Applied Soft Computing 60 (2017) 436–446

Contents lists available at ScienceDirect

Applied Soft Computing

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

Multi-source homogeneous data clustering for multi-target detection from cluttered background with misdetection

Tiancheng Li [a,*,1], Fernando De la Prieta Pintado [a], Juan M. Corchado [a], Javier Bajo [b]

[a] BISITE Group, Faculty of Science, University of Salamanca, Calle Espejo s/n, Salamanca 37008, Spain
[b] Department of Artificial Intelligence, Technical University of Madrid, Madrid 28040, Spain

ARTICLE INFO

Article history:
Received 22 June 2015
Received in revised form 10 April 2017
Accepted 5 July 2017
Available online 10 July 2017

Keywords:
Constrained clustering
Object identification
Multi-target detection
Sensor fusion

ABSTRACT

This paper investigates a particular data mining problem which is to 'identify' an unknown number of targets based on homogeneous observations that are collected via multiple independent sources. This particular clustering problem corresponds to a significant problem of multi-target detection in the multi-sensor/scan context. No prior information is given about either the level of clutter (namely noisy data) or the number of targets/clusters, both of which have to be learned online from the data. In addition, the data-points from the same source cannot be grouped into the same cluster (namely the cannot link, CL, constraint) and the sizes of the generated clusters need to be bounded by the number of data sources. In the proposed approach, a density-based clustering mechanism is proposed firstly to identify dense regions as clusters and to remove clutter at the coarser level; the CL constraint is then applied for finer data mining and to distinguish overlapping clusters. Illustrative datasets are employed to demonstrate the validity of the present clustering approach for multi-target detection and estimation in cluttered environments which are affected by both misdetection and clutter.

1. Introduction

Cluster analysis has been one of the most important technologies widespread in a variety of disciplines as a powerful tool for discovering the hidden structure/pattern in data. Due to the significant difference among data models, ranging from computer sciences to social sciences, small datasets to very large databases, and numerical data to data categories, a vast number of clustering algorithms have been proposed. It is fair to say that no single method applies to all cases, but all are model-specific and dedicated. The explosive development of clustering has also prompted reviews and surveys regarding general models e.g. [6,16], sequential [3]/time-series model [4,9], high-dimensional models [5,7], clustering with constraints [1,2], and clustering validity measure [8].

Clustering is usually taken as part of unsupervised learning as no prior information is available concerning the class of data-points. Nevertheless, in many problems a fair amount of a priori information is often available and can therefore be employed for more effective and efficient clustering, namely semi-supervised clustering [1,29]. Clearly, a priori information such as the number of clusters is critical both to the clustering result and to the clustering speed. For instance, if the number of clusters, namely the parameter k, can be correctly predefined, the k-means method is particularly preferable [6] otherwise it can be computationally NP hard [43,42]. Cluster center initialization can significantly affect the convergence speed and the output of the k-means algorithm [10,11]. Automatically determining number of clusters k has been one of the most difficult issues in data clustering. Most methods focus on model selection or matching in which clustering algorithms are run with different values of k; the best value of k is then chosen based on a predefined criterion such as information criterion values [12], rate distortion theory [13], etc.

Particularly, the 'cannot link' (CL) and the converse 'must link' (ML) constraints are two efficient rules for encoding a priori knowledge [14,19,20] and can significantly affect the outcome. The former corresponds to the requirement that two data-points should be assigned to different clusters, whereas in the latter the cluster labels

* Corresponding author.
E-mail addresses: t.c.li@usal.es, tiancheng.li1985@gmail.com (T. Li), fer@usal.es (F. De la Prieta Pintado), corchado@usal.es (J.M. Corchado), jbajo@fi.upm.es (J. Bajo).
[1] Dr. Li holds a Marie Skłodowska-Curie individual fellow position with the BISITE research group.

http://dx.doi.org/10.1016/j.asoc.2017.07.012
1568-4946/

of two targets should be the same. This allows the user to incorporate expertise into the clustering process by explicitly specifying the required or desirable property in the clustering outcome. Typical examples in this line include constrained k-means clustering [20], constrained hierarchical clustering [21] and the graph-cut based clustering with cluster size constraint [22].

Constrained clustering has received considerable attention in broad applications. Some constrained clustering algorithms do not allow any violation of the constraints, i.e., in all the iterations of the algorithm the resulting partitions must satisfy all the constraints. While this strict constraint may be interesting in certain cases, it can be computationally intractable. To overcome this limitation, more flexible alternatives have been proposed to minimize the number of violated constraints e.g. [23]. In such a case, the constraints are often referred to as soft constraints.

In this paper, we address a specific data mining problem in which the data originates from a number of independent and homogeneous sources, leading to a CL constraint whereby the data from the same source cannot be grouped into the same cluster. In other words, all the data in the same cluster must belong to different sources while data-points from the same source have to be partitioned into different clusters, even if they are very closely distributed. This multi-source data clustering (MSDC) problem originates from a significant engineering problem involved in the context of multi-sensor multi-target tracking [24]. These data-points, excluding an unknown number of outliers/clutter, belong to an unknown number of clusters, each of which corresponds to a target of interest. This MSDC problem where the observations from different sources are homogeneous and approximately i.i.d. (independent and identically distributed) in the state space is different from the similarly-called multi-source/multi-view data clustering problem [3,25–28] or the wireless sensor network–based clustering [40,41] where no i.i.d. condition holds, different sources/views are heterogeneous, and clutter/CL constraints may not be involved.

However, the multi-source CL constraint is more or less related, but still significantly different from some existing work, such as the pairwise CL constraint [19–22], where few specified data-point-pairs cannot be linked, or the semi-supervised learning [29,30], which labels the sources of the data (where training is carried out). Furthermore, overlapping clusters are involved where clusters can be overlapped if the corresponding targets are closely distributed. In existing research, mixed data among overlapping clusters are considered to be outliers [35], to belong to one or multiple clusters [31–33] or to belong to a given cluster to a certain degree [34]; see also [36,37]. None of these existing clustering approaches, however, exactly meet our requirements.

We frame multi-sensor multi-target detection as a constrained clustering problem (without using any traditional filter) and correspondingly propose a clustering method that is able to filter clutter and to detect the latent targets of interest from cluttered environments. Our approach is the hybrid of a density-based clustering process (at the first/coarser level) and a distance-based clustering process (at the second/finer level). Two respective processes have complementary goals: in the coarser level of density based clustering, the clutter will be identified and removed from further consideration. In the finer level of distance-based clustering, each intermediate cluster formed via the density rule will be revised to meet the CL constraint and to partition the over-size clusters. A short and earlier version of the proposed clustering method appeared in [39]. Compared to our conference paper, new content primarily consists of three aspects: 1) a novel and more computationally efficient solution for the finer distance-based clustering; 2) computing complexity and memory analysis of the algorithm and 3) explicit application for multi-target detection.

The rest of the paper is organized as follows. Section 2 formulates the problem model. Section 3 presents the detail of the proposed clustering method. The simulation results are shown in Section 4 and Section 5 concludes the paper.

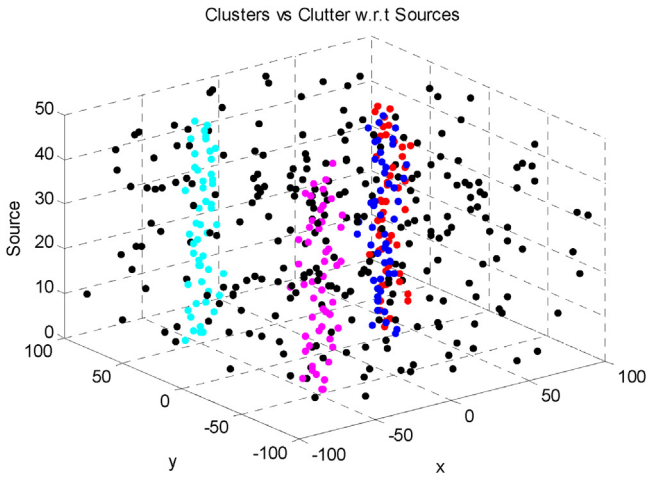

Fig. 1. Multi-source independent and identically distributed data: black dots represent clutter while different colors mark observations of different targets.

2. Problem definition and formulation

2.1. Multi-source data fusion

Multi-object/target detection and estimation (MODE) involves estimating the state of an unknown number of moving or stationary targets based on the noisy observations in the presence of clutter (namely outlier/noise in the clustering context, which also refers to false alarms that correspond to no target). Examples include buildings/crops in remote sensing images, or diseased cells/cracks in X-ray tomographic images, just to name a few. This is a scientific problem of dominating importance in a large variety of commercial, government and military realms, and has been extensively studied in the past half century based on various filters [24].

The general multi-target detection scenario can be modeled by the following assumptions:

(A.1) each target generates observations (in the format of data-points) independently of others and one target generates no more than one observation in each sensor at each scan;

(A.2) the target observation is coupled with unimodal noises, commonly e.g., zero-mean Gaussian;

(A.3) the sensor may miss-detect any targets with a probability;

(A.4) the clutter is assumed to be generated randomly over the scenario, independently of the targets, whose distribution density is much lower than that of the real observations of targets around the true position of targets.

Given that the targets are stationary against time, the observations received at different scans are independent and identically distributed (i.i.d.). The 'i.i.d.' condition approximately holds when massive homogeneous sensors, e.g., large scale wireless sensor networks, are used to monitor the same scenario, forming multi-views of the same scenario. It can also be loosely relaxed to accommodate the general scenario where targets are moving with a relatively low speed that is insignificant compared to the revisit frequency of the sensor (therefore, their movement is negligible between different scans, similar to the case of constant targets). Both multi-scan and multi-sensor can be collectively referred to as multi-source. In this paper, we formulate this multi-source data based MODE problem from the clustering viewpoint.

For illustration purpose, Fig. 1 gives the observations of targets (colored data-points) and clutter (black data-points) collected in 50 sources under the above assumptions (A.1-4), which are mapped

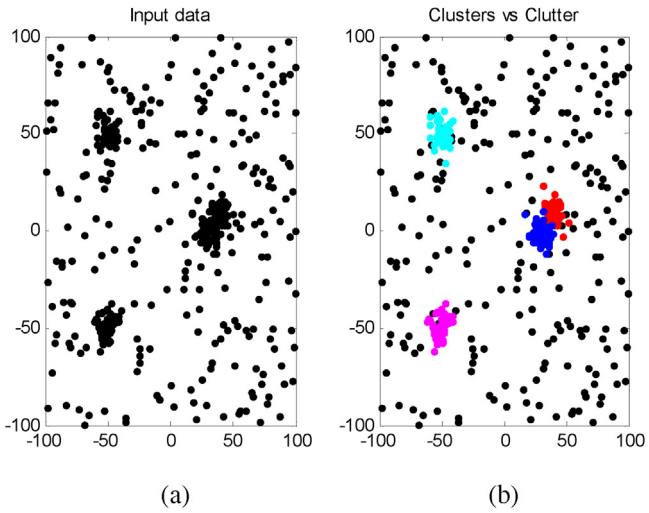

Fig. 2. Multi-source independent and identically distributed data mapped in the planar space: (a) all the data (b) desired clustering result (ground truth).

into the same planar $x-y$ space as shown in Fig. 2(a). Our goal is to distinguish the observations of each target from those of the others, and from the clutter, as shown in Fig. 2(b). The desired clusters can overlap with each other when targets are closely distributed in the observation space. If we can cluster the observation data properly, the number of targets and their positions can then be further estimated.

Intuitively, the observations of a particular target are subject to a unimodal distribution (as the observation noise is assumed to be unimodal in A.2) and so will cluster (around the true state of the target) while the random-appearing clutter will not. Meanwhile, the data input from the same source are independent (as assumed in A.1) which leads to the multi-source CL constraint in which data from the same source cannot be linked. The data distribution density and the CL constraint form two key factors to partition the observations of targets from clutter and from those of each other in our approach.

2.2. Problem formulation

The MSDC problem described above can be formulated as a CL-constrained clustering problem. Consider a dataset X consisting of data points

$$x_i \in P, i = 1, \ldots, N \tag{1}$$

where d is the dimensionality, P is the parameter space, x and N is the number of data-points to be clustered. In this paper we focus on the spatial data-points defined on numeric values where each data-point represents a coordinate in the state space.

The dataset can be written with respect to the sources. Denoting all the data-points from the sth source as $S_s = \left\{x_1^s, x_2^s, \ldots x_{m_s}^s\right\}$ where m_s is the number of data-points in the sth source, the multi-source dataset is defined as

$$X := \left\{S_1, S_2, \ldots S_n\right\} = \left\{x_1^1, x_2^1, \ldots x_{m_1}^1, x_1^2, x_2^2, \ldots x_{m_2}^2, \ldots, x_1^n, x_2^n, \ldots x_{m_n}^n\right\} \tag{2}$$

where n is the number of sources. The i.i.d. condition specifies that different sources of data-points are subject to the same spatial distribution, to say q, written as

$$S_s \sim q, \forall : s \in \left\{1, 2, \ldots, n\right\} \tag{3}$$

The goal of clustering here is to assign the data-points from different sources to a finite number of k subsets, called clusters C_1, $C_2 \ldots C_k$. Particularly, the CL constraint requires that

$$c \neq \left(x_i^s, x_j^s\right), \forall i, j \in \left\{1, 2, \ldots, m_s\right\}, s \in \left\{1, 2, \ldots, n\right\} \tag{4}$$

where $c \neq \left(x_i^s, x_j^s\right)$ means that x_i^s, x_j^s cannot belong to the same cluster. This is equivalent to defining the distance between data-points from the same source as infinite.

It is necessary to note that there are often noisy data-points (called clutter) that shall be excluded and shall not be associated to any target. Here, we refer to them as a set of outliers C_o. The union of these subsets is equal to a full data set:

$$X = C_1 \cup C_2 \cup \ldots \cup C_k \cup C_o \tag{5}$$

Moreover, these subsets do not interact in our approach, i.e.,

$$C_i \cap C_j = \Phi, \forall i, j \in \left\{1, 2, \ldots k, o\right\} \tag{6}$$

But, this is violated in other clustering methods [14], such as soft clustering.

As addressed so far, the goal of clustering can be described as: to group the multi-source data given in (2) to the multiple clusters given in (5) while satisfying the CL constraint (4) and non-interacting condition (6). Given that the distance between two data-points from the same source is defined as infinite (to include the CL constraint), the partitioning of the oversized cluster shall maximally minimize the distance sum between points within the same cluster. One challenge comes from the unknown number of clutter and targets/clusters, which may render the optimization problem computationally intractable. Particularly, the clutter could be significant as the number of noisy data can be larger than that of the real data. When the number of clusters is treated as a variable, the optimal clustering problem is basically NP hard.

2.3. Multi-source CL constraint and the size of clusters

The CL constraint (4) will limit the number of data-points in each cluster (namely the size of the cluster) to an expected level, which cannot be larger than the number of sources. That is, the sizes of the generated clusters have to be subject to a flexible constraint

$$|C_i| \lesssim n, \forall i \in \left\{1, 2, \ldots k\right\} \tag{7}$$

where $|C_i|$ means the number of data-points in cluster C_i, namely the size of the cluster, $\lesssim$ means "smaller than or approximately equal to" in which "approximately equal to" is because of the clutter that may fall in the cluster, and n is the total number of related sources that gives a loosely defined upper limit. As one cluster corresponds to one target, the notation i also indexes a target.

More precisely, we can estimate the expectation of the size of each cluster, namely the expected number of observations received from each target i. The cluster size is given by the number of detections in the cluster, which depends on both the number of sensors whose field of view (FoV) covers that cluster, and their respective detection probabilities. Denoting the number of the sensors whose FOV covers target/cluster i as n_i and the detection probability of sensor s on target i as $p_D^s(i) \leq 1$, a simple estimate of the cluster size is given by

$$\mathrm{E}[|C_i|] = \sum_{s=1}^{n_i} p_D^s(i) \lesssim n_i \tag{8}$$

In a simple case, the detection probability of each sensor $p_D(i)$ is a constant, denoted as p_D, over the entire scenario, which will render all clusters of similar size, namely$\forall i, j \in \{1, 2, \ldots k\{: \mathrm{E}[|C_i|] = \mathrm{E}\left[|C_j|\right]$.

When multiple objects appear close, their detections can be easily clustered into one cluster. As a criterion to realize the CL constraint, the over-sized cluster has to be divided into multiple individual sub-clusters if the size of the cluster exceeds limitation n_i. To partition the cluster, another threshold ρ is needed to give the average number of detections in a single-object cluster. It shall be

designed with respect to the expected number $\mathrm{E}[|C_i|]$ of detections for a single local object, e.g., $\rho_i = l \times \mathrm{E}[|C_i|]$, where $0 < l < 1$ is scalable and involves a trade-off between missing detections (if too high l) and causing false alarms (if too low). In reality, it is very rare that more than two clusters overlap and we recommend $l \in [0.6, 0.9]$. In this paper we use $l = 0.8$ which is demonstrated to work reliably for most of the cases we found. The number k_i of sub-clusters contained in cluster C_i satisfies

$$k_i\rho_i \le |C_i| < (k_i + 1)\,\rho_i \tag{9}$$

As long as there are not too many closely distributed targets, k_i can be approximately calculated by

$$k_i \approx \left[\frac{|C_i|}{\mathrm{E}[|C_i|]}\right] \tag{10}$$

where $[\cdot]$ denotes the rounding operation which gives the nearest integer to the content. If a priori information about $p_D^s(i)$ is unavailable for calculating $\mathrm{E}[|C_i|]$ in (8), an alternative to estimate the number of sub-clusters that shall be formed can be given by the average number of data-points in each cluster that are originating from the same source (for all sources). This can be written as

$$k_i \approx \left[\frac{1}{n_i}\sum_{s=1}^{n_i}|\{x_j^s \in C_i | j \in \{1, 2, \ldots m_s\}\}|\right] \tag{11}$$

Our clustering solution will be presented next with an analytical complexity analysis and an online learning method to estimate the key required parameter.

3. Multi-source *n*-points clustering

3.1. The main procedure

As aforementioned, a key piece of information that can be employed to cluster the data is the distribution density of the data-points, for which the data-points that significantly cluster are more likely to be the observations from targets. This inherently resembles the density-based clustering in which clusters are high density regions in the feature space separated by low density regions. In addition, the CL constraint (4) has to be taken into consideration for finer data mining based on the intermediate density-based clustering results. Overall, the proposed clustering scheme shown in Algorithm 1 consists of three main steps: 1) search across sources to identify different groups of data-points that are closely connected in the feature space; 2) for each group of an adequate number of connected data-points, determine whether to form it as a single cluster (which should be approximately equal or slightly less than $\mathrm{E}[|C_i|]$) or divide it into multiple sub-clusters (if significantly exceeds $\mathrm{E}[|C_i|]$); and 3) if the CL constraint needs to be strictly satisfied, revisit each cluster to make sure that the CL constraint is satisfied.

The optimal clustering problem concerned with unknown statistics about either the targets or the clutter for which the number of targets needs to be identified and clutter-distinguished from real detections, is generally a NP hard problem. To solve the problem efficiently, at the first level, we will not explicitly consider the CL constraint. Instead, the first step focuses on associating closely distributed data-points across different sources. To do so, a "*neighbor radius*" (NR) parameter ε_i is needed to distinguish close data-points from clutter, where i indicates different clusters. In Section 3.2, we will show that this parameter can be online learned.

Remark 1. NR parameter ε_i corresponds to the maximum distance between a data-point and its neighbors from the other sources for their direct connection to be included in the same cluster, which resembles the neighbor radius parameter used in DBSCAN [17] (density-based spatial clustering of applications with noise). It can be designed with respect to the standard deviation σ_i of the cluster distribution, e.g., $\varepsilon_i = (1\sim3)\,\sigma_i$ where σ_i corresponds to the magnitude of the noise affecting the observation on target/cluster i. Clearly, the larger the observation noise, the larger ε_i. If statistical knowledge of the observation noise is infeasible, it can be approximately estimated from the dataset as shown in Section 3.2.

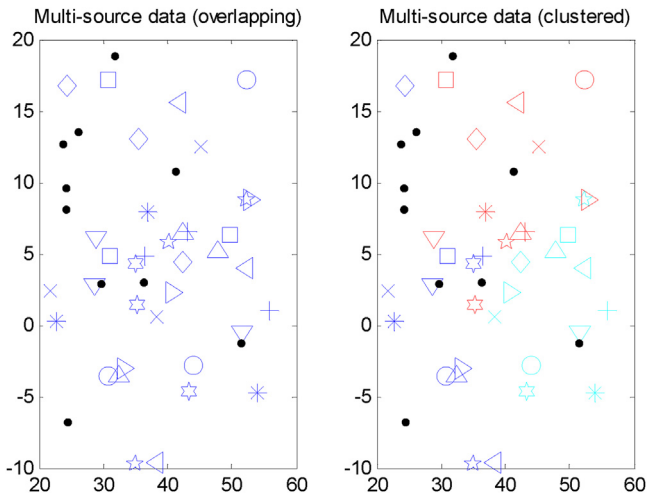

Fig. 3. Finer clustering of overlapping data-points from 10 sources (clutter is denoted by black ".").

Remark 2. The proposed density-based clustering scheme, namely Step 1 of Algorithm 1, needs to visit each data-point, possibly multiple times, for which the computing complexity is mostly governed in practice by the number of region query invocations. Like DBSCAN, our approach executes exactly one such query for each point. Given that an efficient tree indexing structure is used and NR parameter ε is chosen properly, so that it executes a neighborhood query in $O(\log N)$ (i.e. on average only $O(\log N)$ points are returned for each point), an overall average runtime complexity of $O(N \log N)$ is obtained. Without the use of an accelerating index structure, or degenerated data (e.g. all points within a neighbor distance), the worst case will be $O\left(N^2\right)$.

To conduct the CL constraint in the second step, a detection of the number of data-points in each cluster shall be applied to distinguish and further partition oversized clusters into several sub-clusters. Given the number of sub-clusters to divide from an oversized cluster via (10) or (11), the k-means algorithm is competent to obtain sub-clusters of an approximately equivalent size [39]. To avoid violating the CL constraint, we can simply set the distance between two data-points that are from the same source as infinite. If k and d are fixed, the optimal k-mean clustering can be carried out in time $O\left(N^{dk+1}\right)$ [43], which is indeed costly. Instead, a variety of heuristic algorithms such as Lloyd's algorithm and its accelerated versions [42] can be used. The running time of Lloyd's algorithm is $O(wk|C_i|d)$ for w iterations needed until convergence, k centers, and $|C_i|$ (which is smaller than N) points in d dimensions. However, w may be superlinear with respect to $|C_i|$, even exponential in the worst case [43], although it can be specified as small when the data inherently have a clustering structure.

To partition the oversized cluster more efficiently, we propose a solution of certain linear complexity, which is described in Algorithm 2. The procedure is illustrated in Fig. 3 for three overlapping clusters where the data-points are from 10 sources. In the figure, different marks represent different sources; the color on the right subfigure represents different cluster labels.

Remark 3. The runtime complexity of Algorithm 2 mainly depends on the query searching of data-points from each source,

namely Step 2.2. Each cluster needs to execute maximally one query searching for each point so the runtime complexity is$O(k|C_i|d)$, which does not need iteration and is much lower than that of the k-means.

Remark 4. In the proposed clustering schemes based on either density or distance, the distances between data-points are needed. The distance matrix of size $\left(N^2-N\right)/2$ can be materialized to avoid distance re-computations for speedup, but this needs $O\left(N^2\right)$ data storage memory, whereas a non-matrix based implementation only needs $O(N)$ memory.

In addition, if the resulting (intermediate) clusters based on the coarser density-clustering are not oversized, they will not be revised by the finer distance-based clustering, which may leave a few data-points violating the CL constraint. Therefore, as an alternative, each cluster may be double-checked and revised at the end to make sure that the CL constraint is fully respected. For any cluster that violates the CL constraint, a simple solution is to remove all the data-points violating the CL constraint except the one that is nearest to the center of the cluster. This is referred to as strictly constrained clustering; otherwise, without this step, our approach is referred to as soft constrained clustering.

Based on the presented procedure, the clustering results for the data-set given in Fig. 1 are shown in Fig. 4. In the figure, clustered data-points are circled with different colors. Again, the color of the circles is independent of the color of the data-points. As can be seen, the results appear very reasonable. Particularly, it is possible to distinguish between the overlapping clusters (red and blue), although there are a few mismatching data-points.

Algorithm 1 multi-source n-points clustering

Step 1:	Calculate the distances between any two data-points from different sources in the parameter space. Two data-points will be identified as connected and classified into the same group C_i if their distance is smaller than a threshold vector ε_i; see **Remark 1** and **Algorithm 3**. Any group C_i of size larger than $l\times \mathrm{E}[|C_i|]$ data-points forms a cluster.
Step 2:	Calculate (10) or (11) for each cluster obtained in the first step. If $k_i\geq 2$, the 'oversized' cluster has to be further partitioned into k_i sub-clusters based on the CL constraint; see **Remark** 2 and **Algorithm 2**.
Step 3 (alternative):	Revisit each cluster: all the data-points violating the CL constraint except the one that is nearest to the center of the cluster will be removed from the cluster.

Algorithm 2 Partitioning overlapping clusters

Step 2.1	Identify the source s that contributes the minimum number of data-points to the underlying oversized cluster.
Step 2.2	Starting from a data-point in source s, associate it with the nearest data-points in all the other sources to form a group; assuming the group has n_i data points in total, it forms a new sub-cluster if and only if $n_i\geq l\times \mathrm{E}[|C_i|]$.
Step 2.3	Repeat Step 2.2 till all the data-points in source s are grouped into sub-clusters or until all k_i sub-clusters are formed
Step 2.4	Apply Step 2.1–2.3 to the remaining sources excluding s if the total number of sub-cluster is still smaller than k_i.

3.2. Online estimating NR parameter ε

In the case that the statistic property of the sensors is available, the NR threshold ε can be determined correspondingly as addressed in Remark 1. Otherwise, it needs to be estimated online. It can be estimated through unsupervised learning of the data; see algorithm 3 given below. The idea is to approximate a constant value for it as follows:

$$\varepsilon=\min_{j\in\{1,2,\ldots,m_L\}}\left(d_j(T,L)\right) \tag{12}$$

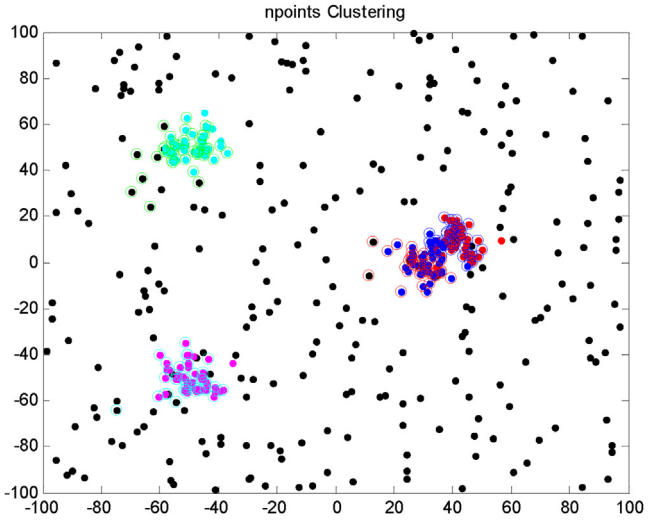

Fig. 4. Clustered data-points (circle 'o' with different color), corresponding to Figs. 1 and 2.

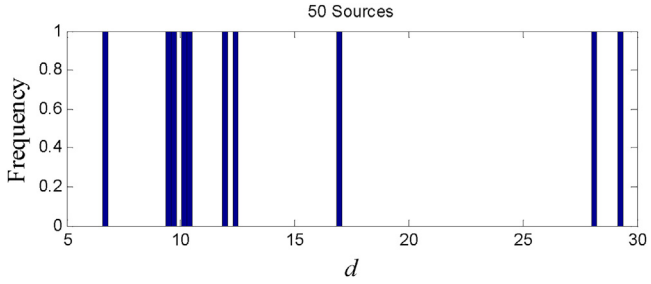

Fig. 5. Rank of d_j, corresponding to Figs. 1 and 2.

where

$$L=\underset{s}{\operatorname{argmax}}|S_s| \tag{13}$$

$$d_j(T,L)=\underset{\substack{m\in\{1,2,\ldots,m_p\}\\ p\in\{1,2,\ldots,n\},\,p\neq L}}{T\text{thmin}}\ d\left(x_j^L,x_m^p\right) \tag{14}$$

where L represents the source containing the largest number of data-points, $\underset{m,p\,\text{s.t.}\,G}{T\text{thmin}}d\left(x_j^L,x_m^p\right)$ gives the Tth smallest value of the Euler distance $d\left(x_j^L,x_m^p\right)$ between data-points x_j^L and x_m^p in the parameter space for any m, p satisfying condition G.

Parameter T, which specifies one of the middle neighbors to calculate its distance to the underlying data-point as an average estimate of the "neighbor radius", does not need to be specified precisely. One rule of thumb is to determine T between $\max\left(\frac{n_i}{10},2\right)$ and $\frac{n_i}{2}$, e.g., $n_i/5$ or just a constant value 5. For instance ($T=5$), the rank of d_j as defined in Step 2 of Algorithm 3 for the dataset given in Figs. 1 and 2 is given in Fig. 5. As shown, the resulting minimum $d_j(T,L)$ is roughly 6.6.

It is possible that all the data-points are clutter in the scenario or, conversely, are from a single target. In the former case, the data-points are uniformly distributed over the whole scene while in the latter, they are centralized to one center. If the NR parameter ε is available, it will be easy to identify them, as very few data-points can be clustered in the former case while almost all the data-points will be clustered to one in the latter case. If the parameter ε is not given, their difference can be determined by the rank distribution of the neighbor radius as defined in (14). The variance of d_j in the

latter case ought to be significantly larger than that in the former case.

Algorithm 3 Estimating the NR parameter ε

Step 1 Identify the source L that contains the most data-points $L = \underset{i}{\text{argmax}}|S_i|$.

Step 2 Calculate the distances of each data-point of S_L to its T nearest data-points from the other sources, denoting the largest of them as $d_j, j = 1, 2, \ldots, |S_L|$;

Step 3 Rank d_j for all data-points from S_L, obtaining the minimum value $\min_j d_j$ which can be estimated as the required ε.

3.3. Further discussion

Denoting the state-mean and (co)variance of the detections in the same cluster as $\mathbf{m}_i$ and $\mathbf{P}_i$, $i = 1, \ldots n$ respectively, where i indicates different sources, the cluster center can be given by the best linear unbiased estimator as follows

$$m_{\text{fused}} = \frac{\sum_{i=1}^{n} \mathrm{P}_i^{-1} m_i}{\sum_{i=1}^{n} \mathrm{P}_i^{-1}} \tag{15}$$

If all these detections are subject to the Gaussian distribution, the final target state-estimate is also Gaussian and its variance is

$$\mathbf{P}_{\text{fused}} = \left(\sum_{i=1}^{n} \mathbf{P}_i^{-1} \right)^{-1} \tag{16}$$

Clearly, $\mathbf{P}_{\text{fused}} < \mathbf{P}_i, \forall\, i = 1, \ldots n$ which indicates that the clustering algorithm promises a more accurate estimate than the original data.

We note that clutter may still be scattered in the cluster even after the density–based clustering, which will deactivate (16). As long as the clutter is uniformly distributed over the scenario, the clutter distribution in the cluster area will be still loosely centralized around the mean of the cluster, and the estimate given by (15) remains unbiased, that is, centralized around the real position of the target. In other words, even clutter may be unexpectedly taken into account in the cluster, it may not deflect the cluster. This allows us not to strictly follow the CL constraint.

However, if the clutter is not uniformly distributed in the cluster area, e.g., clutter depends on the real detections, or if the real target detection is biased (i.e., the observation noise is not zero-mean and assumption A.2 is violated), the above unbiasedness claim will not hold. We omit these complicated cases in this paper but we iterate that the density based clustering scheme, which serves as the pre-processing step of our approach to remove clutter, has the strength to discover clusters with an arbitrary shape, and the efficiency to handle clutter. Further on, more advanced density-based clustering approaches can be employed. For example, the study done in [15] improved DBSCAN to cluster data with dense adjacent clusters. The method of OPTICS (ordering points to identify the clustering structure) [18] extends the idea of DBSCAN to data of varying density. Both methods can be useful to our approach, especially for large scale problems or when the sensors are non-homogeneous.

While the present clustering approach does not make direct assumption about the target dynamics for accommodating poor a priori information, we note that available target motion information, if any, should be used to improve the clustering. For example, when the number of targets is known to be varying insignificantly over time, the estimated number of targets in the preceding iteration can be used in the next iteration as a reference of the potential number of clusters. Moreover, if the target moving speed (and turn rate) is known to be nearly constant, it shall also be used in a way to initialize the potential cluster centres at each iteration by propagating the cluster centres obtained in the preceding iteration to speed up the clustering search.

Table 1
Computing time of different clustering methods (s).

	k-means	DBSCAN 2	DBSCAN 6	Multi-source n-points
Fig. 6	0.0075	0.0156	0.0119	0.0436
Fig. 7	0.0136	0.0422	0.0399	0.0834

4. Demonstration and evaluation

Although existing approaches offer no explicit mechanism to deal with the multi-source CL constraint, we can still implement the popular DBSCAN and k-means method in their best possible parameter setting for an illustrative comparison with the proposed multi-source n-points clustering.

4.1. Given NR parameter ε

In our clustering method, NR parameter $\varepsilon = 10$ is two times the standard deviation of the observation noise. Parameter $k = 6$ is used for the k-means clustering, which puts its performance into the best possible situation in our case. The DBSCAN algorithm needs two parameters ε and m. Parameter ε gives the neighborhood radius and is therefore set as $\varepsilon = 10$. Parameter m gives the minimum number of points in a neighborhood for its inclusion in a cluster; two different values $m = 2, 4$ are adopted. The simulation results for the data size $n = 20, 50$ are given in Figs. 6 and 7 respectively. The colors of the circles (which represent different clusters) are assigned randomly in each run and are independent of the color of the data-points, which indicates the true clusters for different targets. The results show a significant advantage of our approach over the other methods that are unable to deal with overlapping clusters. Particularly, the basic k-means method suffers from clutter (outliers) most. However, we note that advanced k-means such as the MPCK-means algorithm [14] may be employed to deal with constraints, which is still inefficient to handle clutter, and so we did not investigate their efforts.

The average computing time of different clustering methods over 100 trials is provided in Table 1 for the datasets given in Figs. 6 and 7. It shows that the proposed clustering is somewhat slower than the others but is still quite fast, considering that the CL constraint has been fully satisfied.

4.2. Unknown NR parameter ε

Based on the same dataset as that given in the last section, we assume that NR parameter ε is unknown, which has to be learned online through Algorithm 3 from the dataset. The upper and bottom sub-figures of Fig. 8 give the distribution of dj for the dataset shown in Fig. 6 ($n = 20$) and 7 ($n = 50$) respectively, which is obtained at a very fast computational speed (which costs 0.0013 and 0.0025 s respectively for each run in the Matlab platform). The outcomes for estimating NR parameter ε are approximately 6.6 and 4.6 respectively. Based on this, the clustering results yielded by our proposed approach are given in Fig. 9. The estimated parameters are shown to be suitable in that they enable good clustering results, which is very close to the results shown in Figs. 6 and 7. To quantitatively compare with the results shown in the last section, we employ the average purity of clusters defined as follows:

$$AP = \frac{1}{k} \sum_{i=1}^{k} \frac{|C_i^d|}{|C_i|} \times 100\% \tag{17}$$

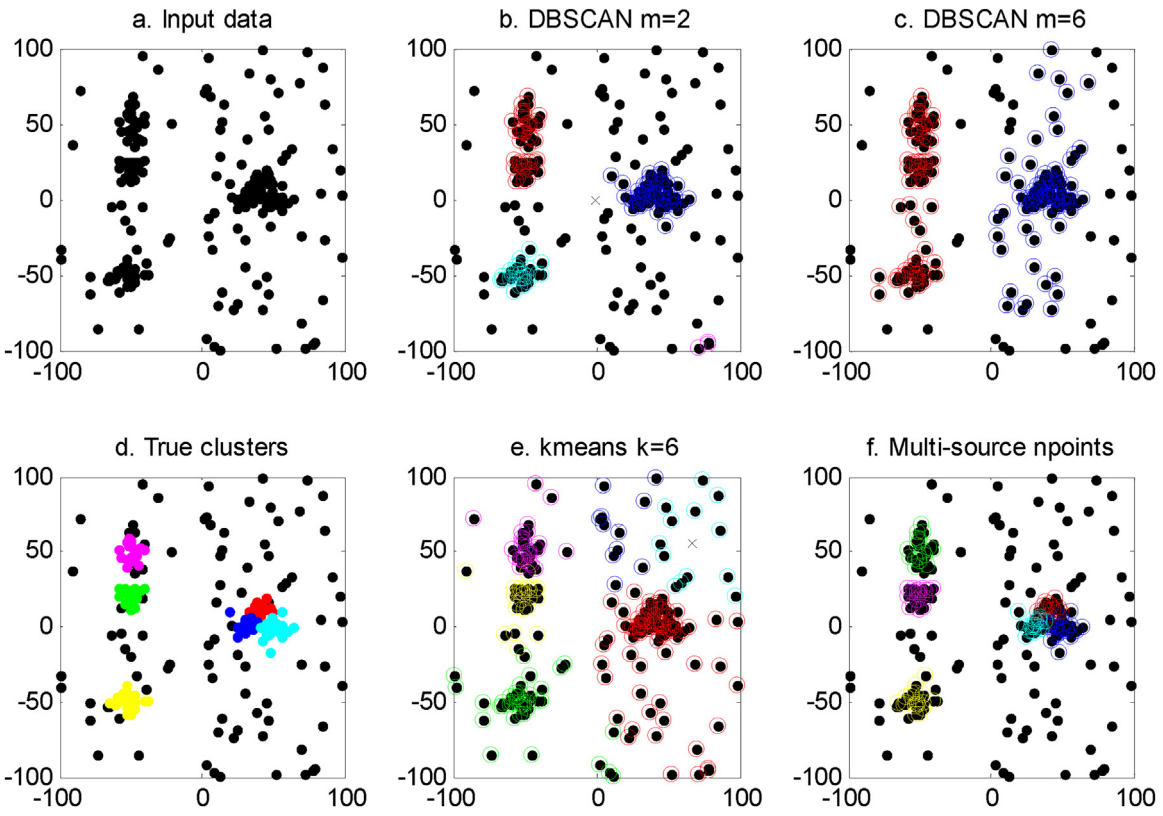

Fig. 6. Outcomes of different clustering methods on dataset from 20 i.i.d sources.

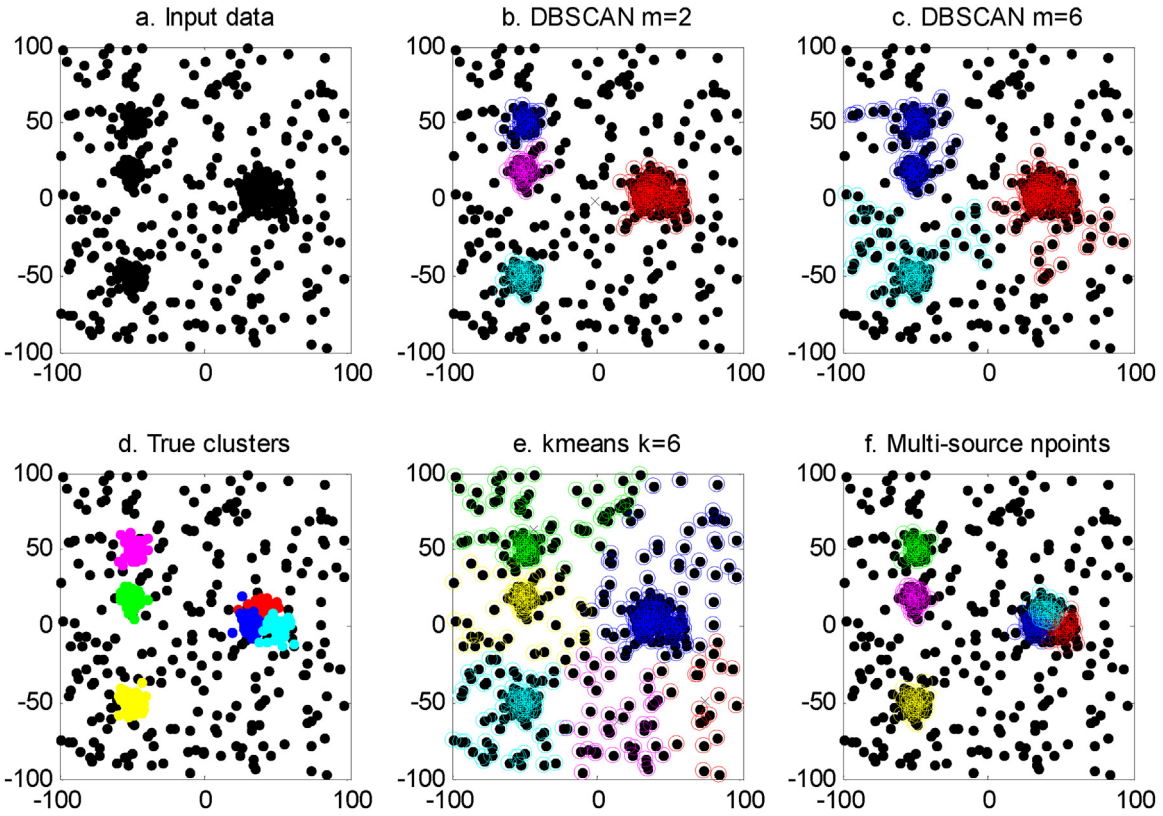

Fig. 7. Outcomes of different clustering methods on dataset from 50 i.i.d sources.

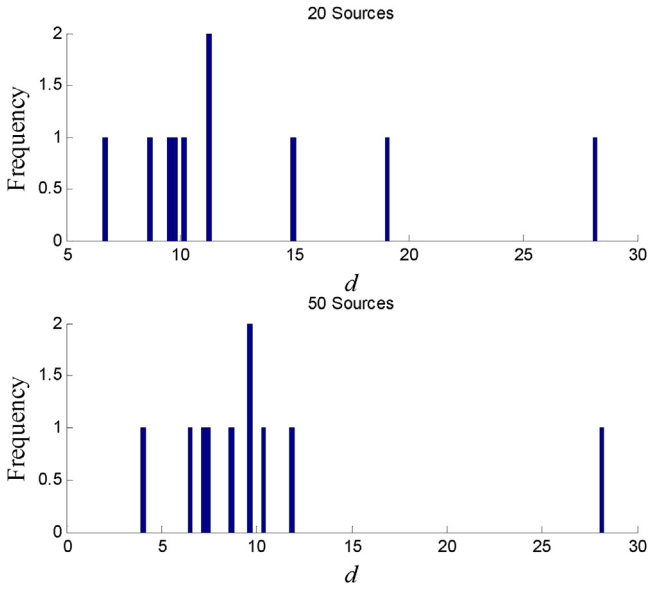

Fig. 8. Rank of d_j, w.r.t. Figs. 6 (upper) and 7 (bottom) respectively.

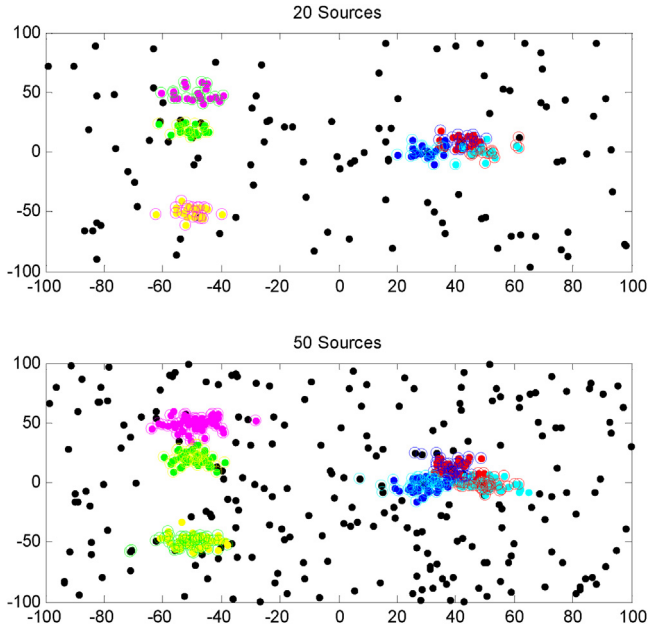

Fig. 9. Clustering results using estimated NR parameter ε w.r.t. Figs. 6 (upper) and 7 (bottom) respectively.

Table 2
Average purity of the multi-source n-points clustering method (%).

	Known NR parameter $\varepsilon = 10$	Online learned NR parameter ε
Fig. 6/Fig. 9(upper)	95	90.8
Fig. 7/Fig. 9(bottom)	96.3	93

where k is the number of clusters, $|C_i^d|$ is the number of data-points that correspond to the same target i.

The average purity of clusters provides a simple and transparent measure for cluster evaluation. For the two cases as given in Table 2, it shows that the performance of the multi-source n-points clusters based on online learned parameter ε performs closely to that using a NR parameter ε properly specified a priori, but at the expense of slightly more computation. This demonstrates that the online learning procedure for NR parameter ε given in Algorithm 3 is both effective and computationally efficient.

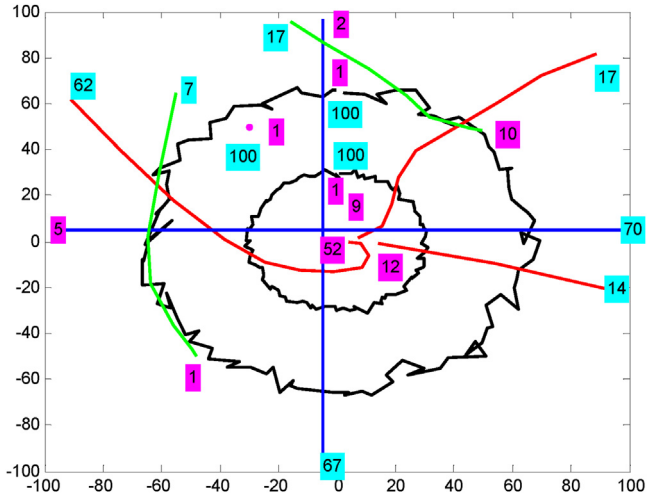

Fig. 10. Trajectories of targets with fully unknown movement.

4.3. Clustering-based MODE

In this simulation, we consider applying the proposed clustering for massive sensor MODE in a challenging scene in which the target motion model and system noises are completely unknown and are highly time varying. For instance, massive cameras are set up to monitor an indoor room. The information available is only from the observations, which include the pixel position identified on the frames/images (by using some image processing technologies, which are beyond the scope of this paper) that are extracted from the real-time video stream. We need to identify the observations that are from the same target and extract their distribution mean in the state space as the estimate of the target position.

The targets can appear and disappear anywhere and/or anytime in the scene, whether jointly, adjacently or solitarily, and they may split, merge or cross each other, just to name a few possibilities. Few traditional filter-based estimators that rely heavily on target dynamic modelling are capable of handling such challenging unknown scenarios. However, we can still apply cluster analysis on the i.i.d. observations received from massive sensors, given that the observations are mapped into the same coordinate.

The ground truth of the trajectories of these targets over the view region $[-100,100] \times [-100,100]$ is given in Fig. 10, with the starting and ending times of each trajectory noted. These targets start at different times and exist for different lengths, as shown in Fig. 11. Clutter is uniformly distributed over different regions with an average rate of $r = 10$ false data-points per scan. For simulation only, the observation noises are set as mutually independent zero-mean Gaussian with variance 5. To note, both the clutter density and the observation noises are unknown to our clustering method. Significantly different from our simulation presented in [38] which utilizes a parameter ε properly given a priori, Algorithm 3 is used to learn the parameter ε from the dataset.

The center of each cluster obtained by our approach is given as the estimate of the target position. The optimal sub-pattern assignment (OSPA) metric [44] is used to evaluate the estimation accuracy. For finite subsets $X = \{x_1, x_2, \ldots, x_m\}$ and $Y =$

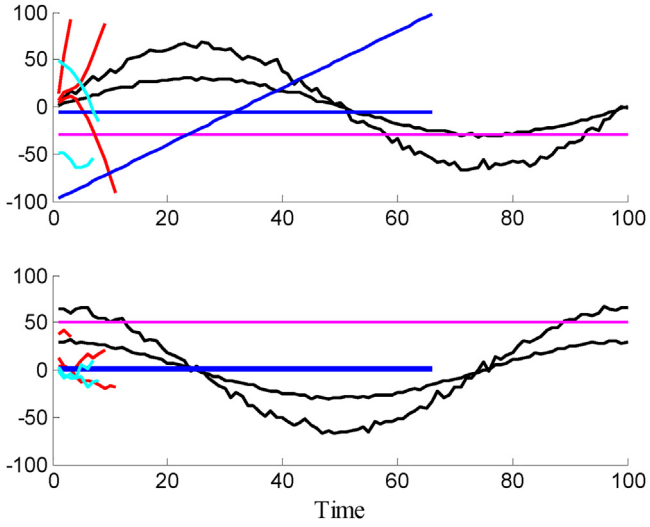

Fig. 11. Trajectories of targets in $x-y$ dimension separately.

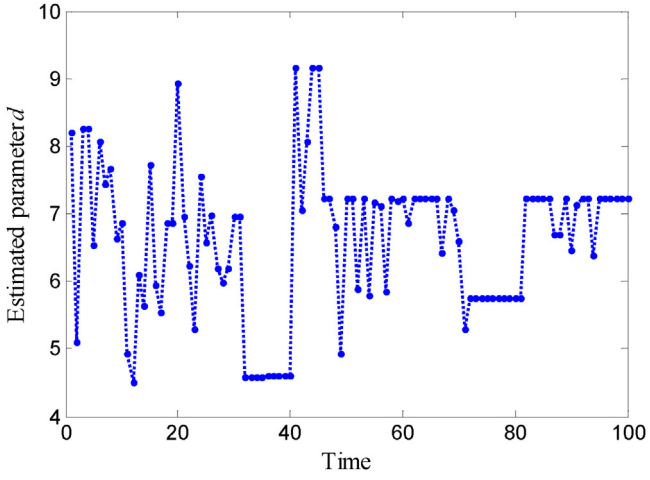

Fig. 12. Online learned NR parameter ε against time.

$\{y_1, y_2, \ldots, y_n\}$ where $m, n \in \mathbb{N}_0 = \{0, 1, 2, \ldots\}$, the OSPA metric of order p between X and Y is defined as (if $m \leq n$)

$$\bar{d}_p^{(c)}(X, Y) = \left(\frac{1}{n} \left(\min_{q \in \Pi_n} \sum_{i=1}^{m} d^{(c)}\left(x_i, y_{q(i)}\right)^p + c^p (n-m) \right) \right)^{\frac{1}{p}} \quad (18)$$

where $d^{(c)}(x, y) = \min(c, d(x, y))$, the cut off value $c > 0$ and $d(x, y)$ is the Euler distance. $\bar{d}_p^{(c)}(X, Y) = \bar{d}_p^{(c)}(Y, X)$ if $m \geq n$ and $\bar{d}_p^{(c)}(X, Y) = 0$ if $m = n = 0$.

The order parameter p determines the sensitivity to outliers, and the cut-off parameter c determines the relative weighting of the penalties assigned to cardinality and localization errors. Clearly, a better target detection capacity that renders more accurate estimation of the target number will significantly reduce the OSPA metric. The parameters used are $c = 100$, $p = 2$.

First, we use ten sensors. The learned NR parameter ε against time is given in Fig. 12, which shows that the obtained estimate is roughly between 4.5 and 9 (different from [38] which used the default $\varepsilon = 10$). Correspondingly, estimates from ten sensors, true target positions and the mean estimates given by each obtained clusters are respectively given in Fig. 13 for $t = 16$. The average of the estimated number of targets and the mean OSPA over 100 Monte Carlo runs are given in Fig. 14. The average OSPA over 100 steps $\times$ 100 MC runs is 10.77, which is arguably a very good result

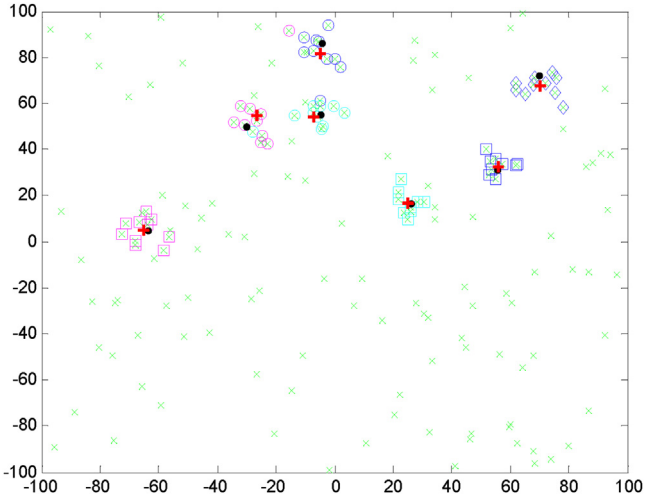

Fig. 13. Observation reports (green "x") of 10 sensors and their clustered results (different color "o", "□" or "◊"), true target positions (black ".") and estimates (red "+") at time $t = 16$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

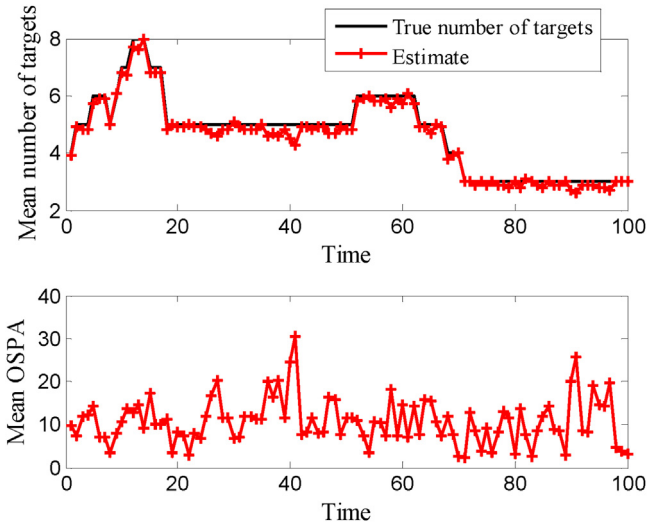

Fig. 14. Mean estimated number of targets and mean OSPA over 100 Monte Carlo trials.

with regard to the cutoff parameter $c = 100$ and order parameter $p = 2$ used for OSPA and fully unknown tracking background.

Fig. 15 gives all the estimates from 100 sensors, the true state positions and the mean estimates from clusters for time $t = 16$. This clearly shows that the estimate of the real targets can be better distinguished statistically from the false alarms when the number of sensors used is large, compared with the results of Fig. 13. Fig. 16 gives the mean OSPA and the mean processing time, both against the number of sensors used. The results show that with an increase in the number of sensors, the proposed clustering will get a more accurate state estimation, which is consistent with the analysis given in (16). This, however, comes at the cost of proportionally increasing processing time, as its computation is linearly proportional to the size of the dataset. This demonstrates again that the proposed clustering method is qualified to serve independently as a reliable and accurate filter-free MODE estimator with the use of massive sensors in the challenging time-varying cluttered environment of unknown statistics about the targets, the clutter and even the sensors. Its estimation accuracy increases statistically with the increase of the number of sensors.

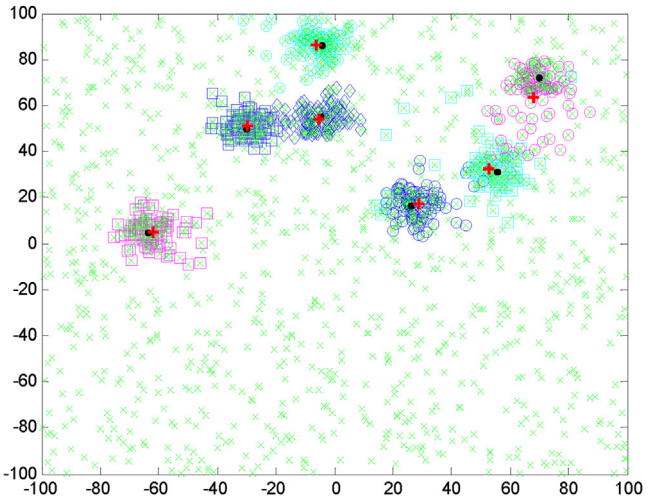

Fig. 15. Observation reports (green "x") of 100 sensors and their clustered results (different color "o", "□" or "◇"), true target positions (black ".") and estimates (red "+") at time $t = 16$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

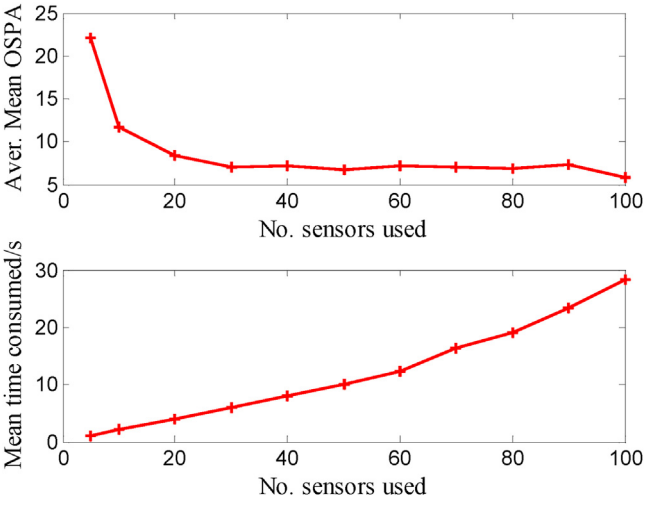

Fig. 16. Mean OSPA of 100 steps × 100 MC runs and mean processing time of 100 steps against different numbers of sensors used.

Furthermore, we employed 10 sensors with different clutter rates averaging from 0 (that is, no clutter is generated at each sensor) to 20 per scan for each sensor. The mean OSPA and processing time of our approach, which are given in Fig. 17, show, unsurprisingly, that with the increase of the clutter rate, the algorithm consumes more processing time and yields a worse estimation. With a very high clutter rate 20 per scan, the average OSPA is 23.94. We note that high clutter rate is a threat to any target detection algorithm and our approach is not an exception.

5. Conclusion

We have investigated a multi-source homogeneous data clustering model which poses "cannot- link" (CL) constraints on the data from the same source. The dataset may be affected by a high level of clutter, misdetection and the number of potential clusters is unknown. In our approach, the first/coarser level clustering is based on density for fast computing, while the second/finer level is based on CL constrained distance to partition closely connected clusters. In addition, an online learning procedure is proposed for parameter estimation, thus allowing the clustering method to

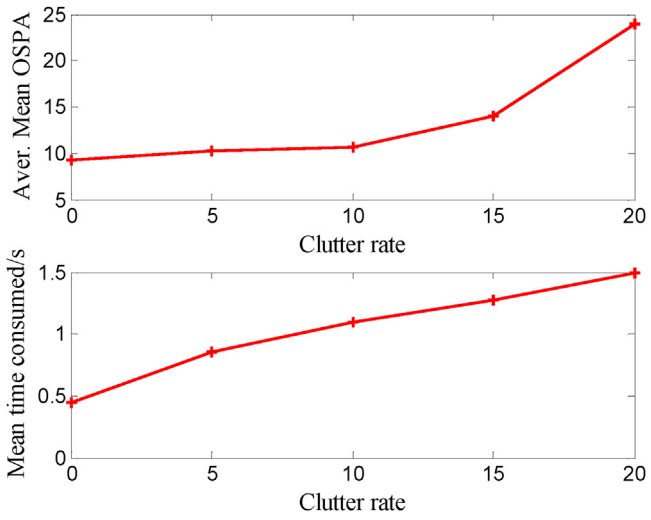

Fig. 17. Mean OSPA of 100 steps × 100 MC runs and mean processing time of 100 steps against different clutter rates.

be fully available without manual parameter setting in advance. This is highly preferable for challenging target detection environments with very little prior information about the background and the sensors. Simulations including synthetic data and challenging multi-sensor multi-target detection applications demonstrate the validity and efficiency of the present clustering method for multi-target detection in challenging cluttered environments. The clustering approach has shown consistent performance for fusing multi-sensor data as that its estimate accuracy will increase with the increase of the number of sensors. This is superior to many model-based approaches for which the inevitable model error will limit the gain yielded by using more data.

Future work will extend the proposed multi-source n-point clustering method for time series sensor data streams as well as heterogeneous and asynchronous sensor data.

Acknowledgments

This work is in part supported and funded by Marie Skłodowska-Curie Individual Fellowship (Grant number 709267) under the European Union's Framework Programme for Research and Innovation Horizon 2020 and the Spanish Ministry, Ministerio de Economía y Competitividad and FEDER funds (Project Ref. TIN2015-65515-C4-3-R).

References

[1] N. Grira, M. Crucianu, N. Boujemaa, Unsupervised and semi-supervised clustering: a brief survey, A Review of Machine Learning Techniques for Processing Multimedia Content, Report of the MUSCLE European Network of Excellence (6th Framework Programme) (2005).
[2] S. Basu, I. Davidson, K. Wagstaff, Constrained clustering: advances in algorithms, theory and applications Data Mining and Knowledge Discovery, vol. 3, Chapman & Hall/CRC, 2008.
[3] G. Dong, J. Pei, Sequence Data Mining, Advances in Database Systems, vol. 33, Kluwer, 2007.
[4] S. Aghabozorgi, A.S. Shirkhorshidi, T.Y. Wah, Time-series clustering—a decade review, Inf. Syst. 53 (2015) 16–38.
[5] E. Elhamifar, R. Vidal, Sparse subspace clustering: algorithm, theory, and applications, IEEE Trans. Pattern Anal. Mach. Intell. 35 (11) (2013) 2765–2781.
[6] A.K. Jain, Data clustering: 50 years beyond K-means, Pattern Recognit. Lett. 31 (8) (2010) 651–666.
[7] H.-P. Kriegel, P. Kröger, A. Zimek, Clustering high-dimensional data: a survey on subspace clustering, pattern-based clustering, and correlation clustering, ACM Trans. Knowl. Discov. Data 3 (1) (2009).
[8] F. de Morsier, D. Tuia, M. Borgeaud, V. Gass, J.-P. Thiran, Cluster validity measure and merging system for hierarchical clustering considering outliers, Pattern Recogn. 48 (4) (2015) 1478–1489.

[9] A. Amini, T.Y. Wah, H. Saboohi, On density-based data streams clustering algorithms: a survey, J. Comput. Sci. Technol. 29 (1) (2014) 116–141.
[10] M. Erisoglu, N. Calis, S. Sakallioglu, A new algorithm for initial cluster centers in k-means algorithm, Pattern Recogn. Lett. 32 (14) (2011) 1701–1705.
[11] D. Arthur, S. Vassilvitskii, k-means++: the advantages of careful seeding, Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics Philadelphia, PA, USA. 2007, pp. 1027–1035.
[12] C. Goutte, L.K. Hansen, M.G. Liptrot, Rostrup feature-space clustering for fMRI meta-analysis, Hum. Brain Mapp. 13 (3) (2001) 165–183.
[13] C.A. Sugar, G.M. James, Finding the number of clusters in a data set: an information theoretic approach, J. Am. Stat. Assoc. 98 (2003) 750–763.
[14] M. Bilenko, S. Basu, R.J. Mooney, Integrating constraints and metric learning in semi-supervised clustering, in: Proceedings of the Twenty-first International Conference on Machine Learning (ICML'04), New York, NY, USA, 2004, page 11.
[15] T.N. Tran, K. Drab, M. Daszykowski, Revised DBSCAN algorithm to cluster data with dense adjacent clusters, Chemom. Intell. Lab. Syst. 120 (2013) 92–96.
[16] M. Filippone, F. Camastra, F. Masulli, S. Rovetta, A survey of kernel and spectral methods for clustering, Pattern Recogn. 41 (1) (2008) 176–190.
[17] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, AAAI Press, 1996, pp. 226–231.
[18] M. Ankerst, M.M. Breunig, H.-P. Kriegel, J. Sander, OPTICS: Ordering points to identify the clustering structure, in: ACM SIGMOD International Conference on Management of Data, ACM Press, 1999, pp. 49–60.
[19] D. Klein, S.D. Kamvar, C.D. Manning, From instance-level constraints to space-level constraints: making the most of prior knowledge in data clustering, Proceedings of the Nineteenth International Conference on Machine Learning (2002), p. 307–314.
[20] K. Wagstaff, C. Cardie, S. Rogers, S. Schroedl, Constrained K-means clustering with background knowledge, Proceedings of the Eighteenth International Conference on Machine Learning (2001) 577–584.
[21] I. Davidson, S.S. RAVI, Hierarchical clustering with constraints: theory and practice, Knowl. Discov. Data Min. 14 (2007) 1.
[22] K. Taşdemir, B. Yalçin, I. Yildirim, Approximate spectral clustering with utilized similarity information using geodesic based hybrid distance measures, Pattern Recogn. 48 (4) (2015) 1465–1477.
[23] T.F. Covões, E.R. Hruschka, J. Ghosh, A Study of K-means-based algorithms for constrained clustering, Intell. Data Anal. 17 (3) (2013) 485–505.
[24] Y. Bar-Shalom, P.K. Willett, X. Tian, Tracking and Data Fusion: A Handbook of Algorithms, YBS Publishing, 2011.
[25] B. Long, P.S. Yu, Z. Zhang, A general model for multiple view unsupervised learning, SDM (2008) 822–833.
[26] T. Xia, D. Tao, T. Mei, Y. Zhang, Multiview spectral embedding, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 40 (6) (2010) 1438–1446.
[27] M. Hua, J. Pei, Clustering in applications with multiple data sources—a mutual subspace clustering approach, Neurocomputing 92 (2012) 133–144.
[28] S. Děroski, Multi-relational data mining: an introduction, SIGKDD Explor. Newsl. 5 (1) (2003) 1–16.
[29] X. Zhu, Semi-supervised learning literature survey. Tech. Rep. 1530, Computer Sciences, University of Wisconsin-Madison, 2005.
[30] S. Basu, A. Banerjee, R.J. Mooney, Active semi-supervision for pairwise constrained clustering, in: Proceedings of the SIAM International Conference on Data Mining (SDM-2004), Lake Buena Vista, FL, April, 2004, pp. 333–344.
[31] F. Bonchi, A. Gionis, A. Ukkonen, Overlapping correlation clustering, Knowl. Inf. Syst. 35 (1) (2013) 1–32.
[32] M.K. Goldberg, M. Hayvanovych, M. Magdon-Ismail, Measuring similarity between sets of overlapping clusters, IEEE Second International Conference on Social Computing (2010) 303–308.
[33] Y.-L. Chen, H.-L. Hu, An overlapping cluster algorithm to provide non-exhaustive clustering, Eur. J. Oper. Res. 173 (3) (2006) 762–780.
[34] G. Cleuziou, An extended version of the k-means method for overlapping clustering, International Conference on Pattern Recognition (2008) 1–4.
[35] N.A. Yousri, M.S. Kamel, M.A. Ismail, A distance-relatedness dynamic model for clustering high dimensional data of arbitrary shapes and densities, Pattern Recogn. 42 (7) (2009) 1193–1209.
[36] A. Pérez-Suárez, J.F. Martínez-Trinidad, J.A. Carrasco-Ochoa, J.E. Medina-Pagola, An algorithm based on density and compactness for dynamic overlapping clustering, Pattern Recogn. 46 (11) (2013) 3040–3055.
[37] C.-E. ben N'Cir, G. Cleuziou, N. Essoussi, Generalization of c-means for identifying non-disjoint clusters with overlap regulation, Pattern Recognit. Lett. 45 (2014) 92–98.
[38] T. Li, J.M. Corchado, S. Sun, J. Bajo, Clustering for filtering: multi-target detection and estimation using multiple/massive sensors, Inf. Sci. 388–389 (2017) 172–190.
[39] T. Li, J.M. Corchado, J. Bajo, S. Sun, Multi-source Data Clustering, 18th International Conference on Information Fusion, Washington, D.C., U.S, July 6–9, 2015.
[40] P. Kuila, P.K. Jana, A novel differential evolution based clustering algorithm for wireless sensor networks, Appl. Soft Comput. 25 (2014) 414–425.
[41] S.A. Sert, H. Bagci, A. Yazici, MOFCA: multi-objective fuzzy clustering algorithm for wireless sensor networks, Appl. Soft Comput. 30 (2015) 151–165.
[42] G. Hamerly, J. Drake, Accelerating Lloyd's algorithm for k-means clustering, in: M. Emre Celebi (Ed.), Partitional Clustering Algorithms, 2014, pp. 41–78.
[43] M. Inaba, N. Katoh, H. Imai, Applications of weighted Voronoi diagrams and randomization to variance-based k-clustering, Proceedings of 10th ACM Symposium on Computational Geometry (1994) 332–339.
[44] D. Schuhmacher, B.T. Vo, B.N. Vo, A consistent metric for performance evaluation in multi-object filtering, IEEE Trans. Signal Process. 56 (8) (2008) 3447–3457.