A SURVEY OF CLUSTERING METHODS VIA OPTIMIZATION METHODOLOGY

XIAOTIAN LI[1], LINJU CAI[1], JINGCHAO LI[1], CARISA KWOK WAI YU[2], YAOHUA HU[1,*]

[1]*Shenzhen Key Laboratory of Advanced Machine Learning and Applications, College of Mathematics and Statistics, Shenzhen University, Shenzhen 518060, P. R. China*

[2]*Department of Mathematics, Statistics and Insurance, The Hang Seng University of Hong Kong, Shatin, Hong Kong*

Abstract. Clustering is one of fundamental tasks in unsupervised learning and plays a very important role in various application areas. This paper aims to present a survey of five types of clustering methods in the perspective of optimization methodology, including center-based methods, convex clustering, spectral clustering, subspace clustering, and optimal transport based clustering. The connection between optimization methodology and clustering algorithms is not only helpful to advance the understanding of the principle and theory of existing clustering algorithms, but also useful to inspire new ideas of efficient clustering algorithms. Preliminary numerical experiments of various clustering algorithms for datasets of various shapes are provided to show the preference and specificity of each algorithm.

Keywords. Machine learning; Clustering methods; Optimization; Numerical algorithms; Optimal transport

1. Introduction

In the era of big data, machine learning plays a very important role in various application areas, such as pattern recognition, image science, bioinformatics, data compression and computer graphics. In general, machine learning approaches are divided into supervised learning and unsupervised learning; the former one aims to infer a function from labeled training data set (e.g., classification), and the latter one aims to learn undetected patterns in a data set with no pre-existing labels (e.g., clustering). It was predicted by professor Yann LeCun that "Next artificial intelligence revolution is unsupervised learning".

1.1. Clustering.

Clustering is one of fundamental tasks in unsupervised learning. It aims to find a partition to segment the unlabeled data into several different groups (called clusters) such that the objects in the same group have a higher degree of similarity to each other than to those in different groups. Motivated by its vast applications, tremendous efforts have been devoted to the development of clustering algorithms; see [40, 66] and references therein. Originating from the classical k-means algorithm proposed by MacQueen [36] in 1967, many exclusive and efficient algorithms have been proposed and developed for clustering. Existing clustering algorithms can mainly be divided into several categories: hierarchical clustering [10], center-based methods [18, 25, 56], density-based methods [28, 48], grid-based methods [35, 50], spectral-based methods [41, 51, 62], model-based methods [17, 69] and bayesian and nonparametric methods [5]. Moreover, in the last two decades, with effective acquisition of high-dimensional

*Corresponding author.

E-mail addresses: 2017191135@email.szu.edu.cn (X. Li), 515698921@qq.com (L. Cai), jingchaoli@szu.edu.cn (J. Li), carisayu@hsu.edu.hk (C. K. W. Yu), mayhhu@szu.edu.cn (Y. Hu).

data (in face images, videos or web pages), subspace clustering [15, 52, 60] and multi-view clustering [19, 67] have been developed and widely applied.

1.2. **Optimization.** Mathematical optimization is a fundamental tool for solving practical problems in many disciplines from economics and engineering to artificial intelligence and data science. It aims to select the best element (with regard to certain criterion/objective) from constraint set of feasible alternatives. Convex optimization plays a key role in mathematical optimization because it has nice theoretical properties and fast numerical algorithms; however it maybe too restrictive for certain practical problems. In contrast to convex optimization, nonconvex optimization usually provides a much more accurate representation of reality, but it is inconvenient to design efficient and globally convergent optimization algorithms.

In applications, a wide class of problems usually have certain special structures, and designing numerical algorithms due to the special structures has become an active topic in mathematical optimization. Exploiting the special structures and inspired by the ideas in numerical optimization methodology, a large number of exclusive and efficient optimization algorithms have been developed and applied for structured optimization problems and large-scale applications; see [7, 42, 47] and references therein.

1.3. **This paper.** In this paper, we aim to provide a survey of clustering algorithms in the perspective of mathematical optimization; particularly, we build up a connection between optimization models and clustering algorithms. This connection is not only helpful to advance the understanding of the principle and theory of existing clustering algorithms, but also useful to inspire new ideas of efficient clustering algorithms. For example, the center-based clustering algorithm can be equivalently converted to an optimization model that minimizes the total distance from each point to the corresponding cluster. Convex clustering algorithms [23] are based on a convex optimization problem in which the objective function is a sum-of-norms. In subspace clustering algorithms, the similarity matrix is obtained by solving an optimization problem under the self-expressiveness property. For clustering of distribution data, optimal transport (OT) based clustering algorithms aim to minimize the total Wasserstein distance (in place of Euclidean distance in k-means) from each point to the corresponding Wasserstein barycenter. Optimization models of clustering (mentioned above or discussed in the sequel) can be efficiently solved by optimization algorithms including alternating direction method of multipliers (ADMM) [8], block coordinate descent (BCD) [63], majorization-minimization (MM) [29], and proximal gradient method (PGM) [24]. For the details and principles of these optimization algorithms, one can refer to Appendix A.

In this paper, we review five types of clustering methods in the perspective of optimization methodology: center-based clustering, convex clustering, spectral clustering, subspace clustering, and OT based clustering. The remainder of this paper is organized as follows. In sections 2-6, we review the mathematical ideas and optimization models of five types of clustering algorithms, respectively. Preliminary numerical experiments of clustering algorithms are conducted for datasets of various shapes in section 7.

1.4. **Notations.** We consider the d-dimensional Euclidean space $\mathbb{R}^d$ with inner product $\langle\cdot,\cdot\rangle$ and its associated norm $\|\cdot\|$. Vectors and matrices are represented in bold lowercase letters and uppercase letters, respectively. $\mathbf{X} \in \mathbb{R}^{d\times n}$ denotes the data matrix, where n is the number of samples and d is the dimension of each sample; each sample $\mathbf{x}_i \in \mathbb{R}^d$ is a column vector.

$\mathbf{A} := (\mathbf{a}_1, \ldots, \mathbf{a}_k) \in \mathbb{R}^{d\times k}$ denotes the matrix of k cluster centers. $A_i \subseteq \{1, \ldots, n\}$ represents the index set of samples that are allocated to i-th cluster center $\mathbf{a}_i$. $\mathbf{z} \in \mathbb{R}^n$ denotes the cluster assignment given by algorithms. The ℓ_p norm ($p > 0$) of $\mathbf{x} \in \mathbb{R}^d$ and the Frobenius norm of $\mathbf{X} \in \mathbb{R}^{d\times n}$ are defined by $\|\mathbf{x}\|_p := (\sum_{i=1}^d |x_i|^p)^{\frac{1}{p}}$ and $\|\mathbf{X}\|_F := \sqrt{\sum_{i=1}^d \sum_{j=1}^n x_{ij}^2}$, respectively.

2. Center-based clustering

Center-based clustering is a type of most classical clustering algorithms, in which it is assumed that each cluster has a center. This assumption is rational for the spherical clusters, while might not hold for the general manifold.

2.1. k-means and center-based optimization framework.

The principle of center-based clustering is to find a set of cluster centers such that the total Euclidean distance of the samples from their nearest centers is minimal. For each sample $\mathbf{x}_i$, the distance from its nearest cluster center is obtained among the cluster centers $\{\mathbf{a}_j\}$:

$$\min_{1\le j\le k} \|\mathbf{x}_i - \mathbf{a}_j\|^2.$$

Hence the center-based clustering can be cast into the following optimization problem

$$\min_{\mathbf{A}} F(\mathbf{a}_1, \cdots, \mathbf{a}_k) := \sum_{i=1}^n \min_{1\le j\le k} \|\mathbf{x}_i - \mathbf{a}_j\|^2. \tag{2.1}$$

Problem (2.1) is non-smooth and non-convex because of the presence of minimizing function, and thus, is in general NP-hard to solve; see [37].

The classic k-means algorithm is one of the most well-known algorithms for center-based clustering, which was introduced by MacQueen [36] and Lloyd [34]. The main idea of the k-means is to approximately solve problem (2.1) through two alternative steps: (i) assign samples to their nearest centers, and (ii) update centers by samples involved. Hence, the k-means initially generates a series of random cluster centers, and then allocates the sample points to their nearest centers and update centers by the arithmetic mean of sample points contained in each cluster, alternatively, and finally arrives at the k cluster centers and the associated partition of data. The classic k-means algorithm is formally stated as follows.

Algorithm 1: k-means

1 **Input:** data matrix $\mathbf{X}$, number of clusters k.
2 **Output:** center matrix $\mathbf{A}$ and allocation label $\mathbf{z}$.
3 Randomly initialize a center matrix $\mathbf{A}$;
4 **Assign points**: for each point $\mathbf{x}_i$, calculate its distance from each center and assign it to the nearest one:

$$z_i := \arg\min_j \|\mathbf{x}_i - \mathbf{a}_j\|^2;$$

5 **Recalculate centers**: for each cluster, update the arithmetic mean of sample points involved in it as the new cluster center of them:

$$\mathbf{a}_j := \frac{1}{|A_j|}\sum_{i\in A_j}\mathbf{x}_i;$$

Repeat steps 4 and 5 until convergence.

In despite of its wide applications, the k-means algorithm suffers from several known drawbacks [65]: (i) the clustering assignment is highly sensitive to the initialization of cluster centers; (ii) the k-means performs well on spherical data with equal radii, but instable for other types of data. For the spherical cluster with different radii (i.e., they have different variances), it is natural to use Mahalanobis distance that takes the covariance matrix into account. More generally, using a general distance-like function $d(\cdot,\cdot)$ (see [56, Definition 1] for the definition) in place of the squared Euclidean distance in (2.1), the center-based clustering can be reformulated as

$$\min_{\mathbf{A}} F(\mathbf{a}_1,\cdots,\mathbf{a}_k) := \sum_{i=1}^{n}\min_{1\le j\le k} d(\mathbf{x}_i,\mathbf{a}_j). \tag{2.2}$$

Problem (2.2) inherits the non-smooth and non-convex property of (2.1) due to the presence of minimizing function. To deal with the non-smoothness of problem (2.2), Teboulle [56] introduced an idea of exact smooth (ES) mechanism. In particular, let $\Delta\subseteq\mathbb{R}^k$ be a unit simplex, i.e.,

$$\Delta := \{\mathbf{y}\in\mathbb{R}^k : \sum_{j=1}^{k} y_j = 1,\ \mathbf{y}\ge 0\}.$$

Then the component function of (2.2) can be re-written as

$$\min_{1\le j\le k} d(\mathbf{x}_i,\mathbf{a}_j) = \min\{\langle \mathbf{w}^i, (d(\mathbf{x}_i,\mathbf{a}_1),\cdots,d(\mathbf{x}_i,\mathbf{a}_k))^T\rangle : \mathbf{w}^i\in\Delta\}$$

for each $i=1,\cdots,n$. Hence problem (2.2) is equivalent to the following smooth one

$$\begin{array}{ll}\min\limits_{\mathbf{A},\mathbf{W}} & F(\mathbf{a}_1,\cdots,\mathbf{a}_k) := \sum_{i=1}^{n}\sum_{j=1}^{k} w_j^i d(\mathbf{x}_i,\mathbf{a}_j)\\ \text{s.t.} & \sum_{j=1}^{k} w_j^i = 1,\ \mathbf{w}^i\ge 0.\end{array} \tag{2.3}$$

In (2.3), $\mathbf{A}$ and $\mathbf{W}$ denote the matrices of cluster centers and assignment weights, respectively. In particular, if w_j^i only takes 0 or 1, then sample i is assigned to cluster j whenever $w_j^i=1$; otherwise, w_j^i denotes the probability of sample i belongs to cluster j and we can proceed the clustering assignment with the maximal probability. The former one is called hard clustering, and the latter one is called soft clustering. Teboulle [56] proposed a hard clustering algorithm with distance-like functions (HCD) to solve the ES problem (2.3).

Besides the clustering assignment, the minimizing term in (2.1) and (2.2) also contributes to take arithmetic mean in recalculating cluster centers. Given a general averaging scheme $M(\cdot)$, the center-based clustering (2.2) can be represented as

$$\min_{\mathbf{A}} F(\mathbf{a}_1,\cdots,\mathbf{a}_k) := \sum_{i=1}^{n} M(d(\mathbf{x}_i,\mathbf{a}_1),d(\mathbf{x}_i,\mathbf{a}_2),\cdots,d(\mathbf{x}_i,\mathbf{a}_k)). \tag{2.4}$$

2.2. **Distance measure.** Different types of distance measures are appropriate for clustering algorithms on different types of data. For example, the Euclidean distance is used in k-means, which performs well on spherical data with equal radii. For other types of data, several types of distance measures are preferred in the corresponding clustering algorithms.

2.2.1. *Elliptic norm.* Given a positive definite matrix $\mathbf{Q} \in \mathbb{R}^{d\times d}$, the elliptic norm is defined by

$$\|\mathbf{v}\|_{\mathbf{Q}} := \langle \mathbf{v}, \mathbf{Q}\mathbf{v}\rangle^{\frac{1}{2}} \quad \text{for each } \mathbf{v} \in \mathbb{R}^d. \tag{2.5}$$

Example 2.1. Euclidean distance and Mahalanobis distance are special cases of elliptic norm.

(i) When $\mathbf{Q} = \mathbf{I}$ (the identity matrix), the elliptic norm (2.5) is reduced to the Euclidean distance.
(ii) When $\mathbf{Q} = \Sigma^{-1}$ (Σ is the covariance matrix of the data involved), the elliptic norm (2.5) is reduced to the Mahalanobis distance.

Remark 2.1. The probabilistic distance clustering (d-clustering), proposed by Ben-Israel and Iyigun [6], adopts model (2.3) with the principle that the probability of each sample point belonging to a certain cluster is inversely proportional to the distance from the center. When the elliptic norm is used in d-clustering, problem (2.3) has closed-form solutions (see [6, Corollaries 1-3]).

2.2.2. *Bregman distance.* Bregman distance has been extensively applied in information science and machine learning. Let $\phi : \mathbb{R}^d \to \mathbb{R}\cup\{+\infty\}$ be a Legendre function satisfying the following conditions:

(a) ϕ is proper, lower semicoutinuous, and convex with $\operatorname{dom}\phi \subseteq \operatorname{cl} C$ and $\operatorname{dom}\nabla\varphi = C$;
(b) ϕ is strictly convex and continuous on $\operatorname{dom}\phi$, and continuously differentiable on C.

The Bregman distance based on ϕ is denoted by $d_\phi : \mathbb{R}^d \times \mathbb{R}^d \to [0,+\infty]$ and defined by

$$d_\phi(\mathbf{x},\mathbf{y}) := \begin{cases} \phi(\mathbf{x}) - \phi(\mathbf{y}) - \langle \mathbf{x}-\mathbf{y}, \nabla\phi(\mathbf{y})\rangle, & \text{for } \mathbf{x}\in\mathbb{R}^d, \mathbf{y}\in C, \\ +\infty, & \text{otherwise.} \end{cases} \tag{2.6}$$

Separable Bregman distances are the most commonly used in the literature. In detail, when $C := \prod_{i=1}^n C_i$ is of separable structure, the Legendre function is written as the summation of one-dimensional functions

$$\phi(\mathbf{x}) := \sum_{i=1}^{n} \varphi(x_i),$$

where $\varphi : \mathbb{R} \to \mathbb{R}_+ \cup \{+\infty\}$ satisfies conditions (a) and (b) and is twice differentiable on C_i. By the separable structure and (2.6), one has $d_\phi(x,y) = \sum_{i=1}^n d_\varphi(x_i,y_i)$.

Several popular Bregman kernels are described as follows, in which the squared Euclidean distance, the Kullback-Leibler divergence and the Itakura-Saito divergence are Bregman distances generated by the energy, the Boltzmann-Shannon entropy and the Burg entropy, respectively; see Examples 2.2-2.4. For more examples of Bregman distance, one can refer to [2, Table 1].

Example 2.2. Let $\varphi(t) := t^2$ be the energy. The Bregman distance can be calculated by (2.6) as

$$d_\phi(\mathbf{x},\mathbf{y}) = \|\mathbf{x}\|^2 - \|\mathbf{y}\|^2 - \langle \mathbf{x}-\mathbf{y}, \nabla\phi(\mathbf{y})\rangle = \|\mathbf{x}-\mathbf{y}\|^2,$$

which is the squared Euclidean distance.

Example 2.3. Let $\varphi(t) := t\log t$ be the Boltzmann-Shannon entropy. The Bregman distance can be calculated by (2.6) as

$$d_\phi(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n} x_i \log\frac{x_i}{y_i} - x_i + y_i = d_{\mathrm{KL}}(\mathbf{x},\mathbf{y}),$$

which is the Kullback-Leibler divergence.

Example 2.4. Let $\varphi(t) := -\log t$ be the Burg entropy. The Bregman distance can be calculated by (2.6) as

$$d_\phi(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n} \log\frac{y_i}{x_i} + \frac{x_i}{y_i} - 1 = d_{\mathrm{IS}}(\mathbf{x},\mathbf{y}),$$

which is the Itakura-Saito divergence.

Remark 2.2. The k-means algorithm is based on the squared Euclidean distance. The information theoretic clustering algorithm [14] is based on the Kullback-Leibler divergence. The Linde-Buzo-Gray (LBG) algorithm [31] is based on Itakura-Saito divergence. The hard clustering algorithm based on Bregman distance and distance-like function were proposed by [2] and [56], respectively.

2.2.3. *ℓ_1 distance.* The ℓ_1 distance is also called the Manhattan distance. The ℓ_1 distance between $\mathbf{x},\mathbf{y} \in \mathbf{R}^d$ is

$$\|\mathbf{x}-\mathbf{y}\|_1 := \sum_{i=1}^{d} |x_i - y_i|.$$

The proportional data, i.e., satisfying $\|\mathbf{x}_i\|_1 = 1$ and $\mathbf{x}_i > 0$ for each $i = 1,\ldots,n$, are encountered in many domains, such as skill allocation in workforce management, consumption patterns in marketing studies, and topic distributions in text mining. For the proportional data, the ℓ_1 distance is a preferred metric because it offers intuitive and actionable interpretations. Hence, the k-means algorithm based on ℓ_1 distance was proposed by [27] to solve the following problem

$$\begin{array}{ll} \min\limits_{\mathbf{A}} & F(\mathbf{a}_1,\mathbf{a}_2,\cdots,\mathbf{a}_k) := \sum_{i=1}^{n} \min_{1\le j\le k} \|\mathbf{x}_i - \mathbf{a}_j\|_1 \\ \text{s.t.} & \|\mathbf{a}_j\|_1 = 1, \mathbf{a}_j > 0. \end{array}$$

2.3. **Generalized means.** In k-means (Algorithm 1), the arithmetic mean is adopted to update cluster centers at each iteration. Alternatively, the geometric mean and harmonic mean can be used for certain purposes, as well as the general nonlinear mean. In details, for a continuous and monotone function $g : \mathbb{R}_+ \to \mathbb{R}_+$, the nonlinear mean M_g is defined by

$$M_g(\mathbf{y}) := g^{-1}\left(\frac{1}{k}\sum_{i=1}^{k} g(y_i)\right), \tag{2.7}$$

where g^{-1} is the inverse function of g. Several popular nonlinear means are listed as follows.

Example 2.5. When the nonlinear kernel $g(y) := y$, $\log y$ and $\frac{1}{y}$, the nonlinear mean (2.7) is reduced to arithmetic mean, geometric mean and harmonic mean, respectively.

Example 2.6. When the nonlinear kernel $g(y) := y^s$ with $s \in \mathbb{R}$, the nonlinear mean (2.7) is called the power means.

(i) If $s = 1$, then M_g is reduced to the arithmetic mean.
(ii) If $s = -1$, then M_g is reduced to the harmonic mean.
(iii) If s tends to 0, then M_g converges to the geometric mean.
(iv) If s tends to $-\infty$, we have

$$\lim_{s\to-\infty} M_g(\mathbf{y}) = \min_{1\le i\le k} y_i. \tag{2.8}$$

In view of (2.8), the center-based clustering (2.2) can be considered as problem (2.4) with power means and $s \to -\infty$. Inspired by this idea, a power k-means algorithm was introduced in [65], in which the power s is iteratively pushed to $-\infty$ and an MM optimization framework is applied to solve the sum-minimization problem of s-power means at each iteration. Particularly, when $s \equiv -1$, it is reduced to the k-harmonic means.

The power k-means algorithm shares the same computational complexity of $\mathcal{O}(nkd)$ with k-means. It was shown in [65, Propositions 2.1 and 3.2] that the power k-means algorithm converges uniformly to a solution of problem (2.1).

3. Convex clustering

Convex clustering was originally proposed by Hocking et al. [23] and Lindsten et al. [32], which is also known as the SON (sum-of-norms) model or clustering path. Convex clustering has significant and stable clustering capability for the manifold data; see, e.g., [11].

3.1. **Basic model and framework.** Instead of the forward style of allocating each sample to certain cluster in center-based clustering, the convex clustering adopts a backward style: it assigns a cluster representative for each sample and remove the close representatives (also called cluster recovery). In particular, let $\mathbf{u}_i$ be the cluster representative of sample $\mathbf{x}_i$ for each $i = 1,\dots,n$. The principle of the convex clustering is that (i) sample $\mathbf{x}_i$ shall be near to its cluster representative $\mathbf{u}_i$; and (ii) different clusters shall be separative while samples in the same cluster have the same representative. Hence the convex clustering can be transformed into the following SON minimization problem:

$$\min_{\mathbf{U}\in\mathbb{R}^{d\times n}} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{u}_i\|^2 + \lambda \sum_{i<j} \|\mathbf{u}_i - \mathbf{u}_j\|_q, \tag{3.1}$$

where $\|\cdot\|_q$ is the ℓ_q norm (usually $q := 1, 2$ or $+\infty$). In (3.1), the first term aims to minimize the total distance between samples and their representatives, the second term is to penalize the difference between cluster representatives, and the parameter $\lambda > 0$ provides a tradeoff between them. By solving (3.1), one can obtain the solution of representative matrix $\mathbf{U}$, in which the number of disparate values of $\{\mathbf{u}_i\}_{i=1}^n$ is the one of clusters. Thus, in convex clustering, the number of clusters is not required to be provided in advance that is an advantage over center-based clustering.

When dealing with large-scale data sets, a k-nearest neighborhood (KNN) graph of the data set was recommended in [55] to improve the accuracy of convex clustering (3.1), which is formulated as

$$\min_{\mathbf{U}\in\mathbb{R}^{n\times d}} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{u}_i\|^2 + \lambda \sum_{(i,j)\in\mathscr{E}} w_{ij}\|\mathbf{u}_i - \mathbf{u}_j\|_q, \tag{3.2}$$

where $\mathscr{E}$ is the edge set of KNN-graph, and $w_{ij} > 0$ is the weight of edge. The KNN-graph is constructed by each point's k-nearest neighbors. The principle of the weight is that the larger the $\|\mathbf{x}_i - \mathbf{x}_j\|$, the smaller the w_{ij}. A common one is via the Gaussian kernel:

$$w_{ij} := \begin{cases} \exp\left(-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{2\sigma^2}\right), & \text{if } (i,j)\in\mathscr{E}, \\ 0, & \text{otherwise}, \end{cases} \tag{3.3}$$

where $\sigma > 0$ is a parameter controlling the width of neighborhoods; and another option was introduced by [49] that

$$w_{ij} := \begin{cases} \frac{\sum_{t=1}^n N_t}{n\sqrt{N_i N_j}}, & \text{if } (i,j)\in\mathscr{E}, \\ 0, & \text{otherwise}, \end{cases}$$

where N_i is the number of edges with vertex i.

The cluster recovery theory of models (3.1) and (3.2) were established in [43, Theorem 1] and [54, Theorem 5], respectively. Two algorithms based on ADMM and AMA frameworks, respectively, were proposed to solve problem (3.2) in [11]. An efficient algorithm based on semismooth Newton was proposed by [54] to solve problem (3.2), and the superlinear convergence theory was established in [54, Theorems 12 and 13].

3.2. **Robust continuous clustering.** Composing a robust estimator $\rho(\cdot)$ on the penalty term of (3.2), the robust continuous clustering (RCC) was proposed by [49] that is formulated as

$$\min_{\mathbf{U}\in\mathbb{R}^{d\times n}} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{u}_i\|^2 + \lambda \sum_{(i,j)\in\mathscr{E}} w_{i,j}\rho(\|\mathbf{u}_i - \mathbf{u}_j\|_2). \tag{3.4}$$

The Geman-McClure estimator [20] is a typical and robust M-estimator, namely,

$$\rho(t) := \frac{\mu t^2}{\mu + t^2}.$$

An algorithm based on BCD framework was proposed by [49] to solve RCC (3.4) with the Geman-McClure estimator, and it was shown in [49] that RCC (3.4) owns more robust performance than SON (3.2).

Moreover, in order to attain robust clustering and dimensionality reduction (DR) simultaneously for high-dimensional data sets, the RCC-DR was proposed by [49] by combining RCC

and dictionary learning techniques:

$$\min_{\mathbf{U},\mathbf{Z},\mathbf{D}} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{D}\mathbf{z}_i\|^2 + \gamma \sum_{i=1}^{n} \|\mathbf{z}_i\|_1 + \sum_{i=1}^{n} \|\mathbf{z}_i - \mathbf{u}_i\|^2 + \lambda \sum_{(i,j)\in\mathscr{E}} w_{i,j}\rho(\|\mathbf{u}_i - \mathbf{u}_j\|),$$

where $\mathbf{D} \in \mathbb{R}^{d\times d'}$ is a dictionary, and $\mathbf{z}_i \in \mathbb{R}^{d'}$ is a sparse code.

3.3. **KNN-graph alternative.** Note that SON (3.2) and RCC (3.4) are both based on the KNN-graph which is constructed from the original data, the quality of graph has significant impact on the performance of clustering algorithms. However, it is unavoidable that a KNN-graph contains some false connections. In order to reduce the influence of false connections in KNN-graph, an auxiliary variable $\{l_{i,j} : (i,j) \in \mathscr{E}\}$ is introduced to build up the RCC model as

$$\min_{\mathbf{U},\mathbf{L}} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{u}_i\|^2 + \lambda \sum_{(i,j)\in\mathscr{E}} w_{i,j}(l_{i,j}\|\mathbf{u}_i - \mathbf{u}_j\|^2 + \varphi(l_{i,j})), \tag{3.5}$$

where $\varphi(\cdot)$ is a penalty function on the removed connections, so as to make the connections as in KNN-graph be used in a self-adapting manner. Particularly,

$$l_{ij} \to \begin{cases} 1, & \text{if } (\mathbf{x}_i, \mathbf{x}_j) \text{ are in the same cluster,} \\ 0, & \text{if } (\mathbf{x}_i, \mathbf{x}_j) \text{ are in different clusters,} \end{cases} \quad \text{and} \quad \varphi(t) \to \begin{cases} 1, & \text{if } t \to 0, \\ 0, & \text{if } t \to 1. \end{cases}$$

When $\varphi(t) := \mu(\sqrt{t} - 1)^2$, (3.5) is reduced to (3.4) with Geman-McClure estimator; see [49].

Besides the penalty approach on the KNN-graph, another alternative is the adaptive graph shrinking (AGS) technique [57]. Its idea is to update the graph at each iteration and to take advantage of higher quality graph than a fixed KNN-graph in RCC. Particularly, in place of the fixed weight matrix $\mathbf{W}$ of KNN-graph in RCC (3.5), the AGS technique views $\mathbf{W}$ as a variable of non-negative and symmetric matrix. Consequently, an AGS-based RCC was introduced in [57], namely,

$$\begin{aligned} \min_{\mathbf{U},\mathbf{L},\mathbf{W}} \quad & \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{u}_i\|^2 + \alpha\|\mathbf{W}\|_F^2 - \beta\mathbf{1}^{\mathbf{T}}\log(\mathbf{W}\mathbf{1}) + \lambda \sum_{(i,j)\in\mathscr{E}} w_{i,j}(l_{i,j}\|\mathbf{u}_i - \mathbf{u}_j\|^2 + \varphi(l_{i,j})) \\ \text{s.t.} \quad & \mathbf{W}^{\mathbf{T}} = \mathbf{W}, \mathbf{W} \geq 0, \end{aligned}$$

where the second term aims to penalize the weight either too big or too small, and the third term is a logarithmic barrier that is able to avoid the isolated vertex. An AGS algorithm based on ADMM framework was proposed by [57] to solve the above problem, which takes the KNN-graph as initialization and updates the graph with progressively higher quality at each iteration. It was shown in [57] that the AGS algorithm outperforms RCC (3.5) with a fixed KNN-graph.

4. Spectral clustering

Spectral clustering is to transform the clustering problem into an optimal partition problem of the graph constructed by the data set. Compared with the traditional clustering algorithms such as k-means, spectral clustering has several significant advantages; for example, spectral clustering is capable to handle data set with arbitrary shape and release the stalemate of local optima [62]. It was revealed by experimental evaluations [9] that spectral clustering outperforms k-means on accuracy and robustness.

Spectral clustering algorithm stems from spectral graph theory [12]. Let $\mathbf{G} := \{\mathbf{V}, \mathbf{E}\}$ be an undirected graph, where $\mathbf{V} := \{\mathbf{x}_i\}_{i=1}^n$ and $\mathbf{E}$ are the sets of vertices (i.e., sample points) and edges, respectively. Let $\mathbf{W} := (w_{ij})$ denote the adjacency matrix of the graph with $w_{ij} \geq 0$ being a non-negative weight of the edge linking $\mathbf{x}_i$ and $\mathbf{x}_j$ (denotes the similarity of $\mathbf{x}_i$ and $\mathbf{x}_j$; $w_{ij} = 0$ means that $\mathbf{x}_i$ and $\mathbf{x}_j$ are not connected). The principle of spectral clustering is to find a partition of the graph such that the edges between different subgraphs have low weights and the edges within a subgraph have high weights, i.e., the sample points in the same cluster are more similar than the ones in other clusters.

Spectral clustering methods usually involve three steps:

(1) construct a Laplacian matrix;
(2) perform eigen-decomposition of the Laplacian matrix;
(3) attain clustering assignments from eigenvectors space via k-means.

4.1. **Similarity graph and Laplacian matrix.** The similarity graph plays a key role in the construction of graph Laplacian matrix, which is the major tool for spectral clustering. Three popular approaches for constructing the similarity graph and the adjacency matrix are listed as follows. For more classes of similarity graphs, one can refer to [62] and references therein.

(i) ε-neighborhood graph. It connects the vertices whose distance is smaller than ε, and provides an unweighted graph. Particularly, letting $s_{ij} := \|x_i - x_j\|^2$, the adjacency matrix $\mathbf{W}$ is constructed by

$$w_{ij} := \begin{cases} \varepsilon, & \text{if } s_{ij} \leq \varepsilon, \\ 0, & \text{if } s_{ij} > \varepsilon. \end{cases}$$

(ii) KNN-graph. It connects every vertex with its k-nearest neighbors. The adjacency matrix $\mathbf{W}$ can be constructed by (3.3).

(iii) Fully connected graph. The edge between arbitrary two vertices is connected and weighted by a similarity function: the more similar the vertices, the larger the function value. A typical similarity function is the Gaussian similarity function

$$w_{ij} := \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right),$$

where the parameter σ controls the width of neighborhoods.

Given an adjacency matrix $\mathbf{W}$ as mentioned above, the degree of a vertex $\mathbf{x}_i$ is defined by $d_i := \sum_{j=1}^n w_{ij}$ (the total weights of edges linking $\mathbf{x}_i$), and $\mathbf{D} := \mathrm{diag}(d_1, \cdots, d_n)$ is the degree matrix of the graph. Then an unnormalized graph Laplacian matrix [12] is defined by

$$\mathbf{L} := \mathbf{D} - \mathbf{W}. \tag{4.1}$$

The Laplacian matrix $\mathbf{L}$ is positive semi-definite, and the eigenvector corresponding to its smallest eigenvalue (i.e., $\lambda_{\min} = 0$) is the one vector (i.e., $\mathbf{1}$). Two types of normalized graph Laplacian matrices [12] are defined by

$$\mathbf{L}_{sym} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}} \quad \text{and} \quad \mathbf{L}_{rw} = \mathbf{D}^{-1}\mathbf{L}.$$

The former one is a symmetric matrix, and the latter one is closely related to a random walk.

4.2. **Graph cut.** The total weights of edges between two subgraphs of $\mathbf{G}$ is called *cut*, i.e.,

$$\mathrm{cut}(\mathbf{G}_1,\mathbf{G}_2)=\sum_{i\in\mathbf{G}_1,j\in\mathbf{G}_2} w_{ij}.$$

As mentioned above that the principle of spectral clustering is to find an optimal partition such that the edges between different subgraphs have low weights, one popular idea of spectral clustering is the so-called minimum cut [3]. Several variants have been devised to balance the partition structures such as ratio cut [21], normalized cut [51], and minmax Cut.

4.2.1. *Minimum cut.* Consider the case of two subgraphs for example. An auxiliary indicator vector $\mathbf{f}\in\mathbb{R}^n$ is introduced to indicate the subgraph of each vertex, i.e.,

$$f_i:=\begin{cases} c_1, & \text{if } \mathbf{x}_i\in\mathbf{G}_1,\\ c_2, & \text{if } \mathbf{x}_i\in\mathbf{G}_2,\end{cases}$$

where c_1 and c_2 are labels of subgraphs. Then the cut can be reformulated by (4.1) as

$$\mathrm{cut}(\mathbf{G}_1,\mathbf{G}_2)=\frac{\sum_{i=1}^n\sum_{j=1}^n w_{ij}(f_i-f_j)^2}{2(c_1-c_2)^2}=\frac{\mathbf{f^T L f}}{(c_1-c_2)^2}. \tag{4.2}$$

Therefore, setting $c_1:=1$ and $c_2:=-1$, the minimum cut model is expressed as

$$\begin{aligned}\min_{\mathbf{f}}\quad & \mathrm{cut}(\mathbf{G}_1,\mathbf{G}_2):=\mathbf{f^T L f}\\ \text{s.t.}\quad & \mathbf{f^T f}=n,\mathbf{f}\perp\mathbf{1},\mathbf{f}\in\{-1,1\}^n.\end{aligned} \tag{4.3}$$

It was noticed in [64] that the minimum cut favors cutting small sets of isolated nodes in the graph, and thus may lead to an unbalanced partition.

4.2.2. *Ratio cut.* In order to obtain a balanced partition, the ratio cut was introduced in [21] and defined by

$$\mathrm{R_{cut}}(\mathbf{G}_1,\mathbf{G}_2):=\mathrm{cut}(\mathbf{G}_1,\mathbf{G}_2)\left(\frac{1}{|\mathbf{G}_1|}+\frac{1}{|\mathbf{G}_2|}\right), \tag{4.4}$$

where $|\mathbf{G}_i|$ means the number of nodes in $\mathbf{G}_i$. Letting $c_1:=\sqrt{\frac{|\mathbf{G}_1|}{|\mathbf{G}_2|}}$ and $c_2:=-\sqrt{\frac{|\mathbf{G}_2|}{|\mathbf{G}_1|}}$ and by (4.2), (4.4) can be re-written as $\mathrm{R_{cut}}(\mathbf{G}_1,\mathbf{G}_2)=\frac{\mathbf{f^T L f}}{|\mathbf{G}|}$, and thus the ratio cut model is

$$\begin{aligned}\min_{\mathbf{f}}\quad & \mathbf{f^T L f}\\ \text{s.t.}\quad & \mathbf{f^T f}=\sqrt{n},\mathbf{f}\perp\mathbf{1},\mathbf{f}\in\left\{-\sqrt{\tfrac{|\mathbf{G}_2|}{|\mathbf{G}_1|}},\sqrt{\tfrac{|\mathbf{G}_1|}{|\mathbf{G}_2|}}\right\}^n.\end{aligned}$$

Extended to a partition into k subgraphs $\{\mathbf{G}_i\}_{i=1}^n$, the indicator matrix $\mathbf{H}=(h_{ij})\in\mathbb{R}^{n\times k}$ is defined by

$$h_{ij}:=\begin{cases}\frac{1}{\sqrt{|\mathbf{G}_j|}}, & \text{if } \mathbf{x}_i\in\mathbf{G}_j,\\ 0, & \text{otherwise},\end{cases}$$

and the corresponding ratio cut model is

$$\begin{aligned}\min_{\mathbf{H}}\quad & \mathrm{trace}(\mathbf{H^T L H})\\ \text{s.t.}\quad & \mathbf{H^T H}=\mathbf{I},\mathbf{h}_j\in\left\{0,\tfrac{1}{\sqrt{|\mathbf{G}_j|}}\right\}^n.\end{aligned} \tag{4.5}$$

4.2.3. *Normalized cut.* Another balance technique is to use the degree of vertices in the subgraph. Consequently, the normalized cut is defined by

$$\mathrm{N_{cut}}(\mathbf{G}_1,\mathbf{G}_2) := \mathrm{cut}(\mathbf{G}_1,\mathbf{G}_2)\left(\frac{1}{\mathrm{vol}(\mathbf{G}_1)}+\frac{1}{\mathrm{vol}(\mathbf{G}_2)}\right),$$

where $\mathrm{vol}(\mathbf{G}_i) := \sum_{\mathbf{x}_j\in\mathbf{V}_i} d_j$. By using the similar arguments for (4.5), the normalized cut model can be expressed as

$$\begin{aligned} \min_{\mathbf{H}} \quad & \mathrm{trace}(\mathbf{H^T L H}) \\ \text{s.t.} \quad & \mathbf{H^T D H} = \mathbf{I},\, \mathbf{h}_j \in \left\{0, \tfrac{1}{\sqrt{\mathrm{vol}(\mathbf{G}_j)}}\right\}^n. \end{aligned} \tag{4.6}$$

4.2.4. *Bi-criteria cut.* Liu et al. [33] introduced a bi-criteria cut:

(i) minimize the total weight of removing edges;
(ii) maximize the number of connected components in the graph.

Let κ denote the number of connected components in the graph. It was shown in [33, Lemma 1] that

$$\kappa = n - \mathrm{rank}(\mathbf{L}).$$

Then, via the scalarization technique, the bi-criteria cut model can be formulated as

$$\begin{aligned} \min_{\mathbf{Z}} \quad & \mathrm{rank}(\mathbf{L}(\mathbf{W}\circ\mathbf{Z})) - \beta\,\mathrm{trace}(\mathbf{WZ}) \\ \text{s.t.} \quad & \mathbf{Z}\in S^n\cap\{0,1\}^{n\times n},\, \mathrm{supp}(\mathbf{Z})\subseteq\mathrm{supp}(\mathbf{W}), \end{aligned} \tag{4.7}$$

where $\circ$ denotes the Hardmard product, $\mathbf{Z}$ is a 0-1 matrix that reflects whether the edge survives after partition, S^n denotes the set of symmetric matrices in $\mathbb{R}^{n\times n}$, and $\mathrm{supp}(\cdot)$ indicates the support set.

4.3. **Spectral clustering algorithms.** The spectral clustering problems mentioned in the preceding subsection are discrete optimization and thus NP-hard to solve. One natural approach for spectral clustering is the continuous relaxation, i.e., the indicator variable $\mathbf{f}$ (or $\mathbf{H}$, $\mathbf{Z}$) is relaxed to be variable in $\mathbb{R}^n$.

Through the continuous relaxation approach, problems (4.3), (4.5) and (4.6) are relaxed to the minimization problems on Rayleigh quotient

$$\mathrm{trace}(\mathbf{H^T L H}) = \sum_{i=1}^{k}\mathrm{RayQ}(\mathbf{L},\mathbf{h}_i), \quad \text{where} \quad \mathrm{RayQ}(\mathbf{L},\mathbf{f}) := \frac{\mathbf{f}^\top\mathbf{L}\mathbf{f}}{\mathbf{f}^\top\mathbf{f}}.$$

By Courant-Fischer Theorem [12], the minimal solution of Rayleigh quotient is the eigenvector corresponding to the smallest nonzero eigenvalue of $\mathbf{L}$, and the minimal solution of the latter one is $\mathbf{H}\in\mathbb{R}^{n\times k}$ consisting of the eigenvectors corresponding to the first k smallest nonzero eigenvalue of $\mathbf{L}$; see, e.g., [51].

In the bi-criteria cut problem (4.7), relaxing $\mathbf{Z}$ to a continuous variable in $[0,1]^{n\times n}$ and replacing the rank function by the trace norm of the first k smallest eigenvalues, a continuous optimization problem of bi-criteria cut spectral clustering was introduced in [33] as

$$\begin{aligned} \min_{\mathbf{Z},\mathbf{H}} \quad & \mathrm{trace}(\mathbf{H^T L}(\mathbf{W}\circ\mathbf{Z})\mathbf{H}) - \beta\,\mathrm{trace}(\mathbf{WZ}) \\ \text{s.t.} \quad & \mathbf{Z}\in S^n\cap[0,1]^{n\times n},\, \mathrm{supp}(\mathbf{Z})\subseteq\mathrm{supp}(\mathbf{W}),\, \mathbf{H^T H} = \mathbf{I}_k. \end{aligned}$$

An algorithm based on BCD framework was proposed by [33] to solve the above problem.

Obtaining the optimal solution $\mathbf{H}$ via the continuous relaxation technique mentioned above, one standard approach to clustering assignment is treating each row of $\mathbf{H}$ as a representation of the origin data and applying k-means to partition $\mathbf{H}$ into k clusters.

5. SUBSPACE CLUSTERING

Subspace clustering is designed to handle high-dimensional data sets such as images, audios and videos. These types of data are generally multi-dimensional tensors, which are quite high-dimensional data while only a small number of features therein contribute to the clustering.

5.1. **Sparse subspace clustering.** Subspace clustering is a spectral clustering based method. As mentioned in section 4, the input of spectral clustering is the similarity matrix of data set defined by similarity graph~~s~~. Instead, subspace clustering acquires the similarity matrix by discovering the latent subspace information of high-dimensional data set and via an optimization problem.

Subspace clustering is based on the *self-expressiveness property* [60] of data: each data point in a union of subspaces can be efficiently re-constructed by a linear combination of other points in the data set. More precisely, each data point $\mathbf{x}_i$ can be expressed as

$$\mathbf{x}_i = \mathbf{X}\mathbf{c}_i,\ i = 1,2,\cdots,n$$

where $\mathbf{c}_i \in \mathbb{R}^n$ is a weight vector satisfying $c_{ii} = 0$ that eliminates a trivial solution. Equivalently,

$$\mathbf{X} = \mathbf{X}\mathbf{C} \quad \text{with} \quad \operatorname{diag}(\mathbf{C}) = \mathbf{0}.$$

One key observation (assumption) is that each data point is represented only by points from the same subspace, and thus the weights contributed by data points from other subspaces are all zeros. This leads to a sparsity structure of the expression matrix $\mathbf{C}$. By using the sparsity structure of $\mathbf{C}$, Vidal [60] introduced a sparse subspace clustering (SSC) optimization problem

$$\begin{aligned} \min_{\mathbf{C}} \quad & \|\mathbf{C}\|_1 \\ \text{s.t.} \quad & \mathbf{X} = \mathbf{X}\mathbf{C},\ \operatorname{diag}(\mathbf{C}) = \mathbf{0}, \end{aligned} \tag{5.1}$$

where $\|C\|_1 := \sum_i \sum_j |c_{ij}|$ is a (convex) sparsity promoting norm.

Obtained the expression matrix $\mathbf{C}$ via SSC (5.1), the similarity matrix $\mathbf{W}$ can be constructed by

$$\mathbf{W} = |\mathbf{C}| + |\mathbf{C}^{\mathbf{T}}|, \tag{5.2}$$

and then apply spectral clustering technique to achieve clustering assignments. To conclude, the framework of SSC is stated in Algorithm 2.

Algorithm 2: Sparse subspace clustering

1 **Input:** data matrix $\mathbf{X}$, ~~parameter~~ ~~λ~~.
2 **Output:** clustering assignments $\mathbf{Z}$.
3 Calculate the expression matrix $\mathbf{C}$ by solving (5.1).
4 Construct the similarity matrix $\mathbf{W}$ by (5.2).
5 Apply spectral clustering using similarity matrix $\mathbf{W}$ to attain clustering assignments $\mathbf{Z}$.

5.2. **Generalization of SSC.** From practical considerations and application circumstances, the high-dimensional data set always contains noise, outlier and missing entries. To deal with the phenomena of noise and missing data, several generalizations of the SSC have been devised and widely applied; see [15, 45] and references therein.

(i) Noise and sparse outlying entries. To handle the noisy high-dimensional data set, Elhamifar and Vidal [15] proposed a noise-aware SSC optimization model, i.e.,

$$\begin{array}{ll}\min\limits_{\mathbf{C},\mathbf{E}} & \|\mathbf{C}\|_1+\lambda\,\mathrm{pen}(\mathbf{E})\\ \text{s.t.} & \mathbf{X}=\mathbf{X}\mathbf{C}+\mathbf{E},\ \mathbf{1}^{\mathbf{T}}\mathbf{C}=\mathbf{1}^{\mathbf{T}},\ \mathrm{diag}(\mathbf{C})=\mathbf{0}.\end{array}$$

where $\mathrm{pen}(\cdot)$ could be ℓ_1 norm, squared Frobenius norm and other penalty terms adjusted to the types of noise $\mathbf{E}$, and the constraint $\mathbf{1}^{\mathbf{T}}\mathbf{C}=\mathbf{1}^{\mathbf{T}}$ is for the self-expressiveness property in affine subspaces. For example, when the data set is corrupted by random noise or sparse outlying entries, the penalty term $\mathrm{pen}(\cdot)$ is suggested to adopt the squared Frobenius norm or the ℓ_1 norm, respectively; see, e.g., [15].

(ii) Missing entries. It is commonly believed that the high-dimensional data in reality always underlie at a low-dimensional subspace. This leads to the low-rank property of high-dimensional data. The low-rank matrix completion is a popular technique to deal with missing entries, and has been applied to subspace clustering; see [16, 45] and references therein. For example, the sparse representation with missing entries and matrix completion (SRME-MC) [16] is formulated as the following optimization problem

$$\begin{array}{ll}\min\limits_{\mathbf{X_0},\mathbf{C},\mathbf{E}} & \|\mathbf{C}\|_1+\alpha\|\mathbf{X}_0\|_*+\lambda\|\mathbf{E}\|_q\\ \text{s.t.} & \mathbf{X}_0=\mathbf{X}_0\mathbf{C}+\mathbf{E},\ \mathbb{P}_\Omega(\mathbf{X_0})=\mathbb{P}_\Omega(\mathbf{X}),\ \mathrm{diag}(\mathbf{C})=\mathbf{0},\end{array}\tag{5.3}$$

where $\|\cdot\|_*$ is the nuclear norm promoting the low-rank structure, Ω is the index set of known entries of $\mathbf{X}$, $\mathbf{X}_0$ is the completed data from $\mathbf{X}$, and $\mathbb{P}_\Omega(\cdot)$ indicates the entries with positions in Ω. An algorithm based on ADMM framework was proposed by [16] to solve SRME-MC (5.3).

6. Optimal Transport based clustering

Optimal transport (OT) was originally raised by G. Monge [39] in the 18th century and represented by L. Kantorovich [26], which aims to find a transportation map for matching distribution data with the minimum transportation cost. In the theory of computational OT [46], the Wasserstein distance is a metric for distributions that is defined by the minimal transportation cost between distributions, and can be understood as an extension of Euclidean distance to distribution data; the Wasserstein barycenter [1] is the mean of a set of distributions measured by the Wasserstein distance, and thus can be understood as a cluster center of distributions in the sense of the Wasserstein distance. By virtue of the Wasserstein distance and Wasserstein barycenter, a Wasserstein clustering was introduced by [22, 68] that extends the k-means to cluster distribution data.

6.1. **Optimal transport.** Let X and Y be two separable metric spaces, and let $c: X\times Y\to\mathbb{R}_+$ be a Borel-measurable function, which represents the cost of transporting a unit mass from $x\in X$ to $y\in Y$. Given Borel probability measures μ and ν on X and Y, Monge's formulation

[39] of OT is to find a (measure-preserving) transport map $T : X \to Y$ that minimizes the total transportation cost; namely,

$$\begin{array}{ll} \min\limits_{T} & \int_X c(x,T(x))\mathrm{d}\mu(x) \\ \text{s.t.} & T_{\#}\mu = \nu, \end{array}$$

where $T_{\#}\mu(\cdot) := \mu(T^{-1}(\cdot))$ is the push-forward of μ by T. Equivalently, Kantorovich's formulation [26] of OT aims to find a transportation plan with minimal total transportation cost; that is,

$$\begin{array}{ll} \min\limits_{\pi} & \int_{X\times Y} c(x,y)\mathrm{d}\pi(x,y) \\ \text{s.t.} & \pi \in \Gamma(\mu,\nu), \end{array} \tag{6.1}$$

where $\Gamma(\mu,\nu) := \{\pi : (\Gamma_X)_{\#}\pi = \mu, (\Gamma_Y)_{\#}\pi = \nu\}$ denotes the set of transportation plans from μ to ν, i.e., joint distributions with marginals μ and ν. Clearly, the discrete version of Kantorovich's OT can be cast into a linear optimization problem

$$\begin{array}{ll} \min\limits_{\Pi\in\mathbb{R}_+^{n\times m}} & \langle C,\Pi\rangle \\ \text{s.t.} & \sum_{j=1}^m \Pi_{i,j} = u_i, \quad \sum_{i=1}^n \Pi_{i,j} = v_j, \end{array} \tag{6.2}$$

where the cost matrix $C_{i,j} := c(x_i,y_j)$, the transport matrix $\Pi_{i,j} := \pi(x_i,y_j)$, and the distributions $\mu := (u_i)_1^n$ and $\nu := (v_i)_1^m$.

6.2. **Wasserstein distance and barycenter.** The Wasserstein distance is the minimal transportation cost of Kantorovich's OT (6.1) with $c(x,y) := \|x-y\|^p$, that is

$$W_p(\mu,\nu) := \left(\inf_{\pi\in\Gamma(\mu,\nu)} \int_\pi \|x-y\|^p \mathrm{d}\pi(x,y)\right)^{\frac{1}{p}}; \tag{6.3}$$

or the minimal transportation cost of discrete OT (6.2) with $C_{ij} := \|x_i - y_j\|^p$. It is also called the earth mover's distance (EMD) in computer science and the Mallows distance in statistics.

The Wasserstein distance is one of the most fundamental metrics on spaces of probability measures and enjoys several significant benefits: (i) it incorporates the geometry of the ground space, (ii) it not only describes the distance between two distributions, but also explains how to transport one distribution to another, and (iii) it is applicable to distributions with different dimensions, even to discrete and continuous distributions; see [44] and references therein. The Wasserstein distance provides an useful tool for the clustering of distributions because it captures key shape characteristics of the distributions. Particularly, the 2-Wasserstein distance (i.e., (6.3) with $p = 2$ and written as $W(\cdot,\cdot)$ for the sake of simplicity) is the most common one because of the geometric properties such as Wasserstein barycenters.

Given a set of distributions $\{\mu_i\} \subseteq \mathscr{P}(X)$ and a set of non-negative weights $\{\lambda_i\}$. The Wasserstein barycenter [1] is defined by the mean of the set of distributions under the Wasserstein distance, that is

$$\overline{\mu} := \arg\min_{\mu} \sum_{i=1}^N \lambda_i W^2(\mu_i,\mu). \tag{6.4}$$

Several popular and efficient optimization algorithms were developed to compute Wasserstein barycenters [13, 68]. It was reported in [59] that the Wasserstein barycenter $\overline{\mu}$ preserves the shape of distributions $\{\mu_i\}$.

It is worth noting that problem (6.4) has an analytical solution as a (weighted) arithmetic mean of distributions, when the Wasserstein distance is replaced by the Euclidean distance. Hence the Wasserstein barycenter could be understood as a cluster center of distributions in the sense of Wasserstein distance.

6.3. **Wasserstein clustering.** Recalling the center-based clustering in section 2.1, it seeks to assign a set of cluster centers such that the total Euclidean distance of the samples from their nearest centers is minimal.

Consider a set of discrete distributions $\{(\mathbf{x}_i, \mu_i), i = 1, \cdots, m\}$. As mentioned in the preceding section, it is better to use the Wasserstein distance to measure the distance between distributions than the Euclidean distance. By virtue of the Wasserstein distance and inspired by the idea of center-based clustering, a Wasserstein clustering, also called discrete distribution clustering (D2-clustering), was introduced by [30] to find a set of Wasserstein barycenters $\{\nu_j\}_1^k$ such that the total Wasserstein distance of the distributions from their nearest barycenters is minimal:

$$\min_{\nu} \sum_{i=1}^{n} \min_{1\le j\le k} W^2(\mu_i, \nu_j). \tag{6.5}$$

Inspired by the idea of k-means, a natural approach for approximately solving Wasserstein clustering (6.5) was proposed by [30] through two alternative steps: (i) assign sample distributions to their nearest Wasserstein barycenters in terms of Wasserstein distance (6.3), and (ii) apply (6.4) to update Wasserstein barycenters of distributions involved.

Moreover, a variational Wasserstein clustering was proposed by [38] based on the variational OT technique that aims to find a set of discrete sparse barycenters to best represent a continuous probability measure, or its discrete empirical representation.

7. Numerical Experiments

This section is contributed to carry out numerical experiments to compare clustering algorithms mentioned in the preceding sections for datasets of various shapes. All numerical experiments are implemented in Matlab R2018b and executed on a personal desktop (Intel Core i7-7700HQ, 2.80GHz, 24.00GB of RAM).

7.1. **Generated datasets and clustering algorithms.** Six datasets of typical shapes are generated to explore the numerical performance of clustering algorithms. In Figure 1, three datasets in the first row are spherical data: "blob" has four clusters with equal radii; "aniso" has three clusters and each one is not isotropic; "varied" has three clusters with different radii. Three datasets in the second row are manifolds.

In numerical experiments, clustering algorithms mentioned in the preceding sections are conducted to compared with a state-of-the-art density-based clustering algorithm [48]. In order to facilitate the reading of numerical results, we list the abbreviations of clustering algorithms in Table 1.

7.2. **Evaluation index.** Normalized mutual information (NMI) [53] and adjusted mutual information (AMI) [61] are two criteria widely used to evaluate numerical results of clustering. Given a cluster assignment $\mathbf{z}$ and the true label $\mathbf{z}_0$, NMI and AMI are defined by

$$\mathrm{NMI}(\mathbf{z}, \mathbf{z}_0) = \frac{\mathrm{MI}(\mathbf{z}, \mathbf{z}_0)}{f(\mathrm{H}(\mathbf{z}), \mathrm{H}(\mathbf{z}_0))} \quad \text{and} \quad \mathrm{AMI}(\mathbf{z}, \mathbf{z}_0) = \frac{\mathrm{MI}(\mathbf{z}, \mathbf{z}_0) - \mathbb{E}(\mathrm{MI}(\mathbf{z}, \mathbf{z}_0))}{f(\mathrm{H}(\mathbf{z}), \mathrm{H}(\mathbf{z}_0)) - \mathbb{E}(\mathrm{MI}(\mathbf{z}, \mathbf{z}_0))},$$

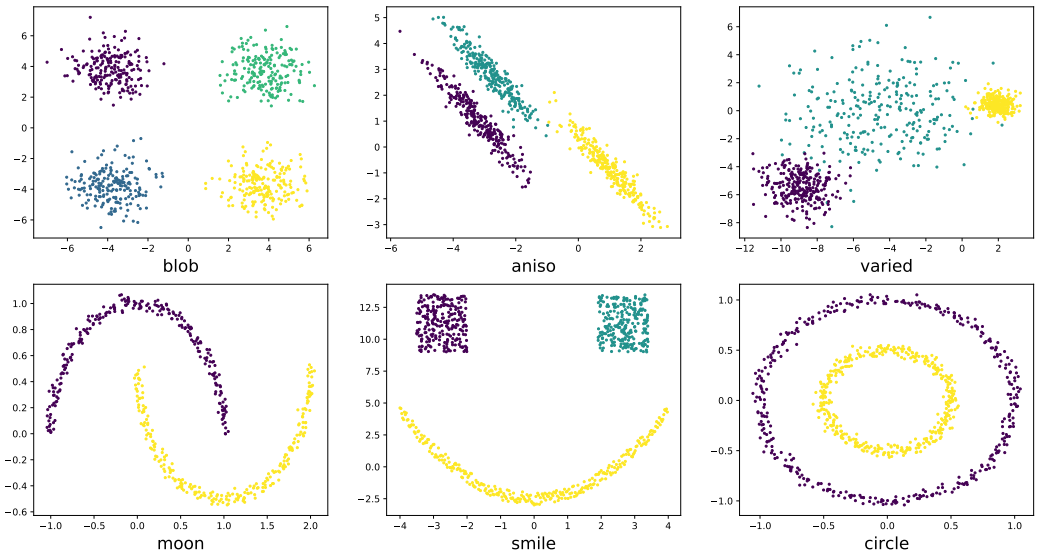

FIGURE 1. Experimental datasets

TABLE 1. List of clustering algorithms used in numerical experiments

Abbreviations	Algorithms
HCD	Hard clustering with Euclidean distance [56] for center-based clustering
RCC	Robust continuous clustering [49] for convex clustering
SPC	Continuous bi-criteria cut [33] for spectral clustering
SSC	ADMM for sparse subspace clustering [15]
CFSFDP	Clustering by fast search and find of density peaks [48]

where $\mathrm{MI}(\cdot,\cdot)$ is the mutual information, $\mathrm{H}(\cdot)$ is the information entropy, $\mathbb{E}(\cdot)$ is the expectation, $f(\cdot,\cdot)$ can be min/max function, arithmetic mean and geometric mean; in this experiment, we use arithmetic mean. Clearly, $\mathrm{NMI} \in [0,1]$ and $\mathrm{AMI} \in [0,1]$. The larger the NMI and AMI, the more accurate the algorithm.

7.3. **Numerical results.** Numerical results and performance of clustering algorithms on datasets mentioned above are illustrated in Table 2 and Figure 2, respectively. The following observations are indicated from numerical results:

(i) For center-based clustering, its performance on spherical data depends on the distribution of radii: the more uniform the radii, the better the performance; its performance on manifold data is not well, except for the separable manifold "smile".
(ii) Convex clustering and spectral clustering work well on manifold data, in which NMI and AMI are generally larger than 0.8.
(iii) Subspace clustering fails for data "moon", while other algorithms generally perform well. It is indicated that subspace clustering may not suitable for low-dimensional data, because the hypothesis of sparse self-expressiveness property may not hold for low-dimensional data.

In conclusion, center-based clustering is suitable for spherical data; convex clustering and spectral clustering are able to handle manifold data; and subspace clustering is capable to deal with high-dimensional data.

TABLE 2. Numerical results of clustering algorithms

	HCD		RCC		SPC		SSC		CFSFDP	
Datasets	NMI	AMI	NMI	AMI	NMI	AMI	NMI	AMI	NMI	AMI
blob	1.000	1.000	0.987	0.987	0.765	0.746	1.000	1.000	1.000	1.000
aniso	0.593	0.592	0.295	0.074	0.010	0.005	0.902	0.901	0.954	0.954
varied	0.784	0.783	0.803	0.799	0.565	0.518	0.742	0.741	0.642	0.642
moon	0.208	0.206	1.000	1.000	1.000	1.000	0.016	0.014	1.000	1.000
smile	1.000	1.000	1.000	1.000	0.960	0.960	0.884	0.884	1.000	1.000
circle	0.000	0.000	0.994	0.994	0.000	0.000	0.000	0.000	0.003	0.002

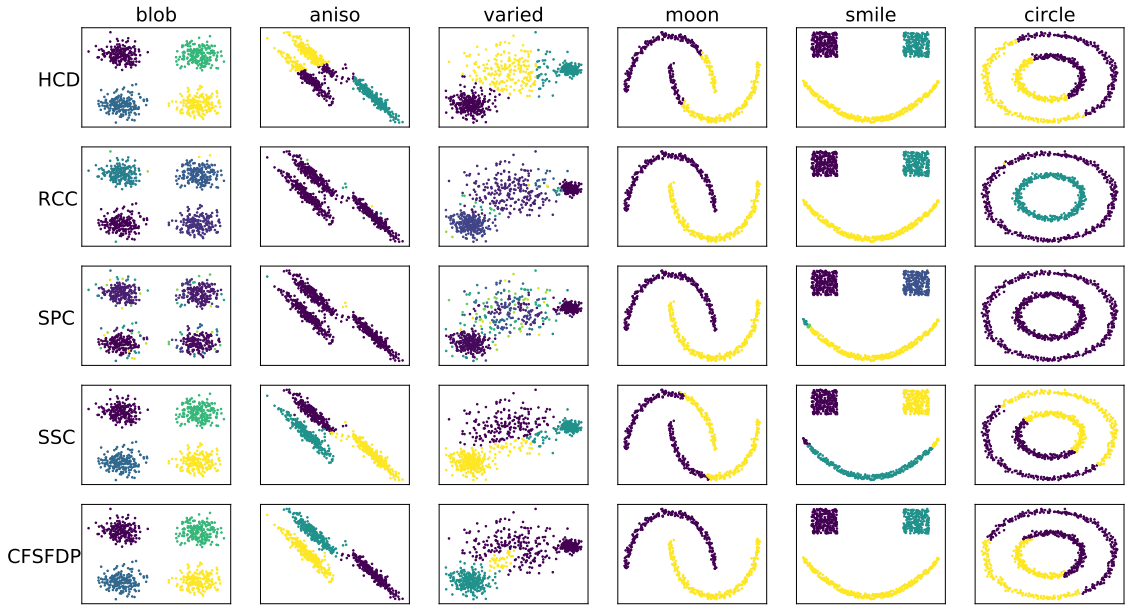

FIGURE 2. Performance of clustering algorithms

ACKNOWLEDGMENTS

The authors are grateful to the editor and the anonymous reviewer for their valuable comments and suggestions toward the improvement of this paper. Yaohua Hu's work was supported in part by the National Natural Science Foundation of China (12071306, 11871347), Natural Science Foundation of Guangdong Province of China (2019A1515011917, 2020A1515010372, 2020B1515310008), Project of Educational Commission of Guangdong Province of China (2019KZDZX1007), Natural Science Foundation of Shenzhen (JCYJ20190808173603590, JCYJ20170817100950436) and Interdisciplinary Innovation Team of Shenzhen University. Carisa Kwok Wai Yu's work was supported in part by grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (UGC/FDS14/P02/17).

APPENDIX A. OPTIMIZATION ALGORITHM

The connection between optimization models and clustering algorithms has been built up in the preceding sections, which is not only helpful to advance the understanding of the principle of existing clustering algorithms, but also useful to inspire new ideas of efficient clustering algorithms. In the era of big data, it is an important issue to design and develop efficient and fast optimization algorithms by virtue of certain structure of large-scale optimization problems; see [7, 42, 47] and references therein. Several popular first-order optimization algorithms in clustering are presented as follows.

A.1. **BCD.** The block coordinate descent (BCD) method [63] has a long history in optimization and has been extensively applied in machine learning, especially in parallel and distributed computation. Consider an unconstrained convex optimization problem with a block-wise variable

$$\min_{\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_n} f(\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_n).$$

The original BCD adopts the Gauss-Seidel scheme to alternately minimize the objective function according to each block $\mathbf{x}_i$ in place of the full variable $\mathbf{x}$; one can refer to [63] for other-types of BCD. The original BCD is formally described as follows.

Algorithm 3: BCD framework

Input: function f.
Output: optimal solution $\mathbf{x}^*$.
Initialize $\mathbf{x}^0 = (\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{x}^n)$.
while *not convergent* **do**
  for $k = 1, \cdots$ **do**
    for $i = 1, \cdots, n$ **do**
      $\mathbf{x}_i^k := \arg\min_{\mathbf{x}_i} f(\mathbf{x}_1^k, \cdots, \mathbf{x}_{i-1}^k, \mathbf{x}_i, \mathbf{x}_{i+1}^{k-1}, \cdots, \mathbf{x}_n^{k-1})$.
    end
  end
end

A.2. **ADMM.** The alternating direction method of multipliers (ADMM) [8] is a widely used algorithm in machine learning. Particularly, consider a composite convex optimization problem

$$\begin{aligned} \min_{\mathbf{x},\mathbf{y}} \quad & f(\mathbf{x}) + g(\mathbf{y}) \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} = \mathbf{c}. \end{aligned} \tag{A.1}$$

Its augmented Lagrangian function is

$$\mathscr{L}_\lambda(\mathbf{x},\mathbf{y},\mathbf{z}) := f(\mathbf{x}) + g(\mathbf{y}) + \mathbf{z}^{\mathbf{T}}(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} - \mathbf{c}) + \frac{\lambda}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} - \mathbf{c}\|^2,$$

where $\mathbf{z}$ is a dual multiplier and $\lambda > 0$ is a parameter. The idea of ADMM is to apply the Gauss-Seidel decomposition technique to solve the above joint minimization problem of augmented Lagrangian function $\mathscr{L}_\lambda$; consequently, minimize $\mathscr{L}_\lambda$ according to variable $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$, respectively, at an alternating manner.

Algorithm 4: ADMM framework

Input: data $\mathbf{A}$, $\mathbf{B}$, $\mathbf{c}$, parameter λ.
Output: optimal solution $\mathbf{x}^*, \mathbf{y}^*$.
while *not convergent* **do**
  for $k = 0, 1, \cdots$ **do**
    $\mathbf{x}^{k+1} := \arg\min_{\mathbf{x}} \mathscr{L}_\lambda(\mathbf{x}, \mathbf{y}^k, \mathbf{z}^k)$,
    $\mathbf{y}^{k+1} := \arg\min_{\mathbf{y}} \mathscr{L}_\lambda(\mathbf{x}^{k+1}, \mathbf{y}, \mathbf{z}^k)$,
    $\mathbf{z}^{k+1} := \mathbf{z}^k + \lambda(\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{y}^{k+1} - \mathbf{c})$.
  end
end

A.3. **AMA.** The alternating minimization algorithm (AMA) [58] is a popular algorithm for solving problem (A.1). It shares a similar pattern with ADMM, except that AMA assumes f is strongly convex and update the first block with $\lambda = 0$.

Algorithm 5: AMA framework

Input: data $\mathbf{A}$, $\mathbf{B}$, $\mathbf{c}$, parameter λ.
Output: optimal solution $\mathbf{x}^*, \mathbf{y}^*$.
while *not convergent* **do**
  for $k = 0, 1, \cdots$ **do**
    $\mathbf{x}^{k+1} := \arg\min_{\mathbf{x}} \mathscr{L}_0(\mathbf{x}, \mathbf{y}^k, \mathbf{z}^k)$,
    $\mathbf{y}^{k+1} := \arg\min_{\mathbf{y}} \mathscr{L}_\lambda(\mathbf{x}^{k+1}, \mathbf{y}, \mathbf{z}^k)$,
    $\mathbf{z}^{k+1} := \mathbf{z}^k + \lambda(\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{y}^{k+1} - \mathbf{c})$.
  end
end

A.4. **MM.** The majorization-minimization (MM) method [29] is a popular algorithm for solving nonconvex optimization problems. Consider a general (nonconvex) optimization problem

$$\min_{\mathbf{x}} f(\mathbf{x}).$$

The idea of MM is to use a series of (convex) surrogate functions $\{g_k\}$ satisfying

$$g_k(\mathbf{x}_k) = f(\mathbf{x}_k) \quad \text{and} \quad g_k(\mathbf{x}) \geq f(\mathbf{x}) \quad \text{for each } \mathbf{x} \in \mathbb{R}^n$$

(minimizing g_k is much easier than f). At each iteration, we minimize the surrogate function g_k in place of f, namely,

$$\mathbf{x}^{k+1} := \arg\min_{\mathbf{x}} g_k(\mathbf{x}),$$

and hence the descent property of the objective function is guaranteed as

$$f(\mathbf{x}^{k+1}) \leq g_k(\mathbf{x}^{k+1}) \leq g_k(\mathbf{x}^k) = f(\mathbf{x}^k).$$

Algorithm 6: MM framework

Input: surrogate functions g_k.
Output: optimal solution $\mathbf{x}^*$.
Find a feasible point $\mathbf{x}^0$.
while *not convergent* **do**
  for $k = 0, 1, \cdots,$ **do**
    $\mathbf{x}^{k+1} := \arg\min_{\mathbf{x}} g_k(\mathbf{x})$.
  end
end

A.5. **PGM.** The proximal gradient method (PGM) [4] is a famous algorithm for solving (possibly nonsmooth and nonconvex) composite optimization problem and has been extensively applied in machine learning and image science. Consider a composite optimization problem

$$\min_{\mathbf{x}} f(\mathbf{x}) + \varphi(\mathbf{x}), \tag{A.2}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is convex and continuously differentiable, and $\varphi : \mathbb{R}^n \to \mathbb{R}$ is possibly nonconvex and nonsmooth. The idea of PGM is to blend the gradient descent method and proximal point method; that is, employ the gradient decent operator to minimize the smooth function f, and then apply the proximal operator to minimize the nonconvex and nonsmooth g at each iteration. Consequently, the framework of PGM is stated as follows.

Algorithm 7: PGM framework

Input: parameter t.
Output: optimal solution $\mathbf{x}^*$.
Initialize $\mathbf{x}^0$.
while *not convergent* **do**
  for $k = 0, 1, \cdots,$ **do**
    $\mathbf{y}^{k+1} := \mathbf{x}^k - t\nabla f(\mathbf{x}^k)$,
    $\mathbf{x}^{k+1} := \mathrm{prox}_{t\varphi}(\mathbf{y}^{k+1}) = \arg\min_{\mathbf{x}} \varphi(\mathbf{x}) + \frac{1}{2t}\|\mathbf{x} - \mathbf{y}^{k+1}\|^2$.
  end
end

The proximal operator $\mathrm{prox}_{t\varphi}(\cdot)$ is still an optimization subproblem. Fortunately, it has analytical solutions for certain functions φ which are widely used in sparse optimization, such as ℓ_p norm ($p = 0, \frac{1}{2}, \frac{2}{3}, 1$), $\ell_{2,p}$ norm ($p = 0, \frac{1}{2}, \frac{2}{3}, 1$) and nuclear norm; one can refer to [24] for details.

In fact, PGM can be understood as a version of MM for solving problem (A.2) with the surrogate function being the second-order Taylor expansion of f at point $\mathbf{x}^k$ plus φ, i.e.,

$$g_k(\mathbf{x}) := f(\mathbf{x}^k) + \langle \nabla f(\mathbf{x}^k)), \mathbf{x} - \mathbf{x}^k \rangle + \frac{1}{2t}\|\mathbf{x} - \mathbf{x}^k\|^2 + \varphi(\mathbf{x}) \text{ with } t < \sup_{\mathbf{x}} \|\nabla^2 f(\mathbf{x}))\|.$$

REFERENCES

[1] M. Agueh and G. Carlier. Barycenters in the Wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2):904–924, 2011.

[2] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh. Clustering with Bregman divergences. *Journal of Machine Learning Research*, 6(58):1705–1749, 2005.

[3] E. R. Barnes. An algorithm for partitioning the nodes of a graph. *SIAM Journal on Algebraic Discrete Methods*, 3(4):541–550, 1982.

[4] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[5] A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik. Support vector clustering. *Journal of Machine Learning Research*, 2:125–137, 2001.

[6] A. Ben-Israel and C. Iyigun. Probabilistic d-clustering. *Journal of Classification*, 25(5):5–26, 2008.

[7] D. P. Bertsekas. *Convex Optimization and Algorithms*. Athena Scientific, Massachusetts, 2015.

[8] S. Boyd, N. Parikh, and E. Chu. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[9] D. Cai, X. He, and J. Han. Document clustering using locality preserving indexing. *IEEE Transactions on Knowledge and Data Engineering*, 17(12):1624–1637, 2005.

[10] G. Carlsson and F. Mémoli. Characterization, stability and convergence of hierarchical clustering methods. *Journal of Machine Learning Research*, 11(47):1425–1470, 2010.

[11] E. C. Chi and K. Lange. Splitting methods for convex clustering. *Journal of Computational and Graphical Statistics*, 24(4):994–1013, 2015.

[12] F. R. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

[13] M. Cuturi and A. Doucet. Fast computation of Wasserstein barycenters. *Proceedings of Machine Learning Research*, 32:685–693, 2014.

[14] I. S. Dhillon, S. Mallela, and R. Kumar. A divisive information-theoretic feature clustering algorithm for text classification. *Journal of Machine Learning Research*, 3:1265–1287, 2003.

[15] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2765–2781, 2013.

[16] J. Fan and T. W. Chow. Sparse subspace clustering for data with missing entries and high-rank matrix completion. *Neural Networks*, 93:36–44, 2017.

[17] C. Fraley and A. E. Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631, 2002.

[18] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, 2007.

[19] L. Fu, P. Lin, A. V. Vasilakos, and S. Wang. An overview of recent multi-view clustering. *Neurocomputing*, 402:148–161, 2020.

[20] S. Geman and D. E. McClure. Statistical methods for tomographic image reconstruction. *Bulletin of the International Statistical Institute*, 52(4):5–21, 1987.

[21] L. Hagen and A. B. Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 11(9):1074–1085, 1992.

[22] N. Ho, X. Nguyen, M. Yurochkin, H. H. Bui, V. Huynh, and D. Phung. Multilevel clustering via Wasserstein means. *Proceedings of the 34th International Conference on Machine Learning*, pp. 1501–1509, 2017.

[23] T. D. Hocking, A. Joulin, F. Bach, and J.-P. Vert. Clusterpath an algorithm for clustering using convex fusion penalties. *Proceedings of the 28th International Conference on Machine Learning*, pp. 746–752, 2011.

[24] Y. Hu, C. Li, K. Meng, J. Qin, and X. Yang. Group sparse optimization via $\ell_{p,q}$ regularization. *Journal of Machine Learning Research*, 18(30):1–52, 2017.

[25] A. K. Jain. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8):651 –666, 2010.

[26] L. V. Kantorovich. On the translocation of masses. *Comptes Rendus (Doklady) de l'Academie des Sciences de l'URSS*, 37:199–201, 1942.

[27] H. Kashima, J. Hu, B. Ray, and M. Singh. K-means clustering of proportional data using L1 distance. *Proceedings of the 19th International Conference on Pattern Recognition*, pp. 1–4, 2008.

[28] H.-P. Kriegel, P. Kröger, J. Sander, and A. Zimek. Density-based clustering. *WIREs Data Mining and Knowledge Discovery*, 1(3):231–240, 2011.

[29] K. Lange, D. R. Hunter, and I. Yang. Optimization transfer using surrogate objective functions. *Journal of Computational and Graphical Statistics*, 9(1):1–20, 2000.

[30] J. Li and J. Z. Wang. Real-time computerized annotation of pictures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(6):985–1002, 2008.

[31] Y. Linde, A. Buzo, and R. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28(1):84–95, 1980.

[32] F. Lindsten, H. Ohlsson, and L. Ljung. Clustering using sum-of-norms regularization: With application to particle filter output computation. *Proceedings of the 2011 IEEE Statistical Signal Processing Workshop*, pp. 201–204, 2011.

[33] X. Liu, M. Ng, R. Zhang, and Z. Zhang. A new continuous optimization model for spectral clustering. *Mathematica Numerica Sinica*, 40(4):354–366, 2018.

[34] S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[35] E. W. Ma and T. W. Chow. A new shifting grid clustering algorithm. *Pattern Recognition*, 37(3):503–514, 2004.

[36] J. MacQueen. Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297, 1967.

[37] M. Mahajan, P. Nimbhorkar, and K. Varadarajan. The planar k-means problem is NP-hard. *Theoretical Computer Science*, 442:13–21, 2012.

[38] L. Mi, W. Zhang, X. Gu, and Y. Wang. Variational Wasserstein clustering. *Proceedings of European Conference on Computer Vision*, pp. 336–352, 2018.

[39] G. Monge. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Academie Royale des Sciences de Paris*, 1781.

[40] O. Nasraoui and C.-E. Ben N'Cir. *Clustering Methods for Big Data Analytics*. Springer, Switzerland, 2019.

[41] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, pp. 849–856, 2002.

[42] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 1999.

[43] A. Panahi, D. Dubhashi, F. D. Johansson, and C. Bhattacharyya. Clustering by sum of norms: Stochastic incremental algorithm, convergence and cluster recovery. *Proceedings of the 34th International Conference on Machine Learning*, 70:2769–2777, 2017.

[44] V. M. Panaretos and Y. Zemel. Statistical aspects of Wasserstein distances. *Annual Review of Statistics and Its Application*, 6(1):405–431, 2019.

[45] V. M. Patel, H. Van Nguyen, and R. Vidal. Latent space sparse and low-rank subspace clustering. *IEEE Journal of Selected Topics in Signal Processing*, 9(4):691–701, 2015.

[46] G. Peyré and M. Cuturi, M. (2017). *Computational optimal transport*. arXiv:1610.06519, 2017.

[47] R. Polyak. Regularized Newton method for unconstrained convex optimization. *Mathematical Programming*, 120:125–145, 2009.

[48] A. Rodriguez and A. Laio. Clustering by fast search and find of density peaks. *Science*, 344(6191):1492–1496, 2014.

[49] S. A. Shah and V. Koltun. Robust continuous clustering. *Proceedings of the National Academy of Sciences*, 114(37):9814–9819, 2017.

[50] G. Sheikholeslami, S. Chatterjee, and A. Zhang. Wavecluster: A multi-resolution clustering approach for very large spatial databases. *The VLDB Journal*, 8:289–304, 2000.

[51] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[52] M. Soltanolkotabi, E. Elhamifar, and E. J. Candes. Robust subspace clustering. *The Annals of Statistics*, 42(2):669–699, 2014.

[53] A. Strehl and J. Ghosh. Cluster ensembles – A knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2002.

[54] D. Sun, K.-C. Toh, and Y. Yuan. Convex clustering: Model, theoretical guarantee and efficient algorithm. *arXiv*, 1810.02677, 2018.

[55] K. M. Tan and D. Witten. Statistical properties of convex clustering. *Electronic Journal of Statistics*, 9:2324–2347, 2015.

[56] M. Teboulle. A unified continuous optimization framework for center-based clustering methods. *Journal of Machine Learning Research*, 8:65–102, 2007.

[57] J. Tian, N. Hu, T. Kwong, and Y. Y. Tang. Clustering by adaptive graph shrinking. *Proceedings of the 2019 International Conference on Wavelet Analysis and Pattern Recognition*, pp. 1–6, 2019.

[58] P. Tseng. Applications of a splitting algorithm to decomposition in convex programming and variational inequalities. *SIAM Journal on Control and Optimization*, 29(1):119–138, 1991.

[59] I. Verdinelli and L. Wasserman. Hybrid Wasserstein distance and fast distribution clustering. *Electronic Journal of Statistics*, 13(2):5088–5119, 2019.

[60] R. Vidal. Subspace clustering. *Signal Processing Magazine*, 28(2):52–68, 2011.

[61] N. X. Vinh, J. Epps, and J. Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11(95):2837–2854, 2010.

[62] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17:395–416, 2007.

[63] S. J. Wright. Coordinate descent algorithms. *Mathematical Programming*, 151:3–34, 2015.

[64] Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1101–1113, 1993.

[65] J. Xu and K. Lange. Power k-means clustering. *Proceedings of the 36th International Conference on Machine Learning*, 97:6921–6931, 2019.

[66] R. Xu and D. Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678, 2005.

[67] Z. Xue, J. Du, D. Du, and S. Lyu. Deep low-rank subspace ensemble for multi-view clustering. *Information Sciences*, 482:210–227, 2019.

[68] J. Ye, P. Wu, J. Z. Wang, and J. Li. Fast discrete distribution clustering using Wasserstein barycenter with sparse support. *IEEE Transactions on Signal Processing*, 65(9):2317–2332, 2017.

[69] S. Zhong and J. Ghosh. A unified framework for model-based clustering. *Journal of Machine Learning Research*, 4:1001–1037, 2003.