Aletti and Micheletti *J Big Data* (2017) 4:48
https://doi.org/10.1186/s40537-017-0109-0

Journal of Big Data

METHODOLOGY Open Access

A clustering algorithm for multivariate data streams with correlated components

Giacomo Aletti and Alessandra Micheletti*

*Correspondence:
alessandra.micheletti@unimi.it
Department of Environmental Science and Policy & ADAMSS Center, Università degli Studi di Milano, Milan, Italy

Abstract

Common clustering algorithms require multiple scans of all the data to achieve convergence, and this is prohibitive when large databases, with data arriving in streams, must be processed. Some algorithms to extend the popular K-means method to the analysis of streaming data are present in literature since 1998 (Bradley et al. in Scaling clustering algorithms to large databases. In: KDD. p. 9–15, 1998; O'Callaghan et al. in Streaming-data algorithms for high-quality clustering. In: Proceedings of IEEE international conference on data engineering. p. 685, 2001), based on the memorization and recursive update of a small number of summary statistics, but they either don't take into account the specific variability of the clusters, or assume that the random vectors which are processed and grouped have uncorrelated components. Unfortunately this is not the case in many practical situations. We here propose a new algorithm to process data streams, with data having correlated components and coming from clusters with different covariance matrices. Such covariance matrices are estimated via an optimal double shrinkage method, which provides positive definite estimates even in presence of a few data points, or of data having components with small variance. This is needed to invert the matrices and compute the Mahalanobis distances that we use for the data assignment to the clusters. We also estimate the total number of clusters from the data.

Keywords: Big data, Data streams, Clustering, Mahalanobis distance

Introduction

Clustering is the (unsupervised) division of a collection of data into groups, or *clusters*, such that points in the same cluster are similar, while points in different clusters are different. When a large volume of (not very high dimensional) data is arriving continuously, it is impossible and sometimes unnecessary to store all the data in memory, in particular if we are interested to provide real time statistical analyses. In such cases we speak about *data streams*, and specific algorithms are needed to analyze progressively the data, store in memory only a small number of summary statistics, and then discard the already processed data and free the memory [7]. Data streams are for example collected and analyzed by telecommunication companies, banks, financial analysts, companies for online marketing, private or public groups managing networks of sensors to monitor climate or environment, technological companies working in IoT, etc. In this framework, there are many situations in which clustering plays a fundamental role, like customer segmentation in big e-commerce web sites, for personalized marketing solutions, image

analysis of video frames for objects recognition, recognition of human movements from data provided by sensors placed on the body or on a smartwatch, monitoring of hacker attacks to a telecommunication system, etc.

Related literature

The methods for cluster analysis present in literature can be roughly classified into two main families: *probability-based methods* (see e.g. [1]), which are based on the assumption that clusters come from a mixture of distributions, from a given family. In such case the clustering problem is reduced to the parameter estimation. These algorithms are well suited to detect the presence of non-spherical or nested clusters, but are based on specific assumptions on the data distribution, the number K of clusters is fixed at the very beginning, and, more important, they require multiple scans of the dataset to estimate the parameters of the model. Thus they cannot be applied to massive datasets or data streams.

The second family of clustering algorithms is composed by *distance-based approaches*. Given a dataset of size n, grouped into K clusters, such methods have usually the goal to find the K *centers* of the clusters which minimize the mean squared distance between the data and their closest centers. These methods usually take different names depending on the type of considered distance. If the Euclidean distance is used, the corresponding method is the classical and very popular *K-means* method (see e.g. [11]), which is probably the most diffused clustering algorithm, because of its simplicity. Anyway the exact solution of the minimization problem connected with K-means is NP-hard, and only local search approximations are implemented. The method is sensitive to the presence of outliers and to the initial guess for the centers, but improvements both in terms of speed and accuracy of the algorithm have been implemented in K-means++ [2], which exploits a randomized seeding technique. Unfortunately both the classical K-means and the K-means++ algorithms require multiple scans of the dataset or a random selection from the entire dataset, in order to solve the minimization problem. Since data streams cannot be scanned several times and we cannot (randomly) access to the entire dataset all together, also these methods are not suitable for clustering data streams.

When the elements to be clustered are not points in $\mathbb{R}^d$ but more complex objects, like functions or polygons, other clustering algorithms are used, like PAM, CLARA, CLARANS, [12, 15], which are based on non-euclidean distances defined on suitable spaces. These methods are looking for *medoids*, instead of means, which are the "most central elements" of each cluster and are selected from the points in the dataset. Also these algorithms cannot be efficiently applied to analyse data streams, since they either require multiple scans of the sample, or the extraction of a subsample to identify the centroids or medoids, then all data are scanned according to such identification and the medoids are not any more updated with the information coming from the whole dataset. Actually such popular methods are suited for data which are very high dimensional (e.g. functions) or for geometrical or spatial random objects, but not for datasets with an high number of (rather small dimensional) data.

The key element in smart algorithms to treat data streams is to find methods to represent the data with summary statistics which are retained in memory, while the single data are discarded. Such summary statistics must be updated when each new

observation, or group (chunk) of observations, is processed, since a second scan of the data is not allowed. This strategy to analyse data streams is followed in O'Callaghan et al. [16], where the STREAM algorithm is proposed as an extension of BIRCH [19]. The STREAM method solves a so called *K-Median* problem, which is a generalization of K-means where the Euclidean distance is replaced by a general distance. The performance of the STREAM method with respect to computational costs and quality of the clustering, measured in terms of sum of squared distances (SSQ) of data points from the assigned clusters centers, is also studied, in particular in comparison with K-means, providing good theoretical and experimental results. In the STREAM algorithm the number K of clusters is not specified in advance, and is evaluated by an iterative combination between SSQ and the number of used centers. The main defect of the STREAM algorithm is that it uses a *global* metric D on the space of the data points, and thus does not take into account that different clusters may have different specific variability. Further the metric D is supposed completely known and is not estimated from the data.

In many situations the quality of the clustering is improved if a *local metric* is used. A local metric is a distance which takes into account the shape of the "cloud" of data points in each cluster to assign the new points (see Fig. 1).

A first attempt to use a local distance is given by the Bradley–Fayyad–Reina (BFR) algorithm [3, 14], which solves the K-means problem by using a distance based on the

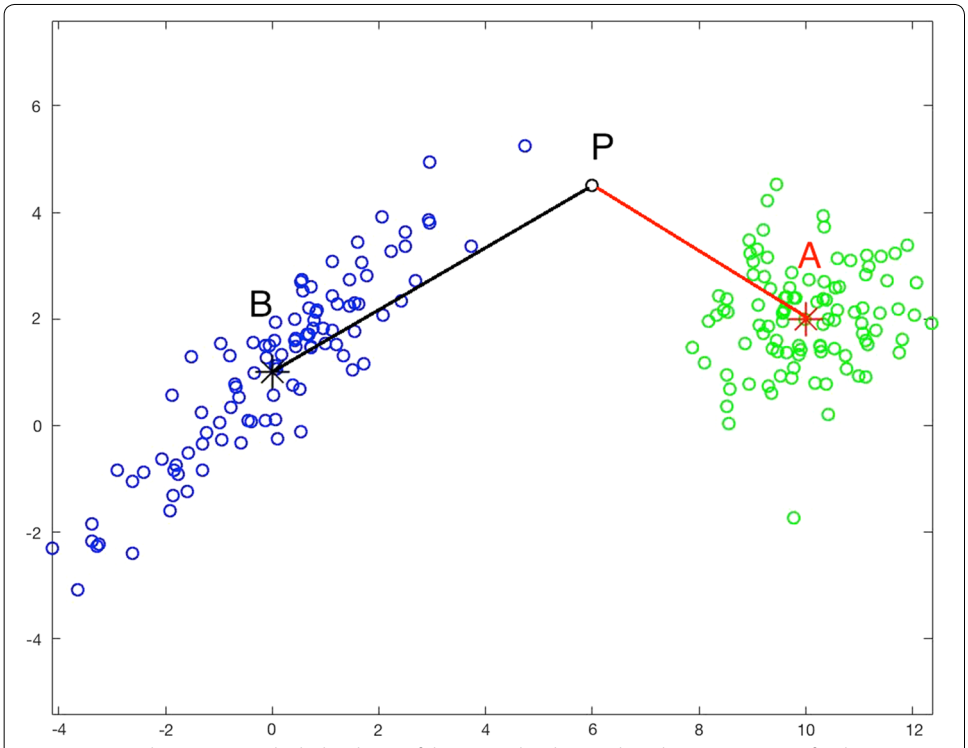

Fig. 1 A typical situation in which the shape of the point clouds must be taken into account for the points assigment: two clusters of gaussian points are represented, both composed by 100 data points. The new data point P is closer to the cluster center A if we use a global metric, like e.g. the Euclidean distance, but the point is more likely to belong to the cluster centered at B

variance of each component of the random vectors belonging to the different clusters. The BFR algorithm is based on the assumption that the clusters' distribution results from a mixture of multivariate normal distributions, whose parameters are estimated from the data streams. The BFR Algorithm for clustering is based on the definition of three different sets of data:

(a) The *retained set (RS)* The set of data points which are not recognized to belong to any cluster, and need to be retained in the buffer;
(b) The *discard set (DS)* The set of data points which can be discarded after updating the summary statistics;
(c) The *compression set (CS)* The set of summary statistics which are representative of each cluster.

Each data point is assigned to one of these sets on the basis of its local distance from the center of each cluster. Here the Mahalanobis distance is used, computed with respect to the sample covariance matrix of each cluster.

The main weakness of the BFR Algorithm resides in the assumption that the covariance matrix of each cluster is diagonal, which means that the components of the analyzed multivariate data should be uncorrelated. With such assumption, at each step of the algorithm only the means and variances of each component of the clusters centers must be retained, reducing thus the computational costs. Further, in this setting the estimated covariance matrices are invertible even in presence of clusters composed just by two p-dimensional gaussian data points. Anyway such assumptions geometrically imply that the level surfaces (ellipsoids) of the gaussians including the data points in each cluster should be oriented with main axes parallel to the reference system.

Aims and overview of the paper

We here propose a method to clusterize data streams, using a *local metric* which is estimated in real time from the data. Such metric is based on the Mahalanobis distance of the data points from each cluster center $\mathbf{c}_i$, computed using an estimator of the covariance matrix of the corresponding ith cluster. In the following we will always represent vectors as column vectors and we will assume that our data are vectors in $\mathbb{R}^p$.

Definition Let $\mathbf{x}$ be a data point and $\mathbf{c}_i$ be the center of the ith cluster. Assume that the elements of the ith cluster come from a population having covariance matrix Σ_i. Then the Mahalanobis distance of $\mathbf{x}$ from $\mathbf{c}_i$ is given by

$$\Delta(\mathbf{x}, \mathbf{c}_i) = (\mathbf{x} - \mathbf{c}_i)^T \Sigma_i^{-1} (\mathbf{x} - \mathbf{c}_i).$$

We assume that the data points are vectors in $\mathbb{R}^d$ with correlated components and we thus estimate all the terms of the covariance matrix of each cluster, including the off diagonal ones. We use the sample mean of each cluster as centers $\mathbf{c}_i$.

We divide the data in the same three sets defined in the BFR algorithm, we don't fix a priori the number K of clustersρ, and we evaluate and update such number using a density condition. Thus in our procedure from time to time new clusters will be formed, composed only by a few data points, not sufficient to obtain a positive definite estimate of the corresponding covariance matrix using the classical sample covariance estimator.

We thus use an optimal double shrinkage estimator of the covariance matrix, which provides always positive definite matrices, that are then inverted to compute the Mahalanobis distance.

In our setting we will relax a little bit the assumption of gaussianity stated in the BFR algorithm, assuming that the data come from a mixture of "bell shaped" distributions, but possibly having a bigger multivariate kurtosis (i.e. fatter queues) than a gaussian.

Our algorithm is thus an improvement of the BFR algorithm, relaxing some of its assumptions. Since with our method also the covariance terms of the clusters must be retained, there is an increase in the computational costs with respect to BFR, but such increase can be easily controlled and is affordable if the processed data are not extremely high dimensional. Therefore our algorithm is targeted to problems with data streams composed by data points of "medium" dimension, i.e. a dimension not so small to apply visualization techniques to identify the clusters (2D or 3D problems), which usually work better, but much smaller than the number of available data.

The paper is then structured as follows: in "The covariance matrices of the clusters" section we face the problem of the estimate of the covariance matrix of each cluster. We modify a Steinian linear shrinkage estimator in order to obtain a positive definite estimator of the covariance matrix, which can be applied also to non-gaussian cases, and which can be incrementally updated during the data processing. In "Summary statistics and primary data compression" section we introduce the summary statistics that will be retained in memory for each cluster, and we show that they can easily be updated when new data streams are processed. We then describe the way by which the data points are assigned to the three sets RS, CS, DS. In "Secondary data compression" section we describe the *secondary compression*, that is the way by which the points in RS and CS can be merged to pre-existing clusters or are put together to form new clusters. In "Results on simulated and real data and discussion" section we apply our method first to synthetic data, and we compare heuristically its performances with the case in which the data points are assumed to have uncorrelated components, like in the BFR algorithm. We then apply our method to cluster the real dataset KDD-CUP'99 (http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html), a network intrusion detection dataset, that was used also to test the STREAM algorithm. We apply our algorithm to all the variables in the dataset which are declared continuous. Actually some of such variables have a very small variance; anyway our optimal double shrinkage estimator of the covariance matrices of the clusters guarantees positive definite estimates also in this situation, stabilizing thus the local Mahalanobis distances that we use in our procedure. The results are coherent with the structure of the dataset, whose data should be divided into five clusters, as we obtain.

In this paper we don't study the asymptotic properties of our algorithm, but we limit ourselves to show heuristically that our algorithm provides better results of other methods to cluster data streams present in literature, just with a small increase in the computational costs.

The covariance matrices of the clusters

Our algorithm is based on the Mahalanobis distance, it is hence crucial to estimate the covariance matrices of the clusters in an optimal way. Let us first observe that when a new cluster is formed, it contains too few data points to obtain a positive definite estimate of the covariance matrix, using the sample covariance matrix, at least until $N \leq p$, where N is the number of data in a cluster and p the data points dimension.

To solve this problem in an optimal way, we exploit the optimal double shrinkage estimator given in [10, Equation (3.6)] by

$$\hat{\Sigma} = (1 - \hat{\lambda}_I - \hat{\lambda}_D)S + \hat{\lambda}_I \frac{tr(S)}{p} I_p + \hat{\lambda}_D D_S, \tag{1}$$

where S is the sample covariance matrix, D_S is its diagonal matrix, I_p is the identity matrix of order p, and $0 \leq \hat{\lambda}_I + \hat{\lambda}_D \leq 1$ are weighting the convex combination of the three matrices. This estimator is optimal in terms of quadratic loss [8, 10, 18], and it leads to covariance matrix estimators that are non-singular, well-conditioned, expressed in closed form and computationally cheap regardless of p. Therefore, in these terms, it is the optimal choice among the possible alternatives, where the first term $(1 - \hat{\lambda}_I - \hat{\lambda}_D)$ should be initially settled close to 0, and then its value is increasing to 1 when $N \to \infty$. We note that when $\hat{\lambda}_I = 0$ and $\hat{\lambda}_D = 1$ we obtain the local distance used in BFR. In [10], $\hat{\lambda}_I, \hat{\lambda}_D$ are given as functions of the quantities

$$\begin{pmatrix} \lambda_I \\ \lambda_D \end{pmatrix} = \begin{pmatrix} \mathrm{tr}[(S - \frac{\mathrm{tr}(S)}{p} I_p)^2] & \mathrm{tr}[(S - \frac{\mathrm{tr}(S)}{p} I_p)(SD_S)] \\ \mathrm{tr}[(S - \frac{\mathrm{tr}(S)}{p} I_p)(SD_S)] & \mathrm{tr}[(S - D_S)^2] \end{pmatrix}^{-1} \begin{pmatrix} \mathrm{tr}(S^2) - \widehat{\mathrm{tr}[\Sigma^2]} \\ \mathrm{tr}(S^2) - \mathrm{tr}(SD_S) - \mathrm{tr}[\Sigma \widehat{(\Sigma - D_\Sigma)}] \end{pmatrix}, \tag{2}$$

where $\widehat{\mathrm{tr}[\Sigma^2]}$ and $\mathrm{tr}[\Sigma \widehat{(\Sigma - D_\Sigma)}]$ are unbiased estimators of the corresponding quantities $\mathrm{tr}[\Sigma^2]$, $\mathrm{tr}[\Sigma(\Sigma - D_\Sigma)]$, Σ is the true covariance matrix of the considered cluster, and D_Σ its diagonal matrix. Unfortunately (see [10, 18]), both these estimators are based on the scalar statistics

$$Q^{(N)} = \frac{1}{N-1} \sum_{i=1}^{N} ((\mathbf{x}_i - \bar{\mathbf{x}}_N)^\top (\mathbf{x}_i - \bar{\mathbf{x}}_N))^2,$$

proposed by [8], where

$$\bar{\mathbf{x}}_N = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}_n, \tag{3}$$

is the centroid of the considered cluster, composed by N data points. In data stream framework, we note that $Q^{(N)} - Q^{(N-1)}$ is not a function of few summary statistics, which can be updated when a new data point is added to the cluster. In fact, $\bar{\mathbf{x}}_N$ is changing with N and $Q^{(N)}$ must be then recomputed, due to the quadratic term in its definition, using all the data in the cluster when a new point is added. To overcome this

problem, we prove in the following section the existence of two unbiased estimators for $tr[\Sigma^2]$ and $tr[\Sigma(\Sigma - D_\Sigma)]$ based on the following statistics Q_N:

$$Q_N = \begin{cases} \left((\mathbf{x}_2 - \mathbf{x}_1)^\top (\mathbf{x}_2 - \mathbf{x}_1)\right)^2 & \text{if } N = 2; \\ Q_{N-1} + \left((\mathbf{x}_N - \overline{\mathbf{x}}_{N-1})^\top (\mathbf{x}_N - \overline{\mathbf{x}}_{N-1})\right)^2 & \text{if a new point } \mathbf{x}_N \text{ is added to a cluster of } N-1 \text{ points;} \\ Q_{N_1} + Q_{N_2} & \text{if a cluster is made by merging two clusters of } N_1 \text{ and } N_2 \text{ points.} \end{cases} \tag{4}$$

The key point is that Q_N is defined recursively, and it is a function of Q_{N-1}, the new added point, and the centroid of the cluster at the time of the update.

In the following we will describe the details of our method and the assumptions that must be satisfied to apply it.

A model for the estimate of the covariance matrices

Our dataset is given by a sequence of p-dimensional vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots$. Each observation $\mathbf{x}_n$ is independent on the others and, if belonging to the cluster $\underline{k}$, it is generated as

$$\mathbf{x}_n = \boldsymbol{\mu}_{\underline{k}} + \sum_{\underline{k}}^{\frac{1}{2}} \mathbf{z}_n$$

where $\boldsymbol{\mu}_{\underline{k}}$ is the mean vector and $\Sigma_{\underline{k}}^{\frac{1}{2}}$ is a matrix such that $\Sigma_{\underline{k}} = \Sigma_{\underline{k}}^{\frac{1}{2}} (\Sigma_{\underline{k}}^{\frac{1}{2}})^\top$ is strictly positive definite. The following hypothesis of uncorrelation is assumed on the first four moments:

$$E[\mathbf{z}_n] = \mathbf{0}, \qquad \mathrm{Cov}(\mathbf{z}_n) = E[\mathbf{z}_n \mathbf{z}_n^\top] = \mathbf{I}, \qquad E\left[\prod_{i=1}^{q} z_{n,i}^{\gamma_i}\right] = \prod_{i=1}^{q} E\left[z_{n,i}^{\gamma_i}\right], \tag{5}$$

for any integers $\gamma_1, \ldots, \gamma_q$ satisfying $0 \leq \sum_1^q \gamma_i \leq 4$, and where $z_{n,i}$ is the ith component of the vector $\mathbf{z}_n = (z_{n,1}, \ldots, z_{n,q})^\top$.

Assume that the sequence $\mathbf{x}_1, \mathbf{x}_2, \ldots$ belongs to the same cluster with $\Sigma_{\underline{k}}^{\frac{1}{2}} = \Sigma^{\frac{1}{2}}$. Then the sequence $\mathbf{y}_1, \mathbf{y}_2, \ldots$ defined as $\mathbf{y}_n = \mathbf{x}_n - \boldsymbol{\mu}_{\underline{k}} = \Sigma^{\frac{1}{2}} \mathbf{z}_n$, is formed by independent vectors with null expectation. Then, as a consequence of (5), we have that

$$E[\mathbf{y}_i^\top \mathbf{y}_j] = \begin{cases} E[\mathbf{z}_i^\top (\Sigma^{\frac{1}{2}})^\top \Sigma^{\frac{1}{2}} \mathbf{z}_i] = \mathrm{tr}((\Sigma^{\frac{1}{2}})^\top \Sigma^{\frac{1}{2}}) = \mathrm{tr}(\Sigma) & \text{if } i = j; \\ 0 & \text{otherwise.} \end{cases}$$

Moreover, $E[\mathbf{y}_i^\top \mathbf{y}_j \mathbf{y}_k^\top \mathbf{y}_l] \neq 0$ only in the following situation: when $i = j = k = l$ then

$$E[\mathbf{y}_i^\top \mathbf{y}_j \mathbf{y}_k^\top \mathbf{y}_l] = E[(\mathbf{y}_i^\top \mathbf{y}_i)^2] = \kappa_{11} + 2\mathrm{tr}(\Sigma^2) + (\mathrm{tr}\Sigma)^2, \tag{6a}$$

where κ_{11} is defined in [8] as

$$\kappa_{11:} = E[\mathbf{z}_i' \Sigma \mathbf{z}_i \mathbf{z}_i' \Sigma \mathbf{z}_i] - 2\mathrm{tr}(\Sigma^2) - (\mathrm{tr}\Sigma)^2.$$

Note that $\kappa_{11} = 0$ for gaussian data, thus it is an indicator of deviation from gaussianity in terms of kurtosis. In case of gaussian data its estimation can be neglected [4].

When $(i = j) \neq (k = l)$ then

$$E[\mathbf{y}_i^\top \mathbf{y}_j \mathbf{y}_k^\top \mathbf{y}_l] = E[(\mathbf{y}_i^\top \mathbf{y}_i)(\mathbf{y}_k^\top \mathbf{y}_k)] = E[(\mathbf{y}_i^\top \mathbf{y}_i)]E[(\mathbf{y}_k^\top \mathbf{y}_k)] = (\mathrm{tr}\Sigma)^2; \tag{6b}$$

when $(i = l) \neq (j = k)$ then

$$\begin{aligned} E[\mathbf{y}_i^\top \mathbf{y}_j \mathbf{y}_k^\top \mathbf{y}_l] &= E[\mathbf{z}_i^\top (\Sigma^{\frac{1}{2}})^\top \Sigma^{\frac{1}{2}} (\mathbf{z}_j \mathbf{z}_j^\top)(\Sigma^{\frac{1}{2}})^\top \Sigma^{\frac{1}{2}} \mathbf{z}_i] \\ &= E[\mathbf{z}_i^\top (\Sigma^{\frac{1}{2}})^\top \Sigma^{\frac{1}{2}} E[\mathbf{z}_j \mathbf{z}_j^\top](\Sigma^{\frac{1}{2}})^\top \Sigma^{\frac{1}{2}} \mathbf{z}_i] \\ &= E[\mathbf{z}_i^\top (\Sigma^{\frac{1}{2}})^\top \Sigma^{\frac{1}{2}} (\Sigma^{\frac{1}{2}})^\top \Sigma^{\frac{1}{2}} \mathbf{z}_i] \\ &= \mathrm{tr}((\Sigma^{\frac{1}{2}})^\top \Sigma^{\frac{1}{2}} (\Sigma^{\frac{1}{2}})^\top \Sigma^{\frac{1}{2}}) = \mathrm{tr}(\Sigma^{\frac{1}{2}} (\Sigma^{\frac{1}{2}})^\top \Sigma^{\frac{1}{2}} (\Sigma^{\frac{1}{2}})^\top) = \mathrm{tr}(\Sigma^2); \end{aligned} \tag{6c}$$

when $(i = k) \neq (j = l)$ the same as above, since $\mathbf{y}_k^\top \mathbf{y}_l = \mathbf{y}_l^\top \mathbf{y}_k$, hence

$$E[\mathbf{y}_i^\top \mathbf{y}_j \mathbf{y}_k^\top \mathbf{y}_l] = \mathrm{tr}(\Sigma^2). \tag{6d}$$

Lemma 1 *As a consequence of* (6d),

$$E\left[\left(\mathbf{y}_N^\top \sum_{i=1}^{N-1} \mathbf{y}_i\right)^2\right] = (N-1)\mathrm{tr}(\Sigma^2).$$

Lemma 2 *As a consequence of all the relations* (6),

$$E\left[\sum_{i,j,k,l=1}^{N-1} \mathbf{y}_i^\top \mathbf{y}_j \mathbf{y}_k^\top \mathbf{y}_l\right] = (N-1)\kappa_{11} + 2(N-1)^2\mathrm{tr}(\Sigma^2) + (N-1)^2(\mathrm{tr}\Sigma)^2.$$

Lemma 3 *As a consequence of* (6b),

$$E\left[\sum_{i,j=1}^{N-1} \mathbf{y}_N^\top \mathbf{y}_N \mathbf{y}_i^\top \mathbf{y}_j\right] = (N-1)(\mathrm{tr}\Sigma)^2.$$

Optimal shrinkage estimation

We now use the previous results to solve the problem of finding the optimal estimates of $\hat{\lambda}_I, \hat{\lambda}_D$ in (1), as a function of the statistics S (sample covariance matrix of the data in the same cluster), of Q_N given in (4), and of two quantities $\mathbb{S}_N$ and $\mathbb{T}_N$ that can be updated inductively. As can be seen in (2), the problem here is the unbiased estimation of the terms $\mathrm{tr}[\Sigma^2]$ and $\mathrm{tr}(\Sigma^2) - \mathrm{tr}(D_\Sigma^2)$. The derivation of this estimate is given in the next section, after a technical result given hereafter.

We may use the following additional relations in our estimates [8, 10]

$$E[\mathrm{tr}(S^2)] = \frac{1}{N}\kappa_{11} + \frac{N}{N-1}\mathrm{tr}(\Sigma^2) + \frac{1}{N-1}(\mathrm{tr}\Sigma)^2 \tag{7a}$$

$$E[(\mathrm{tr}S)^2] = \frac{1}{N}\kappa_{11} + \frac{2}{N-1}\mathrm{tr}(\Sigma^2) + (\mathrm{tr}\Sigma)^2 \tag{7b}$$

$$E[\mathrm{tr}(D_S^2)] = \frac{1}{N-1}\kappa_{11} + \frac{N+1}{N-1}\mathrm{tr}(D_\Sigma^2) + \frac{R_N}{N-1}, \tag{7c}$$

once we have recalled that the quantity R_N is negligible (see, again, [10, 18]). When the data are distributed as gaussians, a direct estimation without κ_{11} based on (7a–c) may be done (see [4]), since $\kappa_{11} = 0$.

When this is not the case, we may use the statistics Q_N already introduced in (4) and we will prove in Lemma 4 that

$$E[Q_N] = \mathbb{S}_N \kappa_{11} + \mathbb{T}_N (2\mathrm{tr}(\Sigma^2) + (\mathrm{tr}\Sigma)^2), \tag{8}$$

where $\mathbb{S}_N$ and $\mathbb{T}_N$ are two quantities that may be simply calculated inductively as:

$$\mathbb{S}_N = \begin{cases} 2 & \text{if } N = 2; \\ \mathbb{S}_{N-1} + \left(1 + \frac{1}{(N-1)^3}\right) & \text{if a new point is added} \\ & \text{to a cluster of } N-1 \text{ points}; \\ \mathbb{S}_{N_1} + \mathbb{S}_{N_2} & \text{if a cluster is made by merging} \\ & \text{two clusters of } N_1 \text{ and } N_2 \text{ points}; \end{cases} \tag{9}$$

$$\mathbb{T}_N = \begin{cases} 4 & \text{if } N = 2; \\ \mathbb{T}_{N-1} + \left(1 + \frac{1}{(N-1)}\right)^2 & \text{if a new point is added} \\ & \text{to a cluster of } N-1 \text{ points}; \\ \mathbb{T}_{N_1} + \mathbb{T}_{N_2} & \text{if a cluster is made by merging} \\ & \text{two clusters of } N_1 \text{ and } N_2 \text{ points}; \end{cases} \tag{10}$$

Lemma 4 *With the notations of* (3), (4), (9) *and* (10) *we have*

$$E[Q_N] = \begin{cases} 2\kappa_{11} + 4(2\mathrm{tr}(\Sigma^2) + (\mathrm{tr}\Sigma)^2) & \text{if } N = 2; \\ E[Q_{N-1}] + \left(1 + \frac{1}{(N-1)^3}\right)\kappa_{11} & \text{if a new point is added} \\ \quad + (1 + \frac{1}{N-1})^2 (2\mathrm{tr}(\Sigma^2) + (\mathrm{tr}\Sigma)^2) & \quad \text{to a cluster of } N-1 \text{ points}; \\ E[Q_{N_1}] + E[Q_{N_2}] & \text{if a cluster is made by merging} \\ & \quad \text{two clusters of } N_1 \text{ and } N_2 \text{ points}; \end{cases}$$

and hence

$$E[Q_N] = \mathbb{S}_N \kappa_{11} + \mathbb{T}_N (2\mathrm{tr}(\Sigma^2) + (\mathrm{tr}\Sigma)^2).$$

See Appendix for the proof.

Unbiased estimators of $\mathrm{tr}(\Sigma^2)$ and $\mathrm{tr}(\Sigma^2) - \mathrm{tr}(D_\Sigma^2)$

Let $\mathbf{X} = (\mathrm{tr}(S^2), (\mathrm{tr}S)^2, \mathrm{tr}(D_S^2), Q_N)^\top$ and $\mathbf{Y} = (\kappa_{11}, \mathrm{tr}(\Sigma^2), (\mathrm{tr}\Sigma)^2, \mathrm{tr}(D_\Sigma^2))^\top$. We are interested in an unbiased estimator of the vector

$$\mathbf{Z} = \begin{pmatrix} \mathrm{tr}(\Sigma^2) \\ \mathrm{tr}(\Sigma^2) - \mathrm{tr}(D_\Sigma^2) \end{pmatrix} = B\mathbf{Y}, \quad \text{where } B = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix}.$$

The system composed by (7a–c) and (8) may be read as

$$E(\mathbf{X}) = A\mathbf{Y}, \quad \text{where } A = \begin{pmatrix} \frac{1}{N} & \frac{N}{N-1} & \frac{1}{N-1} & 0 \\ \frac{1}{N} & \frac{2}{N-1} & 1 & 0 \\ \frac{1}{N-1} & 0 & 0 & \frac{N+1}{N-1} \\ \mathbb{S}_N & 2\mathbb{T}_N & \mathbb{T}_N & 0 \end{pmatrix}.$$

Now, the matrix A may be shown to be invertible, and hence $\hat{\mathbf{Z}} = BA^{-1}\mathbf{X}$ is a linear (in $\mathbf{X}$) unbiased estimator for $\mathbf{Z}$, since

$$E(\hat{\mathbf{Z}}) = E(BA^{-1}\mathbf{X}) = BA^{-1}E(\mathbf{X}) = BA^{-1}A\mathbf{Y} = B\mathbf{Y} = \mathbf{Z}.$$

For sake of completeness, we give here the elements of the matrix $BA^{-1} = [C_{kl}]_{\substack{k=1,2\\l=1,\dots,4}}$. Let $K = (N+2+\frac{2}{N-1})\mathbb{S}_N - 3\mathbb{T}_N$, we have

$$\begin{aligned}
C_{11} &= \frac{(N-1)(N\mathbb{S}_N - \mathbb{T}_N)}{K(N-2)};\\
C_{12} &= \frac{N\mathbb{S}_N - (N-1)\mathbb{T}_N}{K(N-2)};\\
C_{13} &= 0;\\
C_{14} &= \frac{1}{K};\\
C_{21} &= \frac{(N+1+\frac{2}{N-2})\mathbb{S}_N - (3+\frac{1}{N-2}-\frac{2}{N+1})\mathbb{T}_N}{K};\\
C_{22} &= \frac{-(1+\frac{2}{N-2})\mathbb{S}_N + (\frac{1}{N-2}+\frac{1}{N+1})\mathbb{T}_N}{K};\\
C_{23} &= -1 + \frac{2}{N+1};\\
C_{24} &= \frac{\frac{1}{N-1}}{K}.
\end{aligned}$$

Summary statistics and primary data compression

In this section we define the summary statistics that will be retained in memory for each cluster and we describe the first phase of our clustering procedure. As in the BFR algorithm, we first perform the *primary data compression*, that is the identification of items which can be assigned to a cluster, and then discarded (Discard Set, DS), after updating the corresponding summary statistics contained in the Compression Set CS. Data compression refers thus to representing groups of points by their summary statistics and purging these points from RAM. In our algorithm, like in BFR, primary data compression will be followed by a secondary data-compression, which takes place over data points in the Retained Set (RS), not compressed in the primary phase.

Assume that data points $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathbb{R}^p, N \geq 2$ must be compressed in the same cluster. We will retain only the following summary statistics

$$\Sigma_N = \sum_{i=1}^{N} \mathbf{x}_i \mathbf{x}_i^\top, \quad \mathbf{s}_N = \sum_{i=1}^{N} \mathbf{x}_i, \quad N, \tag{11}$$

and the statistics $Q_N, \mathbb{S}_N, \mathbb{T}_N$ defined in (4), (9), (10), respectively.

In particular the statistics $\mathbf{s}_N$ are needed to compute the sample means $\bar{\mathbf{x}}_N = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i$, that are used as clusters centers, while the matrices Σ_N are used to compute the unbiased sample covariance matrices of the clusters $S = \frac{1}{N-1}\sum_{i=1}^{n}(\mathbf{x} - \bar{\mathbf{x}}_N)(\mathbf{x} - \bar{\mathbf{x}}_N)^\top$, which are needed, together with $Q_N, \mathbb{S}_N, \mathbb{T}_N$, to compute the optimal double shrinkage estimators described in the previous section.

The summary statistics (11) can also be easily updated when a new data point $\mathbf{x}_{N+1}$ must be added to the cluster, without processing again the already compressed points. In fact

$$\Sigma_{N+1} = \Sigma_N + \mathbf{x}_{N+1}\mathbf{x}_{N+1}^\top, \quad \mathbf{s}_{N+1} = \mathbf{s}_N + \mathbf{x}_{N+1},$$

while the other summary statistics have already been defined recursively.

Note that the matrix Σ_N is symmetric, thus at each step of the algorithm we have to retain in memory only $\frac{p(p+1)}{2} + p + 4 = \frac{p^2}{2} + \frac{3}{2}p + 4$ summary statistics for each cluster, where p is the dimension of the data points. Thus, in case of K clusters, our computational costs are of the order of Kp^2. In addition, note that we should simply sum the corresponding statistics if we want to merge two clusters.

Similarly to the BFR algorithm, in order to assign a point to a cluster we use the squared Mahalanobis distance from its center (sample mean), i.e. we assign a new data point $\mathbf{x}$ to cluster h with center $\bar{\mathbf{x}}_h$ and estimated covariance matrix $\hat{S}_h$, if h is the index which minimizes

$$\Delta^2_{\hat{S}_h}(\mathbf{x}, \bar{\mathbf{x}}_h) = (\mathbf{x} - \bar{\mathbf{x}}_h)^T (\hat{S}_h)^{-1} (\mathbf{x} - \bar{\mathbf{x}}_h).$$

Differently from the BFR algorithm, here we estimate the covariance matrices of the clusters with the optimal double shrinkage estimators described in the previous section. In order to avoid the inversion of a matrix and thus to reduce the computational costs, we observe that the Mahalanobis distance between two points $\mathbf{x}, \mathbf{y}$, computed with respect to a covariance matrix S, can be rewritten as follows (see e.g. [17, Expression A.7.10]):

$$\Delta^2_S(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T S^{-1} (\mathbf{x} - \mathbf{y}) = \frac{\det[S + (\mathbf{x} - \mathbf{y})(\mathbf{x} - \mathbf{y})^T]}{\det(S)} - 1 \tag{12}$$

In our algorithm we will actually use expression (12) for the computation of all the Mahalanobis distances.

We also compare $\mathbf{x}$ with each point $\mathbf{x}_o$ in the retained set RS, if any, by computing

$$\Delta^2_{\hat{S}_P}(\mathbf{x}, \mathbf{x}_o) = (\mathbf{x} - \mathbf{x}_o)^T (\hat{S}_P)^{-1} (\mathbf{x} - \mathbf{x}_o),$$

where $\hat{S}_P$ matrix is the pooled covariance matrix based on $\hat{S}_h$ of all the K clusters:

$$\hat{S}_P = \frac{n_{h_1}\hat{S}_{h_1} + n_{h_2}\hat{S}_{h_2} + \cdots n_{h_K}\hat{S}_{h_K}}{n_{h_1} + n_{h_2} + \cdots + n_{h_K}}, \tag{13}$$

and where n_h is the number of points in cluster h. With $\hat{S}_P$, we emphasize the weighted importance of directions that are more significant for the clusters when we compute the distance between two "isolated" points. Since the retained set contains the points which do not belong clearly to one specific cluster, with this comparison we check if they can be aggregated with the new incoming data, to form new clusters.

We then approximate locally the distribution of the clusters with a p-variate Gaussian and we build confidence regions around the centers of the clusters (see [9]).

Following the approach stated in [3], which is motivated by the assumption that the mean is unlikely to move outside of the computed confidence interval, we perturb $\bar{\mathbf{x}}_h$ by moving it in the farthest position from $\mathbf{x}$ in its confidence region, while we perturb the centers of the other clusters by moving them in the closest positions with respect to $\mathbf{x}$ and we check if the cluster center closer to $\mathbf{x}$ is still $\bar{\mathbf{x}}_h$. If yes, we assign $\mathbf{x}$ to cluster h, we update the corresponding summary statistics and we put $\mathbf{x}$ in the discard set; otherwise, we put $\mathbf{x}$ in the retained set (RS) (see Fig. 2). If in the first comparisons the point $\mathbf{x}$ is closer to a point $\mathbf{x}_o$ of the retained set than to any cluster, we form a new secondary cluster with the two points if $\mathbf{x}_o$ remains the closest to $\mathbf{x}$ after the centers' perturbation. In this case we add the corresponding summary statistics to the compressed set CS, and we put $\mathbf{x}$ and $\mathbf{x}_o$ in the discard set. Otherwise we put $\mathbf{x}$ and $\mathbf{x}_o$ in RS (see Fig. 3).

Let us see the procedure of centers' perturbation in deeper detail.

Confidence regions

It is well-known [9] that a confidence region for the mean $\boldsymbol{\mu}$ based on $\overline{\mathbf{x}}$ and $\hat{S}$ may be based on the Hotelling's T-squared distribution

$$t^2 = n(\overline{\mathbf{x}} - \boldsymbol{\mu})^\top S^{-1}(\overline{\mathbf{x}} - \boldsymbol{\mu}) \sim T^2_{p,n-1} = \frac{p(n-1)}{n-p} F_{p,n-p},$$

where $F_{p,n-p}$ is the F-distribution with parameters p and $n-p$.

Then, if we denote by $CI_{\underline{k}}$ the confidence region for the mean of cluster $\underline{k}$, i.e.

$$CI_{\underline{k}} = \{\boldsymbol{\mu} : n(\overline{\mathbf{x}}_{\underline{k}} - \boldsymbol{\mu})^\top \hat{S}^{-1}_{\underline{k}}(\overline{\mathbf{x}}_{\underline{k}} - \boldsymbol{\mu}) \leq T^2_{p,n-1}(1-\alpha)\}$$

then the perturbation $p_{\underline{k}}(\mathbf{x})$ for the data point $\mathbf{x}$ is

$$p_{\underline{k}}(\mathbf{x}) = \begin{cases} \sup_{\boldsymbol{\mu} \in CI_{\underline{k}}} (\mathbf{x} - \boldsymbol{\mu})^\top \hat{S}^{-1}_{\underline{k}}(\mathbf{x} - \boldsymbol{\mu}) & \text{if } \underline{k} = j; \\ \inf_{\boldsymbol{\mu} \in CI_{\underline{k}}} (\mathbf{x} - \boldsymbol{\mu})^\top \hat{S}^{-1}_{\underline{k}}(\mathbf{x} - \boldsymbol{\mu}) & \text{if } \underline{k} \neq j; \end{cases}$$

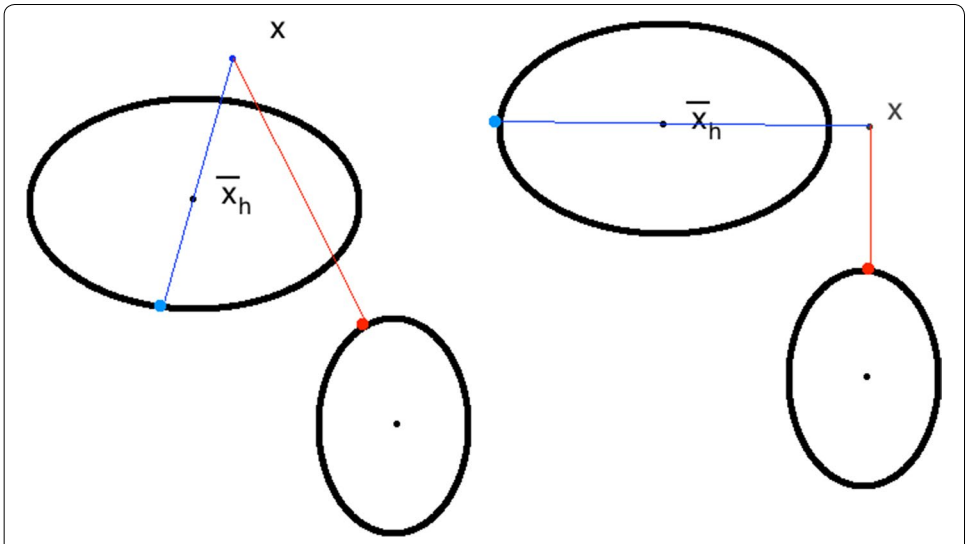

Fig. 2 The two possible situations after the perturbation of the clusters centers. Left: the point $\mathbf{x}$ is assigned to cluster h; right: $\mathbf{x}$ is moved to RS

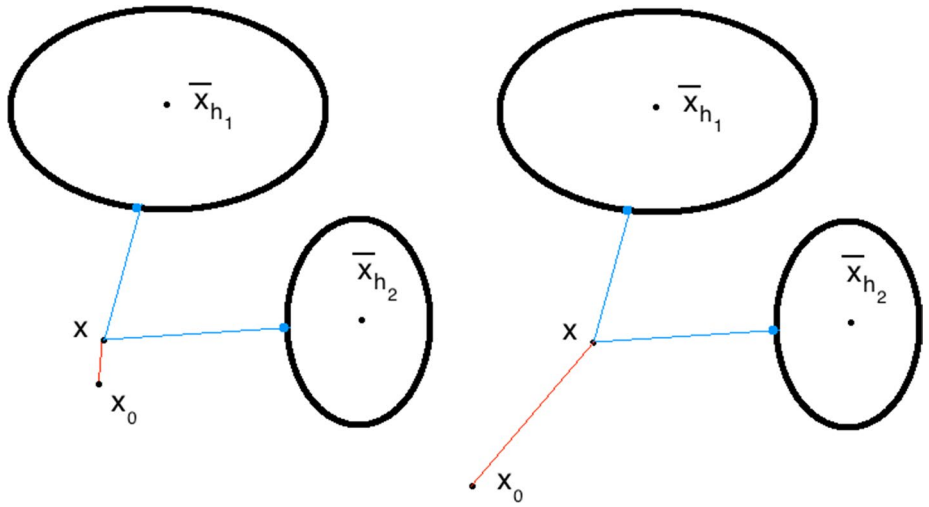

Fig. 3 The two possible situations, after the centers' perturbation, when $\mathbf{x}$ is closer to a point $\mathbf{x}_o$ of the retained set in the first comparison. Left: the points $\mathbf{x}$ and $\mathbf{x}_o$ are joined to form a new cluster; right: $\mathbf{x}$ and $\mathbf{x}_o$ are moved to RS

Denoting by $t_\alpha = T^2_{p,n-1}(1-\alpha)$, if we introduce a Lagrange multiplier λ^*, the problems of minimization or maximization stated in the definition of $p_{\underline{k}}(\mathbf{x})$ can be solved by differentiating the following lagrangian form $\mathcal{L}$:

$$\mathcal{L}(\boldsymbol{\mu},\lambda) = (\mathbf{x}-\boldsymbol{\mu})^\top \hat{S}_{\underline{k}}^{-1}(\mathbf{x}-\boldsymbol{\mu}) - n\lambda^*\big((\overline{\mathbf{x}}_{\underline{k}}-\boldsymbol{\mu})^\top \hat{S}_{\underline{k}}^{-1}(\overline{\mathbf{x}}_{\underline{k}}-\boldsymbol{\mu}) - \tfrac{t_\alpha}{n}\big).$$

The resolution $\nabla_{\boldsymbol{\mu}}\mathcal{L} = \mathbf{0}$ gives $\boldsymbol{\mu} = \frac{\mathbf{x}-\lambda\overline{\mathbf{x}}_{\underline{k}}}{1-\lambda}$, where $\lambda = n\lambda^*$. In particular, the optimal $\boldsymbol{\mu}$ is the linear combination of $\mathbf{x}$ and $\overline{\mathbf{x}}_{\underline{k}}$ in $CI_{\underline{k}}$ which is farther from $\mathbf{x}$ or closer to $\mathbf{x}$, when $\underline{k} = j$ or $\underline{k} \neq j$, respectively. The constrain reads

$$\left(\overline{\mathbf{x}}_{\underline{k}} - \frac{\mathbf{x}-\lambda\overline{\mathbf{x}}_{\underline{k}}}{1-\lambda}\right)^\top \hat{S}_{\underline{k}}^{-1}\left(\overline{\mathbf{x}}_{\underline{k}} - \frac{\mathbf{x}-\lambda\overline{\mathbf{x}}_{\underline{k}}}{1-\lambda}\right) = \frac{t_\alpha}{n} \quad\Longrightarrow\quad \frac{t_\alpha}{n} = \frac{(\overline{\mathbf{x}}_{\underline{k}}-\mathbf{x})^\top \hat{S}_{\underline{k}}^{-1}(\overline{\mathbf{x}}_{\underline{k}}-\mathbf{x})}{(1-\lambda)^2}.$$

Denoting by $\Delta^2_{\underline{k},\mathbf{x}} = (\overline{\mathbf{x}}_{\underline{k}}-\mathbf{x})^\top \hat{S}_{\underline{k}}^{-1}(\overline{\mathbf{x}}_{\underline{k}}-\mathbf{x})$, we have $\lambda = 1 \pm \sqrt{n\Delta^2_{\underline{k},\mathbf{x}}/t_\alpha}$ and

$$p_{\underline{k}}(\mathbf{x}) = \left(\mathbf{x} - \frac{\mathbf{x}-\lambda\overline{\mathbf{x}}_{\underline{k}}}{1-\lambda}\right)^\top \hat{S}_{\underline{k}}^{-1}\left(\mathbf{x} - \frac{\mathbf{x}-\lambda\overline{\mathbf{x}}_{\underline{k}}}{1-\lambda}\right) = \frac{\lambda^2}{(1-\lambda)^2}\Delta^2_{\underline{k},\mathbf{x}}.$$

Summarizing we obtain the following perturbations of the clusters centers, referred to the data point $\mathbf{x}$,

$$p_{\underline{k}}(\mathbf{x}) = \begin{cases} \left(\sqrt{\Delta^2_{\underline{k},\mathbf{x}}} + \sqrt{t_\alpha/n}\right)^2 & \text{if } \underline{k} = j; \\ \left(\sqrt{\Delta^2_{\underline{k},\mathbf{x}}} - \sqrt{t_\alpha/n}\right)^2 & \text{if } \underline{k} \neq j \text{ and } \Delta^2_{\underline{k},\mathbf{x}} \geq t_\alpha. \\ 0 & \text{if } \underline{k} \neq j \text{ and } \Delta^2_{\underline{k},\mathbf{x}} < t_\alpha. \end{cases}$$

Secondary data compression

The purpose of secondary data compression is to identify "tight" sub-clusters of points among the data that we cannot discard in the primary phase. In [3] this is made using the euclidean metric.

We adopt a similar idea, but we use a local metric, based on the Mahalanobis distance. We exploit a technique based on hierarchical clustering, mimicking the Ward's method [6, 13].

Given two clusters h_1 and h_2 with $n_{h_1} \geq 2, n_{h_2} \geq 2$ points, and centroids $\bar{\mathbf{x}}_{h_1}$ and $\bar{\mathbf{x}}_{h_2}$, respectively, then the squared Mahalanobis distance of one centroid to the other cluster may be measured as $\Delta^2_{\hat{S}_{h_i}}(\bar{\mathbf{x}}_{h_1}, \bar{\mathbf{x}}_{h_2})$, $i = 1, 2$. Accordingly, to decide whether two clusters are close or not, we compare the weighted combination of those distances

$$\Delta^2_{h_1,h_2} = \frac{\mathrm{tr}(\hat{S}_{h_1})}{\mathrm{tr}(\hat{S}_{h_1}) + \mathrm{tr}(\hat{S}_{h_2})} \Delta^2_{\hat{S}_{h_1}}(\bar{\mathbf{x}}_{h_1}, \bar{\mathbf{x}}_{h_2}) + \frac{\mathrm{tr}(\hat{S}_{h_2})}{\mathrm{tr}(\hat{S}_{h_1}) + \mathrm{tr}(\hat{S}_{h_2})} \Delta^2_{\hat{S}_{h_2}}(\bar{\mathbf{x}}_{h_1}, \bar{\mathbf{x}}_{h_2}),$$

with the the squared Mahalanobis distance $\Delta^2_{\hat{S}_{h_1 h_2}}(\bar{\mathbf{x}}_{h_1}, \bar{\mathbf{x}}_{h_2})$ of the two centroids, evaluated with the pooled covariance matrix of the two clusters $\hat{S}_{h_1 h_2} = \frac{n_{h_1}\hat{S}_{h_1} + n_{h_2}\hat{S}_{h_2}}{n_{h_1}+n_{h_2}}$.

The distance between a single retained point $\mathbf{x}$ and a cluster h is computed by the squared Mahalanobis distance $\Delta^2_{\hat{S}_h}(\mathbf{x}, \bar{\mathbf{x}}_h)$ between the point and the cluster centroid, based on the estimated covariance matrix of the cluster, while the distance between two retained points $\mathbf{x}_1, \mathbf{x}_2$ is computed by their squared Mahalanobis distance $\Delta^2_{\hat{S}_P}(\mathbf{x}_1, \mathbf{x}_2)$ based on the pooled covariance matrix (13) of all the clusters.

Based on the hierarchical tree built with such distances, we sequentially merge two clusters or points only if a suitable density condition is fulfilled. This condition is different for the different types of merging that we can perform:

- We merge two clusters h_1 and h_2 if $\Delta^2_{h_1,h_2} < \theta_0 \Delta^2_{\hat{S}_{h_1 h_2}}(\bar{\mathbf{x}}_{h_1}, \bar{\mathbf{x}}_{h_2})$;
- We merge a retained point $\mathbf{x}$ and a cluster h if $\Delta^2_{\hat{S}_h}(\mathbf{x}, \bar{\mathbf{x}}_h) < \theta_1(tr(S_h))$;
- We merge two retained points $\mathbf{x}_1$ and $\mathbf{x}_2$ if $\Delta^2_{\hat{S}_P}(\mathbf{x}_1, \mathbf{x}_2) < \theta_2$.

Here θ_i, $i = 0, 1, 2$, are thresholds, chosen by the user. For what concerns θ_2, we suggest to use a significant quantile of the χ-square distribution that arises under the null hypothesis

H_0: the retained points come from a gaussian distribution with covariance matrix given by the pooled covariance matrix (13) of all the clusters.

Results on simulated and real data and discussion

Results on synthetic data

Synthetic data were created for the cases of 5 and 20 clusters. Data were sampled from 5 or 20 independent p-variate Gaussians, with elements of their mean vectors (the true means) uniformly distributed on $[-5, 5]$. The covariance matrices were generated by computing products of the type $\Sigma = UHU^T$, where H is a diagonal matrix with elements on the diagonal distributed as a *Beta* (0.5, 0.5) rescaled to the interval [0.5, 2.5], and U is the orthonormal matrix obtained by the singular value decomposition of a symmetric

matrix MM^T, where the elements of the $p \times p$ matrix M are uniformly distributed on $[-2, 2]$. In either cases of 5 or 20 clusters, we generated 10,000 vectors for each cluster, having dimensions $p = 5, 10, 20$.

This procedure guarantees that these clusters are rather well-separated Gaussians, in particular for higher vector dimensions.

We applied both our procedure and the BFR algorithm to these synthetic data, to compare the performance of the two methods. In both cases, we computed the secondary data compression once out of 25, or out of 50 data points. In the tests on data from 20 clusters we started from a lower number of initial clusters (equal to 10), in order to check the ability of our algorithm to detect the correct number of clusters. The results are reported in Table 1.

We note that the number of clusters is sometimes underestimated by our method, in particular in the case of 20 clusters. In such cases, if the point clouds in different clusters are gathered in rather close ellipsoids, then the correct detection of the clusters may be more difficult. Anyway in all cases the estimates provided by our algorithm are equal or better than those obtained with the BFR algorithm.

We also point out that in the case of 20 clusters with $p = 10$, and secondary compression performed once out of 50 processed data, the overestimation of the number of clusters obtained with our algorithm is compensated by the presence of two small clusters, composed by a few hundreds of data points, which can then be revisited as groups of outliers. Anyway also in this case our results are better than those obtained with BFR.

Table 1 Results of the application of our proposed algorithm (PA) and of the BFR algorithm to synthetic data

N. of true clusters	Algorithm	Dimension *p* of data points	N. of data in each chunk	N. of estimated clusters	N. of retained points (outliers)
5	BFR	5	25	6	0
5	PA	5	25	5	0
5	BFR	5	50	6	0
5	PA	5	50	5	0
5	BFR	10	25	5	0
5	PA	10	25	5	0
5	BFR	10	50	5	0
5	PA	10	50	5	0
5	BFR	20	25	5	0
5	PA	20	25	5	0
5	BFR	20	50	5	0
5	PA	20	50	5	0
20	BFR	10	25	12	0
20	PA	10	25	17	0
20	BFR	10	50	13	0
20	PA	10	50	22	1
20	BFR	20	25	11	0
20	PA	20	25	19	0
20	BFR	20	50	20	0
20	PA	20	50	20	0

We call chunk the number of processed data out of which we apply secondary compression

The method seems to be sensitive to the frequency of the secondary compression only in presence of many clusters.

Note that our method gives always a correct estimation of the number of clusters in all cases with five true clusters, while the BFR method overestimates the correct number in particular when the data dimension is small ($p = 5$). This is reasonable since in lower dimensional spaces the shape and orientation of the point clouds must be correctly estimated and taken into account to identify the clusters in a proper way (see Fig. 4).

We tested also cases with bigger values of p, but in such cases both algorithms are able to detect the correct number of clusters, in an equivalent way, since a few clusters in high dimensional spaces are almost always well separated, because of "curse of dimensionality" reasons.

Results on a real dataset

We applied our algorithm to a real dataset to detect network intrusions. Detecting intrusions is a typical data streaming problem, since it is essential to identify the event while it is happening. In our experiments we used the KDD-CUP'99 (http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html) intrusion detection dataset which consists of 2 weeks of raw TCP dump data. This dataset is related to a local area network simulating a true Air Force environment with occasional attacks. Variables collected for each connection include the duration of the connection, the number of bytes transmitted from source to destination (and viceversa), the number of failed login attempts, etc. We applied our algorithm to the 34 variables that are declared to be continuous.

Some of these variables actually are almost constant, giving thus an estimated zero sample variance in many clusters. In such situation, if the BFR algorithm is applied, singular covariance matrices are estimated for some clusters. Consequently the Mahalanobis distance becomes unstable. Our optimal double shrinkage estimators are thus necessary to overcome this instability, and as a byproduct, they can take into account the deviation of the kurtosis from the Gaussian case.

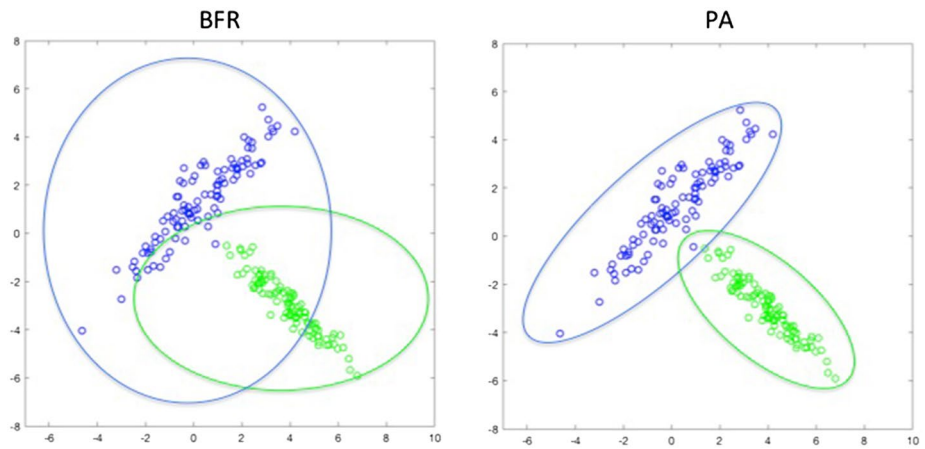

Fig. 4 Example of a typical situation in 2D: the BFR algorithm is approximating the shape of the clusters using ellipsoids parallel to the main axes. Our proposed algorithm (PA) is using ellipsoids with the correct orientation, thus the uncertainty region (overlapping of the ellipsoids) is reduced

The same dataset was analysed in [16], via the STREAM algorithm, but they used the Euclidean distance, which is a global distance that gives the same importance to all the variables.

We obtained stable results. We applied the secondary compression every 100 data, starting from 4 clusters composed by less than 20 points. We observed the presence of 6–8 big clusters starting from about 100,000 processed data. We processed about 646,000 data, ending with five big clusters, composed by the following number of points: 133,028; 121,661; 242,206; 53,235; 95,977. Note that we detected the final correct number of clusters, since in this dataset there are four possible types of attacks, plus no attacks. The four types of attacks are denial-of-service; unauthorized access from a remote machine (e.g. guessing password); unauthorized access to local superuser (root) privileges; surveillance and other probing (e.g., port scanning).

In Fig. 5 we show the effectiveness of secondary compression on the stabilization of the number of clusters. Actually when the number of identified clusters is bigger than 8 after secondary compression, the exceeding ones are formed just by a few points, and can then be reinterpreted as groups of outliers. For example when 300,113 data have been processed, we find 15 clusters composed respectively by the following number of points: 95,141; 22,451; 50,098; 79,943; 30,683; 11,834; 7762; 1228; 712; 118; 100; 33; 4; 4; 2. Note that 7 out of 15 clusters are quite small, containing less than 1000 data points.

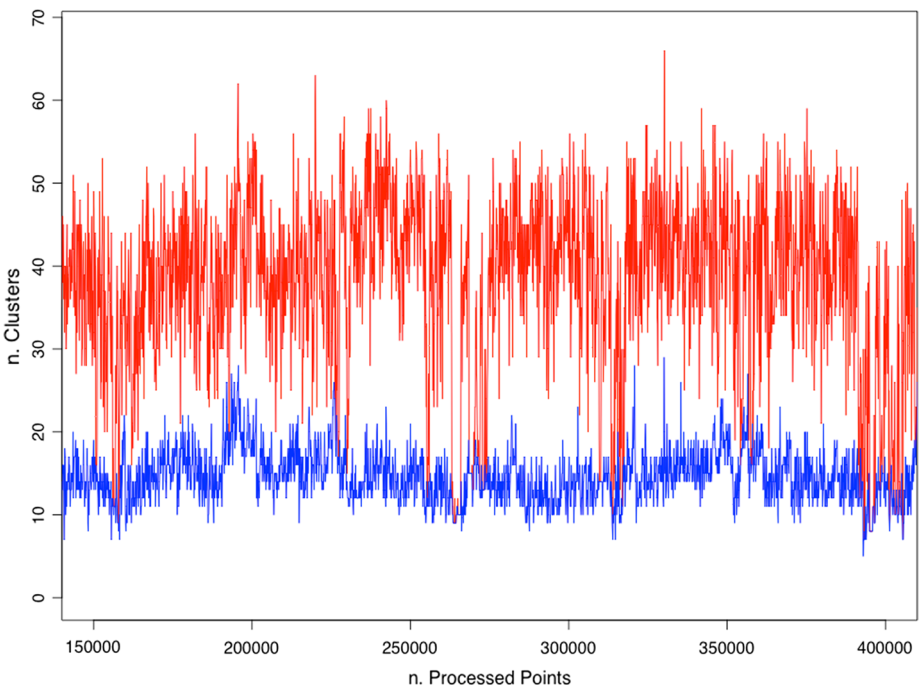

Fig. 5 Effect of the secondary compression. Red line: number of clusters obtained before secondary compression. Blue line: number of clusters obtained after secondary compression. Secondary compression is applied once out of 100 iterations

Conclusion

We have introduced a new algorithm to cluster data streams with correlated components. Our algorithm in some parts imitates the BFR algorithm, since, like BFR, it uses a local distance approach, based on the computation of the Mahalanobis distance. In order to compute such distance, positive definite estimators of the covariance matrices of the clusters are needed, also when the clusters contain just a few data points. We obtained such estimators by considering a Steinian double shrinkage method, which leads to covariance matrix estimators that are non-singular, well-conditioned, expressed in a recursive way and thus computable on data streams. Further such estimators provide positive definite estimates also when some components of the data points have a small variance, or the data distribution has a kurtosis different from the Gaussian case.

We applied both our proposed method and the BFR algorithm to synthetic gaussian data, and we compared their performance. From the numerical results we conclude that our method provides rather good clustering on synthetic data, and performs better than the BFR algorithm in particular in presence of few clusters in spaces of rather low dimension. This is reasonable since the BFR algorithm approximates the "clouds" of data with ellipsoids having axes parallel to the reference system, and this leads to a wrong classification when the clusters are elongated, not much separated, and with axes rotated with respect to the reference system. In such situations our algorithm is able to capture in a more proper way the geometry of the clusters, and thus improves the classification.

Anyway the secondary compression could be possibly improved by applying some incremental model-based technique (see [5]), but modified in such a way to avoid multiple scans of the sample.

We also applied our algorithm to a real dataset, obtaining good results in terms of correct identification of the number of clusters, and stability of our algorithm.

The advantage of our algorithm with respect to other methods present in literature, like BFR or STREAM, is that it relaxes the assumptions on the processed data streams, and can thus be effectively applied to a wider class of cases, on which it performs better. In the cases where the assumptions of the other methods are satisfied, our algorithm provides equivalent results. It can then be systematically substituted to other methods to analyze data streams, in all cases in which the data points are not too much high dimensional.

Authors' contributions
The contribution of each author is equal. Both authors read and approved the final manuscript.

Acknowledgements
G. Aletti is a member of "Gruppo Nazionale per il Calcolo Scientifico (GNCS)" of the Italian "Istituto Nazionale di Alta Matematica (INdAM)".

Competing interests
The authors declare that they have no competing interests.

Availability of data and materials
All data and materials are available upon request.

Consent for publication
Not applicable.

Ethics approval and consent to participate
Not applicable.

Funding
This work has been partially supported by the Universitá degli Studi di Milano Grant Project 2017 "Stochastic modelling, statistics and study of the invariance properties of stochastic processes with geometrical and space-time structure in applications", and by ADAMSS Center funds for Big Data research.

Appendix: Proofs

Proof of Lemma 4 For $N = 2$, as a consequence of the model:

$$\begin{aligned}
E[\big((\mathbf{x}_2 - \mathbf{x}_1)^\top(\mathbf{x}_2 - \mathbf{x}_1)\big)^2] &= E[\big((\mathbf{y}_2 - \mathbf{y}_1)^\top(\mathbf{y}_2 - \mathbf{y}_1)\big)^2] \\
&= E[(\mathbf{y}_2^\top\mathbf{y}_2)^2] + 4E[(\mathbf{y}_2^\top\mathbf{y}_1)^2] + E[(\mathbf{y}_1^\top\mathbf{y}_1)^2] + 2E[(\mathbf{y}_2^\top\mathbf{y}_2)(\mathbf{y}_1^\top\mathbf{y}_1)] \\
&\quad - 2E[(\mathbf{y}_2^\top\mathbf{y}_2)(\mathbf{y}_2^\top\mathbf{y}_1)] - 2E[(\mathbf{y}_2^\top\mathbf{y}_1)(\mathbf{y}_1^\top\mathbf{y}_1)^2] \\
&= 2E[(\mathbf{y}_1^\top\mathbf{y}_1)^2] + 4E[(\mathbf{y}_2^\top\mathbf{y}_1)^2] + 2E[(\mathbf{y}_1^\top\mathbf{y}_1)]^2 \\
&\quad - 2E[(\mathbf{y}_2^\top\mathbf{y}_2)\mathbf{y}_2^\top]E[\mathbf{y}_1] - 2E[\mathbf{y}_2^\top]E[\mathbf{y}_1(\mathbf{y}_1^\top\mathbf{y}_1)^2] \\
&= 2E[(\mathbf{y}_1^\top\mathbf{y}_1)^2] + 4E[(\mathbf{y}_2^\top\mathbf{y}_1)^2] + 2E[(\mathbf{y}_1^\top\mathbf{y}_1)]^2.
\end{aligned}$$

Since $E[(\mathbf{y}_1^\top\mathbf{y}_1)] = \mathrm{tr}\Sigma$, by (6a) and (6d), we obtain the first part of the thesis.

Let us add a point to a cluster of $N-1$ points. We obtain

$$\begin{aligned}
E[Q_N] - E[Q_{N-1}] &= E[Q_N - Q_{N-1}] \\
&= E\big[\big((\mathbf{x}_N - \overline{\mathbf{x}}^{(N)})^\top(\mathbf{x}_N - \overline{\mathbf{x}}^{(N)})\big)^2\big] = E\big[\big((\mathbf{y}_N - \overline{\mathbf{y}}^{(N)})^\top(\mathbf{y}_N - \overline{\mathbf{y}}^{(N)})\big)^2\big] \\
&= E[(\mathbf{y}_N^\top\mathbf{y}_N - 2\mathbf{y}_N^\top\overline{\mathbf{y}}^{(N)} + \overline{\mathbf{y}}^{(N)\top}\overline{\mathbf{y}}^{(N)})^2] \\
&= \underbrace{E[(\mathbf{y}_N^\top\mathbf{y}_N)^2]}_{A} + \underbrace{E[4(\mathbf{y}_N^\top\overline{\mathbf{y}}^{(N)})^2]}_{B} + \underbrace{E[(\overline{\mathbf{y}}^{(N)\top}\overline{\mathbf{y}}^{(N)})^2]}_{C} + \underbrace{E[2\mathbf{y}_N^\top\mathbf{y}_N\overline{\mathbf{y}}^{(N)\top}\overline{\mathbf{y}}^{(N)}]}_{D} \\
&\quad - \underbrace{E[4\mathbf{y}_N^\top\mathbf{y}_N\mathbf{y}_N^\top\overline{\mathbf{y}}^{(N)}]}_{E} - \underbrace{E[4\overline{\mathbf{y}}^{(N)\top}\overline{\mathbf{y}}^{(N)}\mathbf{y}_N^\top\overline{\mathbf{y}}^{(N)}]}_{F}.
\end{aligned}$$

As above, the fact that $\mathbf{y}_n$ is independend from $\overline{\mathbf{y}}^{(n)}$, and both have expectation null, imply

$$\begin{aligned}
E &= 4E[\mathbf{y}_n^\top\mathbf{y}_n\mathbf{y}_n^\top\overline{\mathbf{y}}^{(n)}] = 4E[\mathbf{y}_n^\top\mathbf{y}_n\mathbf{y}_n^\top]E[\overline{\mathbf{y}}^{(n)}] = 0 \\
F &= E[4\overline{\mathbf{y}}^{(n)\top}\overline{\mathbf{y}}^{(n)}\mathbf{y}_n^\top\overline{\mathbf{y}}^{(n)}] = 4E[\overline{\mathbf{y}}^{(n)\top}\overline{\mathbf{y}}^{(n)}\overline{\mathbf{y}}^{(n)\top}\mathbf{y}_n] = 4E[\overline{\mathbf{y}}^{(n)\top}\overline{\mathbf{y}}^{(n)}\overline{\mathbf{y}}^{(n)\top}]E[\mathbf{y}_n] = 0
\end{aligned}$$

By (6a), $A = \kappa_{11} + 2\mathrm{tr}(\Sigma^2) + (\mathrm{tr}\Sigma)^2$. By Lemma 1, $B = \frac{4}{N-1}\mathrm{tr}(\Sigma^2)$. By Lemma 2, $C = \frac{1}{(N-1)^3}\kappa_{11} + \frac{2}{(N-1)^2}\mathrm{tr}(\Sigma^2) + \frac{1}{(N-1)^2}(\mathrm{tr}\Sigma)^2$. By Lemma 3, $D = \frac{2}{N-1}(\mathrm{tr}\Sigma)^2$. Then

$$\begin{aligned}
E[Q_N] - E[Q_{N-1}] &= \Big(1 + \frac{1}{(N-1)^3}\Big)\kappa_{11} + \Big(1 + \frac{2}{N-1} + \frac{1}{(N-1)^2}\Big)(2\mathrm{tr}(\Sigma^2) + (\mathrm{tr}\Sigma)^2) \\
&= \Big(1 + \frac{1}{(N-1)^3}\Big)\kappa_{11} + \Big(1 + \frac{1}{N-1}\Big)^2(2\mathrm{tr}(\Sigma^2) + (\mathrm{tr}\Sigma)^2).
\end{aligned}$$

The case of merging two clusters is a simple consequence of (4), (9) and (10). □

Publisher's Note

Received: 7 August 2017 Accepted: 6 December 2017
Published online: 20 December 2017

References

1. Aggarwal CC. Data clustering: algorithms and applications. 1st ed. Boca Raton: Chapman and Hall/CRC; 2013.
2. Arthur D, Vassilvitskii S. K-means++: the advantages of careful seeding. In: Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms, society for industrial and applied mathematics. Philadelphia: SODA '07. 2007. p. 1027–35. http://dl.acm.org/citation.cfm?id=1283383.1283494.
3. Bradley PS, Fayyad UM, Reina C. Scaling clustering algorithms to large databases. In: KDD. 1998. p. 9–15
4. Fisher TJ, Sun X. Improved Stein-type shrinkage estimators for the high-dimensional multivariate normal covariance matrix. Comput Stat Data Anal. 2011;55(5):1909–18. https://doi.org/10.1016/j.csda.2010.12.006.
5. Fraley C, Raftery A, Wehrens R. Incremental model-based clustering for large datasets with small clusters. J Comput Gr Stat. 2005;14(3):529–46. https://doi.org/10.1198/106186005X59603.
6. Gan G, Ma C, Wu J. Data clustering: theory, algorithms and applications. ASA-SIAM Ser Stat Appl Probab. 2007;20:1–466. https://doi.org/10.1137/1.9780898718348.
7. Garofalakis M, Gehrke J, Rastogi R, editors. Data stream management: processing high-speed data streams. Data-centric systems and applications. Cham: Springer; 2016. https://doi.org/10.1007/978-3-540-28608-0.
8. Himeno T, Yamada T. Estimations for some functions of covariance matrix in high dimension under non-normality and its applications. J Multivar Anal. 2014;130:27–44. https://doi.org/10.1016/j.jmva.2014.04.020.
9. Hotelling H. The generalization of student's ratio. Ann Math Stat. 1931;2(3):360–78. https://doi.org/10.1214/aoms/1177732979.
10. Ikeda Y, Kubokawa T, Srivastava MS. Comparison of linear shrinkage estimators of a large covariance matrix in normal and non-normal distributions. Comput Stat Data Anal. 2016;95:95–108. https://doi.org/10.1016/j.csda.2015.09.011.
11. Jain AK. Data clustering: 50 years beyond k-means. Pattern Recogn Lett. 2010;31(8):651–66. https://doi.org/10.1016/j.patrec.2009.09.011.
12. Kaufman L, Rousseeuw P. Finding groups in data an introduction to cluster analysis. New York: Wiley; 1990.
13. Legendre P, Legendre LF. Numerical ecology, vol. 24. London: Elsevier; 2012.
14. Leskovec J, Rajaraman A, Ullman J. Mining of massive datasets. 2nd ed. Cambridge: Cambridge University Press; 2014.
15. Ng RT, Han J. Clarans: a method for clustering objects for spatial data mining. IEEE Trans Knowl Data Eng. 2002;14(5):1003–16. https://doi.org/10.1109/TKDE.2002.1033770.
16. O'Callaghan L, Mishra N, Meyerson A, Guha S, Motwani R. Streaming-data algorithms for high-quality clustering. In: Proceedings of IEEE international conference on data engineering. 2001. p. 685
17. Rencher A. Multivariate statistical inference and applications. Wiley series in probability and statistics: texts and references section. New York: Wiley; 1998.
18. Touloumis A. Nonparametric Stein-type shrinkage covariance matrix estimators in high-dimensional settings. Comput Stat Data Anal. 2015;83:251–61. https://doi.org/10.1016/j.csda.2014.10.018.
19. Zhang T, Ramakrishnan R, Livny M. Birch: an efficient data clustering method for very large databases. SIGMOD Rec. 1996;25(2):103–14. https://doi.org/10.1145/235968.233324.