P. Perner, Are Case-Based Reasoning and Dissimilarity-based Pattern Recognition two Sides of the same Coin?, In: P. Perner (Eds.) , Machine Laerning and Data Mining in Pattern Recognition, Springer Verlag, lnai 2123, p. 35-51.

Are Case-Based Reasoning and Dissimilarity-Based Classification two Sides of the same Coin?

Petra Perner

Institute of Computer Vision and applied Computer Sciences
Arno-Nitzsche-Str. 45, 04277 Leipzig
e-mail: ibaiperner@aol.com, http://www.ibai-research.de

Abstract. Case-Based Reasoning is used when generalized knowledge is lacking. The method works on a set of cases formerly processed and stored in the case base. A new case is interpreted based on its similarity to cases in the case base. The closest case with its associated result is selected and presented as output of the system. Recently, Dissimilarity-based Classification has been introduced due to the curse of dimensionality of feature spaces and the problem arising when trying to make image features explicitly. The approach classifies samples based on their dissimilarity value to all training samples. In this paper, we are reviewing the basic properties of these two approaches. We show the similarity of Dissimilarity based Classification to Case-Based Reasoning. Finally, we conclude that Dissimilarity based Classification is a variant of Case-Based Reasoning and that most of the open problems in Dissimilarity-based Classification are research topics of Case-Based Reasoning.

1 Introduction

Case-Based Reasoning (CBR) has been developed within the artificial intelligence community. It uses past experiences to solve new problems. Therefore, past problems are stored as cases in a case base and a new case is classified by determining the most similar case from the case base. Although, CBR has been used with great success, for image related applications the examples are rare [1]-[7] and not well known within the pattern recognition community.

Recently, Dissimilarity-based classification (DSC)[8][9] has been introduced within the pattern recognition community. Objects are represented by their dissimilarity value to all objects in the case base. Classification is done based on the dissimilarity values. It is argued that dissimilarity based representations of objects are simpler to access than feature based representations and that this approach helps to comeover the curse of dimensionality of feature spaces.

In this paper, we are reviewing the basic properties of these two approaches. CBR is described in detail in Section 2. DSC is reviewed in Section 3. Finally, we compare these two approaches in Section 4. We show that DSC relies on the same basic idea as

CBR. While CBR has covered all aspects of the development of a CBR system which range from fundamental theory to software engineering aspects, DSC work is very preliminary and does not cover all aspects that make such systems work in practice. Finally, we can conclude that DSC is a special variant of CBR that is influenced by the traditional ideas of pattern recognition.

2 Case-Based Reasoning

Rule-based systems or decision trees are difficult to utilize in domains where generalized knowledge is lacking. However, often there is a need for a prediction system even though there is not enough generalized knowledge. Such a system should a) solve problems using the already stored knowledge and b) capture new knowledge making it immediately available to solve the next problem. To accomplish these tasks case based reasoning is useful. Case-based reasoning explicitly uses past cases from the domain expert's successful or failing experiences.
Therefore, case-based reasoning can be seen as a method for problem solving as well as a method to capture new experiences. It can be seen as a learning and knowledge discovery approach since it can capture from new experiences some general knowledge such as case classes, prototypes and some higher level concepts. The theory and motives behind CBR techniques are described in depth in [10][11][43]. An overview about recent CBR work can be found in [12].
To point out 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, we follow the notion of Wess et al. [13]: A case-based reasoning 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:

$$\boxed{\text{Concept = Case Base + Measure of Similarity}}$$

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 $C = (CB_1, sim_1), (CB_2, sim_2), \ldots, (CB_i, sim_i)$ for the same concept C. Furthermore, the equation gives a hint how a case-based learner can improve its classification ability. There are three possibilities to improve a case-based system. The system can

- store new cases in the case base *CB*,
- change the measure of similarity *sim*,
- or change *CB* and *sim*.

During the learning phase a case-based system gets a sequence of cases $X_1, X_2, \ldots, X_i$ with $X_i = (x_i, class(x_i))$ and builds a sequence of pairs $(CB_1, sim_1), (CB_2, sim_2), \ldots, (CB_i, sim_i)$ with $CB_i \subseteq \{X_1, X_2, \ldots, X_i\}$. 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.

2.1 The Case-Based Reasoning Process

The CBR reasoning process is comprised of six phases (see Figure 1):

- Current problem description
- Problem indexing
- Retrieval of similar cases
- Evaluation of candidate cases
- Modification of selected case, if necessary
- Application to current problem: human action.

The current problem is described by some keywords, attributes or any abstraction that allows describing the basic properties of a case. Based on this a set of close cases description are indexed. The index can be a structure such as for example a classifier or any hierarchical organization of the case base. Among the set of close the closest case cases is determined and is presented as the result of the system. If necessary this case is modified so that it fits to 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 no similar case could be found in case base, then case base management starts.

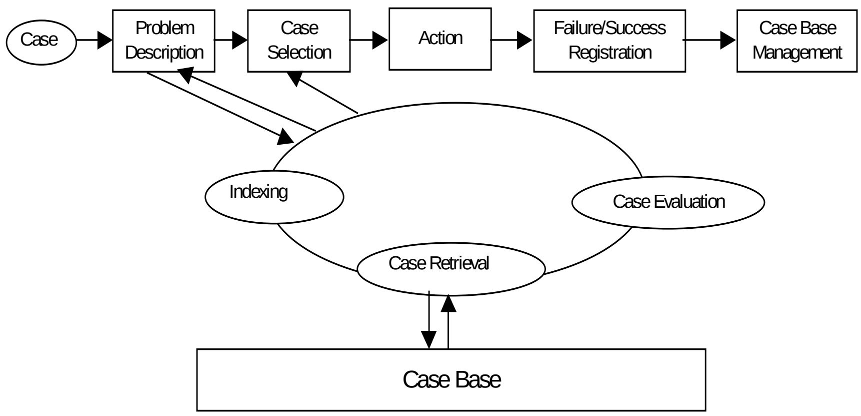

Fig. 1. Case-Based Reasoning Process

2.2 CBR Maintenance

CBR management (see Figure 2) will operate on new cases as well as on cases already stored in case base.

If a new case has to be stored into the case base then it 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 incorporated into the case base in order to close this gap. From the new case has to be extracted a predetermined case description, which should be formatted into the predefined case format. Afterwards the case can be stored into case base.

Selective case registration means that no redundant cases will be stored into case base and that the case will be stored at the right place depending on the chosen organization of the case base. 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 cases would be retrieved from case base that are not applicable to the current problem. Then, 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 sorting out cases that do not apply to the current problem. The weights in the similarity measure might be updated in order to retrieve only a small set of similar cases.

CBR maintenance is a complex process and works over all knowledge containers (vocabulary, similarity, retrieval, case base) [14] of a CBR system. Consequently, architectures and systems have been developed which support this process [7][15][16].

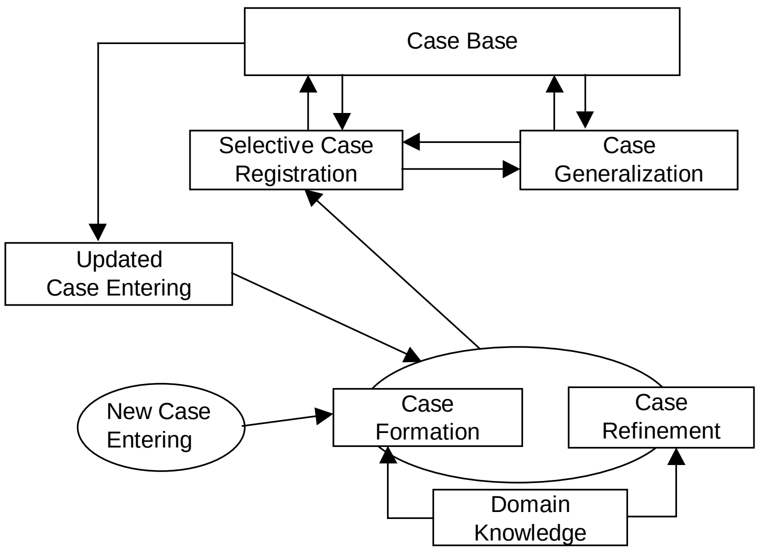

Fig. 2. CBR Maintenance

2.3 Design Consideration

The main problems concerned with the development of a CBR system are:

- What is the right case description?
- 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?

2.4 Case Description

There are different opinions about the formal description of a case. Each system utilizes a different representation of a case. Formally, we like to understand for a case the following definition:

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

For image related tasks, we have two main different types of information that make up a case that are image-related information and non-image related information. Image related information could be the 1D, 2D or 3D images of the desired application. Non-image related information could be information about the image acquisition such as the type and parameters of the sensor, and information about the objects or the illumination of the scene. It depends on the type of application what type of information should be taken into consideration for the interpretation of the image. In case of the medical CT image segmentation described in [3] we used patient-specific parameter such as age and sex, slice thickness and number of slices. Jarmulak [1] took into consideration the type of sensor for the railway inspection application. Based on this information the system controls the type of case base that the system is using during reasoning.

How the 2D or 3D image matrix is represented depends on the purpose and not seldom on the developer's point of view. In principle it is possible to represent an image by one of various abstraction levels. An image may be described by the pixel matrix itself or by parts of this matrix (pixel-based representation). It may be described by the objects contained in the image and their features (feature-based representation). Furthermore, it may be described by a more complex model of the image scene comprising of objects and their features as well as the spatial relation between the objects (attributed graph representation or semantic networks).

Jarmular [1] has solved this problem by a four level hierarchy for a case and different case bases for different sensor types. At the lowest level of the hierarchy are stored the objects described by features such as their location, orientation, and type (line, parabola, or noise) parameters. The next level consists of objects of the same channel within the same subcluster. In the following level the subcluster is stored and at the highest level the whole image scene is stored. This representation allows him to match the cases on different granularity levels. Since the whole scene may have distortions caused by noise and imprecise measurements, he can reduce the influence of noise by retrieving cases on these different level.

Grimnes and Aamodt [2] developed a model based image interpretation system for the interpretation of abdominal CT images. The image content is represented by a semantic network where concepts can be general, special cases or, heuristic rules. Not well understood parts of the model are expressed by cases and can be revised during the usage of the system by the learning component. The combination of the partial well-understood model with cases helps them to overcome the usually burden of modeling. The learning component is based on failure driven learning and case integration. Non-image information is also stored such as sex, age, earlier diagnosis, social condition etc.

Micarelli et. al [4] have also calculated image properties from their images and stored them into the case base. They use the Wavelet transform since it is scale-independent. By doing so they only take into consideration the rotation of the objects in their similarity measure.

In all this work, CBR is only used for the high-level unit. We have studied different approaches for the different processing stages of an image interpretation system. For the image segmentation unit [3], we studied two approaches: 1. a pixel-based approach and 2. a feature-based approach that described the statistical properties of an image. Our results show that the pixel-based approach can give better results for the purpose of image segmentation. For the high-level approach of an ultra sonic image interpretation system, we used a graph-based representation [7].

However, if we do not store the image matrix itself as a case, but we store the representation of a higher-level abstraction instead of, we will lose some information. An abstraction means we have to make a decision between necessary and unnecessary details of an image. It might happen that having not seen all objects at the same time we might think that one detail is not of interest since our decision is only based on a limited number of objects. This can cause problems later on. Therefore, to keep the images themselves is always preferable but needs a lot of storage capacity. The different possible types of representation require different types of similarity measures.

2.5 Similarity

An important point in case-based reasoning is the determination of similarity between a case A and a case B. We need an evaluation function that gives us a measure for similarity between two cases. This evaluation function reduces each case from its case description to a numerical similarity measure *sim*. These similarity measures show the relation to other cases in the case base.

2.5.1 Formalization of Similarity

The problem with similarity is that it has no meaning unless one specifies the kind of similarity.

Smith [17] distinguishes into 5 different kinds of similarity:

- Overall similarity
- Similarity
- Identity
- Partial similarity and
- Partial identity.

Overall similarity is a global relation that includes all other similarity relations. All colloquial similarity statements are subsumed here.

Similarity and identity are relations that consider all properties of objects at once, no single part is left unconsidered. A red ball and a blue ball are similar, a red ball and a red car are dissimilar. The holistic relation's similarity and identity are different in the degree of the similarity. Identity describes objects that are not significantly different. All red balls are similar. Similarity contains identity and is more general.

Partial similarity and partial identity compare the significant parts of objects. One aspect or attribute can be marked. Partial similarity and partial identity are different with respect to the degree of similarity. A red ball and a pink cube are partially similar but a red ball and a red cube are partially identical.

The described similarity relations are in connection with many respects. Identity and similarity are unspecified relations between whole objects. Partial identity and similarity are relations between single properties of objects. Identity and similarity are equivalence relations that mean they are reflexive, symmetrical, and transitive. For partial identity and similarity these relations does not hold. From identity follows similarity and partial identity. From that follows partial similarity and general similarity.

It seems advisable to require from a similarity measure the reflexivity that means an object is similar to itself. Symmetry should be another property of similarity. However, Bayer et. al [18] show that these properties are not bound to belong to similarity in colloquial use. Let us consider the statements "A is similar to B" or "A is the same as B". We notice that these statements are directed and that the roles of A and B can not be exchanged. People say: "A circle is like an ellipse." but not "An ellipse is like a circle." or "The sun looks like the father." but not "The father looks like to the sun.". Therefore, symmetry is not necessarily a basic property of similarity. However, in the above examples it can be useful to define the similarity relation to be symmetrical. The transitivity relation must also not necessarily hold. Let us consider the block world: a red ball and a red cube might be similar; a red cube and a blue square are similar; but a red ball and a blue square are dissimilar. However, a concrete similarity relation might be transitive.

Similarity and identity are two concepts that strongly depend on the context.

The context defines the essential attributes of the objects that are taken into consideration when similarity is determined. An object "red ball" may be similar to an object "red chair" because of the color red. However the object "ball" and "chair" are

dissimilar. These attributes may be relevant depending on weather they are given priority or saliency in the considered problem.
This little example shows that the calculation of similarity between the attributes must be meaningful. It makes no sense to compare two attributes that do not make a contribution to the considered similarity.

Since attributes can be numerical and categorical or a combination of both we need to pay attention to this by the selection of the similarity measure. Not all similarity measures can be used for categorical attributes or can deal at the same time with numerical and categorical attributes.

2.5.2 Similarity Measures for Images

Images can be rotated, translated, different in scale, or may have different contrast and energy but they might be considered as similar. In contrast to that, two images may be dissimilar since the object in one image is rotated by 180 degrees. The concept of invariance in image interpretation is closely related to that of similarity. A good similarity measure should take this into consideration.

The classical similarity measures do not allow this. Usually, the images or the features have to be pre-processed in order to be adapted to the scale, orientation or shift. This process is a further processing step which is expensive and needs some a-priori information which are not always given. Filters such as matched filters, linear filters, Fourier or Wavelet filters are especially useful for invariance under translation and rotation which has also been shown by [4]. There has been a lot of work done to develop such filters for image interpretation in the past. The best way to achieve scale invariance from an image is by means of invariant moments, which can also be invariant under rotation and other distortions. Some additional invariance can be obtained by normalization (reduces the influence of energy).

Depending on the image representation (see Figure 3) we can divide similarity measures into:

- pixel (Iconic)-matrix based similarity measures,
- feature-based similarity measures, (numerical or symbolical or mixed type) and,
- structural similarity measures [18]-[23][34].

Since a CBR image interpretation system has also to take into account non-image information such as about the environment or the objects etc, we need similarity measures which can combine non-image and image information. A first approach to this, we have shown in [3].

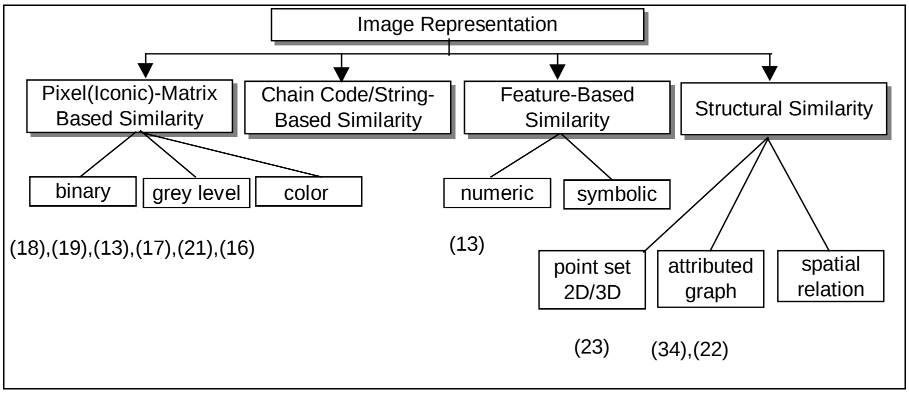

Fig. 3. Image Representations and Similarity Measure

To better understand the concept of similarity systematic studies on the different kinds of image similarity have be done. Zamperoni et. al [19] studied how pixel-matrix based similarity measures behave under different real world influences such as translation, noise (spikes, salt and pepper noise), different contrast and so on. Image feature-based similarity measures have been studied from a broader perspective by Santini and Jain [20]. Those are the only substantiate works we are aware of. Otherwise at every new conference on pattern recognition new similarity measures [21]-[31] are proposed for specific purposes and the different kinds of image representation but it is missing a more methodological work.

2.6 Organization of Case Base

Cases can be organized into a flat case base or in a hierarchical fashion. In a flat organization, we have to calculate similarity between the problem case and each case in memory. It is clear that this will take time even if the case base is very large. Systems with a flat case base organization usually run on a parallel machine to perform retrieval in a reasonable time and do not allow the case base to grow over a predefined limit. Maintenance is done by partitioning the case base into case clusters and by controlling the number and size of these clusters [33].

To speed up the retrieval process a more sophisticated organization of case base is necessary. This organization should allow separating the set of similar cases from those cases not similar to the recent problem at the earliest stage of the retrieval process. Therefore, we need to find a relation *p* that allows us to order our case base:

Definition: A binary relation p on a set CB is called a partial order on CB if it is reflexive, antisymmetric, and transitive. In this case, the pair $\langle CB, p\rangle$ is called a partial ordered set or poset.

The relation can be chosen depending on the application. One common approach is to order the case base based on the similarity value. The set of case can be reduced by the similarity measure to a set of similarity values. The relation <= over these similarity values gives us a partial order over these cases. The derived hierarchy consists of nodes and edges. Each node in this hierarchy contains a set of cases that do not exceed a specified similarity value. The edges show the similarity relation between the nodes. The relation between two successor nodes can be expressed as follows: Let z be a node and x and y are two successor nodes of z then x subsumes z and y subsumes z. By tracing down the hierarchy, the space gets smaller and smaller until finally a node will not have any successor. This node will contain a set of close cases. Among these cases is to find the closest case to the query case. Although, we still have to carry out matching the number of matches will have decreased through the hierarchical ordering. The nodes can be represented by the prototypes of the set of cases assigned to the node. When classifying a query through the hierarchy the query is only matched with the prototype. Depending on the outcome of the matching process, the query branches right or left of the node.

Such kind of hierarchy can be created by hierarchical or conceptual clustering [34], k-d trees [35] and decision trees [1]. There are also set-membership based organizations known, such as semantic nets [2] and object-oriented representations [36].

2.8 Learning in a CBR System

CBR management is closely related to learning. It aims to improve the performance of the system.

Let *X* be a set of cases collected in a case base *CB*. The relation between each case in case base can be expressed by the similarity value *sim*. The case base can be partitioned into *n* case classes *C*: $CB = \bigcup_{i=1}^{n} C_i$ such that the intra case class similarity is high and the inter case class similarity is low. The set of cases in each class *C* can be represented by a representative who generally describes the cluster. This representative can be the prototype, the mediod, or an a-priori selected case. Whereas the prototype implies that the representative is the mean of the cluster which can easily be calculated from numerical data. The mediod is the case whose sum of all distances to all other cases in a cluster is minimal. The relation between the different case classes *C* can be expressed by higher order constructs expressed e.g. as super classes that gives us a hierarchical structure over the case base.
There are different learning strategies that can take place in a CBR system:

1. Learning takes place if a new case x has to be stored into the case base such that: $CB_{n+1} = CB_n \cup \{x\}$. That means that the case base is incrementally updated according to the new case.
2. It may incrementally learn the case classes and/or the prototypes representing the class.
3. The relationship between the different cases or case classes may be updated according the new case classes.
4. The system may learn the similarity measure.

2.8.1 Learning new Cases and Forgetting old Cases

Learning new cases means just adding cases into the case base upon some notification. Closely related to case adding is case deletion or forgetting cases which have shown low utility. This should control the size of the case base. There are approaches that keep the size of the case base constant and delete cases that have not shown good utility within a fixed time window [37]. The failure rate is used as utility criterion. Given a period of observation of N cases, if the CBR component exhibits M failures in such a period, we define the failure rate as $f_r = M / N$. Other approaches try to estimate the "coverage" of each case in memory and by using this estimate to guide the case memory revision process [38].

The adaptability to the dynamic of the changing environment that requires storing new cases in spite of the case base limit is addressed in [33]. Based on intra class similarity is decided whether a case is to be removed from or to be stored in a cluster.

2.8.2 Learning of Prototypes

Learning of prototypes has been described in [39] for flat organization of case base and for hierarchical representation of case base in [34]. The prototype or the representative of a case class is the most general representation of a case class. A class of cases is a set of cases sharing similar properties. The set of cases does not exceed a boundary for the intra class dissimilarity. Cases that are on the boundary of this hyperball have maximal dissimilarity value. A prototype can be selected a-priori by the domain user. This approach is preferable if the domain expert knows for sure the properties of the prototype. The prototype can be calculated by averaging over all cases in a case class or the median of the cases is chosen. If only a few cases are available in a class and subsequently new cases are stored in the class then it is preferable to incrementally update the prototype according to the new cases.

2.8.3 Learning of Higher Order Constructs

The ordering of the different case classes gives an understanding of how these case classes are related to each other. For two case classes which are connected by an edge similarity relation holds. Case classes that are located at a higher position in the hierarchy apply to a wider range of problems than those located near the leaves of the hierarchy. By learning how these case classes are related to each other, higher order constructs are learnt [39].

2.8.4 Learning of Similarity

By introducing feature weights we can put special emphasis on some features for the similarity calculation. It is possible to introduce local and global feature weights. A feature weight for a specific attribute is called local feature weight. A feature weight that averages over all local feature weights for a case is called global feature weight. This can improve the accuracy of the CBR system. By updating these feature weights we can learn similarity [40][41].

3 Dissimilarity-Based Classification

Dissimilarity-based pattern recognition (DSC) [8] - also named featureless classification in earlier papers by the authors [42] - means building classifiers based on distance values. Usually, dissimilarity measures can be transformed into similarity measures. Therefore, it could be also named as similarity-based pattern classification. The authors argue that it becomes especially useful when the original data is described by many features or when experts cannot formulate the attributes explicitly, but they are able to provide a dissimilarity measure, instead. Dissimilarity values express a magnitude of difference between two objects and become zero only when the objects are identical. They further argue: Given such a description one does not deal with overlapping classes, provided that distances are truthful representations of the objects. However, exactly the last statement is a crucial point in similarity-based approaches.

DSC works as following: The distance measures between all cases *x* are calculated. Likewise in hierarchical clustering, the final representation is an *n x n* distance matrix. In the learning process, the decision rules are constructed on the complete *n x n* pairwise distance matrix, see Figure 4.

A new case is then classified by using their distances to the *n* training cases, see Figure 5. That means a new sample must be compared to all training samples and the dissimilarity measures must be calculated before they are passed to the classifier.

The classifier can be any of the known classification algorithms such as for example a Support-Vector classifier, decision trees, a linear /quadratic classifier, nearest neighbor or Fishers linear discriminant. It has been studied how each classifier performs when the dissimilarity between the objects is calculated based on different similarity measures such as Euclidean distance, Hamming distance, Max-Norm, Box-Cox Transformation, and City Block [9].

Besides the complete *n x n* distance matrices, also their *n x m (m<n)* reduced versions are studied, which are sets of dissimilarities computed between *n* training samples and *m* prototypes chosen from their collection.

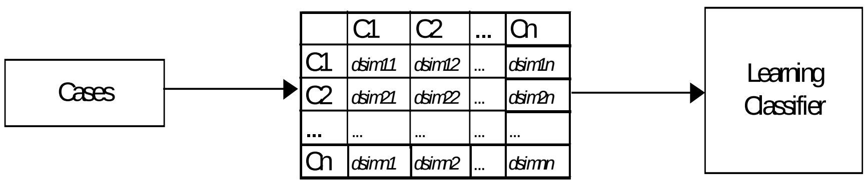

Fig. 4. Learning Dissimilarity-Based Classifier

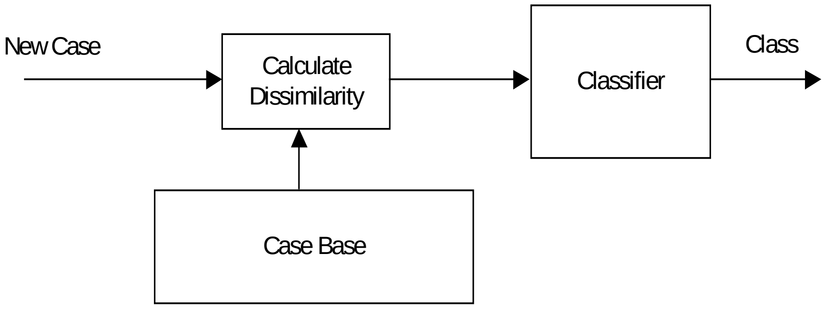

Fig. 5. Dissimilarity-based Classification

The main problems concerned with the development of dissimilarity-based classification are:

- How to access the dissimilarity between the objects?
- What is a proper dissimilarity measure for the problem?
- What is the best type of classifier for the dissimilarity based representation of the objects?
- How to select prototypes?
- What is a representative number of design samples?
- How to organize the system for fast computation?

4. Comparison between CBR and DSC

We have reviewed Case-based Reasoning and Dissimilarity-based Classification. While CBR has been around for more than 10 years, DSC was introduced some years ago. The main focus of the work in DSC is to show that it is possible to built classifiers based on (dis)similarity measures. The study shows that these classifiers do not necessarily work better in terms of accuracy than feature-based classifier [8]. The intention of this work is to overcome the problem of specifying the right image features for classification. Likewise as CBR, DSC relies on the properly chosen similarity measure. The problems with determining similarity have been neglected in the DSC work.

It is argued by Duin et. al [42] that experts are rather able to rank objects based on their dissimilarity instead of describing them by features. However, similarity can have different perspectives as we have shown in Section 2.5. There is no unique way to assess similarity. One person finds two images similar because of the geometric relation between objects in these two images. Another person finds the same images dissimilar since this person does not judge similarity based on the geometric relations between the objects but this person uses the color of the objects in the image to judge similarity. Knowledge engineering experiments for knowledge based image interpretation systems and experiments with repertory grids for determining defect classification knowledge [45] have shown that experts can not easily judge which objects are similar and to what degree they are similar. Also different experts in the field, who are trained to read for example medical images or images showing manufacturing defects, judge similarity of images differently. A consensus of opinion can only be achieved by trying to make explicit the image features and the strategy used by the experts to determine similarity. Therefore, DSC approach does not avoid the knowledge engineering problem; it puts it only in another direction. The assessment of similarity is not a well-understood concept yet. CBR tries to make a step into this direction.

CBR tries to avoid calculation of similarity between all cases and the recent case in order to reduce the computational burden. Therefore, the organization of the case base plays an important role in CBR. The case base should be organized in such a way that similar cases are grouped together and dissimilar cases are separated from them. This should ensure during retrieval of similar cases that such groups of cases that are dissimilar to the recent case are sorted out at an early stage of the retrieval process. This organization is based on the similarity relation between the cases in the case base. The recent case is classified through the organization structure based on its similarity to the cases in the case base. The organization of the case base is related to the classification in DSC. The classifiers in DSC also try to find the boundaries between the subspace of similar cases. While the calculation of similarity between the recent case and the cases in the case base stays explicit during the classification in CBR, in DSC this calculation must be carried out before the recent case is given to the classifier. The computational burden in DSC is enormous even for small case bases.

CBR has been introduced by the artificial intelligence community. Naturally, this community focuses on methods which make knowledge explicit. The assessment of similarity should stay explicit to the user in order to understand the concept of similarity better. Under this requirement, classifiers such as support vector machines, linear discriminate analysis are not sufficient. Following the trend in pattern recognition which relies on numbers instead of on symbolic knowledge, the classifiers are different in DSC from those in CBR.

DSC has similarities to hierarchical clustering [44]. In hierarchical clustering the $n \times n$ similarity matrix is also used and based on this similarity matrix hierarchical groups of similar cases are calculated. While in clustering the classification rules is not made explicit, in DSC the rules are learnt by the used classifier. Conceptual clustering [43] are methods which make the classification rules explicit. To this respect DSC is similar to conceptual clustering. However, conceptual clustering explains the way similarity has been accessed and does not require the calculation of similarity beforehand. In DSC the similarity of the actual object to all cases in the case base must always be calculated before the classification process.

Conceptual clustering methods are used to built index trees for CBR systems [34][35]. They are always used in an incremental fashion in order to update them according to new acquired cases. DSC does not consider the aspect of incremental learning. Learning is only understood as learning of classifier from the initial similarity matrix. DSC does not consider the different types of learning such as learning of new cases; prototype learning and learning of similarity which are necessary to ensure that the system will improve their performance. It is assumed that such kind of classifiers can be built on sets with small sample size [9]. This might be true if the sample set is a good representative of the domain. However, it has been shown in CBR that maintenance of the case base is an important issue.

CBR community has focussed on all aspects of CBR from basic principles to software engineering aspects and developed a lot of good ideas that have been shown excellent performance in practice. The work on DSC is preliminary and does not consider the engineering aspect. Many topics that have been worked out in CBR are relevant for DSC such as how to define similarity, incremental learning, prototype selection, software engineering aspects and so on.

Finally, we think that DSC is only a variant of CBR and that DSC can benefit from the concepts developed in CBR.

5. Conclusion

We have compared Case-based Reasoning and Dissimilarity-based Classification. Both approaches use the (dis)similarity measure between the new case and cases in the cases base to classify the new case. The difference between CBR and DSC is that in DSC the (dis)similarity measure between the new case and all cases in the case base must be calculated before the classification. It is clear that such an approach is computationally expensive. The classification algorithms used in DSC are traditional pattern recognition algorithms such as support vector machines, linear discriminat function and decision trees. The assessment of similarity stays always explicit during the reasoning process in CBR. Traditionally this community tries to develop methods that have explanation capability.

While CBR considers all aspects of the similarity based reasoning the work on DSC does not. Finally, we think that DSC can learn a lot from CBR.

References

1. Jarmulak, Case-Based Classification of Ultrasonic B-Scans: Case-Base Organisation and Case Retrieval, In: B. Smyth and P. Cunningham, Advances in Case-Based Reasoning, lnai 1488, Springer Verlag 1998, p. 100-111.
2. M. Grimnes and A. Aamodt, A Two Layer Case-Based Reasoning Architecture for Medical Image Understanding, In: I. Smith and B. Faltings (Eds.), Advances in Case-Based Reasoning, lnai 1168, Springer Verlag 1996, pp 164-178.
3. P. Perner, An Architeture for a CBR Image Segmentation System, Journal on Engineering Application in Artificial Intelligence, Engineering Applications of Artificial Intelligence, vol. 12 (6), 1999, p. 749-759.
4. A. Micarelli, A. Neri, and G. Sansonetti, A Case-Based Approach to Image Recognition, In: E. Blanzieri and L. Portinale (Eds.), Advances in Case-Based Reasoning, lnai 1898, Springer Verlag 2000, p. 443-454.
5. W. Cheetham and J. Graf, Case-Based Reasoning in Color Matching, In: Leake, D.B. and Plaza, E. (Eds.) Case-Based Reasoning Research and Development, Springer Verlag 1997, 1-12.
6. V. Ficet-Cauchard, C. Porquet, and M. Revenu, CBR for the Reuse of Image Processing Knowledge: A Recursive Retrieval/Adaption Strategy, In: K.-D. Althoff, R. Bergmann, and L. Karl Branting (Eds.), Case-Based Reasoning Research and Development, 1999, lnai 1650, p. 438-453.

7. P. Perner, Using CBR Learning for the Low-Level and High-Level Unit of a Image Interpretation System, ICAPR´98, Plymouth, peer reviewed conference, In: Sameer Singh (Eds.), Advances in Pattern Recognition, Springer Verlag 1998, p. 45-54.
8. E. Pekalska and R.P.W. Duin, Classifier for dissimilarity-based pattern recognition, In Proc.: A. Sanfeliu et. al (Eds.), 15th Intern. Conference on Pattern Recognition, Barcelona 2000, IEEE Computer Society PR 00750, p. 12-16.
9. R. Duin, Classifiers in Almost Empty Spaces, In Proc.: A. Sanfeliu et. al (Eds.), 15th Intern. Conference on Pattern Recognition, Barcelona 2000, IEEE Computer Society PR 00750, p. 1-7.
10. E. Blanzieri and L. Portinale (Eds.), Advance in Case-Based Reasoning, Springer Verlag, lnai 1898, 2000.
11. K.-D. Althoff, R. Bergmann, and L.K. Branting (Eds.), Case Based Reasoning Research and Development, Springer Verlag, lnai 1650, 1999.
12. K.-D. Althoff, Case-Based Reasoning, In: S.K. Chang (ed.) Handbook of Software Engineering and Knowledge Engineering, vol. I, World Scientific (to appear).
13. Wess St., Globig Chr. Case-Based and Symbolic Classification. In: Wess St., Althoff K.-D., Richter M.M. (eds.). Topics in Case-Based Reasoning. Springer Verlag 1994, pp 77-91.
14. M.M. Richter, Introduction (to Case-Based Reasoning), In: M. Lenz et. al (Eds.) Case-Based Reasoning Technology: From Foundations to Applications, Springer Verlag 1998, lnai 1400.
15. F. Heister and W. Wilke, An Architecture for Maintaining Case-Based Reasoning Systems, In: B. Smyth and P. Cunningham (Eds.), Advances in Case-Based Reasoning, lnai 1488, Springer Verlag 1998, p. 221-232.
16. J. Lluis Arcos and E. Plaza, A reflective Architecture for Integrated Memory-Based Learning and Reasoning, In: St. Wess, K.-D. Althoff, and M.M. Richter (Eds.) Topics in Case-based Reasoning, Springer Verlag 1993, p. 289-300.
17. L.B. Smith, From global similarities to kinds of similarities: the construction of dimensions in development. In: St. Vosniadou and A. Ortony (Eds.), Similarity and Analogical Reasoning, Cambridge University Press, 1989
18. M. Bayer, B. Herbig, and St. Wess, Similarity and Similarity Measures, In: S. Wess, K.D. Althoff, F. Maurer, J. Paulokat, R. Praeger, and O. Wendel (Eds.), Case-Based Reasoning Bd. I, SEKI WORKING PAPER SWP-92-08 (SFB)
19. P. Zamperoni and V. Starovoitov, „How dissimilar are two gray-scale images“, In Proc. of 17. DAGM Symposium 1995,Springer Verlag, pp.448-455
20. S. Santini and R. Jain, Similarity Measures, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 21, No. 9,1999, pp. 871-883
21. Y. Horikowa, Pattern Recognition with Invariance to Similarity Transformations Based on Third-Order Correlations, In Proceedings of IAPR´96, Volume II, Track B, pp 200-204
22. F. Leitao, A Study of String Dissimilarity Measures in Structural Clustering, In: S. Singh (Eds.), Advances in Pattern Recognition, Springer Verlag 1999, pp 385-394.
23. G. Mehrotra, Similar Shape Retrieval Using a Structural Feature Index, Information Systems, vol. 18 (5), 1993, pp. 525-537.
24. C. Cortelazzo, G. Deretta, G.A., Mian, P. Zamperoni, Normalized weighted Levensthein distance and triangle inequality in the context of similarity discrimination of bilevel images, Pattern Recognition Letters, vol. 17, no. 5, 1996, pp. 431-437
25. A. Crouzil, L. Massipo-Pail, S. Castan, A New Correlation Criterion Based on Gradient Fields Similarity, In Proceedings of IAPR´96, vol. I, Track A, p. 632-636
26. Moghadda, Nastar, Pentland, A Bayesian Similarity Measure for Direct Image Matching, In Proc. of ICPR´96, vol. II, Track B, pp. 350-358.
27. Moghadda, Jebra, Pentland, Efficient MAP/ML Similarity Matching for Visual Recognition, In Proc. of ICPR´98, vol. I, pp. 876-881

28. Wilson, Baddely, Owens, A new metric for gray-scale image comparison, Intern. Journal of Computer Vision, vol. 24, no. 1, pp. 5-19
29. B. Messmer and H. Bunke, Efficient subgraph isomorphism detection: a decomposition approach, IEEE Trans. on Knowledge and Data Engineering, vol 12, No. 2, 2000, pp. 307-323
30. A. van der Heiden and A. Vossepoel A Landmark-Based Approach of Shape Dissimilarity, In Proc. of ICPR 1999, vol. I, Track A, pp. 120-124
31. P. Perner, Content-Based Image Indexing and Retrieval in a Image Database for Technical Domains, In: Multimedia Information Analysis and Retrieval, Horace H.S. Ip and A. Smuelder (Eds.), LNCS 1464, Springer Verlag 1998, p. 207-224
32. A. Voß (Eds.), Similarity Concepts and Retrieval Methods, Fabel Report No. 13, 1993, ISSN 0942-413X
33. J. Surma and J. Tyburcy, A Study on Competence-Preserving Case Replacing Strategies in Case-Based Reasoning, In: B. Smyth and P. Cunningham (Eds.), Advances in Case-Based Reasoning, lnai 1488, Springer Verlag 1998, p. 233-238.
34. P. Perner, Different Learning Strategies in a Case-Based Reasoning System for Image Interpretation, Advances in Case-Based Reasoning, B. Smith and P. Cunningham (Eds.), LNAI 1488, Springer Verlag 1998, S. 251-261.
35. St. Wess, K.-D. Althoff, and G. Derwand, Using k-d Trees to Improve the Retrieval Step in Case-Based Reasoning, In: St. Wess, K.-D. Althoff, and M.M. Richter (Eds.) Topics in Case-based Reasoning, Springer Verlag 1993, p. 167-182.
36. R. Bergmann and A. Stahl, Similarity Measures for Object-Oriented Case Representations, In Proc.: Advances in Case-Based Reasoning, B. Smith and P. Cunningham (Eds.), LNAI 1488, Springer Verlag 1998, p. 25-36.
37. L. Portinale, P. Torasso, and P. Tavano, Speed-Up, Quality and Competence in Multi-modal Case-Based Reasoning, In: K.-D. Althoff, R. Bergmann, and L. K. Branting (Eds.) Case-Based Reasoning Research and Development, lnai 1650, Springer Verlag 1999, p. 303-317.
38. B. Smyth and E. McKenna, Modelling the Competence of Case-Bases, In: B. Smyth and P. Cunningham (Eds.), Advances in Case-Based Reasoning, lnai 1488, Springer Verlag 1998, p. 208-220.
39. Perner P., Paetzold W. An Incremental Learning System for Interpretation of Images. In: D. Dori and A. Bruckstein (eds.). Shape, Structure, and Pattern Recognition. World Scientific Publishing Co., 1995, pp 311-323.
40. D. Wettscherek, D.W. Aha and T. Mohri, A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms, in Artifical Intelligence Review (also available on the Web from http://www.aic.nrl.navy.mil/~aha.
41. A. Bonzano and P. Cunningham, Learning Feature Weights for CBR: Global versus Local
42. R.P.W. Duin, D. de Ridder, and D.M.J. Tax, Featureless Classification, Kybernetica, vol. 34, no. 4, 1998, p. 399-404.
43. G. Briscoe and T. Caelli, A Compendium of Machine Learning, Vol. 1: Symbolic Machine Learning, Ablex Publishing Corporation , Norwood, New Jersey, 1996
44. A.K. Jain and R.C. Dubes Algorithm for Clustering Data, Prentice Hall 1998
45. P. Perner, How to use Repertory Grid for Knowledge Acquisition in Image Interpretation. HTWK Report 2, 1994.