

Cluster Analysis via Projection onto Convex Sets

Le-Anh Tran^a, Daehyun Kwon^b, Henock Mamo Deberneh^c, and Dong-Chul Park^{a,*}

^a Dept. of Electronics Engineering, Myongji University, Gyeonggi, South Korea

^b Dept. of Information Technology Polish Management, Soongsil University, Seoul, South Korea

^c Dept. of Biochemistry and Molecular Biology, University of Texas Medical Branch, Texas, USA

Abstract. This paper proposes a data clustering algorithm that is inspired by the prominent convergence property of the *Projection onto Convex Sets* (POCS) method, termed the *POCS-based clustering algorithm*. For disjoint convex sets, the form of simultaneous projections of the POCS method can result in a minimum mean square error solution. Relying on this important property, the proposed POCS-based clustering algorithm treats each data point as a convex set and simultaneously projects the cluster prototypes onto respective member data points, the projections are convexly combined via adaptive weight values in order to minimize a predefined objective function for data clustering purposes. The performance of the proposed POCS-based clustering algorithm has been verified through a large scale of experiments and data sets. The experimental results have shown that the proposed POCS-based algorithm is competitive in terms of both effectiveness and efficiency against some of the prevailing clustering approaches such as the K-Means/K-Means++ and Fuzzy C-Means (FCM) algorithms. Based on extensive comparisons and analyses, we can confirm the validity of the proposed POCS-based clustering algorithm for practical purposes. The code of this work is publicly available at <https://github.com/tranleanh/pocs-based-clustering>.

Keywords: POCS, convex sets, clustering algorithm, unsupervised learning, machine learning

1. Introduction

Cluster analysis or clustering is a kind of unsupervised learning task that aims to categorize similar data points while separating them from dissimilar ones without any prior knowledge. There exist numerous types of clustering approaches, each method is best suited to a particular distribution of data, thus it is very difficult to introduce a thorough list of all the clustering algorithms because of the variation of information, the research fields, and the rapid development of modern technology [1]. Though there exists a great number of approaches to clustering, there are three most popularly used categories including *partitional*, *density-based*, and *hierarchical* methods [2]. Among them, the simplest form of clustering is partitional clustering [3–8] which considers the clustering error (sum of the distances from all cluster centers to their corresponding data members) as the main criterion to be optimized.

One of the most popular partitional methods is the K-Means algorithm [9] which has been widely applied to numerous general applications due to its simplicity and efficiency [10]. The K-Means clustering algorithm alternates between assigning cluster membership for each data point to the nearest cluster center and updating the center of each cluster. The objective of the algorithm is to find a set of prototypes that minimize the clustering error and the algorithm converges when there is no further change in the

*Corresponding author: Dong-Chul Park, Dept. of Electronics Engineering, Myongji University, Gyeonggi, South Korea.
E-mail: parkd@mju.ac.kr.

assignment of instances to clusters [9]. The convergence quality of the K-Means clustering algorithm heavily depends on the initial prototypes. Besides, this algorithm is not guaranteed to find the optimum and is known to be sensitive to noise and outliers [11]. In order to overcome the drawbacks of the *naïve K-Means*, Arthur and Vassilvitskii introduced the K-Means++ algorithm [12] which can improve both the speed and accuracy of the naïve version by enhancing the quality of prototype initializing procedure via a careful seeding method.

Another well-known clustering method in the partitional category is the Fuzzy C-Means (FCM) algorithm. In the FCM algorithm, a data point can belong to multiple subgroups, and the certainty degree for that data point belonging to a certain cluster is represented by a membership function. The performance of the FCM algorithm is also highly dependent on the initialization of prototypes and the initial membership value [13]. In addition, the drawbacks of the FCM algorithm include extended computational time and incapability in handling noisy data and outliers [14]. In order to overcome the FCM algorithm's shortcomings, the Gradient-based Fuzzy C-Means (GBFCM) algorithm was introduced by Park and Dagher [15] in which the minimization process of the objective function is proceeded by solving two equations alternatively in an iterative fashion.

On the other extreme, *Projection onto Convex Sets* (POCS) is a powerful method of finding the common point of convex sets in several convex programming problems which was introduced in the mid-1960s [16, 17]. The main goal of the POCS approach is to find a vector that resides in the intersection of convex sets. It has been shown that successive projections between two or more convex sets with non-empty intersections converge to a point that exists in the intersection of the convex sets [16]. When closed convex sets are disjoint, the sequential projections do not converge to a single point, instead, they converge to greedy limit cycles which are dependent on the order of the projections [18]. POCS has been applied to solve various problems including communication systems [19], super-resolution [20], deconvolution and extrapolation [21], embedding analysis [22], and tomography [23]. POCS can also be applied to point-matching problems to determine the correspondence between two sets of points extracted from different images. Lian *et al.* [24] proposed a clustering successive POCS (SPOCS) algorithm for a fast point-matching problem. The algorithm utilizes SPOCS for enforcing a two-way constraint. The concept of clustering is used to decrease the computational load of the SPOCS technique. The algorithm can reduce computational complexity with an insignificant decline in precision.

In this paper, we present a novel clustering algorithm using the convergence property of POCS. The proposed POCS-based clustering algorithm considers each data point as a convex set and all the data points in a data set as disjoint convex sets, then performs projections from the cluster prototypes onto each of their respective constituent instances in order to update the membership of data and compute a new set of prototypes. At the beginning, a careful seeding method is applied to initialize cluster prototypes, and each data point is then assigned to its nearest cluster. The process of simultaneously projecting the cluster prototypes onto corresponding member data points is repeated until the convergence criterion is satisfied and the final clusters are found. Part of this research was presented in [25] where the basic idea and the preliminary experiments were introduced.

The remainder of this paper is structured as follows. In Section 2, a brief review of the POCS concepts and notations is presented. The POCS-based clustering algorithm is proposed in Section 3. Section 4 provides extensive experiments and assessments on a wide range of applications and datasets in order to verify the validity of the proposed POCS-based clustering algorithm. Section 5 concludes this paper.

2. Preliminaries

The theory of convex set and function has a rich history and has been a focus of research as it has been one of the most powerful tools in the theory of optimization [26–28]. This work proposes to use the concepts of convex set and POCS theory for data analysis and clustering problems.

2.1. Convex Set

A convex set is a collection of vectors (data) having the following property: given a non-empty set C , which is a subset of the Hilbert space H ($C \subseteq H$), for all $\vec{x}_1, \vec{x}_2 \in C$, the set C is called a convex set if for every real value λ with $\lambda \in [0, 1]$ the following condition holds true:

$$\vec{x} := \lambda \vec{x}_1 + (1 - \lambda) \vec{x}_2 \in C, \quad (1)$$

and thus $\vec{x} = \vec{x}_1$ when $\lambda = 1$ and $\vec{x} = \vec{x}_2$ if $\lambda = 0$. In this sense, a line segment connecting any two points \vec{x}_1 and \vec{x}_2 is completely subsumed in the set C [29].

2.2. Projection onto Convex Sets (POCS)

The concept of projection of a point onto a plane deals with minimization problems, which is to find a point on the plane that is closest to the center of the projection. Given a vector \vec{z} with $\vec{z} \notin C$, the projection of \vec{z} onto C , denoted as P_C , is the unique point \vec{x}_0 ($\vec{x}_0 \in C$):

$$\vec{x}_0 = P_C(\vec{z}), \quad (2)$$

such that the distance between \vec{z} and \vec{x}_0 is minimum:

$$\vec{x}_0 = \arg \min_{\vec{x} \in C} \|\vec{z} - \vec{x}\|. \quad (3)$$

If $\vec{z} \in C$, then the projection of \vec{z} onto C is itself ($\vec{z} = P_C(\vec{z})$). The constrained optimization task is expressed as Eq. (3), where \vec{x} denotes all the points belonging to the set C .

2.3. Alternating POCS

Alternating projections between two (or more) convex sets with non-empty intersections converge to a point that falls within the intersection of the convex sets [16]. This prominent property of alternating POCS can be applied to solve many tasks which can be described under the convex optimization problems. Given n closed and convex sets with non-empty intersections, the alternating (or successive) projections on the sets can be expressed as:

$$\vec{x}[q+1] = P_{C_n}(\dots(P_{C_2}(P_{C_1}(\vec{x}[q])))\dots), \quad (4)$$

where q denotes the iteration index. Starting from any initial point, the solution to the task resides in the intersection of the convex sets, represented by the following equation:

$$\vec{x}[\infty] \in \bigcap_{i=1}^n C_i = C_1 \cap C_2 \cap \dots \cap C_n. \quad (5)$$

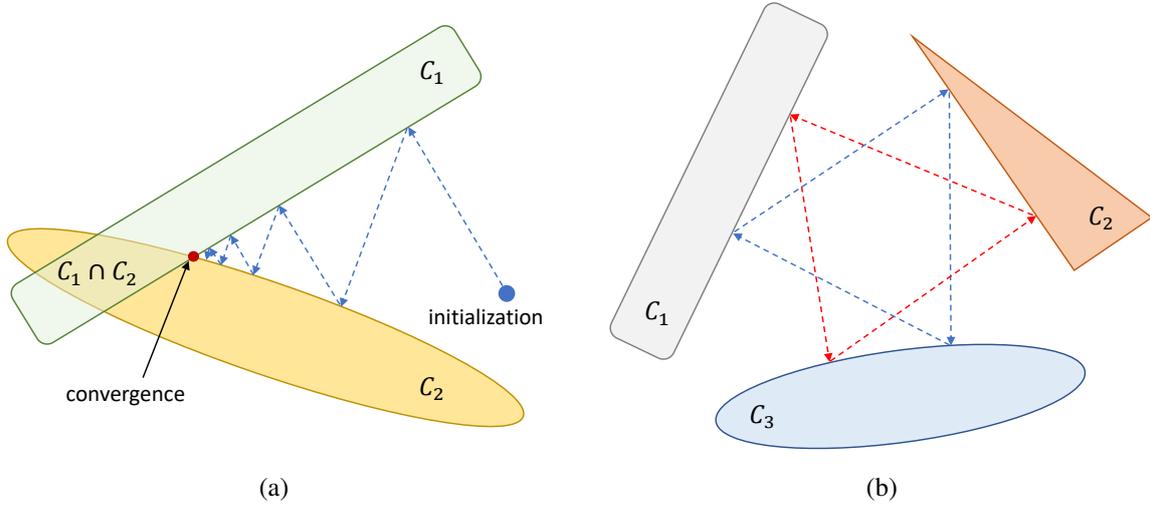


Fig. 1. Alternating POCS: (a) The successive projections converge to a point that belongs to the intersection of the sets for intersecting convex sets, while (b) they converge to greedy limit cycles that depend on the order of projections for disjoint convex sets.

Note that the intersection is also convex. A graphical illustration of the alternating projections onto intersecting convex sets is given in Figure 1a.

On the other hand, when the convex sets are disjoint, the sequential projections do not converge to a single point. Instead, they converge to greedy limit cycles which are dependent on the order of the projections. Especially, in the case of only two non-intersecting convex sets, the projections converge to a greedy limit cycle between two points (one for each set) that are closest in the distance sense [30]. Figure 1b depicts an illustration of alternating projections on three disjoint convex sets.

2.4. Parallel POCS

The parallel mode of POCS sometimes is referred to as *simultaneous weighted projections* [31]. Given an initial point and n convex sets with non-empty intersections, the point is simultaneously projected onto all convex sets, and each projection has a weight of importance w_i such that:

$$\sum_{i=1}^n w_i = 1. \quad (6)$$

All the projections in each iteration are convexly combined to find a minimum mean square error solution which can be applied to solve many minimization problems. The updated data point after one iteration of parallel projections can be expressed as:

$$\bar{x}[q+1] = \sum_{i=1}^n w_i P_{C_i}(\bar{x}[q]), \quad (7)$$

where q denotes the iteration index. The main advantages of the parallel mode of POCS when compared with the alternating version include computational efficiency and improved execution time. The geometrical representation of the parallel POCS method is given in Figure 2a.

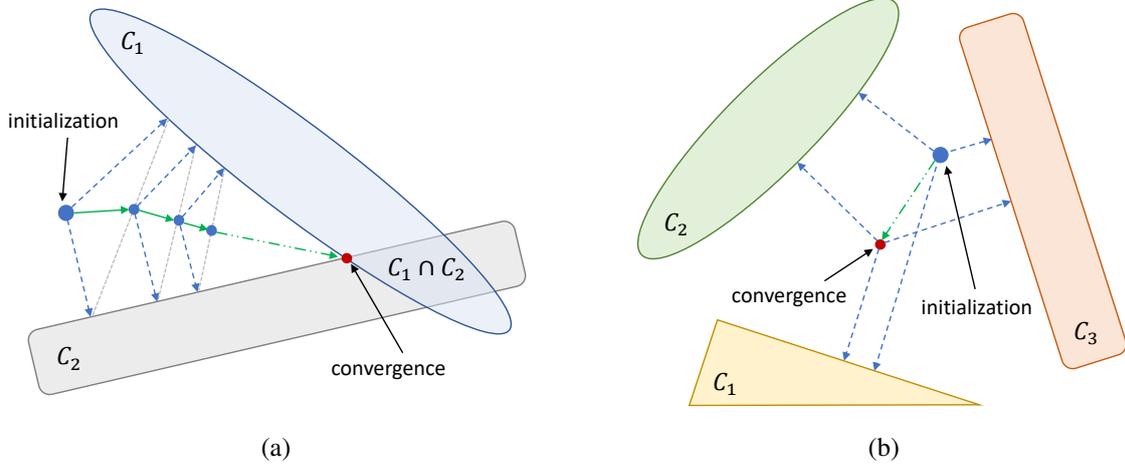


Fig. 2. Parallel POCS: (a) The simultaneous projections converge to a point that belongs to the intersection of the sets for intersecting convex sets, whereas (b) they converge to a minimum mean square error solution for disjoint convex sets.

If the sets are disjoint convex sets, the parallel projections after some iterations will converge to a minimum mean square error solution which is written as:

$$\vec{x}[\infty] = \arg \min_{\vec{x}} \sum_{i=1}^n w_i \|\vec{x} - P_{C_i}(\vec{x})\|, \tag{8}$$

with a constraint as in Eq. (6). A graphical interpretation of the convergence process of the parallel POCS method is illustrated in Figure 2b.

3. Methodology

As discussed in Section 2, the iterative projections (for both alternating and parallel forms) of a random data point onto closed and intersecting convex sets converge to a point that resides on the intersection of the sets. For disjoint convex sets, the alternating POCS converges to greedy limit cycles which are dependent on the order of projections. Alternatively, the parallel mode of projection converges to a point that minimizes the weighted sum of the distances from the point to the sets. The parallel form of POCS performs the projections simultaneously, this property can improve the computational efficiency when compared with that of the alternating version. Relying on the prominent properties of the parallel form of POCS, in this section, we devise the POCS-based clustering algorithm.

Following the definition of the convex set presented in Section 2.1, an empty set \emptyset , a singleton set $\{x_0\}$, line segments, hyperplanes, and Euclidian balls are considered convex sets. In this sense, for clustering tasks, a data point (or a vector) is a singleton set with only one element denoted by $\{\vec{x}_0\}$ and can be considered a convex set, and the projection of a cluster prototype onto a data point is that data point. In order to prove that, a mathematical proof is given as follows. Let V represent a vector space over \mathbb{R} , and $\vec{x}_0 \in V$, then the singleton set $S = \{\vec{x}_0\}$ is a convex set.

Algorithm 1 POCS-based Clustering Algorithm**Input:** $S = \{\vec{v}_n \mid n = 1, 2, \dots, N\}$ (dataset), K (number of clusters), I (number of iterations)**Output:** $X = \{\vec{x}_j \mid j = 1, 2, \dots, K\}$ (cluster prototypes)**Steps:** $X = \{\vec{x}_j = \emptyset \mid j = 1, 2, \dots, K\}$

$$\vec{c} = \frac{1}{N} \sum_{n=1}^N \vec{v}_n$$

$$\vec{x}_1 \leftarrow \arg \min_{\vec{v}} \|\vec{v} - \vec{c}\|$$

 $N_X \leftarrow 1$ **for** $j = 2, \dots, K$ **do** **for** $i = 1, \dots, (N - N_X)$ **do**

$d_i \leftarrow \|\vec{v}_i - \vec{x}_{nearest}\|^2$

▶ Process the remaining data points

end

$D^2 = \{d_i \mid i = 1, \dots, (N - N_X)\}$

$Prob_{D^2} \leftarrow \text{normalize}(D^2)$

 Sample \vec{v}_x from the remaining data points with probability $Prob_{D^2}$

$\vec{x}_j \leftarrow \vec{v}_x$

$N_X \leftarrow N_X + 1$

endAssign each data point in S to its nearest prototype $q \leftarrow 1$ **while** $q \leq I$ **do** **for** $j = 1, \dots, K$ **do** **for** $i = 1, \dots, N_j$ **do**

$w_{ji} \leftarrow \frac{\|\vec{x}_j - P_{ji}(\vec{x}_j)\|}{\sum_{m=1}^{N_j} \|\vec{x}_j - P_m(\vec{x}_j)\|}$

end

$\vec{x}_j \leftarrow \vec{x}_j + \sum_{i=1}^{N_j} w_{ji} (P_{ji}(\vec{x}_j) - \vec{x}_j)$

end $q \leftarrow q + 1$ **if** $X[q] = X[q - 1]$ **then**

break

▶ Converged!

end

Re-assign each data point to its nearest prototype

end**Proof.** For any $\vec{x}, \vec{y} \in S$, we have $\vec{x} = \vec{y} = \vec{x}_0$. It follows that:

$$\begin{aligned} & \forall \lambda \in [0, 1] : \lambda \vec{x} + (1 - \lambda) \vec{y} \\ & = \lambda \vec{x}_0 + (1 - \lambda) \vec{x}_0 \\ & = \vec{x}_0, \end{aligned}$$

and we have $\vec{x}_0 \in S$, hence, S is a convex set. \square

On the other hand, cluster prototype initialization is a vastly important procedure that directly affects clustering performance. Beginning with arbitrary prototypes which are typically chosen uniformly at random from the data points has been widely applied due to its positive effects on empirical speed and simplicity. However, this prototype initialization method comes at the price of unstable clustering accuracy [9]. In order to handle the issues of the random seeding technique, a careful seeding procedure can be adopted to improve accuracy while assuring execution speed [12]. To this end, we adopt a careful prototype initialization procedure which is described as follows: the first prototype is chosen as the data point which is closest to the center of the whole dataset, then each of the next prototypes is chosen successively among the rest of data points based on a weighted probability, called D^2 weighting [12], that is proportional to the squared distances from the data points to their respective nearest prototypes that have been chosen earlier.

Consequently, we can devise the POCS-based clustering algorithm as follows: Given a set of data points and a parameter K denoting the predefined number of clusters, the proposed POCS-based clustering algorithm first adopts a careful prototype initialization procedure, then the algorithm treats each data point as a convex set and all data points as disjoint convex sets, and applies a simultaneous weighted projection method to find cluster prototypes and their corresponding member data points via minimizing an objective function J which is defined as follows:

$$J = \sum_{j=1}^K \sum_{i=1}^{N_j} w_{ji} \|\vec{x}_j - P_{ji}(\vec{x}_j)\|, \quad (9)$$

and:

$$w_{ji} = \frac{\|\vec{x}_j - P_{ji}(\vec{x}_j)\|}{\sum_{m=1}^{N_j} \|\vec{x}_j - P_m(\vec{x}_j)\|}, \quad (10)$$

where N_j represents the number of data points in the j^{th} cluster, $P(\vec{x}_j)$ is the projection of the prototype \vec{x}_j onto its member points, and w denotes the weight of importance of the projection. Note that with the formulation of weight calculation as in Eq. (10), the constraint:

$$\sum_{i=1}^{N_j} w_{ji} = 1, \quad (11)$$

can be satisfied as described in Eq. (6).

Algorithm 1 shows the pseudocode of the proposed POCS-based clustering algorithm which includes three main steps. In the first step, the algorithm initializes cluster prototypes by applying a careful prototype initializing method. In the second step, each data point is assigned to the nearest prototype. In the last step, for each cluster, the algorithm performs simultaneous projections and computes new cluster prototypes using the following equation:

$$\vec{x}_j[q+1] = \vec{x}_j[q] + \sum_{i=1}^{N_j} w_{ji} (P_{ji}(\vec{x}_j[q]) - \vec{x}_j[q]), \quad (12)$$

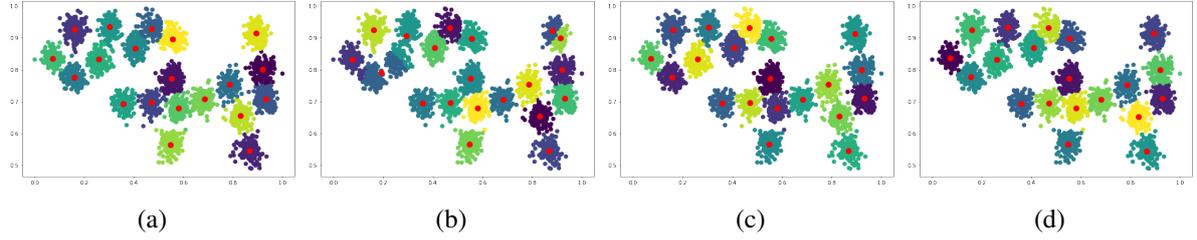


Fig. 3. Visual clustering results of various clustering methods on A1 dataset (red points denote cluster centroids): (a) FCM, (b) K-Means, (c) K-Means++, and (d) POCS-based.

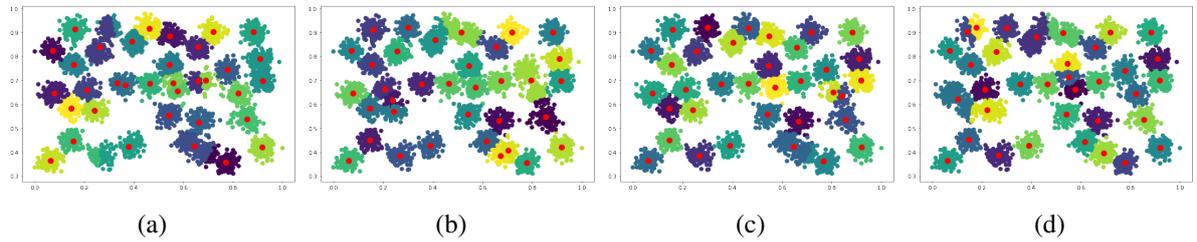


Fig. 4. Visual clustering results of various clustering methods on A2 dataset (red points denote cluster centroids): (a) FCM, (b) K-Means, (c) K-Means++, and (d) POCS-based.

where q denotes the iteration index. The second and the third steps are iteratively repeated until convergence which is when the cluster prototypes no longer change.

The clustering problem we are dealing with is to solve the optimization problem for the objective function given by Eq. (9). That is, starting from a set of initial prototypes:

$$X[0] = \{\vec{x}_j[0] \mid j = 1, 2, \dots, K\}, \quad (13)$$

the algorithm converges to a set of optimized prototypes which minimizes the weighted sum of the distances from the prototypes to their corresponding member points:

$$X[\infty] = \{\vec{x}_j[\infty] \mid j = 1, 2, \dots, K\} = \arg \min_{\{\vec{x}\}} \sum_{j=1}^K \sum_{i=1}^{N_j} w_{ji} \|\vec{x}_j - P_{ji}(\vec{x}_j)\|. \quad (14)$$

4. Experiments and Analyses

In order to provide a comprehensive evaluation, we have compared the performance of the proposed POCS-based algorithm against those of other prevailing methods including the K-Means, K-Means++, and FCM algorithms on four sets of experiments with various types of datasets and applications. The first experiment exploits the synthetic 2-D data sets that are publicly available on the website *Clustering datasets* [32]. The second experiment deals with high-dimensional data sets which were selected from the UCI database repository [33]. The third experiment is to examine the applicability of the proposed POCS-based algorithm to face clustering. The last experiment investigates the effectiveness and efficiency of the proposed algorithm for big data clustering problems. The experimental results were evaluated based on different types of metrics which are discussed in each experiment section. The processor

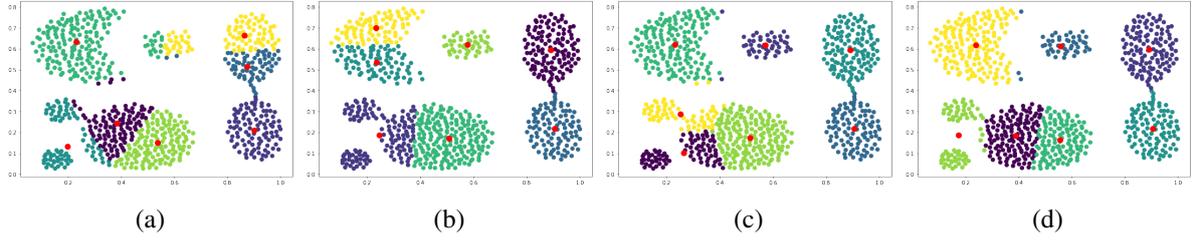


Fig. 5. Visual clustering results of various clustering methods on Aggregation dataset (red points denote cluster centroids): (a) FCM, (b) K-Means, (c) K-Means++, and (d) POCS-based.

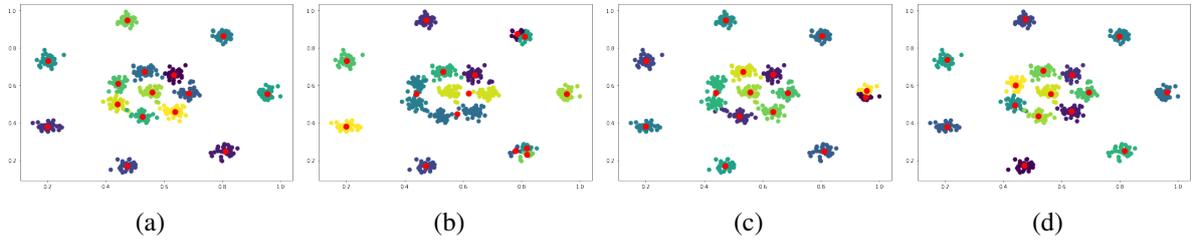


Fig. 6. Visual clustering results of various clustering methods on R15 dataset (red points denote cluster centroids): (a) FCM, (b) K-Means, (c) K-Means++, and (d) POCS-based.

Table 1
Details of the 2-D synthetic datasets.

	A1	A2	Aggregation	R15	S1	S2
#Clusters	20	35	7	15	15	15
#Attributes	2	2	2	2	2	2
#Instances	3,000	5,250	788	600	5,000	5,000

used in all the experiments was Intel(R) Core(TM) i7-4790K CPU @ 4.00GHz on a 64-bit operating system.

4.1. Evaluation on Synthetic 2-D Data Sets

Assessments of clustering results on 2-D space allow observers to easily evaluate visual clustering solutions produced by different clustering algorithms. To this end, experiments to examine the performances of the proposed POCS-based clustering algorithm on six synthetic 2-D data sets including A1, A2, Aggregation, R15, S1, and S2 were conducted [32]. The specifications of these data sets are summarized in Table 1.

Figures 3-8 illustrate typical clustering results produced by different methods in 2-D plots. The found cluster centroids are marked out in red color and located in the vicinity of the center of each data group.

Figure 3 and Figure 4 show the representative visual clustering results of the algorithms under consideration on the A1 and A2 datasets. Both A1 and A2 datasets contain 2-D data points that form 20 and 35 clusters, respectively. The numbers of data points for the A1 and A2 datasets are 3,000 and 5,250, respectively. As shown in Figure 3 and Figure 4, the cluster memberships and prototypes vary for different algorithms. Despite the mild overlapping among the clusters, all the algorithms under consideration were able to identify most of the clusters. Different from the A1 and A2 datasets, the Aggregation

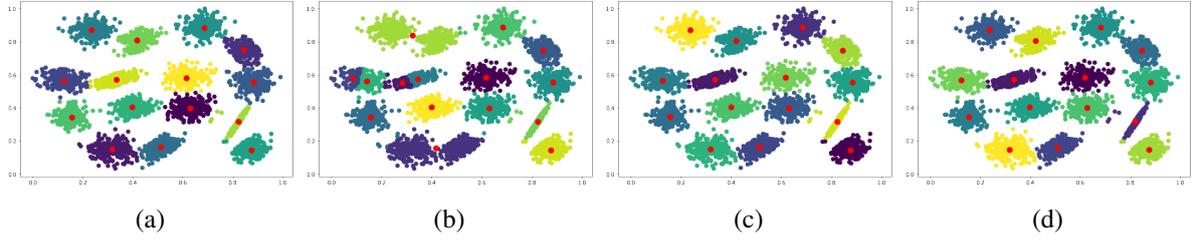


Fig. 7. Visual clustering results of various clustering methods on S1 dataset (red points denote cluster centroids): (a) FCM, (b) K-Means, (c) K-Means++, and (d) POCS-based.

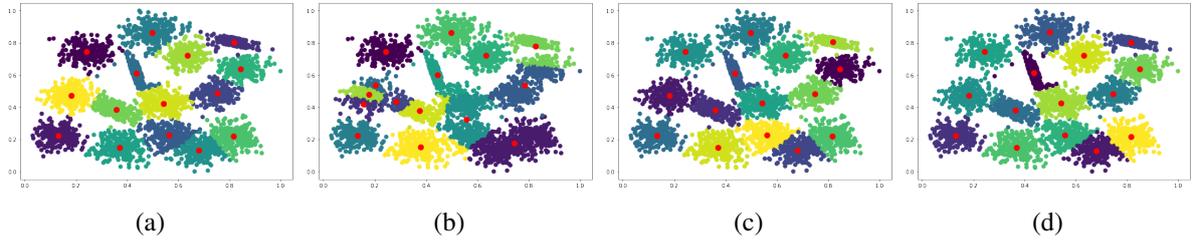


Fig. 8. Visual clustering results of various clustering methods on S2 dataset (red points denote cluster centroids): (a) FCM, (b) K-Means, (c) K-Means++, and (d) POCS-based.

Table 2

Execution time of different algorithms on various datasets (sec.)						
	A1	A2	Aggregation	R15	S1	S2
FCM	0.122	1.264	0.026	0.038	0.243	0.285
K-Means	0.073	0.233	0.012	0.010	0.128	0.184
K-Means++	0.044	0.132	0.016	0.007	0.044	0.069
POCS-based	0.042	0.121	0.018	0.009	0.063	0.075

dataset shown in Figure 5 has 7 clusters with a total of 788 instances distributed in different densities. As the outcomes shown in Figure 5, the FCM, K-Means++, and POCS-based algorithms are considered to yield more favorable results than the K-Means method in this dataset. Despite the fact that these algorithms are not able to correctly segregate all the clusters, the results can be considered natural since these clustering methods are developed based on Euclidean distance for their basic operations and may not be well applicable to data sets with disproportionately distributed data points. The R15 dataset shown in Figure 6 contains 600 data points which are divided into 15 clusters. One of the clusters locates in the vicinity of the center of the data space and the remaining clusters surround the center cluster on two layers of circular orientation. As can be seen from Figure 6, which displays the clustering results, most of the algorithms under consideration were able to correctly identify the cluster prototypes and groups, whereas only the result yielded by the K-Means approach was not very favorable. The S1 and S2 datasets depicted in Figure 7 and Figure 8, respectively, have 5,000 data points that form 15 clusters for each data set. As the results shown in Figure 7 and Figure 8, the algorithms under consideration were able to correctly segregate most of the cluster groups.

In addition, execution time and clustering error were adopted as comparison metrics to quantitatively assess the performance of each algorithm. On every dataset, each algorithm was executed 10 times

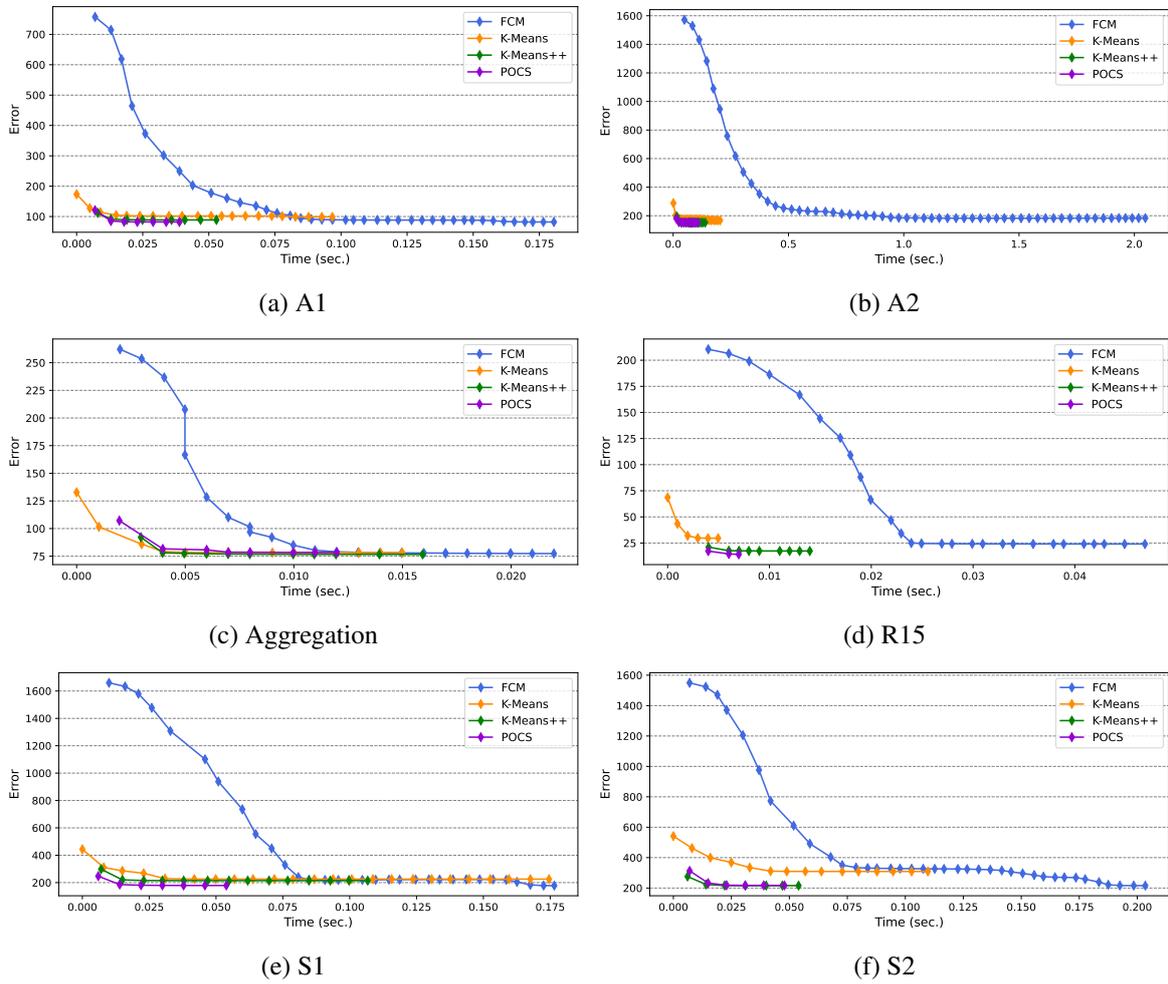


Fig. 9. Typical convergence processes of various clustering methods on the 2-D synthetic datasets: (a) A1, (b) A2, (c) Aggregation, (d) R15, (e) S1, and (f) S2.

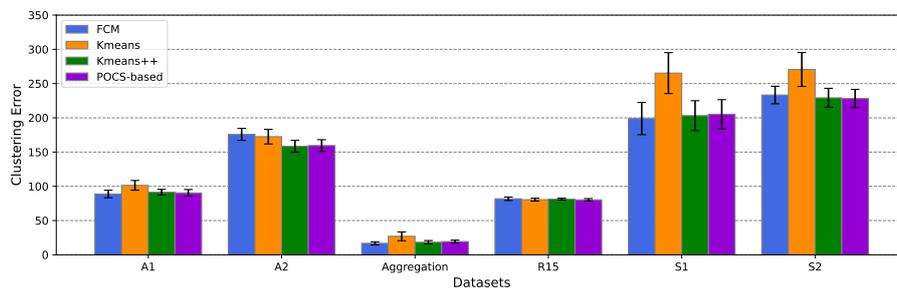


Fig. 10. Comparison of various clustering methods in terms of clustering error on the synthetic 2-D datasets.

and the mean values of convergence speed and clustering error are presented. Table 2 summarizes the measurements of the average convergence time of different algorithms on various synthetic 2-D data

Table 3
Details of the UCI benchmark datasets.

	Iris	BCW	Seeds	Ionos	Sonar	Wine	Glass	Ecoli	PD	QSAR	Chess
#Clusters	3	2	3	2	2	3	6	8	2	2	2
#Attributes	4	10	7	34	60	13	10	8	754	1,024	36
#Instances	150	699	210	351	208	178	214	336	756	1,687	3,196

sets. Based on the results shown in Table 2, the convergence speed of the POCS-based algorithm can be competitively compared with those of the K-Means/K-Means++ approaches and clearly outperforms that of the FCM algorithm. Typical convergence processes of those clustering methods on the synthetic 2-D data sets are illustrated in Figure 9, while Figure 10 shows the comparisons in terms of clustering error of different algorithms after convergence. The clustering error E is defined as:

$$E = \sum_{k=1}^K \sum_{i=1}^{I_k} \|x_k - d_{k,i}\|, \quad (15)$$

where K is the number of clusters, I is the number of data points in one cluster, while x_k and $d_{k,i}$ denote the final prototype and the i^{th} member point of the k^{th} cluster, respectively. As shown in Figure 10, the differences in clustering error among the FCM, K-Means++, and POCS-based algorithms are marginal, whereas the error measurements for the K-Means method are exceptionally higher on S1 and S2 datasets.

4.2. Evaluation on High-dimensional Data Sets

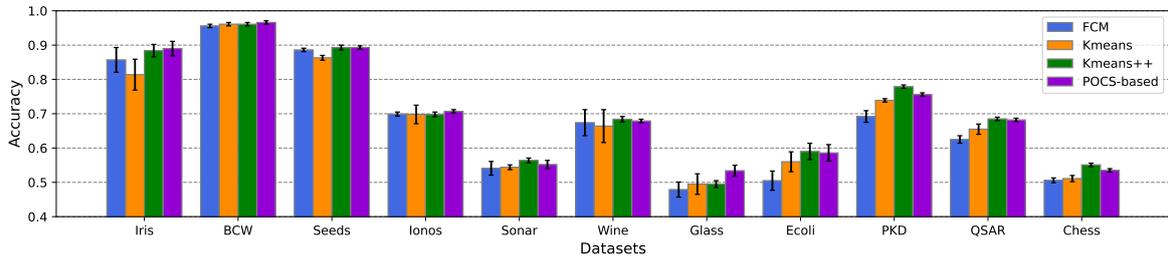
In this section, experiments on 11 benchmark data sets with up to 1,024 attributes have been conducted so as to investigate the performance of the proposed POCS-based clustering algorithm on clustering tasks with high-dimensional data. The data sets were selected from the UCI dataset repository, including Iris, Breast-cancer-Wisconsin (BCW), Seeds, Ionosphere (Ionos), Sonar, Wine, Glass, Ecoli, Parkinson's Disease (PD), QSAR Androgen Receptor (QSAR), and King-Rook vs. King-Pawn (Chess) [33]. A pre-defined number of clusters is also given for each data set. The specifications of the selected benchmark data sets are presented in Table 3. The performances of the algorithms under consideration were compared based on various standard metrics including *Accuracy*, *Precision*, *Recall*, and *F1* score which are measured as follows:

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN}, \quad (16)$$

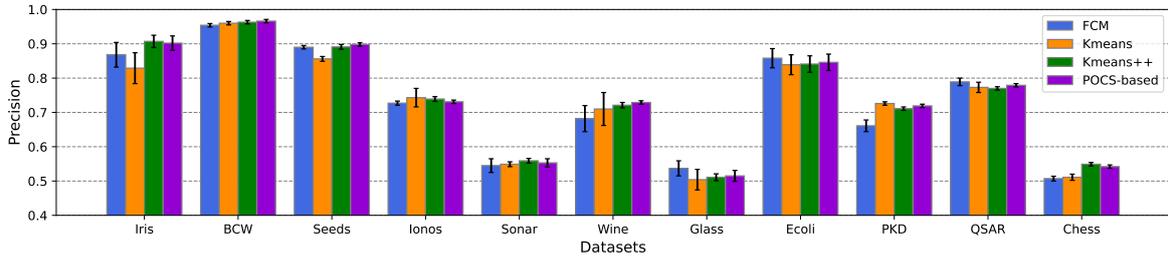
$$Precision = \frac{TP}{TP+FP}, \quad (17)$$

$$Recall = \frac{TP}{TP+FN}, \quad (18)$$

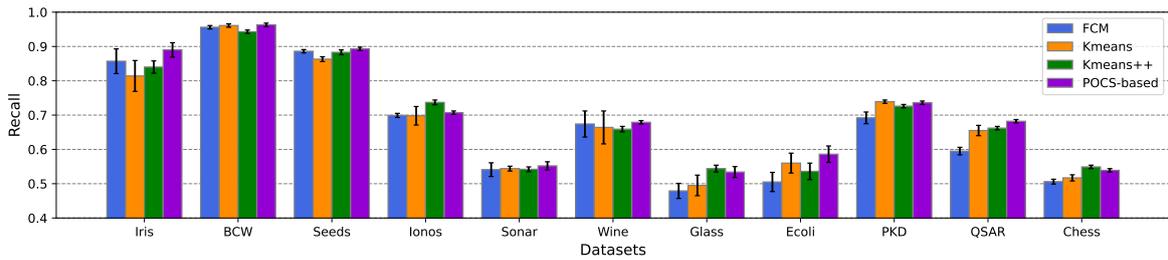
$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN}, \quad (19)$$



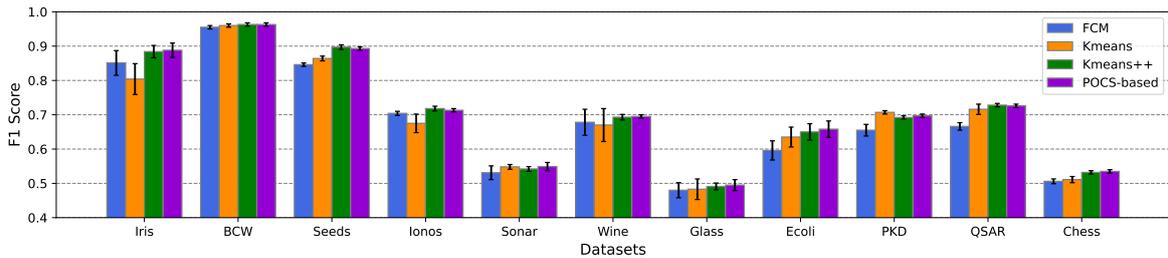
(a) Accuracy



(b) Precision



(c) Recall



(d) F1 Score

Fig. 11. Comparisons of various methods in terms of Accuracy, Precision, Recall, and F1 Score on high-dimensional datasets.

where TP , TN , FP , FN represent true positive, true negative, false positive, and false negative, respectively. Despite the fact that clustering is an unsupervised learning task, the classes can be retrieved when the label information of input data is given. To this end, we determine the class of a cluster by adopting the confusion matrix method [34]. Each algorithm was executed 10 times on each data set and the measurement values were averaged to present the final results.

Table 4

Average execution time (sec.) of different algorithms on various UCI benchmark datasets.

	Iris	BCW	Seeds	Ionos	Sonar	Wine	Glass	Ecoli	PD	QSAR	Chess
FCM	0.005	0.006	0.005	0.008	0.011	0.008	0.015	0.025	0.176	0.614	0.071
K-Means	0.002	0.005	0.003	0.003	0.003	0.001	0.003	0.009	0.025	0.141	0.027
K-Means++	0.002	0.004	0.005	0.010	0.005	0.004	0.007	0.019	0.103	0.103	0.023
POCS-based	0.002	0.005	0.004	0.002	0.007	0.006	0.005	0.007	0.069	0.223	0.025

Table 5

Details of the face data subsets.

	Subset 1	Subset 2	Subset 3	Subset 4	Subset 5
Original Dataset	FCF	LFW	LFW	LFW	LFW
#Clusters	5	7	13	25	86
#Attributes	128	128	128	128	128
#Instances (per cluster)	18-20	41-50	31-40	21-30	11-20
#Instances (total)	118	307	443	613	1,251

The comparisons in terms of *Accuracy*, *Precision*, *Recall*, and *F1* score are summarized in Figure 11. As shown in Figure 11, we can find clearly that the algorithms under the test show similar performances with marginal differences in the magnitude of the evaluation metrics. The POCS-based algorithm can be considered to produce competitive performances compared with the K-Means++ approach and outperform the FCM and K-Means algorithms in most cases. The measurements in terms of average execution time were also examined and summarized in Table 4. From the results summarized in Table 4, the POCS-based algorithm can converge as fast as the K-Means/K-Means++ methods while clearly surpassing the FCM algorithm.

In general, the proposed POCS-based algorithm has shown promising performance when compared with the other classical clustering algorithms in high-dimensional data clustering problems.

4.3. Evaluation on Face Clustering

In this section, we investigate the applicability of the proposed POCS-based clustering algorithm to face clustering by performing clustering as a downstream task. Five subsets of face image data with different numbers of clusters and densities (instances per cluster) were prepared based on two widely used data sets for face clustering tasks, Five Celebrity Faces (FCF) [35] and Labeled Faces in the Wild (LFW) [36]. The FCF dataset contains images of five celebrities (Ben Afflek, Elton John, Jerry Seinfeld, Madonna, and Mindy Kaling) that are divided into training and validation sets. However, since this is a small data set, we merged the training and validation sets to obtain a single data set of 118 images. On the other hand, the LFW dataset is a test set for face verification [37] with 13,233 images of 5,749 identifications which were detected and centered by the Viola–Jones face detector [38]. Since there is a significant imbalance in the face cluster densities with a lot of subjects having less than 10 instances, we created four image subsets from the LFW dataset such that the cluster densities in each subset are relatively equivalent and distributed in various ranges (11-20, 21-30, 31-40, and 41-50). The details of the prepared data subsets in this experiment are summarized in Table 5.

The face clustering system in this experiment has three main steps. In the first step, MultiTask Cascaded Convolutional Neural Network (MTCNN) [39] is adopted with default settings to extract the face region in a given input image. MTCNN is a robust and accurate face detector that has been applied to

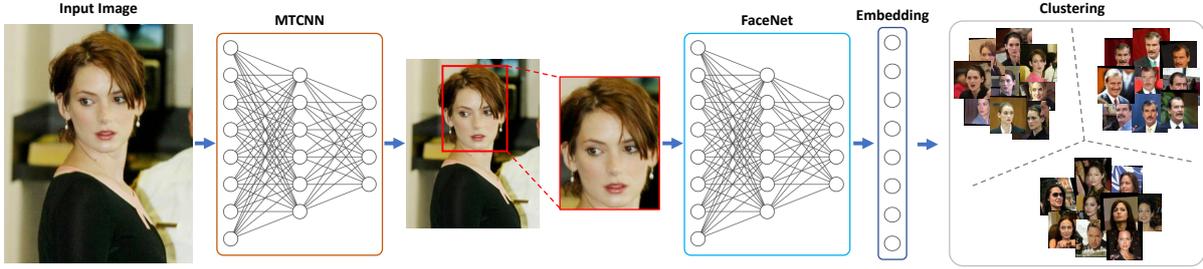


Fig. 12. The pipeline of the face clustering system in this experiment.

Table 6

Comparisons of various clustering algorithms in terms of accuracy (%) and execution time (sec.) in face clustering task.

	Subset 1		Subset 2		Subset 3		Subset 4		Subset 5	
	Acc.	Time								
FCM	93.6	0.18	92.4	0.25	93.9	2.43	91.3	3.38	82.3	4.74
K-Means	98.4	0.01	96.8	0.01	96.5	0.02	95.2	0.03	84.4	0.09
K-Means++	99.5	0.01	99.6	0.02	98.2	0.03	98.4	0.05	91.6	0.19
POCS-based	99.5	0.01	99.7	0.02	98.0	0.03	98.5	0.05	91.9	0.20

numerous face recognition studies [40–42]. In the second step, FaceNet [37] is utilized for face embedding extraction. The input and output shapes of FaceNet are $160 \times 160 \times 3$ and 128×1 , respectively. Thus, each face region detected by MTCNN is cropped and resized to 160×160 pixels before being fed to FaceNet. As a result, a set of 128-D face embeddings is obtained to be used for the clustering process at the last step. The FaceNet model used in this experiment has been verified on the LFW dataset with an accuracy of 99.63% [37], hence, the model can guarantee to produce high-quality feature embeddings for downstream tasks such as classification and clustering. Figure 12 illustrates the pipeline of the face clustering system in this experiment.

Table 6 summarizes the performances of the FCM, K-Means, K-Means++, and POCS-based clustering algorithms in terms of mean recognition accuracy and average processing speed after 10 times of execution for each algorithm. As can be seen from Table 6, the POCS-based clustering algorithm can perform competitively compared with the K-Means++ algorithm, while adequately outperforming the K-Means algorithm and considerably surpassing the FCM approach in terms of accuracy. For the subsets with the number of clusters lower than or equal to 25, the proposed POCS-based clustering algorithm can provide superior accuracy (higher than 98%), whereas giving adequate results on the more challenging dataset (Subset 5) with 91.9% accuracy. In terms of processing time, the K-Means algorithm is considered the most efficient approach when compared with the other methods. Even though the K-Means, K-Means++, and POCS-based algorithms achieve relatively equivalent convergence speeds in Subset 1, Subset 2, and Subset 3, the K-Means algorithm is considered to converge faster when the number of clusters becomes greater (Subset 4 and Subset 5). However, the fast convergence of the K-Means method comes at a price of deteriorated accuracy. Hence, we justify that the POCS-based algorithm can guarantee a better trade-off between effectiveness and efficiency over the K-Means algorithm. On the other hand, the performance of the FCM algorithm is not able to surpass those of the other algorithms in terms of both accuracy and execution time in this experiment.

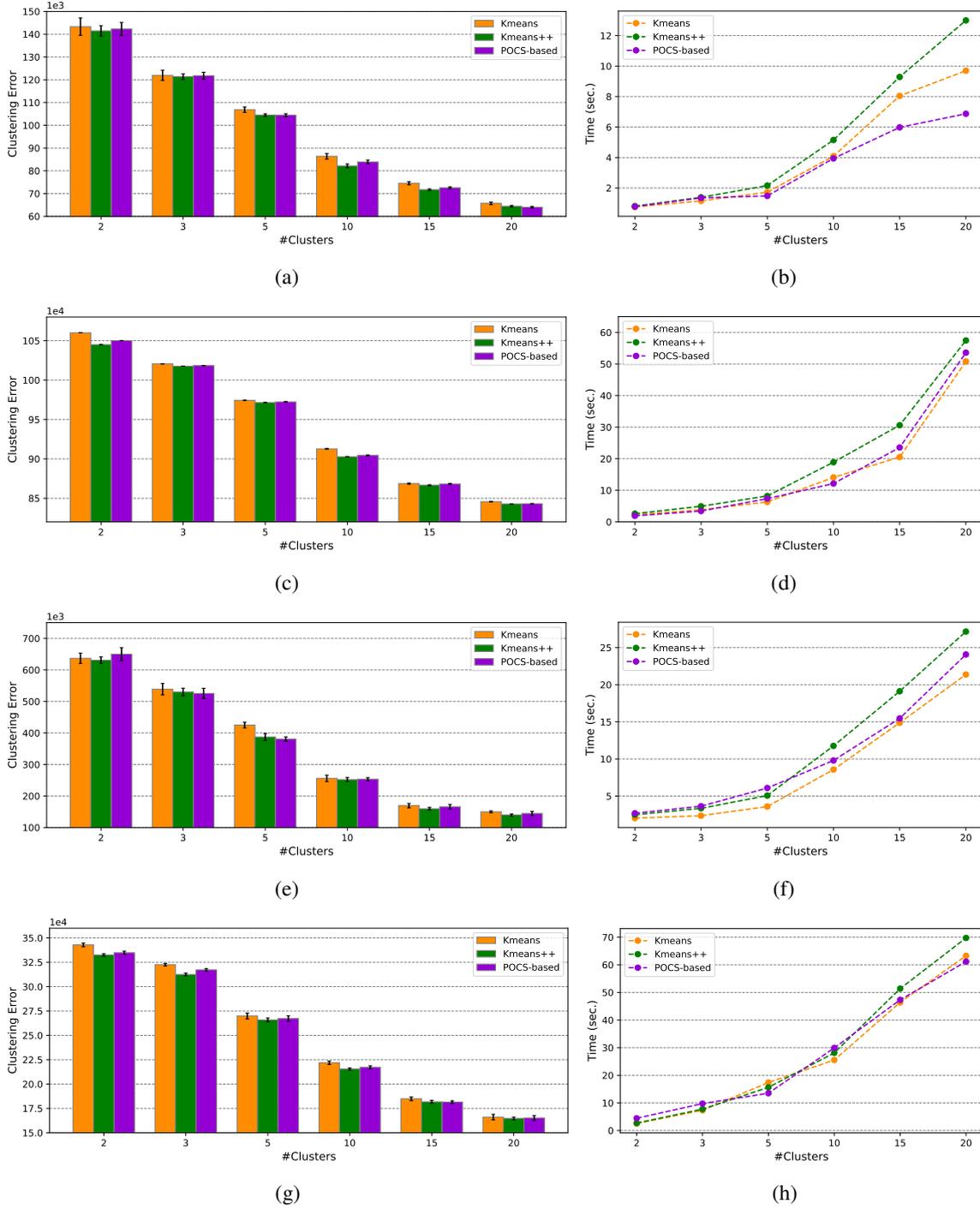


Fig. 13. Comparisons in terms of clustering error (bar charts) and convergence time (line graphs) with different numbers of clusters on big data sets: (a,b) 3D Road Network, (c,d) Poker Hand, (e,f) MetroPT-3, and (g,h) Electric Power Consumption.

Table 7
Details of the big data sets.

	3D Road Network [45]	Poker Hand [43]	MetroPT-3 [44]	Electric Power Consumption [46]
#Attributes	4	10	15	9
#Instances	434,874	1,025,010	1,516,948	2,075,259

4.4. Evaluation on Big Data Sets

The performance of the POCS-based clustering algorithm on big data is examined in this section. To this end, four realistic data sets including 3D Road Network [45], Poker Hand [43], MetroPT-3 [44], and Electric Power Consumption [46] were utilized to conduct experiments. The descriptions of these data sets are summarized in Table 7.

In this examination, we only compare the POCS-based algorithm against the K-Means and K-Means++ approaches, since the FCM algorithm produces close results as compared to the K-Means method while its efficiency tends to deteriorate when the volume of data becomes larger [47]. Similar to preceding experiments, the K-Means, K-Means++, and POCS-based clustering approaches have been compared in terms of the average values of clustering error and convergence speed after 10 times of execution for each algorithm. Since a predefined number of clusters is not provided for the data sets in this experiment, we compare the performances of the algorithms under consideration via different numbers of clusters K , with $K = \{2, 3, 5, 10, 15, 20\}$.

The comparisons in terms of clustering error and execution time are given in Figure 13. As shown in Figure 13, in terms of clustering error, all the algorithms under examination generally provide competitive performances, and the error is lower when the number of clusters becomes higher. The K-Means++ algorithm is considered to obtain the best results in most cases, whereas the POCS-based algorithm usually produces better results than the K-Means method. With regard to the execution time, all three algorithms converge with competitive speed measures when $K = \{2, 3\}$, whereas the differences can be observed more apparently when K becomes greater. Specifically, the K-Means++ algorithm converges slower than the K-Means and POCS-based algorithms when the number of clusters is greater than 5. Overall, for the clustering tasks with big data, the POCS-based algorithm can work with a more balanced trade-off between effectiveness and efficiency. Specifically, the POCS-based algorithm can be compared competitively with the K-Means algorithm in terms of convergence speed while producing favorable clustering error as compared with the K-Means++ approach.

Consequently, the proposed POCS-based clustering algorithm generally yields a competitive performance when compared with those of other clustering algorithms on various types of data sets and applications under our investigations. These results imply that the POCS-based clustering algorithm has potential in a wide range of clustering tasks.

4.5. Data Sensitivity Analysis

Due to the strong dependency on the POCS optimization method, the proposed clustering approach inherits significant characteristics of the POCS algorithm including a possibility of unexpected convergence. The POCS methods have generally demonstrated robustness in finding local minimum solutions based on predefined objective functions with low complexity [48]. However, there is a specific scenario in which the minimum mean square error solutions of POCS can potentially produce undesirable outcomes [31]. This scenario occurs when a dataset includes outliers. In such case, the proposed algorithm

assigns higher weight values to outliers according to the importance constraint described in Eq. (10) and results in outliers' greater influence when compared to other data points. Note that this only happens when the number of data points in the data set is relatively small. Consequently, the algorithm converges to a position midway between the cluster and the outliers. However, we cannot judge whether this convergence is considered desirable or not because we may not know whether the outlier data are valid or noisy data in advance. Note that this problem is common in unsupervised learning algorithms. However, when the number of data points is sufficient and the outlier rate is under control, the proposed POCS-based algorithm can show competitive performances across various types of data and tasks which have been demonstrated through extensive experiments in the paper.

5. Conclusions

A new clustering method, called Projection Onto Convex Set (POCS)-based clustering algorithm, is proposed in this paper. The proposed POCS-based clustering algorithm treats each data point in a given data set as a convex set and applies the convergence property of the parallel POCS method in order to optimize the cluster prototypes. Based on extensive experiments on various synthetic and real-world datasets, the proposed POCS-based algorithm has shown superior performance when compared with the Fuzzy C-Means (FCM) clustering algorithm in most cases and is competitive with the K-Means/K-Means++ algorithms. Furthermore, the execution speed and simplicity are additional important advantages of the POCS-based algorithm over the FCM clustering approach. The proposed clustering algorithm also shows its applicability when it is integrated with other deep learning-based models to serve as a downstream task in a more complex system like face recognition. In general, we have empirically demonstrated that the proposed POCS-based algorithm can be considered a promising tool for a wide range of data clustering problems. However, the proposed algorithm still poses a challenge in achieving satisfactory convergence when confronted with substantial noise within a sparse database. In this respect, our future work is targeted toward addressing this particular concern.

References

- [1] D. Xu and Y. Tian, A comprehensive survey of clustering algorithms, *Annals of Data Science* **2** (2015), 165–193.
- [2] A. Javed, B.S. Lee and D.M. Rizzo, A benchmark study on time series clustering, *Machine Learning with Applications* **1** (2020), 100001.
- [3] D.-C. Park, Centroid neural network for unsupervised competitive learning, *IEEE Transactions on Neural Networks* **11**(2) (2000), 520–528.
- [4] D.-C. Park and Y.-J. Woo, Weighted centroid neural network for edge preserving image compression, *IEEE transactions on neural networks* **12**(5) (2001), 1134–1146.
- [5] D.-C. Park, O.-H. Kwon and J. Chung, Centroid neural network with a divergence measure for GPDF data clustering, *IEEE transactions on neural networks* **19**(6) (2008), 948–957.
- [6] D.-C. Park, Classification of audio signals using Fuzzy c-Means with divergence-based Kernel, *Pattern Recognition Letters* **30**(9) (2009), 794–798.
- [7] D.-C. Park, Centroid neural network with weighted features, *Journal of Circuits, Systems, and Computers* **18**(08) (2009), 1353–1367.
- [8] M.T. Ngoc and D.-C. Park, Centroid neural network with pairwise constraints for semi-supervised learning, *Neural Processing Letters* **48**(3) (2018), 1721–1747.
- [9] J.A. Hartigan and M.A. Wong, Algorithm AS 136: A k-means clustering algorithm, *Journal of the royal statistical society. series c (applied statistics)* **28**(1) (1979), 100–108.
- [10] L.-A. Tran and M.-H. Le, Robust u-net-based road lane markings detection for autonomous driving, in: *2019 International Conference on System Science and Engineering (ICSSE)*, IEEE, 2019, pp. 62–66.

- [11] X. Jin and J. Han, K-Medoids Clustering, Encyclopedia of Machine Learning, Springer US, 2010.
- [12] D. Arthur and S. Vassilvitskii, K-means++ the advantages of careful seeding, in: *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, 2007, pp. 1027–1035.
- [13] M.-C. Hung and D.-L. Yang, An efficient fuzzy c-means clustering algorithm, in: *Proceedings 2001 IEEE International Conference on Data Mining*, IEEE, San Jose, CA, USA, 2001, pp. 225–232.
- [14] Z. Wang, Comparison of four kinds of fuzzy C-means clustering methods, in: *2010 Third International Symposium on Information Processing*, IEEE, Qingdao, Shandong, China, 2010, pp. 563–566.
- [15] D.C. Park and I. Dagher, Gradient based fuzzy c-means (GBFCM) algorithm, in: *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, Vol. 3, IEEE, Orlando, FL, USA, 1994, pp. 1626–1631.
- [16] L.M. Bregman, The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming, *USSR computational mathematics and mathematical physics* 7(3) (1967), 200–217.
- [17] L. Gubin, B.T. Polyak and E. Raik, The method of projections for finding the common point of convex sets, *USSR Computational Mathematics and Mathematical Physics* 7(6) (1967), 1–24.
- [18] M. Tofghi, K. Kose and A.E. Cetin, Denoising using projection onto convex sets (pocs) based framework, *arXiv preprint arXiv:1309.0700* (2013).
- [19] H. Artes, F. Hlawatsch and G. Matz, Efficient POCS algorithms for deterministic blind equalization of time-varying channels, in: *Globecom'00-IEEE. Global Telecommunications Conference. Conference Record (Cat. No. 00CH37137)*, Vol. 2, IEEE, San Francisco, CA, USA, 2000, pp. 1031–1035.
- [20] M. Elad and A. Feuer, Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images, *IEEE transactions on image processing* 6(12) (1997), 1646–1658.
- [21] R.J. Marks II and D.K. Smith, Gerchberg-type linear deconvolution and extrapolation algorithms, in: *Transformations in Optical Signal Processing*, Vol. 373, SPIE, Seattle, United States, 1984, pp. 161–178.
- [22] L.-A. Tran, T.-D. Nguyen, T.-D. Do, C.N. Tran, D. Kwon and D.-C. Park, Embedding Clustering via Autoencoder and Projection onto Convex Set, in: *2023 International Conference on System Science and Engineering (ICSSE)*, IEEE, 2023, pp. 128–133.
- [23] F. Kamalabadi and B. Sharif, Robust regularized tomographic imaging with convex projections, in: *IEEE International Conference on Image Processing 2005*, Vol. 2, IEEE, Genova, Italy, 2005, p. II–205.
- [24] W. Lian, Y. Liang, Q. Pan, Y.-m. Chen and H.-c. Zhang, A Clustering Successive POCS Algorithm for Fast Point Matching, in: *2006 International Conference on Machine Learning and Cybernetics*, IEEE, Dalian, China, 2006, pp. 3903–3908.
- [25] L.-A. Tran, H.M. Deberneh, T.-D. Do, T.-D. Nguyen, M.-H. Le and D.-C. Park, POCS-based Clustering Algorithm, in: *2022 International Workshop on Intelligent Systems (IWIS)*, IEEE, 2022, pp. 1–6.
- [26] A. Ben-Tal and A. Nemirovski, Robust convex optimization, *Mathematics of operations research* 23(4) (1998), 769–805.
- [27] S. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge university press, Cambridge, United Kingdom, 2004.
- [28] J. Dattorro, *Convex optimization & Euclidean distance geometry*, Meboo Publishing, Palo Alto, California, United States, 2010.
- [29] S. Theodoridis, *Machine learning: a Bayesian and optimization perspective*, Academic press, Cambridge, Massachusetts, United States, 2015.
- [30] D. Youla and V. Velasco, Extensions of a result on the synthesis of signals in the presence of inconsistent constraints, *IEEE Transactions on Circuits and Systems* 33(4) (1986), 465–468.
- [31] R.Y. Albert, R.J. Marks, K.E. Schubert, C. Baylis, A. Egbert, A. Goad and S. Haug, Dilated POCS: Minimax Convex Optimization, *IEEE Access* 11 (2023), 32733–32742.
- [32] P. Fränti and S. Sieranoja, K-means properties on six clustering benchmark datasets, *Applied intelligence* 48(12) (2018), 4743–4759.
- [33] D. Dua and C. Graff, UCI Machine Learning Repository, 2017. <http://archive.ics.uci.edu/ml>.
- [34] S.V. Stehman, Selecting and interpreting measures of thematic classification accuracy, *Remote sensing of Environment* 62(1) (1997), 77–89.
- [35] Kaggle, 5 Celebrity Faces Dataset, 2016. <https://www.kaggle.com/dansbecker/5-celebrity-faces-dataset>.
- [36] G.B. Huang, M. Mattar, T. Berg and E. Learned-Miller, Labeled faces in the wild: A database for studying face recognition in unconstrained environments, in: *Workshop on faces in 'Real-Life' Images: detection, alignment, and recognition*, 2008.
- [37] F. Schroff, D. Kalenichenko and J. Philbin, Facenet: A unified embedding for face recognition and clustering, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, Boston, Massachusetts, 2015, pp. 815–823.
- [38] P. Viola and M. Jones, Rapid object detection using a boosted cascade of simple features, in: *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, Vol. 1, Ieee, 2001, p. I–I.
- [39] K. Zhang, Z. Zhang, Z. Li and Y. Qiao, Joint face detection and alignment using multitask cascaded convolutional networks, *IEEE signal processing letters* 23(10) (2016), 1499–1503.
- [40] A. Koubaa, A. Ammar, A. Kanhouch and Y. AlHabashi, Cloud versus edge deployment strategies of real-time face recognition inference, *IEEE Transactions on Network Science and Engineering* 9(1) (2021), 143–160.

- [41] K. Krishnapriya, V. Albiero, K. Vangara, M.C. King and K.W. Bowyer, Issues related to face recognition accuracy varying based on race and skin tone, *IEEE Transactions on Technology and Society* **1**(1) (2020), 8–20.
- [42] Z. Deng, X. Peng, Z. Li and Y. Qiao, Mutual component convolutional neural networks for heterogeneous face recognition, *IEEE Transactions on Image Processing* **28**(6) (2019), 3102–3114.
- [43] R. Cattral and F. Oppacher, Poker Hand, 2007, DOI: <https://doi.org/10.24432/C5KW38>.
- [44] N. Davari, B. Veloso, R. Ribeiro and J. Gama, MetroPT-3 Dataset, 2023, DOI: <https://doi.org/10.24432/C5VW3R>.
- [45] M. Kaul, 3D Road Network (North Jutland, Denmark), 2013, DOI: <https://doi.org/10.24432/C5GP51>.
- [46] G. Hebrail and A. Berard, Individual household electric power consumption, 2012, DOI: <https://doi.org/10.24432/C58K54>.
- [47] S. Ghosh and S.K. Dubey, Comparative analysis of k-means and fuzzy c-means algorithms, *International Journal of Advanced Computer Science and Applications* **4**(4) (2013).
- [48] M. Rydstrom, E.G. Strom and A. Svensson, Robust sensor network positioning based on projections onto circular and hyperbolic convex sets (POCS), in: *2006 IEEE 7th Workshop on Signal Processing Advances in Wireless Communications*, IEEE, 2006, pp. 1–5.