Geometric data analysis
Lecture 6/7 – Probability distributions

Jean Feydy
HeKA team, Inria Paris, Inserm, Université Paris-Cité

Thursday, 9am–12pm – 7 lectures
Faculté de médecine, Hôpital Cochin, rooms 2001 + 2005
Validation: project + quizz
Recap of the previous lectures

To mitigate the **curse of dimensionality**, we use:

- **Expert** knowledge: high-quality features.
- Relevant **families** of functions: kernels, convolutional networks.
- Relevant **neighborhood** structures: graphs.

Main challenge: local implementation $\Rightarrow$ global understanding.

Produce **guidelines** and **insights** for practitioners.
Lecture 5 – From discrete graphs to **continuous spaces**:  
- The Poincaré disk.  
- Local metrics and geodesics.

**Lecture 6** – From discrete samples to **continuous distributions**:  
- **Why do we care** about probability distributions?  
- Information **geometry**, kernels and optimal transport.  
- **Lab session** on gradient descent.

⟹ Chapter 3 of my PhD thesis, *Geometric data analysis, beyond convolutions*. 
What is a probability distribution?
Probability distribution \( \alpha = \text{weights } a_i \text{ at locations } x_i \)

**Histogram:**
- **variable** weights \( a_i \),
- **fixed** locations \( x_i \).

**Sample:**
- **fixed** weights \( 1/N \),
- **variable** locations \( x_i \).

**Weighted point cloud:**
- **variable** weights \( a_i \),
- **variable** locations \( x_i \).

Discrete sum \( \alpha = \sum_{i=1}^{N} a_i \delta_{x_i} \implies \) Continuous density \( \alpha = \int_x a(x) \, dx \).

Today, we assume that \( a \geq 0 \) and sums up to 1.
We must handle both **discrete** and **continuous** distributions.

We must choose if $\alpha$ is closer to $\beta$ (same mean value) or to $\gamma$ (same support).
One-sample test: 
- discrete observation $\alpha$, 
- continuous model $\beta$.

Two-sample test: 
- two discrete observations $\alpha$ and $\beta$.

Null hypothesis: $\alpha$ and $\beta$ come from the same distribution.

Test: reject if $d(\alpha, \beta)$ is too large.
Example: Splitting a population evenly for a clinical trial

**Problem 1:** ensure that the treatment and control groups have similar characteristics.

**Problem 2:** given a large population, pick a group of control patients that have similar characteristics to our treated patients.
**Application 2: Classification = regression in a space of distributions**

**Linear** regression:
- Encode class labels as **integer numbers**
  \[ l(x) \in \{1, 2, 3\} . \]
- Predict a **score** \( s(x) \) at every location \( x \).
- Minimize the **least square error**:
  \[
  \frac{1}{N} \sum_{i=1}^{N} |l(x) - s(x)|^2.
  \]

Massive **bias** depending on the **ordering** of the labels.

2 input features, 3 classes.
**Logistic** regression:

- Encode class labels as **probability distributions** \( \delta(x) \in \mathbb{P}\{1, 2, 3\} \).
- Predict a vector of **scores** \( s_i(x) \) at every location \( x \) and turn it into a probability distribution using the **SoftMax**: 
  \[
  \alpha(x) = \left( e^{s_1(x)}, e^{s_2(x)}, e^{s_3(x)} \right) / \sum e^{s_i(x)}
  \]
- Minimize the **relative entropy**:
  \[
  \frac{1}{N} \sum_{i=1}^{N} KL(\delta(x), \alpha(x)) .
  \]

**Invariant** to the **ordering** of the labels.
Generative Adversarial Networks and Variational Auto-Encoders minimize a distance between a synthetic sample and a reference data sample.

Diffusion and score-based models estimate a gradient of the distance to the support of a reference data sample.
Application 4: Shape registration [KCC17]

Curve: one weight per **segment**.

Surface: one weight per **triangle**.

Segmentation mask: one weight per **voxel**.

Encoding shapes as distributions guarantees an **invariance to resamplings**.

We may work with **basic** \((x, y, z)\) coordinates or with **better features**.
Registration algorithms **minimize a distance** between a deformable model $\alpha$ and the fixed target $\beta$. 

0. Input data
1. Pre-alignment
2. Deep registration
3. Fine-tuning
Application 5: Meta-analyses on histograms and distributions

3D shape **texture**

\[ \sim \text{Distribution of } \textbf{curvatures} \]

\[ \kappa_1 \geq \kappa_2 \text{ on the surface.} \]

**UMAP representation** of a population of textures, from the matrix of Wasserstein **distances between curvature histograms**.

Distances enable the processing of **populations of histograms**.

This is relevant to make **group-level** analyses.
A point about implementations

**Histogram:**
- **explicit** weights $a_i$,
- **implicit** locations $x_i$.

**Sample:**
- **implicit** weights $1/N$,
- **explicit** locations $x_i$.

**Weighted point cloud:**
- **explicit** weights $a_i$,
- **explicit** locations $x_i$.

Depending on the application, we may choose a **different encoding** for our distributions.
A point about implementations

**Histogram:**
- **explicit** weights $a_i$, 
- **implicit** locations $x_i$.

**Sample:**
- **implicit** weights $1/N$, 
- **explicit** locations $x_i$.

**Weighted point cloud:**
- **explicit** weights $a_i$, 
- **explicit** locations $x_i$.

Understanding that **different implementations** correspond to **the same operation** is key to insightful research in the field.
A point about implementations

Convolution of the **density map** $a[i, j]$ with a filter $g[i, j]$.

Additive noise:

$$x_i \mapsto x_i + w_i$$
where $w_i \sim \mathcal{N}(0, \sigma^2)$.

Soft distance:

$$\log\text{-likelihood } \ell(x) = \log \left( \sum_i a_i e^{-\|x-x_i\|^2/2\sigma^2} \right).$$

Understanding that **different implementations** correspond to **the same operation** is key to insightful research in the field.
A point about notations

If \( \alpha = \sum_{i=1}^{N} a_i \delta_{x_i} \) is a probability distribution and \( f : x \mapsto f(x) \in \mathbb{R} \) is a continuous function,

\[
\sum_{i=1}^{N} a_i f(x_i) = \int f(x) \, d\alpha(x) = \langle \alpha , f \rangle = \mathbb{E}_{X \sim \alpha} [f(X)].
\]

To study spaces of probability distributions, the \( \langle \alpha , f \rangle \) notation is superior as it highlights the linearity with respect to both distributions and functions:

\[
\langle \frac{1}{2} \alpha + \frac{1}{2} \beta , f \rangle = \frac{1}{2} \langle \alpha , f \rangle + \frac{1}{2} \langle \beta , f \rangle ,
\]

\[
\langle \alpha , f + g \rangle = \langle \alpha , f \rangle + \langle \alpha , g \rangle .
\]
Major distances between distributions
When designing a distance between histograms:

- Should we leverage the distance $\|x - y\|$ on the “ground space” of labels?
- How harshly should we penalize errors on the estimation of the support?
The total variation distance

The space of probability distributions on

\[ \{x_1, \ldots, x_K\} \]

is a **simplex** of dimension \( K - 1 \).

The Total Variation is the L1–Manhattan distance:

\[
TV(\alpha, \beta) = \sum_i |a(x_i) - b(x_i)|.
\]

This distance:

- **Maxes out** at 2 with **disjoint** supports.
- Pays no attention to \( \|x_i - x_j\| \).
- Pays no attention to **unlikely events**.
If $\beta = (b(1), \ldots, b(K))$ is a model distribution on $\{1, \ldots, K\}$, the likelihood of observing a sample $x$ is $L_\beta(x) = b(x)$.

Assuming independence, the joint likelihood of a sample $(x_1, \ldots, x_N)$ is:

$$L_\beta(x_1, \ldots, x_N) = b(x_1) \cdots b(x_N).$$

Finding a sample $(x_1, \ldots, x_N)$ that maximizes the likelihood is equivalent to minimizing:

$$\ell_\beta(x_1, \ldots, x_N) = - \frac{1}{N} \log [L_\beta(x_1, \ldots, x_N)] = \frac{1}{N} \sum_{i=1}^{N} \log \left[ \frac{1}{b(x_i)} \right]$$

If the $x_i$ are drawn independently according to a data distribution $\alpha$, this converges to:

$$\ell_\beta(\alpha) = \lim_{N \to +\infty} \sum_{k=1}^{K} \frac{\# \{i \mid x_i = k\} \log [1/b(k)]}{N} = \sum_{k=1}^{K} a(k) \log [1/b(k)]$$
In practice, the data distribution \( \alpha \) is fixed and we try to find a model distribution \( \beta \) which is as likely as possible.

This is equivalent to minimizing the relative entropy or Kullback–Leibler divergence:

\[
\text{KL}(\alpha, \beta) = \ell_\beta(\alpha) - \ell_\alpha(\alpha) = \sum_{k=1}^{K} a(k) \log \left[ \frac{a(k)}{b(k)} \right].
\]

We have that \( \text{KL}(\alpha, \alpha) = 0 \) and \( \text{KL}(\alpha, \beta) \geq 0 \), since \( \log \) is concave:

\[
\log \left[ \frac{b(k)}{a(k)} \right] \leq \frac{b(k)}{a(k)} - 1
\]

\[
\implies \quad \log \left[ \frac{a(k)}{b(k)} \right] \geq 1 - \frac{b(k)}{a(k)}
\]

\[
\implies \quad \sum_{k=1}^{K} a(k) \log \left[ \frac{a(k)}{b(k)} \right] \geq \sum_{k=1}^{K} a(k) \left[ 1 - \frac{b(k)}{a(k)} \right] = 0.
\]
First properties of the relative entropy

\[ \text{KL}(\alpha, \beta) = \sum_{k=1}^{K} a(k) \log \left( \frac{a(k)}{b(k)} \right) = \int x a(x) \log \left( \frac{a(x)}{b(x)} \right) \, dx : \]

- Is **not symmetric** – remember it as \( \text{KL}(\text{data} \mid \text{model}) \).
- Is tied to an assumption of **independence**.
- Historically: compression on communication networks \( \implies \) .zip format.

Crucially, the relative entropy:

- Pays no attention to \( \|x_i - x_j\| \).
- Pays **a lot** of attention to **unlikely events**: \( \log(0^+) = -\infty \).
The Gauss map defines a **parametric surface**:

\[
\mathcal{N} : (m, \sigma) \in \mathbb{R} \times \mathbb{R}_+ \mapsto \mathcal{N}(m, \sigma) \in \mathbb{P}(\mathbb{R}) .
\]

Direct computations show that:

\[
\text{KL}(\mathcal{N}(m + dm, \sigma + d\sigma), \mathcal{N}(m, \sigma)) = \frac{1}{2} dm^2 + d\sigma^2 + o(dm^2, d\sigma^2)
\]

\[
\Rightarrow \quad \text{Poincaré metric on the upper half-plane.}
\]

With its **invariance to translation and scalings**, the relative entropy induces a **hyperbolic** geometry on the surface of Gaussian distributions.
Kernel norms: recover compatibility with the addition

For sources \( \alpha = \sum_i a_i \delta_{x_i} \) and targets \( \beta = \sum_j b_j \delta_{y_j} \), choose a symmetric function \( g \) that induces a convolution kernel \( k = g \ast g \) and use:

\[
d_k(\alpha, \beta) = \| g \ast \alpha - g \ast \beta \|^2_{L^2}
\]

\[
= \langle \alpha - \beta, k \ast (\alpha - \beta) \rangle
\]

\[
= \sum_i \sum_j a_i a_j k(x_i, x_j)
- 2 \sum_i \sum_j a_i b_j k(x_i, y_j)
+ \sum_i \sum_j b_i b_j k(y_i, y_j).
\]
Kernel norms: recover compatibility with the addition

**Kernel norms** (aka. Hilbert or Sobolev norms, Maximum Mean Discrepancies):

- Are **quadratic** with respect to the weights $a_i$ and $b_j$.
- Are compatible with the addition – the geodesic from $\alpha$ to $\beta$ is:

  $$t \in [0, 1] \mapsto (1 - t) \alpha + t \beta.$$

- Have **wildly different behaviors** depending on $k(x, y)$: see the lab session.

Crucially, these formulas:

- Pay **a lot** of attention to $\|x_i - y_j\|$.
- Pay little attention to **unlikely events**, except if they are associated to **large values of** $k(x, y)$. 
Optimal transport (OT) generalizes sorting to spaces of dimension $D > 1$

If $A = (x_1, \ldots, x_N)$ and $B = (y_1, \ldots, y_N)$ are two clouds of $N$ points in $\mathbb{R}^D$, we define:

$$\text{OT}(A, B) = \min_{\sigma \in \mathcal{S}_N} \frac{1}{2N} \sum_{i=1}^{N} \| x_i - y_{\sigma(i)} \|^2$$

Generalizes sorting to metric spaces.

**Linear problem** on the permutation matrix $P$:

$$\text{OT}(\alpha, \beta) = \min_{P \in \mathbb{R}^{N \times N}} \sum_{i,j=1}^{N} P_{i,j} \cdot \frac{1}{2} \| x_i - y_j \|^2,$$

s.t. $P_{i,j} \geq 0$, $\sum_j P_{i,j} = a_i$, $\sum_i P_{i,j} = b_j$.

Each source point is transported onto the target.

$\sigma : [1, 5] \rightarrow [1, 5]$
Practical use

Alternatively, we understand OT as:

- Nearest neighbor projection + incompressibility constraint.
- Fundamental example of linear optimization over the transport plan $P_{i,j}$.

This theory induces two main quantities:

- The transport plan $P_{i,j} \simeq$ the optimal mapping $x_i \mapsto y_{\sigma(i)}$.
- The “Wasserstein” distance $\sqrt{\text{OT}(A, B)}$. 
The optimal transport plan

Before

After
The optimal transport plan

Before

After
The optimal transport plan

Before

After
The optimal transport plan
The Wasserstein metric on statistical manifolds [PC18]

The Gauss map defines a **parametric surface**:

\[ \mathcal{N} : (m, \sigma) \in \mathbb{R} \times \mathbb{R}_+ \mapsto \mathcal{N}(m, \sigma) \in \mathbb{P}(\mathbb{R}) . \]

Direct computations show that:

\[
2 \text{OT}(\mathcal{N}(m_1, \sigma_1), \mathcal{N}(m_2, \sigma_2)) = (m_1 - m_2)^2 + (\sigma_1 - \sigma_2)^2.
\]

\[ \implies \text{Euclidean metric on the upper half-plane.} \]

Optimal transport **lifts the geometry of the sample space** to the surface of Gaussian distributions.
Two canonical distances between Gaussian distributions [PC18]

Gaussians + **Wasserstein** metric  
=  **Euclidean**.

Gaussians + relative **entropy**  
=  **Poincaré**.
Geometric solutions to least square problems [AC11]

Barycenter $A^* = \arg\min_A \sum_{i=1}^{4} \lambda_i \text{Loss}(A, B_i)$.

**Euclidean** barycenters.
$\text{Loss}(A, B) = \|A - B\|_{L^2}^2$

**Wasserstein** barycenters.
$\text{Loss}(A, B) = \text{OT}(A, B)$
How should we solve the OT problem?

Key dates for discrete optimal transport with N points:

- [Kan42]: **Dual** problem of Kantorovitch.
- [Kuh55]: **Hungarian** methods in $O(N^3)$.
- [Ber79]: **Auction** algorithm in $O(N^2)$.
- [KY94]: **SoftAssign** = Sinkhorn + simulated annealing, in $O(N^2)$.
- [GRL+98, CR00]: **Robust Point Matching** = Sinkhorn as a loss.
- [Cut13]: Start of the **GPU era**.
- [Mér11, Lév15, Sch19]: **multi-scale** solvers in $O(N \log N)$.

- **Solution**, today: **Multiscale Sinkhorn algorithm, on the GPU.**

  $\implies$ Generalized **QuickSort** algorithm,
  $\approx O(N \log N)$ if D is small, fast $O(N^2)$ otherwise.
Scaling up optimal transport to anatomical data

Progresses of the last decade add up to a $\times 100 - \times 1000$ acceleration:

Sinkhorn GPU $\times 10 \rightarrow$ + KeOps $\times 10 \rightarrow$ + Annealing $\times 10 \rightarrow$ + Multi-scale

With a precision of 1%, on a gaming GPU:

```
pip install geomloss
+ modern GPU (1 000 €)
```

10k points in 30-50ms

100k points in 100-200ms
Recap on classical distances between probability distributions

The **Total Variation**:  
- **Invariant** to the ground metric \( \| x_i - y_j \| \), only cares about **large** weights \( a_i \) and \( b_j \).

The **relative entropy** KL:  
- **Invariant** to the ground metric \( \| x_i - y_j \| \), cares about the **ratio** \( a_i / b_j \).

**Kernel** norms:  
- More or less **faithful** to the ground metric **depending** on \( k \), easy to scale on GPUs.

**Optimal transport** distances:  
- **Extremely faithful** to the ground metric \( \| x_i - y_j \| \).  
- **Scalability** is recent – still open on general graphs and manifolds.
Open problem 1: Topology-aware distances

“OT that preserves the neighborhood structure”? 

The problem has been studied for decades:

- **Optimization**: Quadratic assignment…
- **Optimal Transport**: Gromov–Wasserstein…
- **Fluid mechanics**: Camassa–Holm equation…
- **Shape analysis**: LDDMM, SVF…
- **Statistics**: Stein Variational Gradient Descent…
- **Deep learning**: Neural ODEs…

Mature tools exist but remain $\geq 100$ slower than Optimal Transport.
Open problem 2: The curse of dimensionality

In **high dimension**, the matrix of Euclidean distances stops being informative.

Standard kernels and OT metrics are overwhelmed by **statistical noise**.

How can we compute **meaningful distances and gradients**?

GANs and VAEs **minimize a distance** between a **synthetic sample** and a **reference data sample**.
Dual norms: a fundamental insight from functional analysis

\[
\text{Loss}(\alpha, \beta) = \max_{f \in B} \langle \alpha - \beta, f \rangle,
\]

look for \(\theta^* = \arg \min_{\theta} \max_{f \in B} \langle \alpha(\theta) - \beta, f \rangle\)

- \(B = \{ \| f \|_\infty \leq 1 \} \implies \text{Loss} = \text{TV norm}:\)
  - Zero geometry, always saturates on disjoint samples.
  - **Too many** test functions.
- \(B = \{ \| f \|_{L^2}^2 + \| \nabla f \|_{L^2}^2 + \cdots \leq 1 \} \implies \text{Loss} = \text{kernel norm}:\)
  - Screening artifacts – see lab session.
  - In high dimension, samples are at equal distance from each other.
  
  \textbf{“Smooth”} functions are either “constant” or “bounded”: fall back on TV behavior.
Dual norms: link with the GANs literature

\[
\text{Loss}(\alpha, \beta) = \max_{f \in B} \langle \alpha - \beta, f \rangle,
\]

look for \( \theta^* = \arg\min_{\theta} \max_{f \in B} \langle \alpha(\theta) - \beta, f \rangle \)

- \( B = \{ f \text{ is 1-Lipschitz} \} \implies \text{Loss} = \text{Wasserstein-1}: \)
  - Modern solvers are nearly as efficient as a **closed formula**.
  - **Useless** in \((\mathbb{R}^{512 \times 512}, \| \cdot \|_2)\): the ground cost makes no sense.
- \( B \approx \{ f \text{ is 1-Lipschitz} \} \cap \{ f \text{ is a CNN} \} \implies \text{Loss} = \text{Wasserstein–GAN} : \)
  - Use **perceptual** test functions.
  - No simple formula: use **gradient ascent**.
    Leads to a cumbersome min-max optimization.
Open problem 2: Understand the impact of domain-specific test functions $f$

Similar story for **diffusion models**: we use **CNNs** (U-Nets...) to predict the gradient of the distance to the set of **natural images**.

The **interplay** between **mathematical insights** derived from toy models and **numerical experiments** on modern hardware is at the **heart of ML research**.

Let’s play with gradient descent to **build an intuition** about classical formulas!

Diffusion and score-based models estimate a gradient of the **distance to the support** of a **reference data sample**.
References
M. Agueh and G. Carlier.

**Barycenters in the Wasserstein space.**


Dimitri P Bertsekas.

**A distributed algorithm for the assignment problem.**

Haili Chui and Anand Rangarajan.

**A new algorithm for non-rigid point matching.**


Marco Cuturi.

**Sinkhorn distances: Lightspeed computation of optimal transport.**

Jean Feydy.

**Data science workshop notes.**


Session 12.

Jean Feydy.

**Geometric data analysis, beyond convolutions.**

Steven Gold, Anand Rangarajan, Chien-Ping Lu, Suguna Pappu, and Eric Mjolsness.

**New algorithms for 2d and 3d point matching: Pose estimation and correspondence.**


Leonid V Kantorovich.

**On the translocation of masses.**

Irene Kaltenmark, Benjamin Charlier, and Nicolas Charon.

A general framework for curve and surface comparison and registration with oriented varifolds.


Harold W Kuhn.

The Hungarian method for the assignment problem.

Jeffrey J Kosowsky and Alan L Yuille.

The invisible hand algorithm: Solving the assignment problem with statistical physics.


Bruno Lévy.

A numerical algorithm for l2 semi-discrete optimal transport in 3d.

Quentin Mérigot.

A multiscale approach to optimal transport.


Gabriel Peyré and Marco Cuturi.

Computational optimal transport.

A. Savin.

**Lion at the berlin zoo.**


Art Libre.

Bernhard Schmitzer.

**Stabilized sparse scaling algorithms for entropy regularized transport problems.**

Zhengyang Shen, Jean Feydy, Peirong Liu, Ariel H Curiale, Ruben San Jose Estepar, Raul San Jose Estepar, and Marc Niethammer.

**Accurate point cloud registration with robust optimal transport.**