Wasserstein Embeddings

Zamolodtchikov Petr, Vacher Adrien

June 6, 2019

1 Introduction

In this project, we studied the framework devellopped by [Frogner et al., 2019] which allows to visualize each element of a dataset as a point cloud in a low dimension space.
The central idea of their paper is to embed the input data as non-parametric probability distributions in a Wasserstein space. These emebeddings are computed by matching the Wasserstein distance between the distributions with the inherent semantic structure of the data. While the exact Wasserstein distance is costly to compute, it can be efficiently approximated with the Sinkhorn divergence [Cuturi, 2013].
When the supporting space of the distributions is low dimensional (typically $\mathbb{R}^2$ or $\mathbb{R}^3$), it is possible to directly visualize the embeddings. This isn't the case for euclidean embeddings for instance which require a high number of dimensions for accuratly representing the data. If we want this visual representation to be accurate, we must wonder wether low dimensional Wasserstein embeddings have a sufficient representational capacity. The authors tried to answer this question by comparing the Wasserstein embeddings with other common embedding techniques on different types of graphs.

Our report is structured as follows: in Section 2, we give a theoretical insight on Wasserstein spaces and explicit the algorithm to approximate the earth mover's distance. We also show in this section that the problem of matching the semantic structure of the data can be casted as a differntiable minimzation problem.
In Section 3, we investigate with several experiments the representational capacity of Wasserstein spaces. The first experiment compares the equilateral dimension of finite Wasserstein spaces with the euclidean equilateral dimension. The second experiment is a reproduction of the experiment on graphs ran by [Frogner et al., 2019]. Finally, the last experiment shows the visual representation obtained with Wasserstein embeddings on well-clustered time series.

2 Embedding in Wasserstein spaces

Let $\mathcal{X} = (E, d)$ be a metric space and let P_p be the collection of all probability measures on E with finite p^{th} moment in E. the $p-$Wasserstein distance between two probability measures $\mu, \nu \in P_p$ is then defined as:

$$W_p(\mu, \nu) = \Big[\inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{X}} d(x_1, x_2)^p d\pi(x_1, x_2) \Big]^{1/p}$$

Where $\Pi(\mu, \nu)$ is the set of probability measures on $E \times E$ which first marinal is μ and second is ν. There one can define the Wasserstein metric space (P_p, W_p).

The idea here is to embed a set of data $X = (x_i)_{i=1}^n$ into a Wasserstein space accordingly to some relationship $r(.,.)$ between the x_i's. Therefore we want to learn a function ϕ mapping the x_i's into (P_p, W_p) such that $\forall i, j \in \{1, \dots, n\}, r(x_i, x_j) \simeq W_p(\phi(x_i), \phi(x_j))$. The quality of an embedding lies in how close $W_p(\phi(x_i), \phi(x_j))$ is to $r(x_i, x_j)$.

We are going to work with discrete distributions with finite support over $\mathbb{R}^k$. distributions will then be in the form $\mu = \sum_{i=1}^{N(\mu)} w_i \delta_{x^i}$ where $\mathbf{x} = (x^1, \dots, x^{N(\mu)}) \in \mathbb{R}^{k \times N(\mu)}$ are the support points and $w = (w^1, \dots, w^{N(\mu)}) \in \mathbb{R}^{N(\mu)}$ are the weight corresponding to those points which are summing to 1.

For two discrete measures $\mu = \sum_{i=1}^N w^i \delta_{x^i}$ and $\nu = \sum_{i=1}^M w'^i \delta_{y^i}$, let $D \in \mathbb{R}^{N \times M}$ be the matrix of pairwise distances between elements ok $\mathbf{x}$ and $\mathbf{y}$ where $D_{i,j} = d(x^i, y^j)$.

Computing the Wasserstein distance between two distributions is hard in general. In the previous configuration the optimization problem becomes:

$$W_p(\mu, \nu)^p = \min_{T \geq 0} \operatorname{tr}(D^p T)$$
$$\text{s.t } T\mathbf{1} = \mathbf{x}_\mu,\ T\mathbf{1} = \mathbf{x}_\nu$$

With D^p taken elementwise.

This equation can be approximately solved by solving the corresponding Sinkhorn divergence optimization problem:

$$W_p^\lambda(\mu, \nu)^p = \min_{T \geq 0} \operatorname{tr}(D^p T) + \lambda(T(\log(T) - \mathbf{1}\mathbf{1}^\top)^\top)$$
$$\text{s.t } T\mathbf{1} = \mathbf{x}_\mu,\ T^\top\mathbf{1} = \mathbf{x}_\nu$$

This last problem can be very efficiently solved with matrix balancincing. The following procedure converges to the Sinkhorn divergence:

As for $\lambda > 0$ the optimal solution take the form $T* = \text{Diag}(r)\exp(-Dp/\lambda)\text{Diag}(c)$ where r, c are vectors, one can optimize over r and c by alternately updating them:

$$r \longrightarrow u/Kc \qquad c \longleftarrow v/Kc$$

Where $K = \exp(-D^p/\lambda)$ and the division is applied elementwise. More details on this method can be found in [Cuturi, 2013].

We will consider only uniformly weighted distributions as the authors claim that allowing the weights to vary does not improve the asymptotic approximation error.

Ultimately, given $X \in E^n$ and $r(.,.)$ we want to learn a function $\phi : E \mapsto (P_p(E), W_p)$ that minimizes a certain criterion:

$$\phi^* = \text{Argmin} \frac{1}{N} \sum_{i=1}^N \mathcal{L}\Big(W_p(\phi(x_i), \phi(y_i)), r(x_i, y_i)\Big)$$

Where N is the total number of pairs (x, y) to consider.
The metrics used in each case will be specified later.

In all the experiments, the aim is to minimize the mean quadratic distortion of the embeddings:

$$\phi^* = \text{Argmin} \frac{1}{\binom{n}{2}} \sum_{i<j} \frac{(W_1^\lambda(\phi(x_i), \phi(x_j)) - d(x_i, x_j))^2}{d(x_i, x_j)}$$

Even though there are no closed form solutions to compute this minimum, one can remark that the mean quadratic distortion is differentiable with respect to the embbedings $(\phi(x_i))_{1 \leq i \leq n}$. Therefore, we can perform gradient descent to approximate this minimum. In all the experiments, we used an ADAM optimizer.

3 Experiments

3.1 Experimental equilateral dimension

Objective of the experiment In this section, we discuss an experiment aiming to compare the *equilateral dimension* between (a subset of) Wasserstein spaces and a euclidian space with the same dimension.

The equilateral dimension of a metric space is the maximum number of points that are all at equal distances from each other. This notion can be useful when the embedding aims to preserve some sort of semantic similiraty (by opposition to the case when we want the embedding to be an isometry between two metric spaces): in the case of words, one may wish for the words "dog" "cat" and "fish" to have the same level of similarity.

Lemma 1. *For $n \geq 1$ an integer, the equilateral dimension of $\mathbb{R}^n$ endowed with the euclidean norm $||.||_2$ is $n+1$.*

To the best of our knowledge, the equilateral dimension of finite Wassestein spaces (with a finite number of dirac masses) were not derived yet. We tried to recover experimentally the equilateral dimension of $\mathcal{W}_1^\lambda(\mathbb{R}^2)$ with finite distributions. we detail in the next paragraph our protocol.

Protocol We denote by $\mathcal{W}_1^\lambda(\mathbb{R}^2)(d)$ the subset of distributions with d equally weighted support points in $\mathbb{R}^2$ endowed the λ-Sinkhorn divergence.‘

For several values of d, we tried to build as many equilateral distributions in $\mathcal{W}_1^\lambda(\mathbb{R}^2)(d)$ as possible. We chose the equilateral distance to be 1. Formally, if we call ϕ_i the i-th embedding, we would like to recover the highest value of n such that the infimum on the ϕ_i of the Mean Quadratic Distorsion ($MQD(n)$):

$$MQD(n) = \frac{2}{n(n-1)} \sum_{1 \leq i < j \leq n} (\mathcal{W}_\lambda^1(\phi_i, \phi_j) - 1)^2$$

is zero.

However, since the argmin is approximated with a gradient descent, we can never reach an exactly null distorsion. Therefore, in order to recover the experimental equilateral dimension $\hat{n}_{equi}^d$, we plotted the evolution of the logarithmic empirical minimum distorsion for increasing values of n. We assigned to $\hat{n}_{equi}^d$ the value of n for which we observed a break in the slope. We acknowledge that this criteria is purely qualtitative and has low theoretical foundations. In particular, the empirical minimal distorsion structurally increases with n.

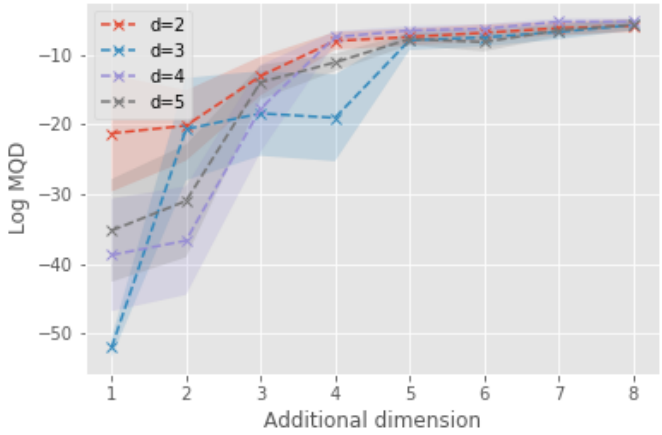


Figure 1: Evolution the Mean Quadratic Distorsion

However, since the true minimimas should go from exactly 0 to strictly positive, we thought it was plausible to observe two distinct slopes.

Experimental results The Figure 1 shows the evolution of the logarithmic minimal distorsion for several values of d. We averaged our results on twenty runs. The ordinate axis repesents the 'additional dimension': for $\mathcal{W}_1^\lambda(\mathbb{R}^2)(d)$ we define the additional dimension k as $2d + k$. Indeed, the total dimension of $\mathcal{W}_1^\lambda(\mathbb{R}^2)(d)$ is $2 \times d$. Therefore computing the logarithmic minimal distorsion for $n = 2d+k$ allows a direct simultaneous comparision with the euclidean case.
The curves suggest that the breaking point is attained for $\hat{n}^d_{equi} \approx 2d + 4$. We point out that for lower values of n, the upper bound of empirical minmimal MQDs are lower than 10^{-10}. Compared to the euclidean equilateral dimension, it is a gain of three units.

Even though this result is not spectacular in itself, we want to point out that in their experiment on graphs, [Frogner et al., 2019] compared the distorsion induced by different type of embeddings for a very specific metric. Hence, it doesn't provide a global result on the "embedding power" of wasserstein spaces vs euclidean spaces. While in our case, the equilateral dimension is a criteria instrinsic to the spaces themselves.

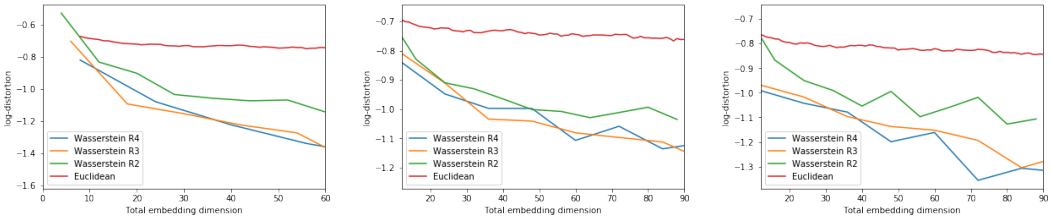
3.2 Comparison on Graph Embeddings

Following the experiments mentioned by the authors, we implemented the Wasserstein embedding procedure on graphs. In details we embed four types of graphs in Wasserstein spaces over $\mathbb{R}^2$, $\mathbb{R}^3$ and $\mathbb{R}^4$ supported distributions. As we want to compare the embedding sharpness of Wasserstein spaces to Euclidean spaces, we are comparing them for identical number of degrees of freedom, i.e if we embed one observation in an $\mathbb{R}^k$ Wasserstein space in which distributions have d support points, we compare the corresponding embedding to an Euclidean embedding in $\mathbb{R}^{kd}$. The dimension of the Euclidean space will then be called total embedding dimension.

Four different graph generation models are used, the following details the parameters used for those models:

- Barbasi-Albert: $n = 40$ nodes
- Watts-Strogatz: $n = 40$ nodes, mean degree: $k = 3$
- Stochastic Block Model: $n = 40$ nodes, each partition of the graph is generated randomly to sum at 40, $\mathbf{P}\mathbf{P}^\top / \max(P_{i,j}) \in \mathbb{R}^{n \times n}$ the probability matrix, with $\mathbf{P}_{i,j} \sim \mathcal{U}([0, 1])$

For each set of parameters (model, embedding space, total embedding dimension) we draw 10 different graphs and average the criterion over the runs on those graphs. The Wasserstein and the Euclidean embeddings are both optimized accordingly to the same criterion: the distortion defined above and use the same convergence criterion. The confidence intervals of level .9 for a point x of the graph are $[x - .04, x + .04]$



Left to right: Comparison on Barbasi-Albert graphs, Comparison on Watts-Strogatz graphs, Comparison on Stochastic Block Model graphs

In all of the experiments, Wasserstein embeddings outperform the Euclidean ones. Surprisingly the Euclidean embeddings on Stochastic Block Model generated graphs are not as close to the Wasserstein ones as they are in the original paper. This may come from computational limitations, indeed the Euclidean embeddings converge very slowly compared to the Wasserstein ones.

Those experiments are confirming the fact that $\mathbb{R}^3$ is nearly as good as an embedding ground space as $\mathbb{R}^4$ and therefore may be sufficient in many cases.

Ultimately Wasserstein embeddings seem to be well adapted for many different situations, their performance being better than Euclidean embeddings on all of the graph structures presented here, the fact that they are more flexible is confirmed.

3.3 Time series embedding and visualization

Setting In this last experiment, we embedded time series from the Weekly Sales Transaction dataset. Each time serie corresponds to the weekly purchased quantitiy of a particular product. We limited ourselves to $n = 40$ time series for computational reasons mainly.

We chose the Dynamic Time Wrapping distance [Silva and Batista,] as the similarity measure between the time series. Therefore, our emmbeddings $(\phi_i)_{1 \leq i \leq n}$ aimed to minimize the following quadratic distorsion:

$$MQD = \frac{2}{n(n-1)} \sum_{1 \leq i < j \leq n} \frac{(\mathcal{W}_\lambda^1(\phi_i, \phi_j) - \mathrm{dtw}(t_i, t_j))^2}{\mathrm{dtw}(t_i, t_j)}$$

with t_i the i-th time serie and $\text{dtw}(t_i, t_j)$ the dynamic time wrapping distance between the series t_i and t_j.

In this experiment, since we wanted to visualize directly the embeddings, we chose $\mathbb{R}^2$ to be our ground space. Also for computational issues, we chose embeddings with $d = 30$ support points.

Objective In this experiment, we wanted to demonstrate the ability of wasserstein embeddings to accurately represent complex objects in a low dimension space. In particular, if the time series had some kind of pattern, we wanted to see how it translated in a 2D point cloud of Wasserstein embeddings.

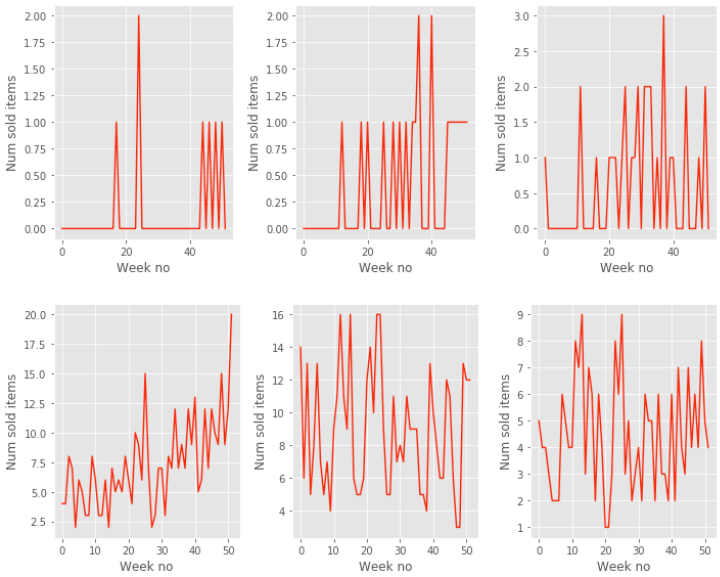


Figure 2: Clusters found by DBSCAN

To find the time series with a similar pattern, we used the DBSCAN [Ester et al., 1996] algorithm. This algorithm is an unsupervised clustering algorithm which does not know *a priori* the number of clusters. Given a matrix of pairwise distances, it returns the clustered data-points.

We applied the DBSCAN algorithm to our 40 time series and it returned 2 clusters. The Figure 2 shows a sample of time series for each cluster. In the first cluster, the series exhibit a very similar pattern: they alternate between plateau and spikes. In the second clusters, the time series look more "random" and don't seem to express a particular pattern. In the rest of the report, we will identify the series of the first cluster as "flat/spiked" and the series of the second cluster as "regular".

Visual results Once we had embedded our time series, we merged the distributions belonging to the same cluster. Therefore we obtained two discrete distributions. The Figure 3 shows both the raw distributions and the 2D densities smoothed with a gaussian kernel [Parzen, 1962].

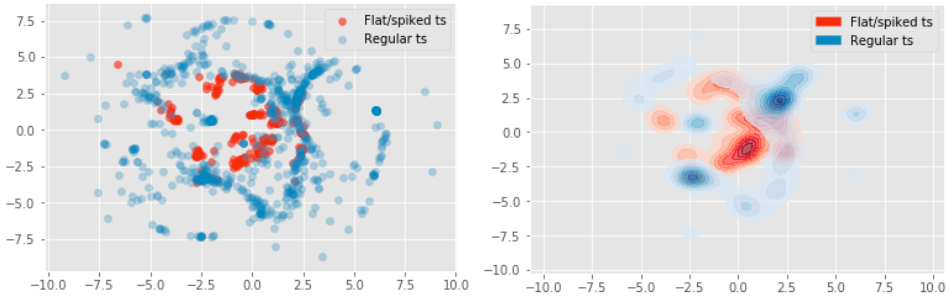


Figure 3: Raw and merged distributions

The visual result is quite satisfactory: the embeddings of flat spiked time series are concentrated while those of regular ones are diffuse. Indeed, the flat spiked time series share a strong similar pattern, therefore they should be close to each other.

4 Conclusion

First we detailed the theory underlying those ideas. We then built an experiment to visualize the better representation capacity of those embeddings. Another experiment aimed at reproducing the results of the original paper by comparing Wasserstein embeddings to Euclidean ones on different graph structures and different dimensionality scales. Finally we applied the embedding procedure to a time series data set and interpreted the enhanced visualization the embeddings provided.
The idea developped in [Frogner et al., 2019] is to try and embed variously structured data into Wasserstein spaces using the fast Sinkhorn divegence comutations and shaping the embeddings according to a distorsion criterion. The main point of proposing Wasserstein spaces for embeddings lies in their increased representation capacities compared to Euclidean embeddings amongst various others as mentionned in [Frogner et al., 2019]. This flexibility gain allows one to embed high dimensional structures in low dimentional spaces such as $\mathbb{R}^2$ or $\mathbb{R}^4$ with very low distorsion, which permits an increased vizualisation power.

References

[Cuturi, 2013] Cuturi, M. (2013). Sinkhorn distances: Lightspeed computation of optimal transport. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 2292–2300, USA. Curran Associates Inc.

[Ester et al., 1996] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pages 226–231. AAAI Press.

[Frogner et al., 2019] Frogner, C., Mirzazadeh, F., and Solomon, J. (2019). Learning entropic wasserstein embeddings. In *International Conference on Learning Representations*.

[Parzen, 1962] Parzen, E. (1962). On estimation of a probability density function and mode. *Ann. Math. Statist.*, 33(3):1065–1076.

[Silva and Batista,] Silva, D. F. and Batista, G. E. A. P. A. *Speeding Up All-Pairwise Dynamic Time Warping Matrix Calculation*.

Appendix

Proof of Lemma 3.1

First we are going to show that for v_1, v_2, v_3 three distinct equidistant vectors at distance d from each other, $< v_1 - v_2, v_1 - v_3 >= d^2/2$ where $<,>$ is the standard euclidean dot product.

$$\begin{aligned} ||v_2 - v_3||^2 &= ||v_2 - v_1 + v_1 - v_3||^2 \\ &= ||v_2 - v_1||^2 - 2 < v_1 - v_2, v_1 - v_3 > + ||v_1 - v_3||^2 \\ &= 2d^2 - 2 < v_1 - v_2, v_1 - v_3 > \end{aligned}$$

Since $||v_2 - v_3||^2 = d^2$, we conclude that $< v_1 - v_2, v_1 - v_3 >= \frac{d^2}{2}$.

With this small result, we will show that for $v_1, \cdots v_p$ $(p \leq n+1)$ p equidistant vectors, $(v_1 - v_2, \cdots, v_1 - v_p)$ are linearly independent.

Be $(\alpha_2, \cdots, \alpha_p)p$ real numbers. Suppose that $\sum_{i=2}^{p} \alpha_i (v_1 - v_i) = 0$. Taking the scale product with $v_1 - v_j$ for some $2 \leq j \leq p$ on both sides we have:

$$\alpha_j d^2 + \frac{d^2}{2} \sum_{i \neq j} \alpha_i = 0$$

$$\iff \alpha_j + \sum_{i=2}^{p} \alpha_i = 0$$

Summing the last equality on $j \geq 2$, we have that $\sum_{i=2}^{p} \alpha_i = 0$. Now, since we have for $2 \leq j \leq p$: $\alpha_j + \sum_{i=2}^{p} \alpha_i = 0$, we can deduce $\alpha_j = 0$.