Metrics for probabilistic geometries

A. Tossi, S. Hauberg, A. Vellido, N. D. Lawrence

UAI 2014

October 16, 2014

Outline

Background

- Dimensionality reduction (GPLVM)
- Riemannian Geometry

Motivation

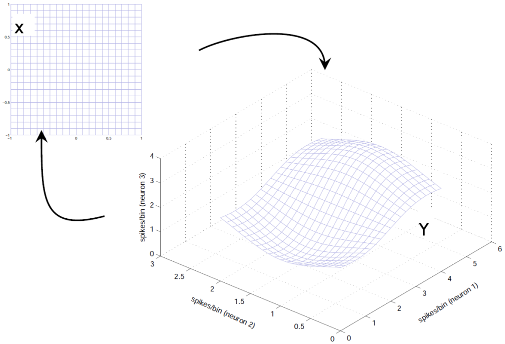
- Euclidian metric has undesired properties in Low dim space

Proposed work

- GPLVM + Riemannian Geometry

Background I : Dimensionality reduction

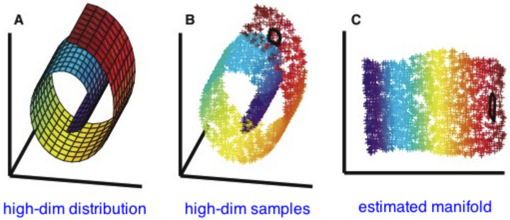
High dimensional data D live on a low dimensional manifold d



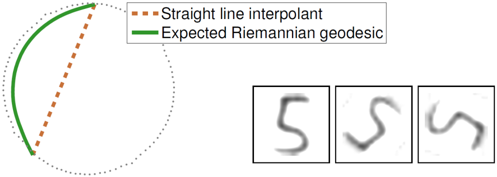
Embedding: $x \leftrightarrow y$

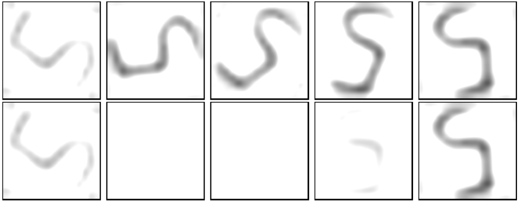
Background I : Dimensionality reduction

Toy example: The swiss roll



Motivation





Background I : Dimensionality reduction

Many existing methods

- PCA, Isomap, LLE, Autoencoders, MVU ...

Background I : Dimensionality reduction

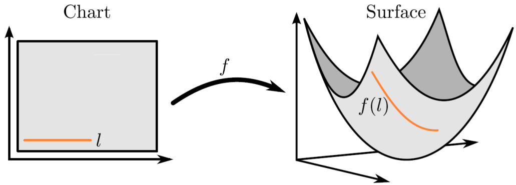
Many existing methods

- PCA, Isomap, LLE, Autoencoders, MVU ...

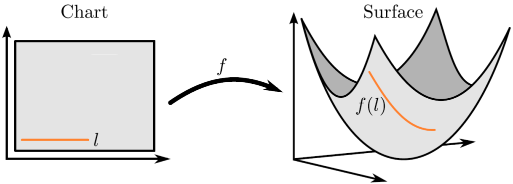
Focus on a probabilistic one:

- Gaussian Process Latent Variable Model (GPLVM)

Background II : A bit of Riemannian Geometry



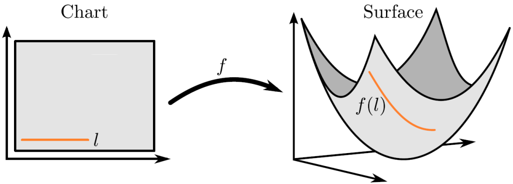
Background II : A bit of Riemannian Geometry



Curve

- $l \ : [0,1] \rightarrow \mathbb{R}^2$(curve = $1D$ object)
- $f \circ l \ : [0,1] \rightarrow \mathbb{R}^3$

Background II : A bit of Riemannian Geometry

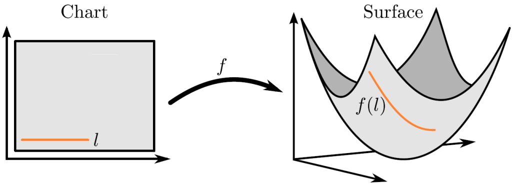


Curve

- l : $[0,1] \to \mathbb{R}^2$(curve = $1D$ object)
- $f \circ l$: $[0,1] \to \mathbb{R}^3$

Length of curve?

Background II : A bit of Riemannian Geometry



Curve

- $l \ : [0,1] \rightarrow \mathbb{R}^2$(curve = $1D$ object)
- $f \circ l \ : [0,1] \rightarrow \mathbb{R}^3$

Length of curve?

$$\begin{aligned} Length(f(l)) &= \int_0^1 \left\| \frac{\partial f(l(t))}{\partial t} \right\| dt \\ &= \int_0^1 \left\| J \frac{\partial l(t)}{\partial t} \right\| dt, \quad J = \frac{\partial f}{\partial l}|_{l=l(t)} \end{aligned}$$

Background II : A bit of Riemannian Geometry

Definition

(Riemannian metric). A Riemannian metric tensor G on a manifold $\mathcal{M}$ is symmetric and positive definite matrix which defines a smoothly varying inner product

For $x \in \mathcal{M}$,and $a, b \in T_x\mathcal{M}$ (tangent space at x)

$$\langle a, b\rangle_x = a^T J^T J b = a^T G(x) b$$

Background II : A bit of Riemannian Geometry

Definition

(Riemannian metric). A Riemannian metric tensor G on a manifold $\mathcal{M}$ is symmetric and positive definite matrix which defines a smoothly varying inner product

For $x \in \mathcal{M}$,and $a, b \in T_x\mathcal{M}$ (tangent space at x)

$$\langle a, b\rangle_x = a^T J^T J b = a^T G(x) b$$

$$Length(f(l)) = \int_0^1 \sqrt{\langle l'(t), l'(t)\rangle_{l(t)}} dt$$

Background II : A bit of Riemannian Geometry

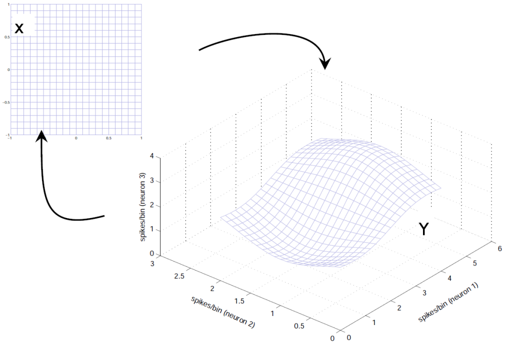
Definition

(Geodesic curve). Given two points $x_1, x_2 \in \mathcal{M}$, a *geodesic* curve is a length minimizing curve connecting the points

$$l_g = \underset{l}{argmin} \ \text{Length}(l)$$

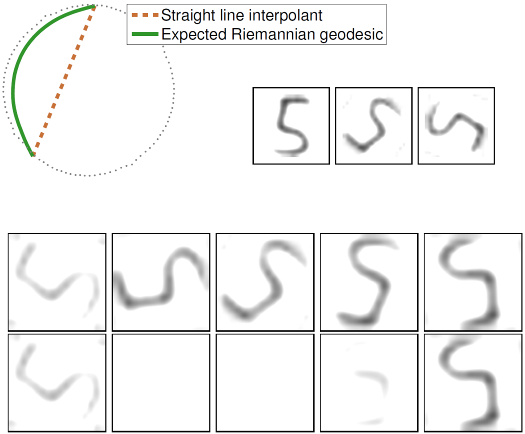
$$l(0) = x_1, \ l(1) = x_2$$

Back to Dimensionality reduction



Embedding: $x \leftrightarrow y$

Motivation



GP Regression

GP Regression

$$y|x = f(x),\ f \sim \mathcal{GP}(K)$$

GP Regression

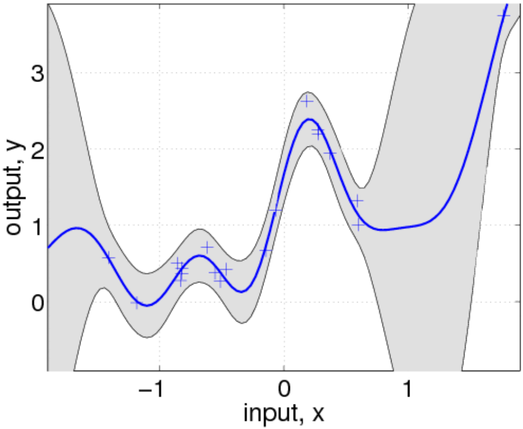
$$y|x = f(x),\ f \sim \mathcal{GP}(K)$$

$$y^*|x^*, X, Y \sim \mathcal{GP}(\tilde{m}, \tilde{K})$$

GP Regression

$$y|x = f(x),\ f \sim \mathcal{GP}(K)$$

$$y^*|x^*, X, Y \sim \mathcal{GP}(\tilde{m},\tilde{K})$$



GPLVM

What kind of embedding?

Linear

$$y = Ax$$

GPLVM

What kind of embedding?

Non-linear

$$y = f(x)$$

GPLVM

What kind of embedding?

Non-linear + noise

$$y = f(x) + \epsilon$$

GPLVM

What kind of embedding?

Non-linear + prior on f

$$f \sim \mathcal{GP}(0, \mathcal{K}_0)$$

$$y = f(x)$$

GPLVM

What kind of embedding?

Non-linear + prior on f + noise

$$y|x \sim \mathcal{GP}(0, K(x, x))$$

GPLVM

What kind of embedding?

Non-linear + noise + prior

$$x \sim \mathcal{N}(0, \mathbb{I})$$
$$y|x \sim \mathcal{GP}(0, K(x, x))$$

GPLVM

What kind of embedding?

Non-linear + noise + prior

$$x \sim \mathcal{N}(0, \mathbb{I})$$

$$y_j | x \sim \mathcal{GP}(0, K(x, x)) \text{ (per feature)}$$

GPLVM

Embedding (optimize over latent)

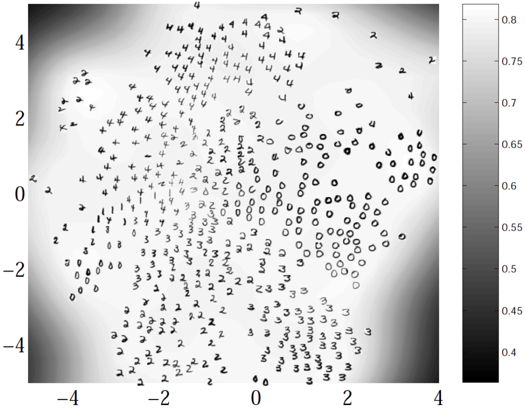
$$\begin{aligned}\hat{X} &= \underset{X}{argmax}\; p(Y|X)p(X) \\ &= \underset{X}{argmax} \int df\; p(Y|f)p(f|X)p(X)\end{aligned}$$

GPLVM

Posterior on mapping function (uncertainty)

$$p(f|\hat{X}, Y) \propto p(Y|\hat{X})$$

GPLVM



What metric to use for $\mathcal{M}$?

We have data Y and its representation $\hat{X}$, thus a posterior embedding

$$f^*|\hat{X}, Y, x^* \textbf{ (Gaussian)}$$

What metric to use for $\mathcal{M}$?

We have data Y and its representation $\hat{X}$, thus a posterior embedding

$$f^*|\hat{X}, Y, x^* \textbf{ (Gaussian)}$$

Which induces a posterior on the gradient of f

$$\nabla f^*|\hat{X}, Y, x^* \textbf{(Gaussian)}$$

What metric to use for $\mathcal{M}$?

We have data Y and its representation $\hat{X}$, thus a posterior embedding

$$f^*|\hat{X}, Y, x^* \textbf{ (Gaussian)}$$

Which induces a posterior on the gradient of f

$$\nabla f^*|\hat{X}, Y, x^* \textbf{(Gaussian)}$$

We can choose

$$G(x) = E\left[J^T J\right] \textbf{ (non-central Wishart)}, \text{ with } J_{ij} = \frac{df_i}{dx_j}$$

Some details

Posterior on gradient per feature

$$p(J|X,Y) = \prod_{j=1}^{p} \mathcal{N}(\mu_J^{(j)}, \underbrace{\Sigma_J}_{shared})$$

Some details

Posterior on gradient per feature

$$p(J|X,Y) = \prod_{j=1}^{p} \mathcal{N}(\mu_J^{(j)}, \underbrace{\Sigma_J}_{shared})$$

Metric Tensor

$$G = J^T J = \mathcal{W}(p, \Sigma_J, \mathbb{E}[J^T]\mathbb{E}[J])$$

Some details

Posterior on gradient per feature

$$p(J|X,Y) = \prod_{j=1}^{p} \mathcal{N}(\mu_J^{(j)}, \underbrace{\Sigma_J}_{shared})$$

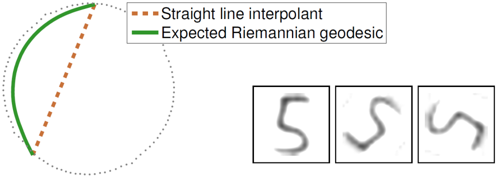
Metric Tensor

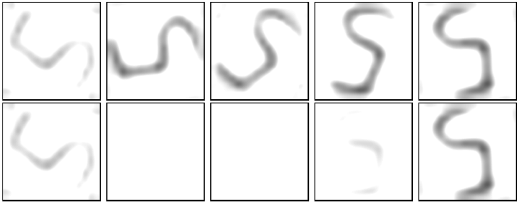
$$G = J^T J = \mathcal{W}(p, \Sigma_J, \mathbb{E}[J^T]\mathbb{E}[J])$$

Expected Metric Tensor

$$E[G] = E\left[J^T J\right] = \mathbb{E}[J^T]\mathbb{E}[J] + \underbrace{p\Sigma_J}_{uncertainty}$$

Motivation



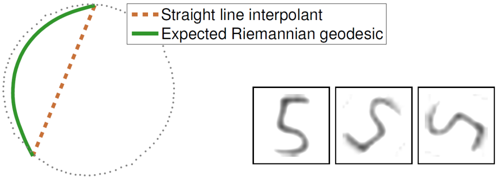


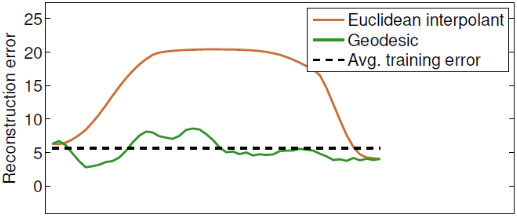
Experiments

- Rotating objects $\rightarrow$dimred $\rightarrow$ interpolate **[Reconstruction error]**
- Motion capture $\rightarrow$(dynamic) dimred $\rightarrow$ interpolate **[Limb length preservation]**

Experiments

- Rotating objects $\rightarrow$dimred $\rightarrow$ interpolate **[Reconstruction error]**





$$\text{Avg. Training error} = \frac{1}{N}\sum_{n=1}^{N} \|\mathbb{E}[f(x_n)] - y_n\|$$

Experiments

- Motion capture →(dynamic) dimred → interpolate **[Limb length preservation]**