6

Memory Structures and Organization in Case-Based Reasoning

I. Bichindaritz

Institute of Technology, University of Washington 1900 Commerce street, 98402 Tacoma USA

Summary. Case-based reasoning (CBR) methodology stems from research on building computational memories capable of analogical reasoning, and require for that purpose specific composition and organization. This main task in CBR has triggered very significant research work and findings, which are summarized and analyzed in this article. In particular, since memory structures and organization rely on declarative knowledge and knowledge representation paradigms, a strong link is set forth in this article between CBR and data mining for the purpose of mining for memory structures and organization. Indeed the richness of data mining methods and algorithms applied to CBR memory building, as presented in this chapter, mirrors the importance of learning memory components and organization mechanisms such as indexing. The article proceeds through an analysis of this link between data mining and CBR, then through an historical perspective referring to the theory of the dynamic memory, and finally develops the two main types of learning related to CBR memories, namely mining for memory structures and mining for memory organization.

6.1 Introduction

Case-based reasoning (CBR) systems have tight connections with machine learning and data mining. They have been tagged by machine learning researchers as *lazy* learners because they defer the decision of how to generalize beyond the training set until a target new case is encountered [37], by opposition to most other learners, tagged as *eager*. Even though a large part of the inductive inferences are definitely performed at *Retrieval* time in CBR [3], mostly through sophisticated similarity evaluation, most CBR systems also perform inductive inferences at *Retain* time. There is a long tradition within this research community to study what is a memory and what its components and organization should be. Indeed CBR methodology focuses more on the memory part of its intelligent systems [51] than any other artificial intelligence (AI) methodology, and this often entails learning declarative memory structures and organization. Therefore this article proposes to focus on studying

I. Bichindaritz: *Memory Structures and Organization in Case-Based Reasoning*, Studies in Computational Intelligence (SCI) **73**, 175–194 (2008)
www.springerlink.com

CBR from the *memory* standpoint more than the inference standpoint, which opens a different, and complementary, perspective on the lazy/eager learning comparison. For CBR, this antagonism can be more adequately called the lazy/eager retrieval dilemma or pondering when it is more appropriate to learn: at *Retrieval* time or at *Retain* time [1]. The lazy learners are also the eager retrievers, and the eager learners are the lazy retrievers. Both though are full-fledged CBR systems. More precisely, the approach taken here has an analogy in the medical domain. Instead of studying the patient – CBR is here the object of our study – from the physiological standpoint, we will adopt the anatomical perspective, and attempt to answer this question: what structures and organization constitute the anatomy of a CBR system memory? We will see that a variety of data mining tasks and methods are performed in CBR, and that this richness reinforces the well known fact that CBR systems are indeed powerful data mining systems. The second section explains the relationship between CBR and data mining, and the motivation behind mining in CBR. The third section revisits data mining in early CBR systems. The fourth section concentrates on mining for memory structures, and the fifth on mining for memory organization. It is followed by a discussion on CBR and data mining, and by the conclusion.

6.2 Data Mining and Case-based Reasoning

Data mining is the analysis of observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner [22]. CBR systems are generally classified as data mining systems simply because they answer this definition. From a set of data – called cases in CBR – they perform one of the classical data mining tasks such as prediction for instance, which gives the case base a competency beyond what the data provide. In this chapter, we will focus more on another aspect, namely what data mining tasks and methods are used in CBR and what is their result in the CBR memory.

First of all, since data mining emerged in the nineties from scaling up machine learning algorithms to large datasets, let us review what machine learning authors have been saying about CBR. Machine learning authors consider case-based reasoning systems as either analogical reasoning systems [11, 15, 33, 55] or instance based learners [37]. Michalski presents the analogical inference, at the basis of case-based retrieval, as a dynamic induction performed during the matching process [33]. Mitchell refers to CBR as a kind of instance based learner [37]. This author labels these systems as *lazy* learners because they defer the decision about how to generalize beyond the training data until each new query instance is encountered. He also praises CBR systems for not committing to a global approximation once and for all during the training phase of machine learning, but for being able to generalize specifically for each target case, therefore, to fit its approximation bias, or induction bias, to the case at hand. He points here to the drawback of over-generalization that is well known for eager learners, to which instance-based learners are exempt [37].

These authors focus their analysis on the inferential aspects of learning in case-based reasoning [2, 17, 25]. Historically CBR systems have evolved from the early work of Schank in the theory of the dynamic memory [51], where this author proposes to design intelligent systems primarily by modeling their memory. Ever since Schank's precursory work on natural language understanding, one of the main goals of case-based reasoning has been to unify as much as possible memory and inferences for the performance of intelligent tasks. Therefore, focusing on studying how case-based reasoning systems learn, or mine, their memory structures and organization can prove at least as fruitful as studying and classifying them from an inference standpoint.

From a memory standpoint, learning in CBR consists in the creation, update, and organization of the structures and organization in memory. It is often referred to as case base maintenance [54, 58]. In the general cycle of CBR, learning takes place within the general reasoning cycle – see Aamodt and Plaza [1] for this classical cycle. It completely serves the reasoning, and therefore one of its characteristics is that it is an *incremental* type of mining. It is possible to fix it after a certain point, though, in certain types of applications, but it is not a tradition in CBR: *learning is an emergent behavior from normal functioning* [26]. Ideally, CBR systems start reasoning from an empty memory, and their reasoning capabilities stem from their progressive learning from the cases they process. The decision to stop learning because the system is judged competent enough is not taken from definitive criteria. It is the consequence of individual decisions made about each case, to keep it or not in memory depending upon its potential contribution to the system. Thus often the decisions about each case, each structure in memory, allow the system to evolve progressively toward states as different as ongoing learning, in novice mode, and its termination, in expert mode. If reasoning, and thus learning, are directed from the memory, learning answers to a process of prediction of the conditions of cases recall (or retrieval). As the theory of the dynamic memory showed, recall and learning are closely linked [51]. Learning in case-based reasoning answers a disposition of the system to anticipate future situations: *the memory is directed toward the future.* The anticipation deals both with avoiding situations having caused a problem, and with reinforcing the performance in success situations.

More precisely, learning in case-based reasoning takes the following forms:

1. *Adding a case to the memory*: it is at the heart of CBR systems, traditionally one of the main phases in the reasoning cycle, and the last one: *Retain* [1]. It is the most primitive learning kind, also called learning by consolidation or rote learning
2. *Explaining*: the ability of a system to find explanations for its successes and failures, and by generalization the ability to anticipate

3. *Choosing the indices*: it consists in anticipating *Retrieval*, the first reasoning step
4. *Learning memory structures*: these may be learnt by generalization from the cases or be provided from the start to hold the indices, for example. These learnt memory structures can play additional roles, such as facilitating the reuse or the retrieval
5. *Organizing the memory*: the memory comprises a network of cases, given memory structures, and learnt memory structures, organized in efficient ways. Flat and hierarchical memories have been traditionally described
6. *Refining cases*: cases may be updated, refined based upon the CBR result
7. *Refining knowledge*: the knowledge at the basis of the case-based reasoning can be refined, such as modifying the similarity measure (weight learning) or situation assessment refinement

6.3 Data Mining in Early CBR Systems

Roger Schank pioneered the methodology that would become CBR from a cognitive background [51]. Based on cognitive science research, he proposed a model of dynamic memory [51] capable of evolving the events it encounters and of learning both from successes and failures. The theory of the dynamic memory presents memory structures and organization that were later implemented in some of the first CBR systems. Their principles have been followed up to current CBR sytems.

6.3.1 The Theory of the Dynamic Memory

Memory structures are of two types, domain dependent and domain independent. The domain dependent structures are called *scripts*, defined as generalized standardized episodes. All the other structures are either organizational structures or generalized structures. The organizational structures are *generalized scenes, MOPs* (memorization organization packets), and *meta-MOPs*. Generalizations of these are, respectively, *universal scenes, u-MOPs*, and *universal u-MOPs* [51] (see Fig. 6.1). The domain independent structures are *TOPs* (thematic organization packets).

6.3.2 Generalization Based Memory

The memory of IPP (integrated partial parser) is a generalization-based memory [27,28]. IPP is a natural language understanding system working from textual information from news about international terrorism. Text understanding is presented in [28] as memory directed, but accomplishes a case-based type of search through the memory for specific events linked with the text in entry,

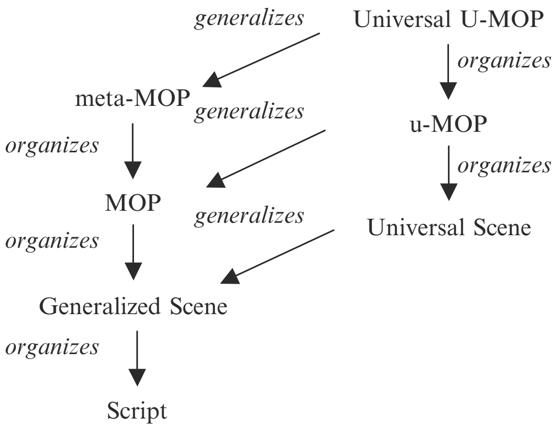

Fig. 6.1. The two dimensions of the dynamic memory and its domain dependent structures

giving it a meaning. IPP is a system implementing closely the theory of the dynamic memory.

The memory structures are on the one hand the initial structures, and on the other hand the structures learnt by the system from documents. The initial structures reproduce the stereotypical aspects of the situations: these are the S-MOPs (simple MOPs) and the AUs (action units). The S-MOPs or Simple MOPs describe abstract situations such as acts of extortion and attack. The AUs or Action Units represent concrete events such as shootings or hostage liberations. The AUs are components of the S-MOPs. The learnt structures are the spec-MOPs, containing the traits common to the structures indexed under them. These common traits are conjunctions of triplets <attribute1, attribute2, value>, such as, for example:

(TARGET NATION DOMINICAN-REPUBLIC)
(TARGET TYPE GOVERNMENT)
(LOCATION COUNTRY COLUMBIA)
(METHOD AU OCCUPATION)...

In this last example, the method used is represented by an AU: Occupation. At the highest level, the memory is a set of S-MOPs, under which are indexed some spec-MOPs forming a network. Each spec-MOP contains a discrimination network being a set of indexes to the events (AUs and role values associated with them) close to this spec-MOP. Moreover, these indexes take as values the differences between the events and the spec-MOP from which they depend.

The memory is organized as a generalization based memory, which means that the S-MOPs are the most general structures, and that the degree of generalization decreases with the deepness from the root of the structures traversed in memory following the indexes. Thus the events not having structures indexed under them are the most specific structures.

Learning in IPP follows the same mechanism as in GBM (generalization-based memory) [30], which is a concept learning system particularly interesting because it can also be linked, from its methodology, to case-based reasoning systems. Works by Lebowitz deal with hierarchical clustering as a way of implementing the theory of the dynamic memory [27]. GBM's memory is composed of GEN-NODES (for generalized nodes), also called concepts and instances. The GEN-NODES, similar to MOPs, are built by factoring common traits in the form of <attribute, value> pairs of the instances indexed underneath them. Indexing is similar to that performed in IPP. Attribute values are qualitative. Moreover a discrimination network, called D-NET, is associated to each GEN-NODE, like in IPP. The D-NET indexes the instances depending upon it by the traits differing from the norm carried by the GEN-NODE. Learning in such a system is particularly flexible. Through the bias of a counter carried by a node, each of these traits can be confirmed, by incrementation, or infirmed, by decrementation, during the search through the memory for a new instance presented to the system. When a minimum threshold is reached, a system parameter, the trait is removed from the concept, and the concept is removed when it does not comprise any trait anymore. Inversely, the system constantly searches for new concepts to create. The system memory is a dynamic memory containing both instances and concepts. It was used in two systems, UNIMEM [31] and RESEARCHER [29].

Some issues for the system have been the dependency of learnt concepts upon the order of presentation of the instances. To remedy it, Lebowitz proposed to postpone as much as possible the formation of new concepts [32]. When a new instance cannot be incorporated to a concept, because of a trait for which the counter is not high, the system prefers to wait before rejecting this concept that the concept evolution permits to definitively incorporate the instance or not.

6.4 Mining for Memory Structures

Memory structures in CBR are not only cases. A case is defined as a contextualized piece of knowledge representing an experience that teaches a lesson fundamental to achieving the goals of a reasoner [26]. For many systems, cases are represented as truthfully as possible from the application domain. Additionally, data mining methods have been applied to cases themselves, features, and generalized cases. These techniques can be applied concurrently to the same problem or selectively. If the trend is now to use them selectively, probably in the near future CBR systems will use these methods more and more concurrently.

6.4.1 Case Mining

Case mining refers to the process of mining potentially large data sets for cases [60]. Researchers have often noticed that cases simply do not exist in electronic format, that databases do not contain well-defined cases, and that the cases need to be created before CBR can be applied. Another option is to start CBR with an empty case base. When large databases are available, preprocessing these to learn cases for future CBR permits to capitalize on the experience dormant in these databases. Yang and Cheng propose to learn cases by linking several database tables [60]. *Clustering* and *support vector machines (SVM)* techniques permit to mine for cases in [60].

6.4.2 Feature Mining

Feature mining refers to the process of mining data sets for features. Many CBR systems select the features for their cases and/or generalize them. Wiratunga et al. notice that transforming textual documents into cases requires dimension reduction and/or feature selection [59], and show that this preprocessing improves the classification in terms of CBR accuracy and efficiency. These authors induce a kind of decision tree called boosted *decision stumps* because they comprise only one level, in order to select features, and *induce rules* to generalize the features. In biomedical domains, in particular when data vary continuously, the need to abstract features from streams of data is particularly prevalent. Recent, and notable, examples include Montani et al., who reduce their cases time series dimensions through *discrete Fourier transform* [39], approach adopted by other authors for time series [42]. Niloofar and Jurisica propose an original method for generalizing features. Here the generalization is an abstraction that reduces the number of features stored in a case [41]. Applied to the bioinformatics domain of micro arrays, the system uses both *clustering* techniques to group the cases into clusters containing similar cases, and *feature selection* techniques. The goal in their system is to abstract cases in a domain where there are many attributes, and few samples, where the pitfall is the famous "curse of dimensionality." The clustering method chosen is *spectral clustering* and the feature selection technique is *logistic regression*. Applying these methods to the case base improved the case-based reasoning along several dimensions, among which improved accuracy, less error, and less undecided cases (those for which there is a tie in the similarity score) [41].

6.4.3 Generalized Case Mining

Generalized case mining refers to the process of mining databases for generalized and/or abstract cases. Generalized cases are named in varied ways, such as prototypical cases, abstract cases, prototypes, stereotypes, templates, classes, categories, concepts, and scripts – to name the main ones [36]. Although all these terms refer to slightly different concepts, they represent structures that have been abstracted or generalized from real cases either by the CBR system or by an expert. When these prototypical cases are provided by a domain expert, this is a knowledge acquisition task [7, 8]. More frequently they are learnt from actual cases. In CBR, prototypical cases are often learnt to structure the memory. Therefore most of the prototypical cases presented here will also be listed in the section on structured memories.

Many authors mine for *prototypes*, and simply refer to *induction* for learning these. CHROMA [4] uses induction to learn prototypes corresponding to general cases, which each contain a pair <*situation*, *plan*>, where the situation is an object whose slots have several values possible – values are elements of a set. Bellazzi et al. organize their memory around prototypes [10]. The prototypes can either have been acquired from an expert or induced from a large case base. Schmidt and Gierl point that prototypes are an essential knowledge structure to fill the gap between general knowledge and cases in medical domains [53]. The main purpose of this prototype learning step is to guide the retrieval process and to decrease the amount of storage by erasing redundant cases. A generalization step becomes necessary to learn the knowledge contained in stored cases. They use several threshold parameters to adjust their prototypes, such as the number of cases the prototype is filled with, and the minimum frequency of each contraindication for the antibiotic therapy domain [53].

Others specifically refer to *generalization*, so that their prototypes correspond to generalized cases. An example of system inducing prototypes by generalization is a computer aided medical diagnosis system interpreting electromyography for neuropathy diagnosis [34]. The first prototypes are learnt from the expert by supervised learning, then the prototypes are automatically updated by the system by generalizing from cases. Prototypes can fusion if one is more general than the other ones, or new prototypes can be added to the memory. Malek proposes to use a *neural network* to learn the prototypes in memory for a classification task, such as diagnosis [35]. A similar connectionist approach is proposed by [50]. Portinale and Torasso [47] in ADAPTER organize their memory through E-MOPs [26] learnt by generalization from cases for diagnostic problem-solving. E-MOPs carry the common characteristics of the cases they index, in a discrimination network of features used as indices to retrieve cases. Mougouie and Bergmann [40] present a method for learning generalized cases. This method, called the Topkis-Veinott method, provides a solution to the computation of similarity for generalized cases over an n-dimensional Real values vector. Maximini et al. [36] have studied the different structures induced from cases in CBR systems. They point out that several different terms exist, such as generalized case, prototype, schema, script, and abstract case. The same terms do not always correspond to the same type of entity. They define three types of cases. A point case is what we refer to as a real case. The values of all its attributes are known. A generalized case is an arbitrary subspace of the attribute space.

There are two forms: the attribute independent generalized case, in which some attributes have been generalized (interval of values) or are unknown, and the attribute dependent generalized case, which cannot be defined from independent subsets of their attributes.

Yet other authors refer to *abstraction* for learning abstract cases. Branting proposes case abstractions for its memory of route maps [16]. The abstract cases, which also contain abstract solutions, provide an accurate index to less abstract cases and solutions. Perner [44] learns prototypes by abstracting cases as well.

Finally, many authors learn *concepts* through *conceptual clustering*. MNAOMIA [12–14] learns concepts and trends from cases through *conceptual clustering* similar to GBM [30] (see Fig. 6.2). Perner learns a hierarchy of classes by *hierarchical conceptual clustering*, where the concepts represent clusters of prototypes [44].

Dìaz-Agudo and Gonzàlez-Calero use *formal concept analysis* (FCA) – a mathematical method from data analysis – as another induction method for extracting knowledge from case bases, in the form of *concepts* [18]. The concepts learnt comprise some cases, and have both an intent– the set of attributes shared by these cases represented by a concept. Retrieval step is a classification in a concept hierarchy, as specified in the FCA methodology, which provides such algorithms. The concepts can be seen as an alternate form of indexing structure. The authors point to one notable advantage of this method, during adaptation. The FCA structure induces dependencies among the attributes that guide the adaptation process [19].

6.5 Mining for Memory Organization

Efficiency at case retrieval time is conditioned by a judicious memory organization. Two main classes of memory are presented here: unstructured – or flat – memories, and structured memories.

6.5.1 Flat Memories

Flat memories are memories in which all cases are organized at the same level. Retrieval in such memories processes all the cases in memory. Classical nearest neighbor (NN) retrieval is a method of choice for retrieval in flat memories. Flat memories can also contain prototypes, but in this case the prototypical cases do not serve as indexing structures for the cases. They can simply replace a cluster of similar cases that has been deleted from the case base during case base maintenance activity. They can also have been acquired from experts. Flat memories are the memories of predilection of NN retrieval methods [3].

Among these are so called memory-based systems, such as ANON [43]. Although capable of case-based reasoning, they have also their own characteristics. The memory of memory-based systems is completely directed by the

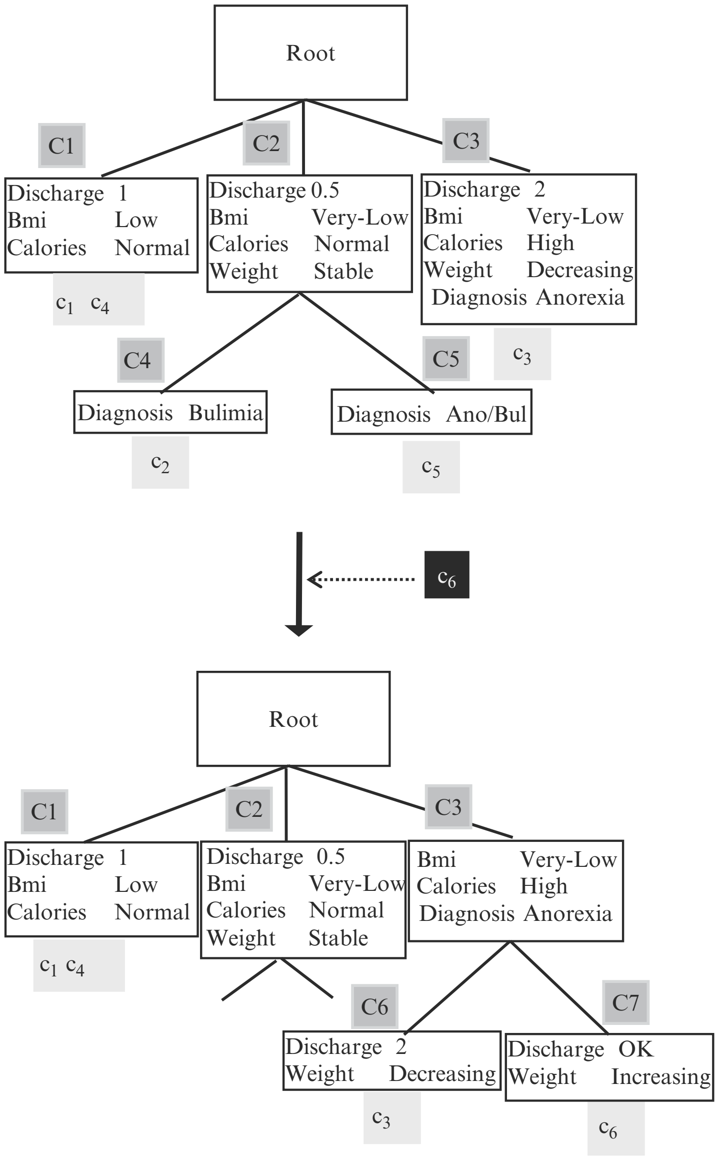

Fig. 6.2. Conceptual clustering example in MNAOMIA: a new case c6 being CBR processed creates two new concepts C6 and C7, and concept C3 is abstracted by losing two features

inferences. Implemented on parallel machines, indexation is replaced by the attribution of labels to the different cases, corresponding to the traditional indices of case-based reasoning. Feature extraction and the search through the memory correspond to the same inferences. There is generally a compromise between the importance of the inferences during the features extraction to constitute the indices, and during the search through the memory. Two approaches are possible: maximizing the inferences during the extraction of the features to constitute the indices, and maximizing the inferences during the search through the memory. The features serving as indices in both approaches are by the way different: having a semantic connotation in the former and a syntactic one in the latter. ANON proposes to integrate both by using first the features with syntactic load to perform a preselection, then the features with semantic load are extracted on the retained cases, powered by processors working in parallel. These memories are called active [43] because they must perform an important inferential effort during the search through the memory, to compensate for the absence of a learnt structuring.

6.5.2 Structured Memories

Among the different structured organizations, the accumulation of generalizations or abstractions facilitates the evaluation of the situation and allows a control of indexation.

Like GBM and related early systems, many CBR memories are organized *hierarchically* through *generalization* or *abstraction*. SWALE [24, 52] is a case-based explanation system. An explanation is a causal chain. Its cases are generalized explanations. When a new case must be integrated to the memory, it constructs a schema generalized from this case and the retrieved case used for reasoning. This constructed schema, called an explanation schema, is similar to a MOP. The authors refer to a generalization process for learning this schema. SWALE's memory is organized hierarchically, and the most abstract explanation schemas are located near the root, while the least abstract ones are positioned near the leaves, made up of the explanation cases. CANDIDE [9] is a system for language acquisition from similar cases. Based on a conceptual clustering algorithm [46], its memory is organized in categories, induced by abstraction of the common elements of two cases. Nevertheless, learning goes through three phases, the first one being totally supervised (an expert provides a set of cases and a generalization hierarchy). The second phase interacts with the expert, helping him by extracting similar memorized cases. The third phase is then an unsupervised recognition phase. This form of learning is close to knowledge acquisition. Similarly, AQUA [49] builds categories by induction. But the traits chosen for the generalization are selected in function of their pertinence. As previously, pertinence is evaluated by the construction of a causal explanation. The generalization is constrained by explanations: it is an explanation based generalization [38], and is a form of axiomatic learning. Branting's case abstractions assist case indexing, matching, and adaptation [16]. The abstract solutions contain the most important aspects of the less abstract solutions. Matching is less expensive because new cases are compared with the abstract cases, which contain many less features than the specific cases and also are less numerous. The adaptation effort is less costly too because many nonpertinent features need not be adapted. The system has been applied to route finding. This author evaluates the comparative performance of ground-level CBR, heuristic search (A*), REFINEMENT (hierarchical problem solving), and S_{CBR} (Stratified case-based reasoning) [16]. Like S_{CBR}, REFINEMENT is a form of hierarchical problem solving, in which a solution at one level of the hierarchy guides the search at a lower level in the hierarchy. The difference between REFINEMENT and S_{CBR} is that S_{CBR} starts by matching with the most specific case in the hierarchy, not systematically the most abstract ones only. The match in this particular domain is the match between a given abstract start position and a given abstract goal position. In both methods, the search starts from the root of the hierarchy and proceeds top-down. S_{CBR} proved to be an improvement over ground-level CBR and heuristic search in terms of number of levels of abstraction, the size of the case library, and the resemblance among cases. S_{CBR} was also an improvement over REFINEMENT in the same dimensions when the number of levels in the hierarchy reached at least three [16]. Some hierarchies are nonrefinable, which means that a solution may show at an abstract level, but not be valid at a more specific level. In this type of hierarchy, S_{CBR} outperforms REFINEMENT particularly in nonrefinable hierarchies.

Structured memories, dynamic, present the advantage of being declarative. The important learning efforts in declarative learning are materialized in the structures and the dynamic organization of their memories. Perner learns a hierarchy of classes by *hierarchical conceptual clustering*, where the concepts are clusters of prototypes [44]. She notes the advantages of this method: a more compact case base, and more robust (error-tolerant). The same author explains that an important aspect of case base maintenance – beyond the classic trio addition, removal and revision of cases – is learning the memory organization as well as the prototypes in memory [45]. Case-based organization is based on approximate graph subsomption. The nodes in the graph can be represented by a prototype. MNAOMIA [14] proposes to use *incremental concept learning* [20,21], which is a form of hierarchical clustering, to organize the memory. Concepts are composed of pairs of <*attribute*, *value*> common to all the cases indexed under these concepts. This system integrates highly data mining with CBR because it reuses the learnt structures to answer higher level tasks such as generating hypotheses for clinical research (see Fig. 6.3), as a side effect of CBR for clinical diagnosis and treatment decision support. Therefore this system illustrates that by learning memory structures in the form of concepts, the classical CBR classification task improves, and at the same time the system extracts what it has learnt, thus adding a knowledge discovery dimension to the classification tasks performed.

Another notable class of systems is composed of those who perform *decision tree induction* [48,56] to organize their memory. INRECA [5] project studied how to integrate CBR and decision tree induction. They propose to preprocess the case base by an induction tree, namely a decision tree. The system is based on similar approach in KATE and PATDEX from the authors.

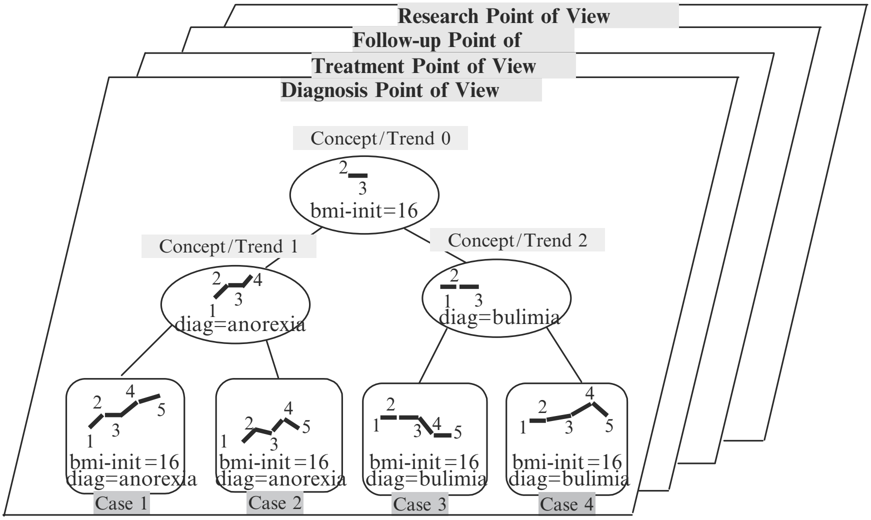

Fig. 6.3. Hierarchical memory organization in MNAOMIA: concepts are learnt during CBR for diagnosis, treatment, and/or follow-up, and can be reused by research task

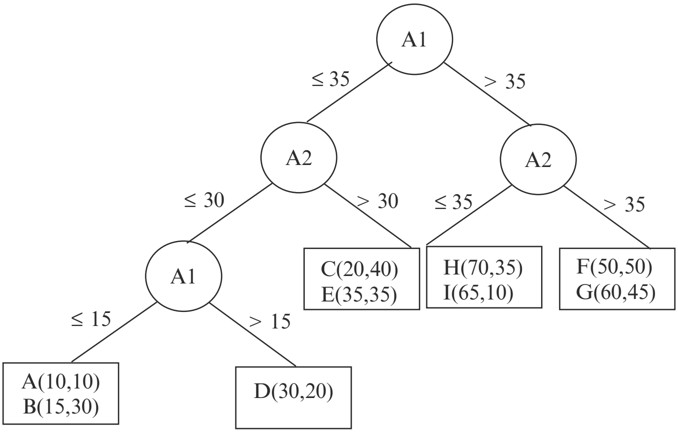

Fig. 6.4. Tree memory organization in INRECA using k-d trees

The decision tree partitions the case base around nodes composed of a single attribute and two branches per node, splitting the values on each branch in the median, based on the interquartile distance. Later refined into an *INRECA tree* [6] (see Fig. 6.4), which is a hybrid between a decision tree and a k-d tree, this method allows both similarity based retrieval and decision tree retrieval, is incremental, and speeds up the retrieval. The structures are a set of classes, each class carrying a rule to determine whether a case belongs to it or not. Each condition in the rule is sufficient and concerns a single attribute.

The similarity measure between cases takes into account the classes. Jarmulak uses a *tree induction* algorithm to induce the top level of the structure, and a *clustering* algorithm to cluster similar cases in the leaves [23]. This system is applied to imaging for the classification of ultrasonic B-scans.

Another important method is to organize the memory like a hierarchy of objects, by *subsomption*. Retrieval is then a classification in a hierarchy of objects, and functions by substitution of values in slots. CHROMA [4] uses its prototypes, induced from cases, to organize its memory. The retrieval step of CBR retrieves relevant prototypes by using subsomption in the object oriented language NOOS to find the matching prototypes. The prototypes contain a pair <*situation*, *plan*> where the situation is an object. Bellazzi et al. [10] also show a memory organization around classes of prototypes in the domain of Diabetes Mellitus. The memory organization is a tree-like structured taxonomy: each class in the hierarchy is a prototypical description of the set of problems or situations it represents. The leaves in the taxonomy are basic classes containing a single case in the case library. The authors stress that in many domains, some knowledge about the structuring of the domain is available, or can be induced. This is the case in object oriented approaches both for database management and programming languages. Every knowledge-based methodology derived from frames and semantic nets rely on that type of knowledge. In such a hierarchy, the retrieval step is two folded: first a classification in a hierarchy of objects, in this system a Bayesian classification, followed by a NN technique on the cases in the classes selected by the first step. This method is called PBR (pivoting based retrieval). An evaluation shows that PBR retrieves cases linearly with the size of the case base, in comparison with the NN technique, which grows quadratically with the number of cases [10].

Many systems use personalized memory organizations structured around several layers or *networks*. Malek and Rialle in the domain of neuropathy diagnosis construct a memory of prototypical cases that is reused in the retrieval phase. The memory structure has two levels: the upper level contains prototypes, each of them representing a group of cases; the lower level contains analyzed patient cases organized into groups of similar cases [34]. A small memory of prototypes learnt by generalization decreases the retrieval time in comparison with a large memory of cases. Malek uses a neural network to learn the prototypes in memory for a classification task, such as diagnosis [35]. Here the memory is organized in three layers: an input layer containing one unit for each attribute, a hidden layer containing the prototypes, and an output layer containing one unit for each class.

Another type of memory organization is the *formal concept lattice*. Dìaz-Agudo and Gonzàlez-Calero organize through formal concept analysis (FCA) the case base around *Galois lattices* [18]. Retrieval step is a classification in a concept hierarchy, as specified in the FCA methodology, which provides such algorithms. The concepts can be seen as an alternate form of indexing structure.

Yet other authors take advantage of the *B-tree structure* implementing databases. West and McDonald propose a method using database SQL query language to retrieve cases over a large case base stored in a database [57]. The method makes implicit use of the optimized B-tree structure underlying the relational databases implementation for fast retrieval, and computes the similarity measure during retrieval. The look up time for retrieving from a case library of any size is constant – and low – and the inclusion of the similarity assessment varies less than linearly [57].

6.6 Discussion

CBR systems make efficient use of most data mining tasks defined for descriptive modeling. We can list among the main ones encountered cluster analysis, rule induction, hierarchical cluster analysis, and decision tree induction. The motivations for performing an incremental type of data mining during CBR are several folds, and their efficiency has been measured to validate the approach. The main motivations are the following:

– Increase efficiency of retrieval mostly, but also of reuse, revise, and retain steps
– Increase robustness, tolerance to noise
– Increase accuracy of reasoning
– Improve storage needs
– Follow a cognitive model
– Add a synthetic task such as generating new research hypotheses as a side effect of normal CBR functioning

The memory organization maps directly into the retrieval method used. For example, object-oriented taxonomies will retrieve cases by subsomption mechanism and not by NN retrieval as in flat memories. Generalized cases and the like are used both as indexing structures and organizational structures. We can see here a direct mapping with the theory of the dynamic memory, which constantly influences the CBR approach. The general idea is that the learnt memory structures and organizations condition what inferences will be performed and how. This is a major difference with database approaches, which concentrate only on retrieval, and also with data mining approaches, which concentrate only on the structures learnt, and not on how they will be used. The ideal CBR memory is one which at the same time speeds up the retrieval step and improves the accuracy and robustness of the task performed by the reasoner, and particularly the reuse performed, influencing positively both the retrieval, the reuse, and the other steps. Researchers do not want to settle for a faster retrieval at the expense of less accuracy due to an overgeneralization. And they succeed at it.

6.7 Conclusion

Data mining in CBR consists mainly in incremental mining for memory structures and organization with the goal to improve performance of retrieval, reuse, revise, and retain steps. Memory structures mining comprises case mining, feature mining, and generalized case mining. CBR memories are rich in a variety of generalized structures such as concepts, prototypes, and abstract cases. These structures can be organized in flat memories or in structured memories, among hierarchies, conceptual hierarchies, decision trees, object-oriented taxonomies, formal concept lattices, and B-trees. Researchers are aiming at the ideal memory as described in the theory of the dynamic memory, which follows a cognitive model, while also improves performance and accuracy in retrieve, reuse, revise, and retain steps. Many have succeeded in showing that their memories indeed both decrease retrieval time and increase accuracy of reasoning. This demanding goal is what motivates the constant search for novel mining methods specific for CBR, and that cannot be met by methodologies that simply do not share the same goals. The variety of approaches as well as the specific and complex purpose lead to thinking that there is still space for future models and theories of CBR memories.

References

1. Aamodt A, Plaza E (1994) Case-Based Reasoning: Foundational Issues, Methodologies Variations, and Systems Approaches. AI Communications, IOS Press, Vol. 7: 1:39–59
2. Aha DW (1991) Case-Based Learning Algorithms. In: Bareiss R (ed) Proceedings of a Workshop on case-based reasoning (DARPA), Washington, D.C. Morgan Kaufmann, San Mateo, California, pp 147–157
3. Aha DW (1997) Lazy Learning. Artificial Intelligence Review 11:7–10
4. Armengol E, Plaza E (1994) Integrating induction in a case-based reasoner. In: Keane M, Haton JP, Manago M (eds) Proceedings of EWCBR 94. Acknosoft Press, Paris, pp 243–251
5. Auriol E, Manago M, Althoff KD, Wess S, Dittrich S (1994) Integrating Induction and Case-Based Reasoning: Methodological Approach and First Evaluations. In: Keane M, Haton JP, Manago M (eds) Proceedings of EWCBR 94. Acknosoft Press, Paris, pp 145–155
6. Auriol E, Wess S, Manago M, Althoff KD, Traphöner R (1995) INRECA: A Seamless Integrated System Based on Inductive Inference and Case-Based Reasoning. In: Veloso M, Aamodt A (eds) Proceedings of ICCBR 95. Springer-Verlag, Lecture Notes in Artificial Intelligence 1010, Berlin, Heidelberg, New York, pp 371–380
7. Bareiss R (1989a) Exemplar-Based Knowledge Acquisition. Academic Press, Inc., San Diego, CA
8. Bareiss R (1989b) The Experimental Evaluation of a Case-Based Learning Apprentice. In: Hammond KJ (ed) Proceedings of a Workshop on case-based reasoning (DARPA). Morgan Kaufmann, San Mateo, CA, pp 162–166

9. Beck HW (1991) Language Acquisition from Cases. In: Bareiss R (ed) Proceedings of a Workshop on case-based reasoning (DARPA), Washington, D.C. Morgan Kaufmann, San Mateo, California, pp 159–169
10. Bellazzi R, Montani S, Portinale L (1998) Retrieval in a Prototype-Based Case Library: A Case Study in Diabetes Therapy Revision. In: Smyth B, Cunningham P (eds) Proceedings of ECCBR 98. Springer-Verlag, Lecture Notes in Artificial Intelligence 1488, Berlin, Heidelberg, New York, pp 64–75
11. Bichindaritz I (1990) Alexia: un système de résolution de problèmes par analogie utilisant une mémoire des exemples méta-indexée par un modèle causal appliqué à la détermination de l'étiologie de l'H.T.A. Rapport de stage de DEA, EHEI, Université René Descartes-Paris V
12. Bichindaritz I (1994) A case-based assistant for clinical psychiatry expertise. In: Proceedings 18th Symposium on Computer Applications in Medical Care. AMIA, Washington DC, pp 673–677
13. Bichindaritz I (1995a) Case-Based Reasoning and Conceptual Clustering: For a Co-operative Approach. In: Watson I, Fahrir M (eds) Advances in Case-Based Reasoning. Springer-Verlag, Lecture Notes in Artificial Intelligence 1020, Berlin, Heidelberg, New York, pp 91–106
14. Bichindaritz I (1995b) A case-based reasoner adaptive to several cognitive tasks. In: Veloso M., Aamodt A (eds) Proceedings of ICCBR 95. Springer-Verlag, Lecture Notes in Artificial Intelligence 1010, Berlin, Heidelberg, New York, pp 391–400
15. Bichindaritz I, Séroussi B (1992) Contraindre l'analogie par la causalité. Technique et sciences informatiques Volume 11, n 4: 69–98
16. Branting KL (1997) Stratified Case-Based Reasoning in Non-Refinable Abstraction Hierarchies. In: Leake D, Plaza E (eds) Proceedings of ICCBR 97. Springer-Verlag, Lecture Notes in Artificial Intelligence 1266, Berlin, Heidelberg, New York, pp 519–530
17. Cain T, Pazzani MJ, Silverstein G (1991) Using Domain Knowledge to Influence Similarity Judgment. In: Bareiss R (ed) Proceedings of a Workshop on case-based reasoning (DARPA), Washington, D.C. Morgan Kaufmann, San Mateo, California, pp 191–202
18. Dìaz-Agudo B, Gonzàlez-Calero P (2001) Classification Based Retrieval Using Formal Concept Analysis. In: Aha DW, Watson I (eds) Proceedings of ICCBR 01. Springer-Verlag, Lecture Notes in Artificial Intelligence 2080, Berlin, Heidelberg, New York, pp 173–188
19. Dìaz-Agudo B, Gervàz P, Gonzàlez-Calero P (2003) Adaptation Guided Retrieval Based on Formal Concept Analysis. In: Ashley K, Bridge DG (eds) Proceedings of ICCBR 03. Springer-Verlag, Lecture Notes in Artificial Intelligence 2689, Berlin, Heidelberg, New York, pp 131–145
20. Fisher D (1987) Knowledge acquisition via incremental conceptual clustering. Machine Learning 2:139–172
21. Gennari JH, Langley P, Fisher D (1989) Models of Incremental Concept Formation. Artificial Intelligence 40:11–61
22. Hand D, Mannila H, Smyth P (2001) Principles of Data Mining. The MIT Press, Cambridge, Massachusetts
23. Jarmulak J (1998) Case-Based Classification of Ultrasonic B-Scans: Case-Base Organization and Case Retrieval. In: Smyth B, Cunningham P (eds) Proceedings of ECCBR 98. Springer-Verlag, Lecture Notes in Artificial Intelligence 1488, Berlin, Heidelberg, New York, pp 100–111

24. Kass A, Leake D, Owens C (1986) SWALE, A Program that Explains. In: Schank RC (ed) Explanation Patterns. Understanding Mechanically and Creatively. Laurence Erlbaum Associates, Publishers, Hillsdale, New Jersey, pp 232–256
25. King J, Bareiss R (1989) Similarity Assessment and Case-Based Reasoning. In: Hammond KJ (ed) Proceedings of a Workshop on case-based reasoning (DARPA), Pensacola Beach, Florida. Morgan Kaufmann, San Mateo, CA, pp 67–71
26. Kolodner JL (1993) Case-Based Reasoning. Morgan Kaufmann Publishers, San Mateo, California
27. Lebowitz M (1982) IPP. In: Schank RC (ed) Dynamic memory: A theory of reminding and learning in computers and people. Cambridge University Press, Cambridge, MA, pp 197–207
28. Lebowitz M (1983) Generalization From Natural Language Text. Cognitive Science 7:1–40
29. Lebowitz M (1985) Researcher: An Experimental Intelligent Information System. In: Proceedings IJCAI 85, pp 858–862
30. Lebowitz M (1986) Concept Learning in a Rich Input Domain: Generalization-Based Memory. In: Michalski RS, Carbonell JG, Mitchell TM (eds) Machine Learning: An Artificial Intelligence Approach, Vol 2. Morgan Kaufmann, Los Altos, CA
31. Lebowitz M (1987) Experiments with Incremental Concept Formation: UNIMEM. Machine Learning 2:103–138
32. Lebowitz M (1988) Deferred Commitment in UNIMEM: Waiting to learn. In: Proceedings 5th Machine Learning Conference, Ann Arbor, Michigan. pp 80–86
33. Michalski RS (1993) Toward a Unified Theory of Learning. In: Buchanan BG, Wilkins DC (eds) Readings in knowledge acquisition and learning, automating the construction and improvement of expert systems. Morgan Kaufmann Publishers, San Mateo, California, pp 7–38
34. Malek M, Rialle V (1994) A Case-Based Reasoning System Applied to Neuropathy Diagnosis. In: (Keane M, HAton JP, Manago M (eds) Proceedings of EWCBR 94. Acknosoft Press, Paris, pp 329–336
35. Malek M (1995) A Connectionist Indexing Approach for CBR Systems. In: Veloso M, Aamodt A (eds) Proceedings of ICCBR 95. Springer-Verlag, Lecture Notes in Artificial Intelligence 1010, Berlin, Heidelberg, New York, pp 520–527
36. Maximini K, Maximini R, Bergmann R (2003) An Investigation of Generalized Cases. In: Ashley KD, Bridge DG (eds) Proceedings of ICCBR 03. Springer-Verlag, Lecture Notes in Artificial Intelligence 2689, Berlin, Heidelberg, New York, pp 261–275
37. Mitchell TM (1997) Machine Learning. Mc Graw Hill, Boston, Massachusetts
38. Mitchell TM, Keller RM, Kedar-Cabelli S (1986) Explanation-based generalization: A unifying view. Machine Learning 1(1): 47–80
39. Montani S, Portinale L, Bellazzi R, Leornardi G (2004) RHENE: A Case Retrieval System for Hemodialysis Cases with Dynamically Monitored Parameters. In: Funk P, Gonzàlez Calero P (eds) Proceedings of ECCBR 04. Springer-Verlag, Lecture Notes in Artificial Intelligence 3155, Berlin, Heidelberg, New York, pp 659–672

40. Mougouie B, Bergmann R (2002) Similarity Assessment for Generalized Cases by Optimization Methods. In: Craw S, Preece A (eds) Proceedings of EWCBR 02. Springer-Verlag, Lecture Notes in Artificial Intelligence 2416, Berlin, Heidelberg, New York, pp 249–263
41. Niloofar A, Jurisica I (2004) Maintaining Case-Based Reasoning Systems: A Machine Learning Approach. In: Funk P, Gonzàlez Calero P (eds) Proceedings of ECCBR 04. Springer-Verlag, Lecture Notes in Artificial Intelligence 3155, Berlin, Heidelberg, New York, pp 17–31
42. Nilsson M, Funk P (2004) A Case-Based Classification of Respiratory sinus Arrhythmia. In: Funk P, Gonzàlez Calero P (eds) Proceedings of ECCBR 04. Springer-Verlag, Lecture Notes in Artificial Intelligence 3155, Berlin, Heidelberg, New York, pp 673–685
43. Owens C (1993) Integrating Feature Extraction and Memory Search. In: Kolodner JL (ed) Case-Based Learning. Kluwer Academic Publishers, Boston, pp 117–145
44. Perner P (1998) Different Learning Strategies in a Case-Based Reasoning System for Image Interpretation. In: Smyth B, Cunningham P (eds) Proceedings of ECCBR 98. Springer-Verlag, Lecture Notes in Artificial Intelligence 1488, Berlin, Heidelberg, New York, pp 251–261
45. Perner P (2003) Incremenetal Learning of Retrieval Knowledge in a Case-Based Reasoning System. In: Ashley KD, Bridge DG (eds) Proceedings of ICCBR 03. Springer-Verlag, Lecture Notes in Artificial Intelligence 2689, Berlin, Heidelberg, New York, pp 422–436
46. BW, Bareiss R, Holte RC (1990) Concept Learning and Heuristic Classification in Weak-Theory Domains. Artificial Intelligence, 45:229–263
47. Portinale L, Torasso P (1995) ADAPTER: An Integrated Diagnostic System Combining Case-Based and Abductive Reasoning. In: Veloso M, Aamodt A (eds) Proceedings of ICCBR 95. Springer-Verlag, Lecture Notes in Artificial Intelligence 1010, Berlin, Heidelberg, New York, pp 277–288
48. Quinlan JR (1986) Induction of decision trees. Machine Learning 1(1):81–106
49. Ram A (1993) Indexing, Elaboration and Refinement: Incremental Learning of Explanatory Cases. In: Kolodner JL (ed) Case-Based Learning. Kluwer Academic Publishers, Boston, pp 7–54
50. Reategui E, Campbell JA, Borghetti S (1995) Using a Neural Network to Learn General Knowledge in a Case-Based System. In: Veloso M, Aamodt A (eds) Proceedings of ICCBR 95. Springer-Verlag, Lecture Notes in Artificial Intelligence 1010, Berlin, Heidelberg, New York, pp 528–537
51. Schank RC (1982) Dynamic memory. A theory of reminding and learning in computers and people. Camdridge University Press, Cambridge
52. Schank RC, Leake DB (1989) Creativity and Learning in a Case-Based Explainer. Articifial Intelligence 40:353–385
53. Schmidt R, Gierl L (1998) Experiences with Prototype Designs and Retrieval Methods in Medical Case-Based Reasoning Systems. In: Smyth B, Cunningham P (eds) Proceedings of ECCBR 98. Springer-Verlag, Lecture Notes in Artificial Intelligence 1488, Berlin, Heidelberg, New York, pp 370–381
54. Smyth B, McKenna E (1999) Building Compact Competent Case-Bases. In: Proceedings of ICCBR 99. Springer-Verlag, Berlin, Heidelberg, New York, pp 329–342

55. Veloso MM, Carbonell JG (1993) Derivational Analogy in PRODIGY: Automating Case Acquisition, Storage and Utilization. In: Kolodner JL (ed) Case-Based Learning. Kluwer Academic Publishers, Norwell, MA, pp 55–84
56. Utgoff PE (1989) Incremental induction of decision trees. Machine Learning, 4, 2:161–186
57. West GM, McDonald JR (2003) An SQL-Based Approach to Similarity Assessment within a Relational Database. In: Ashley K, Bridge DG (eds) Proceedings of ICCBR 03. Springer-Verlag, Lecture Notes in Artificial Intelligence 2689, Berlin, Heidelberg, New York, pp 610–621
58. Wilson DC, Leake DB (2001) Mainting Case-based Reasoners: Dimensions and Directions. Computational Intelligence Journal, Vo. 17, No. 2:196–213
59. Wiratunga N, Koychev I, Massie S (2004) Feature Selection and Generalisation for Retrieval of Textual Cases. In: Funk P, Gonzàlez Calero P (eds) Proceedings of ECCBR 04. Springer-Verlag, Lecture Notes in Artificial Intelligence 3155, Berlin, Heidelberg, New York, pp 806–820
60. Yang Q, Cheng H (2003) Case Mining from Large Databases. In: Ashley K, Bridge DG (eds) Proceedings of ICCBR 03. Springer-Verlag, Lecture Notes in Artificial Intelligence 2689, Berlin, Heidelberg, New York, pp 691–702