

Approximating Hierarchical MV-sets for Hierarchical Clustering

Authored by:

Shaul Markovitch
Assaf Glazer
Michael Lindenbaum
Omer Weissbrod

Abstract

The goal of hierarchical clustering is to construct a cluster tree, which can be viewed as the modal structure of a density. For this purpose, we use a convex optimization program that can efficiently estimate a family of hierarchical dense sets in high-dimensional distributions. We further extend existing graph-based methods to approximate the cluster tree of a distribution. By avoiding direct density estimation, our method is able to handle high-dimensional data more efficiently than existing density-based approaches. We present empirical results that demonstrate the superiority of our method over existing ones.

1 Paper Body

Data clustering is a classic unsupervised learning technique, whose goal is dividing input data into disjoint sets. Standard clustering methods attempt to divide input data into discrete partitions. In Hierarchical clustering, the goal is to find nested partitions of the data. The nested partitions reveal the modal structure of the data density, where clusters are associated with dense regions, separated by relatively sparse ones [27, 13]. Under the nonparametric assumption that the data is sampled i.i.d. from a continuous distribution F with Lebesgue density f in \mathbb{R}^d , Hartigan observed that f has a hierarchical structure, called its cluster tree. Denote $L_f(c) = \{x : f(x) \geq c\}$ as the level set of f at level c . Then, the connected components in $L_f(c)$ are the high-density clusters at level c , and the collection of all high-density clusters for $c \in [0, \infty)$ has a hierarchical structure, where for any two clusters A and B , either $A \subset B$, $B \subset A$, or $A \cap B = \emptyset$.

$L_f(c) \supset L_f(c') \iff c \geq c'$

Figure 1: A univariate, tri-modal density function and its corresponding cluster tree are illustrated. Figure 1 shows a plot of a univariate, tri-modal density function. The cluster tree of the density $F \approx 0.67$ function is shown on

top of the density function. The high-density clusters are nodes in the cluster tree. Leaves are associated with modes in the density function. 66.7%

66.7%

33.3%

33.3%

1

$F \geq 0.5$

Given the density f , the cluster tree can be constructed in a straightforward manner via a recursive algorithm [23]. We start by setting the root node with a single cluster containing the entire space, corresponding to $c = 0$. We then recursively increase c until the number of connected components increases, at which point we define a new level of the tree. The process is repeated as long as the number of connected components increases. In Figure 1, for example, the root node has two daughter nodes, which were found at level $c = 0.11$. The next two descendants of the left node were found at level $c = 0.23$. A common approach for hierarchical clustering is to first use a density estimation method to obtain f [18, 5, 23], and then estimate the cluster tree using the recursive method described above. However, one major drawback in this approach is that a reliable density estimation is hard to obtain, especially in high-dimensional data. An alternative approach is to estimate the level sets directly, without a separate density estimation step. To do so, we define the minimum volume set (MV-set) at level c as the subset of the input space with the smallest volume and probability mass of at least c . MV-sets of a distribution, which are also level sets of the density f (under sufficient regularity conditions), are hierarchical by definition. The well-known One-Class SVM (OCSVM) [20] can efficiently find the MV-set at a specified level c . A naive approach for finding a hierarchy of MV-sets is to train distinct OCSVMs, one for each MV-set, and enforce hierarchy by intersection operations on the output. However, this solution is not well suited for finding a set of hierarchical MV-sets, because the natural hierarchy of MV-sets is not exploited, leading to a suboptimal solution. In this study we propose a novel method for constructing cluster trees by directly estimating MV-sets, while guaranteeing convergence to a globally optimum solution. Our method utilizes the q -OneClass SVM (q -OCSVM) method [11], which can be regarded as a natural extension of the OCSVM, to jointly find the MV-sets at a set of levels $\{c_i\}$. By avoiding direct density estimation, our method is able to handle high-dimensional data more efficiently than existing density-based approaches. By jointly considering the entire spectrum of desired levels, a globally optimum solution can be found. We combine this approach with a graph-based heuristic, found to be successful in high-dimensional data [2, 23], for finding high density clusters in the approximated MV-sets. Briefly, we construct a fully connected graph whose nodes correspond to feature vectors, and remove edges between nodes connected by low-density regions. The connected components in the resulting graph correspond to high density clusters. The advantage of our method is demonstrated empirically on synthetic and real data, including a reconstruction of an evolutionary tree of human populations using the high-dimensional 1000 genomes dataset.

Our novel method for hierarchical clustering belongs to a family of non-parametric clustering methods. Unlike parametric methods, which assume that each group i is associated with a density f_i belonging to some family of parametric densities, non-parametric methods assume that each group is associated with modes of a density f [27]. Non-parametric methods aim to reveal the modal structure of f [13, 28, 14]. Hierarchical clustering methods can be divided into agglomerative (bottom up) and divisive (top down) methods. Agglomerative methods (e.g. single-linkage) start with n singleton clusters, one for each training feature vector, and work by iteratively linking two closest clusters. Divisive methods, on the other hand, start with all feature vectors in a single cluster and recursively divide clusters into smaller sub-clusters. While single-linkage was found, in theory, to have better stability and convergence properties in comparison to average-linkage and complete-linkage [4], it is frequently criticized by practitioners due to the chaining effect. Single-linkage ignores the density of feature vectors in clusters, and thus may erroneously connect two modes (clusters) with a few feature vectors connecting them, that is, a ?chain? of feature vectors. Wishart [27] suggested overcoming this effect by conducting a one-level analysis of the data. The idea is to estimate a specific level set of the data density ($L_f(c)$), and to remove noisy features 2

outside this level that could otherwise lead to the chaining effect. The connected components left in $L_f(c)$ are the clusters; expansions of this idea can be found in [9, 26, 6, 3]. Indeed, this analysis is more resistant to the chaining effect. However, one of its major drawbacks is that no single level set can reveal all the modes of the density. Therefore, various studies have proposed estimating the entire hierarchical structure of the data (the cluster tree) using density estimates [13, 1, 22, 18, 5, 23, 17, 19]. These methods are considered as divisive hierarchical clustering methods, as they start by associating all feature vectors to the root node, which is then recursively divided to sub-clusters by incrementally exploring level sets of denser regions. Our proposed method belongs to this group of divisive methods. Stuetzle [22] used the nearest neighbor density estimate to construct the cluster tree and pointed out its connection to single-linkage clustering. Kernel density estimates were used in other studies [23, 19]. The bisecting K-means (BiKMean) method is another divisive method that was found to work effectively in cluster analysis [16], although it provides no theoretical guarantee for finding the correct cluster tree of the underlying density. Hierarchical clustering methods can be used as an exploration tool for data understanding [16]. The nonparametric assumption, by which density modes correspond to homogenous feature vectors with respect to their class labels, can be used to infer the hierarchical class structure of the data [15]. An implicit assumption is that the closer two feature vectors are, the less likely they will be to have different class labels. Interestingly, this assumption, which does not necessarily hold for all distributions, is being discussed lately in the context of hierarchical sampling methods for active learning [8, 7, 25], where the correctness of such a hierarchical modeling approach is said to depend on

the Probabilistic Lipschitzness assumption about the data distribution.

3

Approximating MV-sets for Hierarchical Clustering

Our proposed method consists of (a) estimating MV-sets using the q-OCSVM method; (b) using a graph-based method for finding a hierarchy of high density regions in the MV-sets, and (c) constructing a cluster tree using these regions. These stages are described in detail below.

Estimating MV-Sets

We begin by briefly describing the One-Class SVM (OCSVM) method. Let $X = \{x_1, \dots, x_n\}$ be a set of feature vectors sampled i.i.d. with respect to F . The function f_C returned by the OCSVM algorithm is specified by the solution of this quadratic program: $\min_{w, \xi} \frac{1}{2} \|w\|^2 + \sum_{i=1}^n \xi_i$, s.t. $(w \cdot x_i) - \xi_i \leq 0$, where w is a vector of the slack variables. Recall that all training examples x_i for which $(w \cdot x_i) - \xi_i \leq 0$ are called support vectors (SVs). Outliers are referred to as examples that strictly satisfy $(w \cdot x_i) - \xi_i > 0$. By solving the program for $\tau = 1 - \epsilon$, we can use the OCSVM to approximate the MV-set $C(\tau)$. Let $0 = \tau_1 < \tau_2 < \dots < \tau_q < 1$ be a sequence of q quantiles. The q-OCSVM method generalizes the OCSVM algorithm for approximating a set of MV-sets $\{C_1, \dots, C_q\}$ such that a hierarchy constraint $C_i \supset C_j$ is satisfied for $i < j$. Given X , the q-OCSVM algorithm solves this primal program: $\min_{w, \xi_j} \frac{1}{2} \|w\|^2 + \sum_{j=1}^q \sum_{i=1}^n \xi_{j,i}$

$$\begin{aligned} & \min_{w, \xi_j} \frac{1}{2} \|w\|^2 + \sum_{j=1}^q \sum_{i=1}^n \xi_{j,i} \\ & \text{s.t. } (w \cdot x_i) - \xi_{j,i} \leq 0, \quad j = 1, \dots, q, \quad i = 1, \dots, n \end{aligned} \quad (2)$$

This program generalizes Equation (1) to the case of finding multiple, parallel half-space decision functions by searching for a global minimum over their sum of objective functions: the coupling between q half-spaces is done by summing q OCSVM programs, while forcing these programs to share the same w . As a result, the q half-spaces in the solution of Equation (2) differ only by their bias terms, and are thus parallel to each other. This program is convex, and thus a global minimum can be found in polynomial time.

Glazer et al. [11] proves that the q-OCSVM algorithm can be used to approximate the MV-sets of a distribution.

Generalizing q-OCSVM for Finding an Infinite Number of Approximated MV-sets

The q-OCSVM finds a finite number of q approximated MV-sets, which capture the overall structure of the cluster tree. However, in order to better resolve differences in density levels between data points, we would like the solution to be extended for defining an infinite number of hierarchical sets. Our approach for doing so relies on the parallelism property of the approximated MV-sets in the q-OCSVM solution. An infinite number of approximated MV-sets are associated with separating hyperplanes in F that are parallel to the q hyperplanes in the q-OCSVM solution. Note that every projected feature vector x lies on a unique separating hyperplane that is parallel to the q hyperplanes defined by the solution, and the distance $\text{dis}(x) = (w \cdot x) - \xi$ is sufficient to determine

whether x is located inside each of the approximated MV-sets. We would like to know the probability mass associated with each of the infinite hyperplanes. For this purpose, we could similarly estimate the expected probability mass of the approximated MVset defined for any $x \in \mathbb{R}^d$. When $\varphi(x)$ lies strictly on one of the $i \in [q]$ hyperplanes, then x is considered as lying on the boundary of the set approximating $C(\varphi_i)$. When $\varphi(x)$ does not satisfy this condition, we use a linear interpolation to define φ for its corresponding approximated MV-set: Let φ_i, φ_{i+1} be the bias terms associated with the i and $i + 1$ approximated MV-sets that satisfy $\varphi_i \leq (w \cdot \varphi(x)) \leq \varphi_{i+1}$. Then we linearly interpolate $(w \cdot \varphi(x))$ along the $[\varphi_{i+1}, \varphi_i]$ interval for an intermediate $\varphi \in (\varphi_i, \varphi_{i+1})$. For the completion of the definition, we set $\varphi_0 = \max_{x \in X} (w \cdot \varphi(x))$ and $\varphi_{q+1} = \min_{x \in X} (w \cdot \varphi(x))$. 3.2

Finding a Hierarchy of High-Density Regions

To find a hierarchy of high density regions, we adopt a graph-based approach. We construct a fully-connected graph whose nodes correspond to feature vectors, and remove edges between nodes separated by low-density regions. The connected components in the resulting graph correspond to high density regions. The method proceeds as follows. Let $\varphi(x)$ be the expected probability mass of the approximated MV-set defined by x . Let $\varphi_{i,s}$ be the maximal value of $\varphi(x)$ over the line segment connecting the feature vectors x_i and x_s in X : $\varphi_{i,s} = \max_{t \in [0,1]} \varphi(tx_i + (1-t)x_s)$. (3)

Let G be a complete graph between pairs of feature vectors in X with edges equal to $\varphi_{i,s} - 1$. High density clusters at level φ are defined as the connected components in the graph $G(\varphi)$ induced by removing edges from G with $\varphi_{i,s} \leq \varphi$. This method guarantees that two feature vectors in the same cluster of the approximated MV-set at level φ would surely lie in the same connected component in $G(\varphi)$. However, the opposite would not necessarily hold φ when $\varphi_{i,s} \leq \varphi$ and a curve connecting x_i and x_s exists in the cluster, x_i and x_s might erroneously be found in different connected components. Nevertheless, it was empirically shown that erroneous splits of clusters are rare if the density function is smooth [23]. One way to implement this method for finding high density clusters is to iteratively find connected components in $G(\varphi)$, when at each iteration φ is incrementally increased (starting from $\varphi = 0$), until all the clusters are found. However, [23] observed that we can simplify this method by working only on the graph G and its minimal spanning tree T . Consequently, we can compute a hierarchy of high-density regions in two steps: First, construct G and its minimal spanning tree T . Then, remove edges from T in descending order of their weights such that the connected components left after removing an edge with weight φ correspond to a high density cluster at level φ . Connected components with a single feature vector are treated as outliers and removed. 1 We calculated $\varphi_{i,s}$ in G by checking the $\varphi(x)$ values for 20 points sampled from the line segment between x_i and x_s . The same approach was also used by [2] and [23].

4

3.3

Constructing a Cluster Tree

The hierarchy resulting from the procedure described above does not form a full partition of the data, as in each edge removal step a fraction of the data is left outside the newly formed high density clusters. To construct a full partition, feature vectors left outside at each step are assigned to their nearest cluster. Additionally, when a cluster is split into sub-clusters, all its assigned feature vectors are assigned to one of the new sub-clusters. The choice of kernel width has a strong effect on the resulting cluster tree. On the one hand, a large bandwidth may lead to the inner products induced by the kernel function being constant; that is, many examples in the train data are projected to the same point in F . Hence, the approximated MV-sets could eventually be equal, resulting in a cluster tree with a single node. On the other hand, a small bandwidth may lead to the inner products becoming closer to zero; that is, points in F tend to lie on orthogonal axes, resulting in a cluster tree with many branches and leaves. We believe that the best approach for choosing the correct bandwidth is based on the number of modes that we expect to find for the density function. By using a grid search over possible γ values, we can choose the bandwidth that results in a cluster tree in which the expected number of modes is the same as the number we expect.

4

Empirical Analysis

We evaluate our hierarchical clustering method on synthetic and real data. While the quality of an estimated cluster tree for the synthetic data can be evaluated by comparing the resulting tree with the true modal structure of the density, alternative quality measures are required to estimate the efficiency of hierarchical clustering methods on high-dimensional data when the density is unknown. In the following section we introduce our proposed measure.

The Quality Measure

One prominent measure is the F -measure, which was extended by [16] to evaluate the quality of estimated cluster trees. Recall that classes refer to the true (unobserved) class assignment of the observed vectors, whereas clusters refer to their tree-assigned partition. For a cluster j and class i , define $n_{i,j}$ as the number of feature vectors of class i in cluster j , and n_i, n_j as the number of feature vectors associated with class i and with cluster j , respectively. The F -measure for cluster j and class i is given by $F_{i,j} = \frac{2 \cdot \text{Recall}_{i,j} \cdot \text{Precision}_{i,j}}{\text{Recall}_{i,j} + \text{Precision}_{i,j}}$, where $\text{Recall}_{i,j} = \frac{n_{i,j}}{n_i}$ and $\text{Precision}_{i,j} = \frac{n_{i,j}}{n_j}$. The F -measure for the cluster tree is $F = \max_{i,j} F_{i,j}$. (4) The F -measure was found to be a useful tool for the evaluation of hierarchical clustering methods [21], as it quantifies how well we could extract k clusters, one for each class, that are relatively "pure" and large enough with respect to their associated class. However, we found it difficult to use this measure directly in our analysis, because it appears to prefer overfitted trees, with a large number of spurious clusters. We suggest correcting this bias via cross-validation. We split the data X into two equal-sized train and test sets, and construct a tree using the train set. Test examples are recursively assigned to clusters in the tree in a top-down manner, and the F -measure is calculated according to the resulting tree. When analytical boundaries of clusters in the tree are not available (such

as in our method), we recursively assign each test example in a cluster to the sub-cluster containing its nearest neighbor in the train set, using Euclidean distance. 4.2

Reference Methods

We compare our method with methods for density estimation, that can also be used to construct a graph G . For this purpose, since $f(x)$ is used instead of $\hat{f}(x)$, we had to adjust the way we construct G .

A kernel density estimator (KDE) and nearest neighbor density estimator (NNE), similar to the one used by [23], are used as competing methods. In addition, we compare our method with the bisecting K-means (BiKMean) method [21] for hierarchical clustering. 4.3

Experiments with Synthetic Data

We run our hierarchical clustering method on data sampled from a synthetic, two-dimensional, trimodal distribution. This distribution is defined by a 3-Gaussian mixture distribution. 20 i.i.d. points were sampled for training our q-OCSVM method, with $\mu_1 = 0.25$, $\mu_2 = 0.5$, $\mu_3 = 0.75$ (3-quantiles), and with a bandwidth h , which results in a cluster tree with 3 modes. The left side of Figure 2 shows the data sampled, and the 3 approximated hierarchical MV-sets. The resulting $Q=3, N=20, h=15$ 3-modes cluster tree is shown in the right side of Figure 2.

Leaf 1: {1}
 1.5 1
 Branch 4: {1 2}, $P=0.68$
 0.5 0
 Leaf 2: {2}
 0.5
 Branch 5: {1 2 3}, $P=0.85$
 0.5 1.5 2
 Leaf 3: {3} 0.5
 2.5 2.5
 2
 1.5
 1
 0.5
 0
 0.5
 1
 1.5
 2
 2.5

Figure 2: Left: Data sampled for training our q-OCSVM method and the 3 approximated MV-sets; Right: The cluster tree estimated from the synthetic

data. The most frequent label in each mode, denoted in curly brackets next to each leaf, defines the label of the mode. Branches are labeled with the probability mass associated with their level set. . We used our proposed and reference method on the data to obtain cluster trees with different numbers of modes (leaves). The number of modes can be tweaked by changing the value of γ for the qOCSVM and KDE methods, and by pruning nodes of small size for the NNE and BiKMean methods. 20 test examples were i.i.d. sampled from the same distribution to estimate the resulting F -measures. The left side of Figure 3 shows the F -measure for each method in terms of changes in the number of modes in the resulting tree. For all methods, the F -measure is bounded by 0.8 as long as the number of modes is greater than 3, correctly suggesting the presence of 3 modes for the data. 4.4

The olive oil dataset

The olive oil dataset [10] consists of 572 olive oil examples, with 8 features each, from 3 regions in Italy (R1, R2, R3), each one further divided into 3 sub-areas. The right side of Figure 3 shows the F -measure for each method in terms of changes in the number of modes in the tree. The q-OCSVM method dominates the other three methods when the number of modes is higher than 5, with an average $F = 0.62$, while its best competitor (KDE) has an average $F = 0.55$. It can be seen that the variability of the F -measure plots is higher for the q-OCSVM and KDE methods than for the BiKMeans and NNE methods. This is a consequence of the fact that the structure of unpruned nodes remains the same for the BiKMeans and NNE methods, whereas different γ values may lead to different tree structures for the q-OCSVM and KDE methods. The cluster trees estimated using the q-OCSVM and KDE methods are shown in Figure 4. For each method, we chose to show the cluster tree with the smallest number of modes with leaves corresponding to all 8 labels. The q-OCSVM method groups leaves associated with the 8 areas into 3 clusters, which perfectly corresponds to the hierarchical structure of the labels. In contrast, modes estimated using the KDE method cannot be grouped into 3 homogeneous clusters. 2 When a density estimator f is used, $\pi_{i,s} = \min_{t \in [0,1]} p(t f(x_i) + (1-t)f(x_s))$ are set to be the edge weights, $G(c)$ is induced by removing edges from G with $\pi_{i,s} \leq c$, and T is defined as the maximal spanning tree of G (instead of the minimal).

6

CC vs. F

CC vs. F

0.85

0.65

0.8

0.6

0.75

0.55 F?Measure

F?Measure

0.9

0.7 0.65

0.45 q-OCSVM Cluster Tree

0.6
0.4 qOCSVM KDE BiKMeans NNE
0.55 0.5
0.5
1
2
3 Number of modes
4
qOCSVM KDE BiKMeans NNE
0.35 0.3
5
0
2
4
6
8 10 12 Number of modes
14
16
18
R1
20 R2

Figure 3: Left: The F -measures of each method are plotted in terms of the number of modes in the estimated cluster trees. The F -measures are calculated using the synthetic test data; Right:R3F measure for the olive oil dataset, calculated using 286 test examples, is shown in terms of the number of modes in the cluster tree. KDE Cluster Tree

q-OCSVM Cluster Tree
R1 R1
R2 R2 R3 R1 R3
R3 R1 R1
KDE Cluster Tree

Figure 4: Left: Cluster tree for the olive oil data estimated with q-OCSVM; Right: Cluster tree for R1 the olive oil data estimated with KDE. One prominent advantage of our method is that we can use the estimated probability mass of branches in the tree to better understand theR2 modal structure of the data. For instance, we can learn from Figure 4 that the R2 cluster is found in a relatively sparse MV-set at level 0.89, while R3 its two nodes are found in a much denser MV-set at level 0.12. Probability masses for high density clusters can also be estimated using the KDE method, but unlike our method, theoretical guarantees R1 R3 are not provided. R1 R1

The 1000 genomes dataset
4.5

We have also evaluated our method on the 1000 genomes dataset [24]. Hierarchical clustering approaches naturally arise in genetic population studies, as they can reconstruct trees that describe evolutionary history and are often the

first step in evolutionary studies [12]. The reconstruction of population structure is also crucial for genetic mapping studies, which search for genetic factors underlying genetic diseases. In this experiment we evaluated our method's capability to reconstruct the evolutionary history of populations represented in the 1000 genomes dataset, which consists of whole genome sequences of 1,092 human individuals from 14 distinct populations. We used a trinary representation wherein each individual is represented as a vector of features corresponding to 0, 1 or 2. Every feature represents a known genetic variation (with respect to the standard human reference genome 3), where the number indicates the number of varied genome copies. We used data processed by the 1000 Genomes Consortium, which initially contained 2.25 million variations. To reduce dimensionality, we used the 1,000 features that had the highest information gain with respect to the populations. We excluded from the analysis highly genetically admixed populations (Colombian, Mexican and Puerto Rican ancestry), because the evolutionary history of admixed populations cannot be represented by a tree. After exclusion, 911 individuals remained in the analysis.

<http://genomereference.org>

7

q-OCSVM Cluster Tree

CC vs. F 0.5

0.45

European

0.4

R1

0.35 F?Measure

0.3 0.25 0.2

R2 East Asian

0.15 0.1

0

KDE Cluster Tree

5

10

15 Number of modes

20

R3

African

qOCSVM BiKMeans SL

0.05

25

Figure 5: Left: F -measure for the 1000 genomes dataset, calculated using 455 test examples; Right: Cluster tree for the 1000 genomes data estimated with q-OCSVM. The labels are GBR (British in England and Scotland), TSI (Toscani in Italia), CEU (Utah Residents with Northern and Western R1 European ancestry), FIN (Finnish in Finland), CHB (Han Chinese in Beijing, China), CHS (Southern Han Chinese), ASW (Americans of African Ancestry in SW USA), YRI (Yoruba in Ibadan, Nigera), and LWK (Luhya in Webuye, Kenya). The

left side of Figure 5 shows that q-OCSVM dominates the other methods for every number of modes tested, demonstrating its superiority in high dimensional settings. Namely, it achieves an F-measure of 0.4 for 2 modes, whereas competing methods obtain an F-measure of 0.35. KDE was not evaluated as it is not applicable due to the high data dimensionality. To obtain a meaningful tree, we increased the number of modes until leaves corresponding to all three major human population groups (African, East Asian and European) represented in the dataset appeared. The tree obtained by using 28 modes is shown in the right side of Figure 5, indicating that q-OCSVM clustering successfully distinguishes between these three population groups. Additionally, it corresponds with the well-established theory that a divergence of a single ancestral population into African and Eurasian populations took place in the distant past, and that Eurasians diverged into East Asian and European populations at a later time [12]. The larger number of leaves representing European populations may result from the larger number of European individuals and populations in the 1000 genomes dataset.

5

Discussion

In this research we use the q-OCSVM method as a plug-in method for hierarchical clustering in high dimensional distributions. The q-OCSVM method estimates the level sets (MV-sets) directly without a density estimation step. Therefore, we expect to achieve more accurate results than approaches based on density estimation. Furthermore, since we know ρ for each approximated MV-set, we believe our solution would be more interpretable and informative than a solution provided by a density estimation-based method.

2 References

- [1] Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: ordering points to identify the clustering structure. *ACM SIGMOD Record*, 28(2):49–60, 1999.
- [2] Asa Ben-Hur, David Horn, Hava T Siegelmann, and Vladimir Vapnik. Support vector clustering. *The Journal of Machine Learning Research*, 2:125–137, 2002.
- [3] Gérard Biau, Benoît Cadre, and Bruno Pelletier. A graph-based estimator of the number of clusters. *ESAIM: Probability and Statistics*, 11(1):272–280, 2007.
- [4] Gunnar Carlsson and Facundo Mémoli. Characterization, stability and convergence of hierarchical clustering methods. *The Journal of Machine Learning Research*, 99:1425–1470, 2010.
- [5] Gunnar Carlsson and Facundo Mémoli. Multiparameter hierarchical clustering methods. In *Classification as a Tool for Research*, pages 63–70. Springer, 2010.
- [6] Antonio Cuevas, Manuel Febrero, and Ricardo Fraiman. Cluster analysis: a further approach based on density estimation. *Computational Statistics & Data Analysis*, 36(4):441–459, 2001.
- [7] Sanjoy Dasgupta. Two faces of active learning. *Theoretical Computer Science*, 412(19):1767–1781, 2011.
- [8] Sanjoy Dasgupta and Daniel Hsu. Hierarchical sampling for active learning. In *ICML*, pages 208–215. ACM, 2008.
- [9]

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In KDD, volume 96, pages 226?231, 1996. [10] M Forina, C Armanino, S Lanteri, and E Tiscornia. Classification of olive oils from their fatty acid composition. Food Research and Data Analysis, pages 189?214, 1983. [11] Assaf Glazer, Michael Lindenbaum, and Shaul Markovitch. q-ocsvm: A q-quantile estimator for high-dimensional distributions. In Advances in Neural Information Processing Systems, pages 503?511, 2013. [12] I. Gronau, M. J. Hubisz, et al. Bayesian inference of ancient human demography from individual genome sequences. Nature Genetics, 43(10):1031?1034, Oct 2011. [13] John A Hartigan. Clustering Algorithms. John Wiley & Sons, Inc., New York, 1975. [14] Anil K Jain. Data clustering: 50 years beyond k-means. Pattern Recognition Letters, 31(8): 651?666, 2010. [15] Daphne Koller and Mehran Sahami. Hierarchically classifying documents using very few words. In ICML, pages 170?178. Morgan Kaufmann Publishers Inc., 1997. [16] Bjornar Larsen and Chinatsu Aone. Fast and effective text mining using linear-time document clustering. In SIGKDD, ACM, pages 16?22, 1999. [17] Alvaro Martinez-Perez. A density-sensitive hierarchical clustering method. arXiv preprint arXiv:1210.6292, 2012. [18] Philippe Rigollet and Régis Vert. Optimal rates for plug-in estimators of density level sets. Bernoulli, 15(4):1154?1178, 2009. [19] Alessandro Rinaldo, Aarti Singh, Rebecca Nugent, and Larry Wasserman. Stability of density-based clustering. Journal of Machine Learning Research, 13:905?948, 2012. [20] Bernhard Schölkopf, John C. Platt, John C. Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. Estimating the support of a high-dimensional distribution. Neural Computation, 13(7):1443?1471, 2001. [21] Michael Steinbach, George Karypis, and Vipin Kumar. A comparison of document clustering techniques. In KDD Workshop on Text Mining, 2000. [22] Werner Stuetzle. Estimating the cluster tree of a density by analyzing the minimal spanning tree of a sample. Journal of Classification, 20(1):025?047, 2003. [23] Werner Stuetzle and Rebecca Nugent. A generalized single linkage method for estimating the cluster tree of a density. Journal of Computational and Graphical Statistics, 19(2), 2010. [24] The 1000 Genomes Project Consortium. An integrated map of genetic variation from 1,092 human genomes. Nature, 491:1, 2012. [25] Ruth Urner, Sharon Wulff, and Shai Ben-David. Plal: Cluster-based active learning. In COLT, pages 1?22, 2013. [26] G. Walther. Granulometric smoothing. The Annals of Statistics, pages 2273?2299, 1997. [27] David Wishart. Mode analysis: A generalization of nearest neighbor which reduces chaining effects. Numerical Taxonomy, 76:282?311, 1969. [28] Rui Xu, Donald Wunsch, et al. Survey of clustering algorithms. IEEE Transactions on Neural Networks, 16(3):645?678, 2005. 9