Signal Processing 92 (2012) 1970–1984

Contents lists available at SciVerse ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Graph based k-means clustering

Laurent Galluccio [a,c], Olivier Michel [b], Pierre Comon [a,*], Alfred O. Hero III [d]

[a] I3S, UMR6070 CNRS, University of Nice-Sophia Antipolis, 2000 route des Lucioles, 06903 Sophia Antipolis Cedex, France
[b] Gipsa-Lab UMR 5216, 961 rue de la Houille Blanche, BP 46, 38402 Saint Martin d'Heres Cedex, France
[c] Laboratoire Cassiopée UMR 6202, University of Nice Sophia Antipolis, CNRS, Nice Cote d'Azur Observatory, Boulevard de l'Observatoire, B.P. 4229, 06304 Nice Cedex 4, France
[d] Department of Electrical Engineering and Computer Science, University of Michigan, 1301 Beal Avenue, Ann Arbor, MI 48109-2122, USA

ARTICLE INFO

Article history:
Received 23 November 2010
Received in revised form
6 December 2011
Accepted 9 December 2011
Available online 21 January 2012

Keywords:
Data partitioning
Unsupervised classification
Graph-theoretic methods
Minimal spanning trees
Similarity measures
Information theoretic measures
Multi-spectral imaging

ABSTRACT

An original approach to cluster multi-component data sets is proposed that includes an estimation of the number of clusters. Using Prim's algorithm to construct a minimal spanning tree (MST) we show that, under the assumption that the vertices are approximately distributed according to a spatial homogeneous Poisson process, the number of clusters can be accurately estimated by thresholding the sequence of edge lengths added to the MST by Prim's algorithm. This sequence, called the Prim trajectory, contains sufficient information to determine both the number of clusters and the approximate locations of the cluster centroids. The estimated number of clusters and cluster centroids are used to initialize the generalized Lloyd algorithm, also known as k-means, which circumvents its well known initialization problems. We evaluate the false positive rate of our cluster detection algorithm, using Poisson approximations in Euclidean spaces. Applications of this method in the multi/hyper-spectral imagery domain to a satellite view of Paris and to an image of Mars are also presented.

1. Introduction

1.1. Motivation and contribution

One of the recurrent problems in pattern recognition, machine learning or data mining is data clustering [1–3]. This is also a major subject of research in the remote sensing community with the emergence of hyper-spectral sensors, which generate a significant amount of data. Clustering consists of partitioning a data set into groups of points that have high similarity. Similarity in this context means that data belonging to a cluster have similar features.

In this paper a new method is introduced for estimating the number of clusters present and determining good centroid locations to initialize the k-means algorithm [4], which is widely known to be extremely sensitive to centroid initialization. The presented method relies on properties of the minimal spanning tree (MST) as grown by Prim's algorithm [5], and in particular the Prim trajectory defined as the rate of growth of the MST as Prim's algorithm successively adds edges. The Prim trajectory describes the evolution of the cost of adding edges to the MST as a function of the algorithm iteration index. Data are L dimensional feature vectors sampled from an unknown multimodal distribution. It is assumed that distinct groups or clusters manifest themselves as modes in the density. We show that Prim trajectory is as a *one-dimensional* unfolded representation of the underlying data probability density function, exhibiting peaks and valleys. Thus, we propose to estimate the number of clusters by thresholding Prim trajectory, in which sharply

* Corresponding author. Tel.: +33 4 92942717; fax: +33 92942896.
E-mail addresses: laurent.galluccio@oca.eu (L. Galluccio), olivier.michel@gipsa-lab.inpg.fr (O. Michel), pcomon@i3s.unice.fr (P. Comon).
URL: http://www.i3s.unice.fr/~pcomon (P. Comon).

0165-1684/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2011.12.009

peaked modes are associated with deep valleys [6]. A similar Prim-trajectory clustering approach was used for a bioinformatics application in Olman et al. [7].

The proposed method can be viewed as an algorithm for initializing an iterative partitioning algorithm like k-means, which is the focus of this paper. k-means is one of the simplest and most popular approaches for solving clustering problems despite its severe sensitivity to initialization. While k-means appears as a final step in the proposed algorithm, other partitioning algorithms could be used. Alternative approaches can be used to identify the number of clusters. Jordan and Ng or Shi and Malik's spectral clustering methods [8–10] estimate the number of clusters by thresholding the eigen-spectrum of the graph Laplacian [11]. This however remains a difficult task in the presence of outliers, as ordered eigenvalues usually do not exhibit an obvious threshold. As our method involves only those vertices in the graph that are close to the most concentrated regions (modal peaks) of the data this problem is circumvented.

The problem of setting the threshold is tackled by adopting a hypothesis testing framework. Under the null (no clusters) hypothesis we model the data as a realization of a spatial homogeneous Poisson spatial point process over feature space. A Neyman–Pearson hypothesis test procedure is then used to detect the modes.

The construction of the Prim trajectory generally requires the computation of a distance matrix between the data points, which may be impractical for large amounts of data. We propose an algorithm for calculating Prim's trajectory that avoids the difficult computation of huge similarity matrices. The algorithm is based on a data-driven hierarchical classification. Our Prim-based algorithm can be applied to a variety of distance metrics or similarity measures in feature space. Often a non-symmetric information theoretic similarity measure can be used to obtain improved clustering results. We show that the use of such information measures can significantly improve clustering performance on real-world data imaging sets.

1.2. Context and brief review of previous works

The general data clustering problem has long been studied and many approaches have been proposed (see [1–3] for review). One can divide the existing methods into two classes: hierarchical and partitional clustering algorithms. Hierarchical algorithms evolve the number of clusters, e.g., as in agglomerative or divisive clustering, while partitional algorithms fix the number of clusters, e.g., as in modal or k-means clustering.

Most hierarchical clustering algorithms are based on popular single-link or complete-link algorithms. These methods often suffer from prohibitive computational time due to the need to construct a dendrogram on a large data sets. Their stability in the presence of outliers and their sensitivity to the applied dendogram thresholds are problematic. One of the most popular partitional clustering algorithm is the generalized Lloyd algorithm used to implement k-means [4], which will be the focus of this paper. The present work is developed in the context of unsupervised clustering. Neither the underlying distributions nor the number and locations of meaningful clusters are known. Hence no prior information is required.

In k-means clustering k points are randomly chosen as the initial cluster centers. The choice of the number k of clusters is usually based on some heuristic. Each data point is assigned to the group that has the closest center. The cluster centers of mass are then recomputed. The assignment and re-computation steps are iterated until the intra-cluster variance converges to a minimum (see Section 2.1). This algorithm, as well as the classical ISODATA algorithm [12], belongs to the class of squared error algorithms which aim at minimizing some objective function in order to cluster the data.

The k-means algorithm clusters the data into non-overlapping convex groups and it always converges (although not necessarily to the global optimum). As with any partitional clustering algorithm, it is highly sensitive to the initial parameters: the number k of clusters and their centroids, respectively. In practice, to obtain the best clustering results, the k-means algorithm is often applied many times to different random initializations and the minimizer of the intra-cluster variance is selected as the final clustering result. However, this is at the cost of increased computation. The main focus of this paper is automatic initialization of k-means by estimating both the number of clusters present in the data set and the positions of initial cluster centroids. Initialization is crucial since improper initialization could lead the algorithm to converge to some undesirable local minimum of the objective function. Initialization of the k-means algorithm has been extensively studied during the last decade (see [13,14] for a short comparative study of initialization methods for k-means). This paper proposes an improved method to initialize the k-means algorithm.

The issue of estimating the right number of clusters has been widely addressed. One of the most widespread approaches is to use statistical model selection criteria such as Akaike information criterion (AIC), Bayesian information criterion (BIC), minimum description length (MDL) [15], Tibshirani's gap [16], Hartigan's index [17] or Krzanowski and Lai's index [18]. Milligan and Cooper [19] have studied and compared 30 different model selection criteria and they concluded that Calinski and Harabasz's index [20] was the best. Most of the aforementioned techniques are based on a clustering step with k-means, followed by a merge or split procedure to optimize these criteria. We briefly review some of these methods below.

In the vector quantization community, Bischof et al. [21] proposed an algorithmic complexity called MDL, to select the number of clusters (starting with a large value of k). Pelleg and Moore [22] developed a variant of k-means referred to as X-means, starting from a minimum number of clusters followed by application of k-means. Their agglomerative cluster splitting procedure is based on the BIC criterion. Hamerly and Elkan [23] proposed to learn the integer k required in k-means by assuming a Gaussian mixture model. Starting with a small value of k, the algorithm splits the clusters that fail a test of spherical Gaussianity. Between each statistical test, the k-means algorithm is applied to refine the solution. Nevertheless,

this algorithm referred to as *G*-means, performs poorly in the presence of non-spherical or non-elliptical clusters. Tibshirani et al. [16] defined the Gap statistic to determine the optimum number of clusters. Their method is based on the output of one of the two clustering algorithms: k-means or hierarchical classification. The authors propose to compare the logarithm of the within-cluster dispersion to its expectation under a reference null-distribution. This method performs better if clusters are well separated and if the number of clusters is small.

The location of the initial cluster centroids has serious impact on the performance of partitional clustering algorithms. Forgy's approach [24] consists in choosing initial centers by picking them at random. This method often converges to local minima. A refinement was recently proposed by Arthur and Vassilvitskii [25] called k-means ++. The first initial center is randomly chosen among the data points. The next seed is selected among the remaining data points with a certain probability, depending on the distance between data points and their closest center. This operation is repeated until k centers are chosen.

Katsavounidis et al.'s algorithm [26] starts by initializing the first cluster as the point with the maximal norm. Then, the distance of all data points from the first cluster center is calculated. The point with the largest distance is chosen as the second cluster center. After that, for every data point, the distance with its closest center is computed; and the point with the largest distance is chosen as the next cluster center. This process is repeated until k centers are chosen. Unfortunately, this method is sensitive to outliers. Kaufman and Rousseeuw's method [27] selects the first center as the most centrally located point. Then, a heuristic rule described in [27] is used to select other cluster centers until k centers are chosen.

A good way to properly initialize the k-means algorithm is to place every cluster centroid at the modes of the joint probability density of the data. Motivated by this idea, Bradley and Fayyad [28] propose a refinement of initial conditions of k-means near the modes of the estimated distribution, by the use of a recursive procedure executing k-means on small random sub-samples of the data. Global k-means was introduced by Likas et al. in [29], and is also motivated by the same idea that modes play an important role. Their method is based on a recursive partitioning of the data space into disjoint subspaces by using $k-d$ trees. They then define a cutting hyperplane as a linear space perpendicular to the highest variance axis. This is similar to the data partitioning that occurs in adaptive vector quantization and regression trees [30].

Our method is motivated similar to the Bradley and Fayyad's motivation of their recursive algorithm [28]: the modes of the underlying unknown multivariate density are the most pertinent features for initializing a partitional algorithm. The originality of our approach resides in (1) the use of Prim trajectory to map the multivariate density to a singly indexed 1D space, and (2) the formulation of mode detection as a binary hypothesis testing problem (Section 2). This allows us to specify a fully unsupervised procedure, which exhibits both reliability and robustness in the presence of noise and outliers. An implementation of our approach to large data sets, relying on the use of a pre-conditioning data driven classification tree, is described in Section 3. Finally, in the last section, experimental results are presented, and alternative information theoretic similarity measures are introduced. Information theoretic similarity measures turn out to be better adapted to feature vectors that behave like spectra, i.e., non-negative valued and normalized. Our new Prim-based clustering approach is tested in this context.

2. Automatic detection of the relevant number of clusters

In this section, the definition of MST and Prim's algorithm are briefly reviewed. Then an original approach is proposed for initializing k-means by jointly estimating the number of relevant clusters and the locations of their corresponding centroids.

2.1. Formulation

Let V be a set of N data points or feature vectors in $\mathbb{R}^L$. In our unsupervised framework, it is assumed that the points are sampled from an unknown density $P(\mathbf{v})$ and that distinct groups manifest themselves as distinct modes of the density $P(\mathbf{v})$.

Each data point (actually a vector in $\mathbb{R}^L$) is interpreted as a vertex in a graph so that V is the graph vertex set. The goal is to partition V into k classes. Denote $C:(C_1,\ldots,C_k)$ the set of clusters, and let $M:(\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_k)$ be the set of corresponding centroids. Let us emphasize that no prior information is introduced: in the context of unsupervised clustering, neither the underlying distributions nor the number and locations of meaningful clusters are known.

The k-means algorithm aims at finding a partition of the data set such as the following error function (within-cluster summary distance to centroids) is minimized:

$$J(C,\boldsymbol{M})=\sum_{j=1}^{k}\sum_{\mathbf{v}\in C_j}\|\mathbf{v}-\boldsymbol{\mu}_j\|^2 \tag{1}$$

This minimization is well known in the pattern classification and vector quantization literature [31,32]. In this paper the classical Lloyd algorithm is used which is summarized in the final loop of Algorithm 1 given in the sequel.

2.2. Minimum spanning tree and prim's algorithm

Let T be an undirected acyclic graph (or tree) connecting all vertices in V. The graph is specified by its set of vertices and its set of edges $E=(e(\mathbf{v}_i,\mathbf{v}_j),(i,j)\in\{1,\ldots,N\})$. The length or weight $|e(\mathbf{v}_i,\mathbf{v}_j)|$ of an edge measures a distance or dissimilarity between two vertices $(\mathbf{v}_i,\mathbf{v}_j)$.

The total length of the tree T is the sum of all edge lengths: $\mathcal{L}(T)=\sum_{e\in T}|e|$. The minimal spanning tree T^* is the one having minimal length among all spanning trees:

$$\mathcal{L}(T^*)=\min_{T}\sum_{e\in T}|e| \tag{2}$$

There exist many algorithms for constructing the MST (see [33] for state of art of MST's algorithms). This paper focuses on Prim's algorithm [5] as some of its features will be used for estimating the presence of relevant clusters in the data set. Prim's algorithm is briefly described below.

Let T_i be the partially connected graph at iteration $i-1$, hence i vertices are connected to the MST. At the ith iteration, one non-connected vertex, say $\mathbf{v}_i$, and one connected vertex of T_i are selected, so that the dissimilarity measure between them is minimal. T_i becomes T_{i+1} with this new vertex $\mathbf{v}_i$ and the associated edge of minimal length. This operation is repeated until no unconnected vertex remains; $N-1$ such iterations are required to construct the complete MST from a set of N points. A consequence is that any vertex is connected to its nearest neighbor, and that any tree fragment is connected to its nearest neighbor by the shortest possible link.

A MST is unique if there are no ties in the pairwise distances. The choice of the initial vertex has no consequence on the resulting MST: the same set of edges will be obtained, although they will be constructed at different iteration indexes. The construction of the MST associated with a vertex set V only requires the knowledge of the dissimilarities between the pairs of vertices. These will be assumed to be symmetric quantities in the sequel, in agreement with the fact that only undirected graphs will be considered. The importance of the choice of the metric provided to the MST computing routine is discussed in Section 4.2.

2.3. Exploiting the MST and the Prim's trajectory

The use of the MST for describing multivariate data in $\mathbb{R}^L$ in order to exhibit clusters is not a new idea [3]. Existing methods share a common feature: they are based on graph cuts. Single-cut clustering [34] is easily performed using the MST, but it is known to be highly unstable in the presence of noise and/or outliers in the data set. Slagle et al. [35] proposed to build a one dimensional function of the length of edges connecting vertices within a short subgraph, to detect the largest ones, subsequently applying graph cuts to form clusters. Stuetzle [6] proposed to consider multiple cuts involving all edges larger than a threshold in order to preserve minimal cardinality (the "runt size") of the resulting clusters (subgraphs). Tuning the adequate "runt size" turns out to be difficult and critical for accurate clustering performance.

The Prim trajectory makes use of the MST construction scheme to propose candidate subsets of vertices that are representative of modes of the density $P(\mathbf{v})$. Instead of trying to detect edges to cut the Prim algorithm decides which vertices to keep, which we use to form the seeds of clusters. Like the method presented in [35] the Prim trajectory projects the data into a one dimensional space.

Prim trajectory and the modes of $P(\mathbf{v})$: We formally define the Prim trajectory here. Prim's algorithm starts with a root vertex and then constructs edges by a sequence of nearest neighbor operations. Let $g(i)=|e_i|$ be the length of a new edge built by Prim's algorithm at iteration i. The ordered set of edge lengths $[g(i), i=1\ldots N-1]$ is defined as *Prim trajectory*. The edge function g allows us to "unfold" the MST built in an L dimensional space into a one-dimensional function (see Fig. 1).

A rough estimate of the L-variate density $P(\mathbf{v})$ in the vicinity of $\mathbf{v}_i$ and $\mathbf{v}_j$ may be inferred from the weight of the edge connecting them: if $\mathbf{v}_i$ and $\mathbf{v}_j$ are connected by the MST then, for $k=1$, the 1-NN density estimate is given by $p(\mathbf{v}_i)\propto 1/\min_j|e(\mathbf{v}_i,\mathbf{v}_j)|^L$. This widely used density estimator motivates our approach: a series of consecutive connections with low weight (a valley in g) is the signature of a peak in the density estimate of $P(\mathbf{v})$. Then, if enough vertices are involved in the valley, a mode of $P(\mathbf{v})$ will be detected by thresholding g.

Discussion:

- While the obtained MST does not depend on the vertex onto which it is rooted, g changes with the choice of the root. However, Prim's algorithm aggregates new edges of minimal weight at each iteration and the vertices sampled from the same mode will always be connected within a subset of consecutive iterations. Fig. 1 illustrates this property. A MST is built from two different Prim's trajectories. Both trajectories exhibit deep valleys, associated to series of consecutive iterations connecting close neighbors from the same mode.
- The choice of the threshold to apply to $g(i)$ is crucial, since it defines the sensitivity of the mode detection to overlapping of clusters. As g is built from a finite size sample, a high threshold will lead to detection of spurious modes, whereas a low threshold may lead to missing some modes. The example in [36] illustrates this behavior; the threshold here was arbitrarily set to $\epsilon_t=\sigma(e_i)$, the standard deviation of the edge lengths. In Fig. 1(e) and (f), if a mode is associated to each group of successively connected points below the threshold, a lot of small spurious modes are introduced due to the high variability of the 1 NN density estimator for small N. This was addressed and an entropy based approach was discussed in [36] for the case of Euclidean distances, but the resultant algorithm has high computation cost. We formulate a systematic approach to threshold selection using a binary hypothesis testing framework in the next subsection.

2.4. Setting the threshold

Consider the following question. What is the probability that a subset of vertices appear as a close connected neighbors in the absence of mode, thus leading to the erroneous detection of a mode? The closeness will depend on the given threshold, and the erroneous detection will be a false alarm. If we specify a null model for $P(\mathbf{v})$ a Neyman–Pearson detection framework can be adopted. We specify the hypothesis testing procedure as follows. First, we propose a model for the spatial vertex distribution under the null hypothesis H_0 (no cluster). Second, we express the probability of finding a subset of k consecutive connections whose weights are below a given threshold ϵ_t. Third, we relate the probability of false alarm to the parameters N, ϵ_t and k.

Modeling the vertex distribution under H_0: Under H_0 we model the N vertices as a realization of a homogeneous Poisson process over the bounded domain $\mathcal{V}\subset\mathbb{R}^L$. This

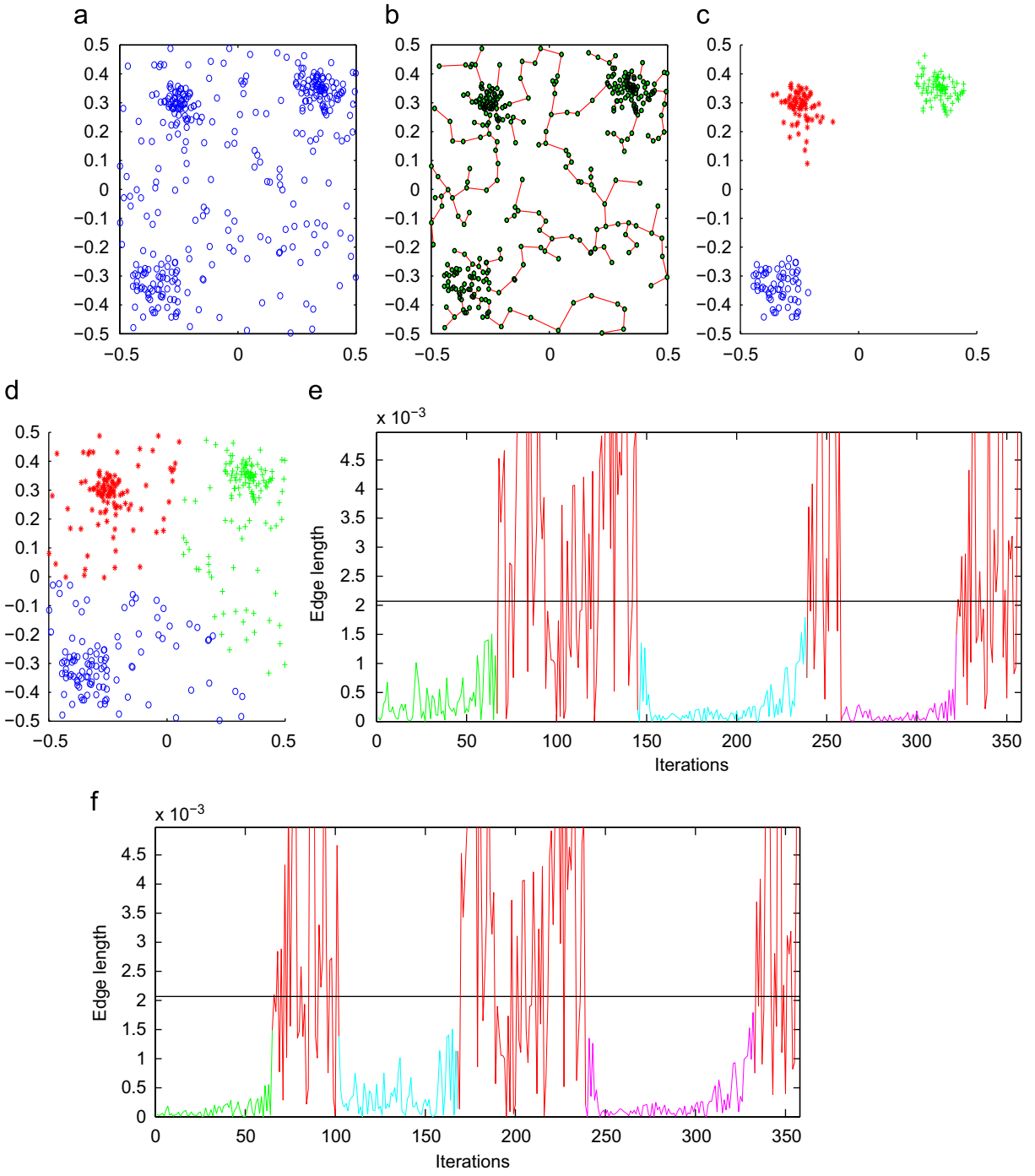

Fig. 1. (a) Toy data set, (b) construction of a MST, (c) extraction of the relevant groups of vertices present below the threshold in Prim's trajectory, (d) clustering with k-means, (e,f) Prim's trajectories with different initial vertices used in the MST construction.

implies that under H_0 and given N, the vertices are samples of a multi-dimensional independent and identically distributed (iid) uniformly distributed over $\mathcal{V}$. Note that modeling the distribution of vertices of a MST as a Poisson process was assumed by Steele in [37].

Let $\mathbf{v}_i$ be the vertex connected at iteration i in the Prim construction algorithm. Let $p_{N,\epsilon,i}$ be the probability under H_0 that another vertex is found within $B_{(\mathbf{v}_i,\epsilon)}$ of radius ϵ, centered on $\mathbf{v}_i$:

$$p_{N,\epsilon,i} = \int_{B_{(\mathbf{v}_i,\epsilon)}} f(r)\, dr$$

where $f(r) = 1/\mathrm{vol}(B(\mathbf{v}_i,\epsilon))$. Conditioned on the vertex $\mathbf{v}_i$, define k the number of vertices that fall in the ball $B_{(\mathbf{v}_i,\epsilon)}$, excluding vertex $\mathbf{v}_i$. As the vertices are assumed to be iid samples conditioned on N, k follows a binomial distribution

with parameters N and $p_{N,\epsilon,i}$. Therefore, the probability that k vertices are in $B_{(\mathbf{v}_i,\epsilon)}$ is given by a binomial probability law with parameters N, $p_{N,\epsilon,i}$, whose cumulative distribution function is

$$F_{N,\epsilon,\mathbf{v}_i}(k)=\sum_{j=0}^{k}\binom{N-1}{j}p_{N,\epsilon,i}^{j}(1-p_{N,\epsilon,i})^{N-1-j}$$

It is well known that the convergence of the binomial converges to the Poisson probability with rate parameter β as $N\to\infty$ and $\epsilon\to 0$, with $\beta=\lim_{N\to\infty}(Np_{N,\epsilon,i})$. Consequently, we obtain

$$F_{N,\epsilon,\mathbf{v}_i}(k)\overset{N\to\infty}{\longrightarrow}\mathrm{e}^{-\beta}\sum_{j=0}^{k}\frac{\beta^j}{\Gamma(j+1)}$$

Note that $1-F_{N,\epsilon,\mathbf{v}_i}(k)$ is simply an Erlang distribution with rate β.

Since $f(v)=1/|\mathcal{V}|$ where $|\mathcal{V}|$ stands for the volume of $\mathcal{V}$, the Poisson rate is

$$\beta\simeq\frac{NC_L}{|\mathcal{V}|}\epsilon^L=\lambda\epsilon^L$$

where C_L is the volume of the L-dimensional unit ball. Parameter $\lambda=NC_L/|\mathcal{V}|$ emphasizes the respective importance of the number of observations (N) and the volume of the support ($\mathcal{V}$).

Probability of detecting a mode under H_0: With the results above the probability that there exists at least one edge (or weight) of length smaller than ϵ is

$$F_{\mathbf{v}_i}(\epsilon)=1-\mathrm{e}^{-\beta} \tag{3}$$

Then the probability that under H_0, k consecutive connections of length smaller than ϵ and that the $(k+1)$ th has a larger length is given by

$$P_{k,\epsilon}=(1-F_{\mathbf{v}_i}(\epsilon))\prod_{i=1}^{k}F_{\mathbf{v}_i}(\epsilon)=(1-F_{\mathbf{v}_i}(\epsilon))(1-\mathrm{e}^{-\lambda\epsilon^L})^k \tag{4}$$

$$P_{k,\epsilon}=(1-\mathrm{e}^{-\lambda\epsilon^L})^k\mathrm{e}^{-\lambda\epsilon^L} \tag{5}$$

Note that the obtained expressions above satisfy the expected asymptotic equalities:

$$\lim_{\beta\to\infty}P_{k,\epsilon}=0\quad\text{for any } k<N \text{ value}$$

$$\lim_{\beta\to 0}P_{k\epsilon}=0\quad\text{for } k\neq 0$$

Furthermore, it is easily verified that

$$\sum_{k=0}^{\infty}P_{k,\epsilon}=\mathrm{e}^{-\beta}\sum_{k=0}^{\infty}(1-\mathrm{e}^{-\beta})^k=1$$

The expression relating β to ϵ needs to be reconsidered here, in the context of a MST construction with Prim's algorithm. Actually, one connects a new vertex to an existing growing subtree. In the limit of large N and large subtrees, this amounts to testing the presence of a vertex in a neighborhood which is no longer a sphere but a half sphere. This is illustrated in Fig. 2. Consequently the normalization constant C_L (volume of the unit sphere in the L-dimensional Euclidean space) will be replaced by $C_L/2$.

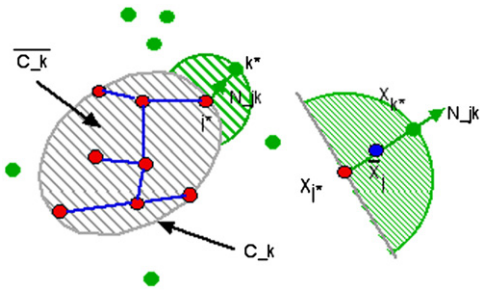

Fig. 2. Left: C_k denotes the contour of the support where connected vertices are found. The neighborhood which is considered for finding a vertex that could be connected to v_i is shown. In the limit of large N, this latter neighborhood is approximated as the half sphere laying on the tangent (hyper)plane to C_k (right).

The above approximations were derived under the assumptions of N and large Prim subtree size. We validate that this asymptotic approximation is accurate for finite sample size by comparing the simulated average number of vertices that are consecutively connected by edges of length smaller than a threshold ϵ. In the previous derivations, k denotes the number of vertices in a neighborhood centered on a vertex $\mathbf{v}$, but excluding $\mathbf{v}$; the number of edges connecting successive neighbors and the number of vertices involved in a cluster therefore differ by one vertex. As a consequence, the mean number of these vertices, denoted by $\langle k\rangle$ is obtained as the average of $(k+1)$ where k is the number of edges whose length are less than ε. As a cluster will be considered only if there is at least one such edge, we obtain

$$\langle k\rangle=\sum_{k=1}^{\infty}(k+1)P_{k,\epsilon}=\mathrm{e}^{-\lambda\epsilon^L}\sum_{k=1}^{\infty}k(1-\mathrm{e}^{-\lambda\epsilon^L})^k+1-\mathrm{e}^{-\lambda\epsilon^L}$$

$$=(1-\mathrm{e}^{-\lambda\epsilon^L})+\frac{(1-\mathrm{e}^{-\lambda\epsilon^L})}{\mathrm{e}^{-\lambda\epsilon^L}}=2\sinh(\lambda\epsilon^L)=2\sinh\left(\frac{C_L}{2}\frac{N}{|\mathcal{V}|}\epsilon^L\right) \tag{6}$$

Fig. 4 shows the average size of candidate modes detected as a function of the threshold value. All the detections are false alarms, as the vertices are drawn from a uniform distribution over $[0,1]^2$. This simulation suggests that there is good match between the asymptotic theory and the finite N simulations.

Threshold as a function of the probability of false alarm (PFA): A false alarm mode occurs whenever some edges of length smaller than ϵ are created under the null hypothesis that the vertices are realizations from a homogeneous Poisson process. Note that the null hypothesis then refers to the no-cluster hypothesis. To avoid these false alarms, a simple strategy is to apply additional criteria. In the sequel we will only declare a mode present if there are least k "small" edges that are contiguous in the Prim trajectory. This requires relating k to the probability of false alarm, denoted PFA.

From (3) the expression of the false alarm probability, the probability of occurrence of at least k consecutive connections of length less than ϵ, is given by

$$P_{FA}(k,\epsilon)=(1-\mathrm{e}^{-\lambda\epsilon^L})^k \tag{7}$$

If L-dimensional Euclidean space is considered, the volume of the half sphere of radius ϵ is $B_L(\epsilon)=\frac{1}{2}C_L\epsilon^L$.

Finally, since $\lambda=NC_L/2|\mathcal{V}|$:

$$P_{FA}(k,\epsilon)=(1-e^{-(C_L/2)\epsilon^L N/|\mathcal{V}|})^k \tag{8}$$

This formula determines the relationship between k_m (minimum number of vertices required to detect a mode) and the threshold value ϵ in the framework of a Neyman–Pearson test (Fig. 3), for a fixed PFA:

$$k=\frac{\log(P_{FA})}{\log\left(1-\exp\left(-\frac{C_L}{2}\epsilon^L\frac{N}{|\mathcal{V}|}\right)\right)} \tag{9}$$

The ratio $N/|\mathcal{V}|$ in Eq. (9) reveals the role of the density of vertices. By maintaining k and the PFA constant over the sample space, Eq. (9) specifies value of the threshold that assures a given false alarm rate PFA. In practice, for a given PFA one has to either fix the threshold ϵ and deduce the minimal k or fix the minimal k and compute the threshold.

The proposed clustering algorithm is summarized in Algorithm 1. As illustrated in Section 4, this algorithm performs well for identifying distinct groups in data sets.

Algorithm 1. Initialization of k-means with Prim's algorithm.

INPUT: V: data set of N points $\in \mathbb{R}^L$
d: dissimilarity matrix ($N\times N$)
Output: *Index*: matrix of labels
1: Construct the Prim trajectory on the data set V using d.
2: Threshold g with $\epsilon=\sigma(e_i)$.
3: Evaluate k for a chosen PFA (e.g., 0.05) (k is the minimum number of vertices that should be connected for a detected mode) (Eventually re-adjust ϵ to obtain k in the desired range).
4: Compute the number k of modes, after discarding all candidates with less than k connected vertices
5: Compute the centers of mass $\boldsymbol{\mu}$ of every detected mode C.
6: **repeat**
7: Assign v to nearest cluster C_j where $Index(\mathbf{v}_i)=\mathrm{argmin}_{j\in 1,\ldots,k}\|\mathbf{v}_i-\boldsymbol{\mu}_j\|$.
8: Recompute mean $\boldsymbol{\mu}_j$ as $\boldsymbol{\mu}_j=\frac{\sum_{\mathbf{v}_i\in C_j}\mathbf{v}_i}{card(C_j)}$.
9: **until** convergence
10: **return Index**

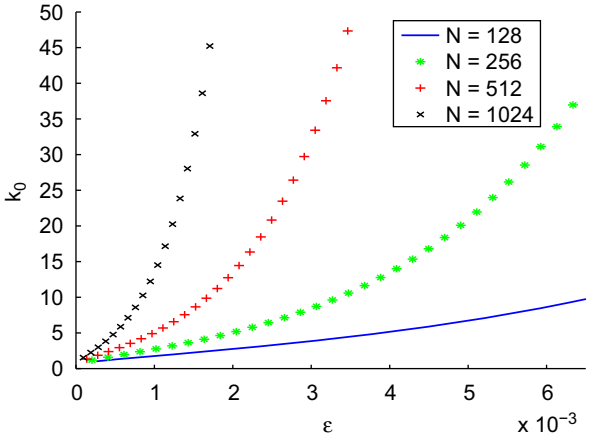

Fig. 3. Minimum number of points in a detected cluster as a function of the threshold value ϵ from sets of uniformly distributed vertices over $[0,1]^2$, for a $P_{FA}=0.05$ and $N=128,256,512,1024$ vertices respectively.

3. Implementation of MST's for large data sets

The construction of the Prim's trajectory or the MST in the previous section requires evaluation of a ranking of the set of all pairwise distances. If the sample set of N vertices is in an L-dimensional space, this will induce a computational load of $O(LN^2/2)$ flops for evaluating the similarity matrix, and an additional $O(N\log(N))$ logical operations for the ranking step (for example by using the "quick-sort" algorithm [38]). This clearly becomes prohibitive for large data sets.

However, the Prim's trajectory construction can be reformulated as a series of nearest neighbor searches. Finding nearest neighbors of a given point in space within an efficient computation time has been widely addressed. One of the most famous search algorithm is the space partitioning technique based on a k dimensional binary search tree, referred to as the $k-d$ tree [39]. The optimized $k-d$ tree method chooses the hyperplane passing through the median point perpendicular to the coordinate axis where the distribution exhibits the largest spread. Based on a search for nearest neighbors using the $k-d$ tree and priority queues to reduce these searches, Bentley and Friedman [40] proposed several algorithms of MST construction. Guttman [41] proposed the R-tree: Although $k-d$ tree methods use partitioning element that is a hyperplane, the R-tree uses hyper-rectangular regions. However these methods cannot easily handle high-dimensional data, since their complexity exponentially grows with the dimension. Other methods such as $v-p$ trees [42], quad-trees [43] or hB-trees [44] have the same computational difficulty for large data sets.

Here, we reduce the computational load by using a preconditioning data-driven hierarchical classification tree, a procedure we denote as CT. The tree can be learned from a small randomly chosen subset of R data points involving only $O(LR\log R)$ logical operations. It is designed to easily identify the neighborhoods of each vertex by comparing the vertex coordinates along each of the L dimensions to a set of L thresholds. The size of the neighborhood can be set up in such a way that in the average, each neighborhood contains approximately $M\ll N$ vertices. Consequently, the number of pairwise distances that need to be evaluated is of the order of $M^2/2$. Thus the computational load is lowered to the order of $O(NLM^2/2)$ flops and $O(NM\log M)$ logical operations, giving substantial computational savings when $M\ll N$. The constructions of MST using the pre-conditioning classification tree (CT) described below will be referred to as "nearest-neighbor MSTs" (NN MST) in the sequel.

Assume that after l iterations of the CT growing procedure a partition Π was created. At the next iteration Π is refined by splitting each of its constitutive cells, called parent cells, into 2^L children cells. This step is conducted by employing a median split over each axis. In order to control the number of cells, only the cells with a minimum number (n_{min}) of vertices are split further, the others being declared terminal cells, or leaves in the CT. The advantage of such a procedure is that the cells tend to fall in areas of the sample space where the sample density is high. Furthermore, the CT procedure allows to maintain an approximately constant number of vertices in every

leaf of the tree. The median splitting rule allows to define subcells with approximately equal marginal probabilities for arbitrary underlying distribution.

A criterion for setting n_{min} using the variance of the median estimate was derived in [45] and refined in [30]. The same idea is used here, and developed for each of the L-axes. The L dimensional CT is obtained by merging all one-dimensional CTs by a logical 'and'.

Consider a cell Π in the L-dimensional space. Then $\Pi = \times_{j=1}^{L} \pi_j$ where $\pi_j = [\alpha_j, \beta_j]$. π_j is a one-dimensional cell defined on axis j; let N_{π_j} be the number of vertices in π_j. The coordinates are assumed to be iid with marginal continuous density function $f_{\mathbf{v}_j|\pi_j}$ on the interval $[\alpha_j, \beta_j]$. The sample median $\hat{T}_{\pi_j}$ is asymptotically Gaussian distributed [45]:

$$\hat{T}_{\pi_j} \sim \mathcal{N}\left(T_{\pi_j}, \frac{1}{4N_{\pi_j}(f_{\mathbf{v}|\pi_j}(T_{\pi_j}))^2}\right) \tag{10}$$

where T_{π_j} is the theoretical median: $T_{\pi_j} = (\beta_j + \alpha_j)/2$. A criterion to select n_{min} is specified by requiring that N_{π_j} is large enough to ensure that the density of $\hat{T}_{\pi_j}$ has maximum mass inside the interval $[\alpha, \beta]$. It will be assumed that the sample medians $\hat{T}_{\pi_j}$, $j \in \{1, \ldots, L\}$ are statistically independent. Furthermore, it is assumed that $f_{\mathbf{v}|\pi_j}$ is a uniform distribution over π_j (which will be practically true for cells whose size is small compared to the variations of the underlying distribution) we obtain

$$1 - \prod_{j=1}^{L} \Pr(|\hat{T}_{\pi_j} - T_{\pi_j}| \le (\beta_j - \alpha_j)/2) \le \epsilon \tag{11}$$

Under the Gaussian approximation (10), and integrating over all coordinate axes, we obtain

$$1 - Pr(|Z| < \sqrt{N_{\Pi}})^L \le \epsilon$$

where Z is a standard normal random variable (zero mean and unit variance):

$$Pr(|Z| < \sqrt{N_{\Pi}}) = erf(\sqrt{N_{\Pi}/2})$$

where *erf* is the error function: $erf(x) = (2/\sqrt{\pi}) \int_0^x e^{-t^2}\, dt$.

This motivates the following stopping rule for the CT cell subdivision:

if $N_\pi \ge 2[erf^{-1}((1-\epsilon)^{1/L})]^2 = n_{min}$ then subdivide π

else stop

The structure for the CT associated to the L-dimensional distribution is obtained by storing every one-dimensional CT. Each CT is associated to an array describing the hierarchy and the thresholds. Thus, nearest neighbors of a vertex are found by performing a small number of logical tests to identify the terminal cell to which it belongs, thus sparing the costly flops needed to evaluate all N^2 dissimilarities.

4. Results and discussions

In this section, we illustrate the proposed clustering algorithm for three applications. The first application is real-world data extracted from the UCI Machine Learning Repository [46]. The second application is the segmentation of multi-spectral satellite image. The third application is to classification of chemical species in hyper-spectral imaging of planet Mars.

4.1. Experiments on the estimation of the number of clusters

Estimation of the number k of clusters is performed on a set of simulated data, for which the actual k is known. All previous methods rely upon the analysis of the results of a clustering algorithm to select k. We used the k-means algorithm and values ranging from $k=1$ to $k=10$ were tested. The tests were conducted on four different data models described below, and 50 independent data samples were generated for each model.

Model 1: Three spherical clusters in $\mathbb{R}^2$. The clusters were generated by simulating a mixture of three spherical normal variables with means $(0,0)$,$(0,5)$,$(5,-3)$.

Model 2: Four spherical clusters in $\mathbb{R}^3$. Each cluster is randomly chosen to contain 25 or 50 observations, with means randomly chosen from an $\mathcal{N}([0,0,0],5I_3)$ distribution, where I_3 is the identity matrix in $\mathbb{R}^3$.

Model 3: Four spherical clusters in $\mathbb{R}^{10}$. Each cluster is randomly chosen to contain 25 or 50 observations, with means randomly chosen from an $\mathcal{N}(\underline{0}, 3.6I_{10})$ distribution.

Model 4: Two elongated clusters in $\mathbb{R}^3$. Cluster 1 $= \{\mathbf{x} + \mathbf{b} / x_1 = x_2 = x_3 = t, t \in [-.5, .5]\}$ where $\mathbf{b}$ is a centered Gaussian noise in $\mathbb{R}^3$, with covariance matrix $.1I_3$; Cluster 2 is similarly obtained except that the noise is centered in $m = [10, 10, 10]$. The two clusters are thus stretched along the main diagonal.

For comparison we also simulated the methods of Calinski and Harabaz, Hartigan, Tibshirani, and Kaufman and Rousseeuw. Calinski and Harabasz [20] proposed a test based on the optimization of a ratio between the inter and intra-cluster variances. Hartigan [17] derived a method that evaluates the relatives variation of the intra-cluster sum of squares when k increases. Tibshirani et al. [16] constructed Tibshirani's Gap: a criterion that relies on the intra-cluster measure, but introduced a corrective term accounting for the statistical variation of the intra-cluster measure under a null hypothesis. Kaufman and Rousseeuw [27] pursued analogous ideas, measuring the average intra-cluster distance and the minimal inter-cluster separation as contrasted to variances.

A description of the simulation is given in Fig. 5. For each of the 50 data samples and for every model, the estimation of the number of clusters present in the data set according to the various criteria described above is recorded. The obtained results are reported in Table 1. Our method was applied with fixed parameters $\varepsilon = \sigma_e$, and $k_m = 3$, where σ_e stands for the standard variation of the edge-length distribution over the MST. (Note the corresponding PFA is obtained by applying Eq. (8).) The method proposed in this paper exhibits improved performance as compared to other methods for all the dimensions of the sample data studied here.

4.2. Similarity measures

The choice of a particular distance measure between points has a great impact on the resulting cluster identification. Using different metrics leads to different MSTs,

and it makes sense to adapt the dissimilarity measure to the physical structure of the sample data. It is for instance well known (see e.g., [47]) that for feature vectors that are spectral(-like), information divergences dissimilarity measures can lead to better results. Furthermore, these divergences handle the case of missing data, e.g., missing spectral measurements at a given wavelength. Let $\mathbf{x}=\{x_1,\ldots,x_L\}$ and $\mathbf{y}=\{y_1,\ldots,y_L\}$ be two non-negative feature vectors in $\mathbb{R}^{*L}$. Let $\tilde{x}_i=x_i/\sum_{j=1}^{L}x_j$ and $\tilde{\mathbf{x}}=\{\tilde{x}_1,\ldots,\tilde{x}_L\}$ as the normalized quantities, respectively. We define $\tilde{\mathbf{y}}$ similarly.

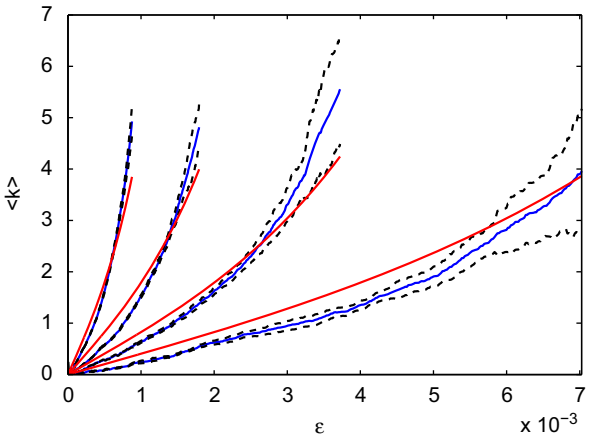

Fig. 4. Average size of false alarm detected cluster from sets of uniformly distributed vertices over $[0,1]^2$, and $N=128,256,512,1024$ vertices respectively (curves from the right to the left). Theoretical curve (red) and numerical simulation (blue). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

There are many information divergence measures, e.g., Csiszár [48] or Bregman [49]). Here we focus on two particular measures: the symmetrized Kullback–Leibler and the Rényi divergences [50]. The symmetrized Kullback–Leibler divergence is defined as

$$d_{KL}(\mathbf{x},\mathbf{y})=\sum_{i=1}^{L}(\tilde{x}_i-\tilde{y}_i)\log\frac{\tilde{x}_i}{\tilde{y}_i} \tag{12}$$

Alternatively, the symmetrized α Rényi divergence (α satisfies $0<\alpha<1$) can similarly be used as a dissimilarity measure:

$$d_{\alpha}(\mathbf{x},\mathbf{y})=\frac{1}{\alpha-1}\left(\log\sum_{i=1}^{L}\tilde{x}_i^{\alpha}\tilde{y}_i^{1-\alpha}+\log\sum_{i=1}^{L}\tilde{y}_i^{\alpha}\tilde{x}_i^{1-\alpha}\right) \tag{13}$$

Properties and advantages of the Rényi divergence have been detailed by Hero et al. [51]. Note that when $\alpha\to1$, the α-divergence (13) converges to the Kullback–Leibler divergence (12). Divergences do not satisfy the triangular inequality in general; they are semi-metrics.

In the remote sensing community, alternative similarity measures have been proposed for hyper-spectral data. One such measure is based on the spectral angle mapper (SAM) [52]:

$$\theta(\mathbf{x},\mathbf{y})=\arccos\left(\frac{\langle\mathbf{x},\mathbf{y}\rangle}{\|\mathbf{x}\|\|\mathbf{y}\|}\right) \tag{14}$$

where $\langle\cdot,\cdot\rangle$ is the dot product, and $\|\cdot\|$ is the Euclidean norm. The angle θ belongs to the interval $(0,\pi/2)$. This measure is used to define the angle existing between spectra. It measures a similarity rather than a dissimilarity.

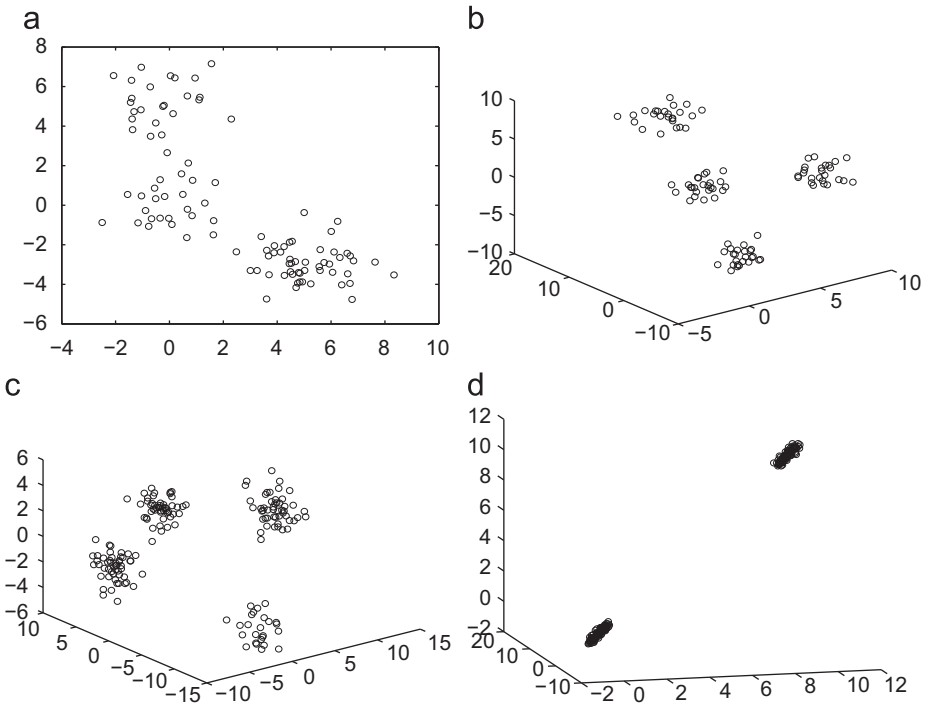

Fig. 5. (a) Model 1: $N=150$; (b) Model 2: $N=175$; (c) Model 3: $N=175$; (d) Model 4: $N=202$.

Table 1
Estimation of the number of clusters (k^* being the true number of clusters).

Method	1	2	3	4	5	>5
Model 1 ($k^*=3$)						
CH	0	7	19	10	8	6
KLai	0	4	8	8	7	23
Sil	0	29	16	5	0	0
Hart	0	14	11	9	5	11
Gap-uniform	0	33	17	0	0	0
Gap-PC	9	21	18	2	0	0
Prim	1	8	**40**	1	0	0
Model 2 ($k^*=4$)						
CH	0	2	6	8	9	25
KLai	0	2	4	4	5	37
Sil	0	10	21	12	5	2
Hart	0	11	17	9	6	7
Gap-uniform	10	18	19	2	1	0
Gap-PC	21	13	13	3	0	0
Prim	0	2	20	28	0	0
Model 3 ($k^*=4$)						
CH	0	0	6	12	7	25
KLai	0	5	7	3	5	30
Sil	0	4	23	13	4	6
Hart	0	2	21	17	4	6
Gap-uniform	1	16	22	8	3	0
Gap-PC	11	11	16	8	3	1
Prim	0	3	4	**43**	0	0
Model 4 ($k^*=2$)						
CH	0	46	0	3	1	0
KLai	0	47	0	0	0	3
Sil	0	50	0	0	0	0
Hart	0	5	12	11	12	10
Gap-uniform	0	22	17	10	1	0
Gap-PC	0	**50**	0	0	0	0
Prim	0	**50**	0	0	0	0

Table 2
Results obtained on the Iris data set.

Method	Distortion	Accuracy
Prim initialization (Euclidean)	1.97	**0.8933**
Prim initialization (Kullback–Leibler)	3.83	0.6667
Prim initialization (Rényi)	3.11	0.6667
Forgy (random)	2.99	**0.8933**
Bradley Fayyad	1.41	0.8267
KKZ	1.42	0.8467
Kaufman Rousseeuw	**1.40**	**0.8933**
k-Means ++	2.01	0.8867
Global k-means	1.41	**0.8933**

These measures may be used for evaluating dissimilarities, in the case where "probability-like" or "spectral-like" data set is analyzed. Such a choice is discussed in Sections 4.1 and 4.3. In this paper, the focus is on the k-means algorithm. k-Means can be adapted to match the metric used in the construction of the tree, similar to the Bregman clustering approach [53,54]. The assignment step in Algorithm 1 then uses the Bregman divergence instead of L2 norm.[1]

[1] Information theoretic divergences and Euclidean distance belong to the family of Bregman divergences.

4.3. Clustering experiments with Prim initialization of k-means

First, we present results of comparisons for real data from the UCI Machine Learning Repository data sets [46] . We compare our method with other k-means initialization

Table 3
Results obtained on the Wine data set.

Method	Distortion	Accuracy
Prim initialization (Euclidean)	2.05×10^4	0.6742
Prim initialization (Kullback–Leibler)	$\mathbf{1.92}\times10^4$	0.6910
Prim initialization (Rényi)	$\mathbf{1.92}\times10^4$	0.6910
Forgy (random)	4.00×10^4	0.6966
Bradley Fayyad	8.15×10^4	0.6966
KKZ	2.92×10^4	0.5843
Kaufman Rousseuw	8.56×10^4	0.6573
k-Means ++	4.15×10^4	**0.7022**
Global k-means	6.86×10^4	**0.7022**

Table 4
Results obtained on the Image Segmentation data set.

Method	Distortion	Accuracy
Prim initialization (Euclidean)	6.39×10^4	0.5173
Prim initialization (Kullback–Leibler)	5.20×10^4	0.5446
Prim initialization (Rényi)	6.39×10^4	0.5173
Forgy (random)	5.52×10^4	0.4251
Bradley Fayyad	4.91×10^4	0.5385
KKZ	1.44×10^5	0.3567
Kaufman Rousseuw	3.79×10^4	0.3355
k-Means ++	3.38×10^5	0.4771
Global k-means	$\mathbf{2.38}\times10^4$	0.4606

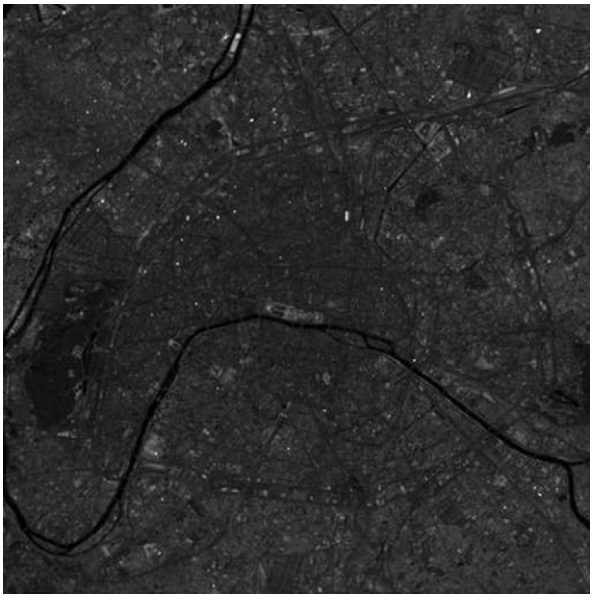

Fig. 6. Multi-spectral image of Paris.

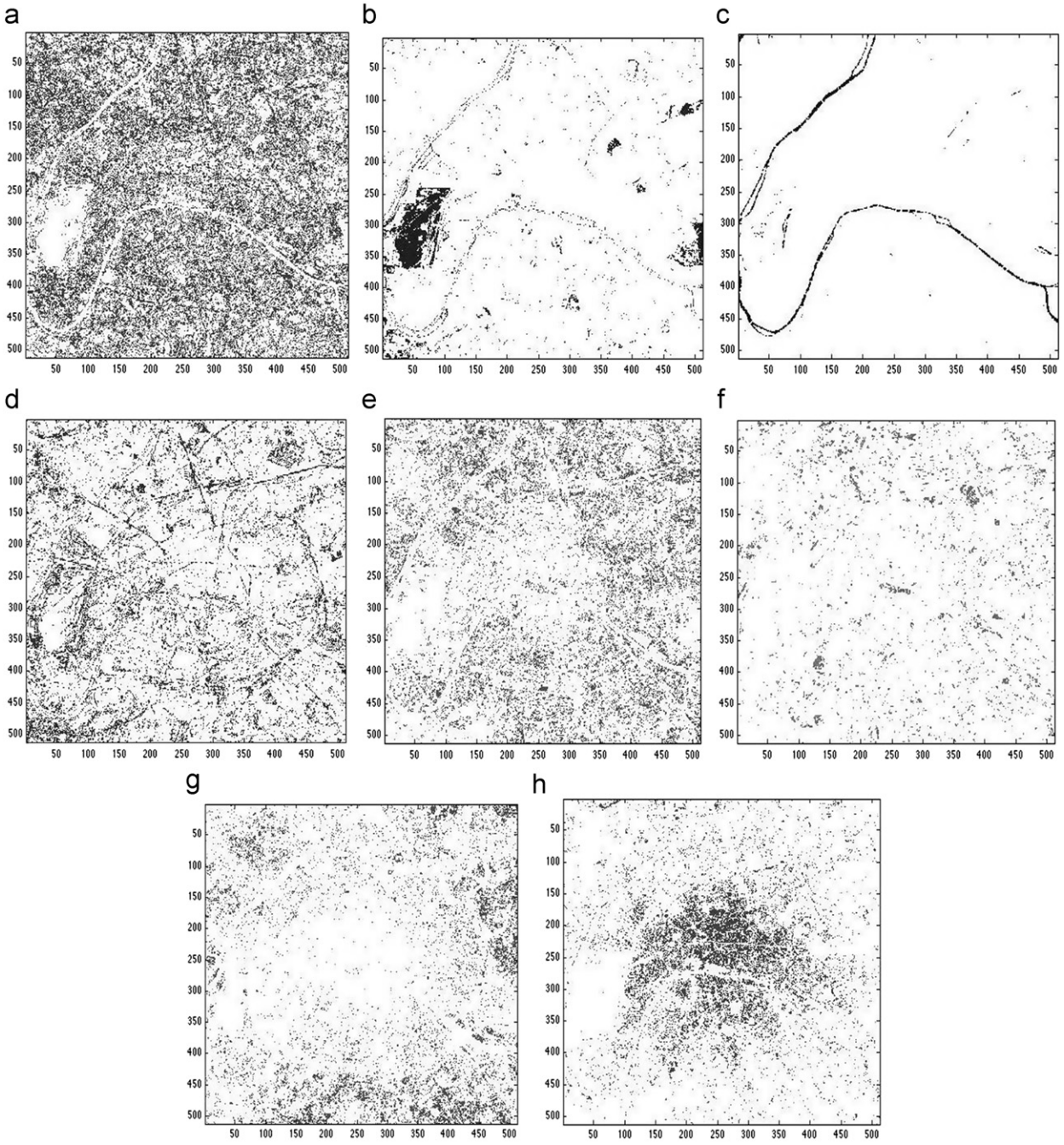

Fig. 7. Clusters obtained on the multi-spectral image of Paris: (a) cluster 1, (b) cluster 2: trees, parks, (c) cluster 3: water elements, (d) cluster 4: roads, (e) cluster 5, (f) cluster 6, (g) cluster 7 and (h) cluster 8.

methods: Forgy's approach (randomly) [24], k-means ++ [25], Global k-means [29]; Katsavounidis et al. (KKZ) [26]; Kaufman and Rousseeuw's method [27] and Bradley and Fayyad's algorithm [28]. Since for these UCI data sets the actual number of clusters is known, it is used for all methods requiring it as an input parameter. Note that for our methods, this value is estimated (with $k_m=3$, $\varepsilon=\sigma_e$). The comparisons presented in Tables 2–4, are based on the error of k-means defined by Eq. (1) (referenced as distortion in the table) and the accuracy measure (percentage of correctly labeled points). For these UCI data labels of data sets are only used to assess the error of the different methods. The results obtained are given in Tables 2–4.

The UCI data sets used are described below. The *Iris data* contains three classes (versicolor, virginica, setosa) of 50 instances each in four dimensions, where one class is linearly separable from the others and the two other classes interleave. The *Wine data* contains 178 instances

in 13 dimensions, with three clusters (different cultivars) of different sizes (59, 71, 48). The features are the quantity of constituent found in every type of wine. The *Image Segmentation* contains 2310 instances in 19 dimensions which describe aspects from seven classes of images (brick, sky, foliage, cement, window, path, grass). Each class is represented by 330 instances.

The reader will notice that our Prim initialization methods performs very well on the *Iris data*, both in terms of distortion and accuracy measure and competes with other k-means initialization methods. Information divergences appear to be more interesting for the other data. On the *Wine data* set, results obtained with the Prim initialization both with Euclidean and informational divergences compete very well with other standard initialization methods. On the image segmentation data set, which contains more instances and more clusters than the previous data sets, our Prim initialization method produces significantly better cluster performance than obtained with classical initialization methods. For all these tests, the threshold of the Prim trajectory has been set to the standard deviation of the edge length, as explained above. The minimal number of vertices required to accept a detection of a mode was set to maintain the PFA to a value below 10%, where this minimal number was determined using our asymptotic Poisson approximation presented in Section 2.

4.4. Multi-spectral image of Paris

We next consider a Landsat Thematic Mapper multi-spectral (seven bands) image of size 512×512 with 30 m resolution (Fig. 6).[2] Each image was recorded from a device operating at a different wavelength. The images cover an area corresponding to the city of Paris, France. The seven images are perfectly registered. The cluster analysis is performed on the four largest principal components of the seven bands, so in the following only four dimensional features are considered instead of the initial seven bands. The affinity measures used were information divergences between the 4-point spectra associated with each pixel, considered as feature vectors. The nearest neighbor MST algorithm described in Section 3 was applied sub-sampling the images by a factor of 4. The pixels were assumed to be independent, and no information related to the image structure (e.g., adjacency of pixels) was used in this test. Such prior information could be used as additional feature information to improve clustering but this was not investigated here.

Fig. 7 shows the eight identified clusters, for which three are easy to understand. Cluster 2 is characteristic of trees and grassy regions (one can recognize recreation areas and natural parks in Paris surroundings (Boulogne and Vincennes) and trees along the shores of the Seine). Cluster 3 exhibits the "water areas" in Paris, and the Seine river together with some known ponds are easily extracted. Cluster 4 is clearly associated with roads, asphalt and concrete. Other clusters cannot be fairly interpreted without cross-validation with, for example, pollution imaging or gas detection systems.

As the "groundtruth" cluster representation of the data is unknown, the quality of the different classifications was assessed by evaluating the Davies–Bouldin (DB) index [55] to complement the average intra cluster variance, which is minimized by k-means. DB index accounts for the separation of the clusters:

$$DB(C) = \frac{1}{k}\sum_{l=1}^{k} R_l \quad \text{where } R_l = \max_{j \neq l} R_{jl} \tag{15}$$

where $R_{ij} = (s_i + s_j)/\delta_{ij}$, and with $\delta_{ij} = d(\mu_i, \mu_j)$, and $s_i = (1/|C_i|)\sum_{v \in C_i} d(v, \mu_i)$. Thus, a lower DB index indicates a better clustering result. Table 5 summarizes the results. Our Prim initialization approach combined with informational divergences gives the best results overall.

4.5. Mars hyper-spectral image

Finally we present the results of applying our method to a hyper-spectral image of Mars. This data is very high dimensional and thus serves as a validation that our proposed method is scalable. The other initialization methods could not be compared due to their excessive computational complexity. This Mars image was provided by the Mars Express (European Space Agency) on board imaging spectrometer instrument OMEGA (Observatoire pour la Minéralogie, l'Eau, les Glaces et l'Activité). It covers the south polar cap of Mars in the Martial summer. Two infrared channels are scanned: 128 spectral planes are recorded at wavelength ranging from 0.93 μm to 2.73 μm with a resolution of 0.013 μm and 128 spectral planes from 2.55 μm to 5.11 μm with a resolution of 0.02 μm. Hence, the image is of size 300×120 pixels in 256 wavelengths. Some preprocessing of the raw data was applied by "Laboratoire de Planetologie de Grenoble" in order to remove erroneous or unreliable values, e.g., due to faulty pixels of the CCD camera, bad calibration values, etc. More information about this data set can be found in [56].

In recent years, two methods have been introduced to classify chemical species on this hyper-spectral image of planet Mars. The first method developed by Schmidt et al. [57] is a supervised approach based on a wavelet representation called wavanglet. This method requires a priori information, specifically the number of classes and the

Table 5
Results obtained on the multi-spectral image.

Clustering methods	Davies–Bouldin index
NN MST (Euclidean) + k-means	111.77
NN MST (SAM) + k-means	157.27
NN MST (Kullback–Leibler) + k-means	**74.87**
NN MST (Rényi) + k-means	**88.02**
Expectation maximization (Mixture of Gaussians)	160.19
k-Means (randomly initialized)	155.13

[2] The image can be downloaded at http://physics.ship.edu/mrc/astro/NASA_Space_Science/observe.arc.nasa.gov/nasa/education-/tools/stepby/arcpage/arc20.html.

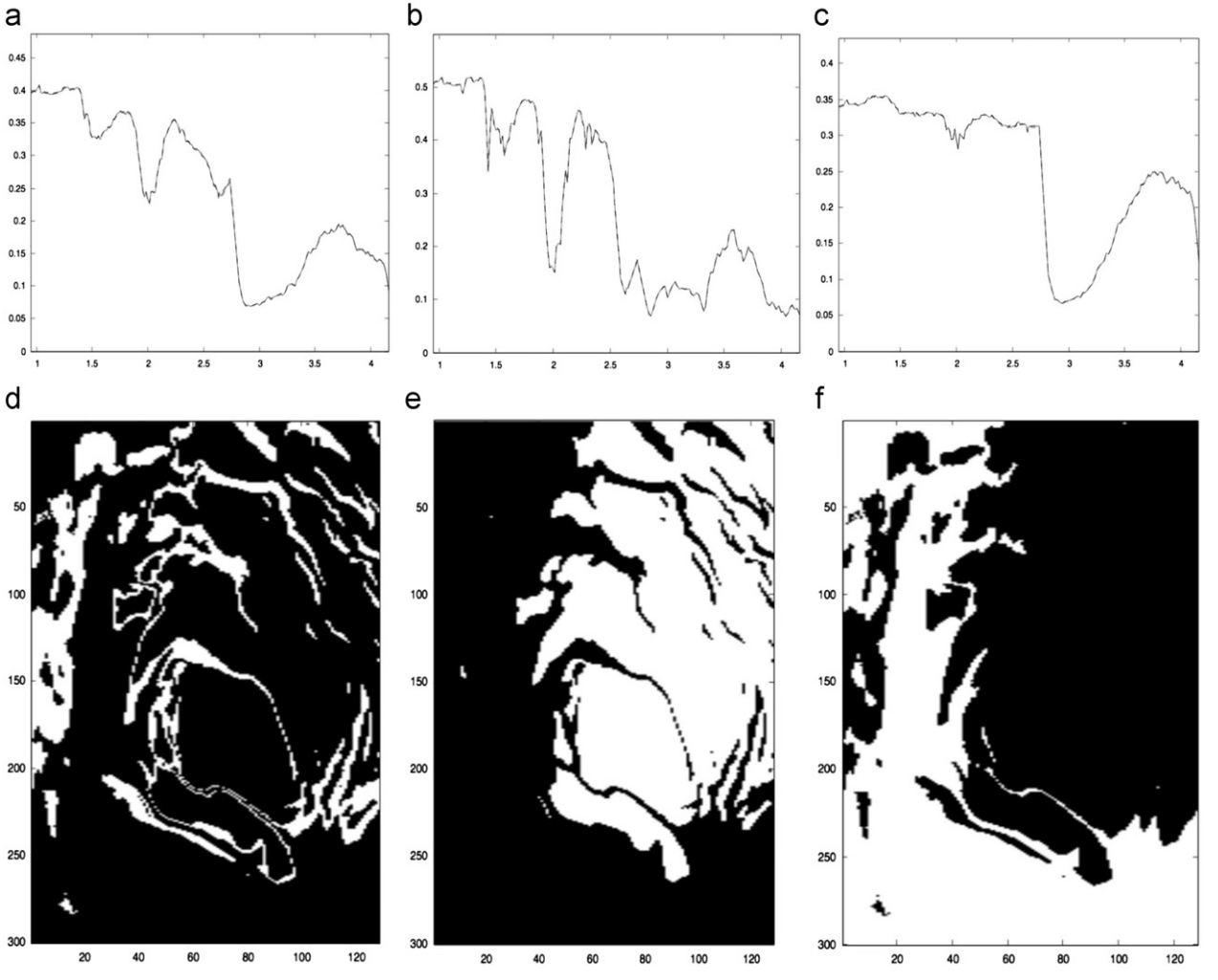

Fig. 8. Mean and $\neg \pm$ standard deviation spectra obtained: (a) H_2O ice, (b) CO_2 ice, (c) dust. Clusters obtained with our method: (d) H_2O ice, (e) CO_2 ice and (f) dust.

reference spectra (generated from a physical model of chemical compounds signature). The classification is made by computing the spectral angle in the best subspace of a wavelet filtered space. The results obtained with this method reveal three binary masks (a pixel belongs to one class) of CO_2 ice, H_2O ice and dust. Note that some pixels have not been classified. As the method requires reference spectra, it requires an a priori report of experts to identify all physical compounds present in the scene, and subsequently generate (or extract from the image) reference spectra.

The second method was developed by Moussaoui et al. [58] and is based on blind source separation method referred to as Bayesian positive source separation (BPSS). BPSS requires costly importance sampling, e.g., Monte Carlo Markov Chain (MCMC), for its implementation.

In Fig. 8 clusters obtained with our Prim initialization of k-means are displayed. The dissimilarity measure used was the symmetrized KL divergence. The image contains only a few homogeneous groups, so the computation of centroids does not require the whole image. We resized the image by column and line sub-sampling (one line over 5 and one column over 3 are kept). The three obtained clusters correspond to H_2O ice, CO_2 ice and dust. The representation of the results consists of binary masks (the pixel is white if it belongs to this cluster). H_2O ice compounds are on the peripheral of CO_2 ice compounds, which have some sense since at these places the CO_2 sublimates to reveal H_2O ice. The corresponding mean spectrum can also be seen in Fig. 8. Our results are fully consistent with the clustering results obtained by Moussaoui et al. [58].

5. Conclusions and future works

In this paper, we have proposed a new approach for clustering multi-dimensional data. The method is based on the estimation of the number of clusters and the centers of the clusters from the Prim construction of a minimum spanning tree, followed by an initialization of the classical k-means clustering algorithm.

New criteria were derived for setting the false alarm rate of a test over Prim's trajectory associated with a MST built over the set of data. The false alarm rates were derived under the null hypothesis that the data points are distributed as a homogeneous Poisson process. We demonstrated the utility of the information divergence as a dissimilarity measure for astrophysical multi-spectral image analysis. In this paper, the threshold value is constant along Prim's trajectory. This rate could be easily made a function of the position of the connected vertices. This has not been developed here but opens new perspectives.

Dimension reduction and its relationship to spectral clustering methods applied to graphs using information divergence or MST-based distances (for example, dual rooted tree distances) could also be investigated [59,8].

Acknowledgments

We thank the OMEGA team at IAS/Orsay for his support with sequencing and data download activities. A. Hero's contribution to this paper was partially supported by the US National Science Foundation Grant no. CCF 0830490. The authors thank Dr. E. Slezak (OCA, University of Nice, France) for useful discussions on the application of these methods.

References

[1] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Computing Surveys 31 (3) (1999) 264–323.
[2] R. Xu, D. Wunsch II, Survey of clustering algorithms, IEEE Transactions on Neural Networks 16 (3) (2005) 645–678.
[3] S. Theodoridis, K. Koutroumbas, Pattern Recognition, 3rd ed., Academic Press, 2006.
[4] J.B. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, University of California Press, Berkeley, 1967, pp. 281–287.
[5] R. Prim, Shortest connection networks and some generalizations, Bell System Technical Journal 36 (1957) 1389–1401.
[6] W. Stuetzle, Estimating the cluster tree of a density by analyzing the minimal spanning tree of a sample, Journal of Classification 20 (5) (2003) 25–47.
[7] V. Olman, D. Xu, Y. Xu, Identification of regulatory binding sites using minimum spanning trees, in: Proceeding of the 8th Pacific Symposium on Biocomputing, vol. 3, Lihue, Hawaii, USA, 2003, pp. 327–338.
[8] F.R. Bach, M.I. Jordan, Learning spectral clustering, with application to speech separation, Journal of Machine Learning Research 7 (2006) 1963–2001.
[9] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: analysis and an algorithm, in: Advances on 15th Annual Conference on Neural Information Processing Systems, vol. 14, Vancouver, British Columbia, Canada, 2001.
[10] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 888–905.
[11] F.R.K. Chung, Spectral graph theory, in: Conference Board on the Mathematical Sciences, no. 92, American Mathematical Society, 1997.
[12] G.H. Ball, D.J. Hall, ISODATA, A Novel Method of Data Analysis and Classification, Technical Report, Stanford University, Stanford, CA, 1965.
[13] J.M. Peña, J.A. Lozano, P. Larrañaga, An empirical comparison of four initialization methods for the k-means algorithm, Pattern Recognition Letters 20 (1999) 1027–1040.
[14] S.J. Redmond, C. Heneghan, A method for initialising the k-means clustering algorithm using *kd*-trees, Patter Recognition Letters 28 (2007) 965–973.
[15] J. Rissanen, Stochastic Complexity in Statistical Inquiry, World Scientific Series in Computer Science, vol. 15, 1989.
[16] R. Tibshirani, G. Walther, T. Hastie, Estimating the number of clusters in a dataset via the gap statistic, Journal of the Royal Statistical Society: Series B 63 (2001) 411–423.
[17] J. Hartigan, Clustering Algorithms, Wiley, New York, 1975.
[18] W.J. Krzanowski, Y.T. Lai, A criterion for determining the number of groups in a data set using sum of squares clustering, Biometrics 44 (1985) 23–34.
[19] G.W. Milligan, M.C. Cooper, An examination of procedures for determining the number of clusters in a data set, Psychometrika 50 (1985) 159–179.
[20] T. Calinski, J. Harabasz, A dendrite method for cluster analysis, Communications in Statistics 3 (1) (1974) 1–27.
[21] H. Bischof, A. Leonardis, A. Selb, MDL principle for robust vector quantisation, Pattern Analysis & Applications 2 (1999) 59–72.
[22] D. Pelleg, A. Moore, *X*-means: extending *K*-means with efficient estimation of the number of clusters, in: Proceedings of the 17th International Conference on Machine Learning, Palo Alto, CA, USA, 2000, pp. 727–734.
[23] G. Hamerly, C. Elkan, Learning the k in k-means, in: Advances on 7th Annual Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 2003, pp. 281–288.
[24] E. Forgy, Cluster analysis of multivariate data: efficiency vs. interpretability of classifications, Biometrics 21 (1965) 768–769.
[25] D. Arthur, S. Vassilvitskii, k-means++: the advantages of careful seeding, in: Proceedings of the 18th annual ACM-SIAM symposium on Discrete algorithms (SODA), Society for Industrial and Applied Mathematics, New Orleans, LA, USA, 2007, pp. 1027–1035.

[26] I. Katsavounidis, C.C.J. Kuo, Z. Zhen, A new initialization technique for generalized Lloyd iteration, IEEE Signal Processing Letters 1 (10) (1994) 144–146.
[27] L. Kaufman, P.J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics, Wiley, New York, 1990.
[28] P.S. Bradley, U.M. Fayyad, Refining initial points for k-means clustering, in: Proceedings of 15th International Conference on Machine Learning, Madison, WI, USA, 1998, pp. 91–99.
[29] A. Likas, N. Vlassis, J.J. Verbeek, The global k-means clustering algorithm, Pattern Recognition 36 (2) (2003) 451–461.
[30] A. Badel, O.J.J. Michel, A.O. Hero, Tree structured non linear signal modeling and prediction, IEEE Transactions on Signal Processing 47 (11) (1999) 13037–13041.
[31] A. Gersho, R.M. Gray, Vector Quantization and Signal Compression, Kluwer Academic Publishers, 1992.
[32] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, Wiley Intersciences, 2001.
[33] R.L. Graham, P. Hell, On the history of the minimum spanning tree problem, Annals of the History of Computing 7 (1) (1985) 43–57.
[34] J. Gower, G. Ross, Minimum spanning trees and single linkage cluster analysis, Applied Statistics 18 (1969) 54–64.
[35] J.R. Slagle, C.-L. Chang, R.C.T. Lee, Experiments with some cluster analysis algorithms, Pattern Recognition 6 (3–4) (1974) 181–187.
[36] O.J.J. Michel, P. Bendjoya, P. RojoGuer, Unsupervised clustering with MST: application to asteroid data, in: Proceedings of 4th Physics in Signal and Images Processing, Toulouse, France, 2005.
[37] J.M. Steele, Probability theory and combinatorial optimization, in: CBMF Regional Conferences in Applied Mathematics, vol. 69, Society for Industrial and Applied Mathematics (SIAM), 1997.
[38] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes 3rd Edition: The Art of Scientific Computing, vol. 3, Cambridge University Press, 2007.
[39] J.H. Friedman, J.L. Bentley, R.A. Finkel, An algorithm for finding best matches in logarithmic expected time, ACM Transactions on Mathematical Software 3 (3) (1977) 209–226.
[40] J.L. Bentley, J.H. Friedman, Fast algorithms for constructing minimal spanning trees in coordinate spaces, IEEE Transactions on Computers 27 (2) (1978) 97–105.
[41] A. Guttman, R-trees: a dynamic index structure for spatial searching, in: B. Yormark (Ed.), Proceedings of Annual Meeting SIGMOD, Boston, MA, USA, 1984, pp. 47–57.
[42] P. Yianilos, Data structures and algorithms for nearest neighbor search in general metric spaces, in: Proceedings of the 4th Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1993, pp. 311–321.
[43] R.A. Finkel, J.L. Bentley, Quad trees a data structure for retrieval on composite keys, Acta Informatica 4 (1) (1974) 1–9.
[44] D.B. Lomet, B. Salzberg, The hB-tree: a multiattribute indexing method with good guaranteed performance, ACM Transactions on Database Systems 15 (4) (1990) 625–658.
[45] A.M. Mood, F.A. Graybill, D.C. Boes, Introduction to the Theory of Statistics, 3rd ed., McGraw-Hill, 1974.
[46] A. Asuncion, D.J. Newman, UCI Machine Learning Repository, 2007. URL ⟨http://www.ics.uci.edu/∼ mlearn/MLRepository.html⟩.
[47] C.-I. Chang, An information-theoretic approach to spectral variability, similarity, and discrimination for hyperspectral image analysis, IEEE Transactions on Information Theory 46 (5) (2000) 1927–1932.
[48] I. Csiszár, Information-type measures of difference of probability distributions and indirect observation, Studia Scientiarum Mathematicarum Hungarica 2 (1967) 229–318.
[49] L.M. Bregman, The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming, USSR Computational Mathematics and Physics 7 (1967) 200–217.
[50] T. Cover, J. Thomas, Elements of Information Theory, Wiley-Interscience, 1991.
[51] A. Hero, B. Ma, O. Michel, J. Gorman, Applications of entropic spanning graphs, IEEE Signal Processing Magazine 19 (5) (2002) 85–95.
[52] N. Keshava, Distance metrics and band selections in hyperspectral processing with applications to material identification and spectral libraries, IEEE Transactions on Geoscience and Remote Sensing 42 (7) (2004) 1552–1565.
[53] I.S. Dhillon, S. Mallela, R. Kumar, A divisive information-theoretic feature clustering algorithm for text classification, Journal of Machine Learning Research 3 (2003) 1265–1287.
[54] A. Banerjee, S. Merugu, I.S. Dhillon, J. Gosh, Clustering with Bregman divergences, Journal of Machine Learning Research 6 (2005) 1705–1749.
[55] D.L. Davies, D.W. Bouldin, A cluster separation measure, IEEE Transactions on Pattern Analysis and Machine Intelligence 1 (2) (1979) 224–227.
[56] F. Schmidt, Martian Surface Classification Using Omega Hyperspectral Images. Spatiotemporal Study of the CO_2 and H_2O Seasonal Deposits, Ph.D. Thesis, Universitè Joseph-Fourier, Grenoble I, 2007.
[57] F. Schmidt, S. Douté, B. Schmitt, WAVANGLET: an efficient supervised classifier for hyperspectral images, IEEE Transactions on Geoscience and Remote Sensing 45 (5) (2007) 1374–1385.
[58] S. Moussaoui, H. Hauksdóttir, F. Schmidt, C. Jutten, J. Chanussot, D. Brie, S. Douté, J. Benediksson, On the decomposition of Mars hyperspectral data by ICA and Bayesian positive source separation, Special Issue of Neurocomputing on Advances in Blind Signal Processing 71 (2008) 2194–2208.
[59] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Computation 15 (6) (2003) 1373–1396.