Combining Case Based Reasoning with Neural Networks

Roderick Murray-Smith, Sunil Thakar
Daimler-Benz Systems Technology Research, Alt-Moabit 91b, D-1000 Berlin 21, Germany.
E-mail: murray@DBresearch–berlin.de, thakar@DBresearch–berlin.de

Abstract: This paper presents a neural network based technique for mapping problem situations to problem solutions for Case-Based Reasoning (CBR) applications. Both neural networks and CBR are instance-based learning techniques, although neural nets work with numerical data and CBR systems work with symbolic data. This paper discusses how the application scope of both paradigms could be enhanced by the use of hybrid concepts. To make the use of neural networks possible, the problem's situation and solution features are transformed into continuous features, using techniques similar to CBR's definition of similarity metrics. Radial Basis Function (RBF) neural nets are used to create a multivariable, continuous input-output mapping. As the mapping is continuous, this technique also provides generalisation between cases, replacing the domain specific solution adaptation techniques required by conventional CBR. This continuous representation also allows, as in fuzzy logic, an associated membership measure to be output with each symbolic feature, aiding the prioritisation of various possible solutions. A further advantage is that, as the RBF neurons are only active in a limited area of the input space, the solution can be accompanied by local estimates of accuracy, based on the sufficiency of the cases present in that area as well as the results measured during testing. We describe how the application of this technique could be of benefit to the real world problem of sales advisory systems, among others.

Key Words: Case-Based Reasoning, Radial Basis Function Neural Networks, Sales Advisory Systems.

1. Introduction

Case-based Reasoning (CBR) is a method of using previous episodes to suggest solutions to new problems [10]. CBR allows a reasoner to solve problems efficiently when previous similar experiences are available. Problem solving using case-based reasoning usually involves retrieving relevant previous cases, adapting the solution(s) from the previous case(s), if necessary, to solve the problem and storing the current episode as a new case to be used in the future [18].

Artificial Neural Networks are also instance based learning systems, which use training sets of examples in the form of input-output vector pairs. The networks then optimise their parameters and possibly structure to learn the mapping from input to output. They work with numerical data, as opposed to symbolic, and expect the data to be pre-processed to a form where the Euclidean distance between situations has as much meaning as possible.

A hybrid approach, combining principles from the two approaches is described with a suitable application domain, namely that of sales advisory systems.

2. The Problem Domain: Sales Advisory Systems

In the present business environment, especially in the manufacturing industry, the functions of sales organisations have become increasingly complex as products are becoming multivariant and customer demands high and specific. Due to the increased complexity of sales situations, sales advisory systems are of utmost importance. In this context, our work begins with a real world problem. Let us consider the following scenario:

Scenario: A potential customer is looking for an automobile and cannot decide alone which particular model of a specific automobile manufacturer with what different accessories to buy. A potential dialogue between the salesperson and customer could be of the following nature:

C00: Hi! I'm looking for a car for myself.
S01: Hi! What kind of car are you looking for?
C01: I don't know! Can't make up my mind...
S02: Well, never mind! What do you want this car for? [business, home]
C02: Business & Home
S03: What kind of cars do you prefer? [fast,medium,slow]
C03: Fast
S04: What do you look for in your car? [safety,performance,comfort]
C04: Performance
S05: How much money would you like to spend?
C05: approx. 70000 DM
.
S10: What profession are you in?
C10: Engineer
S11: How old are you?
C11: 45
.
S15: Now, a final question! What other brand of car would you consider?
C15: Trabant GTE

Once the customer's wishes have been collected, the salesperson then tries to apply following kinds of rules to map the customers preferences to various product models and match the accessories which best fulfill the customer's requirements. A typical rule based approach could be like this:

IF purpose to buy a car = Business & Home
& customer has kids = yes
& customer prefers = fast
& customer objective = performance
.
THEN suggest automobile range = medium

another rule which narrows down this suggestion may be as follows:

IF Age of Kids = 0-3
THEN Suggest automobile Type = X00TD (station wagon)
& Suggest accessories = baby seats

Once the basic model of an automobile is picked by the salesperson then other customer requirements must be fulfilled by adding certain accessories to the basic model. Typically, this part of the counselling is the most difficult one, as the salespeople have not only to satisfy the customer but to also achieve the maximum for themselves and their manufacturer. In other words, a classic win-win situation. To deal with this problem the salespeople usually have two methods; either to provide the customer with the complete list of accessories available for the selected basic model and let the customer decide what he/she wants, which is tedious and time consuming, or they fall back to their past experience and the knowledge about the trends to suggest a combination of accessories which best suit the customer needs. In practice, a more experienced salesperson prefers the latter approach. This is a typical example of the trade-off between *a priori* knowledge and observations needed.

A typical case which is stored in a salesperson's memory may look as follows: *A customer whose profession is engineer, aged between 30-45, has 2 kids of age between 0-3, prefers fast cars with high performance and is prepared to pay ~70000DM .... bought a car of Model X00–TD with accessories x,y,........,z.*

If, in the due course of time, a similar kind of customer appears, then the salesperson uses the past experience and remembers a similar instance of a previous sale and presents this retrieved instance either as it is or takes the retrieved case as a reference, adjusts it to the new situation and then presents it to the customer.

2.1 The need to develop sales advisory systems

The scenario described above tries to sketch a typical consultation situation as things are today. A close look at this scenario description shows the complexity involved in delicate decision making situations where no 'best' answer exists when developing appropriate acquisition plans that satisfy customers requirements, while meeting applicable financial and organisational objectives. Here, the question is not simply to optimize some specific objective. Any proposed solution will be the result of balancing many competing goals. Furthermore, merely suggesting a plan would be insufficient. Explanations are required to persuade and convince the decision maker that the proposed solutions are reasonable.

The present situation in this field is that the salesperson does everything from requirements analysis to product configuration; from present organisational and financial situation analysis to suggesting an appropriate solution plan manually. This task requires an enormous amount of knowledge and experience of a salesperson in various subfields ranging from product component and configuration knowledge to financial marketing etc. Today's ever changing product development and financial market situations do not allow all salespeople to have the same degree of experience and knowledge about every subfield involved in the decision making, hence making the consultation task even more difficult. Computer support in this situation can be of great help.

The purpose of our current work is not only to facilitate and accelerate the sales consultation process but to improve upon the quality of the consultation as well. The quality of the sales consultation is defined by us in two ways: (1) large number of alternative solutions to be considered in the minimum time and (2) successful outcome of the consultation, i.e. how well does the sales object offered to the customer suit the customer's environment or specific needs.

3. Case Based Reasoning

Crucial steps in a case-based reasoning (CBR) process include finding a good match to a new problem, adapting a previous solution to successfully solve the new problem and deciding how to index and store a new case for later effective retrieval [1].

In the course of CBR research, several guidelines for index selection have been proposed: (1) indices should be predictive, (2) indices should be abstract enough to make a case useful in a variety of future situations, (3) indices should be concrete enough to be recognizable in future cases and (4) predictions that can be made should be useful [11].

Accordingly, in the literature one finds different types of case indexing/matching approaches to help resolve a given case : (1) Template matching, (2) Nearest Neighbour matching, (3) Inductive indexing and (4) Prototype Indexing [4].

According to the different needs of the application domain these techniques can either be used individually or can be combined to take advantage of the inherent strengths of each. For example, if dynamic case retrieval capabilities are required then (1) and (2) allow this, requiring little or no pre-indexing of cases but at the expense of retrieval accuracy and speed. (3) and (4) allow the user to build a more accurate and efficient retrieval structure at the expense of dynamic adjustment. Due to the nature of our application domain (Section 2) none of the above mentioned techniques seemed to be sufficient, hence we developed a hybrid technique based upon Radial Basis Function (RBF) Networks, with pre- and post-processing based on ideas from the CBR approach.

Another issue which we are going to tackle in our paper is case adaptation: When a new problem is encountered, the system first retrieves one or more cases that are similar to the new problem. Typically, no case matches the new problem exactly, so the system must adapt one of the retrieved solutions to the new problem. Previously, many CBR systems were designed to solve new problems by adapting solutions to similar, previously solved problems. For example; GINA [6], CYRUS [9], PROTOS [3], EACH [19], CHEF [7], CABOT [5] and PERSUADER [22]. In all these systems, their authors have experimented with different case adaptation techniques and all the solutions suggested in this regard are domain dependent. In our approach, case adaptation is inherent to the neural network approach, and generic in nature (Section 4).

Kolodner mentions the two styles – problem-solving and interpretive – of CBR [11]. In the problem solving style of case-based reasoning, solutions to new problems are derived using old solutions as a guide. CBR of this type supports a variety of problem solving tasks, including planning, diagnosis and design. In the interpretive style new situations are evaluated in the context of old situations. This style is generally useful for situation classification; the evaluation of solution; argumentation; the justification of a solution, interpretation or plan; and the projection of effects of a decision or plan. Our present work belongs to the latter category.

All of the CBR systems developed to date [20] have been for classical domains like medicine, law or cooking, which all require exact solutions rather than approximations. What makes our domain different – and somewhat unconventional – from the traditional domains of CBR is that in our application no tightly constrained solutions are required and the proposed solutions can always be modified. What *is* important here is that a good approximation/consensus should be reached in fewer steps, i.e. the system should produce a short list of high quality suggested configurations.

4. A Hybrid Approach

4.1 Comparing CBR to Numerical methods

The CBR approach to machine learning comes overwhelmingly from the symbolic processing community. The fundamental problems being addressed are, however, often common to other fields such as statistics, modelling and pattern recognition. The most obvious analogy is between Cased Based Reasoning with low-level indices and Nearest-Neighbour classifiers. In Nearest-Neighbour classifiers, the output class is that of the training example closest to the current inputs. The indexing problem is solved by making all *n* features of the problem dimensions in hyperspace, then calculating the Euclidean distance from the current inputs to all other examples in the training set. In the *k*-Nearest Neighbours algorithm the nearest *k* cases are taken into account for classification.

This enables us to find the nearest relevant cases, but suffers from the same problems as case-based reasoning, in that a large number of cases are needed, and that the processing cost increases with the number of examples. In many applications it is also desirable to be able to form a smooth decision surface over the input space, as opposed to a collection of piecewise constant areas. A smooth decision surface, interpolating between cases, can then accomplish the same task as the adaptation of cases in the standard CBR implementations. The numerical formulation is, however, a much more general formulation of adaptation.

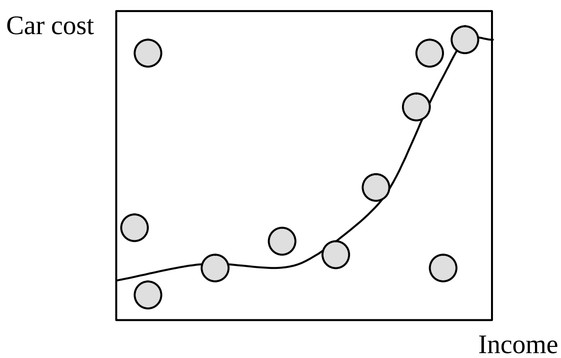
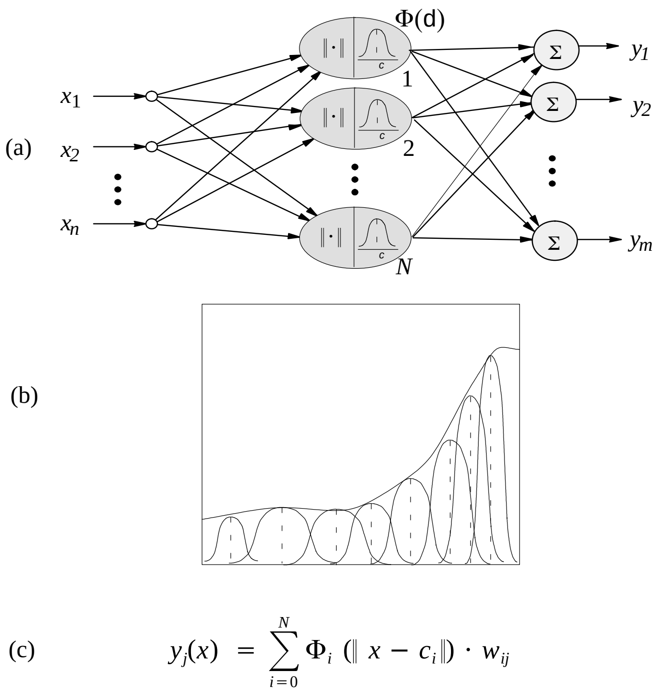


Figure 1: Should we use the individual cases, or fit a decision surface to the data which tries to minimise a given optimisation criterion?

This idea is shown in Figure 1 for a simple system with one input feature, customer income, mapping onto the cost of car bought. This can be described by several cases, with adaptation between them, or by a decision surface. The surface (in this case a line) is given certain *a priori* constraints, so that it should smoothly interpolate between data points, performing an implicit generalisation between cases, averaging out the effects of noise, and robustly rejecting the outliers or inconsistent data in the training set. The surface shape can be further adapted when new cases arrive.

4.2 Local Basis Function Networks

Neural networks with local basis functions, such as Radial Basis Functions (RBF) and others (potential functions or multivariate spline bases) have been used for function approximation and modelling in various forms for many years [2][15][16][21][13][15] and are receiving a growing amount of attention from the neural network community.



$$y_j(x) = \sum_{i=0}^{N} \Phi_i (\| x - c_i \|) \cdot w_{ij}$$

Figure 2: (a) A Radial Basis Function Network.
(b) A one-dimensional mapping represented by basis functions.
(c) The mathematical description of the network.

The basic RBF net is shown in Figure 2. The output is a linear combination of many locally active non-linear *basis functions*. Each unit's centre is placed in the input space, and the *receptive field* of the unit (the volume of the input space in which its activation is non-zero) is defined by its radius. The *basis* or *activation function* (similar to the membership function of a fuzzy set) of the unit is usually designed so that the activation decreases towards zero as the input point moves away from the unit's centre, e.g. B-Splines or Gaussian bells are common choices. The units, with their respective weights can therefore be viewed as locally accurate models, whose validity for a given input is indicated by their own activation functions for this input. In CBR terminology, the centres could be thought of as prototypical situations, the solution is held in the weights between the unit and output, and the radius and shape of the basis function are used to perform adaptation together with other units.

Formulated this way, the problem of finding the correct weights is that of *Generalised Linear Least Squares* optimisation [17]. As this is a linear optimisation problem, the global minimum should always be found. The optimisation of the units' centres and radii is, however, a more difficult non-linear optimisation problem. Specht [21] used one unit for each training example, or case, each centred on an example. This is a simple technique, but one which scales up very poorly. We would therefore prefer methods which use the redundancy in any training set to reduce the number of units needed to learn the desired training data. Recent work has used variations on clustering algorithms, such as self-organising maps, or *k*-means clustering for placing the centres. The radii are then set related to the proximity of the unit's neighbours as in [13]. Another option is to dynamically add new units to the network, whenever an input occurs which is not near the centre of any of the units' receptive fields. This is easily detected, as the maximum response from any neuron is then less than the given tolerance level.

Because of the use of clustering and the interpolative nature of the network, this enables a relatively compact representation, compared to straightforward nearest-neighbour classifiers, leading to a faster response time. Neural networks with local basis functions are especially suited for combination with CBR systems, but it is perfectly feasible to use other networks, such as mulit-layer perceptrons, to learn the input-output relationship.

4.3 The continuity restriction

RBF nets are designed to learn continuous, non-linear multivariable mappings. It is therefore important that a given problem can be framed as a continuous input-output mapping if these nets are to be used. This will mean that symbolic case features *and* solution features must be mapped onto a continuous space.

Is the problem we are examining an example of a problem which can be treated in such a way? Examining some features obtained from our customer in Section 2, we see that some variables, such as *cost* (DM), could be used directly by normalising them within a given range. Others, like *performance* (fast, medium, slow) may be represented in a fuzzy form, which can be directly used by the network as described in Section 4.4. Discrete inputs, such as *number of children*, can also be easily converted. The difficulties start when a particular symbolic feature is complicated and therefore difficult to map onto a continuous variable without *a priori* knowledge. A simple example in this application is the *profession* of the customer, where classifying the similarity of the various jobs depends on sociological knowledge. To solve this requires a domain specific pre-processing for particular variables. This process is shown in Figure 3. For some types of feature, it may be advisable to break the feature down into sub-features which are easier to quantify.

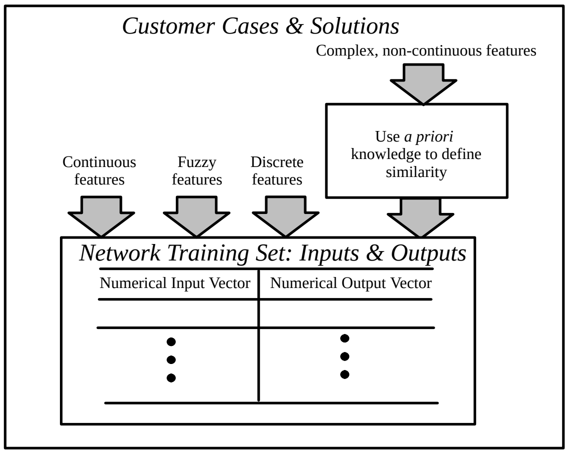


Figure 3: Creating the network's training set with pre-processing techniques from Fuzzy Logic and Case Based Reasoning.

4.4 Using fuzzy features

The functional equivalence of a class of fuzzy logic systems (those using the product operator for inference) and basis function networks has been recognised [8]. This enables the direct integration of fuzzy set descriptions of input features into basis function nets, thus utilising the *a priori* knowledge present in the choice of the membership functions. The set of basis function 'neurons' can be extended to include the fuzzy set's membership functions for a particular input dimension, by forming the tensor product ($\otimes$) of the available adaptable basis functions, with the fuzzy sets for the relevant features, as shown below.

$$act = \Phi(\| \mathrm{x}_{1..k-1} - \mathrm{c}\|) \otimes \mathrm{F}_k(x_k) \otimes, \ldots, \otimes \mathrm{F}_n(x_n)$$

where $\mathrm{F}_a(.)$ is the vector of the f_a membership function values of the relevant feature *a*, and *act* is the $f_1 \times f_2 \times ... \times f_n \times M$ matrix of activation values for the neurons (M is the number of units originally in $\Phi(\,)$). This leads to N units, where $N = M \cdot \prod_{i=1}^{k} f_i$. Treating *act* as a vector for simplicity, the network's output can be described:

$$y_j(x) = \sum_{i=0}^{N} act_i \cdot w_{ij}$$

A further interesting aspect of the similarity between fuzzy systems and basis function networks, is that the final trained network can be translated back into a collection of fuzzy rules. In complex practical cases, however, there may be so many rules that they provide little help in interpreting the system.

4.5 The system should know what it doesn't know

A major shortcoming of learning systems is that they cannot be better than their training set and that the quality of the training set is very difficult to judge. (It is often more difficult to create the representative training set than it is to learn it!) If the variance of the features is known, this measure of the input uncertainty can be used during the run-time phase to calculate the confidence limits on the outputs (also very relevant for applications where cases can be incomplete, i.e. some features are not available, and the outputs must be approximated using the other features).

This is an area where networks with local basis functions have an advantage over other learning systems, as this allows the local calculation of confidence limits for the accuracy of the network. Also, as only a limited number of training examples have significantly influenced the output of the network at any point in the input space, the user can also be warned if the given basis function contained insufficient training data to be able to learn to produce a meaningful output in the given area [12][14].

This can help the designer when testing the system, as it is easier to find gaps in the training set or areas of complex structures in the data, which are harder to learn. It is also useful when the system is in use, as it can warn the user about situations where there were very few, or possibly no previous cases.

4.6 A hybrid sales advisory system

Given the training set in numerical form, the network can then be trained to reproduce the mapping from input to output using the methods described in Section 4.2. The trained network can then be inserted into the structure shown in Figure 4.

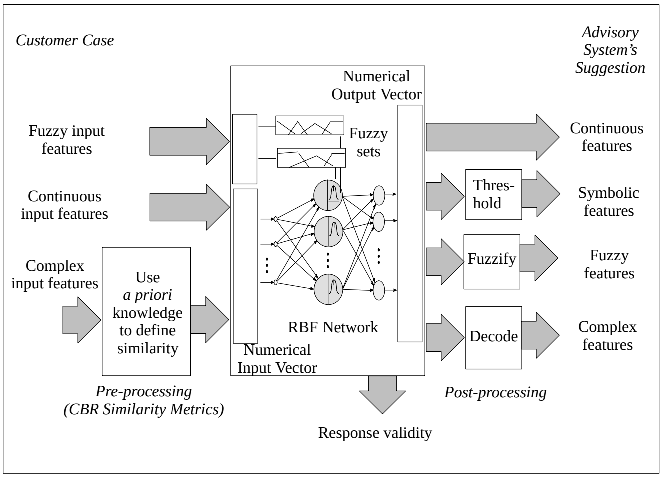


Figure 4: The Trained Hybrid System with pre-processing techniques from CBR and mapping done with RBF network.

This structure uses the same pre-processing as was used to create the training data, thus transforming the situation into a point in the continuous situation feature space. The trained network then maps this point to a point in the solution feature space. The vector of outputs describing this point must be decoded to useable symbols. This will involve rescaling continuous features, assigning membership values to various fuzzy sets, thresholding for binary or symbolic data and dedicated post-processing to decode the more complex features; in effect, the inverse of the pre-processing applied to the target outputs of the training set. The results thus describe the recommended solution for the given input situation.

Another opportunity offered by the continuous representation of the problem is that solution features which are by nature symbolic can be output as fuzzy sets, instead of just thresholding them. The tolerance bands also give a rough guide to the range of possible solutions, allowing the system to easily suggest a variety of interesting possibilities. This could provide extra information to the salesperson, predicting the likely importance of individual features to the current customer, and can be used automatically by the system to create an ordered list of possible solutions.

If the system notifies the salesperson that the current situation is a novel one, it is up to the salesperson to use his/her own initiative. It is, however, important to observe that ‘gaps’ in the training set may not always be a hindrance to the system, as it may be *vital* to know where these gaps are. If, as in this application, the training data was acquired from thousands of real sales cases, the fact that very few examples exist for a particular type of customer could possibly point to an interesting area for a marketing push. This is an example of using the trained system as a ‘market model’ for the particular product, enabling marketing people to simulate what a given type of customer is likely to buy. The advantage of using such a system is that analysis is based not on the subjective and possibly outdated opinions of a manager who may have little direct contact with customers, or a particular local market, but on a model trained with the actual up-to-date sales statistics from the area in question.

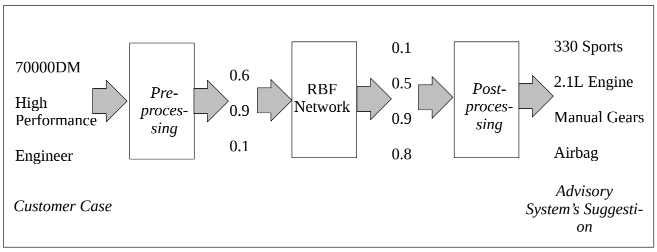


Figure 5: An illustrative example of the system in use.

5. Summary and Conclusions

Two instance-based learning paradigms were combined to create a powerful new learning system. Although there are many similarities between the two approaches, as discussed in Section 4, each has its own strengths and weaknesses. The improvements brought by the hybrid scheme are listed below, as are the advantages for the sales advisory systems domain.

5.1 Relevance to the Sales Advisory System

The CBR approach is well suited to the application, due to the ready availability of cases to train the system, the changing nature of the system (making rule-based techniques unsuitable, due to their brittleness), the explanation ability of such systems and the potential use of the trained system as a market model.

As described in Section 2, the sales advisory area is well suited to the hybrid solution suggested in this paper because of the mixture of data types (numerical, fuzzy or symbolic), large amounts of noisy or inconsistent data and relatively flexible constraints. The suggested hybrid system makes several further contributions to the advisory systems area: Improved efficiency – faster response, lower memory requirement, better prioritisation of recommendations.

5.2 Extensions to the CBR approach

For the class of problems described above, the CBR problem can be viewed as that of forming a mapping in the feature space, as is the case with classical modelling or numerical pattern recognition, allowing the introduction of many of the techniques developed over the last few decades, while retaining the symbolic processing abilities of CBR systems. The advantages gained can be summed up thus:

A memory efficient, generally applicable method can be used to form a continuous mapping to perform the indexing function rapidly. The case solution and adaptation is implicit in the result of the mapping (generalisation), and does not have to be specially considered from domain to domain. (This may lead to less demanding requirements of large numbers of examples in some applications.)

The continuous outputs of the network can be converted into a fuzzy representation of the output features, rather than directly into symbols, easing the process of prioritisation of suggestions.

The validity network concept can be used to give a local measure of accuracy and reliability of the system's answers, based on the amount of training data in the area of interest and the success of the network during the test phase in this area.

A further step could be to introduce the dynamic systems aspects used in many neural networks applications to the CBR world, enabling CBR systems to cope with time-series information.

5.3 Extensions to the RBF approach

The neural network approach can also be extended by some of the concepts in CBR: The important aspect is the use of symbolic data with neural networks, especially higher level features; the CBR community has developed many techniques for dealing with symbolic data, by defining domain specific similarity metrics for complex concepts. These can be used as pre- and post-processing techniques allowing the use of neural networks to perform continuous mappings. This is obviously useful for many applications using such a mixture of feature types.

The explanation systems component of many CBR systems is not ideal for all applications, but could certainly be useful in some: One of the major criticisms of neural networks is that they supply an answer, but are unable to justify the results. If the user asked for a justification of a result from the network, the system could choose the most similar cases from the training set using the standard nearest-neighbour retrieval procedures, to show what the exact content of the most relevant cases is. This is likely to be especially useful in mixed symbolic and statistical classification problems in which the consequences of a false classification are very serious, and where a human user wants to be able to ask the system to explain itself using the historical cases nearest to the current situation e.g. medical diagnosis applications, where symptom descriptions may be symbolic, but relevant measured data (e.g. ECGs, blood pressure time series etc.) is numerical.

All in all, the two research bodies would benefit from a higher degree of interaction, considering the large overlap in content, and similar fundamental problems. The potential mutual benefits offered by the differing experience, applications and insight should interest both research groups, not to mention the wide range of applications which can only be solved by such a combination of symbolic and numerical processing.

6. References

[1] Agnar Aamodt, *Knowledge-Intensive Case-Based Reasoning and Sustained Learning*, Proceedings ECAI'90.

[2] M. Aizerman, E. Braverman, L. Ronzonoer, *Theoretical foundations of the potential function method in pattern recognition learning*, Automatika i Telemekhanika, 25 pp 147-169, 1964.(Russ.)

[3] Bareiss, Porter, *Protos:An Exemplar-Based Learning Apprentice*, Proceedings of the Fourth International Workshop on Machine Learning (pp. 12-23) Irvine, Ca; Morgan Kaufmann, 1987.

[4] Ralph Barletta, Steve Mott, *Techniques for employing Case-Based Reasoning in Automated Customer Service Help Desks*,Proceedings of Workshop on Artificial Intelligence for Customer Service and Support, Calif. 1992.

[5] James P. Callan, Tom E. Fawcett, Edwina L. Rissland, *CABOT: An Adaptive Approach To Case-Based Search*, Proceedings IJCAI'91.

[6] K. A. De Jong, A. C. Schultz, *Using experience-based learning in game playing*, Proceedings of the Fifth International Conference on Machine Learning (pp.284–290), Ann Arbor, MI, Morgan Kaufmann, 1988.

[7] Kristian Hammond, *Learning to Anticipate and Avoid Planning Problems through the Explanation of Failures*, Proceedings of the Fifth National Conference on Artificial Intelligence (pp. 556-560); Philadelphia, PA; Morgan Kaufmann, 1986.

[8] J. S. Jang, C. T. Sun, *Functional Equivalence Between Radial Basis Function Networks and Fuzzy Inference Systems*, IEEE Trans. on Neural Networks, pp156-158, Vol. 4, No. 1, January 1993.

[9] J. Kolodner, *Maintaining Organisation in a Dynamic long-term Memory*, Cognitive Science, 7, 1983.

[10] J. Kolodner, R. Simpson jr., *A Case for Case-Based reasoning*, Proceedings of Sixth Annual Conference of the Cognitive Science Society, Hillsdale NJ, Lawrence Erlbaum Associates, 1984.

[11] Janet L. Kolodner, *Improving Human Decision Making through Case-Based Decision Aiding*, AI Magazine, Summer 1991.

[12] J. A. Leonard, M. A. Kramer, L. H. Ungar, *A Neural Network Architecture that Computes its Own Reliability*, MIT Industrial Liason Program Report, 3–7–92, Dept. of Chemical Engineering

[13] J. Moody, C. Darken, *Fast Learning in Networks of Locally Tuned Processing Units*, Neural Computation 1, p281–294, 1989.

[14] Roderick Murray-Smith, *A Fractal Radial Basis Function Neural Net for Modelling*, Inter. Conf. on Automation, Robotics and Computer Vision, Singapore, Vol 1, NW-2.6.1–NW-2.6.5, 1992.

[15] Thomas Poggio, Federico Girosi, *Networks for Approximation and Learning*, Proc. IEEE, Vol 78, No. 9, Sept. 1990.

[16] M. J. D. Powell, *Radial Basis Functions for multivariable interpolation: A review*, In: J. C. Mason, M. G. Cox, eds., *Algorithms for Approximation*, Clarendon Press, London, 1987.

[17] W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, *Numerical Recipes (C): The Art of Scientific Computing*, Chapter 14, Cambridge University Press, 1989.

[18] Michael Redmond, *Distributed Cases for Case-Based reasoning: Facilitating Use of Multiple Cases*, Proceedings AAAI'90.

[19] S. Salzberg, *Exemplar-Based Learning: Theory and Implementation* (TR–10–88), Harvard University, Center for Research in Computing Technology, Cambridge, MA.

[20] Stephen Slade, *Case-Based Reasoning: A Research Paradigm*, AI Magazine, Spring 1991.

[21] Donald F. Specht, *A General Regression Neural Network*, IEEE Trans. Neural Networks, Vol. 2, No. 6, November 1991

[22] Katia P. Sycara, *Using Case-Based Reasoning for Plan Adaptation and Repair*, In Proceedings of DARPA Workshop on Case-Based Reasoning; San Mateo, Calif.; Morgan Kaufmann, 1988.

[23] Katia P. Sycara, D. Navinchandra, *Index Transformation Techniques for Facilitating Creative use of Multiple Cases*, Proceeedings IJCAI'91.