Parametric information geometry with the package Geomstats

Antoine Collas - Postdoc

Work done with Alice Le Brigant, Jules Deschamps and Nina Miolane
Paper at ACM Transactions on Mathematical Software

https://github.com/geomstats/geomstats

What is a statistical manifold ?

Random vector, negative log-likelihood

$$\boldsymbol{x} \sim f(.;\theta), \quad \theta \in \mathcal{M}$$

$$\mathcal{L}\left(\theta; \boldsymbol{x}\right) = -\log f(\boldsymbol{x}; \theta)$$

Fisher information metric

$$\langle \xi, \eta \rangle_{\theta}^{\mathrm{FIM}} = \mathrm{vec}(\xi)^T \underbrace{\mathbb{E}_{\boldsymbol{x} \sim f(.;\theta)} \left[\mathrm{Hess}\, \mathcal{L}\left(\theta; \boldsymbol{x}\right)\right]}_{\text{Fisher information matrix}} \mathrm{vec}(\eta)$$

(Set of constraints, Fisher information metric) = a Riemannian manifold

Example: covariance matrices

$$\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}), \quad \theta = \boldsymbol{\Sigma} \in \mathcal{S}_p^{++}$$

$$\langle \boldsymbol{\xi}, \boldsymbol{\eta} \rangle_{\boldsymbol{\Sigma}}^{\mathcal{S}_p^{++}} = \mathrm{Tr}(\boldsymbol{\Sigma}^{-1} \boldsymbol{\xi} \boldsymbol{\Sigma}^{-1} \boldsymbol{\eta})$$

Code example: geodesic interpolation of binomial distributions

For well-known statistical manifolds:

- Fisher information metric, exponential/logarithmic maps, geodesics, and distances,
- closed-form expressions or numerical solvers from the **numerics module** of Geomstats.

```
from geomstats import backend as gs
from geomstats.information_geometry.binomial import BinomialDistributions

manifold = BinomialDistributions(5)

point_a = .4
point_b = .7

times = gs.linspace(0, 1, 100)
geodesic = manifold.metric.geodesic(initial_point=point_a, end_point=point_b)(times)

middle = geodesic(.5)
print(middle)

>>>  0.5550055679356352
```

Code example: custom manifold

Custom statistical manifolds from its probability distribution function:

- Fisher information metric from automatic differentiation,
- geodesics computed with the **numerics module** of Geomstats.

```python
class MyInformationManifold(InformationManifoldMixin):
    def __init__(self):
        self.dim = 2
    def point_to_pdf(self, point):
        means = point[..., 0]
        stds = point[..., 1]
        def pdf(x):
            constant = (1. / gs.sqrt(2 * gs.pi * stds**2))
            return  constant * gs.exp(-((x - means) ** 2) / (2 * stds**2))
        return pdf

metric = FisherRaoMetric(
    information_manifold=MyInformationManifold(), support=(-10, 10))
```

Machine learning

Compatible with

- NumPy, Autograd, PyTorch, and TensorFlow,
- **learning module** of Geomstats: K-means, K-medoids, Nearest centroid classifier, PCA, ...

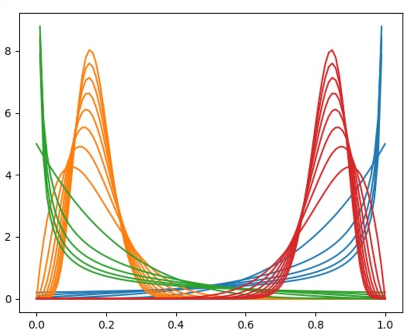

Figure 1: Probability density functions of beta distributions and a K-means clustering (colours) using the statistical manifold of beta distributions.