Open access • Book Chapter • DOI:10.1007/978-3-642-32717-9_21

Information Theoretic Clustering Using Minimum Spanning Trees — Source link

Andreas Müller, Sebastian Nowozin, Christoph H. Lampert

Institutions: University of Bonn, Microsoft, Institute of Science and Technology Austria

Published on: 28 Aug 2012

Topics: Cluster analysis, Mutual information, Spanning tree and Rand index

Information Theoretic Clustering using Minimum Spanning Trees

Andreas C. Müller*[1], Sebastian Nowozin[2], and Christoph H. Lampert[3]

[1] University of Bonn, Germany
[2] Microsoft Research, Cambridge, UK
[3] IST Austria, Klosterneuburg, Austria

Abstract. In this work we propose a new information-theoretic clustering algorithm that infers cluster memberships by direct optimization of a non-parametric mutual information estimate between data distribution and cluster assignment. Although the optimization objective has a solid theoretical foundation it is hard to optimize. We propose an approximate optimization formulation that leads to an efficient algorithm with low runtime complexity. The algorithm has a single free parameter, the number of clusters to find. We demonstrate superior performance on several synthetic and real datasets.

1 Introduction

Clustering data is one of the fundamental problems in machine learning. In clustering, the goal is to divide data points into homogeneous subsets, called clusters. Many different formulations of the clustering problem are given in the literature. Most algorithms are based on ad-hoc criteria such as intra-cluster similarity and inter-cluster dissimilarity. An alternative approach is to formalize clustering using an information theoretic framework, where one considers inputs as well as cluster assignments as random variables. The goal is then to find an assignment of data points to clusters that maximizes the mutual information between the assignments and the observations.

In this work, we rely on a non-parametric estimator of the data entropy to find clusterings of maximum mutual information. The use of non-parametric estimates allows a data-driven approach, without making strong assumptions on the form of the data distribution. As a consequence, we obtain a very flexible model that, e.g., allows non-convex clusters. The resulting objective is easy to evaluate, but difficult to optimize over. We overcome this by proposing an efficient approximate optimization based on the Euclidean minimum spanning tree algorithm. Because the estimator and the optimization are both parameter-free, the only free parameter of the algorithm is the number of clusters, which makes it very easy to use in practice. The contributions of this work are:

- Proposing the use of a MST-based entropy estimator in information theoretic clustering.
- Give a fast algorithm for a relaxed version of the resulting problem.
- Show the practicality on a number of synthetic and real datasets.

* This work was partially funded by the B-IT research school.

2 Related Work

The most commonly used clustering algorithm is the k-Means algorithm, also known as Lloyd's algorithm [14, 13]. While k-Means often works well in practice, one of its main drawbacks is the restriction in cluster shape. They are given by the Voronoi tessellation of the cluster means and therefore always convex.

Another widely used method is spectral clustering [20, 16], which solves a graph partitioning problem on a similarity graph constructed from the data. While spectral clustering is much more flexible than k-Means it is quite sensitive to the particular choice of graph construction and similarity measure. It is also computationally expensive to compute, because clustering n points requires computing the eigenvalues and -vectors of an $n \times n$ matrix.

Information theoretic approaches to clustering were first investigated in the context of document classification. In this setting, training examples are described by a discrete distribution over words, leading to the task of *distributional clustering*, which was later related to the Information Bottleneck method by [21]. This setting was described in detail by [4]. In distributional clustering, it is assumed that the distribution of the data is known explicitly (for example as word counts), which is not the case in our setting.

Later, Banerjee et al. [1] introduced the concept of Bregman Information, generalizing mutual information of distributions, and showed how this leads to a natural formulation of several clustering algorithms. Barber [2] construct a soft clustering by using a parametric model of $p(Y \mid X)$. The framework of mutual information based clustering was extended to non-parametric entropy estimates by Faivishevsky and Goldberger [5]. They use a nearest neighbor based estimator of the mutual information, called MeanNN, that takes into account all possible neighborhoods, therefore combining global and local influences. The approximate mutual information is maximized using local search over labels.

Clustering algorithms based on minimum spanning trees have been studied early on in the statistics community, due to their efficiency. One of the earliest methods is single-link agglomerative clustering [8]. Single-link agglomerative clustering can be understood as a minimum spanning tree-based approach in which the largest edge is removed until the desired number of components is reached. Zahn [23] refines this criterion by cutting edges that are longer than other edges in the vicinity. This approach requires tuning several constants by hand. More recently, Grygorash et al. [9] proposed a hierarchical MST-based clustering approach that iteratively cuts edges, merges points in the resulting components, and rebuilds the spanning tree. We will limit our discussion to the most widely used algorithm from [8].

3 Information Theoretic Clustering Using Nonparametric Entropy-Estimates

In general, the goal of clustering can be formulated as follows: given a finite collection of samples $\mathbf{x} = (x_1, \ldots, x_n)$, we want to assign cluster-memberships $\mathbf{y} = (y_1, \ldots, y_n), y_i \in \{1, \ldots k\}$ to these samples. We adopt the viewpoint of information theoretic clustering of Gokcay and Principe [6], where the x_i are

considered i.i.d. samples from a distribution $p(X)$, and the y_i are found such that the mutual information $I(X,Y)$ between the distribution $p(X)$ and the assigned labels $p(Y)$ is maximized. We can rewrite this objective as

$$I(X,Y) = D_{\mathrm{KL}}(p(X,y) \parallel p(X)p(Y)) = H(X) - \sum_{y=1}^{k} p(Y{=}y)H(X \mid Y{=}y) \quad (1)$$

where

- $D_{\mathrm{KL}} = \int_{\mathcal{X}} p(X) \ln(\frac{p(X)}{q(X)})dX$ is the Kullback-Leibler divergence,
- $H(X) = \int_{\mathcal{X}} p(X) \ln(p(X))dX$ is the differential entropy, and
- $H(X \mid Y{=}y) = \int_{\mathcal{X}} p(X \mid Y{=}y) \ln(p(X \mid Y{=}y))dX$ is the conditional differential entropy.

Expressing the mutual information in terms of the entropy is convenient, since the objective then decomposes over the values of Y. Additionally, $H(X)$ is independent of the distribution of Y and therefore does not influence the search over $\mathbf{y}$.

Because we are given only a finite sample from $p(X)$, there is no way to exactly compute $I(X,Y)$, and this is still true if we fix a set of cluster indicators y_i Possible ways to overcome this are:

1. Fit a parametric model $\hat{p}(X,Y \mid \theta)$ to the observations.
2. Use a non-parametric model $\hat{x}$ to approximate $p(X,Y)$.
3. Estimate $H(X \mid Y)$ directly using a non-parametric estimate.

We choose the third option, as it is the most flexible while avoiding the curse of dimensionality that comes with using non-parametric density estimates.

Let $\mathbf{x}_y$ be the set of x_i with label y. Given a non-parametric density estimator H_{est} we have $H_{est}(\mathbf{x}_y) \approx H(X \mid Y{=}y)$, leading to the clustering problem

$$\max_{\mathbf{y}} \quad -\sum_{y=1}^{k} p(Y{=}y)H_{\mathrm{est}}(\mathbf{x}_y), \quad (2)$$

where the probability $p(Y{=}y)$ is given by the empirical frequency of y, $p(Y=y) = \frac{n_y}{n}$ for $n_y = \frac{|\{i|y_i=y\}|}{n}$.

3.1 Minimum Spanning Tree Based Entropy Estimation

From now on, we assume that $\mathcal{X} = \mathbb{R}^d$ and $p(X)$ is absolute continuous. This setting allows the use of the non-parametric entropy estimate of Hero III and Michel [10], that constructs a minimum spanning tree of the data and obtains an estimate of the data entropy from the logarithm of the length of the spanning tree. More precisely, the entropy estimate of a dataset $\mathbf{x} = (x_1, \ldots, x_n)$ is given by

$$H_{mst}(\mathbf{x}) = d\log(L) - (d-1)\log(n) + \log(\beta_d). \quad (3)$$

where L is the length of a minimum spanning tree $T(\mathbf{x})$ of $\mathbf{x}$ and β_d is an unknown, but data-independent constant. The estimator H_{mst} is consistent in the sense that $H_{mst}(\mathbf{x}) \to H(X)$ for $n \to \infty$ [10]. Using Equation (3) as a non-parametric entropy estimate in Equation (2) yields the problem to maximize $\hat{I}(\mathbf{x}, \mathbf{y})$ with

$$\hat{I}(\mathbf{x}, \mathbf{y}) := -\sum_{y=0}^{k} p(y)\Big[d\log(L_y) - (d-1)\log n_y\Big] + C, \tag{4}$$

$$= -\sum_{y=0}^{k} p(y)\Big[d\log(\bar{L}_y) + \log n_y\Big] + C' \tag{5}$$

$$= -d\sum_{y=0}^{k} p(y)\log(\bar{L}_y) - \sum_{y=0}^{k} p(y)\log p(y) + C'' \tag{6}$$

where n_y is the cardinality of $\mathbf{x}_y$, L_y is the length of the minimum spanning tree $T(\mathbf{x}_y)$ and C, C' and C'' are constants independent of $\mathbf{y}$. We defined $\bar{L}_y := \frac{L_y}{n_y}$, the mean edge length per node in $T(\mathbf{x}_y)$.

Equation (6) has a natural interpretation: The first term penalizes long spanning trees, weighted by the size of the cluster. The second term favors a high entropy of $p(y)$, leading to balanced clusters. Note that there is a natural trade-off between enforcing intra-cluster similarity, expressed through L and the balancing of cluster sizes. This trade-off is similar to formulating an objective in terms of a loss and a regularizer. In contrast to the "loss+regularizer" setup, where the trade-off needs to be specified by the user, the trade-off in Equation (6), given by the factor d, is a direct consequence of the entropy estimator.

The reliance on the dimensionality of the ambient space $\mathbb{R}^d$ can be seen as the requirement that d is actually the intrinsic dimensionality of the data. This requirement is made explicit in our assumptions of an absolute continuous data density: If the support of $p(X)$ was a lower-dimensional sub-manifold of $\mathbb{R}^d$, $p(X)$ could not be absolute continuous.

3.2 Finding Euclidean Minimum Spanning Tree Clusterings

The objective given by Equation (4) is a non-linear combinatorial optimization problem. It has two properties that make it hard to optimize:

1. The objective depends in a non-linear way on L_y. This makes linear programming techniques, that proved successful for other combinatorial task, not directly applicable.
2. L_y is defined in terms of minimum spanning trees. This set is hard to characterize, as changing the cluster membership of a single node may change the two minimum spanning trees involved completely.

For the above reasons, we propose a simple procedure to approximately solve Equation (4). Consider a graph G with nodes $\mathbf{x}$ and edge weights given by the Euclidean distances between points. The connected components of G induce a

Algorithm 1 Information Theoretic MST-based Clustering

Input: Points $\mathbf{x}$, desired number of clusters k.
Output: Clustering $\mathbf{y}$ of $\mathbf{x}$

$G \leftarrow T(\mathbf{x})$
for $i = 0, \ldots, k-1$ **do**
  for $G_j, j = 0, \ldots, i$ connected components of G **do**
    $e_j \leftarrow \text{SplitCluster}(G_j)$
  $l \leftarrow \arg\max_j \hat{I}(G_j \setminus e_j)$
  $G \leftarrow G \setminus e_l$

function SplitCluster(G)
  Pick arbitrary root x_0 of G.
  for node x starting from leaves **do**
    $w_x \leftarrow \sum_{c \in \text{children}(x)} w_c + d(x, c)$
    $n_x \leftarrow 1 + \sum_{c \in \text{children}(x)} n_c$
  for node x starting from root **do**
    $w'_x \leftarrow w'_{\text{par}(x)} + w_{\text{par}(x)} - w_x - d(x, \text{par}(x))$
  for $e \in E(G), e = (c, p)$, p parent of c **do**
    $v_c \leftarrow w'_p + w_p - w_c - d(p, c)$
    $m_c \leftarrow n - n_c$
    $\text{objective}(e) \leftarrow dm_c \ln(m_c) - (d-1)m_c \ln(v_c) + dn_c \ln(n_c) - (d-1)n_c \ln(w_c)$
  $e^* \leftarrow \arg\max_{e \in E(G)} \text{objective}(e)$

clustering $\mathbf{y}(G)$ of $\mathbf{x}$, by assigning x_i and x_j the same cluster if and only if they are in the same connected component of G. Define

$$\hat{I}(G) := -\sum_{y=0}^{k} p(y) \Big[d \log(L_{G,y}) - (d-1) \log n_y \Big], \tag{7}$$

where y enumerates the connected components $G_0, \ldots, G_k$ of G, $n_y = |V(G_y)|$ is the number of nodes in G_y and $L_{G,y} = \sum_{e \in E(G_y)} w(e)$ is the sum of the weights of all edges in the connected component G_y. Then $\hat{I}(G) \geq \hat{I}(\mathbf{x}, \mathbf{y}(G))$, by the definition of the minimum spanning tree, and equality holds if and only if G_y is the minimum spanning tree of its nodes for all y. We try to find a graph G with k components, such that $\hat{I}(G)$ is maximal. We can restrict ourself to optimizing over the set $\mathcal{F}$ of forests over $\mathbf{x}$ with k components, as adding edges inside connected components will only decrease the objective. Thus we can formulate the clustering problem equivalently as $\max_{G \in \mathcal{F}} \hat{I}(G)$.

Optimization over forests remains hard, and we further restrict ourself to solutions from $\mathcal{G} := \{F \in \mathcal{F} \mid F \text{ subgraph of } T(\mathbf{x})\}$ for a given minimum spanning tree $T(\mathbf{x})$, leading to the problem $\max_{G \in \mathcal{G}} \hat{I}(G)$. This restriction allows for a very fast, combinatorial optimization procedure.

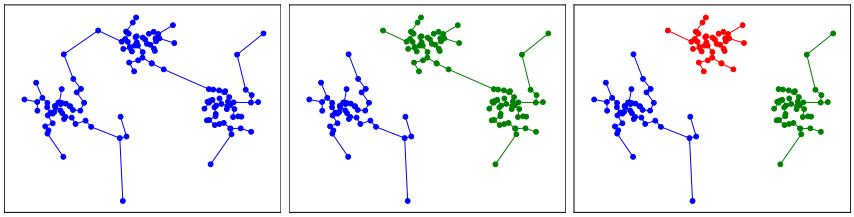

Fig. 1. Illustration of the optimization algorithm for $k = 3$ on synthetic dataset. *Left*: Euclidean minimum spanning tree of the data. *Center*: The edge that yields the best two-cluster partition in terms of Equation (4) was removed, yielding two connected components. *Right*: Another edge from the forest was removed, resulting in the desired number of three components. Note that the edge that are removed are not the longest edges but form a trade-off between edge length and cluster size.

For the two class case, optimization of the above objective can be solved exactly and efficiently by searching over all of $\mathcal{G}$. This amounts to searching for the edge e that maximizes $\hat{I}(T(\mathbf{x}) \setminus e)$. The naive algorithm that computes the objective for each edge separately has run time that is quadratic in the number of data points. To improve upon this, we use a dynamic programming approach as described in Algorithm 1, in function SplitCluster, which has only linear complexity. Using this algorithm, run time in the two cluster case is dominated by computing $T(\mathbf{x})$. We extend this algorithm to the case of more than two clusters in a greedy way: Starting with the full spanning tree of $\mathbf{x}$, we remove the edge yielding the lowest value of Equation (7) until the number of components equals the number of desired clusters. The overall procedure is summarized in Algorithm 1, an illustration can be found in Figure 1. We refer to Algorithm 1 as *Information Theoretic MST-based (ITM) clustering*.

We use the dual-tree Boruvka algorithm [15] to compute the minimum spanning tree, which has runtime close to $O(n \log(n)\alpha(n))$. Here α is the inverse of the Ackerman function, which grows so slowly as to be considered constant in practice. The dynamic programming solution of Algorithm 1 has a run time of $O(n)$ per removed edge, leading to an overall run time of $O(n \log(n)\alpha(n) + nk)$. The $O(nk)$ comes from a worst case scenario, in which each step in the hierarchical clustering procedure only splits off a constant number of points. In a more realistic setting, we expect that the individual clusters are much smaller than the original dataset. In this case, the $O(nk)$ factor would improve to $O(n \log(k))$.

4 Experiments

We compared ITM to the popular k-Means algorithm [14, 13], to the MeanNN algorithm of Faivishevsky and Goldberger [5] and to single-link agglomerative clustering [8]. The similarities between single-link agglomerative clustering and the proposed MST-based optimization make it a good baseline for tree-based clustering approaches.

A comparison of ITM, MeanNN and the baseline methods, k-Means and single link agglomerative clustering, in terms of their objective, optimization

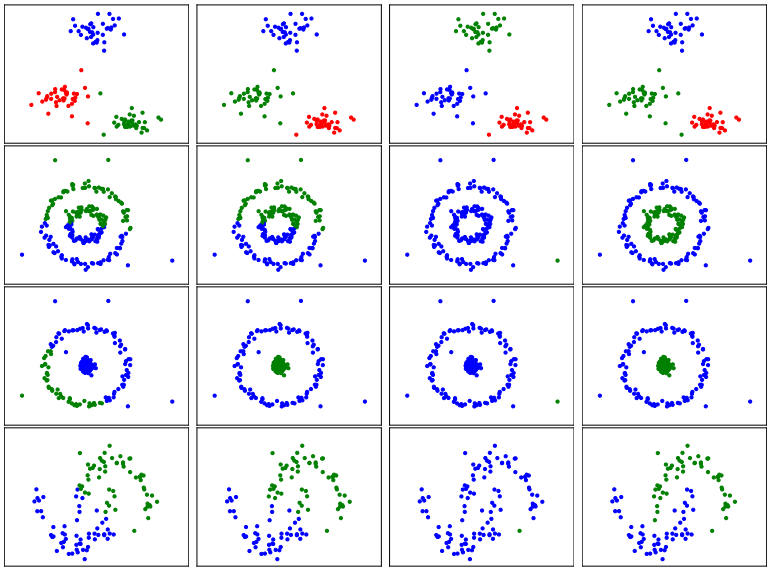

Fig. 2. Comparison of k-Means (left), MeanNN (center left), single link (center right) and ITM (right) on four synthetic datasets. Without the need to tune parameters, ITM can adjust to different cluster shapes. MeanNN is able to recover non-convex clusters (third row) but often produces similar results to k-Means (second and last row). Single link clustering is very sensitive to noise, as it does not take cluster size into account.

and complexity can be found in Table 1. We implemented the ITM clustering procedure as well as MeanNN in Python. We used the k-Means implementation available in the scikit-learn library [17]. We use the dual tree Boruvka algorithm implemented in the mlpack machine learning library [3]. The source code is available online[†].

4.1 Experimental Setup

For both k-Means and MeanNN, we restart the algorithm ten times using different random initializations, keeping the result with the best objective value. As ITM is deterministic there is no need for random restarts. All of the algorithms we compare work with a fixed number of clusters, which we set to the number of classes in the dataset for all experiments.

As single link agglomerative clustering is sensitive to outliers, we set a hard limit on the minimum number of samples per cluster of five for the quantitative analysis.

4.2 Qualitative Results

Figure 2 shows qualitative results on three synthetic datasets. For well separated, convex clusters, all four algorithms produce the same clustering (see top row). If the structure of the data is more complex, the advantage of the proposed method is apparent. Note that there was no need to specify any other parameters than

[†] https://github.com/amueller/information-theoretic-mst

Table 1. Comparing properties of related algorithms.

Algorithm	Objective	Deterministic	Complexity
k-Means	$\sum_y \sum_{i,y_i=y} \|x_i - \mu_y\|^2$	No	$O(nk)$ per iteration
MeanNN	$\sum_y \log\left(\frac{1}{\|\mathbf{x}_y\|} \sum_{i,j,y_i=y_j=y} \|x_i - x_j\|^2\right)$	No	$O(n^2)$ per iteration
Single Link	–	Yes	$O(n \log n)$
ITM	$\sum_{y=0}^{k} dp(y) \log(\bar{L}_y) + p(y) \log p(y)$	Yes	$O(\alpha(n) n \log n + nk)$

the number of clusters to produce these results. It is also noteworthy that the results of MeanNN are very close to those produces by k-Means in most cases. This similarity can be explained by the close relation of the objective functions, listed in Table 1.

4.3 Quantitative Results

We present results on several standard datasets from the UCI repository, selecting datasets that span a wide range of combinations of number of samples, features and clusters. To satisfy the assumption of absolute continuity of the data distribution, we restrict ourself to data with continuous features.

We evaluated the experiments using the *adjusted Rand index (ARI* [11] and *normalized mutual information (NMI)* [22], two popular measures of cluster quality [7, 12]. The Rand index [19] between two clusterings counts on how many pairs of points two clusterings agree. The adjusted Rand index contains a calibration against chance performance.

Table 2 summarizes the results. The two entropy-based methods (MeanNN, ITM) have a clear advantage of the other methods, with ITM finding better clusterings than MeanNN in the majority of cases. The single link agglomerative clustering procedure produces reasonable results on datasets with little noise and well-separated clusters, but fails otherwise. When inspecting the results, we observed that ITM produced several very small clusters on the *faces* dataset. Indeed, increasing the minimum cluster size to 6 or more improved the results to 0.59/0.84. A possible explanation for this is that very small clusters make the entropy estimate less reliable. The single-link method also benefited from this, improving its results to 0.42/0.82. The run time of computing the ITM clustering was dominated by the computation of the MST of the data. The implementation in mlpack took 60 seconds on a desktop computer for *usps*, the largest dataset in our experiments. The other methods had run times in the order of seconds, but given the different implementations we used, this should not be interpreted as a general statement about the speed of the individual methods.

5 Conclusions

In this work we proposed the use of a minimum spanning tree based, non-parametric entropy estimator in information theoretic clustering, ITM. Thereby

Table 2. Scores (ARI/NMI) of k-Means, MeanNN, single link agglomerative clustering and ITM on several benchmark datasets (higher is better). The best score for each dataset is printed in bold.

Dataset				Results			
Description	n	d	k	k-Means	MeanNN	SL	ITM
digits	1797	64	10	0.62 / 0.71	0.67 / 0.76	0.10 / 0.50	**0.85** / **0.89**
faces	400	4096	40	0.41 / 0.76	**0.49** / **0.80**	0.08 / 0.69	0.02 / 0.49
iris	150	4	3	0.72 / 0.76	0.75 / 0.78	0.55 / 0.72	**0.88** / **0.87**
usps	9298	256	10	0.52 / 0.61	**0.54** / **0.65**	0.00 / 0.04	0.44 / 0.58
vehicle	846	18	4	0.10 / 0.15	0.09 / 0.11	0.00 / 0.04	**0.10** / **0.14**
vowel	990	10	11	0.17 / 0.37	0.19 / **0.40**	0.00 / 0.16	**0.20** / 0.39
waveform	5000	21	2	**0.37** / 0.35	0.30 / **0.38**	0.00 / 0.00	0.23 / 0.22

we extended the work of Faivishevsky and Goldberger [5] to a more flexible and efficient entropy estimate. We proposed an approximate optimization method by formulating the clustering problem as a search over graphs. The resulting algorithm is deterministic has sub-quadratic run time. Empirical comparisons showed that the proposed method outperforms standard algorithms and the non-parametric entropy based clustering of [5] on multiple benchmark datasets. We demonstrated that ITM is able to detect non-convex clusters, even in the presence of noise. In contrast to other algorithms that can handle non-convex clusters, ITM has no tuning parameters, as the objective presents a natural trade-off between balancing cluster sizes and enforcing intra-cluster similarity.

A limitation of the proposed algorithm is that it is based on the assumption of an absolute continuous data distribution. This assumption eliminates the possibility of using categorical variables and data that lies on a submanifold of the input space. In future work we plan to investigate a way to overcome this limitation, for example by estimating the intrinsic dimensionality of the data [18]. We will also investigate optimizations of the objective Equation (7) that go beyond the proposed method. Move-making algorithms seem a promising way to refine solutions found by Algorithm 1. Branch and bound techniques could provide an alternative approach.

References

[1] Banerjee, A., Merugu, S., Dhillon, I., Ghosh, J.: Clustering with Bregman divergences. Journal of Machine Learning Research 6 (2005)

[2] Barber, F.: Kernelized infomax clustering. In: Neural Information Processing Systems (2006)

[3] Curtin, R.R., Cline, J.R., Slagle, N.P., Amidon, M.L., Gray, A.G.: MLPACK: A scalable C++ machine learning library. In: BigLearning: Algorithms, Systems, and Tools for Learning at Scale (2011)

[4] Dhillon, I., Mallela, S., Kumar, R.: A divisive information theoretic feature clustering algorithm for text classification. Journal of Machine Learning Research 3 (2003)

[5] Faivishevsky, L., Goldberger, J.: A nonparametric information theoretic clustering algorithm. In: International Conference on Machine Learning (2010)
[6] Gokcay, E., Principe, J.: Information theoretic clustering. Pattern Analysis and Machine Intelligence 24 (2002)
[7] Gomes, R., Krause, A., Perona, P.: Discriminative clustering by regularized information maximization. In: Neural Information Processing Systems (2010)
[8] Gower, J., Ross, G.: Minimum spanning trees and single linkage cluster analysis. Applied Statistics (1969)
[9] Grygorash, O., Zhou, Y., Jorgensen, Z.: Minimum spanning tree based clustering algorithms. In: International Converence on Tools with Artificial Intelligence (2006)
[10] Hero III, A., Michel, O.: Asymptotic theory of greedy approximations to minimal k-point random graphs. Information Theory 45 (1999)
[11] Hubert, L., Arabie, P.: Comparing partitions. Journal of Classification 2 (1985)
[12] Kamvar, K., Sepandar, S., Klein, K., Dan, D., Manning, M., Christopher, C.: Spectral learning. In: International Joint Conference of Artificial Intelligence (2003)
[13] Lloyd, S.: Least squares quantization in PCM. Information Theory 28 (1982)
[14] MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Berkeley Symposium on Mathematical Statistics and Probability (1967)
[15] March, W.B., Ram, P., and Gray, A.G.: Fast Euclidean minimum spanning tree: algorithm, analysis, applications. In: International Conference on Knowledge Discovery and Data Mining (2010)
[16] Ng, A., Jordan, M., Weiss, Y.: On spectral clustering: analysis and an algorithm. In: Neural Information Processing Systems (2002)
[17] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. Journal of Machine Learning Research 12 (2011)
[18] Pettis, K., Bailey, T., Jain, A., Dubes, R.: An intrinsic dimensionality estimator from near-neighbor information. Pattern Analysis and Machine Intelligence 1 (1979)
[19] Rand, W.: Objective criteria for the evaluation of clustering methods. Journal of the American Statistical association (1971)
[20] Shi, J., Malik, J.: Normalized cuts and image segmentation. Pattern Analysis and Machine Intelligence 22 (2000)
[21] Slonim, N., Tishby, N.: Agglomerative information bottleneck. Neural Information Processing Systems (1999)
[22] Strehl, A., Ghosh, J.: Cluster ensembles—a knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research 3 (2003)
[23] Zahn, C.: Graph-theoretical methods for detecting and describing gestalt clusters. IEEE Transactions on Computers 100 (1971)