

Case-Based Reasoning

13. Case-Based Reasoning for Biomedical Informatics and Medicine

Periklis Andritsos, Igor Jurisica, Janice I. Glasgow

Case-based reasoning (CBR) is an integral part of artificial intelligence. It is defined as the process of solving new problems through their comparison with similar ones with existing solutions. The CBR methodology fits well with the approach that healthcare workers take when presented with a new case, making its incorporation into a clinical setting natural. Overall, CBR is appealing in medical domains because a case base already exists, storing symptoms, diagnoses, treatments, and outcomes for each patient. Therefore, there are several CBR systems for medical diagnosis and decision support. This chapter gives an overview of CBR systems, their lifecycle, and different settings in which they appear. It also discusses major applications of CBR in the biomedical field, the methodologies used, and the systems that have been adopted. Section 13.1 provides the necessary background of CBR, while Sect. 13.2 gives an overview of techniques. Section 13.3 presents different systems in which CBR has been successfully applied, and Sect. 13.4 presents biomedical appl-

| | | |
|--------|--|-----|
| 13.1 | Case-Based Reasoning | 208 |
| 13.1.1 | CBR Content Modeling..... | 208 |
| 13.1.2 | CBR System Lifecycle | 208 |
| 13.1.3 | Case Maintenance | 209 |
| 13.1.4 | Adaptation of CBR in Medicine..... | 210 |
| 13.2 | Feature Selection for CBR | 210 |
| 13.2.1 | Feature Subset Selection | 210 |
| 13.2.2 | Feature Ranking..... | 210 |
| 13.3 | Case-Based Reasoning Systems | 211 |
| 13.3.1 | The T _{A3} CBR System..... | 212 |
| 13.4 | Case-Based Reasoning in Biomedicine ... | 214 |
| 13.4.1 | The eXit*CBR System | 214 |
| 13.4.2 | Diagnosis with CBR Systems | 214 |
| 13.4.3 | Medical Imaging and CBR Systems | 216 |
| 13.4.4 | Protein Crystallization Planning with CBR | 217 |
| 13.5 | Conclusions | 218 |
| | References | 219 |

ications. A concluding discussion closes the chapter in Sect. 13.5.

Many areas of bioinformatics have benefited from artificial intelligence problem-solving techniques. Computational biology approaches have been applied and offered enormous advances to a wide breadth of medical applications including diagnosis, prognosis, etc. [13.1, 2]. These fields are often characterized by complex data, many unknowns, incomplete theories, and rapid evolution. In decision-making, reasoning is often based on experience, rather than on general knowledge. In this chapter we consider one such machine-learning approach, *case-based reasoning* (CBR), and discuss how it has been applied to problems in bioinformatics.

CBR using analogy-based reasoning is a multidisciplinary area of research that deals with the reuse

of experiences, called *cases* [13.3, 4]. CBR is defined as a plausible, high-level model for cognitive processing [13.5] as well as a computational paradigm for problem-solving [13.3]. This paradigm uses a bottom-up approach by exploiting knowledge gathered after solving specific problem situations. In contrast, top-down approaches start with the problem domain theory, which is usually expressed using a specific language, e.g., rule-based language, frames, first-order logic, and semantics networks. In brief, CBR is well suited for capturing both objective details as well as contextual ones [13.6].

Our goal is to present the specific challenges involved in analyzing the underlying datasets. Hence, we will explore CBR techniques as they are applied to im-

age data, as well as the diagnosis of diseases using numerical and textual data.

The rest of this chapter is organized as follows. Section 13.1 provides the necessary background of CBR, while Sect. 13.2 gives an overview of techniques used to

perform feature selection on data to be handled by CBR systems. Section 13.3 presents different systems in which CBR has been successfully applied, while Sect. 13.4 focuses on applications in the biomedical domain. Finally, Sect. 13.5 presents concluding discussions.

13.1 Case-Based Reasoning

When building a CBR system, one of the fundamental components is how cases are represented, i. e., the exemplar solutions that we have stored from previous experience. These are solutions that will be used together with the problem description and possibly evaluation of the solution. At design time, one must decide on the following two issues:

1. The model (data structure) used to store the case-base content,
2. The organization of the case memory.

13.1.1 CBR Content Modeling

Case-base content is usually application dependent. Typically, there are three main ways to represent the cases:

1. As a set of features in a vector, usually called the vector-space model,
2. As text, structured (e.g., inside a database management system) or semistructured [e.g., in extensible markup language (XML) documents],
3. As complex objects, such as graphs.

There are also hybrid approaches that mix, for instance, text-based and vector-based models.

Vector-based modeling is of particular interest since this is a representation inherent in most machine-learning techniques, such as feature selection and extraction as well as clustering and classification methods [13.7]. Therefore, most CBR systems have adopted this representation so that these techniques can be applied on available datasets. Objects in these datasets are modeled as vectors, and therefore they often need to be converted into a space where relevant similarity measures are defined and can be incorporated as part of the system; For example, one of the typical transformations of document corpuses into vectors of decimal numbers (or scores) is done by means of the *term-frequency*, *inverse document-frequency* (tf-idf) weighting method. This method computes the frequency of a term t in

a document $d \in D$ as

$$\text{tf}(x) = \|t \in t \in d\| ,$$

as well as the inverse document frequency of x in the corpus D as

$$\text{idf}(x) = \log \frac{\|D\|}{\|d : t \in d\|} .$$

Then, the **tf-idf** of term x is defined as $\text{tf-idf}(x) = \text{tf}(x) \times \text{idf}(x)$, giving higher scores to terms that appear many times in a document and have a low document frequency, i. e., that are considered highly informative with respect to others. After computing these scores for each term inside the documents, we can represent each document by means of the **tf-idf** scores of their terms.

Finally, many CBR approaches view problem-solving as automatic classification or function approximation tasks [13.8, 9].

Regardless of the case representation, all cases include the problem description, the solution, and the outcome. The first refers to the set of features that are matched when we have a new problem in the system. This must include all the information needed to first discover that a case can be successfully reused for solving a similar problem. The solution models the information for which we are searching, e.g., the diagnosis of a disease or the plan to perform protein crystallization. Finally, the outcome provides an evaluation of the applicability or quality of the solution for the given problem.

13.1.2 CBR System Lifecycle

The problem-solving cycle of a typical CBR system is shown in Fig. 13.1.

Retrieve (1)

Given the description of a new problem, the CBR system retrieves a set of cases stored in the case base. The retrieval uses a similarity metric to compare the problem component of the new case that we are about to build with the problem descriptions of the cases in the base.

In a problem where documents are represented as vectors of *tf-idf* scores, as described above, one common similarity metric that is used is called *cosine similarity* and calculates the angle between corresponding vectors of documents. Given the *tf-idf* vectors v_i and v_j of two documents $d_i \in D$ and $d_j \in D$, respectively, the *cosine similarity*, $\approx (v_i, v_j)$ is given by the expression

$$\approx (v_i, v_j) = \frac{v_i \cdot v_j}{\|v_i\| \cdot \|v_j\|},$$

where the nominator is the inner product between the two document vectors and the denominator the arithmetic product of their norms. For very large case bases, other techniques such as indexing and clustering of cases may need to be incorporated to make this step more efficient [13.10]. This is useful in particular when the context does not change. By context we mean the constraints that are involved in the problem-solving without intervening in it [13.11].

Reuse (2)

This step includes the testing of the solved case in the real dataset. Cases can be reused through evaluation against a domain expert, a simulated model, or known solutions (*test set*). In other words, this step is based on the ability to associate concepts to facts by analogy, called *analogy-based reasoning* [13.12]. The similar cases that are retrieved are reused to build the best solution. This solution could simply be the solution of the most similar case in the base, or an integration of solutions that are extracted from the retrieved cases to build a new candidate solution.

Revise (3)

The candidate solution is then adapted to fit any specific constraints of the current situation; For instance, an extracted therapy should be adapted for a new patient suffering from a particular disease.

Review (4)

The solution that the system builds should now be evaluated by applying it (or simulating the application) to the current problem. If we detect failure, we have to go back and revise the solution or keep the solution as a negative example for future use [13.13]. The *reuse*, *revise*, and *review* stages are also called *case adaptation* [13.5].

Retain (5)

The new case may or may not be added to the case base, depending on its similarity to existing cases and poten-

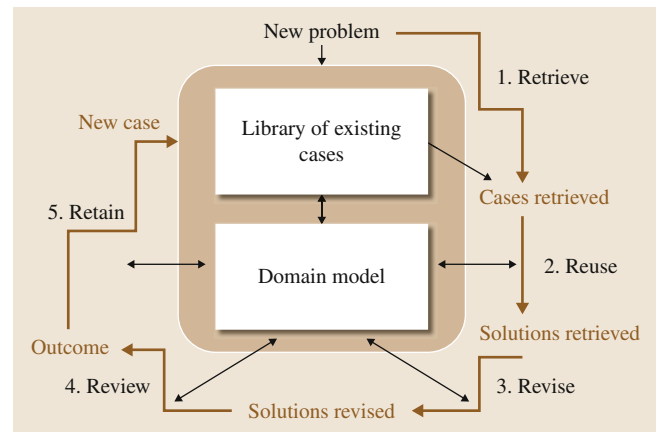


Fig. 13.1 Problem-solving cycle of a CBR system

tial value during problem-solving, since there would be little or no value to add identical or highly similar cases to the case base.

13.1.3 Case Maintenance

The last component of the CBR cycle has attracted particular attention, as unnecessary growth in the size of the case base may negatively affect its performance. Case-based maintenance methods have been proposed in response to this problem through deletions, additions of selected cases, and merging of similar cases. Yang and Wu [13.14] introduced a density-based clustering and information theory-based approach that results in case bases of smaller sizes where retrieval is guided by comparing the information contents of new problems and the clusters built. However, this approach is moving away from the CBR strategy (lazy learning) toward traditional machine-learning approaches, and thus would be applicable in more stable domains. Lawanna and Daengdej [13.15] give a concise set of case deletion and addition heuristics. Their heuristics evaluate the utility of candidate cases by measuring their *coverage* and *reachability*. Coverage is the set of problems that the case can solve, while reachability is the set of cases that can be used to provide a solution for a target problem. Given a value for both measures, cases are retained or deleted. Finally, Arshadi and Jurisica [13.16] approach the problem of case maintenance from a machine-learning point of view. The goal of their method, termed mixture of experts for CBR (MOE4CBR), is to increase the prediction accuracy of CBR classifiers in high-dimensional domains by using a mixture of experts where an ensemble of CBR sys-

tems is integrated with clustering and feature selection to improve performance. They employ spectral clustering to group samples, and each group forms a separate CBR system. Logistic regression is then applied to select those features that more accurately predict the class labels.

13.1.4 Adaptation of CBR in Medicine

The CBR cycle described above fits well with the approach that a healthcare worker takes when presented with a new case, making its incorporation into a clinical setting natural. Overall, CBR is appealing in medical domains because a case base already exists, storing symptoms, diagnoses, treatments, and outcomes for each patient. Therefore, there are several CBR systems for medical diagnosis and decision support [13.17–21]. These systems increasingly use knowledge engineering techniques [13.22]. As more complex domains are tackled by CBR systems, where representing cases

and adapting the solutions of retrieved cases become difficult, systematic approaches to CBR development are needed [13.23, 24]. This is important in these domains because it elucidates knowledge that aids in the construction of a meaningful case representation, meaningful in the sense that it allows for retrieved cases to be matched as closely as possible to the target case in order to reuse their solutions with little adaptation. CBR still has clear benefits in these domains as long as the knowledge engineering efforts required to construct such a case representation are less than what would be required to construct an entire general model [13.24]. Overall, the effectiveness of CBR depends on the domain stability, coverage, as well as the quality and quantity of cases in a case base. With an increased number of unique cases, the problem-solving capabilities of CBR systems may improve at the expense of a decrease in efficiency. In addition, the scalability of a system depends on the model used and the similarity-based retrieval algorithm.

13.2 Feature Selection for CBR

13.2.1 Feature Subset Selection

As one of the key components of CBR, a similarity measure is used to assess closeness of a given *problem case* to cases in the case base, considering any specific conditions (context). This similarity is computed over sets of features that are considered important for the problem at hand. The choice of case features that best distinguish classes of instances has a large impact on the similarity measure and has become an important preprocessing step of case-based reasoning for different domains, termed *feature selection* [13.25]. Techniques that involve the selection of features according to how well they predict given class labels are called *supervised* [13.26]. They usually perform an exhaustive search of all possible subsets of features, and therefore proper heuristics may be used to reduce the complexity. On the other hand, *unsupervised* techniques do not incorporate given class labels but instead employ importance or proximity measures in order to select appropriate sets of features [13.27]

There are two types of feature selectors:

1. *Wrapper* selectors, which use the learning algorithm as a black box with the goal of minimizing the fitting error for a particular problem [13.28], and,

2. *Filter* selectors, which choose features by evaluating some preset criteria independent of the learning algorithm.

In practice, filter selectors have much lower complexity than wrappers, and the features selected often yield comparable prediction errors [13.29].

13.2.2 Feature Ranking

Several feature selection search algorithms have been proposed specifically for case-based reasoning, such as exhaustive search, hill-climbing, and forward sequential selection [13.30, 31]. Such techniques employ objective functions that can be used for *feature ranking*, such as the *tf-idf* criterion discussed above, Fisher's criterion, *t*-test, and logistic regression [13.7, 9, 32]. Highly ranked features are deemed more *valuable*, as opposed to features that have lower ranks. The latter ones are of importance in certain domains, such as in the search for prognostic gene signatures for clinical outcomes [13.33].

Depending on the application domain, a different feature selection approach may need to be applied so that appropriate similarity measures can be employed. Feature selection is highly dependent upon the type of

data with which we are dealing. Numerical data stored as cases of CBR systems have well-defined geometrical characteristics where similarity measures, such as Euclidean distance, can be applied. On the other hand, in a text-based CBR system a case includes the information extracted from the text. This information comprises a set of important keywords that can also be ranked using, for example, the *tf-idf* score and stored in CBR cases. This approach usually integrates CBR with information retrieval [13.34–36] and leverages statistical information inherent in the documents.

One domain in which we are interested in building case bases is that of protein–protein interactions (PPI) from text contained in publication abstracts [13.37]. In this application, features are keywords that are related to the way that PPIs are expressed in collections of papers that deal with specific biological experiments for their detection. As an input to the system, we use sets of terms related to PPI experiments, such as *mass spectrometry*. We use clustering to separate the expressions that are very similar to each other and dissimilar to other PPI expressions. We finally perform keyword scoring in order to measure the utility of each keyword using a scoring function called *category utility* [13.38, 39]. Intuitively, this function measures the *utility* of the individual keywords in the clusters of similar PPI experiments. If we have a corpus D of sentences relevant to PPIs and a cluster C that has been created from this corpus, the utility of a keyword $k \in C$ is calculated as [13.40]

$$CU(k, C) = P(k|C)^2 - P(k)^2,$$

which calculates the difference between the expectation of the keyword k when the clustering C is given with its expectation in the full corpus. Using this measure, the keywords that are more responsible for the creation of the cluster will have a higher probability of appearance in it and, hence, a higher CU score.

More complex objects such as images often require more advanced preprocessing before storing them in a CBR. Implicit image features are extracted using

different techniques such as morphological characterization that applies image segmentation and feature extraction to determine and quantify image texture, distinctive objects contained within images, intensity, presence or absence of straight lines, light distribution, dark points, etc. [13.41]. Image and nonimage features can be combined, and similarity between stored and raw cases can be measured [13.42]. If the purpose of using a CBR system is to classify the images, many of the features can become redundant if they have similar predictive and expressive power. Hence, the challenge is to select a minimal number of features from each image.

The following is a list of techniques proposed for feature extraction in protein crystallization:

- Laplacian pyramid filter [13.43]: The Laplacian filter is used to decompose the image into three different levels. The Laplacian filter is used to extract the boundary information and image features. The multiscale representation is capable of extracting the following useful features of the image (invariant to orientation): mean, standard deviation, skewness, kurtosis, energy, entropy, autocorrelation, and power.
- Extraction of contours [13.44]: Using edge-detection techniques, proper contours are extracted, and the identification of the type of line segments helps in the classification of images. Useful features for which these techniques produce scores are the maximum length and number of line segments as well as the ratio of linear regions.
- Texture features [13.45].
- Gabor wavelet decompositions for edge detection, noise filtering, image compression, texture analysis, and synthesis [13.46].
- Fourier and wavelet analysis [13.47].

Feature selection can be an inherent component of a CBR system, or it can be performed as a preprocessing step. Systems such as eXiT*CBR [13.48] incorporate basic preprocessing and feature selection methods to facilitate experimentation.

13.3 Case-Based Reasoning Systems

Early systems were mainly applied in datasets with a low number of symbolic attributes that contained discrete values [13.49]. With richer heterogeneous real-world data sources available, appropriate treatment and interpretation of large and complex types

of data constitute key issues for developing CBR systems.

CBR systems are being developed in both industrial and scientific applications. Moreover, a variety of systems have been developed to solve problems in de-

sign [13.50], cost estimation [13.51], business process planning and customer support call-centers [13.52], finance [13.53], and legal reasoning [13.54]. Most of the systems follow the standard problem-solving life-cycle described above. Depending on the data that they need to handle, they differ in the way they preprocess and store the cases, as well as the algorithmic approach used to assess the similarity of existing cases with new problems. *Göker et al.* [13.49] suggest a software engineering methodology for CBR. Similar to the development of software applications, they propose a workflow model, and their methodology deems the user, the organization, and the domain at hand as its main characteristics, while it provides a classification of each one.

CBR systems have also been incorporated as modules within other decision-making processes; For instance, *Ahmed et al.* [13.55] describe a system that uses intelligent agents that deploy CBR systems whose role is to assist these agents to gain experience by collecting past solved cases and adapt them to the current context. This enables flexible and modular maintenance systems where different suppliers can deliver agents that eventually develop to become experts in specific tasks. Their framework is geared towards knowledge transfer in complex technical fields, cost reduction, and faster response times.

External knowledge is often included in CBR systems as part of their main knowledge source and can be used from retrieval to reasoning. *D'Aquin et al.* [13.56] have integrated the C-OWL context ontology with a CBR system. Within this system, semantic relations between contexts and the associated reasoning mechanisms in a particular problem are reused and shared in other problems. In a similar fashion, *Sauer et al.* [13.57] discuss the advantages of integrating linked open data (LOD) with a CBR system. In particular they use the DBpedia ontology to retrieve information and similarities of diseases to be used in a system that is concerned with the prevention, management, and research of health problems associated with travel, and covers all medical aspects a traveler has to take care of before, during, and after a journey. DBpedia elements are queried and used in the construction of cases as well as when existing cases are compared against new problems. Their goal is to achieve better semantic similarity of the results and simplify the task of manually filling the case base with existing knowledge.

Case-based reasoning has also been applied to facilitate job runtime estimation [13.13, 58]. In this work, past performance acts as a good indicator for job

scheduling optimization in a grid environment and a CBR system is used to predict the runtime of long-term applications in heterogeneous systems. Using the TA3 CBR system, *Xia et al.* [13.13] have investigated job characteristics and ranked them according to their runtime statistics; i.e., those characteristics with low runtime standard deviation are ranked higher. In a similar fashion they rank machine characteristics and build cases that are stored in the CBR system. Novel similarity measures for job and machine characteristics are defined to be used when new runtimes need to be predicted.

13.3.1 The TA3 CBR System

The last application mentioned above uses the TA3 case-based reasoning system [13.59]. TA3 represents cases as attribute–value pairs whose domains are defined in what is called a case description. More formally, a case C is represented as

$$C = \langle a_0 : V_0 \rangle, \langle a_1 : V_1 \rangle, \dots, \langle a_n : V_n \rangle,$$

where a_i represent attributes and V_i their corresponding values, $0 \leq i \leq n$. Given this representation, a case for which we already know the solution is represented by C_{source} while a case that constitutes a new problem is denoted by C_{input} . Finally, a set of source cases $\{C_{\text{source}}^1, C_{\text{source}}^2, \dots, C_{\text{source}}^k\}$ constitutes a case base, i.e., the search space of cases in the CBR system.

There are three classes of data defined in a case description:

1. Description: the nonpredictive data,
2. Problem: the predictive data,
3. Solution: the classification, diagnosis, or outcome.

Focusing on the problem class, attributes are grouped into categories. The advantage of grouping attributes is that it allows the assignment of different constraints and priorities depending on the relevance of an attribute or collection of attributes (i.e., their value in matching similar cases). During retrieval an explicit context is used for similarity assessment, and the process is guided by incremental transformations of the context. A context is simply a subset of the problem part of the case description with constraints applied to the attribute–value pairs. More formally, a context T is defined by a finite set of attributes and related constraints on their values

$$T = \langle a_0 : CV_0 \rangle, \langle a_1 : CV_1 \rangle, \dots, \langle a_m : CV_m \rangle,$$

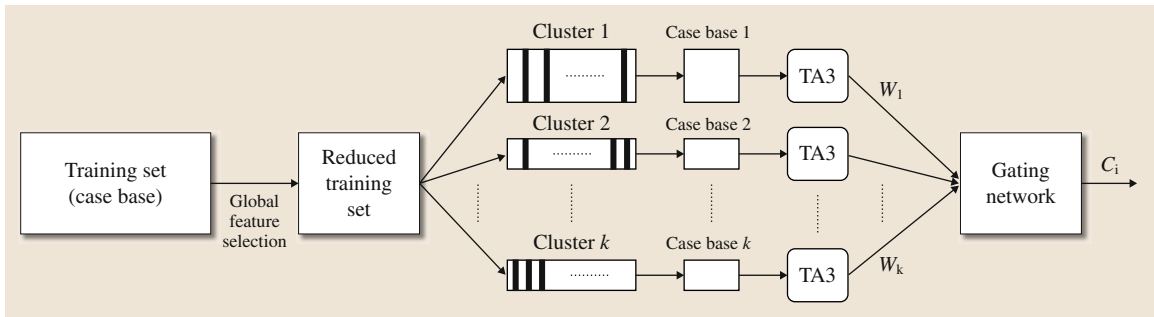


Fig. 13.2 The E4CBR system (after [13.8])

where a_i denotes the attributes and CV_i is the set of values that the attribute can have. This minimizes the effect that irrelevant or less relevant attributes may have when trying to match similar cases. Category membership can be assigned either by an expert with domain knowledge of the relevance of different attributes or by a machine-learning approach.

The retrieval process uses modified nearest-neighbor matching: predictive attributes are grouped to allow different priorities and constraints, an explicit context is used during similarity assessment, and the retrieval algorithm is guided by incremental transformations of the context. A context is simply a subset of the problem part of the case description with constraints applied to the attribute–value pairs.

The flexible nature of *TA3* lies in the fact that retrieval strategies can be dynamically defined for a particular domain and the specific application. The system also includes a genetic algorithm for knowledge discovery purposes; given two or more test sets representing different classes of cases, this functionality maximizes the distances between different classes and minimizes the distances within the same class. The distance between two cases is defined as the amount of relaxations needed to make the two cases similar. The information gained by this process may not only determine previously unknown relations in the data, but may provide a new context with which to guide the retrieval process with greater prediction accuracy.

The *TA3* system uses a structured query language (SQL) database to store the case base and thus handles very large inputs; For instance, it is being used to store and analyze protein crystallization experiments. There are 12 000 experiments, each of which has 9216 attributes, and the data are derived from 110, 592, 000 images. The repository grows at a rate of 200 experiments per month [13.41]. Before *TA3* can suggest crystallization strategies for a new protein, we need to

compute 12 375 features from the images and classify them into 10 categories [13.60].

Finally, the *TA3* system can also be used as a classifier. The attribute–value pairs can be accompanied by class labels, and nearest-neighbor techniques can be modified to classify new cases. To extend the capabilities of *CBR*, *Arshadi* and *Jurisica* [13.8] implemented E4CBR, where an ensemble of *CBR* classifiers is combined with clustering and feature selection. A set of case features is selected first, and then clustering of the cases into disjoint groups is employed, where each group of cases forms the case base of one of the member classifiers (Fig. 13.2). In each case base a subset of features is *locally* selected individually. To predict the label of an unseen case, each classifier in the ensemble provides a prediction, and the aggregation component of E4CBR combines the predictions by weighting each classifier using a *CBR* approach; a classifier with more cases similar to the test case receives a higher weight.

Similarly, *Spasic* et al. [13.61] have proposed MaStErClass, a system that classifies terms by using natural-language processing (NLP) to perform feature selection, comparison of terms, and classification in order to update existing semantic networks with new terms. More precisely, they used a string-based similarity measure called *edit distance* (*ED*) to find similarities between individual terms and keyphrases. Intuitively, *ED* counts the insertions, deletions, and updates that should be performed to convert one string (or keyphrase) to another [13.7]. In addition to *ED*, they incorporate a tree similarity measure to take into account the similarity of corresponding terms in a semantic hierarchy or ontology. This measure is defined as

$$ts(C_1, C_2) = \frac{2 \cdot \text{common}(C_1, C_2)}{\text{depth}(C_1) + \text{depth}(C_2)},$$

where C_1 and C_2 are the two given concepts, while

- $\text{common}(C_1, C_2)$ denotes the number of common concepts in the paths from the root to C_1 and C_2
- $\text{depth}(C_i)$ denotes the number of concepts from the root to C_i , $i = 12$.

13.4 Case-Based Reasoning in Biomedicine

CBR systems in health science include systems with tasks in diagnosis (SHRINK [13.62], Protos [13.63], CASEY [13.64], MEDIC [13.65], BOLERO [13.66]), assessment test planning and clinical research planning (MNAOMIA [13.67]), tutoring (CADI [13.68]), and image analysis (MacRad [13.69], ImageCreek [13.70]).

Nilsson and Sollenborn provide a classification of developments in systems applied in the medical domain, according to their purpose [13.71]. They observe that the majority of these systems belong to the areas of *diagnosis*, *classification*, and *planning*. At the same time, systems such as TA3, which we described in Sect. 13.3.1, are *domain independent* and may be applied in the medical domain.

13.4.1 The eXit*CBR System

Several domain-independent CBR systems have been proposed, including eXit*CBR [13.48]. eXit*CBR is a framework that supports the development and experimentation with CBR systems in general. The system supports a classification component tailored to diagnostic tasks. According to the authors, the advantage of using eXit*CBR lies in the fact that it is modular and helps the user to preprocess the data and visualize the results of different parameter settings. The system includes the following components:

- *Experiment interpreter*: This is the core of the framework that interprets a configuration file given by the user and applies one of the two methods that the system currently supports: *batch processing* and *cross-validation*. In the first method the interpreter reads a set of training data, generates a case base, and uses a different test dataset to obtain the results. Results are interpreted according to a performance measure given as input by the user. The cross-validation method performs multiple runs of a CBR configuration with different datasets (split into training and test data). The performance of the system is averaged at the end of the processing.
- *Preprocessor*: The system accepts a configuration file and the type of preprocessing to be performed. Currently, it supports discretization, normalization, and feature selection.
- *Postprocessor*: This module starts when the CBR engine has finished and involves the application of performance measures that the user is interested in. It also specifies the type of visualization to be used for interpreting the results. Default visualization involves receiver operating characteristic (ROC) curves for the diagnosis system, while it can also calculate an area under curve (AUC) value. The latter measures the accuracy under a ROC curve value and assesses the accuracy of the results. The closer the results are to a value of 1.0, the better the accuracy is.

The system can also assist in the generation of datasets, which can be created according to different user preferences.

13.4.2 Diagnosis with CBR Systems

Other CBR frameworks that have been used in the medical domain are jCOLIBRI [13.72] and MyCBR [13.73]. The former is similar to eXit*CBR in that it is modular, but instead of configuration files, it incorporates an ontology to be used in the different phases of CBR. The ontology has been developed as a plug-in to the PROTEGE (<http://protege.stanford.edu/>) editor for ontologies. In this system, ontologies are used to define properties of the features to be used.

The eXit*CBR tool has been used in a breast cancer diagnosis scenario. Data from patients and healthy population are stored in a database. The 1199 attributes represent habits, such as smoker or not, sport activities and eating, disease characteristics, such as type and size of tumor, as well as the gynecological history of the women involved. The authors describe their experiments with the eXit*CBR tool when it is given different input parameters, such as the number of cases to be retrieved or the number of attributes to be used [13.48]. They also demonstrate how small changes in the initial configuration and preprocessing help in rapid visualization and assessment of the results.

Breast cancer decision support has also been addressed in the KASIMIR system [13.74]. The focus of this work is a methodology that adopts knowledge from experts. The authors describe adaptation patterns such

as cases of inapplicable decisions and consequences of particular decisions. They also take into account missing data during the retrieval phase. Clusters also help in the discovery of patterns, and they have been used in recent work on the diagnosis of melanoma by *Armengol* [13.75]. In this work explanations produced by the system are used to describe clusters and these explanations become part of the system's domain theory, which can be valuable to domain experts.

The *TA3 CBR* system has been used in two medical diagnostic case studies, one in attention-deficit hyperactivity disorder (*ADHD*) and one in stroke diagnosis. *ADHD* is a problem with symptoms of inattentiveness, overactivity, impulsivity, or a combination thereof. It is a neuropsychiatric disorder that appears in both children and adults of different ages. One of the problems with *ADHD* is the lack of objective tests for its proper diagnosis. *Brien et al.* [13.20] have tackled this problem by proposing a methodology that incrementally improves a *CBR* system, namely the *TA3* system that we described above.

Standard diagnoses of *ADHD* usually include interviews with parents and teachers as well as rating scales of hyperactivity and impulsivity, clinical history, cognitive assessments, and neurological examinations. When other disorders are present, diagnosis of *ADHD* becomes harder and the aforementioned tests less accurate. Therefore, their validity and reliability are questionable.

Brien et al. explore a very specific symptom that can lead to more accurate diagnosis of *ADHD*, that of *saccadic eye movement*. Saccades are rapid movements of the eye that bring new visual targets onto the fovea of the retina. They take place either consciously or automatically as a response to stimuli that appear in front of the person. *ADHD* can be diagnosed through saccadic eye movements since the regions in the brain responsible for controlling them are well understood and related to the regions that cause *ADHD*. The data collected for each subject in the saccade tasks include the diagnosis group (*ADHD* or control), any drugs that were used, the age, sex, handedness, hyperactivity, impulsivity, and saccadic reaction time (*SRT*) [13.20]. Two particular tasks, i.e., the prosaccade and antisaccade tasks, have been used to investigate whether the eye movement was performed voluntarily or as a response to stimuli. In both tasks the subjects are looking at fixed objects called fixation points (*FP*). A new visual target (*T*) appears to the left or right side of the fixation point, and the subject is asked to look toward the target (prosaccade) or away from it (antisaccade). The sac-

| Description | | | | |
|-------------------|-------------------|-------|----------------|----------|
| Subject code: aba | Hyperactivity: 87 | | | |
| Handedness: 10 | Impulsivity: 76 | | | |
| Problem | | | | |
| Priority 0 | | | | |
| Age: 8 | Sex: Male | | | |
| Task variables | | | | |
| Task | Mean SRT (ms) | CV | Dir. error (%) | Exp. (%) |
| Anti/Gap/Left | 363.00 | 39.35 | 93.33 | 0.00 |
| Anti/Gap/Right | 458.00 | 50.01 | 75.76 | 12.50 |
| Anti/Over/Left | 351.67 | 42.07 | 77.78 | 16.67 |
| Anti/Over/Right | 483.33 | 44.50 | 62.50 | 0.00 |
| Pro/Gap/Left | 338.00 | 59.82 | 12.50 | 7.14 |
| Pro/Gap/Right | 343.06 | 58.70 | 5.88 | 9.38 |
| Pro/Over/Left | 431.50 | 46.88 | 3.45 | 7.14 |
| Pro/Over/Right | 471.24 | 44.73 | 9.38 | 0.00 |
| Solution | | | | |
| Diagnosis: ADHD | | | | |

Fig. 13.3 Example case description from model 1. *SRT* = Saccadic reaction time. *CV* = coefficient of variation in *SRT*. *Dir. error* = percentage of direction error. *Exp.* = percentage of express saccades. *Anti* = antisaccade task. *Pro* = prosaccade task. *Over* = overlap condition. *Gap* = gap condition (after [13.20])

cadic reaction time (*SRT*) is measured during these tests and compared against controls to make a more objective decision about *ADHD*.

An example of the results from the *TA3* system is given in Fig. 13.3.

The data used involve saccadic eye movement from adults and children. In both categories, measurements of people with *ADHD* are compared against controls, and the accuracy of the results is 72% for children and 76% for adults. The results of the overall study using *TA3* are comparable to those of the continuous performance test (*CPT*) [13.76], the most objective clinical laboratory test for assessing attention and vigilance.

In the diagnosis and prognosis of stroke, conventional assessment techniques are carried out by clinicians, who attempt to measure the degree of impairment with subjective measures. Normally, an early diagnosis is made by assessing the symptoms, reviewing medical history, conducting tests to confirm the

occurrence of a brain attack, and measuring the degree of impairment. Conventional stroke assessment scales convert motor status to a score in an ordinal (nonnumeric) scale. Typically, each patient performs a certain task where the main emphasis is laid on task completion rather than specific details. Therefore it is a nonqualitative scoring. In qualitative scoring, other factors are considered as well, such as measurement of the amount of assistance required, alteration in the normal (gross) position, and time utilized to complete a test.

CBR can be used to create a repository of data from stroke patients who have an explicit diagnosis and prognosis and are receiving subsequent rehabilitation. For a new stroke patient, whose diagnosis is yet to be confirmed and who has an indefinite prognosis, CBR retrieves similar cases from the case base, which may provide useful information to the clinicians, hence facilitating them in reaching a potential solution for stroke diagnosis [13.21]. The work by Baig exploits data that are collected via the kinesiological instrument for normal and altered reaching movement (KINARM) [13.77], a robotic device that monitors and manipulates upper body movements. It records quantifiable kinetic/kinematic measures, such as reaction time, velocity, joint torque, and hand trajectories of both stroke and control subjects for specific motor and sensory tasks. The stroke diagnosis framework involving KINARM and the TA3 CBR systems is shown in Fig. 13.4. After collecting stroke patient data using KINARM, the data are preprocessed to find errors and perform feature selection using data mining through the WEKA tool [13.78]. WEKA also classifies the data into three contexts, A, B, and C, which are defined on dif-

ferent sets of attributes and will be used for retrieval. The data are then stored in a persistent database management system from which cases of known problems, symptoms, and outcomes are extracted and passed to the TA3 CBR system. The final stage is the decision regarding the stroke diagnosis and prognosis of patients. A detailed list of all the attributes used in the case base can be found in [13.21].

Given a new problem (data from a new patient) the case is first classified into one of the three contexts and then compared against the known cases. The main objectives when experimenting with stroke data and the CBR system are:

- To differentiate a *stroke* subject from a *control*,
- To classify the type of stroke as *hemorrhagic* or *ischemic*,
- To classify the stroke subject as *right-brain affected* or *left-brain affected*,
- To determine the prognosis of a stroke patient in terms of *affected vascular territory* and identify the *lesion location*.

The evaluation of the system showed good performance in terms of sensitivity (51%), specificity (98%), and accuracy (82%) [13.21]. A sensitivity of 100% would mean that the test recognizes all sick people as such, whereas a specificity of 100% would mean that the test recognizes all healthy people as healthy.

13.4.3 Medical Imaging and CBR Systems

All the aforementioned systems handle textual and numerical data in medical diagnosis applications. CBR systems have also been found useful at handling bioinformatics data that have been produced as the result of image analysis. Tasks such as interpretation, classification of images, and planning of experiments have been guided by their use; For example, the ImageCreek system [13.70] dealt with the problem of image interpretation. More precisely, this system was used to interpret computer tomography (CT) images, which can then be used in disease diagnosis. The ImageCreek system includes two case-base reasoners, one for segment identification, called *Segment ImageCreek*, and one for image interpretation, called *Wholistic ImageCreek*. This image segment-based CBR system has been used in a study of abdominal CT images that come with a set of hypotheses. The system uses these hypotheses to interpret new images or, according to domain expertise, change the existing ones.

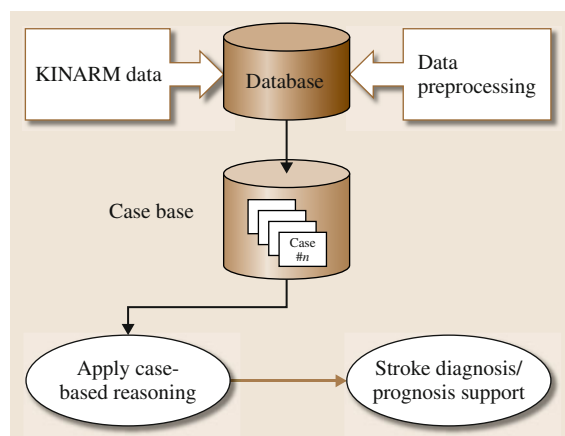


Fig. 13.4 CBR system for stroke patient data

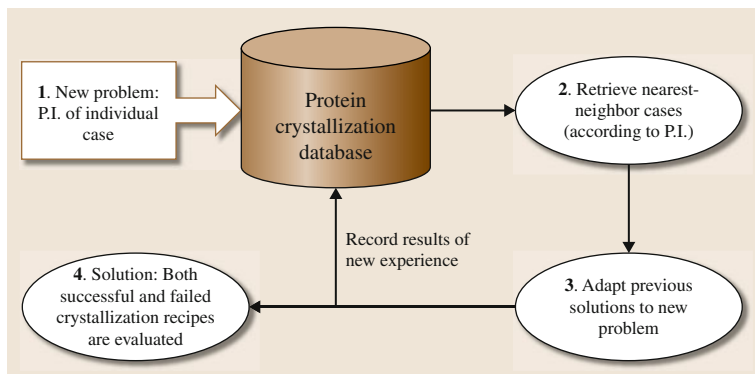


Fig. 13.5 Case-based planning of protein crystallization (P.I. – precipitation index)

13.4.4 Protein Crystallization Planning with CBR

Images are also useful for understanding how proteins acquire their three-dimensional structure, and protein crystallization techniques assist in this task. The fragile nature of crystals as well as the many environmental factors make the process of protein crystallization inherently difficult. Therefore, attempting to crystallize a protein without a proven protocol can be challenging and time consuming. CBR systems have been proposed to aid in the planning of protein crystallization experiments, since one of the difficulties in planning crystal growth experiments is that the history of experiments is not well known [13.80].

Figure 13.5 illustrates the process of using a CBR system in the planning of protein crystallization experiments, where solubility experiments give a quantitative score of similarity for the proteins. Hence, planning strategies that are used in one protein could also be ap-

plied to another. New crystallization problems can then be approached by execution and analysis of a set of precipitation reactions, followed by automated identification of similar proteins and analysis of the recipes used to crystallize them (that is, crystal growth method, temperature and pH ranges, concentration of a protein, crystallization agent, etc.) [13.79].

Case retrieval involves a modified k -nearest-neighbor similarity matching that compares the precipitation indexes of an existing case with a new problem that is given as input [13.79, 81]. Adaptation follows as a next step, and previous solutions are modified to address the new problem. Combined with domain knowledge, the system acts both as an adviser to the crystallographer by suggesting parameter settings for further experiments, as well as an evaluator of potential experiments that the user might propose. The adaptation module constitutes a dynamic process that evolves over time as new knowledge becomes available. Once a plan (in the form of a set of experiments) has been derived

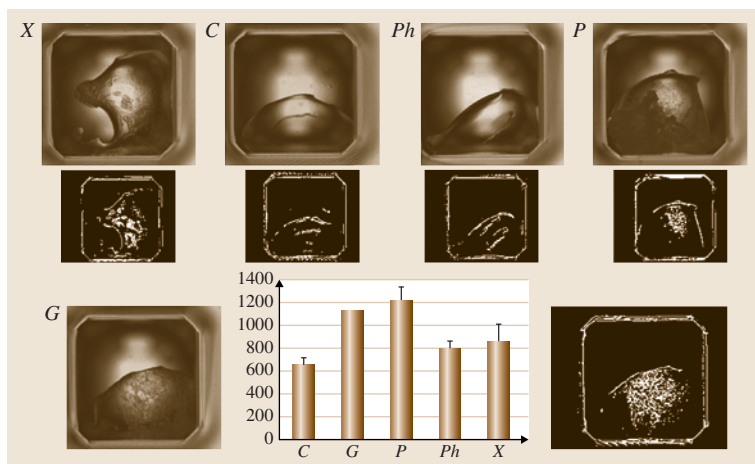


Fig. 13.6 Image analysis and classification (for explanation see text, after [13.79])

Table 13.1 Domain-independent CBR systems

| System | Case representation | Preprocessing | New case processing |
|----------|--|--|-----------------------------|
| TA3 | Attribute–value pairs | Feature selection | k -NN ^a search |
| eXIT*CBR | CSV ^b files of attribute descriptions and weights | Discretization, normalization, feature selection | Case classification |

^a k -NN: k -nearest neighbor; ^b CSV: comma-separated values

Table 13.2 Different biomedical tasks in CBR systems

| Condition/experiment | Type | System | Case comparison |
|-------------------------|------------------------|----------------------|--|
| Breast cancer | Diagnosis Prognosis | eXIT*CBR, KASIMIR | Preclassified cases are compared with new ones |
| ADHD | Diagnosis | TA3 | Similarity of saccade eye movement measurements |
| Stroke | Diagnosis | TA3 | Similarity of kinetic/kinematic measures from KINARM |
| Image analysis | Segmentation | ImageCreek | Existing CT scan hypotheses compared against new |
| Protein crystallization | Planning | TA3 | k -NN compares precipitation indexes of existing/new cases |

and executed for a novel protein, the results are recorded as a new case that reflects this experience. Cases with both positive and negative outcomes are equally valuable for future decision-making processes and are also required for the application of data-mining techniques to the case base.

Figure 13.6 shows example results of image analysis and classification. After performing image segmentation, multiple classes of crystallization results can be detected. The figure demonstrates example images that have been classified as crystal (X), phase separation

(Ph), precipitate (P), clear (C), and gel (G). The corresponding contour images have been utilized to compute the Euler number, which, in turn, has been utilized to cluster similar images. The bar graph also shows that crystal and phase separation overlap, but can be separated from clear drops and precipitates. The results have been shown to have 89% accuracy [13.79, 82].

Finally, for a summary of the two main domain-independent CBR systems discussed, as well as the different biomedical cases where they are being used, see Tables 13.1 and 13.2.

13.5 Conclusions

This chapter presents an overview of CBR systems and their use. We discuss their architecture and the process they follow to solve new problems as well as the challenges involved. Especially in healthcare applications, CBR systems can help solving problems that would otherwise be too difficult to manage using other methods and techniques.

The systems and applications we review illustrate the different roles that CBR systems can play, from diagnosis and prognosis of diseases to image classification and experiment planning. CBR systems can play a significant role in supporting medical decisions nowadays. As the available real-life datasets appear in many different sources and formats, such as data coming from sensors, image, and video as well as data related to semantic properties of entities, e.g., ontologies, CBR systems have been adapted to fit these diverse needs. New data characteristics call for new data preprocessing

techniques, new proximity measures, as well as aggregation and evaluation of results.

Due to the size and heterogeneity of data sources, there is a trend for integration of CBR systems with existing knowledge discovery tools in order to improve the data quality of the cases and their efficiency. Recently CBR has been used in text mining, information retrieval, and natural-language processing. There are also problems in which CBR integrates existing semantic information as part of the domain, which can help address entity resolution problems as well as semantic similarity among new and past experiences.

Finally, new CBR systems facilitate experimentation with different datasets and parameter settings, which allow more effective quality assessment of the results. Combined with advanced visualization techniques, CBR systems can improve healthcare delivery by optimizing decision-support processes.

References

- 13.1 M.N. Nguyen, J.M. Zurada, J.C. Rajapakse: Toward better understanding of protein secondary structure: Extracting prediction rules, *IEEE/ACM Trans. Comput. Biol. Bioinforma.* **8**, 858–864 (2011)
- 13.2 Z. Zhu, S. Sun, M. Bern: Classification of protein crystallization imagery, *IEEE EMBS Annu. Conf.* (2004) pp. 1628–1631
- 13.3 A. Aamodt, E. Plaza: Case-based reasoning: Foundational issues, methodological variations, and system approaches, *AI Commun.* **7**(1), 39–59 (1994)
- 13.4 I. Jurisica, D. Wigle: *Knowledge Discovery in Proteomics* (Chapman Hall, London 2004)
- 13.5 J. Kolodner: *Case-based Reasoning* (Morgan Kaufmann, New York 1993)
- 13.6 S.V. Pantazi, J.F. Arocha, J.R. Moehr: Case-based medical informatics, *BMC Med. Inform. Decis. Mak.* **4**, 1–23 (2004)
- 13.7 R. Baeza-Yates, B. Ribeiro-Neto: *Modern Information Retrieval* (Addison-Wesley-Longman, Amsterdam 1999)
- 13.8 N. Arshadi, I. Jurisica: An ensemble of case-based classifiers for high-dimensional biological domains, *Int. Conf. Case-Based Reason.* (2005) pp. 21–34
- 13.9 N. Arshadi, I. Jurisica: Data mining for case-based reasoning in high-dimensional biological domains, *IEEE Trans. Knowl. Data Eng.* **17**(8), 1127–1137 (2005)
- 13.10 Z.-J. Huang, B.-Q. Wang: A novel swarm clustering algorithm and its application for CBR retrieval, 2nd *Int. Conf. Inf. Eng. Comput. Sci. (ICIECS)* (2010) pp. 1–5
- 13.11 S. Montani: How to use contextual knowledge in medical case-based reasoning systems: A survey on very recent trends, *Artif. Intell. Med.* **51**(2), 125–131 (2011)
- 13.12 S.J. Russell: *Use of Knowledge in Analogy and Induction* (Morgan Kaufmann, New York 1989)
- 13.13 E. Xia, I. Jurisica, J. Waterhouse, V. Sloan: Run-time estimation using the case-based reasoning approach for scheduling in a grid environment, *Int. Conf. Case-Based Reason.* (2010) pp. 525–539
- 13.14 Q. Yang, J. Wu: Keep it simple: A case-base maintenance policy based on clustering and information theory, *Can. Conf. AI* (2000) pp. 102–114
- 13.15 A. Lawanna, J. Daengdej: Methods for case maintenance in case-based reasoning, *Int. J. Comput. Inform. Eng.* **4**, 10–18 (2010)
- 13.16 N. Arshadi, I. Jurisica: Maintaining case-based reasoning systems: A machine learning approach, *Eur. Conf. Case-Based Reasoning* (2004) pp. 17–31
- 13.17 K.-D. Althoff, R. Bergmann, S. Wess, M. Manago, E. Auriol, O.I. Larichev, A. Bolotov, Y.I. Zhuravlev, S.I. Gurov: Case-based reasoning for medical decision support tasks: The INRECA approach, *Artif. Intell. Med.* **12**(1), 25–41 (1998)
- 13.18 M. Frize, R. Walker: Clinical decision-support systems for intensive care units using case-based reasoning, *Med. Eng. Phys.* **22**, 671–677 (2000)
- 13.19 I. Jurisica, J. Mylopoulos, J.I. Glasgow, H. Shapiro, R.F. Casper: Case-based reasoning in IVF: Prediction and knowledge mining, *Artif. Intell. Med.* **12**(1), 1–24 (1998)
- 13.20 D. Brien, J.I. Glasgow, D. Munoz: The application of a case-based reasoning system to attention-deficit hyperactivity disorder, *Int. Conf. Case-Based Reason.* (2005) pp. 122–136
- 13.21 M. Baig: *Case-Based Reasoning: An Effective Paradigm for Providing Diagnostic Support for Stroke Patients*, Master Thesis (Queen's University, Kingston 2008)
- 13.22 R.G. Ross, D. Hommer, D. Breiger, C. Varley, A. Radant: Eye movement task related to frontal lobe functioning in children with attention deficit disorder, *J. Am. Acad. Child Adolesc. Psychiatry* **33**, 869–874 (1994)
- 13.23 A. Aamodt: Modeling the knowledge contents of CBR systems, *Int. Conf. Case-Based Reason.* (2001) pp. 32–37, *Naval Res. Note AIC-01-003*
- 13.24 P. Cunningham, A. Bonzano: Knowledge engineering issues in developing a case-based reasoning application, *Knowl. Based Syst.* **12**(7), 371–379 (1999)
- 13.25 N. Xiong, P. Funk: Combined feature selection and similarity modelling in case-based reasoning using hierarchical memetic algorithm, *IEEE Congr. Evol. Comput.* (2010) pp. 1–6
- 13.26 L. Gazendam, C. Wartena, R. Brussee: Thesaurus based term ranking for keyword extraction, *DEXA Workshops* (2010) pp. 49–53
- 13.27 Y. Matsuo, M. Ishizuka: Keyword extraction from a single document using word co-occurrence statistical information, *Int. J. Artif. Intell. Tools* **13**(1), 157–169 (2004)
- 13.28 R. Kohavi, G.H. John: Wrappers for feature subset selection, *Artif. Intell.* **97**(1–2), 273–324 (1997)
- 13.29 H. Peng, F. Long, C.H.Q. Ding: Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy, *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(8), 1226–1238 (2005)
- 13.30 K. Gopal: Efficient case-based reasoning through feature weighting, and its application in protein crystallography. Ph.D. Thesis (Texas A & M University, College Station 2007)
- 13.31 C. Kirsopp, M.J. Shepperd, J. Hart: Search heuristics, case-based reasoning and software project effort prediction, *Genet. Evol. Comput. Conf.* (2002) pp. 1367–1374

- 13.32 I. Guyon, A. Elisseeff: An introduction to variable and feature selection, *J. Mach. Learn. Res.* **3**, 1157–1182 (2003)
- 13.33 P.C. Boutros, S.K. Lau, M. Pintilie, N. Liu, F.A. Shepherd, S.D. Der, M.-S. Tsao, L.Z. Penn, I. Jurisica: Prognostic gene signatures for non-small-cell lung cancer, *Proc. Natl. Acad. Sci. USA* **106**(8), 2824–2828 (2009)
- 13.34 K. Börner, E. Pippig, E.-C. Tammer, C.-H. Coulon: Structural similarity and adaptation, *Eur. Winter Conf. Brain Research* (1996) pp. 58–75
- 13.35 E.L. Rissland, J.J. Daniels: The synergistic application of CBR to IR, *Artif. Intell. Rev.* **10**(5–6), 441–475 (1996)
- 13.36 M. Lenz, H.-D. Burkhard, P. Pirk, E. Auriol, M. Manago: CBR for diagnosis and decision support, *AI Commun.* **9**(3), 138–146 (1996)
- 13.37 Y. Niu, D. Otasek, I. Jurisica: Evaluation of linguistic features useful in extraction of interactions from PubMed; Application to annotating known, high-throughput and predicted interactions in I²D, *Bioinformatics* **26**(1), 111–119 (2010)
- 13.38 P. Andritsos, P. Tsaparas, R.J. Miller, K.C. Sevcik: LIMBO: Scalable clustering of categorical data, *Eur. Conf. Case-Based Reasoning* (2004) pp. 123–146
- 13.39 P. Andritsos, R.J. Miller, P. Tsaparas: Information-theoretic tools for structure discovery in large data sets, *SIGMOD* (2004) pp. 731–742
- 13.40 M. Gluck, J. Corter: Information, uncertainty, and the utility of categories, *Proc. 7th Annu. Conf. Cogn. Sci. Soc. (COGSCI)* (1985) pp. 283–287
- 13.41 C.A. Cumbaa, I. Jurisica: Protein crystallization analysis on the world community grid, *J. Struct. Funct. Genomics* **11**, 61–69 (2010)
- 13.42 P. Perner: Similarity-based image segmentation for determination of brain/liquor ratio in CT image by Alzheimer dementia, *Bildverarb. für die Med.* (1998)
- 13.43 G. Xu, C. Chiu, E.D. Angelini, A.F. Laine: An incremental and optimized learning method for the automatic classification of protein crystal images, *IEEE EMBS Annu. Int. Conf.* (2006) pp. 6526–6529
- 13.44 K. Kawabata, M. Takahashi, K. Saitoh, H. Asama, T. Mishima, M. Sugahara, M. Miyano: Evaluation of crystalline objects in crystallizing protein droplets based on line-segment information in greyscale images, *Acta Crystallogr. D* **62**, 239–245 (2006)
- 13.45 K. Saitoh, K. Kawabata, H. Asama, T. Mishima, M. Sugahara, M. Miyano: Evaluation of protein crystallization states based on texture information derived from greyscale images, *Acta Crystallogr. D* **61**, 873–880 (2005)
- 13.46 S. Pan, G. Shavit, M. Penas-Centeno, D.-H. Xu, L. Shapiro, R. Ladner, E. Riskin, W. Hol, D. Meldrum: Automated classification of protein crystallization images using support vector machines with scale-invariant texture and gabor features, *Acta Crystallogr. D* **62**, 271–279 (2006)
- 13.47 J. Wilson: Automated Classification of Images from Crystallization Experiments, *Ind. Conf. Data Min.* (2006) pp. 459–473
- 13.48 B. López, C. Pous, P. Gay, A. Pla, J. Sanz, J. Brunet: eXIT*CBR: A framework for case-based medical diagnosis development and experimentation, *Artif. Intell. Med.* **51**(2), 81–91 (2011)
- 13.49 M. Göker, C. Baudin, M. Manago: Development of industrial knowledge management applications with case-based reasoning. In: *Successful Case-Based Reasoning Applications*, ed. by S. Montani, L. Jain (Springer, Berlin, Heidelberg 2010) pp. 53–82
- 13.50 C.L. Cobb, A.M. Agogino: Case-based reasoning for evolutionary mems design, *J. Comput. Inform. Sci. Eng.* **10**(3), 39–48 (2010)
- 13.51 K.J. Kim, K. Kim: Preliminary cost estimation model using case-based reasoning and genetic algorithms, *J. Comput. Civ. Eng.* **24**(6), 499–505 (2010)
- 13.52 C.-H. Leon Lee, A. Liu, H.-H. Huang: Using planning and case-based reasoning for service composition, *J. Adv. Comput. Intell. Intell. Inform.* **14**, 540–548 (2010)
- 13.53 M. Tingyu, M. Biao: Case based reasoning applied in personal financing: Representing cases based on XML, *J. Adv. Comput. Intell. Intell. Inform.* **14**, 540–548 (2010)
- 13.54 A. Wyner, T. Bench-Capon: Argument schemes for legal case-based reasoning, *Proc. 2007 Conf. Leg. Knowl. Inform. Syst.* (2007) pp. 139–149
- 13.55 M.U. Ahmed, S. Begum: Case-based reasoning for medical and industrial decision support. In: *Successful Case-Based Reasoning Applications*, ed. by S. Montani, L. Jain (Springer, Berlin, Heidelberg, 2010) pp. 7–52
- 13.56 M. d’Aquin, J. Lieber, A. Napoli: Decentralized case-based reasoning for the semantic web, *Int. Seman. Web Conf.* (2005) pp. 142–155
- 13.57 C.S. Sauer, K. Bach, K.-D. Althoff: Integration of linked-open data in case-based reasoning systems, *LWA Workshop* (2010) pp. 269–274
- 13.58 L.N. Nassif, J. Marcos, S. Nogueira, A. Karmouch, M. Ahmed, F. de Andrade: Job completion prediction using case-based reasoning for grid computing environments, *Concurr. Comput. Pract. Exp.* **19**(9), 1253–1269 (2007)
- 13.59 I. Jurisica: *TAB: Theory, implementation and applications of similarity-based retrieval for case-based reasoning*. Ph.D. Thesis (University of Toronto, Toronto 1998)
- 13.60 V.K. Chaudhri, I. Jurisica, M. Koubarakis, D. Plexousakis, T. Topaloglou: The KBMS project and beyond, *Concept. Model.* (2009) pp. 466–482
- 13.61 I. Spasic, S. Ananiadou, J. Tsujii: MaSterClass: A case-based reasoning system for the classification of biomedical terms, *Bioinformatics* **21**(11), 2748–2758 (2005)

- 13.62 J. Kolodner, R.M. Kolodner: Using experience in clinical problem solving: Introduction and framework, *IEEE Trans. Syst. Man Cybern.* **17**, 420–431 (1987)
- 13.63 R. Bareiss, B.W. Porter, C.C. Wier: Protos: An exemplar-based learning apprentice, *Int. J. Man-Mach. Stud.* **29**(5), 549–561 (1988)
- 13.64 P. Koton: Reasoning about evidence in causal explanations, *AAAI* (1988) pp. 256–263
- 13.65 R.M. Turner: Using schemas for diagnosis, *Comput. Methods Programs Biomed.* **30**, 199–207 (1989)
- 13.66 B. López, E. Plaza: Case-based planning for medical diagnosis, *Int. Symp. Methodol. Intell. Syst. (ISMIS)* (1993) pp. 96–105
- 13.67 I. Bichindaritz: A case-based reasoner adaptive to different cognitive tasks, *Int. Conf. Case-Based Reason.* (1995) pp. 390–400
- 13.68 K.D. Fenstermacher: CADI – An intelligent, multimedia tutor for cardiac auscultation, *AAAI/IAAI, Vol. 2* (1996) p. 1387
- 13.69 R.T. Macura, K.J. Macura: MacRad: Radiology image resource with a case-based retrieval system, *Int. Conf. Case-Based Reason.* (1995) pp. 43–54
- 13.70 M. Grimnes, A. Aamodt: A two layer case-based reasoning architecture for medical image understanding, *Eur. Winter Conf. Brain Res.* (1996) pp. 164–178
- 13.71 M. Nilsson, M. Sollenborn: Advancements and trends in medical case-based reasoning: An overview of systems and system development, *Proc. 17th Int. Florida Artif. Intell. Res. Soc. Conf. (AAAI, Menlo Park, 2004)* pp. 178–183
- 13.72 J.A. Recio-García, B. Díaz-Agudo, M.-A. Gómez-Martín, N. Wiratunga: Extending jCOLIBRI for textual CBR, *Int. Conf. Case-Based Reason.* (2005) pp. 421–435
- 13.73 A. Stahl, T. Roth-Berghofer: Rapid prototyping of CBR applications with the open source tool myCBR, *Eur. Conf. Case-Based Reasoning* (2008) pp. 615–629
- 13.74 M. d’Aquin, J. Lieber, A. Napoli: Adaptation knowledge acquisition: A case study for case-based decision support in oncology, *Comput. Intell.* **22**(3–4), 161–176 (2006)
- 13.75 E. Armengol: Classification of melanomas in situ using knowledge discovery with explained case-based reasoning, *Artif. Intell. Med.* **51**(2), 93–105 (2011)
- 13.76 H.E. Rosvold, A.F. Mirsky, I. Sarason, E.D. Bransome Jr., L.H. Beck: A continuous performance test of brain damage, *J. Consult. Clin. Psychol.* **20**(5), 343–350 (1956)
- 13.77 S.H. Scott: Apparatus for measuring and perturbing shoulder and elbow-joint positions and torques during reaching, *J. Neurosci. Methods* **89**, 119–127 (1999)
- 13.78 I.H. Witten, E. Frank, L. Trigg, M. Hall, G. Holmes, S.J. Cunningham: *Weka: Practical Machine Learning Tools and Techniques with Java Implementations* (1999), available online at <http://www.cs.waikato.ac.nz/>
- 13.79 I. Jurisica, J.I. Glasgow: Applications of case-based reasoning in molecular biology, *AI Mag.* **25**(1), 85–96 (2004)
- 13.80 A. Ducruix, R. Giege: *Crystallization of Nucleic Acids and Proteins. A Practical Approach* (Oxford Univ. Press, Oxford 1992)
- 13.81 I. Jurisica, P. Rogers, J.I. Glasgow, R.J. Collins, J.R. Wolfley, J.R. Luft, G.T. DeTitta: Improving objectivity and scalability in protein crystallization: Integrating image analysis with knowledge discovery, *IEEE Intell. Syst.* **16**(6), 26–34 (2001)
- 13.82 I. Jurisica, C.A. Cumbaa, A. Lauricella, N. Fehrman, C. Veatch, R. Collins, J. Luft, G. DeTitta: Automatic classification of protein crystallization screens on 1536-well plates, *Proc. Annu. Conf. Am. Crystallogr. Assoc. (ACA03)* (2003)