures. The same plastic is a slightly different color when molded at different temperatures, because plastic tends to yellow at higher temperatures. In order to minimize the color shift, extra colorant loadings need to be used. A formula must also be robust enough to hide these color changes in the base plastic. One way to characterize this attribute of hiding variations due to process conditions is to measure the color of the material under normal processing conditions and under abusive processing conditions. The difference in color between these two processing conditions is then measured in dE* units using the CIE L*a*b* color scale. Table 6.10 shows the process color change in dE* units mapped between the grey and yellow color subclasses. Visually, a larger change in color due to processing conditions can be tolerated in a light yellow color than a grey color as shown by this mapping based on historical data.

Table 6. 10 Linguistic Terms for Color Shift

Fuzzy Rating	Grey dE*	Yellow dE*
Excellent	0.05	0.05
Good	0.10	0.15
Fair	0.2	0.4
Poor	0.5	1.0

6.4.4 Aggregate Fuzzy Preference Values

Each of the above properties including spectral color match, metamerism index, loading level, cost, optical density, and color shift due to processing conditions, is based on different scales of units. By mapping each of these properties to a global scale through the use of fuzzy preferences and linguistic terms such as Excellent, Good, Fair, and Poor, it becomes possible to compare one attribute with another.

The next step is to create a summation of the preference value of each attribute. This can be done with a weight of unity for each attribute or the end user can supply weights of their own if they desire to emphasize one attribute over another. Dividing this summation term by the summation of the weights gives the global preference value for the system.

6.4.5 Results

GE Plastics has obtained significant savings in both time and money by using the CBR software for color matching since the beginning of 1995. The hard savings that have been documented come from the optimization of pigment and dye concentrations and the reduction in the number of trial batches required for a color match. When the tool was first rolled out at the end of 1994, the optimization algorithms for pigment and dye concentrations was run against the historical database of formulas resulting in the identification of formulas that could have their colorant concentrations reduced while still maintaining the quality of the color match. These changes lead to a reduction in the raw material cost of the final product generating significant cost reductions to the company. The reduction in the number of trial batches required to obtain a color match has lead GE Plastics to cut its lead time for a custom color match down from an average of 2 weeks to a 48 hour turn-around time. This lead time reduction has been a great help to our customers. Many of our customers have been trying to shorten the development time for their new products. If we can shorten the color selection time by two weeks then it can shorten their development cycle time by two weeks at no cost to them.

6.5 Adaptive Fuzzy Classification in Aircraft Engines

The CBR application shown here diagnoses aircraft engines based on sensor readings taken at the engines. This is a difficult diagnostic problem because the baseline for the sensor values changes unpredictably – within bounds – over time. To cope with that situation, the implemented system continuously adapts cases to match criteria set up for baseline cases. This can conceptually be done by either adapting the baseline cases or by adapting the complement set. To allow the process to be applied to a multitude of different engines, we chose to adapt the baseline case and to keep the complement set fixed. The adaptation was performed marrying statistical process control techniques with fuzzy knn [7]. In a further step, the incoming cases are not actually stored. Rather, a prototype case is updated with the new information. This allows a substantial reduction of storage requirements because only a few parameters are needed to be carried forward through time. In addition, it eliminates the need to delete older samples because that information is already integrated into the parameters. The remaining system is comprised of a fixed set of cases evenly spaced through the problem domain. Incoming cases are pre-processed to fit that scheme and then are used to update the pre-processing scheme. The operation is shown for abnormal condition detection on a continuous data stream from engine measurements of an aircraft engine from GE Aircraft Engines (GEAE).

6.5.1 Introduction

It is desirable to be able to detect incipient failures before they cause costly secondary damage or result in equally undesired engine shutdowns. To that end, service providers have long tracked the behavior of engines by measuring and analyzing a multitude of system parameters to become aware of changes that might be indicative of developing failures [15]. However, often times faults may not be recognized distinctively because the “normal” operation condition varies unpredictably. This is due in part to large amounts of noise, poor corrections of first principle models or regression models, changes of schedules, maintenance, poorly calibrated and deteriorating sensors, faulty data acquisition, etc.

When the sensor readings are analyzed in trend analysis tools, the data and parameter settings are evaluated for deviations from normal operating conditions. An alarm is raised when threshold settings are exceeded, assuming the reading is not an outlier. To create awareness of potential problems before they turn into events that result in revenue loss (such as grounding of an airplane, damage to the engine, departure delays, etc.) the settings used in trend analysis are typically below the limit that would cause an immediate shutdown. Once made aware, the analyst examines trend charts to determine whether something real has happened. If he/she can corroborate the initial finding, remedial actions (boroscopy, engine washing, overhauls, etc.) are suggested. A fully automated notification is hard to come by

because of the need to detect all abnormal conditions. This implies that thresholds have to be set to a level that can insure zero false negatives. However, in the presence of very noisy and drifting systems, this means that alarms are generated excessively. That in turn means the user must look at trends more often than desired to weed out the faulty alarms from the true ones.

Prior work in this area was done, among others, by Karamouzis and Feyoock who proposed a system integrating model based and CBR techniques [20]. This system contained a self-organizing memory structured as a frame-based abstraction hierarchy for storing previously encountered problems. The cases were aircraft accident reports. Sensor inputs were gas temperature, engine pressure ratio, and human input such as sound of explosion and smell of smoke in passenger cabin. The time sequence was also of importance in establishing a causal relation such as an aberrant reading of the fan speed before an abnormal observation of the compressor speed. The system incorporated a functional dependency implemented via a digraph and a causality submodel describing transitions between various states of the system. The system searched its case library for the most similar case by a weighted count of corresponding symptoms and gave higher degree of similarity for symptoms occurring early in the fault occurrence.

Reibling and Bublin proposed a system that integrates pattern-matching techniques with causal networks for OMS to model normal and abnormal behavior [26].

Fernandez-Montesinos used Kohonen's feature maps and expert systems to support the interpretation of in-flight monitoring data and to ensure more consistent engine condition monitoring (ECM) [16]. ECM was supported by performance trend analysis software (ADEPT) which captured deterioration and failure of engine parts. Shift patterns allowed the localization of problems within the engine modules. The two main tasks tackled were recognition of a pattern indicating a possible problem, and the interpretation and further analysis. Kohonen's feature maps were chosen because of their self-organizing properties and the incremental extension of the domain of the patterns. Output of the Kohonen's feature map was an error code that indicated whether the module recognized a problem. As starting points for the network, fingerprints provided by the engine manufacturer (GE) were used. An expert system controlled the proposed neural net and interpreted the output.

Records and Choi [25] investigated strategies to filter spurious symptoms in aircraft engine fault monitoring systems (resulting from the monitoring system's inability to accurately assess the expected value from an engine model). They used a back-end knowledge based approach and a front-end neural network that generated expectation values for the monitored sensor.

Gomm [18] and Patel [24] both suggested the use of a self-adaptive neural network with on-line learning capabilities. Gomm used radial basis function networks to which new output nodes were automatically added when a new process fault is encountered. Adaptation was achieved using recursive linear algorithms that train the localized network parameters.

6.5.2 Fuzzy Adaptive Clusters

The approach for adaptive CBR combines multi-variate data evaluation using

fuzzy clusters with an exponential filter, which lets the clusters adapt to changing environments. The adaptive properties of the clusters allow the centroids to move and their shape to vary [6,7].

6.5.2.1 Fuzzy Clusters

In contrast to traditional trend analysis tools, we evaluated fault conditions in multivariate feature space. Small changes in one variable may result in small changes of other system variables and this behavior can be detected more easily in multivariate space. For example, engine variables exhaust gas temperature (*EGT*), fuel flow (*w_f*), and turbine speed (*n₂*) are correlated for some faults. If air leaks from the compressor, a “bleed” problem, the engine will run less efficiently. To maintain a demanded thrust level the controller will cause more fuel to be injected into the combustors. This in turn will raise the turbine speed as well as the exhaust gas temperature. Depending on the amount of bleed, the change in any of the engine parameters may be too small to be recognized alone. However, the change is more pronounced in a higher-dimensional space, and should therefore be detected more easily. Yet, depending on the amount of noise, there may still be overlap between the clusters, i.e., variables cannot be partitioned clearly, and the potential for misclassification remains high. Adding more features to the classification scheme does not necessarily improve the result because the decreasing information gain in Euclidean space is blurred additionally by decreasing feature quality, assuming one starts out using the highest-quality feature and adding others in decreasing order of value. Beyond a certain threshold, adding more features may actually decrease the classification ability.

Within the *EGT-w_f-n₂* space, data have different meaning depending on where they are located. Data behaving normally will be found mostly in a cluster labeled “normal.” Data representing an alarm condition, such as *EGT*, *w_f*, and *n₂* data that are larger than data in the normal cluster are located in the “leak fault” cluster. We avoid using rate of change as a feature based on the assumption that the abnormal condition will show up at least an order of magnitude faster than a slow normal drift (for example due to expected wear). The clusters are of non-uniform size with typically nonlinearly decreasing boundaries. That is, the evaluation of the cluster membership is carried out such that the degree of membership is largest at the cluster center, nonlinear slope outward. This allows training data to be separated more easily. We use the fuzzy *c*-means clustering algorithm [27] for this purpose which seeks to minimize an objective function defined by

$$\min J(U, V) = \sum_{j=1}^n \sum_{i=1}^k u_{ij}^m d^2(x_j, v_i)$$

where

U is a set of a fuzzy k-partition of the data set

V is a set of k prototypes

x is the feature vector of the normalized data

v is the centroid of a cluster

u is the degree of membership x_j on cluster i

m is a weighing component which controid the fuzziness of the cluster
 $d^2(x_j, v_i)$ is an inner product metric (e.g., distance) between x_j and v_i
 n is the number of data points
 k is the number of clusters

After choosing the initial number of clusters and their position, the membership is determined by

$$u_{ij} = \frac{\left(\frac{1}{d^2(x_i, v_j)} \right)^{\frac{1}{m-1}}}{\sum_{k=1}^n \left(\frac{1}{d^2(x_k, v_j)} \right)^{\frac{1}{m-1}}}$$

Next, a new centroid v_i is determined using

$$v_i = \frac{\sum_{j=1}^n (u_{ij})^m x_j}{\sum_{j=1}^n (u_{ij})^m}$$

At this point the u_{ij} are updated again and the cycle is repeated until a stopping criterion such as

$$\sum_{j=1}^k \sum_{i=1}^n (u_{ij} - u_{ij_{new}})^2 < \epsilon$$

is reached.

6.5.2.2 Cluster Adaptation

Since thermal, chemical, and mechanical wear degrade the performance of the engine, normal behavior is constantly redefined. Although these changes are expected, they are hard to predict with necessary accuracy because they are driven to a large degree by a whole host of external factors. For example, the effect of maintenance such as replacement of parts, cleaning, etc. is very hard to capture. To retain desirable classification properties these changes must also be echoed in changes of the clusters. Therefore we guided the centroids with slow exponential weighted moving average filters. Only data belonging to a cluster were used for updating the centroid of that particular cluster. For example, data of type "normal" were used to update the cluster "normal" but not the cluster "leak fault" and vice versa. After choosing the initial number of clusters, their position and membership, the adaptation of a fuzzy cluster is performed with an exponential weighted moving average filter, which we express as [7]:

$$\tilde{v}^{new} = \tilde{v}^{old} + \tilde{\alpha} \left(\frac{\sum_{i=1}^n \tilde{u}_i^m y_i}{\sum_{i=1}^n \tilde{u}_i^m} - \tilde{v}^{old} \right)$$

where

superscript “ \sim ” denotes the winner

superscript “ $\hat{\sim}$ ” denotes the loser

\tilde{v}^{new} is the new cluster position

\tilde{v}^{old} is the old cluster position

$\tilde{\alpha}$ is the adaptive learning coefficient calculated (for two clusters) by

$$\tilde{\alpha} = \alpha (1 - |\tilde{u} - \hat{u}|)$$

where

α is a constant term, e.g., $\alpha=0.05$

u is the membership value

y_i is the data point under consideration

The adaptation rate, given by $\tilde{\alpha}$, is designed to minimize ambiguity. When the degree of membership in a winning cluster is ambiguous, \tilde{u} and \hat{u} are almost the same. In that case $\tilde{\alpha}$ becomes large, thus trying to separate the indeterminate situation. The winning cluster is moved farther toward the new data point. In addition, it causes a steeper slope near the cluster boundaries and less where the fault classification is already working well. Thus, it has the effect of focusing its learning where it is needed most, as dictated by recent data. As a result, the centroids move with changing normal conditions. This way, noise will be dampened as well, since it gives some weight to past data and acts as an averaging scheme. Like all filters, it will also lag behind true changes in the system.

6.5.2.3 Alertness Filter

Another approach to increase robustness is to judiciously track the cumulative occurrence of suspect readings, rather than triggering the alarm after just one fault classification. We used an *alertness filter* [7] for doing this, which forces the reoccurrence of several suspicious readings before an alarm is issued. If only one suspicious reading is encountered, the value of an “alertness counter” - which would be zero under normal conditions - is incremented. If a second suspicious reading is encountered immediately afterwards, the counter value is increased. Otherwise, it is decreased. In the latter case, the first suspicious reading is treated just as an outlier and no maintenance is recommended. If it is not an outlier, i.e., the condition persists, the level of the alertness will go up until a full alarm is issued. At that point the user will be alerted. Further fault readings will not increase the alertness level once the alarm threshold is reached. The alertness level threshold is limited to a user-specified number. This saturation avoids lengthy recovery from alarm conditions when the system recovers either spontaneously or through maintenance. The purpose of the alertness filter is to further improve the classification rate of the system.

4.5.3 Results

The adaptive CBR scheme was tested with historical gas turbine data. Different events of similar fault conditions were examined and divided into training and test cases. For training, the use of both deliberate placement and random seed points (for the cluster centroids) were investigated. Placing the starting points into the general expected neighborhood resulted in faster learning and convergence. Convergence was on average accomplished after about 20 iterations, depending on the start conditions and final acceptable convergence error. Typically, there is a much larger amount of data for normal conditions and only few data for fault conditions. To prevent a fitting bias for normal conditions, those data were undersampled so that “normal” and “faulty” operation were equally represented. Figure 6.15 shows the distribution of clusters after training in $EGT-n_2$ space.

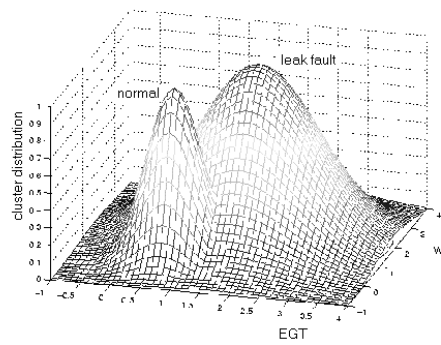


Figure 6. 15: Fuzzy cluster for normal and leak fault conditions displayed in $EGT-w_f$ space

After training, the system was used for classification of on-line test data. The false negative rate is as low as with traditional trending tools (none observed with the test data available) and the false positive rate improved from 95% to less than 1%. Figure 6.16 shows the adaptation of a cluster during operation.

The “x” denotes the location of the centroid of cluster “normal” and the squares denote the location of a fault cluster. The normal region changes initially slowly with normal wear and then undergoes a considerably change in one dimension as seen by the vertical jump of the operating point upward in the graph. Because there were no changes in the other two dimensions, the algorithm adapts correctly to the new location while retaining the ability to detect faults. The adaptation also exhibits the lag introduced by the exponential filter.

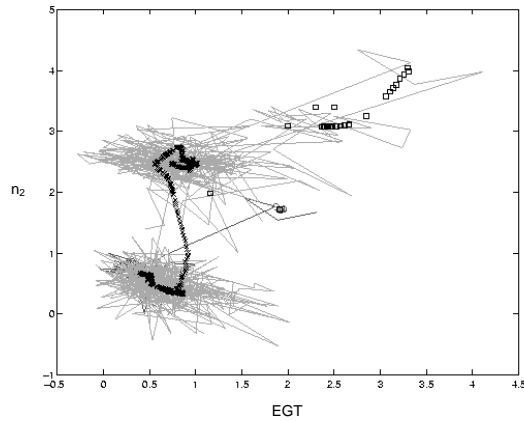


Figure 6.16: Adaptation of clustering during operation [7]

6.6 Summary and Conclusion

We demonstrated how SC techniques were used in a variety of fielded CBR systems. In doing so we were able to leverage the tolerance for imprecision and uncertainty which is intrinsic to SC techniques. The payoff of this conjunctive use of SC and CBR techniques is a more accurate and robust solution than a solution derived from the use of any single technique alone. This synergy comes at comparatively little expense because typically the methods do not try to solve the same problem in parallel but they do it in a mutually complementary fashion.

In the examples of medical equipment diagnostics and residential property valuation, we showed how simple membership functions can be used to implement the case selection and confidence evaluation. The membership functions allowed a greater selection accuracy through their noise tolerance. The ability to determine a confidence in a result made the systems much more useful to the end users. In addition, confidence membership functions also facilitated maintainability.

The third example, a plastic color matching system, combined several selection criteria. Handling multiple selection criteria allowed the color matchers to detect potential problems during the case selection phase while previously these problems were not detected until the color match was complete. This avoided considerable amount of rework and frustration by the color matchers.

Finally, we have presented the use of adaptive fuzzy clusters for fault classification of aircraft engine sensor data. The advantages of this system were its ability to deal with extremely noisy sensors and to adjust to unpredictable slow drift. Dramatic performance improvement was demonstrated during testing. This application also featured hybrid use of soft computing techniques by employing both statistical process control techniques as well as fuzzy techniques within the same system.

A step in further improving system performance is the use of more hybrid

systems, such as evolutionary computing for parameter learning or the exploitation of parallel systems. The latter may be designed to rely to the maximum amount on non-overlapping data and use different techniques to arrive at their conclusions in an information fusion [17] scheme where the outputs of these heterogeneous models will be compared, contrasted, and aggregated.

6.7 References

- [1] Aamodt A (1994) Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AICOM Vol. 7 No. 1, pp. 39-59.
- [2] Althoff K-D (1995) A Review of Industrial Case-Based Reasoning Tools. AI Intelligence, Oxford, UK
- [3] Appraisal Institute (1994) Appraising Residential Properties. Part VI Chicago IL
- [4] Billmeyer F (1981) Principles of Color Technology: Second Edition. John Wiley & Sons, New York
- [5] Bonissone P, Cheetham W (1998) Fuzzy Case-Based Reasoning for Residential Property Valuation. in Handbook of Fuzzy Computation Institute of Physics Publishing, Section G14.1, Bristol, UK.
- [6] Bonissone P, Goebel K (1999) Soft Computing Techniques for Diagnostics and Prognostics. AI in Equipment Maintenance Service and Support, 1999 AAAI Spring Symposium, Technical Report SS-99-04, AAAI Press, Menlo Park, CA
- [7] Bonissone P, Chen Y, Goebel K, Khedkar P (1999) Hybrid Soft Computing Systems: Industrial and Commercial Applications. Proceedings of the IEEE. September 1999. Vol 87. No 9. pp 1641-1667.
- [8] Case K, Shiler R (1987) Prices of Single-Family Homes Since 1970: New Indexes for Four Cities. New England Economic Review Sept./Oct.
- [9] Cheetham W (1996) Case-Based Reasoning with Confidence. Ph.D Thesis Rensselaer Polytechnic Institute
- [10] Cheetham W, Graf J (1997) Method and System for Formulating a Color Match. U.S. Patent No. 5,668,633
- [11] Cheetham W, Graf J (1997) Method and System for Selecting a Previous Color Match from a Set of Previous Matches U.S. Patent No. 5,841,421
- [12] Cuddihy P, Shah R (1995) Method and System for Analyzing Error Logs for Diagnostics. US Patent 5463768

- [13] Cuddihy P, Cheetham W (1998) System and Method for Estimating a Measure of Confidence in a Match Generated from a Case-based Reasoning System. US Patent 5799148
- [14] Cuddihy P, Cheetham W (1999) ELSI: A Medical Equipment Diagnostic System. Third International Conference on Case-Based Reasoning Munich, Germany
- [15] Doel, D. (1990) The Role for Expert Systems in Commercial Gas Turbine Engine Monitoring, Proc. of the Gas Turbine and Aeroengine Congress and Exposition, Brussels, Belgium.
- [16] Fernandez-Montesinos M., Janssens P., Vingerhoeds R. (1994) Enhancing Aircraft Engine Condition Monitoring. Safety, Reliability and Applications of Emerging Intelligent Control Technologies. A Postprint Volume from the IFAC Workshop on Emerging Intelligent Control Technologies, pp161-166. Hong Kong.
- [17] Goebel K (1999) Diagnostic Information Fusion for Manufacturing Processes, in Proceedings of the Second International Conference on Information Fusion, vol. 1, pp 331-336, Sunnyvale, CA.
- [18] Gomm J-B (1995) Process-Fault Diagnosis Using a Self-Adaptive Neural Network with On-Line Learning Capabilities. Proc. IFAC On-Line Fault Detection and Supervision in the Chemical Process Industry. pp 69-74 Newcastle upon Tyne, UK
- [19] Gonzalez A (1992) A Case-Based Reasoning Approach to Real Estate Property Appraisal. Expert Systems with Applications Vol. 4 pp. 229-246
- [20] Karamouzis S, Feyock S (1993) A Performance Assessment of a Case-Based Diagnostic System for Aircraft Malfunctions. Proc. of the Sixth Int. Conf. Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Edinburgh, Scotland. pp 71-78.
- [21] Keller J, Gray M, Givens, J (1985) A Fuzzy K-Nearest Neighbor Algorithm in IEEE Transactions for Systems Man & Cybernetics. vol. SMC-15, no 4, pp 580-585
- [22] Klir G, Folger (1988) Fuzzy Sets, Uncertainty, and Information. New Jersey, Prentice Hall
- [23] Kolodner J (1993) Case-based Reasoning. San Mateo, CA, Morgan Kaufmann
- [24] Patel V-C, Kadirkamanathan, Thompson H-A (1996) A Novel Self-Learning Fault Detection System for Gas Turbine Engines. Proc. UKACC Int. Conf. on Control '96, Exeter, UK. pp 867-872
- [25] Records R, Choi J (1994) Spurious Symptom Reduction in Fault Monitoring Using a Neural Network and Knowledge as Hybrid System.

Proc. of the Twelfth National Conf. on Artificial Intelligence, vol. 2, Seattle, WA. pp 865-870.

- [26] Reibling R, Bublin S (1993) Diagnostic Reasoning Technology for the On-Board Maintenance System. Proc. IEEE 1993 National Aerospace and Electronics Conf. NAECON 1993. vol. 2. Dayton, OH. pp 930-936.
- [27] Ruspini E (1963) A New Approach to Clustering, in Information Control. vol. 15 no.1 pp22-32
- [28] Zadeh L (1965) Fuzzy Sets. Information and Control 8:338-353