Annals of Operations Research (2024) 339:1481–1497
https://doi.org/10.1007/s10479-022-04642-2

ORIGINAL RESEARCH

# Orthogonal nonnegative matrix factorization problems for clustering: A new formulation and a competitive algorithm

**Ja'far Dehghanpour**[1] · **Nezam Mahdavi-Amiri**[1]

Accepted: 18 February 2022 / Published online: 17 March 2022


**Abstract**
Orthogonal Nonnegative Matrix Factorization (ONMF) with orthogonality constraints on a matrix has been found to provide better clustering results over existing clustering problems. Because of the orthogonality constraint, this optimization problem is difficult to solve. Many of the existing constraint-preserving methods deal directly with the constraints using different techniques such as matrix decomposition or computing exponential matrices. Here, we propose an alternative formulation of the ONMF problem which converts the orthogonality constraints into non-convex constraints. To handle the non-convex constraints, a penalty function is applied. The penalized problem is a smooth nonlinear programming problem with quadratic (convex) constraints that can be solved by a proper optimization method. We first make use of an optimization method with two gradient projection steps and then apply a post-processing technique to construct a partition of the clustering problem. Comparative performance analysis of our proposed approach with other available clustering methods on randomly generated test problems and hard synthetic data-sets shows the outperformance of our approach, in terms of the obtained misclassification error rate and the Rand index.

**Keywords** Orthogonal Nonnegative Matrix Factorization · Isoperimetry problem · Clustering · Optimization problem with orthogonality constraints

## 1 Introduction

Optimization problems with orthogonality constraints posed as

$$\min_{X\in\mathbb{R}^{n\times k}} \quad F(X) \quad s.t. \quad X^TX = I_k, \tag{1a}$$

have wide applications in various areas such as polynomial optimization, combinatorics, eigenvalue problems, and clustering (Jiang and Dai 2015). These problems are difficult to solve since the orthogonality constraints may lead to several local solutions and, in particular,

✉ Nezam Mahdavi-Amiri
nezamm@sharif.edu

Ja'far Dehghanpour
94@sharif.edu

[1] Faculty of Mathematical Sciences, Sharif University of Technology, P. O. Box: 11155-9415, Tehran, Iran

several of these problems are NP-hard. There is no guarantee that a global solution will be obtained except for a few simple cases; e.g., finding the extreme eigenvalues. It is not easy to even generate a sequence of feasible points since it can be numerically expensive to preserve the orthogonality constraints. Most existing methods for preserving constraints either use re-orthogonalization of the matrix or generate points along geodesics of $\mathcal{M}_n^k = \{X \in \mathbb{R}^{n\times k} : X^T X = I\}$. Matrix factorizations such as singular value decomposition (SVD) are needed for the former, and the latter must solve partial differential equations (PDEs) or compute exponential matrices. To avoid these difficulties, we propose an alternative formulation that converts the orthogonality constraints into non-convex equality (or inequality) constraints. To handle these non-convex constraints, a penalty function is applied. The penalized problem is a smooth nonlinear programming problem with quadratic constraints that can be solved by a proper optimization method.

The nonnegative matrix factorization (NMF) problem proposed by Paatero and Tapper (1994) has a wide range of applications, such as pattern recognition, chemical engineering, fault diagnosis, and outlier detection (Banker et al. 2017; Duan et al. 2009; Tosyali et al. 2020). The orthogonal nonnegative matrix factorization (ONMF) problem can be interpreted as the NMF problem, with an additional orthogonality constraint that significantly changes the nature of the problem, making it suitable for clustering (Li et al. 2020; Peng et al. 2020). Ding et al (2006) studied NMF with orthogonality constraint for the first time and showed its effectiveness in clustering. Following that, several ONMF algorithms have been developed for a wide range of applications (Pompili et al. 2014). Most of these algorithms use a multiplicative updating framework on the Stiefel manifold $\mathcal{M}_n^k$ (iteratively updating matrices by taking the element-wise product with other computed non-negative matrices (He et al. 2020; Pan and Ng 2018)). Other approaches include hierarchical alternating least squares (HALS) (Kimura et al. 2015), and penalty function utilization for the orthogonality constraints (Del Buono 2009).

Here, we introduce an approach for solving the isoperimetry and ONMF problems using an efficient optimization algorithm. First, we present alternative formulations of the isoperimetry and ONMF problems, converting the orthogonality constraints into a smooth nonlinear programming problem with convex constraints in Section 2.2 and Section 3, respectively. Then, we solve these reformulated problems, in particular the ONMF, efficiently using a dedicated algorithm. It is remarkable that, instead of solving the ONMF problem for the matrix solutions, we convert the problem into subproblems and solve for vector solutions (see Section 3.1). Finally, we apply a post-processing technique to extract a solution to the clustering problem. Comparative computational results are provided in Section 4 and our concluding remarks are given in Section 5.

In summary, here our contributions are listed as follows:

- Reformulation of the matrix orthogonality constraint as a set of non-convex smooth constraints.
- Application of a penalty function including non-convex smooth constraints as penalty terms, leaving out only convex constraints.

## 2 Related works

This section reviews several works, namely $k$-means and isoperimetry problems, that are closely related to our work here. The $k$-means is one of the most popular unsupervised learning approaches being used for solving the well-known clustering problem. Many recent

developments of $k$-means have been reported in the literature (see Fard et al. 2020; Fränti and Sieranoja 2018; Huang et al. 2021; Moreno et al. 2020; Sinaga and Yang 2020; Xia et al. 2020; Yu et al. 2018 for more details). The isoperimetry is an approach for finding a cluster structure in a data-set, characterizing the greatest similarity within a cluster and the greatest dissimilarity between the other clusters, by minimizing the sum of the weights of the edges connecting the specified cluster to the other clusters ( Dinler et al. 2020; Qin et al. 2017).

### 2.1 *K*-means

A fundamental problem of clustering, known as Minimum Sum-of-Squares Clustering (MSSC), is to partition $n$ points based on a minimum sum-of - squares model into $k$ clusters. Given a set $X$ of $n$ points in an $m$-dimensional Euclidean space, denoted by

$$X = \{x_i = (x_{i1}, \ldots, x_{im})^T \in \mathbb{R}^m,\ i = 1, \ldots, n\},$$

the partitional MSSC deals with the assignment of the $n$ points into $k$ disjoint clusters denoted by $A = (A_1, \ldots, A_k)$ centered at cluster centers $c_j (j = 1, \ldots, k)$ based on the total sum-of-squared Euclidean distances of the points $x_i$ from their respective assigned cluster centroids $c_i$, that is,

$$f(X, A) = \sum_{j=1}^{k} \sum_{i=1}^{|A_j|} ||x_i^{(j)} - c_j||^2, \tag{2}$$

where $|A_j|$ is the number of points in $A_j$, and $x_i^{(j)}$ is the $i$th point in $A_j$. Note that if the clusters are known, then the function $f(X, A)$ achieves its minimum when each points is assigned to its closest cluster center. On the other hand, if the points in clusters $A_j$ are fixed, then the function

$$f(X, A_j) = \sum_{i=1}^{|A_j|} ||x_i^{(j)} - c_j||^2 \tag{3}$$

is minimal when

$$c_j = \frac{1}{|A_j|} \sum_{i=1}^{|A_j|} x_i^{(j)}. \tag{4}$$

The classical $k$-means algorithm (McQueen 1967) is described as follows:

(1) Construct $k$ clusters created randomly in a domain containing all the points.
(2) Assign each point to the closest cluster center.
(3) Recalculate cluster centers using current cluster points.
(4) If the predefined criteria are met, then stop; otherwise, go to (2).

Another way to model the MSSC problem is based on the assignment problem. Let $Y = [y_{ij}] \in \mathbb{R}^{n \times k}$ be the assignment matrix defined by

$$y_{ij} = \begin{cases} 1, & \text{if } x_i \text{is assigned to} A_j \\ 0, & \text{otherwise.} \end{cases}$$

As a consequence, the cluster center of the cluster $A_j$ is defined by

$$c_j = \frac{\sum_{l=1}^{n} y_{lj} x_l}{\sum_{l=1}^{n} y_{lj}},$$

which is the mean of all the points in the cluster. Using this, Peng and Wei (2007) introduced the following model for the $k$-means problem:

$$\min_{y_{ij}} \quad \sum_{j=1}^{k}\sum_{i=1}^{n} y_{ij}||x_i - c_j||^2 \tag{5a}$$

$$s.t. \quad \sum_{j=1}^{k} y_{ij} = 1 \ \ (i = 1, \ldots, n), \tag{5b}$$

$$\sum_{i=1}^{n} y_{ij} \geq 1 \ \ (j = 1, \ldots, k), \tag{5c}$$

$$y_{ij} \in \{0, 1\} \ \ (i = 1, \ldots, n, \ j = 1, \ldots, k). \tag{5d}$$

The constraints (5b) ensure that each point $x_i$ is assigned to exactly one cluster, and (5c) ensures that there are exactly $k$ clusters. We can show (5a) in matrix form by Ferebinous norm as follows (Bauckhage 2015):

$$\sum_{j=1}^{k}\sum_{i=1}^{n} y_{ij}||x_i - c_j||^2 = ||X - CY||_F^2,$$

where $X \in \mathbb{R}^{m\times n}$ is a matrix of data vectors $x_i \in \mathbb{R}^m$, $C \in \mathbb{R}^{m\times k}$ is a matrix of cluster centroids $c_j \in \mathbb{R}^m$ and $Y \in \mathbb{R}^{k\times n}$ is the assignment matrix.

### 2.2 Isoperimetry problem

Given a weighted graph $G = (X, E)$, for any partition (subpartition) $A = \{A_1, \ldots, A_k\}$, we define the vector $v$ as follows:

$$v = \left(\frac{w(A_1)}{|A_1|}, \ldots, \frac{w(A_k)}{|A_k|}\right),$$

where $w(A_i)$ is the sum of the weights of edges between $A_i$ and $A_i^c$ $(i.e., \ X - A_i)$ and $|A_i|$ is the number of vertices in the cluster $A_i$, for $1 \leq i \leq k$. We are to find a partition (or subpartition) of vertices so that the norm of $v$ is minimized. For $p = 1$, the mean version of the isoperimetry problem is defined as

$$IPP_k^m(G) = \min_{\{A_i\}_1^k \in D_k(X)} \|v\|_1 = \min_{\{A_i\}_1^k \in D_k(X)} \frac{1}{k}\left(\sum_{i=1}^{k} \frac{c(A_i)}{|A_i|}\right),$$

where $D_k(X)$ is the collection of all the $k$-subpartitions of the set $X$. Actually, the isoperimetry problem is a relaxed version of the normalized cut problem, in such a way that some points may not be assigned to any cluster.

In Dehghanpour-Sahron and Mahdavi-Amiri (2020), Dehghanpour and Mahdavi-Amiri formulated the isoperimetry problem as follows:

$$\min_{Y\in\mathbb{R}^{n\times k}} \quad tr(Y^T LY) \tag{6a}$$

$$s.t. \quad Y^T Y = I_k, \tag{6b}$$

$$Y \geq 0, \tag{6c}$$

where $L$ is the Laplacian matrix and $I_k$ is the $k \times k$ identity matrix. The orthogonal constraint $Y^T Y = I_k$, together with the nonnegativity constraint $Y \geq 0$ ensure that each row of the matrix $Y$ has at most one non-zero entry. Thus, matrix $Y$ in (6) is closest to that in (5a) and shows the cluster assignment of the data set (here, the binary condition is omitted).

As noted, the orthogonality constraint (6b) and the nonnegative constraint (6c) indicate that there is at most one non-zero element in each row of $Y$. We note that every vector $x \in \mathbb{R}^n$ has at most one nonzero element if and only if $||x||_1 = ||x||_2$. Thus, constraint (6b) for matrix $Y$ can be written as follows:

$$||\hat{y}_i^T||_1^2 = ||\hat{y}_i^T||_2^2 \ \ (i = 1, \ldots, n), \tag{7a}$$

$$||y_j||_2^2 = 1 \ \ (j = 1, \ldots, k), \tag{7b}$$

where $\hat{y}_i$ and $y_j$ are the $i$th row and the $j$th column of $Y$, respectively. So, we have a new model for the clustering problem as follows:

$$\min_{Y} \quad tr(Y^T L Y) \tag{8a}$$

$$s.t. \quad ||\hat{y}_i^T||_1^2 = ||\hat{y}_i^T||_2^2 \ \ (i = 1, \ldots, n), \tag{8b}$$

$$||y_j||_2^2 = 1 \ \ (j = 1, \ldots, k), \tag{8c}$$

$$Y \geq 0. \tag{8d}$$

Problem (8) is still difficult to solve for two reasons. First, the constraint (8b) is nonconvex and second, constraints (8b) and (8c) are written across the rows and across the columns of $Y$, respectively. Therefore, it is difficult to applying methods based on decomposition, specially when size of the problem is large. To deal with these issues, we propose a penalized function as follows:

$$\min_{Y} \quad tr(Y^T L Y) + \frac{\rho}{2} \sum_{i=1}^{n} (||\hat{y}_i^T||_1^2 - ||\hat{y}_i^T||_2^2) \tag{9a}$$

$$s.t. \quad ||y_j||_2^2 = 1 \ \ (j = 1, \ldots, k), \tag{9b}$$

$$Y \geq 0, \tag{9c}$$

where scalar $\rho > 0$ is the penalty parameter.

Next, we use the same idea to reformulate the ONMF problem.

## 3 Orthogonal nonnegative matrix factorization

Orthogonal nonnegative matrix factorization (ONMF), an approximate matrix factorization technique with matrix orthogonality conditions and nonnegativity constraints, has recently been shown to work remarkably well for clustering tasks (Pompili et al. 2014). We consider an orthogonal nonnegative matrix factorization problem as follows. Given a $d$ by $n$ nonnegative matrix $M$ and a rank $k$ factorization (with $k < n$), we are to solve

$$\min_{U \in \mathbb{R}^{d \times k}, V \in \mathbb{R}^{k \times n}} \|M - UV\|_F^2 \tag{ONMF}$$

$$s.t. \quad VV^T = I,$$

$$U \geq 0, \ V \geq 0.$$

Here, we consider each column of the matrix $M$ as a point in $\mathbb{R}^d$ and construct the data-set $X = \{x_1, \ldots, x_n\}$, with $x_i$ being the $i$th column of $M$.

Similarly, if we apply to (ONMF) the same procedure used before to convert problem (6) into the problem (9), we get

$$\min_{U\in\mathbb{R}^{d\times k}, V\in\mathbb{R}^{k\times n}} \|M - UV\|_F^2 \tag{10a}$$

$$s.t. ||v_j||_1^2 = ||v_j||_2^2 \ (j = 1, \ldots, n), \tag{10b}$$

$$||\hat{v}_i^T||_2^2 = 1 \ (i = 1, \ldots, k), \tag{10c}$$

$$U \geq 0, \ V \geq 0. \tag{10d}$$

And, the penalized function of the ONMF problem achieves as follow:

$$\min_{U\in\mathbb{R}^{d\times k}, V\in\mathbb{R}^{k\times n}} \|M - UV\|_F^2 + \frac{\rho}{2}\sum_{j=1}^{n}(||v_j||_1^2 - ||v_j||_2^2) \tag{11a}$$

$$s.t. \quad ||\hat{v}_i^T||_2^2 = 1 \ (i = 1, \ldots, k), \tag{11b}$$

$$U \geq 0, \ V \geq 0, \tag{11c}$$

where $\hat{v}_i, i = 1, \ldots, k$, and $v_j, j = 1, \ldots, n$, are the $i$th row and the $j$th column of $V$, respectively. Note that $V$ is an assignment matrix with $v_{ij} \neq 0$ indicating that $x_j$ is in cluster $A_i$, and nonzero elements of the matrix $HV$ are the same as the ones in $V$, where $H$ is a diagonal matrix with $0 < h_{ii} \leq 1$ $(i = 1, \ldots, k)$. So, from clustering point of view, both $V$ and $HV$ provide the same cluster assignment. Also, $(U, V)$ and $(UH^{-1}, HV)$ provide the same objective value in the ONMF problem, i.e. ,

$$\|M - UV\|_F^2 = \|M - UH^{-1}HV\|_F^2.$$

Morever,

$$HV = \begin{pmatrix} h_{11} & 0 & 0 & 0 \\ 0 & h_{22} & 0 & 0 \\ & & & \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & h_{kk} \end{pmatrix} \begin{pmatrix} \hat{v}_1^T \\ \hat{v}_2^T \\ . \\ . \\ \hat{v}_k^T \end{pmatrix} = \begin{pmatrix} h_{11}\hat{v}_1^T \\ h_{22}\hat{v}_2^T \\ . \\ . \\ h_{kk}\hat{v}_k^T \end{pmatrix},$$

where $||h_{ii}\hat{v}_i^T||_2^2 = h_{ii}||\hat{v}_i^T||_2^2 \leq 1$ $(i = 1, \ldots, k)$. Therefore, clustering is not affected if we replace $||\hat{v}_i^T||_2^2 = 1$ $(i = 1, \ldots, k)$, by $||\hat{v}_i^T||_2^2 \leq 1$ $(i = 1, \ldots, k)$, and choose $(UH^{-1}, HV)$ instead of $(U, V)$ in (11). As a result, we have a new model for the ONMF problem as follows:

$$\min_{U\in\mathbb{R}^{d\times k}, V\in\mathbb{R}^{k\times n}} \|M - UV\|_F^2 + \frac{\rho}{2}\sum_{j=1}^{n}(||v_j||_1^2 - ||v_j||_2^2) \tag{12a}$$

$$s.t. \quad ||\hat{v}_i^T||_2^2 \leq 1 \ (i = 1, \ldots, k), \tag{12b}$$

$$U \geq 0, \ V \geq 0. \tag{12c}$$

Now, problem (12) is a nonlinear programming problem with convex (quadratic) constraints, and an efficient optimization method can be applied to solve it.

The Lagrangian function corresponding to the isoperimetry problem without constraint (6c) can be written as

$$\mathcal{L}(Y, \Lambda) = tr(Y^T LY) - \frac{1}{2}tr(\Lambda(Y^T Y - I)), \tag{13}$$

where $\Lambda \in \mathbb{R}^{k\times k}$ is comprised of the Lagrange multipliers. We define $\nabla F(Y) = \mathcal{D}_Y \mathcal{L}(Y, \Lambda)$. Dehghanpour-Sahron and Mahdavi-Amiri (2020) showed that if the similarity matrix $C$ is placed in problem (ONMF) rather than the matrix $M$, the solution of the problem is equivalent to the solution of the isoperimetry problem. As a result, we can solve the problem (12) instead of the problem (8). Numerical results show that problem (12) has advantages: first, it is easier to solve than problem (8), and second, in problem (12), instead of solving the main problem, we can solve $k$ sub-problems, which significantly reducing the computing time.

### 3.1 Solving ONMF problem

We first apply the algorithm proposed by Bolte et al. (2014) known as the PALM algorithm to obtain a solution of problem (12). This algorithm uses two gradient projection steps for $V$ and $U$. Suppose that $F_\rho(U, V) = \|M - UV\|_F^2 + \frac{\rho}{2}\sum_{j=1}^{n}(||v_j||_1^2 - ||v_j||_2^2)$. At iteration $l$ of the algorithm, the variable $V$ is obtained by solving the following problem:

$$V^{l+1} = \arg\min_{V} ||V - B^l||_F^2 \quad s.t.\ V \geq 0,\ ||\hat{v}_i^T||_2 \leq 1,\ i = 1, \ldots, k, \tag{14}$$

where $B^l = V^l - \frac{1}{t^l}\nabla_V F_\rho(U^l, V^l)$ and $t^l > 0$ is a step size. It is remarkable that the objective function and constraints of (14) can be separated with respect to the rows of $V$, and we can decompose the updating of $V$ to $k$ subproblems as (15). At iteration $l$, the $\hat{v}_i$ $(i = 1, \ldots, k)$ are obtained as follows:

$$\tilde{v}_i^{l+1} = \arg\min_{\hat{v}_i^T \geq 0,\ ||\hat{v}_i||_2 \leq 1} ||\hat{v}_i - \hat{b}_i^l||_2^2, \tag{15}$$

$$\hat{v}_i^{l+1} = \tilde{v}_i^{l+1} + \tau(\tilde{v}_i^{l+1} - \hat{v}_i^l), \tag{16}$$

where $\hat{b}_i^l$ $(i = i, \ldots, k)$ are the rows of the matrix $B^l$. We note that problem (15) has a simple solution as follows: partition $\hat{b}_i^l = [\hat{b}_{i-}^l, \hat{b}_{i+}^l]$, where $\hat{b}_{i-}^l = \{\hat{b}_i^l(j) | \hat{b}_i^l(j) \leq 0\}$ and $\hat{b}_{i+} = \{\hat{b}_i^l(j) | \hat{b}_i^l(j) > 0\}$, for $j = 1, \ldots, n$. Then, we get

$$\hat{v}_i^*(j) = \begin{cases} 0, & \text{if } \hat{b}_i^l(j) \in \hat{b}_{i-}^l \\ \dfrac{\hat{b}_i^l(j)}{\max ||\hat{b}_{i+}^l||_2, 1}, & \text{if } \hat{b}_i^l(j) \in \hat{b}_{i+}^l. \end{cases}$$

Equation (16) introduced by Pock and Sabach (2016) is a correction step for the rows of matrix $V$, where $\tau \in (0, 1)$ is a combination parameter. Numerical results show that if $\tau$ is chosen carefully, the resulting correction step turns to reduce the number of iterations of the algorithm significantly and as a result, the running time of the algorithm is reduced. Finally, using a post-processing technique proposed by Dehghanpour-Sahron and Mahdavi-Amiri (2020), we construct a partition for the clustering problem. The aim of this technique is to obtain a 0-1 assignment matrix $V$, which $v_{ij} = 1$ indicates $x_j$ is in cluster $A_i$. It constructs the assignment matrix by rounding the elements of the input matrix to 0 and 1 with predefined criteria. We explain this technique as follows. Suppose that the matrix $V$ is an output of the proposed algorithm (Algorithm 1 below). In each column of the matrix $V$, the maximal element is preserved and the other elements are set to be zero (this ensures that every column of $V$ has only one nonzero element). In each row $i$ of the matrix $V$, the maximal element is

found and named to be $M_i$. In the $i$th row, for $V_{ij} \neq M_i,\ \forall 1 \leq j \leq n$, if $V_{ij} < \frac{4M_i}{3n}$, then we set $V_{ij} = 0$, and an assignment matrix $V$ is obtained.

**Algorithm 1:** Penalty Function Method (PFM) for Solving Problem (ONMF).

Give $U^0 \in \mathbb{R}^{d\times k}$, $V^0 \in \mathbb{R}^{k\times n}$, $\rho > 0, \alpha > 1, \tau \in (0, 1)$ and set $r = 0$.
**While** $||\nabla F(Y)||_F > \epsilon$ **do**

Set $l = 0$, $U^0 = U^r$, $V^0 = V^r$.
**Repeat**
**For** $i = 1$ till $k$ **do**

$$\tilde{v}_i^{l+1} = \arg \min_{\hat{v}_i \geq 0,\ ||\hat{v}_i^T||_2 \leq 1} ||\hat{v}_i^T - \hat{b}_i^l||_2^2,$$

$$\hat{v}_i^{l+1} = \tilde{v}_i^{l+1} + \tau(\tilde{v}_i^{l+1} - \hat{v}_i^l).$$

**Endfor**
Find step size $t^l$, satisfying the Armijo-Wolfe line search condition.
$U^{l+1} = \max\{U^l - \frac{1}{t^l}\nabla_U F_\rho(U^l, V^{l+1}), 0\}$.
Set $l = l + 1$.
**Until** $||V^l - V^{l-1}||_F \leq \epsilon$
Set $\rho = \alpha\rho$ and $r = r + 1$.
Set $V^r = V^l$ and $U^r = U^l$.

**Endwhile**
Apply post-processing technique to matrix $V^r$ to get a new matrix $V$.

Post-processing technique.

Set $V = V^r$.
**For** all columns of $V$ **do**

Preserve the maximal element and set the other elements to zero to get the matrix $V$.

**Endfor**
**For** $i = 1$ till $k$ **do**

Find the maximal element in row $i$ of $V$ and store it in $M_i$.

**For** $j = 1$ till $n$ **do**

**If** $V_{ij} < \frac{4M_i}{3n}$ **then** set $V_{ij} = 0$ **else** set $V_{ij} = 1$.

**Endfor**

**Endfor**

### 3.2 Convergence analysis of the proposed algorithm

We first notice that any local minimal solution of (12) is feasible for (10). We know that for a non-convex problem, a local minimal solution cannot be computed in general, and we can only obtain a stationary point under some proper conditions (Bertsekas 1999). Suppose that $(U^\rho, V^\rho)$ is bounded and is a stationary point of (12) and $(U^\rho, V^\rho) \to (U^*, V^*)$ as

**Table 1** Compared clustering algorithms

| Notation | Description |
|---|---|
| NJW | Ng-Jordan-Weis algorithm (Ng et al. 2002) |
| DJS | Daneshgar-Javadi-ShariyatRasavi algorithm (Daneshgar et al. 2013) |
| SKA | Standard $k$-means algorithm (Arthur and Sergi 2007) |
| PGAG | Pompili-Gillis-Absil-Glineur algorithm (Pompili et al. 2014) |
| KP | Kim-Park algorithm (Kim and Park 2011) |
| KTK | Kimura-Tanaka-Kudo algorithm (Kimura et al. 2015) |
| YFS | Yang-Fu-Sidiropoulos algorithm (Yang et al. 2017) |
| SY | Sinaga-Yang algorithm (Sinaga and Yang 2020) |
| DMA | Dehghanpour-Mahdavi-Amiri algorithm (Dehghanpour-Sahron and Mahdavi-Amiri 2020) |
| PFM | Our proposed algorithm (Penalty Function Method) |

**Table 2** Misclassification error rates and Rand indices on 4 test problems of [41]

| Data-set | $n$ | $k$ | NJW | DJS | SKA | PGAG | KP | KTK | YFS | SY | DMA | PFM |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2moon | 300 | 2 | **0** | **0** | 0.223 | 0.131 | 0.14 | 0.135 | 0.12 | 0.009 | **0** | **0** |
| 4donut | 700 | 4 | 0.31 | 0.14 | 0.2471 | 0.14 | 0.18 | 0.154 | 0.127 | 0.07 | 0.06 | **0.03** |
| 6moon | 900 | 6 | 0.291 | 0.261 | 0.2744 | 0.242 | 0.37 | 0.268 | 0.205 | 0.18 | 0.17 | **0.09** |
| spiral | 1300 | 2 | 0.24 | **0.07** | 0.2631 | 0.37 | 0.44 | 0.412 | 0.3 | 0.156 | 0.18 | 0.14 |
| **Rand-Index** | | | 0.7528 | 0.8675 | 0.7450 | 0.7387 | 0.6647 | 0.718 | 0.758 | 0.843 | 0.8679 | **0.8902** |

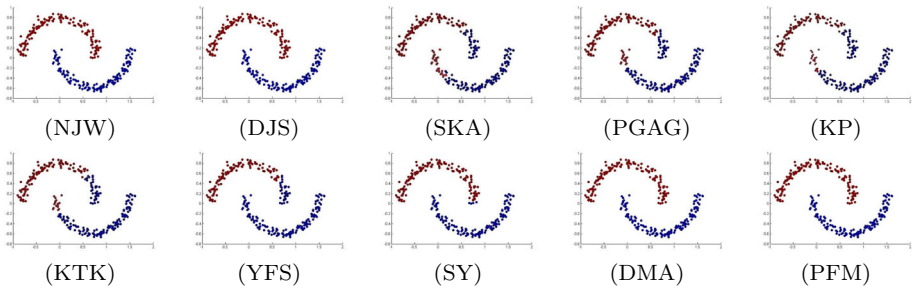


**Fig. 1** Results due to ten algorithms on the “2moon" test problem

**Table 3** Misclassification error rates and Rand indices on 4 test problems of [42]

| Data-set | $n$ | $k$ | NJW | DJS | SKA | PGAG | KP | KTK | YFS | SY | DMA | PFM |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Flame | 240 | 2 | 0.35 | 0.22 | 0.162 | 0.047 | 0.012 | 0.06 | 0.034 | 0.007 | 0.012 | **0.004** |
| R15 | 600 | 15 | 0.21 | 0.05 | 0.003 | 0.22 | 0.03 | 0.248 | 0.174 | 0.003 | 0.003 | **0.0017** |
| Aggregation | 788 | 7 | 0.36 | 0.16 | 0.14 | 0.22 | 0.18 | 0.246 | 0.168 | 0.102 | 0.09 | **0.03** |
| D31 | 3100 | 31 | 0.23 | 0.28 | 0.1074 | 0.21 | 0.16 | 0.153 | 0.248 | 0.08 | 0.02 | **0.01** |
| **Rand-Index** | | | 0.7447 | 0.7722 | 0.8976 | 0.7953 | 0.8606 | 0.8139 | 0.7753 | 0.9205 | 0.9170 | **0.9410** |

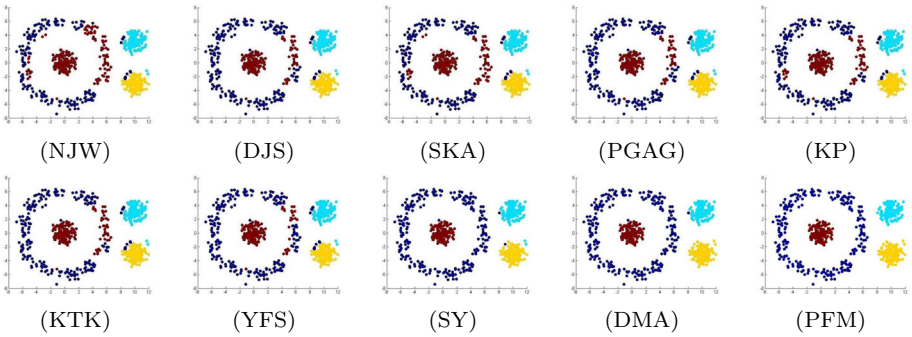


**Fig. 2** Results due to ten algorithms on the "4donut" test problem

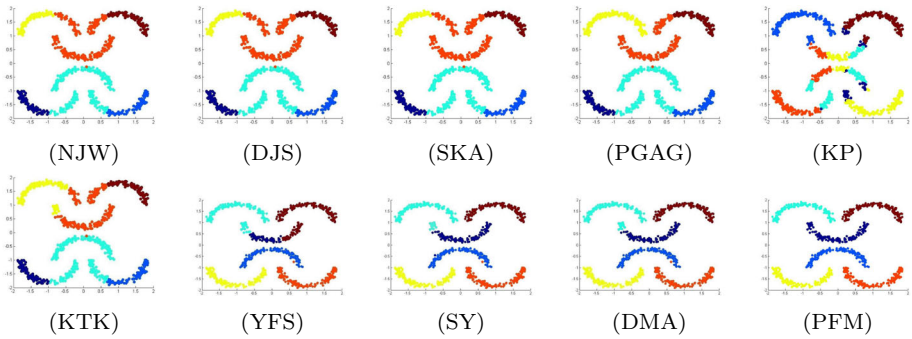


**Fig. 3** Results due to ten algorithms on the "6moon" test problem

$\rho \to \infty$. It is well known that if the Mangasarian-Fromovitz constraint qualification (MFCQ) (Facchinei and Pang 2007) holds for (10) at $(U^*, V^*)$, then $(U^*, V^*)$ is a stationary point of problem (10). According to the obtained convergence results for the algorithms of (Bolte et al. 2014) and (Pock and Sabach 2016), with a proper choice of $t^l$ (obtained from a proper line search to ensure convergence of the iterates (Bertsekas 1999)) along with increasing penalty parameter $\rho$, Algorithm 1 can obtain a bounded stationary point of problem (12). Moreover, the convergence rate of Algorithm 1 depends on the utilized algorithm for solving subproblems (15). In Shefi and Teboulle (2016), it is reported that the PALM algorithm has a global convergence with an asymptotic sublinear convergence rate.

## 4 Comparative results

We implemented our proposed algorithm and other clustering methods on MATLAB R2012a environment in a Windows 7 machine with a 2.40GHz CPU and 4.00 GB RAM.

The numerical results are presented in two parts. In Sect. 4.1, we compared our proposed algorithm (PFM) with other related algorithms (listed in Table 1) on some hard artificial benchmark problems. Moreover, we also report numerical results of PFM on randomly generated graphs in Sect. 4.2 for an extensive evaluation of the performance of our proposed algorithm. In all the tables, the best performance (having minimum error and highest Rand index) is highlighted in bold and the second best is specified by an underline.

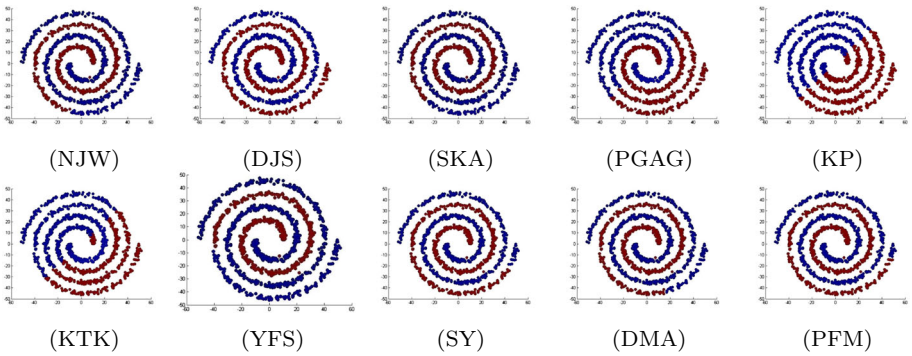


**Fig. 4** Results due to ten algorithms on the "spiral" test problem

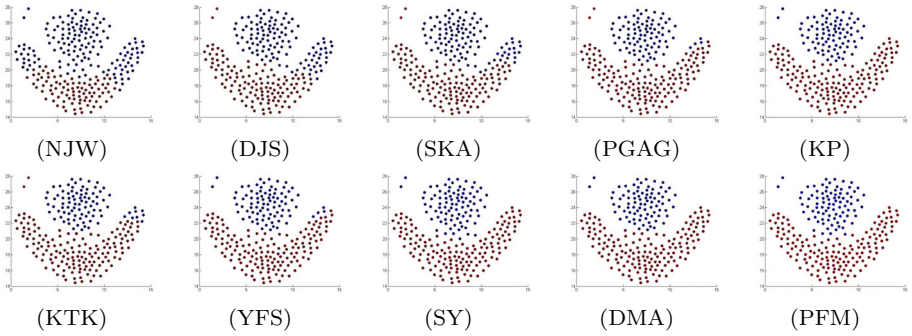


**Fig. 5** Results due to ten algorithms on the "Flame" test problem

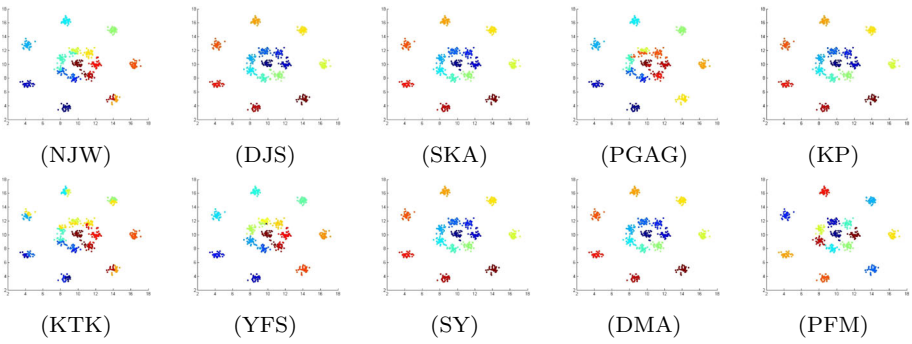


**Fig. 6** Results due to ten algorithms on the "R15" test problem

*Hyper-parameter settings*

Here, we provide parameters used in our proposed algorithm. We set the tolerance for the stopping criterion as $\epsilon = 10^{-6}$, penalization parameter as $\rho = 10^{-5}$, and $\alpha = 1.1$. The correction step $\tau$ must be chosen carefully to reduce the number of iterations of the algorithm. We set this to be 0.01, 0.12, and 0.3 for Tables 2, 3, 4, 5 and 6, respectively; these values have been decided based on our experimentations. Note that, for exiting the inner loop of PFM, we should use the relative error as $||V^l - V^{l-1}||_F \leq \epsilon ||V^l||_F$. Since $||V^l||_F$ is a fixed number

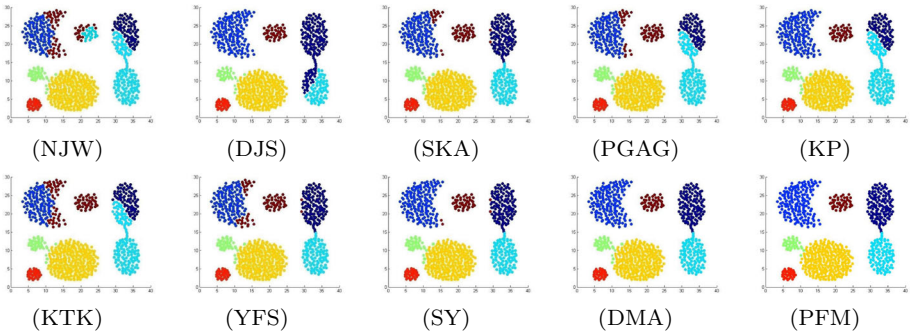


**Fig. 7** Results due to ten algorithms on the "Aggregation" test problem

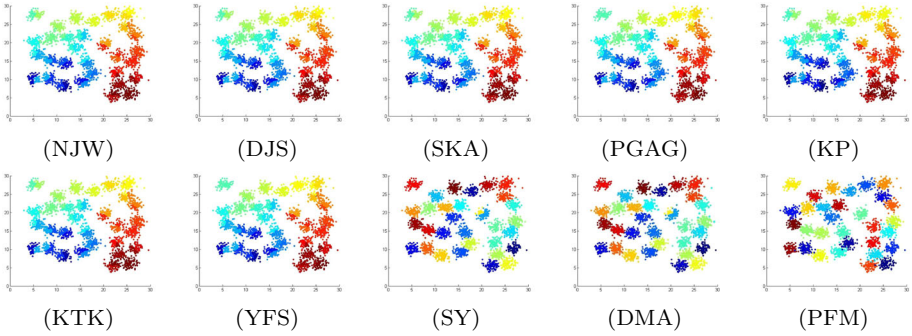


**Fig. 8** Results due to ten algorithms on the "D31" test problem

due to orthogonality of $V^l$ (it is approximately $\sqrt{k}$, where $k$ is the number of clusters), the absolute and relative errors are almost equivalent, and thus we made use of absolute error.

### 4.1 Artificial data-sets

Here, we reported the numerical results obtained for the ten algorithms on two groups (four test problems of [41] and four test problems of [42]) of the hard benchmark clustering problems, as shown in Tables 2 and 3. For each entry of the tables, we reported the misclassification rate (the ratio of incorrect labelings to the total number of objects) for the corresponding algorithm. Also, to evaluate and compare the performance of the clustering methods, we reported the Rand index (a measure of similarity between the data clusters which has a value between 0 and 1, and the higher the value, the better the clustering) in the last row of the tables. In Tables 2 and 3, for each table, we reported this value as the average of the Rand indices over all the test problems obtained by each algorithm. From the obtained results it is obvious that PFM outperforms the other algorithms; PFM has the best or the second-best misclassification error rates and also the highest Rand index over all the test problems. Figures 1, 2, 3, 4, 5, 6, 7 and 8 illustrate the performance of ten algorithms corresponding to Tables 2 and 3. By observing the 2-dimensional shape of the test problems (depicted in Figs. 1, 2, 3, 4, 5, 6, 7 and 8), it is clear that PFM performs well in constructing the expected clusters.

**Table 4** The average misclassification rates and Rand indices for 100 randomly generated test problems: $n = 1000, k = 5$

| Algorithms | MR Mean | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Parameters | NJW | DJS | PGAG | KP | SKA | KTK | YFS | SY | DMA | PFM |
| $\mu_w = 0.01, \mu_t = 0.9$ | 0.208 | 0.028 | 0.083 | 0.01 | 0.191 | 0.09 | 0.07 | 0.007 | 0.0098 | **0.005** |
| $\mu_w = 0.01, \mu_t = 0.01$ | 0.443 | 0.164 | 0.363 | 0.094 | 0.345 | 0.37 | 0.312 | 0.02 | 0.009 | **0.004** |
| $\mu_w = 0.02, \mu_t = 0.02$ | 0.088 | 0.03 | 0.177 | 0.182 | 0.061 | 0.2 | 0.134 | 0.06 | 0.008 | **0.004** |
| $\mu_w = 0.03, \mu_t = 0.03$ | **0** | **0** | 0.221 | 0.001 | **0** | 0.241 | 0.18 | 0.012 | **0** | **0** |
| $\mu_w = 0.04, \mu_t = 0.04$ | **0** | **0** | 0.002 | 0.002 | **0** | 0.005 | 0.003 | **0** | **0** | **0** |
| $\mu_w = 0.05, \mu_t = 0.05$ | 0.185 | 0.01 | 0.098 | 0.191 | 0.107 | 0.09 | 0.042 | 0.09 | 0.007 | **0** |
| $\mu_w = 0.01, \mu_t = 0.1$ | 0.183 | **0.01** | 0.082 | 0.09 | 0.106 | 0.084 | 0.062 | 0.016 | 0.012 | **0.01** |
| $\mu_w = 0.02, \mu_t = 0.1$ | 0.186 | 0.05 | 0.265 | **0** | 0.118 | 0.22 | 0.162 | 0.014 | **0** | **0** |
| $\mu_w = 0.03, \mu_t = 0.1$ | 0.184 | 0.085 | 0.098 | 0.091 | 0.107 | 0.092 | 0.064 | 0.02 | **0.01** | **0.01** |
| $\mu_w = 0.04, \mu_t = 0.1$ | **0** | **0** | 0.244 | **0** | **0** | 0.252 | 0.2 | 0.001 | **0** | **0** |
| $\mu_w = 0.05, \mu_t = 0.1$ | 0.16 | 0.038 | 0.165 | 0.15 | 0.091 | 0.18 | 0.132 | **0.005** | 0,006 | 0,006 |
| $\mu_w = 0.1, \mu_t = 0.01$ | 0.26 | 0.021 | **0** | **0** | 0.167 | 0.03 | 0.03 | 0.002 | **0** | **0** |
| $\mu_w = 0.1, \mu_t = 0.02$ | 0.03 | 0.025 | 0.013 | 0.013 | 0.01 | 0.016 | 0.01 | 0.001 | 0.003 | **0** |
| $\mu_w = 0.1, \mu_t = 0.03$ | 0.126 | 0.016 | 0.134 | 0.144 | 0.08 | 0.1 | 0.08 | 0.01 | 0.02 | **0.01** |
| $\mu_w = 0.1, \mu_t = 0.04$ | 0.203 | 0.029 | 0.279 | 0.034 | 0.221 | 0.28 | 0.22 | 0.012 | **0** | **0** |
| $\mu_w = 0.1, \mu_t = 0.05$ | 0.21 | 0.106 | 0.183 | 0.133 | 0.234 | 0.2 | 0.164 | 0.019 | 0.016 | **0.011** |
| $\mu_w = 0.02, \mu_t = 0.03$ | 0.184 | **0.013** | 0.09 | 0.1 | 0.114 | 0.1 | 0.04 | 0.05 | 0.014 | 0.015 |
| $\mu_w = 0.02, \mu_t = 0.04$ | 0.19 | 0.076 | 0.087 | 0.09 | 0.128 | 0.092 | 0.078 | 0.046 | **0.01** | **0.01** |
| $\mu_w = 0.02, \mu_t = 0.05$ | 0.009 | **0** | 0.02 | 0.081 | 0.009 | 0.04 | 0.014 | **0** | **0** | **0** |
| $\mu_w = 0.03, \mu_t = 0.02$ | 0.14 | 0.04 | 0.15 | 0.145 | 0.091 | 0.15 | 0.128 | 0.005 | 0.003 | **0** |
| $\mu_w = 0.03, \mu_t = 0.04$ | 0.16 | 0.1 | 0.163 | 0.132 | 0.102 | 0.184 | 0.132 | 0.062 | 0.017 | **0.012** |
| $\mu_w = 0.04, \mu_t = 0.02$ | 0.4 | 0.16 | 0.32 | 0.094 | 0.32 | 0.24 | 0.14 | 0.12 | 0.018 | **0.012** |
| $\mu_w = 0.04, \mu_t = 0.03$ | 0.08 | 0.03 | 0.2 | 0.24 | 0.051 | 0.26 | 0.18 | 0.09 | 0.007 | **0** |
| $\mu_w = 0.04, \mu_t = 0.05$ | 0.015 | 0.01 | 0.09 | 0.19 | 0.01 | 0.14 | 0.06 | 0.005 | 0.009 | **0** |
| $\mu_w = 0.05, \mu_t = 0.02$ | 0.2 | 0.1 | 0.08 | **0.01** | 0.12 | 0.094 | 0.064 | 0.05 | 0.088 | 0.08 |
| $\mu_w = 0.05, \mu_t = 0.03$ | 0.18 | 0.02 | 0.1 | 0.18 | 0.06 | 0.12 | 0.05 | 0.028 | **0.015** | 0.02 |
| **Rand-Index** | 0.8452 | 0.9553 | 0.8574 | 0.9078 | 0.8906 | 0.8258 | 0.8688 | 0.9579 | 0.9888 | **0.9902** |
| Time (sec.) | 4.75 | 3.42 | 7.51 | 9.87 | 4.0131 | 8.34 | 7.128 | 5.12 | 5.94 | 4.1 |

### 4.2 Random graph generation and testing

Here, we investigate the performance of clustering algorithms and compare the obtained results on some randomly generated test problems. For a comparable performance evaluation, several sample random graphs were generated using the benchmark generator of Lancichinetti and Fortunato (2009) with parameters $\mu_t$ and $\mu_w$. Table 4 shows the misclassification rate for the algorithms on constructed randomly generated test problems using several values of the parameters. We note that all algorithms were executed once for these benchmarks and "MR Mean" is the average of the total error for each test problem. The last two rows of the table respectively give Rand indices of the clustering algorithms and the average running times of these algorithms on all the data-sets. For more statistical analysis, we utilized the Dolan and Moré (2002) performance profiles to compare the performance

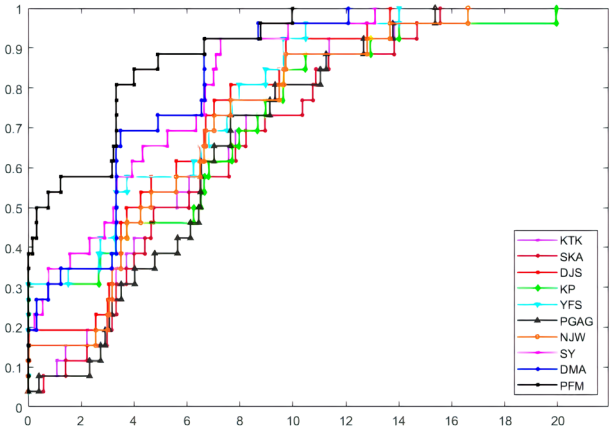


**Fig. 9** The Dolan-Moré performance profiles comparing the misclassification rates by NJW, DJS, KP, PGAG, SKA, KTK, YFS, SY, DMA and PFM

**Table 5** The average misclassification rates and Rand indices for 100 randomly generated test problems: $n = 10000$, $\mu_w = 0.1$, $\mu_t = 0.01, 0.02, 0.03, 0.04, 0.05$.

| Algorithm | MR Mean | | | | | | |
|---|---|---|---|---|---|---|---|
| | $\mu_t = 0.01$ | $\mu_t = 0.02$ | $\mu_t = 0.03$ | $\mu_t = 0.04$ | $\mu_t = 0.05$ | **Rand-Index** | Time (sec.) |
| NJW | 0.36 | 0.23 | 0.24 | 0.33 | 0.221 | 0.7238 | 213.71 |
| DJS | 0.221 | 0.195 | 0.216 | 0.29 | ]0.2 | 0.7756 | 158.8 |
| PGAG | 0.28 | 0.193 | 0.234 | **0.219** | 0.211 | 0.7726 | 337.95 |
| KP | 0.3 | 0.193 | 0.224 | 0.34 | 0.22 | 0.7446 | 444.15 |
| SKA | 0.26 | 0.196 | 0.217 | 0.292 | 0.202 | 0.7666 | 180.58 |
| KTK | 0.3 | 0.2 | 0.224 | 0.3 | 0.24 | 0.7566 | 583.18 |
| YFS | 0.24 | 0.17 | 0.2 | 0.25 | 0.196 | 0.7828 | 481.55 |
| SY | 0.19 | 0.12 | 0.22 | 0.24 | 0.2 | 0.7886 | 324.75 |
| DMA | 0.19 | 0.145 | 0.2 | 0.29 | 0.18 | 0.7990 | 226.31 |
| PFM | **0.15** | **0.1** | **0.16** | 0.24 | **0.12** | **0.8202** | 190.45 |

of the clustering algorithms. Fig. 9 shows a comparison of the obtained misclassification rates for the considered algorithms. We constructed this profile using all the test problems corresponding to Tables 4.

Tables 5 and 6 provide the average misclassification rates and Rand indices of the related algorithms on 100 randomly generated test problems, with parameters $\mu_w = 0.1$ and $\mu_t = 0.01, 0.02, 0.03, 0.04, 0.05$, respectively, for 10000 and 20000 points. In these tables, we reported the Rand index as the average of the Rand indices of all the test problems.

## 5 Concluding remarks

We proposed an alternative formulation of the ONMF problem by converting the orthogonality constraints into convex constraints. Using a penalty function, we proposed a proper optimization method to solve this problem. We first used a gradient-based opti-

**Table 6** The average misclassification rates and Rand indices for 100 randomly generated test problems: $n = 20000$, $\mu_w = 0.1$, $\mu_t = 0.01, 0.02, 0.03, 0.04, 0.05$.

| Algorithm | MR Mean | | | | | | |
|---|---|---|---|---|---|---|---|
| | $\mu_t = 0.01$ | $\mu_t = 0.02$ | $\mu_t = 0.03$ | $\mu_t = 0.04$ | $\mu_t = 0.05$ | **Rand-Index** | Time (sec.) |
| NJW | 0.45 | 0.39 | 0.36 | 0.43 | 0.441 | 0.5858 | 601.13 |
| DJS | 0.321 | 0.342 | 0.316 | 0.34 | 0.4 | 0.6562 | 494.31 |
| PGAG | 0.38 | **0.293** | 0.334 | 0.35 | 0.41 | 0.6466 | 1117.74 |
| KP | 0.4 | 0.35 | 0.344 | 0.38 | 0.43 | 0.6192 | 1420.08 |
| SKA | 0.351 | 0.362 | 0.331 | 0.336 | 0.4012 | 0.6440 | 531.12 |
| KTK | 0.392 | 0.323 | 0.324 | 0.28 | 0.42 | 0.6336 | 928.73 |
| YFS | 0.34 | 0.32 | 0.312 | 0.34 | 0.382 | 0.6646 | 682.48 |
| SY | 0.33 | 0.3 | **0.28** | 0.3 | 0.365 | 0.6801 | 748.16 |
| DMA | 0.323 | 0.342 | 0.32 | 0.27 | 0.365 | 0.6760 | 653.24 |
| PFM | **0.28** | 0.3 | 0.3 | **0.251** | **0.347** | **0.6908** | 540.16 |

mization algorithm and then applied a post-processing technique to extract a solution to the clustering problem. Utilizing different test problems, we considered the performance of our proposed algorithm in comparison with other available clustering algorithms, namely, Ng-Jordan-Weiss (NJW), Daneshgar-Javadi-ShariyatRazavi (DJS), Standard $k$-means (SKA), Pompili-Gillis-Absil-Glineur (PGAG), Kim-Park (KP), Kimura-Tanaka-Kudo (KTK), Yang-Fu-Sidiropoulos (YFS), Sinaga-Yang (SY) and Dehghanpour-Mahdavi-Amiri (DMA). Numerical results confirmed the practicality of our formulation and showed the capability of our proposed approach for constructing the expected clustering.

We compared our proposed algorithm with nine related clustering algorithms on hard synthetic data sets and some randomly generated test problems. For a proper statistical analysis, we utilized the Dolan-Moré performance profiles to compare the obtained misclassification rate errors. Numerical results confirmed our proposed method to be successful in clustering; PFM had the best or the second-best misclassification rate and also the highest Rand index among all the compared methods.

**Acknowledgements** The authors thank the Research Council of Sharif University of Technology for supporting this work.

## Decarations

**Conflict of interest** Authors declare that they have no conflict of interest.

**Ethical approval** This article does not contain any studies with human participants or animals performed by any of the authors.

## References

Arthur, D., Sergi, V. (2007) K-means++: The Advantages of Careful Seeding. SODA ' 07: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, 1027-1035 .

Banker, R. D., Chang, H., & Zheng, Z. (2017). On the use of super-efficiency procedures for ranking efficient units and identifying outliers. *Ann Oper Res, 250*(1), 21–35.

Bauckhage, C. K-means clustering is matrix factorization. arXiv preprint arXiv:1512.07548, (2015).

Bertsekas, D. P. (1999). *Nonlinear Programming* (2nd ed.). Belmont, Massachusetts: Athena Scientific.
Bolte, J., Sabach, S., & Teboulle, M. (2014). Proximal alternating linearized minimization for non-convex and non-smooth problems. *Math Program, 146*, 459–494.
Daneshgar, A., Javadi, R., & Razavi, S. S. (2013). Clustering and outlier detection using isoperimetric number of trees. *Pattern Recognition, 46*(12), 3371–3382.
Dehghanpour-Sahron, J., & Mahdavi-Amiri, N. (2020) A competitive optimization approach for data clustering and orthogonal non-negative matrix factorization. 4OR, 27 pages, , https://doi.org/10.1007/s10288-020-00445-y.
Del Buono N. (2009). A penalty function for computing orthogonal non-negative matrix factorizations. (pp. 1001–1005)
Ding, C., Li, T., Peng, W., & Park, H. (2006). Orthogonal nonnegative matrix t-factorizations for clustering. (pp. 126–135)
Dinler, D., Tural, M. K., & Ozdemirel, N. E. (2020). Centroid based Tree-Structured Data Clustering Using Vertex/Edge Overlap and Graph Edit Distance. *Ann Oper Res, 289*(1), 85–122.
Dolan E D, & Moré J J (2002). Benchmarking optimization software with performance profiles. *Mathematical Programming, 91*(2), 201–213.
Duan, L., Xu, L., Liu, Y., et al. (2009). Cluster-based outlier detection. Ann. *Oper Res, 168*, 151–168.
Facchinei, F., & Pang, J. S. (2007). *Finite-dimensional variational inequalities and complementarity problems*. Springer Science and Business Media.
Fard, M. M., Thonet, T., & Gaussier, E. (2020). Deep k-means: Jointly clustering with k-means and learning representations. *Pattern Recognition Letters, 138*, 185–192.
Fränti, P., & Sieranoja, S. (2018). K-means properties on six clustering benchmark datasets. *Applied Intelligence, 48*(12), 4743–4759.
He, P., Xu, X., Ding, J., & Fan, B. (2020). Low-rank nonnegative matrix factorization on Stiefel manifold. *Information Sciences, 514*, 131–148.
Jiang, B., & Dai, Y. H. (2015). A framework of constraint preserving update schemes for optimization on Stiefel manifold. *Mathematical Programming, 153*(2), 535–575.
Kim, J., & Park, H. (2011). Fast non-negative matrix factorization: An active-set-like method and comparisons. *SIAM Journal on Scientific Computing, 33*(6), 3261–3281.
Kimura, K., Tanaka, Y., & Kudo, M. (2015). A fast hierarchical alternating least squares algorithm for orthogonal nonnegative matrix factorization.
Kimura, K., Kudo, M., & Tanaka, Y. (2016). A column-wise update algorithm for nonnegative matrix factorization in Bregman divergence with an orthogonal constraint. *Machine learning, 103*(2), 285–306.
Lancichinetti, A., & Fortunato, S. (2009). Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical Review E, 80*, 016118.
Huang, S., Kang, Z., Xu, Z., & Liu, Q. (2021). Robust deep k-means: An effective and simple method for data clustering. *Pattern Recognition, 117*, 107996.
Lawrence, H., & Phipps, A. (1985). Comparing partitions. *Journal of Classification, 2*(1), 193–218.
McQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Computer and Chemistry, 4*, 257–272.
Li, W., Li, J., Liu, X., & Dong, L. (2020). Two fast vector-wise update algorithms for orthogonal nonnegative matrix factorization with sparsity constraint. *Journal of Computational and Applied Mathematics, 375*, 112785.
Moreno, S., Pereira, J., & Yushimito, W. (2020). A hybrid K-means and integer programming method for commercial territory design: a case study in meat distribution. *Ann Oper Res, 286*(1), 87–117.
Ng, A. Y., Jordan, M. I., & Weiss, Y (2002) On spectral clustering: analysis and an algorithm. In: Advances in Neural Information Processing Systems, 849-856 .
Paatero, P., & Tapper, U. (1994). Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics, 5*(2), 111–126.
Pan, J., & Ng, M. K. (2018). Orthogonal nonnegative matrix factorization by sparsity and nuclear norm optimization. *SIAM Journal on Matrix Analysis and Applications, 39*(2), 856–875.
Peng, J., & Wei, Y. (2007). Approximating k-means-type clustering via semidefinite programming. *SIAM Journal on Optimization, 18*(1), 186–205.
Peng, S., Ser, W., Chen, B., & Lin, Z. (2020). Robust orthogonal nonnegative matrix tri-factorization for data representation. *Knowledge-Based Systems, 201,* 106054.
Pock, T., & SabachS. (2016). Inertial proximal alternating linearized minimization (iPALM) for nonconvex and nonsmooth problems. *SIAM Journal on Imaging Sciences, 9*(4), 1756–1787.
Pompili, F., Gillis, N., Absil, P. A., & Glineur, F. (2014). Two algorithms for orthogonal non-negative matrix factorization with application to clustering. *Neurocomputing, 141*, 15–25.

Qin, Z., Wan, T., & Zhao, H. (2017). Hybrid clustering of data and vague concepts based on labels semantics. *Ann Oper Res, 256*(2), 393–416.

Shefi, R., & Teboulle, M. (2016). On the rate of convergence of the proximal alternating linearized minimization algorithm for convex problems. *EURO J Comput Optim, 4*, 27–46.

Sinaga, K. P., & Yang, M. S. (2020). Unsupervised K-means clustering algorithm. IEEE. *Access, 8*, 80716–80727.

Tosyali, A., Kim, J., Choi, J., et al. (2020). New node anomaly detection algorithm based on nonnegative matrix factorization for directed citation networks. *Ann Oper Res, 288*, 457–474.

Xia, S., Peng, D., Meng, D., Zhang, C., Wang, G., Giem, E., & Chen, Z. (2020). A fast adaptive k-means with no bounds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. https://doi.org/10.1109/TPAMI.2020.3008694

Yang, B., Fu, X., & Sidiropoulos, N. D. (2017). Learning from hidden traits: Joint factor analysis and latent clustering. *IEEE Transactions on Signal Processing, 65*(1), 256–269.

Yu, S. S., Chu, S. W., Wang, C. M., Chan, Y. K., & Chang, T. C. (2018). Two improved k-means algorithms. *Applied Soft Computing, 68*, 747–755.

http://www.vision.caltech.edu/lihi/Demos/SelfTuningClustering.html.

http://cs.joensuu.fi/sipu/datasets/.