CoDA: Interactive Cluster Based Concept Discovery

Stephan Günnemann Ines Färber Hardy Kremer Thomas Seidl

Data Management and Data Exploration Group
RWTH Aachen University, Germany
{guennemann, faerber, kremer, seidl}@cs.rwth-aachen.de

ABSTRACT

Large data resources are ubiquitous in science and business. For these domains, an intuitive view on the data is essential to fully exploit the hidden knowledge. Often, these data can be semantically structured by concepts. Since the determination of concepts requires a thorough analysis of the data, data mining methods have to be applied. In the field of subspace clustering, some techniques have recently shown to be effective for this task. Although these methods generate concept-based patterns, the user has to provide domain knowledge to gain reasonable concepts out of the data.

Our demonstration *CoDA* (*Concept Determination and Analysis*) is a tool that supports the user in the final step of concept definition. More concretely, the user is guided through an iterative, interactive process in which concepts are suggested, analyzed, and potentially refined. The core aspect of *CoDA* is an intuitive, concept-driven presentation of subspace clusters such that concepts can be visually captured.

1. INTRODUCTION

In today's applications such as life sciences, e-commerce and sensor networks large amounts of data have to be administrated in databases. With growing size it becomes virtually impossible to manually keep an overview over the data. One way to solve this problem is to semantically structure the database. In many applications one can observe certain structures in the data if only some characteristics of the database objects are considered. This is especially true for high-dimensional data. Fig. 1 shows an example where objects apparently group according to different attributes. These groupings for certain attribute subsets, called subspace clusters, represent a manifestation of an abstract concept. The green objects in the left plot form the group "healthy living" in the concept of "health awareness", while the red objects form the group "unhealthy living" in the same concept. The right plot in Fig. 1 depicts another concept "enthusiasm for technology" that is defined by other attributes and groupings. This example may provide insights for the process of customer segmentation in the economic field. Similar observations can be made in other scenarios as well. For example, in sensor analysis of the environment the measured events can be assigned to abstract concepts.

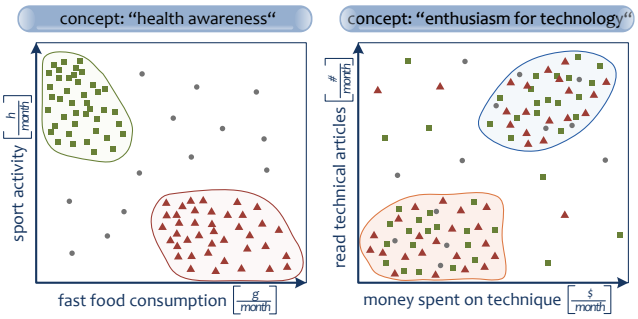


Figure 1: Different concepts in databases

Groupings of objects can be found by clustering algorithms, and since clusters are induced by concepts for which different attributes may be relevant, a consideration of attribute subsets instead of all attributes for concept discovery is needed. Subspace clustering techniques were developed for the task of finding clusters in differing subspace projections of the data [7]. Some approaches as [5, 4, 1] focus already on the specific task of grouping objects according to underlying concept structures: they find clusters in strongly differing subspace projections, providing the key for discovering the inherent concept structure. The obtained clusters can be seen as a manifestation of a concept, e.g. the clusters 'smokers', 'joggers', or 'vegans' belong to the concept 'health awareness'. Since the concepts are generative, i.e. they actually induce the clusters, they cannot be automatically concluded out of clusters. Accordingly, the mentioned subspace clustering techniques achieve concept-based aggregations of objects but are not capable of abstracting from these aggregations in the sense of named concepts.

In real-world applications, however, the interest lies in the explicit discovery and naming of the underlying concepts. This task cannot be solved automatically by unsupervised learning methods as subspace clustering but requires the domain knowledge of an expert. Our tool *CoDA* supports the user in revealing the concepts out of a given subspace clustering. It therefore provides the user with concept-oriented cluster visualization and interactive exploration to enable him to uncover the inherent concept structures. The main

Proceedings of the VLDB Endowment, Vol. 3, No. 2

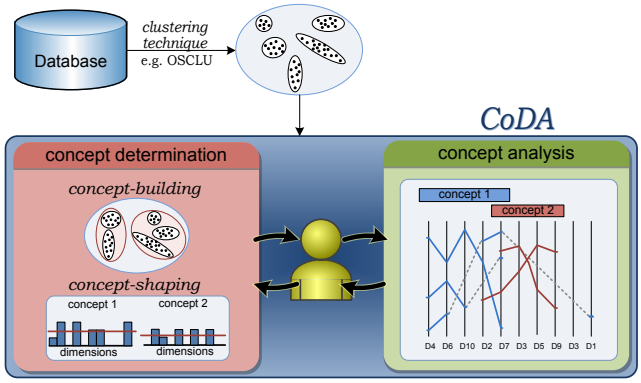


Figure 2: Workflow of *CoDA*

challenges tackled with *CoDA* are:

- Determination of concept structures
- Analysis of single concepts as well as inter-concept dependencies

Each concept can be described by its occurring clusters on the one hand and its characteristic attributes on the other hand. Since the related clusters are not known beforehand, the idea is to capture the concepts through the structure of relevant attributes of the clustering. The relevant attributes are of particular importance for a semantic labeling of clusters and concepts. The process of concept determination can be divided into two phases as depicted in Fig. 2. Given a subspace clustering of database objects, in a first determination step an interim grouping of clusters representing concepts is calculated based on their relevant subspaces. In a second determination step the user sets the significant attributes for each represented concept. In the analysis phase the user takes a closer look at the concept compositions and gives feedback to refine or to recalculate the concept structures. Thus, the whole process of concept discovery is iterative and highly dependent on user interaction.

2. CODA

In this section, we introduce our tool *CoDA* (*Concept Determination and Analysis*). *CoDA* is integrated into the OpenSubspace framework [10, 11] that adds subspace clustering functionality to the well-known WEKA Data Mining Software. In this framework, several subspace clustering algorithms are integrated; for *CoDA*, these algorithms can provide subspace clusterings to analyze them for concept structures. Figure 3 shows a screenshot of the framework with the *CoDA* integration. Subspace clusterings are created in the *SubspaceClusterer* tab, and *CoDA* is realized in the *CoDA* tab. Since *CoDA* comprises two phases, i.e. concept determination and concept analysis, these phases are realized by two distinct tabs. The final concepts are determined by cyclic usage of the two interdependent phases.

2.1 Concept determination

In the following we present how concepts are determined with *CoDA*. Remember, concepts induce clusters and not vice versa. Since in most application scenarios the inherent concepts of the data are unknown, *CoDA* determines these concepts for a given subspace clustering by integrating users and their domain knowledge into the search process. The phase of concept determination has two goals: First, to assign the given clusters to possible concepts. Second, to determine the significant dimensions of these concepts.

Clusters that share relevant dimensions are expected to describe the same concept and are therefore automatically grouped together. These groupings, however, do not consider semantic knowledge; the user has to refine them in the concept analysis phase. The assigned clusters of a possible concept can have different relevant dimensions, preventing an automatic determination of the concept's significant dimensions. It is therefore the task of the user to select these dimensions. This process is called concept shaping. The two steps cluster grouping and concept shaping are now presented in more detail.

Finding concepts based on cluster grouping. The first step aims at grouping subspace clusters such that the resulting groups possibly represent meaningful concepts. This is achieved by grouping the given subspace clusters according to their relevant dimensions and knowledge that was obtained in previous iterations of the concept analysis; the latter will be explained in more detail in Section 2.2. The clusters of one group belong very likely to the same concept and therefore represent this concept. In *CoDA*, the found concepts are displayed in the left part of the *concept determination* tab (cf. Fig. 3). The details of a concept's corresponding subspace clusters can be inspected by the user: by clicking on a cluster the cluster's objects and the relevant dimensions are shown. This is a functionality that is already implemented in the OpenSubspace framework and has shown to be very intuitive.

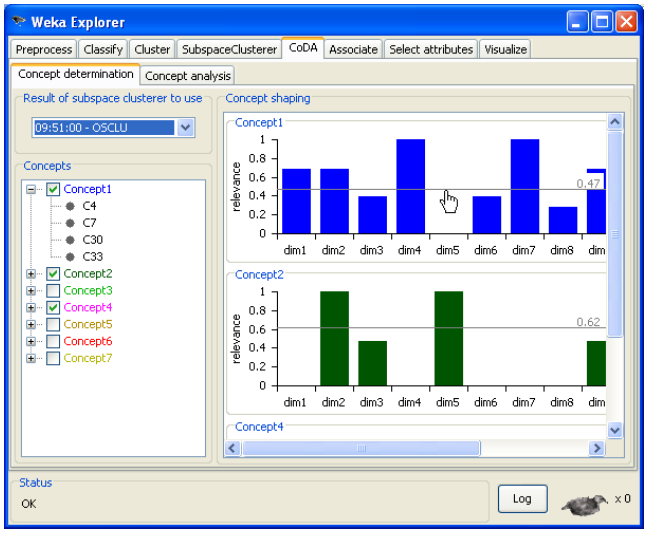
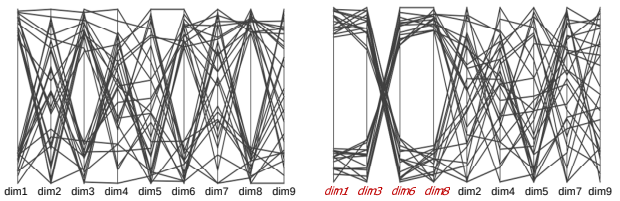


Figure 3: Concept determination tab of *CoDA*

Technically, the grouping of clusters to achieve meaningful concepts is realized by constrained-based clustering [12, 13, 2]. In this clustering, the similarity between two clusters C_i and C_j is solely determined through their relevant dimensions, i.e. the similarity of their subspaces S_i and S_j. It is formally defined by: $sim(C_i, C_j) = |S_i \cap S_j| / |S_i \cup S_j|$. Knowledge obtained in previous iterations of the concept analysis is included into the clustering process by encoding this knowledge as constraints. More concrete, we provide



(a) Naive dimension ordering (b) Concept-based ordering

Figure 4: Dimension ordering in parallel coordinates plots and their influence on visual interpretation

must-links and cannot-links, i.e. the user can specify which clusters belong to the same concept and which do not.

Concept shaping.
In the following, we describe how the preliminary concepts found in the previous step are concretized by determining the significant dimensions of a concept. These dimensions are the basis for specifying the semantics of the final concepts in the preceding concept analysis phase. Since the corresponding subspace clusters of a concept have different relevant dimensions, the significant dimensions cannot be determined automatically. *CoDA* provides a bar chart for each concept that visualizes the relevance of each dimension (cf. Fig. 3). This allows a visual discrimination of significant and non-significant dimensions, the latter ones are of no relevance for a concept. Based on this intuitive discrimination, the user can specify a threshold for each concept that determines the significant dimensions. Formally, the relevance of a single dimension d_i for a concept cpt and its assigned clusters $C_j = (O_j, S_j)$ with object set O_j and subspace set S_j, is determined by:

$$rel(d_i, cpt) = \frac{1}{\sum_{C_j \in cpt} |O_j|} \sum_{C_j \in cpt} |O_j| \cdot |\{d_i\} \cap S_j|$$

The output of this phase is a set of concepts and their selected significant dimensions.

2.2 Concept analysis

In the previous phase of *CoDA* the user determines the concepts; the second phase described in the following allows an in-depth analysis of these results (cf. Fig. 5). First, the analysis enables the user to comprehend the domain-specific semantic of a concept, e.g. by examining the actual characteristics of the clusters induced by the concept. Second, the user can improve the concept determination of subsequent steps by identifying any discrepancies in the current step.

Concept-centric parallel coordinates.
Our *CoDA* uses parallel coordinates to visualize the concepts and their induced subspace clusters in an intuitive way. Parallel coordinates are a technique to illustrate high-dimensional data sets [6]. For our concept analysis step, however, we face a particular problem. Because each concept is associated only with a subset of dimensions, i.e. its significant ones, an intuitive illustration is challenging. A naive use of parallel coordinates would lead to a representation where significant and non-significant dimensions are interweaved. Considering the example in Fig. 4(a) where two subspace clusters of the same concepts and with the relevant dimensions $\{1, 3, 6, 8\}$ are plotted. A visual interpretation of this plot and thus a knowledge extraction is difficult since the non-significant dimensions hinder a condensed view of the data. For a clear representation it is important to group the significant dimensions of a concept together. In Fig. 4(b) the dimensions are permuted such that $\{1, 3, 6, 8\}$ are adjacent.

With our *CoDA* the user is able to analyze inter-concept dependencies, i.e. several concepts (with different sets of significant dimensions) are visualized simultaneously. To facilitate a clear visual impression for the user *CoDA* performs a sophisticated arrangement of the dimensions, such that for each concept under consideration its significant dimensions are grouped together as good as possible. The arrangement of dimensions is easy to realize for each concept individually but when several concepts are considered simultanously the problem of arranging dimensions gets more complicated. Technically, the optimal ordering of dimensions is solved by using matrix bandwidth minimization techniques [8, 9].

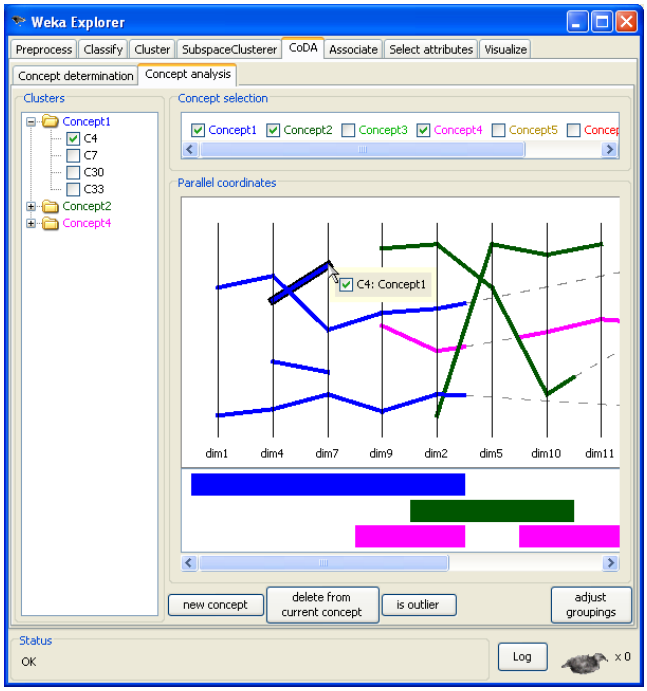


Figure 5: Concept analysis tab of *CoDA*

The concept analysis tab of *CoDA* is depicted in Fig. 5. The user is able to select a set of concepts to be analyzed with our tool. Based on the selected concepts and their significant dimensions the arrangement of the dimensions is automatically determined. Within the parallel coordinates diagram the clusters of the corresponding concepts are plotted. By using color codes the different concepts and their induced clusters can be distinguished. To enable a visual interpretation, *CoDA* describes a cluster by a single representation (e.g. the cluster mean) instead of all its objects, and we also skip the irrelevant dimensions of each subspace cluster. However, keep in mind that the relevant dimensions of the clusters do not necessarily correspond to the significant dimensions of the concepts. To provide the user with a comparison of these dimension sets and to give an easy overview for analyzing the inter-concept dependencies, *CoDA* additionally shows the significant dimensions of each concept in a bar diagram below the parallel coordinates.

User interaction for concept improvement.
By comparing the relevant dimensions of clusters and the significant dimensions of concepts, the user is able to detect any discrepancies in the concept determination so far. In Fig. 5 for example the cluster $C4$ fits not very well to its currently assigned concepts. Adjusting the concept determination based on the concept analysis is thus crucial for a meaningful overall interpretation.

The easiest way to modify the current concepts is by readjusting the significance thresholds in the concept determination tab (cf. Fig. 3). Thereby the user changes the significant dimensions of the concepts and consistency with the induced clusters can be realized. Note that this interaction does not influence the cluster grouping. For adjusting these groupings, *CoDA* implements more complex interactions such that the user is able to initiate a regrouping of the clusters to form novel concepts.

Consider the cluster $C4$ and the other cluster in the same subspace in Fig. 5. Based on the previous analysis and with the knowledge of the application domain, the user identifies that these two clusters do not belong to the current concept but they build an own concept. In *CoDA* these clusters can be selected and the user can enforce this set to represent a new concept (button 'new concept' in Fig. 5). Similarly, the user can resolve conflicts if a cluster is wrongly assigned to a concept (button 'delete from current concept'): the selected cluster has to be assigned to another concept. Even stricter, the user can classify clusters as outliers that do not belong to any concept (button 'is outlier'). The user's decision, which interaction is reasonable, can be further confirmed by a detailed analysis of each cluster individually. By clicking on single clusters a pop-up appears that does not just plot the single representative for the cluster but also the exact object values within the parallel coordinates plot.

After doing several of these interactions, the user can initiate a readjustment of the current cluster groupings (button 'adjust groupings'). As a result, refined and more sound concepts are identified. Technically, we realize the interactions and the regrouping by using constraint based clustering [12, 13, 2]. The different types of interactions are implemented with particular must-link and cannot-link constraints between the subspace clusters.

The refined concepts, i.e. the novel grouping of clusters, cause new and refined significant dimensions for each concept. Accordingly, *CoDA* guides the user to the concept determination tab where novel thresholds within the bar charts can potentially be set, realizing a cyclic dependency between the determination and analysis of concepts to increase the quality of each step. By performing multiple iterations of this process the user can gain a deeper understanding of the concept structure of large databases.

3. DEMONSTRATION SCENARIO

In the demonstration setup of *CoDA*[1] several real world data sets from the UCI KDD archive [3] can be examined with regard to their concept structure. On these datasets several subspace clustering techniques can be applied, allowing not only to examine the different datasets for their concepts but also the output of the clustering algorithms regarding their suitability for this process. The different phases of concept discovery in *CoDA* can be tested by the participants and they can verify the soundness of the results. Overall, *CoDA* supports the user to find and understand concepts in databases.

Future Work.
For future development *CoDA* can be integrated in the clustering process, such that the user feedback is already used while the clusters are determined. By this integration, the clustering result represents the concepts more precisely. In some domains concepts are interrelated, e.g. by a hierarchical structure. We plan to extent *CoDA* such that the user is enabled to derive hierarchical dependencies out of the concepts.

Acknowledgment
This work has been supported by the UMIC Research Centre, RWTH Aachen University, Germany.

[1] http://dme.rwth-aachen.de/OpenSubspace/CoDA

4. REFERENCES

[1] Y. Cui, X. Z. Fern, and J. G. Dy. Non-redundant multi-view clustering via orthogonalization. In *ICDM*, pages 133–142, 2007.

[2] I. Davidson. Clustering with constraints. In L. Liu and M. T. Özsu, editors, *Encyclopedia of Database Systems*, pages 393–396. Springer US, 2009.

[3] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[4] S. Günnemann, I. Färber, E. Müller, and T. Seidl. ASCLU: Alternative subspace clustering. In *MultiClust at KDD*, 2010.

[5] S. Günnemann, E. Müller, I. Färber, and T. Seidl. Detection of orthogonal concepts in subspaces of high dimensional data. In *CIKM*, pages 1317–1326, 2009.

[6] A. Inselberg and B. Dimsdale. Parallel coordinates: A tool for visualizing multi-dimensional geometry. In *IEEE Visualization*, pages 361–378, 1990.

[7] H.-P. Kriegel, P. Kröger, and A. Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *TKDD*, 3(1), 2009.

[8] E. Mäkinen and H. Siirtola. Reordering the reorderable matrix as an algorithmic problem. In *Diagrams*, pages 453–467, 2000.

[9] R. Martí, V. Campos, and E. Piñana. A branch and bound algorithm for the matrix bandwidth minimization. *European Journal of Operational Research*, 186(2):513–528, 2008.

[10] E. Müller, I. Assent, S. Günnemann, T. Jansen, and T. Seidl. OpenSubspace: An open source framework for evaluation and exploration of subspace clustering algorithms in weka. In *OSDM at PAKDD*, pages 2–13, 2009.

[11] E. Müller, S. Günnemann, I. Assent, and T. Seidl. Evaluating clustering in subspace projections of high dimensional data. *PVLDB*, 2(1):1270–1281, 2009.

[12] K. Wagstaff and C. Cardie. Clustering with instance-level constraints. In *ICML*, pages 1103–1110, 2000.

[13] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl. Constrained k-means clustering with background knowledge. In *ICML*, pages 577–584, 2001.