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

OSPA Barycenters for Clustering Set-Valued Data

Marcus Baum[*§], **Balakumar Balasingam**[§], **Peter Willett**[§], and **Uwe D. Hanebeck**[*]
[*] Karlsruhe Institute of Technology (KIT), Germany.
[§] University of Connecticut, CT, USA.
Email: baum@engineer.uconn.edu, bala@engr.uconn.edu, willett@engr.uconn.edu, uwe.hanebeck@kit.edu

***Abstract*—We consider the problem of clustering set-valued observations, i.e., each observation is a set that consists of a finite number of real vectors. For this purpose, we develop a k-means algorithm that employs the OSPA distance for measuring the distance between sets. In particular, we introduce a novel alternating optimization algorithm for the OSPA barycenter of sets with varying cardinalities that is required for calculating cluster centroids efficiently. The benefits of clustering set-valued data with respect to the OSPA distance are illustrated by means of simulated experiments in the context of target tracking and recognition.**

***Keywords*—Clustering, k-means, set-valued data, point sets, OSPA distance, Wasserstein distance, barycenter.**

I. Introduction

Cluster analysis [25], [26] arises in many fields such as machine learning, data mining, image processing, signal processing, and sensor data fusion. The objective of cluster analysis is to find structure in a data set by partitioning the observations into clusters that contain (in some way) similar observations. One of the most common clustering techniques is the k-means algorithm [26]. The standard k-means algorithm clusters vector-valued observations with respect to the squared error. For this purpose, it aims to find clusters that minimize the Within-Cluster Sum of Squares (WCSS), i.e., the sum of squared distances between the data and the cluster centroid. The k-means algorithm starts with initial clusters that are iteratively improved using an assignment and update step. In the assignment step, each observation is associated to the closest cluster centroid. In the update step, the cluster centroids are updated by calculating the arithmetic mean of the cluster members.

In many applications, the available data to be clustered is not in vector format and the squared error criterion cannot be applied. In this context, many variants of k-means have been developed in the past. For example, a variant for spherical data has been developed in [15]; data that consists of intervals instead of vectors has been considered in [21]; and the squared error criterion has been replaced by the more general Bregman divergence in [3]. It was proposed to employ the Wasserstein distance instead of the squared error for comparing histograms in works such as [24], [37], [38]. In general, extending the standard k-means approach to non-Euclidean data types consists of two steps: (1) Define a reasonable similarity measure on the data. (2) Develop an algorithm to compute cluster centroids (also called prototypes). Typically, the second step is the more challenging as it involves a potentially difficult optimization problem.

This work considers the clustering of observations that are set-valued instead of vector-valued, i.e., we are given a collection of point sets, where each point set consists of a finite number of real vectors. Set-valued data frequently occurs in many applications, most especially of interest to us is multi-object tracking with sensors that do not provide object labels, and feature extraction methods that extract features without a specific order. In this work, we propose to cluster set-valued data with respect to the Optimal Sub-Pattern Assignment (OSPA) distance [34]. The OSPA distance, which is a variant of the Wasserstein distance [22], [36], is the standard metric in multi-object tracking. Hence, we argue that clustering with the OSPA distance is especially suitable for multi-object tracking applications.

A. Contributions

The main contributions of this work are the followings: First, we present the first (provably correct) alternating optimization algorithm for computing the OSPA barycenter for sets with varying cardinalities. Second, we integrate this algorithm into the k-means framework in order to cluster set-valued data with respect to the OSPA distance. Finally, the benefits of OSPA-based set clustering are demonstrated by means of two example scenarios in the context of automatic target recognition and unsupervised learning.

B. Related Work

The OSPA barycenter has its origin in the Minimum Mean OSPA (MMOSPA) estimator [20] that is used to optimize multi-target tracking algorithms to perform well with respect to the OSPA distance. In this context, many algorithms have been developed for calculating OSPA barycenters in case all sets have the same cardinalities [6], [8], [12], [13], [20]. Preliminary investigation of the unknown target case can be found in [2]. In computer vision, a related concept called the Wasserstein barycenter recently gained significant interest [1], [14], [27], [32], [38]. For a detailed discussion about the relation between Wasserstein barycenters and MMOSPA estimation, we refer to [7].

Clustering set-valued data is also considered in [31] using a random finite set model [29], where a cluster is modeled as a Poisson random finite set characterized by its mean, covariance, and Poisson rate. This model-based approach is fundamentally different from our distance-based (model-free) approach. The work [19] treats the clustering of set-valued data with a similiarity function that became popular for point set registration [18]. However, the employed similiarity function is not a true metric on sets and it is not suitable if the cardinalities of the cluster centroids are unknown.

There are several works about clustering histograms [24], [37], [38] with the Wasserstein and earth mover's distance

(EMD) [30], [33], [38], which are both related to the OSPA distance. However, histograms are one-dimensional discrete quantities for which centroids can be computed analytically as only the weights are free parameters. For point sets, only the point locations are free parameters so that the calculation of centroids is quite different. In [28], a novel similarity measure named "intersection coefficient" is developed in order to cluster non-ordered discrete data sets.

A further related problem is the unsupervised discovery of structures in point clouds, e.g., from a laser scanner or depth sensor [17]. These approaches typically do not work with the raw point clouds but perform learning on extracted features.

C. Structure

The following Section II introduces the concept of an OSPA barycenter and derives the novel alternating optimization algorithm for its calculation. Based on these results, Section III develops a set-variant of the k-means algorithm. Section IV presents simulations and the conclusions are given in Section V.

II. OSPA Barycenter

An essential component of a clustering algorithm such as k-means is the ability to calculate a cluster centroid (also called prototype). In the standard k-means formulation for vector-valued data, the cluster centroid is given by the mean of the cluster members. However, as we have to deal with sets instead of vectors, it is unclear how to define a reasonable centroid using the arithmetic mean. Instead, we employ the Fréchet mean [10], which generalizes the concept of a mean to general metric spaces: If we agree on a distance for sets – in this case the OSPA distance – the "mean set" is defined as the set that minimizes the squared distance to all other sets.

A. Definitions

The Optimal Sub-Pattern Assignment (OSPA) distance is a distance measure on finite point sets. It has been introduced in [34] for the purpose of evaluating the performance of multi-target trackers. Essentially, the OSPA distance is based on the Wasserstein distance [22], [36], which is important in many areas such as computer vision. For sets with the same cardinality, the OSPA distance and the Wasserstein distance coincide.

In this work, we focus on the OSPA distance of order 2 and the thresholded Euclidean distance as a base distance.

Definition 1 (2-OSPA Distance)**.** The Optimal Sub-Pattern Assignment (OSPA) distance [34] of order 2 between two finite point sets $X = \{\underline{x}_1, \ldots, \underline{x}_m\} \subset \mathbb{R}^d$ and $Y = \{\underline{y}_1, \ldots, \underline{y}_n\} \subset \mathbb{R}^d$ is defined as

$$\text{OSPA}_c(X,Y) := \left(\frac{1}{n} \min_{\pi \in \Pi_n} \sum_{i=1}^{m} d_c(\underline{x}_i, \underline{y}_{\pi(i)})^2 + c^2 \cdot (n-m) \right)^{1/2} \quad (1)$$

if $m \leq n$, and $\text{OSPA}_{p,c}(X,Y) := \text{OSPA}_{p,c}(Y,X)$ if $m > n$, where the base distance

$$d_c(\underline{x}, \underline{y}) := \min\{c, ||\underline{x} - \underline{y}||\} \quad (2)$$

is the Euclidean distance with threshold $c > 0$.

Remark 1. The threshold c is also called cut-off parameter or clamping factor.

Remark 2. In order to evaluate (1), an optimal assignment problem has to be solved. For this purpose, several efficient algorithms are available in literature. For example, the Hungarian algorithm [22], [34] leads to a cubic runtime in $\max(m, n)$.

In order to avoid case distinctions, we will use a slightly rewritten version of (1) as described as follows.

Remark 3. If we introduce the symbol $\underline{\infty}$, which stands for a point in infinity, i.e., $d^c(\cdot, \underline{\infty}) = c$, the squared OSPA distance (1) can be written as

$$\text{OSPA}(X,Y)^2 = \frac{1}{\max\{m,n\}} \min_{\pi \in \Pi_{\max\{m,n\}}} \sum_{l=1}^{\max\{m,n\}} d^c(\underline{y}_l, \underline{x}_{\pi(l)})^2 \ , \quad (3)$$

where $y_l := \underline{\infty}$ if $l > n$ and $\underline{x}_l := \underline{\infty}$ if $l > m$.

Equipped with the OSPA distance, we are now ready to define the Fréchet mean [10] for point sets, which will be called OSPA barycenter. For this purpose, we are given n_p finite point sets

$$\{X_i\}_{i=1}^{n_p} \ , \quad (4)$$

where the i-th point set is given by

$$X_i = \{\underline{x}_{1,i}, \ldots, \underline{x}_{m(i),i}\} \subset \mathbb{R}^d \quad (5)$$

in which $m(i)$ is the number of points in the set. Note that the number of points varies from set to set. The point clouds are associated to weights

$$\{w_i\}_{i=1}^{n_p} \subset \mathbb{R} \quad (6)$$

with $\sum_{i=1}^{n_p} w_i = 1$.

Definition 2 (OSPA Barycenter)**.** The OSPA barycenter [20] for a collection of point clouds (4) with weights (6) is defined as the set $Y = \{\underline{y}_1, \ldots, \underline{y}_n\} \subset \mathbb{R}^d$ with cardinality n that minimizes the *Mean Squared OSPA* distance

$$\text{MOSPA}(Y, \{X_i\}_{i=1}^{n_p}, \{w_i\}_{i=1}^{n_p}) := \sum_{i=1}^{n_p} w_i \cdot \text{OSPA}^c(Y, X_i)^2 = \sum_{i=1}^{n_p} \frac{w_i}{\max\{n, m(i)\}} \cdot \min_{\pi \in \Pi_{\max\{n,m(i)\}}} \sum_{l=1}^{\max\{n,m(i)\}} d^c(\underline{y}_l, \underline{x}^{(i)}_{\pi(l)})^2 \ . \quad (7)$$

Many algorithms have been developed for calculating OSPA barycenters in case all sets have the same cardinalities $n = m(1) = \ldots = m(n_p)$, see [6], [8], [12], [13], [20]. For example, in [20], an alternating optimization algorithm for sets with equal cardinalities and the Euclidean distance as the base metric ($c = \infty$) has been proposed.

Algorithm 1 Computes the barycenter (9) for vector-valued data $\{\underline{x}^{(i)}\}_{i=1}^{n_p}$ with weights $\{w^{(i)}\}_{i=1}^{n_p}$, see (8), using a base distance (2) with threshold c.

Choose an initial estimate $y^{(0)} \in \mathbb{R}^d$
$t := 0$
repeat
  Determine all points with distance less than c to $\underline{y}^{(t)}$

$$\mathcal{I}^{(t+1)} = \{\ i \in \{1, \dots, n_p\} \mid ||\underline{y}^{(t)} - \underline{x}_i|| \leq c\} \quad (10)$$

  Update barycenter

$$\underline{y}^{(t+1)} := \frac{1}{\sum_{i \in \mathcal{I}^{(t+1)}} w_i} \sum_{i \in \mathcal{I}^{(t+1)}} w_i \cdot \underline{x}_i$$

  $t := t + 1$
until Converged
return $\underline{y}^{(t)}$

B. *Alternating Optimization Algorithm for OSPA Barycenters*

Calculating an OSPA barycenter in its most general form as in (7) is extremely challenging. First, one has to optimize over all possible cardinalities. Second, a thresholded base distance – which is required for sets with different cardinalities – renders the problem much more difficult as one cannot resort to the variational property of the "mean".

In the following, we will derive a novel (provably correct) alternating optimization algorithm for solving (7), i.e., the algorithm is capable to deal with different cardinalities and with the thresholded base distance.

1) Barycenter for Vectors Using a Thresholded Ground Distance: For didactic purposes, we first consider the sub-problem of finding the barycenter of a vector-valued data set with respect to the thresholded base distance (2). Hence, given are n_p vectors

$$\{\underline{x}_i\}_{i=1}^{n_p} \subset \mathbb{R}^d \quad (8)$$

and corresponding weights $\{w_i\}_{i=1}^{n_p} \subset \mathbb{R}$. Then, the problem is to find a vector $\underline{\hat{y}} \in \mathbb{R}^d$ so that

$$\underline{\hat{y}} = \arg\min_{\underline{y}} \sum_{i=1}^{n_p} w_i d^c(\underline{y}, \underline{x}_i)^2 \ . \quad (9)$$

For $c = \infty$, i.e., the Euclidean distance, the solution is given by the mean $\sum_{i=1}^{n_p} w_i \underline{x}_i$. The case $c < \infty$ is challenging as the thresholded distance is not differentiable, e.g., it is not obvious how to derive a Newton method.

We propose an iterative optimization algorithm, see Algorithm 1. The key idea is as follows: Based on an initial barycenter estimate $\underline{y}^{(0)} \in \mathbb{R}^d$, an "improved" barycenter is given by the mean of all $\underline{x}_i$, $i \in \{1, \dots, n_p\}$, for which $||\underline{y}^{(0)} - \underline{x}_i|| < c$ holds.

It can be shown that each iteration of Algorithm 1 improves the quality of the current barycenter estimate.

Proof of Algorithm 1: First, we define cost function

$$\text{Cost}^c(y, \mathcal{I}) := \sum_{i \in \mathcal{I}} w_i ||\underline{y} - \underline{x}_i||^2 + c \cdot \sum_{i \in \{1, \dots, n_p\} \setminus \mathcal{I}} w_i \ . \quad (11)$$

Then, we have to show that

$$\text{Cost}^c(\underline{y}^{j+1}, \mathcal{I}^{(t+1)}) \leq \text{Cost}^c(\underline{y}^{j}, \mathcal{I}^{(t)}) \ , \quad (12)$$

which is achieved in the following steps:

1) $\text{Cost}^c(\underline{y}^{(t+1)}, \mathcal{I}^{(t)}) \leq \text{Cost}^c(\underline{y}^{j}, \mathcal{I}^{(t)})$ due to the properties of the mean.
2) Let $\mathcal{I}^{(t+1)} = \mathcal{I}^{(t)} \setminus \mathcal{D} \cup \mathcal{A}$, where $\mathcal{A}$ denotes the particles that are moved "below" the threshold and $\mathcal{D}$ are the particles that are moved "above" the threshold c.
3) $\text{Cost}^c(\underline{y}^{(t+1)}, \mathcal{I}^{(t)} \setminus \mathcal{D}) \leq \text{Cost}^c(\underline{y}^{(t+1)}, \mathcal{I}^{(t)})$ as all points in $\mathcal{D}$ have a distance **more** than c from $\underline{y}^{(t+1)}$.
4) $\text{Cost}^c(\underline{y}^{(t+1)}, \mathcal{I}^{(t)} \setminus \mathcal{D} \cup A) \leq \text{Cost}^c(\underline{y}^{(t+1)}, \mathcal{I}^{(t)} \setminus \mathcal{D})$ as all points in $\mathcal{A}$ have a distance **less** than c from $\underline{y}^{(t+1)}$.

$\Rightarrow$ Equation (12) holds.

■

We note that Algorithm 1 resembles an instance of the mean shift principle [11]. However, a direct derivation based on the mean shift principle is *not* possible as the thresholded distance is not differentiable. Also, (9) can be interpreted as an M-estimator from robust statistics [23], where the traditional squared error loss is replaced by a loss function that forgives outliers (in our case the thresholded squared error loss).

2) General Case: Equipped with the previously introduced algorithm for dealing with the thresholded distance, we are now in the position to develop the alternating optimization algorithm for the general OSPA barycenter of sets with varying cardinalities (7). The algorithm shares the widely-used idea of alternating solving subproblems, see for example [16], [19], [20], [35].

For all possible cardinalities n of Y, we start with an initial barycenter estimate and repeat the following two steps until convergence is reached:

1) Calculate all optimal permutations in (7) with respect to the current barycenter.
2) Determine an improved estimate by minimizing (7) for the case that the permutations are given.

For the first step, n_p assignment problems have to be solved as the OSPA distance has to be calculated n_p times, see Remark 2. The second step requires to find the barycenter for *vectors* with respect to the thresholded distance such as (9). Hence, for the second step, we can transfer the idea from Algorithm 1.

Algorithm 2 shows the pseudo-code of overall barycenter algorithm. Note that we integrated only one iteration from Algorithm 1 in order to avoid nested optimizations. The correctness of Algorithm 2 follows from the correctness of Algorithm 1.

III. Clustering Set-Valued Data

This section considers the problem of clustering set-valued data. More precisely, given a data set consisting of n_p observations

$$\{X_i\}_{i=1}^{n_p} \ , \quad (14)$$

Algorithm 2 Calculates the OSPA barycenter (7) of a set of point sets $\{X_i\}_{i=1}^{n_p}$ with weights $\{w_i\}_{i=1}^{n_p}$.

for $n = 1, \ldots, n_{\max}$ **do**
  Choose an initial estimate $Y_n^{(0)} := \{\underline{y}_1^{(0)}, \ldots, \underline{y}_n^{(0)}\}$
  $t := 0$.
  repeat
    Calculate optimal permutations w.r.t $Y_n^{(t)}$

$$\pi_i := \underset{\pi \in \Pi_{\max\{n,m(i)\}}}{\arg\min} \sum_{l=1}^{\max\{n,m(i)\}} d^c(\underline{y}_l^{(t)}, \underline{x}_{\pi(l),i})^2 \ .$$

    Caclulate improved barycenter

$$Y_n^{(t+1)} = \{\underline{y}_1^{(t+1)}, \ldots, \underline{y}_n^{(t+1)}\}$$

    with

$$\underline{y}_l^{(t+1)} := \frac{1}{\sum_{i \in \mathcal{I}} \tilde{w}_i} \cdot \sum_{i \in \mathcal{I}_l^{(t)}} \tilde{w}_i \cdot \underline{x}_{\pi_i(l),i} \ ,$$

    for all $l \in \{1, \ldots, m(i)\}$, where

$$\mathcal{I}_l^{(t)} := \{ \ i \in \{1, \ldots, n_p\} \mid ||\underline{y}_l^{(t)} - \underline{x}_{\pi_i(l),i}|| \leq c \text{ and } \underline{x}_{\pi_i(l),i} < \underline{\infty}\}$$

    and

$$\tilde{w}_i = w_i / \max\{n, m(i)\} \ .$$

    $t := t + 1$
  until Converged
  Compute cost

$$\mathcal{C}(n) := \text{MOSPA}(Y_n^{(t)}, \{X_i\}_{i=1}^{n_p}, \{w_i\}_{i=1}^{n_p})$$

  Save barycenter

$$\hat{Y}_n := Y_n^{(t)}$$

end for
Find cardinality with minimum cost

$$\hat{n} := \arg\min_n \mathcal{C}(n) \qquad (13)$$

return Barycenter $\hat{Y}_{\hat{n}}$ with cardinality $\hat{n}$

Algorithm 3 Set-k-means: Performs a clustering (16) of the set-valued data $\{X_i\}_{i=1}^{n_p}$ with respect to the OSPA distance.

Choose initial clusters $\{\hat{Y}_j^{(0)}\}_{j=1}^k$
$t := 0$
repeat
  Cluster assignment step:
  For all $j \in \{1, \ldots, k\}$

$$\mathcal{C}_j^{(t+1)} := \{i \in \{1, \ldots, n_p\} \mid \text{OSPA}_c(X_i, \hat{Y}_j)^2 \leq \text{OSPA}_c(X_i, \hat{Y}_r)^2 \quad \text{for all } r = 1 \ldots k\} \ . \qquad (18)$$

  Cluster centroid update step:
  For all $j \in \{1, \ldots, k\}$

$$\hat{Y}_j^{(t+1)} := \arg\min_Y \frac{1}{|\mathcal{C}_j^{(t+1)}|} \sum_{i \in \mathcal{C}_j^{(t+1)}} \text{OSPA}_c(X_i, Y)^2 \ . \qquad (19)$$

  $t := t + 1$
until Converged
return $\{\hat{Y}_l^{(t)}\}_{j=1}^k$

where each observation is a finite point set. The i-th point set is given by

$$X_i = \{\underline{x}_{1,i}, \ldots, \underline{x}_{m(i),i}\} \subset \mathbb{R}^d \qquad (15)$$

in which $m(i)$ is the number of points in the set.

The objective is to partition the data set into k clusters $\mathbf{C} = \{\mathcal{C}_j\}_{j=1}^k$, where $\mathcal{C}_j \subset \{1, \ldots, n_p\}$. Each cluster is associated with a cluster centroid. For the vector-valued k-means, the cluster centroid is the mean of all cluster members. For the point cloud case, we propose to employ the OSPA barycenter (7) of the cluster members as the cluster centroid, i.e., each cluster $\mathcal{C}_j$ is associated to the cluster centroid

$$\hat{Y}_j = \arg\min_Y \frac{1}{|\mathcal{C}_j|} \sum_{i \in \mathcal{C}_j} \text{OSPA}_c(X_i, Y)^2 \ . \qquad (16)$$

According to the k-means approach, the quality of the clusters is assessed with the help of the Within-Cluster Sum of Squares (WCSS) [9] so that we aim at finding a clustering $\hat{\mathbf{C}}$ with

$$\hat{\mathbf{C}} = \arg\min_{\mathcal{C}} \sum_{j=1}^k \sum_{i \in \mathcal{C}_j} \text{OSPA}_c(\hat{Y}_j, X_i)^2 \ . \qquad (17)$$

Hence, the objective is to find clusters that minimize the sum of squared OSPA distances of the observations to the cluster centroid.

In order to minimize (17), we propose the following "Set-k-means" algorithm that starts with a (random) initial guess of the cluster centers $\{\hat{Y}_j^{(0)}\}_{j=1}^k$ and iteratively calculates improved clusters. The algorithm results from the standard k-means algorithm from systematic replacement the squared error with the OSPA distance, where instead of the arithmetic mean the OSPA barycenter is used.

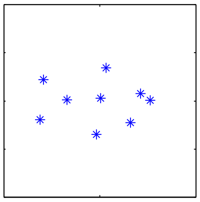
1) *Cluster assignment step:* Recalculates the cluster assignments by assigning each observation to its nearest cluster center with respect to the OSPA distance.
2) *Cluster set-centroid update step:* Based on the updated clusters, new cluster centroids are calculated using the OSPA barycenter.

As both steps decrease the set version of the WCSS (17), the Set-k-means algorithm eventually converges. In general, it is not necessary that (19) finds the optimal barycenter; an improved estimate is sufficient.

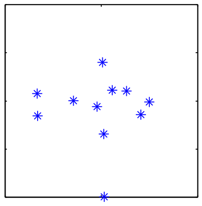
The calculation of (18) in Algorithm 3 requires the repeated evaluation of the OSPA distance, which can be done as described in Remark 2. The barycenter in (19) can be computed efficiently with Algorithm 2.

IV. Evaluation

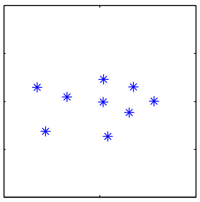
We demonstrate the benefits of the Set-k-means algorithm by means of two simulated experiments.



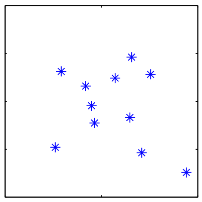
(a) Point set X_1.



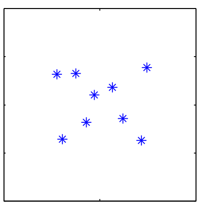
(b) Point set X_2.



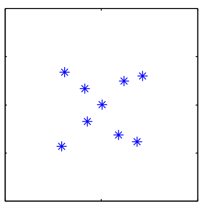
(c) Point set X_3.



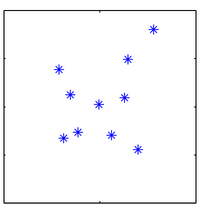
(d) Point set X_4.



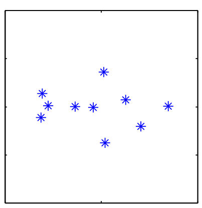
(e) Point set X_5.



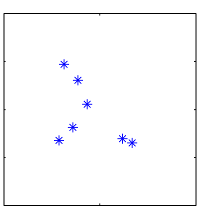
(f) Point set X_6.



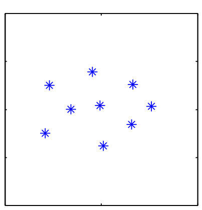
(g) Point set X_7.



(h) Point set X_8.

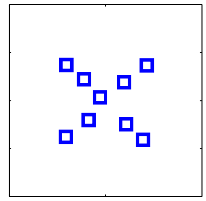


(i) Point set X_9.

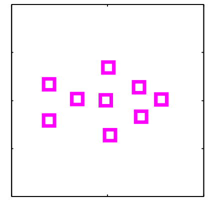


(j) Point set X_{10}.

Fig. 1: Example data set $\{X_i\}_{i=1}^{10}$.



(a) Cluster centroid Y_1.



(b) Cluster centroid Y_2.

Fig. 2: Clustering results.

A. *Experiment 1*

The first experiment is performed with an eye to *unsupervised* target recognition and classification. Fig. 1 depicts $n_p = 10$ sets (4). Each set X_i, $i = 1 \dots, 10$, represents a radar scan of an extended target, where each point in X_i is a single reflection, see [4], [5]. Note that the sets have different cardinalities.

Assume we know that there are two different types of extended targets. Then, the following two questions are of high interest: How does the typical target of type 1 and type 2 look, and which set X_i belongs to which type? Actually, this is a clustering problem where the number of clusters is known ($k = 2$) and the data to be clustered is set-valued.

Fig. 2 depicts the clustering results of Algorithm 2 when applied to the data set in Fig. 1. We used the OSPA distance with threshold $c = 0.5$ (the size of the surveillance area is 2×2). As the data set is quite small and there are no runtime issues, we ran Algorithm 2 n_p times, using all observations X_i as an initialization.

It can clearly be seen that the calculated cluster centroid for cluster 1 is a star-shaped point pattern formed by 9 points, and the cluster centroid of cluster 2 is an arrow-shaped point pattern, also with 9 points. Note that the number of points per cluster centroid was not known in advance. It is determined as part of the clustering algorithm.

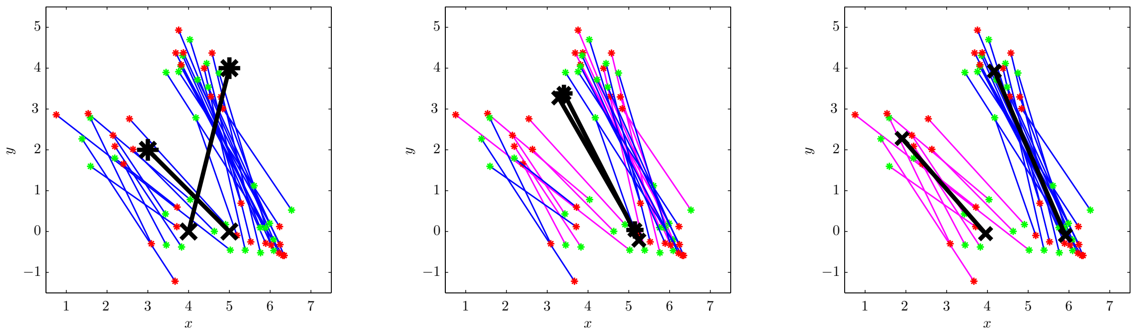
Remark 4. The data set was generated by artificially perturbing the ground truth with Gaussian noise and adding uniformly distributed "false points". However, as Set-k-means is *not* model-based, all the details of how the point sets are generated are not relevant. The only free parameter to adjust is the threshold c in (2).

B. *Experiment 2*

The purpose of the second experiment is to show how Set-k-means can be used to detect structure in a data set that remains hidden with the standard vector k-means. Suppose each observation in the data set consists of two two-dimensional features. The feature values are generated as follows: With equal probability draw the features either according to

- $\underline{x}_{1,i} \sim \mathcal{N}(\left[2,2\right]^T, \frac{1}{2}\,\mathbf{I}_2)$ and $\underline{x}_{2,i} \sim \mathcal{N}(\left[4,4\right]^T, \frac{1}{2}\,\mathbf{I}_2)$

  or
- $\underline{x}_{1,i} \sim \mathcal{N}(\left[4,2\right]^T, \frac{1}{2}\,\mathbf{I}_2)$ and $\underline{x}_{2,i} \sim \mathcal{N}(\left[6,0\right]^T, \frac{1}{2}\,\mathbf{I}_2)$,



(a) Illustration of the data set and initial cluster centroids.

(b) Clustering results of the standard k-means algorithm.

(c) Clustering results of the Set-k-means algorithm.

Fig. 3: Experiment 2: Each observation in the data consists of two two-dimensional features $\underline{x}_{1,i}$ (green markers) and $\underline{x}_{2,i}$ (red markers), where $i \in \{1, \ldots 30\}$ is the index. The lines connect the first feature $\underline{x}_{1,i}$ with the second feature $\underline{x}_{2,i}$. In (a), the black markers indicate the two initial cluster centers (asterisk and cross indicate features 1 and 2, respectively). In (b) and (c), the black markers show the cluster centroids and the line color for the observations indicates the cluster.

where $\mathbf{I}_2 = \text{diag}([1, 1])$ is the two-dimensional identity matrix. As this is a Gaussian mixture density with two components, we expect two clusters with cluster centroids at means of the Gaussians.

However, the data is now corrupted by an additional noise source: We introduce an association uncertainty by randomly switching the values of $\underline{x}_{1,i}$ and $\underline{x}_{2,i}$. As a consequence, the ordering of the features in the observations gets lost. The resulting data set consisting of 30 observations is visualized in Fig. 3a.

First, we apply the standard k-means algorithm to the data (with $k = 2$). For this purpose, we stack the two features in one single vector $\left[\underline{x}_{1,i}^T, \underline{x}_{2,i}^T\right]^T$. Fig. 3b shows the clustering results. Due to the random switching of the feature values, k-means is not able to find the underlying structure of the data. Next, we treat the observations as sets, i.e., $X_i = \{\underline{x}_{1,i}, \underline{x}_{2,i}\}$ and apply the Set-k-means Algorithm 3 this work. As all sets have the same cardinalities, we set the thresholding parameter of the OSPA distance to $c = \infty$, i.e., we use the Euclidean distance. In this case, the OSPA distance coincides with the Wasserstein distance. As depicted in Fig. 3c, the Set-k-means algorithm is capable of perfectly identifying the underlying structure of the data. The reason is that it does not make any assumption on the ordering of the features and optimizes the OSPA/Wasserstein distance.

V. Conclusions

Set-valued data occurs nearly everywhere – in computer vision, machine learning, signal processing, data science, and many more. A fundamental problem is to identify the structure of set-valued data, i.e., to summarize similar sets in a cluster.

In this work, we have derived an algorithm for clustering set-valued data with respect to a true distance metric on sets – the OSPA distance. The key technique is a novel algorithm for efficiently calculating the OSPA barycenter of sets with varying cardinalities. The final algorithm is easy to implement and comes with a low computational complexity. This approach is not model-based, i.e., no assumption on the generation of the data is exploited. The only free parameter to adjust is the threshold for the base distance in the OSPA distance. We have demonstrated two application scenarios for the Set-k-means algorithm. (i) Set-k-means can be used in automatic *target recognition* for identifying target types. (ii) In *unsupervised learning* problems, Set-k-means might be able to discover structures that are not visible with standard methods.

In future work, we will investigate initialization techniques for the Set-k-means algorithm. Also, we will consider further applications of both the OSPA barycenter algorithm and Set-k-means. For example, the OSPA barycenter algorithm can be used in multi-target trackers estimate the number of targets systematically, such as in [2]. Specifically, we think that our approach is especially useful for applications involving sparse point clouds, where the extraction of features is not possible.

Acknowledgment

This research was partially supported by ONR under contract N000014-13-1-0231.

References

[1] M. Agueh and G. Carlier, "Barycenters in the Wasserstein Space," *SIAM Journal on Mathematical Analysis*, vol. 43, no. 2, pp. 904–924, 2011.

[2] B. Balasingam, M. Baum, and P. Willett, "MMOSPA Estimation with Unknown Number of Objects," in *IEEE China Summit and International Conference on Signal and Information Processing (IEEE ChinaSIP 2015)*, 2015.

[3] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh, "Clustering with Bregman Divergences," *J. Mach. Learn. Res.*, vol. 6, pp. 1705–1749, Dec. 2005.

[4] M. Baum and U. D. Hanebeck, "Shape Tracking of Extended Objects and Group Targets with Star-Convex RHMs," in *Proceedings of the 14th International Conference on Information Fusion (Fusion 2011)*, Chicago, Illinois, USA, Jul. 2011.

[5] ——, "Extended Object Tracking with Random Hypersurface Models," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, pp. 149–159, Jan. 2014.

[6] M. Baum, P. Willett, and U. D. Hanebeck, "Calculating Some Exact MMOSPA Estimates for Particle Distributions," in *Proceedings of the 15th International Conference on Information Fusion (Fusion 2012)*, Singapore, Jul. 2012.

[7] ——, "On Wasserstein Barycenters and MMOSPA Estimation," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1511–1515, Oct. 2015.

[8] ——, "Polynomial-Time Algorithms for the Exact MMOSPA Estimate of a Multi-Object Probability Density Represented by Particles," *IEEE Transactions on Signal Processing*, vol. 63, no. 10, pp. 2476–2484, May 2015.

[9] P. Berkhin, "A Survey of Clustering Data Mining Techniques," in *Grouping Multidimensional Data*, J. Kogan, C. Nicholas, and M. Teboulle, Eds. Springer Berlin Heidelberg, 2006, pp. 25–71.

[10] R. Bhattacharya and V. Patrangenaru, "Large Sample Theory of Intrinsic and Extrinsic Sample Means on Manifolds," *The Annals of Statistics*, vol. 31, no. 1, pp. 1–29, Feb. 2003.

[11] D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach Toward Feature Space Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, May 2002.

[12] D. F. Crouse, P. Willett, Y. Bar-Shalom, and L. Svensson, "Aspects of MMOSPA Estimation," in *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference, Orlando, FL*, Dec. 2011.

[13] D. F. Crouse, "Advances in Displaying Uncertain Estimates of Multiple Targets," in *SPIE – Signal Processing, Sensor Fusion, and Target Recognition XXII*, vol. 8745, 2013, pp. 874 504–874 504–31.

[14] M. Cuturi and A. Doucet, "Fast Computation of Wasserstein Barycenters," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, Oct. 2014.

[15] I. S. Dhillon and D. S. Modha, "Concept Decompositions for Large Sparse Text Data Using Clustering," *Mach. Learn.*, vol. 42, no. 1-2, pp. 143–175, Jan. 2001. [Online]. Available: http://dx.doi.org/10.1023/A:1007612920971

[16] H. Ding and J. Xu, "Finding Median Point-Set Using Earth Movers Distance," in *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI 2014)*, 2014.

[17] F. Endres, C. Plagemann, C. Stachniss, and W. Burgard, "Unsupervised Discovery of Object Classes from Range Data using Latent Dirichlet Allocation," in *Robotics: Science and Systems*, vol. 2. Seattle, Washington, 2009, p. 113120.

[18] S. Gold, A. Rangarajan, C.-P. Lu, S. Pappu, and E. Mjolsness, "New Algorithms for 2D and 3D Point Matching: Pose Estimation and Correspondence," *Pattern Recognition*, vol. 31, no. 8, pp. 1019 – 1031, 1998.

[19] S. Gold, A. Rangarajan, and E. Mjolsness, "Learning with Preknowledge: Clustering with Point and Graph Matching Distance Measures," *Neural Computation*, vol. 8, no. 4, pp. 787–804, 1996.

[20] M. Guerriero, L. Svensson, D. Svensson, and P. Willett, "Shooting Two Birds with Two Bullets: How to Find Minimum Mean OSPA Estimates," *Proceedings of the 13th International Conference on Information Fusion (Fusion 2010)*, 2010.

[21] D. Guru, B. B. Kiranagi, and P. Nagabhushan, "Multivalued Type Proximity Measure and Concept of Mutual Similarity Value Useful for Clustering Symbolic Patterns," *Pattern Recognition Letters*, vol. 25, no. 10, pp. 1203 – 1213, 2004.

[22] J. Hoffman and R. Mahler, "Multitarget Miss Distance and its Applications," in *Proceedings of the Fifth International Conference on Information Fusion*, vol. 1, Jul. 2002, pp. 149–155 vol.1.

[23] P. J. Huber, *Robust statistics*. Springer, 2011.

[24] A. Irpino and R. Verde, "Dynamic Clustering of Interval Data using a Wasserstein-based Distance," *Pattern Recognition Letters*, vol. 29, no. 11, pp. 1648 – 1658, 2008.

[25] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data Clustering: A Review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, Sep. 1999.

[26] A. K. Jain, "Data Clustering: 50 Years beyond K-Means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651 – 666, 2010.

[27] S. Justin, R. Rustamov, L. Guibas, and A. Butscher, "Earth Mover's Distances on Discrete Surfaces," in *ACM Transactions on Graphics (SIGGRAPH)*, 2014.

[28] S. Kumar, S. Sural, A. Watve, and S. Pramanik, "CNODE: Clustering of Set-Valued Non-Ordered Discrete Data," *IJDMMM*, vol. 1, no. 3, pp. 310–334, 2009.

[29] R. Mahler, ""Statistics 101" For Multisensor, Multitarget Data Fusion," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 53 –64, Jan. 2004.

[30] O. Pele and M. Werman, "Fast and Robust Earth Mover's Distances," in *IEEE 12th International Conference on Computer Vision (ICCV)*, 2009.

[31] D. Phung and B. Vo, "A Random Finite Set Model for Data Clustering," in *Proceedings of International Conference on Fusion (FUSION)*, Salamanca, Spain, Jul. 2014.

[32] J. Rabin, G. Peyre, J. Delon, and M. Bernot, "Wasserstein Barycenter and Its Application to Texture Mixing," in *Scale Space and Variational Methods in Computer Vision*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, vol. 6667, pp. 435–446.

[33] Y. Rubner, C. Tomasi, and L. Guibas, "The Earth Mover's Distance as a Metric for Image Retrieval," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, 2000.

[34] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A Consistent Metric for Performance Evaluation of Multi-Object Filters," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3447 –3457, Aug. 2008.

[35] K. Turner, Y. Mileyko, S. Mukherjee, and J. Harer, "Fréchet Means for Distributions of Persistence Diagrams," *ArXiv e-prints*, Jun. 2012.

[36] C. Villani, *Optimal Transport*. Springer Verlag, 2009.

[37] J. Wagner and B. Ommer, "Efficient Clustering Earth Mover's Distance," in *Computer Vision – ACCV 2010*, ser. Lecture Notes in Computer Science, R. Kimmel, R. Klette, and A. Sugimoto, Eds. Springer Berlin Heidelberg, 2011, vol. 6493, pp. 477–488.

[38] G. Zen and E. Ricci, "Earth Mover's Prototypes: A Convex Learning Approach for Discovering Activity Patterns in Dynamic Scenes," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2011, pp. 3225–3232.