Acquisition of Concept Descriptions by Conceptual Clustering

Silke Jänichen and Petra Perner

Institute of Computer Vision and applied Computer Sciences, IBaI, Körnerstr. 10, 04107 Leipzig
ibaiperner@aol.com, www.ibai-institut.de

Abstract. Case-based object recognition requires a general case of the object that should be detected. Real world applications such as the recognition of biological objects in images cannot be solved by one general case. A case-base is necessary to handle the great natural variations in the appearance of these objects. In this paper we will present how to learn a hierarchical case base of general cases. We present our conceptual clustering algorithm to learn groups of similar cases from a set of acquired structural cases. Due to its concept description it explicitly supplies for each cluster a generalized case and a measure for the degree of its generalization. The resulting hierarchical case base is used for applications in the field of case-based object recognition.

Keywords: Case Mining, Case-Based Object Recognition, Cluster Analysis

1 Introduction

In case-based object recognition a group of similar objects is represented by a generalized case for the purpose of efficient matching. If this representative case is not known *a-priori* it must be learnt from real examples. There arise special problems if the objects of interest have a great variation so they can not be generalized by one single case. A case base is necessary which describes the different appearances of the objects. But then it is also not known in advance how many cases are necessary to detect all objects with a sufficiently high accuracy.

Clustering techniques can be used to mine for groups of similar cases in a set of acquired cases. For each group it is possible to determine a generalized case to represent this group. Because we do not know the number of cases in advance we will use hierarchical cluster analysis method to learn a hierarchy of increasing generalized cases. Applying a hierarchical instead of a flat case-base for case-based object recognition might speed up the recognition process especially in CBR applications with very large case bases.

When learning a representative case of a cluster this case should average over all cases in this cluster by generalizing common properties of the instances. We offer two different approaches to calculate such a representative. While the first one is to learn an artificial case that is positioned in the centroid, the second one selects that case out of a cluster which has the minimum distance to all other cases in this cluster.

It is also important to know the permissible dissimilarity from this generalized case. The degree of generalization of the cases decreases from the top to the bottom of the hierarchal case base and has to be taken into account in the matching process. The more groups are established in a hierarchy level the less generalized these representatives will be. When matching those cases for object recognition the similarity measure has to be set according to the degree of its abstraction. The less generalized the cases are, the higher the required similarity for the matched objects.

We will present in this paper our study on learning generalized cases. First we review related work on clustering in Section 2 and describe the material used for our study in Section 3. After having reviewed some agglomerative clustering methods in Section 4 we describe our novel algorithm for learning general cases in Section 5. The description of the calculation of cluster representatives is given in Section 6. We discuss experimental results in Section 7 and, finally, give conclusions in Section 8.

2 Related Work

Cluster analysis [1], [2] is used to mine for groups of similar observations in a set of unordered observations. In conclusion, similar cases should belong to the same cluster for strong internal compactness and dissimilar cases should belong to different clusters for a maximum of external separation.

There are plenty of different clustering algorithms [3], [4] and which one is best suited depends on the dataset and on the special properties and aims coupled with the cluster analysis. One main difference between several clustering algorithms is the resulting organization of the instances. Clustering algorithms can be distinguished into overlapping, unsharp, and disjunctive. While overlapping clustering algorithms allow that one case is located in one or more clusters, unsharp clustering algorithms assign to each case membership values related to all clusters. Disjunctive clustering algorithms are best suited for our application because every case has to be assigned to exactly one cluster.

Another main criterion concerning the choice of a clustering method is if the number of resulting groups is known. If the number of clusters is known *a-priori* partitioning clustering [5], [6] can be used, where an initial partition of the cases becomes optimized. If it is unknown or impossible to determine the number of clusters in advance it might be better to create a sequence of partitions using hierarchical clustering methods.

A hierarchical clustering method [1], [4] divides the set of all input cases into a sequence of disjunctive partitions. They can be distinguished between agglomerative and divisive methods. Initially, in the agglomerative methods each case is hosted in its separate cluster. With increasing distances the clusters become merged in cluster that are more general until finally all cases are hosted in the same cluster. The opposite is given in the divisive methods, where initially all cases are hosted in one cluster and were splitted until all cases form their own cluster. The main drawback of these algorithms is that once a cluster has been formed there is no way to redesign this cluster if necessary when other examples have been seen.

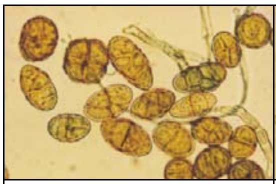
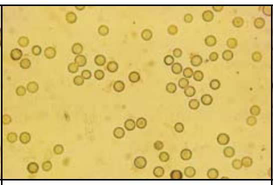
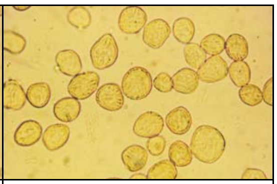
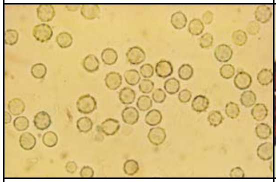
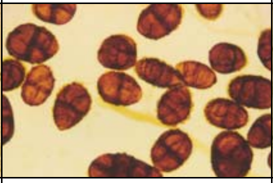
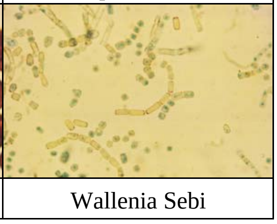
Another main problem with these conventional clustering algorithms is that it is only possible to draw conclusions about the composition of the clusters. They do not explain why a cluster was established and they supply no real indication about the quality of single partitions. To determine the partition with the optimal number of clusters different cluster validity indices [7], [8] can be used to prove the quality of single partitions. However these indices have to be calculated in an off-line phase after the clustering has been done. Besides conventionally clustering methods supply no precise description about the clusters. One has to calculate this manually for each cluster in a post-processing step which is not sufficient for our purpose.

Alternatively, different conceptual clustering algorithms [9], [10], [11] were developed. They establish clusters with a utility function, which can be built on a probabilistic concept [9], [11], or a similarity based concept [10]. On the basis of this function they explain why a set of cases confirm a cluster and automatically supply a comprehensive description of the established concepts. Their concept forming strategy is more flexible than the one of the conventional hierarchical clustering algorithms.

3 Material

In our application we are studying the shapes of six different airborne fungal spores. Table 1 shows one of the images for each analyzed fungal strain. The objects have a great variance in the appearance so that it is impossible to represent their shape by only one case. But for the purpose that these object shapes should be effectively detected in new images, it is indispensable to generalize the shapes.

Table 1. Images of six different fungal strains

		
Alternaria Alternata	Aspergillus Niger	Rhizopus Stolonifer
		
Scopulariopsis Brevicaulis	Ulocladium Botrytis	Wallenia Sebi

From the real images we acquired a set of shapes for each species. These shapes were pair-wise aligned to obtain a measure of distance between them. A detailed description of our shape acquisition and alignment process was presented in [12]. The alignment of every possible pair of shapes leads us to $N \times N$ distance measures between N acquired cases. These distances can be collected in a matrix where each row and each column corresponds to a shape instance. This square symmetric distance matrix will be used as input for the hierarchical cluster analysis.

4 Agglomerative Clustering Methods

There are plenty of different agglomerative clustering methods. Each method has its special characteristic and should be used in compliance with the aims of the application, e.g. detection of outlier. We will analyze how they can be used for our problem of learning groups of similar cases and group representatives with its concept description.

Usually in agglomerative clustering methods the resulting sequence of partitions is graphically represented by a dendrogram (see Fig. 1). The set of all input cases is shown on the left side. In the initial partition each case forms its own cluster. They become merged with increasing distances from left to right until all cases are combined in only one cluster. The distance where two clusters become one cluster for the first time depends on the particular clustering method. This distance is called cophenetic proximity measure and is drawn on the abscissa of the dendrogram. Note that this proximity measure is not equal to the pair-wise dissimilarity measure. However the aim while calculating the cophenetic proximity measure is that the real proximity relation between the objects should not be distorted.

To obtain the partition of one level in the hierarchy the dendrogram has to be cut at some distance. The cut-point drawn in Fig. 1 splits the input cases into three clusters.

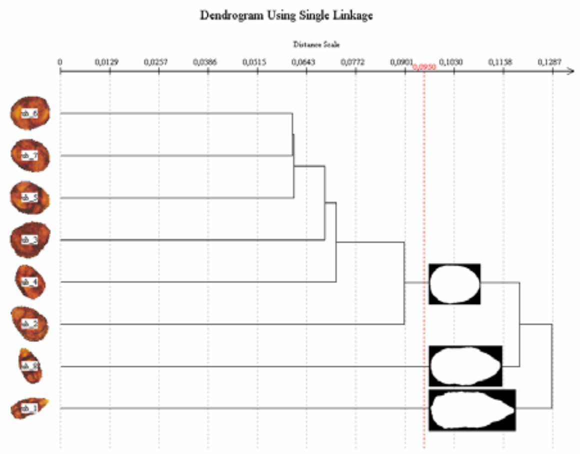


Fig. 1. Dendrogram of eight instances of strain *Ulocladium Botrytis* using single linkage with generalized cases calculated at distance of *0.0950*

Since the clusters are merged together on specific converted distances, every method has an own ultra-metric [2]. In the single linkage method two clusters become merged to a new cluster at their minimum distance, the smallest distance between a case in one cluster and its nearest neighbor in the second cluster. Graphically this can be interpreted as the shortest link between two clusters. The length of this link is the cophenetic proximity measure in the dendrogram where the two clusters become united. It usually leads to long, elongated clusters in the formation of chains which is disadvantageous in most applications. Note that only one single case out of the cluster establishes the link to the second cluster. This means the other cases have no influence on the cophenetic proximity measure. The calculated pair-wise distances between two cases within a cluster might be much greater than the cophenetic proximity measure at this hierarchy level.

By contrast the complete linkage method merges two partitions at their maximum distance, the greatest of all pair-wise distances between a case in the first and a case in the second cluster. If this method has merged the two clusters at some cophenetic proximity, it is guaranteed that every pair-wise distance between two arbitrary cases within the new cluster is smaller than or equal to this measure. Complete linkage usually establishes homogeneous, spherical clusters. Outlier cases stay unrecognized. This might be disadvantageous, because a single outlier might prevent the merging of two groups, although all other cases are very similar. But, the cophenetic proximity measures obtained with complete linkage give a first impression about the expansions of the clusters. An agglomerative clustering method which is a midway between these to extreme methods is average linkage: Two clusters are merged at the average distance of all pair-wise distances between comprised cases. In addition it is possible to weight each cluster according to the number of hosted cases. The average linkage methods establish spherical clusters while outlier cases stay unrecognized for a long time.

In the centroid method the cophenetic proximity between two clusters is the distance between their centroids. The weighting included in the centroid method emphasizes clusters which consist of many cases while small sized clusters tend to get lost [13]. Therefore, in the median method, the weights of the comprised clusters are not included when calculating the centroid of the new cluster. These methods are most suited for our purpose because we are interested in determining a representative case in the cluster centroids. But they also give no real indication about the degree of generalization of the two clusters at a hierarchy level. If necessary we have to calculate this measure in an off-line phase.

In summary it can be said that the agglomerative hierarchical clustering methods give a good impression about the organization of the underlying dataset. However, these algorithms only produce a sequence of partitions but give no further indication about why this cluster was established. Thus all other information concerning a more detailed description of a cluster, e.g. cluster mean, inner-cluster-variance, have to be calculated manually. This fact is a main drawback in all applications where the number of classes is not known in advance. The agglomerative clustering methods are simple but also rigid and inflexible. They offer merging as the only possibility to incorporate a case into a hierarchy. If a case is merged once it is impossible to separate it or to change the cluster again. If it turns out later that a classification was wrong, this is irreversible. Besides that these clustering methods can not be used for incremental learning.

5 Our Conceptual Clustering Algorithm

Conceptual clustering is a type of flexible learning the hierarchy by observations. The partitioning of the cases is controlled by a category utility function [1]. Conceptual clustering algorithms can be distinguished by the type of this utility function which can be based on a probabilistic [9], [11] or a similarity concept [10]. Our conceptual clustering algorithm presented here is based on similarities, because we do not consider logical but numerical concepts. The algorithm works directly with structural objects. In our study this is a set of acquired cases, each comprised by an ordered array of contour points. In contrast to agglomerative clustering methods where the distance matrix is used as input it is not necessary to calculate pair-wise distances in advance.

In addition to merging cases our algorithm allows incorporating new cases into existing nodes, opening new nodes, and splitting of existing nodes at every position in the hierarchy. Each new case is successively incorporated, so the algorithm dynamically fits the hierarchy to the data. The resulting sequence of partitions is represented by a directed graph (concept hierarchy) where the root node contains the complete set of input cases and each terminal node represents an individual case.

Initially the concept hierarchy only consists of an empty root node. The algorithm implements a top-down method. A new case is placed into the actual concept hierarchy level by level beginning with the root node until a terminal node is reached. In each hierarchy level one of these four different kinds of operations is performed:

- The new case is incorporated into an existing child node,
- A new empty child node is created where the new case is incorporated,
- Two existing nodes are merged to form a single node where the new case is incorporated, and
- An existing node is splitted into its child nodes.

The new case is tentatively placed into the next hierarchy level by applying all of these operations. Finally that operation is performed which gives the best score of the partition according to the evaluation criteria. A proper evaluation function prefers compact and well separated clusters. These are clusters with small inner-cluster variances and high inter-class variances. Thus we calculate the score of a partition by

$$SCORE = \frac{1}{m} \sum_{i=1}^{m} p_i \left(SB_i - SW_i \right) , \tag{1}$$

where m is the number of clusters in this partition, p_i is the relative frequency of the i-th cluster, SB_i is the inter-cluster variance and SW_i is the inner-cluster variance of the i-th cluster. The normalization according to m is necessary to compare partitions of different size. The relative frequency p_i of the i-th cluster is

$$p_i = \frac{n_i}{n} , \tag{2}$$

where n_i is the number of cases in the i-th cluster and n is the number of cases in the parent node. The output of our algorithm for applying the eight exemplary shape cases of strain *Ulocladium Botrytis* is shown in Fig. 3. On top level the root node is shown which comprises the set of all input cases. Successively the tree is partitioned into nodes until each input case forms its one cluster.

We introduced a pruning criterion into the algorithm which can be used optionally. It says that the clusters in one partition are removed if the sum of their inner-cluster-variances is *zero*. Fig. 2 shows the complete, un-pruned concept hierarchy, where a new case was incorporated supplementary. The darker nodes were those clusters which had to be updated because the new case was incorporated into them. The white nodes in the hierarchy are clusters which were not attached.

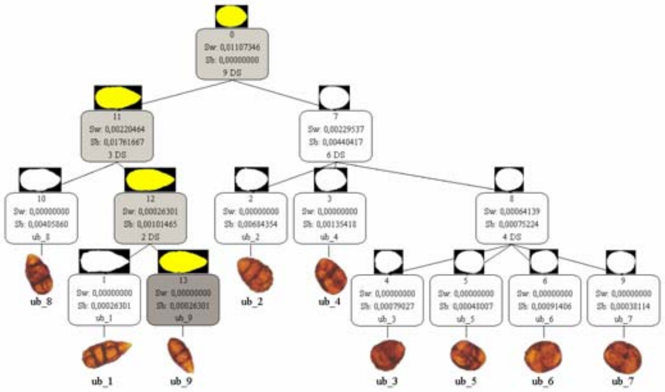


Fig. 2. Complete, not-pruned concept hierarchy after incrementally incorporating a new case. The *darker* nodes are those clusters which are modified because the new case was inserted. Their new representative cases are depicted as *yellow* shapes. The *white* nodes were not attached while the hierarchy was modified to fit the new data

The main advantage of our conceptual clustering algorithm is that it brings along a concept description. Thus, in comparison to agglomerative clustering methods it is easy to understand why a set of cases forms a cluster. The algorithm calculates the inner-cluster-variances direct on the cases within this cluster or rather on their contour points instead of using a given distance matrix. During the clustering process the representative case, and also the variances and maximum distances in relation to this representative case are calculated since they are part of the concept description. The algorithm is of incrementally fashion because it is possible to incorporate new cases into the existing learnt hierarchy.

6 Calculation of General Cases

The representative case of a cluster is a more general representation of all cases hosted in this cluster. Since this case should average over all cases in that cluster, a good case might be positioned in the centroid of the cluster. In our conceptual clustering algorithm the concept description is based on the inner-cluster-variance.

The inner-cluster-variance of a cluster X is calculated by

$$SW_X = \frac{1}{n_x} \sum_{i=1}^{n_x} d(x_i, \bar{\mu}_X)^2 \ , \tag{3}$$

where $\bar{\mu}_X$ is the centroid and n_x is the number of cases in the cluster X. Thus, a direct output of the clustering process is the calculation of the cluster centroids.

In our application the cluster centroid is an artificial mean shape defined by an ordered set of points. To calculate this shape it is necessary to determine corresponding points [12] between the shapes in the cluster. For each set of corresponding points between all shapes in one cluster we calculate its centroid. The centroid of a set of n_S corresponding 2D-points $s_i(x,y)$, $i = 1,2,\dots,n_s$ is given by

$$\bar{\mu}_S(x,y) = \frac{1}{n_S} \sum_{i=1}^{n_S} s_i(x,y) \ . \tag{4}$$

All calculated mean points are set as points on the contour of the representative shape of this cluster. This results in an artificial mean shape case positioned in the centroid of the cluster.

A second approach is to select the medoid as a natural representative case for a cluster. The medoid x_{medoid} of a cluster X is the shape case which is positioned closest to the cluster centroid. It is the case which has the minimum distance to all other cases in the cluster

$$\bar{\mu}_X = x_{medoid} = \min_{x \in X} \sum_{i=1}^{n_x} d(x_i, x) \ . \tag{5}$$

In addition to the representative of a cluster we are interested in leaning the maximum permissible distance from this generalized case. The maximum permissible distance D_X to the representative case is

$$D_X = \max_{x \in X} \ d(x, \bar{\mu}_X) \ . \tag{6}$$

When matching objects with a hierarchical case-base of increasing specialized cases it is important to know the degree of generalization for each case. This measure will be used as threshold for the similarity score while matching.

7 Experimental Results

Our conceptual clustering algorithm was directly applied to the set of shape cases instead of the matrix of pair-wise distances between those cases. The pruned version of the resulting hierarchy for eight exemplary cases is shown in Fig. 3. The established groups appear useful and logical. If we compare this hierarchy to the outputs of the agglomerative clustering algorithms it is very similar to the median method, which is based on the distances between un-weighted cluster centroids. The outputs are similar but the main difference is how these results were obtained.

In comparison to the agglomerative methods our conceptual clustering algorithm is incremental and more flexible. If during the process it turns out that a classification was wrong, it is still possible to split or merge a formed cluster afterwards. If a new case is incorporated into the concept hierarchy, the algorithm dynamically fit the hierarchy to the new data. It has linear time complexity $O(N)$. By contrast the agglomerative clustering methods have to calculate the complete hierarchy again if a new case should be incorporated supplementary. Thus, conceptual clustering is better suited for huge databases and all applications where it is necessary to adapt the hierarchy by learning new cases over time.

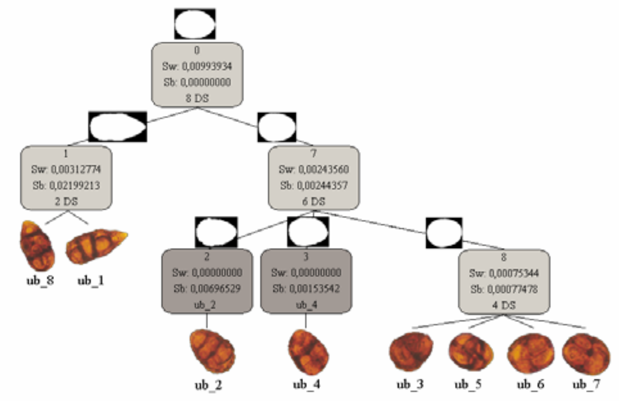


Fig. 3. The pruned version of the concept hierarchy resulting from the eight instances of strain *Ulocladium Botrytis* is shown. On the top of each node the generalized representative of this cluster is shown

Our algorithm brings the right concept description for our purpose of learning case groups and generalized cases. The calculated general cases represent the clusters and are stored into the case base. The measures inner-cluster variance, inter-cluster variance, and maximum permissible distance to the cluster centroid help us to understand on what hierarchy level we should stop to generalize the cases so that we can achieve good results during the matching process.

8 Conclusion

We have described how to learn a hierarchical case base of general cases from a set of acquired cases. It has shown that classical hierarchical clustering methods give a good impression about the organization of the cases but fail if further information is necessary. Our presented conceptual clustering algorithm is directly working on the set of structural cases, while the resulting hierarchy is similar to those of classical hierarchical clustering methods.

We have also shown that our algorithm is more flexible since the establishing of the hierarchy is not only based on merging, but it is also possible to split, incorporate, and create cluster. In addition to that it allows incremental incorporation of new cases while the hierarchy is only adapted to fit the new data. Due to its concept description our conceptual clustering algorithm supplies for each cluster a generalized case and a measure for the degree of its generalization. This output in form of a hierarchical case base with decreasingly generalized cases is the basis for efficient application in case-based object recognition.

Acknowledgement

This project is sponsored by the European Comission within the project “Multimedia Understanding through Semantics, Computation, and Learning” No. 507752.

References

1. P. Perner, Data Mining on Multimedia Data, Springer Verlag Berlin, 1998.
2. H.J. Mucha, Clusteranalyse mit Mikrocomputern, Akademie Verlag, Berlin, 1992.
3. A.K. Jain and R.C. Dubes, Algorithms for Clustering Data, Prentice Hall, 1988.
4. E. Rasmussen, Clustering Algorithms, In W.B. Frakes and R. Baeza-Yates (Eds), Information Retrieval, pp. 419-442, Prentice Hall, 1992.
5. J.B. MacQueen, Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281-297, Berkeley, University of California Press, 1967.
6. S.K. Gupta, K.S. Rao, and V. Bhatnagar, K-means Clustering Algorithm for Categorical Attributes. In M.K. Mohania and A. Min Toja (Eds.) Proceedings of the First International Conference on Data Warehousing and Knowledge Discovery, pp. 203-208, Springer Verlag, lncs 1676, 1999.
7. J.C. Dunn, Well separated clusters and optimal fuzzy partitions, J. Cybern. Vol. 4, pp. 95-104, 1974.
8. D.L. Davies and D.W. Bouldin, A cluster separation measure, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 1, No. 2, pp. 224-227, 1979.
9. D. Fisher and P. Langley, Approaches to conceptual clustering, Proceedings of the Ninth International Joint Conference on Artificial Intelligence, pp. 691-697, Los Angeles, 1985.
10. P. Perner, Different Learning Strategies in a Case-Based Reasoning System for Image Interpretation, In B. Smith and P. Cunningham (Eds.), Advances in Case-Based Reasoning, pp. 251-261, Springer Verlag, lnai 1488, 1998.
11. W. Iba and P. Langley, Unsupervised Learning of Probabilistic Concept Hierarchies, In G. Paliouras, V. Karkaletsis, & C. D. Spyropoulos (Eds)., Machine learning and its applications. Springer Verlag, 2001.

12 P. Perner and S. Jänichen, Case Acquisition and Case Mining for Case-Based Object Recognition, In: Peter Funk and Pedro A. Gonzalez Calero (Eds.), Advances in Case-Based Reasoning, Proceedings of the ECCBR 2004, pp. 616-629, Springer Verlag, 2004.

13. G.N. Lance and W.T. Williams, A General Theory of Classification Sorting Strategies, 1. Hierarchical Systems, pp. 373-380, Comp. J. 9, 1966.