On accelerated alternating minimization

Sergey Guminov[12] , Pavel Dvurechensky[3], Alexander Gasnikov[1,2]

submitted: February 19, 2020

[1] Moscow Institute of Physics and Technology
Institutskiy Pereulok, 9
Dolgoprudny, 141701 Moscow Region
Russian Federation
E-Mail: sergey.guminov@phystech.edu
           gasnikov@yandex.ru

[2] Institute for Information Transmission Problems of RAS
Bolshoy Karetny per. 19, build.1
127051 Moscow
Russian Federation
E-Mail: sergey.guminov@phystech.edu
           gasnikov@yandex.ru

[3] Weierstrass Institute
Mohrenstr. 39
10117 Berlin
Germany
E-Mail: pavel.dvurechensky@wias-berlin.de

On accelerated alternating minimization

Sergey Guminov , Pavel Dvurechensky, Alexander Gasnikov

Abstract

Alternating minimization (AM) optimization algorithms have been known for a long time and are of importance in machine learning problems, among which we are mostly motivated by approximating optimal transport distances. AM algorithms assume that the decision variable is divided into several blocks and minimization in each block can be done explicitly or cheaply with high accuracy. The ubiquitous Sinkhorn's algorithm can be seen as an alternating minimization algorithm for the dual to the entropy-regularized optimal transport problem. We introduce an accelerated alternating minimization method with a $1/k^2$ convergence rate, where k is the iteration counter. This improves over known bound $1/k$ for general AM methods and for the Sinkhorn's algorithm. Moreover, our algorithm converges faster than gradient-type methods in practice as it is free of the choice of the step-size and is adaptive to the local smoothness of the problem. We show that the proposed method is primal-dual, meaning that if we apply it to a dual problem, we can reconstruct the solution of the primal problem with the same convergence rate. We apply our method to the entropy regularized optimal transport problem and show experimentally, that it outperforms Sinkhorn's algorithm.

1 Introduction

Alternating minimization (AM) optimization algorithms have been known for a long time [24, 7]. These algorithms assume that the decision variable is divided into several blocks and minimization in each block can be done explicitly, i.e. they assume the availability of small-dimensional minimization oracle (SDM-oracle). AM algorithms have a number of applications in machine learning problems. For example, iteratively reweighted least squares can be seen as an AM algorithm. Other applications include robust regression [18] and sparse recovery [10]. Famous Expectation Maximization (EM) algorithm can also be seen as an AM algorithm [19, 3].

In this paper, we are mostly motivated by optimal transport applications, which are widespread in the machine learning community [8, 9, 4]. The ubiquitous Sinkhorn's algorithm can be seen as an alternating minimization algorithm for the dual to the entropy-regularized optimal transport problem. Recent Greenkhorn algorithm [2], which is a greedy version of Sinkhorn's algorithm, is a greedy modification of an AM algorithm.

Sublinear $1/k$ convergence rate was proved for AM algorithm in [5]. Despite the same convergence rate as for the gradient method, AM-algorithms converge faster in practice as they are free of the choice of the step-size and are adaptive to the local smoothness of the problem. At the same time, there are accelerated gradient methods which use a momentum term to have a faster convergence rate of $1/k^2$ [20]. Our goal in this paper is to combine the idea of alternating minimization and momentum acceleration to propose an accelerated alternating minimization method.

Related work. Besides mentioned above works on AM algorithms, we mention [6, 25, 28], where non-asymptotic convergence rates for AM algorithms were proposed and their connection with cyclic

coordinate descent was discussed, but the analyzed algorithms are not accelerated. Accelerated versions are known for random coordinate descent methods [22, 16, 26, 17, 13, 1, 14, 23]. These methods use momentum term and block-coordinate steps, rather than full minimization in blocks.

A hybrid method, which uses exact minimization in the last block and random coordinate descent steps in other blocks was proposed in [11]. This method and its analysis can be extended[1] to obtain an accelerated alternating minimization method. We underline that our method and analysis is different from the approach in [11].The AAR-BCD method presented in [11] only utilises steps over a single coordinate block and has convergence rate dependent on the block-wise Lipschitz constants of all but one block, similarly to the convergence rate of the AM method established in [5]. The methods presented in this paper, on the other hand, are modifications of accelerated gradient methods. They inherit some nice properties of the original methods, like primal-duality, adaptivity to the smoothness of the objective, or convergence to a stationary point for non-convex objectives. However, their iterations include steps over the whole coordinate space.

Concerning the optimal transport problem, the most used algorithm is Sinkhorn's algorithm [27, 8]. Its convergence rate was analyzed in [2] and improved in [12] to $1/k$, which is expected in view of [5] and the fact that Sinkhorn's algorithm can be seen as an AM algorithm. An accelerated gradient descent method in application to OT problem was also analyzed in [12] with a better dependence on k in the rate, but worse dependence on the dimension of the problem.

Our contributions. We introduce an accelerated alternating minimization method with the $1/k^2$ convergence rate for unconstrained problems, and also an accelerated alternating minimization method with a $1/k$ convergence rate in terms of the squared norm of the gradient for non-convex problems. The main idea is to combine block-wise minimization and the extrapolation (also known as momentum) step which is usually used in accelerated gradient methods. We also show that the proposed method is primal-dual, meaning that if we apply it to a dual problem, we can reconstruct the solution of the primal problem with the same convergence rate. We apply our method to the dual of the entropy-regularized optimal transport problem and show experimentally, that it outperforms Sinkhorn's algorithm. In some sense, our algorithm can be considered as an accelerated Sinkhorn's algorithm.

2 Accelerated alternating minimization algorithm

In this paper we consider the minimization problem

$$f(x) \to \min_{x \in \mathbb{R}^N}, \tag{1}$$

where $f(x)$ is assumed to be convex and continuously differentiable and the space is equipped with the Euclidean norm $\|\cdot\|$. We also assume that the problem has at least one solution, denoted by x_*. The set $\{1, \ldots, N\}$ of indices of the orthonormal basis vectors $\{e_i\}_{i=1}^N$ is divided into n disjoint subsets (blocks) $I_k, k \in \{1, \ldots, n\}$. Let $S_k(x) = x + \mathrm{span}\{e_i : i \in I_k\}$, i.e. the affine subspace containing x and all the points differing from x only over the block k. We use x_i to denote the components of x corresponding to the block i and $\nabla_i f(x)$ to denote the gradient corresponding to the block i. We will further require that for any $k \in \{1, \ldots, n\}$ and any $z \in \mathbb{R}^N$ the problem $f(x) \to \min_{x \in S_i(z)}$ has a solution, and this solution is easily computable. Let us also assume that the gradient of $f(x)$ is L-Lipschitz: $\forall x, y \in \mathbb{R}^N \quad \|\nabla f(x) - \nabla f(y)\| \leqslant L\|x - y\|$. We call such functions L-smooth.

---

[1]Personal communication with Jelena Diakonikolas.

For the general case of number of blocks $n \geqslant 2$ the Alternating Minimization algorithm may be written as Algorithm 1. There are multiple common block selection rules, such as the cyclic rule or the Gauss-Southwell rule. More generally, it is also possible to update more than one block on each iteration [15].

Algorithm 1 Alternating Minimization

Input: Starting point x_0.

Output: x^k

1: **for** $k \geqslant 0$ **do**

2:    Choose $i \in 1, \ldots, n$

3:    Set $x^{k+1} = \underset{x \in S_i(x^k)}{\operatorname{argmin}} f(x)$

4: **end for**

Below we present two versions of accelerated alternating minimization algorithms for the solution of problem (1). The first version, Algorithm 2, tries to adapt to the smoothness constant L and choose the step size for the extrapolation step based on an updated proxy L_k to L. The second method, Algorithm 3 is free of this parameter and uses a univariate minimization to find the step size for the extrapolation step. We underline that, although our methods do share some similarities with existing ones, they do differ significantly. We use a greedy approach to determine the block which is updated, unlike how it is usually done in random coordiant descent methods. At the same time, the difference with the greedy coordinate descent is twofold: a) we have a momentum term, b) we use full relaxation in the block instead of a coordinate step. Block-wise minimization differentiates our methods from standard accelerated gradient descent methods.

The first algorithm we propose is listed below as Algorithm 2, which incorporates block-wise minimization steps into an accelerated gradient method. On each iteration, we perform an exact minimization over the block corresponding to the largest in norm block of the gradient at the current iterate. This block selection rule is also known as the Gauss-Southwell rule [15].

Algorithm 2 Accelerated Alternating Minimization 1

Input: starting point x_0, initial estimate of the Lipschitz constant L_0.

Output: x^k

1: $x^0 = y^0 = v^0$.

2: **for** $k \geqslant 0$ **do**

3:     Set $L_{k+1} = L_k/2$

4:     **while** True **do**

5:         Set $a_{k+1} = \frac{1}{2L_{k+1}} + \sqrt{\frac{1}{4L_{k+1}^2} + a_k^2 \frac{L_k}{L_{k+1}}}$

6:         Set $\tau_k = \frac{1}{a_{k+1}L_{k+1}}$

7:         Set $y^k = \tau_k v^k + (1 - \tau_k)x^k$ {Extrapolation step}

8:         Choose $i_k = \underset{i \in \{1,\dots,n\}}{\operatorname{argmax}} \|\nabla_i f(y^k)\|^2$

9:         Set $x^{k+1} = \underset{x \in S_{i_k}(y^k)}{\operatorname{argmin}} f(x)$ {Block minimization}

10:         Set $v^{k+1} = v^k - a_{k+1}\nabla f(y^k)$ {Update momentum term}

11:         **if** $f(x^{k+1}) \leqslant f(y^k) - \frac{\|\nabla f(y^k)\|^2}{2L_{k+1}}$ **then**

12:             **break**

13:         **end if**

14:         Set $L_{k+1} = 2L_{k+1}$.

15:     **end while**

16: **end for**

The convergence rate of this algorithm is given by the following theorem

Theorem 2.1. *If $L_0 \leqslant 4nL$, then after k steps of Algorithm 2 it holds that*

$$f(x^k) - f(x_*) \leqslant \frac{4nL\|x^0 - x_*\|^2}{k^2}. \tag{2}$$

This convergence rate is n times worse than that of an adaptive accelerated gradient method [12], or, equivalently, this means that in the worst case it may take $\sqrt{n}$ times more iterations to guarantee accuracy ε compared to an adaptive accelerated gradient method. To prove the convergence rate of the AAM-1 method, we will need a technical result, the proof of which may be found in the supplementary material.

Lemma 2.2. *For any $u \in \mathbb{R}^N$ and any $k \geqslant 0$*

$$a_{k+1}^2 L_{k+1} f(x^{k+1}) - \left(a_{k+1}^2 L_{k+1} - a_{k+1}\right) f(x^k) + \frac{1}{2}\|v^k - u\|^2 - \frac{1}{2}\|v^{k+1} - u\|^2 \leqslant a_{k+1}f(u).$$

Proof of Theorem 2.1. Note that $a_{k+1} = \frac{1}{2L_{k+1}} + \sqrt{\frac{1}{4L_{k+1}^2} + a_k^2 \frac{L_k}{L_{k+1}}}$ satisfies the equation $a_{k+1}^2 L_{k+1} = a_k^2 L_k + a_{k+1}$. We also have $a_1 = \frac{1}{L_{k+1}}$. With that in mind, we sum up the inequality in the statement of Lemma 2.2 for $k = 0, \dots, T-1$ and set $u = x_*$:

$$L_T a_T^2 f(x^T) + \frac{1}{2}\|v^0 - x_*\|^2 - \frac{1}{2}\|v^T - x_*\|^2 \leqslant \sum_{k=0}^{T-1} a_k f(x_*) = L_T a_T^2 f(x_*).$$

Denote $A_k = a_k^2 L_k$. Since $v^0 = x^0$, we now have that for any $T \geqslant 1$

$$f(x^T) - f(x_*) \leqslant \frac{\|x^0 - x_*\|^2}{2A_T}.$$

It remains to estimate A_T from below. We will now show by induction that $A_k \geqslant \frac{nk^2}{8L}$. From the L-smoothness of the objective we have

$$f(x^{k+1}) = \operatorname*{argmin}_{x \in S_{i_k}(y^k)} f(x) \leqslant f(y^k - \frac{1}{L}\nabla_{i_k} f(y^k)) \leqslant f(y^k) - \frac{1}{2L}\|\nabla_{i_k} f(y^k)\|^2.$$

Also, since i_k is chosen by the Gauss–Southwell rule, it is true that

$$\|\nabla_{i_k} f(y^k)\|^2 \geqslant \frac{1}{n}\|\nabla f(y^k)\|^2.$$

As a result,

$$f(x^{k+1}) \leqslant f(y^k) - \frac{1}{2nL}\|\nabla f(y^k)\|^2.$$

This implies that the condition in line 11 of Algorithm 2 is automatically satisfied if $L_{k+1} \geqslant nL$. Combined with the fact that we multiply L_{k+1} by 2 if this condition is not met, this means that if $L_{k+1} \leqslant 2Ln$ at the beginning of the while loop during iteration k, then it is sure to hold at the end of the iteration too. This is guaranteed by our assumption that $L_0 \leqslant 4Ln$.

We have just shown that $L_k \leqslant 2Ln$ for $k \geqslant 1$. The base case $k = 0$ is trivial. Now assume that $A_k \geqslant \frac{k^2}{8nL}$ for some k. Note that $A_{k+1} = L_k a_k^2 + a_k = A_k + a_k$ and $L_{k+1} = \frac{A_{k+1}}{a_{k+1}^2}$.

$$a_{k+1} = \frac{1}{2L_{k+1}} + \sqrt{\frac{1}{4L_{k+1}^2} + a_k^2 \frac{L_k}{L_{k+1}}} \geqslant \frac{1}{4nL} + \sqrt{\frac{1}{16n^2L^2} + a_k^2 \frac{L_k}{2nL}} \geqslant$$

$$\geqslant \frac{1}{4nL}\left(1 + \sqrt{1 + 8A_k nL}\right) \geqslant \frac{k+1}{4nL}.$$

Finally,

$$A_{k+1} = A_k + a_{k+1} \geqslant \frac{k^2 + 2(k+1)}{8nL} \geqslant \frac{(k+1)^2}{8nL}.$$

By induction, we have $A_k \geqslant \frac{k^2}{8nL}$ for all $k \geqslant 1$ and

$$f(x^k) - f(x_*) \leqslant \frac{4nL\|x^0 - x_*\|^2}{k^2}.$$

$\square$

We also note that the assumption $L_0 \leqslant 4L$ is not really crucial. In fact, if $L_0 > 4L$, then after $O(\log_2 \frac{L_0}{4L})$ iterations L_k is surely lesser than $4L$, so overestimating L only results in a logarithmic in $\frac{L_0}{L}$ amount of additional iterations needed to converge.

Unlike the AM algorithm, this method requires computing the whole gradient of the objective, which makes the iterations of this algorithm considerably more expensive. Also, even when the number of blocks is 2, the convergence rate of Algorithm 2 depends on the smoothness parameter L of the whole objective, and not on the Lipschitz constants of each block on its own, which is the case for the

AM algorithm [5]. On the other hand, if we compare the AAM-1 algorithm to an adaptive accelerated gradient method, we will see that the theoretical worst-case time complexity of the AAM-1 method is only $\sqrt{n}$ times worse, while in practice block-wise minimization steps may perform much better than gradient descent steps simply because they directly use some specific structure of the objective.

Our second accelerated alternating minimization algorithm uses an exact minimization over an interval instead of the classic extrapolation step.

Algorithm 3 Accelerated Alternating Minimization 2

Input: Starting point x_0.

Output: x^k

1: Set $A_0 = 0$, $x^0 = v^0$.
2: **for** $k \geqslant 0$ **do**
3:     Set $\beta_k = \underset{\beta \in [0,1]}{\operatorname{argmin}} f\left(x^k + \beta(v^k - x^k)\right)$
4:     Set $y^k = x^k + \beta_k(v^k - x^k)$ {Extrapolation step}
5:     Choose $i_k = \underset{i \in \{1,...,n\}}{\operatorname{argmax}} \|\nabla_i f(y^k)\|^2$
6:     Set $x^{k+1} = \underset{x \in S_{i_k}(y^k)}{\operatorname{argmin}} f(x)$ {Block minimization}
7:     Find largest a_{k+1} from the equation

$$f(y^k) - \frac{a_{k+1}^2}{2(A_k + a_{k+1})} \|\nabla f(y^k)\|^2 = f(x^{k+1})$$

8:     Set $A_{k+1} = A_k + a_{k+1}$
9:     Set $v^{k+1} = v^k - a_{k+1}\nabla f(y^k)$ {Update momentum term}
10: **end for**

The convergence rate of the AAM-2 algorithm is given by the following theorem, the proof of which may be found in the supplementary material.

Theorem 2.3. *After k steps of Algorithm 3 it holds that*

$$f(x^k) - f(x_*) \leqslant \frac{2nL\|x^0 - x_*\|^2}{k^2}.$$

Again, our inclusion of block-wise minimization steps results in the decrease of worst-case convergence rate only by a factor of $\sqrt{n}$ compared to accelerated gradient methods. One of the advantages of this algorithm over the previous one is that it also has guaranteed convergence to a stationary point for non-convex objectives.

Theorem 2.4. *For a (possibly non-convex) L-smooth objective f after k steps of Algorithm 3 it holds that*

$$\min_{i=0,...,k} \|\nabla f(y^i)\|^2 \leqslant \frac{2nL(f(x^0) - f(x_*))}{k}.$$

Line-search implementation   Unlike the AAM-1 algorithm, this method does not require to repeat iterations to estimate L locally with the sequence L_{k+1}. Instead it requires solving a one-dimensional subproblem, which may computationally difficult. However, as it may be seen from the convergence

analysis, this step may be significantly simplified: if we denote by $\tilde{\beta}_k$ the exact solution to the problem $\text{argmin}_{\beta \in [0,1]} \, f\left(x^k + \beta(v^k - x^k)\right)$ and set $\tilde{y}^k = x^k + \tilde{\beta}_k(v^k - x^k)$, it is actually sufficient to find $\beta_k \in [0,1]$ and $y^k = v^k + \beta_k(x^k - v^k)$ such that $f(y^k) \leqslant f(x^k)$ and $\tilde{\beta}_k \leqslant \beta_k$. This modification significantly decreases the time needed for this method to converge to the approximate solution in our experiments.

3 Primal-dual analysis of the algorithms

We consider the minimization problem

$$(P_1) \qquad \min_{x \in Q \subseteq E} \{f(x) : Ax = b\},$$

where E is a finite-dimensional real vector space, Q is a simple closed convex set, A is a given linear operator from E to some finite-dimensional real vector space H, $b \in H$ is given. The Lagrange dual problem to Problem (P_1) is

$$(D_1) \qquad \max_{\lambda \in \Lambda} \left\{ -\langle \lambda, b \rangle + \min_{x \in Q} \left(f(x) + \langle A^T \lambda, x \rangle \right) \right\}.$$

Here we denote $\Lambda = H^*$. It is convenient to rewrite Problem (D_1) in the equivalent form of a minimization problem

$$(P_2) \quad \min_{\lambda \in \Lambda} \left\{ \langle \lambda, b \rangle + \max_{x \in Q} \left(-f(x) - \langle A^T \lambda, x \rangle \right) \right\}.$$

We denote

$$\varphi(\lambda) = \langle \lambda, b \rangle + \max_{x \in Q} \left(-f(x) - \langle A^T \lambda, x \rangle \right). \tag{3}$$

Since f is convex, $\varphi(\lambda)$ is a convex function and, by Danskin's theorem, its subgradient is equal to (see e.g. [21])

$$\nabla \varphi(\lambda) = b - Ax(\lambda) \tag{4}$$

where $x(\lambda)$ is some solution of the convex problem

$$\max_{x \in Q} \left(-f(x) - \langle A^T \lambda, x \rangle \right). \tag{5}$$

In what follows, we make the following assumptions about the dual problem (D_1)

■ The gradient of the objective function $\varphi(\lambda)$ is L-Lipschitz.

■ The dual problem (D_1) has a solution λ^* and there exist some $R > 0$ such that

$$\|\lambda^*\|_2 \leqslant R < +\infty. \tag{6}$$

It is worth noting that the quantity R will be used only in the convergence analysis, but not in the algorithm itself.

Our primal-dual algorithm based on Algorithm 2 for Problem (P_1) is listed below as Algorithm 4.

Algorithm 4 Primal-Dual Accelerated Alternating Minimization 1

Input: initial estimate of the Lipschitz constant L_0.

1: $A_0 = \alpha_0 = 0$, $\eta_0 = \zeta_0 = \lambda_0 = 0$.

2: **for** $k \geqslant 0$ **do**

3:     Set $L_{k+1} = L_k/2$

4:     **while** True **do**

5:         Set $a_{k+1} = \frac{1}{2L_{k+1}} + \sqrt{\frac{1}{4L_{k+1}^2} + a_k^2 \frac{L_k}{L_{k+1}}}$

6:         Set $\tau_k = \frac{1}{a_{k+1}L_{k+1}}$

7:         Set $\lambda^k = \tau_k \zeta^k + (1 - \tau_k)\eta^k$

8:         Choose $i_k = \underset{i \in \{1,...,n\}}{\operatorname{argmax}} \|\nabla_i \phi(\lambda^k)\|^2$

9:         Set $\eta^{k+1} = \underset{\eta \in S_{i_k}(\lambda^k)}{\operatorname{argmin}} \phi(\eta)$

10:       Set $\zeta^{k+1} = \zeta^k - a_{k+1}\nabla f(\lambda^k)$

11:       **if** $f(\eta^{k+1}) \leqslant f(\lambda^k) - \frac{\|\nabla f(\lambda^k)\|^2}{2L_{k+1}}$ **then**

12:         Set $\hat{x}^{k+1} = \frac{a_{k+1}x(\lambda^k) + L_k a_k^2 \hat{x}^k}{L_{k+1}a_{k+1}^2}$.

13:         **break**

14:       **end if**

15:       Set $L_{k+1} = 2L_{k+1}$.

16:     **end while**

17: **end for**

Output: The points $\hat{x}^{k+1}$, η^{k+1}.

The key result for this method is that it guarantees convergence in terms of the constraints and the duality gap for the primal problem, provided that it is strongly convex.

Theorem 3.1. *Let the objective $f(x)$ in the problem (P_1) be γ-strongly convex , and let $\|\lambda^*\| \leqslant R$. Then, for the sequences $\hat{x}^k, \eta^k$, $k \geqslant 0$, generated by Algorithm 4,*

$$\|A\hat{x}^k - b\| \leqslant \frac{16n\|A\|_{E \to H}R}{\gamma k^2}, \quad |\varphi(\eta^k) + f(\hat{x}^k)| \leqslant \frac{16n\|A\|_{E \to H}R^2}{\gamma k^2}, \tag{7}$$

where $\|A\|_{E \to H}$ is the norm of A as a linear operator from E to H.

Of course, Algorithm 3 may also be applied to the dual problem, and the same result as in Theorem 3.1 holds, although with a slightly different constant. The details are in the supplementary material.

4   Application to the Optimal Transportation problem

In this section for any matrix A and any vector a we use e^A, e^a, $\ln A$, $\ln a$ to denote their entrywise exponents and natural logarithms respectively and $\operatorname{diag}(a)$ to denote the diagonal matrix with entries of a as its diagonal elements. For a matrix A we denote $\|A\|_\infty = \max_{i,j}|A_{ij}|$ and $\|A\|_1 = \sum_{i,j}|A_{ij}|$.

Of particular interest to us is the discrete optimal transportation problem

$$f(X) = \langle C, X \rangle \to \min_{X \in \mathcal{U}(r,c)}, \qquad (8)$$

$$\mathcal{U}(r,c) = \{X \in \mathbb{R}_+^{N \times N} : X\mathbf{1} = r, X^T\mathbf{1} = c\},$$

where X is the transportation plan, $C \in \mathbb{R}_+^{N \times N}$ is a given cost matrix, $r, c \in \mathbb{R}^N$ are given elements of the probability simplex, and $\langle A, B \rangle$ denotes the Frobenius product of matrices defined as $\langle A, B \rangle = \sum_{i,j=1}^{N} A_{ij}B_{ij}$.

One way to approach this problem is to regularize it with some strongly convex term and pass to the dual problem. With the entropic regularization term $\gamma\langle X, \ln X \rangle$, $\gamma > 0$, the dual problem may be written as a minimization problem

$$\varphi(u, v) = \gamma(\mathbf{1}^T B(u,v)\mathbf{1} - \langle u, r \rangle - \langle v, c \rangle - 1) \to \min_{u,v \in \mathbb{R}^N} \qquad (9)$$

with $K := e^{-C/\gamma}$ and $B(u,v) = \operatorname{diag}(e^u)K\operatorname{diag}(e^v)$. The full derivation is provided in the supplementary material. Since the regularized problem is strongly convex, the dual is L-smooth [21, 12].

The variables in the dual problem naturally decompose into two blocks u and v. Moreover, minimization over any one block may be performed analytically:

$$\operatorname*{argmin}_{h \in \mathbb{R}^N} \varphi(u + h, v) = \ln r - \ln\left(B\left(u, v\right)\mathbf{1}\right),$$

$$\operatorname*{argmin}_{h \in \mathbb{R}^N} \varphi(u, v + h) = \ln c - \ln\left(B\left(u, v\right)^T\mathbf{1}\right).$$

The AM algorithm for this problem is the well-known Sinkhorn's algorithm [8, 12]. The primal-duality of our algorithms established in the previous section allows us to apply our methods to the dual problem and then recover the approximate solution of the primal problem. The final algorithm takes the following form:

Algorithm 5 Approximate OT by PDAAM

Input: Accuracy ε

1: Set $\gamma = \frac{\varepsilon}{3\ln N}$
2: **for** $k \geqslant 0$ **do**
3:     Perform an iteration of PDAAM-1 on Problem (9).
4:     Find $\hat{X}$ as the projection of $\hat{X}^k$ onto $\mathcal{U}(r,c)$.
5:     **if** $\langle C, \hat{X} - \hat{X}^k \rangle \leqslant \frac{\varepsilon}{6}$ and $f(\hat{X}^k) + \varphi(\eta^k) \leqslant \frac{\varepsilon}{6}$ **then**
6:         Return $\hat{X}$
7:     **end if**
8: **end for**

---

5 Numerical experiments

We performed numerical experiments on the problem of computing the second Wasserstein distance between two 2D-histograms. We randomly chose pairs of images from the MNIST to form the measures

r and c, then we performed a preprocessing step described in [12]:

$$(\tilde{r}, \tilde{c}) = \left(1 - \frac{\varepsilon}{8}\right)\left((r, c) + \frac{\varepsilon}{8n}(1, 1)\right).$$

This should improve the numerical stability of the used algorithms without significantly influencing the solution. We compared the Sinkhorn's algorithm (for which we set $\gamma = \frac{\varepsilon}{4 \ln N}$) [8, 12], which is the explicit form of the AM algorithm for the problem (9), the PDASTM method from [12], which is a primal-dual accelerated gradient method, and our Algorithm 4 and its version based on the AAM-2 method, in which the line-search step was implemented making use of the modification described in the end of section 2. We chose 5 values of accuracy $\varepsilon \in [0.005, 0.1]$. We ran the Sinkhorn's algorithm until the stopping criterion $\|B1 - r\|_1 + \|B^T 1 - c\|_1 \leqslant \frac{\varepsilon}{8\|C\|_\infty}$ is fulfilled. The other three algorithms were stopped as soon as the inequalities $\langle C, \hat{X} - \hat{X}^k \rangle \leqslant \frac{\varepsilon}{6}$ and $f(\hat{X}^k) + \varphi(\eta^k) \leqslant \frac{\varepsilon}{6}$ were satisfied. For each value of ε we randomly selected 20 pairs of images and averaged the computation times of each algorithm.
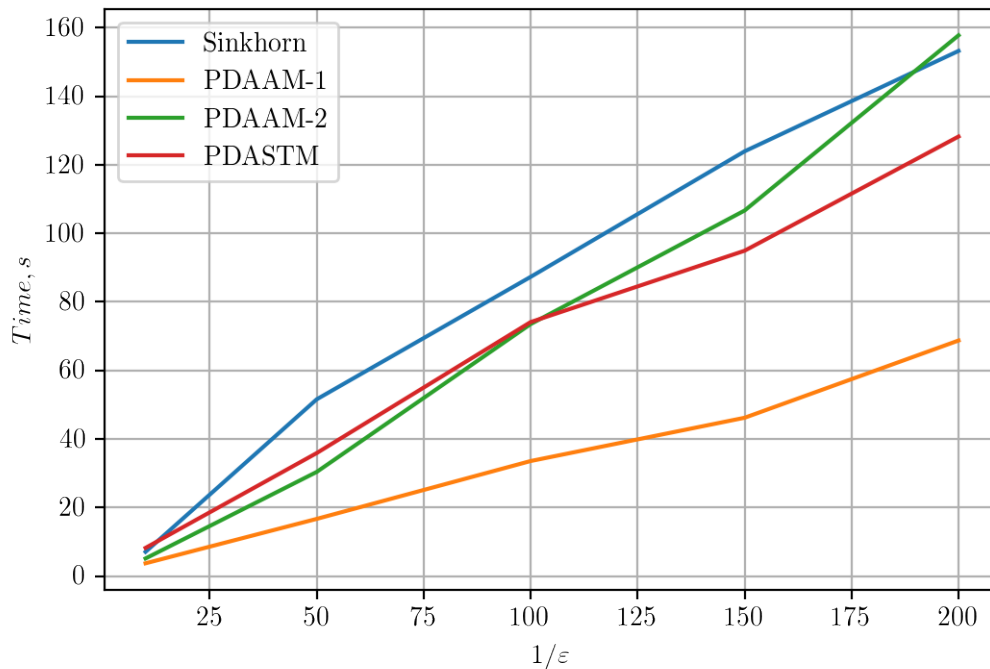

Figure 1: Average computation time to guarantee ε-accurate solution for the primal OT problem.

In this experiment, our PDAAM-1 algorithm converged significantly faster than Sinkhorn's algorithm and the PDASTM method. The PDAAM-2 algorithm showed performance comparable to that of the PDASTM method for larger ε. However, for higher accuracies the performance of the PDAAM-2 method worsens slightly. This may be due to the accumulation of error from the inexact solution to the line-search subproblem. Since the PDAAM-1 method outperformed the PDASTM algorithm, we may conclude that using block-wise minimizations instead of descent along the full gradient results in improved performance for the OT problem, even though the theoretical convergence rate of PDAAM-1 is actually worse by a factor of 2. We also believe that with an appropriate choice of the line-search accuracy the PDAAM-2 method may also demonstrate high performance for other tasks.

Acknowledgements

References

[1] Zeyuan Allen-Zhu, Zheng Qu, Peter Richtarik, and Yang Yuan. Even faster accelerated coordinate descent using non-uniform sampling. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1110–1119, New York, New York, USA, 20–22 Jun 2016. PMLR. First appeared in arXiv:1512.09103.

[2] Jason Altschuler, Jonathan Weed, and Philippe Rigollet. Near-linear time approxfimation algorithms for optimal transport via sinkhorn iteration. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 1961–1971. Curran Associates, Inc., 2017. arXiv:1705.09634.

[3] Andreas Andresen and Vladimir Spokoiny. Convergence of an alternating maximization procedure. *Journal of Machine Learning Research*, 17(63):1–53, 2016.

[4] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *arXiv:1701.07875*, 2017.

[5] A. Beck. On the convergence of alternating minimization for convex programming with applications to iteratively reweighted least squares and decomposition schemes. *SIAM Journal on Optimization*, 25(1):185–209, 2015.

[6] A. Beck and L. Tetruashvili. On the convergence of block coordinate descent type methods. *SIAM Journal on Optimization*, 23(4):2037–2060, 2013.

[7] Dimitri P Bertsekas and John N Tsitsiklis. *Parallel and distributed computation: numerical methods*, volume 23. Prentice hall Englewood Cliffs, NJ, 1989.

[8] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2292–2300. Curran Associates, Inc., 2013.

[9] Marco Cuturi and Arnaud Doucet. Fast computation of wasserstein barycenters. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 685–693, Bejing, China, 22–24 Jun 2014. PMLR.

[10] Ingrid Daubechies, Ronald DeVore, Massimo Fornasier, and C. Sinan Güntürk. Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics*, 63(1):1–38, 2010.

[11] Jelena Diakonikolas and Lorenzo Orecchia. Alternating randomized block coordinate descent. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1224–1232, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[12] Pavel Dvurechensky, Alexander Gasnikov, and Alexey Kroshnin. Computational optimal transport: Complexity by accelerated gradient descent is better than by Sinkhorn's algorithm. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1367–1376, 2018. arXiv:1802.04367.

[13] Olivier Fercoq and Peter Richtárik. Accelerated, parallel, and proximal coordinate descent. *SIAM Journal on Optimization*, 25(4):1997–2023, 2015. First appeared in arXiv:1312.5799.

[14] Alexander Gasnikov, Pavel Dvurechensky, and Ilnura Usmanova. On accelerated randomized methods. *Proceedings of Moscow Institute of Physics and Technology*, 8(2):67–100, 2016. In Russian, first appeared in arXiv:1508.02182.

[15] Mingyi Hong, Meisam Razaviyayn, Zhi-Quan Luo, and Jong-Shi 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(1):57–77, 2016.

[16] Yin Tat Lee and Aaron Sidford. Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems. In *Proceedings of the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, FOCS '13, pages 147–156, Washington, DC, USA, 2013. IEEE Computer Society. First appeared in arXiv:1305.1922.

[17] Qihang Lin, Zhaosong Lu, and Lin Xiao. An accelerated proximal coordinate gradient method. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3059–3067. Curran Associates, Inc., 2014. First appeared in arXiv:1407.1296.

[18] P. McCullagh and J.A. Nelder. *Generalized Linear Models, Second Edition*. Chapman and Hall/CRC Monographs on Statistics and Applied Probability Series. Chapman & Hall, 1989.

[19] G. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley Series in Probability and Statistics. Wiley, 1996.

[20] Yurii Nesterov. A method of solving a convex programming problem with convergence rate $o(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

[21] Yurii Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.

[22] Yurii Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012. First appeared in 2010 as CORE discussion paper 2010/2.

[23] Yurii Nesterov and Sebastian U. Stich. Efficiency of the accelerated coordinate descent method on structured optimization problems. *SIAM Journal on Optimization*, 27(1):110–123, 2017. First presented in May 2015 `http://www.mathnet.ru:8080/PresentFiles/11909/7_nesterov.pdf`.

[24] J.M. Ortega and W.C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 1970.

[25] A. Saha and A. Tewari. On the nonasymptotic convergence of cyclic coordinate descent methods. *SIAM Journal on Optimization*, 23(1):576–601, 2013.

[26] Shai Shalev-Shwartz and Tong Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 64–72, Bejing, China, 22–24 Jun 2014. PMLR. First appeared in arXiv:1309.2375.

[27] Richard Sinkhorn. Diagonal equivalence to matrices with prescribed row and column sums. II. *Proc. Amer. Math. Soc.*, 45:195–198, 1974.

[28] Ruoyu Sun and Mingyi Hong. Improved iteration complexity bounds of cyclic block coordinate descent for convex problems. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 1306–1314, Cambridge, MA, USA, 2015. MIT Press.