Learning of General Cases

Silke Jänichen and Petra Perner

Institute of Computer Vision and applied Computer Sciences, IBaI,
Koernerstr. 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 variation in appearance of these objects. We present our conceptual clustering algorithm to learn a hierarchy of decreasingly generalized 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 efficient matching. If this representative case is not known *a-priori* it must be learnt from real examples. Special problems arise if the objects of interest have a great variation, so they cannot be generalized by one single case. A case base is necessary which describes the different appearances of the objects. But then it is 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 the hierarchical cluster-analysis method to learn a hierarchy of increasingly 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 because it 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.

S.K. Pal et al. (Eds.): PReMI 2005, LNCS 3776, pp. 774 – 779, 2005.

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. 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 is the resulting organization of the instances. In our application, we prefer disjunctive clustering algorithms, because every case has to be assigned to exactly one cluster. If the number of clusters is known *a-priori*, partitioning clustering [5] can be used. 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.

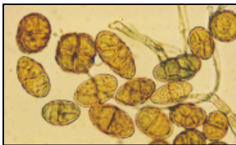
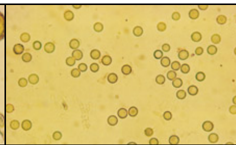
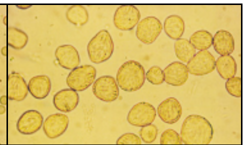
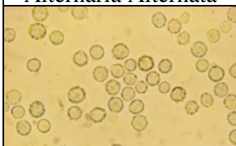
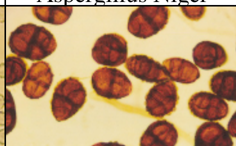
A hierarchical clustering method [1], [4] divides the set of all input cases into a sequence of disjunctive partitions. They can be distinguished between agglomerative and divisive methods. The main drawback of conventional hierarchical clustering algorithms is that once a cluster has been formed, there is no way to redesign this cluster if necessary after other examples have been seen. Another main problem 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 optimal number of clusters, different cluster validity indices [6] can be used. However, these indices have to be calculated in an off-line phase after the clustering has been done. Besides, conventional clustering methods supply no precise description about the clusters. One has to calculate this manually for each cluster in a post-processing step.

Alternatively, different conceptual clustering algorithms [7], [8] were developed. They establish clusters with a utility function, which can be built on a probabilistic concept [8], or a similarity-based concept [8]. 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 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 appearance, so that it is impossible to represent their shape by only one case. 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

Table 1. Images of six different fungal strains

		
Alternaria Alternata	Aspergillus Niger	Rhizopus Stolonifer
		
Scopul. Brevicaulis	Ulocladium Botrytis	Wallenia Sebi

in [9]. The alignment of every possible pair of shapes leads us to $N \times N$ distances between N acquired cases. These distances can be collected in a square symmetric distance matrix, which will be used as input for the hierarchical cluster analysis.

4 Agglomerative Clustering Methods

There are plenty of different agglomerative clustering methods. We have analyzed how they can be used for our problem of learning groups of similar cases and group representatives with its concept description. In agglomerative clustering methods, initially each case forms its own cluster. They become merged with increasing distances until all cases are combined in only one cluster. The distance where two clusters become one for the first time is called cophenetic proximity measure. Note that this proximity measure is not equal to the pair-wise dissimilarity measure. The clusters are merged together on specific converted distances, so every method establishes an own ultra-metric [2].

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 has to be calculated manually. 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. Once a case is merged, 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. Beside that, these clustering methods cannot be used for incremental learning.

5 Our Conceptual Clustering Algorithm

Conceptual clustering is a type of flexible learning of the hierarchy by observations. The partitioning of the cases is controlled by a category utility function [1], which can be based on a probabilistic [7], or a similarity concept [8]. Our conceptual clustering algorithm presented here is based on similarities, because we do not consider logical but numerical concepts.

The algorithm results in a sequence of partitions (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. Each new case is successively incorporated, so the algorithm dynamically fits the hierarchy to the data. A new case is tentatively 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 a node where the new case is incorporated, and
- An existing node is split into its child nodes.

Finally that operation is performed which gives the best score to 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 algorithm automatically determines how many clusters are necessary to obtain the best score.

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-

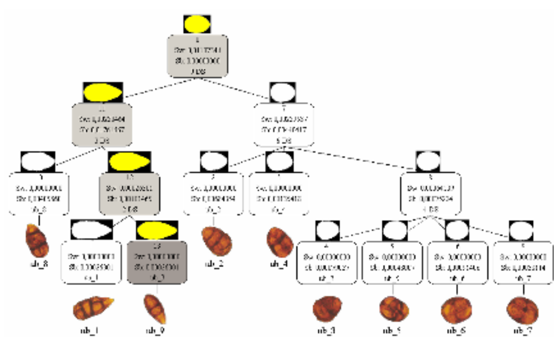


Fig. 1. Concept hierarchy after incrementally incorporating a new case. The *darker* nodes are those clusters which are modified because the new case was inserted. For these nodes the concept description and the representative cases had to be updated.

variances is *zero*. Fig. 1 shows the complete, not pruned concept hierarchy, where a new case was incorporated supplementarily. The darker nodes were those clusters, which had to be updated because the new case was incorporated into them.

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

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

where $\overline{\mu}_X$ is the centroid and n_x is the number of cases in the cluster X. Since the cluster centroid is represented by an artificial case, 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 that has the minimum distance to all other cases in the cluster

$$\overline{\mu}_X = x_{medoid} = arg \min_{x \in X} \sum_{i=1}^{n_x} d(x_i, x) . \tag{3}$$

When matching objects with a hierarchical case-base of increasingly specialized cases, it is important to know the degree of generalization for each case. This measure will be used as a threshold for the similarity score while matching. Therefore, we calculate the maximum permissible distance D_X from this generalized case

$$D_X = \max_{x \in X} d(x, \overline{\mu}_X) . \tag{4}$$

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 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 fits 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. 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. 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 clusters. 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

The project is sponsored by the European Commission 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, Proc. of 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281-297, Berkeley, University of California Press, 1967
6. 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
7. D. Fisher and P. Langley, Approaches to conceptual clustering, Proc. of the 9th International Joint Conference on Artificial Intelligence, pp. 691-697, Los Angeles, 1985
8. 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
9. 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, Proc. of the ECCBR 2004, pp. 616-629, Springer Verlag, 2004