Learning Weighted Submanifolds with Variational Autoencoders and Riemannian Variational Autoencoders

Nina Miolane
Stanford University
nmiolane@stanford.edu

Susan Holmes
Stanford University
susan@stat.stanford.edu

Abstract

Manifold-valued data naturally arises in medical imaging. In cognitive neuroscience, for instance, brain connectomes base the analysis of coactivation patterns between different brain regions on the analysis of the correlations of their functional Magnetic Resonance Imaging (fMRI) time series – an object thus constrained by construction to belong to the manifold of symmetric positive definite matrices. One of the challenges that naturally arises in these studies consists of finding a lower-dimensional subspace for representing such manifold-valued and typically high-dimensional data. Traditional techniques, like principal component analysis, are ill-adapted to tackle non-Euclidean spaces and may fail to achieve a lower-dimensional representation of the data – thus potentially pointing to the absence of lower-dimensional representation of the data. However, these techniques are restricted in that: (i) they do not leverage the assumption that the connectomes belong on a pre-specified manifold, therefore discarding information; (ii) they can only fit a linear subspace to the data. In this paper, we are interested in variants to learn potentially highly curved submanifolds of manifold-valued data. Motivated by the brain connectomes example, we investigate a latent variable generative model, which has the added benefit of providing us with uncertainty estimates – a crucial quantity in the medical applications we are considering. While latent variable models have been proposed to learn linear and nonlinear spaces for Euclidean data, or geodesic subspaces for manifold data, no intrinsic latent variable model exists to learn nongeodesic subspaces for manifold data. This paper fills this gap and formulates a Riemannian variational autoencoder with an intrinsic generative model of manifold-valued data. We evaluate its performances on synthetic and real datasets by introducing the formalism of weighted Riemannian submanifolds.

1. Introduction

Representation learning aims to transform data x into a lower-dimensional variable z designed to be more efficient for any downstream machine learning task, such as exploratory analysis of clustering, among others. In this paper, we focus on representation learning for manifold-valued data that naturally arise in medical imaging. Functional Magnetic Resonance Imaging (fMRI) data are often summarized into "brain connectomes", that capture the coactivation of brain regions of subjects performing a given task (memorization, image recognition, or mixed gamble task, for example). As correlation matrices, connectomes belong to the cone of symmetric positive definite (SPD) matrices. This cone can naturally be equipped with a Riemannian manifold structure, which has shown to improve performances on classification tasks [1]. Being able to learn low-dimensional representations of connectomes within the pre-specified SPD manifold is key to model the intrinsic variability across subjects, and tackle the question: do brain connectomes from different subjects form a lower-dimensional subspace within the manifold of correlation matrices? If so, each subject's connectome x can be represented by a latent variable z of lower dimension. Anticipating potential downstream medical tasks that predict behavioral variables (such as measures of cognitive, emotional, or sensory processes) from z, we seek a measure of uncertainty associated with z. In other words, we are interested in a posterior in z given x.

While the literature for generative models capturing lower-dimensional representations of Euclidean data is rich, such methods are typically ill-suited to the analysis of manifold-valued data. Can we yet conclude that lower-dimensional representations within these manifolds are not achievable? The aforementioned techniques are indeed restricted in that: either (i) they do not leverage any geometric knowledge as to the known manifold to which the data, such as the connectomes, belong; or (ii) they can only fit a linear (or geodesic, *i.e.* the manifold equivalent of linear) subspace to the data. In this paper, we focus on alternatives

with a latent variable generative model that address (i) and (ii).

1.1. Related Work

There is a rich body of literature on manifold learning methods. We review here a few of them, which we evaluate based on the following desiderata:

- Is the method applicable to manifold-valued data?
- For methods on Euclidean data: does the method learn a linear or a nonlinear manifold, see Figure 1 (a, b)?
- For methods geared towards Riemannian manifolds: does the method learn a geodesic (*i.e.* the manifold equivalent of a linear subspace) - or a nongeodesic subspace, see Figure 1 (c, d)?
- Does the method come with a latent variable generative model?

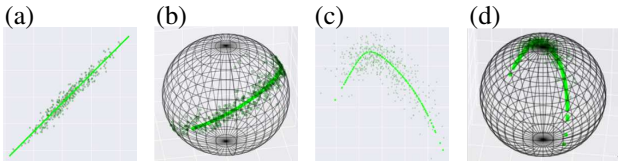

Figure 1. (a) Learning a 1D linear subspace in a 2D Euclidean space; (b) Learning a geodesic in a 2D manifold (sphere); (c) Learning a 1D nonlinear subspace in a 2D Euclidean space; (d) Learning a nongeodesic 1D subspace in a 2D manifold (sphere).

1.1.1 Learning Linear and Geodesic Subspaces

Principal Component Analysis (PCA) [15] learns a linear subspace, while Probabilistic PCA (PPCA) and Factor Analysis (FA) [25] achieve the same goal within a probabilistic framework relying on a latent variable generative mode; see Figure 1 (a). These techniques are based on vector space's operations that make them unsuitable for data on manifolds. As a consequence, researchers have developed methods for manifold-valued data, which take into account the geometric structure; see Figure 1 (b).

Principal Geodesic Analysis (PGA) [8, 23], tangent PGA (tPGA) [8], Geodesic Principal Component Analysis (gPCA) [11], principal flows [14], barycentric subspaces (BS) [17] learn variants of "geodesic" subspaces, *i.e.* generalizations in manifolds of linear spaces in Euclidean spaces. Probabilistic PGA [29] achieves the same goal, while adding a latent variable model generating data on a manifold.

However, these methods are restricted in the type of submanifold that can be fitted to the data, either linear or geodesic - a generalization of linear subspaces to manifolds. This restriction can be considered both a strength and a weakness. While it protects from overfitting with a submanifold that is too flexible, it also prevents the method from capturing possibly nonlinear effects. With current dataset sizes exploding (even within biomedical imaging datasets which have been historically much smaller), it seems that the investigation of flexible submanifold learning techniques takes on crucial importance.

1.1.2 Learning Non-Linear and Nongeodesic Subspaces

While methods for learning nonlinear manifolds from Euclidean data are numerous (see Figure 1 (c)), those providing a latent variable generative models are scarce. Kernel PCA [21], multi-dimensional scaling and its variants [6, 3], Isomap [24], Local Linear Embedding (LLE) [20], Laplacian eigenmaps [2], Hessian LLE [7], Maximum variance unfolding [28], and others, learn lower-dimensional representations of data but do not provide a latent variable generative model, nor a parameterization of the recovered subspace.

In contrast, principal curves and surfaces (PS) [9] and autoencoders fit a nonlinear manifold to the data, with an explicit parameterization of this manifold. However, this framework is not directly transferable to non-Euclidean data and has been more recently generalized to principal curves on Riemannian manifolds [10]. To our knowledge, this is the only method for nongeodesic submanifold learning on Riemannian manifolds (see Figure 1 (d)). A probabilistic approach to principal curves was developed in [4] for the Euclidean case, but not the manifold case. Similarly, variational autoencoders (VAEs) [12] were developed to provide a latent variable generative model for autoencoders. However, they do not apply to manifold-valued data.

In order to create a latent variable generative model for manifold-valued data, we can either generalize principal curves on manifolds by adding a generative model or generalize VAEs for manifold-valued data. Principal curves require a parameterization of the curve that involves a discrete set of points. As the number of points needed grows exponentially with the dimension of the estimated surface, scaling this method to high dimensional principal surfaces becomes more difficult. As a consequence, we choose to generalize VAEs to manifold-valued data. This paper introduces Riemannian VAE, an intrinsic method that provides a flexible generative model of the data on a pre-specified manifold. We emphasize that our method does not amount to embedding the manifold in a larger Euclidean space, training the VAE, and projecting back onto the original manifold - a strategy that does not come with an intrinsic generative model of the data. We implement and compare both meth-

ods in Section 6.

1.2. Contribution and Outline

This paper introduces the intrinsic Riemannian VAE, a submanifold learning technique for manifold-valued data. After briefly reviewing the (Euclidean) VAE, we present our Riemannian generalization. We show how Riemannian VAEs generalize both VAE and Probabilistic Principal Geodesic Analysis. We provide theoretical results describing the family of submanifolds that can be learned by the Riemannian method. To do so, we introduce the formalism of weighted Riemannian submanifolds and associated Wasserstein distances. This formalism also allows giving a sense to the definition of consistency in the context of submanifold learning. We use this to study the properties of VAE and Riemannian VAE learning techniques, on theoretical examples and synthetic datasets. Lastly, we deploy our method on real data by applying it to the analysis of connectome data.

2. Riemannian Variational Autoencoders (rVAE)

2.1. Review of (Euclidean) VAE

We begin by setting the basis for variational autoencoders (VAEs) [12, 19]. Consider a dataset $x_1, ..., x_n \in \mathbb{R}^D$. A VAE models each data point x_i as the realization of a random variable X_i generated from a nonlinear probabilistic model with lower-dimensional latent variable Z_i taking value in $\mathbb{R}^L$, where $L < D$, such as:

$$X_i = f_\theta(Z_i) + \epsilon_i, \tag{1}$$

where $Z_i \sim \mathcal{N}(0, \mathbb{I}_L)$ *i.i.d.* and ϵ_i represents *i.i.d.* measurement noise distributed as $\epsilon_i \sim \mathcal{N}(0, \sigma^2 \mathbb{I}_D)$. The function f_θ belongs to a family $\mathcal{F}$ of nonlinear generative models parameterized by θ, and is typically represented by a neural network, called the decoder, such that: $f_\theta(\bullet) = \Pi_{k=1}^K g(w_k \bullet + b_k)$ where Π represents the composition of functions, K the number of layers, g an activation function, and the w_k, b_k are the weights and biases of the layers. We write: $\theta = \{w_k, b_k\}_{k=1}^K$. This model is illustrated on Figure 2.

The VAE pursues a double objective: (i) it learns the parameters θ of the generative model of the data; and (ii) it learns an approximation $q_\phi(z|x)$, within a variational family $\mathcal{Q}$ parameterized by ϕ, of the posterior distribution of the latent variables. The class of the generative model $\mathcal{F}$ and the variational family $\mathcal{Q}$ are typically fixed, as part of the design of the VAE architecture. The VAE achieves its objective by maximizing the evidence lower bound (ELBO) defined as:

$$\mathcal{L}_1(x, \theta, \phi) = \mathbb{E}_{q_\phi(z|x)} \left[\log \frac{p_\theta(x, z)}{q_\phi(z|x)} \right], \tag{2}$$

which can conveniently be rewritten as:

$$\begin{aligned} \mathcal{L}_1(x, \theta, \phi) &= l(\theta, x) - \mathrm{KL}\left(q_\phi(z|x) \parallel p_\theta(z|x)\right) \\ &= \mathbb{E}_{q_\phi(z)}\left[\log p_\theta(x|z)\right] - \mathrm{KL}\left(q_\phi(z|x) \parallel p(z)\right) \\ &= \mathcal{L}_{\text{rec}}(x, \theta, \phi) + \mathcal{L}_{\text{reg}}(x, \phi), \end{aligned}$$

where the terms $\mathcal{L}_{\text{rec}}(x, \theta, \phi)$ and $\mathcal{L}_{\text{reg}}(x, \phi)$ are respectively interpreted as a reconstruction objective and as a regularizer to the prior on the latent variables.

From a geometric perspective, the VAE learns a manifold $\hat{N} = N_{\hat{\theta}} = f_{\hat{\theta}}(\mathbb{R}^L)$ designed to estimate the true submanifold of the data $N_\theta = f_\theta(\mathbb{R}^L)$. The approximate distribution $q_\phi(z|x)$ can be seen as a (non-orthogonal) projection of x on the subspace $N_{\hat{\theta}}$ with associated uncertainty.

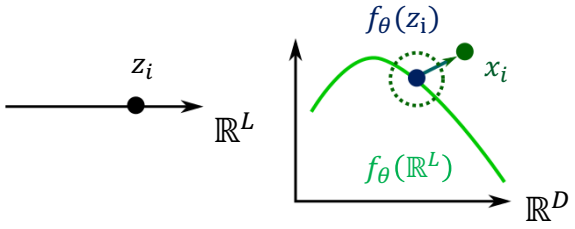

Figure 2. Generative model for the variational autoencoder with latent space $\mathbb{R}^L$ and data space $\mathbb{R}^D$. The latent variable z_i is sampled from a standard multivariate normal distribution on $\mathbb{R}^L$ and embedded into $\mathbb{R}^D$ through the embedding f_θ. The data x_i is generated by addition of a multivariate isotropic Gaussian noise in $\mathbb{R}^D$.

2.2. Riemannian VAE (rVAE)

We generalize the generative model of VAE for a dataset $x_1, ..., x_n$ on a Riemannian manifold M. We need to adapt two aspects of the (Euclidean) VAE: the embedding function f_θ parameterizing the submanifold, and the noise model on the manifold M. We refer to supplementary materials for details on Riemannian geometry, specifically the notions of Exponential map, Riemannian distance and Fréchet mean.

2.2.1 Embedding

Let $\mu \in M$ be a base point on the manifold. We consider the family of functions $f_\theta : \mathbb{R}^L \mapsto \mathbb{R}^D \simeq T_\mu M$ that are parameterized by a fully connected neural network of parameter θ, as in the VAE model. We define a new family of functions with values on M, by considering: $f^M_{\mu,\theta}(\bullet) = \mathrm{Exp}^M(\mu, f_\theta(\bullet))$ as an embedding from $\mathbb{R}^L$ to M, where $\mathrm{Exp}^M(\mu, \bullet)$ is the Riemannian exponential map of M at μ.

2.2.2 Noise model

We generalize the Gaussian distribution from the VAE generative model, as we require a notion of distribution on manifolds. There exist several generalizations of the Gaussian distribution on Riemannian manifolds [16]. To have a tractable expression to incorporate into our loss functions, we consider the minimization of entropy characterization of [16]:

$$p(x|\mu,\sigma) = \frac{1}{C(\mu,\sigma)} \exp\left(-\frac{d(\mu,x)^2}{2\sigma^2}\right), \tag{3}$$

where $C(\mu,\sigma)$ is a normalization constant:

$$C(\mu,\sigma) = \int_M \exp\left(-\frac{d(\mu,x)^2}{2\sigma^2}\right) dM(x), \tag{4}$$

and $dM(x)$ refers to the volume element of the manifold M at x. We call this distribution an (isotropic) Riemannian Gaussian distribution, and use the notation $x \sim \mathcal{N}^M(\mu,\sigma^2\mathbb{I}_D)$. We note that this noise model could be replaced with a different distribution on the manifold M, for example a generalization of a non-isotropic Gaussian noise on M.

2.2.3 Generative model

We introduce the generative model of Riemannian VAE (rVAE) for a dataset $x_1,...,x_n$ on a Riemannian manifold M:

$$X_i|Z_i = \mathcal{N}^M\left(\text{Exp}^M(\mu, f_\theta(Z_i)), \sigma^2\right) \text{ and } Z_i \sim \mathcal{N}(0,\mathbb{I}_L), \tag{5}$$

where f_θ is represented by a neural network and allows to represent possibly highly "nongeodesic" submanifolds. This model is illustrated on Figure 3.

From a geometric perspective, fitting this model learns a submanifold $N_{\hat{\theta}} = \text{Exp}^M(\mu, f_{\hat{\theta}}(\mathbb{R}^L))$ designed to estimate the true $N_\theta = \text{Exp}^M(\mu, f_\theta(\mathbb{R}^L))$ in the manifold M. The approximate distribution $q_\phi(z|x)$ can be seen as a (non-orthogonal) projection of x on the submanifold $N_{\hat{\theta}}$ with associated uncertainty.

2.2.4 Link to VAE and PPGA

The rVAE model is a natural extension of both the VAE and the Probabilistic PGA (PPGA) models. We recall that, for $M = \mathbb{R}^D$, the Exponential map is an addition operation, $\text{Exp}^{\mathbb{R}^D}(\mu, y) = \mu + y$. Furthermore, the Riemannian Gaussian distribution reduces to a multivariate Gaussian $\mathcal{N}^{\mathbb{R}^D}(\mu,\sigma^2\mathbb{I}_D) = \mathcal{N}(\mu,\sigma^2\mathbb{I}_D)$. Thus, the Riemannian VAE model coincides with the VAE model when $M = \mathbb{R}^D$. Furthermore, the Riemannian VAE model coincides with the model of PPGA:

$$X_i|Z_i \sim \mathcal{N}^M\left(\text{Exp}^M(\mu, WZ_i), \sigma^2\right) \text{ and } Z_i \sim \mathcal{N}(0,\mathbb{I}_L), \tag{6}$$

when the decoder is a linear neural network: $f_\theta(z) = Wz$ for $z \in \mathbb{R}^L$.

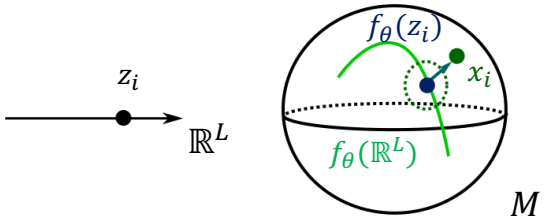

Figure 3. Generative model for the Riemannian variational autoencoder with latent space $\mathbb{R}^L$ and data space M. The latent variable z_i is sampled from a standard multivariate normal distribution on $\mathbb{R}^L$ and embedded into M through the embedding $f_{\mu,\theta}$. The data x_i is generated by addition of a Riemannian multivariate isotropic Gaussian noise in M.

Inference in PPGA was originally introduced with a Monte-Carlo Expectation Maximization (MCEM) scheme in [29]. In contrast, our approach fits the PPGA model with variational inference, as we will see in Section 4. Variational inference methods are known to be less accurate but faster than Monte-Carlo approaches. Consequently, our training procedure allows to speed-up learning within the PPGA model.

3. Expressiveness of rVAE

The Riemannian VAE model parameterizes an embedded submanifold N defined by a smooth embedding f_θ^M as:

$$N = f_\theta^M(\mathbb{R}^L) = \text{Exp}^M(\mu, f_\theta(\mathbb{R}^L)), \tag{7}$$

where f_θ is the function represented by the neural net, with a smooth activation function, and the parameter μ is absorbed in the notation θ in f_θ^M. The flexibility in the nonlinear function f_θ allows rVAE to parameterize embedded manifolds that are not necessarily geodesic at a point. A question that naturally arises is the following: can rVAE represent any smooth embedded submanifold N of M? We give results, relying on the universality approximation theorems of neural networks, that describe the embedded submanifolds that can be represented with rVAE.

3.1. Weighted Riemannian submanifolds

We introduce the notion of weighted submanifolds and suggest the associated formalism of Wasserstein distances to analyze dissimilarities between general submanifolds of M and submanifolds of M parameterized by rVAE.

Definition 1 (Weighted (sub)manifold) *Given a complete N-dimensional Riemannian manifold (N, g^N) and a smooth probability distribution $\omega : N \to \mathbb{R}$, the weighted manifold (N, ω) associated to N and ω is defined as the triplet:*

$$(M, g^N, d\nu = \omega.dN), \tag{8}$$

where dN denotes the Riemannian volume element of N.

The Riemannian VAE framework parameterizes weighted submanifold defined by:

$$N_\theta : (f_\theta^M(\mathbb{R}^L), g_M, f_\theta^M * \mathcal{N}(0, \mathbb{I}_L)), \tag{9}$$

so that the submanifold N_θ is modeled as a singular (in the sense of the Riemannian measure of M) probability density distribution with itself as support. The distribution is associated with the embedding of the standard multivariate Gaussian random variable $Z \sim \mathcal{N}(0, \mathbb{I}_L)$ in M through f_θ^M, which we denote: $f_\theta^M * \mathcal{N}(0, \mathbb{I}_L))$.

3.2. Wasserstein distance on weighted submanifolds

We can measure distances between weighted submanifolds through the Wasserstein distances associated with their distributions.

Definition 2 (Wasserstein distance) *The 2-Wasserstein distance between probability measures ν_1 and ν_2 defined on M, is defined as:*

$$d_2(\nu_1, \nu_2) = \left(\inf_{\gamma \in \Gamma(\nu_1, \nu_2)} \int_{M \times M} d_M(x_1, x_2)^2 d\gamma(x_1, x_2) \right)^{1/2}, \tag{10}$$

where $\Gamma(\nu_1, \nu_2)$ denotes the collection of all measures on $M \times M$ with marginals ν_1 and ν_2 on the first and second factors respectively.

Wasserstein distances have been introduced previously in the context of variational autoencoders with a different purpose: [26] use the Wasserstein distance with any cost function between the observed data distribution and the learned distribution, penalized with a regularization term, to train the neural network. In contrast, we use the Wasserstein distance with the square of the Riemannian distance as the cost function to evaluate distances between submanifolds. Therefore, we evaluate a distance between the data distribution and the learned distribution before the addition of the Gaussian noise. We do not use this distance to train any model; we only use it as a performance measure.

3.3. Weighted submanifold approximation result

The following result describes the expressiveness of rVAEs. For $T \in \mathbb{R}_+^*$, we denote μ_T the standard multivariate normal in $\mathbb{R}^L$ truncated at a distance T from the origin.

Proposition 1 *Let (N_T, ν_T) be a weighted Riemannian submanifold of M, embedded in a submanifold L of M homeomorphic to $\mathbb{R}^L$ and for which there exists an embedding f^M that verifies: $\nu_T = f^M * \mu_T$. Let assume the existence of $\mu \in M$ such that $N \subset V(\mu)$, where $V(\mu)$ is the maximal domain of global bijection of the Riemannian exponential of M at μ. Then, for any $0 < \epsilon < 1$, there exists a Riemannian VAE with decoder represented by a neural network f_θ, parameterized by θ, such that:*

$$d_2(N_T, N_{\theta,T}) < C_{T,D}^2 D \epsilon^2, \tag{11}$$

where d_2 is the 2-Wasserstein distance and $N_{\theta,T} = (f_\theta(\mathbb{R}^L, f_\theta * \mu_T))$, and $C_{T,D}$ a constant that depends on T and D.

Proof 1 *The proof is provided in the supplementary materials.*

As Hadamard manifolds are homeomorphic to $\mathbb{R}^L$ through their Riemannian Exponential map, the assumption $N \subset V(\mu)$ is always verified in their case. This suggests that it can be better to equip a given manifold with a Riemannian metric that has non-positive curvature: in this case, there exists an rVAE architecture that can represent any submanifold N_T, under the remaining assumptions of Proposition 1, with arbitrary precision in terms of the 2-Wasserstein distance. When learning submanifolds of the space of SPD matrices, as in Section 7, we will therefore choose either a flat Riemannian metric (Euclidean, Inverse-Euclidean, Log-Euclidean, Power-Euclidean or Square-Root metric) or a Riemannian metric with negative curvature (Affine-Invariant, Polar affine or Fisher metric).

4. Learning and inference for rVAEs

We show how to train rVAE by performing learning and inference in model (5).

4.1. Riemannian ELBO

As with VAE, we use stochastic gradient descent to maximize the ELBO:

$$\begin{aligned} \mathcal{L}_1(x, \theta, \phi) &= \mathcal{L}_{\text{rec}}(x, \theta, \phi) + \mathcal{L}_{\text{reg}}(x, \phi) \\ &= \mathbb{E}_{q_\phi(z)} \left[\log p_\theta(x|z) \right] - \text{KL} \left(q_\phi(z|x) \parallel p(z) \right), \end{aligned}$$

where the reconstruction objective $\mathcal{L}_{\text{rec}}(x, \theta, \phi)$ and the regularizer $\mathcal{L}_{\text{reg}}(x, \phi)$ are expressed using probability densities from model (5), and a variational family chosen to be the multivariate Gaussian:

$$\begin{aligned} q_\phi(z|x) &= \mathcal{N}(h_\phi(x), \sigma_\phi^2(x)), \\ p(z) &= \mathcal{N}(0, \mathbb{I}_L), \\ p(x|z) &= \mathcal{N}^M(\text{Exp}^M(\mu, f_\theta(Z_i)), \sigma^2 \mathbb{I}_D). \end{aligned}$$

The reconstruction term writes:

$$\mathcal{L}_{\text{rec}}(x,\theta,\phi) = \int_z \Big(-\log C(\sigma^2, r(\mu,z,\theta)) - \frac{d_M(x, \text{Exp}(\mu, f_\theta(z)))^2}{2\sigma^2} \Big) q_\phi(z|x) dz,$$

while the regularizer is:

$$\begin{aligned}\mathcal{L}_{\text{reg}}(x,\phi) &= \int_z \log \frac{q_\phi(z|x)}{p(z)} q_\phi(z|x) dz \\ &= \frac{1}{2} \sum_{l=1}^{L} \left(1 + \log(\sigma_l^{(i)})^2 - (\mu_l^{(i)})^2 - (\sigma_l^{(i)})^2 \right),\end{aligned}$$

where C is the normalization constant, that depends on $r(\mu,z,\theta)$, to the injectivity radius of the Exponential map at the point $\text{Exp}(\mu, f_\theta(z))$ [18]. We note that, although in the initial formulation of the VAE, the σ depends on z and θ and should be estimated during training, the implementations usually fix it and estimate it separately. We perform the same strategy here. In practice, we use the package `geomstats` [13] to plug-in the manifold of our choice within the rVAE algorithm.

4.2. Approximation

To compute the ELBO, we need to perform an approximation as providing the exact value of the normalizing constant $C(\sigma^2, r(\mu,z,\theta))$ is not trivial. The constant C depends on the σ^2 and the geometric properties of the manifold M, specifically the injectivity radius r at μ.

For Hadamard manifolds, the injectivity radius is constant and equal to ∞, thus $C = C(\sigma)$ depends only on σ. As we do not train on σ, we can discard the constant C in the loss function. For non-Hadamard manifolds, we consider the following approximation of the C, that is independent of the injectivity radius:

$$C = \frac{1 + O(\sigma^3) + O(\sigma/r)}{\sqrt{(2\pi)^D \sigma^{2D}}}. \tag{12}$$

This approximation is valid in regimes with σ^2 low in comparison to the injectivity radius, in other words, when the noise's standard deviation is small in comparison to the distance to the cut locus from each of the points on the submanifold. After this approximation, we can discard the constant C from the ELBO as before.

4.3. An important remark

We highlight that our learning procedure does not boil down to projecting the manifold-valued data onto some tangent space of M and subsequently applying a Euclidean VAE. Doing so implicitly models the noise on the tangent space as a Euclidean Gaussian, as shown in the supplementary materials. Therefore, the noise would be modulated by the curvature of the manifold. This is an undesirable property, because it entangles the probability framework with the geometric prior, *i.e.* the random effects with the underlying mathematical model.

5. Goodness of fit for submanifold learning

We consider the goodness of fit of rVAEs (and VAEs) using the formalism of weighted submanifolds that we introduced in Section 3. In other words, assuming that data truly belong to a submanifold $N_\theta = f_{\mu,\theta}(\mathbb{R}^L)$ and are generated with the rVAE model, we ask the question: how well does rVAE estimate the true submanifold, in the sense of the 2-Wasserstein distance? For simplicity, we consider that rVAE is trained with a latent space $\mathbb{R}^L$ of the true latent dimension L. Inspired by the literature of curve fitting [5], we define the following notion of consistency for weighted submanifolds.

Definition 3 (Statistical consistency) *We call the estimator $N_{\hat{\theta}}$ of N_θ statistically consistent if:*

$$\operatorname*{plim}_{n \to +\infty} d_{W_2}(N_{\hat{\theta}}, N_\theta) = 0. \tag{13}$$

Denoting $N_{\hat{\theta}}$ the submanifold learned by rVAE, we want to evaluate the function: $d(n,\sigma) = d_{W_2}(N_{\hat{\theta}}, N_\theta)$, for different values of n and σ, where $\hat{\theta}$ depends on n and σ.

5.1. Statistical inconsistency on an example

We consider data generated with the model of probabilistic PCA (PPCA) with $\mu = 0$ [25], *i.e.* a special case of a rVAE model:

$$X_i = wZ_i + \epsilon_i, \tag{14}$$

where: $w \in \mathbb{R}^{D \times L}$, $Z \sim \mathcal{N}(0, \mathbb{I}_L)$ *i.i.d.* and $\epsilon \sim \mathcal{N}(0, \mathbb{I}_D)$ *i.i.d.*. We train a rVAE, which is a VAE in this case, on data generated by this model. We chose a variational family of Gaussian distributions with variance equal to 1. Obviously, this is not the learning procedure of choice in this situation. We use it to illustrate the behavior of rVAEs and VAEs.

The case $D = 1$ and $L = 1$ allows to perform all computations in closed forms (see supplementary materials). We compute the distance between the true and learned submanifold in terms of the 2-Wasserstein distance:

$$d_2(\nu_\theta, \nu_{\hat{\theta}}) = w - \sqrt{\frac{\hat{\sigma}^2}{2} - 1} \to w - \sqrt{\frac{w^2 - 1}{2}} \neq 0,$$

where $\hat{\sigma}^2$ is the sample variance of the x_i's. We observe that the 2-Wasserstein distance does not converge to 0 as $n \to +\infty$ if $w \neq 1$ or -1. This is an example of statistical inconsistency, in the sense that we defined in this section.

5.2. Experimental study of inconsistency

We further investigate the inconsistency with synthetic experiments and consider the following three Riemannian manifolds: the Euclidean space $\mathbb{R}^2$, the sphere S^2 and the hyperbolic plane H_2. The definitions of these manifolds are recalled in the supplementary materials. We consider three Riemmanian VAE generative models respectively on $\mathbb{R}^2$, S^2 and H_2, with functions f_θ that are implemented by a three layers fully connected neural network with softplus activation. Figure 4 shows synthetic samples of size $n = 100$ generated from each of these models. The true weighted 1-dimensional submanifold corresponding to each model is shown in light green.

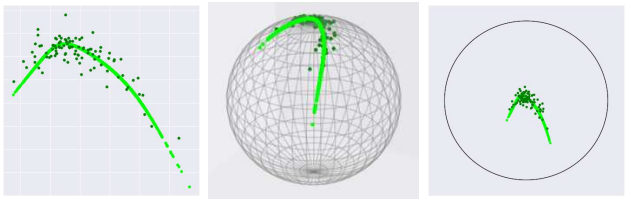

Figure 4. Synthetic data on the manifolds $\mathbb{R}^2$ (left), S^2 (center) and H_2 in its Poincaré disk representation (right). The light green represents the true weighted submanifold, the dark green points represents data points generated with rVAE.

For each manifold, we generate a series of datasets with sample sizes $n \in \{10, 100\}$ and noise standard deviation such that $\log \sigma^2 \in \{-6, -5, -4, -3, -2\}$. For each manifold and each dataset, we train a rVAE with the same architecture than the decoder that has generated the data, and standard deviation fixed to a constant value.

Figure 5 shows the 2-Wasserstein distance between the true and the learned weighted submanifold in each case, as a function of σ, where different curves represent the two different values of n. These plots confirm the statistical inconsistency observed in the theoretical example. For $\sigma \neq 0$, the VAE and the rVAE do not converge to the submanifold that has generated the data as the sample size increases. This observation should be taken into consideration when these methods are used for manifold learning, *i.e.* in a situation where the manifold itself is essential.

Additionally, we observe that this statistical inconsistency translates into an asymptotic bias that leads rVAEs and VAEs to estimate flat submanifolds, see Figure 6. We provide an interpretation to a statement in [22], where the authors compute the curvature of the submanifold learned with a VAE on MNIST data and observe a "surprinsingly little" curvature. Our experiments indicate that the true submanifold possibly has some curvature, but that its estimation does not because of noise regime around the submanifold is "too high". Interesting, this remark challenges the very assumption of the existence of a submanifold: if the noise around the manifold is large, does the manifold assumption still hold?

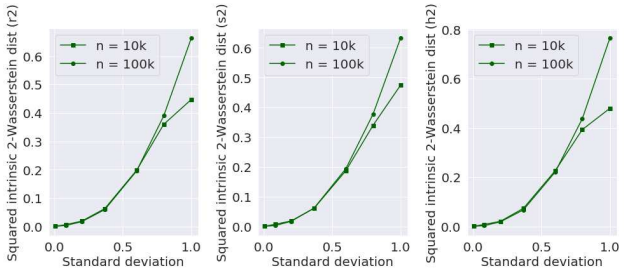

Figure 5. Goodness of fit for submanifold learning using the 2-Wasserstein distance. First column: $\mathbb{R}^2$; Second column: S^2; Third column: H_2.

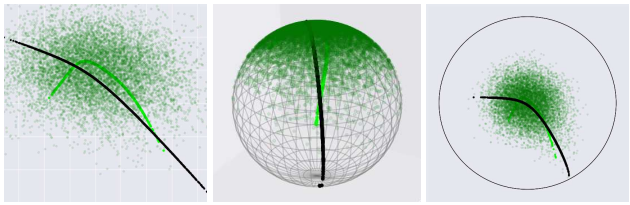

Figure 6. Data points (dark green) generated for $n = 10k$ and $\log \sigma^2 = -2$ from the true submanifolds shown in Figure 4 (light green). Learned submanifold (black). First column: $\mathbb{R}^2$; Second column: S^2; Third column: H_2 in its Poincaré disk representation.

6. Comparison of rVAE with submanifold learning methods

We perform experiments on simulated datasets to compare the following submanifold learning methods: PGA, VAE, rVAE, and VAE projected back on the pre-specified manifold. We generate datasets on the sphere using model (5) where the function f_θ is a fully connected neural network with two layers, and softplus nonlinearity. The latent space has dimension 1, and the inner layers have dimension 2. We consider different noise levels $\log \sigma^2 = \{-10, -2, -1, 0\}$ and sample sizes $n \in \{10k, 100k\}$.

We fit PGA using the tangent PCA approximation. The architecture of each variational autoencoder - VAE, rVAE and VAE projected - has the capability of recovering the true underlying submanifold correctly. Details on the architectures are provided in the supplementary materials. Figure 7 shows the goodness of fit of each submanifold learning procedure, in terms of the extrinsic 2-Wasserstein distance in the ambient Euclidean space $\mathbb{R}^3$. The PGA is systematically off, as shown in the Figures from the supplementary materials, therefore we did not include it in this plot.

We observe that rVAE outperforms the other submanifold learning methods. Its flexibility enables to outperforms PGA, and its geometric prior allows to outperforms VAE. It also outperforms the projected VAE, although the difference in performances is less significative. Projected VAEs

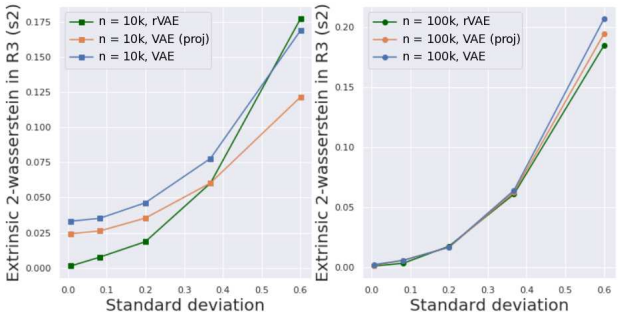

Figure 7. Quantitative comparison of the submanifold learning methods using the 2-Wasserstein distance in the embedding space $\mathbb{R}^3$. From left to right: ; quantitative comparison for $n = 10k$ and different values of σ; quantitative comparison for $n = 100k$ and different values of σ.

might be interesting for applications that do not require an intrinsic probabilistic model on the Riemannian manifold.

7. Experiments on brain connectomes

In the last section, we turn to the question that has originated this study: do brain connectomes belong to a submanifold of the SPD(N) manifold? We compare the methods of PCA, PGA, VAE and rVAE on resting-state functional brain connectome data from the "1200 Subjects release" of the Human Connectome Project (HCP) [27]. We use $n = 812$ subjects each represented by a 15×15 connectome. Details on the dataset are provided in the supplementary materials.

The VAE represents the brain connectomes as elements x of the vector space of symmetric matrices and is trained with the Frobenius metric. In contrast, the Riemannian VAE represents the brain connectomes as elements x of the manifold SPD(N), which we equip with the Riemannian Log-Euclidean metric. We chose equivalent neural network architectures for both models. Details on the architectures and the training are provided in the supplementary materials. We perform a grid search for the dimension of the latent space over $L \in \{10, 20, 40, 60, 80, 100\}$. The latent dimension L controls the dimension of the learned submanifold, as well as the model's flexibility.

Results from PCA and PGA do not reveal any low-dimensional linear or geodesic subspace, as we do not observe an "elbow" in Figure 8. Training a VAE enables us to ask the question: does a low-dimensional *nonlinear* subspace exist? The results are presented in Figure 9 (left): the VAE detects a 5D subspace. However, we observe that the subspace represents only around 30% of the variability.

Training rVAE allows to look for nongeodesic subspaces, which is a larger class than linear, geodesic and nonlinear subspaces. Therefore, rVAE uniquely allows to tackle the scientific question: what is the dimension of the subspace/submanifold of brain connectomes? Can we find

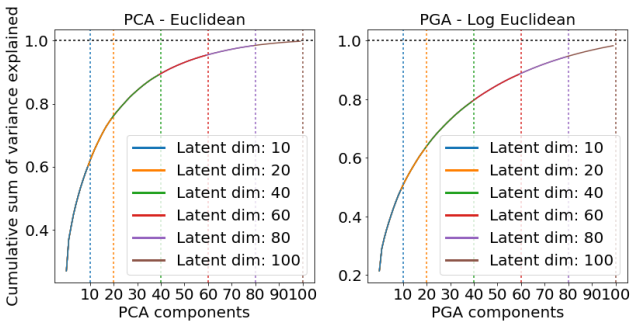

Figure 8. Cumulative sum of variance captured by the principal components, for Principal Component Analysis (left) and Principal Geodesic Analysis (right).

a nongeodesic subspace of dimension higher than 5? Figure 9 (right) answers the question: rVAE only detects a 5D nongeodesic subspace, again representing around 30% of the variability. This may mean that there is simply no submanifold of dimension higher than 5 in this dataset, as we have ruled out 4 classes of submanifolds through PCA, PGA, VAE and rVAE. We hope that this result will inspire researchers to run the 4 methods on another connectome dataset to see if this finding holds.

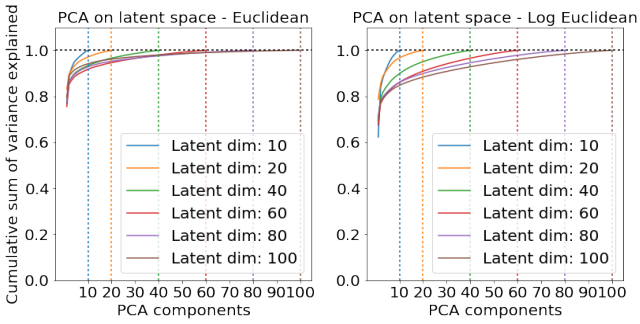

Figure 9. Cumulative sum of variance captured by the principal components within the latent space, for the VAE (left); Right: Riemannian VAE (right).

8. Conclusion

We introduced the Riemannian variational autoencoder (rVAEs), which is an intrinsic generalization of VAE for data on Riemannian manifolds and an extension of probabilistic principal geodesic analysis (PPGA) to nongeodesic submanifolds. The rVAE variational inference method also allows performing approximate, but potentially faster, inference in PPGA. We provided theoretical and experimental results on rVAE using the formalism of weighted submanifold learning.

References

[1] Alexandre Barachant, Stéphane Bonnet, Marco Congedo, and Christian Jutten. Classification of covariance matrices using a Riemannian-based kernel for BCI applications. *Neurocomputing*, 112:172–178, 2013. 1

[2] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003. 2

[3] Alexander M. Bronstein, Michael M. Bronstein, and Ron Kimmel. Generalized multidimensional scaling: A framework for isometry-invariant partial matching. *Proceedings of the National Academy of Sciences of the United States of America*, 103(5):1168–1172, 2006. 2

[4] Kui Yu Chang and Joydeep Ghosh. A unified model for probabilistic principal surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(1):22–41, 2001. 2

[5] Nikolai Chernov. *Circular and linear regression : fitting circles and lines by least squares*. Monographs on statistics and applied probability. CRC Press/Taylor & Francis, Boca Raton, 2011. 6

[6] Trevor Cox and Michael Cox. *Multidimensional Scaling*. Springer h edition, 2000. 2

[7] David Donoho and Carrie Grimes. Hessian eigenmaps: New locally linear embedding techniques for high-dimensional data. *TR2003-08, Dept. of Statistics.*, (650):1–15, 2003. 2

[8] Thomas Fletcher, Conglin Lu, Stephen M Pizer, and Sarang Joshi. Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE transactions on medical imaging*, 23(8):995–1005, 2004. 2

[9] Trevor Hastie and Werner Stuetzle. Principal Curves. *Journal of the American Statistical Association*, 84(406):502–516, 1989. 2

[10] Søren Hauberg. Principal Curves on Riemannian Manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(9):1915–1921, 2016. 2

[11] Stephan Huckemann, Thomas Hotz, and Axel Munk. Intrinsic shape analysis: Geodesic PCA for riemannian manifolds modulo isometric lie group actions. *Statistica Sinica*, 20(1):1–58, 2010. 2

[12] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*, 2014. 2, 3

[13] N. Miolane, A. Le Brigant, B. Hou, C. Donnat, M. Jorda, J. Mathe, X. Pennec, and S. Holmes. Geomstats: a python module for computations and statistics on manifolds. *Submitted to JMLR*, 2019. 6

[14] Victor M. Panaretos, Tung Pham, and Zhigang Yao. Principal flows. *Journal of the American Statistical Association*, 109(505):424–436, 2014. 2

[15] Karl Pearson. LIII. On lines and planes of closest fit to systems of points in space . *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901. 2

[16] Xavier Pennec. Intrinsic Statistics on Riemannian Manifolds: Basic Tools for Geometric Measurements. *Journal of Mathematical Imaging and Vision*, 25(1):127–154, 2006. 4

[17] Xavier Pennec. Barycentric subspace analysis on manifolds. *Annals of Statistics*, 46(6A):2711–2746, 2018. 2

[18] Mikhail Postnikov. *Riemannian Geometry*. Encyclopaedia of Mathem. Sciences. Springer, 2001. 6

[19] Danilo J. Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *Proceedings of the 31st International Conference on Machine Learning*, 2014. 3

[20] Sam T. Roweis and Lawrence K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290(22):2323–2326, 2000. 2

[21] Bernhard Schoelkopf, Alexander Smola, and Klaus-Robert Mueller. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. 1998. 2

[22] Hang Shao, Abhishek Kumar, and P. Thomas Fletcher. The riemannian geometry of deep generative models. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2018-June:428–436, 2018. 7

[23] Stefan Sommer, Francois Lauze, and Mads Nielsen. Optimization over geodesics for exact principal geodesic analysis. *Advances in Computational Mathematics*, 40(2):283–313, 2014. 2

[24] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319, 2000. 2

[25] Michael E. Tipping and Christopher M. Bishop. Probabilistic Principal Component Analysis. *Source: Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 61(3):611–622, 1999. 2, 6

[26] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schölkopf. Wasserstein Auto-Encoders. Technical report, 2018. 5

[27] David C. Van Essen, Stephen M. Smith, Deanna Barch, Timothy E. J. Behrends, Essa Yacoub, and Kamil Ugurbil. The WU-Minn Human Connectome Project: An Overview David. *Neuroimage*, 80:62–79, 2013. 8

[28] Kilian Q. Weinberger and Lawrence K. Saul. An introduction to nonlinear dimensionality reduction by maximum variance unfolding. *Proceedings of the National Conference on Artificial Intelligence*, 2:1683–1686, 2006. 2

[29] Miaomiao Zhang and P. Thomas Fletcher. Probabilistic principal geodesic analysis. *Advances in Neural Information Processing Systems*, pages 1–9, 2013. 2, 4