SPOT: Sliced Partial Optimal Transport

-

fast python implementation, algorithm improvements, and problems

Auteurs:
Quentin SPINAT

December 27, 2020

Contents

Abstract

This is paper is the project report for the 2020 MVA course *Computational Optimal Transport* taught by Gabriel Peyré. The purpose of it is to study the paper *SPOT: Sliced Partial Optimal Transport* [1].

Optimal Transport is a mathematical field that focus on finding the best way to match two sets with respect to a matching cost. Here we will focus on a subproblem which consists in finding the best injective assignment of a finite set of points to a second bigger set of points. Although it is a very precise problem, it is often encountered : color transfer from an Image to a bigger one, matching a small point clouds to a bigger one, ...

Nicolas Bonneel et al. proposed a method they called FIST : Fast Iterative Sliced Transport to solve this problem. It consists in solving the assignment problem considering only 1D optimal assignments, which are fast and easy to compute. Their algorithm is told to be quasilinear in time with respect to the datasets sizes.

In this project, I clarify the explanations of the original paper (which is full of typos, especially in the algorithm description), make a fast and user-friendly implementation of this algorithm in Python, and apply it to color transfer and point cloud registration.

Even though I wasn't able to parallelize my code due to python limitations, the implementation I made revealed to be faster than the original implementation in C++ for color transfer application. However I couldn't be able to reproduce good results for point cloud registration.

Keywords— Optimal Transport, Optimal assignment, SGD, Shape matching, Color transfer

––––––––––––––––––

Introduction

Problem

Optimal Transport is a mathematical field that focus on finding the best way to match two sets with respect to a matching cost. Here we will focus on a subproblem which consists in finding the best injective assignment of a finite set of points to a second bigger set of points. Although it is a very precise problem, it is often encountered : color transfer from an Image to a bigger one, matching a small point clouds to a bigger one.

Previous Work

Optimal transport is a computationally complex problem, often expressed as a linear program due to Kantorovich. The variables being optimized form what is known as a transport plan to describe the amount of mass from each bin of

an input (possibly high-dimensional) histogram that needs to travel to each bin of a target histogram. In this definition, it is assumed that both histograms are normalized. Directly solving this linear program can be very costly in time and memory, and when dealing with a large amount of points, algorithms become very greedy, especially because of the curse of dimensionality. Faster alternate approaches have been developed to tackle particular cases, but cannot be applied to the problem we try to overcome.

This is why the field of sliced optimal transport has emerged. It consists in projecting the point clouds onto random one-dimensional subspaces, and solving one-dimensional transport problems [Pitie et al. Rabin et al. [2], Bonneel et al. [3], [4]]. Slicing methods are very fast because of how simpler one-dimensional optimal transport is to solve. However, Sliced optimal transport is only an approximation and comes at the cost of being optimal only on the projections and not necessarily on the higher dimensional space. But optimal transport for non-normalized histograms has received little attention, even in one-dimension, which makes the sliced approach not possible for point clouds of different sizes yet.

When histograms are not normalized, the transport problem needs to be relaxed. Notably, the "unbalanced" optimal transport of Benamou et al. [5] replaces hard constraints by soft constraints in a Monge formulation to accommodate for densities of unequal masses within a PDE framework. However, it suffers from a big space complexity. Figalli introduces *partial optimal transport* that keeps all constraints of the linear program hard [6], but replaces equality with inequality constraints. The problem we are interested in is a special case of this linear program for the problem of matching point clouds.

New Contributions

Nicolas Bonneel et al. [1] proposed a Sliced Partial Optimal Transport (SPOT) method they called FIST : Fast Iterative Sliced Transport to solve the problem of optimal injective assignment between point clouds of different sizes. This is a sliced method which consists in minimizing the mean over all the one-dimensional projections of the Wassertein distance between the two 1D projected point clouds. It hence makes it possible to solve the original problem by doing a Stochastic Gradient Descent and solving only the 1D projected subproblems. To solve these one-dimensional partial optimal transport subproblems efficiently, they propose an algorithm quasilinear in time with respect to the datasets sizes.

Applications

In this project, we focus on two applications of this new method : color transfer problem and point cloud registration.

Color Transfer

Transferring colors between images has become a classical image processing problem. The purpose is to change the color distribution of an input image, without

changing its content, to match the style of a target image. Numerous optimal transport solutions have already been developed [Bonneel et al. [4], [7]; Pitié et al. [2]; Pitié et al. [8]; Rabin et al. [9]]. In addition to their use of optimal transport, a common point to these approaches is that they consider the problem of matching normalized histograms. A consequence of that is often acknowledged as a limitation: their content should not differ. For instance, matching an image with 80% trees and 20% sky to an image containing the opposite ratio will inevitably lead to trees becoming blue or sky becoming green. Several approaches also require the images to have exactly the same number of pixels – this is precisely the case for sliced transportation [Bonneel et al. [4]; Rabin et al. [9]].

Bonneel et al. proposed two solutions to overcome these problems, but I only focus on one of them : this solution simply enlarges the target images to give more freedom to each input pixel to be matched to target pixel values. This method is really fast and produces really good results

Point Cloud Registration

A common problem in point cloud processing is that of registering points under a given transformation model. For instance, one tries to match a point cloud with another by supposing the transformation between them is rigid, or constrained to a similarity, or affine, or homographic, etc. A well-known algorithm to solve this problem is the Iterative Closest Point (ICP) algorithm. Given a point cloud X_0 to be matched against X_1 , this algorithm proceeds by first matching points of X_0 to their nearest neighbors in X_1 , and, given this assignment, the best transformation in the class of allowed transformations is found by minimizing an energy. The algorithm then transforms the initial point cloud using the computed transformation. The process is repeated until convergence. However, this algorithm requires points clouds to be relatively close to start with and suffers from local minima. In practice, extremely bad behaviors may arise when considering similarity transforms (rotation, translation and scaling). In that case, the lack of injectivity of the nearest neighbor map tends to estimate progressively smaller scaling factors as iterations increase, occasionally leading to a trivial zero scale solution.

This motivates the use of a metric which accounts for an injective mapping, such as the sliced metric proposed by Bonneel et al. They thus propose to replace the nearest neighbor matching by a partial sliced optimal transport matching they call *Fast Iterative Sliced Transport* (FIST).

1 Partial Transport in 1-D

This section will explain the core of this new method, which is an efficient algorithm for Partial Transport in one dimension. It relies on the fact that the nearest neighbor assignment of two point sets can be found in linear time in one-dimension. However, this nearest neighbor assignment not being injective. However, we need to smartly modify it. This can easily be done in quadratic time. To go further and improve the algorithm performances, we need then to introduce a way to decompose the original problem into independent easier problems, which can be solved in parallel. With this problem decomposition, the whole algorithm becomes quasilinear in time.

In all the problem we consider that X is a set of 1-D points of cardinal m, Y a set of 1-D points of cardinal $n < m$, and that X and Y are sorted by increasing value (this can be done in quasilinear time).

1.1 Nearest Neighbor Assignment

In one dimension, the nearest neighbor assignment $t : [\![1, m]\!] \to [\![1, n]\!]$ from X to Y can be found in linear time. Since $t(i) \leqslant t(i + 1)$, the algorithm consists in scanning X and Y simultaneously from left to right and compare their value.
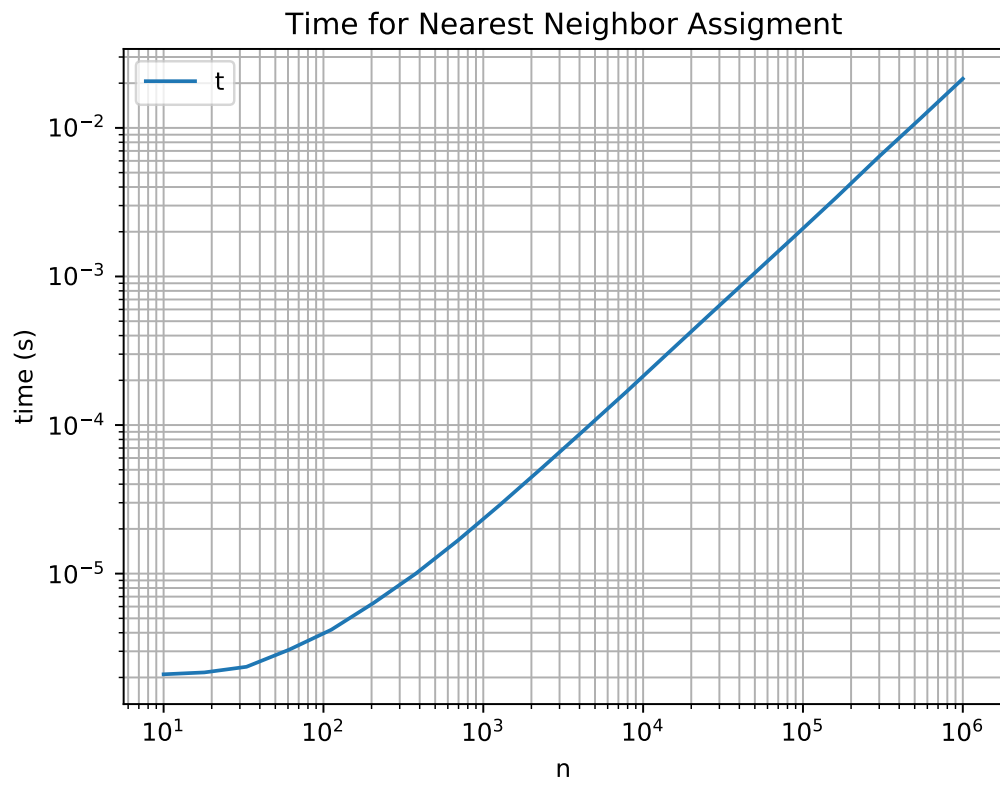

Algorithm 1 Nearest Neighbor Assignment

    Input: X, Y
    Output: t

1: $i \leftarrow 0$
2: $j \leftarrow 0$
3: **while** $i < m$ **do**
4:     **if** $X_i \leqslant Y_j$ **then**
5:         $t[i] \leftarrow j$
6:         $i \leftarrow i + 1$
7:     **else if** $j = n - 1$ **then**:
8:         $t[i] \leftarrow n - 1$
9:         $i \leftarrow i + 1$
10:     **else if** $Y_{j+1} < X_i$ **then**
11:         $j \leftarrow j + 1$
12:     **else if** $|X_i - Y_j| < |X_i - Y_{j+1}|$ **then**:
13:         $t[i] \leftarrow j$
14:         $i \leftarrow i + 1$
15:     **else**
16:         $t[i] \leftarrow j + 1$
17:         $j \leftarrow j + 1$
18:         $i \leftarrow i + 1$

Mettre une image de l'assignement.

Time for Nearest Neighbor Assigment

Conclusion

Remerciements

References

[1] Nicolas Bonneel and David Coeurjolly. Spot: Sliced partial optimal transport. *ACM Transactions on Graphics (SIGGRAPH)*, 38(4), 2019.

[2] Francois Pitie, Anil C Kokaram, and Rozenn Dahyot. N-dimensional probability density function transfer and its application to color transfer. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1434–1439. IEEE, 2005.

[3] Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernot. Wasserstein barycenter and its application to texture mixing. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 435–446. Springer, 2011.

[4] Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45, 2015.

[5] Jean-David Benamou. Numerical resolution of an "unbalanced" mass transport problem. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 37(5):851–868, 2003.

[6] Alessio Figalli. The optimal partial transport problem. *Archive for rational mechanics and analysis*, 195(2):533–560, 2010.

[7] Nicolas Bonneel, Kalyan Sunkavalli, Sylvain Paris, and Hanspeter Pfister. Example-based video color grading. *ACM Trans. Graph.*, 32(4):39–1, 2013.

[8] François Pitié, Anil C Kokaram, and Rozenn Dahyot. Automated colour grading using colour distribution transfer. *Computer Vision and Image Understanding*, 107(1-2):123–137, 2007.

[9] Julien Rabin, Julie Delon, and Yann Gousseau. Regularization of transportation maps for color and contrast transfer. In *2010 IEEE International Conference on Image Processing*, pages 1933–1936. IEEE, 2010.