Fundamentals of machine learning

Course 10: Density estimation & generative models

Titouan Vayer & Mathurin Massias
email: titouan.vayer@inria.fr, mathurin.massias@inria.fr,

March 25, 2025

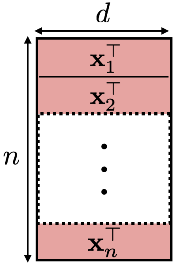
ENS DE LYON

Table of contents

Table of contents

Unsupervised dataset

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_n^\top \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1d} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nd} \end{bmatrix}$$



Unsupervised learning

- The dataset contains the samples $(\mathbf{x}_i)_{i=1}^n$ where n is the number of samples of size d.
- d and n define the dimensionality of the learning problem.
- Data stored as a matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ that contains the training samples as rows.

Understanding the data

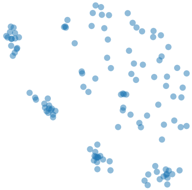
- The samples come from a certain distribution $p(\mathbf{x})$ ($\mathbf{x}_i \sim p(\mathbf{x})$).
- $p(\mathbf{x})$ is unknown !
- Density estimation: find $\hat{p}(\mathbf{x}) \approx p(\mathbf{x})$.
- One can generate new samples from this approximate distribution.

Understanding the data

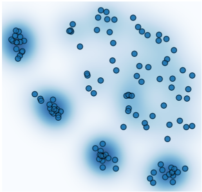
- The samples come from a certain distribution $p(\mathbf{x})$ ($\mathbf{x}_i \sim p(\mathbf{x})$).
- $p(\mathbf{x})$ is unknown !
- Density estimation: find $\hat{p}(\mathbf{x}) \approx p(\mathbf{x})$.
- One can generate new samples from this approximate distribution.
- $p(\mathbf{x})$ is usually complicated: find a understandable/compact representation of it.
- Clustering: group points together.
- Find most "representative" points of $p(\mathbf{x})$.

Table of contents

Probability density estimation



$\Longrightarrow$



Objective

$$\{\mathbf{x}_i\}_{i=1}^n \quad \Rightarrow \quad \hat{p} \in \mathcal{P}(\mathbb{R}^d)$$

- Estimate a probability density $\hat{p}(\mathbf{x})$ from the IID samples in the data.
- Probability density : $\hat{p}(\mathbf{x}) \geq 0, \ \forall \mathbf{x}$ and $\int \hat{p}(\mathbf{x}) d\mathbf{x} = 1$.
- Optional : generate new data from $\hat{p}(\mathbf{x})$.

Parameters

- Type of distribution (Histogram, Gaussian, ...).
- Parameters of the law $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

Methods

- Gaussian mixture.
- Parzen/kernel density estimation.
- Generative neural networks.

Maximum likelihood estimation

Principle

- Given a parametrized distribution $p(\mathbf{x}|\boldsymbol{\theta})$, find the most likely parameters $\boldsymbol{\theta}^\star$ given observed data $(\mathbf{x}_i)_{i\in[\![n]\!]}$.
- Hope for $p(\mathbf{x}) \approx p(\mathbf{x}|\boldsymbol{\theta}^\star)$.
- Maximize the likelihood:

$$\max_{\boldsymbol{\theta}} \text{Likelihood}(\boldsymbol{\theta}) = p(\mathbf{x}_1, \cdots, \mathbf{x}_n|\boldsymbol{\theta}) \overset{i.i.d.}{=} \prod_i^n p(\mathbf{x}_i|\boldsymbol{\theta}) \qquad (1)$$

Maximum likelihood estimation

Principle

- Given a parametrized distribution $p(\mathbf{x}|\boldsymbol{\theta})$, find the most likely parameters $\boldsymbol{\theta}^\star$ given observed data $(\mathbf{x}_i)_{i \in [\![n]\!]}$.
- Hope for $p(\mathbf{x}) \approx p(\mathbf{x}|\boldsymbol{\theta}^\star)$.
- Maximize the likelihood:

$$\max_{\boldsymbol{\theta}} \text{ Likelihood}(\boldsymbol{\theta}) = p(\mathbf{x}_1, \cdots, \mathbf{x}_n|\boldsymbol{\theta}) \overset{i.i.d.}{=} \prod_i^n p(\mathbf{x}_i|\boldsymbol{\theta}) \tag{1}$$

Multivariate Gaussian distribution

- $p_{\mathcal{N}}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ the density of a multivariate Gaussian distribution

$$p_{\mathcal{N}}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-d/2} \det(\boldsymbol{\Sigma})^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right) .$$

- $\boldsymbol{\mu} \in \mathbb{R}^d$ the mean, $\boldsymbol{\Sigma} \succ 0$ the covariance matrix.
- The MLE estimates $(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}})$ is known and given by

$$\hat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i \text{ and } \hat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{i=1}^n (\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})^\top .$$

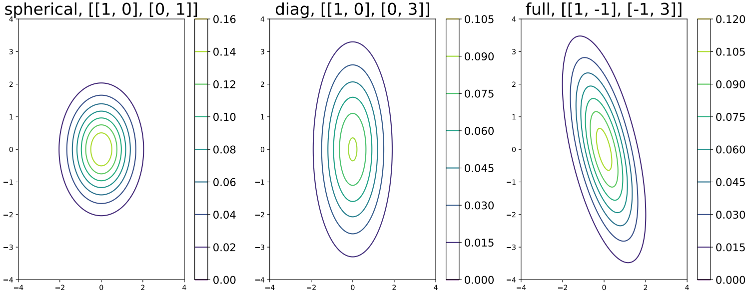
Maximum likelihood estimation

Principle

- Given a parametrized distribution $p(\mathbf{x}|\boldsymbol{\theta})$, find the most likely parameters $\boldsymbol{\theta}^\star$ given observed data $(\mathbf{x}_i)_{i\in[\![n]\!]}$.
- Hope for $p(\mathbf{x}) \approx p(\mathbf{x}|\boldsymbol{\theta}^\star)$.
- Maximize the likelihood:

$$\max_{\boldsymbol{\theta}} \text{ Likelihood}(\boldsymbol{\theta}) = p(\mathbf{x}_1, \cdots, \mathbf{x}_n|\boldsymbol{\theta}) \overset{i.i.d.}{=} \prod_i^n p(\mathbf{x}_i|\boldsymbol{\theta}) \tag{1}$$

Visualizing 2D Gaussian $\boldsymbol{\mu} = 0, \boldsymbol{\Sigma}$



The principle of GMM

Mixture of Gaussians

- Look for $p(\mathbf{x}) \approx p(\mathbf{x}|\boldsymbol{\theta})$ where, for $\boldsymbol{\theta} = (\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)_{k \in [\![K]\!]}$,
$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k p_{\mathcal{N}}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$
- $\pi_k \geq 0, \sum_{k=1}^{K} \pi_k = 1$ are the weights of each Gaussian.
- "Mixture" of multiple Gaussian.

The principle of GMM

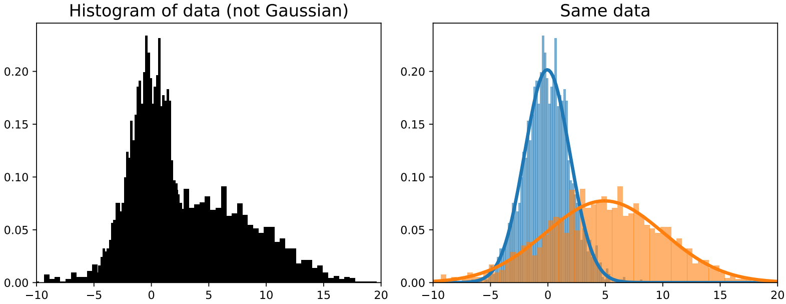
Mixture of Gaussians

- Look for $p(\mathbf{x}) \approx p(\mathbf{x}|\boldsymbol{\theta})$ where, for $\boldsymbol{\theta} = (\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)_{k \in [\![K]\!]}$,

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k p_{\mathcal{N}}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- $\pi_k \geq 0, \sum_{k=1}^{K} \pi_k = 1$ are the weights of each Gaussian.
- "Mixture" of multiple Gaussian.

Why ?



The principle of GMM

Mixture of Gaussians

- Look for $p(\mathbf{x}) \approx p(\mathbf{x}|\boldsymbol{\theta})$ where, for $\boldsymbol{\theta} = (\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)_{k \in [\![K]\!]}$,

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k p_{\mathcal{N}}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- $\pi_k \geq 0, \sum_{k=1}^{K} \pi_k = 1$ are the weights of each Gaussian.
- "Mixture" of multiple Gaussian.

Interpretation in terms of random variables

$\mathbf{x}_1, \cdots, \mathbf{x}_n \sim \text{GMM}(\boldsymbol{\theta})$ if:

- $z_1, \cdots, z_n \sim \text{Multinomial}(\boldsymbol{\pi}, 1)$ (clusters of each point).
- z_i represents the latent cluster for datapoint $\mathbf{x}_i$.
- $\mathbf{x}_i | z_i \sim \mathcal{N}(\boldsymbol{\mu}_{z_i}, \boldsymbol{\Sigma}_{z_i})$ *i.i.d.*

EM algorithm

Maximizing the likelihood

- There is no closed form for $\max_{\boldsymbol{\theta}}$ Likelihood($\boldsymbol{\theta}$).
- We often rely on the Expectation-Maximization algorithm (EM)
  Dempster, Laird, and Rubin 1977

EM algorithm

Maximizing the likelihood

- There is no closed form for $\max_{\boldsymbol{\theta}} \text{Likelihood}(\boldsymbol{\theta})$.
- We often rely on the Expectation-Maximization algorithm (EM) Dempster, Laird, and Rubin 1977
- **Step 1:** Initialize $\boldsymbol{\theta}^{(0)} = (\pi_k^{(0)}, \boldsymbol{\mu}_k^{(0)}, \boldsymbol{\Sigma}_k^{(0)})_{k \in [\![K]\!]}$. Then alternate:

EM algorithm

Maximizing the likelihood

- There is no closed form for $\max_{\boldsymbol{\theta}} \text{Likelihood}(\boldsymbol{\theta})$.
- We often rely on the Expectation-Maximization algorithm (EM) Dempster, Laird, and Rubin 1977
- **Step 1:** Initialize $\boldsymbol{\theta}^{(0)} = (\pi_k^{(0)}, \boldsymbol{\mu}_k^{(0)}, \boldsymbol{\Sigma}_k^{(0)})_{k \in [\![K]\!]}$. Then alternate:
- **Step 2 (Expectation):** Given $\boldsymbol{\theta}^{(\text{current})}$ estimate $p(\mathbf{x}_i | \boldsymbol{\theta}^{(\text{current})})$.
- In particular find soft assignments

$$\mathbb{P}(\mathbf{x}_i \in C_k | \boldsymbol{\theta}^{(\text{current})}) = \frac{\pi_k p_{\mathcal{N}}(\mathbf{x}_i | \boldsymbol{\mu}_k^{(\text{current})}, \boldsymbol{\Sigma}_k^{(\text{current})})}{\sum_{j=1}^{K} \pi_j p_{\mathcal{N}}(\mathbf{x}_i | \boldsymbol{\mu}_j^{(\text{current})}, \boldsymbol{\Sigma}_j^{(\text{current})})} .$$

EM algorithm

Maximizing the likelihood

- There is no closed form for $\max_{\boldsymbol{\theta}} \text{Likelihood}(\boldsymbol{\theta})$.
- We often rely on the Expectation-Maximization algorithm (EM) Dempster, Laird, and Rubin 1977
- **Step 1:** Initialize $\boldsymbol{\theta}^{(0)} = (\pi_k^{(0)}, \boldsymbol{\mu}_k^{(0)}, \boldsymbol{\Sigma}_k^{(0)})_{k \in [\![K]\!]}$. Then alternate:
- **Step 2 (Expectation):** Given $\boldsymbol{\theta}^{(\text{current})}$ estimate $p(\mathbf{x}_i | \boldsymbol{\theta}^{(\text{current})})$.
- In particular find soft assignments

$$\mathbb{P}(\mathbf{x}_i \in C_k | \boldsymbol{\theta}^{(\text{current})}) = \frac{\pi_k p_{\mathcal{N}}(\mathbf{x}_i | \boldsymbol{\mu}_k^{(\text{current})}, \boldsymbol{\Sigma}_k^{(\text{current})})}{\sum_{j=1}^{K} \pi_j p_{\mathcal{N}}(\mathbf{x}_i | \boldsymbol{\mu}_j^{(\text{current})}, \boldsymbol{\Sigma}_j^{(\text{current})})} .$$

- **Step 3 (Maximization):** Estimate $\boldsymbol{\theta}^{(\text{next})}$ given $p(\mathbf{x}_i | \boldsymbol{\theta}^{(\text{current})})$ (most likely parameters) : closed form.

EM algorithm

Maximizing the likelihood

- There is no closed form for $\max_{\boldsymbol{\theta}} \text{Likelihood}(\boldsymbol{\theta})$.
- We often rely on the Expectation-Maximization algorithm (EM) Dempster, Laird, and Rubin 1977
- **Step 1:** Initialize $\boldsymbol{\theta}^{(0)} = (\pi_k^{(0)}, \boldsymbol{\mu}_k^{(0)}, \boldsymbol{\Sigma}_k^{(0)})_{k \in [\![K]\!]}$. Then alternate:
- **Step 2 (Expectation):** Given $\boldsymbol{\theta}^{(\text{current})}$ estimate $p(\mathbf{x}_i | \boldsymbol{\theta}^{(\text{current})})$.
- In particular find soft assignments

$$\mathbb{P}(\mathbf{x}_i \in C_k | \boldsymbol{\theta}^{(\text{current})}) = \frac{\pi_k p_{\mathcal{N}}(\mathbf{x}_i | \boldsymbol{\mu}_k^{(\text{current})}, \boldsymbol{\Sigma}_k^{(\text{current})})}{\sum_{j=1}^{K} \pi_j p_{\mathcal{N}}(\mathbf{x}_i | \boldsymbol{\mu}_j^{(\text{current})}, \boldsymbol{\Sigma}_j^{(\text{current})})} .$$

- **Step 3 (Maximization):** Estimate $\boldsymbol{\theta}^{(\text{next})}$ given $p(\mathbf{x}_i | \boldsymbol{\theta}^{(\text{current})})$ (most likely parameters) : closed form.

Remarks

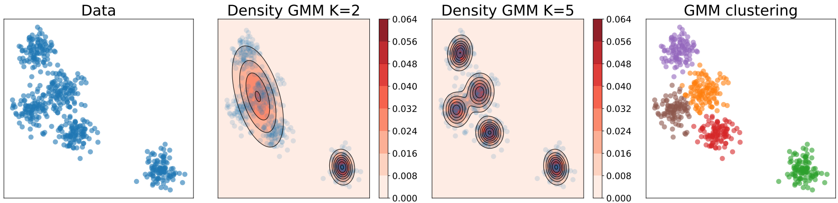
- Alternating strategy similar to Lloyd's algorithm !
- E step: assign point to cluster, M step: find clusters parameters.

Examples in 2D

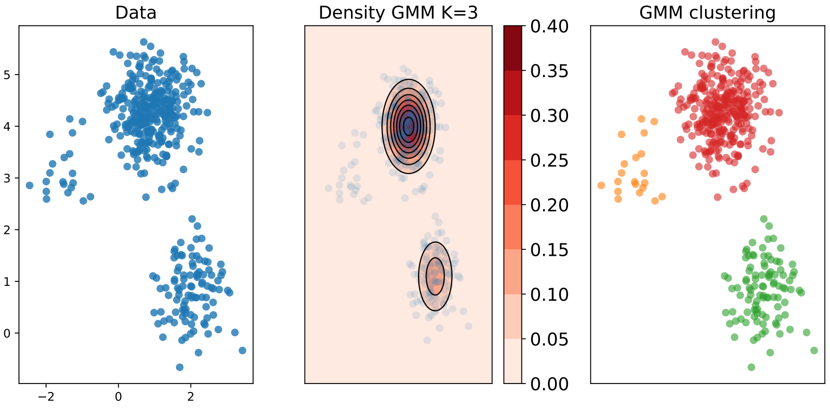
It is also a clustering algorithm !

- GMM can assign points to clusters.
- Given a point $\mathbf{x}_i$ find its most likely cluster via

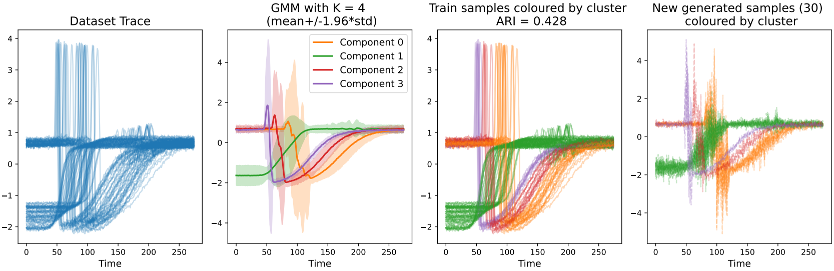
$$\arg\max_{k \in [\![K]\!]} \mathbb{P}(\mathbf{x}_i \in C_k | \boldsymbol{\theta}) = \frac{\pi_k p_{\mathcal{N}}(\mathbf{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j p_{\mathcal{N}}(\mathbf{x}_i | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} .$$



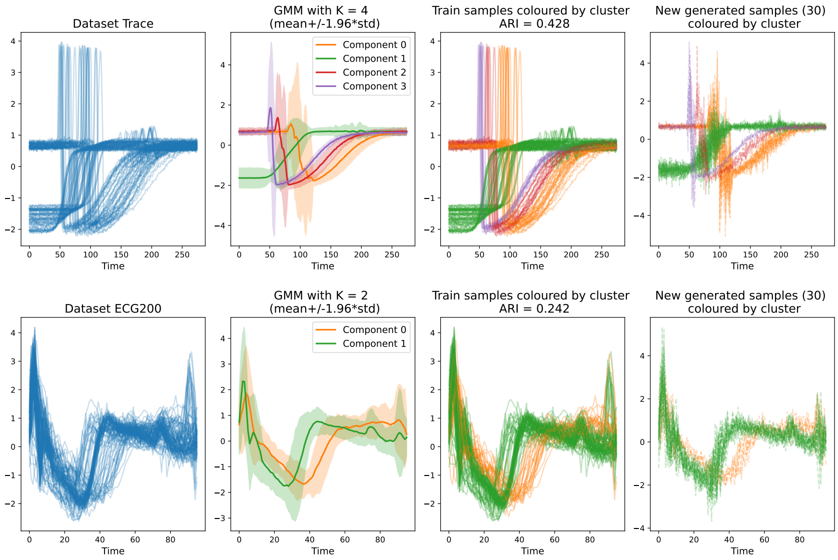
Examples in 2D



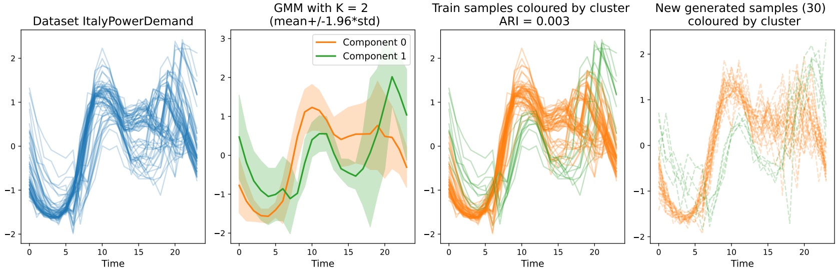
GMM modeling of time series



GMM modeling of time series



GMM modeling of time series



GMM modeling of time series

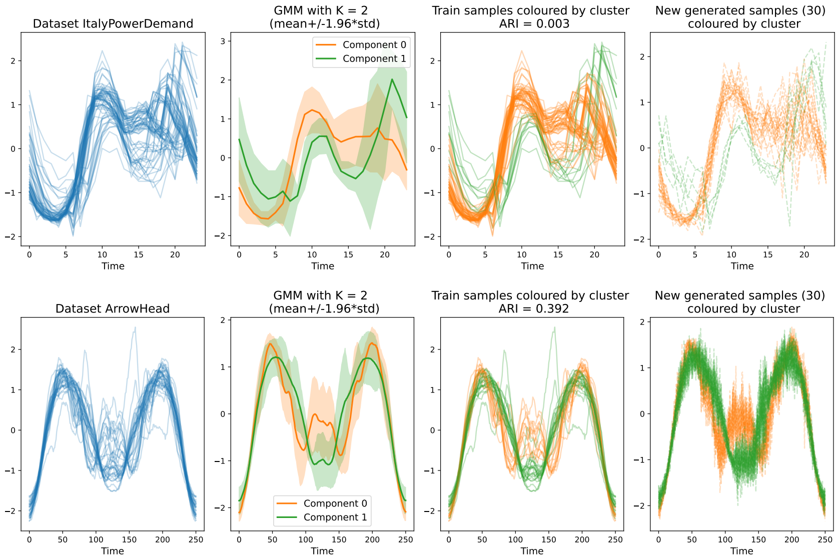


Table of contents

Kernel Density Estimation (KDE)

Non parametric density estimation

- Find $\hat{p}(\mathbf{x}) \approx p(\mathbf{x})$ without having to estimate parameters $\boldsymbol{\theta}$

"Kernel" function

- $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ a pointwise non-negative function
- κ measures **similarity between points**

Kernel Density Estimation (KDE)

Non parametric density estimation

- Find $\hat{p}(\mathbf{x}) \approx p(\mathbf{x})$ without having to estimate parameters $\boldsymbol{\theta}$

"Kernel" function

- $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ a pointwise non-negative function
- κ measures **similarity between points**
- In practice when $\mathcal{X} \subseteq \mathbb{R}^d$, $k(\mathbf{x}, \mathbf{x}') = q_h(\mathbf{x} - \mathbf{x}') = h^{-d} q(\frac{1}{h}(\mathbf{x} - \mathbf{x}'))$
- $q : \mathbb{R}^d \to \mathbb{R}_+$ that has large values around 0, $h > 0$ the **bandwidth**

Kernel Density Estimation (KDE)

Non parametric density estimation

- Find $\hat{p}(\mathbf{x}) \approx p(\mathbf{x})$ without having to estimate parameters $\boldsymbol{\theta}$

"Kernel" function

- $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ a pointwise non-negative function
- κ measures **similarity between points**
- In practice when $\mathcal{X} \subseteq \mathbb{R}^d$, $k(\mathbf{x}, \mathbf{x}') = q_h(\mathbf{x} - \mathbf{x}') = h^{-d} q(\frac{1}{h}(\mathbf{x} - \mathbf{x}'))$
- $q : \mathbb{R}^d \to \mathbb{R}_+$ that has large values around 0, $h > 0$ the **bandwidth**
- e.g. box kernel $q(\mathbf{x}) = \mathbf{1}_{\|\mathbf{x}\|_2 \leq 1}$, gaussian $q(\mathbf{x}) = \exp(-\|\mathbf{x}\|_2/2)$
- ⚠ same name but not the same as kernels in SVM !

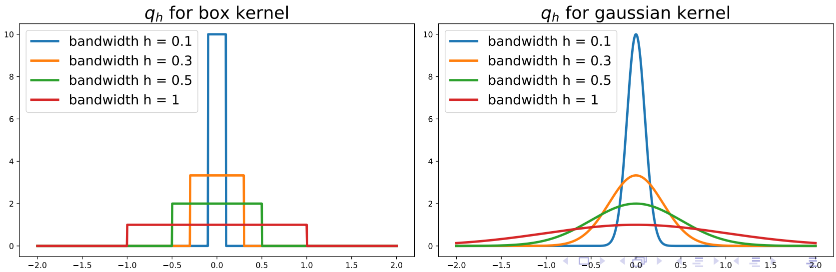
Kernel Density Estimation (KDE)

Non parametric density estimation

- Find $\hat{p}(\mathbf{x}) \approx p(\mathbf{x})$ without having to estimate parameters $\boldsymbol{\theta}$

"Kernel" function

- $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ a pointwise non-negative function
- κ measures **similarity between points**
- In practice when $\mathcal{X} \subseteq \mathbb{R}^d$, $k(\mathbf{x}, \mathbf{x}') = q_h(\mathbf{x} - \mathbf{x}') = h^{-d} q(\frac{1}{h}(\mathbf{x} - \mathbf{x}'))$
- $q : \mathbb{R}^d \to \mathbb{R}_+$ that has large values around 0, $h > 0$ the **bandwidth**
- e.g. box kernel $q(\mathbf{x}) = \mathbf{1}_{\|\mathbf{x}\|_2 \leq 1}$, gaussian $q(\mathbf{x}) = \exp(-\|\mathbf{x}\|_2/2)$



Kernel Density Estimation (KDE)

KDE estimation Rosenblatt 1956; Parzen 1962

- The approximate distribution is:

$$\hat{p}(\mathbf{x}) \propto \frac{1}{n}\sum_{i=1}^{n} \kappa(\mathbf{x}, \mathbf{x}_i) = \frac{1}{n}\sum_{i=1}^{n} q_h(\mathbf{x} - \mathbf{x}_i)$$

- $\kappa(\mathbf{x}, \mathbf{x}_i)$ is close to 1 when $\mathbf{x}$ is similar to $\mathbf{x}_i$

Kernel Density Estimation (KDE)

KDE estimation Rosenblatt 1956; Parzen 1962

- The approximate distribution is:

$$\hat{p}(\mathbf{x}) \propto \frac{1}{n}\sum_{i=1}^{n} \kappa(\mathbf{x}, \mathbf{x}_i) = \frac{1}{n}\sum_{i=1}^{n} q_h(\mathbf{x} - \mathbf{x}_i)$$

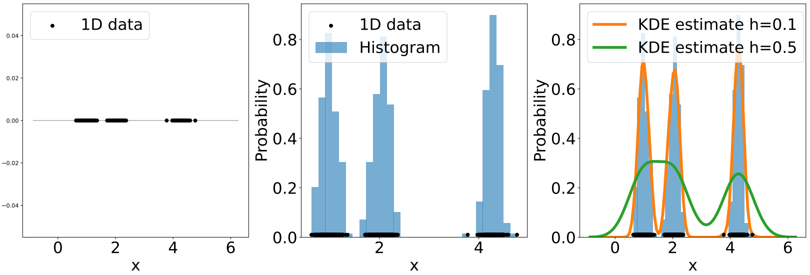
- $\kappa(\mathbf{x}, \mathbf{x}_i)$ is close to 1 when $\mathbf{x}$ is similar to $\mathbf{x}_i$
- Normalizing factor so that $\int \hat{p}(\mathbf{x})\mathrm{d}\mathbf{x} = 1$, requires

$$\int \kappa(\mathbf{x}, \mathbf{x}_i)\mathrm{d}\mathbf{x} = \int q(\mathbf{x})\mathrm{d}\mathbf{x} = 1 \text{ for common kernels.}$$

- Complexity of calculating $\hat{p}(\mathbf{x})$ usually in $\mathcal{O}(nd)$

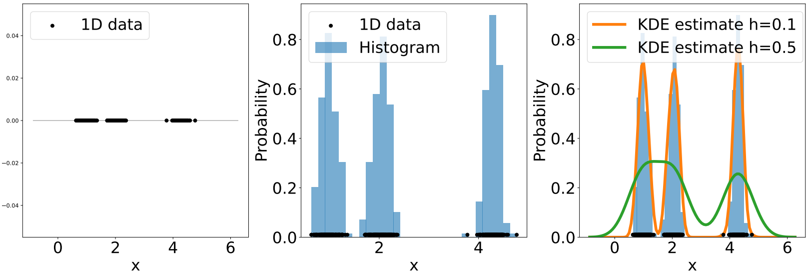
Kernel Density Estimation (KDE)

Example in 1D



Kernel Density Estimation (KDE)

Example in 1D



Example in 2D

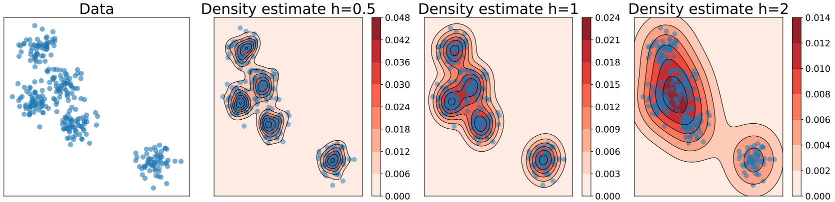
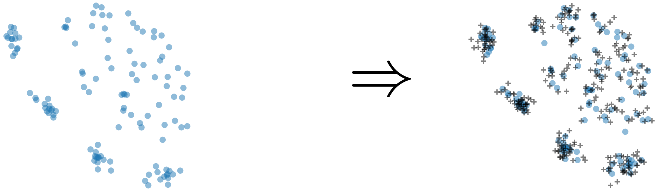


Table of contents

Generative modeling



Objective

$$\{\mathbf{x}_i\}_{i=1}^n \quad \Rightarrow \quad g \text{ such that } p(\mathbf{x}) \approx g(\mathbf{z}) \text{ with } \mathbf{z} \sim \mathcal{N}$$

- Estimate a mapping function $g(\mathbf{z})$ that generates similar samples to $\{\mathbf{x}_i\}_{i=1}^n$.
- Latent variable $\mathbf{z}$ follows a known Normal or Unif distribution.
- Optional : recover the distribution (change of variable formula).

Parameters

- Type of distribution for $\mathbf{z}$.
- Type of function for g (NN)

Methods

- Generative neural networks.
- GMM.

Divergence

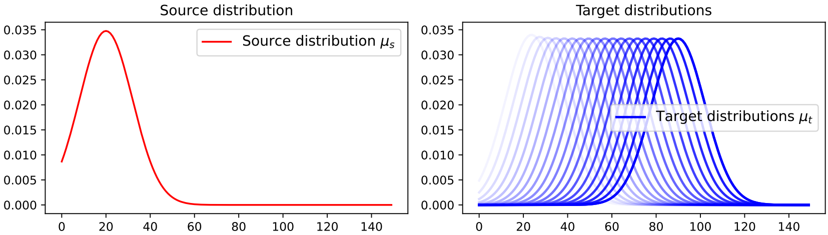
$D : \mathcal{P}(\Omega) \times \mathcal{P}(\Omega) \to \mathbb{R}$ is a **divergence** if it satisfies :

- for any distributions μ_s and μ_t, $D(\mu_s || \mu_t) \geq 0$.
- $D(\mu_s || \mu_t) = 0 \iff \mu_s = \mu_t$.

Divergence

$D : \mathcal{P}(\Omega) \times \mathcal{P}(\Omega) \rightarrow \mathbb{R}$ is a **divergence** if it satisfies :

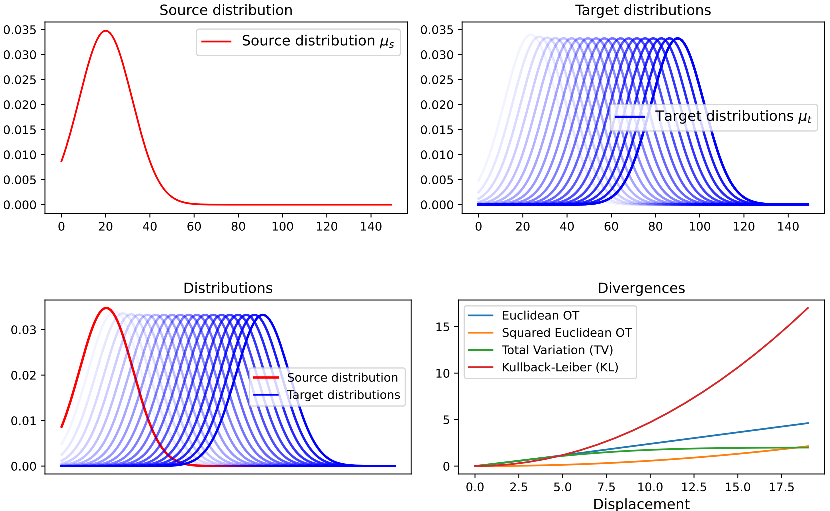
- for any distributions μ_s and μ_t, $D(\mu_s||\mu_t) \geq 0$.
- $D(\mu_s||\mu_t) = 0 \iff \mu_s = \mu_t$.



Divergence

$D : \mathcal{P}(\Omega) \times \mathcal{P}(\Omega) \to \mathbb{R}$ is a **divergence** if it satisfies :

- for any distributions $\color{red}{\mu_s}$ and $\color{blue}{\mu_t}$, $D(\color{red}{\mu_s}||\color{blue}{\mu_t}) \geq 0$.
- $D(\color{red}{\mu_s}||\color{blue}{\mu_t}) = 0 \iff \color{red}{\mu_s} = \color{blue}{\mu_t}$.



Kullback-Leiber divergence

The definition

If μ_s is absolutely continuous with respect to μ_t then

$$\mathsf{KL}(\mu_s||\mu_t) = \int_\Omega \log(\frac{\mathrm{d}\mu_s}{\mathrm{d}\mu_t}(\mathbf{x}))\mathrm{d}\mu_s(\mathbf{x})\,.$$

where $\frac{\mathrm{d}\mu_s}{\mathrm{d}\mu_t}$ is the Radon–Nikodym derivative.

Kullback-Leiber divergence

The definition

If μ_s is absolutely continuous with respect to μ_t then

$$\mathrm{KL}(\mu_s||\mu_t) = \int_\Omega \log(\frac{\mathrm{d}\mu_s}{\mathrm{d}\mu_t}(\mathbf{x}))\mathrm{d}\mu_s(\mathbf{x})\,.$$

where $\frac{\mathrm{d}\mu_s}{\mathrm{d}\mu_t}$ is the Radon–Nikodym derivative.

Examples

- If distributions have densities with respect to Lesbegue $\mu_s = f\mathrm{d}\mathbf{x}, \mu_t = g\mathrm{d}\mathbf{x}$

$$\mathrm{KL}(\mu_s||\mu_t) = \int_\Omega \log(\frac{f(\mathbf{x})}{g(\mathbf{x})})f(\mathbf{x})\mathrm{d}\mathbf{x}\,.$$

Kullback-Leiber divergence

The definition

If μ_s is absolutely continuous with respect to μ_t then

$$\mathrm{KL}(\mu_s||\mu_t) = \int_\Omega \log(\frac{\mathrm{d}\mu_s}{\mathrm{d}\mu_t}(\mathbf{x}))\mathrm{d}\mu_s(\mathbf{x})\,.$$

where $\frac{\mathrm{d}\mu_s}{\mathrm{d}\mu_t}$ is the Radon–Nikodym derivative.

Examples

- If distributions have densities with respect to Lesbegue $\mu_s = f\mathrm{d}\mathbf{x}, \mu_t = g\mathrm{d}\mathbf{x}$

$$\mathrm{KL}(\mu_s||\mu_t) = \int_\Omega \log(\frac{f(\mathbf{x})}{g(\mathbf{x})})f(\mathbf{x})\mathrm{d}\mathbf{x}\,.$$

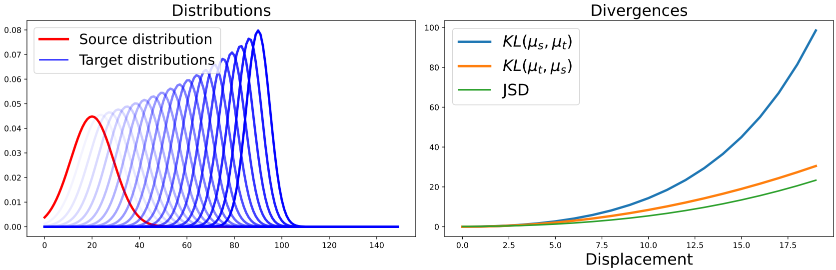
- If the distributions are discrete **with the same support** $(\mathbf{x}_1, \cdots, \mathbf{x}_n)$:

$$\mathrm{KL}(\mu_s||\mu_t) = \sum_{i=1}^{n} \log(\frac{\mu_s(\mathbf{x}_i)}{\mu_t(\mathbf{x}_i)})\mu_s(\mathbf{x}_i)\,.$$

Relation with maximum likelihood estimation

(On the board)

Kullback Leiber divergence is asymmetric

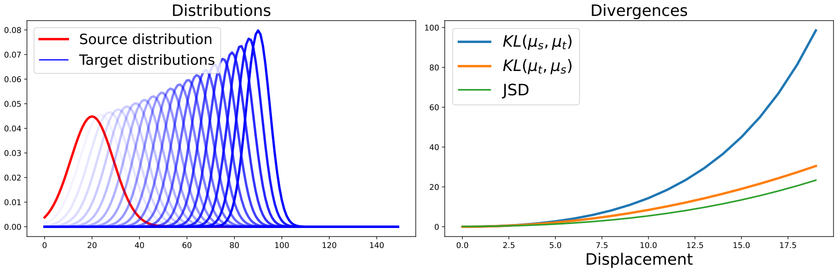


Jensen-Shannon divergence

We can derive a "symmetric" version of KL:

$$\mathrm{JSD}(\mu_s, \mu_t) = \frac{1}{2}\,\mathrm{KL}(\mu_s||\overline{\mu}) + \frac{1}{2}\,\mathrm{KL}(\mu_t||\overline{\mu}) \text{ with } \overline{\mu} = \frac{1}{2}(\mu_s + \mu_t)\,.$$

Kullback Leiber divergence is asymmetric



Jensen-Shannon divergence

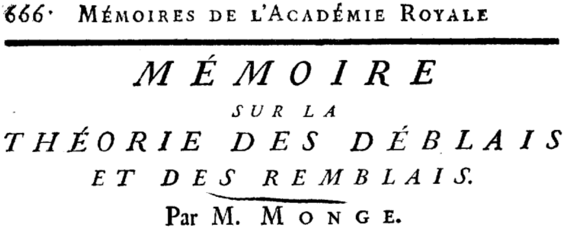
We can derive a "symmetric" version of KL:

$$\mathrm{JSD}(\mu_s, \mu_t) = \frac{1}{2}\,\mathrm{KL}(\mu_s||\overline{\mu}) + \frac{1}{2}\,\mathrm{KL}(\mu_t||\overline{\mu}) \text{ with } \overline{\mu} = \frac{1}{2}(\mu_s + \mu_t)\,.$$

Drawbacks

- $\mathrm{KL}(\mu_s||\mu_t)$ undefined when support of distributions are different.
- Distance between the points in the support not used.
- It is **not** a distance.

The origins of optimal transport


666· MÉMOIRES DE L'ACADÉMIE ROYALE

MÉMOIRE

SUR LA

THÉORIE DES DÉBLAIS

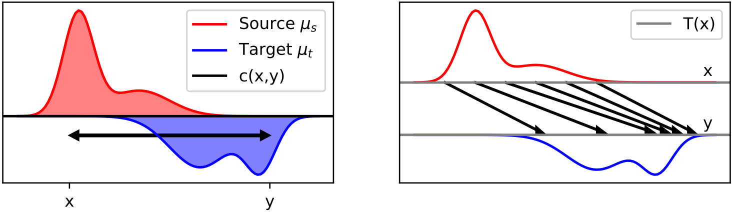
ET DES REMBLAIS.

Par M. MONGE.

Problem

- How to move dirt from one place (déblais) to another (remblais) while minimizing the effort ?
- Find a mapping T between the two distributions of mass (transport).
- Optimize with respect to a displacement cost $c(x, y)$ (optimal).

The origins of optimal transport

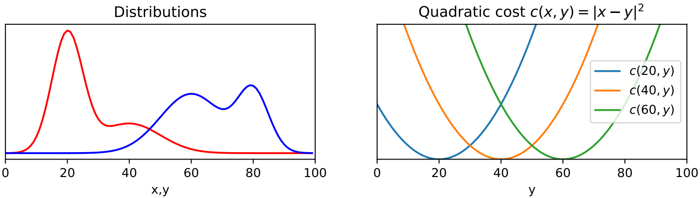


Problem

- How to move dirt from one place (déblais) to another (remblais) while minimizing the effort ?
- Find a mapping T between the two distributions of mass (transport).
- Optimize with respect to a displacement cost $c(x, y)$ (optimal).

Optimal transport (Monge formulation)

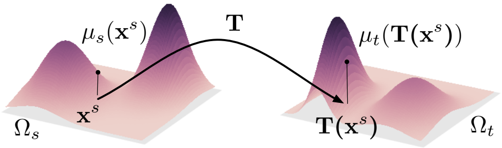
- Mathematical tools aiming at comparing distributions



- Probability measures μ_s and μ_t on Ω with a cost function $d : \Omega \times \Omega \to \mathbb{R}^+$.
- The Monge formulation aim at finding a mapping $T : \Omega_s \to \Omega_t$

$$\inf_{T\#\mu_s=\mu_t} \int_\Omega d(\mathbf{x}, T(\mathbf{x}))\mathrm{d}\mu_s(\mathbf{x}) \tag{2}$$

What is $T\#\mu_s = \mu_t$?



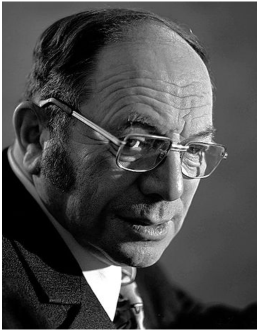
- $T\#$ is the so called push forward operator
- If $\mathbf{x} \sim \mu_s$ then $T(\mathbf{x}) \sim T\#\mu_s$.
- Condition $T\#\mu_s = \mu_t$ is equivalent to:

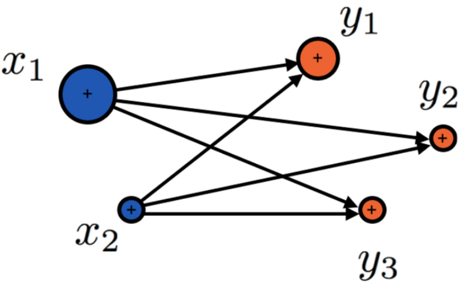
$$\mu_t(A) = \mu_s(T^{-1}(A))$$

- For $\mu_s = \sum_{i=1}^n a_i \delta_{x_i}$,

$$T\#\mu_s = \sum_{i=1}^n a_i \delta_{T(x_i)} \,.$$

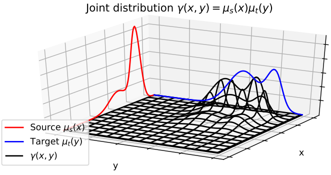
Kantorovich relaxation

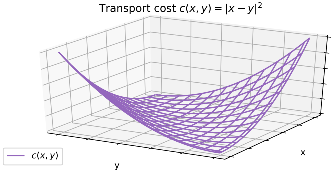




- Leonid Kantorovich (1912–1986), Economy nobelist in 1975, proposed a different formulation of the problem
- With applications mainly for resource allocation problems

Kantorovich relaxation





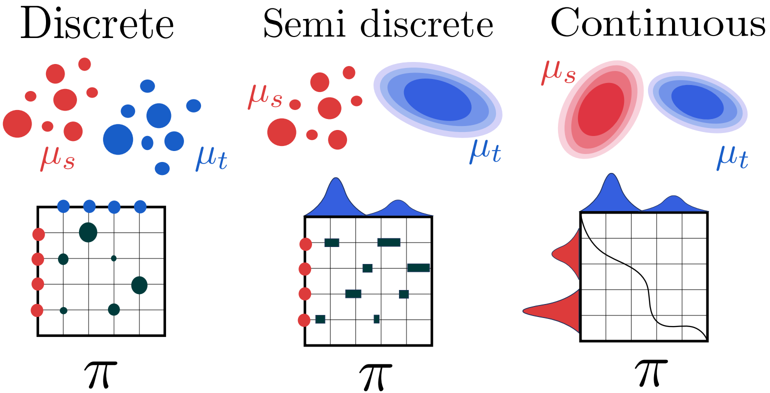
$\mu_s = \sum_{i=1}^{n} a_i \delta_{x_i}$ and $\mu_t = \sum_{j=1}^{m} b_j \delta_{y_j}$ on a common ground space equipped with a distance

- The Kantorovich formulation seeks for a probabilistic coupling $\pi \in \Pi(\mu_s \times \mu_t)$ between μ_s and μ_t.
- π is a joint probability measure with prescribed marginals μ_s and μ_t.
- Computes the Wasserstein distance :

$$\mathcal{W}_p(\mu_s, \mu_t) = \left(\min_{\pi \in \Pi(\mu_s, \mu_t)} \sum_{i,j} d(x_i, y_j)^q \pi_{i,j} \right)^{\frac{1}{p}} \quad (3)$$

Probabilistic couplings

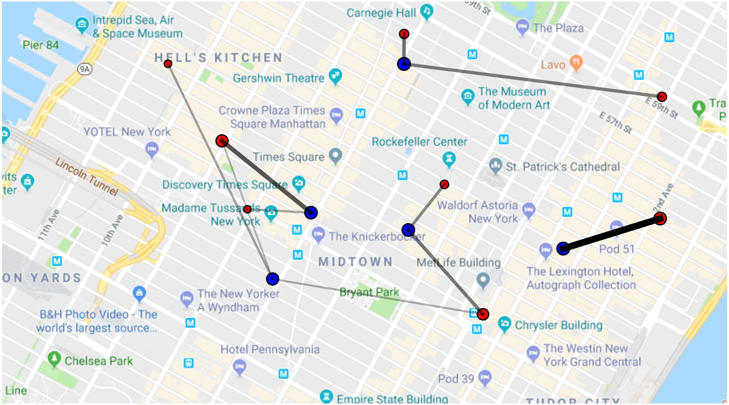
The resulting coupling π associates in a "fuzzy" way the points of the distributions.



Properties of Wasserstein distance

(On the board)

Illustration with bakeries and cafés



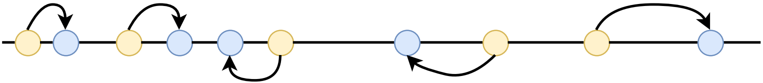
Special case: 1D distribution

We consider the case where $d(x, y) = |x - y|$

- if $x_1 < x_2$ and $y_1 < y_2$ then

$$d(x_1, y_1) + d(x_2, y_2) < d(x_1, y_2) + d(x_2, y_1)$$

- Any optimal transport plan respects the ordering of the elements
- The solution is given by the monotone rearrangement of μ_s onto μ_t.
- Very simple algorithm to compute the transport in $O(N \log N)$, by sorting both x_i and y_i



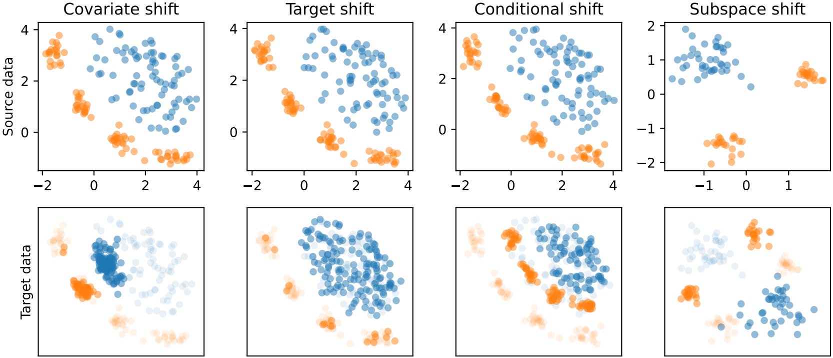
In the case $\Omega = \mathbb{R}^d$, $d(x, y) = \|x - y\|$ and $p = 1$

The optimal transport problem then aim to find $f \in \text{Lip}^1$ (set of 1-Lipschitz functions) as

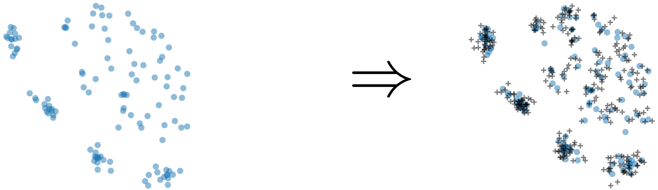
$$\sup_{f \in \text{Lip}^1} \int f d(\mu_s - \mu_t) = \sup_{f \in \text{Lip}^1} \mathbb{E}_{\mathbf{x} \sim \mu_s}[f(x)] - \mathbb{E}_{\mathbf{y} \sim \mu_t}[f(y)] \tag{4}$$

- Known as **Kantorovich-Rubinstein duality**

Optimal transport for domain adaptation



Generative modeling



Objective

$$\{\mathbf{x}_i\}_{i=1}^n \quad \Rightarrow \quad g \text{ such that } p(\mathbf{x}) \approx g(\mathbf{z}) \text{ with } \mathbf{z} \sim \mathcal{N}$$

- Estimate a mapping function $g(\mathbf{z})$ that generates similar samples to $\{\mathbf{x}_i\}_{i=1}^n$.
- Latent variable $\mathbf{z}$ follows a known Normal or Unif distribution.
- Optional : recover the distribution (change of variable formula).

Parameters

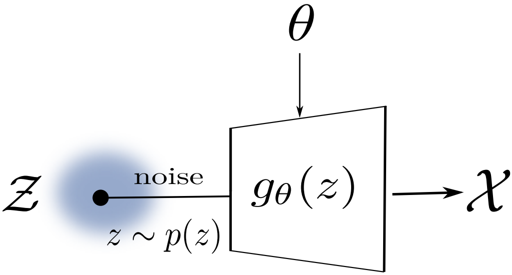
- Type of distribution for $\mathbf{z}$.
- Type of function for g (NN)

Methods

- Generative neural networks.
- KDE, GMM.

Generative modeling

- Latent space $\mathcal{Z}$ which we can sample using known $p(\mathbf{z})$.
- Use parametric functions $g_\theta : \mathcal{Z} \to \mathbb{R}^d$.
- Goal: optimize θ such that when we sample $\mathbf{z}$ from $p(\mathbf{z})$ the output $g_\theta(\mathbf{z})$ looks like being generated by $p(\mathbf{x})$.



Generative modeling by divergence minimization

Generator function

- $g_{\boldsymbol{\theta}} : \mathbb{R}^p \to \mathbb{R}^d$ is a function (neural network) and $p(\mathbf{z}) \in \mathcal{P}(\mathbb{R}^p)$.
- Notation: $g_{\boldsymbol{\theta}} \# p(\mathbf{z})$ is the distribution of the random variable $g_{\boldsymbol{\theta}}(\mathbf{z})$ with $\mathbf{z} \sim p(\mathbf{z})$.

Generative modeling by divergence minimization

Generator function

- $g_{\boldsymbol{\theta}} : \mathbb{R}^p \to \mathbb{R}^d$ is a function (neural network) and $p(\mathbf{z}) \in \mathcal{P}(\mathbb{R}^p)$.
- Notation: $g_{\boldsymbol{\theta}} \# p(\mathbf{z})$ is the distribution of the random variable $g_{\boldsymbol{\theta}}(\mathbf{z})$ with $\mathbf{z} \sim p(\mathbf{z})$.

Minimizing the divergence between distributions

- Find the parameters $\boldsymbol{\theta}$ that optimize

$$\min_{\boldsymbol{\theta}} D(p_{\text{data}}, g_{\boldsymbol{\theta}} \# p(\mathbf{z}))$$

- Learn a generator $g_{\boldsymbol{\theta}}$ that minimize the divergence D between the generated data and the empirical data distribution $p_{\text{data}} = \frac{1}{n} \sum_{i=1}^{n} \delta_{\mathbf{x}_i}$.

Generative modeling by divergence minimization

Generator function

- $g_{\boldsymbol{\theta}} : \mathbb{R}^p \to \mathbb{R}^d$ is a function (neural network) and $p(\mathbf{z}) \in \mathcal{P}(\mathbb{R}^p)$.
- Notation: $g_{\boldsymbol{\theta}} \# p(\mathbf{z})$ is the distribution of the random variable $g_{\boldsymbol{\theta}}(\mathbf{z})$ with $\mathbf{z} \sim p(\mathbf{z})$.

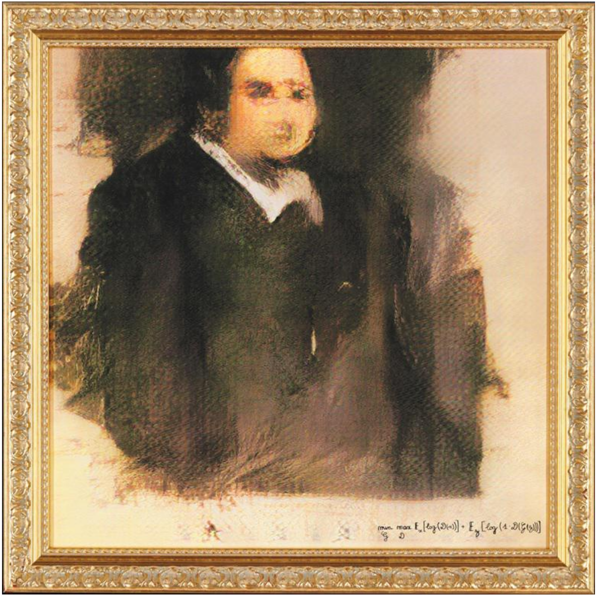
Minimizing the divergence between distributions

- Find the parameters $\boldsymbol{\theta}$ that optimize

$$\min_{\boldsymbol{\theta}} D(p_{\text{data}}, g_{\boldsymbol{\theta}} \# p(\mathbf{z}))$$

- Learn a generator $g_{\boldsymbol{\theta}}$ that minimize the divergence D between the generated data and the empirical data distribution $p_{\text{data}} = \frac{1}{n} \sum_{i=1}^{n} \delta_{\mathbf{x}_i}$.
- Different divergences can be used:
  - Jensen-Shannon (JS): classical GAN Goodfellow et al. 2014.
  - Wasserstein (Optimal Transport) Arjovsky, Chintala, and Bottou 2017.

Examples



Examples

Style GAN: https://arxiv.org/pdf/1812.04948.pdf
https://www.whichfaceisreal.com/index.php
https://www.instagram.com/openaidalle/

Diffusion models

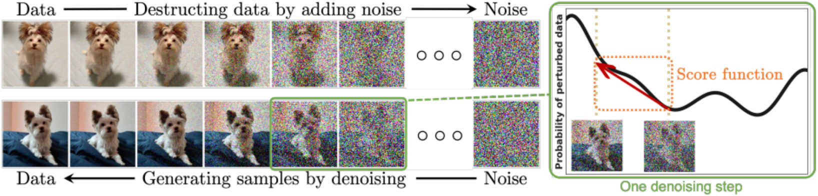


Figure: From Yang et al. 2024

$$
\begin{aligned}
\text{Forward: } & q(\mathbf{x}_t|\mathbf{x}_0) = p_{\mathcal{N}}(\mathbf{x}_t|\sqrt{\overline{\alpha}_t}\mathbf{x}_0, (1-\overline{\alpha}_t)\mathbf{I}) \\
\text{Backward: } & p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t) = p_{\mathcal{N}}(\mathbf{x}_t|\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t), \boldsymbol{\Sigma}_{\boldsymbol{\theta}}(\mathbf{x}_t, t))
\end{aligned} \tag{5}
$$

References I

Arjovsky, Martin, Soumith Chintala, and Léon Bottou (2017). *Wasserstein GAN*.

Dempster, Arthur P, Nan M Laird, and Donald B Rubin (1977). "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the royal statistical society: series B (methodological)* 39.1, pp. 1–22.

Goodfellow, Ian et al. (2014). "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems*. Vol. 27.

Parzen, Emanuel (1962). "On estimation of a probability density function and mode". In: *The annals of mathematical statistics* 33.3, pp. 1065–1076.

Rosenblatt, Murray (1956). "Remarks on some nonparametric estimates of a density function". In: *The annals of mathematical statistics*, pp. 832–837.

Yang, Ling et al. (2024). *Diffusion Models: A Comprehensive Survey of Methods and Applications*. arXiv: 2209.00796 [cs.LG].