slautern

Rissland, E. L., Basu, C., Daniels, J. L., McCarthy, J., Rubinstein, Z. B. & Skalag, D. B. (1991). A Blackboard-Based Architecture for Case-Based Reasoning: An Initial Report. In: R. Bareiss (ed.), *Proc. of the 3rd DARPA Workshop on Case-Based Reasoning*, Morgan Kaufmann, 77-92

Rissland, E. L. & Skalag, D. B. (1989). Combining Case-Based and Rule-Based Reasoning: A Heuristic Approach. *Proc. IJCAI-89*, 524-530

Rumelhart, D. E. & Zipser, D. (1985). Feature Discovery by Competitive Learning. *Cognitive Science 9*, 75-112

Schank, R. C. (1982). *Dynamic Memory: A Theory of Learning in Computers and People.* Cambridge, UK: Cambridge University Press

Tversky, A. (1977). Features of Similarity. *Psychological Review 84*, 327-352

Utgoff, P. (1988). ID5: An Incremental ID3. *Proc. of the 5th International Conference on Machine Learning, Irvine, CA: Morgan Kaufmann*

*Van de Velde, W. & Aamodt, A. (1992). Machine Learning Issues in Common*KADS. Esprit-Project P5248, Technical Report KADS-II/TII.4.3/TR/VUB/002/3.0

van Someren, M. W., Zheng, L. L. & Post, W. (1990). Cases, Models or Compiled Knowledge: a Comparative Analysis and Proposed Integration. In: B. J. Wielinga, J. Boose, B. Gaines et al. (eds.), Current Trends in Knowledge Acquisition *(Proc.* EKAW-90*). Amsterdam: IOS Press*

Wess, S. (1991). PATDEX/2 - *ein System zum adaptiven, fallfokussierenden Lernen in technischen Diagnosesituationen.* Seki-Working-Paper SWP-91-01*, University of Kaiserslautern*

Wess, S. (1993). PATDEX - *ein Ansatz zur wissensbasierten und inkrementellen Verbesserung von Ähnlichkeitsbewertungen in der fallbasierten Diagnostik. In: F. Puppe & A. Günter (eds.),* Proc. of the 2nd German Conference on Expert Systems*, Hamburg, Springer Verlag*

In: R. Bareiss (ed.), *Proc. 3rd DARPA Workshop on Case-Based Reasoning*, Morgan Kaufmann, 121-132

Jacquemain, K. J. (1988). *Effiziente Datenstrukturen und Algorithmen für mehrdimensionale Suchprobleme*. Hochschultexte Informatik (Bd. 5), Heidelberg: Hüthig Verlag

Janetzko, D. & Strube, G. (1992). Case-based Reasoning and Model-based Knowledge Acquisition. In: F. Schmalhofer, G. Strube & Th. Wetter (eds.), *Contemporary Knowledge Engineering and Cognition*, Springer Verlag

Jantke, K. P. (1992). Case-Based Reasoning in Inductive Inference. *Proc. COLT-92*

Jantke, K. P. & Lange, S. (1989). Algorithmic Learning Theory (in German: Algorithmisches Lernen). In: J. Grabowski, K. P. Jantke & H. Thiele (eds.), *Grundlagen der Künstlichen Intelligenz*, Akademie-Verlag, 246-277

Jantke, K. P., Richter, M. M., Althoff, K.-D., Lange, S. & Wess, S. (1991). IND-CBL - Vergleich ausgewählter Ansätze aus dem induktiven und dem fallbasierten Lernen. *DFG project proposal*

Kolodner, J. L. (1980). Retrieval and Organisational Strategies in Conceptual Memory: A Computer Model. *Ph.D. Thesis*, Yale University

Koopmans, L. H. (1987). *Introduction to Contemporary Statistical Methods*. Second Edition, Duxbury Press, Boston

Manago, M. & Kodratoff, Y. (1987). Model Driven Learning of Disjunctive Concepts. *Progress in Machine Learning (Proc. of the 2nd European Working Session on Learning)*, edited by Bratko & Lavrac, Sigma Press (distributed by John Wiley & Sons)

Manago, M. & Kodratoff, Y. (1990). KATE: A Piece of Computer Aided Knowledge Engineering. *Proc. of the 5th AAAI Workshop on Knowledge Acquisition for Knowledge-Based Systems*, edited by B. R. Gaines & J. Boose, Banff, Canada, AAAI Press

Manago, M., Althoff, K.-D., Auriol, E., Traphöner, R., Wess, S., Conruyt, N. & Maurer, F. (1993). Induction and Reasoning from Cases. In: Richter, Wess et al. (1993), 313-318

Michalski, R. S. (1983). Theory and Methodology of Inductive Learning. In: R. S. Michalski, J. G. Carbonell & T. M. Mitchell (eds.), *Machine Learning: An Artificial Intelligence Approach*, Tioga Publishing Co.

Morik, K. (1991). Balanced Cooperative Modelling Using Mobal - An Introduction. *Technical Report (Gmd-F3-Nachrichten AC Special Nr. 3)*, GMD, Sankt Augustin

Öchsner, H. (1992). Mehrdimensionale Zugriffspfadstrukturen für das ähnlichkeitsbasierte Retrieval von Fällen. *Diploma Thesis*, University of Kaiserslautern

Öchsner, H. & Wess, S. (1992). Ähnlichkeitsbasiertes Retrieval von Fällen durch assoziative Suche in einem mehrdimensionalen Datenraum. In: Althoff, Wess et al. (1992), 101-106

Pews, G., Weiler, F. & Wess, S. (1992). Bestimmung der Ähnlichkeit in der fallbasierten Diagnose mit simulationsfähigen Maschinenmodellen. In: Althoff, Wess et al. (1992), 47-50

Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning 1*, 81-106

Rehbold, R. (1991). Integration modellbasierten Wissens in technische Diagnostik-Expertensysteme. *Doctoral Dissertation*, University of Kaiserslautern

Richter, M. M. (1992). Classification and Learning of Similarity Measures. *Proc. of the 16th Annual Conference of the Gesellschaft für Klassifikation e.V.*, Springer Verlag

Richter, M. M. & Wess, S. (1991). Similarity, Uncertainty and Case-Based Reasoning in PATDEX. *Automated Reasoning - Essays in Honour of Woody Bledsoe*, Kluwer Academic Publishers

Richter, M. M., Wess, S., Althoff, K.-D. & Maurer, F. (eds.) (1993). *Proc. of the First European Workshop on Case-Based Reasoning*, Seki-Report SR-93-12, University of Kaiser-

8 Acknowledgement

Funding for this research has been provided by the Commission of the European Communities (Esprit contract P6322, the INRECA project). The partners of INRECA are AcknoSoft (prime contractor, France), tecInno (Germany), Irish Medical Systems (Ireland), and the University of Kaiserslautern (Germany).

9 References

Aamodt, A. (1991). A Knowledge-Intensive, Integrated Approach to Problem Solving and Sustained Learning. *Doctoral Dissertation*, University of Trondheim

Aben, M., van Someren, M. W. & Terpstra, P. (1992). Functional and Representational Integration in Knowledge Acquisition. *Proc. International Machine Learning Conference, Workshop on "Computational Architectures for Supporting Machine Learning and Knowledge Acquisition" in Aberdeen*

Althoff, K.-D. (1992). Eine fallbasierte Lernkomponente als integrierter Bestandteil der Moltke-Werkbank zur Diagnose technischer Systeme. *Doctoral Dissertation*, University of Kaiserslautern; also: Sankt Augustin (Germany): Diski 23, infix Verlag

Althoff, K.-D., Bergmann, R., Maurer, F., Wess, S., Manago, M., Auriol, E., Conruyt, N., Traphöner, R., Bräuer, M. & Dittrich, S. (1993). Integrating Inductive and Case-Based Technologies for Classification and Diagnostic Reasoning. *Proc. ECML-93 Workshop on Integrated Learning Architectures (edited by E. Plaza)*

Althoff, K.-D., Maurer, F. & Rehbold, R. (1990). Multiple Knowledge Acquisition Strategies in MOLTKE. In: B. J. Wielinga, J. Boose, B. Gaines et al. (eds.), *Current Trends in Knowledge Acquisition* (Proc. EKAW-90). Amsterdam: IOS Press, 21-40

Althoff, K.-D., Maurer, F., Traphöner & Wess, S. (1992). MOLTKE - An Integrated Workbench for Fault Diagnosis in Engineering Systems. *Proc.* EXPERSYS-92*, Paris*

Althoff, K.-D. & Wess, S. (1991a). Case-Based Knowledge Acquisition, Learning, and Problem Solving in Diagnostic Real World Tasks. *Proc.* EKAW-91*, Glasgow & Crieff*

Althoff, K.-D. & Wess, S. (1991b). Case-Based Reasoning and Expert System Development. In: F. Schmalhofer, G. Strube & T. Wetter (eds.), *Contemporary Knowledge Engineering and Cognition*, Springer Verlag

Althoff, K.-D., Wess, S., Bartsch-Spörl, B. & Janetzko, D. (eds.) (1992). Ähnlichkeit von Fällen beim fallbasierten Schliessen. *Proc. of the first Meeting of the German Special Interest Group on Case-Based Reasoning*, Seki-Working-Paper SWP-92-11, University of Kaiserslautern

Bentley, J. L. (1975). Multidimensional Search Trees Used for Associative Searching. *Communications of the ACM 18*, 509-517

Branting, L. K. & Porter, B. W. (1991). Rules and Precedents as Complementary Warrants. *Proc. AAAI-91*, 3-9

Friedman, J. H., Bentley, J. L. & Finkel, R. A. (1977). An Algorithm for Finding Best Matches in Logarithmic Expected Time. *ACM Trans. math. Software 3*, 209-226

Gentner, D. & Forbus, K. D. (1991). Mac/Fac: A Model of Similarity-Based Retrieval. *Proc. of the 13th Annual Conference of the Cognitive Science Society*, 504-509

Globig, Ch. & Wess, S. (1993). Symbolic Learning and Nearest Neighbour Classification. *Technical Report*, University of Kaiserslautern

Hinrichs, T. R. & Kolodner, J. L. (1991). The Roles of Adaptation in Case-Based Design.

generated. Such a hypothesis consists of a pair of a set of cases and an associated similarity measure. Questions to be answered are: Which cases will be entered into the case base, which will be removed from it, and how to realise the similarity measure. Since the criteria of Jantke and Lange can be analogously applied to case-based learning, it can be viewed as a special instance of inductive learning. A common theoretical framework is necessary to achieve results on this topic. First steps in this direction have been described by Jantke, Richter et al. (1991), Jantke (1992), and Globig and Wess (1993).

Both inductive learning and case-based learning have in common that they derive "global" knowledge from "local" observations (which, of course, are uncertain, respectively). However, they use different techniques to achieve this: Inductive learning bases mainly on logical concept descriptions ("logical reasoning"), whereas case-based reasoners often use analytic descriptions ("geometric reasoning") (cf., e.g., Richter, 1992). One consequence from this is that inductive learners mostly start with the "dropping of complete dimensions" in contrast to case-based reasoners which "decompose complete dimensions into intervals". It depends on the use of a learning result which particular technique is then the more successful one. Therefore, the INRECA approach integrates both learning strategies within a broader architecture for identification and diagnostic reasoning.

Up to now, much work has been done on the integration of different knowledge representation and processing schemes to improve knowledge acquisition. E.g., a comparative analysis as well as a proposed integration of models, cases and compiled knowledge have been given by van Someren, Zheng and Post (1990). The MOLTKE architecture also bases on these three schemes (cf. Althoff, Maurer & Rehbold, 1990; Althoff, Maurer et al., 1992; Althoff, 1992). The GRANUL system integrates several existing knowledge acquisition tools into one coherent system that supports several styles of knowledge acquisition (Aben, van Someren & Terpstra, 1992). The MOBAL system is an interesting example for the integration of manual and automatic knowledge acquisition methods (the balanced cooperative modelling issue, cf. Morik, 1991). Van de Velde and Aamodt (1992) have analysed the possible use of machine learning techniques within the KADS approach to expert system development. Rissland and Skalag (1989) introduced the notion of mixed paradigm reasoning for the integration of different reasoning schemes (reasoning from cases, rules, constraints, deep models etc.). Examples here are CABARET (Rissland, Basu et al., 1991), CREEK (Aamodt, 1991), PATDEX/MOLTKE (Althoff & Wess, 1991a; Richter & Wess, 1991), GREBE (Branting & Porter, 1991), and JULIA (Hinrichs & Kolodner, 1991), among others. A first suggestion for the integration of case-based reasoning and model-based knowledge acquisition is given in Janetzko and Strube (1992).

7 Conclusion

We have introduced basic parts of the architecture of the Inreca system that uses induction and case-based reasoning for solving classification tasks. INRECA is being applied to real world problems in the areas of technical maintenance as well as the pharmaceutical industry. Results from this applications might change the suggested architecture.

are no more sufficiently similar, the increase of the weights belonging to the attributes in C and U, the decrease of the weights belonging to the attributes in E, as well as the normalisation of the weights. The weights belonging to the attributes in A remain invariant. Since there is a remaining degree of freedom in the underlying equation formula, we choose the following: high weights belonging to the attributes in E are highly decreased, low weights only to a low degree. In addition, low weights belonging to the attributes in C and U are highly increased, high weights only to a low degree. Here, the goal is to "support" attributes which had only a small responsibility for the misclassification, and vice versa.

5.2 Domain Knowledge

The overall similarity assessment process can be improved by the use domain knowledge. Default values can be used to increase the number of known attribute values. Causal and heuristic determination rules can be used to derive new attribute values from known ones. Since such knowledge increases the available information, similarity is estimated on a broader basis. For the automatic generation (of a part) of that knowledge and its detailed use cf. Althoff (1992), Rehbold (1991), Althoff and Wess (1991a), and Wess (1991).

6 Discussion

The overall scenario we assumed is comparable to Gentner and Forbus' MAC/FAC model[5] (Gentner & Forbus, 1991; cf. figure 5). We used a fixed-order processing technique as the basic case retrieval mechanism which can be compared to the MAC phase. The described extensions (chapter 5) then correspond to the FAC phase.

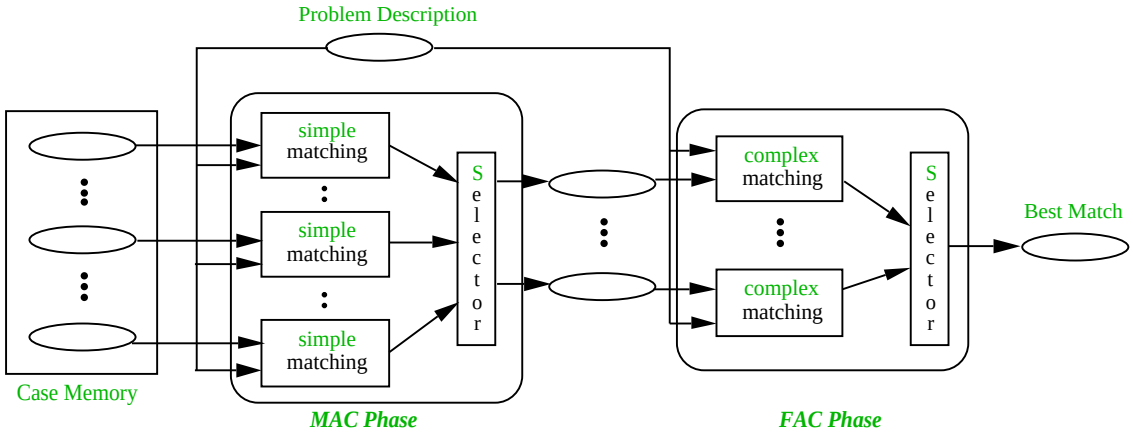

Figure 5: MAC/FAC Model

Characteristics of inductive learning have been summarised in, e.g., Jantke and Lange (1989). From this abstract point of view, case-based learning could be described as follows: From a given sequence of cases, learning hypotheses are incrementally

[5]Many Are Called but Few Are Chosen

- Attributes are selected as discriminating attributes only if the percentage of occurring unknown values is not too high. Otherwise, they are not used for indexing at all.
- Every node within the k-d tree "remembers" which diagnoses are included within the cases belonging to the respective node's subtree.
- While searching the k-d tree the tests BOB and BWB are applied using the diagnosis-dependent similarity measures of all diagnoses which occur in the respective node's subtree.

If only a few attributes can be used for indexing (because of many unknown values), it might happen that the leaf nodes contain more than (bucket size) b cases. For such cases we have, of course, linear retrieval costs. Within the buckets, the cases are sorted by their included diagnoses. Thus, the BWB test can be performed more efficiently.

In real applications, we are not always interested in the most similar case(s), only if such case(s) are sufficiently similar. This leads us to the definition of diagnosis-dependent thresholds $\sigma(D_i)$, which must be exceeded by the global similarity measure sim in order to terminate the overall classification with a certain diagnosis D_i as output: $sim(C_i, C_q) > \sigma(D^{(i)})$ if C_i sufficiently similar to C_q.

5.1 Automatic Adaptation of the Similarity Measure

Experiments in our laboratory with given case bases of correctly classified cases CB_{cor} (iteratively selecting, and temporarily removing, one case for the use as query case) showed that the similarity measure often did not classify correctly, though only one case has been removed from the case base. But, this can be improved applying an adaptive learning process. The goal is to learn new weights, i.e. new entries of the relevance matrix R. This process has an initial and a learning phase, the training set is the case base CB_{cor}.

Initial phase: the initial weights w_{ji} are determined according to the observed frequencies in the base.

Learning phase: the query cases C_q are taken from the case base CB_{cor}, i.e. every case of CB_{cor} will be a query case once. Such a query case is then temporarily removed from case base. The system determines the most similar case C_{sim}. Since the query cases are selected from CB_{cor}, it is possible to compare the respective diagnoses of C_q and C_{sim}. If $D^{(sim)} = D^{(q)}$, then nothing will be changed. For $D^{(sim)} \neq D^{(q)}$ we distinguish two possibilities:

- C_{sim} contains less known attribute values than C_q, i.e. the known values of C_{sim} are a subset of the known values of C_q. Here, the diagnosis $D^{(sim)}$ was obviously only correct by accident and C_{sim} is eliminated from the case base.
- In all other situations C_{sim} remains in the case base but the weights are updated.

The numerical form of the learning rule is not of interest here (cf. Wess, 1991). The leading principles are the achievement of $sim(C_{sim}, C_q) = \sigma(D^{(sim)})$, i.e. C_{sim} and C_q

$$\sum_{j=1}^{n} w_{ji} := \sum_{j=1}^{n} w_j(D_i) := 1$$

For every value range R_j, we introduce the distinguished value of *unknown*. During the generation of the k-d tree it has the meaning of *don't care*, during retrieval that of a missing value.

We also introduce global, i.e. diagnosis-independent, weights for special groups of attributes. Such groups are defined using the distinguished values of unknown$^{(1)}$ ... unknown$^{(k)}$ as well as the additional information whether an attribute value is a pathologic[4] one, or not. Let $C_i \in CB, C_i := (c_{i1}, c_{i2}, \ldots, c_{ik})$, be some case of the case base and C_q a query case, $C_q := (c_{q1}, c_{q2}, \ldots, c_{qk})$, where C_i includes the diagnosis $D^{(i)}$ and C_q's diagnosis is not known (per definition). We distinguish the following sets of attributes:

- $E := \{j \mid \mu_j(c_{ij}, c_{qj}) > \Omega_j\}$ *Equivalent attribute values*
- $C := \{j \mid \mu_j(c_{ij}, c_{qj}) \leq \Omega_j\}$ *Conflicting attribute values*
- $U := \{j \mid c_{qj} = unknown^{(j)}\}$ *Unknown attribute values*
- $A := \{j \mid c_{ij} = unknown^{(j)} \wedge c_{qj}$ is pathologic $\}$ *Additional attribute values*

Note, that the decision whether two values are equivalent or conflicting, i.e. belong to E or C, is made by use of the respective local similarity measure μ_j as well as a range-dependent threshold $\Omega_j \in [0, 1]$. Based on the above defined attribute sets, we introduce the following improved global similarity measure sim:

$$sim(C_i, C_q) = \frac{\alpha E}{(\alpha E + \beta C + \eta U + \gamma A)} \quad \alpha, \beta, \eta, \gamma > 0$$

where E, C, U, and A denote the following expressions:

$$\begin{aligned} E &:= \sum_{j \in E} w_j(D_i) * \mu_j(c_{ij}, c_{qj})) \\ C &:= \sum_{j \in C} w_j(D_i) * (1 - \mu_j(c_{ij}, c_{qj}))) \\ U &:= \sum_{j \in U} w_j(D_i)) \\ A &:= |A| \end{aligned}$$

Practical experience led us to the use of $\alpha = 1, \beta = 2, \gamma = 1$, and $\eta = 1/2$.

Since we have introduced the distinguished values unknown$^{(1)}$ ...unknown$^{(k)}$ as well as diagnosis-dependent similarity measures, we have to extend the k-d tree mechanism:

[4]Pathologic (or abnormal) attribute values within a query case are very important and must be explained by a similar case in the case base. Thus, they are weighted maximally (=1)

attribute) which correspond to the respective subspace. These geometric bounds are used to compute a similarity interval whose upper bound then answers the question to explore, or not. The closest point C_{min} within the actual nodes subspace is computed as the projection onto the actual nodes geometric bounds. C_{min} is on the actual nodes bounding box on the edge facing the query case C_q. If there is no overlapping in any of the k dimensions between the nodes bounding box and the k-dimensional ball round C_q then C_{min} is a corner of the bounding box. If C_q is within the bounding box then $C_q = C_{min}$ (cf. also figure 3).

Before the recursive search procedure terminates the BALL-WITHIN-BOUNDS test is applied. It is TRUE if the k-dimensional ball round C_q is completely within the bounding box of the actual tree node. If this is the case, no overlapping with other bounding boxes is possible any more. Thus, the search is finished. Two cases $C_1^{(i)}$ and $C_2^{(i)}$ per dimension $i \in \{1, \ldots, k\}$ are generated (building an interval according to the geometric bounds of the actual tree node's bounding box) to test whether the m most similar cases are all within that bounding box.

5 Extensions

The associative search mechanism, as described above, is used for the basic memorisation and retrieval task in our case-based reasoner. But, there exist a lot of real diagnostic problems which cannot be handled satisfactorily up to now (cf. Althoff & Wess, 1991a; Manago, Althoff et al., 1993; Wess, 1993). Our approach is to introduce extensions for the global similarity measure, the k-d tree representation and search, as well as the overall similarity assessment process (e.g., use of domain knowledge). Within this paper, we want to focus on the integration of an adaptive learning mechanism to automatically improve the global similarity measure. It is the second kind of improvement of our case-based reasoner using induction. Another reason is that this learning strategy is already built on top of other important extensions which then can be introduced implicitly by this procedure. The used learning strategy is similar to competitive learning (cf. Rumelhart & Zipser, 1985) and has been described in Wess (1993) and Richter (1992). Here, we concentrate on the combination of this learning strategy with the above described basic memorisation and retrieval mechanism.

We now stepwise introduce all necessary extensions. First, we improve the global similarity measure using global and local weights. The latter are defined by use of a relevance matrix R which includes a special weight for every attribute/diagnosis pair. A local weight w_{ji} denotes the relative importance (relevance) of an attribute A_j for the diagnosis D_i. Such weights effect the ball tests BOB and BWB because (only) here the global similarity measure sim is used. The consequence for the k-ball round the query case C_q is that there is a tendency to exact matching on important dimensions, and that there is an increasing degree of flexibility for less important dimensions. The relevance matrix is defined as follows:

$$\mathrm{R} = \begin{pmatrix} & D_1 & D_2 & \ldots & D_m \\ A_1 & w_{11} & w_{12} & \ldots & w_{1m} \\ A_2 & w_{21} & w_{22} & \ldots & w_{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_n & w_{n1} & w_{n2} & \ldots & .w_{nm} \end{pmatrix}$$

quartile similarity iqr is the lowest: $d := \{i \mid iqr^{(i)} \leq iqr^{(j)}\}$. This easily corresponds to the use of inter quartile distances where that attribute is selected of which the respective quartiles have the maximal distance.

Since every inner node should partition the case set into two equally-sized subsets, for every discriminating attribute d the respective median p for the value range R_d is computed: $p := median\{a_j \mid (a_1, \ldots, a_k) \in CB, j = d\}$. Then optimal k-d trees for the partitions $CB_{\leq}$ and $CB_{>}$ are generated: $CB_{\leq} := \{(a_1, \ldots, a_k) \in CB \mid a_d \leq p\}, CB_{>} := \{(a_1, \ldots, a_k) \in CB \mid a_d > p\}$.

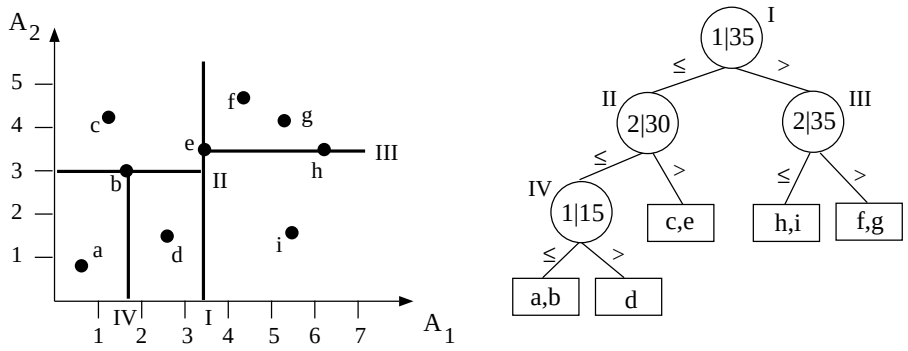

Figure 4: An Exemplary Two Dimensional Search Space and the Corresponding k-d Tree

4.2 Searching a k-d Tree

For finding the m most similar cases for a given working case (or query case)[3], we apply recursive tree search. Thus, as input we need the query case C_q, the number m of most similar cases, the k-d tree represented by its root node, and the global similarity measure $sim : [0,1]^k \rightarrow [0,1]$, and

$$sim(C_h, C_i) \rightarrow F(\mu_1(c_{h1}, c_{i1}), \mu_2(c_{h2}, c_{i2}), \ldots, \ \mu_k(c_{hk}, c_{ik})) \ \ C_h, C_i \in CB$$

One simple example for the (monotonic) function F is:

$$sim(C_h, C_i) := F(\mu_1(c_{h1}, c_{i1}), \ldots, \mu_k(c_{hk}, c_{ik})) := \frac{1}{k} \sum_{j=1}^{k} \mu_j(c_{hj}, c_{ij})$$

During search a priority queue is continuously updated which includes the m most similar cases. If the recursive search procedure examines a leaf node, the similarity of all included cases is computed and, if necessary, the priority queue is updated. If the examined node is an inner node, then the search procedure is recursively called for that son node which should include the query case. If this call terminates, it is tested whether it is also necessary to examine the other son node by using the BOUNDS-OVERLAP-BALL test. It is TRUE if the cases of the actual tree node have to be explored.

The inner nodes are correct generalisations of the all the cases they represent in that sense that they include the geometric (upper and lower) bounds (for every indexing

[3]For a query case the value of the distinguished attribute *diagnosis* is unknown

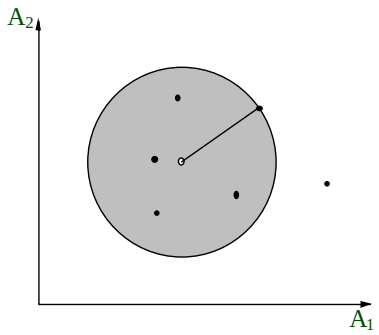

Figure 3: Bounds-Test for Nearest Neighbour Search

The average case effort (measured by the number of comparisons; cf. Jacquemain, 1988) for generating a k-d tree is $O[k * n * log_2 n]$, for the worst case $O[k * n^2]$. The average costs for retrieving the most similar case are $O[log_2 n]$, if the tree is optimally organised. For the worst case, the retrieval costs are $O[n]$. The retrieval mechanism is correct in that sense that it always finds the m most similar cases. The costs for the reorganisation of the k-d tree (making the tree an optimal one again) are $O[l * log_2 l]$, where l is the number of leaf nodes belonging to the non-balanced subtree, i.e. the costs to rebuild the whole tree are $O[n * log_2 n]$.

4 k-d Trees

We describe the basic procedures for generating and searching a k-d tree. Here, we already include some modifications, e.g. changing distance to similarity measures. This is necessary because we need the notion of similarity for the case-based reasoning component. The similarity measure is split into local measures for each value range and a global measure that is composed from the local ones. We need the local measures during the construction of the k-d tree for selecting the next discriminating attribute. The global measure is used for searching the tree. Starting from this basic retrieval mechanism, we introduce several extensions that are necessary in the context of diagnostic reasoning.

4.1 Building a k-d Tree: Basic Algorithm

For generating an optimal k-d tree, we need as input the case base $CB := \{C_i \mid C_i := (c_{i1}, c_{i2}, \ldots, c_{ik}), i \in \{1, \ldots, n\}, c_{ij} \in R_j := R(A_j)\}$, the indexing attributes $A_1 \ldots A_k$, the value ranges $R_1 \ldots R_k$, the local similarity measures $\mu_1 \ldots \mu_k$, $\mu_i : R_i \times R_i \rightarrow [0, 1]$, and the bucket size b which defines how many cases are at most allowed to be included in one leaf node. Every case includes a distinguished attribute (called *diagnosis*) which is, of course, not used for indexing.

If $|CB| \leq b$ then only one leaf node is generated and the construction process terminates. Otherwise, an inner node is generated. For every attribute $A_i, i \in \{1, \ldots, k\}$, the quartiles $q_1^{(i)}$ and $q_3^{(i)}$ of its in CB occurring values are computed. The inter quartile similarity is defined as $iqr^{(i)} := \mu_i(q_1^{(i)}, q_3^{(i)})$. As a discriminating attribute d, which is attached to the generated inner node, we select that one of which the inter

suggest another kind of integration of induction and case-based reasoning by building a case-based reasoner that uses inductive techniques to improve its performance. The improvement will be in two different ways:

- reducing the average case complexity of the case retrieval step
- correcting misclassifications of the similarity measure

The main focus will be on the first kind of improvement (chapters 3-4), the second kind will be one major aspect discussed in chapter 5. We hope that the introduction of the fixed-order processing view helps to make transparent that using an efficient information retrieval technique, namely multidimensional retrieval structures for associative search, for case retrieval is a step towards the integration of induction and case-based reasoning. We will describe the basic retrieval algorithms in the next two chapters. To overcome certain restrictions of these algorithms, especially to keep the advantages of the case-based reasoning approach, we will introduce certain extensions for these algoritms. These extensions also allow the above mentioned second kind of inductive improvement, namely the heuristic adaptation of the (global) similarity measure to avoid misclassifications.

3 Multidimensional Retrieval Structures

We developed a retrieval mechanism that is based on a k-d tree, a multi-dimensional binary search tree (Wess, 1993b; Bentley, 1975; Friedman, Bentley & Finkel, 1977). This mechanism is built on top of an object-oriented data base (Öchsner & Wess, 1992). This leads us, e.g., to the following correspondences: case = entity/object, case base = data base, problem = query, similarity-based case retrieval = best-match search. Within the k-d tree an incremental best-match search is used to find the m most similar cases (nearest neighbours) within a set of n cases with k specified indexing attributes. The search is guided by application-dependent similarity measures based on user-defined value ranges. The used similarity measures are constructed according to Tverskys contrast model (Tversky, 1977), but the user is free to define other ones. He is only restricted to use ordered value ranges as well as monotonic and symmetric similarity functions, which is not a problem for many real applications. The k-d tree uses the inhomogeneity of the search space for density-based structuring. The balanced retrieval structure results in a small number of accesses to external media.

Every node within the k-d tree represents a subset of the cases of the case base, the root node represents the whole case base. Every inner node partitions the represented case set into two disjoint subsets. The next discriminating attribute within the tree is selected based on the inter quartile distance of the attributes' value ranges (cf. Koopmans, 1987). Splitting in the median of the discriminating attribute makes the k-d tree an optimal one (the tree is optimal if all leaf nodes are at adjoining levels).

Search in the k-d tree is done via recursive tree search and the use of two test procedures: BALL-WITHIN-BOUNDS (BWB) and BOUNDS-OVERLAP-BALL (BOB) (cf. figure 3). These procedures check whether it would be reasonable to explore certain areas of the search space in more detail, or not. Such tests can be carried out without retrieving the respective cases. The geometric bounds of the considered subspaces are used to compute a "similarity interval" whose upper bound then "answers" the question to explore, or not.

its derived concept descriptions. Therefore, the generation of concept descriptions (normally) has to be carried out again. For the case-based reasoning system, the consideration of new cases (normally) is no problem, because they only have to be included into the case base. But, the underlying similarity measure is, of course, not guaranteed to classify all new cases correctly. Thus, we may have to improve the measure based on the extended case base.

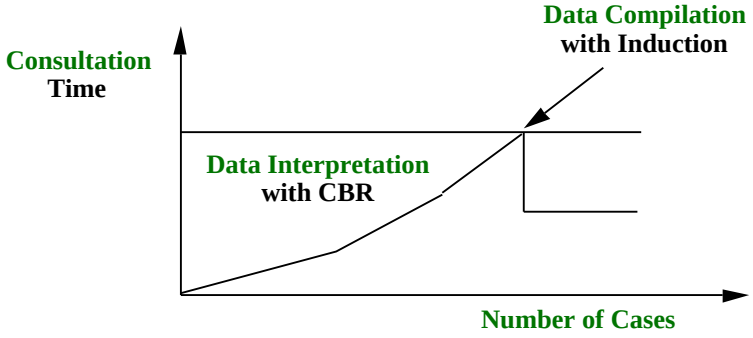

Figure 2: Combining Interpretation and Compilation

Having the above described commonalty in mind, we can use the above stated compilation-interpretation contrast to come up with an (abstract) idea for integration. This view is summarised in figure 2. Case-based reasoning is used as a flexible interactive mechanism to directly interpret the presented cases. If the number of cases strongly increases such that the time needed for consultation becomes too high, induction can be used as a compilation-like procedure which allows to reason with abstract knowledge being derived from the presented cases. If new cases are presented, case-based reasoning can be used again etc. Thus, we arrive at an interlocking of both reasoning schemes (cf. Althoff, Bergmann et al., 1993).

For the development of one single homogeneous architecture based on inductive and case-based reasoning technology that goes beyond this interlocking of the respective reasoning schemes, we generalise our view on induction. Through-out the following chapters, we consider induction as a technique which constructs abstractions from data for efficient processing. We denote the underlying construction mechanism by *fixed-order processing*, where the discovered structure within the given data corresponds to the "fixed order". With respect to the derived abstract structure, the procedure is static, i.e. inflexible in some sense. For instance, a TDIDT-like (top-down induction of decision trees; Quinlan, 1986) procedure derives a decision tree from the given cases. If certain assumptions are fulfilled[2] (no unknown or missing attribute values, no noise, no exceptions), such a decision tree enables an efficient consultation. Case-based reasoning does not apply such a kind of fixed-order processing. Therefore, its efficiency normally is worse, but it is more flexible in its reaction on data which do not meet the above mentioned requirements.

For instance, Althoff, Bergmann et al. (1993) describe the cooperation of a TDIDT-like inductive system and a case-based reasoner. The decision tree is used to preprocess the entered attribute values in a way that the number of interesting cases can be reduced. Thus, it works like a fixed indexing structure for the case retrieval where the induction and the case-based reasoning module are on the same level. We now

[2]At least to a high degree

We present the INRECA integrated learning system[1] which goes first steps into this direction. It includes inductive and case-based reasoning techniques. Currently, it is tested on two applications, name ly fault diagnosis of machine tools as well as the identification of marine sponges (cf. Manago, Althoff et al., 1993). While a more cooperative kind of integration of induction and case-based reasoning is described in Althoff, Bergmann et al. (1993), within this paper we focus on a deep integration of these technologies.

First, we motivate our approach on a more intuitive basis. Chapter 2 results in a more or less concrete guideline for integrating inductive and case-based reasoning based on mechanisms known from the field of information retrieval. We introduce multidimensional retrieval structures for associative search, especially k-d trees and describe the basic algorithms for tree construction and search. These basic data structures and algorithms are then extended to meet all the requirements of real complex diagnostic problems. Finally, we discuss our approach from several scientific points of view.

2 Inductive and Case-Based Reasoning

Case-based reasoning is a technology that allows to find analogies between a current working case and past experiences (reference cases). It makes direct use of past experiences to solve a new problem by recognising its similarity with a specific known problem and by, at least partially, applying the known solution to reach a solution for the actual new problem (cf., e.g., Kolodner, 1980; Schank, 1982; Althoff & Wess, 1991a+b).

Induction is a technology that automatically extracts knowledge from training examples (reference cases). It derives general knowledge from the cases: From an extensional description of concepts (i.e. the examples), it derives an intensional description of these concepts in the form of a decision tree, a set of most general rules (most general version of the concepts), or a characteristic description of the examples (most specific version of the concepts) (cf., e.g., Michalski, 1983; Quinlan, 1986; Manago & Kodratoff, 1987; 1990). This general knowledge is then used to solve new problems.

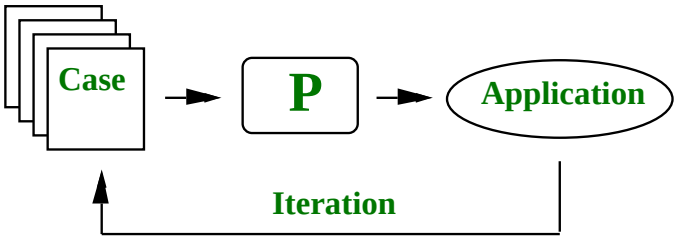

Figure 1: Inductive and Case-Based Processing of Cases

Induction and case-based reasoning both are processes that prepare cases for a certain application (cf. figure 1). If new cases occur, the inductive system has to update

[1]This description does not necessarily reflect the official opinion of the whole INRECA consortium. Ongoing applications might change this.

Induction and Case-Based Reasoning for Classification Tasks

K.-D. Althoff[1], M. Manago[2], R. Bergmann[1], F. Maurer[1], S. Wess[1], E. Auriol[2], N. Conruyt[2], R. Traphöner[3], M. Bräuer[3], S. Dittrich[3]

[1]University of Kaiserslautern, D-67653 Kaiserslautern, Germany

[2]AcknoSoft, 58a rue du Dessous des Berges, F-75013 Paris, France

[3]tecInno GmbH, Sauerwiesen 2, D-67661 Kaiserslautern, Germany

Summary: We present two techniques for reasoning from cases to solve classification tasks: Induction and case-based reasoning. We contrast the two technologies (that are often confused) and show how they complement each other. Based on this, we describe how they are integrated in one single platform for reasoning from cases: The INRECA system.

1 Introduction

Induction and case-based reasoning are two technologies for the development of experience-based expert systems that have received considerable attention during the past decade. They provide methodologies for knowledge acquisition, validation of the knowledge base, and expert system maintenance. However, a confusion is often made between induction and case-based reasoning by tool vendors or even by academic researchers: Several systems presented with the label "case-based reasoning" are simply inductive tools and, on the other hand, some incremental versions of induction tools work in a case-based reasoning fashion. We distinguish between case-based reasoning and induction by considering that the first technique makes direct use of past experiences (cases) at the problem solving stage (diagnosis) while the second one only uses an abstraction of the cases. In other words: induction *compiles* past experiences into general heuristics which are then used to solve problems. Case-based reasoning directly *interprets* past experiences (cf. also Manago, Althoff et al., 1993; Wess, 1993a; Althoff, 1992).

Many systems are often at the frontier of the two approaches. For example, ID5 (cf. Utgoff, 1988) refers back to the cases in order to incrementally modify the decision tree. The question is, however, if such a system is purely an inductive system since it remembers past cases. We prefer to clearly distinguish the two kinds of systems in order to perform a cost and merit analysis which gives clues on how to integrate the two technologies such that they can indeed benefit from each other. Note that the fundamental distinction that we make between the two kinds of systems is not so much in the underlying technology. For example, information theory as in ID3 (cf. Quinlan, 1983) might be used to implement a case-based reasoning system.

The key distinction lies in how the technology is used. We believe that the integration of induction and case-based reasoning is one key issue for improving the development of diagnostic expert systems and will expand the set of applications that can be tackled. While both technologies in their own right are able to solve special instances of diagnostic problems, the combination of these approaches may result in more than "the sum of the respective single approaches". Up to now, no satisfying systems are available that base on a really deep integration of the underlying technologies.