Vol.2022-AVM-116 No.10
2022/2/25

緩和最適輸送問題のためのブロック座標Frank-Wolfeアルゴリズムの拡張手法と画像処理への応用

福永 拓海[1,a)]　笠井 裕之[1,2,b)]

Abstract: 確率分布間の距離を表現可能な最適輸送問題は幅広い分野で注目されている．最適輸送問題は厳密な質量保存を表す制約条件を有する線形計画問題で定式化されるが，一般に線形計画問題を高速に解くことは難しい．当該問題の解決のため，制約条件を緩めた緩和最適輸送問題が提案されており，高速化の実現と応用分野への有効性が確認されている．以前発表した研究では，その緩和問題のうち凸緩和最適輸送問題に注目し，Frank-Wolfe アルゴリズムに基づいた高速最適化手法を提案した．しかし，Frank-Wolfe (FW) アルゴリズムは劣線形性で収束するため，その収束速度は依然遅い．本稿では，Frank-Wolfe アルゴリズムの改良手法である，ギャップサンプリングを考慮したブロック座標 Frank-Wolfe (BCFW-GA) アルゴリズムを提案し，そのアルゴリズムの計算量と最悪収束反復数を示す．数値実験から，改良手法と画像処理に対する凸緩和最適輸送問題の有効性を議論する．

1. Introduction

The Optimal Transport (OT) problem has been focused and applied to widely various fields recently, thanks to representability of distances between probability distributions [1]. Because it is defined as a convex linear programming, many dedicated solvers such as an interior-point method and a network-flow method enable us to obtain this solutions. However, it remains hard to solve efficiently because its computational cost increases cubically in terms of the data size.

To avoid this issue, an *entropy-regularized* approach has been widely used because it enables us to bring about the Sinkhorn algorithm [2], which is faster and enables a parallel implementation. In addition, although a stabler variant has been also proposed to cope with its numerical unsuitability and non-robustness against for small values of the regularizer, it still remain slow [3].

In another line of directions, some papers report that the strict mass-conversation constraint in the OT problem diminishes the performance of some applications where mass need not be necessarily preserved. For this particular problem, some researchers have recently utilized a *constraint-relaxed* approach, which relaxes such strict constraints. This approach has gained importance on various machine learning fields such as color transfer [4] and multi-label learning [5]. However, it still suffers from a slow convergence property. To address this slow convergence, a faster algorithm has been proposed by use of the Frank Wolfe (FW) and block-coordinate Frank-Wolfe (BCFW) for semi-relaxed problem [6]. These approaches are more effective thanks to *projection-free* property of FW algorithm. In addition, it enables us to obtain a sparser solution.

This paper gives a fast variant of BCFW for semi-relaxed problem improving the previously proposed algorithm [6]. We focus on another approach to accelarate the convergence speed, which is an *adaptive sampling* strategy that is popular approach in block-coordinate methods [7], [8], [9], [10]. This paper particularly utilizes the approach considering the duality gap. Herein, we denote the BCFW with *gap-adaptive* sampling as BCFW-GA.

[1] Department of Computer Science and Communications Engineering, Graduate School of Fundamental Science and Engineering, Waseda University, Japan
[2] Department of Communications and Computer Engineering, School of Fundamental Science and Engineering, Waseda University, Japan
[a)] f_takumi1997@suou.waseda.jp
[b)] hiroyuki.kasai@waseda.p

2. Preliminary and related work

$\mathbb{R}^n$ is denoted as n-dimensional Euclidean space and $\mathbb{R}^n_+$ is denoted as the set of vectors in which all elements are non-negative. $\mathbb{R}^{m\times n}$ is denoted as the set of $m\times n$ matrices and $\mathbb{R}^{m\times n}_+$ is denoted as the set of $m \times n$ matrices in which all elements are non-negative. We denote vectors as bold lower-case letters $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \ldots$ and matrices as bold-face letters $\mathbf{A}, \mathbf{B}, \mathbf{C}, \ldots$. The i-th element of $\boldsymbol{a}$ and the element at the (i, j) position of $\mathbf{A}$ are represented as $\boldsymbol{a}_i$ and $\mathbf{A}_{i,j}$ respectively. When a matrix $\mathbf{A}$ is denoted as $(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n)$, $\boldsymbol{a}_i$ represents the i-th column vector of $\mathbf{A}$. $\boldsymbol{e}_i$ is the canonical standard unit vector, of which the i-th element is 1, and others are zero. We denote [m] as the set $\{1, 2, \cdots, m$. The probability simplex is denoted as $\Delta_m = \{\boldsymbol{a} \in \mathbb{R}^m : \sum_i a_i = 1\}$. $\boldsymbol{\delta}_{\boldsymbol{a}}$ is the delta function at the vector $\boldsymbol{a}$. $\langle\cdot,\cdot\rangle$ and $\langle\cdot,\cdot\rangle_F$ represent the inner product and the Frobenius norm. Given two matrices $\mathbf{A}, \mathbf{B}$, the Frobenius norm is denoted as $\langle \mathbf{A}, \mathbf{B}\rangle_F := \sum_{i=1}^n \langle \boldsymbol{a}_i, b_i\rangle = \sum_{i=1}^m \sum_{j=1}^n A_{i,j}B_{i,j}$.

2.1 Optimal transport (OT)

Given two empirical probability distributions $\boldsymbol{\nu} = \sum_{i=1}^m a_i\boldsymbol{\delta}_{x_i}$, $\boldsymbol{\mu} = \sum_{i=1}^n b_i\boldsymbol{\delta}_{y_i}$ and the cost matrix $\mathbf{C}$, the OT problem between distributions is defined as:

$$\min_{\mathbf{T}\in\mathcal{U}(\boldsymbol{a},\boldsymbol{b})} \langle \mathbf{T}, \mathbf{C}\rangle_F, \tag{1}$$

where the domain $\mathcal{U}(\boldsymbol{a}, \boldsymbol{b})$ is defined as

$$\mathcal{U}(\boldsymbol{a}, \boldsymbol{b}) = \{\mathbf{T} \in \mathbb{R}_+^{m\times n} : \mathbf{T}\mathbf{1}_n = \boldsymbol{a}, \mathbf{T}^T\mathbf{1}_m = \boldsymbol{b}\}. \tag{2}$$

The resultant transport matrix $\mathbf{T}^*$ brings a powerful distances between distributions, which is known to *Wasserstein distance*. Many problems appearing in machine learning and statistical learning can be defined in the OT problem. We refer the interested readers to [1] for more comprehensive survey .

2.2 Relaxed optimal transport

Domain constraint relaxation. One approach is to relax the constraint domain [11]. Ferradans et al. propose to allow each point of $\mathbf{X}$ to be transported to multiple points of $\mathbf{Y}$ and versa. This method enables the transport matrix to increase or decrease the mass between two points which are low distances. The noteworthy point is that the relaxed domain keeps the linear constraints as the original, thus, existing solvers of linear programming can be used. We also have other relaxed formulations considering only $\mathbf{T}\mathbf{1}_n = \boldsymbol{a}$ or $\mathbf{T}^T\mathbf{1}_m = \boldsymbol{b}$ as $\min_{\mathbf{T}\mathbf{1}_n = \boldsymbol{a}} \langle \mathbf{T}, \mathbf{C}\rangle$ or $\min_{\mathbf{T}^T\mathbf{1}_m=\boldsymbol{b}} \langle \mathbf{T}, \mathbf{C}\rangle$. Because these optimal solutions are summation of minimum costs of each row or column vector, they can be solved faster than linear programming. In practice, this method is useful for document classification [12], and its extended formulation have recently been developed in context of style transfer [13], [14].

Regularized constraint relaxation. Another approach adds the regularized term of the domains defined in (2) into the objective function [15]. Relaxing both marginal constraints in (2) yields the following relaxed formulation:

$$\min_{\mathbf{T}\geq\mathbf{0}} \langle \mathbf{T}, \mathbf{C}\rangle + \frac{1}{2}\Phi(\mathbf{T}\mathbf{1}_n, \boldsymbol{a}) + \frac{1}{2}\Phi(\mathbf{T}^T\mathbf{1}_m, \boldsymbol{b}),$$

where $\Phi(\boldsymbol{x}, \boldsymbol{y})$ is a smooth divergence measure. We also have an alternative formulation, which relaxes one of the two constraints in (2). This is called a *semi-relaxed* problem and is defined as the following:

$$\min_{\mathbf{T}\geq\mathbf{0},\mathbf{T}^T\mathbf{1}_m=\boldsymbol{b}} \langle \mathbf{T}, \mathbf{C}\rangle + \Phi(\mathbf{T}\mathbf{1}_n, \boldsymbol{a}). \tag{3}$$

A similar formulation is also proposed and, is solved by use of augmented Lagrangian [16]. Another formulation specifically focuses on both color transfer and barycenter and is solved by use of the proximal splitting method and the coordinate descent method.[11]. Rabin et al. also propose the weighted regularization term $\| \kappa - \mathbf{1}_n \|_1$ as well as Relaxed Weighted OT so that the ratio of the source image approaches that of the target image [4]. Recently, this approach is used in graph dictionary learning [17].

2.3 Frank-Wolfe and block-coordinate algorithms

The Frank-Wolfe (FW) algorithm is one of the constraint convex optimization methods using conditional gradient [18]. Although FW has *sublinear* convergence rate, its *projection-free* property is preferable in the case where the convex constraint is simple and the feasible point can be found easily. More specifically, at every iteration, the feasible point $\boldsymbol{s}$ is first found by minimizing the *linearization* of f over the convex feasible set $\mathcal{M}$. To find the feasible point $\boldsymbol{s}$, we need to solve the following subproblem :

$$\boldsymbol{s} = \arg\min_{\boldsymbol{s}'\in\mathcal{M}} \langle \boldsymbol{s}', \nabla f(\boldsymbol{x}^{(k)})\rangle \tag{4}$$

where $\boldsymbol{x}^{(k)}$ represents the k-th current point. The convexity of the domain $\mathcal{M}$ and the linearity of the objective enable us to solve (4) by linear programming. Finally, the next iterate $\boldsymbol{x}^{(k+1)}$ can be obtained by a convex combination as $\boldsymbol{x}^{(k+1)} = (1-\gamma)\boldsymbol{x}^{(k)} + \gamma\boldsymbol{s}$ where γ is a stepsize. Therefore, the generated iterates can belong to the feasible set $\mathcal{M}$ if the initial point $\boldsymbol{x}^{(0)}$ is in $\mathcal{M}$.

Nevertheless, it is necessary in the FW algorithm to solve the minimization problem in each iteration. For this issue, For this issue, if the variable $\mathcal{M}$ can be *block-separable* as a cartesian product $\mathcal{M} = \mathcal{M}^{(1)} \times \mathcal{M}^{(2)} \times \cdots \times \mathcal{M}^{(n)} \subset \mathbb{R}^m$ over $n \geq 1$, we can perform a *single cheaper* update on only $\mathcal{M}^{(i)}$ instead of on an entire of $\mathcal{M}$. In this line of algorithms, the block-coordinate Frank-Wolfe (BCFW) algorithm has been proposed, for example, in the structural SVM problem in [19] and in the MAP inference [20]. This algorithm can be applied to the constrained convex problem of the form

$$\min_{\boldsymbol{x}\in\mathcal{M}^{(1)}\times\mathcal{M}^{(2)}\times\cdots\times\mathcal{M}^{(n)}} f(\boldsymbol{x}).$$

We assume that each factor $\mathcal{M}^{(i)}$ is convex, with $m = \sum_{i=1}^n m_i$. We solve the subproblem on the factor which is selected randomly. As a result, the BCFW algorithm can be implemented in cheaper iteration. When $n = 1$, this algorithm is reduced to the FW algorithm.

2.4 Block-coordinate Frank-Wolfe (BCFW) for semi-relaxed OT problem

Our previous paper addresses the semi-relaxed problem with $\Phi(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2\lambda} \| \boldsymbol{x} - \boldsymbol{y} \|_2^2$ because it is not only smooth but also convex [6]. This problem is formally defined as

$$\min_{\substack{\mathbf{T}\geq\mathbf{0},\\ \mathbf{T}^T\mathbf{1}_m=\boldsymbol{b}}} \left\{ f(\mathbf{T}) := \langle \mathbf{T}, \mathbf{C}\rangle + \frac{1}{2\lambda} \| \mathbf{T}\mathbf{1}_n - \boldsymbol{a} \|_2^2 \right\}, \tag{5}$$

where λ is a *relaxation* parameter. The domain is transformed into $\mathcal{M} = b_1\Delta_m \times b_2\Delta_m \times \cdots \times b_n\Delta_m$, where $b_i\Delta_m$ represents the simplex of the summation b_i. After following Frank-Wolfe (FW) algorithm, we describe a block-coordinate Frank-Wolfe (BCFW) algorithm for the semi-relaxed optimal transport problem.

Frank-Wolfe (FW) algorithm. The gradient $\nabla f(\mathbf{T}) \in \mathbb{R}^{mn}$ is given as $(\nabla_1 f(\mathbf{T})^T, \ldots, \nabla_n f(\mathbf{T})^T)^T$ where $\nabla_i f(\mathbf{T}) \in \mathbb{R}^m$ represents the gradient on the i-th variable block $b_i\Delta_m$. The linear subproblem is equivalent to

$$\boldsymbol{s}_i = b_i\boldsymbol{e}_j = b_i \arg\min_{\boldsymbol{e}_k\in\Delta_m, k\in[m]} \langle \boldsymbol{e}_k, \nabla_i f(\mathbf{T}^{(k)})\rangle, \tag{6}$$

where $j \in [m]$ and $\boldsymbol{e}_j$ is the extreme point on probability simplex [21]. The computational cost of the subproblem (6) is greatly improved from $O(n^3 \log n)$ to $O(n)$. A line-search algorithm can be applicable to search an optimal stepsize γ. Concretely, we solve $\min_{\gamma\in[0,1]} f((1-\gamma)\boldsymbol{x} + \gamma\boldsymbol{s})$, and calculate γ directly since the objective of the semi-relaxed problem is quadratic.

Block-coordinate Frank-Wolfe (BCFW) algorithm. The separability of the domain (5) enables us to develop the block-coordinate Frank-Wolfe algorithm for the semi-relaxed problem. The subproblem is identical to (6), but we solve the subproblem only for the i-th column, which is selected randomly. Then, all the other columns of $\mathbf{T}$ remain the same. Similarly to the FW algorithm, an exact line-search (ELS) algorithm can be also used. The optimal stepsize γ_{LS} is calculated as

$$\gamma_{\mathrm{LS}} = \frac{\lambda\langle \boldsymbol{t}_i^{(k)} - \boldsymbol{s}_i, \boldsymbol{c}_i\rangle + \langle \boldsymbol{t}_i^{(k)} - \boldsymbol{s}_i, \mathbf{T}^{(k)}\mathbf{1}_n - \boldsymbol{a}\rangle}{\| \boldsymbol{t}_i^{(k)} - \boldsymbol{s}_i \|^2} \tag{7}$$

where $\boldsymbol{t}_i$ is the i-th column of $\mathbf{T}$, and $\boldsymbol{s}_i$ is the solution of the i-th subproblem in (6). The duality gap can be used for the stopping criterion, and in a practical implementation, we monitor the value of the duality gap because the subproblem is solved at every iteration.

3. Block-coordinate Frank-Wolfe with adaptive sampling (BCFW-GA)

We construct in this paper the BCFW with gap sampling for semi-relaxed optimal transport problem according to methods [8], [22] because they address the duality gap whereas others mainly focus on the Lipschitz constants of the gradients [7], [8], [9], [10]. The main idea behind our proposed approach is as follows: Because the columns with larger duality gaps admit higher improvement to the objective function value, such columns should be sampled more often. In this way, we try to make more significant progress than the uniform-sampling method. For this purpose, after update of $\boldsymbol{t}_i$, the proposed BCFW-GA updates the duality gap for each column. Here, note that $g(\mathbf{T})$ is given as

$$\begin{aligned} g(\mathbf{T}) &= \langle \mathbf{T} - \mathbf{S}, \mathbf{C}\rangle + \frac{1}{\lambda}\langle \mathbf{T}\mathbf{1}_n - \mathbf{S}\mathbf{1}_n, \mathbf{T}\mathbf{1}_n - \boldsymbol{a}\rangle \\ &= \sum_{i=1}^{n}\langle \boldsymbol{t}_i - \boldsymbol{s}_i, \boldsymbol{c}_i\rangle + \frac{1}{\lambda}\langle \sum_{i=1}^{n}(\boldsymbol{t}_i - \boldsymbol{s}_i), \mathbf{T}\mathbf{1}_n - \boldsymbol{a}\rangle = \sum_{i=1}^{n} g_i(\mathbf{T}), \end{aligned}$$

where $g_i(\mathbf{T})$ is given by $g_i(\mathbf{T}) = \langle \boldsymbol{t}_i - \boldsymbol{s}_i, \boldsymbol{c}_i\rangle + \frac{1}{\lambda}\langle \boldsymbol{t}_i - \boldsymbol{s}_i, \mathbf{T}\mathbf{1}_n - \boldsymbol{a}\rangle. \forall i \in [n]$. Therefore, updating the column-wise duality gap $g_i(\mathbf{T})$ every iteration, we select an index i at random in proportion to the probability generated from $(g_1(\mathbf{T}), g_2(\mathbf{T}), \ldots, g_n(\mathbf{T}))$.

In the meantime, the update of $g_i(\mathbf{T})$ apparently depends on $\mathbf{T}$. Hence, every time one single $\boldsymbol{t}_i$ is updated, it is necessary to re-calculate $g_i(\mathbf{T})$ of all other $(n-1)$ columns to obtain its correct probability. Nevertheless, this is intractable, and wastes the benefit of the block coordinate approach. Therefore, in practice, at every $M \times n$ iterations, we periodically update $g_i(\mathbf{T})$ of all the columns to obtain their exact values. This update is specifically called the *global update* in this paper, and the loop of this global update is called an *outer iteration*. In contract to the outer iteration, the update of single $g_i(\mathbf{T})$ within the cycle of the global update is called an *inner iteration*. Within the global update period, i.e., the inner iteration, we store the calculated $g_i(\mathbf{T})$ for each i-th column, and do not perform the global update for the other columns. For the update of $g_j(\mathbf{T})$ of the j-th column ($j \neq i$), we utilize the stored latest (but outdated) $g_j(\mathbf{T})$. Hence, we expect that, when M is reasonably small, the convergence can be achieved, otherwise not.

4. Theoretical analysis

We analyze the convergence behaviour of BCFW-GA proposed in the previous section. In addition, we reveal computational complexity and the worst convergence iteration of the proposed algorithm. In the presentation, we will present them.

5. Numerical evaluations in color transfer problem

We compare BCFW-GA with our previously proposed algorithm. In addition, we investigate the effectiveness of semi-relaxed optimal transport problem for color transfer problem. In the presentation, we will show these numerical evaluation results.

References

[1] Peyre, G. and Cuturi, M.: Computational Optimal Transport, *Foundations and Trends in Machine Learning*, Vol. 11, No. 5-6, pp. 355–607 (2019).
[2] Cuturi, M.: Sinkhorn Distances: Lightspeed Computation of Optimal Transportation Distances, *NIPS* (2013).
[3] Chizat, L., Peyré, G., Schmitzer, B. and Vialard, F.-X.: Scaling algorithms for unbalanced optimal transport problems, *Mathematics of computation*, Vol. 87, pp. 2563–2609 (2018).
[4] Rabin, J., Ferradans, S. and Papadakis, N.: Adaptive color transfer with relaxed optimal transport, *ICIP* (2014).
[5] Frogner, C., Zhang, C., Mobahi, H., Araya-Polo, M. and Poggio, T.: Learning with a Wasserstein Loss, *NIPS* (2015).
[6] 福永, 拓. and 笠井, 裕.: 緩和最適輸送問題のための Frank-Wolfe アルゴリズム高速化手法と色転写問題への応用, Technical report, Waseda University (2021).
[7] Nesterov, Y.: Efficiency of Coordinate Descent Methods on Huge-Scale Optimization Problems, *SIAM Journal on Optimization*, Vol. 22, No. 2, pp. 341–362 (2012).
[8] Perekrestenko, D., Cevher, V. and Jaggi, M.: Faster Coordinate Descent via Adaptive Importance Sampling, *AISTATS* (2017).
[9] Needell, D., Srebro, N. and Ward, R.: Stochastic Gradient Descent, Weighted Sampling, and the Randomized Kaczmarz algorithm, *NIPS* (2014).
[10] Zhao, P. and Zhang, T.: Stochastic Optimization with Importance Sampling, *ICML* (2015).
[11] Ferradans, S., Papadakis, N., Peyré, G. and Aujol, J.-F.: Regularized Discrete Optimal Transport, *SIAM Journal on Imaging Sciences*, Vol. 7, No. 3, pp. 1853–1882 (2013).
[12] Kusner, M., Sun, Y., Kolkin, N. and Weinberger, K.: From Word Embeddings To Document Distances, *ICML* (2015).
[13] Kolkin, N., Salavon, J. and Shakhnarovich, G.: Style Transfer by Relaxed Optimal Transport and Self-Similarity, *CVPR* (2019).
[14] Qiu, T., Ni, B., Liu, Z. and Chen, X.: Fast Optimal Transport Artistic Style Transfer, *MultiMedia Modeling* (2021).
[15] Blondel, M., Seguy, V. and Rolet, A.: Smooth and Sparse Optimal Transport, *AISTATS* (2018).
[16] Benamou, J.-D.: Numerical resolution of an "unbalanced" mass transport problem, *ESAIM: M2AN*, Vol. 37, No. 5, pp. 851–868 (2003).
[17] Vincent-Cuaz, C., Flamary, R., Corneli, M., Vayer, T. and Courty, N.: Semi-Relaxed Gromov Wasserstein Divergence With Applications On Graphs, *arXiv preprint: arXiv:2110.02753* (2021).
[18] Dostl, Z.: *Optimal Quadratic Programming Algorithms: With Applications to Variational Inequalities*, Springer Publishing Company, Incorporated, 1st edition (2009).
[19] Lacoste-Julien, S., Jaggi, M., Schmidt, M. and Pletscher, P.: Block-Coordinate Frank-Wolfe Optimization for Structural SVMs, *ICML* (2013).
[20] Swoboda, P. and Kolmogorov, V.: MAP inference via Block-Coordinate Frank-Wolfe Algorithm, *CVPR* (2019).
[21] Clarkson, K. L.: Coresets, Sparse Greedy Approximation, and the Frank-Wolfe Algorithm, *ACM Transactions on Algorithms*, Vol. 6, No. 4 (2010).
[22] Osokin, A., Alayrac, J.-B., Lukasewitz, I., Dokania, P. and Lacoste-Julien, S.: Minding the Gaps for Block Frank-Wolfe Optimization of Structured SVMs, *ICML* (2016).