Case-Based Reasoning and the Statistical Challenges II

Petra Perner

Abstract. Case-based reasoning (CBR) solves problems using the already stored knowledge, and captures new knowledge, making it immediately available for solving the next problem. Therefore, CBR can be seen as a method for problem solving, and also as a method to capture new experience and make it immediately available for problem solving. The CBR paradigm has been originally introduced by the cognitive science community. The CBR community aims to develop computer models that follow this cognitive process. Up to now many successful computer systems have been established on the CBR paradigm for a wide range of real-world problems. We will review in this paper the CBR process and the main topics within the CBR work. Hereby we try bridging between the concepts developed within the CBR community and the statistics community. The CBR topics we describe are: similarity, memory organization, CBR learning, and case-base maintenance. The incremental aspect arising with the CBR paradigm will be considered as well as the life-time aspect of a CBR system. We will point out open problems within CBR that need to be solved. Finally we show on application how the CBR paradigm can be applied. The applications we are focusing on are meta-learning for parameter selection in technical systems, image interpretation, incremental prototype-based classification and novelty detection and handling. Finally, we summarize our concept on CBR.

Keywords: case-based reasoning, statistics, similarity, incremental learning, normalization, case base organization, novelty detection.

1 Introduction

CBR [3] solves problems using the already stored knowledge, and captures new knowledge, making it immediately available for solving the next problem.

Petra Perner
Institute of Computer Vision and Applied Computer Sciences, IBaI,
Leipzig, Germany
e-mail: pperner@ibai-institut.de,
www.ibai-institut.de

Therefore, CBR can be seen as a method for problem solving, and also as a method to capture new experience and make it immediately available for problem solving. It can be seen as an incremental learning and knowledge-discovery approach, since it can capture from new experience general knowledge, such as case classes, prototypes and higher-level concepts.

The CBR paradigm has originally been introduced by the cognitive science community. The CBR community aims at developing computer models that follow this cognitive process. For many application areas computer models have successfully been developed based on CBR, such as signal/image processing and interpretation tasks, help-desk applications, medical applications and E-commerce-product selling systems.

In this paper we will explain the CBR process scheme in Section 2. We will show what kinds of methods are necessary to provide all the necessary functions for such a computer model. The CBR process model comprised of the CBR reasoning and CBR maintenance process is given in Section 3. Then we will focus on similarity in Section 4. Memory organization in a CBR system will be described in Section 5. Both similarity and memory organization are concerned in learning in a CBR system. Therefore, in each section an introduction will be given as to what kind of learning can be performed. In Section 6 we will describe open topics in CBR research for incremental learning and Section 7 for the life-cycle of a CBR system. In Section 8 we describe the newly established application fields on multimedia. Then, we focus our description on meta-learning for parameter selection, image interpretation, incremental prototype-based classification and novelty detection and handling. In Section 8.1 we will describe meta-learning for parameter selection for data processing systems. CBR based image interpretation will be described in Section 8.2 and incremental prototype-based classification in Section 8.3. New concepts on novelty detection and handling will be presented in Section 8.4.

While reviewing the CBR work, we will try bridging between the concepts developed within the CBR community and the concepts developed in the statistics community.

In the conclusion, we will summarize our concept on CBR in Section 9.

2   Case-Based Reasoning

CBR is used when generalized knowledge is lacking. The method works on a set of cases formerly processed and stored in a case base. A new case is interpreted by searching for similar cases in the case base. Among this set of similar cases the closest case with its associated result is selected and presented to the output.

The differences between a CBR learning system and a symbolic learning system, which represents a learned concept explicitly, e.g. by formulas, rules or decision trees, is that a CBR system describes a concept C implicitly by a pair (CB, sim). The relationship between the case base CB and the measure sim used for classification may be characterized by the equation:

$$Concept = Case\_Base + Measure\_o\_Similarity \tag{1}$$

This equation indicates in analogy to arithmetic that it is possible to represent a given concept C in multiple ways, i.e. there exist many pairs $C = (CB_1, sim_1)$, (CB_2, sim_2), ..., (CB_i, sim_i) for the same concept C.

During the learning phase a case-based system gets a sequence of cases $X_1, X_2, ..., X_i$ with $X_i = (x_i, class(x_i))$ and builds a sequence of pairs $(CB_1, sim_1), (CB_2, sim_2), ..., (CB_i, sim_i)$ with $CB_i \subseteq \{X_1, X_2, ..., X_i\}$. The aim is to get in the limit a pair (CB_n, sim_n) that needs no further change, i.e. $\exists n \; \forall m \geq n (CB_n, sim_n) = (CB_m, sim_m)$, because it is a correct classifier for the target concept C [66].

Formal, we like to understand a case as the following:

Definition 1. A case X is a triple (P, E, L) with a problem description P, an explanation of the solution E and a problem solution L.

The problem description summarizes the information about a case in the form of attributes or features. Other case representations such as graphs, images or sequences may also be possible. The case description is given a-priori or needs to be elicitated during a knowledge acquisition process. Only the most predictive attributes will guarantee us to find exactly the most similar cases.

Equation 1 and definition 1 give a hint as to how a case-based learning system can improve its classification ability. The learning performance of a CBR system is of incremental manner and it can also be considered as on-line learning. In general, there are several possibilities to improve the performance of a case-based system. The system can change the vocabulary V (attributes, features), store new cases in the case base CB, change the measure of similarity sim, or change V, CB and sim in combinatorial manner.

That brings us to the notion of knowledge containers introduced by Richter [51]. According to Richter, the four knowledge containers are the underlying vocabulary (or features), the similarity measure, the solution transformation, and the cases. The first three represent compiled knowledge, since this knowledge is more stable. The cases are interpreted knowledge. As a consequence, newly added cases can be used directly. This enables a CBR system to deal with dynamic knowledge. In addition, knowledge can be shifted from one container to another container. For instance, in the beginning a simple vocabulary, a rough similarity measure, and no knowledge on solution transformation are used. However, a large number of cases are collected. Over time, the vocabulary can be refined and the similarity measure defined in higher accordance with the underlying domain. In addition, it may be possible to reduce the number of cases, because the improved knowledge within the other containers now enables the CBR system to better differentiate between the available cases.

The abstraction of cases into a more general case (concepts, prototypes and case classes) or the learning of the higher-order relation between different cases may reduce the size of the case base and speed up the retrieval phase of the system [57]. It can make the system more robust against noise. More abstract cases which are set

in relation to each other will give the domain expert a better understanding about his domain. Therefore, beside the incremental improvement of the system performance through learning, CBR can also be seen as a knowledge-acquisition method that can help to get a better understanding about the domain [9, 14].

The main problems with the development of a CBR system are the following: What makes up a case?, What is an appropriate similarity measure for the problem?, How to organize a large number of cases for efficient retrieval?, How to acquire and refine a new case for entry in the case base?, How to generalize specific cases to a case that is applicable to a wide range of situations?

3   Case-Based Reasoning Process Model

The CBR reasoning process is comprised of seven phases (see Fig. 1): Current problem description, problem indexing, retrieval of similar cases, evaluation of candidate cases, modification of selected cases, application to a current problem, and critique of the system.
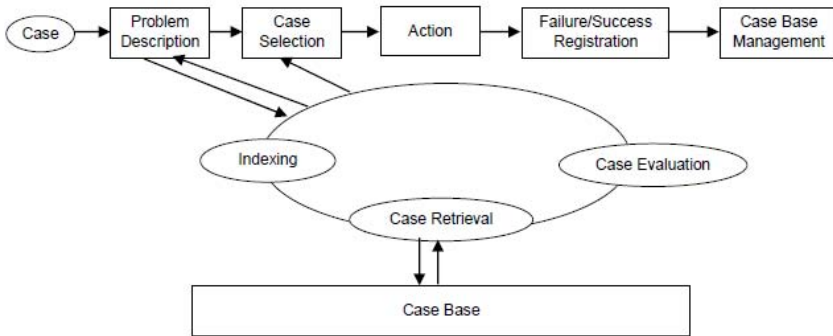

Fig. 1 Case-Based Reasoning Process Model

The current problem is described by some keywords, attributes, features or any abstraction that allows describing the basic properties of a case. Based on this description indexing of case base is done. Among a set of similar cases retrieved from the case base the closest case is evaluated as a candidate case. If necessary this case is modified so that it fits the current problem. The problem solution associated to the current case is applied to the current problem and the result is observed by the user. If the user is not satisfied with the result or if no similar case could be found in the case base, the user or the process itself gives feedback to the system. This critique is used to incrementally improve the system performance by the case-base management process.

The CBR management (see Fig. 2) will operate on new cases as well as on cases already stored in the case base. If a new case has to be stored into the case base, this means there is no similar case in case base. The system has recognized a gap in

the case base. A new case has to be inputted into the case base in order to close this gap. From the new case a predetermined case description has to be extracted which should be formatted into the predefined case format. After that the case is stored into the case base. Selective case registration means that no redundant cases will be stored into the case base and similar cases will be grouped together or generalized by a case that applies to a wider range of problems. Generalization and selective case registration ensure that the case base will not grow too large and that the system can find similar cases fast.

It might also happen that too many non-relevant cases will be retrieved during the CBR reasoning process. Therefore, it might be wise to rethink the case description or to adapt the similarity measure. For the case description more distinguishing attributes should be found that allow separating cases that do not apply to the current
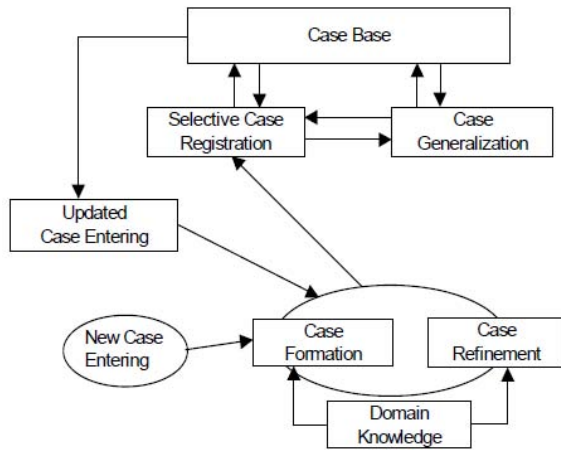

Fig. 2 Case Base Maintenance Process

problem. The weights in the similarity measure might be updated in order to retrieve only a small set of relevant cases.

CBR maintenance is a complex process and works over all knowledge containers (vocabulary, similarity, retrieval, case base) of a CBR system. Consequently, architectures and systems have been developed which support this process [26, 41] and also look into the life-time aspect concerned with case-based maintenance [34].

4   Similarity

Although similarity is a concept humans prefer to use when reasoning over problems, they usually do not have a good understanding of how similarity is formally expressed. Similarity seems to be a very incoherent concept.

From the cognitive point of view, similarity can be viewed from different perspectives [58]. A red bicycle and a blue bicycle might be similar in terms of the concept "bicycle", but both bicycles are dissimilar when looking at the colour. It is important to know what kind of similarity is to be considered when reasoning over two objects. Overall similarity, identity, similarity, and partial similarity need to be modelled by the right flexible control strategy in an intelligent reasoning system. It is especially important in image data bases where the image content can be viewed from different perspectives. Image data bases need to have this flexibility and computerized conversational strategies to figure out from what perspective the problem is looked at and what kind of similarity has to be applied to achieve the desired goal. From the mathematical point of view, the Minkowski metric is the most used similarity measure for technical problems:

$$d_{ii^{\cdot}}^{(p)} = \left[\sum_{j=1}^{J} \left| x_{ij} - x_{i^{\cdot}j} \right|^{p} \right]^{1/p} \tag{2}$$

the choice of the parameter p depends on the importance we give to the differences in the summation. Metrical properties such as symmetry, identity and unequality hold for the Minkowski metric.

If we use the Minkowski metric for calculating the difference between two trajectories of a robot axis, one is the original trajectory and the other one is a reconstructed trajectory obtained by a compression algorithm from the compressed data points stored in the memory of the robot control system [21], it might not be preferable to choose $p = 2$ (Euclidean metric), since the measure averages over all data points, but gives more emphasis to big differences. If choosing $p = 1$ (City-Block metric), big and small differences have the same influence (impact) on the similarity measure. In case of the Max-Norm ($p = \infty$) none of the data point differences should exceed a predefined difference. In practice it would mean that the robot axis is performing a smooth movement over the path with a known deviation from the real path and will never come in the worse situation to perform a ramp-like function. In the robot example the domain itself gives us an understanding about the appropriate similarity metric.

Unfortunately, for most of the applications we do not have any a-priori knowledge about the appropriate similarity measure. The method of choice for the selection of the similarity measure is to try different types of similarity and observe their behaviour based on quality criteria while applying them to a particular problem. The error rate is the quality criterion that allows selecting the right similarity measure for classification problems. Otherwise it is possible to measure how well similar objects are grouped together, based on the chosen similarity measure, and at the same time, how well different groups can be distinguished from each other. This changes the problem into a categorization problem for which proper category measures are known from clustering [27] and machine learning [20].

In general, distance measures can be classified based on the data-type dimension. There are measures for numerical data, symbolical data, structural data and mixed-data types. Most of the overviews given for similarity measures in various works are based on this view [36, 40, 69]. A more general view to similarity is given in Richter [52].

Other classifications on similarity measures focus on the application. There are measures for time-series [53], similarity measures for shapes [56], graphs [41], music classification [64], and others.

Translation, size, scale and rotation invariance are another important aspect of similarity as concerns technical systems.

Most real-world applications nowadays are more complex than the robot example given above. They are usually comprised of many attributes that are different in nature. Numerical attributes given by different sensors or technical measurements and categorical attributes that describe meta-knowledge of the application usually make up a case. These n different attribute groups can form partial similarities $Sim_1, Sim_2, ..., Sim_n$ that can be calculated based on different similarity measures and may have a contextual meaning for itself. The final similarity might be comprised of all the contextual similarities. The simplest way to calculate the overall similarity is to sum up over all contextual similarities: $Sim = \alpha_1 Sim_1 + \alpha_2 Sim_2 ... + \alpha_n Sim_n$ and model the influence of the similarities by different importances α_i. Other schemas for combining similarities are possible as well. The usefulness of such a strategy has been shown for meta-learning of segmentation parameters [38] and for medical diagnosis [60].

The introduction of weights into the similarity measure in equation 1 puts a different importance on particular attributes and views similarity not only as global similarity, but also as local similarity. Learning the attribute weights allows building particular similarity metrics for the specific applications. A variety of methods based on linear or stochastic optimization methods [72], heuristics search [67], genetic programming [16], and case-ordering [61] or query ordering in NN-classification, have been proposed for attribute-weight learning.

Learning distance function in response to users' feedback is known as relevance feedback [5, 10] and it is very popular in data base and image retrieval. The optimization criterion is the accuracy or performance of the system rather than the individual problem-case pairs. This approach is biased by the learning approach as well as by the case description.

New directions in CBR research build a bridge between the case and the solution [8]. Cases can be ordered based on their solutions by their preference relations [71] or similarity relation [50] given by the users or a-priori known from the application. The derived values can be used to learn the similarity metric and the relevant features. That means that cases having similar solutions should have similar case descriptions. The set of features as well as the feature weights are optimized until they meet this assumption.

Learning distance function by linear transformation of features has been introduced by Bobrowski et. al [13].

The necessity to study the taxonomy of similarity measures and a first attempt to construct a taxonomy over similarity measures has been given by Perner in [40] and has been further studied by Cunningham [17]. More work is necessary especially when not only one feature type and representation is used in a CBR system, as it is the case for multimedia data. These multimedia cases will be more complex as the cases that only face on one specific data type. To develop novel similarity measures for text, videos, images, and audio and speech signals and to construct a taxonomy that allows understanding the relation between the different similarity measures will be a challenging task. Similarity aggregation of the different types of similarity measures is another challenging topic.

Research has been described for learning of feature weights and similarity measures [16, 67, 72]. Case mining from raw data in order to get more generalised cases has been described in [28]. Learning of generalised cases and the hierarchy over the case base has also been presented. These works demonstrate that the system performance can be significantly improved by these functions of a CBR system. New techniques for learning of feature weights and similarity measures, case generalisation for different data types are necessary.

5   Organization of Case Base

The case base plays a central role in a CBR system. All observed relevant cases are stored in the case base. Ideally, CBR systems start reasoning from an empty memory, and their reasoning capabilities stem from their progressive learning from the cases they process [11].

Consequently, the memory organization and structure are in the focus of a CBR system. Since a CBR system should improve its performance over time, imposes on the memory of a CBR system to change constantly.

In contrast to research in data base retrieval and nearest-neighbour classification, CBR focuses on conceptual memory structures. While *k-d* trees [7, 65] are space-partitioning data structures for organizing points in a k-dimensional space, conceptual memory structures [20, 41] are represented by a directed graph in which the root node represents the set of all input instances and the terminal nodes represent individual instances. Internal nodes stand for sets of instances attached to that node and represent a super-concept. The super-concept can be represented by a generalized representation of the associated set of instances, such as the prototype, the mediod or a user-selected instance. Therefore a concept C, called a class, in the concept hierarchy is represented by an abstract concept description (e.g. the feature names and its values) and a list of pointers to each child concept $M(C) = \{C_1, C_2, ..., C_i, ..., C_n\}$, where C_i is the child concept, called subclass of concept C.

The explicit representation of the concept in each node of the hierarchy is preferred by humans, since it allows understanding the underlying application domain.

While for the construction of a *k-d* tree only a splitting and deleting operation is needed, conceptual learning methods use more sophisticated operations for the construction of the hierarchy [28]. The most common operations are splitting, merging,

adding and deleting. What kind of operation is carried out during the concept hierarchy construction depends on a concept-evaluation function. There are statistical functions known, as well as similarity-based functions.

Because of the variety of construction operators, conceptual hierarchies are not sensitive to the order of the samples. They allow the incremental adding of new examples to the hierarchy by reorganizing the already existing hierarchy. This flexibility is not known for *k-d* trees, although recent work has led to adaptive *k-d* trees that allow incorporating new examples.

The concept of generalization and abstraction should make the case base more robust against noise and applicable to a wider range of problems. The concept description, the construction operators as well as the concept evaluation function are in the focus of the research in conceptual memory structure.

The conceptual incremental learning methods for case base organization puts the case base into the dynamic memory view of Schank [54] who required a coherent theory of adaptable memory structures and that we need to understand how new information changes the memory.

Memory structures in CBR research are not only pure conceptual structures, hybrid structures incorporating *k-d* tree methods are studied also. An overview of recent research in memory organization in CBR is given in [11].

Other work goes into the direction of bridging between implicit and explicit representations of cases [42]. The implicit representations can be based on statistical models and the explicit representation is the case base that keeps the single case as it is. As far as evidence is given, the data are summarized into statistical models based on statistical learning methods such as Minimum Description Length (*MDL*) or Minimum Message Length (*MML*) learning. As long as not enough data for a class or a concept have been seen by the system, the data are kept in the case base. The case base controls the learning of the statistical models by hierarchically organizing the samples into groups. It allows dynamically learning and changing the statistical models based on the experience (data) seen so far and prevents the model from overfitting and bad influences by singularities.

This concept follows the idea that humans have built up very effective models for standard repetitive tasks and that these models can easily be used without a complex reasoning process. For rare events the CBR unit takes over the reasoning task and collects experience into its memory.

6   The Incremental Aspect of CBR

The incremental aspect of the CBR method presents oneself a new task for which new methods have to be developed.

Case-based reasoning, image or data retrieval is based on similarity determination between the actual case and the cases in a database. It is preferable to normalize the similarity values between 0 and 1 in order to be able to compare different similarity values based on a scale. A scale between 0 and 1 gives us a symbolic understanding of the meaning of the similarity value. The value of 0 indicates identity of the two

cases while the value of 1 indicates the cases are unequal. On the scale of 0 and 1 the value of 0.5 means neutral and values between 0.5 and 0 mean more similarity and values between 0.5 and 1 mean more dissimilarity.

Different normalization procedures are known. The most popular one is the normalization to the upper and lower bounds of a feature value.

When not having any a-priori information about the distribution of the data, the upper and lower bounds of a feature value $[x_{\min,i,k}, x_{\max,i,k}]$ can only be judged based on this limited set of cases at the point in time t_k and must not meet the true values of $x_{\min,i}$ and $x_{\max,i}$ of feature i. Then the scale of similarity might change over time periods t_k which will lead to different decisions for two cases at the points in time t_k and t_{k+l} [4]. It leads to the task of incremental learning the frequency distribution of the data [45] and the question how to use the decision of the CBR system. One method for the later can be to fit the similarity based on expert's judgment about the similarity in the assumption that this value is more close to the true value [46].

Another important aspect is the kind of normalization used. Studies show that normalization has a big impact on the final results [19, 37]. The selection of the right normalization method is another challenging task.

Besides that the incremental aspects is still a task for the organization of the case base and feature learning.

7   The Life-Cycle of a CBR System

The incremental aspect of the CBR put the aspect of the Life-Cycle [18, 33, 34] of a CBR system into a central point of view. One question is: How to deal with changing cases? Cases were the set of features is changing.

The question of the Life-Cycle of a CBR system goes along with the learning capabilities, case base organization and maintenance mechanism, standardization and software engineering [47] for which new concepts should be developed.

8   Applications

CBR has been successfully applied to a wide range of problems. Among them are signal interpretation tasks [49], medical applications [25], and emerging applications such as geographic information systems, applications in biotechnology and topics in climate research (CBR commentaries) [31].

The development of CBR methods for multimedia data is still a challenging topic. So far four different subfields have been developed over the time: CBR for image and video analysis and interpretation [40, 68], CBR for 1-D signals and sequences [2, 48], and CBR for spatio-temporal data [23, 24] (see Fig. 3).

We are focussing here on hot real-world topics such as meta-learning for parameter selection, image&signal interpretation, prototype-based classification and novelty detection & handling.
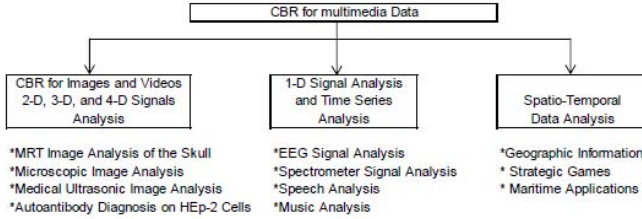
Fig. 3 Overview about Multimedia Topics in CBR

8.1 Meta-learning for Parameter Selection of Data/Signal Processing Algorithms

Meta learning is a subfield of Machine Learning where automatic learning algorithms are applied on meta-data about machine-learning experiments. The main goal is to use such meta-data to understand how automatic learning can become flexible as regards solving different kinds of learning problems, hence to improve the performance of existing learning algorithms. Another important meta-learning task, but not so widely studied yet, is parameter selection for data or signal processing algorithms. Soares et. al [59] have used this approach for selecting the kernel width of a support-vector machine, while Perner [38] and Frucci et. al [22] have studied this approach for image segmentation.

The meta-learning problem for parameter selection can be formalized as follows: For a given signal that is characterized by specific signal properties A and domain properties B find the parameters of the processing algorithm that ensure the best quality of the resulting output signal:

$$f : A \cup B \rightarrow P_i \tag{3}$$

with P_i the i-th class of parameters for the given domain.

Meta-data for signals are comprised of signal-related meta-data and non-signal related meta-data [38]. Likewise in the standard system theory [70], the signal-related meta-data should characterize the specific signal properties that influence the result of the signal processing algorithm. In general the processing of signal-related meta-data from signals should not require too much processing since it is auxiliary process to achieve the final result.

The architecture of Case-based Reasoning for Image Segmentation is shown in Fig. 4. This architecture has been applied to threshold-based image segmentation [38] and the Watershed Transformation [22]. The resulting good segmentation quality compared to standard Watershed segmentation result is shown in Fig. 5 for a biological image.

The signal-related meta-data are for the CBR-based Watershed Transformation statistical grey-level and texture parameters such as mean, standard deviation, entropy, and Haralick's texture-descriptor. The non-signal related meta-data are the category of the images such as biological image, face images, and landscape images.
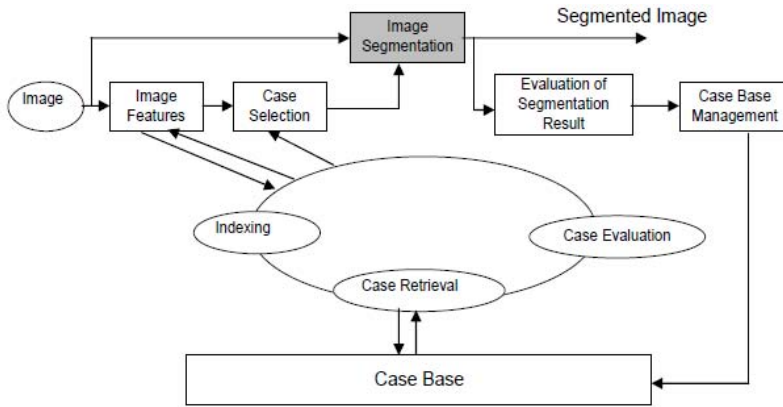
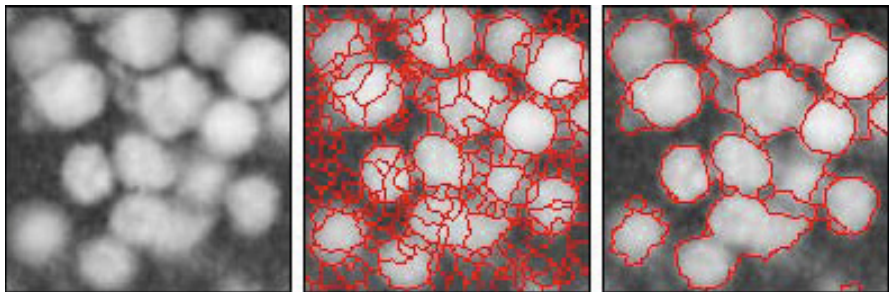**Fig. 4** Meta-Learning for Parameter Selection

Fig. 5 Image Segmentation Results, right Original Image, Watershed Transform, middle, Watershed Transform based on CBR, left

The image segmentation algorithm is the Watershed transformation where the over-segmentation of the result is controlled by weighted merging rules. The weights and the application of the merging rules are controlled by the CBR unit. This unit selects based on the signal characteristics and the category the weights and rules that should be applied for merging basins obtained by the standard Watershed Transformation in the particular image. The output of the segmentation unit is automatically criticised by a specific evaluation measure that compares the input image with the output image and starts the case-base maintenance process if the result is not as good as it should be. Then new case is stored with its meta-data and its segmentation parameters in the case base. Case generalization groups similar cases into a more generalized case so that it is applicable to more signals.

The mapping function f can be realized by any classification algorithm, but the incremental behaviour of CBR fits best to many data/signal processing problems where the signals are not available ad-hoc but appear incrementally. The right similarity metric that allows mapping data to parameter groups and in the last

consequence to good output results should be more extensively studied. Performance measures that allow to judge the achieved output and to automatically criticize the system performances are another important problem. As from the statistics point of view we need more signal-characterizing parameters for the meta-data that bridge between the signal characteristics and the behaviour of the modern, often heuristic signal processing algorithms.

Abstraction of cases to learn domain theory are also related to these tasks and would allow to better understand the behaviour of many signal processing algorithms that cannot be described anymore by standard system theory [70].

8.2   Case-Based Image Interpretation

Image interpretation is the process of mapping the numerical representation of an image into a logical representation such as is suitable for scene description. This is a complex process; the image passes through several general processing steps until the final result is obtained. These steps include image preprocessing, image segmentation, image analysis, and image interpretation. Image pre-processing and image segmentation algorithm usually need a lot of parameters to perform well on the specific image. The automatically extracted objects of interest in an image are first described by primitive image features. Depending on the particular objects and focus of interest, these features can be lines, edges, ribbons, etc. Typically, these low-level features have to be mapped to high-level/symbolic features. A symbolic feature such as fuzzy margin will be a function of several low-level features.

The image interpretation component identifies an object by finding the object to which it belongs (among the models of the object class). This is done by matching the symbolic description of the object to the model/concept of the object stored in the knowledge base. Most image-interpretation systems run on the basis of a bottom-up control structure. This control structure allows no feedback to preceding processing components if the result of the outcome of the current component is unsatisfactory. A mixture of bottom-up and top-down control would allow the outcome of a component to be refined by returning to the previous component.

CBR is not only applicable as a whole to image interpretation, it is applicable to all the different levels of an image-interpretation system [39, 40] (see Fig. 6) and many of the ideas mentioned in the chapters before apply here. CBR-based meta-learning algorithms for parameter selection are preferable for the image pre-processing and segmentation unit [22, 38]. The mapping of the low-level features to the high-level features is a classification task for which a CBR-based algorithm can be applied. The memory organization [41] of the interpretation unit goes along with problems discussed for the case base organization in Section 5. Different organization structures for image interpretation systems are discussed in [40].

Ideally the system should start working with only a few samples and during usage of the system new cases should be learnt and the memory should be updated based on these samples. This view at the usage of a system brings in another topic that is called life-time cycle of a CBR system. Work on this topic takes into account
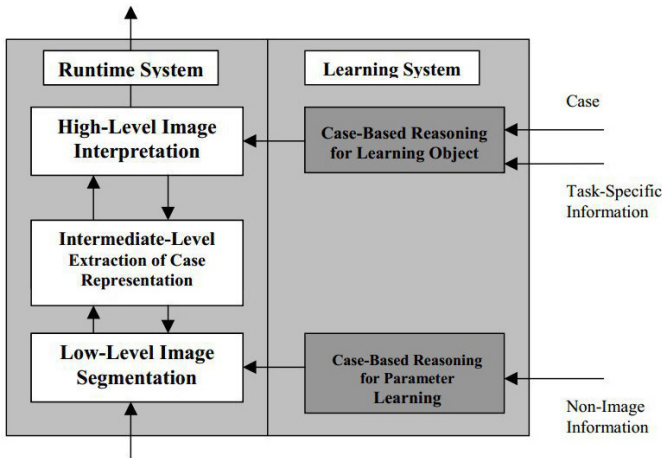
Fig. 6 Architecture of a CBR-Based Image Interpretation System

that a system is used for a long time, while experience changes over time. The case structure might change by adding new relevant attributes or deleting attributes that have shown not to be important or have been replaced by other ones. Set of cases might not appear anymore, since these kinds of solutions are not relevant anymore. A methodology and software architecture for handling the life-time cycle problem is needed so that this process can easily be carried out without rebuilding the whole system. It seems to be more a software engineering task, but has also something to do with evaluation measures for the task of forgetting and relevance of cases that can come from statistics.

8.3   *Incremental Prototype-Based Classification*

The usage of prototypical cases is very popular in many applications, among them are medical applications [55], Belazzi et al. [6] and by Nilsson and Funk [35], knowledge management systems [12] and image classification tasks [44]. The simple nearest-neighbour- approach [1] as well as hierarchical indexing and retrieval methods [55] have been applied to the problem. It has been shown that an initial reasoning system could be built up based on these cases. The systems are useful in practice and can acquire new cases for further reasoning during utilization of the system.

There are several problems concerned with prototypical CBR. If a large enough set of cases is available, the prototypical case can automatically be calculated as the generalization from a set of similar cases. In medical applications as well as in applications where image catalogues are the development basis for the system, the prototypical cases have been selected or described by humans. That means when building the system, we are starting from the most abstract level (the prototype) and
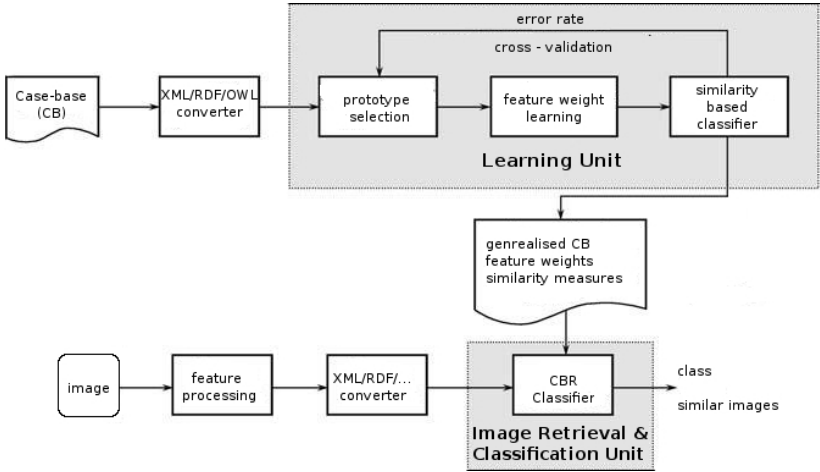
Fig. 7 Architecture of a Prototype-Based Classifier

have to collect more specific information about the classes and objects during the usage of the system.

Since a human has selected the prototypical case, his decision on the importance of the case might be biased and picking only one case might be difficult for a human. As for image catalogue-based applications, he can have stored more than one image as a prototypical image. Therefore we need to check the redundancy of the many prototypes for one class before taking them all into the case base.

According to this consideration, the minimal functions a prototype-based classification system [44] (see Fig. 7) should realize are: classifications based on a proper similarity-measure, prototype selection by a redundancy-reduction algorithm, feature weighting to determine the importance of the features for the prototypes and to learn the similarity metric, and feature-subset selection to select the relevant features from the whole set of features for the respective domain. Cross Validation over the loop of all processing steps can estimate the error rate of such a system (see Fig. 6). However, when the data set is imbalanced that means a class is underrepresented by samples, what always can happen in real domain and when incrementally collecting samples, class-specific error rates have to be calculated to judge the true performance of the system. Otherwise overall error rate will turn out to be good but underrepresented classes will be badly classified [30]. That means that the learning schema for prototype-selection, feature subset selection and feature weighting cannot only rely on the overall error as often. More sophisticated learning strategies are necessary which are incrementally and take whether the class-specific error rate into account or follow the idea of bridging between the cases and the solutions based on preference or similarity relations mentioned in Section 4.

Statistical learning methods so far focus on adaptive k-NN that adapts the distance metric by feature weighting or kernel methods or the number k of neighbours

off-line to the data. Incremental strategies are used for the nearest- neighbour search, but not for updating the weights, distance metric and prototype selection. A system for handwriting recognition is described in [62] that can incrementally add data and adapt the solutions to different users' writing style. A k-NN realization that can handle data streams by adding data through reorganizing a multi-resolution array data structure and concept drift by realizing a case forgetting strategy is described in [29].

8.4   *Novelty Detection by Case-Based Reasoning*

Novelty detection [32], recognizing that an input differs in some respect from previous inputs, can be a useful ability for learning systems.

Novelty detection is particularly useful where an important class is under-represented in the data, so that a classifier cannot be trained to reliably recognize that class. This characteristic is common to numerous problems such as information management, medical diagnosis, fault monitoring and detection, and visual perception.

We propose novelty detection to be regarded as a CBR problem under which we can run the different theoretical methods for detecting the novel events and handling the novel events [42]. The detection of novel events is a common subject in the literature. The handling of the novel events for further reasoning is not treated so much in the literature, although this is a hot topic in open-world applications.

The first model we propose is comprised of statistical models and similarity-based models (see Fig. 8). For now, we assume an attribute-value based representation. Nonetheless, the general framework we propose for novelty detection can be based on any representation. The heart of our novelty detector is a set of statistical models that have been learnt in an off-line phase from a set of observations. Each model represents a case-class. The probability density function implicitly represents the data and prevents us from storing all the cases of a known case-class. It also allows modelling the uncertainty in the data. This unit acts as a novel-event detector by using the Bayesian decision-criterion with the mixture model. Since this set of observations might be limited, we consider our model as being far from optimal and update it based on new observed examples. This is done based on the Minimum Description Length (MDL) principle or the Minimum Message Length (MML) learning principle [63].

In case our model bank cannot classify an actual event into one of the case-classes, this event is recognized as a novel event. The novel event is given to the similarity-based reasoning unit. This unit incorporates this sample into their case base according to a case-selective registration-procedure that allows learning case-classes as well as the similarity between the cases and case-classes. We propose to use a fuzzy similarity measure to model the uncertainty in the data. By doing that the unit organizes the novel events in such a fashion that is suitable for learning a new statistical model.

The case-base-maintenance unit interacts with the statistical learning unit and gives an advice as to when a new model has to be learnt. The advice is based on the
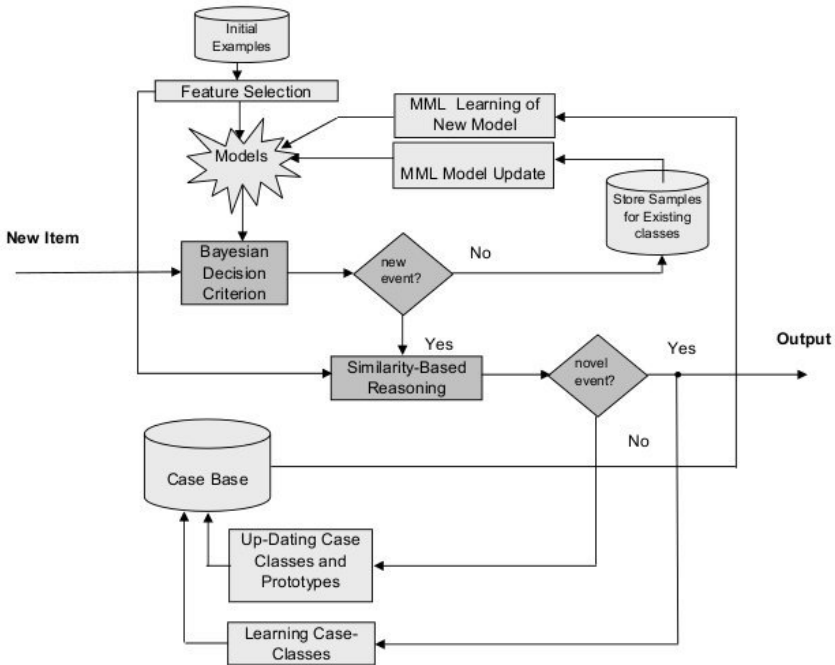
Fig. 8 Architecture of a Statistical and Similarity-Based Novelty Detector and Handling System

observation that a case-class is represented by a large enough number of samples that are most dissimilar to other classes in the case-base.

The statistical learning unit takes this case class and proves based on the MML-criterion, whether it is suitable to learn the new model or not. In the case that the statistical component recommends to not learn the new model, the case-class is still hosted by the case base maintenance unit and further up-dated based on new observed events that might change the inner-class structure as long as there is new evidence to learn a statistical model.

The use of a combination of statistical reasoning and similarity-based reasoning allows implicit and explicit storage of the samples. It allows handling well-represented events as well as rare events.

9 Conclusion

In this paper we have presented our thoughts and work on CBR under the aspect "CBR and Statistical Challenges". The paper is an extension of the work presented in [43]. This work will be constantly extrapolated with future results. CBR solves problems using already stored knowledge, and captures new knowledge, making it immediately available for solving the next problem. To realize this cognitive

model in a computer-based system we need methods known from statistics, pattern recognition, artificial intelligence, machine learning, data base research and other fields. Only the combination of all these methods will give us a system that can efficiently solve practical problems. Consequently, CBR research has shown much success for different application areas, such as medical and technical diagnosis, image interpretation, geographic information systems, text retrieval, e-commerce, user-support systems and so on. CBR systems work efficiently in real-world applications, since the CBR method faces on all aspects of a well-performing and user-friendly system.

We have pointed out that the central aspect of a well-performing system in the real-world is its ability to incrementally collect new experience and reorganize its knowledge based on these new insights. In our opinion the new challenging research aspects should have its focus on incremental methods for prototype-based classification, meta-learning for parameter selection, complex signals understanding tasks and novelty detection. The incremental methods should allow changing the system function based on the newly obtained data.

Recently, we are observing that this incremental aspect is in the special focus of the quality assurance agency for technical and medical application, although this is in opposition to the current quality performance guidelines.

While reviewing the CBR work, we have tried bridging between the concepts developed within the CBR community and the concepts developed in the statistics community. At the first glance, CBR and statistics seem to have big similarities. But when looking closer at it one can see that the paradigms are different. CBR tries to solve real-world problems and likes to deliver systems that have all the functions necessary for an adaptable intelligent system with incremental learning behavior. Such a system should be able to work on a small set of cases and collect experience over time. While doing that it should improve its performance. The solution need not be correct in the statistical sense, rather it should help an expert to solve his tasks and learn more about it over time.

Nonetheless, statistics disposes of a rich variety of methods that can be useful for building intelligent systems. In the case that we can combine and extend these methods under the aspects necessary for intelligent systems, we will further succeed in establishing artificial intelligence systems in the real world.

Our interest is to build intelligent flexible and robust data-interpreting systems that are inspired by the human CBR process and by doing so to model the human reasoning process when interpreting real-world situations.

References

1. Aha, D.W., Kibler, D., Albert, M.K.: Instance-based learning algorithm. Machine Learning 6(1), 37–66 (1991)
2. Ahmed, M.U., Begum, S., Funk, P.: An overview of three medical application using hybrid case-based reasoning. In: Bichindaritz, I., Perner, P., Ruß, G., Schmidt, R. (eds.) Proceedings of the Industrial Conference on Advances in Data Mining (ICDM 2012), Workshop Case-Based Reasoning, pp. 79–94 (2012)

3. Althoff, K.D.: Case-based reasoning. In: Chang, S.K. (ed.) Handbook of Software Engineering and Knowledge Engineering, Fundamentals, vol. 1, pp. 549–588. World Scientific (2001)

4. Attig, A., Perner, P.: The problem of normalization and a normalized similarity measure by online data. Transactions on Case-Based Reasoning 4(1), 3–17 (2011)

5. Bagherjeiran, A., Eick, C.F.: Distance function learning for supervised similarity assesment. In: Perner, P. (ed.) Case-Based Reasoning on Images and Signals. SCI, vol. 73, pp. 91–126. Springer, Heidelberg (2008)

6. Bellazzi, R., Montani, S., Portinale, L.: Retrieval in a prototype-based case-library: A case study in diabetes therapy revision. In: Smyth, B., Cunningham, P. (eds.) EWCBR 1998. LNCS (LNAI), vol. 1488, pp. 64–75. Springer, Heidelberg (1998)

7. Bentley, J.L.: Multidimensional binary search trees used for associative searching. Communication of the ACM 18(9), 509–517 (1975)

8. Bergmann, R., Richter, M.M., Schmitt, S., Stahl, A., Vollrath, I.: Utility-oriented matching: A new research direction for case-based reasoning. In: Schnurr, H.P., et al. (eds.) Professionelles Wissensmanagement, pp. 20–30. Shaker-Verlag (2001)

9. Bergmann, R., Wilke, W.: On the role of abstraction in case-based reasoning. In: Smith, I., Faltings, B. (eds.) EWCBR 1996. LNCS, vol. 1168, pp. 28–43. Springer, Heidelberg (1996)

10. Bhanu, B., Dong, A.: Concepts learning with fuzzy clustering and relevance feedback. In: Perner, P. (ed.) MLDM 2001. LNCS (LNAI), vol. 2123, pp. 102–116. Springer, Heidelberg (2001)

11. Bichindaritz, I.: Memory structures and organization in case-based reasoning. In: Perner, P. (ed.) Case-Based Reasoning on Images and Signals. SCI, vol. 73, pp. 175–194. Springer, Heidelberg (2008)

12. Bichindaritz, I., Kansu, E., Sullivan, K.M.: Case-based reasoning in care-partner: Gathering evidence for evidence-based medical practice. In: Smyth, B., Cunningham, P. (eds.) EWCBR 1998. LNCS (LNAI), vol. 1488, pp. 334–345. Springer, Heidelberg (1998)

13. Bobrowski, L., Topczewska, M.: Improving the K-NN classification with the euclidean distance through linear data transformations. In: Perner, P. (ed.) ICDM 2004. LNCS (LNAI), vol. 3275, pp. 23–32. Springer, Heidelberg (2004)

14. Branting, L.K.: Integrating generalizations with exemplar-based reasoning. In: Proceedings of the 11th Annual Conference of the Cognitive Science Society, pp. 129–146. Lawrence Erlbaum (1989)

15. Commentaries, C.C.: The Knowledge Engineering Review 20(3) (2005)

16. Craw, S.: Introspective learning to build case-based reasoning (CBR) knowledge containers. In: Perner, P., Rosenfeld, A. (eds.) MLDM 2003. LNCS (LNAI), vol. 2734, pp. 1–6. Springer, Heidelberg (2003)

17. Cunningham, P.: A taxonomy of similarity mechanisms for case-based reasoning. IEEE Transactions on Knowledge and Data Engineering 21(11), 1532–1543 (2009)

18. Dingsoyr, T.: A lifecycle process for experience databases. In: Schmitt, S., Vollrath, I. (eds.) Challenges for Case-Based Reasoning - Proceedings of the ICCBR 1999 Workshops, pp. 9–14 (1999)

19. Fayyad, U.M., Piatesky-Shapiro, G., Smyth, P., Utuhrusamy, R. (eds.): Advance in Knowledge Discovery and Data Mining. AAAI Press (1996)

20. Fisher, D.H.: Knowledge acquisition via incremental conceptual clustering. Machine Learning 2(2), 139–172 (1987)

21. Fiss, P.: Data Reduction Methods for Industrial Robots with Direct Teach-In Programming, Diss A. Technical University Mittweida (1985)

22. Frucci, M., Perner, P., Sanniti di Baja, G.: Case-based reasoning for image segmentation by watershed transformation. In: Perner, P. (ed.) Case-Based Reasoning on Signals and Images. SCI, vol. 73, pp. 319–353. Springer, Heidelberg (2008)

23. Gupta, K.M., Aha, D.W., Moore, P.: Case-based collective inference for maritime object classification. In: McGinty, L., Wilson, D.C. (eds.) ICCBR 2009. LNCS (LNAI), vol. 5650, pp. 434–449. Springer, Heidelberg (2009)

24. Hegazy, O.M., Hemeida, I.H., Eldein, M.N., Elhusseiny, J.: Similarity assessment mechanism for spatiotemporal data sets in case-based reasoning. In: Bichindaritz, I., Perner, P., Ruß, G., Schmidt, R. (eds.) Proceedings of the Industrial Conference on Advances in Data Mining (ICDM 2012), Workshop on Case-Based Reasoning, pp. 62–78 (2012)

25. Holt, A., Bichindaritz, I., Schmidt, R., Perner, P.: Medical applications in case-based reasoning. The Knowledge Engineering Review 20(3), 289–292 (2005)

26. Iglezakis, I., Reinartz, T., Roth-Berghofer, T.R.: Maintenance memories: Beyond concepts and techniques for case base maintenance. In: Funk, P., González Calero, P.A. (eds.) ECCBR 2004. LNCS (LNAI), vol. 3155, pp. 227–241. Springer, Heidelberg (2004)

27. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice Hall, Inc., Upper Saddle River (1988)

28. Jänichen, S., Perner, P.: Conceptual clustering and case generalization of two dimensional forms. Computational Intelligence 22(3-4), 177–193 (2006)

29. Law, Y.N., Zaniolo, C.: An adaptive nearest neighbor classification algorithm for data streams. In: Jorge, A.M., Torgo, L., Brazdil, P.B., Camacho, R., Gama, J. (eds.) PKDD 2005. LNCS (LNAI), vol. 3721, pp. 108–120. Springer, Heidelberg (2005)

30. Little, S., Colantonio, S., Salvetti, O., Perner, P.: Evaluation of feature subset selection, feature weighting, and prototype selection for biomedical applications. Journal of Software Engineering & Applications 3(1), 39–49 (2010)

31. Lopez De Mantaras, R., Cunningham, P., Perner, P.: Emergent case-based reasoning applications. The Knowledge Engineering Review 20(3), 325–328 (2005)

32. Markou, M., Singh, S.: Novelty detection: A review — part 1: Statistical approaches. Signal Processing 83(12), 2481–2497 (2003)

33. Minor, M., Hanft, A.: Cases with a life-cycle. In: Schmitt, S., Vollrath, I. (eds.) Challenges for Case-Based Reasoning - Proceedings of the ICCBR 1999 Workshops, pp. 3–8. University of Kaiserslautern, Computer Science (1999)

34. Minor, M., Hanft, A.: The life cycle of test cases in a CBR system. In: Blanzieri, E., Portinale, L. (eds.) EWCBR 2000. LNCS (LNAI), vol. 1898, pp. 455–466. Springer, Heidelberg (2000)

35. Nilsson, M., Funk, P.: A case-based classification of respiratory sinus arrhythmia. In: Funk, P., González Calero, P.A. (eds.) ECCBR 2004. LNCS (LNAI), vol. 3155, pp. 673–685. Springer, Heidelberg (2004)

36. Pękalska, E., Duin, R.P.W.: The Dissimilarity Representation for Pattern Recognition. World Scientific (2005)

37. Perner, J., Zotenko, E.: Characterizing cell types through differentially expressed gene clusters using a model-based approach. Transactions on Case-Based Reasoning 4(1), 3–17 (2011)

38. Perner, P.: An architecture for a CBR image segmentation system. Engineering Application in Artificial Intelligence 12(6), 749–759 (1999)

39. Perner, P.: Using CBR learning for the low-level and high-level unit of a image interpretation system. In: Singh, S. (ed.) Proceedings of the International Conference on Advances in Pattern Recognition (ICAPR 1998), pp. 45–54. Springer (1999)

40. Perner, P.: Why case-based reasoning is attractive for image interpretation. In: Aha, D.W., Watson, I. (eds.) ICCBR 2001. LNCS (LNAI), vol. 2080, pp. 27–43. Springer, Heidelberg (2001)

41. Perner, P.: Case-base maintenance by conceptual clustering of graphs. Engineering Applications of Artificial Intelligence 19(4), 381–393 (2006)

42. Perner, P.: Concepts for novelty detection and handling based on a case-based reasoning scheme. In: Perner, P. (ed.) ICDM 2007. LNCS (LNAI), vol. 4597, pp. 21–33. Springer, Heidelberg (2007)

43. Perner, P.: Case-based reasoning and the statistical challenges. Quality and Reliability Engineering International 24(6), 705–720 (2008)

44. Perner, P.: Prototype-based classification. Applied Intelligence 28(3), 238–246 (2008)

45. Perner, P.: Incremental normalization for CBR. Transactions on Case-Based Reasoning 5(1), 35–50 (2012)

46. Perner, P.: Improving prototype-based classification by fitting the similarity. In: Proceedings of ISA International Conference Intelligent Systems and Agents (2013)

47. Perner, P. (ed.): Machine Learning, Software Engineering, and Standardization. Ibai-Publishing (2013)

48. Perner, P., Attig, A., Machnow, O.: A novel method for the interpretation of spectrometer signals based on delta-modulation and similarity determination. Transactions on Mass-Data Analysis of Images and Signals 3(1), 3–14 (2011)

49. Perner, P., Holt, A., Richter, M.: Image processing in case-based reasoning. The Knowledge Engineering Review 20(3), 311–314 (2005)

50. Perner, P., Perner, H., Müller, B.: Similarity guided learning of the case description and improvement of the system performance in an image classification system. In: Craw, S., Preece, A.D. (eds.) ECCBR 2002. LNCS (LNAI), vol. 2416, pp. 604–612. Springer, Heidelberg (2002)

51. Richter, M.M.: Introduction. In: Lenz, M., Bartsch-Spörl, B., Burkhard, H.-D., Wess, S. (eds.) Case-Based Reasoning Technology. LNCS (LNAI), vol. 1400, pp. 1–16. Springer, Heidelberg (1998)

52. Richter, M.M.: Similarity. In: Perner, P. (ed.) Case-Based Reasoning on Images and Signals. SCI, vol. 73, pp. 1–21. Springer, Heidelberg (2008)

53. Sankoff, D., Kruskal, J. (eds.): Time warps, string edits, and macromolecules: the theory and practice of sequence comparison. Addison-Wesley, Readings (1983)

54. Schank, R.C.: Dynamic Memory: A theory of reminding and learning in computers and people. Cambridge University Press, Cambridge (1983)

55. Schmidt, R., Gierl, L.: Temporal abstractions and case-based reasoning for medical course data: Two prognostic applications. In: Perner, P. (ed.) MLDM 2001. LNCS (LNAI), vol. 2123, pp. 23–34. Springer, Heidelberg (2001)

56. Shapiro, L.G., Atmosukarto, I., Cho, H., Lin, H.J., Ruiz-Correa, S.: Similarity-based retrieval for biomedical applications. In: Perner, P. (ed.) Case-Based Reasoning on Signals and Images. SCI, vol. 73, pp. 355–388. Springer, Heidelberg (2007)

57. Smith, E.E., Medin, D.L.: Categories and Concepts. Havard University Press (1981)

58. Smith, L.B.: From global similarities to kinds of similarities: the construction of dimensions in development. In: Vosniadou, S., Ortony, A. (eds.) Similarity and Analogical Reasoning, pp. 146–178. Cambridge University Press, New York (1989)

59. Soares, C., Brazdil, P.B., Kuba, P.: A meta-learning method to select the kernel width in support vector regression. Machine Learning 54(3), 195–209 (2004)

60. Song, X., Petrovic, S., Sundar, S.: A case-based reasoning approach to dose planning in radiotherapy. In: Wilson, D.C., Khemani, D. (eds.) Worshop Proceedings of the 7th International Conference on Case-Based Reasoning (ICCBR 2007), pp. 348–357 (2007)

61. Stahl, A.: Learning feature weights from case order feedback. In: Aha, D.W., Watson, I. (eds.) ICCBR 2001. LNCS (LNAI), vol. 2080, pp. 502–516. Springer, Heidelberg (2001)
62. Vuori, V., Laaksonen, I., Oja, E., Kangas, J.: Experiments with adaptation strategies for a prototype-based recognition system for isolated handwritten characters. International Journal on Document Analysis and Recognition 3(3), 150–159 (2001)
63. Wallace, C.S.: Statistical and Inductive Inference by Minimum Message Length. Information Science and Statistics. Springer(2005)
64. Weihs, C., Ligges, U., Mörchen, F., Müllensiefen, D.: Classification in music research. Advances in Data Analysis and Classification 1(3), 255–291 (2007)
65. Wess, S., Althoff, K.D., Derwand, G.: Using k-d trees to improve the retrieval step in case-based reasoning. In: Wess, S., Althoff, K.-D., Richter, M.M. (eds.) EWCBR 1993. LNCS, vol. 837, pp. 167–182. Springer, Heidelberg (1994)
66. Wess, S., Globig, C.: Case-based and symbolic classification. In: Wess, S., Althoff, K.-D., Richter, M.M. (eds.) EWCBR 1993. LNCS, vol. 837, pp. 77–91. Springer, Heidelberg (1994)
67. Wettschereck, D., Aha, D.W., Mohri, T.: A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. Artificial Intelligence Review 11(1-5), 273–314 (1997)
68. Wilson, D.C., O'Sullivan, D.: Medical imagery in case-based reasoning. In: Perner, P. (ed.) Case-Based Reasoning on Images and Signals. SCI, vol. 73, pp. 389–418. Springer, Heidelberg (2008)
69. Wilson, D.R., Martinez, T.R.: Improved heterogeneous distance functions. Journal of Artificial Intelligence Research 6(1), 1–34 (1997)
70. Wunsch, G.: Systemtheorie der Informationstechnik. Akademische Verlagsgesellschaft, Leipzig (1971)
71. Xiong, N., Funk, P.: Building similarity metrics reflecting utility in case-based reasoning. Journal of Intelligent & Fuzzy Systems 17(4), 407–416 (2006)
72. Zhang, L., Coenen, F., Leng, P.: Formalising optimal feature weight settings in case-based diagnosis as linear programming problems. Knowledge-Based Systems 15(7), 391–398 (2002)