Primal-Dual Randomized Coordinate Descent for Entropy regularized Optimal Transport using Sparse Structure

Tomohisa Tabuchi

Kasai Laboratory B4

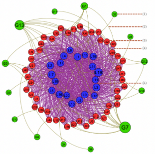
2022/12/23

Abstract of my research topic

- Computational Difficulties
  - The optimal transport problem is computationally expensive.
  - Several algorithms have been proposed to solve the entropy regularized optimal transport.
    - Sinkhorn Algorithm
    - Adaptive Primal-Dual Accelerate Gradient Descent
    - etc.
- Issues
  - Among them, we focus on Accelerated Primal-Dual Randomized Coordinate Descent (APDRCD).
  - APDRCD and other primal-dual algorithms have large dimensionality of the primal (or dual) variables.
  - Updating all variables takes a lot of time per iteration.
- Proposal
  - By taking advantage of the sparsity of matrix A in the constraint condition, the update of the primal variables can be reduced to only nonzero elements of the sampled rows of A.
  - We tried to improve by block sampling using the sparse structure of A.

Optimal Transport

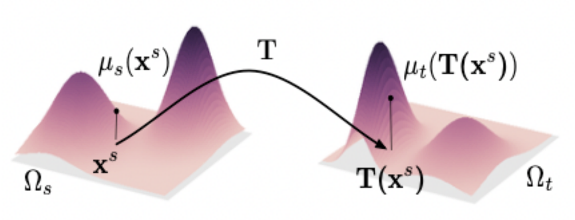
- Optimal transport (OT) is a tool for comparing distances between probability distributions
- It is becoming increasingly popular in machine learning.
  - Clustering
  - Supervised Learning
  - Domain Adaptation
  - Generative Adversarial Network



What is Optimal Transport?

- OT is the problem of finding the minimum energy to transport from a source to a target.
- The total energy for transportation

$$\sum_{i=1,j=1}^{n,m} C_{i,j} X_{i,j}$$



Formulation of the optimal transport problem

- OT formulation

$$\min_{X \in \mathbb{R}_{+}^{n \times m}} \quad \langle C, X \rangle \quad s.t.\ X\mathbf{1}_m = r,\ X^T\mathbf{1}_n = l$$

- $r \in \mathbb{R}^n$, $l \in \mathbb{R}^m$: input vector
- $C \in \mathbb{R}^{n \times m}$: cost matrix
- $X \in \mathbb{R}^{n \times m}$: transport matrix

- OT is a linear programming problem.
- It can be solved by the interior point method, with a best known practical complexity of $\tilde{O}(n^3)$.
- It cannot be used effectively in machine learning where the number of dimensions n becomes large.

Entropy regularized Optimal Transport [Cuturi, 2013]

$$\min_{X \in \mathcal{U}(r,l)} \quad \langle C, X \rangle - \eta H(X)$$

where

$$\mathcal{U}(r,l) \coloneqq \{X \in \mathbb{R}^{n \times m} : X\mathbf{1}_m = r, X^T\mathbf{1}_n = l\}$$

- [Cuturi, 2013] showed that the OT can be easily solved by adding an entropy regularization term.
- Sinkhorn Algorithm

$$u_{k+1} = \frac{r}{Kv_k}, \quad v_{k+1} = \frac{l}{K^T u_{k+1}}$$

- Computational Complexity $O(\frac{n^2}{\epsilon^2})$

Reformulate the entropy regularized optimal transport

$$\min_{x} \quad c^T x - \eta H(x) \quad s.t.\ Ax = b$$

where $(n = 3, m = 3)$

$$A = \begin{pmatrix} 1 & 1 & 1 & & & & & & \\ & & & 1 & 1 & 1 & & & \\ & & & & & & 1 & 1 & 1 \\ 1 & & & 1 & & & 1 & & \\ & 1 & & & 1 & & & 1 & \\ & & 1 & & & 1 & & & 1 \end{pmatrix}, \quad b = \begin{pmatrix} r \\ l \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ l_1 \\ l_2 \\ l_3 \end{pmatrix}$$

- For simplicity, let n=m.
- $A \in \mathbb{R}^{2n \times n^2}$, $b \in \mathbb{R}^{2n}$, $c \in \mathbb{R}^{n^2}$, $x \in \mathbb{R}^{n^2}$
- $H(x) = -\sum_i x_i(\log x_i - 1)$
- $c = (C_{1,1}, C_{1,2}, \cdots, C_{n,n})^T$, $x = (X_{1,1}, X_{1,2}, \cdots, X_{n,n})^T$

General Problem and Primal-Dual Formulation

- We consider the optimization problem:

$$\min_{x \in \mathcal{X}} \quad f(x) \quad s.t. \ Ax = b$$

  - $f(x)$: smooth
  - A : linear operator.
- Lagrange dual problem:

$$\min_{y \in \mathcal{Y}} \left\{ \phi(y) := \langle y, b \rangle + \max_{x \in \mathcal{X}} \quad \left\{ -f(x) - \langle A^T y, x \rangle \right\} \right\}$$

- $\nabla \phi(y)$ is Lipschitz-continuous:

$$||\nabla \phi(y_1) - \nabla \phi(y_1)||_2 \leq L||y_1 - y_2||_2$$

Primal-Dual Method for Optimal Transport

- In the case of OT, Lagrange dual problem is as follows:

$$
\begin{aligned}
&\min_{x \in \mathcal{X}} \quad c^T x - \eta H(x) \quad s.t.\ Ax = b \\
\iff &\min_{y \in \mathcal{Y}} \phi(y) := \langle y, b \rangle + \max_{x \in \mathcal{X}} \quad \Big\{ -c^T x + \eta H(x) - \langle A^T y, x \rangle \Big\}
\end{aligned}
$$

- The updates for the primal and dual variables are as follows:

$$
\begin{aligned}
x^{k+1} &= \arg\max_x \Big\{ -c^T x + \eta H(x) - \langle A^T y^k, x \rangle \Big\} \\
&= \exp\left(-\frac{c + A^T y_k}{\eta} \right) \\
y^{k+1} &= y^k - \frac{1}{L} \nabla \phi(y^k) \\
&= y_k - \frac{1}{L} (A x^{k+1} - b)
\end{aligned}
$$

Primal-Dual Randomized Coordinate Descent Method

- Primal-Dual Randomized Coordinate Descent is a stochastic extension of the primal-dual method:

1. Update primal variables for all $i \in \{1, 2, \ldots, n^2\}$

$$x_i^{k+1} = \exp\left(-\frac{c_i + (A^T y^k)_i}{\eta}\right)$$

2. Randomly sample one coordinate $j_k \in \{1, 2, \ldots, 2n\}$
3. Update dual variables

$$y_{j_k}^{k+1} = y_{j_k}^k - \frac{1}{L}\nabla_{j_k}\phi(y^k)$$

- [Guo et al., 2020] proposed an accelerated algorithm based on this primal-dual randomized coordinate descent method.
  - Accelerated Primal-Dual Randomized Coordinate Descent (APDRCD)

Proposed Method 1

- Issues with APDRCD
  - Update all of the primal variables (all coordinates)
  - Updating all variables takes a lot of time per iteration.
- Our Methods
  - Update only variables x_i with related to y_{j_k} that was updated.
  - No need to update for x_i with related to y_{j_k} that was not updated.

$$x_i^{k+1} = \exp\left(\frac{-c_i - (A^T y^{k+1})_i}{\eta}\right)$$

Proposed Algorithm: Sparsity-Aware APDRCD

1. Randomly sample one coordinate $j_k \in \{1, 2, \ldots, 2n\}$
2. Update dual variables

$$y_{j_k}^{k+1} = y_{j_k}^{k} - \frac{1}{L} \nabla_{j_k} \phi(y^k)$$

3. Find x_i^{k+1} to update using $y_{j_k}^{k+1}$

$$I(j_k) = \left\{ i \in \{1, 2, \ldots, n\} : \ A_{j_k, i} \neq 0 \right\}$$

4. Update primal variables for all $i \in I(j_k)$

$$x_i^{k+1} = \exp\left(\frac{-c_i - (A^T y^{k+1})_i}{\eta} \right)$$

Proposed Method 2

- A has a special sparse structure.

$$A = \begin{pmatrix} 1 & 1 & 1 & & & & & & \\ & & & 1 & 1 & 1 & & & \\ & & & & & & 1 & 1 & 1 \\ 1 & & & 1 & & & 1 & & \\ & 1 & & & 1 & & & 1 & \\ & & 1 & & & 1 & & & 1 \end{pmatrix}$$

- Block sampling might can further simplify variable updates.
- Experiment with various sampling methods that are aware of the sparse structure of A.

greedy

$$A = \begin{pmatrix} 1 & 1 & 1 & & & & & & \\ & & & 1 & 1 & 1 & & & \\ & & & & & & 1 & 1 & 1 \\ 1 & & & 1 & & & 1 & & \\ & 1 & & & 1 & & & 1 & \\ & & 1 & & & 1 & & & 1 \end{pmatrix}$$

cyclic

$$A = \begin{pmatrix} 1 & 1 & 1 & & & & & & \\ & & & 1 & 1 & 1 & & & \\ & & & & & & 1 & 1 & 1 \\ 1 & & & 1 & & & 1 & & \\ & 1 & & & 1 & & & 1 & \\ & & 1 & & & 1 & & & 1 \end{pmatrix}$$

original

$$A = \begin{pmatrix} 1 & 1 & 1 & & & & & & \\ & & & 1 & 1 & 1 & & & \\ & & & & & & 1 & 1 & 1 \\ 1 & & & 1 & & & 1 & & \\ & 1 & & & 1 & & & 1 & \\ & & 1 & & & 1 & & & 1 \end{pmatrix}$$

vertical

$$A = \begin{pmatrix} 1 & 1 & 1 & & & & & & \\ & & & 1 & 1 & 1 & & & \\ & & & & & & 1 & 1 & 1 \\ 1 & & & 1 & & & 1 & & \\ & 1 & & & 1 & & & 1 & \\ & & 1 & & & 1 & & & 1 \end{pmatrix}$$

nblock

$$A = \begin{pmatrix} 1 & 1 & 1 & & & & & & \\ & & & 1 & 1 & 1 & & & \\ & & & & & & 1 & 1 & 1 \\ 1 & & & 1 & & & 1 & & \\ & 1 & & & 1 & & & 1 & \\ & & 1 & & & 1 & & & 1 \end{pmatrix}$$

20block

$$A = \begin{pmatrix} 1 & 1 & 1 & & & & & & \\ & & & 1 & 1 & 1 & & & \\ & & & & & & 1 & 1 & 1 \\ 1 & & & 1 & & & 1 & & \\ & 1 & & & 1 & & & 1 & \\ & & 1 & & & 1 & & & 1 \end{pmatrix}$$

20batch

$$A = \begin{pmatrix} 1 & 1 & 1 & & & & & & \\ & & & 1 & 1 & 1 & & & \\ & & & & & & 1 & 1 & 1 \\ 1 & & & 1 & & & 1 & & \\ & 1 & & & 1 & & & 1 & \\ & & 1 & & & 1 & & & 1 \end{pmatrix}$$

vblock

$$A = \begin{pmatrix} 1 & 1 & 1 & & & & & & \\ & & & 1 & 1 & 1 & & & \\ & & & & & & 1 & 1 & 1 \\ 1 & & & 1 & & & 1 & & \\ & 1 & & & 1 & & & 1 & \\ & & 1 & & & 1 & & & 1 \end{pmatrix}$$

Experiments

- Comparison
  - Execution time between the original APDRCD and the Sparsity-Aware APDRCD.
  - Compared original APDRCD with various block sampling methods.
- Dataset
  - Two probability distributions generated by uniform random numbers are used as input.

Experiments: APDRCD vs Sparsity-Aware APDRCD

- Original APDRCD converges a little faster than Sparsity-Aware.
- Sparsity-Aware APDRCD takes less time for each iteration than original one.

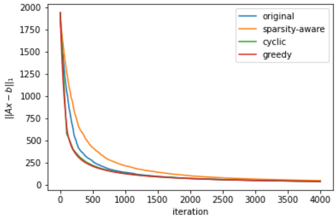


Figure 1: Distance to the transportation polytope per iteration

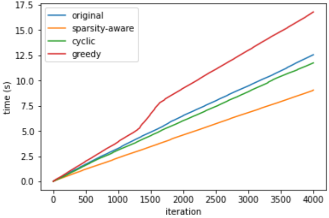


Figure 2: Execution time per iteration

Experiments: APDRCD vs APDRCD with block sampling

- In the error, "nblock" has the fastest descent along with "greedy".
- (Block sampling algorithms are not comparable in execution time to the other algorithms.)

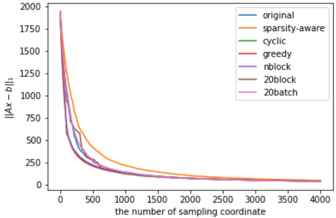


Figure 3: Distance to the transportation polytope/the number of sampled coordinates

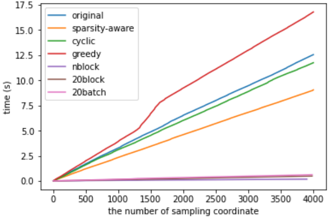


Figure 4: Execution time/the number of sampled coordinates

Comparison between block sampling algorithms

- In the error, "nblock" has a faster descent, but they are all about the same.
- "20batch" takes longer time than "20block".

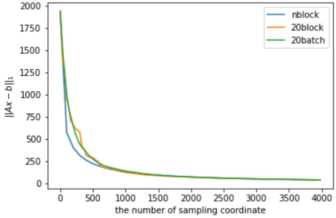


Figure 5: Distance to the transportation polytope/the number of sampled coordinates

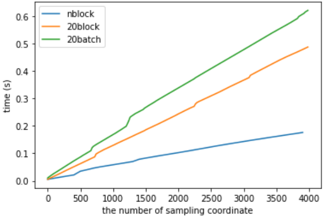


Figure 6: Execution time/the number of sampled coordinates because

Discussion

- Proposal 1
  - "sparsity-aware": shortest execution time
    - reduce the number of variables to be updated per iteration
  - "greedy": longest execution time
    - It took a long time to find the largest gradient of dual objective function.
- Proposal 2
  - "20batch" takes longer time than "20block".
    - It took a lot of time for sampling minibatches.

Future Work

- Experiment on real-world data and data with high dimensionality
- Convergence analysis of APDRCD algorithms with block sampling
- Compare proposed algorithm with other stochastic algorithms
- Find a suitable OT application for this algorithm

References I

- Alacaoglu, A., Fercoq, O., and Cevher, V. (2020).
  Random extrapolation for primal-dual coordinate descent.
  In III, H. D. and Singh, A., editors, Proceedings of the 37th International Conference on Machine Learning, volume 119 of Proceedings of Machine Learning Research, pages 191–201. PMLR.

- Cuturi, M. (2013).
  Sinkhorn distances: Lightspeed computation of optimal transport.
  In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K., editors, Advances in Neural Information Processing Systems, volume 26. Curran Associates, Inc.

References II

- Guo, W., Ho, N., and Jordan, M. (2020).
  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.