

# A Two Stage Optimal Transport based System Identification method

B. Tsakam-Sotché  
Horgen, CH-8810 Switzerland

## Abstract

We present a novel system identification method that overcomes the local minima problem yet converges to accurate system parameters. In the first stage, the algorithm minimizes the wasserstein distance by the mean of an optimal transport solver without resorting to a gradient for the wasserstein distance cost. In the second stage, the algorithm achieves fast convergence using conventional gradient descent methods in the euclidean distance cost. This method is also simple to implement using off the shelf software packages for optimal transport and for gradient descent optimization.

## 1. INTRODUCTION

System identification in the large class of linear time invariant systems, is completely defined by its impulse response. For time discrete signals, the output of the system  $y(n)$  is entirely defined by the input  $x(n)$  and the impulse response  $h(n)$  through the convolution operator:

$$y(n) = x(n) \otimes h(n)$$

The system identification is the inverse problem that consists in inferring  $h(n)$  given the system input and output signals denoted by  $x(n)$  and  $y(n)$ , respectively. Here we concern ourselves with stochastic systems or at least positive signals and positive impulse responses scaled so that all samples sum up to 1. That is:

$$\begin{aligned}\sum_n x(n) &= 1, x(n) \geq 0 \forall n \\ \sum_n h(n) &= 1, h(n) \geq 0 \forall n\end{aligned}$$

The acclaimed Wiener deconvolution suits best frequency domain applications and is even robust to moderate noise additive [Wiener]. In the time domain, gradient based methods are commonly used for this problem. For example, ADMM methods [Gray'2000] allow including a regularization term for robustness noise. The basic gradient descent method seeks a stationary point of the euclidean cost function given the impulse response:

$$cost(h) = [y(n) - x(n) \otimes h(n)]^2$$

When the transform function is a convolution, it can be represented with a matrix multiplication. The system identification problem becomes:

$$Y_{m \times 1} = H_{m \times m} \times X_{m \times 1}$$

The matrix  $H$  a convolution matrix or a circular matrix [Gray'2000 chap 1.1]. Gradient descent of convex and constrained problems using ADMM method converges rapidly  $O(1/k)$  in the number of steps  $k$  [Goldstein'2014].

However if the system's impulse response or transform function involves a translation, the problem becomes non-convex in the euclidean cost. This issue is well documented in the context of image registration (see for example [Taylor'2008]). Thus, the search algorithm might end in a local minima and never reach the global minima.

## 2. PRIOR WORK

Previously, the gaussian clustering problem, also facing the local minima issue, was solved using the wd distance [Kolouri'2017]. We take a similar approach for the system identification problem but our method is based on the transport map. [El Moselhy'2012] lays very relevant theoretical foundations for this work and to our knowledge their method is the closest prior art to ours. We also aim at finding the transform function that minimizes the wasserstein distance in an iterative fashion and subsequently build  $h$  as a composition of partial transform functions. We propose a simpler algorithm with explicit and separated Optimal Transport and Regression steps that does not require any gradient. From the computer vision community, the registration task aims at finding the transform that relates 2 images, given a translation and a 2D (or 3D) rotation or other affine transformations. The registration transform is commonly found using a randomized search and a cost function that involves both topological and euclidean distance terms [Bleyer'2011].

## 3. PROPOSED METHOD

### 3.1. Optimal transport

The computation of the Wasserstein distance  $wd$  amounts to minimal the total cost of transforming an input probability density function (pdf)  $X_{1 \times m}$ , into an output pdf  $Y_{m \times 1}$ , by the mean of a transportation plan matrix  $P_{m \times m}$  subject to a cost matrix  $C_{m \times m}$  such that:

$$X_{1 \times m} = P \times I^T$$

$$Y_{m \times 1} = P \times I$$

$$wd = \min_P \langle C, P \rangle$$

With  $\langle \cdot, \cdot \rangle$  for the Frobenius dot product.

Recent Optimal Transport (OT) solvers effectively compute the wasserstein distance  $wd$  and the transport plan  $P$  [Flamary'2017]. In the remainder, we use  $wd$  to refer to the wd distance where the distance cost in  $C$  increases as  $d^t$  where  $d$  is the euclidean distance in the coordinates (of  $X$  and  $Y$ ) and  $t$  is larger or equal to 1.

### 3.2. Transport plan and convolution

First, we observe that if  $h$  is known, the convolution matrix corresponds to a valid transportation plan. Given that the transport plan is a doubly stochastic matrix where the sum of the rows equals the input pdf  $X$  and the sum of the columns equals the output pdf  $Y$ . Let set the  $i^{\text{th}}$  column  $c_i$  of the transport plan as the impulse response  $h(n)$ , with the centered  $h(0)$  at the diagonal, ie  $c_i = h(n-i)$ , times the amplitude of the corresponding input sample  $x[i]$ . We have a valid column for the transport plan. Doing the same for all  $m$  columns results in a valid transport plan such that satisfies the output signals requirements:  $Y_{m \times 1} = P \times I$  since  $Y_{m \times 1} = H_{m \times m} \times X_{m \times 1}$ .

Using an OT solver, we obtain the distance  $wd$  and the transport map  $P$ . The transport map represents the most parsimonious (wrt the cost matrix  $C$ ) warp of the input signal into the output signal. We now turn our attention to recovering  $H$  from  $P$ .

### 3.3. Transport plan and regression

The warp for a given input sample defines its reallocated weights in the output signal. For 1D signals, the warp for a given sample is entirely defined by its column in the transport map. The OT solver does not require the warp to be independent of the location however that is a requirement for a time invariant system.

To recover the transform out of the transport plan, and enforce the time invariant constraint, we regress the allocations at each offset for each sample into a candidate filter, the new  $h$  as shown in Algorithm I. Note that we could allow the regression parameters to be adaptive with respect to location ( $n$ ), or on other conditionals.

| Algorithm I                                                                                                                                                                                                                                                                                                                                                                                                            |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none"> <li>1. <math>wd, P = \text{OT.solve}(X, Y, C)</math><br/>stop if <math>wd</math> converged</li> <li>2. <math>h = \text{regress}(P)</math>, convert to the circular matrix <math>H</math><br/>regress: weighted average over all samples of the allocation <math>h[\tau]</math> for each offset <math>\tau</math></li> <li>3. <math>X := H \times X</math>, go to step 1.</li> </ol> |

For the regression step ( $\text{regress}(\cdot)$  in step 2.), a simple average is sufficient in the case of a toeplitz cost matrix, however the regression here aims to minimize the resulting  $wd$  depending on the cost matrix  $C$ .

#### Regression: Linear time invariant system

The regression for  $h[\tau]$  at offset  $\tau$  is estimated as follows for a toeplitz cost matrix:

We know  $h[\tau] = \sum_{ci=0}^m x[n] * h[\tau] / [\sum_{ci=0}^m x[n]]$ , with  $\sum_{ci=0}^m x[n] = 1$ ,

In section 3.2 we stated that  $P(ci, ci+\tau) \approx h[\tau] * x[n]$ , where  $ci$  is the column index corresponding to the sample  $x[ci]$  of the input signal. Therefore we use the regression

$$h[\tau] = \sum_{ci=0}^m P(ci, ci+\tau)$$

The regression for  $h[\tau]$  at offset  $\tau$  is estimated as follows for a general (non-toeplitz) cost matrix:

$$h[\tau] = [\sum_{ci=0}^m P(ci, ci+\tau) * C(ci, ci+\tau)] / \sum_{ci=0}^m C(ci, ci+\tau)$$

The cost matrix drives the  $wd$  distance and indicates where  $h$  should make the input signal more similar to the output signal. In practice we use a modified cost matrix  $C1 = C + \kappa$ , where  $\kappa$  is a small constant so that  $C1$  has no zero entries and the weighted sum in  $C1(ci, ci+\tau)$  is always defined.

Looking at the update step 3., we are identifying the system as a sequence of convoluted filters  $h$ . The convoluted filters result in the optimal transform function. When the optimal transform function is found,  $wd$  is stationary at the global minima.

#### Regression: Translation

We have the a priori model:

$$y(n) = x(n+\Delta), \text{ where } \Delta \text{ is an integer}$$

And we want to identify  $\Delta$ .

The linear time invariant or filtering approach is suitable for small translation - within the support of the filter. For a more general case, it is possible to resort to calculating the cross correlation and picking the maximal point. Using a regression on the optimal transport plan, we regress over all the allocations to extract the average transportation vector as the translation vector  $T^*$ :

$$T^* = \sum_{ci=0}^m \sum_{t=0}^m P(ci, ci + t) * t * e^*$$

$$\text{and } T^* = \Delta * e^*$$

Where  $e^*$  is the unit vector for transportation.

### **Regression: Centered affine transform**

We have the a priori model:

$$y(n) = x(\lambda^*(n-\vartheta)), \text{ where } a \text{ is a positive real number and } b \text{ is an integer.}$$

This transform is morphological in nature and gradient descent on a cost based on euclidean distance is not convex. For this type of problem researchers might resort to exhaustive search, randomized algorithms [Bleyer'2011] or use their domain expertise to devise effective heuristics.

We observe that the transportation cost should be close to zero at the center of the affine transform and grow linearly as the distance from the center grows.

say the center is at  $c^*$ , we have:

$$\sum_{t=0}^m P(ci, ci+t) * t = a * (ci-c^*), \text{ which is formulated as a regression}$$

The solution of this regression gives  $\lambda \leftarrow a, \vartheta \leftarrow c^*$

In all three regression cases, this method empirically converges on synthetic 1D data. The exact conditions for convergence remain to be defined.

### 3.4. Two Stage Optimal Transport System Identification

We observed using globally convex toy examples that where the Algorithm I would obtain an optimal solution with an euclidean error of the order  $1e-2$ , a straightforward gradient descent would deliver an euclidean error of the order of  $1e-5$  with nearly the same CPU time.

For applications where a high accuracy is required, we propose a two phase algorithm in the same spirit as Brent's method for root finding [Brent'1973]. The first stage uses OT to avoid the local minima trap and come in the vicinity of the global minima. The second stage using gradient descent guarantees fast convergence in the euclidean sense but within a bounded margin from the optimal point from the wasserstein distance point of view. The second stage is shown in Algorithm II. Here,  $wd\_min$  is the optimal  $wd$  distance obtained from algorithm I. The Two Stage Optimal Transport System Identification or TSOTSI reaches a guaranteed near global optimum solution. It is shown in Algorithm III.

| Algorithm II                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none"> <li>1. calculate the gradient in the euclidean sense of <math>  Y - H \times X  ^2</math><br/> <math>h'[n] = h[n] - \text{grad\_}h[n] * \text{learning\_step}</math></li> <li>2. <math>wd, P = \text{OT.solve}(X, Y, C)</math><br/> if <math>wd &gt; wd\_min + \text{tolerance}</math>, stop</li> <li>3. <math>h = h'</math>, convert <math>h</math> to <math>H</math>, go to 1</li> </ol> |

Note that in Algorithm II, step 1, automatic differentiation may be used [AD]. Alternatively, we may use other gradient descent based methods including ADMM.

| Algorithm III                                                                                                                                                                                                   |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none"> <li>1. find the optimal <math>h</math> in the <math>wd</math> sense using algorithm I</li> <li>2. refine <math>h</math> using gradient descent using algorithm II</li> </ol> |

#### 4. EXTENSIONS

The extension to multi-dimensional signals is straightforward using the appropriate circular matrix [Gray'2000].

Because of the commutativity of the convolution, our method applies to deconvolution. We still need to assess the robustness of our method in the presence of noise.

During the first stage of the proposed method, the wasserstein distance can be plugged-in as a cost in a multi-criterion problem using an Alternating Direction Optimization method [Goldstein'2014], such as ADMM. Within the ADMM framework, additional constraints or projections can be enforced, for example to a subset of parametric transform functions  $h$ . The same additional constraint would also be applied during the second stage further using ADMM.

#### 5. SUMMARY

We presented a novel system identification method that overcomes the local minima problem in cases where the transform function involves a translation or is otherwise not globally convex in the euclidean cost. This method is simple to implement using the transport map obtained through available OT solvers [Justin'2015, Xie'2018].

In practice, it shows strong convergence that can be monitored using the Wasserstein distance at each iteration. While a strong theoretical basis is provided in [El Moselhy'2012], the exact conditions for convergence remain to be defined.

As exemplified in [El Moselhy'2012], the TSOTSI method allows to overcoming the limitations of euclidean distance based gradient descent (local minima) and the convergence issue of Markov Chain Monte-Carlo (sample generation, convergence monitoring) in system identification.

Our method can be applied to positive signal and positive transform function inverse problems, which are common in several domains such as signal processing, machine vision, or chemical interaction analysis.

## References

[AD]

[https://en.wikipedia.org/wiki/Automatic\\_differentiation](https://en.wikipedia.org/wiki/Automatic_differentiation)

[Bleyer'2011]

M. Bleyer, C. Rhemann, C. Rother

“PatchMatch Stereo - Stereo Matching with Slanted Support Windows”

[Brent'1973]

Richard P. Brent

"Chapter 4: An Algorithm with Guaranteed Convergence for Finding a Zero of a Function",

“Algorithms for Minimization without Derivatives”, Englewood Cliffs, NJ: Prentice-Hall

[El Moselhy'2012]

Tarek A. El Moselhy, Youssef M. Marzouk

Bayesian Inference with Optimal Maps

[Flamary'2017]

Rémi Flamary and Nicolas Courty

“POT Python Optimal Transport library” [pythonot.github.io](https://pythonot.github.io)

[Goldstein'2014]

Tom Goldstein, Brendan O'Donoghue, Simon Setzer, and Richard Baraniuk

“Fast Alternating Direction Optimization Methods”

[Gray'2000]

Robert M Gray

“Toeplitz and Circulant Matrices: A review”

[Houdard'2021]

Antoine Houdard, Arthur Leclaire, Nicolas Papadakis, Julien Rabin

"On the Existence of Optimal Transport Gradient for Learning Generative Models"

[Justin'2015]

Solomon Justin, Fernando de Goes, Gabriel Peyré, Marco Cuturi, Adrian Butscher, Andy Nguyen, Tao Du, and Leonidas Guibas.

“Convolutional Wasserstein Distances: Efficient Optimal Transportation on Geometric Domains”

[Kolouri'2017]

Soheil Kolouri, Gustavo K. Rohde, Heiko Hoffmann

“Sliced Wasserstein Distance for Learning Gaussian Mixture Models”

[Taylor'2008]

Camillo Jose Taylor and Arvind Bhusnurmth

Insert PSN Here

“Solving Image Registration Problems Using Interior Point Methods.”

[Wiener]

[https://en.wikipedia.org/wiki/Wiener\\_filter](https://en.wikipedia.org/wiki/Wiener_filter)

[Xie’2018]

Yujia Xie, Xiangfeng Wang, Ruijia Wang, Hongyuan Zha

“A Fast Proximal Point Method for Computing Exact Wasserstein Distance”