HISSCLU: A Hierarchical Density-Based Method for Semi-Supervised Clustering

Christian Böhm
Institute for Computer Science
University of Munich
boehm@dbs.ifi.lmu.de

Claudia Plant
Department of Neuroradiology
Technical University of Munich
plant@lrz.tum.de

ABSTRACT

In situations where class labels are known for a part of the objects, a cluster analysis respecting this information, i.e. semi-supervised clustering, can give insight into the class and cluster structure of a data set. Several semi-supervised clustering algorithms such as HMRF-K-Means [4], COP-K-Means [26] and the CCL-algorithm [18] have recently been proposed. Most of them extend well-known clustering methods (K-Means [22], Complete Link [17]) by enforcing two types of constraints: must-links between objects of the same class and cannot-links between objects of different classes. In this paper, we propose HISSCLU, a hierarchical, density-based method for semi-supervised clustering. Instead of deriving explicit constraints from the labeled objects, HISSCLU expands the clusters starting at all labeled objects simultaneously. During the expansion, class labels are assigned to the unlabeled objects most consistently with the cluster structure. Using this information the hierarchical cluster structure is determined. The result is visualized in a semi-supervised cluster diagram showing both cluster structure as well as class assignment. Compared to methods based on must-links and cannot-links, our method allows a better preservation of the actual cluster structure, particularly if the data set contains several distinct clusters of the same class (i.e. the intra-class data distribution is multimodal). HISSCLU has a determinate result, is efficient and robust against noise. The performance of our algorithm is shown in an extensive experimental evaluation on synthetic and real-world data sets.

1. INTRODUCTION

In many application domains, huge amounts of unlabeled data are available, e.g. unfiltered emails or metabolite concentrations from blood samples of thousands of patients. Labeling unlabeled data according to classes, e.g. regular - spam, healthy - disease A - disease B, is a complex task often requiring domain knowledge by human experts. Therefore class labels are often only given for a part of the objects. Consequently, semi-supervised learning, which considers both, labeled and unlabeled data has attracted much attention in the recent years [4, 13, 18, 26]. In this paper we focus on semi-supervised clustering, i.e. the use of given class labels (maybe only very few) to improve unsupervised clustering. Most approaches, e.g. [18, 26] and also [4, 5] to a certain extent, achieve this goal by enforcing two types of constraints during the clustering process: Cannot-links are applied to prevent objects with different labels from being grouped together. Must-links between identically labeled objects force them into a common cluster.

In this paper, we propose HISSCLU a hierarchical density-based approach to semi-supervised clustering which avoids the use of explicit constraints due to the following reasons: To gain insight into the modality of the data set and to be informed about previously unknown sub-classes, it is not helpful to apply must-links forcing objects into the same cluster which are actually dissimilar. As running example consider the 2-d data set displayed in Figure 1 consisting of four clusters of different object densities. In addition for seven objects a class label is known. These labeled objects from two different classes are marked by the symbols X and $\square$. Both classes are multimodal, i.e. they consist of different sub-classes distributed over different clusters. In addition we have one labeled object of class X which is an outlier. Must-link constraints force this object into a common cluster with the other labeled objects of its class. The information that this object is completely different to all other objects in the data set is lost.

Cannot-links remove any information about how similar the objects of different classes actually are. In our example one cluster is shared by three labeled objects of both classes which are very similar. Taking more than two classes into account, the important information that objects of some classes A and B are more similar than objects of classes C and D (indicating a hierarchical relationship of the classes A and B) would be lost by the enforcement of cannot-links. Instead of deriving constraints from the labeled objects, HISSCLU expands the clusters starting at all labeled objects simultaneously. As an additional value add over comparative methods, HISSCLU assigns class labels to the unlabeled objects during the cluster expansion. The labeling is maximally consistent to both, the cluster structure of all objects and the given class labels of the labeled objects. The result of HISSCLU is visualized in the semi-supervised clus-

EDBT'08, March 25–30, 2008, Nantes, France.

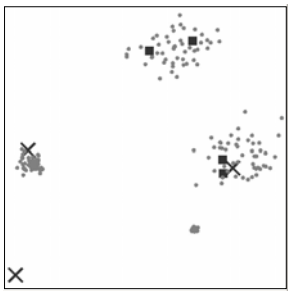


Figure 1: Running Example.

ter diagram giving a concise illustration of the hierarchical class and cluster structure. The cluster diagram provides answers to the following questions even for high-dimensional and moderate to large scale data sets:

- Q1: Are there clusters shared by more than one class?
- Q2: Are there multi-modal classes distributed over more than one cluster?
- Q3: Which are the most similar/dissimilar classes?
- Q4: Is there any class hierarchy and how well does it correspond to the cluster hierarchy?

To determine the hierarchical class and cluster structure we found our approach on a hierarchical density-based clustering notion ([2, 14]). More specifically we give the following problem specification for clustering.

Problem Specification. The objective of our method is to determine a hierarchical clustering of the labeled and unlabeled objects with maximally large class pure sub-clusters of high density. We can identify the following two goals:

- G1: High density: Clusters are regions of high density separated by regions of lower density.
- G2: Class-purity: As many clusters as possible are uniformly labeled.

For a hierarchical clustering approach, G1 means that sub-clusters have a higher density than the super-clusters in which they are nested (cf. [2]). G2 means that maximally large sub-clusters are uniformly labeled. In this paper, we propose an efficient algorithm that exploits the concepts of density-based clustering to address G1 and carefully applies a local distance weighting technique to increase the class-purity of clusters (G2).

Solution Overview. We can exploit the cluster hierarchy for cluster expansion and labeling of the unlabeled objects: Starting from each object O, we can go upwards in the cluster hierarchy until we have reached an inner node n which contains at least one labeled object in its subtree. In many cases, the contained labeled objects are class-pure, and we can safely assign O to this sub-cluster and assign the corresponding class label to O. Whenever the subtree rooted by n is class-impure, we additionally consider spatial coherency for clustering and labeling thus assigning areas of neighboring objects to the same cluster. The border between two areas of differently labeled objects should be positioned in the area of least data density. Since in areas of a relatively uniform data density the border is rather random, we propose the careful use of a local weighting function overriding the random differences of inter-point distances but not overriding actual cluster-boundaries.

Notations. We are considering a database DB of objects from an Euclidean vector space. In addition to d feature attributes, for some of the objects a categorial class label $class(L)$ is given. We call them the pre-labeled objects $L \in \mathbb{L}$. During the run of the algorithm, also the previously unlabeled objects $U \in \mathbb{U}$ obtain a class label. For distance computation $dist(P,Q)$ the Euclidean distance $||P-Q||^2$ of the d feature attributes is used and we will also use the notions of ϵ-neighborhood $N_\epsilon(P) = \{Q \in DB \,|\, dist(P,Q) \leq \epsilon\}$ and the set of k-nearest neighbors of an object P, denoted by $NN_k(P)$. We use the notion $nn\text{-}dist_k(P)$ for the distance of the k-nearest neighbor of an object $P \in DB$.

The paper is organized as follows: In Section 2 we briefly survey related work and summarize our contributions. In Section 3 we introduce our algorithm for cluster expansion and elaborate on labeling of the unlabeled objects. In Section 4 we propose a local weighting function to improve clustering stability. Section 5 illustrates the visualization of the result before we provide an extensive experimental evaluation in Section 6. In Section 7 we summarize the benefits of HISSCLU for data mining. Section 8 concludes the paper.

2. RELATED WORK

In this section we give a brief survey on related work on semi-supervised clustering and label propagation. Since we found our approach on the density-based clustering notion, we also survey basic concepts of density-based clustering.

2.1 Semi-Supervised Clustering

Several constraint-based approaches in the field of semi-supervised clustering have appeared. Most of them extend existing clustering methods, such as Complete Link to incorporate constraints, e.g. [18] and [9] for numerical constraints. In the CCL algorithm [18] complete-link clustering [17] is applied after replacing the Euclidean distance by a shortest path algorithm. The distance matrix is modified by setting the distance of all pairs of labeled objects to zero and the distance between all pairs of labeled objects of different classes to a value larger than the maximal distance appearing in the data set. Due to this rigid and global transformation of the data space, objects of the same class are forced to be in the same cluster. COP-K-Means [26] is a K-Means [22] based algorithm enforcing constraints. Must-link constraints are established between all pairs of identically labeled objects and cannot-links between all pairs of differently labeled objects. Objects are assigned to clusters without violating any of the constraints.

Recently, constraints have been used in a softer way to improve the clustering result, e.g. by using probabilistic models [21] or fuzzy clustering [20]. In [5] the authors propose MPC-K-Means, a K-Means based algorithm that considers both, constraints and the data distribution to assign objects to clusters. A cost function for violating must-link and cannot-link constraints is defined. The clustering objective function minimizes both, the link violation cost and

the deviation of the objects from the cluster centers. In addition a metric learning step is performed after each iteration to globally adapt a weighted Euclidean distance. In [4] the authors generalize this technique proposing a probabilistic framework for semi-supervised clustering to additionally support several non-Euclidean distance measures e.g. cosine similarity for text data.

2.2 Label Propagation

The label propagation algorithm [27] is related to our approach since HISSCLU assigns class labels to the unlabeled objects during the clustering process. Label propagation first constructs an $n \times n$ similarity matrix T (n being the number of all – labeled and unlabeled – objects) and a second matrix Y ($n \times c, c$ being the number of classes) for fuzzy class assignment with the information in $Y_{i,j}$ indicating to what degree Object P_i belongs to Class c_j. Then labels are propagated by iteratively multiplying $Y := T \cdot Y$ until matrix Y converges. This algorithm requires a storage complexity of $O(n^2)$ and is, hence, not scalable to database environments. Moreover, no clustering method and no visualization technique is proposed, which is the main focus of our approach.

2.3 Density Based Clustering

Density-based clustering algorithms, such as DBSCAN [14], Single-Link [17] and OPTICS [2] find clusters of arbitrary shape and number. Clusters are connected dense regions in the feature space that are separated by regions of lower density. In the algorithm DBSCAN [14] this idea is formalized using two parameters, $MinPts$ specifying a minimum number of objects, and ϵ, the radius of a hypersphere. These two parameters determine a density threshold for clustering. An object is called a *core object* of an $(\epsilon, MinPts)$-cluster if there are at least $MinPts$ objects in the ϵ-neighborhood. If one object P is in the ϵ-neighborhood of a core-object Q, then P is said to be *directly density-reachable* from Q. The *density-connectivity* is the symmetric, transitive closure of the *direct density reachability*, and a *density-based* $(\epsilon, MinPts)$*-cluster* is defined as a maximal set of density-connected objects. DBSCAN determines a non-hierarchical, disjoint partitioning of the data set. In contrast to many other partitioning clustering methods such as K-Means and K-Medoid methods, DBSCAN is determinate and robust against noise objects.

OPTICS [2] is a hierarchical extension of DBSCAN but also related to Single-Link [17]. The main idea of OPTICS is to compute a complex hierarchical cluster structure, i.e. all possible clusterings with the parameter ϵ varying from 0 to a given ϵ_{max} simultaneously during a single traversal of the data set. The output of OPTICS is a linear order of the data objects according their hierarchical cluster structure which is visualized in the reachability-plot. OPTICS is equivalent to Single-Link if the $MinPts$-Parameter of OPTICS is set to 1. In this case, OPTICS has the same drawbacks like Single-Link, i.e. missing stability with respect to noise objects and the so-called Single-Link effect: Whenever two actual clusters are connected by a small chain of equidistant objects, the two clusters cannot be separated. OPTICS overcomes this drawback if $MinPts$ is set to higher values.

Compared to DBSCAN, OPTICS turns the definitions of core-objects and density-connectivity around: Instead of specifying a distance parameter and defining whether an object is a core-object or two objects are density-connected, OPTICS defines the core-distance of an object P as the minimal distance ϵ_P from which DBSCAN would consider P as a core object. The reachability distance is analogously defined as the minimal distance $\epsilon_{P,Q}$ between P and Q starting from which DBSCAN would consider these objects as directly density-connected. We summarize these two definitions formally:

DEFINITION 1. *(Core Distance)*
The core distance *of object $O \in DB$ w.r.t. $MinPts \in \mathbb{N}$ is defined as*

$$Core_{MinPts}(O) = nn\text{-}dist_{MinPts}(O).$$

The core distance of an object O measures the density around O. It is defined as the $MinPts$-nearest neighbor distance of O.

DEFINITION 2. *(Reachability Distance)*
The reachability distance *of an object $P \in DB$ relative from object $O \in DB$ w.r.t. $MinPts \in \mathbb{N}$ is defined as*

$$Reach_{MinPts}(O, P) = \max\{Core_{MinPts,\epsilon}(O), dist(O,P).\}$$

Let us note that the original definitions of OPTICS use an additional parameter which is left out here for simplicity reasons. This parameter specifies the maximum reachability distance for cluster expansion and should be set to a high value. The OPTICS algorithm operates on a *seed list SL* which is initialized with an arbitrary, unprocessed object whenever empty. The unprocessed objects are stored in the seed list, ordered by a criterion which is described later. In the main loop of the algorithm, the algorithm selects the top (minimum) element T of SL and appends it to the output. The ordering criterion for each object P in the seed list is the minimum of all reachability distances from any of the objects in the output to P:

$$P.\text{order} = \min_{Q \in \text{ output}} \{Reach_{MinPts}(P, Q).\}$$

Consequently, the seed list is updated after each iteration of the loop: Some new objects which are density-reachable from T may be inserted, and P.order is updated for all those objects for which $Reach_{MinPts}(T, P)$ is less than the previously stored P.order.

2.4 Contributions

Our new method HISSCLU has the following main advantages over previous methods:

- The result is determinate, robust against noise, and the method does not favor clusters of convex shape.
- In contrast to constraint-based methods the original cluster structure is preserved. Therefore, the result gives valuable information about previously unknown class and cluster hierarchies.
- Our method assigns class labels to the unlabeled objects in a way which is maximally consistent with the cluster structure and observes spatial coherency of class labeling.
- We propose a visualization method which allows a clear and concise illustration of the semi-supervised cluster structure even for moderate to high numbers of objects.

3. CLUSTER EXPANSION

In this section we elaborate how our algorithm expands the clusters starting at each of the pre-labeled objects simultaneously and how the class labels are naturally assigned to the unlabeled objects during this process. Two objects share a common density-based cluster if they are density-connected, i.e. if there exists a path of core objects between them and there is no higher distance between path-neighbors than ϵ. For an unlabeled object P we consider the paths each starting from one of the pre-labeled objects $L \in \mathbb{L}$ and ending in P. The object P belongs to the cluster and adopts the class label of that pre-labeled object L for which a path with minimum ϵ exists. We define in the following the notions of a *path* and the *path reachability distance* which corresponds to the minimum ϵ in the DBSCAN algorithm. First of all, we define a path to be an arbitrary sequence of objects starting with a pre-labeled object and, apart from that, containing only unlabeled objects, because we are only interested in such sequences here.

DEFINITION 3. *(Path)*
Let $S = \langle P_1, P_2, ..., P_n \rangle$ be a sequence of objects, where P_1 is a labeled object and $P_2, ..., P_n$ are distinct, unlabeled objects. Then we call S a path from P_1 to P_n.

We now have to define an ordering predicate denoted by $\overset{<}{}_{PRD}$ to compare two or more paths, deciding which of them is *shorter* or *better*. The main criterion for the comparison is the maximum distance between subsequent objects on the path. If two paths share the same maximum distance we consider the second largest distance on the path, and so on. For simplicity reasons we assume for the following definitions that the distance between every pair of objects is different, i.e. $dist(P,Q) = dist(R,Q) \Rightarrow P = R$. In practice tie situations can be solved in a nondeterministic way. Let also $\epsilon \in \mathbb{R}_0^+$, $MinPts \in \mathbb{N}$.

DEFINITION 4. *(Restricted path reachability distance)*
Let $S = \langle P_1, ..., P_n \rangle$ be a path. Then $I(\epsilon) \subseteq \{1, ..., n-1\}$ is the index set such that $i \in I(\epsilon) :\Leftrightarrow Reach_{MinPts}(P_i, P_{i+1}) < \epsilon$. The restricted path reachability distance of a path S w.r.t. ϵ and $MinPts$, denoted by $rPRD_{\epsilon,MinPts}(S)$ is the maximum of those reachability distances of two adjacent objects on S, less than ϵ. $rPRD_{\epsilon,MinPts}(S) =$

$$\begin{cases} 0 \text{ if } I(\epsilon) = \emptyset \\ \max_{i \in I(\epsilon)}\{Reach_{MinPts}(P_i, P_{i+1})\} \text{ otherwise.} \end{cases}$$

The above definition allows us to determine the actual maximum distance on the path by setting $\epsilon = \infty$ or also to determine the maximum distance below some specified threshold ϵ'. We now inductively define the ordering of paths.

DEFINITION 5. *(Ordering different paths)*
Let S_1 and S_2 be two paths. We say that S_1 is of less path reachability distance ($S_1 \overset{<}{}_{rPRD(\epsilon,MinPts)} S_2$) if one of the following conditions applies
1. $rPRD_{\epsilon,MinPts}(S_1) < rPRD_{\epsilon,MinPts}(S_2)$
2. $rPRD_{\epsilon,MinPts}(S_1) = rPRD_{\epsilon,MinPts}(S_2) =: \epsilon' \wedge S_1 \overset{<}{}_{rPRD(\epsilon',MinPts)} S_2$.

The parameter ϵ is now omitted by setting
1. $PRD_{MinPts}(S) := rPRD_{\infty,MinPts}(S)$ and
2. $S_1 \overset{<}{}_{PRD(MinPts)} S_2 :\Leftrightarrow S_1 \overset{<}{}_{rPRD(\infty,MinPts)} S_2$.
We can now define the minimum path between two objects as the minimum according to the $\overset{<}{}_{PRD(MinPts)}$ relation.

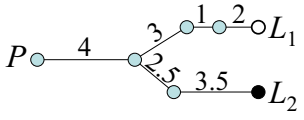


Figure 2: Ordering Different Paths.

DEFINITION 6. *(Minimum path from L to P)*
Let L be a pre-labeled object and P an unlabeled object. Let Σ be the set of all paths from L to P. Then the minimum path $MinPath_{MinPts}(L,P)$ is the path $S \in \Sigma$ for which the following condition holds:

$$\forall S' \in (\Sigma \setminus \{S\}) : S \overset{<}{}_{PRD(MinPts)} S'.$$

Finally, we define that P adopts the class label of that pre-labeled object $L \in \mathbb{L}$ having the *smallest* (according to $\overset{<}{}_{PRD(MinPts)}$) minimum path to P.

DEFINITION 7. *(Assignment of class labels to P)*
Let $P \in \mathbb{U}$. P is assigned to class $C :\Leftrightarrow \exists L \in \mathbb{L}$ such that
1. $C = class(L)$ and
2. $\forall L' \in \mathbb{L} \setminus \{L\}$:
$MinPath_{MinPts}(L,P) \overset{<}{}_{PRD(MinPts)} MinPath_{MinPts}(L',P)$.

For simplicity we drop the index and write $\overset{<}{}_{PRD}$ and $MinPath$ wherever non-ambiguous. An example of two minimum paths is depicted in Figure 2, where we have an unlabeled object P, two pre-labeled objects L_1 and L_2. The maximum path segment (4) is shared by both paths. The second maximum (3 vs. 3.5) is applied, and, therefore, L_1 is the winner: P adopts the class label of L_1. Before providing an efficient algorithm for cluster expansion, we prove an interesting property, the consistency of class labeling with the cluster structure:

THEOREM 1. *(Cluster consistency of object labeling)*
Let $K \subseteq \mathbb{L} \cup \mathbb{U}$ be a cluster that can be detected by DBSCAN with arbitrarily chosen parameters ϵ and $MinPts$. If K does not contain any pre-labeled objects from two or more classes then all objects of K obtain uniform labels.

PROOF. (1) Let $K \cap \mathbb{L} \neq \emptyset$.
(a) Since all objects in cluster K are density-connected we know that the path reachability distance of the minimum path between any unlabeled objects and any pre-labeled objects is less or equal ϵ. (b) As K is the maximum set of density-connected objects we know further that all path reachability distances between unlabeled objects in the cluster and pre-labeled objects outside the cluster are larger than ϵ. From (a) and (b) it follows that the smallest minimum path from any object $P \in K \cap \mathbb{U}$ goes to one of the pre-labeled objects $L \in K \cap \mathbb{L}$ of the cluster. All objects adopt the same label.
(2) Let $K \cap \mathbb{L} = \emptyset$.
Assume we have two different objects $P, Q \in (\mathbb{U} \cap K)$ which obtain different labels by Definition 7. Then P and Q must have different MinPaths S_P, S_Q to different winner objects L_P and L_Q. S_P and S_Q must particularly be different in the larger subsequences S'_P and S'_Q outside the cluster. W.l.o.g. let $S'_P \overset{<}{}_{PRD} S'_Q$. Then we could replace S_Q by some path $\langle S_P, P, ..., Q \rangle$ for which we know $\langle S_P, P, ..., Q \rangle \overset{<}{}_{PRD} S_Q$ which is a contradiction to the minimality of S_Q. □

```
algorithm expansion(DB, 𝕃, MinPts, ϵ, )
  List seedList;
  forall L ∈ 𝕃
    L.reachDist:= UNDEFINED;
    neighbors :=N_ϵ(L);
    updateSeedList(neighbors, L);
  end for
  while not seedList.empty() do
    currentObject = seedList.getMin();
    seedList.deleteMin();
    currentObject.setCoreDist(ϵ, MinPts);
    neighbors :=N_ϵ(currentObject);
    seedList.update(neighbors, currentObject);
  end while
end expansion

procedure updateSeedList(objects neighbors, object P);
  core:= P.CoreDist;
  forall N ∈ neighbors do
    rd: = max(core, dist(P, N);
    if not N ∈ seedList
      N.reachDist := rd;
      N.label := P.label;
      seedList.add(N);
    else
      if rd < N.reachDist
        N.reachDist: = rd;
        N.label: = P.label;
  end for
end updateSeedList
```

Figure 3: Cluster Expansion Algorithm.

Our algorithm (cf. Figure 3) expands clusters starting from all pre-labeled objects simultaneously. In the first step, all objects in the ϵ-neighborhood of the pre-labeled objects are added to the *seedList*. This is implemented by the method *updateSeedList*. The *seedList* is an ordered list of objects according their minimum reachability distance w.r.t. any object processed before. Objects which have not been in the *seedList* before are inserted and get the class label of the object from which they are density reachable. For objects that are already in the *seedList* the algorithm determines if they are density reachable with a smaller distance ϵ now. If this is the case, their reachability distance and class label are updated. In each iteration the algorithm removes and processes that object from the *seedList* which has the smallest reachability distance from all previously considered objects. Our algorithm is efficient with a worst-case time complexity in $O(n^2)$. Finally we prove that our algorithm assigns exactly the same label to each data object as provided by Definition 7. The completeness property of labeling follows directly from Lemma 2.

THEOREM 2. *(Correctness of labeling)*
An unlabeled object P which is a successor of a pre-labeled object L in the cluster expansion algorithm has minimum path reachability distance to L.

PROOF. Assume there exists an object L' with $MinPath(P, L') \overset{<}{}_{PRD} MinPath(P, L)$. Then, by definition of $\overset{<}{}_{PRD}$, we know that either (1) the maximum segment on $MinPath(P, L)$ is larger than that on $MinPath(P, L')$ or (2) $\exists i \in \mathbb{N}$ that the first, second, ..., $(i-1)$ maximal segments are identical and the i-maximal segment is larger on $MinPath(P, L)$ than on $MinPath(P, L')$. In case (1) we know that the maximum segment on $MinPath(P, L')$ has been expanded before the maximum segment on $MinPath(P, L)$. As all other segments on $MinPath(P, L')$ have smaller reachability distance than the maximum segment on $MinPath(P, L)$, P must be assigned to L' which is a contradiction. In case (2) we know that the identical maximum segments are expanded by the algorithm after the distinguishing segment (i-maximum). Analogously to (1) the object P must be assigned to L' which is a contradiction to the assumption. □

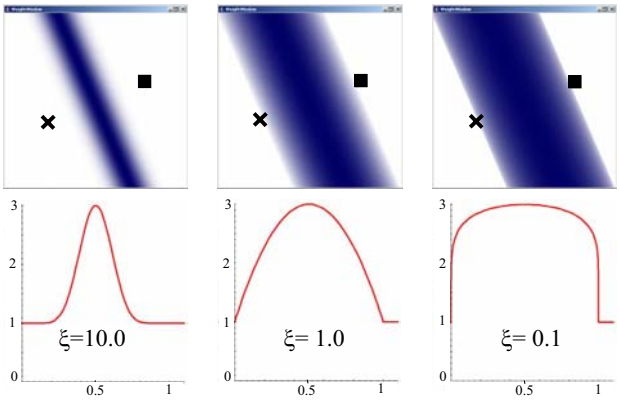


Figure 4: Different Values for ξ.

4. LOCAL LABEL-BASED DISTANCE WEIGHTING

If there are differently pre-labeled objects inside a cluster our cluster expansion algorithm assigns class labels in a way which is geometrically contiguous. The border line between the areas of different label assignments, however, is in this case rather random and the small, accidental variance of the distances decides about the border between regions of different class labels. In this section, we propose a weighting function to clear situations where no natural boundaries of low density between differently pre-labeled objects exist. Our idea is to define a continuous (smooth) weighting function which has a user-defined maximum (ρ) at the perpendicular bisector plane between differently pre-labeled objects and monotonically decreases with increasing distance from the bisector. The weighting function is 1.0 at the pre-labeled objects and in the areas between pre-labeled objects of the same class. Let A and $B \in \mathbb{L}$ be two differently pre-labeled objects that define a perpendicular bisector and one object P which defines the position for which the weight $w_{A,B}(P)$ should be determined. The distance of P to the bisector plane can be computed by the projection P' of object P onto the line between A and B. The distance $d_{AB}(A, P)$ between P' and A is given by the absolute value of the scalar product $d_{AB}(A, P) = |\langle A - P, \frac{A-B}{||A-B||}\rangle|$, and the distance between P' and B equals $d_{AB}(B, P) = |\langle B - P, \frac{A-B}{||A-B||}\rangle|$. By comparing $d_{AB}(A, P)$ and $d_{AB}(B, P)$ we can define an auxiliary function h which equals 1 at the bisector plane where $d_{AB}(A, P) = d_{AB}(B, P)$ and which equals 0 at the two planes parallel to the bisector passing through A and

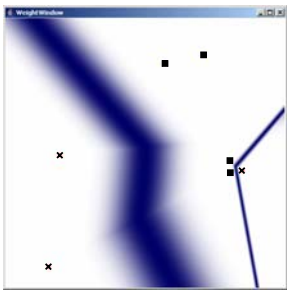


Figure 5: Weighting on Running Example.

B, respectively:

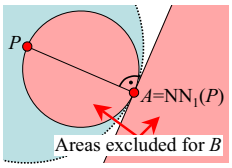
$$h_{A,B}(P) = \frac{d_{AB}(A,P) \cdot d_{AB}(B,P)}{\frac{1}{4}(d_{AB}(A,P) + d_{AB}(B,P))^2}$$

We can easily substitute h to obtain our weight function with the above mentioned value range $[1..\rho]$:

$$w_{A,B}(P) = (h_{A,B}(P))^{\xi} \cdot (\rho - 1) + 1$$

In this formula, $\rho \geq 1$ is a parameter controlling the maximum of the weight function. The exponent $\xi \in \mathbb{R}^+$ controls how fast the weight decreases with increasing distance from the perpendicular plane: $\xi = 1$ corresponds to a parabolic decrease, whereas $\xi > 1$ facilitates a faster decrease (i.e. the perpendicular plane becomes sharper) and $0 < \xi < 1$ a slower decrease. Figure 4 shows an example in 2-d space with different ξ settings ($\xi \in \{0.1, 1.0, 10.0\}$).

Given the object P at which the weighting function has to be evaluated, we now determine the suitable pair $(A, B) \in \mathbb{L} \times \mathbb{L}$ of pre-labeled objects which has to be applied in the formula above. Intuitively, we have to apply that pair (A, B) where A and B have different labels and which defines that perpendicular bisector plane to which P is closest (among all such pairs). We have an additional side condition that P is between A and B when projected onto line AB, and not beyond A or beyond B. One of the objects (say A) is the nearest neighbor of P in the set of pre-labeled objects $\mathbb{L}$, because no planes can be closer to P than those which border the Voronoi cell in which P is positioned. This Voronoi cell is defined by the nearest neighbor of P. To determine the second object B we have to obey the side condition. B must be searched such that the projected P is between A and B.



See the illustration on the left for two different loci which are excluded for B: If B is taken from the semi-space right from the perpendicular plane passing through A, then the projection of P is beyond A. And if B is taken from the inside of the sphere between P and A (which actually corresponds to a Thales' circle) then the projection of P will be beyond B. The latter situation (object B inside the small sphere) cannot occur anyway, because A is the nearest neighbor of P in $\mathbb{L}$, and therefore, there cannot be any other object inside the larger dotted sphere (the nearest neighbor sphere) which completely encloses the Thales' circle. The first side condition can be checked by the sign of the scalar product $\langle P - A, B - A \rangle$ which is negative in the excluded semi-space. To select B we have to remove A and all objects in the excluded semi-space from $\mathbb{L}$. We have to select that of the remaining objects which maximizes the weight function $w_{A,B}(P)$:

$$B = \underset{L \in \mathbb{L} \setminus \{A\}, \langle P-A, L-A \rangle > 0}{\operatorname{argmax}} \left\{ \frac{d_{AL}(A,P) \cdot d_{AL}(L,P)}{\frac{1}{4} \cdot (d_{AL}(A,P) + d_{AL}(L,P))^2} \right\}.$$

Figure 5 visualizes the overall weight function $w(P) = w_{A,B}(P)$ on our 2-d running example first presented in Figure 1. The weight function $w(P)$ is applied for weighting the distance between two different objects P_i and P_j. To obtain a stable and consistent result we have to use the maximum weight when substituting every point x on the line segment $[P_i, P_j]$ in $w(x)$. Due to monotonicity we can derive the following:
1. If P_i and P_j have different nearest neighbors in $\mathbb{L}$ with different class labels, then the line segment crosses one of the perpendicular bisector planes, and therefore, the maximum weight corresponds to ρ.
2. Otherwise, the maximum weight function must be at either of the two end points P_i or P_j because inside the same Voronoi cell, the weight function is monotonic.
We define the following weighting function for a pair of objects:

$$w(P_i, P_j) = \begin{cases} \rho \text{ if class } (NN(P_i)) \neq \text{class } (NN(P_j)) \\ \max\{w(P_i), w(P_j)\} \text{ otherwise.} \end{cases}$$

5. VISUALIZATION

For visualization we merge the $|\mathbb{L}|$ different cluster order lists into a common one. Some of them have to be partially reordered because the start object changes from the pre-labeled object into that object which is closest to the previous cluster in the overall order. This does not affect the overall runtime complexity of $O(n^2)$.

In the semi-supervised cluster diagram each object is represented by a histogram bin with a length corresponding to the reachability distance of the object in the final (merged) cluster order. Therefore, clusters of the data set can be recognized as valleys in the reachability area of the cluster diagram. Hierarchically nested clusters correspond to sub-valleys in a common valley delimited by higher peaks than those between the sub-valleys. The class labeling is coded in different colors. Therefore, the consistence between cluster and class structure can be easily recognized in the cluster diagram. To mark the pre-labeled objects in the diagram, we stretch the corresponding bins slightly below the x-axis of the diagram. To facilitate the evaluation of our tech-

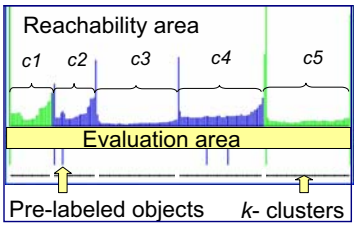
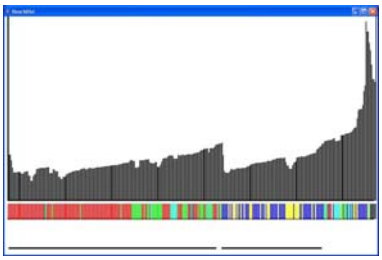
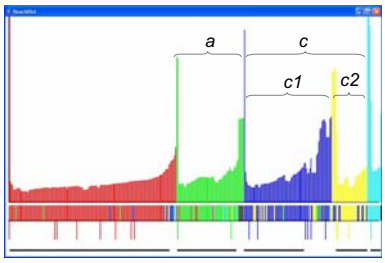


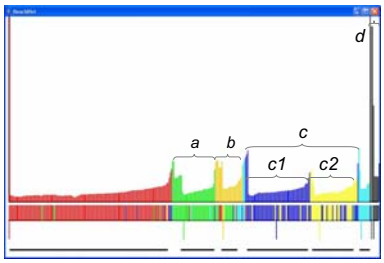
Figure 6: Cluster Diagram of Running Example.



(a) OPTICS Reachability-plot.

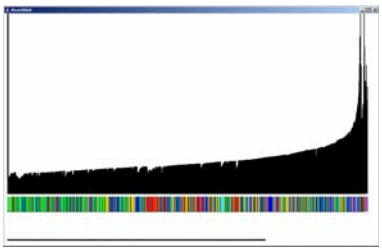


(b) 20 pre-labeled objects (random).

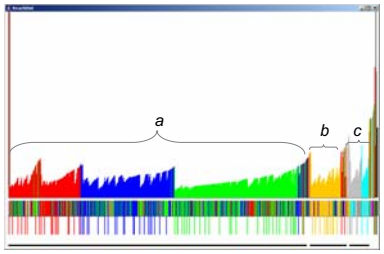


(c) 8 pre-labeled objects (one per class).

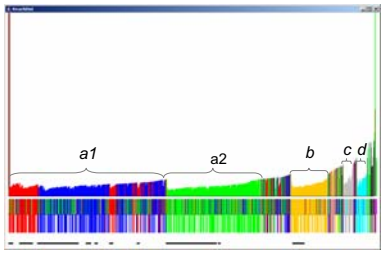
Figure 7: Visualizing Semi-supervised Class- and Cluster-hierarchies: Results on Ecoli Data.



(a) OPTICS Reachability-plot.



(b) 100 pre-labeled objects (random).



(c) 500 pre-labeled objects (random).

Figure 8: Visualizing Semi-supervised Class- and Cluster-hierarchies: Results on Yeast Data.

nique we extract clusters using one certain density threshold $\epsilon = k \cdot maxRd$ (where $maxRd$ denotes the maximum reachability distance) which we call *k-clusters*. The k-clusters are depicted as horizontal lines underneath the diagram. We also draw the histogram bars in the color coding the true class label of the objects in the evaluation area below the cluster diagram, if this information is available.

In Figure 6 the annotated cluster diagram of our running example with 5 extracted clusters is depicted. The cluster diagram shows interesting properties of the data set: Both classes are multimodal, one class is distributed over the clusters c1 and c5, the other class over c2, c3, c4. The clusters c1 and c2 are the most similar clusters among all extracted clusters, i.e. separated by the smallest reachability distance and contain differently pre-labeled objects. For comparison see also Figure 1 depicting the data set.

6. EXPERIMENTAL EVALUATION

For evaluation and for comparison with partitioning methods we extract clusters form the cluster diagram using a certain density threshold. To automatically extract clusters from the OPTICS reachability-plot, the algorithms ξ-cluster [2] and cluster-tree [25] have been proposed. These algorithms can also be applied to a cluster diagram but they do not extract a flat cluster structure. Therefore, we just horizontally cut the reachability-plot and the cluster diagram. Let d_{max} be the maximum reachability distance. We require that clusters are separated by a distance of at least $d_{sep} = k \cdot d_{max}$, with $k \in [0..1]$. In addition, we require that each cluster has at least $MinPts$ objects.

DEFINITION 8. *(k-clustering)*
Let DB be a set of objects and d_{max} be their maximum reachability distance $k, n, MinPts \in \mathbb{N}$ and $n \geq MinPts$. A sequence $S =< S_1, ...S_n >$ in the reachability-plot or the cluster diagram of DB is called a k-cluster if
1. $Reach(S_1) \geq k \cdot d_{max}$.
2. $Reach(S_2), ...Reach(S_n) < k \cdot d_{max}$.
Let $K = \{k_1, ..., k_n\}$ be the set of all k-clusters in DB. All objects $o \in DB$ with $o \notin k_i$ are called noise objects. Let N be the set of noise objects. We call $K \cup N$ a k-clustering.

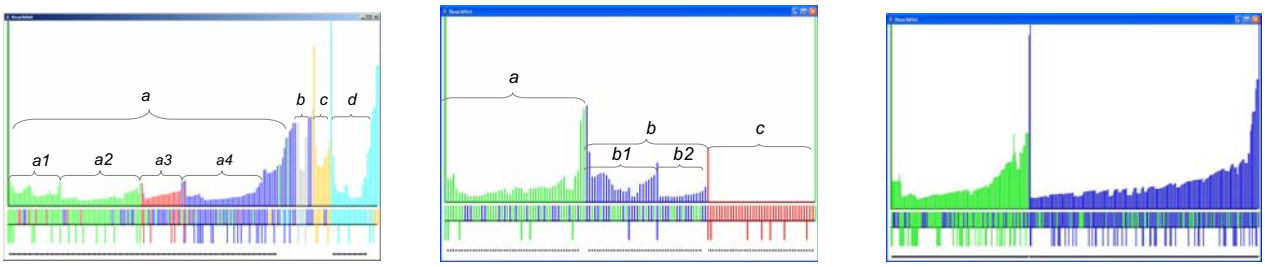
For comparison with partitioning methods we compute the Mutual Information of the k-clustering [11].

DEFINITION 9. *(Mutual Information of the k-clustering)*
Let $C = \{c_1, ...c_i\}$ be the set of classes, $K = \{k_1, ...k_j\}$ be the set of k-clusters and O_k be the set of cluster objects, i. e $O_k = \{o \in DB | o \in k_i\}$. Let $(h(c_i, k_j))$ be the number of objects of class c_i assigned to cluster k_j, $h(k_j)$ the total number of objects assigned to cluster k_j and $h(c_i)$ the total number of objects belonging to class c_i.

$$MI = -\sum_{i=1}^{|C|} \frac{h(c_i)}{|O_k|} \cdot \log_2 \frac{h(c_i)}{|O_k|} + \sum_{i=1}^{|C|} \sum_{j=1}^{|K|} \frac{h(c_i, k_j)}{|O_k|} \cdot \log_2 \frac{h(c_i, k_j)}{h(k_j)}.$$

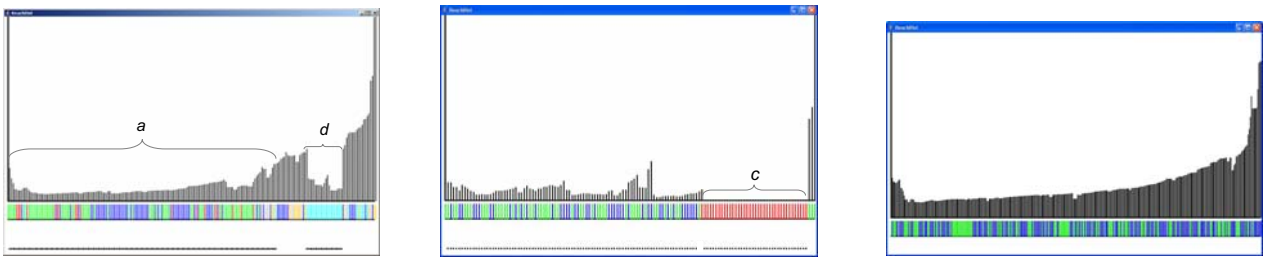
The Mutual Information $MI \in \mathbb{R}^+$ reflects to which degree a k-clustering corresponds to the class label distribution. We scale this quality measure in the range of [0..1].

In the following, we present results on synthetic data, various data sets obtained from the UCI machine learning repository [6] (Ecoli, Yeast, Glass, Liver) and on high dimensional metabolic data. In Section 6.1 we demonstrate that the cluster diagram provides valuable information on the hierarchical class and cluster structure. We compare the cluster diagram with the OPTICS reachability plot here, since no



(a) Glass data. (b) Metabolic data. (c) Liver data.

Figure 9: Results on Glass, Metabolic and Liver Data.



(a) Glass data. (b) Metabolic data. (c) Liver data.

Figure 10: OPTICS reachability-plot of Glass, Metabolic and Liver Data.

semi-supervised clustering method offers a visualization. In Section 6.2 and 6.3 we compare the performance of HISSCLU with COP-K-Means and MPC-K-Means in terms of clustering quality and in Section 6.4 we give hints on parameter selection.

6.1 Visualizing Class- and Cluster-Hierarchies

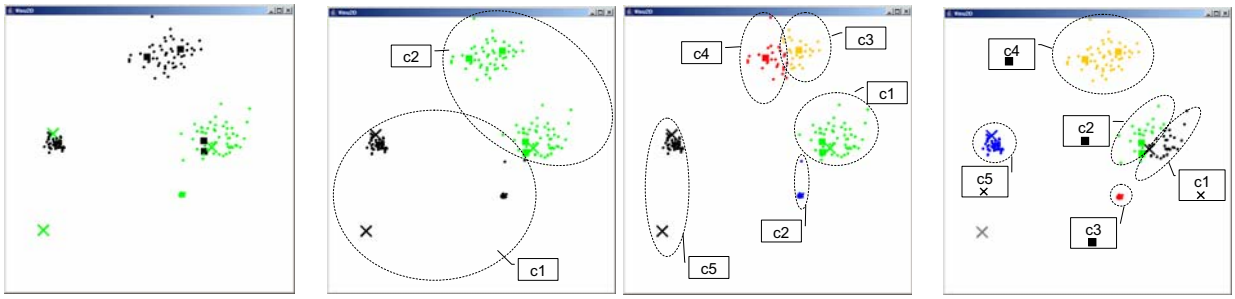
6.1.1 *Ecoli Data*

In Figure 7 the OPTICS reachability-plot and 2 cluster diagrams generated by HISSCLU of the Ecoli data set are depicted ($MinPts = 5, \rho = 2, \xi = 0.5$). We extracted clusters for $k = 0.2$. This 7-dimensional data set on predicting protein localization sites with 336 data objects and 8 classes is highly unbalanced having from 2 to 142 objects per class. The unsupervised reachability-plot in Figure 7(a) shows 2 k-clusters each containing objects of mainly two classes and a large amount of noise. In Figure 7(b) the cluster diagram using 20 randomly sampled objects as pre-labeled objects is shown. We extracted 5 clusters corresponding to the 5 largest classes. It can be seen that some of the classes are multimodal, e.g. the class of periplasm proteins forming cluster (a). The classes of inner membrane proteins without signal sequence (cluster (c1)) and inner membrane proteins with an uncleavable signal sequence (cluster (c2)) are the most similar classes in this data set. The maximum separating reachability distance between these classes is the smallest one among all clusters. In fact these classes share the common cluster (c) at a higher level of the cluster hierarchy. This corresponds well to the biological ground truth since the presence or absence of an uncleavable signal sequence is somewhat arbitrary [23].

Figure 7(c) shows the cluster diagram using one object per class as pre-labeled object (7 k-clusters for $k = 0.15$). Supplied with this supervision, HISSCLU provides interesting information on the rare classes. The class of outer membrane lipoprotein consisting of 5 instances (cluster (d)) shows the maximal difference to all other classes. The two instances of inner membrane proteins with cleavable signal sequence are quite similar to periplasm proteins, forming together cluster (a). Already with this minimal amount of supervision HISSCLU archieves to determine the correct class hierarchy, cf. clusters (c1), (c2) and (c).

6.1.2 *Yeast Data*

The Yeast data set is a 9 dimensional data set consisting of 1448 instances belonging to 10 classes. Similar to Ecoli, this data set on predicting protein localization sites is highly unbalanced (5 to 423 objects per class) but much more challenging for classification [23]. The OPTICS reachability plot of this data set shows no cluster structure at all (cf. Figure 8(a)). In the cluster diagram ($MinPts = 5$, $\rho = 5.0, \xi = 5.0$) already for 100 pre-labeled objects several distinct clusters (extracted for $k = 0.2$) can be observed (cf. Figure 8(b)). Cluster (a) consists predominately of objects from three classes: cytosolic, nuclear and mitochondiral proteins. Cluster (b) represents membrane proteins without N-terminal signal. Cluster (c) with two sub-clusters represents the classes of membrane proteins with uncleaved and cleaved signal. For 500 pre-labeled objects the cluster purity increases, as expected (cf. Figure 8(c)). The two sub-clusters of cluster (c) get more separated forming clusters (c) and (d). For cluster (a) two sub-clusters can now be observed: Cluster (a2) predominately consists of objects of class cytosolic proteins, whereas cluster (a1) is mainly shared by mitochandrial and nuclear proteins. This reflects a fundamental difficulty in identifying nuclear proteins which can also be observed for most classification methods [24] and



(a) COP-k-means $k = 2$. (b) MPC-k-means $k = 2$. (c) MPC-k-means $k = 5$. (d) HISSCLU.

Figure 11: Cluster Assignment.

has a biological reason. The nuclear localization signal is not limited to one portion of a proteins primary sequence. In some cases a protein without a nuclear localization signal may be transported to the nucleus. [16]. However cluster (a1) contains several class-pure sub-clusters of nuclear proteins extracted for $k = 0.05$.

6.1.3 Glass Data

The glass data comprises 9 numerical attributes representing different physical and chemical properties. 214 instances are labeled to 7 classes representing various types of glass. Two clusters have been extracted from the OPTICS reachability plot for $k = 0.2$, cf. Figure 10(a). Only the class "tableware" is well separated forming a distinct cluster (d), the other cluster (a) consists of objects of all other classes. In the cluster diagram (cf. Figure 9(a) the cluster hierarchy becomes obvious. Instances of the classes "building windows float processed" and "building windows non float processed" and "vehicle windows float processed" are very similar, forming together one cluster of objects of type window glass (a). This cluster contains four sub-clusters for $k = 0.2$: Clusters (a1) and (a2) mainly represent objects of the class "building window float processed" and (a4) "building window non float processed", whereas (a3) contains objects of all three classes. Objects of the class "container" forming cluster (b) are more similar to the objects of type window glass (cluster (a)) than to the other two clusters representing the classes "headlamps" (cluster (c)) and "tableware" (cluster (d)).

6.1.4 Metabolic Data

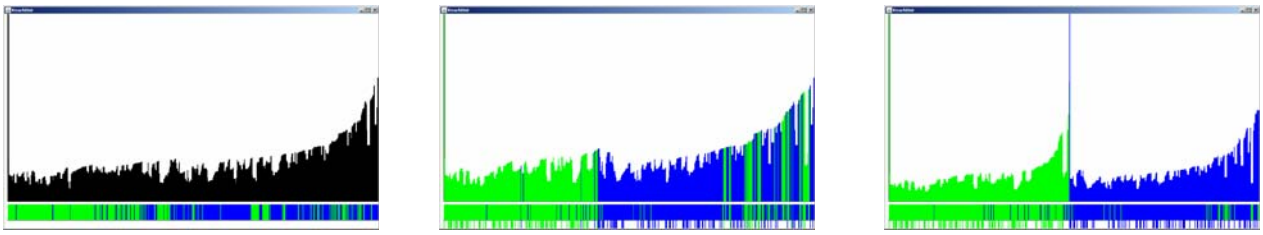
Figure 9(b) shows the cluster diagram of a metabolic data set for $MinPts = 3$, $\rho = 4$, $\xi = 0.5$. This 41-dimensional data set (132 instances) was produced by modern screening methodologies and represents cases of phenylalanine hydroxylase deficiency (PAHD) consisting of two different expressions of this metabolic disorder, the milder form called HPA, PKU, the stronger expression, and a control group [3]. We used 15 sample points as pre-labeled objects. For both diagrams, we extracted k-clusters for $k = 0.2$. Most of the instances of class HPA are in cluster (b) which is more similar to the class-pure cluster (c) representing the healthy control group than to cluster (a) comprising predominantly instances of class PKU. Both, PKU and even more HPA form different sub-clusters corresponding to different substages of the disease with blurred borders. For class HPA there are two distinct sub-clusters marked by (b1) and (b2). In the OPTICS reachability plot (for comparison depicted in Figure 10(b)) there is only one distinct cluster(c) representing the control group.

6.2 Spatial- and Class-Coherent Cluster Assignment.

Due to simultaneous cluster expansion and careful local distance weighting HISSCLU preserves the original cluster structure much better than comparative methods. Figure 11 shows the cluster assignment for COP-K-Means [26] and MPC-K-Means [4] for $k = 2$ (number of classes) and $k = 5$ (number of clusters extracted of the cluster diagram) on our running example with constraints generated between all pairs of pre-labeled objects. Not violating any of the constraints, COP-K-Means obtains an unnatural clustering result where even pre-labeled objects situated in the center of dense clusters are assigned to a different clusters (cf. Figure 11(a)). MPC-K-Means performs better (cf. Figure 11(b)) considering both, constraints and the data distribution when assigning objects to clusters. Due to this there are pre-labeled objects of both classes assigned to one cluster. For $k = 5$ COP-K-Means does not perform better (not depicted) and MPC-K-Means (cf. Figure 11(c)) splits up the cluster on top although there are must-linked objects inside. This reflects the inherent tendency of K-means to detect spatially compact clusters. HISSCLU achieves to assign objects to clusters in the best coherent way with class labels and local cluster structure (cf. Figure 11(d)).

6.3 Making Use of Supervision

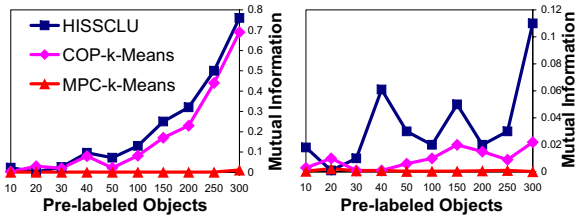
In cases when no natural spatial cluster boundaries exist the information provided by the pre-labeled objects should be used to improve the clustering. To examine how efficient and effective the algorithms make use of the supervision in this case we compared their performance w.r.t. Mutual Information [11] on the liver data set (351 instances, 7 attributes, 2 classes). We selected a two-class data set showing no cluster structure at all in the reachability plot (cf. Figure 10(c)). We used $MinPts = 5$ and generated cluster diagrams with $\rho = 20$ and $\xi = 10$. This strong distance weighting is applied in order to obtain k-clusterings with two clusters without noise (always done for $k = 0.9$) which are directly comparable to the partitionings into two clusters generated by MPC-K-Means and COP-K-Means (cf. Figure 9(c)). The task here is to achieve a clustering maximally respecting the class structure with as less supervision as possible. To provide to all algorithms the same amount of supervision, we randomly sampled objects out



(a) Reachability-plot. (b) $\rho = 5$. (c) $\rho = 50$.

Figure 13: Parameter Selection.

of the data set, which we directly used as pre-labeled objects for HISSCLU. For the other algorithms we generated all possible constraints between all pairs of these objects. MPC-K-Means has been parameterized as described in [4], for COP-K-Means no advanced parameter settings are possible. The Mutual Information reflects to which extend a clustering corresponds to the class labels. Figure 12 shows the Mutual Information on the whole data set and on the unlabeled data w.r.t. the number of pre-labeled objects. MPC-K-Means does not succeed in making use of the constraints due to global distance weighting. More constraints can even be misleading for global weighting in the situation of no natural spatial cluster borders. COP-K-Means performs quite good since the algorithm enforces the constraints without caring about the data distribution. HISSCLU performs even better in spite of not using any explicit constraints. This demonstrates the usability of our local weighting technique.



(a) MI on all data (b) MI on test data

Figure 12: Comparison of Class Purity w.r.t Various Amounts of Supervision on Liver Data.

6.4 Parameter selection

From hierarchical density based clustering we have inherited the parameters ϵ and $MinPts$, which can be set as suggested in [2]. In addition HISSCLU uses the parameter ρ and ξ to establish borders between classes when there are no clear natural cluster boundaries. In order to maximally preserve the original cluster structure it is recommended to start with $\rho = 1.0..1.5, \xi = 0.5$ and to increase it if better separation of classes is desired.

Figure 13 illustrates the impact of the parameter ρ on cluster separation. The reachability plot of 2-d synthetic data set (600 points, two classes which 300 points each) shows no distinct cluster structure (cf. Figure 13(a)). There are no obvious spatial cluster boundaries. We generated cluster diagrams using 200 randomly sampled pre-labeled objects for a fixed $\xi = 5.0$ and varying ρ. Already for $\rho = 5$ the cluster order corresponds well to the class labels. A separation into two large clusters, each containing mainly objects of one class becomes evident. Besides this, small class-pure sub-clusters of both classes can be observed. For $\rho = 50$ the separation is further improved. A distinct cluster boundary well separating the objects of both classes from each other with a mutual information between class labels and cluster-ids of 0.59 (on all data) is established. As illustrated in Figure 4, the parameter ξ additionally allows to adjust the sharpness of the cluster boundaries, which causes only minor changes in the cluster order in this example.

7. BENEFITS OF HISSCLU FOR DATA MINING

In this section, we demonstrate the main benefits of HISSCLU for data exploration and classification. In particular we focus on three aspects which can be best illustrated all together using a synthetic data set. Besides this, we demonstrate single aspects on real world data. HISSCLU may be applied as a preprocessing step for classification. To evaluate the classification accuracy we used a linear support vector machine with standard settings as implemented in the WEKA machine learning toolkit [1]. Let us consider a synthetic data set with 1300 objects belonging to three classes, among them 780 objects for which the class label is known. The classification accuracy with linear SVM using the pre-labeled objects as training data is 80.2%. Most errors occur due to mixing up the classes 1 and 2 and also for class 3 precision and recall can be improved. The cluster diagram of this data set using ρ and $\xi = 5.0$ is depicted in Figure 14(a). HISSCLU achieves to assign correct class labels to 95.4% of the test objects.

7.1 Spotting Outliers in the Training Set

For the performance of most classifiers it is beneficial to identify and remove outliers from the training set [19]. HISSCLU provides a simple but effective automatic way to remove outliers. In our example the cluster diagram shows three clusters $c1, c2, c3$ (extracted for $k = 0.2$) whereas $c1$ and $c2$ are class pure clusters consisting of objects of the classes 3 and 1, respectively. The cluster $c3$ is composed of objects of the classes 1 and 2. Some of the training objects do not belong to any of these clusters. It is also evident from the cluster diagram that most of these outliers belong to class 3. Removing the outliers from the training set results in a refined training set with 750 objects. A linear SVM trained with this training data set achieves a significantly higher accuracy of 85.4% on the test data.

7.2 Identifying Multimodal and Similar Classes

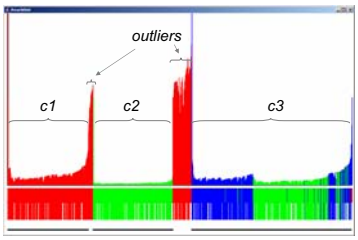
For classification also the information on the multi-modality of classes is very useful. In our example, class 1 is clearly 2-modal. Closely related to multi-modality is the similarity of classes. Some of the objects of class 1 are very similar to the objects of class 2, as they form together cluster $c3$. The other part of the objects of class 1 which can be found in cluster $c2$ are very dissimilar to the objects of class 2. To use this information for classification, we internally split up the training objects of class 1 into the classes 1.1, consisting of training objects of class 1 in cluster $c3$ and 1.2, containing the objects of class 1 in cluster $c2$. We now have four classes in total on which the classifier is trained. Test objects which have been assigned to the labels 1.1. or 1.2 by the classifier are in a postprocessing step assigned to class 1. By doing so we can increase the classification accuracy to 90.4%. Roughly speaking, multi-modality of classes can be interpreted as a too coarse scale in labeling.

The application of HISSCLU to identify similar classes on real world data has been discussed in Section 6. For the data sets on protein localization biological meaningful similarities among classes have been detected. Similarity of classes can be regarded as a too fine scale in labeling, as the classes are not well separable in the data space. On the ecoli data set for example, a classification accuracy of 84.2% is obtained with linear SVM and 10-fold cross-validation, whereas most of the classification errors occur due to mixing up two subtypes of inner membrane proteins. In the cluster diagram these are the most similar classes (cf. Section 6) which also corresponds well to the classification results reported in literature [23, 24]. Merging the two subtypes of inner membrane proteins into one common class increases the classification accuracy to 92.2%.

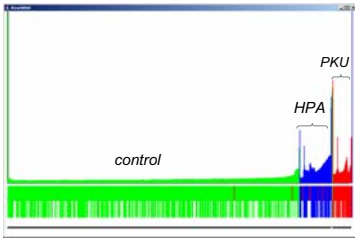
If a separation of similar classes is desired, additional information is required. Feature selection or feature transformation methods, e.g. suitable kernel functions can be used to modify the data space in a way that improves the separability. The cluster diagram can be used to determine for which classes further steps are necessary.

7.3 Using Class Hierarchies for Classification

Many classifiers, e.g. SVM have been originally designed for two-class-problems. In order to apply such binary classifiers in a multi-class setting, the multi-class problem has to be reduced to several binary ones. A common way to achieve this is to build one classifier for each pair of classes as implemented in WEKA. An alternative for adapting binary classifiers to multi-class-problems are Nested Dichotomies [15]. The classes are split into two subsets which are recursively further split. In [12] Dong et al. proposed ensembles of Nested Dichotomies for multi-class classification. For multi-class problems, an ensemble of balanced classification trees has shown superior performance in terms of classification accuracy and efficiency. In presence of a clear class hierarchy, it has been demonstrated in [28] that classification trees respecting this hierarchy perform better than arbitrarily constructed balanced trees. Building only hierarchies which respect the fact classes 1.1 and 2 are the most similar ones in our example data set, the classification accuracy with SVM can be slightly further increased to 91.5%. The improvement is rather small compared with ordinary SVM. This is due to the fact, that the data set shows no clear hierarchy besides the similarity of the classes 1.1 and 2. In presence of a distinct class hierarchy the benefits can be much larger.



(a) Synthetic Example.



(b) Metabolic Example.

Figure 14: Application of HISSCLU - Examples.

The authors propose in [28] to use domain knowledge to specify appropriate class hierarchies, but this is difficult in many applications. Consider a 14-dimensional metabolic data set with three classes representing a healthy control group (class 1), a group of patients with the metabolic disorder HPA (class 2) and a group of patients suffering from the more severe disorder PKU (class 3). It is a nontrivial question to rate how similar these classes actually are. The HISSCLU cluster diagram depicted in Figure 14(b) clearly shows that the instances of class HPA are more similar to the healthy control group than to the instances of class PKU. Giving this hierarchy as an input to the approach [28], a classification accuracy of 97.0% is obtained with linear SVM, compared to 93.5% when using an ordinary pair-wise trained SVM. All other possible hierarchies lead to worse results than the one extracted of the cluster diagram. There are many other approaches aiming at using the information of class hierarchies to improve the classification result, e.g. [7, 8, 10]. The cluster hierarchy can be automatically extracted of the cluster diagram by considering the reachability distances between class pure clusters. However, different approaches may require class hierarchies of different resolutions.

8. CONCLUSIONS

In this paper, we have proposed HISSCLU, a novel method for semi-supervised clustering. Our method founded on a hierarchical, density-based cluster notion with the advantage of a determinate clustering result, high robustness with respect to noise and no favor for clusters of a particular shape (e.g. convex). HISSCLU consists of a method for cluster-consistent assignment of class labels to previously unlabeled objects and a method for the determination of the overall cluster structure of the data set in a way which is consistent to original and obtained class labels. In contrast to most previous methods, HISSCLU avoids the use of constraints in order to preserve the original cluster structure.

In a broad experimental evaluation we demonstrated that HISSCLU has the following advantages over state of the art semi-supervised clustering methods:

- Making use of multimodal class information in the clustering process,
- Detecting and visualizing hierarchical class and cluster structures,
- Spacial and class coherent data partitioning.

The HISSCLU cluster diagram offers the user more insight into the class and cluster structure. It provides answers to important questions, e. g. which classes are most similar? They may have a common superclass. Or, are there any multimodal classes? Classes may be distributed over several clusters in the cluster diagram. Moreover, our method is robust in terms of parameter settings. With a runtime complexity of $O(n^2)$ and memory usage of $O(n)$ HISSCLU is scalable to be used on top of moderate to large size databases.

9. REFERENCES

[1] *"WEKA machine learning package, http://www.cs.waikato.ac.nz/ ml/weka"*. Universitity of Waikato.

[2] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. "OPTICS: Ordering Points to Identify the Clustering Structure". In *SIGMOD Conference*, 1999.

[3] C. Baumgartner, C. Böhm, D. Baumgartner, G. Marini, K. Weinberger, B. Olgemöller, B. Liebl, and A. Roscher. "Supervised machine learning techniques for the classification of metabolic disorders in newborns". In *Bioinformatics*, pages 20(17):2985–2996, 2004.

[4] M. Bilenko, S. Basu, and R. J. Mooney. "A Probabilistic Framework for Semi-Supervised Clustering". In *KDD Conference*, 2004.

[5] M. Bilenko, S. Basu, and R. J. Mooney. "Integrating Constraints and Metric Learning in Semi-Supervised Clustering". In *ICML Conference*, 2004.

[6] C. L. Blake and C. J. Merz. *"UCI Repository of machine learning databases, http://www.ics.uci.edu/∼mlearn/MLRepository.html"*.

[7] N. Cesa-Bianchi, C. Gentile, A. Tironi, and L. Zaniboni. Incremental algorithms for hierarchical classification. In *NIPS*, 2004.

[8] N. Cesa-Bianchi, C. Gentile, and L. Zaniboni. Hierarchical classification: combining bayes with svm. In *ICML*, pages 177–184, 2006.

[9] B.-R. Dai, C.-R. Lin, and M.-S. Chen. On the techniques for data clustering with numerical constraints. In *SDM Conference*, 2003.

[10] O. Dekel, J. Keshet, and Y. Singer. Large margin hierarchical classification. In *ICML*, 2004.

[11] B. E. Dom. " An Information-Theoretic External Cluster-Validity Measure". In *Research Report RJ 10219, IBM*, 2001.

[12] L. Dong, E. Frank, and S. Kramer. Ensembles of balanced nested dichotomies for multi-class problems. In *Proc. of PKDD Conference*, pages 84–95, 2005.

[13] C. Eick, N. Zeidat, and Z. Zhao. " Supervised Clustering - Algorithms and Benefits". In *Proc. of the International Conference on Tools with AI*, 2004.

[14] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In *KDD Conference*, 1996.

[15] J. Fürnkranz. "Round Robin Classification". *Journal of Machine Learning Research*, (2):721–747, 2002.

[16] J. Gracia-Bustos, J. Heitman, and M. Hall. "Nuclear protein localization". In *Biochimica et Biophysica Acta*, pages 1071:83–101, 1991.

[17] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, 1988.

[18] D. Klein, D. Kamvar, and C. Manning. "From Instance-Level Constraints to Space-Level Constraints: Making Most of Prior Knowledge in Data Clustering.". In *ICML Conference*, 2002.

[19] H. Li and M. Niranjan. "Outlier Detection in Benchmark Classification Tasks". In *Proc. of International Conference on Accoustics, Speech ans Signal Processing*, pages 557–560, 2006.

[20] H. Liu and S. teng Huang. Evolutionary semi-supervised fuzzy clustering. *Pattern Recognition Letters*, 24(16):3105–3113, 2003.

[21] Z. Lu and T. Leen. "Semi-supervised Learning with Penalized Probabilistic Clustering". In *NIPS 17*, pages 849–856, 2005.

[22] J. MacQueen. "Some Methods for Classification and Analysis of Multivariate Observations". In *5th Berkeley Symp. Math. Statist. Prob.*, 1967.

[23] K. Nakai and M. Kanehisa. "A Knowledge Base for Predicting Protein Localization Sites in Eukaryotic Cells". *Genomics*, 14(897):897–911, 1991.

[24] C. Plant, C. Böhm, C. Baumgartner, and B. Tilg. "Enhancing instance-based classification with local density.". In *Bioinformatics*, pages 22:981–988, 2006.

[25] J. Sander, X. Qin, Z. Lu, N. Niu, and A. Kovarsky. "Automatic Extraction of Clusters from Hierarchical Clustering Representations.". In *PAKDD*, 2003.

[26] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedel. "Constrained K-Means Clustering with Background Knowledge". In *ICML Conference*, 2001.

[27] X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. technical report., 2002.

[28] A. Zimek. Hierarchical classification using ensembles of nested dichotomies. Master's thesis, TU/LMU Munich, 2005.