Greedy stochastic algorithms for entropy-regularized optimal transport problems

Brahim Khalil Abid
brahim-khalil.abid@polytechnique.edu
École polytechnique[1]

Robert M. Gower
robert.gower@telecom-paristech.fr
LTCI, Télécom-Paristech, Université Paris-Saclay[2]

Abstract

Optimal transport (OT) distances are finding evermore applications in machine learning and computer vision, but their wide spread use in larger-scale problems is impeded by their high computational cost. In this work we develop a family of fast and practical stochastic algorithms for solving the optimal transport problem with an entropic penalization. This work extends the recently developed Greenkhorn algorithm, in the sense that, the Greenkhorn algorithm is a limiting case of this family. We also provide a simple and general convergence theorem for all algorithms in the class, with rates that match the best known rates of Greenkorn and the Sinkhorn algorithm, and conclude with numerical experiments that show under what regime of penalization the new stochastic methods are faster than the aforementioned methods.

1 Introduction

Probability distributions are the backbone of machine learning and statistics: we use them to represent a variety of objects in learning tasks, ranging from statistical models to data representations. Comparing different distributions is often done using information divergences such as Kullback-Leiber divergence, yet this discards much of structural and geometric information present in the distribution. Developing a practical measure that captures the geometry of the probability distribution is a problem to which the optimal transport (OT) distance offers an attractive solution.

First formulated by Monge 1781 then revisited by Kantorovich 1942, OT distances have the inherent particularity of capturing the geometrical properties of the probability measures. However, they have a drawback: computing an OT distance has a typical cost of the order $O(n^3 \log n)$ for histograms of n points (Pele and Werman 2009). This prevents the application of OT distances in large-scale machine learning problems.

The idea of entropy penalization, proposed by Cuturi 2013, represents a key milestone in this field. The benefits of such a regularization scheme are multiple: the regularized problem has a unique solution, greater computational stability, and can be solved efficiently using the Sinkhorn algorithm. This new family of distances has been used in a wide range of applications, such as image classification (Cuturi 2013), unsupervised learning using Restricted Boltzmann Machines (Montavon, Müller, and Cuturi 2016), learning with a Wasserstein Loss (Frogner et al. 2015), domain adaptation (Courty, Flamary, and Tuia 2014), computer graphics (Solomon et al. 2015), and neuroimaging (Gramfort, Peyré, and Cuturi 2015). The growing interest in applications for the Sinkhorn distances has sparked the development of new and efficient algorithms for its calculation, such as stochastic gradient based algorithms by Genevay et al. 2016, and fast methods to compute Wasserstein barycenters (Cuturi and Doucet 2014). To this end, Altschuler, Weed, and Rigollet 2017 have developed the Greenkhorn algorithm, a greedy variant of the Sinkhorn algorithm that selects columns and rows to be updated that most violate the constraints. The authors present both promising numerical results, besting the Sinkhorn algorithm, and an insightful theoretical complexity that is linear in n.

Our contribution: We expand on the idea of greedy

[1]Work done while at Inria, SIERRA team and ENSAE, CREST

[2]Work done while at Inria, SIERRA team.

Proceedings of the 21^{st} International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. PMLR: Volume 84. Copyright 2018 by the author(s).

column and row selection by proposing a family of algorithms that assign a probability of updating each row and column. Moreover, our family allows for any sampling so long as the probabilities are proportional to the violation of each column or row with respect to the transport polytope. We call our algorithm Stochastic Sinkhorn. We explain the idea behind this family of methods, show how Greenkhorn is a limiting case, and propose several other instances of the algorithm. We develop an all encompassing convergence theorem that recovers the best known $O(n/\epsilon^2)$ iteration complexity for the Greenkhorn algorithm. Finally, we exhibit some numerical experiments that explicit the relevance of Stochastic Sinkhorn in a particular regime of penalization, along with a discussion around the computational properties of the algorithms.

1.1 The Optimal Transport problem

The discrete OT problem can be seen as a problem of optimal resource allocation given by a linear program

$$\begin{aligned} T^* \quad \in \quad & \arg\min_{T\in\mathbb{R}_+^{n\times n}} \langle T, C\rangle\,, \\ & \text{subject to} \quad T\mathbf{1} = r,\ T^\top\mathbf{1} = c, \end{aligned} \tag{1}$$

where $r, c \in \Delta_n \stackrel{\text{def}}{=} \{x \in \mathbb{R}^n \mid \sum_{i=1}^n x_i = 1\}$ are respectively the initial and target distributions, $C \in \mathbb{R}_+^{n\times n}$ the transport cost matrix and $\mathbf{1}$ is a vector of all ones of an appropriate dimension. Matrices $T \in \mathbb{R}_+^{n\times n}$ that satisfy the transport constraints in (1) represent valid transportation maps between r and c, where T_{ij} will represent the mass transported from r_i to c_j. The matrix T^* is a transportation map that minimizes the transportation cost, the computed minimum $\langle T^*, C\rangle$ is the optimal transport value and it defines a distance between r and c (Villani 2008). The transportation map T^* can be computed using the network simplex or interior point methods (Pele and Werman 2009), but the computational cost is in both cases $O(n^3\log(n))$. It is this cubic cost in the dimension that makes this notion of distance infeasible in high-dimensional settings, such as in computer vision or high dimensional inference.

1.2 Entropic regularization and the Sinkhorn algorithm

An interesting approach to alleviate the computational burden was proposed by Cuturi 2013 through the introduction of an entropic regularization as follows

$$\begin{aligned} T^*_\lambda \quad = \quad & \arg\min_{T\in\mathbb{R}_+^{n\times n}} \langle T, C\rangle - \frac{1}{\lambda}E(T), \\ & \text{subject to} \quad T\mathbf{1} = r,\ T^\top\mathbf{1} = c, \end{aligned} \tag{2}$$

where the entropy is $E(T) = \sum_{i,j=1}^n -T_{ij}\log(T_{ij})$. Due to the strong convexity introduced by the entropic regularization, the problem (2) now has a unique solution. What is more, using duality theory (2) has a smooth and unconstrained dual formulation. Leveraging on the dual Cuturi showed that (2) can be equivalently re-written as the following matrix scaling problem: find $u, v \in \mathbb{R}_+^n$ such that

$$D(u)AD(v)\mathbf{1} = r \quad \text{and} \quad D(v)A^\top D(u)\mathbf{1} = c, \tag{3}$$

where $A = e^{-\lambda C}$ with the exponential taken elementwise and $D(u)$ denotes a diagonal matrix with the elements of u on the diagonal. With the (u, v) solution to (3), the solution to (2) is simply given by $T^*_\lambda = D(u)AD(v)$. This matrix scaling problem can now be efficiently solved using the celebrated Sinkhorn algorithm, as proposed by Cuturi 2013.

The Sinkhorn algorithm is a fixed point iteration algorithm for solving (3) which alternately scales the row and column sums to match the desired marginals

$$\begin{aligned} u^{k+1} \quad &= \quad r./(Av^k), \\ .v^{k+1} \quad &= \quad c./(A^\top u^k), \end{aligned} \tag{4}$$

where we have used $x./y$ to denote elementwise division of vectors[1]. On top of being a simple and fast algorithm, the Sinkhorn algorithm is also GPU-friendly since its highest cost is a matrix vector product which can be parallelized. The resulting distance $\langle T^*_\lambda, C\rangle$ defined by (2) has been dubbed the Sinkhorn distance.

Notation: For the sake of brevity, we use $r(T) = T\mathbf{1}$ and $c(T) = T^\top\mathbf{1}$ to denote the *row sum* and *column sum* vectors of T, respectively. Let $U_{r,c}$ be the transport polytope defined by

$$U_{r,c} \stackrel{\text{def}}{=} \{T \in \mathbb{R}_+^{n\times n} \,|\, r(T) = r,\ c(T) = c\}.$$

Since we need to solve (3), in order to discuss convergence results, we need to define a distance that measures how far are the scaled iterates from the transport polytope $U_{r,c}$. We will use in all the following work the ℓ_1 distance

$$dist(A, U_{r,c}) \stackrel{\text{def}}{=} \|r(A) - r\|_1 + \|c(A) - c\|_1 \tag{5}$$

which, as argued by Altschuler, Weed, and Rigollet, is much more suitable to compare probability distributions than the ℓ_2 distance. A simple example to see this: taking $p = (\frac{1}{n}, ...\frac{1}{n}, 0....0) \in \Delta_{2n}$ and $q = (0, 0....0, \frac{1}{n}, ...\frac{1}{n}) \in \Delta_{2n}$ as two probability distributions with disjoint supports, we see that $\|p-q\|_1 = 2$, while $\|p-q\|_2 = \frac{1}{\sqrt{n}}$ and thus decreases as n increases, despite being clearly distinct distributions for all n.

[1] In other words $x./y = D(y)^{-1}x$

1.3 Greedy Sinkhorn: Greenkhorn algorithm

Greenkhorn is a greedy version of Sinkhorn proposed by Altschuler, Weed, and Rigollet 2017 where at each iteration only one coordinate of u or v is updated in (4), picking each time the one with highest violation with respect to the corresponding marginal. These violations are computed with the following function

$$\rho(a,b) = b - a + a\log(\frac{a}{b}), \quad \text{for } a,b \in \mathbb{R}_+ \quad (6)$$

$$d_\rho(u,v) = \sum_{i=1}^{n} \rho(u_i, v_i), \quad \text{for } u,v \in \mathbb{R}^n_+. \quad (7)$$

For vectors in the simplex $u, v \in \Delta_n$, we have that $d_\rho(u,v)$ coincides with the Kullback-Leiber divergence. For this reason, d_ρ is known as the generalized Kullback-Leiber divergence. It is not a distance because it is not symmetric, but it verifies $d_\rho \geq 0$ and $d_\rho(u,v) = 0 \Leftrightarrow u = v$. Therefore, if u, v are two vectors of positive entries, $d_\rho(u,v)$ will return some measurement on how far they are from each other. Let $\rho^r(M)$ (resp. $\rho^c(M)$) be the vector of the row sum violations (resp. column sum violations) of a given matrix $M \in \mathbb{R}^{n\times n}_+$ with respect to r (resp. c), where the violations are computed using ρ, that is

$$\rho^r(M) = (\rho(r_i, r_i(M)))_{i=1..n} \in \mathbb{R}^n_+,$$
$$\rho^c(M) = (\rho(c_i, c_i(M)))_{i=1..n} \in \mathbb{R}^n_+.$$

We will refer to the concatenation of these two vectors as the *marginal violations* denoted by

$$\rho(M) = (\rho^r(M), \rho^c(M))_{i=1..n} \in \mathbb{R}^{2n}_+. \quad (8)$$

The marginal violations vector $\rho(M)$ measures how far the matrix M is from the transport polytope $U_{r,c}$ in the sense that $M \in U_{r,c}$ if and only if all entries of $\rho(M)$ are equal to zero.

The Greenkhorn algorithm uses $\rho(M)$ to guide the choice of which row or column should be updated. As the name indicates, the algorithm chooses the row or column index greedily, that is the index of maximal value in $\rho(M)$, see Algorithm 1. This greedy variation of Sinkhorn is expected to perform better in practice, mainly because it does not update rows or columns that already match the correct marginal sum value.

Altschuler, Weed, and Rigollet proved that, to reach an $\epsilon > 0$ approximate solution, the Greenkhorn algorithm and the Sinkhorn algorithm converge in at most $28n\epsilon^{-2}\log(\frac{s}{l})$ and $28\epsilon^{-2}\log(\frac{s}{l})$ iterations, respectively, where $s = \|A\|_1$ is the total mass and l the smallest entry of the matrix A. Altschuler, Weed, and Rigollet also claimed that the Greenkhorn algorithm can be implemented in such a way that the iteration cost is linear in n, consequently the overall complexity of either the Sinkhorn algorithm or Greenkhorn is quadratic in n, which is stark contrast to the cubic dependency of the interior point type methods (Pele and Werman 2009). Since the authors omitted the details on how such a linear iteration complexity can be achieved, we have given the details in Section 4.1. The ϵ^{-2} dependency of Greenkhorn and Sinkhorn is also in contrast with logarithmic dependency on ϵ in interior point based methods. Thus Greenkhorn and Sinkhorn are well suited for the large dimensional setting where we can tolerate an approximate solution. This is typically the case in the problems that we are interested in here, such as problems that arise in large dimensional machine learning.

Algorithm 1: Greenkhorn

Data: $A \in \mathbb{R}^{n\times n}_+$, $r, c \in \mathbb{R}^n_+$, $\epsilon > 0$
initialization: u,$v = \mathbf{1}$
while *dist*$(D(u)AD(v)$,$U_{r,c}) \geq \epsilon$ **do**
  $I = \arg\max_{i=1..2n} \rho(D(u)AD(v))$
  if $I \leq n$ *(corresponds a row update)* **then**
    $u_I = r_I./(Av)_I$
  else
    $v_{I-n} = c_{I-n}./(A^\top u)_{I-n}$
  Result: $u, v \in \mathbb{R}^n_+$ such that $D(u)AD(v) \in U_{r,c}$

2 Greedy Stochastic Sinkhorn

While the greedy strategy in the Greenkhorn algorithm is, in some sense, optimal for one step, it may not be the best strategy over a number of iterations. Here we introduce a more flexible, and less aggressive updating strategy.

At each iteration of the *Stochastic Sinkhorn* algorithm, instead of picking the column or row with the highest violation, as is done in the Greenkhorn algorithm, we will assign to each row and column a probability of being updated. Because we want the columns and rows with highest violation to be updated more frequently, we assign a higher probability to columns and rows with a higher violation. We do this using an *increasing probability function*.

Definition 1 *We say that Ψ is a increasing probability function if*

$$\forall h \in \mathbb{R}^{2n}_+ \quad \Psi(h) = \left(\frac{g(h_k)}{\sum_{i=1}^{2n} g(h_i)}\right)_{k=1..2n} \in \Delta_{2n} \quad (9)$$

where $g : \mathbb{R}_+ \to \mathbb{R}_+$ is an increasing positive function.

Several examples of an increasing probability function

are given as follow

$$\Psi(h) = \left(\frac{1}{2n}\right)_{i=1,\dots,2n}, \quad (10)$$

$$\Psi(h) = \left(\frac{h_i^\alpha}{\sum_{j=1..2n} h_j^\alpha}\right)_{i=1..2n}, \quad (11)$$

$$\Psi(h) = \left(\frac{e^{(h_i/T)}}{\sum_{j=1..2n} e^{(h_j/T)}}\right)_{i=1..2n}, \quad (12)$$

where $T, \alpha > 0$ are parameters. If ρ is our current vector of violations, then $\Psi(\rho) = p \in \Delta_{2n}$ defines a probability distribution. Furthermore, since Ψ is built on top of an increasing function, a larger violation ρ_i will result in a larger probability p_i. See Algorithm 2 for the pseudocode of this family of stochastic algorithms. In the next section we prove that Algorithm 2 converges for any increasing probability function. This is particularly interesting when we consider that the Greenkhorn algorithm is a limiting case of Stochastic Sinkhorn. Indeed, the selection criteria of the Greenhkorn algorithm corresponds to taking the limit over $\alpha \to \infty$ of the probability function (11).

Algorithm 2: Stochastic Sinkhorn

Data: $A \in \mathbb{R}_+^{n\times n}$, $r, c \in \mathbb{R}_+^n$, Ψ, ϵ
Result: $u, v \in R_+^n$ such that $D(u)AD(v) \in U_{r,c}$
initialization: u,$v = \mathbf{1}$
while *dist*$(D(u)AD(v)$,$U_{r,c}) \geq \epsilon$ **do**
  $p = \Psi(\rho(D(u)AD(v))) \in \Delta_{2n}$
  Sample index I with
    $P(I = i) = p_i, \quad \forall i \in \{1, 2, \dots, 2n\}$
  if $I \leq n$ *(corresponds a row update)* **then**
    $u_I = r_I./(Av)_I$
  else
    $v_{I-n} = c_{I-n}./(A^\top u)_{I-n}$

3 Convergence analysis

We now present our main convergence theorem, discuss its consequences and proof.

Theorem 2 *Consider the sequence of matrices $A^k \overset{def}{=} D(u^k)AD(v^k)$ produced by Algorithm 2 with an increasing probability function Ψ as defined in* (9)*. Then for a given $\epsilon > 0$, we have that*

$$\exists k \in \mathbb{N}, \quad k \leq \frac{28n}{\epsilon^2} \log\left(\frac{s}{\ell}\right), \quad (13)$$

such that $\mathbf{E}\left[dist(A^k, U_{r,c})\right] \leq \epsilon$.

We make several interesting remarks on the consequence of this theorem.

1. Since $dist(A^k, U_{r,c})$ is a positive random variable, by Markov's inequality we have that the convergence in expectation given in Theorem 2 also proves that $dist(A^k, U_{r,c})$ converges in probability to zero. The variance also converges to zero at a $O(n/\epsilon)$ rate, as proven in the appendix.
2. The convergence rate given in Theorem 2 is exactly the same rate as given by Altschuler, Weed, and Rigollet 2017 for the Greenkorn algorithm.
3. Remarkably the rate of convergence does not depend on the choice of probability function Ψ. Thus, in theory, a uniform selection of the coordinates gives the same asymptotic convergence as the Greenkhorn selection criteria.

Before moving onto the proof, we need several auxiliary lemmas.

3.1 Useful lemmas

Our analysis is based on the dual objective of (1) given by

$$f(x, y) = \sum_{i,j=1}^{n} A_{ij} e^{x_i + y_j} - \langle r, x\rangle - \langle c, y\rangle . \quad (14)$$

Let $X = D(e^x)$ and $Y = D(e^y)$. By writing out the first order optimality conditions of $f(x, y)$ we arrive at

$$r(XAY) = r \quad \text{and} \quad c(XAY) = c . \quad (15)$$

That is, the row sum and column sum of XAY is r and c, respectively. By denoting $u = e^x$ and $v = e^y$ the given scaling vectors, we see that (15) is the matrix scaling problem (3). Throughout this section we use (u^k, v^k) to denote the (u, v) vectors of Algorithm 2 after completing the kth iteration. We also denote $(x^k, y^k) = (\log(u^k), \log(v^k))$.

The proof of Theorem 2 is based on the four next lemmas. Our first lemma is an extension to the stochastic setting of a lemma by Altschuler, Weed, and Rigollet. It links the expectation of the dual objective value to a type of condition number of the matrix A.

Lemma 3 *Let $((u^k, v^k))_{k\in\boldsymbol{N}}$ and the associated $((x^k, y^k))_{k\in\boldsymbol{N}}$ be a sequence of scaling vectors produced by the Stochastic Sinkhorn Algorithm 2. Then the following inequalities hold*

$$\begin{aligned}\mathbf{E}\left[f(x^k, y^k)\right] - \min_{x,y\in R} f(x, y) &\leq f(0, 0) - \min_{x,y\in R} f(x, y)\\ &\leq \log(\frac{s}{l}),\end{aligned}$$

where $l = \min_{i,j} |A_{ij}|$ and $s = \|A\|_1$. As a direct consequence, we also have

$$f(0, 0) - \mathbf{E}\left[f(x^k, y^k)\right] \leq \log(\frac{s}{l}) .$$

The next lemma is a useful inequality on ordered series of real number.

Lemma 4 (Chebyshev inequality) *Let $a_1 \leq a_2 \leq \ldots \leq a_n \in \mathbb{R}$ and $b_1 \leq b_2 \leq \ldots \leq b_n$ be two ordered sequences. It follows that*

$$\frac{1}{n}\sum_{i=1}^{n} a_i b_i \geq \left(\frac{1}{n}\sum_{j=1}^{n} a_j\right)\left(\frac{1}{n}\sum_{j=1}^{n} b_j\right). \tag{16}$$

Next we have a lemma that is a generalization of the Pinsker inequality.

Lemma 5 *The following generalized Pinsker inequality holds for $v, u \in R^n_+$*

$$\|u - v\|_1 \leq \sqrt{7\|u\|_1 d_\rho(u,v)}\,, \tag{17}$$

where $d_\rho(u,v)$ is defined as in (7).

The final lemma was presented by Altschuler, Weed, and Rigollet 2017, and it links the evolution of dual objective (14) and the marginal violations.

Lemma 6 *Let $((u^k, v^k))_{k\in\boldsymbol{N}}$ and the associated $((x^k, y^k))_{k\in\boldsymbol{N}}$ be a sequence of scaling vectors produced by the Stochastic Sinkhorn Algorithm 2. For a given k, if (u^{k+1}, v^{k+1}) was obtained by updating coordinate I of u^k then the following identity holds*

$$f(x^k, y^k) - f(x^{k+1}, y^{k+1}) = \rho(r_I, r_I(D(u^k)AD(v^k)))$$

and if they were obtained by updating coordinate J of v^k then

$$f(x^k, y^k) - f(x^{k+1}, y^{k+1}) = \rho(c_J, c_J(D(u^k)AD(v^k)))$$

Since $\rho \geq 0$ then the sequence of real numbers $(f(x^k, y^k))_{k\in\boldsymbol{N}}$ is decreasing.

3.2 Proof of Theorem 2

Proof: Let $D_k \overset{\text{def}}{=} \mathbf{E}\left[dist(A^k, U_{r,c})\right]$ and let $k^* \in \mathbb{N}$ be an integer such that $D_k > \epsilon$ for all $k < k^*$ (in other terms, an index such that the algorithm has not converged yet at the corresponding iteration).

Recall that $\rho(A^k)$ is the vector of all $2n$ marginal violations for the matrix A^k, as defined in (8). We will write its components as $\rho_i(A^k)$ for a given index i. Recall that $\Psi(\rho(A^k))$ is the vector of probabilities of picking each row and column, and similarly we will write its components $\Psi_i(\rho(A^k))$, which is then the probability of picking index i. We start the proof by showing that D_k^2 is upper bounded by the following conditional expectation

$$\mathbf{E}\left[\rho_I(A^k) \mid A^k\right] = \sum_{i=1}^{2n} \Psi_i(\rho(A^k))\rho_i(A^k),$$

where I is the index randomly sampled at iteration k. Let $k < k^*$ and since we assume that (9) holds for some function g, we have that

$$\begin{aligned}\mathbf{E}\left[\rho_I(A^k) \mid A^k\right] &= \sum_{i=1}^{2n} \frac{g(\rho_i(A^k))}{\sum_{j=1}^{2n} g(\rho_j(A^k))}\rho_i(A^k) \\ &\overset{(16)}{\geq} \frac{1}{n}\sum_{i=1}^{2n}\rho_i(A^k) \\ &\overset{(17)}{\geq} \frac{\left(\|r - r(A^k)\|_1 + \|c - c(A^k)\|_1\right)^2}{28n},\end{aligned} \tag{18}$$

where we applied Lemma 4 in the first inequality which relies on the monotonicity of g, and the generalized Pinsker inequality (17) in the second inequality with $a = (r, c)$, $b = (r(A^k), c(A^k))$ and used that $\|a\|_1 = \|r\|_1 + \|c\|_1 = 2$. Taking expectation in (18), using the law of total expectation and the fact that $\mathbf{E}\left[X^2\right] \geq \mathbf{E}\left[X\right]^2$ for any random variable X gives

$$\begin{aligned}\mathbf{E}\left[\rho_I(A^k)\right] &\overset{(18)}{\geq} \frac{\mathbf{E}\left[\|r - r(A^k)\|_1 + \|c - c(A^k)\|_1\right]^2}{28n} \\ &= \frac{1}{28n}D_k^2 \quad > \quad \frac{\epsilon^2}{28n}.\end{aligned} \tag{19}$$

To conclude, we now use Lemma 6 to re-write $\mathbf{E}\left[\rho_I(A^k) \mid A^k\right]$ as

$$\begin{aligned}&\mathbf{E}\left[f(x^k, y^k) - f(x^{k+1}, y^{k+1}) \mid x^k, y^k\right] \\ &\quad = \sum_{i=1}^{2n} \Psi_i(\rho(A^k))\rho_i(A^k) = \mathbf{E}\left[\rho_I(A^k) \mid A^k\right].\end{aligned} \tag{20}$$

Thus taking expectation in (20) gives

$$\begin{aligned}\mathbf{E}\left[f(x^k, y^k) - f(x^{k+1}, y^{k+1})\right] &= \mathbf{E}\left[\rho_I(A^k)\right] \\ &\overset{(19)}{>} \frac{\epsilon^2}{28n}.\end{aligned} \tag{21}$$

Summing over $k = 0, \ldots, k^* - 1$ in (21) and using telescopic cancellation we have that

$$f(x^0, y^0) - \mathbf{E}\left[f(x^{k^*}, y^{k^*})\right] > \frac{k^*\epsilon^2}{28n}. \tag{22}$$

Combining the above with

$$f(0,0) - \mathbf{E}\left[f(x^k, y^k)\right] \leq \log\left(\frac{s}{\ell}\right),$$

as proven in Lemma 3, we have that

$$\frac{28n}{\epsilon^2} log\left(\frac{s}{\ell}\right) > k^*. \tag{23}$$

This proves that for a given integer k^*

$$\forall k < k^*, \quad D_k > \epsilon, \quad \Rightarrow \quad \frac{28n}{\epsilon^2} log\left(\frac{s}{\ell}\right) > k^*. \tag{24}$$

The contrapositive of the above statement is given by

$$k^* \geq \frac{28n}{\epsilon^2} log\left(\frac{s}{\ell}\right) \Rightarrow \exists k < k^*, \quad D_k \leq \epsilon. \tag{25}$$

Choosing $k^* = \lceil \frac{28n}{\epsilon^2} log\left(\frac{s}{\ell}\right) \rceil$ concludes the proof. ∎

Corollary 7 *If we choose* Ψ *as either* (10), (11) *or* (12) *then the Stochastic Sinkhorn algorithm converges at a* $O(\frac{n}{\epsilon^2})$ *rate according to Theorem 2.*

Proof: The proof follows by observing that (10), (11) or (12) are increasing probability functions. That is, the functions $x \mapsto e^{x/T}$, $x \mapsto x^\alpha$ and $x \mapsto 1$ are positive increasing real-valued functions.

4 Numerical experiments

In this section we provide some empirical insights into the behaviour of the Stochastic Sinkhorn algorithm. We consider both real and synthetic datasets: MNIST digits and random histograms. The authors of both Cuturi 2013 and Altschuler, Weed, and Rigollet 2017 provided numerical experiments where Sinkhorn and Greenkhorn perform considerably better than other Optimal Transport algorithms. We will show how Stochastic Sinkhorn has a similar efficiency for monotonic probability functions (9). In particular, we will show that for (11) with $\alpha = 1$, Stochastic Sinkhorn outperforms Greenkhorn in the short-term for regimes of small penalization. Finally we will discuss the computational properties of the three algorithms, in particular some drawbacks of Greenkhorn and Stochastic Sinkhorn in comparaison with Sinkhorn, and give insights on how to bypass them. But first, we explicity show how Greenkhorn and Stochastic Sinkorn have linear iteration complexities.

4.1 Updating the marginal violation

The Greenkhorn Algorithm 1 and the Stochastic Sinkhorn Algorithm 2 must re-compute marginal violations

$$\rho(A^k) = \left[\rho(r_i, r_i(A^k))_{i=1..n},\, \rho(c_i, c_i(A^k))_{i=1..n}\right],$$

at each iteration. Calculating $\rho(A^k)$ from scratch at each iteration would cost $O(n^2)$, which would defeat the purpose of both algorithms of having a linear iteration complexity. Fortunately $\rho(A^k)$ can be updated on the fly with only $O(n)$ operations. To see this, suppose we have stored $c(A^k), r(A^k)$ and $\rho(A^k)$ and now we wish to calculate $\rho(A^{k+1})$. Suppose we sample an index I in Algorithm 2 such that $I \in \{1, \ldots, n\}$, consequently we update

$$u_I^{k+1} = r_I ./ (Av^k)_I \tag{26}$$

while $v^{k+1} = v^k$ and $u_i^{k+1} = u_i^k$ for $i \neq I$ remain unaltered. We can thus calculate the ith component of $r(A^{k+1})$ via

$$\begin{aligned} r_i(A^{k+1}) &= (D(u^{k+1})AD(v^k)\mathbf{1})_i = u_i^{k+1} A_{i:} v^k \\ &\overset{(26)}{=} \begin{cases} u_I^{k+1} A_{I:} v^k & \text{if } i = I, \\ r_i(A^k) & \text{if } i \neq I. \end{cases} \end{aligned}$$

The column sum vector can be updated using

$$\begin{aligned} c(A^{k+1}) &= D(v^k) \sum_{i=1}^{n} A_{i:} u_i^{k+1} \\ &= D(v^k) A_{I:} u_I^{k+1} - D(v^k) A_{i:} u_i^k + c(A^k). \end{aligned}$$

Thus both $r(A^{k+1})$ and $c(A^{k+1})$ can be updated using $O(n)$ operations. Since $\rho(r_i, r_i(A^{k+1})) = \rho(r_i, r_i(A^k))$ for $i \neq I$, only $n+1$ components of $\rho(A^{k+1})$ need to re-computed, which costs $O(n)$ operations. The $O(n)$ cost of the case where $I \in \{n+1, \ldots 2n\}$ can be deduced in an analogous way.

4.2 Experiments

We perform experiments on MNIST dataset. We take pairs of elements from the 28×28 pixels MNIST dataset, that we then vectorize into 1D arrays r and c (in the sense that, for example, the 2 by 2 matrix $((1,2),(3,4))$ becomes the vector $(1,2,3,4)$). The cost matrix C is then constructed so that C_{ij} equals the ℓ_1 distance between pixels i and j in the 28×28 grid. We then apply the Sinkhorn, Greenkhorn and Stochastic Sinkhorn algorithms to compute a diagonal scaling of $A = e^{-\lambda C}$. This process is then repeated 20 times, for each time we randomly sample a pair of images from the MNIST dataset. Finally, we report the average performance of the algorithms over these 20 experiments.

The choice of λ defines the penalization, and we highlight the fact that regimes of low penalization, corresponding to higher values of λ (2), are of particular interest since they change the least the solution of the original non-regularized problem (1). For this setting, Stochastic Sinkhorn with (11) for $\alpha = 1$ is clearly the best choice overall, see Figure 1.

We also compared in Figure 2 the Stochastic Sinkhorn for various choices of parameters. As expected, using the probability function (11) with $\alpha \to +\infty$ or (12) with $T \to 0$, the Stochastic Sinkhorn algorithm reduces to the Greenkhorn algorithm

Notice also that the standard deviation for Stochastic Sinkhorn (represented as errorbars) tends to 0, which is a very important property because of the stochastic nature of the algorithms. In fact, this means that not only the expectation of the distance tends to zero,

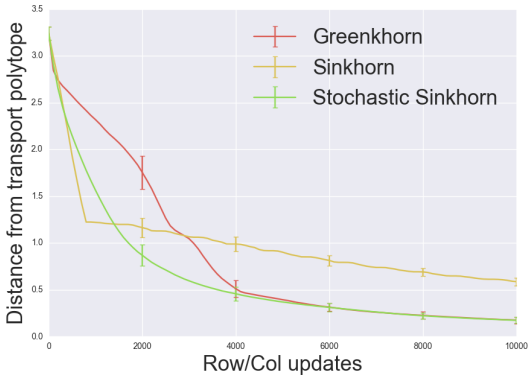

Figure 1: Evolution of distance from transport polytope for Sinkhorn, Greenkhorn and Stochastic Sinkhorn ((11) with $\alpha = 1$) in regimes of low penalization ($\lambda = 10$) on MNIST dataset. For the x-axis, one should read "number of row and column updates" in the sense that one iteration on the x-axis represents one update of a row or a column.

but also the variance, a fact which we prove in the appendix.

4.3 Discussion and block algorithms

While Stochastic Sinkhorn and Greenkhorn empirically perform better than Sinkhorn, they also have two computational drawbacks: firstly they are not parallelizable. In fact, each iteration of Sinkhorn (4) is a rescaling of u and v involving a matrix-vector product that can be parallelized. Greenkhorn and Stochastic Sinkhorn update only one element per iteration which does not involve a similar product that we can parallelize. Secondly, the greedy algorithms compute at each iteration the marginal violations, which, despite only costing $O(n)$, it does represent an additional computational cost per iteration.

One solution to these issues is to re-compute these marginal violations only once every d iterations: for example in Greenkhorn we compute marginal violations $\rho(A^k)$ and we update not only the index of highest value, but the d indexes of highest values. Something similar can be done in Stochastic Sinkhorn by sampling d indexes without replacement instead of just one. By doing so, on the one hand we reduce computation time for computing the marginal violations by a factor d, and on the other hand the algorithms described are now parallelizable because updating d components of u and v does involve a matrix-vector product.

This idea is motivated by the numerical results of Altschuler, Weed, and Rigollet where the authors

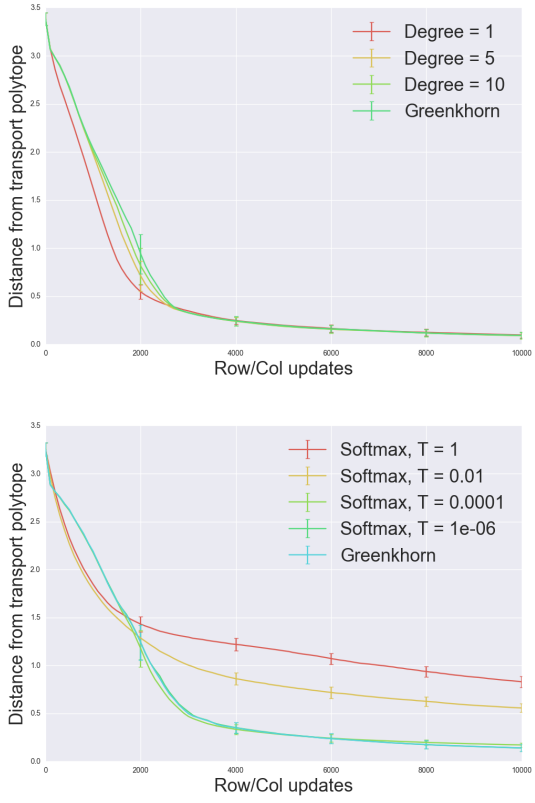

Figure 2: Stochastic Sinkhorn with different probability functions, and Greenkhorn as limiting case. Up: polynomial probabilities (11), down: softmax probabilities (12). For the x-axis, one should read "number of row and column updates" in the sense that one iteration on the x-axis represents one update of a row or a column.

concluded that the efficiency of Greenkhorn is mainly due to the fact that is does not update rows and columns that already match desired sums, more than the fact that it updates indexes with highest marginal violations. This means that the procedure of updating d indexes instead of just one is expected to have a similar efficiency, which is indeed the result we get in our own numerical experiments as shown in Figure 3.

5 Conclusion

We presented a family of stochastic algorithms for entropy-regularized OT problems. We were able to derive convergence rates for a very broad class of probability functions, along with numerical experiments where a simple and intuitive choice of probability functions performed the best. We also proposed and tested simple numerical solutions to the drawbacks of the greedy algorithms.

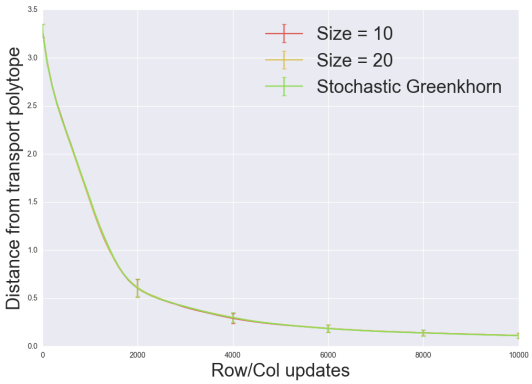

Figure 3: Evolution of distance from transport polytope for Block Stochastic Sinkhorn compared to Stochastic Sinkhorn ((11) with $\alpha = 1$). For the x-axis, one should read "number of row and column updates" in the sense that one iteration on the x-axis represents one update of a row or a column.

Acknowledgements

Both authors are indebted to Marco Cuturi for essentially teaching both of them about OT and many inspiring discussions. RMG acknowledges the support of the FSMP postdoctoral fund. Work done during BKA internship, who acknowledges CREST/ENSAE for funding, and the SIERRA team, Inria Paris, for hosting.

References

Altschuler, Jason, Jonathan Weed, and Philippe Rigollet (2017). "Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration". In: *CoRR* abs/1705.09634.

Courty, N., R. Flamary, and D. Tuia (2014). "Domain adaptation with regularized optimal transport". In: *Proceedings of ECML/PKDD 2014*. LNCS. Nancy, France, pp. 1–16.

Cuturi, Marco (2013). "Sinkhorn Distances: Lightspeed Computation of Optimal Transport". In: *Advances in Neural Information Processing Systems 26*, pp. 2292–2300.

Cuturi, Marco and Arnaud Doucet (2014). "Fast Computation of Wasserstein Barycenters". In: *Proceedings of the 31st International Conference on Machine Learning*. Vol. 32. Proceedings of Machine Learning Research 2. Bejing, China, pp. 685–693.

Frogner, Charlie, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya-Polo, and Tomaso A. Poggio (2015). "Learning with a Wasserstein Loss". In: *Advances in Neural Information Processing Systems (NIPS) 28*.

Genevay, Aude, Marco Cuturi, Gabriel Peyré, and Francis Bach (2016). "Stochastic Optimization for Large-scale Optimal Transport". In: *Advances in Neural Information Processing Systems 29*, pp. 3440–3448.

Gramfort, Alexandre, Gabriel Peyré, and Marco Cuturi (2015). "Fast Optimal Transport Averaging of Neuroimaging Data". In: *International Conference on Information Processing in Medical Imaging*. Vol. 9123. Lecture Notes in Computer Science, pp. 261–272.

Kantorovich, L. (1942). "On the translocation of masses". In: *Acad. Sci. URSS* 37, 199201.

Monge, Gaspard (1781). *Mémoire sur la théorie des déblais et des remblais*.

Montavon, Grégoire, Klaus-Robert Müller, and Marco Cuturi (2016). "Wasserstein Training of Restricted Boltzmann Machines". In: *Advances in Neural Information Processing Systems 29*, pp. 3718–3726.

Pele, Ofir and Michael Werman (2009). "Fast and robust Earth Mover's Distances." In: *ICCV*, pp. 460–467.

Solomon, Justin, Fernando De Goes, Gabriel Peyré, Marco Cuturi, Adrian Butscher, Andy Nguyen, Tao Du, and Leonidas Guibas (2015). "Convolutional wasserstein distances". In: *ACM Transactions on Graphics* 34.4, 66:1–66:11.

Villani, Cédric (2008). *Optimal Transport: Old and New*. 2009th ed. Grundlehren der mathematischen Wissenschaften.