Unsupervised Ground Metric Learning with Tree Wasserstein Distance

Kira Düsterwald [1 2] **Makoto Yamada** [2]

Abstract

Optimal transport (OT) is a powerful geometric machine learning tool for comparing distances between samples. Accurate OT distances rely on the underlying distance between dataset features, or ground metric. Ground metrics are commonly decided with heuristics or learned with supervised algorithms. However, since many interesting datasets are unlabelled, unsupervised ground metric learning approaches have recently been introduced. One promising option employs Wasserstein singular vectors (WSV), which emerge when computing OT distances between features and samples simultaneously. WSV is effective, but computationally expensive ($\mathcal{O}(n^5)$ complexity). Here, we propose to augment this method by embedding samples and features on trees, on which we compute the tree Wasserstein distance (TWD). We demonstrate theoretically and in practice that the algorithm converges to a better approximation of the full WSV approach than entropy regularisation, with faster (cubic) computational efficiency. In addition, we show that the initial tree structure can be chosen flexibly, since tree geometry does not constrain the solution up to the number of edge weights. These results poise unsupervised ground metric learning with TWD as a low-rank approximation of WSV with the potential for widespread low-compute application.

1. Introduction

The geometric nature of data structures can help to solve unsupervised learning tasks. For example, clustering (e.g. k-means) compares groupings based on a distance between samples. Inherently, the heuristic that one chooses for the distance matrix or function determines the outcome of the algorithm. So, how should one choose or learn this distance?

From an optimal transport perspective, the optimal sample distance is the Wasserstein distance: that is, the minimum cost from the best mapping between samples, weighted by the cost, or *ground metric*, between their features. For example, samples might be documents, and features words. Then the distance/cost between two documents is determined by the relative expression of words in each, mappings between them, and the ground metric between words. This turns the problem of learning sample distances into one of learning – or choosing – the (feature) *ground metric*.

State-of-the-art methods usually employ heuristics for the ground metric, commonly Euclidean, with embeddings for the features, e.g. word2vec for documents (Kusner et al., 2015), and gene2vec in bioinformatics (cell identification from gene expression) (Zou et al., 2019; Du et al., 2019). However, embeddings are not always available or may be difficult to learn. We are interested in unsupervised ways to learn the ground metric.

We do so by augmenting an iterative method by Huizing et al. (2022): they suggest learning distances on *samples*, which then act as the ground metric to learn (ground) distances on *features*, which then are the ground metric and improve the metric on samples, and so on.

To illustrate the idea, consider the *words* "opera" and "Vienna", which one might expect to be close in word-space. If the metric between *documents* in the corpus that contain both of these words is close, we learn a smaller distance in the word dictionary between "opera" and "Vienna". This in turn decreases the distance between documents containing these words. Such an approach could be applied to gene expression (features) in cells (samples), V1 neuronal activity (features) corresponding to representation of images (samples), and other unlabelled high-dimensional data.

While effective, Huizing et al. (2022)'s algorithm is expensive to run as it requires multiple pairwise distance computations per iteration. Inspired by Yamada et al. (2022), we propose a method that reduces algorithmic complexity by at least a quadratic factor by representing the data on a tree and learning weights on tree edges, rather than the full pairwise distance matrix. Our method offers a geometric low-rank ap-

[1]Gatsby Computational Neuroscience Unit, University College London, United Kingdom [2]Machine Learning and Data Science Unit, Okinawa Institute of Science and Technology, Japan. Correspondence to: Kira Düsterwald <kira.dusterwald.21@ucl.ac.uk>.

Proceedings of the Geometry-grounded Representation Learning and Generative Modeling at 41st International Conference on Machine Learning, Vienna, Austria. PMLR Vol Number, 2024.

proximation interpretation of the full unsupervised ground metric learning problem. We show that it performs well on a toy dataset.

1.1. Previous and related work

In this section we introduce optimal transport distances, tree Wasserstein distances, inverse optimal transport and unsupervised ground metric learning.

Optimal transport and optimal transport distances: Optimal transport can be thought of as a "natural geometry" for probability measures (Peyré & Cuturi, 2019). The classical OT problem – attributed to Monge (1781) and Kantorovich (1940) –asks how to find an optimal transport plan between two probability distributions subject to some cost.

Consider the empiric discrete OT problem. Let $\mu = \sum_i^n a_i \delta_{x_i}$ and $\nu = \sum_j^n b_j \delta_{x_j}$ be discrete probability distributions ("histograms": normalised bin counts), with δ_{x_i} the Dirac delta function at position x_i. Then the OT problem is to find optimal P satisfying:

$$\mathcal{W}_{1-C}(\mu,\nu) = \min_{P \in \mathbb{R}_+^{n\times n}} \langle P, C\rangle = \min_{P \in \mathbb{R}_+^{n\times n}} \sum_i \sum_j P_{ij} C_{ij} \quad (1)$$

where $P\mathbf{1}_n = a, P^\top \mathbf{1}_n = b$ and C is some cost function. $\mathcal{W}_{1-C}(\mu,\nu)$ is known as the (1)-Wasserstein distance. In this paper, we shall drop the 1 and assume that $\mathcal{W}_C$ includes the pairwise distance matrix C as its ground metric.

For example, in document similarity, x_i, x_j are words (or embedding vectors) and a, b tell us the frequency of words in a given document μ (or ν). Usually, C is assumed to be the Euclidean distance between word embeddings, and then $\mathcal{W}_C(\mu,\nu)$ is the word mover's distance (Kusner et al., 2015). Here, we instead take the approach of *learning* C in an efficient manner, rather than utilising pre-trained embeddings, for documents represented as bag-of-words.

Computationally efficient OT: Solving the OT problem to compute $\mathcal{W}_C$ via linear programming has complexity $\mathcal{O}(n^3 \log n)$. Cuturi (2013) suggested using the celebrated Sinkhorn's algorithm, which speeds up the calculation through entropic-regularised OT to $\mathcal{O}(n^2)$. Another more recent formulation is via tree Wasserstein distance, which has linear complexity (Le et al., 2019).

Tree Wasserstein distance (TWD): The tree Wasserstein distance represents samples on a tree. Another geometric embedding is the sliced Wasserstein distance (SWD), in which $\mathcal{W}_C$ is computed via projection to a one-dimensional subspace, with complexity $\mathcal{O}(n \log n)$ (Kolouri et al., 2016). SWD is a special case of TWD (Le et al., 2019).

Consider a tree $\mathcal{T}$ with $d_\mathcal{T}$ the tree metric (the unique and shortest path between any two nodes on $\mathcal{T}$). Let $\mathbf{w}$ be the set of weights between nodes and $\mathbf{Z}$ the tree parameter ($\mathbf{Z}_{ij} = 1$ if node $i = Parent(j)$, 0 otherwise). Given $d_\mathcal{T}(\mathbf{x}, \mathbf{y})$, the TWD can be written (Le et al., 2019; Yamada et al., 2022): $\mathcal{W}_\mathcal{T}(\mu,\nu) = \inf_\pi \int_{\mathcal{X}\times\mathcal{Y}} d_\mathcal{T}(\mathbf{x},\mathbf{y}) d\pi(\mathbf{x},\mathbf{y})$.

Furthermore, TWD takes on a closed analytical form (Takezawa et al., 2021):

$$\mathcal{W}_\mathcal{T}(\mu,\nu) = ||\text{diag}(\mathbf{w})\mathbf{Z}(\mathbf{a}-\mathbf{b})||_1 \quad (2)$$

From (2), TWD can be computed in $\mathcal{O}(\text{size}(\mathbf{w}))$ (Takezawa et al., 2021). TWD is a good approximation of the full 1-Wasserstein distance (Yamada et al., 2022).

The inverse OT problem and ground metric learning: Leveraging the geometric nature of the Wasserstein/OT distance for unsupervised learning tasks relies on having a good idea of the ground metric between features that is then "lifted" to the OT distance between samples. Heuristics for ground metrics can be useful, especially with embeddings (Kusner et al., 2015; Du et al., 2019), but we are interested in broader, principled ways to find ground metrics.

Inverse optimal transport provides one solution: given a transport plan P, find the distance matrix C. For matching problems based on recommendation systems, for example the marriage dataset where preferred pairings and the associated features (i.e. the transport plan) are known, inverse OT can find the underlying distance between features. While in general this problem is underconstrained, solutions exist given sufficient constraints on C (Paty & Cuturi, 2020; Li et al., 2019; Stuart & Wolfram, 2019). However, inverse OT requires access to the full transport plan – in practice and for high-dimensional features, such as gene-cell data, that is not feasible.

An alternative is to use *partial* information about distance or similarity to learn the ground metric, for example with supervised or semi-supervised learning (Cuturi & Avis, 2014). A related but distinct concept is the idea of supervised learning of *sample* Wasserstein distances (Huang et al., 2016) and supervised TWD (Takezawa et al., 2021): here, the (feature) ground metric is still assumed (Euclidean), but information about distances is used to learn metrics between samples. The unsupervised approach differs in that neither a feature nor a sample ground metric needs to be assumed.

Unsupervised ground metric learning with Wasserstein singular vectors (WSV): Unsupervised techniques harness the relationship between the geometry of features (embedded in samples) and the geometry of samples (embedded in features) (Paty & Cuturi, 2020; Huizing et al., 2022).

Consider some data matrix $X \in \mathbb{R}_+^{n\times m}$. Let $a_i \in \mathbb{R}_+^m$ be a sample (normalised row) of X, and $b_k \in \mathbb{R}_+^n$ be a feature (normalised column). These are just histograms "embedding" a sample as a distribution over features, and vice versa. For example, the a_i (of which there are n) could

be a histogram of words in a document, and the b_k (of which there are m) documents containing a given word. As another example, a_i might represent cells (samples) as histograms over gene expression, where b_k are the genes (features) as histograms over cells that contain each gene.

Huizing et al. (2022) learn ground metric matrices $\mathbb{A} \in \mathbb{R}^{m\times m}, \mathbb{B} \in \mathbb{R}^{n\times n}$ satisfying the fixed point equations:

$$\mathbb{A}_{k,l} = \frac{1}{\lambda}\mathcal{W}_{\mathbb{B}}(b_k, b_l); \mathbb{B}_{i,j} = \frac{1}{\mu}\mathcal{W}_{\mathbb{A}}(a_i, a_j) \tag{3}$$

$\forall k, l \in \{1, .., m\}, i, j \in \{1, ..., n\}$, where $\mathcal{W}_d$ is the 1-Wasserstein distance with ground metric d.

Power iterations are used to compute these matrices, $\Phi_A(\mathbb{A})_{i,j} := W_{\mathbb{A}}(a_i, a_j) + \tau||A||_\infty R(a_i - a_j)$ (and similarly for $\mathbb{B}$). Then the equivalent problem is to find Wasserstein singular vectors (WSV) such that
$\exists(\lambda, \mu) \in (\mathbb{R}^*_+)^2$ s.t. $\Phi_B(\mathbb{B}) = \lambda\mathbb{A}, \Phi_A(\mathbb{A}) = \mu\mathbb{B}$.

The complexity of a single power iteration is $\mathcal{O}(n^2m^2(n\log(n) + m\log(m)))$ (since it requires computing m^2, n^2 Wasserstein distances), which Huizing et al. (2022) propose to reduce via stochastic optimisation and entropic regularisation, the latter improving performance to at best $\mathcal{O}(n^2m^2)$.

1.2. Contributions

We propose to improve on the WSV computational bottleneck by embedding the dataset in trees and approximating $\mathcal{W}_d$ with the TWD. We show that:

- We can learn a set of non-zero tree edge weights via a system of linear equations (Algorithm 1).
- This system is determined by a tree-parameter-based tensor whose rank is equal to the number of edges in the tree or one less the number of edges.
- The TWD singular vectors approach is more computationally efficient and outperforms entropic regularisation in terms of accuracy, as compared to WSV.

This positions our method as a low-rank approximation of the full WSV approach with geometric interpretation, advantages for scaling and low-resource efficiency gains.

2. Unsupervised tree Wasserstein distance ground metric learning

We propose to use the TWD (i.e. distance between leaves) to learn the tree ground metric (i.e. distance between features, embedded in the tree structure).

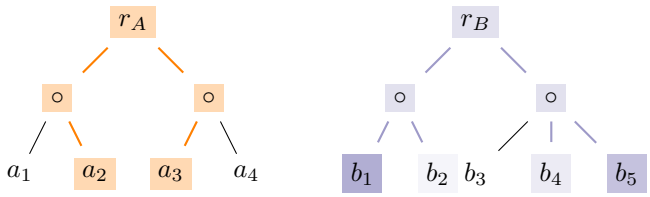

Figure 1. Tree embeddings for samples A as leaves in $\mathcal{T}_A$ (left) and features B as leaves in $\mathcal{T}_B$ (right). The tree metric $d_{\mathcal{T}_A}(a_2, a_3)$ is shown as the shortest path between these leaves in orange. Equivalently, we can use the relative expression of the features $\{b_1, b_2, b_3, b_4, b_5\}$ that represent a_2, a_3 respectively (as shown in different hues of blue) to compute a TWD, $\mathcal{W}_{\mathcal{T}B}(a_2, a_3)$, in $\mathcal{T}_B$.

2.1. Set-up and intuition

Let $A = \{a_i\}, B = \{b_k\}, i \in \{1, ..., n\}, k \in \{1, ..., m\}$ – the set of rows (samples) and set of columns (features) respectively – of the data matrix $X \in \mathbb{R}^{n\times m}_+$ be embedded in two trees $\mathcal{T}_A$ and $\mathcal{T}_B$.

Specifically, let the vectors in $A = \{a_1, ..., a_n\}$ be the leaves of $\mathcal{T}_A$, a_i defined as for WSV, and let $\mathcal{T}_B$ have leaves the vectors in $B = \{b_1, ..., b_m\}$ (Fig. 2.1). Let Z_A, Z_B, $Z \in \{0, 1\}^{N\times N_{\text{leaf}}}$ be the tree parameters for each tree, where $[Z]_{i,j} = 1$ when node i is the parent of node j or itself ($i = j$), and otherwise is 0. For $\mathcal{T}_A$, $N_{\text{leaf}} = n$, and for $\mathcal{T}_B$, $N_{\text{leaf}} = m$. Let $\mathbf{w}_A \in \mathbb{R}^{N-1}_+, \mathbf{w}_B \in \mathbb{R}^{M-1}_+$ be the vectors of edge weights (size one less than the number of nodes in each tree, N, M respectively).

Proposition 2.1. *The WSV fixed point equations* (3) *can be expressed on the trees $\mathcal{T}_A, \mathcal{T}_B$ as:*

$$d_{\mathcal{T}_A}(\mathbf{a_i}, \mathbf{a_j}) = \frac{1}{\lambda}\mathcal{W}_B(\mathbf{a_i}, \mathbf{a_j}); d_{\mathcal{T}_B}(\mathbf{b_k}, \mathbf{b_l}) = \frac{1}{\mu}\mathcal{W}_A(\mathbf{b_k}, \mathbf{b_l})$$

where the singular vector update is to find $\mathbf{w_A}$ (and symmetrically $\mathbf{w_B}$) such that $\forall i, j$

$$\lambda_A \mathbf{w_A}^\top \left(\mathbf{z_i^{(A)}} + \mathbf{z_j^{(A)}} - 2\mathbf{z_i^{(A)}} \cdot \mathbf{z_j^{(A)}}\right) = \left|\left| diag(\mathbf{w_B})\mathbf{Z^{(B)}}(a_i - a_j)\right|\right|_1 \tag{4}$$

Proof: *Appendix A* □

For intuition about Proposition 2.1, choose two samples a_i, a_j that are leaves in one of the trees, WLOG $\mathcal{T}_A$ (Fig. 2.1). The tree metric $d_{\mathcal{T}}$ is the shortest path between these leaves with edges in $\mathcal{T}_A$. Each sample can also be expressed as some histogram over features, which are leaves on $\mathcal{T}_B$. The 1-Wasserstein distance between the two samples is some cost matrix B applied to this expression, that is $\min_P \sum_k \sum_l P_{kl}C_{kl}$ where $P\mathbf{1}_n = a_i$ and $P^\top \mathbf{1}_n = a_j$. From Yamada et al. (2022), $\mathcal{W}_{1-B}(a_i, a_j)$ is equivalent to the TWD between a_i, a_j on $\mathcal{T}_B$, which we can express through summing proportions of $d_{\mathcal{T}_B}$ between the leaves in $\mathcal{T}_B$: $d_{\mathcal{T}_A}(a_2, a_3) = \mathcal{W}_{\mathcal{T}B}(a_2, a_3)$.

2.2. Solutions to the tree WSV problem exist

We can utilise Proposition 2.1 to learn low-rank approximations of $\mathcal{W}_B(\mathbf{a_i}, \mathbf{a_j})$ and $\mathcal{W}_A(\mathbf{b_k}, \mathbf{b_l})$.

Let $y_{ij} = \mathbf{z_i^{(A)}} + \mathbf{z_j^{(A)}} - 2\mathbf{z_i^{(A)}}\mathbf{z_j^{(A)}}$ and $\mathbf{Y}^{(A)}$ be the tensor of all y_{ij}s. Equation 4 presents a system of n^2 linear equations, since each y_{ij} is dimension N and $\mathbf{Y^{(A)}}$ is a tensor of dimension $n \times n \times N$. The system is over-determined since $N << n^2$ from $\max(N) = 2n - 1 < n^2$. Since Proposition 2.1 is still a singular vector problem, it can be solved with power iterations, and satisfies the convergence guarantees in Huizing et al. (2022), if solutions exist.

For non-negative solutions for all components of $\mathbf{w}$ to exist, the matrix multiplier in the system of linear equations needs to have rank at least equal to that of $\mathbf{w_A}$. The matrix multiplier is the tensor $\mathbf{Y}$ unravelled along the $n \times n$ dimensions to create a matrix Y' of size $n^2 \times N$. That we can choose trees for which the rank of Y' is $N - 1$ is proved in Theorem 2.2.

Theorem 2.2. *Let the number of nodes in a tree, including the root and leaves, be N. Then there can be at most one non-leaf node of a tree of degree 2. In addition, if such a node exists, it is the root and Y' has rank $N - 2$. Otherwise, every degree of the tree is at least 3, and Y' has rank $N - 1$.*

Proof: *Appendix B* □

This theorem implies that as long as we restrict ourselves to trees whose nodes have degree all greater than 2, we can ensure non-zero solutions to (4) since the rank of Y' is guaranteed to be $N - 1$, i.e. the length of $\mathbf{w}$.

2.3. Computational complexity and speed-ups

The method detailed above provides a computationally efficient way of computing Wasserstein singular vectors, since each inner power iteration computes m^2 Wasserstein distances in $\mathcal{O}(n)$ rather than $\mathcal{O}(n^2)$. There are a few additional speed-ups possible, too.

First, note that we only need to construct trees once (without edge weights), using ClusterTree or QuadTree with the number of children k set as $k > 2$ (Le et al., 2019; Indyk & Thaper, 2003). We compute Z_A, Z_B once for each of A, B, and $\mathbf{Z^{(A)}}(b_k - b_l)\forall k, l$ and $\mathbf{Z^{(B)}}(a_i - a_j)\forall i, j$ just once.

Similarly, we only learn the $\mathbf{Y}$ tensors once with the tree structure. In fact, we do not need to learn the whole tensor: we just need a square sub-matrix of Y' of rank $N - 1$, from Theorem 2.2. Since $\mathbf{Y}$ is sparse, we employed `scipy` sparse methods to find this basis set, say $\mathbf{U}$. Sparse QR decomposition or singular value decomposition (SVD) can be used; in practice sparse SVD performed faster.

Algorithm 1 Unsupervised Ground Metric Learning with Trees

Input: dataset X, size $n \times m$
do once
$\mathbf{A} \leftarrow \{a_i = \mathbf{X}^{(i)}_{row} / \sum \mathbf{X}^{(i)}_{row}\}$
$\mathbf{B} \leftarrow \{b_k = \mathbf{X}^{(k)}_{col} / \sum \mathbf{X}^{(k)}_{col}\}$
$\mathbf{Z^{(B)}}, \mathbf{Z^{(A)}} \leftarrow \text{ClusterTree}(\mathbf{B}), \text{ClusterTree}(\mathbf{A})$
$\mathbf{Y^{(A)}} \leftarrow \text{tensor}\left\{y_{ij} = \mathbf{z_i^{(A)}} + \mathbf{z_j^{(A)}} - 2\mathbf{z_i^{(A)}}\mathbf{z_j^{(A)}}\right\}$
$\mathbf{Y^{(B)}} \leftarrow \text{tensor}\left\{y_{kl} = \mathbf{z_i^{(B)}} + \mathbf{z_j^{(B)}} - 2\mathbf{z_i^{(B)}}\mathbf{z_j^{(B)}}\right\}$
$\mathbf{U^{(A)}} \leftarrow \text{basis sparse SVD}\left[\text{upper triangular}(\mathbf{Y^{(A)}}))\right]$
$\mathbf{U^{(B)}} \leftarrow \text{basis sparse SVD}\left[\text{upper triangular}(\mathbf{Y^{(B)}}))\right]$
$\mathbf{Z_{diff}}^{(\mathbf{A_{kl}})} \leftarrow \mathbf{Z^{(A_{kl})}}(b_k - b_l)\forall k, l \in \mathbf{B}_{\text{basis indices}}$
$\mathbf{Z_{diff}}^{(\mathbf{B_{ij}})} \leftarrow \mathbf{Z^{(B)}}(a_i - a_j)\forall i, j \in \mathbf{A}_{\text{basis indices}}$
initialise
$\mathbf{w_A}, \mathbf{w_B} \leftarrow$ random vectors size $\mathbf{A}_{leaf}, \mathbf{B}_{leaf}$
repeat
  for $t = 1$ **to** num iterations **do**
    $\mathcal{W}_{\mathcal{T}}(\mathbf{A}) \leftarrow \left\{\left|\left|\text{diag}(\mathbf{w_A})\mathbf{Z_{diff}}^{(\mathbf{A_{kl}})}\right|\right|_1, \forall k, l\right\}$
    $\mathcal{W}_{\mathcal{T}}(\mathbf{B}) \leftarrow \left\{\left|\left|\text{diag}(\mathbf{w_B})\mathbf{Z_{diff}}^{(\mathbf{B_{ij}})}\right|\right|_1, \forall i, j\right\}$
    $\mathcal{W}_{\mathcal{T}}(\mathbf{A})_{norm} \leftarrow \mathcal{W}_{\mathcal{T}}(\mathbf{A}) / \max\left(\mathcal{W}_{\mathcal{T}}(\mathbf{A})\right)$
    $\mathcal{W}_{\mathcal{T}}(\mathbf{B})_{norm} \leftarrow \mathcal{W}_{\mathcal{T}}(\mathbf{B}) / \max\left(\mathcal{W}_{\mathcal{T}}(\mathbf{B})\right)$
    $\mathbf{w_A} \leftarrow \text{NNLS solver}\left[\mathbf{w_A}^\top \mathbf{U^{(A)}} = \mathcal{W}_{\mathcal{T}}(\mathbf{B})_{norm}\right]$
    $\mathbf{w_B} \leftarrow \text{NNLS solver}\left[\mathbf{w_B}^\top \mathbf{U^{(B)}} = \mathcal{W}_{\mathcal{T}}(\mathbf{A})_{norm}\right]$
  end for
until convergence: $\mathbf{w_B} = \mathbf{w_B}(\text{prev}), \mathbf{w_A} = \mathbf{w_A}(\text{prev})$

Thereafter, the goal is to iteratively learn $\mathbf{w_A}$ and $\mathbf{w_B}$:

$$\lambda_A \mathbf{w_A}^\top \underbrace{\left(\mathbf{z_i^{(A)}} + \mathbf{z_j^{(A)}} - 2\mathbf{z_i^{(A)}} \cdot \mathbf{z_j^{(A)}}\right)}_{\mathbf{y^{(B)}}_{ij} \in \mathbf{Y^{(B)}} \text{ - learn as sparse } U^{(B)}} = \left|\left|\text{diag}(\mathbf{w_B}) \underbrace{\mathbf{Z^{(B)}}(a_i - a_j)}_{\text{learn once}}\right|\right|_1$$

(and symmetrically for $\mathbf{w_B}$). We use the same basis set indices to compute the TWD on the right-hand side as appears in $\mathbf{U^{(B)}}$. A solution exists, and can be found with a linear systems solver. We used non-negative least squares (NNLS), which is efficient compared to other solvers since weights are non-negative by assumption.

Using NNLS to solve for the $\mathbf{w}$ vector as a system of $N - 1$ linear equations has cubic complexity. Each compute of a pairwise TWD is linear. This gives overall complexity per power iteration: $\mathcal{O}(N^3) < \mathcal{O}((2n - 1)^3) << \mathcal{O}(n^5)$.

We summarise the full, computationally efficient unsupervised ground metric learning with TWD algorithm in Algorithm 1.

3. Experimental results on toy datasets

We demonstrated the utility of our method on the synthetic dataset $\mathbf{X} \in \mathbb{R}^{n \times m}$, $n = 80, m = 60$, as employed in prior work (Huizing et al., 2022).

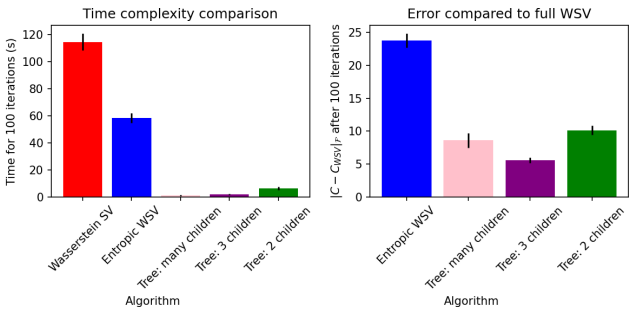

Figure 2. Comparison of mean time complexity (left) and Frobenius norm error (right) for different ground metric learning algorithms. Tree algorithms are specified by restrictions on the minimum number of children per node in the tree. In both, lower is better. Standard error of the mean is shown with black bars. "Many children" refers to ClusterTree initialised with 10 as the minimum children parameter, and 3, 2 with 3, 2 children respectively.

3.1. Dataset construction and experimental methods

The dataset uses various translations of a (unimodal) periodic function on 1-d torus (with boundary conditions), such that $\mathbf{X}_{ik} = \exp\left(-\left(\frac{i}{n} - \frac{k}{m}\right)^2/\sigma^2\right)$. On $\mathbf{X}$, the underlying (and learned) ground metrics look like $\left|\sin(\frac{i}{n} - \frac{j}{n})\right|$ (and symmetrically for m).

We ran experiments 3 times for 100 iterations each on CPU with different methods, using a translation of the periodic function, to which we added a 50% translation of the torus with half the magnitude. We set $\sigma = 0.01$. Thus: $\mathbf{X}_{ik} \propto \exp\left(-\left(\frac{i}{n} - \frac{k}{m}\right)^2/\sigma^2\right) + 0.5\left(-\left(\frac{i}{n} - \frac{k}{m} + 0.5\right)^2/\sigma^2\right)$.

In order to control against potential gains from tree construction based on preferred data ordering, we randomly permuted the dataset rows and columns before each experiment. We compare performance via computation time and the Frobenius norm of the difference between the learned cost and the metric uncovered by the relevant WSV algorithm (Fig. 2).

3.2. Results

All tree-based methods were faster than both standard WSV and entropic regularised WSV by 2-3 orders of magnitude (3-children trees took 1% of the time of the full WSV, for example), as predicted by our theoretical time complexity calculations. Interestingly, the tree-based methods had lower Frobenius norm error when compared to entropic WSV.

As expected from Theorem 2.2, the "many" children trees (at least 10 children) had lower ranks than both 2- and 3-trees, and 3-trees had lower ranks than 2-trees. In general, rank correlated with accuracy and time – that is, trees with fewer children per node computed distance matrices slower, but produced more accurate results (Fig. 2). One exception follows directly from Theorem 2.2: the 2-trees have rank $N - 2$ while the 3-trees are rank $N - 1$, so the full weight vector cannot be learned for all edges on 2-trees. As a result, the 2-trees performed worse in terms of accuracy that the 3-trees (Fig. 2 right).

4. Conclusions and future work

We present here a new, computationally efficient approach to learn ground metrics in an unsupervised manner by harnessing tree Wasserstein distance as an approximation of the 1-Wasserstein distance. The results are interesting both from an efficiency standpoint, and because Theorem 2.2 provides a new geometric underpinning of the restrictions needed for tree structures to flexibly represent data in low-rank approximations, which a toy model supports. Future work includes scaling the experiments to real-world datasets and exploring stochastic WSV (Huizing et al., 2022) against TWD singular vectors as an alternative low-rank approximation.

References

Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. pp. 2292–2300, 2013.

Cuturi, M. and Avis, D. Ground metric learning. *Journal of Machine Learning Research*, 15:533–564, 2014.

Du, J., Jia, P., Dai, Y., Tao, C., Zhao, Z., and Zhi, D. Gene2vec: Distributed representation of genes based on co-expression. *BMC Genomics*, 20: 7–15, 2 2019. ISSN 14712164. doi: 10.1186/S12864-018-5370-X/FIGURES/5. URL https://bmcgenomics.biomedcentral.com/articles/10.1186/s12864-018-5370-x.

Huang, G., Guo, C., Kusner, M. J., Sun, Y., Weinberger, K. Q., and Fei, S. Supervised word mover's distance. 2016.

Huizing, G.-J., Cantini, L., and Peyré, G. Unsupervised ground metric learning using wasserstein singular vectors. 2022.

Indyk, P. and Thaper, N. Fast image retrieval via embeddings. volume 2, pp. 5, 2003.

Kantorovich, L. A new method of solving some classes of extremal problems. *Doklady Akademii Nauk SSSR*, 28: 211–214, 1940.

Kolouri, S., Zou, Y., and Rohde, G. K. Sliced wasserstein kernels for probability distributions. 2016.

Kusner, M. J., Sun, Y., Kolkin, N. I., and Weinberger, K. Q. From word embeddings to document distances. 2015.

Le, T., Yamada, M., Fukumizu, K., and Cuturi, M. Tree-sliced variants of wasserstein distances. 2019. URL https://github.com/lttam/TreeWasserstein.

Li, R., Zhou, H., Zha, H., Ye, X., and Li, Z. Learning to match via inverse optimal transport. *Journal of Machine Learning Research*, 20:1–37, 2019.

Monge. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences de Paris, avec les Mémoires de Mathématique et de Physique pour la même année*, pp. 666–704, 1781.

Paty, F.-P. and Cuturi, M. Regularized optimal transport is ground cost adversarial. 2020.

Peyré, G. and Cuturi, M. Computational optimal transport. *Foundations and Trends in Machine Learning*, 11(5-6): 355–607, 2019.

Stuart, A. M. and Wolfram, M. T. Inverse optimal transport. *SIAM Journal on Applied Mathematics*, 80:599–619, 5 2019. ISSN 00361399. doi: 10.1137/19M1261122. URL https://arxiv.org/abs/1905.03950v1.

Takezawa, Y., Sato, R., and Yamada, M. Supervised tree-wasserstein distance. 2021.

Yamada, M., Takezawa, Y., Sato, R., Bao, H., Kozareva, Z., Ravi, S., Aip, R., Ai, M., and Ai, S. Approximating 1-wasserstein distance with trees. 2022.

Zou, Q., Xing, P., Wei, L., and Liu, B. Gene2vec: gene subsequence embedding for prediction of mammalian n 6-methyladenosine sites from mrna. *RNA*, 25:205–218, 2019. doi: 10.1261/rna. URL http://www.rnajournal.org/cgi/doi/10.1261/rna.

A. Proof of Proposition 2.1

First, note that for all discrete measures and distance metric d, we are guaranteed theoretically that there exists a tree for which $\mathcal{W}_1 = \mathcal{W}_\mathcal{T}$, and we can construct it by setting $d = d_\mathcal{T}$ (Le et al., 2019; Yamada et al., 2022).

Following Yamada et al. (2022), define a tree parameter-based vector for every a_i, a_j as $y^A_{ij} := z_i + z_j - 2z_i \circ z_j$ with $z_i, z_j \in Z_A$, which is known from the tree structure. The tree metric (distance between two leaves on $\mathcal{T}_A$) is given by:

$$d_{\mathcal{T}_A}(a_i, a_j) = w_A^\top y^A_{ij} \tag{5}$$

Note that any b_k is a distribution over leaves in $\mathcal{T}_A$ and vice versa. From these, we can derive Wasserstein distances between any b_k, b_l. TWD can approximate the 1-Wasserstein distance and admits the closed form in equation 2 (Yamada et al., 2022), so $\mathcal{W}_{\mathcal{T}_A}(b_k, b_l) = ||\text{diag}(w_A)Z_A(b_k - b_l)||_1$.

Now, let us assume that instead of learning the *ground* metric for the non-tree case, we want to learn a *tree* metric. The equivalent claim as in Huizing et al. (2022) is that we achieve the fixed points in Proposition 2.1.More formally, as before, we would like:

$$\lambda_B \mathbf{w_B}^\top \mathbf{Y^{(B)}} = \phi_A(\mathbf{w_A}); \lambda_A \mathbf{w_A}^\top \mathbf{Y^{(A)}} = \phi_B(\mathbf{w_B})$$

with $\lambda_A, \lambda_B, \tau \in \mathbb{R}_+$, where $\phi_A(\mathbf{W_A})_{ij} = \mathcal{W}_{\mathcal{T}_A}(a_i, a_j) + \tau||\mathbb{A}||_\infty R(a_i - a_j)$, and $\mathbf{Y}$ is the tensor composed of the n^2 (or m^2) y_{ij} vectors. So, Φs map ground costs to Wasserstein distance matrices.

Let $Z_A(b_k - b_l) = z^A_{kl}$. Since $w_A \in \mathbb{R}^N_+$ is positive, and letting $|.|$ denote element-wise absolute value,

$$\begin{aligned} W_{\mathcal{T}_A}(b_k, b_l) &= ||\text{diag}(w_A) z^A_{kl}||_1 \\ &= \sum_s^M w_A^{(s)} |z_{kl}^{A(s)}| = w_A^\top |z^A_{kl}| \end{aligned}$$

So from (2) and Proposition 2.1:

$$\lambda w_B^\top y^B_{kl} = w_A^\top |z^A_{kl}|, \mu w_A^\top y^A_{ij} = w_B^\top |z^B_{ij}| \tag{6}$$

$\forall k, l \in \{1, .., m\}, i, j \in \{1, .., n\}$.

In full, the singular vector update becomes to find $\mathbf{w_A}$ such that $\forall i, j$

$$\lambda_A \mathbf{w_A}^\top \left(\mathbf{z_i^{(A)}} + \mathbf{z_j^{(A)}} - 2\mathbf{z_i^{(A)}} \cdot \mathbf{z_j^{(A)}} \right) = \left|\left| \text{diag}(\mathbf{w_B}) \mathbf{Z^{(B)}} (a_i - a_j) \right|\right|_1 \tag{7}$$

(and symmetrically for the $\mathbf{w_B}$ iteration). □

B. Proof of Theorem 2.2

As preliminaries, we assert that $N > N_{leaf} = m$, and $\mathbf{Z} \in \{0, 1\}^{N \times l}$ has rank m. Note that $\mathbf{Y} \in \{0, 1\}^{N \times l \times l}$ is strictly a tensor, but in practice we only care for the reshaped matrix $\mathbf{Y} \in \{0, 1\}^{N \times l^2}$.

Assume that trees cannot have redundant nodes (i.e. nodes with exactly 1 child, not including themselves), and let every tree have root node r and leaf nodes $1, ..., l$, which we also use to index the associated positions along vectors z and y.

Note that at most one inner (non-leaf) node can have degree less than 3, and it must be the root. In this case, the root has degree 2. Otherwise all inner nodes are of degree ≥ 3.

Some bounds for N in terms of l can be derived from the observation that rank is upper-bounded by the number of pairs that two distinct leaves can make, noting $y_{i,j} = y_{j,i}$ and $y_{i,i} = 0$. For $l \geq 4$,

$$l^2 > \frac{l(l-1)}{2} \geq 2l - 2 \geq N - 1 \geq l \tag{8}$$

Replacing $N - 1$ for $N - 2$ we get bounds for $l \geq 2$. $2l - 2 \geq N - 1$ in (8) is derived from the upper bound on the number of nodes in the maximal spanning tree, a binary tree.

We prove by induction on the number of tree nodes that the rank of $\mathbf{Y}$ is $N - 1$ for any tree structure in which all nodes have degree ≥ 3, and $N - 2$ otherwise, for any $l \geq 2$.

$N-2$ **base case,** $N=3$**:** The tree can only be a root connected to two leaves. There is only one y-vector, between the two leaves. So the rank is $N-2=1$.

$N-1$ **base case,** $N=4$**:** r connects to each of the three 3 leaves. In general for $N=l+1, l\geq 3$, we can show the rank is $N-1=l$. WLOG, choose a leaf i and consider $y_{i,j}, \forall j \neq i$. Since every z_j is 0 everywhere except for positions indexed by node i and root r, where it is 1, $y_{i,j}$ is 1 at i,j and 0 otherwise. Therefore the set of $l-1$ vectors $\mathbf{S_i}=\{y_{i,j}, i\neq j, \forall j\}$ is linearly independent. Additionally, we can choose any two distinct nodes j,k where $j\neq k\neq i$, and $\mathbf{S}=\{y_{j,k}\cup \mathbf{S_i}\}$ must also be a linearly independent set. This follows because $y_{j,k}$ is 1 at nodes j,k and 0 otherwise; and no linear combination of $y_{i,k}, y_{i,j}$ (with $i=1$ for both vectors, which should be cancelled) can reproduce $y_{j,k}$.

Can we add any more vectors from $\mathbf{Y}$ to $\mathbf{S}$ such that the set is still linearly independent? No. To show this, assume that there exists some $y_{g,h}\notin \text{span}(\mathbf{S})$. Trivially, $g\neq h\neq i$ and $\{g,h\}\neq\{j,k\}$, so $y_{g,h}$ is 0 everywhere except at the g,h positions. But note that we can write $y_{g,h}$ using vectors in $\mathbf{S}$:

$$y_{g,h}=y_{i,g}+y_{i,h}-\big(\underbrace{y_{i,j}+y_{i,k}-y_{j,k}}_{=2 \text{ at } i;\ 0 \text{ otherwise}}\big) \tag{9}$$

So our assumption was incorrect and $\mathbf{Y}\in\text{span}(\mathbf{S})$. Therefore $\text{rank}(\mathbf{Y})=l=N-1$.

Induction hypothesis, $N\leq m$**:** Assume that the rank of *any* tree structure with parameter $N\leq m, m\in\mathbb{N}$, has rank $N-1=m-1$ if all inner nodes are degree 3 or more, and $N-2=m-2$ otherwise. We call this a m-tree.

Inductive step, $N=m+1\geq 4$**:** We define a set of "adjacent" leaf nodes $\mathbf{A}$ to be the set of all leaves connected to the same parent node ρ where ρ has no non-leaf children.

Note first that we can construct any valid $[m+1]$-tree structure from some previous valid m-tree: for any $[m+1]$-tree, consider a complete set of adjacent leaf nodes of depth at least 2 from r – that is, all the leaves connected to the same non-root node ρ (if depth is less than 2, we get the extended base case). These leaves have common ancestry, and their z-parameters differ only at the indices of the leaves themselves. Since $n>1$, there must exist at least one node ρ that is neither r nor a leaf. Collapse the ρ-sub-tree by removing ρ and reconnecting its leaves to the immediate parent node of ρ. Then the new tree is an m-tree (we have lost one node), as claimed.

Every tree must have at least one such set $\mathbf{A}$, and it must have at least 2 elements. We consider the *smallest* $\mathbf{A}$-sub-tree and the $\mathbf{L}/\mathbf{A}$-sub-tree consisting of the rest of the tree, in two cases.

Case 1, $|\mathbf{A}|\geq 3$**:** From the base case, the rank of the $\mathbf{A}$-sub-tree up to the parent node ρ is $|\mathbf{A}|$. Now consider the sub-tree consisting of all the other leaves in the tree, $\mathbf{L}/\mathbf{A}$, up until ρ, where the two sub-trees connect. Imagine ρ is a leaf (ignore its children $\mathbf{A}$). Then the $\mathbf{L}/\mathbf{A}$-sub-tree has tree parameter $N_{\mathbf{L}/\mathbf{A}}=m+1-\mathbf{A},\leq m$, so by the inductive hypothesis it has rank $N-1-\mathbf{A}$ or $N-2-\mathbf{A}$, depending on the degree of r.

How do we join the sub-trees? Let links including ρ now extend to some (the same) new leaf in $\{\mathbf{A}\}$, say a_i. The union of the $y_{i,j}$ making up each sub-tree's basis set is linearly independent. To see this, note that the only $y_{i,j}$ that have a 1 at position ρ were those ending at ρ the leaf on $\{\mathbf{L}/\mathbf{A}\}$. So to express any y_{l_k,a_i}, a link via ρ is needed, that is the rank is lower-bounded by $N-1$ or $N-2$ respectively.

We cannot add any more ys: since the sub-trees are saturated by assumption, any addition y should also be a link, but any link can be expressed via an existing link through ρ and combinations of the sub-tree basis sets (following the argument for the extended base case $N>3$, since we know that each sub-tree has more than 3 nodes). Therefore rank($[m+1]$-tree) $=N-1$ or $N-2$ based on the sub-tree, as required.

Case 2, $|\mathbf{A}|=2$**:** In this case, the $\mathbf{L}/\mathbf{A}$-sub-tree has rank $N-3$ or $N-4$ and the $\mathbf{A}$-sub-tree has internal rank 1 since there is just one pair of leaves to make y_{a_1,a_2}. WLOG, assume the ρ links extend to a_1, with $\{y_{l_i,a_1}\}$ for some (not necessarily single) i, necessarily linearly independent of y_{a_1,a_2}. The total rank is then $N-2$ or $N-3$.

In the 2-case, we can also create another y_{l_j,a_2}, where j can either be the same as i or different.

How do we know that y_{l_j,a_2} is not in the span of the two sub-tree's y vectors? Assume it is in the span. y_{l_j,a_2} has 1s at ρ. The only other ys which shares this property are from y_{l_i,a_1}. So, if y_{l_j,a_2} could be written using already chosen y, its decomposition must include y_{l_i,a_1}. But since a_1 is non-zero in y_{a_2,l_j}, it should be removed through linear combinations with other ys; however, there is just one y involving a_1, y_{a_1,a_2}. So we must subtract y_{a_1,a_2}, resulting in -1 at the a_2 position. But we require a_2 to be 1: there is no linear combination that allows both the ancestral node ρ and a_2 to be 1.

So we must be able to add one link, and so here, too, the rank is $N-1$ or $N-2$ in total (trivially any further ys must be links and so would be in the span).

□