Deep conditional measure quantization

Gabriel Turinici

CEREMADE
Université Paris Dauphine - PSL
Paris, France

OL2A 2023
International Conference on Optimization, Learning Algorithms and Applications

OL2A 2023, Ponta Delgada (PT), Sept. 27 – 29, 2023

Executive summary

Deep generative AI algorithms such as GAN /VAE / Transformer, ... create objects similar to ones in the dataset it was trained on.

In particular conditional generation is encountered when the created objects have to fit some prescribed description : a prompt for chatGPT, some parts of the image when restoring images etc.

Exploiting measure quantization techniques, and Huber-energy kernel based statistical distances we give a procedure to create conditional samples.

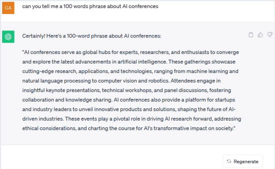
The procedure is tested with satisfactory results on a standard AI dataset (MNIST).

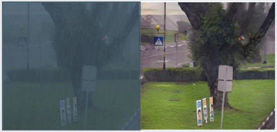
Outline

Introduction and motivation

- we are concerned with GENERATIVE IA algorithms i.e. that create new objects (e.g., images) based on some dataset of examples ; more precisely, we focus on conditional generation which produces new samples based on an initial information. Number of samples Q is given.

Conditional generative AI examples



Left : "chatGPT" writes a text based on a prompt, i.e. conditional on the prompt (ref: Openai Chatgpt) **Middle** generation of images based on a description (Midjourney, DALL-E). (ref : wikipedia Midjourney) **Right** heavy rain image restoration, cf. [1].

Outline

• Given : empirical dataset $\mu_e = \frac{1}{M}\sum_{\ell=1}^{M} \delta_{z_\ell}$ sampled from unknown distribution μ ($z_\ell \sim \mu$), $z_\ell \in \mathbb{R}^N$.
• prescribed information (the condition) denoted "x" or X (as r.v. on $\mathcal{X}$), y= unknown
• Goal: construct Q samples from $\mu_x := \mu(dy|x)$
• Number of samples Q is given: "quantization"

Mathematical formulation

Find $Y = (Y_q)_{q=1}^{Q} \in \mathcal{Y}^Q$ ($q = 1, ..., Q$) that minimizes the distance from $\delta_Y := \frac{\sum_{q=1}^{Q} \delta_{Y_q}}{Q}$ to the target measure μ_x.

Questions:
• how to compute $Y \mapsto dist(\delta_Y, \eta)^2$ objects are probability distributions !
• does a minimum exists as random variable, or, in mathematical terms, a measurable selection exists ?
• how to minimize the distance ?

Technical how to

Question: how to compute $Y \mapsto dist(\delta_Y, \eta)^2$

Distance: use a conditionally negative definite kernel h :

$$d(\mu_1, \mu_2)^2 = -\frac{1}{2} \int_{\mathbb{R}^L} \int_{\mathbb{R}^L} h(x-y)(\mu_1 - \mu_2)(dx)(\mu_1 - \mu_2)(dy). \quad (1)$$

Discrete version :

$$d\left(\frac{1}{J} \sum_{j=1}^{J} \delta_{X_j}, \frac{1}{B} \sum_{b=1}^{B} \delta_{z_b} \right)^2 = \frac{\sum_{j,b=1}^{J,B} h(X_j - z_b)}{JB} - \frac{\sum_{j,j'=1}^{J} h(X_j - X_{j'})}{2J^2} - \frac{\sum_{b,b'=1}^{B} h(z_b - z_{b'})}{2B^2}. \quad (2)$$

Question: what function h to choose ?

Statistical distances: conditionally negative kernels

Definition (conditional negative definite)

A kernel $h(\cdot,\cdot)$ is said to be conditionally negative definite if for any $I \in \mathbb{N}$, $p_1, ..., p_I$ with $\sum p_i = 0$ and any $x_1, ..., x_I$: $\sum_{i,j} p_i p_j h(x_i, x_j) \le 0$.

Theorem ("Gini difference" Gini 1912; "energy distance" Szekelly 1985, '02; "maximum mean discrepancy" Gretton '07, G.T. '21 [6])

The kernel $h(x) = |x|$ is conditionally negative definite.

Caution : (sub) gradient $\nabla|x| = x/|x|^2$ unstable in $x = 0$.

Theorem (Schoenberg 1938 [3], Micchelli 1984 [2], GT 2021 [5])

For any $a \ge 0$, $\alpha \in]0,1[$, $(a^2 + |x|^2)^\alpha - a^{2\alpha}$ and $\|x^2\|/(a^2 + |x|^2)^\alpha$ are conditionally negative definite (explicit Gaussian mixtures).

In particular this is true for our choice $h(x) = \sqrt{a^2 + \|x\|^2}^\alpha - a^\alpha$.

Theoretical results on the existence of conditional generation

The conditional quantization of the law μ is defined as follows: for any $x \in \mathcal{X}$ we look for the minimizer of the distance to μ_x i.e., any $\mathbf{y}^{opt}(x) \in \mathcal{Y}^Q$ such that :

$$d(\delta_{\mathbf{y}^{opt}(x)}, \mu_x)^2 \leq d(\delta_{\mathbf{y}}, \mu_x)^2, \quad \forall \mathbf{y} \in \mathcal{Y}^Q. \tag{3}$$

PROBLEM: in general the minimum is not unique so $\mathbf{y}^{opt}(x)$ is a set-valued function !

Proposition (GT [4])

Suppose that the distribution μ is such that for any $x \in \mathcal{X}$ the distribution μ_x has finite α-order moment. Then there exists a measurable function $\mathbf{y}^{opt} : \mathcal{X} \to \mathcal{Y}^Q$ such that $\mathbf{y}^{opt}(x)$ satisfies equation (3) for any $x \in \mathcal{X}$.

This ensures $\mathbf{y}^{opt}(x)$ is a proper random variable !

Theoretical results on the existence of conditional generation

What about other properties, e.g. continuity ?

- in general this is NOT true (counter examples exist)!
- no general results but a particular case.

Proposition (GT [4])

Let us take $\mathcal{Y} = \mathbb{R}$, $a = 0$, $r = 1$ (i.e. the kernel is the so-called 'energy' kernel). We work under the assumptions of the previous proposition and suppose in addition that the distribution $\mu(dx, dy)$ is absolutely continuous with respect to the Lebesgue measure and admits a continuous density $\rho(x, y)$ which is strictly positive on $\mathcal{X} \times \mathcal{Y}$. Then the conditional quantization $\mathbf{y}^{opt}(x)$ is unique for any $x \in \mathcal{X}$ and continuous as a function of x.

Outline

Stochastic minimization algorithm

deep neural networks are used to construct complex function that maps condition to the quantization

neural network input : the condition x ; output : the Q points $Y \in \mathcal{Y}^Q$; network parameters weights θ

network mapping $x \mapsto f(\theta, x) \in \mathcal{Y}^Q$

θ are optimized to minimize

$$\mathbb{E}_{x \sim \mu^X} \left[d\left(\mu_x, \frac{1}{Q} \sum_{q=1}^{Q} \delta_{f(\theta, x_q)} \right)^2 \right]. \tag{4}$$

Stochastic minimization algorithm

Algorithm DCQM samples from the conditional distribution.

Algorithm A1 Deep Conditional Measure Quantization algorithm : DCMQ

procedure DCMQ
- set batch size B, sampling size J, parameters a (default 10^{-6}), and r (default 1.), minimization algorithm (default = Adam) ; max iterations (default 1000)
- choose a network architecture and initialize layers (default : 5 sequential fully connected layers of size $n_y \times Q$, first input is of size n_x);

while (max iteration not reached) **do**
- sample i.i.d $x_1, ..., x_B$ according to the marginal law μ^X of X;
- for each $b \leq B$ sample i.i.d J times from μ_{x_b} and denote $\tilde{\mathbf{y}}_b$ the sample as a vector in $\mathcal{Y}^J$;
- propagate $x_1, ..., x_B$ through the network to obtain $\mathbf{y}^{dcmq}(x_b) \in \mathcal{Y}^Q$, $b \leq B$
- compute the loss $\mathcal{L} = \frac{1}{B} \sum_{b=1}^{B} d\left(\delta_{\tilde{\mathbf{y}}_b}, \delta_{\mathbf{y}^{dcmq}(x_b)}\right)^2$;
- update the network as specified by the stochastic optimization algorithm (using backpropagation) to minimize the loss $\mathcal{L}$.

end while

end procedure

Stochastic minimization algorithm

Algorithm DCQM-J samples directly from the joint distribution.

Algorithm A2 Deep Conditional Measure Quantization algorithm through joint sampling : DCMQ-J

procedure DCMQ-J
- set batch size B, sampling size J, parameters a (default 10^{-6}), and r (default 1.), minimization algorithm (default = Adam); max iterations (default 1000)
- choose a network architecture and initialize layers (default : cf. fig. 4);

while (max iteration not reached) **do**

sample i.i.d $(x_1, y_1), ..., (x_B, y_B)$ from the dataset ;

propagate $x_1, ..., x_B$ through the network to obtain $\mathbf{y}^{dcmq}(x_b) \in \mathcal{Y}^Q$, $b \le B$;

compute the loss $\mathcal{L} = \frac{1}{B} \sum_{b=1}^{B} d \left(\delta_{x_b, y_b}, \frac{\sum_{q=1}^{Q} \delta_{x_b, \mathbf{y}_q^{dcmq}(x_b)}}{Q} \right)^2$;

update the network as specified by the stochastic optimization algorithm (using backpropagation) to minimize the loss $\mathcal{L}$.

end while

end procedure

Outline

Gaussian tests, algo DCMQ, conditional sampling

The first test : 2D Gaussian that has its mean given by another variable: X, Y 2 independent standard Gaussian variables, $\mu =$ distribution of $X + Y$, For $X = x$, μ_x is a Gaussian variable with mean x.

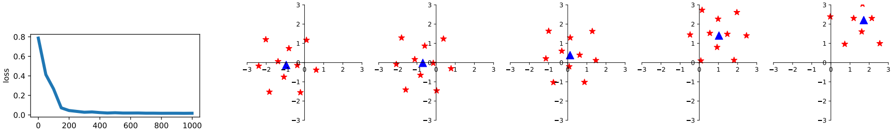


Figure: Conditional quantization with $Q = 10$ points for the test in section 1. **Left image:** Convergence of the loss function. **Right images :** Five points $x_1, \ldots, x_5 \in \mathcal{X} = \mathbb{R}^2$ are sampled from μ^X (plotted as blue triangles); the DNN (after training) is asked to quantize the conditional distribution μ_{x_b} for each $b \leq 5$ (red stars). Recall that μ_{x_b} is a Gaussian shifted by x_b. The quantization points follow precisely the indicated mean.

MNIST dataset

Random images from the handwritten figures MNIST dataset.

Numerical results : MNIST

Test on the MNIST database; the generation is conditional to the information available. White and black pixels are known but gray pixels are missing information to be recovered /restored.

Figure: Example of result: first row = the input (the condition) with unknown points; second row: ground truth: third row : the result of the algorithm

Numerical results : MNIST

Test on the MNIST database

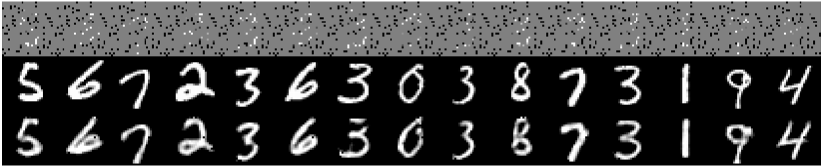


Figure: MNIST reconstruction results. The restoration has very good quality. In some cases the performance is beyond standard human results.

Numerical results : MNIST further results

Test on the MNIST database; the generation is conditional to the information available. White and black pixels are known but gray pixels are missing information to be recovered /restored.

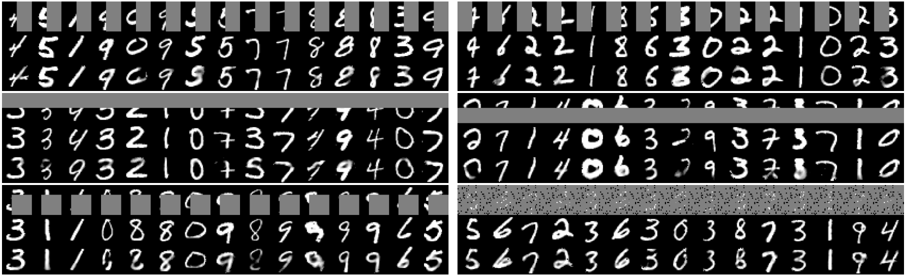


Figure: MNIST reconstruction results. The restoration has very good quality. In some cases the performance is beyond standard human results.

General references I

[1] Ruoteng Li, Loong-Fah Cheong, and Robby T. Tan. "Heavy Rain Image Restoration: Integrating Physics Model and Conditional Adversarial Learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[2] Charles A Micchelli. "Interpolation of scattered data: distance matrices and conditionally positive definite functions". In: *Approximation theory and spline functions*. Springer, 1984, pp. 143–145.

[3] Isaac J Schoenberg. "Metric spaces and completely monotone functions". In: *Annals of Mathematics* (1938), pp. 811–841.

[4] Gabriel Turinici. *Deep Conditional Measure Quantization*. 2023. DOI: 10.48550/ARXIV.2301.06907. URL: https://arxiv.org/abs/2301.06907.

General references II

[5] Gabriel Turinici. *Huber-energy measure quantization*. 2023. arXiv: 2212.08162 [stat.ML].

[6] Gabriel Turinici. "Radon–Sobolev Variational Auto-Encoders". In: *Neural Networks* 141 (2021), pp. 294–305. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2021.04.018.