ENTROPIC WASSERSTEIN COMPONENT ANALYSIS

Antoine Collas[1], *Titouan Vayer*[2], *Rémi Flamary*[3], *Arnaud Breloy*[4]

[1]Université Paris-Saclay, Inria, CEA
[2]Université Lyon, Inria, CNRS, ENS Lyon, UCB Lyon 1, LIP UMR 5668
[3]Ecole Polytechnique, Institut Polytechnique de Paris, CMAP, UMR 7641
[4]LEME, Université Paris Nanterre

ABSTRACT

Dimension reduction (DR) methods provide systematic approaches for analyzing high-dimensional data. A key requirement for DR is to incorporate global dependencies among original and embedded samples while preserving clusters in the embedding space. To achieve this, we combine the principles of optimal transport (OT) and principal component analysis (PCA). Our method seeks the best linear subspace that minimizes reconstruction error using entropic OT, which naturally encodes the neighborhood information of the samples. From an algorithmic standpoint, we propose an efficient block-majorization-minimization solver over the Stiefel manifold. Our experimental results demonstrate that our approach can effectively preserve high-dimensional clusters, leading to more interpretable and effective embeddings. Python code of the algorithms and experiments is available online[1].

Index Terms— Dimension reduction, PCA, Optimal Transport, entropy, block-majorization-minimization

1. INTRODUCTION

Given a set of n samples of dimension d, denoted $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$, a linear dimension reduction consists in projecting the data onto a k-dimensional subspace ($k < p$) as $\mathbf{U}^\top\mathbf{X} \in \mathbb{R}^{k\times n}$, where $\mathbf{U} \in \mathrm{St}(d,k)$ is an orthonormal basis (as $\mathrm{St}(d,k) = \{\mathbf{U} \in \mathbb{R}^{d\times k}, \mathbf{U}^\top\mathbf{U} = \mathbf{I}_k\}$ denotes the Stiefel manifold). The most celebrated method in this framework is probably the principal component analysis (PCA) that selects the k leading eigenvectors of $\mathbf{X}$ for the projection basis [1].

Interestingly, this basis appears as the solution to many underlying optimization problems, whose formulations offer points of view to generalize PCA and thus alleviate several of its shortcomings. For example, PCA minimizes the average squared distance between the samples and their projection on the subspace spanned by $\mathbf{U}$. Generalizations can then come from considering the minimization of robust distances to be less sensitive to outliers [2, 3]. A second example is that any formulation of PCA as an optimization problem can be regularized to promote certain properties of the solution. This is the starting point of many sparse PCA algorithms that aim to obtain a sparse basis $\mathbf{U}$, *i.e.*, promoting the projection to act as a variable selection [4].

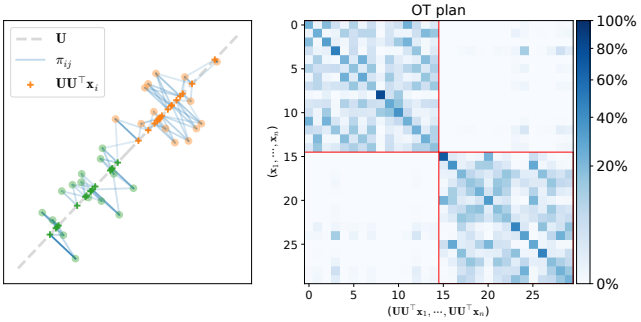


Fig. 1: Illustration of (EWCA) with 2D samples organized in two clusters (in green and orange). On the left are the samples and their 1D projections, and on the right is the corresponding OT transport plan.

In this work, we explore a reformulation of PCA as the solution to an optimal transport (OT) problem [5]. We show that optimizing an entropic OT between samples $\mathbf{X}$ and their projected counterparts $\mathbf{U}\mathbf{U}^\top\mathbf{X}$ encodes neighborhood information between samples. Interestingly, optimizing exact OT (in the special case in the absence of entropic regularization) is equivalent to standard PCA. Thanks to the underlying principles of OT, this new approach is able to capture both global linear embeddings as well as local interactions between samples.

The contributions are the following: i) We reformulate a subspace recovery problem with an OT objective and show that it indeed yields the standard PCA when using least-squares cost and no regularization; ii) We propose a block-coordinate descent (BCD) algo-

[1]https://github.com/antoinecollas/Entropic_Wasserstein_Component_Analysis

979-8-3503-2411-2/23/$31.00 ©2023 IEEE

rithm to solve the corresponding optimization problem and a more efficient alternative using the majorization-minimization framework [6]; *iii*) We perform numerical experiments on genome data [7, 8] that show a gain in accuracy compared to the standard PCA when used as a preprocessing step for classification problems.

2. ENTROPIC WASSERSTEIN COMPONENT ANALYSIS (EWCA)

Entropic Optimal Transport. Given two datasets $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_n), \mathbf{Y} = (\mathbf{y}_1, \cdots, \mathbf{y}_m)$ with $\mathbf{x}_i, \mathbf{y}_j \in \mathbb{R}^d$ and $c : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$ a cost function. Consider two histograms $\mathbf{a} \in \Sigma_n, \mathbf{b} \in \Sigma_m$ (*i.e.* $a_i \geq 0, \sum_{i=1}^n a_i = 1$) the entropic OT problem aims at solving, for $\varepsilon > 0$,

$$\mathrm{OT}_{\varepsilon,c}(\mathbf{a}, \mathbf{b}, \mathbf{X}, \mathbf{Y}) \triangleq \min_{\boldsymbol{\pi} \in \Pi(\mathbf{a},\mathbf{b})} \sum_{i,j=1}^{n,m} c(\mathbf{x}_i, \mathbf{y}_j)\pi_{ij} - \varepsilon \, \mathrm{H}(\boldsymbol{\pi}) \,,$$

where $\Pi(\mathbf{a}, \mathbf{b}) = \{\boldsymbol{\pi} \in \mathbb{R}_+^{n \times m};\ \boldsymbol{\pi}\mathbf{1}_m = \mathbf{a}, \boldsymbol{\pi}^\top \mathbf{1}_n = \mathbf{b}\}$ is the set of couplings between $\mathbf{a}, \mathbf{b}$ and $\mathrm{H}(\boldsymbol{\pi}) = -\sum_{ij} \log(\pi_{ij}/a_i b_j)\pi_{ij}$ is the negative entropy. To simplify the notations we write $\mathrm{OT}_{\varepsilon,c}(\mathbf{X}, \mathbf{Y})$ when $n = m$ and $\mathbf{a} = \mathbf{b} = \frac{1}{n}\mathbf{1}_n$. This problem can be solved using the Sinkhorn-Knopp (SK) algorithm [9]. Specifically, given the Gibbs kernel $\mathbf{K} = \exp(-\mathbf{C}/\varepsilon)$, SK alternates (until convergence) the two steps

$$\mathbf{u} \leftarrow \mathbf{a} \oslash \mathbf{K}\mathbf{v} \text{ // Update left scaling}$$
$$\mathbf{v} \leftarrow \mathbf{b} \oslash \mathbf{K}^\top \mathbf{u} \text{ // Update right scaling}$$

and returns the coupling $\boldsymbol{\pi} = \mathrm{diag}(\mathbf{u})\mathbf{K}\,\mathrm{diag}(\mathbf{v})$. SK involves simple iterations of matrix-vectors products that can run efficiently on GPU.

Entropic Wasserstein Component Analysis. The principle of our method is to consider the optimization problem

$$\min_{\mathbf{U} \in \mathrm{St}(d,k)} \mathrm{OT}_{\varepsilon,c}(\mathbf{X}, \mathbf{U}\mathbf{U}^\top \mathbf{X}) \,, \qquad \text{(EWCA)}$$

with the classical squared ℓ_2 cost function $c(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2^2$. Problem (EWCA) is a non-convex problem that equivalently writes

$$\min_{\substack{\boldsymbol{\pi} \in \Pi(\frac{1}{n}\mathbf{1}_n, \frac{1}{n}\mathbf{1}_n) \\ \mathbf{U} \in \mathrm{St}(d,k)}} \sum_{i,j=1}^{n,n} \|\mathbf{x}_i - \mathbf{U}\mathbf{U}^\top \mathbf{x}_j\|_2^2 \, \pi_{ij} - \varepsilon \, \mathrm{H}(\boldsymbol{\pi}). \quad (1)$$

The objective function shares similarities with PCA but with key differences: the OT plan $\boldsymbol{\pi}$ assigns weights between original and projected samples, while the entropic regularization adjusts the spread of mass between them. Therefore, as shown in Figure 1, the OT plan weights the samples within the neighborhood of the projected points, promoting the clustering of these points. Note that a similar strategy was used in Wasserstein Discriminant Analysis [10] to find a discriminant subspace for the data with a different objective. (EWCA) is a difficult non-convex problem, and we propose two algorithms to solve it in the next section.

Algorithm 1 BCD for solving (1) when $\varepsilon > 0$

$n_{it}, \mathbf{a}, \mathbf{b}, \epsilon > 0, \mathbf{U}^{(0)}$
while not converged **do**
  Let $\mathbf{C}^{(t)} = (c(\mathbf{x}_i, \mathbf{U}^{(t)}(\mathbf{U}^{(t)})^\top \mathbf{x}_j))_{ij}$
  Find $\boldsymbol{\pi}^{(t)}$ with Sinkhorn-Knopp algorithm
  Let $\mathbf{M}^{(t)} = \mathbf{X}\left(2\,\mathrm{sym}(\boldsymbol{\pi}^{(t)}) - \frac{1}{n}\mathbf{I}\right)\mathbf{X}^\top$
  Find $\mathbf{u}_1, \cdots, \mathbf{u}_k$ the eigenvectors corresponding the the k highest eigenvalues of $\mathbf{M}^{(t)}$
  $\mathbf{U}^{(t+1)} = (\mathbf{u}_1, \cdots, \mathbf{u}_k)$
  $t = t + 1$
end while
return $\mathbf{U}^{(t)}, \boldsymbol{\pi}^{(t)}$

Limit cases. Interstingly, when $\varepsilon \to 0$, we have $\boldsymbol{\pi} \to \frac{1}{n}\mathbf{I}_n$ and we recover the PCA objective. Conversely when $\varepsilon \to +\infty$, $\boldsymbol{\pi} \to \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$. From the reformulation (3) in the next section and when the data are centered (*i.e.* $\mathbf{X}\mathbf{1}_n = 0$), solving (1) in $\mathbf{U}$ when $\varepsilon \to +\infty$ corresponds to $\min_{\mathbf{U} \in \mathrm{St}(d,k)} \mathrm{tr}\left(\mathbf{U}^\top [\frac{1}{n}\mathbf{X}\mathbf{X}^\top]\mathbf{U}\right)$ that is finding the k eigenvectors corresponding to the k lowest eigenvalues of the empirical covariance matrix. Therefore, ε allows to interpolate between estimating eigenvectors of the empirical covariance matrix for the k highest and lowest eigenvalues.

3. OPTIMIZATION ALGORITHMS FOR ENTROPIC WASSERSTEIN COMPONENT ANALYSIS

Block coordinate descent (BCD). A first approach to tackle the optimization problem (EWCA) is presented with BCD algorithm. Indeed, the cost function in (1) can be minimized by alternating a minimization over $\boldsymbol{\pi} \in \Pi(\frac{1}{n}\mathbf{1}_n, \frac{1}{n}\mathbf{1}_n)$ (with SK algorithm) and a minimization over $\mathbf{U} \in \mathrm{St}(d,k)$ with fixed $\boldsymbol{\pi}$. The latter step requires solving

$$\min_{\mathbf{U} \in \mathrm{St}(d,k)} \sum_{i,j}^{n,n} \|\mathbf{x}_i - \mathbf{U}\mathbf{U}^\top \mathbf{x}_j\|_2^2 \pi_{ij} \,. \qquad (2)$$

As described in Lemma 1, problem (2) is minimized by finding the k eigenvectors of $\mathbf{M} \triangleq \mathbf{X}\left(2\,\mathrm{sym}(\boldsymbol{\pi}) - \frac{1}{n}\mathbf{I}\right)\mathbf{X}^\top$ associated with the k highest eigenvalues. Consequently, the BCD procedure, summarized in Algorithm 1, alternates between Sinkhorn-Knopp and computing the eigenvectors of $\mathbf{M}$. The complexity of the BCD is presented in Table 1.

Lemma 1. *For any $\boldsymbol{\pi} \in \Pi(\frac{1}{n}\mathbf{1}_n, \frac{1}{n}\mathbf{1}_n)$ the problem (2) is equivalent to*

$$\max_{\mathbf{U}\in \mathrm{St}(d,k)} \mathrm{tr}\left(\mathbf{U}^\top \mathbf{M} \mathbf{U}\right) \tag{3}$$

where

$$\mathbf{M} \triangleq \mathbf{X}\left(2\,\mathrm{sym}(\boldsymbol{\pi}) - \frac{1}{n}\mathbf{I}\right)\mathbf{X}^\top \tag{4}$$

with $\mathrm{sym}(\boldsymbol{\pi}) \triangleq \left(\boldsymbol{\pi} + \boldsymbol{\pi}^\top\right)/2$. Hence, the solution of (2) is given by $\mathbf{U}^\star = (\mathbf{u}_1, \cdots, \mathbf{u}_k)$ where the $\mathbf{u}_i$'s are the eigenvectors corresponding the the k highest eigenvalues of $\mathbf{M}$.

Proof. The cost can be written as $\frac{1}{n}\sum_i \|\mathbf{x}_i\|_2^2 + \frac{1}{n}\sum_i \|\mathbf{U}\mathbf{U}^\top\mathbf{x}_i\|_2^2 - 2\sum_{ij}\langle \mathbf{x}_i, \mathbf{U}\mathbf{U}^\top\mathbf{x}_j\rangle \pi_{ij}$. The second term writes $\frac{1}{n}\sum_i \mathrm{tr}(\mathbf{U}\mathbf{U}^\top\mathbf{x}_i\mathbf{x}_i^\top) = \mathrm{tr}(\mathbf{U}\mathbf{U}^\top\frac{1}{n}\mathbf{X}\mathbf{X}^\top)$ and the third term $-2\mathrm{tr}(\mathbf{U}\mathbf{U}^\top\mathbf{X}\boldsymbol{\pi}\mathbf{X}^\top) = -2\mathrm{tr}(\mathbf{U}\mathbf{U}^\top\mathbf{X}\,\mathrm{sym}(\boldsymbol{\pi})\mathbf{X}^\top)$ since $\mathbf{X}^\top\mathbf{U}\mathbf{U}^\top\mathbf{X}$ is symmetric. By combining both, we obtain (3). To conclude, we used the Ky-Fan theorem. □

Block-majorization-minimization (block-MM). BCD is a simple approach but can become slow on high dimensional data due to the computational complexity of $\mathcal{O}(d^3)$. To alleviate this problem, we seek to solve (2) without actually computing a $p \times p$ matrix by relying on block-MM algorithms over $\mathrm{St}(d,k)$ [6]. In the next Lemma, we first formulate a problem that is equivalent to (2) when restricted to $\mathrm{St}(d,k)$, but whose objective can be globally majorized on $\mathbb{R}^{d\times k}$ by a linear function.

Lemma 2. *For any $\boldsymbol{\pi} \in \Pi(\frac{1}{n}\mathbf{1}_n, \frac{1}{n}\mathbf{1}_n)$ the problem (2) is equivalent to the following minimization problem*

$$\min_{\mathbf{U}\in \mathrm{St}(d,k)} \mathrm{tr}\left(\mathbf{U}^\top \mathbf{P} \mathbf{U}\right) \tag{5}$$

where $\mathbf{P} \preccurlyeq \mathbf{0}$ is the matrix defined as

$$\mathbf{P} \triangleq \alpha_{\boldsymbol{\pi}}\left(\boldsymbol{\Sigma} - 1_{\alpha_{\boldsymbol{\pi}}>0}\lambda_{\max}^{\boldsymbol{\Sigma}}\mathbf{I}\right) - 2\mathbf{X}\left(\mathrm{sym}(\boldsymbol{\pi}) - \lambda_{\min}^{\mathrm{sym}(\boldsymbol{\pi})}\mathbf{I}\right)\mathbf{X}^\top, \tag{6}$$

and where $\lambda_{\max}^{\boldsymbol{\Sigma}}$ is the largest eigenvalue of $\boldsymbol{\Sigma} \triangleq \frac{1}{n}\mathbf{X}\mathbf{X}^\top$, $\lambda_{\min}^{\mathrm{sym}(\boldsymbol{\pi})}$ is the smallest eigenvalue of $\mathrm{sym}(\boldsymbol{\pi})$, $\alpha_{\boldsymbol{\pi}} \triangleq 1 - 2n\lambda_{\min}^{\mathrm{sym}(\boldsymbol{\pi})}$, and $1_{\alpha_{\boldsymbol{\pi}}>0}$ is equal to 1 if $\alpha_{\boldsymbol{\pi}} > 0$ and 0 otherwise.

Proof. The problem (3) is rewritten as

$$\min_{\mathbf{U}\in \mathrm{St}(d,k)} \mathrm{tr}\left(\mathbf{U}^\top \boldsymbol{\Sigma} \mathbf{U}\right) - 2\mathrm{tr}\left(\mathbf{U}^\top \mathbf{X}\,\mathrm{sym}(\boldsymbol{\pi})\mathbf{X}^\top\mathbf{U}\right)$$

which can also be rewritten as

$$\min_{\mathbf{U}\in \mathrm{St}(d,k)} \alpha_{\boldsymbol{\pi}}\mathrm{tr}\left(\mathbf{U}^\top \boldsymbol{\Sigma} \mathbf{U}\right) - 2\mathrm{tr}\left(\mathbf{U}^\top\mathbf{X}\left(\mathrm{sym}(\boldsymbol{\pi}) - \lambda_{\min}^{\mathrm{sym}(\boldsymbol{\pi})}\mathbf{I}\right)\mathbf{X}^\top\mathbf{U}\right).$$

Algorithm 2 Block-MM for solving (1) when $\varepsilon > 0$

$n_{it}, m_{it}, \mathbf{a}, \mathbf{b}, \epsilon > 0$, $\mathbf{U}^{(0)}$
while not converged **do**
  Let $\mathbf{C}^{(t)} = (c(\mathbf{x}_i, \mathbf{U}^{(t)}(\mathbf{U}^{(t)})^\top\mathbf{x}_j))_{ij}$
  Find $\boldsymbol{\pi}^{(t)}$ with Sinkhorn-Knopp algorithm
  for $\mathbf{U}^{(l=0)} = \mathbf{U}^{(t)}$, $l = 1, \ldots, m_{it}$ **do**
    Compute $\mathbf{P}^{(t)}\mathbf{U}^{(l)}$ as in (9)
    $\mathbf{U}^{(l+1)} = \mathrm{qf}(\mathbf{P}^{(t)}\mathbf{U}^{(l)})$ # QR orth.
  end for
  $\mathbf{U}^{(t)} = \mathbf{U}^{(l+1)}$
  $t = t + 1$
end while
return $\mathbf{U}^{(t)}, \boldsymbol{\pi}^{(t)}$

Finally, we observe that when $\alpha_{\boldsymbol{\pi}}$ is non-positive $\mathbf{U} \mapsto \alpha_{\boldsymbol{\pi}}\mathrm{tr}\left(\mathbf{U}^\top\boldsymbol{\Sigma}\mathbf{U}\right)$ is concave over $\mathbb{R}^{d\times k}$ since $\boldsymbol{\Sigma} \succcurlyeq \mathbf{0}$. Otherwise (*i.e.* $\alpha_{\boldsymbol{\pi}}$ is positive), we remark that the restriction of $\alpha_{\boldsymbol{\pi}}\mathrm{tr}\left(\mathbf{U}^\top\boldsymbol{\Sigma}\mathbf{U}\right)$ to $\mathrm{St}(d,k)$ coincide with the concave function $\mathbf{U} \mapsto \alpha_{\boldsymbol{\pi}}\mathrm{tr}\left(\mathbf{U}^\top\left(\boldsymbol{\Sigma} - \lambda_{\max}^{\boldsymbol{\Sigma}}\mathbf{I}\right)\mathbf{U}\right) + k\lambda_{\max}^{\boldsymbol{\Sigma}}$. □

Given the current iterate $\mathbf{P} \preccurlyeq \mathbf{0}$, the objective of (5) is concave over $\mathbb{R}^{d\times k}$, so it can be majorized by its first order Taylor expansion at the point $\mathbf{U}^{(l)} \in \mathrm{St}(d,k)$:

$$\mathrm{tr}\left(\mathbf{U}^\top\mathbf{P}\mathbf{U}\right) \leq 2\mathrm{tr}\left(\mathbf{U}^\top\mathbf{P}\mathbf{U}^{(l)}\right) + \mathrm{const.} \tag{7}$$

The minimizer of the above upper bound on $\mathrm{St}(d,k)$ is the orthogonal projection of $-\mathbf{P}\mathbf{U}^{(l)} \in \mathbb{R}^{d\times k}$ onto $\mathrm{St}(d,k)$, *i.e.*

$$\mathbf{U}^{(l+1)} = \mathrm{pf}(-\mathbf{P}\mathbf{U}^{(l)}) \tag{8}$$

where $\mathrm{pf}(\mathbf{A})$ is the orthogonal factor from the polar factorization of $\mathbf{A} \in \mathbb{R}^{d\times k}$. Multiple iterations of (8) correspond to an MM algorithm that reaches a critical point of (2) [6]. Moreover, the cost function (2) is invariant to the action of $\mathrm{St}(k,k)$. Consequently, any operator that yields the span of $-\mathbf{P}\mathbf{U}^{(l)}$ (*e.g.*, a QR decomposition) is a valid alternative to $\mathrm{pf}(\cdot)$ in order to compute the update (8). An additional reduction in computational cost is realized by directly computing the product $\mathbf{P}\mathbf{U}^{(l)}$ as

$$\mathbf{P}\mathbf{U}^{(l)} = \alpha_{\boldsymbol{\pi}}\left[\frac{1}{n}\mathbf{X}\left(\mathbf{X}^\top\mathbf{U}^{(l)}\right) - 1_{\alpha_{\boldsymbol{\pi}}>0}\lambda_{\max}^{\boldsymbol{\Sigma}}\mathbf{U}^{(l)}\right] - 2\mathbf{X}\left[\mathrm{sym}(\boldsymbol{\pi}) - \lambda_{\min}^{\mathrm{sym}(\boldsymbol{\pi})}\mathbf{I}\right]\left(\mathbf{X}^\top\mathbf{U}^{(l)}\right). \tag{9}$$

Hence, we transformed the BCD update that computes a $d \times d$ matrix and its SVD by computing a $d \times k$ matrix and its QR decomposition. This strategy proves effective when $k \ll d$. The overall block-MM procedure with the qf function that returns the orthogonal factor of the QR decomposition and its complexity are summarized in Algorithm 2 and Table 1 respectively.

Common steps to BCD (Alg. 1) and Block-MM (Alg. 2)	Computation of $\mathbf{C}^{(t)} = (c(\mathbf{x}_i, \mathbf{U}^{(t)}(\mathbf{U}^{(t)})^\top \mathbf{x}_j))_{ij}$	$\mathcal{O}(n^2 d)$
	Computation of $\boldsymbol{\pi}^{(t)}$	$\mathcal{O}(n^2)$
BCD: scales with n (Alg. 1)	Computation of $\mathbf{M}^{(t)}$ (Eq. (4))	$\mathcal{O}(n^2 d + nd^2)$
	Computation of the eigenvectors of $\mathbf{M}$	$\mathcal{O}(d^3)$
	Overall complexity	$\mathcal{O}(n^2 d + nd^2 + d^3)$
Block-MM: scales with d (Alg. 2)	Computation of $\mathbf{P}^{(t)}\mathbf{U}^{(l)}$ (Eq. (9))	$\mathcal{O}(ndk + n^3)$
	Projection of $\mathbf{P}^{(t)}\mathbf{U}^{(l)}$ onto $\mathrm{St}(d,k)$	$\mathcal{O}(dk^2)$
	Overall complexity	$\mathcal{O}(n^2 d + n^3)$

Table 1: Comparison of the computation complexities of Algorithms 1 and 2 with respect to the executed steps.

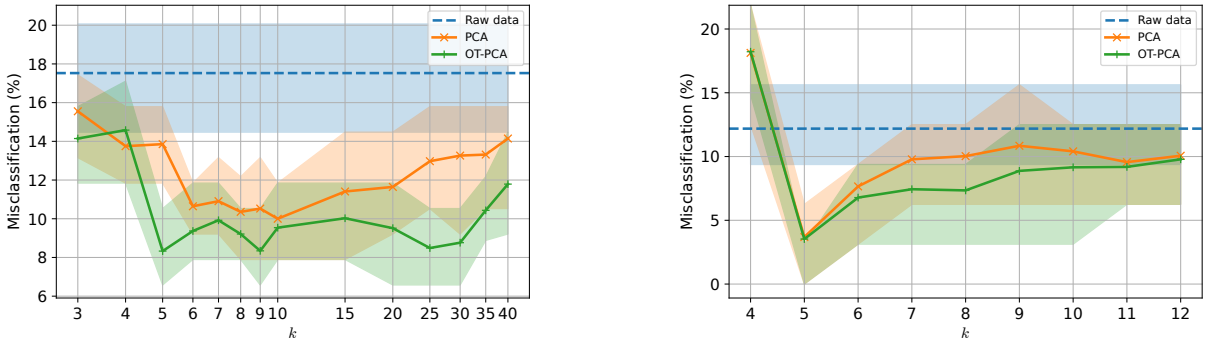
Computation of $\lambda_{\max}^{\boldsymbol{\Sigma}}$ and $\lambda_{\min}^{\mathrm{sym}(\boldsymbol{\pi})}$. Finally, we remark that the computation of the eigenvalues $\lambda_{\max}^{\boldsymbol{\Sigma}}$ and $\lambda_{\min}^{\mathrm{sym}(\boldsymbol{\pi})}$ is not a limitation. Indeed, in practice, both Algorithms 1 and 2 are initialized with the PCA. Thus, the $\lambda_{\max}^{\boldsymbol{\Sigma}}$ is obtained from this initialization. Then, $\lambda_{\min}^{\mathrm{sym}(\boldsymbol{\pi})}$ can be replaced by any lower bound. Since $\mathrm{sym}(\boldsymbol{\pi}) \in \Pi(\frac{1}{n}, \frac{1}{n})$, and using the Gershgorin circle theorem, a lower bound of $\lambda_{\min}^{\mathrm{sym}(\boldsymbol{\pi})}$ is $-\frac{1}{n}$. This replacement is convenient when n is large since it avoids the computation of the SVD of $\mathrm{sym}(\boldsymbol{\pi})$. As the corresponding majorizer is less tight, we observed in practice a slightly lower rate of convergence in iterations when replacing $\lambda_{\min}^{\mathrm{sym}(\boldsymbol{\pi})}$ by $-\frac{1}{n}$ in Equation (9).

4. NUMERICAL EXPERIMENTS

To assess the performance of the developed Algorithms 1 and 2, we leverage two classification datasets: *Breast* [8] and *khan2001* [7]. The *Breast* dataset contains $n = 151$ samples with $d = 54675$ gene expressions each. The goal is to classify these data into 6 classes corresponding to breast cancer subtypes and normal tissues. The *Khan2001* dataset contains $n = 63$ samples of $d = 2308$ gene expression profiles to classify into 4 types of tumors of childhood. Notice that the interpolation effect induced by ϵ in EWCA trades some explained variance (maximized by the PCA) for an alternate representation of the data. The relevance and quality of this representation are analyzed through quantitative and qualitative experiments on both datasets. Moreover, the speed of the block-MM algorithm (Alg. 2) over the BCD algorithm (Alg. 1) is shown in an experiment on the *Breast* dataset.

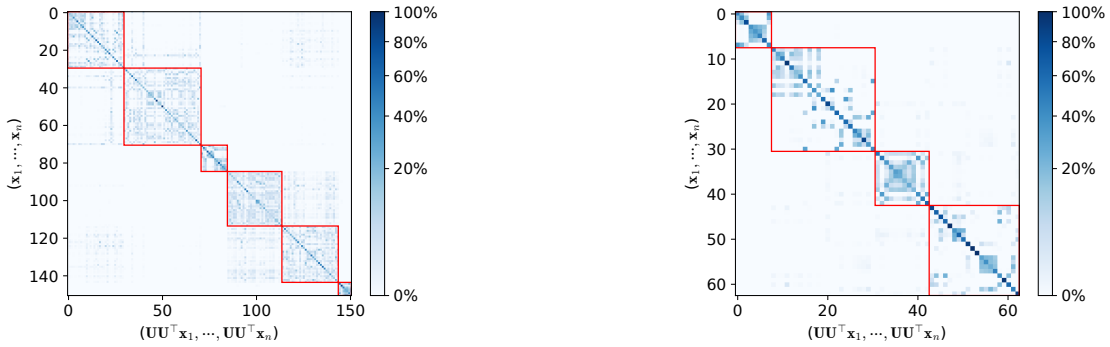
Classification performance. We first compute the misclassification rate, over 100 train-test splits ($50\% - 50\%$), of a 1 nearest neighbor classifier on the raw data $(\mathbf{x}_1, \cdots, \mathbf{x}_n)$. Then, we estimate subspaces with PCA and EWCA across their respective hyperparameters, *i.e.*the subspace dimension k and the intensity of the entropy regularization ε. For each value of k, we compute the misclassification rates, over 100 train-test splits ($50\% - 50\%$), of a 1 nearest neighbor classifier applied on the projected data $(\mathbf{U}^\top \mathbf{x}_1, \cdots, \mathbf{U}^\top \mathbf{x}_n)$. The hyperparameter ε of EWCA is chosen over 20 splits on the train set. The mean misclassification rates, as well as the 1st and 3rd quartiles, are reported across the different tested subspace dimensions k in the Figure 2. We observe that EWCA and PCA give better accuracies than the 1 nearest neighbor classifier applied directly on raw data, showing interest in considering dimension reduction methods. Then, EWCA outperforms PCA on a wide range of values of k on both datasets. For certain values of k, the improvement in classification performance is large. Indeed, on the *Breast* dataset, at $k = 5$, the misclassification rate is down from 14% to 8%; *i.e.* a reduction of half of the error. On the *Khan2001* dataset, at $k = 8$, the misclassification rate is down from 11% to 7.5%; *i.e.* a reduction of a third of the error. This improvement in discriminative capabilities indicates that EWPCA provides linear embeddings that favor clusters within samples in an unsupervised way.

Transport plan interpretation. Then, Figure 3 displays the transport plans $\boldsymbol{\pi}$ estimated by EWCA at $k = 5$ with ε chosen as the best performing on the 20 splits of the train set. The data are ordered by class, and those that belong to the same class are enclosed in a red box. We observe on both datasets that the transport plans values π_{ij} are higher within data that belong to the same class than within data that belong to different classes. This means that given a point $\mathbf{x}_i$ that belongs to the class y_i, the estimated subspace minimizes the discrepancy between $\mathbf{x}_i$ and the projected points $\mathbf{U}\mathbf{U}^\top \mathbf{x}_j$ that belong to the class y_i. This enforces, in an unsupervised way, that points that belong to the same class are close to each other once projected in the estimated subspace. Furthermore, using the transport plan from the *Khan2001* dataset, several clusters can be identified. Indeed, in the red square on the top left corner (class 1) of Figure 3b, two clusters are distinguishable. These two clusters are also observable in the samples from class 1 in Figure 4. The latter plots a TSNE [11] of projected data $(\mathbf{U}^\top \mathbf{x}_1, \cdots, \mathbf{U}^\top \mathbf{x}_n)$ and the transport plan values. This again indicates that EWCA identifies clusters by jointly estimating the transport plan and the subspace to project data on.



(a) *Breast* dataset (b) *khan2001* dataset

Fig. 2: **Misclassification rate (%) versus subspace dimension k (the lower the better)**. Data are classified using a 1 nearest neighbor classifier on 100 splits train-test ($50\% - 50\%$). In addition to the raw data (no preprocessing), two preprocessing are considered: PCA and EWCA (proposed in Algorithm 2). The value of ε of EWCA is chosen as the best performing on 20 splits of the train set. The mean misclassification and the 1^{st} and 3^{rd} quartiles are reported.



(a) *Breast* dataset (b) *khan2001* dataset

Fig. 3: **Transport plan π (%) computed with EWCA (Alg. 2) between raw data $(\mathbf{x}_1, \cdots, \mathbf{x}_n)$ and their projected counterparts $(\mathbf{U}\mathbf{U}^\top\mathbf{x}_1, \cdots, \mathbf{U}\mathbf{U}^\top\mathbf{x}_n)$**. $k = 5$ and the values of ε are the one performing the best on Figure 2. The red squares enclose the data belonging to the same class. Classes are ordered from left to right and top to bottom, *i.e.*the red square on the top left corner is class 1, and the red square on the bottom right corner is the last class.

Computation cost: block-MM versus BCD. So far, we have shown the good performance of EWCA in terms of precisions and given an interpretation of the estimated transport plan. We now leverage the *Breast* dataset to analyze the computational time of the proposed Algorithms 1 and 2. Indeed, we subsample $d \in [\![500, 54675]\!]$ genes and run the two algorithms until convergence. The mean computation time in seconds and the 1^{st} and 3^{rd} quartiles are reported. When $d \leq 2000$, the BCD is faster than the block-MM thanks to its closed form formula on the $\mathbf{U}$-step. However, when $d > 2000$, the block-MM algorithm is much faster than the BCD with a much lower rate of increase. This illustrates the lower computation complexity in d of the block-MM compared to the BCD one (see Table 1).

Sensitivity to hyperparameter ε. In the classification tasks, we selected the hyperparameter ε as the best performing one for a 1 nearest neighbor classifier on 20 splits of the train set. To mitigate this necessity of testing many values of ε, we plot in Figure 6 the heat map of the misclassification rates with respect to k and ε using the same protocol as the one used in Figure 2. On a wide range of ε, we observe that EWCA has similar misclassification rates as PCA, if not better. Hence, EWCA is not too sensitive to the choice of ε.

5. CONCLUSION

We reformulated the PCA algorithm as the minimizer of the squared 2-Wasserstein distance between a dataset and its projected counterpart. Adding an entropy regularizer enabled us to consider pairs of points $(\mathbf{x}_i, \mathbf{U}\mathbf{U}^\top\mathbf{x}_j)$, with $i \neq j$ in this new optimization problem called EWCA. To solve it, we proposed two algorithms, a BCD and a block-MM. The latter showed faster convergence in high-dimensional regimes. When leveraged as a preprocessing step for classification problems on gene expression datasets, we showed that EWCA yields a projection that favors clusters within the data

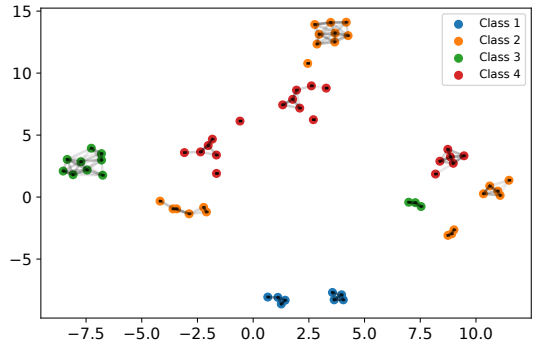


Fig. 4: **TSNE of the projected data $(\mathbf{U}^\top \mathbf{x}_1, \cdots, \mathbf{U}^\top \mathbf{x}_n)$ computed with EWCA (Alg. 2) on the *Khan2001* dataset**. The subspace dimension is chosen as $k = 5$ and the used subspace corresponds to the entropy intensity ε performing the best on Figure 2b. The grey links represent the intensity of the values of the transport plan presented in Figure 3b (values under a certain threshold are set to 0).

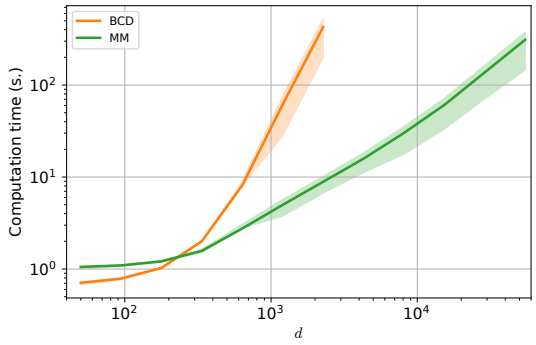


Fig. 5: **Computation time in seconds of EWCA (Alg. 2) for $k = 5$ versus data dimension d of subsampled genes of the *Breast* dataset (the lower the better)**. The mean, 1^{st} and 3^{rd} quartiles computed with 100 sets of subsampled genes are reported.

in an unsupervised way. The joint use of EWCA and its achieved transport map thus offers an interesting alternative to PCA for exploratory data analysis.

Acknowledgements. Numerical experiments have realized with: Matplotlib [12], Scikit-learn [13], Numpy [14], and POT [15]. This work was supported by ANR MASSILIA (ANR-21-CE23-0038-01).

6. REFERENCES

[1] Ian T Jolliffe and Jorge Cadima, "Principal component analysis: a review and recent developments," *Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, pp. 20150202, 2016.

[2] Chris Ding, Ding Zhou, Xiaofeng He, and Hongyuan Zha, "R1-PCA: rotational invariant ℓ_1-norm principal component analysis for robust subspace factorization," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 281–288.

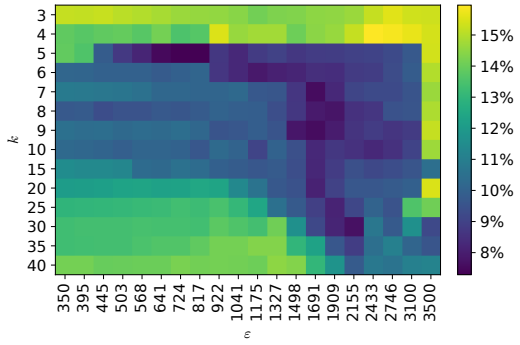


Fig. 6: **Misclassification rate (%) versus subspace dimension k and entropy intensity ε on the *Breast* dataset (the lower the better)**. Data projected on the subspaces estimated by EWCA (proposed in Algorithm 2) are classified using a 1 nearest neighbor classifier on 100 splits train-test $(50\% - 50\%)$.

[3] Gilad Lerman and Tyler Maunu, "Fast, robust and non-convex subspace recovery," *Information and Inference: A Journal of the IMA*, vol. 7, no. 2, pp. 277–336, 2018.

[4] Hui Zou, Trevor Hastie, and Robert Tibshirani, "Sparse principal component analysis," *Journal of computational and graphical statistics*, vol. 15, no. 2, pp. 265–286, 2006.

[5] G. Peyré and M. Cuturi, "Computational optimal transport," *Foundations and Trends in Machine Learning*, vol. 11, pp. 355–607, 2019.

[6] A. Breloy, S. Kumar, Y. Sun, and D. P Palomar, "Majorization-minimization on the stiefel manifold with application to robust sparse pca," *IEEE Transactions on Signal Processing*, vol. 69, pp. 1507–1520, 2021.

[7] J. Khan, J. S. Wei, M. Ringner, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. R. Antonescu, C. Peterson, and P. S. Meltzer, "Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks," *Nat Med*, vol. 7, no. 6, pp. 673–679, 2001.

[8] B. C. Feltes, E. B. Chandelier, B. I. Grisci, and M. Dorn, "Cumida: an extensively curated microarray database for benchmarking and testing of machine learning approaches in cancer research," *Journal of Computational Biology*, vol. 26, no. 4, pp. 376–386, 2019.

[9] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," *Advances in neural information processing systems*, vol. 26, 2013.

[10] R. Flamary, M. Cuturi, N. Courty, and A. Rakotomamonjy, "Wasserstein discriminant analysis," *Machine Learning*, vol. 107, no. 12, pp. 1923–1945, may 2018.

[11] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.

[12] J.D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in science & engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[13] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python ," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[14] C.R. Harris et al., "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, 2020.

[15] R. Flamary et al., "Pot: Python optimal transport," *Journal of Machine Learning Research*, vol. 22, no. 78, pp. 1–8, 2021.