18th International Conference on Information Fusion
Washington, DC - July 6-9, 2015

Multi-source Data Clustering

Tiancheng Li, Juan M. Corchado
BISITE group, Faculty of Science
University of Salamanca
37007, Salamanca, Spain
{t.c.li, corchado}@usal.es

Javier Bajo
Department of Artificial Intelligence
Technical University of Madrid
28660, Madrid, Spain
jbajo@fi.upm.es

Shudong Sun
School of Mechanical Engineering
Northwestern Polytechnical University
710072, Xi'an, China
sdsun@nwpu.edu.cn

Abstract* - *In this paper, we consider a special multi-source data clustering problem for which the data-points from the same source cannot be grouped into the same cluster, namely cannot link (CL) constraint, and the sizes of the generated clusters are subject to maximum thresholds. No prior information is given about the level of clutter (namely noisy data) or the number of clusters. Particularly, the clusters might be closely distributed in the space (overlapping clusters) with one another and have to be carefully partitioned to meet the CL constraint. This particular CL constrained data mining problem corresponds to a significant problem of multi-sensor data fusion (MSDF) raised in the multi-target detection context. A novel clustering method as well as the online parameter learning procedure is proposed for this particular dataset model. Clustering results are provided to demonstrate the validity of the present approach.

Keywords: Clustering; data fusion; target detection.

1 Introduction

One of the most fundamental tasks in data mining is to identify manageable, meaningful groups of data-points of "similar" or "relatively homogeneous" attributes/features, as well as to exclude noisy data, namely cluster analysis or clustering, where a group that contains similar data-points is called a cluster. In the past several decades, clustering has been a very active and vibrant research topic and is still experiencing very rapid growth. Simply for numerical data, there are many algorithms proposed based on different data models, including the popular hierarchical methods [1, 2], k-means clustering [2], spectral clustering [3], subspace clustering [4] and density-based clustering [5, 6], just to name a few. In this paper, we are interested in an efficient clustering approach for multi-sensor data fusion (MSDF) raised in the multi-object detection problem.

Clustering is usually taken as "unsupervised" as no information is available concerning the association of data items to any predefined group. Nevertheless, in many problems including the multi-sensor data fusion problem to be solved in this paper, more or less a prior information is available and therefore can benefit the clustering; see semi-supervised clustering [7]. The semi-supervised clustering is demonstrated to be more effective and efficient in which, one would like to take into account all the available information [1]. Clearly, a prior information such as the potential number of clusters is critical both to the clustering results and to the clustering speed. For example, if the number of clusters, namely the parameter k, can be correctly predefined, the k-means will be particularly preferable for many cases [2]. Cluster center initialization can also significantly affect the speed of convergence and the final output [8, 9].

Particularly, must link (ML) and the converse cannot link (CL) constraints, may be specified/applied for encoding a prior knowledge, namely constrained clustering. The former corresponds to the requirement that two objects should be assigned to the same cluster label, whereas the cluster labels of two objects participating in the latter should be different. The addition of constraints allows users to incorporate domain expertise into the clustering process by explicitly specifying the desirable properties in the final clustering outcome, e.g. constrained k-means clustering [10], constrained hierarchical clustering [11] and the graph-cut based clustering with cluster size (value of the cluster) constraint [3], see also [12, 13, 14].

In this paper, we are facing with a multi-source dataset in which the data originates from n i.i.d (independent and identically distributed) sensors, leading to a strong CL constraint that 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 they are very closely distributed. This multi-source data clustering (MSDC) problem is extracted from a significant engineering problem involved in multi-sensor multi-target tracking [15, 16]. All the sensor data except an unknown number of outlier belong to an unknown number of clusters, each of which corresponds a target of interest. This multi-source i.i.d problem is different to and more specific than the similar-called multi-source/view or subspace data clustering problem proposed in [17-20], where the i.i.d condition does not hold and that data from different sources/views can be heterogeneous.

The CL constraint will significantly affect the clustering output. As a result, the size of the cluster C_i (i.e. the number of data-points in the cluster) is subject to an upper threshold ($|C_i| \leq n$). Data-points from the same source have to be partitioned into different clusters if they are very closely distributed (namely overlapping clusters).

This multi-source CL constraint here is different to the traditional data point-pairwise CL instance constraint [10-14]. It resembles somewhat semi-supervised learning [7, 21], although differently in that training is carried out. Particularly, overlapping clusters are involved if the corresponding targets are closely distributed. In existing

research, mixed data between them are to be considered as outliers, to belong to one or multiple clusters or to belong to a given cluster to a certain degree; see [22-24]. All of these however do not meet our requirements subject to the multi-source CL constraint.

Despite that several attempts have been made to employ mature clustering approaches within filtering [25] and target detection [26, 27] and to address a variety of sensor fusion issues [28], they offer no solution for this particular CL constrained MSDF problem. This paper aims to formulate this problem and proposes a novel clustering method as called multi-source *n*-points clustering.

The rest of the paper is organized as follows. Section 2 formulates the clustering problem model. Section 3 presents the *n*-points clustering idea. The simulation results and discussion are shown in Section 4. Finally, Section 5 concludes the paper.

2 Problem formulation

2.1 Multi-source data fusion

We start from an interesting multi-source data fusion problem raised in the task of multi-target detection in the planar position space. Consider that a sensor is used to monitor a scenario containing an unknown number of targets of interest e.g. infrastructures or crops in remote sensing images or diseased cells or cracks in X-ray scanning tomographic images, to name a few. This scenario can be modeled by the following assumptions:

(A.1) each target generates observation reports (data-points of interesting) independently of others and one target generates no more than one observation at each scan; this forms the CL constraint that observation reports in the same scan are independent of each other and cannot be linked.

(A.2) the observations generated from targets are coupled with unimodal noise, e.g. zero-mean Gaussian;

(A.3) the sensor may miss-detect targets with a probability (miss-detection probability) that is often related with the distance between the target and the sensor;

(A.4) there is clutter within the observations received by the sensor, which are noisy observations generated from no target; the clutter is assumed to be generated randomly, independently of the targets, having a local distribution density that is significantly lower than the density of the 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 can be relaxed to accommodate a scenario where targets are moving with a relatively low speed that is insignificant as compared to the scanning frequency of the sensor (therefore, their movement is very small between different scans, similar to the case of static targets), or where massive homogeneous sensors are used to scan the scenario synchronously and their observation noise on the same target are approximately equivalent, e.g. [15, 16]. We do not intend to detail these scenarios, which would distract from the main contribution of this paper. Both i.i.d. multi-scan and multi-sensor can be collectively referred to as multi-source. The problem is referred to as CL-constrained MSDF for which the observations from the same source cannot be associated to the same target. To solve this problem, we propose a novel CL-constrained clustering method for which the input data-points from the same source cannot belong to the same cluster.

For instance, Fig. 1 gives the real observation reports of different targets (colored data-points) and clutter (black data-points) collected in 50 sources under assumptions (A.1-4), which can be mapped into the same planar x-y space as shown in Fig.2 (a). The goal of the required clustering is to distinguish these groups of observations from each other and from the clutter, as shown in Fig.2 (b). The number of targets and their positions can then be further estimated [15, 16], which is beyond the scope of this paper. Here we are only interested in associating the observation data-points into different clusters corresponding to different individual targets.

Intuitively, the observations of a particular target are subject to a unimodal distribution and so will concentrate locally (around the true state of the target) while the clutter will not. Therefore, the data distribution density as well as the CL constraint form the two key factors to partition the observations of targets from clutter and from that of each other in our approach.

2.2 Problem formulation

The MSDF problem can be formulated as a strict CL-constrained clustering problem. Consider a dataset X consisting of data points

$$x_i = [x_{i1}, x_{i2}, \dots, x_{id}] \in P, i = 1, \dots, N \quad (1)$$

where d is the dimensionality, P is the parameter space, $x_{il} \in P_l$ is the ith attribute or feature, N is the number of data-points to be clustered. It should be noted that the data-points are not specified to be numeric or categorical values. However, in this paper we focus on the spatial data-points defined on numeric values.

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

$$\begin{aligned} X &= \{S_1, S_2, \dots S_n\} \\ &= \{x_1^1, x_2^1, \dots x_{m_1}^1, x_1^2, x_2^2, \dots x_{m_2}^2, \dots, x_1^n, x_2^n, \dots x_{m_n}^n\} \end{aligned} \quad (2)$$

where n is the number of sources (scans or sensors as stated in the MSDF). The i.i.d. condition specifies that different sources of data-points are (approximately) subject to the same spatial distribution, to say q, written as

$$\forall: s \in \{1,2, \dots, n\}, S_s \sim q \quad (3)$$

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

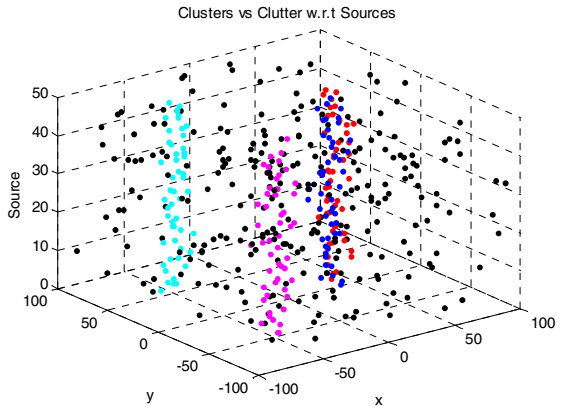

Figure 1. Multi-source i.i.d data (black dots represent clutter while different colors mark observations of different targets)

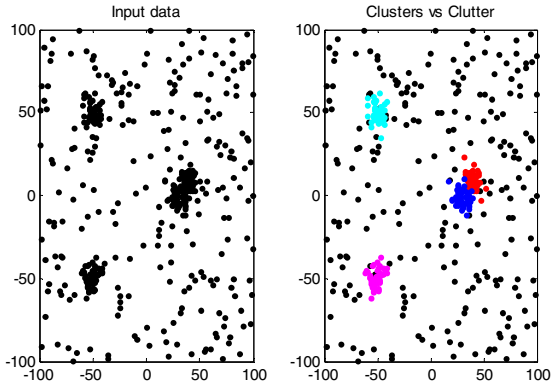

Figure 2. Multi-source i.i.d data mapped in the planar space (a) all the data (b) desired clustering result (ground truth)

$$\forall i,j \in \{1,2,\dots,m_s\}, s \in \{1,2,\dots,n\}: c_{\neq}(x_i^s, x_j^s) \quad (4)$$

where $c_{\neq}(x_i^s, x_j^s)$ means that x_i^s, x_j^s cannot belong to the same cluster. To realize this, the distance between data-points from the same source can be re-defined 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 all of these noisy data-points as a set of outliers C_o. The union of these subsets is equal to a full data set:

$$X = C_1 \cup C_2 \cup \dots \cup C_k \cup C_o \quad (5)$$

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

$$\forall i,j \in \{1,2,\dots k,o\}: C_i \cap C_j = \Phi \quad (6)$$

But, this is violated in some other clustering methods.

As addressed so far, the clustering goal 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. In addition to the CL constraint, another challenge for clustering comes from the unknown number of clutter and targets/clusters. Particularly, the clutter could be significant as the number of noisy data can be larger than that of the real data.

2.3 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 constraints

$$\forall i \in \{1,2,\dots k\}: |C_i| \lesssim n \quad (7)$$

where $|C_i|$ means the number of data-points in cluster C_i, namely the size of the cluster, $\lesssim$ means being slightly smaller than or equal to, and n is the upper limit which can be specified as the total number of sources. As one cluster corresponds to one target, the notation i can also be used to index a target.

More precisely, we can estimate the expectation of the size of each cluster, namely the number of observations received from each target i (more precisely to say, target at position i). With regard to the general condition (A.3), we denote the detection probability of the sensor on target i as $p_D(i) \leq 1$ which is usually a function defined on the position of the cluster/target, then we can statistically have

$$E(|C_i|) = p_D(i) \times n \quad (8)$$

Since the accurate position of targets is unknown, $p_D(i)$ is roughly estimated based on the potential area containing target i. In a simple case, the detection probability of the sensor $p_D(i)$ is a constant that is independent of targets/clusters, i.e. $\forall i,j \in \{1,2,\dots k\}: E(|C_i|) = E(|C_j|)$.

The challenge arising in the multi-sensor case is that the view-scopes of the sensors can be different, causing a different number of sensors observing different areas with interaction. Denoting the total number of sensors as n, the number of the sensors whose view fields cover area/cluster i as n_i and the detection probability of sensor s in the area of target i as $p_{D,s}(i) \leq 1$, (8) will be extended to

$$E(|C_i|) = \sum_{s=1}^{n_i} p_{D,s}(i) \leq n_i \quad (9)$$

This accommodates a more general case for different-positioned heterogeneous sources.

Owning to the CL constraint, a cluster has to be divided into multiple individual sub-clusters if its size exceeds threshold n_i. Here in our approach, the number of sub-clusters in each connected area/cluster C_i is estimated as

$$k_i = \left[\frac{|C_i|}{E(|C_i|)}\right] \quad (10)$$

where $[\cdot]$ represents the rounding operation which gives the nearest integer to the content. In practice without exact knowledge of $p_D(i)$ or $p_{D,s}(i)$ to calculate $E(|C_i|)$ as given in (8)/(9), one can use the average number of data-points that are both from the same source (for all sources) and grouped into the cluster C_i, to estimate the number of sub-clusters that shall be formed. This can be written as

$$k_i = \left[\frac{1}{n_i}\sum_{s=1}^{n_i} |\{x_j^s | j \in \{1,2,\dots m_s\}, x_j^s \in C_i\}|\right] \quad (11)$$

This is very helpful to determine the number of close-distributed targets.

In both calculations of (10) and (11), the parameter n_i is needed which does not need to be very accurate when the level of clutter is low and targets are well distant from each other. Generally, n_i is known or can be approximated on basis of the known configuration of the sensors including their located positions and observing scopes. The following section will detail the proposed CL constrained *n*-points clustering method with an online learning method to estimate the key required parameter.

3 Multi-source *n*-points clustering

3.1 Proposed clustering solution

As previously mentioned, 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 concentrate significantly locally are more likely to be the observations from targets; this inherently resembles the density-based clustering in that 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. Overall, we propose a more efficient clustering solution based on the CL constraints (10) or (11) as shown in Algorithm 1, consisting of two steps 1) searching across sources to identify different groups of closely connected data-points, and 2) for each group of an adequate number of connected data-points (e.g. more than $0.8 \times E(|C_i|)$), determining whether to form it as a single cluster or divide it into multiple sub-clusters.

To carry out the first step, which connects closely distributed data-points across different sources, a distance parameter d_i is needed to distinguish close data-points from clutter, where i indicates different clusters/targets.

Remark 1. Parameter d_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 can be determined with respect to the standard deviation σ_i of the cluster distribution, e.g. $d_i = (1\sim3)\sigma_i$ 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 d_i. If the observation noise is infeasible in practice, a constant value can be estimated from the dataset as shown in section 3.2 but there is no clue/information to specify different values for different clusters/targets.

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 as the final output.

Remark 2. Given the number of sub-clusters to divide from an oversized cluster as shown in (10) or (11), existing clustering methods such as *k*-means are readily able to obtain sub-clusters of approximately equivalent size. However, this might violate the CL constraint somewhat as data-points from the same source might be grouped into the same sub-cluster. Therefore, we must apply the CL constraint (e.g. by setting the distance between two data-points that are from the same source as infinite) in the clustering process using e.g. the CL constrained *k*-means [10] to avoid violating the constraint.

By using the presented procedure, the clustering results for the data-set given in Fig.1 are shown in Fig.3. In the figure, clustered data-points are circled with different colors. Again, the color of the circles are 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 d_i; see **Remark 1** and **Algorithm 3**. Any group C_i containing more than $0.8 \times E(|C_i|)$ data-points forms a cluster; here the parameter 0.8 is only a reference and can be chosen roughly between $0.6\sim0.95$.

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**.

Algorithm 2 Partitioning overlapping clusters

Step 2.1 Identify the source S that contributes the least number of data-points to the underlying oversize 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 $n_i \geq 0.8 \times E(|C_i|)$;

Step 2.3 Repeat Step 2.2 till all the data-points in source S are grouped into separate sub-clusters;

Step 2.4 Do Step 2.1-2.3 in the rest sources excluding S.

Step 2.5 The procedure stop when totally k_i sub-clusters are formed.

3.2 Online estimating parameter *d*

A constant value of the threshold d can be estimated through unsupervised learning of the data; see algorithm 3 given below. This is calculated as:

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

where

$$L = \operatorname{argmax}_i |S_i| \tag{13}$$

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

where L represents the source containing the largest number of data-points, $T\text{th}\min_{m,p\ s.t.\ G} d(x_j^L, x_m^p)$ gives the T th

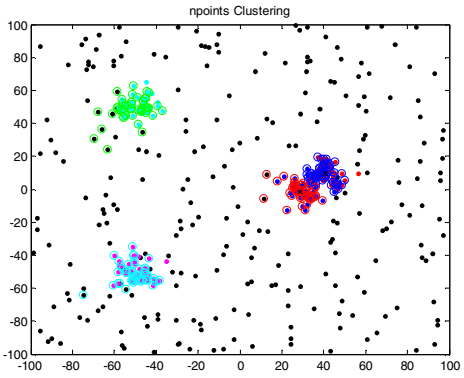

Figure 3. Clustered data-points (circle 'o' with different color), corresponding to Fig.1 & 2.

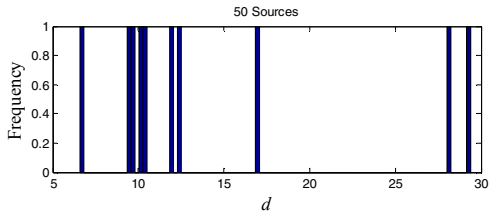

Figure 4. Rank of d_j, corresponding to Fig.1 & 2

smallest value of the Euler distance $d(x_j^L, x_m^p)$ between data-points x_j^L and x_m^p in the parameter space for any m, p satisfying condition G.

Given that the neighbors are ranked according to their distances to the underlying data-point, parameter T specifies one of the neighbors for calculating its distance to the data-points as an average estimate of the "neighbor distance" (it resembles the neighbor radius parameter ϵ used in DBSCAN). T does not need to be precise and as a rule of thumb it can be chosen between $\max\left(\frac{n_i}{5}, 2\right)$ and $0.9 \times n_i$, e.g. $[n_i/2\,]$ or even just a constant number e.g. 5. For instance (choosing $T = 5$), the rank of d_j (as defined in Step 2 of Algorithm 3) for the dataset given in Fig. 1&2 is given in Fig.4. As shown, the minimum $d_j(T, L)$ is roughly 6.6 . Therefore, we can choose $d = 5{\sim}8$.

Algorithm 3 Estimating the distance threshold d

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

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 d.

3.3 Null/Full cluster

It is possible that all the data-points are clutter in the scenario or, conversely, are from a single target. If the parameter d is (approximately) given, it will be easy to identify them, as very few data-points can be clustered if all the data are clutter or almost all the data-points will be clustered to one if all the data-points are from the same target. However, if the parameter d is not given, it is a dilemma to distinguish them unless further information about the level of clutter is provided. Here, we have omitted this rare situation in our current work as we will treat this dilemma to be no cluster existing (full of clutter).

4 Simulations

Although existing clustering methods offer no explicit mechanism to deal with the multi-source CL constraint, we still implement typical DBSCAN and *k*-means methods in their best possible parameter setting for comparison with our multi-source (MS) *n*-points clustering.

4.1 Given parameter d

For the *k*-means clustering, the correct parameter $k = 6$ is used, which puts its performance into the best possible situation. Furthermore, the parameter k is known to be hard to choose when not given by external constraints and is very critical to the clustering performance. Automatically determining the number of clusters has been one of the most difficult problems in data clustering. Most methods focus on model selection or matching. In practice, clustering algorithms are run with different values of *k*; the best value of *k* is then chosen based on a predefined criterion [29, 30].

The DBSCAN needs two parameters ϵ and m. Parameter ϵ gives the neighborhood radius for which we set it as $\epsilon = 10$. Parameter m gives the minimum number of points in a neighborhood for its inclusion in a cluster for which we adopt two different values $m = 2, 4$. The simulation results for $n = 20, 50$ are given in Fig.5 and 6 respectively. The color of the circles (which represents different clusters) are assigned randomly in each run and is independent of the color of the data-points (which represents the true clusters for different individual targets).

The results show the obvious advantage of our approach over other methods which are unable to deal with overlapping/hinged clusters. Particularly, the basic *k*-means is unable to efficiently deal with clutter and is insensible to the density of data-points. Advanced implementations of the basic *k*-means and DBSCAN based on the multi-source constraint might achieve much better results but will also cause more computation.

The average computing time of different clustering methods over 100 Monte Carlo trials is given in Table 1 for the dataset given in Fig.5 and 6. It shows that the proposed CL constrained n-points clustering is somewhat slower than the others but is still quite fast. Regarding that the CL constraint that has been fully satisfied, the proposed clustering approach is arguably computationally fast.

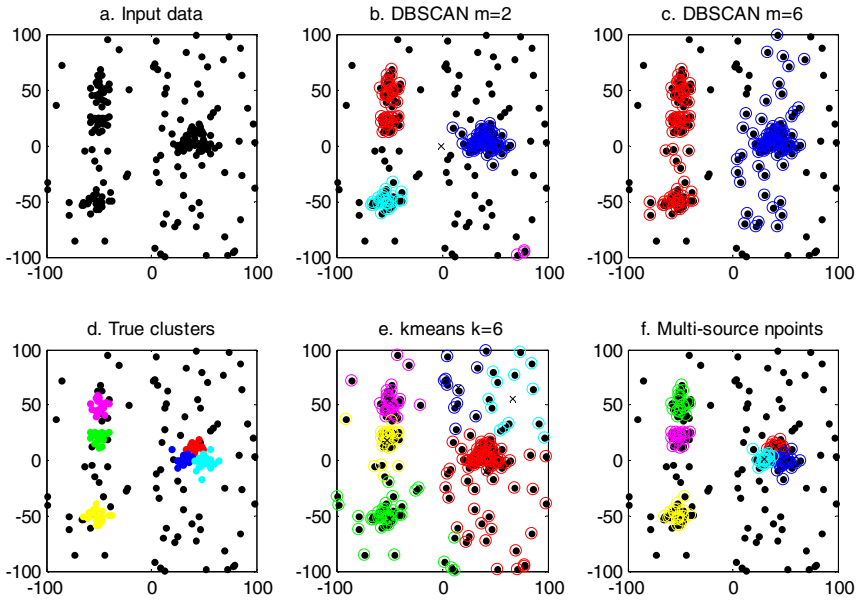

Figure 5. Outcomes of different clustering methods on data from 20 i.i.d sources

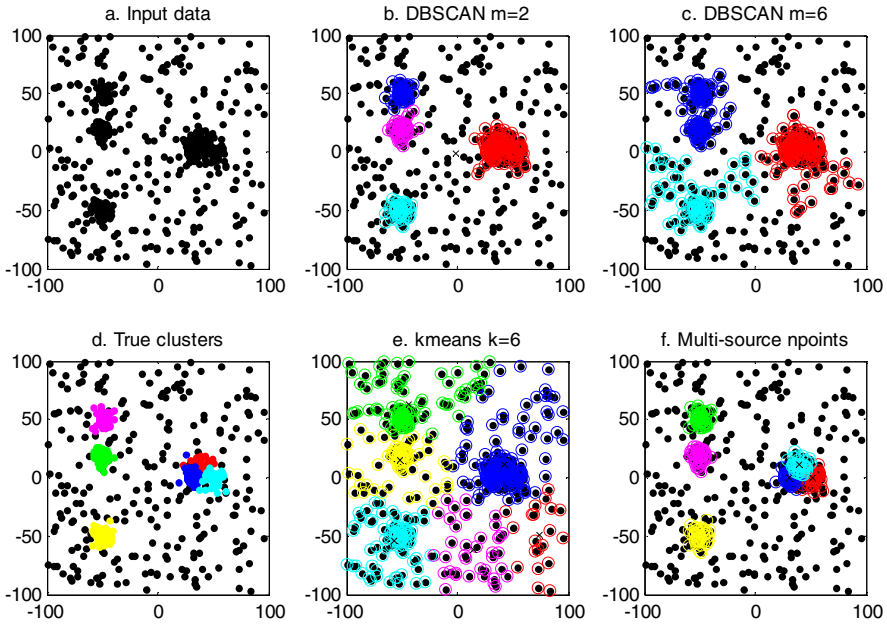

Figure 6. Outcomes of different clustering methods on data from 50 i.i.d sources

Table 1 Computing time of different clustering methods (s)

	k-means	DBSCAN 2	DBSCAN 6	MS n-points
Fig.5	0.0075	0.0156	0.0119	0.0436
Fig.6	0.01363	0.0422	0.0399	0.0834

4.2 Unknown parameter *d*

Based on the same dataset as given in the last section, we assume that parameter *d* is unknown in the proposed multi-source *n*-points clustering, which has to be learned online through algorithm 2 from the dataset. The upper and bottom sub-figures of Fig. 7 give the distribution of *d* for the

dataset shown in Fig. 5 ($n = 20$) and 6 ($n = 50$) respectively. The outcomes for estimating parameter d are approximately 6.6 and 4.6 respectively. Based on these estimates, the clustering results of the multi-source n-points clustering are given in Fig. 8. Compared with the results shown in the last section, the estimated parameters are shown to be suitable and well qualified to provide good clustering results, which is very close to the results shown in Fig.4 and 5. This demonstrates that our online learning procedure for parameter d is effective.

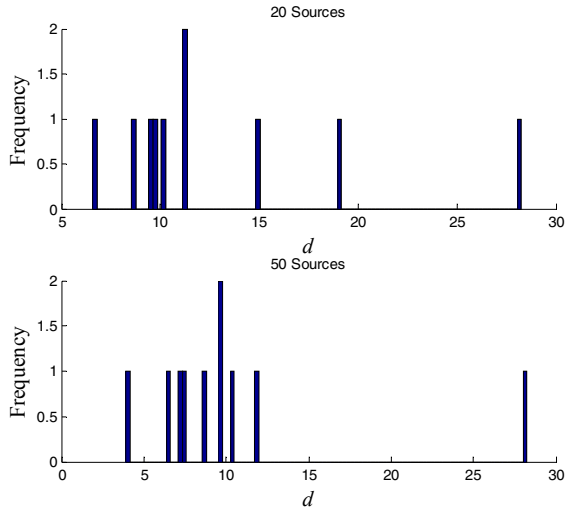

Figure 7. Rank of d_j, w.r.t. Fig.5 and 6 respectively

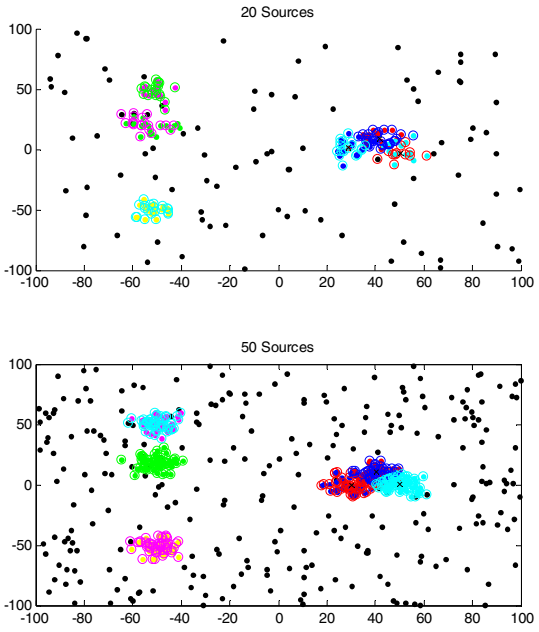

Figure 8. Clustering results using online learned parameter d

5 Conclusion

We have established a multi-source data fusion-oriented clustering model where the data are coming from different sources and the data from the same source cannot group into the same cluster (namely cannot link constraint). The dataset may be affected by a high level of clutter (namely noisy data), clusters may be overlapped with each other and there is no prior information about the number of potential clusters, posing significant challenges for cluster analysis.

Based on this particular albeit significant problem, a new multi-source n-points clustering approach is proposed which resembles the density-based clustering algorithms in which clusters are high density regions in the feature space separated by low density regions (of clutter). Particularly, the proposed clustering method is able to partition closely connected clusters based on the source-label of data-points reasonably and accurately. In addition, an online learning procedure is proposed for estimating the key parameter required by the clustering approach, thus allowing the clustering method to be used for more general cases with little prior knowledge. Simulation results on synthetic data are presented to demonstrate the validity of the present approach with either known or unknown parameter.

The proposed clustering method has a unique potential for massive sensor data fusion to carry out multi-target detection and state-estimation in dynamic cluttered environments. Our future work will employ the proposed multi-source n-point clustering method for the challenging problem of multiple (massive) sensor multiple (massive) object detection and estimation.

Acknowledgments

This work has been supported by Ministry of Economy and Finance of Spain (No. TIN2012-36586-C03-03), FEDER funds and National Natural Science Foundation of China (No. 51475383). Tiancheng Li's work has been supported by the Excellent Doctorate Foundation of Northwestern Polytechnical University and the Postdoctoral Fellowship of the University of Salamanca.

References

[1] A.K. Jain, "Data clustering: 50 years beyond K-means," Pattern Recognition Letters, vol. 31, no. 8, 2010, pp. 651-666.

[2] J.A. Hartigan, Clustering algorithms. John Wiley & Sons, Inc., 1975.

[3] J. Shi, M. Malik, Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Machine Intell. 22, 888–905, 2000.

[4] E. Elhamifar, R. Vidal, "Sparse subspace clustering: algorithm, theory, and applications," IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 11, pp. 2765–2781, 2013.

[5] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, "A density-based algorithm for discovering clusters in large spatial

databases with noise," Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, AAAI Press. pp. 226–231, 1996.

[6] M. Ankerst, M. M. Breunig, H.-P. Kriegel, J. Sander "OPTICS: Ordering Points To Identify the Clustering Structure," ACM SIGMOD international conference on Management of data. ACM Press. pp. 49–60, 1999.

[7] N. Grira, M. Crucianu, N. Boujemaa, "Unsupervised and Semi-supervised Clustering: a Brief Survey," in: A Review of Machine Learning Techniques for Processing Multimedia Content, Report of the MUSCLE European Network of Excellence (6th Framework Programme), 2005.

[8] M. Erisoglu, N. Calis, S. Sakallioglu, "A new algorithm for initial cluster centers in k-means algorithm," Pattern Recognit. Lett., vol. 32, no.14, pp. 1701–1705, 2011.

[9] D. Arthur, S. Vassilvitskii, "k-means++: the advantages of careful seeding," In Proc. the 18th annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics Philadelphia, PA, USA. pp. 1027–1035.

[10] K. Wagstaff, C. Cardie, S. Rogers, S. Schroedl, "Constrained K-means Clustering with Background Knowledge," In Proc. the 18th International Conference on Machine Learning, 2001, p. 577–584.

[11] I. Davidson, S. S. Ravi, "Hierarchical clustering with constraints: Theory and practice," Knowledge Discovery and Data Mining, 14, 1. 2007.

[12] T. Lange, M.H. Law, A.K. Jain, J. Buhmann, "Learning with constrained and unlabelled data," IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognition, vol. 1, pp. 730–737, 2005.

[13] S. Basu, I. Davidson, K. Wagstaff, "Constrained Clustering: Advances in Algorithms, Theory and Applications," Data Mining and Knowledge Discovery, vol. 3, 2008.

[14] I. Davidson, S. S. Ravi, "The complexity of non-hierarchical clustering with instance and cluster level constraints," Data Min. Knowl. Disc. vol.14, pp. 25–61, 2007.

[15] T. Li, J. M. Corchado, J. Bajo and G. Chen, Multi-target detection and estimation with the use of massive independent, identical sensors, Proc. SPIE 9469, Sensors and Systems for Space Applications VIII, 94690G, Baltimore, Maryland, US, April 20-24, 2015.

[16] T. Li, J. M. Corchado, J. Bajo and S. Sun, "Clustering for filtering: Multi-object detection and estimation using multiple/massive sensors," Submitted, 2015.

[17] B. Long, P.S. Yu, Z. Zhang, A general model for multiple view unsupervised learning, in: SDM, 2008, pp. 822–833

[18] T. Xia, D. Tao, T. Mei, Y. Zhang, "Multiview Spectral Embedding," IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol.40, no.6, pp.1438-1446, Dec. 2010.

[19] M. Hua, J. Pei, "Clustering in applications with multiple data sources—A mutual subspace clustering approach," Neurocomputing, vol. 92, pp. 133-144, 2012

[20] J. Ye, Z. Zhao, M. Wu, Discriminative k-means for clustering, In Proc. the 20th Annual Conference on Advances in Neural Information Processing Systems 2007.

[21] S. Basu, A. Banerjee, R. J. Mooney, Active Semi-Supervision for Pairwise Constrained Clustering, In Proc. the SIAM International Conference on Data Mining (SDM-2004), pp. 333-344, 2004.

[22] 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 Recognit., vol.42, no.7, pp. 1193–1209, 2009.

[23] H. Kim, J. Lee, "Clustering based on Gaussian processes," Neural Comput., vol.19, no.11, pp. 3088–3107, 2007.

[24] J. S. Wang, J-C Chiang, "A cluster validity measure with outlier detection for support vector clustering," IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol.38, no.1, pp. 78-89, 2008.

[25] T. Li, S. Sun, T. P. Sattar and J. M. Corchado, "Fight sample degeneracy and impoverishment in particle filters: A review of intelligent approaches," Expert Systems with Applications, vol.41, no. 8, pp. 3944-3954, 2014.

[26] J. Schubert, H. Sidenbladh, "Sequential clustering with particle filters-estimating the number of clusters from data," In Proc. the 8th International Conference on Information Fusion, pp.1-8, 25-28 July 2005.

[27] E. Hanusa, D. Krout and M.R. Gupta, "Clutter rejection by clustering likelihood-based similarities," In Proc. the 14th International Conference on Information Fusion, pp.1-6, 5-8 July 2011.

[28] M. Cetin, L. Chen, J.W. Fisher, A. T. Ihler, R.L. Moses, M.J. Wainwright, A.S. Willsky, "Distributed fusion in sensor networks," IEEE Signal Processing Magazine, vol.23, no.4, pp.42-55, 2006.

[29] C. A. Sugar, G. M. James, "Finding the number of clusters in a data set: An information theoretic approach," Journal of the American Statistical Association, vol. 98, pp: 750–763, 2003.

[30] P. J. Rousseuw, "Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis," Computational and Applied Mathematics, vol. 20, pp. 53–65, 1987.