ITCH: Information-Theoretic Cluster Hierarchies

Christian Böhm[1], Frank Fiedler[1], Annahita Oswald[1], Claudia Plant[2], Bianca Wackersreuther[1], and Peter Wackersreuther[1]

[1] University of Munich, Munich, Germany
{boehm,fiedler,oswald,wackersb,wackersr}@dbs.ifi.lmu.de
[2] Florida State University, Tallahassee, FL, USA
cplant@fsu.edu

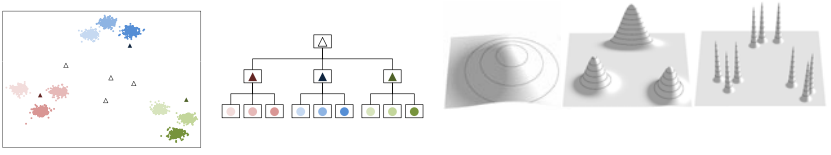
Abstract. Hierarchical clustering methods are widely used in various scientific domains such as molecular biology, medicine, economy, etc. Despite the maturity of the research field of hierarchical clustering, we have identified the following four goals which are not yet fully satisfied by previous methods: First, to guide the hierarchical clustering algorithm to identify only meaningful and valid clusters. Second, to represent each cluster in the hierarchy by an intuitive description with e.g. a probability density function. Third, to consistently handle outliers. And finally, to avoid difficult parameter settings. With ITCH, we propose a novel clustering method that is built on a hierarchical variant of the information-theoretic principle of Minimum Description Length (MDL), referred to as hMDL. Interpreting the hierarchical cluster structure as a statistical model of the data set, it can be used for effective data compression by Huffman coding. Thus, the achievable compression rate induces a natural objective function for clustering, which automatically satisfies all four above mentioned goals.

1 Introduction

Since dendrograms and similar hierarchical representations provide extremely useful insights into the structure of a data set, hierarchical clustering has become very popular in various scientific disciplines, such as molecular biology, medicine, or economy. However, well-known hierarchical clustering algorithms often either fail to detect the true clusters that are present in a data set, or they identify invalid clusters, which are not existing in the data set. These problems are particularly dominant in the presence of noise and outliers and result in the questions "How can we decide if a given representation is really natural, valid, and therefore meaningful?" and "How can we enforce a hierarchical clustering algorithm to identify only the meaningful cluster structure?"

Information Theory for Clustering. We give the answer to these questions by relating the hierarchical clustering problem to that of information theory and data compression. Imagine you want to transfer the data set via an extremely expensive and small-banded communication channel. Then you can interpret the cluster hierarchy as a statistical model of the data set, which defines more or less likely areas of the data space. The knowledge of these probabilities can be used for an efficient compression of the data set: Following the idea of (optimal) Huffman coding, we assign few bits to points in areas of high probability and more bits to areas of low probability. The interesting observation is the following: the compression becomes the more effective, the better our

J.L. Balcázar et al. (Eds.): ECML PKDD 2010, Part I, LNAI 6321, pp. 151–167, 2010.



(a) Data set: △ display outliers. (b) Hierarchical representation. (c) Cluster contents on each level modeled by PDFs.

Fig. 1. Contributions of ITCH

statistical model, the hierarchical cluster structure, fits to the data. This so-called *Minimum Description Length* (MDL) principle has recently received increasing attention in the context of partitioning (i.e. non-hierarchical) clustering methods. Note that it can not only be used to assess and compare the quality of the clusters found by different algorithms and/or varying parameter settings. Rather, we use this concept as an objective function to implement clustering algorithms directly using simple but efficient optimization heuristics.

In this paper, we extend the idea of MDL to the hierarchical case and develop hMDL for *hierarchical* cluster structures. Whereas previous MDL approaches can only evaluate the result of partitioning clustering methods, our new hMDL criterion is able to assess a complete cluster hierarchy. Moreover, hMDL can be used in combination with an optimization heuristic to exactly determine that cluster hierarchy, which optimizes the data compression according to the MDL criterion.

Challenges and Contributions. With an assessment condition for cluster hierarchies, we develop a complete hierarchical clustering approach on top of the idea of hMDL. This proposed algorithm ITCH (**I**nformation-**T**heoretic **C**luster **H**ierarchies) yields numerous advantages, out of which we demonstrate the following four:

1. All single clusters as well as their hierarchical arrangement are guaranteed to be **meaningful**. Nodes only are placed in the cluster hierarchy if they improve the data compression. This is achieved by optimizing the hMDL criterion. Moreover, a maximal consistency with partitioning clustering methods is obtained.
2. Each cluster is represented by an **intuitive description** of its content in form of a Gaussian probability density function (PDF). Figure 1(c) presents an example of a 3-stage hierarchy. The output of conventional methods is often just the (hierarchical) cluster structure and the assignment of points.
3. ITCH is **outlier-robust**. Outliers are assigned to the root of the cluster hierarchy or to an appropriate inner node, depending on the degree of outlierness. For example, in Figures 1(a) and 1(b) the outlier w.r.t. the three red clusters at the bottom level is assigned to the parent cluster in the hierarchy, marked by a red triangle.
4. ITCH is **fully automatic** as no difficult parameter settings are necessary.

To the best of our knowledge, ITCH is the only clustering algorithm that meets *all* of the above issues by now. The remainder of this paper is organized as follows: Section 2 gives a brief survey of related work. Section 3 presents a derivation of our hMDL

criterion and introduces the ITCH algorithm. Section 4 documents the advantages of ITCH on synthetic and real data sets. Section 5 summarizes the paper.

2 Related Work

Hierarchical Clustering. One of the most widespread approaches to hierarchical clustering is the Single Link algorithm [12] and its variants [14]. The resulting hierarchy obtained by the merging order is visualized as a tree, which is called dendrogram. Cuts through the dendrogram at various levels obtain partitioning clusterings. However, for complex data sets it is hard to define appropriate splitting levels, which correspond to meaningful clusterings. Furthermore, outliers may cause the well-known Single Link effect. Also, for large data sets, the fine scale visualization is not appropriate. The algorithm OPTICS [1] avoids the Single Link effect by requiring a minimum object density for clustering, i.e. $MinPts$ number of objects are within a hyper-sphere with radius ϵ. Additionally, it provides a more suitable visualization, the so-called reachability plot. However, the problem that only certain cuts represent useful clusterings still remains unsolved.

Model-based Clustering. For many applications and further data mining steps, it is essential to have a model of the data. Hence, clustering with PDFs, which goes back to the popular EM algorithm [8], is a widespread alternative to hierarchical clustering. After a suitable initialization, EM iteratively optimizes a mixture model of K Gaussian distributions until no further significant improvement of the log-likelihood of the data can be achieved. Usually a very fast convergence is observed. However, the algorithm may get stuck in a local maximum of the log-likelihood. Moreover, the quality of the clustering result strongly depends on an appropriate choice of K, which is a non-trivial task in most applications. And even with a suitable choice of K the algorithm is very sensitive w.r.t. noise and outliers.

Model-based Hierarchical and Semi-supervised Clustering. [22] proposes a hierarchical extension of EM to speed up query processing in an object recognition application. In [6] a hierarchical variant of EM is applied for image segmentation. Goldberger and Roweis [9] focus on reducing the number of clusters in a mixture model. The consistency with the initial clustering is assured by the constraint that objects belonging to the same initial cluster must end up after the reduction in the same new cluster as well. Each of these approaches needs a suitable parameter setting for the number of hierarchy levels. Clustering respecting some kind of hierarchy can also be regarded as *semi-supervised clustering*, i.e. clustering with side information. In most of some recently published papers [13,4,3], this information is introduced by constraints on the objects and typically consists of strong expert knowledge. In contrast, ITCH does not consider any external information. The data itself is our single source of knowledge.

Information Theory in the Field of Clustering. Only a few papers on compression based clustering, that avoid difficult parameter settings have been published so far. X-Means [16], G-Means [11] and RIC [5] focus on avoiding the choice of K in partitioning clustering by trying to balance data likelihood and model complexity. This sensitive trade-off can be rated by model selection criteria, among them the Akaike Information

Criterion (AIC), the Bayesian Information Criterion (BIC) and Minimum Description Length (MDL) [10]. X-Means provides a parameter-free detection of spherical Gaussian clusters by a top-down splitting algorithm, which integrates K-Means clustering and BIC. G-Means extends this idea to detect non-spherical Gaussian clusters. The model selection criterion of RIC is based on MDL, which allows to define a coding scheme for outliers and to identify non-Gaussian clusters.

There is a family of further closely related ideas, such as Model-based Clustering [2], the work of Still and Bialek [20] and the so-called Information Bottleneck Method [21], introduced by Tishby *et al*. This technique aims at providing a quantitative notation of *meaningful* or *relevant* information. The authors formalize this perception by finding the best tradeoff between accuracy and complexity when clustering a random variable X, given a joint probability distribution between X and an observed relevant variable Y. It is used for clustering terms and documents [19]. However, all parameter-free algorithms mentioned so far, do not provide any cluster hierarchy. One recent paper [7] presents a new method for clustering based on compression. In the first step, this method determines a parameter-free, universal, similarity distance, computed from the lengths of compressed data files. Afterwards a hierarchical clustering method is applied. In contrast to ITCH, this work was not designed to handle outliers in an appropriate way. Furthermore, no description of the content of a cluster is available.

3 Information-Theoretic Hierarchical Clustering

The clustering problem is highly related to that of data compression: The detected cluster structure can be interpreted as a PDF $f_\Theta(x)$ where $\Theta = \{\theta_1, \theta_2, ...\}$ is a set of parameters, and the PDF can be used for an efficient compression of the data set n. It is well-known that the compression by *Huffman coding* is optimal if the data distribution really corresponds to $f_\Theta(x)$. Huffman coding represents every point x by a number of bits which is equal to the negative binary logarithm of the PDF:

$$C_{\text{data}}(x) = -\log_2(f_\Theta(x)).$$

The better the point set corresponds to $f_\Theta(x)$, the smaller the coding costs $C_{\text{data}}(x)$ are. Hence, $C_{\text{data}}(x)$ can be used as the objective function in an optimization algorithm. However, in data compression, Θ serves as a *code book* which is required to decode the compressed data set again. Therefore, we need to complement the compressed data set with a coding of this code book, the parameter set Θ. When, for instance, a Gaussian Mixture Model (GMM) is applied, Θ corresponds to the weights, the mean vectors and the variances of the single Gaussian functions in the GMM. Considering Θ in the coding costs is also important for the clustering problem, because neglecting it leads to overfitting. For partitioning (non-hierarchical) clustering structures, several approaches have been proposed for the coding of Θ [16,17,18] (cf. Section 2). These approaches differ from each other because there is no unambiguous and natural choice for a distribution function, which can be used for the Huffman coding of Θ, and different assumptions lead to different objective functions. In case of the hierarchical cluster structure in ITCH, a very natural distribution function for Θ exists: With the only exception of the root node, every node in the hierarchy has a parent node. This parent

node is associated with a PDF which can naturally be used as a code book for the mean vector (and indirectly also for the variances) of the child node. The coding costs of the root node, however, are not important, because every valid hierarchy has exactly one root node with a constant number of parameters, and therefore, the coding costs of the root node are always constant.

3.1 Hierarchical Cluster Structure

In this Section, we formally introduce the notion of a hierarchical cluster structure (HCS). A HCS contains clusters $\{A, B, ...\}$ each of which is represented by a Gaussian distribution function. These clusters are arranged in a tree:

Definition 1 (Hierarchical Cluster Structure). *(1) A HCS is a tree $\mathcal{T} = (\mathcal{N}, \mathcal{E})$ consisting of a set of nodes $\mathcal{N} = \{A, B, ...\}$ and a set of directed edges $\mathcal{E} = \{e_1, e_2, ...\}$ where A is a parent of B (B is a child of A) iff $(A, B) \in \mathcal{E}$. Every node $C \in \mathcal{N}$ is associated with a weight W_C and a Gaussian PDF defined by the parameters μ_C and Σ_C such that the sum of the weights equals one:*

$$\sum_{C \in \mathcal{N}} W_C = 1.$$

(2) If a path from A to B exists in $\mathcal{T}$ (or $A = B$) we call A an ancestor of B (B a descendant of A) and write $B \sqsubseteq A$.
(3) The level l_C of a node C is the height of the descendant subtree. If C is a leaf, then C has level $l_C = 0$. The root has the highest level (length of the longest path to a leaf).

The PDF which is associated with a cluster C is a multivariate Gaussian in a d-dimensional data space which is defined by the parameters μ_C and Σ_C (where $\mu_C = (\mu_{C,1}, ..., \mu_{C,d})^{\mathbf{T}}$ is a vector from a d-dimensional space, called the location parameter, and Σ_C is a $d \times d$ covariance matrix) by the following formula:

$$N(\mu_C, \Sigma_C, x) = \frac{1}{\sqrt{(2\pi)^d \cdot |\Sigma_C|}} \cdot \mathbf{e}^{-\frac{1}{2}(x-\mu_C)^{\mathbf{T}} \cdot \Sigma_C^{-1} \cdot (x-\mu_C)}.$$

For simplicity we restrict $\Sigma_C = \text{diag}(\sigma_{C,1}^2, ..., \sigma_{C,d}^2)$ to be diagonal such that the multivariate Gaussian can also be expressed by the following product:

$$\begin{aligned} N(\mu_C, \Sigma_C, x) &= \prod_{1 \le i \le d} N(\mu_{C,i}, \sigma_{C,i}^2, x_i) \\ &= \prod_{1 \le i \le d} \frac{1}{\sqrt{2\pi\sigma_{C,i}^2}} \cdot \mathbf{e}^{-\frac{(x_i-\mu_i)^2}{2\sigma_{C,i}^2}}. \end{aligned}$$

Since we require the sum of all weights in a HCS to be 1, a HCS always defines a function whose integral is ≤ 1. Therefore, the HCS can be interpreted as a complex, multimodal, and multivariate PDF, defined by the mixture of the Gaussians of the HCS $\mathcal{T} = (\mathcal{N}, \mathcal{E})$:

$$f_{\mathcal{T}}(x) = \max_{C \in \mathcal{N}} \{W_C \, N(\mu_C, \Sigma_C, x)\} \text{ with} \int_{\mathbb{R}^d} f_{\mathcal{T}}(x) \mathbf{d}x \le 1.$$

If the Gaussians of the HCS do not overlap much, then the integral becomes close to 1.

The operations, described in Section 3.3, assign each point $x \in DB$ to a cluster of the HCS $\mathcal{T} = (\mathcal{N}, \mathcal{E})$. We distinguish between the *direct* and the *indirect* association. A point is directly associated with that cluster $C \in \mathcal{N}$ the probability density of which is maximal at the position of x, and we write $C = Cl(x)$ and also $x \in C$, with:

$$Cl(x) = \arg\max_{C \in \mathcal{N}} \{W_C \cdot N(\mu_C, \Sigma_C, x)\}\,.$$

As we have already stated in the introduction, one of the main motivations of our hierarchical, information-theoretic clustering method ITCH is to represent a sequence of clustering structures which range from a very coarse (unimodal) view to the data distribution to a very detailed (multi-modal) one, and that all these views are meaningful and represent an individual complex PDF. The ability to cut a HCS at a given level L is obtained by the following definition:

Definition 2 (Hierarchical Cut). *A HCS $\mathcal{T}' = (\mathcal{N}', \mathcal{E}')$ is a hierarchical cut of a HCS $\mathcal{T} = (\mathcal{N}, \mathcal{E})$ at level L (in symbols: $\mathcal{T}' = HC_L(\mathcal{T})$), if the following properties hold:*
(1) $\mathcal{N}' = \{A \in \mathcal{N} | l_A \geq L\}$,
(2) $\mathcal{E}' = \{(A, B) \in \mathcal{E} | l_A > l_B \geq L\}$,
(3) For each $A \in \mathcal{N}'$ the following properties hold:

$$W'_A = \begin{cases} W_A & \text{if } l_A > L \\ \sum_{B \in \mathcal{N}, B \sqsubseteq A} W_B & \text{otherwise,} \end{cases}$$

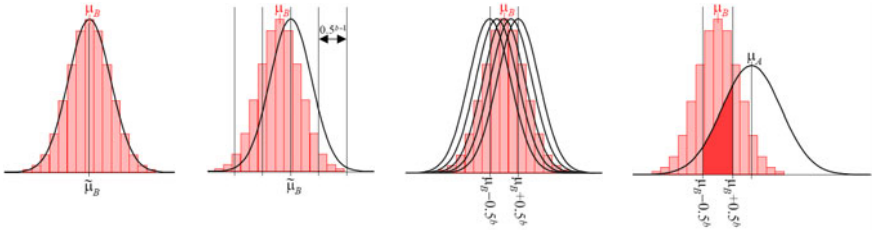
where W_C and W'_C is the weight of node C in $\mathcal{T}$ and $\mathcal{T}'$, respectively.
(4) Analogously, for the direct association of points to clusters the following property holds: Let x be associated with Cluster B in $\mathcal{T}$, i.e. $Cl(x) = B$. Then in $\mathcal{T}'$, x is associated with:

$$Cl'(x) = \begin{cases} B & \text{if } l_B \geq L \\ A | B \sqsubseteq A \wedge l_A = L & \text{otherwise.} \end{cases}$$

Here, the weights of the pruned nodes are automatically added to the leaf nodes of the new HCS, which used to be the ancestors of the pruned nodes. Therefore, the sum of all weights is maintained (and still equals 1), and the obtained tree is again a HCS according to Definition 1. The same holds for the point-to-cluster assignments.

3.2 Generalization of the MDL Principle

Now we explain how the MDL principle can be generalized for hierarchical clustering and develop the new objective function hMDL. Following the traditional MDL principle, we compress the data points according to their negative log likelihood corresponding to the PDF which is given by the HCS. In addition, we penalize the model complexity by adding the code length of the HCS parameters to the negative log likelihood of the data. The better the PDFs of child nodes fit into the PDFs of the parent, the less the coding costs will be. Therefore, the overall coding costs corresponds to the natural, intuitive notion of a good hierarchical representation of data by distribution functions. The discrete assignment of points to clusters allows us to determine the



(a) Exact coding of μ_B. (b) Inexact coding of μ_B. (c) Different Gaussians w.r.t. different grid positions. (d) All possible values for recovered μ_B.

Fig. 2. Optimization of the grid resolution for the hMDL criterion

coding costs of the points clusterwise and dimensionwise, as explained in the following: The coding costs of the points associated with the clusters $C \in \mathcal{N}$ of the HCS $\mathcal{T} = (\mathcal{N}, \mathcal{E})$ corresponds to:

$$C_{\text{data}} = -\log_2 \prod_{x \in DB} \max_{C \in \mathcal{N}} \left\{ W_C \prod_{1 \le j \le d} N(\mu_{C,j}, \sigma^2_{C,j}, x_j) \right\}.$$

Since every point x is associated with that cluster C in the HCS which has maximum probability density, we can rearrange the terms of the above formula and combine the costs of all points that are in the same cluster:

$$\begin{aligned}
&= -\sum_{x \in DB} \log_2 \left(W_{Cl(x)} \prod_{1 \le j \le d} N(\mu_{Cl(x),j}, \sigma^2_{Cl(x),j}, x_j) \right) \\
&= -\left(\left(\sum_{C \in \mathcal{N}} nW_C \log_2 W_C \right) + \left(\sum_{x \in DB;\, 1 \le j \le d} \log_2 N(\mu_{Cl(x),j}, \sigma^2_{Cl(x),j}, x_j) \right) \right) \\
&= -\sum_{C \in \mathcal{N}} \left(nW_C \log_2 W_C + \sum_{x \in C;\, 1 \le j \le d} \log_2 N(\mu_{C,j}, \sigma^2_{C,j}, x_j) \right).
\end{aligned}$$

The ability to model the coding costs of each cluster separately allows us now, to focus on a single cluster, and even on a single dimension of a single cluster. A common interpretation of the term $-nW_C \log_2 W_C$, which actually comes from the weight a single Gaussian contributes to the GMM, is a Huffman coding of the cluster ID. We assume that every point carries the information which cluster it belongs to, and a cluster with many points gets a shortly coded cluster ID. These costs are referred to the *ID cost* of a cluster C. Lets consider two clusters, A and B, where $B \sqsubseteq A$. We now want to derive the coding scheme for the cluster B and its associated points. Several points are associated with B, where the overall weight of assignment sums up to W_B. When coding the parameters of the associated PDF of B, i.e. μ_B, and σ_B, we have to consider two aspects: (1) The *precision* both parameters should be coded to minimize the overall description length depends on W_B, as well as on σ_B. For instance, if only few points are associated with cluster B and/or the variance σ_B is very large, then it is not necessary to know the position of μ_B very precisely and vice versa. (2) The knowledge of the PDF

of cluster A can be exploit for the coding of μ_B, because for likely positions (according to the PDF of A) we can assign fewer bits. Basically, model selection criteria, such as the Bayesian Information Criterion (BIC) or the Aikake Information Criterion (AIC) already address the first aspect, but not the hierarchical aspect. To make this paper self contained, we consider both aspects. In contrast to BIC, which uses the natural logarithm, we use the binary logarithm to represent the code length in *bits*. For simplicity, we assume that our PDF is univariate and the only parameter is its mean value μ_B. We neglect σ_B by assuming e.g. some fixed value for all clusters. We drop these assumptions at the end of this section. When the true PDF of cluster B is coded inexactly by some parameter $\widetilde{\mu}_B$, the coding costs for each point x (which truly belongs to the distribution $N(\mu_B, \sigma_B^2, x)$) in B is increased compared to the *exact* coding of μ_B, which would result in c_{ex} bits per point:

$$c_{\text{ex}} = \int_{-\infty}^{+\infty} -\log_2(N(\mu_B, \sigma_B^2, x)) \cdot N(\mu_B, \sigma_B^2, x) \;\; \mathbf{d}x = \log_2(\sigma_B \sqrt{2\pi \cdot \mathbf{e}}).$$

If $\widetilde{\mu}_B$ instead of μ_B is applied for compression, we obtain:

$$c(\widetilde{\mu}_B, \mu_B) = \int_{-\infty}^{+\infty} -\log_2(N(\widetilde{\mu}_B, \sigma_B^2, x)) \cdot N(\mu_B, \sigma_B^2, x)\, \mathbf{d}x.$$

The difference is visualized in Figure 2(a) and 2(b) respectively: In 2(a) $\widetilde{\mu}_B$ of the coding PDF, depicted by the Gaussian function, fits exactly to μ_B of the data distribution, represented by the histogram. This causes minimum code lengths for the compressed points but also a considerable effort for the coding of μ_B. In Figure 2(b) μ_B is coded by some regular quantization grid. Thereby, the costs for the cluster points slightly increase, but the costs for the location parameter decreases. The difference between $\widetilde{\mu}_B$ and μ_B depends on the bit resolution and on the position of the quantization grid. One example is depicted in Figure 2(b) by five vertical lines, the Gaussian curve is centered by the vertical line closest to μ_B. We derive lower and upper limits of $\widetilde{\mu}_B \in [\mu_B - 1/2^b ... \mu_B + 1/2^b]$ from the number of bits b, spent for coding $\widetilde{\mu}_B$. The real difference between μ_B and $\widetilde{\mu}_B$ depends again on the grid position. Not to prefer clusters that are incidentally aligned with the grid cells, we average over *all* possible positions of the discretization grid. Figure 2(c) presents five different examples of the infinitely many Gaussians that could be recovered w.r.t. different grid positions. Note that all positions inside the given interval have equal probability. Hence, the average coding costs for every possible position of $\widetilde{\mu}_B$ can be expressed by the following integral:

$$\begin{aligned} c_{\text{appx}}(b) &= 2^{b-1} \int_{\mu_B - 1/2^b}^{\mu_B + 1/2^b} c(\widetilde{\mu}_B, \mu_B)\, \mathbf{d}\widetilde{\mu}_B \\ &= \frac{1}{2} \log_2(\pi \cdot \mathbf{e} \cdot \sigma_B^2) + \frac{1}{2} + \frac{\log_2 \mathbf{e}}{6\sigma_B^2} \cdot 4^{-b}. \end{aligned}$$

Coding all $n \cdot W_B$ coordinates of the cluster points as well as the parameter μ_B (neglecting the ID cost) requires then the following number of bits:

$$C_{\text{appx}}(B) = c_{\text{appx}}(b) \cdot n \cdot W_B + b.$$

The optimal number b_{opt} of bits is determined by setting the derivation of the above term to zero.

$$\frac{\mathbf{d}}{\mathbf{d}b} C_{\text{appx}}(B) = 0 \implies b_{\text{opt}} = \frac{1}{2}\log_2\left(\frac{n \cdot W_B}{3 \cdot \sigma_B^2}\right).$$

The unique solution to this equation corresponds to a *minimum*, as can easily be seen by the second derivative.

Utilization of the Hierarchical Relationship. We do not want to code the (inexact) position of μ_B without the prior knowledge of the PDF associated with cluster A. Without this knowledge, we would have to select a suitable range of values and code μ_B at the determined precision b assuming e.g. a uniform distribution inside this range. In contrast, μ_B is a value taken from the distribution function of cluster A. Hence, the number of bits used for coding of μ_B corresponds to the overall density around the imprecise interval defined by μ_B, i.e.

$$c_{\text{hMDL}}(\mu_B) = -\log_2 \int_{\mu_B - 1/2^b}^{\mu_B + 1/2^b} N(\mu_A, \sigma_A^2, x)\, \mathbf{d}x.$$

Figure 2(d) visualizes the complete interval of all possible values for the recovered mean value (marked in red) and illustrates the PDF of the cluster$A$, which is the predecessor of cluster B. $\widetilde{\mu}_B$ can be coded by determining the whole area under the PDF of A where $\widetilde{\mu}_B$ could be. The area actually corresponds to a probability value. The negative logarithm of this probability represents the required code length for μ_B. The costs for coding all points of cluster B and μ_B then corresponds to

$$c_{\text{appx}}(b) \cdot n \cdot W_B + c_{\text{hMDL}}(\mu_B).$$

Note, that it is also possible to optimize b directly by setting the derivative of this formula to zero. However, this is impossible in an analytic way, and the difference to the optimum which is obtained by minimizing $C_{\text{appx}}(B)$ is negligible. In addition, if the parent A of B is not the root of the HCS, μ_B causes some own ID cost. In this case, μ_B is a sample from the complex distribution function of the hierarchical cut (cf. Definition 2), which prunes the complete level of B and all levels below. Hence, the weight of these levels is added to the new leaf nodes (after cutting), and the ID costs of μ_B correspond to:

$$-\log_2\Big(\sum_{X \sqsubseteq A} W_X\Big).$$

A similar analysis can be done for the second parameter of the distribution function, σ_B. Since it is not straightforward to select a suitable distribution function for the Huffman coding of variances, one can apply a simple trick: Instead of coding σ_B, we code $y_B = \mu_B \pm v \cdot \sigma_B$, where v is a constant close to zero. Then, y_B is also a sample from the distribution function $N(\mu_A, \sigma_A^2, x)$ and can be coded similar to μ_B. Therefore, $c_{\text{hMDL}}(\sigma_B) = c_{\text{hMDL}}(\mu_B)$, and we write $c_{\text{hMDL}}(\text{param})$ for the coding costs per parameter instead. In general, if the PDF, which is associated with a cluster has r parameters, then the optimal number of bits can be obtained by the formula:

$$b_{\text{opt}} = \frac{1}{2}\log_2\left(\frac{n \cdot W_B}{3 \cdot r \cdot \sigma_B^2}\right).$$

And the overall coding costs are:

$$C_{\text{hMDL}}(B) = c_{\text{appx}}(b) \cdot n \cdot W_B + r \cdot c_{\text{hMDL}}(\text{param})$$

Until now, only the trade-off between coding costs of points and the parameters of the assigned cluster are taken into account. If we go above the lowest level of the HCS, we have to trade between coding costs of parameters at a lower level and coding costs of the parameters at the next higher level. This can be done in a similar way as before: Let b_B be the precision, which has already been determined for the representation of μ_B and σ_B, the parameters for cluster B, which is a subcluster of A. However, this is the minimum coding costs assuming that μ_A and σ_A have been stored at maximum precision, and that μ_B and σ_B are also given. Now, we assume that μ_B is an arbitrary point selected from the distribution function $N(\mu_A, \sigma_A^2, x)$ and determine an expectation for the cost:

$$\int_{-\infty}^{+\infty} -\log_2 \int_{\mu_B - 1/2^{b_B}}^{\mu_B + 1/2^{b_B}} N(\mu_A, \sigma_A^2, x) \mathbf{d}x \; N(\mu_A, \sigma_A^2, \mu_B) \mathbf{d}\mu_B .$$

Finally, we assume that μ_A is also coded inexactly by its own grid with resolution b_A. Then the expected costs are:

$$2^{b_A - 1} \int_{\mu_A - 1/2^{b_A}}^{\mu_A + 1/2^{b_A}} \int_{-\infty}^{+\infty} \left(-\log_2 \int_{\mu_B - 1/2^{b_B}}^{\mu_B + 1/2^{b_B}} N(y, \sigma_A^2, x) \; \mathbf{d}x \right) \cdot \\ \cdot N(\mu_A, \sigma_A^2, \mu_B) \; \mathbf{d}\mu_B \; \mathbf{d}y.$$

Since it is analytically impossible to determine the optimal value of b_A, we can easily get an approximation of the optimum by simply treating μ_B and σ_B like the points which are directly associated with the cluster A. The only difference is the following. While the above integral considers that the PDF varies inside the interval $[\mu_B - 1/2^{b_B}, \mu_B + 1/2^{b_B}]$ and determines the average costs in this interval, treating the parameters as points only considers the PDF value at one fixed position. This difference is negligible provided that $\sigma_B < \sigma_A$, which makes sense as child clusters should usually be much smaller (in terms of σ) than their parent cluster.

Coding Costs for a Cluster. Summarizing, the coding costs for a cluster can be obtained as follows: (1) Determine the optimal resolution parameter for each dimension according to the formula:

$$b_{\text{opt}} = \frac{1}{2} \log_2 \left(\frac{n \cdot W_B + r \cdot \#\text{ChildNodes}(B)}{3 \cdot r \cdot \sigma_B^2} \right).$$

(2) Determine the coding costs for the data points and the parameters according to:

$$C_{\text{hMDL}}(B) = c_{\text{appx}}(b) \cdot n \cdot W_B + r \cdot c_{\text{hMDL}}(\text{param})$$

(3) Add the costs obtained in step (2) to the ID costs of the points $(-nW_B \log_2(W_B))$ and of the parameters $(-\log_2(\sum_{X \sqsubseteq A} W_X))$. Whereas the costs determined in (2) are individual in each dimension the costs in (3) occur only once per stored point or parameter set of a cluster.

Coding Costs for the HCS. The coding costs for all clusters sum up to the overall coding costs of the hierarchy where we define constant ID costs for the parameters of the root:

$$hMDL = \sum_{C \in \mathcal{N}} \left(C_{\text{hMDL}}(C) - nW_C \log_2(W_C) - \log_2(\sum_{X \sqsubseteq \text{parent of } C} W_X) \right).$$

3.3 Obtaining and Optimizing the HCS

We optimize our objective function in an EM-like clustering algorithm ITCH. Reassignment of objects and re-estimation of the parameters of the HCS are done interchangeably until convergence. Starting from a suitable initialization, ITCH periodically modifies the HCS.

Initialization of the HCS. Clustering algorithms that follow the EM-scheme have to be suitable initialized before starting with the actual iterations of E- and M-step. An established method is to initialize with the result of a K-Means clustering. This is typically repeated several times with different seeds and the result with best mean squared overall deviation from the cluster centers is taken. Following this idea, ITCH uses a initialization hierarchy determined by a bisecting K-Means algorithm taking the hMDL value of the HCS as a stopping criterion for partitioning. First, a root node that contains all points is created. Then this root node is partitioned into two subclusters by applying K-Means with $K = 2$. This is done recursively until the hMDL of the binary HCS does not improve anymore within three steps. This ensures not to get stuck in a local minimum. Finally, after the best hierarchy is selected, μ_C and Σ_C are determined for each node C according to Section 3.2, and equal weights are assigned to the nodes, to ensure that clusters compeed likewise for the data points.

E-step and M-step. Whenever an object is associated directly to a cluster C then it is also indirectly associated with every ancestor of C. Nevertheless, points can also be directly associated not only to leaf nodes but also to inner nodes of the HCS. For instance, if a point P_i is an outlier w.r.t. any of the clusters at the bottom level of the HCS, then P_i has to be associated with an inner node or even the root. As established in Section 3.1, the clusters at all levels of the HCS compete for the data points. A point x is directly associated with that Cluster $C \in \mathcal{N}$ the probability density function of which is maximal:

$$Cl(x) = \arg\max_{C \in \mathcal{N}} \{W_C \cdot N(\mu_C, \Sigma_C, x)\}.$$

In the E-step of our hierarchical clustering algorithm, the direct association $Cl(x)$ for every object x is updated. Whereas, in the E-step only the direct association is used in the M-step which updates the location and scale parameters of all clusters we use both the direct and indirect association. The motivation is the following: The distribution function of every node in the HCS should always represent the whole data set in this branch of the tree, and the root node should even represent the complete data set. Therefore, for the location and scale parameters, all directly and indirectly associated objects are considered, as in the following formulas:

$$\mu_C = \frac{\sum_{B \in \mathcal{N}, B \sqsubseteq C} \left(\sum_{x \in B} x \right)}{\sum_{B \in \mathcal{N}, B \sqsubseteq C} |B|}, \sigma^2_{C,j} = \frac{\sum_{B \in \mathcal{N}, B \sqsubseteq C} \left(\sum_{x \in B} (x_j - \mu_{C,j})^2 \right)}{\sum_{B \in \mathcal{N}, B \sqsubseteq C} |B|}$$

$$\Sigma_C = \text{diag}(\sigma^2_{C,1}, ..., \sigma^2_{C,d}).$$

In contrast, the weight W_C of each cluster should reflect the strenght of the individual Gaussian in the overall mixture of the HCS and sum up to 1 in order to define a valid

PDF with an integral over the complete data space of 1. Therefore, we use the direct associations for calculating the cluster weight with $W_C = |C|$.

Rearrangement of the HCS

The binary HCS that results from the initialization, does not limit our generality. ITCH is flexible enough to convert the initial HCS into a general one. Given a binary hierarchy, which is deeper than any n-ary hierarchy with $n > 2$, ITCH aims in flattening the HCS as far as the rearrangement improves our hMDL criterion. Therefore we trade off the two operations *delete* or *collapse* a node to eliminate clusters that do not pay off any more. Figure 3 visualizes the operations for an extract of a given HCS. By deleting a cluster C, the child nodes of C become child nodes of the parent of C (Figure 3(b)). By collapsing C, all of its child nodes are merged into a new cluster C' (including C), and therefore all of their child nodes become child nodes of C' (Figure 3(c)). Afterwards all points are redistributed, and E- and M-step are performed alternately until convergence. ITCH rearranges the HCS in an iterative way. In each iteration we tentatively delete/collapse each node in the HCS and perform E- and M-steps. Then first, the node and the operation that improves the hMDL criterion best is selected and second, the corresponding local neighborhood (parent, child and sibling nodes) is processed. These two steps are performed alternately until convergence.

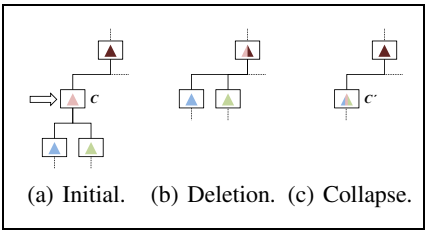


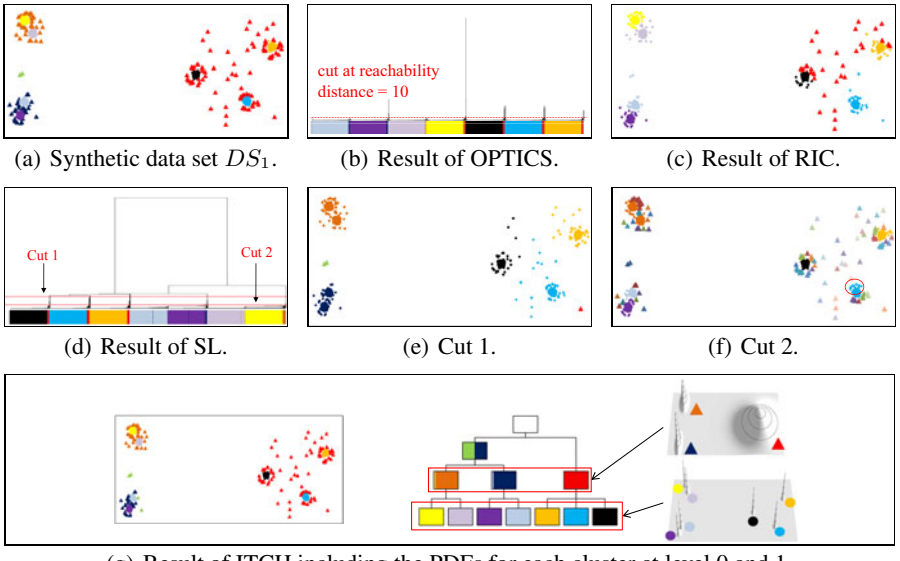
Fig. 3. Restructure operations of ITCH

4 Experimental Evaluation

Since ITCH is a hybrid approach combining the benefits of hierarchical *and* model-based clustering, we compare to algorithms of both classes to demonstrate the effectiveness of ITCH. We selected Single Link (SL) which probably is the most common approach to hierarchical clustering. As especially on noisy data, SL suffers from the so-called Single Link effect, we additionally compare to OPTICS, a more outlier-robust hierarchical clustering algorithm. Unless otherwise mentioned, OPTICS is parameterized with $\epsilon = 10,000$ and $MinPts = 10$. For an extensive description of parameterization strategies, we refer to [1]. Furthermore, we compare to RIC, an outlier-robust and parameter-free state-of-the-art algorithm to model-based clustering. In all plots, we mark cluster points by circles and outliers by triangles respectively. To relieve evaluation w.r.t. outliers, we added a color bar below the dendrograms of SL and the reachability plots of OPTICS, where colors refer to the class labels in the original data.

4.1 Synthetic Data

Experiments on DS_1 demonstrate the superiority of ITCH on hierarchical data sets. DS_1 comprises about 3,500 2-dimensional points that form a hierarchy of 12 clusters with outliers at different levels of the hierarchy. Seven Gaussian clusters are located at



(a) Synthetic data set DS_1. (b) Result of OPTICS. (c) Result of RIC.

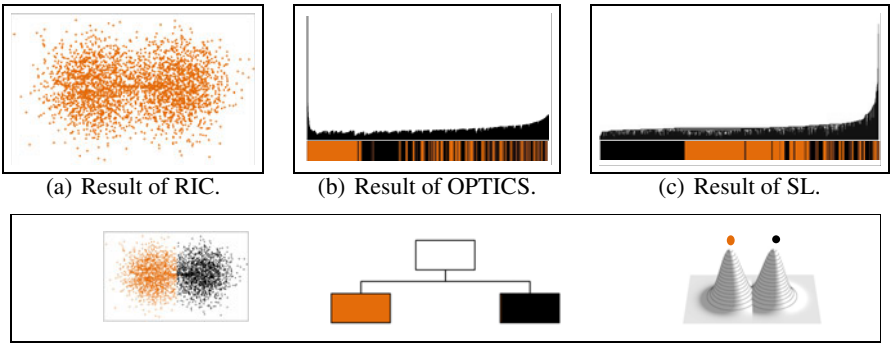
(d) Result of SL. (e) Cut 1. (f) Cut 2.

(g) Result of ITCH including the PDFs for each cluster at level 0 and 1.

Fig. 4. Experimental evaluation on synthetic data set $\mathbf{DS_1}$

the bottom level (Figure 4(a)). Experiments on DS_2 indicate the limitations of existing approaches to form meaningful clusters in extremely noisy non-hierarchical data. DS_2 is composed of two Gaussian clusters with 1,650 points each, overlapping in the marginal area without any global outliers. The quantitative evaluation of the results is always performed w.r.t. the "true" hierarchies present in these data sets.

Experimental Evaluation on $\mathbf{DS_1}$. As Clusters can be recognized as valleys in the reachability plot, OPTICS yields a satisfactory result (Precision: 94.8% Recall: 95.4% w.r.t. reachability distance < 10). But without our added color bar it would be impossible to spot the outliers since high distance peaks can also be caused by the usual jumps (Figure 4(b)). At a first glance, the SL-hierarchy (Figure 4(d)) reflects the true hierarchy quite well. However, a closer look at the data partitioning w.r.t. different cuts does not lead to meaningful clusters. Figure 4(e) illustrates the data that refers to a cut resulting in seven clusters. SL identifies only five clusters and three outliers (Precision: 70.0% Recall: 85.9%). The four clusters on the left side are wrongly combined into two clusters. Even at a much deeper split (Figure 4(f)) this effect remains for the orange cluster. Actually, the cluster quality is getting worse (Precision: 8.5% Recall: 9.0%) as the multiple outliers w.r.t. the three subclusters on the right side cause the well-known SL effect. Even though, each outlier is assigned to a single cluster the points marked by a red circle are not identified as outliers. Altogether, it is extremely hard to find the right parameter to cut through the dendrogram which gives a meaningful cluster representation. In order to apply RIC to the hierarchical data set, we preprocessed DS_1 with SL and applied RIC as postprocessing step in each level of the hierarchy. Figure 4(c) demonstrates the result when applying RIC to Cut 1 of the SL dendrogram (Precision:



(a) Result of RIC. (b) Result of OPTICS. (c) Result of SL.

(d) Result of ITCH including the PDFs for both clusters at level 0.

Fig. 5. Experimental evaluation on synthetic data set $\mathbf{DS_2}$

94,9% Recall: 92,2%). It is obvious that even RIC fails to successfully filter out all outliers. More precisely, RIC assigns points (marked by a dark blue and orange triangle in the original data) that obviously are outliers w.r.t. two clusters on the left upper and lower side misleadingly to clusters. Also a majority of the red outliers are incorrectly identified as cluster points. ITCH is the best method to detect the true cluster hierarchy including outliers fully automatically (Precision: 93.8% Recall: 97.5%), and ITCH provides meaningful models on the data for each level of the hierarchy (Figure 4(g)).

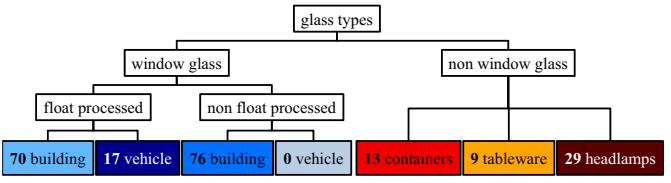
Experimental Evaluation on $\mathbf{DS_2}$. Figure 5(a) demonstrates that RIC merges the two Gaussian clusters into only one cluster (Precision: 50.0% Recall: 100.0%). Also with OPTICS, it is impossible to detect the true structure of DS_2. The color bar in Figure 5(b) indicates that OPTICS assigns the points in an almost arbitrary order. Even when increasing the parameter for the minimum object density per cluster to a large value, OPTICS fails in detecting two clusters. SL miscarries due to the massive SL effect (Figure 5(c)). Here, OPTICS is not suitable to cure that problem. Moreover, the hierarchies generated by OPTICS and SL are overly complex but do not capture any cluster structure. Hence, it is not possible to evaluate these results in a quantitative fashion. Only ITCH discovers a meaningful result without requiring any input parameters (Precision: 99.2% Recall: 99.7%). All clusters that do not pay off w.r.t. our hMDL are pruned and hence, only two Gaussian clusters remain in the resulting flat hierarchy which are described by an intuitive description in form of a PDF (Figure 5(d)).

4.2 Real World Data

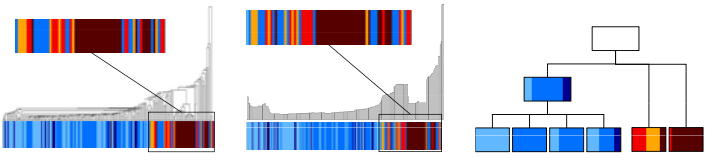
Finally, we show the practical application of ITCH on real data sets available at UCI[1].

Glass Data. The *Glass Identification* data set comprises nine numerical attributes representing different glass properties. 214 instances are labelled according to seven different types of glass that form a hierarchy as presented in Figure 6(a). ITCH perfectly separates *window glass* from *non window glass*. Additionally, *tableware* and *containers* are

[1] `http://archive.ics.uci.edu/ml/`



(a) Hierarchy in the original data set.

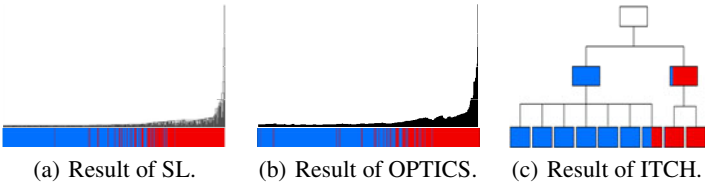


(b) Result of SL. (c) Result of OPTICS. (d) Result of ITCH.

Fig. 6. Hierarchical clustering of 9-dimensional glass data (214 instances)

almost perfectly separated from *headlamps*. The four subclusters of *window glass* are very similar. Hence, ITCH arranges them at the same level. Some outliers are directly assigned to *window glass*. In contrast to ITCH, neither SL nor OPTICS separates *window glass* from *non window glass* perfectly (Figures 6(b) and 6(c)). *Containers* and *tableware* do not form discrete clusters but are constituted as outliers instead. In the dendrogram only the *headlamps* can be identified, whereas in the reachability plot two clusters are visible. Nevertheless, both approaches do not reflect the original hierarchy successfully. As it is not clear where to define an adequate cut through the dendrogram we applied RIC at the bottom level. This results in only two clusters without any separation between *window glass* or *non window glass*.

Cancer Data. The high-dimensional *Breast Cancer Wisconsin* data set contains 569 instances each describing 30 different characteristics of the cell nuclei, where each instance is either labelled by *benign* (blue) or *malignant* (red). OPTICS and SL both fail to detect a clear cluster structure in this data set (Figures 7(a) and 7(b)). Hence, we applied RIC on top of a K-Means clustering with K=15. As stated by the authors we chose K large enough compared to the number of classes. However, RIC also fails and results in three mixed clusters. In contrast, despite the high dimensionality of the data, ITCH almost perfectly separates the *benign* from the *malignant* objects which are then split



(a) Result of SL. (b) Result of OPTICS. (c) Result of ITCH.

Fig. 7. Hierarchical clustering of 30-dimensional breast cancer data (569 instances)

into different subclusters (Figure 7(c)). This result is consistent with previous findings as the two classes exhibit a degree of overlap with each other [15].

4.3 Stability of ITCH

Since we do not want to rely on single results we additionally tested the stability of ITCH over 20 runs for each data set. Figure 8 shows the variance of the hMDL value in percent depending on the mean value. The result of ITCH is highly stable within DS1, DS2 having only a variance of 0.03% and 0.12%, respectively. Also in the real world data sets the result of ITCH shows only little variance.

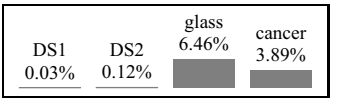


Fig. 8. Stability of the ITCH result over 20 runs

5 Conclusions

We have introduced a new hierarchical clustering method to arrange only natural, valid, and meaningful clusters in a hierarchical structure – ITCH. ITCH is based on an objective function for clustering that was guided by the information-theoretic idea of data compression. We have shown that without difficult parameter settings ITCH finds the *real* cluster hierarchy effectively, and that it provides accurate and intuitive interpretable information in a wide variety of domains, even in the presence of outliers.

Acknowledgements. We thank Johannes Huber for assisting us with the evaluation.

References

1. Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: OPTICS: Ordering Points To Identify the Clustering Structure. In: SIGMOD, pp. 49–60 (1999)
2. Banfield, J.D., Raftery, A.E.: Model-Based Gaussian and Non-Gaussian Clustering. Biometrics 49(3), 803–821 (1993)
3. Basu, S., Bilenko, M., Mooney, R.J.: A Probabilistic Framework for Semi-supervised Clustering. In: KDD, pp. 59–68 (2004)
4. Bilenko, M., Basu, S., Mooney, R.J.: Integrating Constraints and Metric Learning in Semi-supervised Clustering. In: ICML (2004)
5. Böhm, C., Faloutsos, C., Pan, J.Y., Plant, C.: Robust Information-theoretic Clustering. In: KDD. pp. 65–75 (2006)
6. Chardin, A., Pérez, P.: Unsupervised Image Classification with a Hierarchical EM Algorithm. In: ICCV, pp. 969–974 (1999)
7. Cilibrasi, R., Vitányi, P.M.B.: Clustering by Compression. IEEE Transactions on Information Theory 51(4), 1523–1545 (2005)
8. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum Likelihood from Incomplete Data via the EM Algorithm. J. Roy. Stat. Soc. 39, 1–31 (1977)
9. Goldberger, J., Roweis, S.T.: Hierarchical Clustering of a Mixture Model. In: NIPS (2004)
10. Grünwald, P.: A Tutorial Introduction to the Minimum Description Length Principle. CoRR math.ST/0406077 (2004)
11. Hamerly, G., Elkan, C.: Learning the K in K-means. In: NIPS (2003)

12. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice-Hall, Englewood Cliffs (1988)
13. Lu, Z., Leen, T.K.: Semi-supervised Learning with Penalized Probabilistic Clustering. In: NIPS (2004)
14. Murtagh, F.: A Survey of Recent Advances in Hierarchical Clustering Algorithms. Comput. J. 26(4), 354–359 (1983)
15. Pantazi, S., Kagolovsky, Y., Moehr, J.R.: Cluster analysis of wisconsin breast cancer dataset using self-organizing maps (2002)
16. Pelleg, D., Moore, A.W.: X-means: Extending K-means with Efficient Estimation of the Number of Clusters. In: ICML, pp. 727–734 (2000)
17. Rissanen, J.: Stochastic Complexity in Statistical Inquiry Theory. World Scientific Publishing Co., Inc., River Edge (1989)
18. Rissanen, J.: Information and Complexity in Statistical Modeling. Springer Publishing Company, Incorporated (2007)
19. Slonim, N., Tishby, N.: Document Clustering using Word Clusters via the Information Bottleneck Method. In: SIGIR, pp. 208–215 (2000)
20. Still, S., Bialek, W.: How Many Clusters? An Information-Theoretic Perspective. Neural Computation 16(12), 2483–2506 (2004)
21. Tishby, N., Pereira, F.C., Bialek, W.: The information bottleneck method. CoRR physics/0004057 (2000)
22. Vasconcelos, N., Lippman, A.: Learning Mixture Hierarchies. In: NIPS. pp. 606–612 (1998)