Stochastic Proximal Algorithms with SON Regularization: Towards Efficient Optimal Transport for Domain Adaptation

Anonymous Author
Anonymous Institution

Abstract

We propose a new regularizer for optimal transport (OT) which is tailored to better preserve the class structure of the subjected process. Accordingly, we provide the first theoretical guarantees for an OT scheme that respects class structure. We derive an accelerated proximal algorithm with a closed form projection and proximal operator scheme thereby affording a highly scalable algorithm for computing optimal transport plans. We provide a novel argument for the uniqueness of the optimum even in the absence of strong convexity. Our experiments show that the new regularizer does not only result in a better preservation of the class structure in the data but also in additional robustness to the data geometry, relative to previous regularizers.

1 Introduction

Optimal Transport (OT), first proposed by Monge as an analysis problem [Monge, 1781], has become a classic topic in probability and statistics for transferring mass from one probability distribution to another [Villani, 2008, Santambrogio, 2015]. The OT problem seeks to find a transport map from a source distribution to a target distribution while minimizing the cost of the transport. As a richly adopted framework in many different disciplines, OT has also recently been very successfully used in many applications in computer vision, texture analysis, tomographic reconstruction and clustering, as documented in the recent surveys [Kolouri et al., 2017] and [Solomon, 2018]. In many of these applications, OT exploits the geometry of the underlying spaces to effectively yield improved performance over the alternative of obviating it. This improvement, however, comes at a significant computational cost when solving the OT problem. Much attention has focused on efficient computational and numerical algorithms for OT, and a monograph focusing on this topic appeared [Peyré and Cuturi, 2018]. In [Guo et al., 2020a], an "accelerated primal-dual randomized coordinate descent (APDRCD)" algorithm is developed to solve the OT problem. An upper bound is also provided for the complexity of the algorithm and it is shown that it could be used for large-scale purposes.

Several advances in computational approaches to OT have been made in recent years, primarily focusing on applications in domain adaptation. In [Courty et al., 2017], a generalized conditional gradient method is used to compute OT with the help of a couple of regularizers. Cuturi introduced an entropic regularizer and showed that its adoption with the Sinkhorn algorithm yields a fast computation of OT [Cuturi, 2013]; a theoretical guarantee that the Sinkhorn iteration computes the approximation in near linear time was also provided by [Altschuler et al., 2017]. Screenkhorn algorithm proposed in [Alaya et al., 2019] performs screening to eliminate (and accelerate) the solution of the Sinkhorn algorithm. Another computational breakthrough was achieved by [Genevay et al., 2016] who gave a stochastic incremental algorithm to solve the entropic regularized OT problem.

While the entropic regularization of OT has attracted a lot of attention on account of its many merits, it has some limitations, such as the blurring in the optimal transportation plan induced by the entropy term. An intrinsic property of the entropy term is that it keeps the transportation plan strictly positive and therefore completely dense, unlike unregularized OT. This lack of sparsity can be problematic for applications such as domain adaptation [Courty et al., 2017] where the optimal transportation plan is itself of interest and a class structure is present in the data. In this case, an ideal transport plan maps each class in the source domain to its corresponding class in the target domain. In practice, we should transfer one source class to as few target classes as possible. An amelioration of this effect may be achieved by using a small regularization so long as it is carefully engineered, which is generally difficult.

Preliminary work. Under review by AISTATS 2023. Do not distribute.

A case for exploring new regularizers was made in [Courty et al., 2017] in the context of domain adaptation applications. In [Blondel et al., 2018] the primal and dual formulations of OT are regularized with a strongly convex term, and the constraints are relaxed with smooth approximations. [Dessein et al., 2018] also propose a framework to solve discrete optimal transport problems with smooth convex regularization. The L2 regularization of OT plan also yields a sparse plan. It has notably been used in a doubly stochastic scheme in [Seguy et al., 2018]. While promoting sparsity in the transport plan, these techniques are not tailored to the underlying class structure of the data, which may not be known in advance. In this paper, we accordingly propose a novel approach of class–based regularization of the OT problem, based on the recently proposed convex clustering framework of Sum of Norms (SON) [Lindsten et al., 2011a, Hocking et al., 2011], which presents an improvement on the state of the art on at least two grounds:

SON Regularizer Benefits for OT: The SON regularization allows one to discover and exploit the class structure and to preserve the sparsity of the transport plan. While this approach may be superficially reminiscent of a Laplacian regularizer [Courty et al., 2017], the latter only acts indirectly on the transported points and is quadratic in nature, in contrast to our transport plan. This difference is clearly illustrated in the experiments. We theoretically show and experimentally validate that this formulation ensures a transport plan adhering to the class structure. In the source domain, the class structure is given by the labels while it is latent (hidden) in the target domain. We further show that our formulation leads to the discovery of the underlying hidden class structure in the target domain, and provide for the first time, rigorous guarantees on the recovery of class structure. No such results, to the best of our knowledge, are known for other regularizers. We also experimentally show that our regularizer does not only yield a better class structure preservation, but also provides additional robustness compared to other class-based regularizers in [Courty et al., 2017].

Computational Benefits of Stochastic Proximal Algorithm: Our SON regularizer-based formulation also enjoys computational benefits – we propose a scalable stochastic incremental algorithm which operates in the primal formulation and explicitly produces the optimal coupling. In contrast to [Courty et al., 2017] where full gradients are used, our algorithm operates in a stochastic incremental framework. Proximal method has been used in OT but not in a stochastic scheme. Examples include [Alvarez-Melis et al., 2018] that uses notions of submodularity and also [Papadakis et al., 2014]. We first construct an abstract stochastic framework that is based on a combination of proximal and projection iterations , for which we give a generic proof of convergence at rate $O(1/T)$. We also demonstrate its stability in a number of experiments. We subsequently specialize this general scheme for our SON regularizer-based formulation, which leads to an algorithm with computationally low-cost iterations. Beyond the proposed SON-based framework, our proximal scheme can be used to avoid the reported convergence difficulties gradient-based methods [Patrascu and Necoara, 2018].

It is notable that our formulation differs from the *partial domain adaption* [Cao et al., 2018] and *robust domain adaption* [Balaji et al., 2020] settings developed to deal with outlier classes or samples. The former assumes some special form of relationship between the source and target classes, whereas, we do not make such assumptions. Furthermore, the partial domain adaptation literature has not been developed yet around the idea of optimal transport. However, we observe that our framework is able to address the partial domain adaptation task, though it is not specialized for this purpose. The latter proposes a new formulation for OT to reduce the sensitivity to outlier samples. It does not take the class structure in source and target into account.

Summary of contributions: Our main contributions can be summarized as follows:

i.We propose in Section 2 a new regularized formulation of OT that promotes the sparsity of the transport plan, thereby ensuring a preservation of a class structure typically arising in domain adaptation problems.
ii. In Section 2.1, we develop a new proof for the uniqueness of the solution optimum of our convex formulation in spite of its non-strong convexity. We believe that this proposed technique may have wider applicability.
iii. We develop in Section 3 a general accelerated stochastic incremental proximal-projection optimization scheme, for which we give a proof of convergence at a rate $O(1/T)$ without a decaying step size. We specialize the general scheme with an explicit closed form of proximal operators and fast projections to yield a scalable stochastic incremental algorithm for computing our OT formulation, thus producing an optimal coupling (transport plan) explicitly.
iv. We derive the first rigorous results for recovering an OT plan that respects class structure, introduced in Section 4 with details in the appendix 7.
v. Finally in Section 5, we investigate the algorithm on several synthetic and benchmark data sets, and demonstrate the benefits of the new regularizer.

2 Optimal Transport with SON regularization

Consider two finite sets $\{\mathbf{y}_i^s\}_{i=1}^m, \{\mathbf{y}_j^t\}_{j=1}^n$ of points, respectively sampled from the so-called source and target domains. Let $\mathbf{D} = (D_{ij} = d(\mathbf{y}_i^s, \mathbf{y}_j^t))$ be the $m \times n$ dis-

tance matrix with D_{ij} representing a distance between the i^{th} point in the source domain and the j^{th} point in the target domain, being used as the transportation cost of a unit mass between them. We denote the i^{th} row and j^{th} column of $\mathbf{D}$ by $\mathbf{d}^i$ and $\mathbf{d}_j$, respectively. We let the positive probability masses μ_i, ν_j be respectively assigned to the data points $\mathbf{y}_i^s$ and $\mathbf{y}_j^t$. In this discrete setup, the Monge problem amounts to finding a one to one assignment between the points in the two domains (assuming that $m = n$) with a minimal cost, that transforms the source distribution $\{\mu_i\}$ to the target distribution $\{\nu_j\}$ (if feasible). This is generally considered to be a difficult and highly ill-posed problem to solve and hence its linear programming (LP) relaxation, known as the Kantarovich problem is more widely considered, which can be written as

$$\min_{\mathbf{X}\in B(\boldsymbol{\mu},\boldsymbol{\nu})} \langle \mathbf{D}, \mathbf{X}\rangle. \tag{2.1}$$

Here, the variable matrix $\mathbf{X} = (x_{i,j})$ is known as the transport map and $B(\boldsymbol{\mu}, \boldsymbol{\nu}) = \{\mathbf{X} \in R^{m\times n}, \mathbf{X}\mathbf{1}_{n^s} = \boldsymbol{\mu}, \mathbf{X}^T\mathbf{1}_{n^t} = \boldsymbol{\nu}\}$ is the set of all coupling distributions between $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$, respectively denoting the vectors of elements μ_i, ν_j. Moreover, $\langle \mathbf{D}, \mathbf{X}\rangle = \mathrm{Tr}(\mathbf{D}^T\mathbf{X}) = \sum_{i,j} X_{ij}D_{ij}$ is the Euclidean inner product of two matrices. In an ideal case, one hopes that the optimal solution for $\mathbf{X}$ become an assignment (permutation matrix) in which case it is seen to coincide with the solution of the Monge problem. On account of numerical difficulties and statistical instability, the Kantarovich problem is widely used by applying further regularization. In this respect, we introduce the following flexible convex optimization framework for optimal transport via the so-called SON regularizer:

$$\mathbf{X}^* = \arg\min_{\mathbf{X}\in B(\boldsymbol{\mu},\boldsymbol{\nu})} \langle \mathbf{D}, \mathbf{X}\rangle + \lambda \left(\sum_{l,k} R_{l,k}\|\mathbf{x}_l - \mathbf{x}_k\|_2 + \sum_{l,k} S_{l,k}\|\mathbf{x}^l - \mathbf{x}^k\|_2 \right), \tag{2.2}$$

where $\mathbf{x}_l$ and $\mathbf{x}^k$ denote the (transpose of the) l^{th} row and k^{th} column of $\mathbf{X}$, respectively, and $S_{l,k}, R_{l,k}$ are positive kernel coefficients on rows and columns of $\mathbf{X}$, respectively. λ is a tuning parameter. Comparing equation 2.1 to equation 2.2, we observe that the second line of equation 2.2 serves as the proposed regularizer.

The effect of the regularization in equation 2.2 is explained in the original SON paper [Lindsten et al., 2011b]. In short, it enforces many vanishing regularization terms (sparsity), hence yielding identical columns and identical rows in the solution. In other words, the resulting map $\mathbf{X}^*$ after a suitable permutation of rows and columns is a block matrix with constant values in each block. Thus if the data in the source and target domains has a clear partitioning structure as in the well known stochastic block model, then the recovered blocks will reflect such a structure. We show that similar to standard Kanturovich relaxation, under suitable conditions related to the well known stochastic block model, the constraints will further force many blocks to be zero. Each row and column will contain exactly one nonzero block, and the solution reflects an assignment consistent of the classes, rather than individual samples. This is made precise and proved in Section 4. The regularization parameter λ sets a desired balance between the cluster structure and the underlying transport problem. When $\lambda = 0$, equation 2.2 reduces to the standard optimal transport approach by Kantorovich relaxation. For large values of λ, the SON regularization dominates the result. In a typical situation, $\lambda \to \infty$ results in all data in each domain assigned to the same cluster, hence a trivial transformation between single clusters in each domain. Smaller values of λ lead to a larger number of identified classes in the solution. Note that the Laplacian regularizer in [Courty et al., 2017] acts only indirectly on the transported points and is quadratic, whereas ours acts directly on the transport plan and is of ℓ_1 type. Compared to the entropy regularization, which accounts for the mean energy of the distribution (per Boltzman derivation of entropy), the SON formulation specifies a more refined characterization of the distributions. The blurry transport map resulting from an entropy regularization is widely attributed to the dependence on the mean energy, which is avoided in the SON formulation.

The framework in equation 2.2 is useful, especially when the source samples $\mathbf{y}_i^s$ are readily assigned to different classes and the optimal transport is additionally required to map the points within each class to identical or similar points in the target domain. In this case, we may set $R_{l,k} = 0$ if $\mathbf{y}_l^s$ and $\mathbf{y}_k^s$ are in different classes, otherwise set $R_{l,k} = k_s(\mathbf{y}_l^s, \mathbf{y}_k^s)$ for a suitable (differentiable) kernel k_s. On the target side where no class information is ordinarily provided, we may set $S_{l,k} = k_t(\mathbf{y}_l^t, \mathbf{y}_k^t)$ for a suitable kernel k_t of choice. The framework also allows one to use different penalty hyperparameters λ_1 on the rows, and λ_2 on the columns by incorporating them into R and S respectively.

2.1 Uniqueness

An elementary question concerning any optimization formulation, including the Kantorovich problem and its regularization in equation 2.2, is the uniqueness of their optimal solution, and a standard method for verifying uniqueness is to establish strong convexity of the objective function. Even though it is seen that the objective in (2.2) is not strongly convex, we are nevertheless able to identify conditions, under which the solution still remains unique. For this, we develop an alternative approach, which is not only useful in our framework, but may also be generically used in many similar problems including a wide range of linear programming (LP) relaxation problems, and for this reason it is first presented. Our approach is based on the following definition:

Definition 1. *We call a (global) optimal solution $\mathbf{X}_0$ of a convex optimization problem*

$$\min_{\mathbf{X}\in\mathscr{S}} \mathscr{F}(\mathbf{X}),$$

where $\mathscr{F}(.)$ is a convex function and $\mathscr{S}$ is a convex set, a **resistant optimal point** *if adding a linear perturbation term $\langle\tilde{\mathbf{D}},\mathbf{X}\rangle$ with sufficiently small coefficients in $\tilde{\mathbf{D}}$ to the objective leads to an arbitrarily small perturbation of the solution $\mathbf{X}_0$. In mathematical terms for any open neighborhood $\mathcal{N}$ of $\mathbf{X}_0$ there exists an open neighborhood $\mathcal{M}$ of $\tilde{\mathbf{D}}$ around $\tilde{\mathbf{D}}=\mathbf{0}$ such that*

$$\forall\tilde{\mathbf{D}}\in\mathcal{M},\quad \mathcal{N}\cap\arg\min_{\mathbf{X}\in\mathscr{S}}\mathscr{F}(\mathbf{X})+\langle\tilde{\mathbf{D}},\mathbf{X}\rangle\neq\emptyset.$$

Accordingly, we have the following result:

Theorem 2. *A resistant optimal point of a convex optimization problem is its unique optimal point.*

Proof. Suppose that there exists a different optimal point $\mathbf{X}'$. Take $\mathbf{D}_0=\frac{\mathbf{X}_0-\mathbf{X}'}{\|\mathbf{X}_0-\mathbf{X}'\|}$, $r=\|\mathbf{X}_0-\mathbf{X}'\|$ and $\tilde{\mathbf{D}}=\epsilon\mathbf{D}_0$ for arbitrary $\epsilon>0$. Further, define $\mathcal{N}$ as the ball of radius $\delta=r/2$ centered at $\mathbf{X}_0$. Note that for each $\mathbf{Y}\in\mathcal{N}$ we have

$$\mathscr{F}(\mathbf{Y})+\langle\tilde{\mathbf{D}},\mathbf{Y}\rangle\geq\mathscr{F}(\mathbf{X}_0)+\langle\tilde{\mathbf{D}},\mathbf{Y}\rangle=$$

$$\mathscr{F}(\mathbf{X}')+\langle\tilde{\mathbf{D}},\mathbf{X}'\rangle+\langle\tilde{\mathbf{D}},(\mathbf{Y}-\mathbf{X}_0)+(\mathbf{X}_0-\mathbf{X}')\rangle.$$

Now, note that $\langle\tilde{\mathbf{D}},(\mathbf{Y}-\mathbf{X}_0)+(\mathbf{X}_0-\mathbf{X}')\rangle\geq-\delta\epsilon+r\epsilon>0$, which establishes

$$\mathscr{F}(\mathbf{Y})+\langle\tilde{\mathbf{D}},\mathbf{Y}\rangle>\mathscr{F}(\mathbf{X}')+\langle\tilde{\mathbf{D}},\mathbf{X}'\rangle.$$

Hence, $\mathcal{N}\cap\arg\min_{\mathbf{X}\in\mathscr{S}}\mathscr{F}(\mathbf{X})+\langle\tilde{\mathbf{D}},\mathbf{X}\rangle=\emptyset$ and since $\epsilon=\|\tilde{\mathbf{D}}\|$ is arbitrarily small, we conclude that $\mathbf{X}_0$ is not a resistant optimal point. This contradicts the assumption and shows that the solution is unique. □

Theorem 2 is a general way to establish uniqueness. In fact, we can show that the strong convexity condition is a special case of this result:

Theorem 3. *If $\mathscr{F}$ is continuous and strongly convex, then the global minimal point of $\mathscr{F}$ over a convex set $\mathscr{S}$ is resistant.*

Proof. Denote the optimal point by $\mathbf{X}^*$. By strong convexity, there exists a $\gamma>0$ such that for any feasible point $\mathbf{X}\in\mathscr{S}$, we have $\mathscr{F}(\mathbf{X})-\mathscr{F}(\mathbf{X}^*)\geq\frac{\gamma}{2}\|\mathbf{X}-\mathbf{X}^*\|_{\mathrm{F}}^2$. Take $\mathscr{G}=\mathscr{F}+\langle\tilde{\mathbf{D}},\mathbf{X}\rangle$ and note that $\mathscr{G}(\mathbf{X})-\mathscr{G}(\mathbf{X}^*)\geq\frac{\gamma}{2}\|\mathbf{X}-\mathbf{X}^*\|_{\mathrm{F}}^2+\langle\tilde{\mathbf{D}},\mathbf{X}-\mathbf{X}^*\rangle\geq\frac{\gamma}{4}\|\mathbf{X}-\mathbf{X}^*\|_{\mathrm{F}}^2-\frac{2}{\gamma}\|\tilde{\mathbf{D}}\|_{\mathrm{F}}^2$. This shows that $\mathscr{G}>\mathscr{G}(\mathbf{X}^*)$ and hence does not have any global optimal point outside the closed sphere $\{\mathbf{X}\mid\|\mathbf{X}-\mathbf{X}^*\|_{\mathrm{F}}\leq\frac{\sqrt{8}}{\gamma}\|\tilde{\mathbf{D}}\|_{\mathrm{F}}\}$. Since $\mathscr{G}$ is continuous, it also attains a minimum inside the sphere, which then becomes the global optimal point. We conclude that for any $\epsilon>0$, taking $\|\tilde{\mathbf{D}}\|<\frac{\gamma\epsilon}{\sqrt{8}}$ leads to an optimal solution inside a ball of radius ϵ centered at $\mathbf{X}^*$. This shows that the solution is resistant. □

Uniqueness for equation 2.2: One special case of resistant optimal points, that will be useful in our analysis, is when there exists a neighborhood $\mathcal{M}$ of $\mathbf{0}$ such that

$$\forall\tilde{\mathbf{D}}\in\mathcal{M},\quad \mathbf{X}^*\in\arg\min_{\mathbf{X}\in\mathscr{S}}\mathscr{F}(\mathbf{X})+\langle\tilde{\mathbf{D}},\mathbf{X}\rangle.$$

We call such a resistant optimal point an **extremal optimal point**. Later, we consider an analysis where we give conditions on $\mathbf{D}$ to ensure that a desired solution $\mathbf{X}^*$ is achieved. Our strategy for uniqueness in this analysis is to show that under the same conditions, the desired optimal point is also extremal and hence unique, according to Theorem 1. In the case of the problem in equation 2.2, adding the term $\langle\tilde{\mathbf{D}},\mathbf{X}\rangle$ modifies the cost matrix $\mathbf{D}$ to $\mathbf{D}+\tilde{\mathbf{D}}$. Hence, being an extremal optimal point is in this case equivalent to the solution $\mathbf{X}^*$ being maintained following a perturbation of the matrix $\mathbf{D}$ in a sufficiently small open neighborhood. This is easy to achieve in our planted model analysis, because the optimality of $\mathbf{X}^*$ is guaranteed by a set of inequalities on $\mathbf{D}$, which remain valid under small perturbations, simply by requiring the inequalities to be strict. As seen, Theorem 2 and extremal optimality, in particular, can be powerful tools for establishing uniqueness beyond strong convexity.

3 Stochastic Incremental Algorithms

3.1 Accelerated Proximal-Projection Scheme

An important advantage of the framework in equation 2.2 is the possibility of applying stochastic optimization techniques. Since the objective term includes a large number of non-smooth SON terms, our stochastic optimization avoids calculating the (sub)gradient or the proximal operator of the entire objective function, which is numerically infeasible for large-scale problems. Our algorithm is obtained by introducing the following "template function":

$$\phi_{\rho,\boldsymbol{\zeta},\boldsymbol{\eta}}(\mathbf{p},\mathbf{q})=\langle\mathbf{p},\boldsymbol{\zeta}\rangle+\langle\mathbf{q},\boldsymbol{\eta}\rangle+\rho\|\mathbf{p}-\mathbf{q}\|_2,\qquad(3.1)$$

and noting that the objective function in equation 2.2 can be written as

$$\sum_{l\neq k}\phi_{R_{l,k},\frac{1}{2(n-1)}\mathbf{d}_l,\frac{1}{2(n-1)}\mathbf{d}_k}(\mathbf{x}_l,\mathbf{x}_k)+\sum_{l\neq k}\phi_{S_{l,k},\frac{1}{2(m-1)}\mathbf{d}^l,\frac{1}{2(m-1)}\mathbf{d}^k}(\mathbf{x}^l,\mathbf{x}^k),\qquad(3.2)$$

with a total number of $P=m(m-1)+n(n-1)$ summands in the form of the template function. This places the

problem in the setting of *finite sum* optimization problems [Bottou et al., 2018]. However, there are two obstacles to the application of stochastic optimization techniques: First, the terms in (3.2) are not smooth, so gradient methods do not apply and second, equation 2.2 involves a fairly complex constraint. We address these issues in the following.

Non–Smooth Terms: We exploit the highly effective proximal methodology for optimizing non–smooth functions [Parikh and Boyd, 2016, Combettes and Pesquet, 2011] using a proximal operator. Defazio further gives a stochastic acceleration technique using proximal operators for unconstrained problems [Defazio, 2016]. In addition to its fast convergence, the main advantage of this scheme is its potential constant step size convergence in contrast to the ordinary stochastic gradient approach. It unfortunately does not address constrained optimization problems.

Constrained Optimization: Facing a constrained optimization problem, the calculation of the proximal operators over the feasible set is numerically intractable. However, we observe an appealing structure in the constraint which lends itself to a more efficient stochastic implementation: Recalling the definition of an $n-$dimensional standard simplex

$$S^{(n)} = \left\{ \mathbf{x} = (x_i \geq 0)_{i=1}^n \mid \sum_i x_i = 1 \right\},$$

we define the weighted cylinder-simplices $S_l(\mu) = \{\mathbf{X} \mid \mathbf{x}_l \in \mu S^{(n)}\}$ and $S^k(\nu) = \{\mathbf{X} \mid \mathbf{x}^k \in \nu S^{(m)}\}$ respectively corresponding to the l^{th} row and k^{th} column of $\mathbf{X}$ with weights $\mu, \nu \geq 0$. We then observe that the constraint set $B(\boldsymbol{\mu}, \boldsymbol{\nu})$ is equal to $B(\boldsymbol{\mu}, \boldsymbol{\nu}) = (\bigcap_{l=1}^m S_l(\mu_l)) \cap (\bigcap_{k=n}^m S^k(\nu_k))$, which is an intersection of $Q = m + n$ weighted cylinder-simplices.

In summary, the optimization problem in equation 2.2 can be written in the following abstract form:

$$\min_{x \in \mathbb{R}^D} \sum_{p=1}^{P} \phi_p(x) \quad \text{st} \quad x \in \bigcap_{q=1}^{Q} S_q, \tag{3.3}$$

where each term ϕ_p denotes a template function term in the objective and each set S_q is a weighted cylinder-simplex. The values of P, Q in equation 2.2 are given above and $D = mn$. [Bertsekas, 2011], [Wang and Bertsekas, 2016] and [Patrascu and Necoara, 2018] give general stochastic incremental schemes that combine gradient, proximal and projected schemes for optimizing such finite sum problems with convex constraints. These do not, however, use acceleration and their respective convergence is only guaranteed with a variable and vanishing step size, which is practically difficult to control and often yields extremely slow convergence.

Our Proposed Method: We herein jointly exploit the two ideas in [Defazio, 2016] and [Wang and Bertsekas, 2016] to obtain an accelerated proximal scheme for constrained framework in equation 2.2. Further, we shortly show in Lemma 5 that the proximal operator can be computed in closed form for our problem. Together with the projection to the simplex from [Condat, 2016, Duchi et al., 2008], this gives a stochastic incremental algorithm with much less costly iterations.

We extend the acceleration techniques of unconstrained optimization as in the Defazio's scheme (known as Point–SAGA) to the constrained setting. Point–SAGA utilizes individual "memory" vectors for each term in the objective function, which store a calculated subgradient of a selected term in every iteration. These vectors are subsequently used as an estimate of the subgradient at subsequent iterations. We extend this scheme by introducing similar memory vectors to constraints. Each memory vector $\mathbf{h}_m$ for a constraint S_m stores the last observed normal (separating) vector to S_m. At each iteration either an objective term ϕ_p or a constraint component S_q is considered by random selection. Accordingly, we propose the following rule for updating the solution:

$$\mathbf{x}_{t+1} = \begin{cases} \text{prox}_{\mu\phi_{p_t}}\left(\mathbf{x}_t + \mu \mathbf{g}_{p_t}\right), & \phi_{p_t} \text{ is selected} \\ \text{proj}_{S_{q_t}}\left(\mathbf{x}_t + \mu \mathbf{h}_{q_t}\right) & S_{q_t} \text{ is selected} \end{cases}, \tag{3.4}$$

where t is the iteration number, $\mu > 0$ is the fixed step size and p_t, q_t denote the selected index in this iteration (only one of them exists). At each iteration, the corresponding memory vector to the selected term is also updated. Depending on the choice of ϕ_{p_t} or S_{q_t}, either $\mathbf{g}_{p_t} \leftarrow \mathbf{g}_{p_t} + \mathbf{a}_t$ or $\mathbf{h}_{q_t} \leftarrow \mathbf{h}_{q_t} + \mathbf{a}_t$, where

$$\mathbf{a}_t = \rho \frac{\mathbf{x}_t - \mathbf{x}_{t+1}}{\mu} - \alpha \left(\sum_n \mathbf{g}_n + \sum_m \mathbf{h}_m \right), \tag{3.5}$$

where $\rho \in (0\ 1)$ and $\alpha > 0$ are design constants. The vector $\mathbf{a}_t$ consists of two parts: the first part $\rho \frac{\mathbf{x}_t - \mathbf{x}_{t+1}}{\mu}$ calculates a sub-gradient or a normal vector at point $\mathbf{x}_{t+1}$ corresponding to the selected term. The second term, the sum of the memory terms, implements acceleration. Our algorithm bears marked differences with Point-SAGA. While acceleration by the sum of memory vectors is also employed in Point-SAGA, it is moved in our scheme from the update rule of $\mathbf{x}_t$ to the update rule of $\mathbf{g}_t$. Also, the design parameters ρ and α are introduced to improve convergence. Similar to Point-SAGA we only need to calculate the sum of memory terms once in the beginning and later update it by simple manipulations. As we later employ initialization of the memory vectors by zero, the first summation trivially leads to zero.

Convergence Analysis We show that for a generic convex optimization problem of the form in equation 3.3, the algorithmic scheme in section 3.1 converges with a guaranteed rate, under the following mild assumptions:

Assumption 1. The functions ϕ_p are $\beta-$Lipschitz.

Assumption 2. We require the monotone inclusion problem

$$\mathbf{0} \in \sum_{p=1}^{P} \partial \phi_p(\mathbf{x}) + \sum_{q=1}^{Q} \partial I_{S_q}(\mathbf{x}) \tag{3.6}$$

to have a solution at $\mathbf{x} = \mathbf{x}^*$ with a finite optimal value ϕ^* and $\mathbf{g}_p^* \in \partial \phi_p(\mathbf{x}^*)$ and $\mathbf{h}_q^* \in \partial I_{S_q}(\mathbf{x}^*)$ satisfying $\sum_p \mathbf{g}_p^* + \sum_q \mathbf{h}_q^* = \mathbf{0}$. Furthermore, we assume that

$$\sum_p \left\|\mathbf{g}_p^*\right\|^2 + \sum_q \left\|\mathbf{h}_q^*\right\|^2 = O(\beta^2 R) \tag{3.7}$$

where $R = P + Q$ is the total number of terms.

Here, $\partial\phi(\mathbf{x})$ and $\partial I_S(\mathbf{x})$ respectively denote the subdifferential of the function ϕ and the cone of normal vectors to the set S at $\mathbf{x}$. It is well-known that any solution to equation 3.6 is an optimal feasible solution to equation 3.3.

Assumption 3. We assume that the algorithm is initialized with $\mathbf{g}_p = \mathbf{h}_q = \mathbf{0}$.

Then, we can show the following result.

Theorem 4. *Suppose that Assumption 1-3 are satisfied. Then for $\alpha, \rho, \mu > 0$ and $\alpha < 2(1-\rho)$ the following holds true:*

1. *Defining $\bar{\mathbf{x}}_t = \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbf{x}_\tau$ and $\eta = (1 + Q/P)(\|\mathbf{x}_0 - \mathbf{x}^*\|^2/\beta\mu + \beta\mu R)$ we have*

$$\mathbb{E}\left[\sum_p \phi_p(\bar{\mathbf{x}}_t)\right] - \phi^* \leq c\beta P\left(\frac{\eta}{t} + \sqrt{\frac{\beta\mu\eta}{t}}\right),$$

$$\mathbb{E}\left[\sum_q \operatorname{dist}^2(\bar{\mathbf{x}}_t, S_q)\right] \leq c\beta\mu P\frac{\eta}{t}, \tag{3.8}$$

   where c is a constant depending on ρ, α and the underlying constant in equation 3.7.

2. *Moreover,*

$$\sum_{\tau=0}^{\infty} \mathbb{E}[\|\mathbf{x}_{\tau+1} - \mathbf{x}_\tau\|^2] \leq c\frac{\mu\beta P}{P+Q}\eta. \tag{3.9}$$

Proof. The proof is given in section 10. □

Comment: As expected, the results only depend on $\beta\mu$, except for the optimality gap being linearly proportional to β. Applying this technique to our problem of interest and assuming that m, n are of the same order, we observe that $P = O(n^2)$ and $Q = O(n)$. Since many terms in our objective function are for regularization, it is fair to consider the relative optimality gap obtained by driving the optimality gap to the number of objective terms. We observe that this quantity is controlled by η/t. We conclude that $t \sim \eta$ iterations is required to achieve a desired relative optimality gap. The total feasibilty gap is controlled by $\beta\mu P\eta$. If we take $\beta\mu \sim 1/n$, we obtain $\eta \sim n$ and the relative optimality gap vanishes in $O(n)$ iterations. Then, the total optimality gap and feasibility gap will vanish in $O(n^3)$ and $O(n^2)$ iterations, respectively. In the absence of the regularization terms, we may reorganize the objective to have only $O(n)$ terms. In this case, taking $\beta\mu \sim 1/\sqrt{n}$, we get $\eta \sim O(\sqrt{n})$ and we require $O(n^{\frac{3}{2}})$ and $O(n)$ iterations to control the total optimality and feasibility gaps. Compared to the results of [Guo et al., 2020b], which establishes convergence in $O(n^{\frac{5}{2}})$ our convergence rates are better. Moreover, the $O(n^2)$ dependence can be improved by reducing the number of terms in the objective function. It has been pointed out that in the sum of norms approach many terms may be redundant and only $O(n)$ terms corresponding to pre-selected pairs (i, j) can be sufficient.

3.2 Proximal Operator for the SON-Regularized Kantorovich Relaxation

We next show that we can explicitly compute the proximal operator for each term in (3.2):

Theorem 5. *The proximal operator of the template function $\phi_{\rho,\boldsymbol{\zeta},\boldsymbol{\eta}}$ is given by $\mathcal{T}_{\mu\rho}(\mathbf{p} - \mu\boldsymbol{\zeta}, \mathbf{q} - \mu\boldsymbol{\eta})$, where*

$$\mathcal{T}_\lambda(\mathbf{a}, \mathbf{b}) = \left(\tfrac{\mathbf{a}+\mathbf{b}}{2} + \mathcal{T}_\lambda\left(\tfrac{\mathbf{a}-\mathbf{b}}{2}\right), \tfrac{\mathbf{a}+\mathbf{b}}{2} - \mathcal{T}_\lambda\left(\tfrac{\mathbf{a}-\mathbf{b}}{2}\right)\right), \tag{3.10}$$

and

$$\mathcal{T}_\lambda(\mathbf{c}) = \begin{cases} \frac{\|\mathbf{c}\|-\lambda}{\|\mathbf{c}\|}\mathbf{c} & \|\mathbf{c}\| \geq \lambda \\ 0 & \text{otherwise} \end{cases}.$$

Proof. The proximal operator of $\phi_{k,\boldsymbol{\zeta},\boldsymbol{\eta}}$ is defined as

$$\operatorname*{argmin}_{(\mathbf{x},\mathbf{y})\in\mathbb{R}^m\times\mathbb{R}^m} \frac{1}{2\mu}\|\mathbf{x} - \mathbf{p}\|_2^2 + \frac{1}{2\mu}\|\mathbf{y} - \mathbf{q}\|_2^2 + \phi_{\rho,\boldsymbol{\zeta},\boldsymbol{\eta}}(\mathbf{x}, \mathbf{y}). \tag{3.11}$$

We introduce a change of variables by $\mathbf{u} = (\mathbf{x}+\mathbf{y})/2, \mathbf{v} = (\mathbf{x}-\mathbf{y})/2$. First note that $\|a\|^2 + \|b\|^2 = (\|a+b\|^2 + \|a-b\|^2)/2$. Hence

$$\frac{1}{2\mu}\|\mathbf{x} - \mathbf{p}\|_2^2 + \frac{1}{2\mu}\|\mathbf{y} - \mathbf{q}\|_2^2 = \frac{1}{\mu}\left(\left\|\mathbf{u} - \frac{\mathbf{p}+\mathbf{q}}{2}\right\|^2 + \left\|\mathbf{v} - \frac{\mathbf{p}-\mathbf{q}}{2}\right\|^2\right)$$

Furthermore,

$$\langle\mathbf{x}, \boldsymbol{\zeta}\rangle + \langle\mathbf{y}, \boldsymbol{\eta}\rangle = \langle\mathbf{u}+\mathbf{v}, \boldsymbol{\zeta}\rangle + \langle\mathbf{u}-\mathbf{v}, \boldsymbol{\eta}\rangle = \langle\mathbf{u}, \boldsymbol{\zeta}+\boldsymbol{\eta}\rangle + \langle\mathbf{v}, \boldsymbol{\zeta}-\boldsymbol{\eta}\rangle$$

Hence equation 3.11 can be written as

$$\operatorname*{argmin}_{(\mathbf{u},\mathbf{v})\in\mathbb{R}^m\times\mathbb{R}^m} \frac{1}{\mu}\left(\left\|\mathbf{u} - \frac{\mathbf{p}+\mathbf{q}}{2}\right\|^2 + \left\|\mathbf{v} - \frac{\mathbf{p}-\mathbf{q}}{2}\right\|^2\right) + \langle\mathbf{u}, \boldsymbol{\zeta}+\boldsymbol{\eta}\rangle + \langle\mathbf{v}, \boldsymbol{\zeta}-\boldsymbol{\eta}\rangle + 2\rho\|\mathbf{v}\|_2 \tag{3.12}$$

$$= \underset{(\mathbf{u},\mathbf{v})\in\mathbb{R}^m\times\mathbb{R}^m}{\operatorname{argmin}} \frac{1}{\mu}\left(\left\|\mathbf{u} - \frac{\mathbf{p}+\mathbf{q}-\mu\boldsymbol{\zeta}-\mu\boldsymbol{\eta}}{2}\right\|^2\right) + \frac{1}{\mu}\left(\left\|\mathbf{v} - \frac{\mathbf{p}-\mathbf{q}-\mu\boldsymbol{\zeta}+\mu\boldsymbol{\eta}}{2}\right\|^2\right) + 2\rho\|\mathbf{v}\|_2 \tag{3.13}$$

This is separable over $\mathbf{u}$ and $\mathbf{v}$, and can be analytically solved. We get

$$\mathbf{u} = \frac{\mathbf{p}+\mathbf{q}-\mu\boldsymbol{\zeta}-\mu\boldsymbol{\eta}}{2},$$

$$\mathbf{v} = \mathcal{T}_{\mu\rho}\left(\frac{\mathbf{p}-\mathbf{q}-\mu\boldsymbol{\zeta}+\mu\boldsymbol{\eta}}{2}\right)$$

The result is obtained by setting $\mathbf{x} = \mathbf{u}+\mathbf{v}, \mathbf{y} = \mathbf{u}-\mathbf{v}$. □

Efficient Computation: While the objective in (3.2) may appear complex as it involves n^2 terms, the associated algorithm is stochastic and incremental, thus only involving one term in (3.2) for each iteration, thus greatly reducing the complexity as a result. The simplification of the algorithm is also due to the proximal update detailed in Theorem 5 (and subsequent projection) used in each iteration update of a pair of rows or columns. We further note that an early stopping typical of stochastic schemes is likely, making a full-run to convergence unnecessary (see Section 4.4 in [Bottou et al., 2018]), and in practice avoiding the impact of the n^2 terms on the performance. When the underlying data satisfies the structure of the stochastic block model, the problem size is essentially $B^2 \ll n^2$, as the number of required iterations is determined by an adequate sampling of all blocks.

Just-in-Time Update: In our problem of interest in equation 2.2, the number of variables quadratically grows with the problem size. For such problems, incremental algorithms may become infeasible in large-scale. Note that each iteration of our algorithm includes proximal and projection operators, that update only a small group of variables. This allows us to apply the Just-in-Time approach in [Schmidt et al., 2017] to resolve the problem with the number of variables. In our problem, each term $\phi_n(x)$ and constraint S_m only involves a small subset $x_{I_n} := (x_i, i \in I_n)$ of the variables, where $I_n \subseteq [D]$. Hence, the projection and proximal operators alter only a small subset of variables, dramatically reducing the amount of computation. We exploit this to give an algorithm that has much cheaper per-iteration cost. Note that the vanilla algorithm explained in equation 3.4 and equation 3.5 still operates on the full set of variables as the memory vectors become non-sparse by the updating rule in equation 3.5. We resolve this issue by following the Just-in-Time approach in [Schmidt et al., 2017] and modifying equation 3.5 to

$$\mathbf{a}_t = \rho\frac{\mathbf{x}_t - \mathbf{x}_{t+1}}{\mu} - \alpha\left(\sum_n \mathbf{g}_n + \sum_m \mathbf{h}_m\right)_{I_t} \tag{3.14}$$

where I_t denotes the set of variables involved in the t^{th} iteration and we define $(\mathbf{y})_I$ for a vector $\mathbf{y} = (y_1, y_2, \ldots, y_d)$ as a vector $\mathbf{y}' = (y'_1, y'_2, \ldots, y'_d)$ such that

$$y'_i = \begin{cases} \frac{Ky_i}{K_i} & i \in I \\ 0 & i \notin I \end{cases}$$

where $K = M + N$ and K_i is the number of objective terms ϕ_n and constraint sets S_m including the i^{th} variable x_i.

4 Class Based Regularization: Guarantees

In the appendix (section 7), we show that our SON regularizer is able to *provably* compute transport plans that respect the class structure in the manner, explained in section 2. Our approach is to analyse it under a setting such as the well known *stochastic block model* (SBM) [Holland et al., 1983, Snijders and Nowicki, 1997], also known as the *planted partition model* [Condon and Karp, 2001] which has been used widely as a canonical generative model for the data with clear class structure. In this model, we already have a latent ground truth for the class structure which the algorithm is supposed to recover.

In the supervised version of the domain adaptation problem, the class structure is given explicitly in the source via the labels, but not in the target domain. In the unsupervised version, the class structure is unknown in both domains. In both cases, it is reasonable to assume that a latent (hidden) class structure exists. We show that our algorithm can discover this hidden class structure in both domains (unsupervised) or in the target (supervised) and computes a transport plan that respects the class structure in the two domains.

5 Experiments

We now investigate different OT domain adaptation models on several datasets. We start by real-world datasets, and in the appendix (section 11), we illustrate the value of several other properties of our method, such as early stopping, class diversity and unsupervised domain adaptation. We compare our method (OT-SON) with the other regularized optimal transport-based methods OT-l1l2, OT-lpl1 and OT-Sinkhorn, as developed and used in [Courty et al., 2017, Cuturi, 2013, Perrot et al., 2016].

5.1 Real-world experiments

5.1.1 MNIST and USPS

In these experiments, we compare the different models on the real-world images of digits. For this, we consider the MNIST data as the source and the USPS data as the target. To further increase the difficulty of the problem, we use

all 10 classes of the source (MNIST) data, and we discard some of the classes of the target (USPS) data. In our experiments, each object (image) is represented by 256 features. By discarding the different subsets from the USPS data, we consider several pairs of source and target datasets. i) real1: the USPS classes are $1, 2, 3, 5, 6, 7, 8$, ii) real2: the USPS classes are $0, 2, 4, 5, 6, 7, 9$, iii) real3: the USPS classes are $0, 1, 3, 5, 7, 9$, and iv) real4: the USPS classes are: $0, 1, 3, 4, 6, 8, 9$. We note that these settings where the number of classes is different between the domains are the typical cases in practice. Therefore, our class-specific domain adaption approach is more suitable and robust to class imbalance, as it avoids splitting a class in one domain among multiple classes in another domain.

The transformed source samples are used to train a 1-nearest neighbor classifier. We then use this (parameter-free) classifier to estimate the class labels in the target data and then compute the respective accuracy. Table 1 shows the accuracy results for different OT-based models for different values of the regularization parameters λ_1 and λ_2 (i.e., $\lambda_1, \lambda_2 \in \{10^{-5}, ..., 10^{3}\}$)[1]. Specifically, for each OT method we use different values of regularization parameters and we report the best accuracy achieved by that method. We observe, i) OT-SON yields the highest accuracy scores, and ii) it is significantly more robust to variation of the regularization parameters, in comparison to the other methods. Moreover, the other methods are prone to yielding numerical errors for small regularizations.

model	real1	real2	real3	real4
OT-SON	**0.550**	**0.564**	**0.608**	**0.628**
OT-l1l2	0.421	0.507	0.500	0.621
OT-lpl1	0.457	0.521	0.516	0.592
OT-Sinkhorn	0.414	0.521	0.508	0.621

Table 1: Accuracy scores on MNIST and USPS.

5.1.2 Caltech Office

A commonly used dataset for domain adaptation is an object recognition dataset known as *Caltech Office* [Saenko et al., 2010, Griffin et al., 2006] which consists of four different domains: A (Amazon, 958 samples) , W (Webcam, 295 samples), C (Caltech, 1123 samples), and D (DSLR, 157 samples). These domains have 10 classes of objects in common that are represented with two sets of features: SURF (800 features) and DeCAF (4096 features). Similar to the setting in [Courty et al., 2017], we use all possible pairs of the four domains as source and target with SURF features, and we remove the classes 3, 5, and 7 from the target domain. Similar to the previous study, we assume imbalanced source and target classes in order to make the task more realistic. Table 2 compares the classification accuracies achieved by our method with the scores obtained by the three other methods. We calculate the accuracy similar to Section 5.1.1. In these experiments we use class information in the source domain together with Gaussian kernels. Specifically, we set $R_{l,k} = 0$ if y_l^s and y_k^s are not in the same classes and otherwise we set $R_{l,k} = \lambda \exp(-\|y_l^s - y_k^s\|^2)$ for different values of λ. We observe that our method yields the best results in seven cases. In other cases, our method is still competitive compared to the alternatives. In addition, we conclude that the different methods may perform differently on different datasets. Even though the setting of imbalanced classes is more important in practice and we have focused more in this paper, for the sake of completeness, we also compare our method with the three other methods in a setting where the classes are balanced. We use the same number of classes in the source (W) and target (C) domains. The accuracy results for the different methods are respectively: i) OT-SON: 0.260, ii) OT-l1l2: 0.244, iii) OT-lpl1: 0.247, and iv) OT-Sinkhorn: 0.243. We observe that even in this setting, our method yields the highest accuracy.

S→T	OT-SON	OT-l1l2	OT-lpl1	OT-Sinkhorn
A→C	0.3552	0.3229	**0.3565**	0.3018
A→D	0.3274	0.3097	**0.3539**	0.3008
A→W	**0.3144**	0.2577	0.3092	0.2422
C→A	**0.4601**	0.3714	0.4120	0.3548
C→D	0.3982	0.3274	**0.4424**	0.3362
C→W	**0.3556**	0.2474	0.3144	0.2474
D→A	**0.3248**	0.2812	**0.3248**	0.2812
D→C	0.2720	0.2658	**0.2956**	0.2621
D→W	0.7216	**0.7628**	0.5876	0.7525
W→A	**0.2661**	0.2225	0.2616	0.2210
W→C	**0.2149**	0.1962	0.2124	0.2012
W→D	**0.8230**	0.7964	0.6637	0.8141

Table 2: Accuracy scores on Caltech Office (S: source, T: target).

6 Conclusion

We developed a regularized optimal transport algorithm which produces sparse maps which are suitable for problems with class specifications and geometric kernels. We provided theoretical guarantees for the sparsity of the resulting transform, and developed constrained incremental algorithms which are generally suitable for non-smooth problems and enjoy theoretical convergence guarantees. Our experimental studies have substantiated the effectiveness of our proposed approach in different illustrative settings and datasets.

[1] For OT-l1l2 and OT-lpl1, λ_1 is the entropic regularization parameter and λ_2 is class regularization parameter.

References

Alaya, M. Z., Berar, M., Gasso, G., and Rakotomamonjy, A. (2019). Screening sinkhorn algorithm for regularized optimal transport. In Wallach, H., Larochelle, H., Beygelzimer, A., d Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32, pages 12169–12179. Curran Associates, Inc.

Altschuler, J., Weed, J., and Rigollet, P. (2017). Near-linear time approximation algorithms for optimal transport via sinkhorn iteration. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 1964–1974.

Alvarez-Melis, D., Jaakkola, T., and Jegelka, S. (2018). Structured optimal transport. In Storkey, A. and Perez-Cruz, F., editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 1771–1780. PMLR.

Balaji, Y., Chellappa, R., and Feizi, S. (2020). Robust optimal transport with applications in generative modeling and domain adaptation. *Advances in Neural Information Processing Systems (NeurIPS) 2020*, abs/2010.05862.

Bertsekas, D. (2011). Incremental proximal methods for large scale convex optimization. *Math. Program.*, 129(163).

Blondel, M., Seguy, V., and Rolet, A. (2018). Smooth and sparse optimal transport. In Storkey, A. and Perez-Cruz, F., editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 880–889. PMLR.

Bottou, L., Curtis, F. E., and Nocedal, J. (2018). Optimization methods for large-scale machine learning. *Siam Reviews*, 60(2):223–311.

Cao, Z., Ma, L., Long, M., and Wang, J. (2018). Partial adversarial domain adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Chang, H. and Yeung, D. (2008). Robust path-based spectral clustering. *Pattern Recognition*, 41(1):191–203.

Chizat, L., Peyré, G., Schmitzer, B., and Vialard, F. (2018). Scaling algorithms for unbalanced optimal transport problems. *Math. Comput.*, 87(314):2563–2609.

Combettes, P. L. and Pesquet, J.-C. (2011). Proximal splitting methods in signal processing. In Bauschke, H. H., Burachik, R. S., Combettes, P. L., Elser, V., Luke, D. R., and Wolkowicz, H., editors, *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer, New York.

Condat, L. (2016). Fast projection onto the simplex and the l1 ball. *Math. Program.*, 158(1-2):575–585.

Condon, A. and Karp, R. (2001). Algorithms for graph partitioning on the planted partition model. *Random Structures and Algorithms*, 18(2):116–140.

Courty, N., Flamary, R., Tuia, D., and Rakotomamonjy, A. (2017). Optimal transport for domain adaptation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(9):1853–1865.

Cuturi, M. (2013). Sinkhorn distances:lightspeed computation of optimal transport. In *Adv. in Neural Information Processing Systems*, pages 2292–2300.

Defazio, A. (2016). A simple practical accelerated method for finite sums. *Advances in Neural Information Processing Systems 29 (NIPS 2016)*.

Dessein, A., Papadakis, N., and Rouas, J.-L. (2018). Regularized optimal transport and the rot mover's distance. *J. Mach. Learn. Res.*, 19(1):590–642.

Duchi, J. C., Shalev-Shwartz, S., Singer, Y., and Chandra, T. (2008). Efficient projections onto the l_1-ball for learning in high dimensions. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, pages 272–279.

Genevay, A., Cuturi, M., Peyre, G., and Bach, F. (2016). Stochastic optimization for large-scale optimal transport. In *Adv. in Neural Information Processing Systems*.

Griffin, G., Holub, A., and Perona, P. (2006). Caltech256 image dataset.

Guo, W., Ho, N., and Jordan, M. (2020a). Fast algorithms for computational optimal transport and wasserstein barycenter. In Chiappa, S. and Calandra, R., editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 2088–2097. PMLR.

Guo, W., Ho, N., and Jordan, M. (2020b). Fast algorithms for computational optimal transport and wasserstein barycenter. In Chiappa, S. and Calandra, R., editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 2088–2097. PMLR.

Hocking, T., Vert, J., Bach, F. R., and Joulin, A. (2011). Clusterpath: an algorithm for clustering using convex fusion penalties. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 745–752.

Holland, P., Laskey, K., and Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137.

Kolouri, S., Park, S. R., Thorpe, M., Slepcev, D., and Rohde, G. K. (2017). Optimal mass transport: Signal processing and machine-learning applications. *IEEE Signal Process. Mag.*, 34(4):43–59.

Lindsten, F., Ohlsson, H., and Ljung, L. (2011a). Clustering using sum-of-norms regularization: With application to particle filter output computation. In *IEEE Statistical Signal Pro- cessing Workshop (SSP)*.

Lindsten, F., Ohlsson, H., and Ljung, L. (2011b). Clustering using sum-of-norms regularization: With application to particle filter output computation.

Monge, G. (1781). Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences de Paris*.

Papadakis, N., Peyré, G., and Oudet, E. (2014). Optimal transport with proximal splitting. *SIAM Journal on Imaging Sciences*, 7(1):212–238.

Parikh, N. and Boyd, S. (2016). Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):123–231.

Patrascu, A. and Necoara, I. (2018). Nonasymptotic convergence of stochastic proximal point methods for constrained convex optimization. *Journal of Machine Learning Research*, 18(198):1–42.

Perrot, M., Courty, N., Flamary, R., and Habrard, A. (2016). Mapping estimation for discrete optimal transport. In *Advances in Neural Information Processing Systems*, pages 4197–4205.

Peyré, G. and Cuturi, M. (2018). *Computational Optimal Transport*. Athena Scientific Belmont.

Saenko, K., Kulis, B., Fritz, M., and Darrell, T. (2010). Adapting visual category models to new domains. In Daniilidis, K., Maragos, P., and Paragios, N., editors, *Computer Vision – ECCV 2010*, pages 213–226, Berlin, Heidelberg. Springer Berlin Heidelberg.

Santambrogio, F. (2015). *Optimal transport for applied mathematicians*. Birkäuser, NY.

Schmidt, M., Le Roux, N., and Bach, F. (2017). Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1-2):83–112.

Seguy, V., Damodaran, B. B., Flamary, R., Courty, N., Rolet, A., and Blondel, M. (2018). Large scale optimal transport and mapping estimation. In *International Conference on Learning Representations*.

Snijders, T. and Nowicki, K. (1997). Estimation and prediction for stochas- tic blockmodels for graphs with latent block structure. *Journal of Classification*, 14(1):75–100.

Solomon, J. (2018). Optimal transport on discrete domains. *CoRR*, abs/1801.07745.

Villani, C. (2008). *Optimal transport: old and new*, volume 338. Springer Science & Business Media.

Wang, M. and Bertsekas, D. (2016). Stochastic first-order methods with random constraint projection. *SIAM J. Optimization*, 26(1):681–717.