Y. Ioannidis et al. (eds.): EDBT 2006, LNCS 3896, pp. 1151–1154, 2006.

VICO: Visualizing Connected Object Orderings

Stefan Brecheisen, Hans-Peter Kriegel, Matthias Schubert, and Michael Gruber

Institute for Informatics, University of Munich
{brecheis,kriegel,schubert,gruber}@dbs.ifi.lmu.de

Abstract. In modern databases, complex objects like multimedia data, proteins or text objects can be modeled in a variety of representations and can be decomposed into multiple instances of simpler sub-objects. The similarity of such complex objects can be measured by a variety of distance functions. Thus, it quite often occurs that we have multiple views on the same set of data objects and do not have any intuition about how the different views agree or disagree about the similarity of objects. VICO is a tool that allows a user to interactively compare these different views on the same set of data objects. Our system is based on OPTICS, a density-based hierarchical clustering algorithm which is quite insensitive to the choice of parameters. OPTICS describes a clustering as a so-called cluster order on a data set which can be considered as an image of the data distribution. The idea of VICO is to compare the position of data objects or even complete clusters in a set of data spaces by highlighting them in various OPTICS plots. Therefore, VICO allows even non-expert users to increase the intuitive understanding of feature spaces, distance functions and object decompositions.

1 Introduction

In modern databases, complex objects like multimedia data, proteins or text objects can be modeled in a variety of representations and can be compared by a variety of distance or similarity functions. Thus, it quite often occurs that we have multiple views on the same set of data objects and do not have any intuition about how the different views on data objects agree or disagree about the similarity of objects. VICO is a tool for comparing these different views on the same set of data objects. Our system is heavily based on OPTICS, a density-based hierarchical clustering algorithm, which is quite insensitive to its parametrizations. OPTICS describes a clustering as a so-called cluster order on a data set. A cluster order can be considered as an image of the data distribution in one representation. The idea of VICO is to select data objects or even complete clusters in one OPTICS plot and additionally highlight the same objects in all other displayed views on the data. VICO has the following three main applications: First, if more than one distance function for a given data set is available, it allows direct comparisons of the distance functions. Second, in a multi-represented setting, where multiple feature transformations for an object are available, the relationships between the given data representations can be examined by comparing the clusterings resulting w.r.t. these representations.

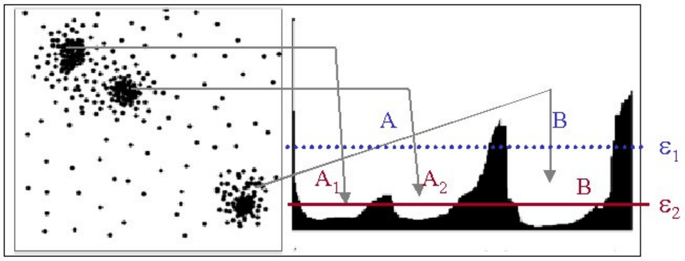


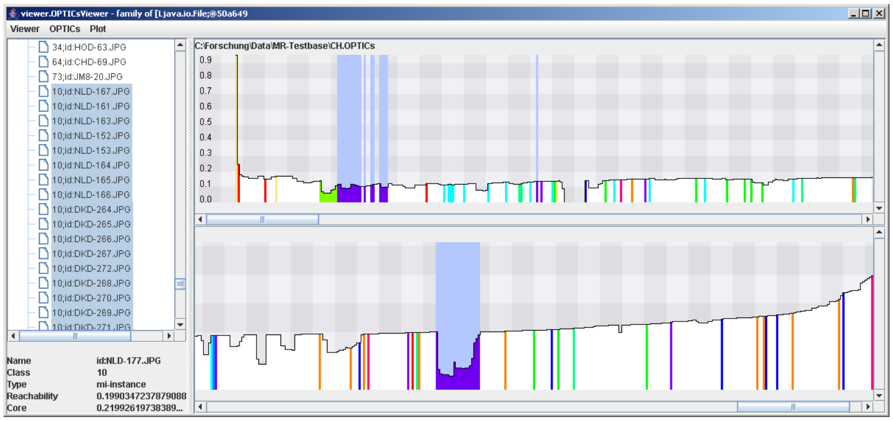
Fig. 1. Reachability plot (right) computed by OPTICS for a sample 2-D dataset (left).

Third, the connection between multi-instance objects and their single instances can be examined by comparing the clustering of multi-instance objects to the clusterings w.r.t. single instances.

2 Algorithmic Foundation

In the following, we will introduce the basic concepts behind OPTICS [1] which is the clustering algorithm VICO employs to generate the density plot of a given data representation. OPTICS is a density-based hierarchical clustering algorithm that extends DBSCAN by deriving a cluster hierarchy that is displayed within the so-called reachability plot. The central concepts of OPTICS are the core distance of an object expressing the size of the neighborhood around an object containing at least *MinPts* other objects. In other words, the core distance of object o is the smallest distance for which o would be considered a core point with respect to *MinPts*. The reachability distance of an object p from o denoted as $d_{reach}(p, o)$ is the maximum of the true distance between o and p and the core distance of o. OPTICS performs a best first run in a complete directed graph where the objects are the nodes and an edge between the objects p and o is labeled with $d_{reach}(p, o)$. After starting its traversal with an arbitrary node, OPTICS always pursues the edge first that provides the smallest reachability distance and starts with an already reached object. When traversing the data from one object to any other object the reachabilty of the correponding link is collected in the so-called reachability plot. Valleys in this plot indicate clusters: objects having a small reachability value are more similar to their predecessor objects than objects having a higher reachability value.

The reachability plot generated by OPTICS can be cut at any level ε parallel to the abscissa. It represents the density-based clusters according to the density threshold ε: A consecutive subsequence of objects having a smaller reachability value than ε belong to the same cluster. An example is presented in Fig. 1: For a cut at the level ε_1, we retrieve two clusters denoted as A and B. Compared to this clustering, a cut at level ε_2 would yield three clusters. The cluster A is split into two smaller clusters denoted by A_1 and A_2 and cluster B is decreased


viewer.OPTICsViewer - family of [Ljava.io.File;@50a649
Viewer
OPTICs
Plot
C:\Forschung\Data\MR-Testbase\CH.OPTICs
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

34;id:HOD-63.JPG
64;id:CHD-69.JPG
73;id:JM8-20.JPG
10;id:NLD-167.JPG
10;id:NLD-161.JPG
10;id:NLD-163.JPG
10;id:NLD-152.JPG
10;id:NLD-153.JPG
10;id:NLD-164.JPG
10;id:NLD-165.JPG
10;id:NLD-166.JPG
10;id:DKD-264.JPG
10;id:DKD-265.JPG
10;id:DKD-266.JPG
10;id:DKD-267.JPG
10;id:DKD-272.JPG
10;id:DKD-268.JPG
10;id:DKD-270.JPG
10;id:DKD-269.JPG
10;id:DKD-271.JPG

Name	id:NLD-177.JPG
Class	10
Type	mi-instance
Reachability	0.1990347237879088
Core	0.21992619738389...

Fig. 2. VICO displaying OPTICS plots of multi-represented data.

in size. Usually, for evaluation purposes, a good value for ε would yield as many clusters as possible.

3 Comparing Data Spaces Using VICO

The main purpose of VICO is to compare different feature spaces that describe the same set of data. For this comparison, VICO relies on the interactive visual exploration of reachability plots. Therefore, VICO displays any available view on a set of data objects as adjacent reachability plots and allows comparions between the local neighborhoods of each object. Fig. 2 displays the main window of VICO. The left side of the window contains a so-called tree control that contains a subtree for each view of the data set. In each subtree, the keys are ordered w.r.t. the cluster order of the corresponding view. The tree control allows a user to directly search for individual data objects. In addition to the object keys displayed in the tree control, VICO displays the reachability plot of each view of the data set.

Since valleys in the reachability plot represent clusters in the underlying representation, the user gets an instant impression of the richness of the cluster structure in each representation. However, to explore the relationships between the representations, we need to find out whether objects that are clustered in one representation are also similar in the other representation. To achieve this type of comparison, VICO allows the user to select any data object in any reachability plot or the tree control. By selecting a set of objects in one view, the objects are highlighted in any other view as well. For example, if the user looks at the reachability plot in one representation and selects a cluster within this plot, the corresponding object keys are highlighted in the tree control and identify the objects that are contained in the cluster. Let us note that it is possible to visualize the selected objects as well, as long as there is a viewable object representation. In addition to the information about which objects are clustered together, the set of objects is highlighted in the reachability plots of the other representations as well. Thus, we can easily decide whether the objects in one representation are placed within a cluster in another representation as well or if they are spread among different clusters or are part of the noise. If there exist contradicting reachability plots for the same set of data objects, it is interesting to know which of these representations is closer to the desired notion of similarity. Thus, VICO allows the user to label data objects w.r.t. some class value. The different class values for the objects are displayed by different colors in the reachability plot. Thus, a reachability plot of a data space that matches the user's notion of similarity should display clusters containing objects of the same color. Fig. 2 displays a comparison of two feature spaces for an image data set. Each image is labelled with w.r.t. the displayed motive.

Another feature of VICO is the ability to handle multi-instance objects. In a multi-instance representation, one data object is given by a set of separated feature objects. An example are CAD parts that can be decomposed to a set of spatial primitives, which can be represented by a single feature vector. This way, the complete CAD part is represented by a set of feature vectors, which can be compared by a variety of distance functions. To find out which instances are responsible for clusters of multi-instance objects, VICO allows us to cluster the instances without considering the multi-instance object they belong to. Comparing this instance plot to the plot derived on the complete multi-instance objects allows us to analyze which instance clusters are typical for the clusters on the complete multi-instance object. Thus, for multi-instance settings, VICO highlights all instances belonging to some selected multi-instance object.

4 Architecture and Implementation

VICO is implemented in Java 1.5 and thus, runs on any platform supporting the current version of the Java Runtime Environment. VICO includes an integrated version of OPTICS allowing the user to load and cluster data sets described in a variety of file formats like CSV and ARFF files. For this version of OPTICS there are several distance measures already implemented like the Euclidian, Manhattan or Cosine distance. Furthermore, VICO already implements various distance functions for multi-instance objects, e.g. the Hausdorff distance. The system is based on an extensible architecture, so that additional components like new distance functions can be integrated easily by implementing Java interfaces. Finally, VICO can directly load preprocessed reachability plots as well and also export reachability plots that were computed by the integrated implementation of OPTICS.

References

1. Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: "OPTICS: Ordering Points to Identify the Clustering Structure". In: Proc. ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'99), Philadelphia, PA. (1999) 49–60