Geodesic Finite Mixture Models

Edgar Simo-Serra
esimo@iri.upc.edu
Carme Torras
torras@iri.upc.edu
Francesc Moreno-Noguer
fmoreno@iri.upc.edu

Institut de Robòtica i Informàtica
Industrial (CSIC-UPC)
08028, Barcelona, Spain

Abstract

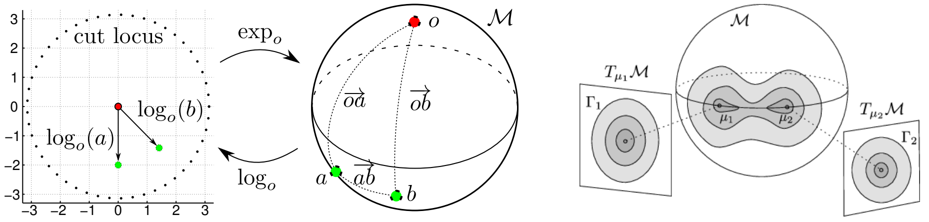
We present a novel approach for learning a finite mixture model on a Riemannian manifold in which Euclidean metrics are not applicable and one needs to resort to geodesic distances consistent with the manifold geometry. For this purpose, we draw inspiration on a variant of the expectation-maximization algorithm, that uses a minimum message length criterion to automatically estimate the optimal number of components from multivariate data lying on an Euclidean space. In order to use this approach on Riemannian manifolds, we propose a formulation in which each component is defined on a different tangent space, thus avoiding the problems associated with the loss of accuracy produced when linearizing the manifold with a single tangent space. Our approach can be applied to any type of manifold for which it is possible to estimate its tangent space. In particular, we show results on synthetic examples of a sphere and a quadric surface, and on a large and complex dataset of human poses, where the proposed model is used as a regression tool for hypothesizing the geometry of occluded parts of the body.

1 Introduction

The use of Riemannian manifolds and their statistics has recently gained popularity in a wide range of applications involving non-linear data modeling. For instance, they have been used to model shape changes in the brain [7], deformations of anatomical parts [10] and human motion [5, 32]. In this work we tackle the problem of approximating the Probability Density Function (PDF) of a potentially large dataset that lies on a *known* Riemannian manifold. We address this by creating a completely data-driven algorithm consistent with the manifold, i.e., an algorithm that yields a PDF defined exclusively on the manifold.

A standard procedure to operate on a manifold is to use the logarithmic map to project the data points onto the tangent space of the mean point on the manifold [10, 13, 32]. After this linearization, Euclidean statistics are computed and projected back to the manifold using the exponential map. This process is iteratively repeated until convergence of the computed statistics. Unfortunately, while this approximation is effective to model data with a reduced extent, it is prone to fail when dealing with data that covers wide regions of the manifold.

In the proposed finite mixture model, we overcome this limitation by simultaneously considering multiple tangent spaces, distributed along the whole manifold as seen in Fig. 1. We draw inspiration on the unsupervised algorithm from [9], which given data lying in an



(a) S^2 Geodesics

(b) Geodesic Finite Mixture Model

Figure 1: **Geodesics and Mixture Models.** (a) Representation of geodesics on the S^2 manifold. The tangent space ensures that $\|\log_o(x)\|$ is the true geodesic distance of $\overrightarrow{ox}$. However, $\|\log_o(a) - \log_o(b)\|$ is not the geodesic distance of $\overrightarrow{ab}$. (b) Illustration of the proposed mixture model approach. Each mixture component has its own tangent space, ensuring the consistency of the model while minimizing accuracy loss.

Euclidean space, automatically computes the number of model components that minimize a message length cost. By representing each component as a distribution on the tangent space at its corresponding mean on the manifold, we are able to generalize the algorithm to Riemannian manifolds and at the same time mitigate the accuracy loss produced when using a single tangent space. Furthermore, since our model is *semi-parametric*, we can handle an arbitrarily large number of samples. This is in contrast to existing *non-parametric* approaches [22] whose complexity grows with the training set size.

Results will show that our manifold-based finite mixture model can be effectively used both for non-linear data modeling and regression. In addition, we will show that the modeling of articulated structures can greatly benefit from our approach, allowing for instance, hallucinating subsets of missing joints of the human body without any specific training.

2 Related Work

Recently, there has been an influx of theoretical results in statistics on Riemannian manifolds [23] that have allowed for their widespread usage in modeling. For example, there exists several PCA generalizations to non-linear data such as the Principal Geodesic Analysis [10, 32] and the Geodesic PCA [13]. Yet, these methods only use one single tangent space located at the geodesic mean, which can lead them to have significant accuracy error when input data is spread out widely on the manifold. Other algorithms based on kernels [7] and Markov Chain Monte Carlo sampling [5] have been specifically used in regression tasks on manifolds, but they have not been applied to stochastic modeling problems.

There have also been recent developments addressing classification models on Riemannian manifolds. For binary cases, the classifier is usually built in a “flattened” version of the manifold, obtained via the tangent space [34]. Multiple category classification problems have been addressed by replacing the tangent space mapping with rolling maps [6], and by using extensions of the Kernel methods to Riemannian manifolds [16]. In any event, these approaches have been exclusively used to tackle classification problems, which are out of the scope of the current paper, focused on PDF modeling.

With regards to density estimation on Riemannian manifolds, various non-parametric approaches [21, 22] have been proven to be powerful. However, as their complexity is dependent on the number of training samples, they scale poorly for large datasets. In contrast, semi-parametric models such as the mixture model we propose here can handle large amounts of data efficiently. [1] is an interesting approach that is in spirit is similar to ours, as it proposes a technique to perform Expectation Maximization (EM) on manifolds. However, this work considers data-driven manifolds, resulting in a high computational overhead for large training sets. In addition, [1] neither estimates the number of clusters, nor makes use of the tangent space, which allows our model to be defined "on the manifold".

As an alternative to build probabilistic models, one could use Gaussian Processes (GP) strategies, such as GP-Latent Variable Models [18] and the constrained GP [35]. Again, despite some approximations to improve computation efficiency [19, 24], the algorithmic complexity of GP makes them unfeasible for large datasets, as those we consider here.

3 Geodesic Finite Mixture Model

We next describe our approach, starting with some basic notions on Riemannian geometry and statistics on manifolds. We then integrate these tools in a mixture modeling algorithm to build consistent generative models.

3.1 Manifolds, Geodesics and Tangent Spaces

Manifolds arise naturally in many real-world problems. One of the more well-known is the manifold representing spatial rotations. For example, when studying human motion, it is a common practice to use the spatial rotations of the different body parts to obtain a subject-agnostic representation of the whole body pose. This is usually done with angle representations that have an inherent periodicity and thus are not a vector space. By considering the Riemannian manifold of spatial rotations it is possible to use tangent spaces as a local vector space representation, and use powerful statistical tools based on Euclidean metrics. For an in depth description of Riemannian manifolds we refer the reader to [3].

Geodesic distances, which we shall denote as $d(\cdot,\cdot)$, are the shortest distance along the manifold between two arbitrary points. This distance is generally not equivalent to an Euclidean one. The tangent space is a local vector space representation where the Euclidean distances between the origin and arbitrary points correspond to the geodesic distances on the manifold. Yet, as seen in Fig. 1a, this correspondence does not hold for two arbitrary points. A point $v \in \mathcal{M}$ on the tangent space $T_p\mathcal{M}$ at $p \in \mathcal{M}$ can be mapped to the manifold $\mathcal{M}$ and back to the $T_p\mathcal{M}$ by using the exponential and logarithmic maps respectively:

$$\exp_p : \begin{array}{ccc} T_p\mathcal{M} & \longrightarrow & \mathcal{M} \\ v & \longmapsto & \exp_p(v) = x \end{array} , \qquad \log_p : \begin{array}{ccc} \mathcal{M} & \longrightarrow & T_p\mathcal{M} \\ x & \longmapsto & \log_p(x) = v \end{array} \tag{1}$$

In general there is no closed-form of the $\exp_p$ and $\log_p$ maps for an arbitrary manifold. There are, though, approximations for computing them in Riemannian manifolds [8, 31]. Additionally, efficient closed-form solutions exist for certain manifolds [25]. An interesting mapping for us will be the one between the unit sphere S^2 and its tangent space T_pS^2. Let $x = (x_1, x_2, x_3)^\top$, $y = (y_1, y_2, y_3)^\top$ be two unit spoke directions in S^2 and $v = (v_1, v_2)^\top$ a point in T_pS^2. The $\exp_p$ and $\log_p$ maps are in this case:

$$\exp_p(v) = R_p^{-1}\left(v_1 \frac{\sin\|v\|}{\|v\|}, v_2 \frac{\sin\|v\|}{\|v\|}, \cos\|v\|\right), \qquad \log_p(x) = \left(y_1 \frac{\theta}{\sin\theta}, y_2 \frac{\theta}{\sin\theta}\right) \tag{2}$$

where R_p is the rotation of p to the north pole, $\|v\|$=$(v_1^2+v_2^2)^{\frac{1}{2}}$, $y = R_p x$ and $\theta = \arccos(y_3)$.

3.2 Statistics on Tangent Spaces

While it is possible to define distributions on manifolds [23], we shall focus on approximating Gaussian PDFs using the tangent space. Since it is a vector space, we can compute statistics that, by definition of the tangent space, are consistent with the manifold. For instance, the mean of N points x_i on a manifold can be calculated as $\mu = \arg\min_p \sum_{i=1}^{N} d\left(x_i,\ p\right)^2$ [17]. This is optimized iteratively using the $\exp_p$ and $\log_p$ maps,

$$\mu(t+1) = \exp_{\mu(t)}\left(\frac{\delta}{N}\sum_{i=1}^{N}\log_{\mu(t)}\left(x_i\right)\right) , \tag{3}$$

until $\|\mu(t+1)-\mu(t)\| < \varepsilon$ for some threshold ε, with δ being the step size parameter. Given the mean, it is then possible to estimate the covariance on the tangent space:

$$\Sigma = \frac{1}{N}\sum_{i=1}^{N}\log_{\mu}(x_i)\log_{\mu}(x_i)^{\top} . \tag{4}$$

Knowing the mean value and the concentration matrix ($\Gamma = \Sigma^{-1}$) we can write the distribution that maximizes entropy on the tangent space as a normal distribution centered on the point $\mu \in \mathcal{M}$, corresponding to the origin ($\nu = 0$) in the tangent space:

$$\mathcal{N}_{\mu}(\nu,\ \Gamma) = a\exp\left(-\frac{\log_{\mu}(x)^{\top}\Gamma\log_{\mu}(x)}{2}\right) \tag{5}$$

where a is normalization term that ensures the normal distribution over the tangent space integrates to unity. If $T_{\mu}\mathcal{M} \equiv \mathbb{R}^D$ this term simplifies to $a^{-1} = \sqrt{(2\pi)^k\det(\Sigma)}$.

As the normalization term is dependent on the tangent space, it is not always easy to obtain. However, it is possible to approximate this normalization factor based on maximum entropy criterion computed directly on the manifold [23]. Since this situation is very unusual, it is not further explored in this paper.

3.3 Unsupervised Finite Mixture Modeling

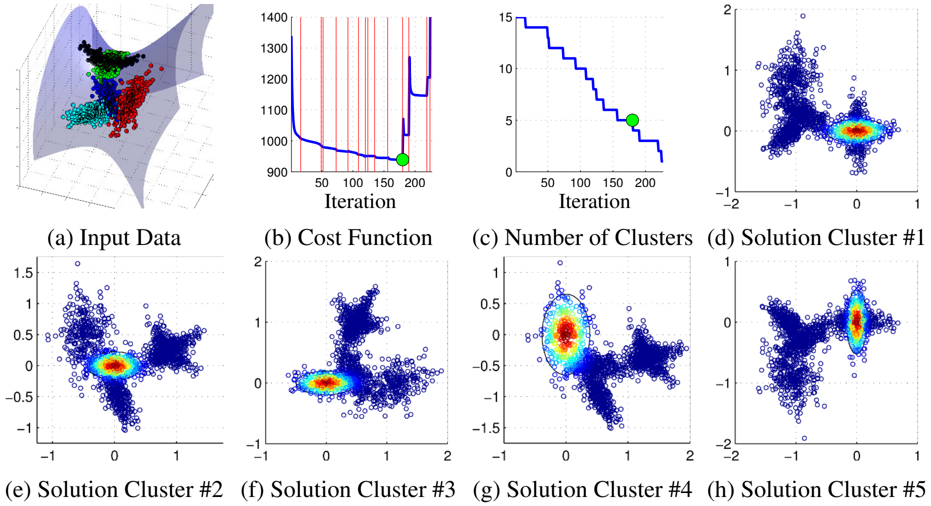
Our approach for fitting the mixture model relies on [9], a variant of the EM algorithm that uses the Minimum Message Length criterion (MML) to estimate the number of clusters and their parameters in an unsupervised manner.

Given an input dataset, this algorithm starts by randomly initializing a large number of mixtures. During the Maximization (M) step, a MML criterion is used to annihilate components that are not well supported by the data. In addition, upon EM convergence, the least probable mixture component is also forcibly annihilated and the algorithm continues until a minimum number of components is reached. [9] is designed to work with data in an Euclidean space. To use it in Riemannian manifolds, we modify the M-step as follows.

We define each mixture component with a mean μ_k and a concentration matrix $\Gamma_k = \Sigma_k^{-1}$ as a normal distribution on its own tangent space $T_{\mu_k}\mathcal{M}$:

$$p(x|\theta_k) \approx \mathcal{N}_{\mu_k}\left(0,\ \Sigma_k^{-1}\right) \tag{6}$$

with $\theta_k = (\mu_k, \Sigma_k)$. Remember that the mean μ_k is defined on the manifold $\mathcal{M}$, while the concentration matrix Γ_k is defined on the tangent space $T_{\mu_k}\mathcal{M}$ at the mean $\nu_k = 0$.



(a) Input Data (b) Cost Function (c) Number of Clusters (d) Solution Cluster #1

(e) Solution Cluster #2 (f) Solution Cluster #3 (g) Solution Cluster #4 (h) Solution Cluster #5

Figure 2: **Quadratic Surface Example.** (a) Section of the manifold with input data generated from 1500 points sampled from a mixture of 5 Gaussians. (b) Evolution of the cost function where vertical lines represent iterations in which a cluster is annihilated. The optimal mixture is marked with a green dot. (c) Evolution of the number of clusters. All the clusters from the solution are shown in (d) to (h).

Also note that the dimensionality of the space embedding the manifold is larger than the actual dimension of the manifold, which in its turn is equal to the dimension of the tangent space. That is, $\dim(\text{Embedding}(\mathcal{M})) > \dim(T_p\mathcal{M}) = \dim(\mathcal{M}) = D$. This dimensionality determines the total number of parameters D_θ specifying each component, and, as we will explain below, plays an important role during component annihilation process. For full covariance matrices it can be easily found that $D_\theta = D + D(D+1)/2$.

We next describe how the EM algorithm is extended from Euclidean to Riemmanian manifolds. Specifically, let us assume that K components survived after iteration $t-1$. Then, in the E-step we compute the *responsibility* that each component k takes for every sample x_i:

$$w_k^{(i)} = \frac{\alpha_k(t-1)p(x_i|\theta_k(t-1))}{\sum_{k=1}^{K}\alpha_k(t-1)p(x_i|\theta_k(t-1))} , \tag{7}$$

for $k = 1,\dots,K$ and $i = 1,\dots,N$, and where $\alpha_k(t-1)$ are the relative weights of each component k.

In the M-step we update the weight α_k, the mean μ_k and covariance Σ_k for each of the components as follows:

$$\alpha_k(t) = \frac{1}{N}\sum_i^N w_k^{(i)} = \frac{w_k}{N}, \qquad \mu_k(t) = \arg\min_p \sum_{i=1}^{N} d\left(\frac{N}{w_k}w_k^{(i)}x^{(i)},\, p\right)^2 \tag{8}$$

$$\Sigma_k(t) = \frac{1}{w_k}\sum_{i=1}^{N}\left(\log_{\mu_k(t)}(x^{(i)})\right)\left(\log_{\mu_k(t)}(x^{(i)})\right)^\top w_k^{(i)} \tag{9}$$

After each M-step, we follow the same annihilation criterion as in [9], and eliminate those components whose accumulated responsibility w_k is below a $D_\theta/2$ threshold. A score for the remaining components based on the Minimum Message Length is then computed. This EM process is repeated until the convergence of the score or until reaching a minimum number of components K_{min}. If this number is not reached, the component with the least responsibility is eliminated (even if it is larger than $D_\theta/2$) and the EM process is repeated. Finally, the configuration with minimum score is retained (see [9] for details), yielding a resulting distribution with the form

$$p(x|\theta) = \sum_{k=1}^{K} \alpha_k p(x|\theta_k) . \tag{10}$$

3.3.1 Regression.

To use the generative model in regression tasks, we need to split the mix into two components, $x = (x_A, x_B)$ with $\mu_k = (\mu_{k,A}, \mu_{k,B})$ and $\Gamma_k = \begin{bmatrix} \Gamma_{k,A} & \Gamma_{k,AB} \\ \Gamma_{k,BA} & \Gamma_{k,B} \end{bmatrix}$. The regression function can be written as:

$$p(x_A|x_B,\theta) = \frac{p(x_A,x_B|\theta)}{p(x_B|\theta_B)} = \frac{\sum_{k=1}^{K} \alpha_k p(x_B|\theta_{k,B}) p(x_A|x_B,\theta_k)}{\sum_{k=1}^{K} \alpha_k p(x_B|\theta_{k,B})} . \tag{11}$$

Observe that this is a new mixture model $p(x_A|x_B,\theta) = \sum_{k=1}^{K} \pi_k p(x_A|x_B,\theta_k)$, with weights:

$$\pi_k = \frac{\alpha_k p(x_B|\theta_{k,B})}{\sum_{j=1}^{K} \alpha_j p(x_B|\theta_{j,B})} , \tag{12}$$

and $p(x_A|x_B,\theta_k) = \mathcal{N}_{\mu_{k,A|B}}(\nu_{k,A|B}, \Gamma_{k,A|B})$, where the mean and concentration matrix can be found to be:

$$\nu_{k,A|B} = \Gamma_{k,AB}\Gamma_{k,B}^{-1} \log_{\mu_k,B}(x_B), \quad \Gamma_{k,A|B} = \Gamma_{k,A} - \Gamma_{k,AB}\Gamma_{k,B}^{-1}\Gamma_{k,BA} . \tag{13}$$

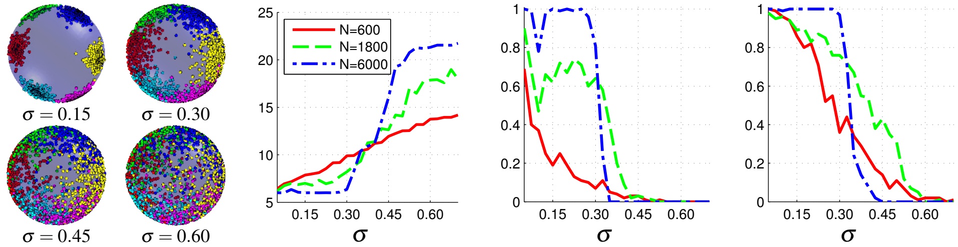
3.4 Implementation Considerations

While using the tangent spaces allows a consistent PDF representation, this comes at a price of higher computational cost. There are, though, several implementation considerations that can be taken into account to improve the efficiency. For instance, as mentioned in [9], we might consider using less expressive covariance matrices (e.g. diagonal ones). However, when using tangent spaces, there is no global coordinate frame representation, as the orthonormal basis of the tangent space depends on the point at which it is calculated. Thus, in general, purely diagonal matrices cannot be used.

Nevertheless, when working with a manifold which is the cartesian product of other manifolds such as $\mathcal{S}_{AB} = \mathcal{S}_A \times \mathcal{S}_B$, it is possible to use a block-diagonal matrix of the form:

$$\Sigma_{\mathcal{S}_{AB}} = \text{diag}(\Sigma_{\mathcal{S}_A}, \Sigma_{\mathcal{S}_B}) = \text{diag}(\Gamma_{\mathcal{S}_A}^{-1}, \Gamma_{\mathcal{S}_B}^{-1}) = \Gamma_{\mathcal{S}_{AB}}^{-1}$$

which by construction is a valid covariance matrix and avoids this issue. The null row and column elements highly simplify the computational cost. By using less expressive covariance matrices, the model has fewer degrees of freedom and generally converges in fewer iterations, in addition to requiring less training data.



(a) Example Input Data (b) Mean Clusters (c) $K = 6$ Solutions (d) $K \leq 9$ Solutions

Figure 3: **Effect of cluster size and number of samples.** We evaluate 100 times on distributions with increasing covariance sizes $\text{diag}(\sigma,\sigma)$ and three values of sampled points N on the S^2 sphere example. As evaluation criterion we consider the number K of estimated clusters. The true number of clusters is 6. (b) Mean number of solution clusters found. (c), (d) Ratio of solutions found with a number of clusters subject to different constraints.

In addition, while in the Euclidean case the mean of a set of points can be computed in closed form, when working with manifolds we need to do this iteratively. In our implementation this is required in the M-step, where the parameters θ_k are estimated as a weighted combination of terms. By considering only a subset of samples S such that $\sum_{i \in S} w_k^{(i)} > \varepsilon_s$ for a certain threshold ε_s, it is possible to improve the computational efficiency without sacrificing accuracy.

4 Results

We validate our algorithm on both synthetic and real data. Although we can handle any type of distribution, for the synthetic experiments we focus on Gaussians[1].

4.1 Synthetic Results

We present two experiments on 2D manifolds defined by

$$\mathcal{M} = \left\{ (x,y,z) \mid cx^2 + y^2 + z^2 = 1 \right\} . \tag{14}$$

For the first experiment we generate 1500 points from a mixture of 5 Gaussians on the manifold of Eq. (14) with $c = -2$, as shown in Fig. 2a. The distribution in this case is set so that there is large overlap, and two Gaussians even share the same mean. The algorithm is initialized with 15 clusters and uses the generic forms of the $\exp_p$ and $\log_p$ operators that rely on the derivative of the implicit manifold equation. Additionally the threshold ε_s is set to 0.999 to speed up the computations. In this scenario the underlying distribution is recovered as shown in Fig. 2.

For the second experiment we analyze the effects of the non-linearities of the $\exp_p$ and $\log_p$ operators by evaluating the method on a sphere (Eq. (14) with $c = 1$) using 6 clusters with parameters $\mu_i = [cos(i\pi/3), 0, sin(i\pi)/3]$ and $\Sigma_i = \text{diag}(\sigma,\sigma)$ for $i = 1,\ldots,6$ and for increasing values of σ. Several examples of input distributions with different covariances

[1] We plan to release the source code of the algorithm to allow other researchers to evaluate and use our method.

Table 1: Results on recovering an original distribution on the Sphere manifold with 6 and 8 clusters. Results obtained from 100 different evaluations and 1000 samples per cluster. We compare our method with using a single tangent space (1-TM) and Von Mises-Fisher distributions (vMF).

	6 Clusters		**8 Clusters**	
	Clusters	**Correct**	**Clusters**	**Correct**
Ours	6.09 (0.32)	0.92	8.00 (0.00)	1.00
1-TM	7.05 (1.38)	0.46	15.25 (2.17)	0.00
vMF	16.59 (1.71)	0.00	19.86 (2.35)	0.00

are shown in Fig. 3a. The effect of the number of input samples is seen by testing with $N = \{600, 1800, 6000\}$. We initialize with 30 clusters and $\varepsilon_s = 0.999$.

The results are shown in Fig. 3. With less spread Gaussians and little overlap, having more data can be seen to be beneficial. However, with more overlap and samples, generally the data gets partitioned into more clusters. These results seem to indicate that the algorithm tends to implicitly favor smaller Gaussians over larger ones, suggesting that there shouldn't be problems with approximating distributions. It is also worth mentioning that these results are for clustering. When estimating a probability density function, the number of clusters is not particularly important as long as the underlying density is properly approximated.

Finally, in order to evaluate the benefit of using multiple tangent spaces over a single one, we perform a comparison on the sphere manifold, in two situations: the same 6 clusters as in Fig. 3 with $\Sigma_i = \text{diag}(0.2, 0.3)$, and when fully covering the sphere with two additional clusters centered at $(0,1,0)$ and $(0,-1,0)$. We also compare against an approach that uses von Mises-Fisher (vMF) distributions [2] and it is specifically designed to cluster data on a sphere. For this, we use our implementation based on [9]. The results are shown in Table 1. In the 6-cluster case our algorithm retrieves the correct number of clusters in a 92% of the experiments, while one single plane only provides a 46% of success. Note that we evaluate the performance of the methods based only on the number of clusters, and not comparing the entire density probability. In the following subsection we will show that the distributions obtained with one single tangent plane are also much less representative as those obtained with the proposed approach. In the 8-cluster case our algorithm's performance improves and it always finds the correct clusters, while a single tangent space always fails, with an average of 15 clusters found. Using vMF distributions results in an oversegmentation of the data in both experiments. This is due to the fact that the vMF distributions use a single parameter κ for the concentration of the data, while our model allows for much more expressive covariances in the form of matrices. These results clearly show the advantage of using multiple tangent planes to better approximate manifold distributions.

4.2 Human Pose Results

Despite having devoted most of our discussion to the mathematical description of our approach and synthetic results that help to clarify its potential, we have to keep in mind that our main motivation is to approximate probability density functions for large datasets constrained on some manifold. This can be done to remove the need of additional criteria from existing approaches in which sampling from manifold-constrained data plays a fundamental role such as [20, 29] or provide a speed-up to the sampling [30]. To illustrate this point,

Table 2: Comparing Our method against von Mises Fisher distributions (vMF), a single tangent space (1-TM) and an Euclidean Gaussian Mixture Model (GMM) in the limb reconstruction task. We show results using three different metrics: Mean Geodesic Joint Error (MGJE), Mean Joint Error (MJE), and Mean Limb Length Error (MLLE).

	Scenario 1			**Scenario 2**			**Scenario 3**		
	MGJE	**MJE**	**MLLE**	**MGJE**	**MJE**	**MLLE**	**MGJE**	**MJE**	**MLLE**
Ours	0.446	105.8	0.0	0.468	110.1	0.0	0.349	81.7	0.0
vMF	0.481	114.5	0.0	0.568	134.8	0.0	0.470	110.2	0.0
1-TM	0.522	123.0	0.0	0.640	148.7	0.0	0.535	124.9	0.0
GMM	1.111	103.1	19.0	1.167	106.6	27.5	1.152	77.6	11.3

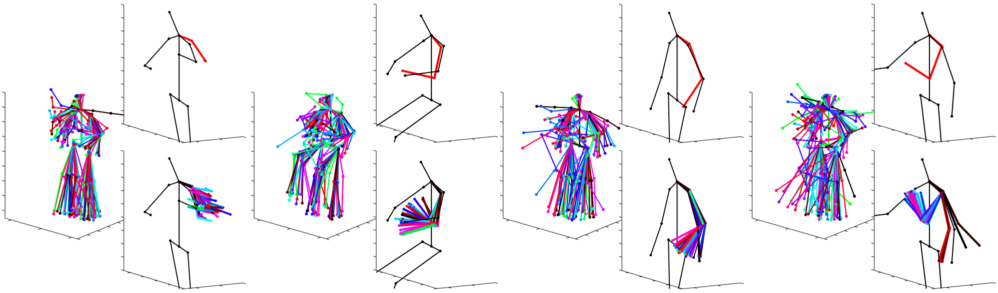
	Scenario 4			**Scenario 5**		
	MGJE	**MJE**	**MLLE**	**MGJE**	**MJE**	**MLLE**
Ours	0.458	108.2	0.0	0.597	135.7	0.0
vMF	0.496	118.0	0.0	0.698	162.3	0.0
1-TM	0.548	130.2	0.0	0.765	175.1	0.0
GMM	1.272	101.0	14.2	1.401	127.3	24.8

we evaluated our approach on the Human3.6M Dataset [14, 15] which has many subjects performing different activities.

As is common practice [15], we model the poses in an individual-agnostic way by normalizing all the limbs by their segments and only modeling the rotation. The rotation of each joint is represented on the S^2 manifold which then are arranged in a tree structure [32, 33]. We note that other manifolds do exist such as one defined by using forward kinematics [12], however, this approach does not have closed form solutions of the $\exp_p$ and $\log_p$ operators. A simplified model of the human with 15 joints is used because the dimensionality of the model greatly increases the amount of the required data. This results in a 24-dimensional manifold, represented by a block-diagonal covariance matrix with 46 non-zero elements.

We divide the dataset into its five base scenarios: S_1, upper body movements; S_2, full body upright variations; S_3, walking variations; S_4, variations while seated on a chair; and S_5, sitting down on the floor. We train on subjects 1, 5, 6, 7, 8, 9 and 11 for each scenario using only the first of the actions in each scenario and test on the second one. The algorithm uses 2000 initial clusters and $\varepsilon_s = 0.999$ for all scenarios.

We evaluate on the task of hallucinating limbs, i.e., probabilistically recovering missing parts x_A from the rest of the body x_B. This is done by obtaining the expectation of the posterior $p(x_A|x_B,\theta)$ using a combination of the conditional means (Eq. (13)) at each tangent space with $x_A = \arg\min_p \sum_{k=1}^{K} d\left(\pi_k \nu_{k,A|B},\ p\right)^2$. We compare our method with von Mises Fisher distributions (vMF), a single tangent space (1-TM) and a Gaussian Mixture Model in Euclidean space (GMM). To compensate the fact that our approach is individual-agnostic, was trained with the GMM with same individual that was being tested. We define three metrics: Mean Geodesic Joint Error (MGJE), that accounts for the rotation error; Mean Joint Error (MJE), representing the Euclidean distance with the ground truth in mm; and Mean Limb Length Error (MLLE), which is the amount of limb length deformation in mm. All four models are trained using exactly the same parameters for all scenarios. The results are summarized in Table 2. Observe that while the MJE is similar for both our method and GMM, our approach has roughly one third of the MGJE error. Additionally, as our model is coherent with the manifold, there is no limb deformation (MLLE). Our model



(a) Scenario 1 (b) Scenario 2 (c) Scenario 3 (d) Scenario 4

Figure 4: **Regression Example**. We use the trained mixtures to perform regression on the left arm (3 limbs) of subjects on the Human3.6M dataset. On the left we plot 30 random samples from the full mixture. On the top-right we plot 30 random samples from the mixture of the conditional distribution corresponding to a particular frames. On the bottom-right the mean of the mixture is shown in red. The black lines indicate the input sample. As shown in Scenario 4, the distribution can be multimodal, however, one of the modes is very close to the true limb position.

also consistently outperforms the vMF and 1-TM approaches, which once again shows that multiple tangent spaces with Gaussians can better capture the underlying distribution. The 1-TM approach has the worst MJE error but we can see it has a MGJE much closer to our model than the GMM. We also outperform the vMF model. We show four specific examples of the estimated distributions for our model in Fig. 4.

5 Conclusions

We have presented a novel data-driven approach for modeling the probability density function of data located on a Riemannian manifold. By using a mixture of distributions, each with its own tangent space, we are able to ensure the consistency of the model while avoiding most of the linearization error caused by using one single tangent space. The approach has been experimentally validated on various synthetic examples that highlight their ability to both correctly approximate manifold distributions and discover the underlying data structure. Furthermore, the approach has been tested on a large and complex dataset, where it is shown to outperform the traditionally used Euclidean Gaussian Mixture Model, von Mises distributions and using a single tangent space in a regression task.

Future works include exploiting the potential of this algorithm on different manifolds and datasets, for instance for representing and further matching tree-like biologic structures [26, 27, 28]. We have presented results using Gaussian distributions and have focused on the S^2 manifold. However, the algorithm presented here should work with any distribution and on any manifold for which the exponential and logarithm map operators are provided, as shown on a quadratic surface. For example, it should be possible to initially estimate unknown and non-parameterizable manifolds [4] and use approximate operators [11].

Acknowledgements

This work has been partially funded by Spanish Ministry of Economy and Competitiveness under projects PAU+ DPI2011-27510 and ERA-Net Chistera project ViSen PCIN-2013-047.

References

[1] C. Archambeau and M. Verleysen. Manifold constrained finite gaussian mixtures. In *Computational Intelligence and Bioinspired Systems*, pages 820–828. Springer, 2005.

[2] A. Banerjee, I. S. Dhillon, J. Ghosh, S. Sra, and G. Ridgeway. Clustering on the unit hypersphere using von mises-fisher distributions. *Journal of Machine Learning Research*, 6(9):1345–1382, 2005.

[3] W. M. Boothby. *An Introduction to Differentiable Manifolds and Riemannian Geometry. Revised 2nd Ed.* Academic, New York, 2003.

[4] M. Brand. Charting a Manifold. In *Neural Information Processing Systems*, 2003.

[5] M. A. Brubaker, M. Salzmann, and R. Urtasun. A Family of MCMC Methods on Implicitly Defined Manifolds. *Journal of Machine Learning Research - Proceedings Track*, 22:161–172, 2012.

[6] R. Caseiro, P. Martins, J. F. Henriques, F. S. Leite, and J. Batista. Rolling riemannian manifolds to solve the multi-class classification problem. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[7] B. C. Davis, E. Bullitt, P. T. Fletcher, and S. Joshi. Population Shape Regression from Random Design Data. In *International Conference on Computer Vision*, 2007.

[8] J.-P. Dedieu and D. Nowicki. Symplectic methods for the approximation of the exponential map and the newton iteration on riemannian submanifolds. *Journal of Complexity*, 21(4):487 – 501, 2005.

[9] M. Figueiredo and A. Jain. Unsupervised Learning of Finite Mixture Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):381–396, 2002.

[10] P. Fletcher, C. Lu, S. Pizer, and S. Joshi. Principal Geodesic Analysis for the Study of Nonlinear Statistics of Shape. *IEEE Transactions on Medical Imaging*, 23(8):995–1005, 2004.

[11] O. Freifeld and M. J. Black. Lie Bodies: A Manifold Representation of 3D Human Shape. In *European Conference on Computer Vision*, 2012.

[12] S. Hauberg, S. Sommer, and K. S. Pedersen. Natural metrics and least-committed priors for articulated tracking. *Image and Vision Computing*, 30(6):453–461, 2012.

[13] S. Huckemann, T. Hotz, and A. Munk. Intrinsic Shape analysis: Geodesic PCA for Riemannian Manifolds Modulo Isometric Lie Group Actions. *Statistica Sinica*, 20: 1–100, 2010.

[14] C. Ionescu, F. Li, and C. Sminchisescu. Latent Structured Models for Human Pose Estimation. In *International Conference on Computer Vision*, 2011.

[15] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.

[16] S. Jayasumana, R. Hartley, M. Salzmann, H. Li, and M. Harandi. Kernel methods on the riemannian manifold of symmetric positive definite matrices. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[17] H. Karcher. Riemannian center of mass and mollifier smoothing. *Communications on Pure and Applied Mathematics*, 30(5):509–541, 1977.

[18] N. D. Lawrence. Probabilistic Non-linear Principal Component Analysis with Gaussian Process Latent Variable Models. *Journal of Machine Learning Research*, 6:1783–1816, 2005.

[19] N. D. Lawrence. Learning for Larger Datasets with the Gaussian Process Latent Variable Model. *Journal of Machine Learning Research - Proceedings Track*, 2:243–250, 2007.

[20] F. Moreno-Noguer and P. Fua. Stochastic exploration of ambiguities for nonrigid shape recovery. *IEEE Transactions Pattern Analylis and Machine Intelligence*, 35(2):463–475, 2013.

[21] A. Ozakin and A. Gray. Submanifold Density Estimation. In *Neural Information Processing Systems*, 2009.

[22] B. Pelletier. Kernel Density Estimation on Riemannian Manifolds. *Statistics & Probability Letters*, 73(3):297 – 304, 2005.

[23] X. Pennec. Intrinsic Statistics on Riemannian Manifolds: Basic Tools for Geometric Measurements. *J. Math. Imaging Vis.*, 25(1):127–154, July 2006.

[24] J. Quiñonero-candela, C. E. Rasmussen, and R. Herbrich. A Unifying View of Sparse Approximate Gaussian Process Regression. *Journal of Machine Learning Research*, 6: 1939–1959, 2005.

[25] S. Said, N. Courtry, N. L. Bihan, and S. Sangwine. Exact Principal Geodesic Analysis for data on $SO(3)$. In *European Signal Processing Conference*, 2007.

[26] E. Serradell, A. Romero, R. Leta, C. Gatta, and F. Moreno-Noguer. Simultaneous correspondence and non-rigid 3d reconstruction of the coronary tree from single x-ray images. In *International Conference on Computer Vision*, 2011.

[27] E. Serradell, P. Glowacki, J. Kybic, F. Moreno-Noguer, and P. Fua. Robust non-rigid registration of 2d and 3d graphs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[28] E. Serradell, P. Glowacki, J. Kybic, F. Moreno-Noguer, and P. Fua. Robust elastic 2d/3d geometric graph matching. In *SPIE Medical Imaging*, 2012.

[29] E. Simo-Serra, A. Ramisa, G. Alenyà, C. Torras, and F. Moreno-Noguer. Single Image 3D Human Pose Estimation from Noisy Observations. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[30] E. Simo-Serra, A. Quattoni, C. Torras, and F. Moreno-Noguer. A Joint Model for 2D and 3D Pose Estimation from a Single Image. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[31] S. Sommer, A. Tatu, C. Chen, D. Jurgensen, M. De Bruijne, M. Loog, M. Nielsen, and F. Lauze. Bicycle Chain Shape Models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[32] S. Sommer, F. Lauze, S. Hauberg, and M. Nielsen. Manifold Valued Statistics, Exact Principal Geodesic Analysis and the Effect of Linear Approximations. In *European Conference on Computer Vision*, 2010.

[33] M. Tournier, X. Wu, N. Courty, E. Arnaud, and L. Reveret. Motion compression using principal geodesics analysis. In *Computer Graphics Forum*, volume 28. Wiley Online Library, 2009.

[34] O. Tuzel, F. Porikli, and P. Meer. Pedestrian Detection via Classification on Riemannian Manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10): 1713–1727, 2008.

[35] A. Varol, M. Salzmann, P. Fua, and R. Urtasun. A Constrained Latent Variable Model. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.