Machine Learning (2021) 110:533–558
https://doi.org/10.1007/s10994-020-05931-2

Coupling matrix manifolds assisted optimization for optimal transport problems

Dai Shi[1] · Junbin Gao[1] · Xia Hong[2] · S. T. Boris Choy[1] · Zhiyong Wang[3]

Received: 26 November 2019 / Revised: 14 September 2020 / Accepted: 4 November 2020 /
Published online: 1 January 2021

Abstract
Optimal transport (OT) is a powerful tool for measuring the distance between two probability distributions. In this paper, we introduce a new manifold named as the coupling matrix manifold (CMM), where each point on this novel manifold can be regarded as a transportation plan of the optimal transport problem. We firstly explore the Riemannian geometry of CMM with the metric expressed by the Fisher information. These geometrical features can be exploited in many essential optimization methods as a framework solving all types of OT problems via incorporating numerical Riemannian optimization algorithms such as gradient descent and trust region algorithms in CMM manifold. The proposed approach is validated using several OT problems in comparison with recent state-of-the-art related works. For the classic OT problem and its entropy regularized variant, it is shown that our method is comparable with the classic algorithms such as linear programming and Sinkhorn algorithms. For other types of non-entropy regularized OT problems, our proposed method has shown superior performance to other works, whereby the geometric information of the OT feasible space was not incorporated within.

Keywords Optimal transport · Doubly stochastic matrices · Coupling matrix manifold · Sinkhorn algorithm · Wasserstein distance · Entropy regularized optimal transport

Editor: Pradeep Ravikumar.

✉ Junbin Gao
junbin.gao@sydney.edu.au

Dai Shi
dai.shi@sydney.edu.au

Xia Hong
x.hong@reading.ac.uk

S. T. Boris Choy
boris.choy@sydney.edu.au

Zhiyong Wang
zhiyong.wang@sydney.edu.au

[1] Discipline of Business Analytics, The University of Sydney Business School, The University of Sydney, Sydney, NSW 2006, Australia

[2] Department of Computer Science, University of Reading, Reading RG6 6AY, UK

[3] School of Computer Science, The University of Sydney, Sydney, NSW 2006, Australia

1 Introduction

An Optimal Transport (OT) problem can be briefly described as to find out the optimized transport plan (defined as transportation polytope) between two or more sets of subjects with certain constraints (Peyre and Cuturi 2019). It was firstly formalized by French mathematician Gaspard Monge in 1781 (Monge 1781), and was relaxed by Kantorovich who provided a solution of Monge's problem in 1942 (Kantorovich 1942) and established its importance to logistics and economics.

As the solution of the OT problem provides the optimized transportation plan between probability distributions, and the advance in computer science allows us to perform a large amount of computation in a high dimensional space, the optimized distance induced by the OT solution, known as the Wasserstein distance (Panaretos and Zemel 2019), Monge–Kantorovich distance (Brezis 2018) and Earth Mover's distance (Rubner et al. 2000), has been treated as a target being analyzed in various aspects such as image processing (Rabin and Papadakis 2015; Ferradans et al. 2014), pattern analysis (Zhao and Zhou 2018; Cuturi 2013; Miller and Lent 2016) and domain adaption (Courty et al. 2016; Maman et al. 2019; Yair et al. 2019).

The OT-based method for comparing two probability densities and generative models are vital in machine learning research where data are often presented in the form of point clouds, histograms, bags-of-features, or more generally, even manifold-valued data set. In recent years, there has been an increase in the applications of the OT-based methods in machine learning. Bousquet et al. (2017) approached OT-based generative modeling, triggering fruitful research under the variational Bayesian concepts, such as Wassertein GAN (Arjovsky et al. 2017; Gulrajani et al. 2017), Wasserstein Auto-encoders (Tolstikhin et al. 2018; Zhang et al. 2019), and Wasserstein variational inference (Ambrogioni et al. 2018) and their computationally efficient sliced version (Kolouri et al. 2019). Another reason that OT gains its popularity is convexity. As the classic Kantorovich OT problem is a constrained linear programming problem or a convex minimization problem where the minimal value of the transport cost objective function is usually defined as the divergence/distance between two distributions of loads (Peyre and Cuturi 2019), or the cost associated with the transportation between the source subjects and targets. Therefore, the convex optimization plays an essential role in finding the solutions of OT. The computation of the OT distance can be approached in principle by interior-point methods, and one of the best is from Lee and Sidford (2014).

Although the methods for finding the solutions of OT have been widely investigated in the literature, one of the major problems is that these algorithms are excessively slow in handling large scale OT problems.A great deal of effort have been paid to find more efficient algorithms under the classic OT problem setting with some specifications. For example, a better algorithm (Haker et al. 2004) was proposed in the image registration and wrapping. Under the homogeneous cost assumption, Jacobs and Lèger (2020) proposed a faster back-and-forth algorithm. Another issue with the classic Kantorovich OT formulation is that its solution plan merely relies on a few routes as a result of the sparsity of optimal couplings, and therefore fails to reflect the practical traffic conditions. These issues limit the wider applicability of OT-based distances for large-scale data within the field of machine learning until a regularized transportation plan was introduced by Cuturi (2013). By applying this new method (regularized OT), we are not only able to reduce the sparsity in the transportation plan, but also speed up the Sinkhorn

algorithm with a linear convergence (Knight 2008). The new research (Schmitzer 2019) further improves the stability of the entropy regularized OT problem.

By offering a unique solution, better computational stability compared with the previous algorithms and being underpinned by the Sinkhorn algorithm, the entropy regularization method has successfully delivered OT approaches into modern machine learning aspects (Villani 2009), such as unsupervised learning using Restricted Boltzmann Machines (Montavon et al. 2016), Wasserstein loss function (Frogner et al. 2015), computer graphics (Solomon et al. 2015) and discriminant analysis (Flamary et al. 2018). Other algorithms that aim for high calculation speed in the area of big data have also been explored, such as the stochastic gradient-based algorithms (Genevay et al. 2016) and fast methods to compute Wasserstein barycenters (Cuturi and Doucet 2014). Altschuler et al. (2017) proposed the Greenkhorn algorithm, a greedy variant of the Sinkhorn algorithm that updates the rows and columns which violate most of the constraints.

In order to meet the requirements of various practical situations, many works have been done to define suitable regularizations. For newly introduced regularizations, Dessein et al. (2018) extended the regularization in terms of convex functions. To apply OT to power functions, the Tsallis Regularized Optimal Transport (trot) distance problem was introduced in Su and Hua (2017). Furthermore, in order to involve OT into series data, the order-preserving Wassertein distance with its regularizor was developed in Courty et al. (2016). In addition, to maintain the locality in OT-assisted domain adaption, the Laplacian regularization was also proposed in Courty et al. (2016). While entropy-based regularizations have achieved great success in terms of calculation efficiency, those problems without such regularization are still challenging. For example, to solve a Laplacian regularized OT problem, Courty *et al.* proposed a generalized conditional gradient algorithm, which is a variant of the classic conditional gradient algorithm (Bertsekas 1999). In this paper, we shall compare the experimental results of several entropy and non-entropy regularized OT problems based on previous studies and the new manifold optimization algorithm proposed in Sect. 4.

Non-entropy regularized OT problems arise the question about the development of a uniform and generalized method that is capable of efficiently and accurately calculating all sort of regularized OT problems. To answer this question, we first consider that all OT problems are constrained optimization problems on the transport plane space, namely the set of polytope (Peyre and Cuturi 2019). Such constrained problems can be regarded as the unconstrained problem on a specific manifold with certain constraints. The well-defined Riemannian optimization can provide better performance than the original constrained problem with the advantage of treating lower dimensional manifold as a new search space. Consequentially, those fundamental numerical iterative algorithms, such as the Riemannian gradient descent (RGD) and Riemannian trust region (RTR), can naturally solve the OT problems, achieving convergence under mild conditions.

The main purpose of this paper is to propose a manifold-based framework to optimize the transportation polytope in which the related Riemannian geometry will be explored. The "Coupling Matrix Manifold" provides an innovative method for solving OT problems under the framework of manifold optimization. The research on the coupling matrix manifold has rooted in our earlier paper (Sun et al. 2016) in which the so-called multinomial manifold was explored in the context of tensor clustering. The optimization on multinomial manifolds has successfully been applied to several density learning tasks (Hong and Gao 2015; Hong et al. 2015; Hong and Gao 2018). More recently, Douik and Hassibi (2019) explored the manifold geometrical structure and the related convex optimization algorithms on three types of manifolds constructed by three types of matrices, namely the

doubly stochastic matrices, symmetric stochastic matrices and positive stochastic matrices. The CMM introduced in this paper can be regarded as the generalization of their doubly positive stochastic manifolds. According to the mathematical and experimental results, our CMM paves the way to solve all types of OT problems,including regularized (commonly solved by using the famous Sinkhorn algorithm) and non-regularized (previously solved by using the classic linear programming algorithm) under the manifold optimization framework (Absil et al. 2008), thus providing a form of unconstrained optimization on manifold to exploit geometry information with higher efficiency.

In summary, the main contribution of this paper are three folds.

1. We define the Coupling Matrix Manifold. We explore the geometric properties of this manifold, including its tangent space, the projection mapping onto the tangent space, a numerically efficient retraction mapping and the calculation of Riemann gradient and Riemann Hessian on the manifold.
2. Following the framework of optimization on manifolds, we formulate the Riemann optimization algorithm on the Coupling Matrix Manifold, so that most OT related optimization problems can be solved in a consistent way.
3. We compare our newly presented algorithm with the existing algorithms in literature for several state-of-the-art OT models.

The remainder of the paper is organized as follows. Section 2 introduces CMM and its Riemannian geometry,including the tangent space, Riemannian gradient, Riemannian Hessian, and Retraction operator, all the ingredients for the Riemannian optimization algorithms. In Sect. 3, we review several OT problems with different regularizations from other studies. These regularization problems will be then converted into the optimization problem on CMM so that the Riemannian version of optimization algorithms (RGD and RTR) can be applied. In Sect. 4, we will conduct several numerical experiments to demonstrate the performance of the new Riemannian algorithms and compare the results with classic algorithms (i.e. Sinkhorn algorithm). Finally Sect. 5 concludes the paper with several recommendations for future research and applications.

2 Coupling matrix manifolds—CMM

In this section, we introduce the CMM and Riemannian geometry of this manifold in order to solve any generic OT problems (Peyre and Cuturi 2019) under the framework of CMM optimization (Absil et al. 2008).

Throughout this paper, we use a bold lower case letter for a vector $\mathbf{x} \in \mathbb{R}^d$, a bold upper case letter for a matrix $\mathbf{X} \in \mathbb{R}^{n\times m}$, and a calligraphy letter for a manifold $\mathscr{M}$. The embedded matrix manifold $\mathscr{M}$ is a smooth subset of vector space $\mathscr{E}$ embedded in the matrix space $\mathbb{R}^{n\times m}$. For any $\mathbf{X} \in \mathscr{M}$, $T_{\mathbf{X}}\mathscr{M}$ is the tangent space of the manifold $\mathscr{M}$ at $\mathbf{X}$ (Absil et al. 2008). $\mathbf{0}_d$ and $\mathbf{1}_d \in \mathbb{R}^d$ are the d-dimensional vectors of zeros and ones, respectively, and $\mathbb{R}^{n\times m}_+$ is the set of all $n \times m$ matrices with real and positive elements.

2.1 The definition of a manifold

Definition 1 Two vectors $\mathbf{p} \in \mathbb{R}^n_+$ and $\mathbf{q} \in \mathbb{R}^m_+$ are coupled if $\mathbf{p}^T\mathbf{1}_n = \mathbf{q}^T\mathbf{1}_m$. A matrix $\mathbf{X} \in \mathbb{R}^{n\times m}_+$ of positive entries is called a coupling matrix of the coupled vectors $\mathbf{p}$ and $\mathbf{q}$ if

$\mathbf{X}\mathbf{1}_m = \mathbf{p}$ and $\mathbf{X}^T\mathbf{1}_n = \mathbf{q}$. The set of all the coupling matrices for the given coupled $\mathbf{p}$ and $\mathbf{q}$ is denoted by

$$\mathbb{C}_n^m(\mathbf{p}, \mathbf{q}) = \{\mathbf{X} \in \mathbb{R}_+^{n\times m} : \mathbf{X}\mathbf{1}_m = \mathbf{p} \text{ and } \mathbf{X}^T\mathbf{1}_n = \mathbf{q}\}. \tag{1}$$

The open subset defined in (1) is indeed a linear manifold. We will introduce an appropriate new metric to make it a Riemannian manifold in the following for the purpose of manifold optimization.

Remark 1 The coupling condition

$$\mathbf{p}^T\mathbf{1}_n = \mathbf{q}^T\mathbf{1}_m \tag{2}$$

is vital in this paper as this condition ensures a non-empty transportation polytope so that the manifold optimization process can be naturally employed. This condition is checked in Lemma 2.2 of De Loera and Kim (2014), and the proof of this lemma is based on the north-west corner rule algorithm described in Queyranne and Spieksma (2009).

Remark 2 The defined space $\mathbb{C}_n^m(\mathbf{p}, \mathbf{q})$ of positive plans is a subset of the classic transport plan space (or polytope)

$$\mathbb{P}_n^m(\mathbf{p}, \mathbf{q}) = \{\mathbf{X} \in \mathbb{R}_0^{n\times m} : \mathbf{X}\mathbf{1}_m = \mathbf{p} \text{ and } \mathbf{X}^T\mathbf{1}_n = \mathbf{q}\},$$

where each entry of a plan $\mathbf{X}$ in $\mathbb{P}_n^m(\mathbf{p}, \mathbf{q})$ is non-negative. In practice, this constraint on $\mathbb{C}_n^m(\mathbf{p}, \mathbf{q})$ may pull the solution plan from being sparsity while the classic linear programming algorithm for the OT problem restricts the entries of a plan to be non-negative, resulting in zero entries, i.e., sparsity. Given the practical requirement of non-sparsity, the entropy regularization is used to enforce such non-sparsity.

Proposition 1 *The subset $\mathbb{C}_n^m(\mathbf{p}, \mathbf{q})$ forms a smooth manifold of dimension $(n-1)(m-1)$ in its embedding space $\mathbb{R}_+^{n\times m}$, named as the Coupling Matrix Manifold.*

Proof Define a mapping $F : \mathbb{R}_+^{n\times m} \to \mathbb{R}^{n+m}$ by

$$F(\mathbf{X}) = \begin{bmatrix} \mathbf{X}\mathbf{1}_m - \mathbf{p} \\ \mathbf{X}^T\mathbf{1}_n - \mathbf{q} \end{bmatrix}.$$

Hence

$$\mathbb{C}_n^m(\mathbf{p}, \mathbf{q}) = F^{-1}(\mathbf{0}_{n+m}).$$

Clearly $DF(\mathbf{X})$ is a linear mapping from $\mathbb{R}_+^{n\times m}$ to $\mathbb{R}^{n+m}$ with

$$DF(\mathbf{X})[\Delta\mathbf{X}] = \begin{bmatrix} \Delta\mathbf{X}\mathbf{1}_m \\ \Delta\mathbf{X}^T\mathbf{1}_n \end{bmatrix}.$$

Hence the null space of $DF(\mathbf{X})$ is

$$\mathbf{K} = \{\Delta\mathbf{X} : \Delta\mathbf{X}\mathbf{1}_m = \mathbf{0}_n, \Delta\mathbf{X}^T\mathbf{1}_n = \mathbf{0}_m\}.$$

As there are only $n + m - 1$ linearly independent constraints among $\Delta\mathbf{X}\mathbf{1}_m = \mathbf{0}_n$, and $\Delta\mathbf{X}^T\mathbf{1}_n = \mathbf{0}_m$, the rank of the null space is $nm - n - m + 1 = (n-1)(m-1)$. Hence the

dimension of the range will be $n + m - 1$. According to the sub-immersion theorem [Proposition 3.3.4 in Absil et al. (2008)], the dimension of the manifold $\mathbb{C}_n^m(\mathbf{p}, \mathbf{q})$ is $(n-1)(m-1)$.

This completes the proof. □

Several special cases of the coupling matrix manifolds that have been explored recently are as follows:

Remark 3 When both $\mathbf{p}$ and $\mathbf{q}$ are discrete distributions, i.e., $\mathbf{p}^T\mathbf{1}_n = \mathbf{q}^T\mathbf{1}_m = 1$ which are naturally coupled. In this case, we call $\mathbb{C}_n^m(\mathbf{p}, \mathbf{q})$ the double probabilistic manifold, denoted by

$$\mathbb{DP}_n^m(\mathbf{p}, \mathbf{q}) = \{\mathbf{X} \in \mathbb{R}_+^{n\times m} : \mathbf{X}\mathbf{1}_m = \mathbf{p}, \mathbf{X}^T\mathbf{1}_n = \mathbf{q} \text{ and } \mathbf{p}^T\mathbf{1}_n = \mathbf{q}^T\mathbf{1}_m = 1\}$$

and the coupling condition becomes:

$$\mathbf{p}^T\mathbf{1}_n = \mathbf{q}^T\mathbf{1}_m = 1$$

Remark 4 The doubly stochastic multinomial manifold (Douik and Hassibi 2019): This manifold is the special case of $\mathbb{C}_n^m(\mathbf{p}, \mathbf{q})$ with $n = m$ and $\mathbf{p} = \mathbf{q} = \mathbf{1}_n$, e.g.

$$\mathbb{DP}_n = \{\mathbf{X} \in \mathbb{R}_+^{n\times n} : \mathbf{X}\mathbf{1}_n = \mathbf{1}_n, \mathbf{X}^T\mathbf{1}_n = \mathbf{1}_n\}.$$

$\mathbb{DP}_n$ can be regarded as the two-dimensional extension of the multinomial manifold introduced in Sun et al. (2016), defined as

$$\mathbb{P}_n^m = \{\mathbf{X} \in \mathbb{R}_+^{n\times m} : \mathbf{X}\mathbf{1}_m = \mathbf{1}_n\}.$$

2.2 The tangent space and its metric

From now on, we only consider the coupling matrix manifold $\mathbb{C}_n^m(\mathbf{p}, \mathbf{q})$ where $\mathbf{p}$ and $\mathbf{q}$ are a pair of coupled vectors. For any coupling matrix $\mathbf{X} \in \mathbb{C}_n^m(\mathbf{p}, \mathbf{q})$, the tangent space $T_{\mathbf{X}}\mathbb{C}_n^m(\mathbf{p}, \mathbf{q})$ is given by the following proposition.

Proposition 2 *The tangent space $T_{\mathbf{X}}\mathbb{C}_n^m(\mathbf{p}, \mathbf{q})$ can be calculated as*

$$T_{\mathbf{X}}\mathbb{C}_n^m(\mathbf{p}, \mathbf{q}) = \{\mathbf{Y} \in \mathbb{R}^{n\times m} : \mathbf{Y}\mathbf{1}_m = \mathbf{0}_n,\ \mathbf{Y}^T\mathbf{1}_n = \mathbf{0}_m\} \tag{3}$$

and its dimension is $(n-1)(m-1)$.

Proof It is easy to prove Proposition 2 by differentiating the constraint conditions. We omit this.

Also it is clear that $\mathbf{Y}\mathbf{1}_m = \mathbf{0}_n$ and $\mathbf{Y}^T\mathbf{1}_n = \mathbf{0}_m$ consist of $m + n$ equations where only $m + n - 1$ conditions are in general independent because $\sum_{ij} Y_{ij} = \mathbf{1}_n^T\mathbf{Y}\mathbf{1}_m = 0$. Hence the dimension of the tangent space is $nm - n - m + 1 = (n-1)(m-1)$. The proof is completed. □

Following Sun et al. (2016); Douik and Hassibi (2019), we still use the Fisher information as the Riemannian metric g on the tangent space $T_{\mathbf{X}}\mathbb{C}_n^m(\mathbf{p},\mathbf{q})$. The motivation of using the Fisher information metric is due to the characteristic of $\mathbf{X}$ in definition (1): $\{\mathbf{X} \in \mathbb{R}_+^{n\times m} : \mathbf{X}\mathbf{1}_m = \mathbf{p} \text{ and } \mathbf{X}^T\mathbf{1}_n = \mathbf{q}\}$ such that the set consists of discrete distributions with fixed marginal source and target distributions (vectors) and the fact that the Fisher information metric is a widely used metric (i.e. Riemannian metric) on the manifold of the probability distributions (Amari and Nagaoka 2000, 2007). Here, for any two tangent vectors $\xi_{\mathbf{X}}, \eta_{\mathbf{X}} \in T_{\mathbf{X}}\mathbb{C}_n^m(\mathbf{p},\mathbf{q})$, the Fisher information metric is defined as

$$g(\xi_{\mathbf{X}}, \eta_{\mathbf{X}}) = \sum_{ij} \frac{(\xi_{\mathbf{X}})_{ij}(\eta_{\mathbf{X}})_{ij}}{X_{ij}} = \mathrm{Tr}((\xi_{\mathbf{X}} \oslash \mathbf{X})(\eta_{\mathbf{X}})^T) \tag{4}$$

where the operator $\oslash$ means the element-wise division of two matrices in the same size.

Remark 5 Equivalently we may use the normalized Riemannian metric as follows

$$g(\xi_{\mathbf{X}}, \eta_{\mathbf{X}}) = (\mathbf{p}^T\mathbf{1}_n) \sum_{ij} \frac{(\xi_{\mathbf{X}})_{ij}(\eta_{\mathbf{X}})_{ij}}{X_{ij}}$$

As one of building blocks for the optimization algorithms on manifolds, we consider how a matrix of size $n \times m$ can be orthogonally projected onto the tangent space $T_{\mathbf{X}}\mathbb{C}_n^m(\mathbf{p},\mathbf{q})$ under its Riemannian metric g.

Theorem 1 *The orthogonal projection from $\mathbb{R}^{n\times m}$ to $T_{\mathbf{X}}\mathbb{C}_n^m(\mathbf{p},\mathbf{q})$ takes the following form*

$$\Pi_{\mathbf{X}}(\mathbf{Y}) = \mathbf{Y} - (\alpha\mathbf{1}_m^T + \mathbf{1}_n\beta^T) \odot \mathbf{X}, \tag{5}$$

where the symbol $\odot$ denotes the Hadamard product, and α and β are given by

$$\alpha = (\mathbf{P} - \mathbf{X}\mathbf{Q}^{-1}\mathbf{X})^{-1}(\mathbf{Y}\mathbf{1}_m - \mathbf{X}\mathbf{Q}^{-1}\mathbf{Y}^T\mathbf{1}_n) \in \mathbb{R}^n \tag{6}$$

$$\beta = \mathbf{Q}^{-1}(\mathbf{Y}^T\mathbf{1}_n - \mathbf{X}^T\alpha) \in \mathbb{R}^m \tag{7}$$

where $\mathbf{P} = \mathrm{diag}(\mathbf{p})$ *and* $\mathbf{Q} = \mathrm{diag}(\mathbf{q})$.

Proof We only present a simple sketch of the proof here. First, it is easy to verify that for any vectors $\alpha \in \mathbf{R}^n$ and $\beta \in \mathbf{R}^m$, $\mathbf{N} = (\alpha\mathbf{1}_m^T + \mathbf{1}_n\beta^T) \odot \mathbf{X}$ is orthogonal to the tangent space $T_{\mathbf{X}}\mathbb{C}_n^m(\mathbf{p},\mathbf{q})$. This is because for any $\mathbf{S} \in T_{\mathbf{X}}\mathbb{C}_n^m(\mathbf{p},\mathbf{q})$, we have the following inner product induced by g,

$$\begin{aligned}\langle \mathbf{N}, \mathbf{S}\rangle_{\mathbf{X}} &= \mathrm{Tr}((\mathbf{N} \oslash \mathbf{X})\mathbf{S}^T) = \mathrm{Tr}((\alpha\mathbf{1}_m^T + \mathbf{1}_n\beta^T)\mathbf{S}^T) \\ &= \alpha^T\mathbf{S}\mathbf{1}_m + \beta^T\mathbf{S}^T\mathbf{1}_n = 0.\end{aligned}$$

By counting the dimension of the tangent space, we conclude that, for any $\mathbf{Y} \in \mathbf{R}^{n\times m}$ and $\mathbf{X} \in \mathbb{C}_n^m(\mathbf{p},\mathbf{q})$, there exist α and β such that the following orthogonal decomposition is valid

$$\mathbf{Y} = \Pi_{\mathbf{X}}(\mathbf{Y}) + (\alpha\mathbf{1}_m^T + \mathbf{1}_n\beta^T) \odot \mathbf{X}$$

Hence

$$\mathbf{Y}\mathbf{1}_m = ((\alpha \mathbf{1}_m^T + \mathbf{1}_n \beta^T) \odot \mathbf{X})\mathbf{1}_m$$

By direct element manipulation, we have

$$\mathbf{Y}\mathbf{1}_m = \mathbf{P}\alpha + \mathbf{X}\beta.$$

Similarly

$$\mathbf{Y}^T\mathbf{1}_n = \mathbf{X}^T\alpha + \mathbf{Q}\beta.$$

From the second equation we can express β in terms of α as

$$\beta = \mathbf{Q}^{-1}(\mathbf{Y}^T\mathbf{1}_n - \mathbf{X}^T\alpha)$$

Taking this equation into the first equation gives

$$\mathbf{Y}\mathbf{1}_m = (\mathbf{P} - \mathbf{X}\mathbf{Q}^{-1}\mathbf{X}^T)\alpha + \mathbf{X}\mathbf{Q}^{-1}\mathbf{Y}^T\mathbf{1}_n$$

This gives both (6) and (7). The proof is completed. □

Remark 6 For numerical stability, we can replace the inverse $(\mathbf{P} - \mathbf{X}\mathbf{Q}^{-1}\mathbf{X})^{-1}$ in (6) with its pseudo-inverse $(\mathbf{P} - \mathbf{X}\mathbf{Q}^{-1}\mathbf{X})^{+}$.

2.3 Riemannian gradient and retraction

The classical gradient descent method can be extended to the case of optimization on manifold with the aid of the so-called Riemannian gradient. As the coupling matrix manifold is embedded in the Enclidean space, the Riemannian gradient can be calculated via projecting the Euclidean gradient onto its tangent space. Given the Riemannian metric which is defined in (4), we can immediately formulate the following lemma, see Sun et al. (2016) and Douik and Hassibi (2019).

Lemma 1 *Suppose that $f(\mathbf{X})$ is a real-valued smooth function defined on $\mathbb{C}_n^m(\mathbf{p}, \mathbf{q})$ with its Euclidean gradient* $\mathrm{Grad}f(\mathbf{X})$*, then the Riemannian gradient* $\mathrm{grad}f(\mathbf{X})$ *can be calculated as*

$$\mathrm{grad}f(\mathbf{X}) = \Pi_{\mathbf{X}}(\mathrm{Grad}f(\mathbf{X}) \odot \mathbf{X}). \tag{8}$$

Proof As $Df(\mathbf{X})[\xi_{\mathbf{X}}]$, the directional derivative of f along any tangent vector $\xi_{\mathbf{X}}$, according to the definition of Riemannian gradient, for the metric $g(\cdot, \cdot)$ in (4) we have:

$$g(\mathrm{grad}f(\mathbf{X}), \xi_{\mathbf{X}}) = Df(\mathbf{X})[\xi_{\mathbf{X}}] = \langle \mathrm{Grad}f(\mathbf{X}), \xi_{\mathbf{X}} \rangle \tag{9}$$

where the right equality comes from the definition of Euclidean gradient $\mathrm{Grad}f(\mathbf{X})$ with the classic Euclidean metric $\langle \cdot, \cdot \rangle$. Clearly we have

$$\langle \mathrm{Grad}f(\mathbf{X}), \xi_{\mathbf{X}} \rangle = g(\mathrm{Grad}f(\mathbf{X}) \odot \mathbf{X}, \xi_{\mathbf{X}}) \tag{10}$$

where $g(\mathrm{Grad}f(\mathbf{X}) \odot \mathbf{X}, \xi_{\mathbf{X}})$ can be simply calculated according to the formula in (4), although $\mathrm{Grad}f(\mathbf{X}) \odot \mathbf{X}$ is not in the tangent space $T_{\mathbf{X}}\mathbb{C}_n^m(\mathbf{p}, \mathbf{q})$. Considering its orthogonal decomposition according to the tangent space, we shall have

$$\mathrm{Grad}f(\mathbf{X}) \odot \mathbf{X} = \Pi_{\mathbf{X}}(\mathrm{Grad}f(\mathbf{X}) \odot \mathbf{X}) + \mathbf{Q} \tag{11}$$

where $\mathbf{Q}$ is the orthogonal complement satisfying $g(\mathbf{Q}, \xi_{\mathbf{X}}) = 0$ for any tangent vector $\xi_{\mathbf{X}}$. Taking (11) into (10) and combining it with (9) gives

$$Df(\mathbf{X})[\xi_{\mathbf{X}}] = g(\Pi_{\mathbf{X}}(\text{Grad}f(\mathbf{X}) \odot \mathbf{X}), \xi_{\mathbf{X}}).$$

Hence

$$\text{grad}f(\mathbf{X}) = \Pi_{\mathbf{X}}(\text{Grad}f(\mathbf{X}) \odot \mathbf{X}).$$

This completes the proof. □

As an important part of the manifold gradient descent process, retraction function retracts a tangent vector back to the manifold (Absil et al. 2008). For Euclidean submanifolds, the simplest way to define a retraction is

$$R_{\mathbf{X}}(\xi_{\mathbf{X}}) = \mathbf{X} + \xi_{\mathbf{X}}$$

In our case, to ensure $R_{\mathbf{X}}(\xi_{\mathbf{X}}) \in \mathbb{C}_n^m(\mathbf{p}, \mathbf{q})$, $\xi_{\mathbf{X}}$ should be in the smaller neighbourhood of $\mathbf{0}$ particularly when $\mathbf{X}$ has smaller entries. This will result an inefficient descent optimization process. To provide a new retraction with high efficiency, following Sun et al. (2016), Douik and Hassibi (2019), we define P as the projection from the set of element-wise positive matrices $\mathbb{R}_+^{n\times m}$ onto the manifold $\mathbb{C}_n^m(\mathbf{p}, \mathbf{q})$ under the Euclidean metric, that is,

$$P(\mathbf{M}) = \underset{\mathbf{P}\in\mathbb{C}_n^m(\mathbf{p},\mathbf{q})}{\arg\min} \|\mathbf{P} - \mathbf{M}\|_F^2.$$

This projection may not be unique because of the openness of $\mathbb{C}_n^m(\mathbf{p}, \mathbf{q})$. Here the following lemma offers one such projection through an algorithm.

Lemma 2 *For any matrix* $\mathbf{M} \in \mathbb{R}_+^{n\times m}$, *there exist two diagonal scaling matrices* $\mathbf{D}_1 = \text{diag}(\mathbf{d}_1) \in \mathbb{R}_+^{n\times n}$ *and* $\mathbf{D}_2 = \text{diag}(\mathbf{d}_2) \in \mathbb{R}_+^{m\times m}$ *such that*

$$P(\mathbf{M}) = \mathbf{D}_1\mathbf{M}\mathbf{D}_2 \in \mathbb{C}_n^m(\mathbf{p}, \mathbf{q})$$

where both $\mathbf{D}_1$ *and* $\mathbf{D}_2$ *can be determined by the extended Sinkhorn–Knopp algorithm* (Peyre and Cuturi 2019).

The Sinkhorn–Knopp algorithm is specified in Algorithm 1 below, which implements the projection P in Lemma 2.

Algorithm 1 The Sinkhorn-Knopp Algorithm

Input: $\mathbf{M} \in \mathbb{R}_+^{n\times m}$, $\mathbf{p} \in \mathbb{R}_+^n$ and $\mathbf{q} \in \mathbb{R}_+^m$, a tolerance $\varepsilon = 1e-10$ and the number of maximal iteration T
Output: $\mathbf{D}_1$ and $\mathbf{D}_2$

Initializing
  $\mathbf{d}_1 = \mathbf{q} \oslash (\mathbf{M}^T\mathbf{1}_m)$; $\mathbf{d}_2 = \mathbf{p} \oslash (\mathbf{M}\mathbf{d}_1)$;
while the iteration is less than T **do**
  $\mathbf{d}_1 = \mathbf{q} \oslash (\mathbf{M}^T\mathbf{d}_2)$; $\mathbf{d}_2 = \mathbf{p} \oslash (\mathbf{M}\mathbf{d}_1)$;
  $\mathbf{D}_1 = \text{diag}(\mathbf{d}_1)$ and $\mathbf{D}_2 = \text{diag}(\mathbf{d}_2)$;
  if $\|\mathbf{D}_1\mathbf{M}\mathbf{d}_2 - \mathbf{p}\| < \varepsilon$ and $\|\mathbf{D}_2\mathbf{M}^T\mathbf{d}_1 - \mathbf{q}\| < \varepsilon$ **then**
    break while;
  end if
end while

Based on the projection P, a simple retraction can be defined as

$$R_{\mathbf{X}}(\xi_{\mathbf{X}}) = P(\mathbf{X} + \xi_{\mathbf{X}}).$$

However it may cause numerical uncertainty in the optimization process when both $\mathbf{X}$ and $\xi_{\mathbf{X}}$ contains smaller entries. Instead we define the following retraction mapping for $\mathbb{C}_n^m(\mathbf{p}, \mathbf{q})$

Lemma 3 *Let P be the projection defined in Lemma 2, the mapping $R_{\mathbf{X}} : T_{\mathbf{X}}\mathbb{C}_n^m(\mathbf{p}, \mathbf{q}) \to \mathbb{C}_n^m(\mathbf{p}, \mathbf{q})$ given by*

$$R_{\mathbf{X}}(\xi_{\mathbf{X}}) = P(\mathbf{X} \odot \exp(\xi_{\mathbf{X}} \oslash \mathbf{X}))$$

is a valid retraction on $\mathbb{C}_n^m(\mathbf{p}, \mathbf{q})$. Here $\exp(\cdot)$ is the element-wise exponential function and $\xi_{\mathbf{X}}$ is any tangent vector at $\mathbf{X}$.

Proof We only present a sketch of the proof here. First we need to prove that (i) $R_{\mathbf{X}}(\mathbf{0}) = \mathbf{X}$ and (ii) $\gamma_{\xi_{\mathbf{X}}}(\tau) = R_{\mathbf{X}}(\tau\xi_{\mathbf{X}})$ satisfies $\left.\frac{d\gamma_{\xi_{\mathbf{X}}}(\tau)}{d\tau}\right|_{\tau=0} = \xi_{\mathbf{X}}$.

For (i), it is obvious that $R_{\mathbf{X}}(\mathbf{0}) = \mathbf{X}$ as $P(\mathbf{X}) = \mathbf{X}$ for any $\mathbf{X} \in \mathbb{C}_n^m(\mathbf{p}, \mathbf{q})$.

For (ii),

$$\begin{aligned}\left.\frac{d\gamma_{\xi_{\mathbf{X}}}(\tau)}{d\tau}\right|_{\tau=0} &= \lim_{\tau\to 0}\frac{\gamma_{\xi_{\mathbf{X}}}(\tau) - \gamma_{\xi_{\mathbf{X}}}(0)}{\tau} \\ &= \lim_{\tau\to 0}\frac{P(\mathbf{X} \odot \exp(\tau\xi_{\mathbf{X}} \oslash \mathbf{X})) - \mathbf{X}}{\tau}\end{aligned}$$

As all $\exp(\cdot)$, $\odot$ and $\oslash$ are element-wise operations, the first order approximation of the exponential function gives

$$P(\mathbf{X} \odot \exp(\tau\xi_{\mathbf{X}} \oslash \mathbf{X})) = P(\mathbf{X} + \tau\xi_{\mathbf{X}}) + o(\tau)$$

where $\lim_{\tau\to 0}\frac{o(\tau)}{\tau} = 0$. The next step is to show that $P(\mathbf{X} + \tau\xi_{\mathbf{X}}) \approx \mathbf{X} + \tau\xi_{\mathbf{X}}$ when τ is very small. For this purpose, consider a smaller tangent vector $\Delta\mathbf{X}$ such that $\mathbf{X} + \Delta\mathbf{X} \in \mathbb{R}_+^{n\times m}$. There exist two smaller diagonal matrices $\Delta\mathbf{D}_1 \in \mathbb{R}_+^{n\times n}$ and $\Delta\mathbf{D}_2 \in \mathbb{R}_+^{m\times m}$ that satisfy

$$P(\mathbf{X} + \Delta\mathbf{X}) = (\mathbf{I}_n + \Delta\mathbf{D}_1)(\mathbf{X} + \Delta\mathbf{X})(\mathbf{I}_m + \Delta\mathbf{D}_2)$$

where $\mathbf{I}$ are identity matrices. By ignoring higher order small quantity, we have

$$P(\mathbf{X} + \Delta\mathbf{X}) \approx \mathbf{X} + \Delta\mathbf{X} + \Delta\mathbf{D}_1\mathbf{X} + \mathbf{X}\Delta\mathbf{D}_2.$$

As both $P(\mathbf{X} + \Delta\mathbf{X})$ and $\mathbf{X}$ are on the coupling matrix manifold and $\Delta\mathbf{X}$ is a tangent vector, we have

$$\begin{aligned}\mathbf{p} = P(\mathbf{X} + \Delta\mathbf{X})\mathbf{1}_m &\approx (\mathbf{X} + \Delta\mathbf{X} + \Delta\mathbf{D}_1\mathbf{X} + \mathbf{X}\Delta\mathbf{D}_2)\mathbf{1}_m \\ &\approx \mathbf{p} + \mathbf{0} + \Delta\mathbf{D}_1\mathbf{p} + \mathbf{X}\Delta\mathbf{D}_2\mathbf{1}_m = \mathbf{p} + \mathbf{P}\delta\mathbf{D}_1 + \mathbf{X}\delta\mathbf{D}_2\end{aligned}$$

where $\delta\mathbf{D} = \mathrm{diag}(\Delta\mathbf{D})$ and $\mathbf{P} = \mathrm{diag}(\mathbf{p})$[1]. Hence,

$$\mathbf{P}\delta\mathbf{D}_1 + \mathbf{X}\delta\mathbf{D}_2 \approx \mathbf{0}.$$

[1] For a matrix $\mathbf{M}$, $\mathrm{diag}(\mathbf{M})$ is the vector formed by $\mathbf{M}$'s diagonal elements. For a vector $\mathbf{v}$, the result of $\mathrm{diag}(\mathbf{v})$ is the matrix whose diagonal elements come from $\mathbf{v}$.

Similarly,

$$\mathbf{X}^T\delta\mathbf{D}_1 + \mathbf{Q}\delta\mathbf{D}_2 \approx \mathbf{0}.$$

That is

$$\begin{bmatrix}\mathbf{P} & \mathbf{X}\\ \mathbf{X}^T & \mathbf{Q}\end{bmatrix}\begin{bmatrix}\delta\mathbf{D}_1\\ \delta\mathbf{D}_2\end{bmatrix} \approx \mathbf{0}.$$

Hence $[\delta\mathbf{D}_1^T, \delta\mathbf{D}_2^T]^T$ is in the null space of the above matrix which contains $[\mathbf{1}_n^T, -\mathbf{1}_m^T]^T$. In general, there exists a constant c such that $\delta\mathbf{D}_1 = c\mathbf{1}_n$ and $\delta\mathbf{D}_2 = -c\mathbf{1}_m$ and this gives

$$\Delta\mathbf{D}_1\mathbf{X} + \mathbf{X}\Delta\mathbf{D}_2 = \mathbf{0}.$$

Combining all results obtained above, we have $P(\mathbf{X} + \tau\xi_{\mathbf{X}}) \approx \mathbf{X} + \tau\xi_{\mathbf{X}}$ as τ is sufficiently smaller. Hence, this completes the proof. □

2.4 The Riemannian Hessian

Theorem 2 *Let* Grad$f(\mathbf{X})$ *and* Hess$f(\mathbf{X})[\xi_{\mathbf{X}}]$ *be the Euclidean gradient and Euclidean Hessian, respectively. The Riemennian Hessian* hess$f(\mathbf{X})[\xi_{\mathbf{X}}]$ *can be expressed as*

$$\text{hess}f(\mathbf{X})[\xi_{\mathbf{X}}] = \Pi_{\mathbf{X}}\Big(\dot{\gamma} - \frac{1}{2}(\gamma \odot \xi_{\mathbf{X}}) \oslash \mathbf{X}\Big)$$

with

$$\begin{aligned}
\mu &= (\mathbf{P} - \mathbf{X}\mathbf{Q}^{-1}\mathbf{X}^t)^+\\
\eta &= \text{Grad}f(\mathbf{X}) \odot \mathbf{X}\\
\alpha &= \mu(\eta\mathbf{1}_m - \mathbf{X}\mathbf{Q}^{-1}\eta^T\mathbf{1}_n)\\
\beta &= \mathbf{Q}^{-1}(\eta^T\mathbf{1}_n - \mathbf{X}^T\alpha)\\
\gamma &= \eta - (\alpha\mathbf{1}_m^T + \mathbf{1}_n\beta^T) \odot \mathbf{X}\\
\dot{\mu} &= \mu(\mathbf{X}\mathbf{Q}^{-1}\xi_{\mathbf{X}}^T + \xi_{\mathbf{X}}\mathbf{Q}^{-1}\mathbf{X}^T)\mu\\
\dot{\eta} &= \text{Hess}f(\mathbf{X})[\xi_{\mathbf{X}}] \odot \mathbf{X} + \text{Grad}f(\mathbf{X}) \odot \xi_{\mathbf{X}}\\
\dot{\alpha} &= \dot{\mu}(\eta\mathbf{1}_m - \mathbf{X}\mathbf{Q}^{-1}\eta^T\mathbf{1}_n)\\
&\quad + \mu(\dot{\eta}\mathbf{1}_m - \xi_{\mathbf{X}}\mathbf{Q}^{-1}\eta^T\mathbf{1}_n - \mathbf{X}\mathbf{Q}^{-1}\dot{\eta}^T\mathbf{1}_n)\\
\dot{\beta} &= \mathbf{Q}^{-1}(\dot{\eta}^T\mathbf{1}_n - \xi_{\mathbf{X}}^T\alpha - \mathbf{X}^T\dot{\alpha})\\
\dot{\gamma} &= \dot{\eta} - (\dot{\alpha}\mathbf{1}_m^T + \mathbf{1}_n\dot{\beta}^T) \odot \mathbf{X} - (\alpha\mathbf{1}_m^T + \mathbf{1}_n\beta^T) \odot \xi_{\mathbf{X}}.
\end{aligned}$$

Proof It is well known (Absil et al. 2008) that the Riemannian Hessian can be calculated from the Riemannian connection ∇ and Riemannian gradient via

$$\text{hess}f(\mathbf{X})[\xi_{\mathbf{X}}] = \nabla_{\xi_{\mathbf{X}}}\text{grad}f(\mathbf{X}).$$

Furthermore the connection $\nabla_{\xi_{\mathbf{X}}}\eta_{\mathbf{X}}$ on the submanifold can be given by the projection of the Levi-Civita connection $\overline{\nabla}_{\xi_{\mathbf{X}}}\eta_{\mathbf{X}}$, i.e., $\nabla_{\xi_{\mathbf{X}}}\eta_{\mathbf{X}} = \Pi_{\mathbf{X}}(\overline{\nabla}_{\xi_{\mathbf{X}}}\eta_{\mathbf{X}})$. For the Euclidean space $\mathbb{R}^{n\times m}$

endowed with the Fisher information, with the same approach used in Sun et al. (2016), it can be shown that the Levi-Civita connection is given by

$$\overline{\nabla}_{\xi_{\mathbf{X}}}\eta_{\mathbf{X}} = D(\eta_{\mathbf{X}})[\xi_{\mathbf{X}}] - \frac{1}{2}(\xi_{\mathbf{X}} \odot \eta_{\mathbf{X}}) \oslash \mathbf{X}.$$

Hence,

$$\begin{aligned}\text{hess}f(\mathbf{X})[\xi_{\mathbf{X}}] &= \Pi_{\mathbf{X}}(\overline{\nabla}_{\xi_{\mathbf{X}}}\text{grad}f(\mathbf{X})) \\ &= \Pi_{\mathbf{X}}\Big(D(\text{grad}f(\mathbf{X}))[\xi_{\mathbf{X}}] - \frac{1}{2}(\xi_{\mathbf{X}} \odot \text{grad}f(\mathbf{X})) \oslash \mathbf{X}\Big)\end{aligned}$$

According to Lemma 1, the directional derivative can be expressed as

$$\begin{aligned}D(\text{grad}f(\mathbf{X}))[\xi_{\mathbf{X}}] &= D(\Pi_{\mathbf{X}}(\eta))[\xi_{\mathbf{X}}] \\ &= D(\eta - (\alpha\mathbf{1}_m^T + \mathbf{1}_n\beta^T) \odot \mathbf{X})[\xi_{\mathbf{X}}] \\ &= D(\eta)[\xi_{\mathbf{X}}] - (D(\alpha)[\xi_{\mathbf{X}}]\mathbf{1}_m^T + \mathbf{1}_n D(\beta)[\xi_{\mathbf{X}}]^T) \odot \mathbf{X} \\ &\quad - (\alpha\mathbf{1}_m^T + \mathbf{1}_n\beta^T) \odot \xi_{\mathbf{X}}.\end{aligned}$$

Taking in the expressions for η, α, β and directly computing directional derivatives give all formulate in the theorem. □

3 Riemannian optimization applied to OT problems

In this section, we illustrate the Riemannian optimization in solving various OT problems, starting by reviewing the framework of the optimization on Riemannian manifolds.

3.1 Optimization on manifolds

Early attempts to adapt standard manifold optimization methods were presented by Gabay (1982) in which steepest descent, Newton and qusasi-Newtwon methods were introduced. The second-order geometry related optimization algorithm such as the Riemannian trust region algorithm was proposed in Absil et al. (2008), where the algorithm was applied on some specific manifolds such as the Stiefel and Grassman manifolds.

This paper focuses only on the gradient descent method which is the most widely used optimization method in machine learning.

Suppose that $\mathbb{M}$ is a D-dimensional Riemannian manifold. Let $f : \mathbb{M} \to \mathbb{R}$ be a real-valued function defined on $\mathbb{M}$. Then, the optimization problem on $\mathbb{M}$ has the form

$$\min_{\mathbf{X}\in\mathbb{M}} f(\mathbf{X}).$$

For any $\mathbf{X} \in \mathbb{M}$ and $\xi_{\mathbf{X}} \in T_{\mathbf{X}}\mathbb{M}$, there always exists a geodesic starting at $\mathbf{X}$ with initial velocity $\xi_{\mathbf{X}}$, denoted by $\gamma_{\xi_{\mathbf{X}}}$. With this geodesic the so-called exponential mapping $\exp_{\mathbf{X}} : T_{\mathbf{X}}\mathbb{M} \to \mathbb{M}$ is defined as

$$\exp_{\mathbf{X}}(\xi_{\mathbf{X}}) = \gamma_{\xi_{\mathbf{X}}}(1), \quad \text{for any } \xi_{\mathbf{X}} \in T_{\mathbf{X}}\mathbb{M}.$$

Thus the simplest Riemannian gradient descent (RGD) consists of the following two main steps:

1. Compute the Riemannian gradient of f at the current position $\mathbf{X}^{(t)}$, i.e. $\xi_{\mathbf{X}^{(t)}} = \text{grad}f(\mathbf{X}^{(t)})$;
2. Move in the direction $-\xi_{\mathbf{X}^{(t)}}$ according to $\mathbf{X}^{(t+1)} = \exp_{\mathbf{X}^{(t)}}(-\alpha\xi_{\mathbf{X}^{(t)}})$ with a step-size $\alpha > 0$.

Step 1) is straightforward as the Riemannian gradient can be calculated from the Euclidean gradient according to (8) in Lemma 1. However, it is generally difficult to compute the exponential map effectively as the computational processes require some second-order Riemannian geometrical elements to construct the geodesic, which sometimes is not unique on a manifold point. Therefore, instead of using the exponential map in RGD, an approximated method, namely the retraction map $R_{\mathbf{X}}$ is commonly adopted to replace the exponential mapping $\exp_{\mathbf{X}}$ in Step 2).

For the coupling matrix manifold $\mathbb{C}_n^m(\mathbf{p}, \mathbf{q})$, a retraction mapping has been proposed in Lemma 3. Hence Step 2) in the RGD can be modified by using the computable retraction mapping as follows,

$$\mathbf{X}^{(t+1)} = R_{\mathbf{X}^{(t)}}(-\alpha\xi_{\mathbf{X}^{(t)}}).$$

Hence for any given OT-based optimization problem

$$\min_{\mathbf{X}\in\mathbb{C}_n^m(\mathbf{p},\mathbf{q})} f(\mathbf{X}),$$

conducting the RGD algorithm comes down to computing the Euclidean gradient $\text{Grad}f(\mathbf{X})$. Similarly, formulating the second-order Riemannian optimization algorithms based on Riemannian Hessian, such as Riemannian Newton method and Riemannian trust region method, boils down to calculating the Euclidean Hessian. See Theorem 2.

3.2 Computational complexity of coupling matrix manifold optimization

In this section we give a simple complexity analysis on optimizing a function defined on the coupling matrix manifold by taking the RGD algorithm as an example. Suppose that we minimize a given objective function $f(\mathbf{X})$ defined on $\mathbb{C}_n^m$. For the sake of simplicity, we consider the case of $m = n$.

In each step of RGD, we first suppose we need the number of flops $E_t(n)$ to calculate the Euclidean gradient $\text{Grad}f(\mathbf{X}^{(t)})$. In most cases shown in the next subsection, we have $E_t(n) = O(n^2)$. Before applying the RGD step, we shall calculate the Riemannian gradient $\text{grad}f(\mathbf{X}^{(t)})$ by the projection according to Lemma 3 which is implemented by the Sinkhorn–Knopp algorithm in Algorithm 1. The complexity of Sinkhorn–Knopp algorithm to have an ϵ-approximate solution is $O(n\log(n)\epsilon^{-3}) = O(n\log(n))$ (Altschuler et al. 2017).

If RGD is coducted T iterations, the overall computational complexity will be

$$O(n\log(n)T) + TE_t(n) = O(n\log(n)T) + O(Tn^2) = O(Tn^2).$$

Remark 7 This complexity is comparable to other optimization algorithms for most OT problems, for example, equivalent to the complexity of the Order-Preserving OT problem

(Su and Hua 2017), see Sect. 3.3.4 below. However as our optimization algorithm has sufficiently exploited the geometry of the manifold, the experimental results are much better than other algorithms, as demonstrated in Sect. 4.

Remark 8 Although the Sinkhorn–Knopp algorithm has a complexity of $O(n\log(n))$, it can only be directly applied to solve the entropy regularized OT problem,, see Application Example 2) in Sect. 3.3 below.

3.3 Application examples

As mentioned before, basic Riemannain optimization algorithms are constructed on the Euclidean gradient and Hessian of the objective function. In the first part of our application example, some classic OT problems are presented to illustrate the calculation process for their Riemannian gradient and Hessian.

3.3.1 The classic OT problem

The objective function of the classic OT problem (Peyré et al. 2019) is

$$\min_{\mathbf{X}\in\mathbb{C}_n^m(\mathbf{p},\mathbf{q})} f(\mathbf{X}) = \mathrm{Tr}(\mathbf{X}^T\mathbf{C}) \tag{12}$$

where $\mathbf{C} = [C_{ij}] \in \mathbb{R}^{n\times m}$ is the given cost matrix and $f(\mathbf{X})$ gives the overall cost under the transport plan $\mathbf{X}$. The solution $\mathbf{X}^*$ to this optimization problem is called the transport plan which induces the lowest overall cost $f(\mathbf{X}^*)$. When the cost C is defined by the distance between the source objects and the target objects, the best transport plan X^* assists in defining the so-called Wasserstein distance between the source distribution $\mathbf{p}$ and the target distribution $\mathbf{q}$ by

$$d(\mathbf{p},\mathbf{q}) = \mathrm{Tr}(\mathbf{X}^{*T}\mathbf{C}).$$

Given that problem (12) is indeed a linear programming problem, it is straightforward to solve the problem by the linear programming algorithms. In this paper, we solve the OT problem under the Riemannian optimization framework. Thus, for the classic OT, obviously the Euclidean gradient and Hessian can be easily computed as:

$$\mathrm{Grad}f(\mathbf{X}) = \mathbf{C}$$

and

$$\mathrm{Hess}f(\mathbf{X})[\xi] = \mathbf{0}.$$

3.3.2 The entropy regularized OT problem

To enforce the non-sparse OT plan, the entropy regularized OT problem was proposed (Peyre and Cuturi 2019). It takes the following form,

$$\min_{\mathbf{X}\in\mathbb{C}_n^m(\mathbf{p},\mathbf{q})} f(\mathbf{X}) = \mathrm{Tr}(\mathbf{X}^T\mathbf{C}) - \lambda\mathbf{H}(\mathbf{X}),$$

where $\mathbf{H}(\mathbf{X})$ is the discrete entropy of the coupling matrix and is defined by:

$$\mathbf{H}(\mathbf{X}) \triangleq -\sum_{ij} \mathbf{X}_{ij}(\log(\mathbf{X}_{ij})).$$

In terms of matrix operation, $\mathbf{H}(\mathbf{X})$ has the form

$$\mathbf{H}(\mathbf{X}) = -\mathbf{1}_n^T(\mathbf{X} \odot \log(\mathbf{X}))\mathbf{1}_m$$

where log applies to each element of the matrix. The minimization is a strictly convex optimization process, and for $\lambda > 0$ the solution $\mathbf{X}^*$ is unique and has the form:

$$\mathbf{X}^* = \text{diag}(\boldsymbol{\mu})\mathbf{K}\text{diag}(\boldsymbol{\nu})$$

where $\mathbf{K} = e^{\frac{-C}{\lambda}}$ is computed entry-wisely (Peyre and Cuturi 2019), and $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ are obtained by the Sinkhorn–Knopp algorithm.

Now, for objective function

$$f(\mathbf{X}) = \text{Tr}(\mathbf{X}^T\mathbf{C}) - \lambda\mathbf{H}(\mathbf{X}),$$

one can easily check that the Euclidean gradient is

$$\text{Grad}f(\mathbf{X}) = \mathbf{C} + \lambda(\mathbb{I} + \log(\mathbf{X})),$$

where $\mathbb{I}$ is a matrix of all 1s in size $n \times m$, and the Euclidean Hessian is, in terms of mapping differential, given by

$$\text{Hess}f(\mathbf{X})[\boldsymbol{\xi}] = \lambda(\boldsymbol{\xi} \oslash \mathbf{X}).$$

3.3.3 The power regularization for OT problem

Dessein et al. (2018) further extended the regularization to

$$\min_{\mathbf{X} \in \mathbb{C}_n^n(\mathbf{p},\mathbf{q})} \text{Tr}(\mathbf{X}^T\mathbf{C}) + \lambda\phi(\mathbf{X})$$

where ϕ is an appropriate convex function. As an example, we consider the squared regularization proposed by Essid and Solomon (2018)

$$\min_{\mathbf{X} \in \mathbb{C}_n^n(\mathbf{p},\mathbf{q})} f(\mathbf{X}) = \text{Tr}(\mathbf{X}^T\mathbf{C}) + \lambda \sum_{ij} X_{ij}^2$$

and we apply a zero truncated operator in the manifold algorithm. It is then straightforward to prove that

$$\text{Grad}f(\mathbf{X}) = \mathbf{C} + 2\lambda\mathbf{X}$$

and

$$\text{Hess}f(\mathbf{X})[\boldsymbol{\xi}] = 2\lambda\boldsymbol{\xi}.$$

The Tsallis Regularized Optimal Transport is used in Muzellec et al. (2017) to define trot distance which comes with the following regularization problem

$$\min_{\mathbf{X} \in \mathbb{C}_n^n(\mathbf{p},\mathbf{q})} f(\mathbf{X}) = \text{Tr}(\mathbf{X}^T\mathbf{C}) - \lambda\frac{1}{1-q}\sum_{ij}(X_{ij}^q - X_{ij}).$$

For the sake of convenience, we denote $\mathbf{X}^q := [X_{ij}^q]_{i=1,j=1}^{n,m}$ for any given constant $q > 0$. Then we have

$$\text{Grad}f(\mathbf{X}) = \mathbf{C} - \frac{\lambda}{1-q}(q\mathbf{X}^{q-1} - \mathbb{I})$$

and

$$\text{Hess}f(\mathbf{X})[\xi] = q\lambda\left[\mathbf{X}^{q-2} \odot \xi\right].$$

3.3.4 The order-preserving OT problem

The order-preserving OT problem is proposed in Su and Hua (2017) and is adopted by Su and Wu (2019) for learning distance between sequences. This learning process takes the local order of temporal sequences and the learned transport defines a flexible alignment between two sequences. Thus, the optimal transport plan only assigns large loads to the most similar instance pairs of the two sequences.

For sequences $\mathbf{U} = (\mathbf{u}_1, ..., \mathbf{u}_n)$ and $\mathbf{V} = (\mathbf{v}_1, ..., \mathbf{v}_m)$ in the respective given orders, the distance matrix between them is

$$\mathbf{C} = [d(\mathbf{u}_i, \mathbf{v}_j)^2]_{i=1,j=1}^{n,m}.$$

Define an $n \times m$ matrix (distance between orders)

$$\mathbf{D} = \left[\frac{1}{\left(\frac{i}{n} - \frac{j}{m}\right)^2 + 1}\right]$$

and the (exponential) similarity matrix

$$\mathbf{P} = \frac{1}{\sigma\sqrt{2\pi}}\left[\exp\left\{-\frac{l(i,j)^2}{2\sigma^2}\right\}\right]$$

where $\sigma > 0$ is the scaling factor and

$$l(i,j) = \left|\frac{\frac{i}{n} - \frac{j}{m}}{\sqrt{\frac{1}{n^2} + \frac{1}{m^2}}}\right|.$$

The (squared) distance between sequences $\mathbf{U}$ and $\mathbf{V}$ is given by

$$d^2(\mathbf{U}, \mathbf{V}) = \text{Tr}(\mathbf{C}^T\mathbf{X}^*) \tag{13}$$

where the optimal transport plan $\mathbf{X}^*$ is the solution to the following order-preserving regularized OT problem

$$\mathbf{X}^* = \underset{\mathbf{X} \in \mathbb{C}_n^m(\mathbf{p},\mathbf{q})}{\arg\min} f(\mathbf{X}) = \text{Tr}(\mathbf{X}^T(\mathbf{C} - \lambda_1\mathbf{D})) + \lambda_2 \text{KL}(\mathbf{X}||\mathbf{P})$$

where the KL-divergence is defined as

$$\mathrm{KL}(\mathbf{X}||\mathbf{P}) = \sum_{ij} X_{ij}(\log(X_{ij}) - \log(P_{ij}))$$

and specially $\mathbf{p} = \frac{1}{n}\mathbf{1}_n$ and $\mathbf{q} = \frac{1}{m}\mathbf{1}_m$ are uniform distributions. Hence

$$\mathrm{Grad}f(\mathbf{X}) = (\mathbf{C} - \lambda_1\mathbf{D}) + \lambda_2(\mathbb{I} + \log(\mathbf{X}) - \log(\mathbf{P}))$$

and

$$\mathrm{Hess}f(\mathbf{X})[\boldsymbol{\xi}] = \lambda_2(\boldsymbol{\xi} \oslash \mathbf{X}).$$

3.3.5 The OT domain adaption problem

OT has also been widely used for solving the domain adaption problems. In this subsection, Courty et al. (2016) formalized two class-based regularized OT problems, namely the group-induced OT (OT-GL) and the Laplacian regularized OT (OT-Laplace). As the OT-Laplace is found to be the best performer for domain adaption, we only apply our coupling matrix manifold optimization to it and thus we summarize its objective function here.

As pointed out in Courty et al. (2016), this regularization aims at preserving the data graph structure during transport. Consider $\mathbf{P}_s = [\mathbf{p}_1^s, \mathbf{p}_2^s, ..., \mathbf{p}_n^s]$ to be the n source data points and $\mathbf{P}_t = [\mathbf{p}_1^t, \mathbf{p}_2^t, ..., \mathbf{p}_m^t]$ the m target data points, both are defined in $\mathbb{R}^d$. Obviously, $\mathbf{P}_s \in \mathbb{R}^{d\times n}$ and $\mathbf{P}_t \in \mathbb{R}^{d\times m}$. The purpose of domain adaption is to transport the source $\mathbf{P}_s$ towards the target $\mathbf{P}_t$ so that the transported source $\widehat{\mathbf{P}}_s = [\widehat{\mathbf{p}}_1^s, \widehat{\mathbf{p}}_2^s, ..., \widehat{\mathbf{p}}_n^s]$ and the target $\mathbf{P}_t$ can be jointly used for other learning tasks.

Now suppose that for the source data we have extra label information $\mathbf{Y}_s = [y_1^s, y_2^s, ..., y_n^s]$. With this label information we sparsify similarities $\mathbf{S}_s = [S_s(i,j)]_{i,j=1}^n \in \mathbb{R}_+^{n\times n}$ among the source data such that $S_s(i,j) = 0$ if $y_i^s \neq y_j^s$ for $i,j = 1,2,...,n$. That is, we define a 0 similarity between two source data points if they do not belong to the same class or do not have the same labels. Then the following regularization is proposed

$$\Omega_c^s(\mathbf{X}) = \frac{1}{n^2}\sum_{i,j=1}^n S_s(i,j)\|\widehat{\mathbf{p}}_i^s - \widehat{\mathbf{p}}_j^s\|_2^2.$$

With a given transport plan $\mathbf{X}$, we can use the barycentric mapping in the target as the transported point for each source point (Courty et al. 2016). When we use the uniform marginals for both source and target and the ℓ_2 cost, the transported source is expressed as

$$\widehat{\mathbf{P}}_s = n\mathbf{X}\mathbf{P}_t. \tag{14}$$

It is easy to verify that

$$\Omega_c^s(\mathbf{X}) = \mathrm{Tr}(\mathbf{P}_t^T\mathbf{X}^T\mathbf{L}_s\mathbf{X}\mathbf{P}_t), \tag{15}$$

where $\mathbf{L}_s = \mathrm{diag}(\mathbf{S}_s\mathbf{1}_n) - \mathbf{S}_s$ is the Laplacian of the graph $\mathbf{S}_s$ and the regularizer $\Omega_c(\mathbf{X})$ is therefore quadratic with respect to $\mathbf{X}$. Similarly when the Laplacian $\mathbf{L}_t$ in the target domain is available, the following symmetric Laplacian regularization is proposed

$$\begin{aligned}\Omega_c(\mathbf{X}) &= (1-\alpha)\mathrm{Tr}(\mathbf{P}_t^T\mathbf{X}^T\mathbf{L}_s\mathbf{X}\mathbf{P}_t) + \alpha\mathrm{Tr}(\mathbf{P}_s^T\mathbf{X}\mathbf{L}_t\mathbf{X}^T\mathbf{P}_s)\\ &= (1-\alpha)\Omega_c^s(\mathbf{X}) + \alpha\Omega_c^t(\mathbf{X}).\end{aligned}$$

When $\alpha = 0$, this goes back to the regularizer $\Omega_c^s(\mathbf{X})$ in (15).

Finally the OT domain adaption is defined by the following Laplacian regularized OT problem

$$\min_{\mathbf{X}\in\mathbb{C}_n^m(\mathbf{1}_n,\mathbf{1}_m)} f(\mathbf{X}) = \mathrm{Tr}(\mathbf{X}^T\mathbf{C}) - \lambda\mathbf{H}(\mathbf{X}) + \frac{1}{2}\eta\Omega_c(\mathbf{X}) \tag{16}$$

Hence the Euclidean gradient and uclidean Hessian are given by

$$\begin{aligned}\mathrm{Grad}f(\mathbf{X}) = \mathbf{C} + \lambda(\mathbb{I} + \log(\mathbf{X})) \\ + \eta((1-\alpha)\mathbf{L}_s\mathbf{X}\mathbf{P}_t\mathbf{P}_t^T + \alpha\mathbf{P}_s\mathbf{P}_s^T\mathbf{X}\mathbf{L}_t).\end{aligned}$$

and

$$\mathrm{Hess}f(\mathbf{X})[\boldsymbol{\xi}] = \lambda(\boldsymbol{\xi} \oslash \mathbf{X}) + \eta((1-\alpha)\mathbf{L}_s\boldsymbol{\xi}\mathbf{P}_t\mathbf{P}_t^T + \alpha\mathbf{P}_s\mathbf{P}_s^T\boldsymbol{\xi}\mathbf{L}_t),$$

respectively.

4 Experimental results and comparisons

In this section, we investigate the performance of our proposed methods. The implementation of the coupling matrix manifold follows the framework of ManOpt Matlab toolbox in http://www.manopt.org from which we call the conjugate gradient descent algorithm as our Riemannian optimization solver in experiments. All experiments are carried out on a laptop computer running on a 64-bit operating system with Intel Core i5-8350U 1.90GHz CPU and 16G RAM with MATLAB 2019a version.

4.1 Synthetic data for the classic OT problem

First of all, we conduct a numerical experiment on a classic OT problem with synthetic data and the performance of the proposed optimization algorithms are demonstrated.

Consider the following source load $\mathbf{p}$ and target load $\mathbf{q}$, and their per unit cost matrix $\mathbf{C}$:

$$\mathbf{p} = \begin{bmatrix}3\\3\\3\\4\\2\\2\\2\\1\end{bmatrix}, \quad \mathbf{q} = \begin{bmatrix}4\\2\\6\\4\\4\end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix}0 & 0 & 1.2 & 2 & 2\\ 2 & 4 & 4 & 4 & 0\\ 1 & 0 & 0 & 0 & 3\\ 0 & 1 & 2 & 1 & 3\\ 1 & 1 & 0 & 1 & 2\\ 2 & 1 & 2 & 0.8 & 3\\ 4 & 0 & 0 & 1 & 1\\ 0 & 1 & 0 & 1 & 3\end{bmatrix}.$$

For this setting, we solve the classic OT problem using the coupling matrix manifold optimization (CMM) and the standard linear programming (LinProg) algorithm, respectively. We visualize the learned transport plan matrices from both algorithms in Fig. 1.

The results reveal that, for the **non-negative** constrained conditions for the entries of transport plan, the linear programming algorithm gives a transportation plan demonstrating sparse patterns, while our coupling matrix manifold imposes the **positivity** constraints, resulting in an relatively denser transportation plan which is preferable to many practical

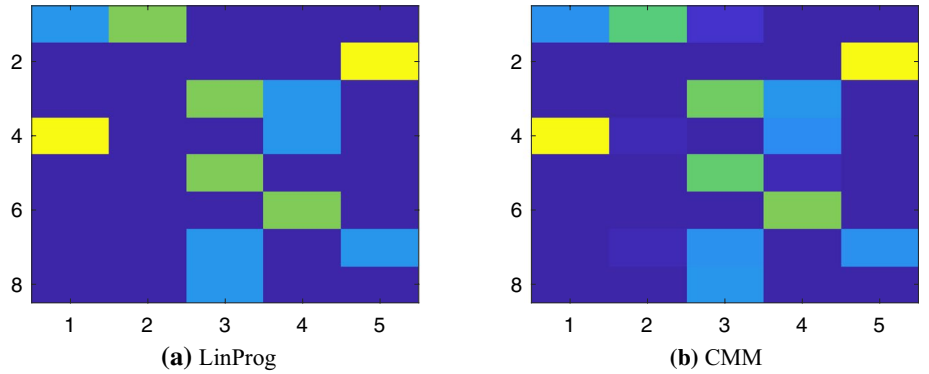


Fig. 1 Two transport plan matrices via: **a** linear Programming and **b** Coupling Matrix Manifold Optimization

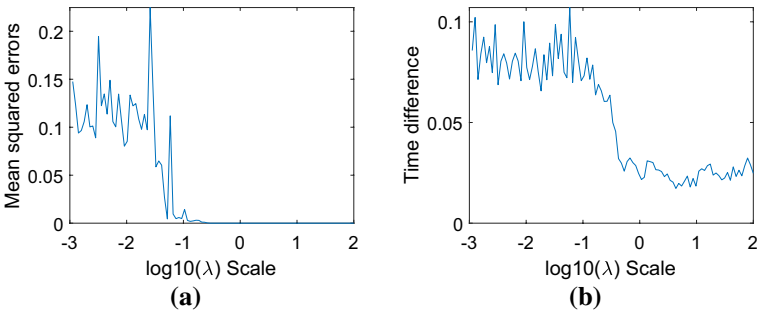


Fig. 2 Algorithm comparison over 100 regularizer values at log scale $[-3, 2]$: **a** the mean Squared errors between the solutions of CMM and Sinkhorn algorithms and **b** time difference (in seconds) between CMM and Sinkhorn algorithms

problems, i.e., in practical logistic planning, one prefers to use all the possible routes from multiple suppliers to retailers rather than to congest on several routes. The proposed manifold optimization performs well in this regard.

Next we consider an entropy regularized OT problem which can be easily solved by the Sinkhorn algorithm. We test both the Sinkhorn algorithm and the new coupling matrix manifold optimization on the same synthetic problem over 100 regularizer λ values on a log scale ranging $[-3, 2]$, i.e., $\lambda = 0.001 = 10^{-3}$ to $100.0 = 10^{2}$. Mean squared error (MSE) is used as a criterion to measure the closeness between transport plan matrices in both algorithms. We run the experiment 10 times each and the mean MSE and the mean time used for 10 runs are reported in Fig. 2.

In the experiments, we observed that when the Sinkhorn algorithm breaks down for $\lambda \leq 0.001$ due to computational instability. On the contrary, the manifold-assisted algorithm generates reasonable results for a wider range of regularizer values. From Fig. 2a, we also observe that both algorithms give almost exactly same transport plan matrices when $\lambda > 0.1668$.

In terms of computational complexity, the Sinkhorn algorithm is generally more efficient than the manifold assisted method in the entropy regularized OT problem, given the

Table 1 The classification accuracy of the kNN classifiers based on two algorithms for the order-preserving Wasserstein distance

Algorithms	1NN	3NN	5NN	7NN	13NN	19NN
S-OWP (Su and Hua 2017)	0.8236	0.8454	0.8454	0.8418	0.8473	0.8290
(std)	0.0357	0.0215	0.0215	0.0220	0.0272	0.0240
CM-OWP	0.8091	0.8309	0.8255	0.8218	0.8109	0.8091
(std)	0.0275	0.0212	0.0194	0.0196	0.0317	0.0315

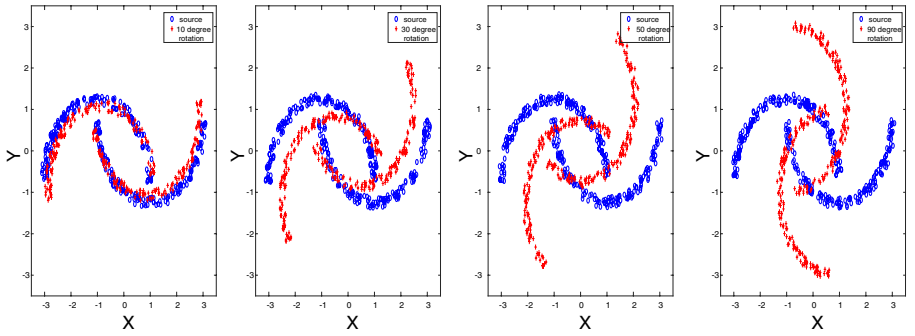
information on computational time difference between CMM and Sinkhorn as shown in Fig. 2b. This is expected as CMM works on manifold optimization where extra computation is needed to maintain the constrain conditions for the manifold. However we shall note that when the regularizer is larger, the time difference between two algorithms is negligible. For the cases of smaller λ values, the CMM is much more stable than the Sinkhorn algorithm although more computational cost is needed, but worthwhile.

4.2 Experiments on the order-preserving OT

In this experiment, we demonstrate the performance in calculating the order-preserving Wasserstein distance (Su and Hua 2017) using a real dataset. The "Spoken Arabic Digits (SAD)" dataset, available from the UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/Spoken+Arabic+Digit), contains 8,800 vectorial sequences from ten spoken Arabic digits. The sequences consist of time series of the mel-frequency cepstrum-coefficients (MFCCs) features extracted from the speech signals. This is a classification learning task on ten classes. The full set of training data has 660 sequence samples per digit spoken repeatedly for 10 times by 44 male and 44 female Arabic native speakers. For each digit, another 220 samples are retained as testing sets.

The experimental setting is similar to that in Su and Hua (2017). Based on the order-preserving Wasserstein distance (OPW) between any two sequences, we directly test the nearest neighbour (NN) classifier. To define the distance in (13), we use three hyperparameters: the width parameter σ of the radius basis function (RBF), two regularizers λ_1 and λ_2. For the comparative purpose, these hyperparameters are chosen to be $\sigma = 1$, $\lambda_1 = 50$ and $\lambda_2 = 0.1$, as in Su and Hua (2017). Our purpose here is to illustrate that the performance of the NN classifier based on the coupling matrix manifold optimization algorithm (named as CM-OPW) is comparable to the NN classification results from Sinkhorn algorithm (named as S-OPW). We randomly choose 10% training data and 10% testing data for each run in the experiments. The classification mean accuracy and their standard error are reported in Table 1 based on five runs.

In this experiment, we also observe that the distance calculation fails for some pairs of training and testing sequences due to numerical instability of the Sinkhorn algorithm. Our conclusion is that the performance of the manifold-based algorithm is comparable in terms of similar classification accuracy. When $k = 1$, the test sequence is also viewed as a query to retrieve the training sequences, and the mean average precision (MAP) is MAP = 0.1954 for the S-OPW and MAP = 0.3654 for CM-OPW. Theoretically the Sinkhorn algorithm is super-fast, outperforming all other existing algorithms; however, it is not applicable to those OT problems with non-entropy regularizations. We demonstrate these problems in the next subsection.



(a) rotation = 10° **(b)** rotation = 30° **(c)** rotation = 50° **(d)** rotation = 90°

Fig. 3 Two moons' example for increasing rotation angles

4.3 Laplacian regularized OT problems: synthetic domain adaption

Courty et al. (2016) analyzed two moon datasets and found that the OM domain adaption method significantly outperformed the subspace alignment method significantly.

We use the same experimental data and protocol as in Courty et al. (2016) to perform a direct and fair comparison between results[2]. Each of the two domains represents the source and the target respectively presenting two moon shapes associated with two specific classes. See Fig. 3.

The source domain contains 150 data points sampled from the two moons. Similarly, the target domain has the same number of data points, sampled from two moons shapes which rotated at a given angle from the base moons used in the source domain. A classifier between the data points from two domains will be trained once transportation process is finished.

To test the generalization capability of the classifier based on the manifold optimization method, we sample a set of 1000 data points according to the distribution of the target domain and we repeat the experiment for 10 times, each of which is conducted on 9 different target domains corresponding to 10°, 20°, 30°, 40°, 50°, 60°, 70°, 80° and 90° rotations, respectively. We report the mean classification error and variance as comparison criteria.

We train the SVM classifiers with a Gaussian kernel, whose parameters were automatically set by 5-fold cross-validation. The final results are shown in Table 1. For comparative purpose, we also present the results based on the DA-SVM approach (Bruzzone and Marconcini 2010) and the PBDA (Germain et al. 2013) from Courty et al. (2016).

From Table 2, we observe that the coupling matrix manifold assisted optimization algorithm significantly improves the efficiency of the GCG (the generalized conditional gradient) algorithm which ignores the manifold constraints although a weaker Lagrangian condition was imposed in the objective function. This results in a sub-optimal solution to the transport plan, producing poorer transported source data points. The results in Table 2 show our coupling matrix manifold optimal transport Laplacian (CM-OT-Lap) algorithm provides a more stable classification results along with different data structures (from 10°

[2] We sincerely thanks to the authors of Courty et al. (2016) for providing us the complete simulated two moon datasets.

Table 2 Mean error rate over 10 realizations for the two moons simulated example

Rotate Angle	10°	20°	30°	40°	50°	70°	90°
SVM (no adapt.)	**0**	0.104	0.24	0.312	0.4	0.764	0.828
DASVM	**0**	**0**	0.259	0.284	0.334	0.747	0.82
PBDA	**0**	0.094	0.103	0.225	0.412	0.626	0.687
OT-Laplace	**0**	**0**	0.004	0.062	0.201	0.402	0.524
CM-OT-Lap (ours)	0.0027	0.0043	**0.0014**	**0.0142**	**0.0301**	**0.0446**	**0.0797**
(variance)	0.0000	0.0002	0.0000	0.0007	0.0013	0.0015	0.0057

DASVM (Bruzzone and Marconcini 2010); PBDA (Germain et al. 2013); OT-Laplace (Courty et al. 2016)

to 90° rotations) with the highest classification error only 0.0466 at 70° rotation. Especially for the problem with highest difficulty in 90°, the CM-OT-Lap resulted in a mean classification error as 0.0797 whereas other methods are with the results as 0.82 (DASVM), 0.687 (PBDA) and 0.524 (OT-Lap) respectively, indicating that computing the transportation map between two data sets can significantly help us to accurately do the classification work. We also provided the variance of the classification results to show the robustness of our method. Our results show relatively low variances.

4.4 Laplacian regularized OT problems: image domain adaption

We now apply our manifold-based algorithm to solve the Laplician regularized OT problem for the challenging real-world adaptation tasks. In this experiment, we test the domain adaption for both handwritten digits images and face images for recognition. We follow the same setting used in Courty et al. (2016) for a fair comparison.

4.4.1 Digit recognition

We use the two-digit famous handwritten digit datasets USPS and MNIST as the source and target domain and verse, respectively, in our experiment[3]. The datasets share 10 classes of features (single digits from 0-9). We randomly sampled 1800 images from USPS and 2000 from MNIST. In order to unify the dimensions of two domains, the MNIST images are re-sized into 16×16 resolution same as USPS. The grey level of all images are then normalized to produce the final feature space for all domains. For this case, we have two settings U-M (USPS as source and MNIST as target) and M-U (MNIST as source and USPS as target).

4.4.2 Face recognition

In the face recognition experiment, we use PIE ("Pose, Illumination, Expression") dataset which contain 32×32 images of 68 individuals with different poses: pose, illuminations and expression conditions.[4] In order to make a fair and reasonable comparison with Courty

[3] Both datasets can be found at http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html.
[4] http://www.cs.cmu.edu/afs/cs/project/PIE/MultiPie/Multi-Pie/Home.html.

Table 3 Overall recognition accuracies in % in both digital and face recognition

Domains	1NN	OT-IT	OT-Lap	CMM-OT-Lap
U-M	39.00	53.66	57.43	**60.67**
M-U	58.33	64.73	64.72	**70.37**
mean	48.66	59.20	61.07	**65.52**
P1-P2	23.79	53.73	58.92	**58.08**
P1-P3	23.50	57.43	57.62	**62.65**
P1-P4	15.69	47.21	47.54	**48.98**
P2-P1	24.27	60.21	62.74	**93.10**
P2-P3	44.45	63.24	64.29	**69.18**
P2-P4	25.86	51.48	53.52	**65.10**
P3-P1	20.95	57.50	57.87	**91.70**
P3-P2	40.17	63.61	65.75	**75.66**
P3-P4	26.16	52.33	54.02	**87.60**
P4-P1	18.14	45.15	45.67	**90.30**
P4-P2	24.37	50.71	52.50	**66.46**
P4-P3	27.30	52.10	52.71	**62.29**
mean	26.22	54.56	56.10	**72.59**

et al. (2016), we select PIE05(C05, denoted as P1, left pose), PIE07(C07, denote as P2, upward pose), PIE09(C09, denoted as P3, downward pose) and PIE29(C29, denoted as P4, right pose). This four domains induce 12 adaptation problems with increasing difficulty (the hardest adaptation is from left to the right). Note that large variability between each domain is due to the illumination and expression.

4.4.3 Experiment settings and result analysis

We generate the experimental results by applying the manifold-based algorithm on two types of Laplacian regularized problems, namely: Problem (16) with $\alpha = 0$ (CMM-OT-Lap) and with $\alpha = 0.5$ (CMM-OT-symmLap). We follow the same experimental settings in Courty et al. (2016). For all methods, the regularization parameter λ was initially set to 0.01, similarly, another parameter, η that controls the performance of Laplacian terms was set to 0.1.

In both Face and digital recognition experiments, 1NN is trained with the adapted source data and target data, and then we report the overall accuracy (OA) score (in %) calculated on testing samples from the target domain. We compare OAs between our CMM-OT solutions to the baseline methods and the results generated by the methods provided in Courty et al. (2016) in Table 3. Note that, we applied both coupling matrix OT Laplacian and coupling matrix OT symmetric Laplacian algorithm for all experiments, and due to the high similarity of the results generated from these two methods, we only list the OA generated from the non-symmetric CMM-OT-Lap algorithm in table.

As a result, the OA based on the solution generated from CMM based OT Laplician algorithm over-performs all other methods in both digital and face recognition experiments, with mean OA = 65.52% and 72.59%, respectively. Averagely, our method is able to increase 4% and 16% of the OA from the previous results. However, in terms of the adaptation problem with the highest difficulty : P1 to P4, we got similar result compared

with previous results, with the OA = 47.54% from Courty et al. (2016) and 48.98% from our method respectively.

5 Conclusions

This paper explores the so-called coupling matrix manifolds on which the majority of the OT objective functions are defined. We formally defined the manifold, explored its tangent spaces, defined a Riemennian metric based on information measure, proposed all the formulas for the Riemannian gradient, Riemannina Hessian and an appropriate retraction as the major ingradients for implementation Riemannian optimization on the manifold. We apply manifold-based optimization algorithms (Riemannian gradient descent and second-order Riemannian trust region) into several types of OT problems, including the classic OT problem, the entropy regularized OT problem, the power regularized OT problem, the state-of-the-art order-preserving Wasserstein distance problems and the OT problem in regularized domain adaption applications. The results from three sets of numerical experiments demonstrate that the newly proposed Riemannian optimization algorithms perform as well as the classic algorithms such as Sinkhorn algorithm. We also find that the new algorithm overperforms the generalized conditional gradient when solving non-entropy regularized OT problem where the classic Sinkhorn algorithm is not applicable.

Acknowledgements This project is partially supported by the University of Sydney Business School ARC Bridging grant. The authors are graeteful to the anonymous reviewers for their constructive comments to improve this work.

References

Absil, P. A., Mahony, R., & Sepulchre, R. (2008). *Optimization algorithms on matrix manifolds*. Princeton: Princeton University Press.

Altschuler, J., Weed, J., & Rigollet, P. (2017). Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration. In *Proceedings of the 31st international conference on neural information processing systems* (pp. 1961–1971), Curran Associates Inc., USA, NIPS'17.

Amari, S., & Nagaoka, H. (2007). *Methods of informantion geometry*. Providence: American Mathematical Society.

Amari, S., & Nagaoka, H. (2000). *Methods of Information Geometry* (pp. 37–40). Oxford University Press, New York, chap Chentsov's theorem and some historical remarks.

Ambrogioni, L., Güçlü U, Güçlütürk, Y., Hinne, M., Maris, E., & van Gerven, M. A. J. (2018). Wasserstein variational inference. In *Proceedings of the 32nd international conference on neural information processing systems* (pp. 2478–2487), Curran Associates Inc., USA, NIPS'18.

Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein GAN. arXiv:1701.07875.

Bertsekas, D. (1999). *Nonlinear programming*. Belmont: Athena Scientific.

Bousquet, O., Gelly, S., Tolstikhin, I., Simon-Gabriel, C. J., & Schölkopf, B. (2017). From optimal transport to generative modeling: The VEGAN cookbook. Tech. rep.

Brezis, H. (2018). Remarks on the Monge–Kantorovich problem in the discrete setting. *Comptes Rendus Mathematique*, *356*(2), 207–213.

Bruzzone, L., & Marconcini, M. (2010). Domain adaptation problems: A DASVM classification technique and a circular validation strategy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(5), 770–787.

Courty, N., Flamary, R., Tuia, D., & Rakotomamonjy, A. (2016). Optimal transport for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(9), 1853–1865.

Cuturi, M. (2013). Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in Neural Information Processing Systems*, *26*, 2292–2300.

Cuturi, M., & Doucet, A. (2014). Fast computation of Wasserstein barycenters. In Xing, E. P., & Jebara, T. (Eds.) *Proceedings of the 31st international conference on machine learning* (pp. 685–693), Bejing, China, vol 32.

De Loera, J. A., & Kim, E. D. (2014). Combinatorics and geometry of transportation polytopes: An update. *Discrete Geometry and Algebraic Combinatorics*, *625*, 37–76.

Dessein, A., Papadakis, N., & Rouas, J. L. (2018). Regularised optimal transport and the rot mover's distance. *Journal of Machine Learning Research*, *19*(15), 1–53.

Douik, A., & Hassibi, B. (2019). Manifold optimization over the set of doubly stochastic matrices: A second-order geometry. *IEEE Transactions on Signal Processing*, *67*(22), 5761–5774.

Essid, M., & Solomon, J. (2018). Quadratically regularized optimal transport on graphs. *SIAM Journal on Scientific Computing*, *40*(4), A1961–A1986.

Ferradans, S., Papadakis, N., Peyre, G., & Aujol, J. F. (2014). Regularized discrete optimal transport. *SIAM Journal on Imaging Sciences*, *7*(3), 1853–1882.

Flamary, R., Cuturi, M., Courty, N., & Rakotomamonjy, A. (2018). Wasserstein discriminant analysis. *Machine Learning*, *107*(12), 1923–1945.

Frogner, C., Zhang, C., Mobahi, H., Araya-Polo, M., & Poggio, T. A. (2015). Learning with a Wasserstein loss. In *Advances in neural information processing systems (NIPS)*, vol 28.

Gabay, D. (1982). Minimizing a differentiable function over a differential manifold. *Journal of Optimization Theory and Applications*, *37*(2), 177–219.

Genevay, A., Cuturi, M., Peyré, G., & Bach, F. (2016). Stochastic optimization for large-scale optimal transport. In Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R. (Eds.) *Advances in neural information processing systems 29* (pp. 3440–3448). Curran Associates, Inc.

Germain, P., Habrard, A., Laviolette, F., & Morvant, E. (2013). APAC-Bayesian approach for domain adaptation with specialization to linear classifiers. In *Proceedings of international conference on machine learning (ICML)* (pp. 738–746). Atlanta, USA.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. (2017). Improved training of Wasserstein gans. In *Proceedings of the 31st international conference on neural information processing systems* (pp. 5769–5779). Curran Associates Inc., Red Hook, NY, USA, NIPS'17.

Haker, S., Zhu, L., Tannenbaum, A., & Angenent, S. (2004). Optimal mass transport for registration and warping. *International Journal of Computer Vision*, *60*(3), 225–240.

Hong, X., & Gao, J. (2015). Sparse density estimation on multinomial manifold combining local component analysis. In *Proceedings of international joint conference on neural networks (IJCNN)* (pp. 1–7). https://doi.org/10.1109/IJCNN.2015.7280301.

Hong, X., & Gao, J. (2018). Estimating the square root of probability density function on Riemannian manifold. *Expert Systems* (in press) https://doi.org/10.1111/exsy.12266.

Hong, X., Gao, J., Chen, S., & Zia, T. (2015). Sparse density estimation on the multinomial manifold. *IEEE Transactions on Neural Networks and Learning Systems*, *26*, 2972–2977.

Jacobs, M., & Lèger, F. (2020). A fast approach to optimal transport: The back-and-forth method. arXiv :190512154 2.

Kantorovich, L. V. (1942). On the translocation of masses. *Doklady Akademii Nauk SSSR (NS)*, *37*, 199–201.

Knight, P. A. (2008). The Sinkhorn–Knopp algorithm: Convergence and applications. *SIAM Journal on Matrix Analysis and Applications*, *30*(1), 261–275.

Kolouri, S., Pope, P. E., Martin, C. E., & Rohde, G. K. (2019) Sliced Wasserstein auto-encoders. In *Proceedings of international conference on learning representation (ICLR)*.

Lee, Y. T., & Sidford, A. (2014). Path finding methods for linear programming: Solving linear programs in o(vrank) iterations and faster algorithms for maximum flow. In *Proceedings of IEEE 55th annual symposium on foundations of computer science* (pp. 424–433). https://doi.org/10.1109/FOCS.2014.52.

Maman, G., Yair, O., Eytan, D., & Talmon, R. (2019). Domain adaptation using Riemannian geometry of SPD matrices. In *International conference on acoustics, speech and signal processing (ICASSP)* (pp. 4464–4468). Brighton, United Kingdom: IEEE.

Miller, M., & Lent, J. V. (2016). Monge's optimal transport distance with applications for nearest neighbour image classification. arXiv:1612.00181.

Monge, G. (1781). Mémoire sur la théorie des déblais et des remblais. Histoire de l'Académie Royale des Sciences de Paris.

Montavon, G., Müller, K. R., & Cuturi, M. (2016). Wasserstein training of restricted Boltzmann machines. *Advances in Neural In-formation Processing Systems*, *29*, 3718–3726.

Muzellec, B., Nock, R., Patrini, G., & Nielsen, F. (2017). Tsallis regularized optimal transport and ecological inference. In *Proceedings of AAAI* (pp. 2387–2393).

Panaretos, V. M., & Zemel, Y. (2019). Statistical aspects of Wasserstein distances. *Annual Review of Statistics and Its Application*, *6*, 405–431.

Peyre, G., & Cuturi, M. (2019). Computational optimal transport: With applications to data science. Foundations and Trends in Machine Learning Series, Now Publishers, https://books.google.com.au/books?id=J0BiwgEACAAJ.

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

Queyranne, M., & Spieksma, F. (2009). Multi-index transportation problems: Multi-index transportation problems MITP. *Encyclopedia of Optimization*, pp. 2413–2419.

Rabin, J., & Papadakis, N. (2015). Convex color image segmentation with optimal transport distances. In *International conference on scale space and variational methods in computer vision*. Springer, pp. 256–269.

Rubner, Y., Tomasi, C., & Guibas, L. J. (2000). The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, *40*(2), 99–121.

Schmitzer, B. (2019). Stabilized sparse scaling algorithms for entropy regularized transport problems. *SIAM Journal on Scientic Computing*, *41*(3), A1443–A1481.

Solomon, J., de Goes, F., Peyré, G., Cuturi, M., Butscher, A., Nguyen, A., et al. (2015). Convolutional Wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Transactions on Graphics*, *34*(4), 66:1–66:11.

Su, B., & Hua, G. (2017). Order-preserving Wasserstein distance for sequence matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1049–1057).

Su, B., & Wu, Y. (2019). Learning distance for sequences by learning a ground metric. In *Proceedings of the 36th international conference on machine learning (ICML)*.

Sun, Y., Gao, J., Hong, X., Mishra, B., & Yin, B. (2016). Heterogeneous tensor decomposition for clustering via manifold optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*, 476–489.

Tolstikhin, I., Bousquet, O., Gelly, S., & Schoelkopf, B. (2018). Wasserstein auto-encoders. In *Proceedings of international conference on learning representation*.

Villani, C. (2009). Optimal transport: Old and new. Berlin: Springer, chap The Wasserstein distances (pp. 93–111).

Yair, O., Dietrich, F., Talmon, R., & Kevrekidis, I.G. (2019). Optimal transport on the manifold of SPD matrices for domain adaptation. arXiv:1906.00616.

Zhang, S., Gao, Y., Jiao, Y., Liu, J., Wang, Y., & Yang, C. (2019). Wasserstein-Wasserstein auto-encoders. arXiv:1902.09323.

Zhao, P,, & Zhou, Z. H. (2018). Label distribution learning by optimal transport. In *Proceedings of the thirty-second AAAI conference on artificial intelligence (AAAI)* (pp. 4506–4513).