A PAM method for computing Wasserstein barycenter with unknown supports in D2-clustering

Yitian Qian* and Shaohua Pan†

October 5, 2018

Abstract

A Wasserstein barycenter is the centroid of a collection of discrete probability distributions that minimizes the average of the $\ell_2$-Wasserstein distance. This paper concerns with the computation of a Wasserstein barycenter under the case where the support points are not pre-specified, which is known to be a severe bottleneck in the D2-clustering due to the large-scale and nonconvexity. We develop a proximal alternating minimization (PAM) method for computing an approximate Wasserstein barycenter, and provide its global convergence analysis. This method can achieve a good accuracy at a reduced computational cost when the unknown support points of the barycenter have low cardinality. Numerical comparisons with the existing representative method on synthetic and real data show that our method can yield a little better objective values within much less computing time, and the computed approximate barycenter renders a better role in the D2-clustering.

Keywords: Wasserstein barycenter; PAM; semi-proximal ADMM; KL property

1 Introduction

The discrete probability distribution is a well-adopted and succinct way to summarize a batch of data, which often serves as a descriptor for complex instances in machine learning such as images, sequences, and documents. In a host of research areas, including multimedia retrieval and document analysis, the celebrated bag of “words” data model is intrinsically a discrete distribution. The widely used normalized histogram is also a special case of discrete distributions with a fixed set of support points across the instances.

A well-accepted criterion for clustering discrete probability distributions is to minimize total within-cluster variation under the Wasserstein distance (also known as the Mallows distance [15] or the earth mover’s distance [21]), similarly as Lloyd’s K-means for

*School of Mathematics, South China University of Technology, Guangzhou
†Corresponding author(shhpan@scut.edu.cn), School of Mathematics, South China University of Technology, Guangzhou, China.
vectors under the Euclidean distance. This clustering problem was originally explored by Li and Wang [14] who coined the phrase D2-clustering. Motivations for using the Wasserstein distance in practice are strongly argued by researchers (see, e.g., [14, 7, 8, 17]), and its theoretical significance is well supported in the optimal transport literature [26]. In the D2-clustering framework, the Wasserstein barycenter, which is the centroid for discrete probability distributions under the $\ell_2$-Wasserstein distance, is solved for the case of an unknown support with a pre-given cardinality. Now we are faced with a large-scale nonconvex optimization problem in which a coupled nonconvex objective function is minimized over a polyhedral constraint set. Although the D2-clustering holds much promise because of the advantages of the Wasserstein distance and its kinship with K-means, the high computational complexity of barycenter has limited its applications.

To scale up the computation of Wasserstein barycenter in the D2-clustering, a divide-and-conquer approach has been proposed in [32], but the method is ad-hoc and lack of convergence guarantee. When an alternating minimization strategy is used, the computation of Wasserstein barycenter is decomposed into a quadratic program with a closed form solution and a linear program (LP) with a super-linear time-complexity on the number of samples $N$. The latter brings a big challenge for the computation of a barycenter because the number of variables in the LP grows quickly with the cluster size, and the classical LP solvers such as the simplex method and the interior point method are not scalable. To overcome this challenge, Ye and Li [30] applied the classical alternating direction method of multipliers (ADMM) for solving this LP, and Wang and Banerjee [27] generalized the classical ADMM to the Bregman ADMM (B-ADMM) by replacing the quadratic distance with a general Bregman distance so as to exploit the structure of the LP. However, they did not provide the global convergence analysis for the outer alternating minimization method. Recently, Ye et al. [31] proposed a three-block B-ADMM for computing a Wasserstein barycenter directly. Although this method has demonstrated a computational efficiency for large-scale data, it is still unclear whether it is globally convergent or not. In fact, for convex programs, it has been shown in [6] that the direct extension of the classical ADMM to the three-block case can be divergent. Just recently, Yang et al. [29] propose a very efficient dual solver for the LP by adopting a symmetric Gauss-Seidel based ADMM, but they did not study the performance of an alternating minimization method with such a subroutine for computing barycenter.

The main contribution of this work is to develop a globally convergent and efficient PAM method for computing an approximate Wasserstein barycenter when the support points are not pre-specified. Since a PAM strategy is adopted, at each iteration there involves two strongly convex quadratic programs (QPs). One of them has a closed form solution, and the other has a polyhedral constraint set and may be good-conditioned by controlling the proximal parameter elaborately. Such strongly convex QPs have much better stability than those LPs appearing in [30, 27], which means that their solutions are much easier to achieve. In Section 5.1 we propose a tailored semi-proximal ADMM for solving the strongly convex QP by exploiting the special structure of the feasible set. We notice that the B-ADMM in [27] also belongs to this line since at each iteration it decomposes the LP into two simple strongly convex Kullback-Leibler (KL) minimization
to solve. However, compared with the B-ADMM, the semi-proximal ADMM not only has a global convergence [9] but also admits the well-established linear rate of convergence [12] and weighted iteration complexity [22]. Numerical comparisons with the three-block B-ADMM in [31] on synthetic data and several classes of real data show that our PAM method yields a little better objective values within much less computing time, and the computed approximate barycenter renders a better role in the D2-clustering.

For a nonconvex and nonsmooth optimization problem, without additional conditions imposed on the problem, the convergence result of an alternating minimization method and more general block coordinate descent methods (see, e.g., [24, 25, 18, 13]) is typically limited to the objective convergence (to a possibly non-minimal value) or the convergence of a certain subsequence of iterates to a critical point. Motivated by the recent excellent works [1, 2, 4], we achieve the global convergence of our PAM method for computing a Wasserstein barycenter by using the Kurdyka-Lojasiewicz (KL) property of the extended objective function. It is worthwhile to point out that under the KL assumption Xu and Yin [28] also developed a globally convergent algorithm based on block coordinate update for a class of nonconvex and nonsmooth optimization problems.

The rest of this paper is organized as follows. Section 2 presents some notations and preliminary knowledge that will be used in this paper. In Section 3, we introduce the optimization model for computing a Wasserstein barycenter when its support points are unknown and propose a PAM method for solving it. Section 4 focuses on the convergence analysis of the PAM. In Section 5, we provide the implementation details of the PAM and compare its performance with that of the B-ADMM [31] on synthetic and real data.

2 Notations and preliminaries

Throughout this paper, $\mathbb{R}^n$ denotes the n-dimensional Euclidean space, $\mathbb{R}_+^n$ is the non-negative orthant cone in $\mathbb{R}^n$, $\mathbb{R}^{m \times n}$ represents the vector space consisting of all $m \times n$ real matrices, equipped with the trace inner product $\langle \cdot, \cdot \rangle$ and its induced Frobenius norm $\| \cdot \|_F$, i.e., $\langle X, Y \rangle = \text{tr}(X^T Y)$ for $X, Y \in \mathbb{R}^{m \times n}$, and $\mathbb{R}^{m \times n}_+$ denotes the polyhedral cone consisting of all $m \times n$ nonnegative real matrices. For a vector $x \in \mathbb{R}^n$, $\| x \|$ and $\| x \|_1$ denote the $\ell_2$-norm and $\ell_1$-norm of $x$, respectively. We use $e$ to denote a column vector of all ones whose dimension is known from the context. For a given set $S$, $\delta_S$ denotes the indicator function on $S$, i.e., $\delta_S(u) = 0$ if $u \in S$, and otherwise $\delta_S(u) = +\infty$; when $S$ is convex, $\mathcal{N}_S(x)$ denotes the normal cone of $S$ at $x$ in the sense of convex analysis [19].

In the rest of this section, the notation $X$ to denote a finite dimensional vector space equipped with the inner product $\langle \cdot, \cdot \rangle$ and its induced norm $\| \cdot \|$. First, we recall from [20] the concept of limit subdifferential for an extended real-valued function.

2.1 Limiting subdifferential

**Definition 2.1** Consider a function $h: X \to (-\infty, +\infty]$ and a point $\tau$ with $h(\tau)$ finite.
(i) A vector \( v \) is said to be a Fréchet subgradient of \( h \) at \( x \), written by \( v \in \partial h(x) \), if
\[
h(x) \geq h(x) + \langle v, x - x \rangle + o(\|x - x\|).
\]

(ii) A vector \( v \in \mathbb{X} \) is said to be a limiting subgradient of \( h \) at \( x \), written by \( v \in \partial h(x) \),
if there are sequences \( x^k \to x \) with \( h(x^k) \to h(x) \) and \( v^k \in \partial h(x^k) \) with \( v^k \to v \).

**Remark 2.1** (i) The sets \( \partial h(x) \) and \( \partial h(x) \) are respectively called the Fréchet subdifferential and the limiting subdifferential of \( h \) at \( x \). Notice that \( \partial h(x) \subseteq \partial h(x) \) and they are always closed, and moreover, \( \partial h(x) \) is always convex. When \( h \) is convex, \( \partial h(x) = \partial h(x) \), which also coincides with the subdifferential of \( h \) at \( x \) in the sense of convex analysis.

(ii) Let \( \{(x^k, \xi^k)\} \) be a sequence in the graph of the set-valued mapping \( \partial h : \mathbb{X} \rightrightarrows \mathbb{X} \), which converges to \( (x, \xi) \). If \( h(x^k) \to h(x) \) as \( k \to \infty \), then \( (x, \xi) \in \text{gph} \partial h \).

(iii) By [20, Theorem 10.1], a necessary (but not sufficient) condition for \( \partial h(x) \subseteq \partial h(x) \), if there are sequences \( x^k \to x \) such that \( \xi^k \) satisfies \( 0 \in \partial h(x^k) \) is called a (limiting) critical point of \( h \). The set of critical points of \( h \) is denoted by \( \text{crit} h \).

For an optimization problem with a closed convex feasible set, it is common to consider its directional stationary point, which is defined as follows.

**Definition 2.2** Let \( \psi : \mathbb{X} \to \mathbb{R} \) be a directionally differentiable function, and \( S \subseteq \mathbb{X} \) be a closed convex set. Consider a point \( x \in S \). If for every \( x \in S \), \( \psi'(x; x - x) \geq 0 \), then \( x \) is called a directional stationary point of the minimization problem \( \min_{x \in S} \psi(x) \).

**Lemma 2.1** Consider the minimization problem \( \min_{x \in S} \psi(x) \) where \( S \subseteq \mathbb{X} \) is a closed convex set. Suppose that \( \psi \) is directionally differentiable and locally Lipschitz at \( x \in S \). Then, \( \hat{x} \) is a directional stationary point of this problem if and only if \( 0 \in \partial \psi(x) + \mathcal{N}(S, \hat{x}) \).

**Proof:** Let \( \psi'(\hat{x}) : \mathbb{X} \to [-\infty, +\infty] \) denote the subderivative function of \( \psi \) at \( \hat{x} \) (see [20, Definition 8.1]). Suppose that \( \hat{x} \) is a directional stationary point. From the directional differentiability and locally Lipschitz of \( \psi \) at \( \hat{x} \) and [20, Exercise 8.4], it follows that
\[
\sup_{z \in \partial \psi'(\hat{x})} \langle z, x - \hat{x} \rangle = \psi'(\hat{x})(x - \hat{x}) = \psi'(x; x - \hat{x}) \geq 0 \quad \forall x \in S.
\]

Notice that \( \partial \psi'(\hat{x}) \) is bounded since \( \psi \) is locally Lipschitz at \( \hat{x} \). From the last inequality and the compactness of \( \partial \psi'(\hat{x}) \), there exists \( v \in \partial \psi'(\hat{x}) \) such that \( \langle v, x - \hat{x} \rangle \geq 0 \) for all \( x \in S \), i.e., \( -v \in \mathcal{N}(S, \hat{x}) \). Thus, we have \( 0 \in \partial \psi'(\hat{x}) + \mathcal{N}(S, \hat{x}) \). Conversely, suppose that \( 0 \in \partial \psi'(\hat{x}) + \mathcal{N}(S, \hat{x}) \). Then there exists \( \xi \in \mathcal{N}(S, \hat{x}) \) such that \( -\xi \in \partial \psi'(\hat{x}) \). Fix an arbitrary \( x \in S \). Since \( \xi \in \mathcal{N}(S, \hat{x}) \), we have \( \xi, x - \hat{x} \leq 0 \). Together with [20, Exercise 8.4],
\[
\psi'(\hat{x}; x - \hat{x}) \geq d \psi'(\hat{x})(x - \hat{x}) \geq \langle -\xi, x - \hat{x} \rangle \geq 0.
\]

By the arbitrariness of \( x \in S \), this shows that \( \hat{x} \) is a directional stationary point.

**Remark 2.2** Lemma [20, Exercise 8.4] implies that, when \( \psi \) is continuously differentiable, \( \text{crit} (\psi + \delta_S) \) coincides with the set of directional stationary points for the problem \( \min_{x \in S} \psi(x) \).
2.2 Kurdyka-Lojasiewicz property

Definition 2.3 Let $h : \mathbb{X} \to (-\infty, +\infty]$ be a proper lower semicontinuous (lsc) function. The function $h$ is said to have the Kurdyka-Lojasiewicz (KL) property at $x \in \text{dom} \partial h$ if there exist $\eta \in (0, +\infty]$, a continuous concave function $\varphi : [0, \eta] \to \mathbb{R}_+$ satisfying

(i) $\varphi(0) = 0$ and $\varphi$ is continuously differentiable on $(0, \eta)$,

(ii) for all $s \in (0, \eta)$, $\varphi'(s) > 0$

and a neighborhood $U$ of $x$ such that for all $x \in U \cap [h(x) < h(\pi) < h(x) + \eta]$

$$\varphi'(h(x) - h(\pi)) \text{dist}(0, \partial h(x)) \geq 1.$$ 

If $h$ satisfies the KL property at each point of $\text{dom} \partial h$, then $h$ is called a KL function.

Remark 2.3 By Definition 2.3 and [1, Lemma 2.1], a proper lsc function has the KL property at every noncritical point. Thus, to show that a proper lsc $h : \mathbb{X} \to (-\infty, +\infty]$ is a KL function, it suffices to check that $h$ has the KL property at any critical point.

As discussed in [1, Section 4], many classes of functions are the KL function; for example, the semialgebraic function. A function $h : \mathbb{R}^n \to (-\infty, +\infty]$ is semialgebraic if its graph is a semialgebraic subset of $\mathbb{R}^{n+1}$. Recall that a subset of $\mathbb{R}^n$ is called semialgebraic if it can be written as a finite union of sets of the form

$$\Omega = \bigcup_{i=1}^{p} \bigcap_{j=1}^{q} \{ x \in \mathbb{R}^n : f_{ij}(x) = 0, g_{ij}(x) > 0 \}$$

where $f_{ij} : \mathbb{R}^n \to \mathbb{R}$ and $g_{ij} : \mathbb{R}^n \to \mathbb{R}$ are polynomial functions for all $1 \leq i \leq p, 1 \leq j \leq q$.

3 PAM method for computing Wasserstein barycenter

In this section we develop a PAM method for computing Wasserstein barycenter, which marks the main difference between D2-clustering and K-means. To this end, we first introduce the Wasserstein barycenter in discrete probability distribution clustering.

3.1 Wasserstein barycenter in discrete distribution clustering

Consider discrete probability distributions with finite supports specified by a set of support points and their associated probabilities $\{(x_1, w_1), \ldots, (x_m, w_m)\}$, where $x_i \in \mathbb{R}^d$ for $i = 1, 2, \ldots, m$ are support vectors, and $w = (w_1, \ldots, w_m)^T$ is the probability vector, i.e., $w \in \Delta := \{ z \in \mathbb{R}_+^m \mid \sum_{i=1}^{m} z_i = 1 \}$. Let $P^\pi = \{(x^\pi_i, w^\pi_i), i = 1, 2, \ldots, m_\pi\}$ and $P^\nu = \{(x^\nu_j, w^\nu_j), j = 1, 2, \ldots, m_\nu\}$ be two given discrete probability distributions. The
\( \ell_2 \)-Wasserstein distance between the distributions \( P^\pi \) and \( P^\nu \), denoted by \( W(\ell_2, P^\pi, P^\nu) \), is the optimal value of the following linear programming problem

\[
\min_{Z_{ij} \geq 0} \left( \sum_{i=1}^{m_\pi} \sum_{j=1}^{m_\nu} Z_{ij} \| x^i_\pi - x^j_\nu \|^2 \right)^{1/2} \\
\text{s.t.} \sum_{j=1}^{m_\nu} Z_{ij} = \pi_i, \quad i = 1, 2, \ldots, m_\pi; \\
\sum_{i=1}^{m_\pi} Z_{ij} = \nu_j, \quad j = 1, 2, \ldots, m_\nu,
\]

while an optimal solution \( \{ Z_{ij}^* \} \) of the problem (1) is called the optimal matching weights between support points \( x^i_\pi \) and \( x^j_\nu \) (or the optimal coupling for \( P^\pi \) and \( P^\nu \)).

For a given set of discrete distributions \( \{ P^t, t = 1, 2, \ldots, N \} \) and the number of clusters \( K \), where \( P^t = \{ (a^t_j, b^t_j) \in \mathbb{R}^d \times \mathbb{R}, j = 1, \ldots, n^t \} \), the goal of D2-clustering is to seek a set of centroid distributions \( Q^* = \{ Q^s, s = 1, 2, \ldots, K \} \) such that

\[
Q^* \in \arg\min_{Q^1, \ldots, Q^K} \sum_{t=1}^{N} \min_{s \in \{1, \ldots, K\}} W^2(Q^s, P^t) \tag{2}
\]

where \( Q^s = \{ (x^s_i, w^s_i) \in \mathbb{R}^d \times \mathbb{R}, i = 1, \ldots, m \} \) for \( s = 1, 2, \ldots, K \). Similarly as in \( K \)-means, D2-clustering also alternates the optimization of the centroids and the assignment of each instance to the nearest centroid, whose iteration steps are described as follows.

**Algorithm 1 (D2-Clustering)**

**Initialize:** Initialize the set of centroids \( \{ Q^{1,0}, Q^{2,0}, \ldots, Q^{K,0} \} \).

For \( k = 1, 2, \ldots \) do

1. for \( t = 1, 2, \ldots, N \) do (assignment step)

   \[
   l^{t,k} := \arg\min_{s \in \{1, \ldots, K\}} W^2(Q^{s,k-1}, P^t) \tag{3}
   \]

   end for

2. for \( s = 1, 2, \ldots, K \) do (optimization step)

   \[
   Q^{s,k} \in \arg\min_{Q} \sum_{t: l^t = s} W^2(Q, P^t) \tag{4}
   \]

   end for

end For

Return the index set \( \{ l^{1,k}, \ldots, l^{N,k} \} \) and the set of centroids \( \{ Q^{1,k}, \ldots, Q^{K,k} \} \).
The major computation challenge in Algorithm 1 is to compute an optimal centroid distribution, called Wasserstein barycenter, for each cluster at each iteration. Different from $K$-means, the optimal centroid distribution in (4) does not have a closed form. In fact, it is intractable since as will be shown later, the problem (4) is actually a nonconvex program in which the number of decision variables $m \sum_{t=1}^{N} n_t$ quickly becomes very large even for a rather small number of distributions containing only 10 support points each.

3.2 PAM method for computing the centroid distribution

Suppose that we have a set of discrete probability distributions $\{P^t : t = 1, 2, \ldots, N\}$, where $P^t = \{(a^t_j, b^t_j) \in \mathbb{R}^d \times \mathbb{R}, j = 1, \ldots, n_t\}$ for $t = 1, \ldots, N$ and $N$ is the sample size for computing a Wasserstein barycenter. The problem (4) is to find an optimal centroid $Q^*$ with respect to the support points and the associated probabilities among all discrete probability distributions $Q = \{(x_1, w_1), \ldots, (x_m, w_m)\}$ such that

$$Q^* \in \arg\min_{Q} \frac{1}{N} \sum_{t=1}^{N} W^2(Q, P^t).$$

For each $t \in \{1, 2, \ldots, N\}$, let $b^t = (b^t_1, \ldots, b^t_{n_t})^T \in \mathbb{R}^{n_t}$ and define $F^t : \mathbb{R}^{md} \rightarrow \mathbb{R}^{m \times n_t}$ by

$$[F^t(x)]_{ij} := \|x_i - a^t_j\|^2 \quad \forall x \in \mathbb{R}^{md}.$$

Write $F(x) := [F^1(x), F^2(x), \ldots, F^N(x)] \in \mathbb{R}^{m \times n}$ and $Z = [Z^1, Z^2, \ldots, Z^N] \in \mathbb{R}^{m \times n}$ with $n = \sum_{t=1}^{N} n_t$. The optimization problem (6) actually takes the following form

$$\min_{Z \in \mathbb{R}^{m \times n}, w \in \mathbb{R}^m, x \in \mathbb{R}^{md}} (Z, N^{-1}F(x))$$

subject to

$$Z^t e - w = 0, \quad t = 1, 2, \ldots, N,$$

$$(Z^t)^T e - b^t = 0, \quad t = 1, 2, \ldots, N,$$

$Z \in \mathbb{R}^{m \times n}, w \in \Delta.$$

The LP solvers developed in [27, 30, 29] are solving the problem (6) with a fixed $x$.

For each $t \in \{1, 2, \ldots, N\}$, let $\Sigma_t := \{Y \in \mathbb{R}^{m \times n_t} | Y^T e = b^t\}$. By using the indicator functions $\delta_{\Sigma_t}$ and $\delta_{\Delta}$, the problem (6) can be written in the following compact form

$$\min_{Z \in \mathbb{R}^{m \times n}, w \in \mathbb{R}^m, x \in \mathbb{R}^{md}} \sum_{t=1}^{N} \left[ N^{-1}\langle Z^t, F^t(x) \rangle + \delta_{\Sigma_t}(Z^t) \right] + \delta_{\Delta}(w)$$

subject to

$$Z^t e - w = 0, \quad t = 1, 2, \ldots, N.$$

The problem (6) or (7) has a nonconvex objective function although its feasible set is convex. Hence, generally one can only expect to obtain a desirable stationary point. The objective function of (7) has a desirable coupled structure, that is, when one of the decision variables $x$ and $Z$ is fixed, it becomes a solvable convex program. Motivated by this, we solve the problem (7) in an alternating way. The iterations are as follows.
Algorithm 3.1 (PAM method for solving (7))

(S.0) Choose $\alpha_0 > \alpha > 0$, $\rho_0 > \rho > 0$ and an starting point $(Z^0, w^0, x^0)$. Set $k := 0$.

(S.1) Compute the linear constrained strongly convex quadratic program

$$
(Z^{k+1}, w^{k+1}) = \arg \min_{Z, w} \left\{ \sum_{t=1}^{N} \left[ N^{-1}(Z^t, F^t(x^k)) + \delta_{\Sigma_t}(Z^t) \right] + \delta_\Delta(w) \\
+ \frac{\alpha_k}{2} \left[ \|Z - Z^k\|_F^2 + \|w - w^k\|_2^2 \right] \right\}
\text{s.t. } Z^t e - w = 0, \quad t = 1, 2, \ldots, N.
$$

(S.2) Compute the unconstrained strongly convex quadratic program

$$
x^{k+1} = \arg \min_{x \in \mathbb{R}^{md}} \left\{ \frac{1}{N} \sum_{t=1}^{N} \left[ \sum_{i=1}^{m} \sum_{j=1}^{n_t} Z^{k+1}_{ij} \|x_i - a^t_j\|^2 \right] + \frac{\rho_k}{2} \|x - x^k\|_2^2 \right\}.
$$

(S.3) Choose $\alpha_{k+1} \in [\alpha, \alpha_k]$ and $\rho_{k+1} \in [\rho, \rho_k]$. Let $k \leftarrow k + 1$, and go to (S.1).

Remark 3.1 (i) Algorithm 3.1 is well defined since each subproblem has a unique optimal solution. In particular, from the optimality condition of (9), it is easy to obtain

$$
x_i^{k+1} = \frac{2 \sum_{t=1}^{N} \sum_{j=1}^{n_t} Z^{k+1}_{ij} a^t_j + \rho_k N x_i^k}{2 \sum_{t=1}^{N} \sum_{j=1}^{n_t} Z^{k+1}_{ij} + \rho_k N} \quad \text{for } i = 1, 2, \ldots, m.
$$

This shows that the main work of Algorithm 3.1 in each iterate is the solution of (8).

(ii) Since the gradient of the objective function in (7) is not globally Lipschitz continuous, we here introduce a proximal strategy instead of a majorization technique as in [23] for each block subproblem. The quadratic proximal term $\frac{1}{2} \alpha_k (\|Z - Z^k\|_F^2 + \|w - w^k\|_2^2)$ ensures that a strongly convex QP instead of an LP subproblem is solved at each iteration. Since the former has much better stability than the latter, its solution is much easier.

(iii) If at the $(k+1)$th iteration $w^k_i = 0$ for some $i \in \{1, \ldots, m\}$, it necessarily holds that

$$
\sum_{t=1}^{N} \sum_{j=1}^{n_t} Z^{k+1}_{ij} = w^{k+1}_i = 0.
$$

From the nonnegativity of $Z^{k+1}_{ij}$, it then follows that $Z^{k+1}_{ij} = 0$ for each $j = 1, \ldots, n_t$ and $t = 1, 2, \ldots, N$. Together with formula (10), we have $x_{i}^{k+1} = x_{i}^{k}$. This means that

$$
(Z_{i}^{\nu}, w_{i}^{\nu}) = 0 \quad \text{and} \quad x_{i}^{\nu} = x_{i}^{k} \quad \text{for all } \nu \geq k + 1,
$$
where $Z_i^{t,v}$ means the $i$th row of the matrix $Z^{t,v}$, and consequently one may delete the decision variables $Z_i \in \mathbb{R}^{1 \times n}$, $w_i \in \mathbb{R}$ and $x_i \in \mathbb{R}^d$ of the problem (5).

For Algorithm 3.1, it is natural to introduce Nesterov’s acceleration technique [16], and then the iteration steps of the accelerated version are described as follows.

Algorithm 3.2 (Accelerated PAM method for solving (7))

(S.0) Choose $0 > \alpha > 0$, $\rho > 0$ and an starting point $(Z^0, w^0, x^0)$. Let $t_0 = 1$ and $\tilde{x}^0 = x^0$. Set $k := 0$.

(S.1) Compute the following strongly convex optimization problems

\[
\begin{align*}
(Z^{k+1}, w^{k+1}) &= \arg\min_{Z, w} \left\{ \sum_{t=1}^{N} \left[ N^{-1} (Z^t, F^t(\tilde{x}^k)) + \delta_{\Sigma_t}(Z^t) \right] + \delta_{\Delta}(w) \right. \\
&\quad \left. + \frac{\alpha_t}{2} \| \Sigma^{-1} \|_F^2 + \| w - w^k \|_2^2 \right\} \quad (11a) \\
\text{s.t.} & \quad Z^t e - w = 0, \quad t = 1, 2, \ldots, N; \\
x^{k+1} &= \arg\min_{x \in \mathbb{R}^{md}} \frac{1}{N} \sum_{t=1}^{N} \sum_{i=1}^{m} \sum_{j=1}^{n} Z_{ij}^t, \nu \| x_i - a_{ij}^t \|_2 + \frac{\rho_t}{2} \| x - \tilde{x}^k \|_2^2. \quad (11b)
\end{align*}
\]

(S.2) Set $t_{k+1} = \frac{1 + \sqrt{t_k^2 + 4 \rho_t}}{2}$ and $\gamma_k = \frac{t_{k+1} - 1}{t_{k+1}}$. Compute $\tilde{x}^{k+1} = x^{k+1} + \gamma_k (x^{k+1} - x^k)$.

(S.3) Choose $\alpha_{k+1} \in [\alpha, \alpha_k]$ and $\rho_{k+1} \in [\rho, \rho_k]$. Let $k \leftarrow k + 1$, and go to (S.1).

Remark 3.2 In Algorithm 3.2 the extrapolation technique is only imposed on the $x$-part instead of all decision variables as in [28], because preliminary tests indicate that the acceleration of the $(Z, w)$-part does not bring the notable improvement on Algorithm 3.1.

4 Convergence analysis of Algorithm 3.1

In this section, by following the analysis technique in [1], we establish the global convergence of the sequence $\{(Z^k, w^k, x^k)\}_{k \in \mathbb{N}}$ generated by Algorithm 3.1 to a stationary point $(Z^*, w^*, x^*)$ of the problem (9) or (11). For any $(Z, w, x) \in \mathbb{R}^{m \times n} \times \mathbb{R}^n \times \mathbb{R}^{md}$, define

\[
f(Z, x) := \frac{1}{N} \sum_{t=1}^{N} \langle Z^t, F^t(x) \rangle \quad \text{and} \quad g(Z, w) := \sum_{t=1}^{N} \left[ \delta_{\Sigma_t}(Z^t) + \delta_{\Gamma_t}(Z^t, w) \right] + \delta_{\Delta}(w) \quad (12)
\]

where, for each $t \in \{1, 2, \ldots, N\}$, $\Gamma_t := \{(Z^t, w) \in \mathbb{R}^{m \times n} \times \mathbb{R}^n \mid Z^t e - w = 0\}$. Clearly, the extended-valued objective function of the problem (10) or (11) takes the form of

\[
\Psi(Z, w, x) := f(Z, x) + g(Z, w) \quad \forall (Z, w, x) \in \mathbb{R}^{m \times n} \times \mathbb{R}^n \times \mathbb{R}^{md}. \quad (13)
\]
The following lemma provides a characterization for the set of stationary points \( \text{crit } \Psi \), which by Remark 2.2 is precisely the set of directional stationary points for (6).

**Lemma 4.1** The point \((\hat{Z}, \hat{\omega}, \hat{x}) \in \text{crit } \Psi \) if and only if the following conditions hold

\[
\begin{cases}
0 \in N^{-1} F(\hat{x}) + \partial g(\hat{Z}, \hat{\omega}); \\
0 \in \partial_w g(\hat{Z}, \hat{\omega}); \\
0 = 2N^{-1} \sum_{i=1}^{\alpha} \sum_{j=1}^{n_i} \hat{Z}_{ij} (\hat{x}_i - \alpha^*_j),
\end{cases}
\]

where \( \partial g(\cdot, \cdot) \) and \( \partial_w g(\cdot, \cdot) \) are the partial subdifferential of \( g \) w.r.t \( Z \) and \( w \), and

\[
\partial g(\hat{Z}, \hat{\omega}) = \partial g(\hat{Z}, \hat{\omega}) \times \partial_w g(\hat{Z}, \hat{\omega}) = \left( \begin{array}{c}
\mathcal{N}_{\Sigma_1}(\hat{Z}^1) \times \mathcal{N}_{\Delta}(\hat{\omega}) + \mathcal{N}_{t}(\hat{Z}^1, \hat{\omega}) \\
\vdots \\
\mathcal{N}_{\Sigma_N}(\hat{Z}^N) \times \mathcal{N}_{\Delta}(\hat{\omega}) + \mathcal{N}_{t}(\hat{Z}^N, \hat{\omega})
\end{array} \right).
\]

**Proof:** By the definition of \( \text{crit } \Psi \), \((\hat{Z}, \hat{\omega}, \hat{x}) \in \text{crit } \Psi \) if and only if \( 0 \in \partial g(\hat{Z}, \hat{\omega}) \). By using [20] Exercise 8.8 and the continuous differentiability of \( f \), from (13) we have

\[
\partial \Psi(\hat{Z}, \hat{\omega}, \hat{x}) = \left( \begin{array}{c}
\nabla_Z f(\hat{Z}, \hat{x}) + \partial g(\hat{Z}, \hat{\omega}) \\
\partial_w g(\hat{Z}, \hat{\omega}) \\
\nabla_x f(\hat{Z}, \hat{x})
\end{array} \right),
\]

which implies the first part of the conclusions. Notice that the problem (6) has a nonempty feasible set; for example, with \( w^0 = \frac{1}{m}e \) and \( Z^{t,0} = \frac{1}{m} [b'; \ldots; b'] \in \mathbb{R}^{m \times n_t} \) for each \( t \), \((Z^0, w^0, x)\) for any \( x \in \mathbb{R}^{md} \) is feasible. Together with the polyhedrality of \( \Sigma_t, \Gamma_t \) and \( \Delta \), from [19] Theorem 23.8 it follows that the second part of the conclusions holds.

In what follows, we study the properties of the sequence \( \{(Z^k, w^k, x^k)\} \) generated by Algorithm 3.1. First, we establish the sufficient decreasing of the sequence \( \{(Z^k, w^k, x^k)\} \).

**Lemma 4.2** Let \( \{(Z^k, w^k, x^k)\}_{k \in \mathbb{N}} \) be the sequence generated by Algorithm 3.1. Then,

(i) the sequence \( \{\Psi(Z^k, w^k, x^k)\}_{k \in \mathbb{N}} \) is nonincreasing, and moreover, for each \( k \in \mathbb{N} \),

\[
\Psi(Z^k, w^k, x^k) - \Psi(Z^{k-1}, w^{k-1}, x^{k-1}) \leq -\frac{\alpha_{k-1}}{2} \left[ \|Z^k - Z^{k-1}\|_F^2 + \|w^k - w^{k-1}\|^2 \right] - \frac{\beta_{k-1}}{2} \|x^k - x^{k-1}\|^2.
\]

(ii) \( \sum_{k=1}^{\infty} \left[ \|Z^k - Z^{k-1}\|_F^2 + \|w^k - w^{k-1}\|^2 + \|x^k - x^{k-1}\|^2 \right] < \infty \), and consequently

\[
\lim_{k \to \infty} \|Z^k - Z^{k-1}\|_F = 0, \quad \lim_{k \to \infty} \|w^k - w^{k-1}\| = 0, \quad \lim_{k \to \infty} \|x^k - x^{k-1}\| = 0.
\]
The following lemma provides a subgradient lower bound for the iterate sequence.

Remark 4.1

For the sequence generated by Algorithm 3.2, now we can not achieve the decreasing of the objective value sequence \( \Psi(Z^k, w^k, x^k) \). This hinders us to obtain its global convergence, and we will leave this for a future research topic.

For convenience, in the rest of this section, we write \( U^k := (Z^k, w^k, x^k) \) for each \( k \in \mathbb{N} \). The following lemma provides a subgradient lower bound for the iterate sequence.
Lemma 4.3 Let \( \{U^k\}_{k \in \mathbb{N}} \) be the sequence yielded by Algorithm 3.1. For each \( k \in \mathbb{N} \), let
\[
\begin{align*}
A^k_Z & := \alpha_{k-1}(Z^{k-1} - Z^k) + N^{-1}(F(x^k) - F(x^{k-1})), \\
A^k_w & := \alpha_{k-1}(w^{k-1} - w^k), \\
A^k_x & := \rho_{k-1}(x^{k-1} - x^k).
\end{align*}
\] (16a)
Then, for each \( k \in \mathbb{N} \), \( (A^k_Z, A^k_w, A^k_x) \in \partial\Psi(Z^k, w^k, x^k) \). If the level set \( \mathcal{L}_f(Z^0, x^0) \) of \( f \) is bounded, then there exists a constant \( M > \) such that with \( \hat{\alpha} := \max_{1 \leq i \leq N} \max_{1 \leq j \leq n_i} \|a^i_j\| \),
\[
\|\langle A^k_Z, A^k_w, A^k_x \rangle\| \leq \sqrt{\max(2\alpha_{k-1}^2, 8(2M + \hat{\alpha})^2, \rho_{k-1}^2)} \|U^k - U^{k-1}\| 
\] (17)
where \( \|U^k - U^{k-1}\| = \sqrt{\|Z^k - Z^{k-1}\|_F^2 + \|w^k - w^{k-1}\|^2 + \|x^k - x^{k-1}\|^2} \).
Prove: By the optimality conditions of the problems (8) and (9), it is easy to obtain
\[
\begin{align*}
\alpha_{k-1}(Z^{k-1} - Z^k) - N^{-1}(F(x^{k-1}) - F(x^k)) & \in N^{-1}F(x^k) + \partial\Psi g(Z^k, w^k), \quad (18a) \\
\alpha_{k-1}(w^{k-1} - w^k) & \in \partial g(Z^k, w^k), \quad (18b) \\
\rho_{k-1}(x^{k-1} - x^k) & = 2N^{-1} \sum_{i=1}^{N} \sum_{j=1}^{n_i} Z_{ij}^{k-1} (x_i^k - a^i_j), \quad (18c)
\end{align*}
\]
where the function \( g \) is defined by (12). Together with the expression of \( \Psi \), we have
\[ (A^k_Z, A^k_w, A^k_x) \in \partial\Psi(Z^k, w^k, x^k). \]
From the expression of \( A^k_Z \) and the relation \( \|u - v\|^2 \leq 2\|u\|^2 + 2\|v\|^2 \), it follows that
\[
\|A^k_Z\|^2_F \leq 2\alpha_{k-1}^2 \|Z^{k-1} - Z^k\|^2_F + \frac{2}{N} \|F(x^{k-1}) - F(x^k)\|^2_F,
\]
\[
= 2\alpha_{k-1}^2 \|Z^{k-1} - Z^k\|^2_F + \frac{2}{N} \sum_{i=1}^{N} \sum_{j=1}^{m} \sum_{j=1}^{n_i} (\|x_i^{k-1} - a^i_j\|^2 - \|x_i^k - a^i_j\|^2)^2. \quad (19)
\]
Since \( \{(Z^k, x^k)\} \subseteq \mathcal{L}_f(Z^0, x^0) \) and the set \( \mathcal{L}_f(Z^0, x^0) \) is bounded, there exists a constant \( M > 0 \) such that \( \|x^k\| \leq M \) for all \( k \). By the relation \( \|u + v\|^2 - \|u\|^2 - \|v\|^2 = 2\langle u, v \rangle \), for each \( i = 1, \ldots, m \) and \( j = 1, \ldots, n_i \), it holds that
\[
\|x_i^{k-1} - a^i_j\|^2 - \|x_i^k - a^i_j\|^2 \leq \|x_i^k - x_i^{k-1}\|^2 + 2(x_i^k - x_i^{k-1}, x_i^k - a^i_j) \leq 2M\|x_i^k - x_i^{k-1}\|^2 + 2(M + \|a^i_j\|)\|x_i^k - x_i^{k-1}\| \leq (4M + 2\hat{\alpha})\|x_i^k - x_i^{k-1}\|. \quad (20)
\]
Substituting the inequality (20) into (19) yields that
\[
\|A^k_Z\|^2_F \leq \max(2\alpha_{k-1}^2, 8(2M + \hat{\alpha})^2)(\|Z^{k-1} - Z^k\|^2_F + \|x^k - x^{k-1}\|^2).
\]
Together with the expressions of \( A^k_w \) and \( A^k_x \), the desired result follows.

By Lemma 4.2 and 4.3, we are in a position to establish a weak convergence result, which states the distance of \( \{U^k\}_{k \in \mathbb{N}} \) to the set of limit points approaches to 0 as \( k \to \infty \).
Proposition 4.1 Let $\{U^k\}_{k \in \mathbb{N}}$ be the sequence generated by Algorithm 3.1 and denote by $\omega(U^0)$ the set of limit points of the sequence $\{U^k\}$ starting from $U^0$. Suppose that the level set $L(f(z_0,x^0))$ of $f$ is bounded. Then, the following assertions hold.

(i) $\emptyset \neq \omega(U^0) \subset \text{crit} \Psi$, and consequently $\lim_{k \to \infty} \text{dist}(U^k, \omega(U^0)) = 0$.

(ii) The extended-valued cost function $\Psi$ is finite and constant on $\omega(U^0)$.

Proof: (i) By Lemma 4.2(iii), the sequence $\{U^k\}_{k \in \mathbb{N}}$ is bounded, which implies that $\omega(U^0) \neq \emptyset$. Let $U^* = (Z^*, w^*, x^*)$ be an arbitrary point from $\omega(U^0)$. Then there is a subsequence $\{(Z^{k_q}, w^{k_q}, x^{k_q})\}_{q \in \mathbb{N}}$ such that $(Z^{k_q}, w^{k_q}, x^{k_q}) \to (Z^*, w^*, x^*)$ as $q \to \infty$. By Lemma 4.2(ii), we have $(Z^{k_q-1}, w^{k_q-1}, x^{k_q-1}) \to (Z^*, w^*, x^*)$ as $q \to \infty$. By Lemma 4.4 for each $q \in \mathbb{N}$, $(A^q, A^q, A^q) \in \partial \Psi(Z^{k_q}, w^{k_q}, x^{k_q})$. By Remark 2.1, we only need to argue that $\lim_{q \to \infty} \Psi(Z^{k_q}, w^{k_q}, x^{k_q}) = \Psi(Z^*, w^*, x^*)$. Since the function $\Psi$ is lsc,

$$\liminf_{q \to \infty} \Psi(Z^{k_q}, w^{k_q}, x^{k_q}) \geq \Psi(Z^*, w^*, x^*). \quad (21)$$

In addition, from the feasibility of $(Z^{k_q}, w^{k_q})$ and $(Z^*, w^*)$ to the problem (8), we have

$$f(Z^{k_q}, x^{k_q-1}) + \frac{\alpha k_q - 1}{2} \left( \|Z^{k_q} - Z^{k_q-1}\|_F^2 + \|w^{k_q} - w^{k_q-1}\|_F^2 \right) \leq f(Z^*, x^{k_q-1}) + \frac{\alpha k_q - 1}{2} \left( \|Z^* - Z^{k_q-1}\|_F^2 + \|w^* - w^{k_q-1}\|_F^2 \right). \quad (22)$$

Taking the limit $q \to \infty$ and using Lemma 4.2(ii) immediately yields that

$$\limsup_{q \to \infty} \Psi(Z^{k_q}, w^{k_q}, x^{k_q}) \leq \limsup_{q \to \infty} \left( \Psi(Z^{k_q}, w^{k_q}, x^{k_q-1}) - \frac{\rho k_q - 1}{2} \|x^{k_q} - x^{k_q-1}\|_F^2 \right)$$

$$= \limsup_{q \to \infty} \left( f(Z^{k_q}, x^{k_q-1}) - \frac{\rho k_q - 1}{2} \|x^{k_q} - x^{k_q-1}\|_F^2 \right)$$

$$\leq f(Z^*, w^*) = \Psi(Z^*, w^*, x^*)$$

where the first inequality is due to (15), and the second one is using inequality (22), the limits $(Z^{k_q}, w^{k_q}, x^{k_q}) \to (Z^*, w^*, x^*)$ and $(Z^{k_q-1}, w^{k_q-1}, x^{k_q-1}) \to (Z^*, w^*, x^*)$, and the continuity of $f$. Along with (21), we obtain $\lim_{q \to \infty} \Psi(Z^{k_q}, w^{k_q}, x^{k_q}) = \Psi(Z^*, w^*, x^*)$.

(ii) Let $U^* = (Z^*, w^*, x^*)$ be an arbitrary point from the set $\omega(U^0)$. There is a subsequence $\{(Z^{k_q}, w^{k_q}, x^{k_q})\}_{q \in \mathbb{N}}$ such that $(Z^{k_q}, w^{k_q}, x^{k_q}) \to (Z^*, w^*, x^*)$ as $q \to \infty$. From the proof of part (i), we have $\lim_{q \to \infty} \Psi(Z^{k_q}, w^{k_q}, x^{k_q}) = \Psi(Z^*, w^*, x^*)$. On the other hand, by Lemma 4.2(i) and the nonnegativity of $\Psi$, the sequence $\{\Psi(Z^k, w^k, x^k)\}_{k \in \mathbb{N}}$ is convergent, and denote its limit by $\varphi^*$. The two sides imply that $\Psi(Z^*, w^*, x^*) = \varphi^*$. Hence, the cost function $\Psi$ is finite and constant on $\omega(U^0)$. $\square$

In order to strengthen the convergence result of Proposition 4.1, we need to take a closer look at the KL property of the extended valued objective function $\Psi$.

Lemma 4.4 The function $\Psi$ is semialgebraic, and hence it satisfies the KL property with $\phi(s) = cs^{1-\theta}$ for some $c > 0$ and $\theta \in (0, 1) \cap \mathbb{Q}$, where $\mathbb{Q}$ is the set of all rational numbers.
Proof: Recall that \( \Psi(Z,w,x) = f(Z,x) + g(Z,w) \) for any \((Z,w,x) \in \mathbb{R}^{m \times n} \times \mathbb{R}^m \times \mathbb{R}^{md} \), where \( f \) and \( g \) are the functions defined by (12). Since \( g \) is an indicator on a polyhedral set which is clearly semialgebraic, \( g \) is semialgebraic by [1, Section 4.3]. Notice that

\[
    f(Z,x) = \frac{1}{N} \sum_{t=1}^{N} \sum_{i=1}^{m} \sum_{j=1}^{n_t} Z_{tij} \|x_i - a_{tj}\|_2^2
\]

which is a polynomial function. Hence, \( f \) is also semialgebraic. Since the sum of two semialgebraic functions is also semialgebraic, the function \( \Psi \) is semialgebraic. The second part of the conclusions follows from [3]. The proof is completed.  

Using Lemma 4.2-4.4 and following the same arguments as those for [1, Theorem 1], we can establish the following global convergence result of Algorithm 3.1.

**Theorem 4.2** Let \( \{U^k\}_{k \in \mathbb{N}} \) be the sequence generated by Algorithm 3.1. Suppose that the level set \( L_{f(Z_0,x_0)} \) of \( f \) is bounded. Then, the following assertions hold.

(i) The sequence \( \{U^k\}_{k \in \mathbb{N}} \) has a finite length, i.e., \( \sum_{k=1}^{\infty} \|U^{k+1} - U^k\| < \infty \).

(ii) The sequence \( \{U^k\}_{k \in \mathbb{N}} \) converges to a critical point \( U^* = (Z^*,w^*,x^*) \) of \( \Psi \).

5 Numerical experiments

We shall apply Algorithm 3.1 and its accelerated version (Algorithm 3.2) for computing a Wasserstein barycenter of discrete probability distributions with unknown sparse finite supports, and compare its performance with that of the three-block B-ADMM in [31] on synthetic data and real data from USPS\(^1\) and MNIST\(^2\) and BBC News\(^3\). Table 1 summarizes the basic information on the datasets, where \( N \) is the data size, \( d \) is the dimension of the support vectors, and \( m \) is the number of support vectors in a barycenter.

| Data            | \( N \)  | \( d \) | \( m \) |
|-----------------|---------|--------|--------|
| Synthetic       | 10,000  | 16     | 60     |
| Image color     | 5,000   | 3      | 8      |
| USPS digits     | 11,000  | 2      | 80     |
| MNIST digits    | 10,000  | 2      | 160    |
| BBC News        | 2,225   | 400    | 25     |

All numerical results are computed by a desktop computer running on 64-bit Windows Operating System with an Intel(R) Core(TM) i7-7700 CPU 3.6GHz and 16 GB RAM.

\(^1\)http://www.cs.toronto.edu/~roweis/data/usps_all.mat 
\(^2\)http://www.cs.toronto.edu/~roweis/data/mnist_all.mat 
\(^3\)http://mlg.ucd.ie/datasets/bbc.html
5.1 Implementation of the PAM method

As mentioned in Remark 3.1, the major computation work of Algorithm 3.1 and 3.2 at each iteration is solving a strongly convex quadratic program of the following form

\[
\min_{Z \in \mathbb{R}^{m \times n}, w \in \mathbb{R}^m} \left\{ \sum_{t=1}^{N} \left[ \langle Z^t, C_t^{t,k} \rangle + \delta\Sigma_t(Z^t) \right] + \delta\Delta(w) + \frac{\alpha_k}{2} \left[ \| Z - Z^k \|^2_F + \| w - w^k \|^2 \right] \right\}
\]

s.t. \(Z^t e - w = 0, \ t = 1, 2, \ldots, N\) \hfill (23)

where \(C_t^{t,k} = N^{-1} F_t(x^k)\) or \(N^{-1} F_t(\tilde{x}^k)\) for \(t = 1, \ldots, N\). Define the linear mappings

\[ G_t(u) := Z_t^{t,k} - \frac{1}{\alpha_k} C_t^{t,k} - \frac{1}{\alpha_k} u e^T, \ t = 1, \ldots, N \quad \text{and} \quad H(\lambda) := w^k + \frac{1}{\alpha_k} \sum_{t=1}^{N} \lambda^t. \]

After an elementary calculation, the dual of the problem (23) takes the following form

\[
\max_{\lambda \in \mathbb{R}^{mN}} \frac{\alpha_k}{2} \left[ \sum_{t=1}^{N} \left( \| G_t(\lambda^t) - \Pi\Sigma_t(G_t(\lambda^t)) \|^2_F - \| G_t(\lambda^t) \|^2_F \right) \right. \]

\[ + \left. \| H(\lambda) - \Pi\Delta(H(\lambda)) \|^2 - \| H(\lambda) \|^2 + M^k \right] \]

where \(\lambda = (\lambda^1; \ldots; \lambda^N) \in \mathbb{R}^{mN}\) and \(M^k = \sum_{t=1}^{N} \| Z_t^{t,k} \|^2_F + \| w^k \|^2\). Due to the large scale, the state-of-art-software such as the interior point method is not applicable to (23). Here we shall develop a tailored semi-proximal ADMM (sPADMM), which is an extension of the classical ADMM designed by Glowinski and Marroco [10] and Gabay and Mercier [11]. For a given \(\beta > 0\), the augmented Lagrangian function of (23) is given by

\[ L_\beta(Z, w; \lambda) := \sum_{t=1}^{N} \left[ \langle Z^t, C_t^{t,k} \rangle + \delta\Sigma_t(Z^t) + \frac{\alpha_k}{2} \| Z^t - Z^{t,k} \|^2_F \right] + \delta\Delta(w) \]

\[ + \sum_{t=1}^{N} \left( \langle \lambda^t, Z^t e - w \rangle + \frac{\beta}{2} \| Z^t e - w \|^2 \right) + \frac{\alpha_k}{2} \| w - w^k \|^2. \]

With the function \(L_\beta\), the iteration steps of the sPADMM are described as follows.
Algorithm 2 sPADMM for solving the subproblem \[23\]

**Initialize:** Choose $\beta > 0$ and $\tau \in (0, \frac{1+\sqrt{5}}{2})$. For $t = 1, \ldots, N$, let $S^{t,k}: \mathbb{R}^{m \times n_t} \rightarrow \mathbb{R}^{m \times n_t}$ be a self-adjoint positive semidefinite linear map such that $\alpha_k I + \beta A^t + S^{t,k} \succeq 0$, where $A^t(X) := Xe e^T$ for $X \in \mathbb{R}^{m \times n_t}$. Choose an initial $(w^0, \lambda^0) \in \mathbb{R}^m \times \mathbb{R}^{mN}$. Set $\nu = 0$.

**Step 1.** Compute the following optimization problems

\[
\begin{align*}
Z^{\nu+1} &= \arg \min_{Z \in \mathbb{R}^{m \times n}} \left\{ L_\beta(Z, w^\nu; \lambda^\nu) + \frac{1}{2} \sum_{t=1}^N \|Z_t - Z_t^{\nu}\|_S^{t,k}^2 \right\} \quad \text{(25a)} \\
w^{\nu+1} &= \arg \min_{w \in \mathbb{R}^m} L_\beta(Z^{\nu+1}, w; \lambda^\nu). \quad \text{(25b)}
\end{align*}
\]

**Step 2.** Update the Lagrange multiplier by the formula

\[
\lambda_t^{\nu+1} := \lambda_t^{\nu} + \tau \beta (Z_t^{\nu+1} e - w_t^{\nu+1}), \quad t = 1, 2, \ldots, N. \quad \text{(26)}
\]

**Step 3.** Set $\nu \leftarrow \nu + 1$, and then go to Step 1.

**Remark 5.1** (i) An immediate choice of $S^{t,k}$ is $S^{t,k} := (\sigma_t - \alpha_k)I - \beta A^t$ for a certain $\sigma_t \geq \alpha_k + \beta \|A^t\|$. By the definition of the linear mapping $A^t: \mathbb{R}^{m \times n_t} \rightarrow \mathbb{R}^{m \times n_t}$, its spectral norm satisfies $\|A^t\| \leq \|ee^T\| \leq n_t$. This means that $\sigma_t = \alpha_k + \beta n_t$ satisfies the requirement. In the subsequent numerical experiments, we always choose such $S^{t,k}$.

(ii) For the global convergence and the linear rate of convergence of Algorithm 2, the reader may refer to [9, 12]; and for its ergodic iteration complexity, may refer to [22].

(iii) By the definition of $L_\beta$ and the choice of $S^{t,k}$ in part (i), for each $t = 1, \ldots, N$,

\[
Z_t^{\nu+1} = \arg \min_{Z_t \in \Sigma_t} \frac{\sigma_t}{2} \|Z_t - \sigma_t^{-1}H^t\|^2_p
\]

with $H^t := [(\sigma_t - \alpha_k)I - \beta A^t](Z_t^{\nu}) + \alpha_k(Z_t^{t,k} - (\alpha_k N)^{-1}F^t(x^k)) + \beta w_t^{\nu}e^T - \lambda_t^{\nu}e^T$, and

\[
w^{\nu+1} = \arg \min_{w \in \Delta} \frac{\beta N + \alpha_k}{2} \left\|w - \frac{1}{\beta N + \alpha_k} \left[ \sum_{t=1}^N \left( \beta Z_t^{\nu+1} e + \lambda_t^{\nu} \right) + \alpha_k w^k \right] \right\|^2.
\]

By the definitions of $\Sigma_t$ and $\Delta$, clearly, both $Z^{\nu+1}$ and $w^{\nu+1}$ have a closed form, which means that the computation work of Algorithm 2 at each iteration is tiny.

By the optimality conditions, the accuracy of an approximate optimal solution $(Z, w, \lambda)$
of (23) and (24) is measured via \( \eta = \max \{ \eta_P, \eta_{\text{gap}} \} \) with \( \eta_P = \max(\eta_Z, \eta_w, \eta_{Z,w}) \) where
\[
\eta_Z = \frac{\sqrt{\sum_{t=1}^{N} \|Z^t - \Pi_{z_t}(Z^t)\|_{F}^2}}{1 + \|Z\|}, \quad \eta_w = \frac{\|w - \Pi_{\Delta}(w)\|}{1 + \|w\|},
\]
\[
\eta_{Z,w} := \sqrt{\sum_{t=1}^{N} \|Z^t e - w\|_2^2}, \quad \eta_{\text{gap}} := \frac{|\text{obj}_P + \text{obj}_D|}{1 + |\text{obj}_P| + |\text{obj}_D|}.
\]
In each iteration of Algorithm 3.1 or 3.2, we run Algorithm 2 until \( \max(\eta_P, 0.1\eta_{\text{gap}}) \leq \epsilon_k \), where \( \epsilon_k \) varies with the iterates of Algorithm 3.1 or 3.2. Specifically, we update \( \epsilon_k \) by
\[
\epsilon_{k+1} = \max(10^{-5}, 0.8\epsilon_k) \quad \text{with} \quad \epsilon_0 = 5 \times 10^{-2}.
\]
We terminate Algorithm 3.1 and 3.2 at \( (Z^{k+1}, w^{k+1}, x^{k+1}) \) if \( \eta_{Z,w,\text{gap}} \leq 3 \times 10^{-5} \) and
\[
\sqrt{\frac{F(x^{k+1}) - F(y^k)}{N}} - \alpha_k (Z^{k+1} - Z^k)_{F}^2 + \|\alpha_k (w^{k+1} - w^k)\|_2^2 + \|\rho_k (x^{k+1} - y^k)\|_2^2 
\]
\[
1 + \|a_1 \cdots a_{n_1} \cdots a_1^N \cdots a_{n_N} \|_F \leq 10^{-2},
\]
where \( y^k = x^k \) for Algorithm 3.1 and \( y^k = \tilde{x}^k \) for Algorithm 3.2. By Lemma 4.1, the error on the left hand side of the last inequality actually measures the approximate accuracy of \( (Z^{k+1}, w^{k+1}, x^{k+1}) \) as a stationary point of the problem (7).

For the parameter \( \rho_k \), since preliminary tests show that a varying \( \rho_k \) does not improve the performance of Algorithm 3.1 and 3.2, we set \( \rho_k \equiv 10^{-5} \). For the parameter \( \alpha_k \) in Algorithm 3.1 or 3.2 we update it by the following rule with \( \alpha_0 = 100 \) and \( \alpha = 10^{-8} \)
\[
\alpha_{k+1} = \begin{cases} 
\min(\alpha_k, 0.1\alpha_k) & \text{if} \quad \frac{\alpha_k}{\alpha_k} \left(\|Z^{k+1} - Z^k\|_F^2 + \|w^{k+1} - w^k\|_2^2\right) > 10^{-5} f(Z^{k+1}, y^k); \\
\alpha_k & \text{otherwise}
\end{cases}
\]
where \( y^k = x^k \) for Algorithm 3.1 and \( y^k = \tilde{x}^k \) for Algorithm 3.2. As shown in Figure 1 below (the time and objective value curves are plotted by using the data from Example 5.2), the varying \( \alpha_k \) can reduce the time much without worsening solution.

![Figure 1: Computing time and objective value of Algorithm 3.2 with fixed and varying \( \alpha_k \)](image)

Figure 1: Computing time and objective value of Algorithm 3.2 with fixed and varying \( \alpha_k \)
5.2 Numerical comparisons with three-block B-ADMM

We shall compare the performance of the PAM method with that of the three-block B-ADMM in [31] for computing the barycenter of discrete distributions with unknown sparse finite supports. We first give a brief introduction to the three-block B-ADMM.

5.2.1 Three-block B-ADMM

By introducing \( Y = [Y^1 \cdots Y^N] \in \mathbb{R}^{m \times n} \), the problem (7) can be equivalently written as

\[
\min_{Z,Y,w,x} \left\{ \sum_{t=1}^{N} \langle Z^t, F^t(x) \rangle \right. \\
\left. \text{s.t. } Z^t = Y^t, \ Z^t \in \Sigma_t, \ (Y^t, w) \in \tilde{\Gamma}_t, \ t = 1, \ldots, N \right\}
\]

(27)

where \( \tilde{\Gamma}_t = \Gamma_t \cap (\mathbb{R}_+^{m \times n_t} \times \mathbb{R}^m) \). Define the KL function \( D^t(Z^t, Y^t) := \sum_{i=1}^{m} \sum_{j=1}^{n_t} Z^t_{ij} \left( \log \left( \frac{Z^t_{ij}}{Y^t_{ij}} \right) - 1 \right) \), \( t = 1, 2, \ldots, N \) where we stipulate \( 0 \log 0 = 0 \). The three-block B-ADMM proposed in [31] replaces the quadratic augmented Lagrangian by the Bregman divergence \([5]\) when updating the split variables, which is exactly the KL regularized Lagrange function defined by

\[
D_\varrho(Z, Y, w; \Lambda) := \sum_{t=1}^{N} \left( \langle Z^t, F^t(x) \rangle + \langle Z^t - Y^t, \Lambda^t \rangle + \varrho D^t(Z^t, Y^t) \right)
\]

where \( \varrho > 0 \) is the regularization parameter. Its iteration steps are described as follows.

**Algorithm 3** (3-block B-ADMM for solving the problem (27))

**Initialize:** Choose \( \varrho > 0 \) and a starting point \((X^0_t, Y^0_t, w^0, x^0, \Lambda^0)\). Set \( k := 0 \).

**Step 1.** Compute the following optimization problems successively:

\[
\begin{cases}
Z^{k+1} = \arg \min_{Z \in \Sigma_1 \times \cdots \times \Sigma_N} D_\varrho(Z, Y^k, w^k, x^k, \Lambda^k), \\
(w^{k+1}, x^{k+1}) \in \arg \min_{w \in \Delta, x \in \mathbb{R}^{md}} D_\varrho(Z^{k+1}, Y^k, w, x; \Lambda^k), \\
Y^{k+1} = \arg \min_{Y \in \Xi_1(w^{k+1}) \times \cdots \times \Xi_N(w^{k+1})} D_\varrho(Y, Z^{k+1}, w^{k+1}, x^{k+1}; \Lambda^k) \\
\end{cases}
\]

(28a)

(28b)

(28c)

where, for a given \( w \in \mathbb{R}^m \), \( \Xi_t(w) := \{ Y^t \in \mathbb{R}^{m \times n_t} \mid Y^t e = w \} \) for \( t = 1, \ldots, N \).

**Step 2.** Update the Lagrange multiplier by the formula

\[
\Lambda^{k+1} := \Lambda^k + \varrho(Z^{k+1} - Y^{k+1}).
\]

(29)

**Step 3.** Set \( k \leftarrow k + 1 \), and then go to Step 1.
Remark 5.2 As discussed in [31], the three subproblems (28a)-(28c) have a closed form solution, which means that the computation work of the B-ADMM at each iteration is tiny. Now it is unclear whether the three-block B-ADMM is convergent or not. As mentioned in the introduction, the direct extension of the classical ADMM to the three-block case can be divergent, not to mention the three-block B-ADMM for solving the nonconvex (27).

5.2.2 Numerical comparisons with 3-block B-ADMM

We test the performance of Algorithm 3.1 and 3.2 and 3-block B-ADMM for computing Wasserstein barycenter of discrete probability distributions from synthetic and real data in Table 1. For Algorithm 3.1 and 3.2 we use the parameters and the stopping criterion described as in Subsection 5.1 and for 3-block B-ADMM we adopt its default setting. In the subsequent experiments, we report the computing time in seconds required by three methods and the objective value (objval) and the infeasibility measure (pinfeas) of the solution yielded by them. Notice that each iterate \((Z^k, w^k, x^k)\) yielded by three methods satisfies the feasibility constraint \((Z^t)^\top e - b^t = 0, t = 1, 2, \ldots, N\). Hence, we use

\[
\text{pinfeas} := \max_{1 \leq t \leq N} \frac{\|Z^t_e - w^t\|}{1 + \|b\|}
\]

to denote the infeasibility measure, where \((Z^t, w^t, x^t)\) means the final iterate. In addition, for all test examples, we run the three methods from the same \((Z^0, w^0, x^0)\).

First of all, we compare the performance of three algorithms on synthetic data.

Example 5.1 We generate a set of 10,000 discrete probability distributions in the same way as [31] does, in which the support vectors are generated by sampling from a multivariate normal distribution and adding a heavy-tailed noise from the student’s t-distribution.

![Figure 2: Numerical results of PAM and 3-block B-ADMM for Example 5.1 under different N](image)
Figure 3: Numerical results of PAM and 3-block B-ADMM for Example 5.1 under different $m$

Figure 2 and 3 plot the computing time, objective value and infeasibility curves of the three methods for solving Example 5.1 under different $N$ with $m = 60$ and under different $m$ with $N = 1000$, respectively. We see that the computing time overall increases as the sample size $N$ or the support size $m$ increases, but Algorithm 3.1 and 3.2 requires less time than 3-block B-ADMM does. Since the latter always attains the maximum number of iterations 2000, its time monotonically increases. As $m$ increases, the objective values and the infeasibility of the three methods decrease, but the objective values and the infeasibility of Algorithm 3.1 and 3.2 are always lower than those of 3-block B-ADMM. In particular, the infeasibility of the former is about $10^{-2}$ less than that of the latter.

Example 5.2 We generate a set of 2,000 discrete probability distributions with sparse finite supports, obtained by clustering pixel colors of images as [14] doing.
Figure 4: Numerical results of PAM and 3-block B-ADMM for Example 5.2 under different $N$.

Figure 5: Numerical results of PAM and 3-block B-ADMM for Example 5.2 under different $m$.

For this example, the three methods demonstrate a similar performance as they do for Example 5.1, i.e., Algorithm 3.1 and 3.2 require less computing time and yield a little better objective value and much lower infeasibility.

**Example 5.3** We obtain a set of 510 discrete distributions from BBC News dataset. The average number of support points is around 25 and the dimension of every support is 400. The texts are treated as a bag of “words”, where the support vector is the vocabulary of the whole document and the weight corresponds to the appearing frequency of words.
Table 2 reports the numerical results of the three methods for the document data in Example 5.3. Similar for the data from Example 5.1-5.2, Algorithm 3.1 and 3.2 yield a little better objective value and much lower infeasibility within less computing time.

| Methods | time(s) | objval | pinfeas |
|---------|---------|--------|---------|
| BBC News (m=25) |
| Algorithm 3.1 | 37.99 | 21.668 | $2.91 \times 10^{-5}$ |
| Algorithm 3.2 | 30.08 | 21.667 | $1.84 \times 10^{-5}$ |
| 3-block B-ADMM | 51.16 | 21.708 | $1.10 \times 10^{-3}$ |

Finally, we compare the performance of three algorithms on two large-scale real data.

Example 5.4 We obtain a set of 1,100 discrete distributions from USPS Handwritten Digits and the average number of support points is around 90, and a set of 1,010 discrete distributions from MNIST Handwritten Digits and the average number of support points is around 160. The digit images are treated as normalized histograms over the pixel locations covered by the digits, where the support vector is the 2D coordinate of a pixel and the weight corresponds to pixel intensity.

Table 3 reports the numerical results of the three methods for the two real data from Example 5.4. We see that 3-block B-ADMM not only requires about twice computing time of Algorithm 3.1 and 3.2 but also its infeasibility is at least 10 times higher than that of the latter. For MNIST Handwritten, the objective value yielded by 3-block B-ADMM is a litter lower than those yielded by Algorithm 3.1 and 3.2 which is reasonable since the infeasibility of the former is much higher than that of the latter.

| Methods | time(s) | objval | pinfeas |
|---------|---------|--------|---------|
| USPS Handwritten Digits (m=80) |
| Algorithm 3.1 | 382.80 | 4.413 | $2.58 \times 10^{-5}$ |
| Algorithm 3.2 | 389.26 | 4.410 | $2.91 \times 10^{-5}$ |
| 3-block B-ADMM | 905.02 | 4.422 | $1.09 \times 10^{-4}$ |
| MNIST Handwritten Digits (m=160) |
| Algorithm 3.1 | 1303.06 | 3.985 | $2.47 \times 10^{-5}$ |
| Algorithm 3.2 | 1278.02 | 3.983 | $2.82 \times 10^{-5}$ |
| 3-block B-ADMM | 3046.10 | 3.977 | $1.62 \times 10^{-4}$ |

From the above numerical comparisons, we conclude that Algorithm 3.1 and 3.2 are noticeably superior to 3-block B-ADMM in terms of the computing time and the infeasibility measure. For those large-scale problems, say Example 5.4, the former reduces about a half of the time for the latter. Moreover, the former yields comparable even a little better objective values. In addition, unlike in the convex setting [23], the accelerated technique has no remarkable improvement on the computing time of Algorithm 3.1.
6 Conclusions

We have developed a globally convergent PAM method for computing an approximate Wasserstein barycenter with unknown supports by designing a tailored semi-proximal ADMM for solving the strongly convex QP subproblems. Numerical comparisons with 3-block B-ADMM on synthetic and several classes of real data show that the proposed PAM method has a significant advantage in reducing the computing time for those large-scale problems. In our future research work, we will focus on the application of the PAM method in the D2-clustering for large-scale image and document data.

Acknowledgements The authors would like to express their sincere thanks to Dr. Jianbo Ye for sharing us with their codes. This work is supported by the National Natural Science Foundation of China under project No.11571120 and No.11701186.

References

[1] H. Attouch, J. Bolte, P. Redont and A. Soubeyran, Proximal alternating minimization and projection methods for nonconvex problems: an approach based on the Kerdyka-Łojasiewicz inequality, Mathematics of Operations Research, 35(2010): 438-457.

[2] H. Attouch, J. Bolte and B. F. Svaiter, Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward-backward splitting, and regularized Gauss-Seidel methods, Mathematical Programming, 137(2013): 91-129.

[3] J. Bolte, A. Daniilidis and A. Lewis, The Lojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems, SIAM Journal on Optimization, 17(2006): 1205-1223.

[4] J. Bolte, S. Sabach and M. Teboulle, Proximal alternating linearized minimization for nonconvex and nonsmooth problems, Mathematical Programming, 146(2014): 459-494.

[5] L. M. Bregman, The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming, USSR Comput. Math. Math. Phys., 7(1967): 200-217.

[6] C. H. Chen, B. S. He, Y. Y. Ye and X. M. Yuan, The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent, Mathematical Programming, 155(2016): 57-79.

[7] M. Cuturi, Sinkhorn distances: Lightspeed computation of optimal transport, in Proc. Adv. Neural Inf. Process. Syst., 2013: 2292-2300.
[8] M. Cuturi and A. Doucet, Fast computation of Wasserstein barycenters, in Proc. Int. Conf. Mach. Learn., 2014, pp. 685-693.

[9] M. Fazel, T. K. Pong, D. F. Sun and P. Tseng, Hankel matrix rank minimization with applications to system identification and realization, SIAM Journal on Matrix Analysis, 34(2013): 946-977.

[10] R. Glowinski and A. Marrocco, Sur l’approximation par éléments finis d’ordre un, et la résolution, par pénalisation-dualité, d’une classe de problèmes de dirichlet non linéaires, Revue Francaise d’Automatique, Informatique et Recherche Opérationnelle, 9(1975): 41-76.

[11] D. Gabay and B. Mercier, A dual algorithm for the solution of nonlinear variational problems via finite element approximation, Computers and Mathematics with Applications, 2(1976): 17-40.

[12] D. R. Han, D. F. Sun and L. W. Zhang, Linear rate convergence of the alternating direction method of multipliers for convex composite programming, Mathematics of Operations Research, 43(2017): 622-637.

[13] M. Hong, M. Razaviyayn, Z. Q. Luo and J. S. Pang, A unified algorithmic framework for block-structured optimization involving big data: with applications in machine learning and signal processing, IEEE Signal Processing Magazine, 33(2016): 57-77.

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

[15] C. L. Mallows, A note on asymptotic joint normality, The Annals of Mathematical Statistics, 43(1972): 508-515.

[16] Y. Nesterov, A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$, Doklady AN SSSR (translated as Soviet Math. Dokl.), 269(1983): 543-547.

[17] O. Pele and M. Werman, Fast and robust Earth mover’s distances, in Proceedings of IEEE International Conference on Computer Vision, 2009, pp. 460-467.

[18] M. Razaviyayn, M. Hong, Z. Q. Luo, A unified convergence analysis of block successive minimization methods for nonsmooth optimization, SIAM Journal on Optimization, 23(2013): 1126-1153.

[19] R. T. Rockafellar, Convex Analysis, Princeton University Press, 1970.

[20] R. T. Rockafellar and R. J-B. Wets, Variational Analysis, Springer, 1998.

[21] Y. Rubner, C. Tomasi and L. J. Guibas, The Earth mover’s distance as a metric for image retrieval, International journal of computer vision, 40(2000): 99-121.
[22] L. Shen and S. H. Pan, Weighted iteration complexity of the sPADMM on the KKT residuals for convex composite optimization, arXiv:1611.03167.

[23] D. F. Sun, K. C. Toh and L. Q. Yang, An efficient inexact ABCD method for least squares semidefinite programming, SIAM Journal on Optimization, 26(2016): 1072-1100.

[24] P. Tseng, Convergence of a block coordinate descent method for nondifferentiable minimization, Journal of Optimization Theory and Application, 109(2001): 475-494.

[25] P. Tseng and S. W. Yun, A coordinate gradient descent method for nonsmooth separable minimization, Mathematical Programming, 117(2009): 387-423.

[26] C. Villani, Optimal Transport: Old and New, New York, NY, USA: Springer, 2008, vol. 338.

[27] H. Wang and A. Banerjee, Bregman alternating direction method of multipliers, in Proc. Adv. Neural Inf. Process. Syst., 2014, pp. 2816-2824.

[28] Y. Y. Xu and W. T. Yin, A globally convergent algorithm for nonconvex optimization based on block coordinate update, Journal of Scientific Computing, 72(2017): 700-734.

[29] L. Yang, J. Li, D. F. Sun and K. C. Toh, A fast globally linearly convergent algorithm for the computation of Wasserstein Barycenters, arXiv:1809.04249.

[30] J. B. Ye and J. Li, Scaling up discrete distribution clustering using ADMM, in Proceedings of International Conference on Image Process, 2014, pp. 5267-5271.

[31] J. B. Ye, P. R. Wu, J. Z. Wang and J. Li, Fast discrete distribution clustering using wasserstein barycenter with sparse support, IEEE Transactions on Signal Processing, 65(2017): 2317-2332.

[32] Y. Zhang, J. Z. Wang and J. Li, Parallel massive clustering of discrete distributions, ACM Transactions on Multimedia Computing, Communications and Applications, 11(2015): 49:1-49:24.