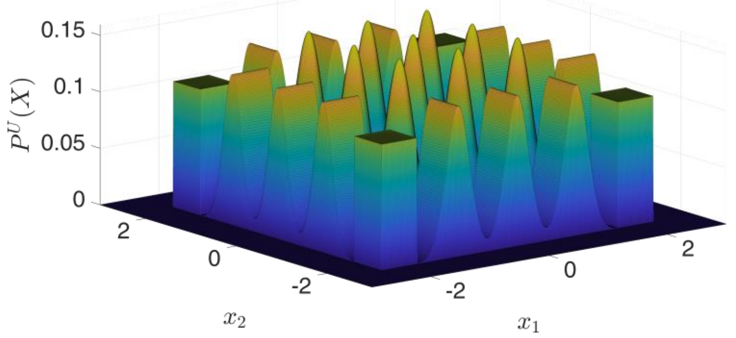
General Probabilistic Surface Optimization

Dmitry Kopitkov

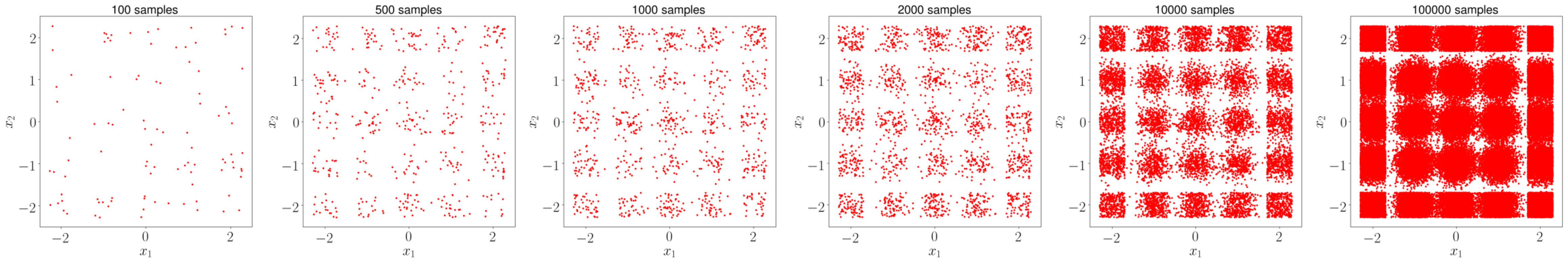
TECHNION
Israel Institute
of Technology

January 2021

Preliminaries



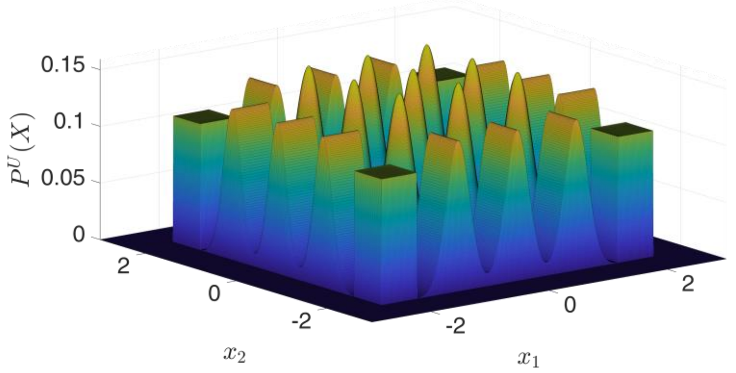
- Probability density function (pdf) $\mathbb{P}^U(X)$ defined over $\mathbb{R}^n$ represents probability/frequency of i.i.d. samples appearing in various neighborhoods/areas of the domain $\mathbb{R}^n$:



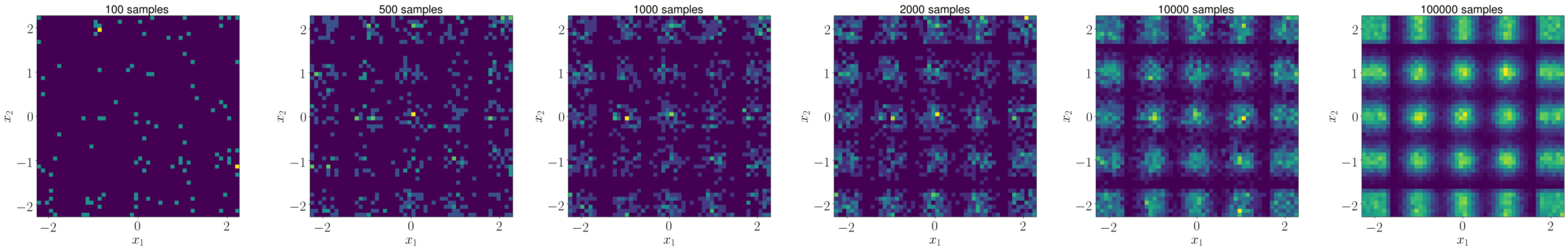
- Knowledge of $\mathbb{P}^U(X)$ is **extremely** useful for many ML problems
- Inferring it from i.i.d. samples $\{X_i^U\}_{i=1}^{N^U}$ is a basic yet very challenging statistical estimation task a.k.a. **density estimation problem**

TECHNION Israel Institute of Technology

Preliminaries



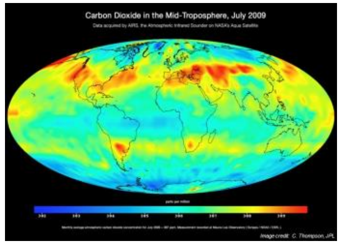
- Possible solutions:
    - MLE, NCE, KDE, etc.
    - Huge research amount was done
- The simplest idea behind all of them is just a histogram:

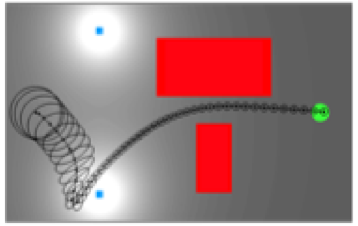
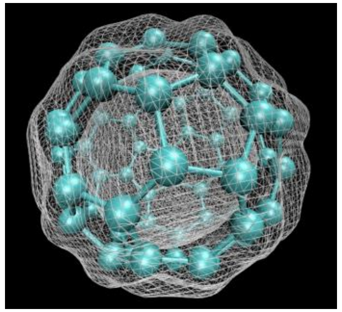


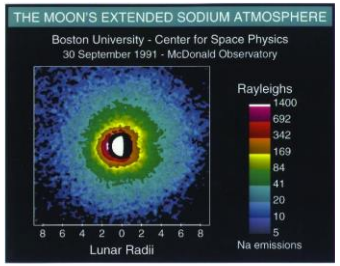
- Define bins over $\mathbb{R}^n$, count samples in each bin, normalize (~divide by total amount of samples)
- Each existing approach has somewhat similar behavior, if we look deep enough into it

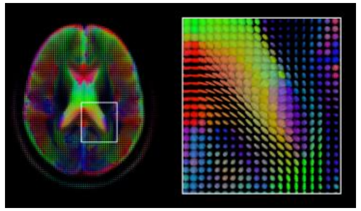
Motivation

- Consider two datasets $\{X_i^U\}_{i=1}^{N^U}$ and $\{X_i^D\}_{i=1}^{N^D}$ from **arbitrary** densities $\mathbb{P}^U(X)$ and $\mathbb{P}^D(X)$
- **Our goal** is to analyze data of these datasets which involves:
  - Density estimation
  - Conditional density
  - Density divergence/ratio
  - Distribution transformation/sampling
- Extremely and widely applicable in:
  - Robotics, computer science, economics, medicine and science in general

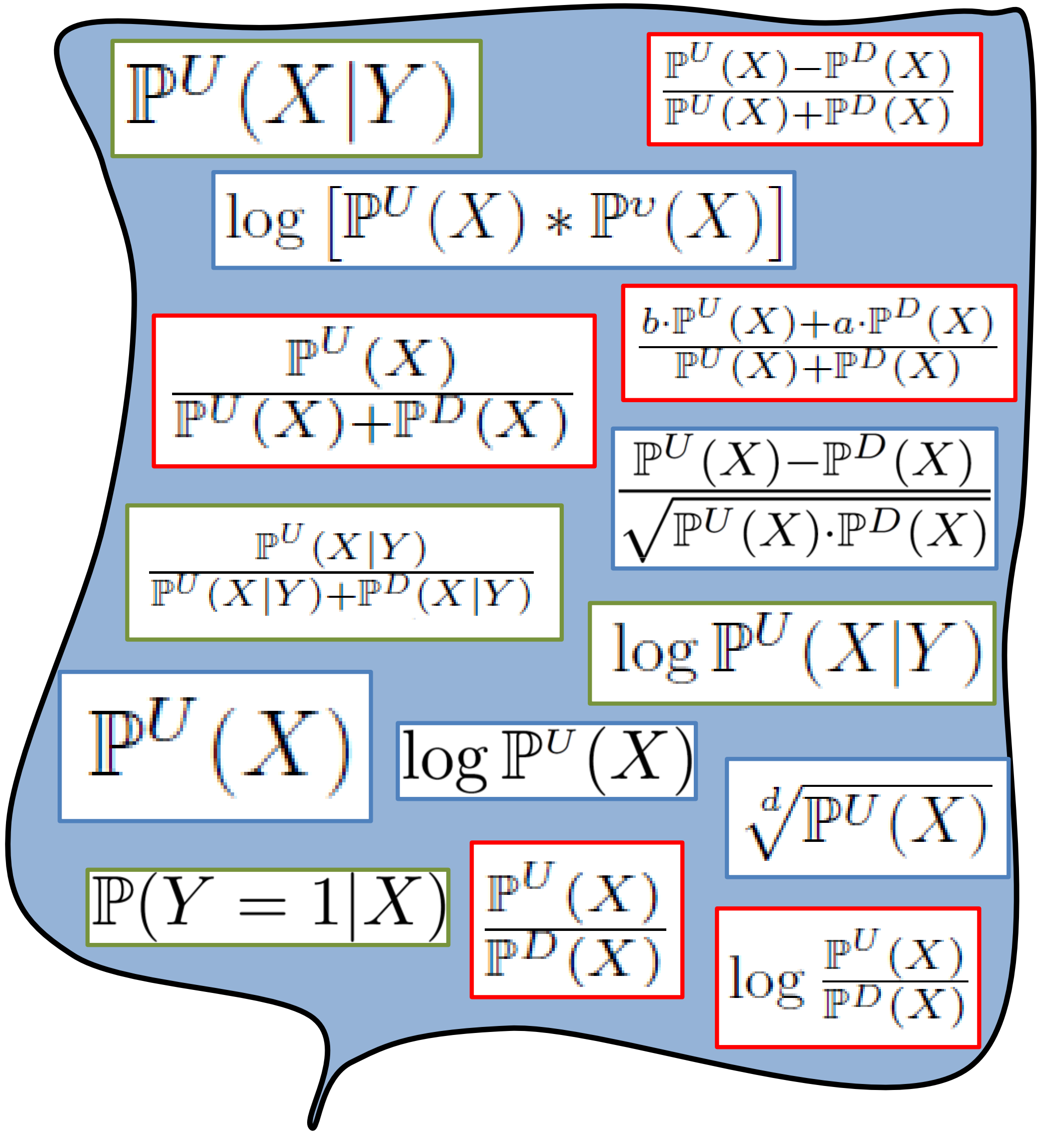








Motivation

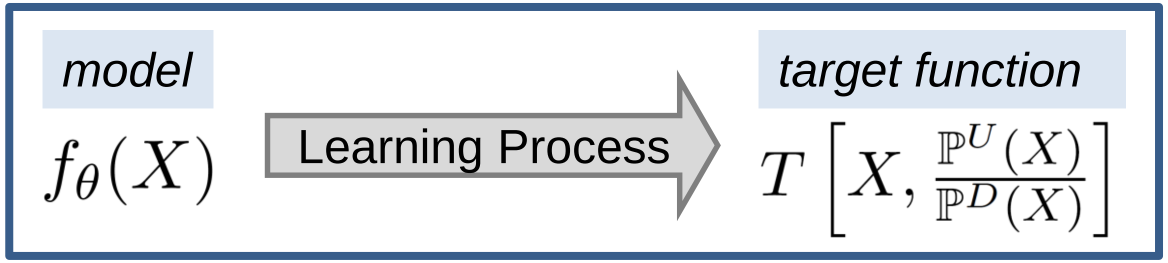
- Estimation of $\mathbb{P}^U(X)$ from $\{X_i^U\}_{i=1}^{N^U}$ is important for:
  - Measurement likelihood model
  - Distribution entropy
  - Image denoising
- Estimation of $\frac{\mathbb{P}^U(X)}{\mathbb{P}^D(X)}$ from $\{X_i^U\}_{i=1}^{N^U}$ and $\{X_i^D\}_{i=1}^{N^D}$:
  - Anomaly detection
  - Divergence learning (e.g. in generative models)
- Estimation of $\log \mathbb{P}^U(X)$ and $\log \frac{\mathbb{P}^U(X)}{\mathbb{P}^D(X)}$ is more numerically stable
- Many problems require us to learn some function $T\left[X, \frac{\mathbb{P}^U(X)}{\mathbb{P}^D(X)}\right]$ of ratio $\frac{\mathbb{P}^U(X)}{\mathbb{P}^D(X)}$
- **Hundreds** of papers with various probabilistic methods for different T exist



Unified Formulation of Statistical Target Functions

- Various target functions can be formulated in a general way as $T\left[X, \frac{\mathbb{P}^U(X)}{\mathbb{P}^D(X)}\right]$:
  - $T\left[X, \frac{\mathbb{P}^U(X)}{\mathbb{P}^D(X)}\right] = \frac{\mathbb{P}^U(X)}{\mathbb{P}^D(X)}$ for $T[X, z] = z$
  - $T\left[X, \frac{\mathbb{P}^U(X)}{\mathbb{P}^D(X)}\right] = \log \frac{\mathbb{P}^U(X)}{\mathbb{P}^D(X)}$ for $T[X, z] = \log z$
  - $T\left[X, \frac{\mathbb{P}^U(X)}{\mathbb{P}^D(X)}\right] = \mathbb{P}^U(X)$ for $T[X, z] = \mathbb{P}^D(X) \cdot z$
  - $T\left[X, \frac{\mathbb{P}^U(X)}{\mathbb{P}^D(X)}\right] = \log \mathbb{P}^U(X)$ for $T[X, z] = \log \mathbb{P}^D(X) + \log z$
- Similarly, many more statistical modalities can be formulated in the same way
- Allows to treat different inference tasks in a unified way

Related Work



- Desired inference task:

- Estimation frameworks:
  - Bregman divergence based methods
  - '*f*'-divergence based methods (e.g. '*f*'-GAN [4,5])

- Divergence-based objective functions:
  - Maximum-Likelihood estimators (based on KL divergence)
  - Noise-contrastive estimators [8]
  - Energy-based unnormalized models (e.g. Boltzman Machines)
  - Critic losses of GANs
  - Many other methods that learn various target functions $T\left[X, \frac{\mathbb{P}^U(X)}{\mathbb{P}^D(X)}\right]$

Research Goals/Questions

- Statistical inference:
  - Deeper understanding of probabilistic modeling
  - How all methods are related to each other?
  - Proposal of new/improved density estimators
  - Make it easy and intuitive!

- Deep Models:
  - Apply neural networks (NNs) to infer intricate probabilistic modalities
  - Understand gradient-based optimization dynamics of NNs
  - Generalization/interpolation, bias-variance, etc.

Contributions

- Statistical inference:
  - Probabilistic Surface Optimization (PSO) estimation framework [1]
  - Offers infinitely many objective functions to learn (almost) any target $T\left[X, \frac{\mathbb{P}^U(X)}{\mathbb{P}^D(X)}\right]$
  - Systematic and simple theory of unsupervised learning
  - Mechanical recovery of existing and novel statistical objective functions

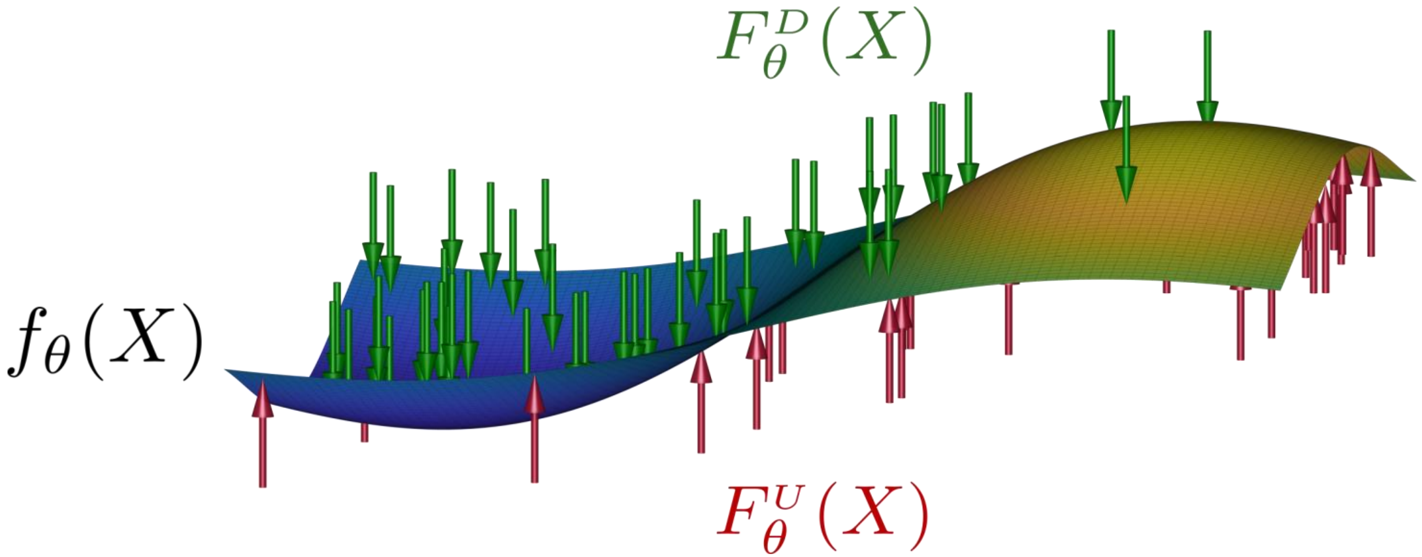
- Deep Models:
  - Relation between PSO performance and the model kernel (a.k.a. Neural Tangent Kernel)
  - Model kernel dynamics and its dependence on NN architecture

*[1] **D. Kopitkov**, V. Indelman, "General Probabilistic Surface Optimization and Log Density Estimation", 2020, <arXiv>*

Contents Outline

Probabilistic Surface Optimization (PSO)

- Consider function space $\mathcal{F}$ containing functions $f_\theta(X) : \mathbb{R}^n \to \mathbb{R}$
- Key idea: view model f_θ as a high-dimensional surface, pushed to equilibrium by virtual forces



- PSO concepts of force equilibrium allow to estimate various statistical modalities of given data (e.g. pdf function), by enforcing $f_\theta(X)$ to converge to any desired target $T\left[X, \frac{\mathbb{P}^U(X)}{\mathbb{P}^D(X)}\right]$

PSO Estimation Framework

- Consider two densities $\mathbb{P}^U(X)$ and $\mathbb{P}^D(X)$ over $\mathbb{R}^n$ with identical support (not mandatory and can be relaxed..), and two corresponding datasets $\{X_i^U\}_{i=1}^{N^U}$ and $\{X_i^D\}_{i=1}^{N^D}$
- Choose any two *magnitude* functions (some minor conditions should hold):

$$M^U(X,s) : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}, \qquad M^D(X,s) : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$$

- Iterate gradient-descent algorithm (GD) via $\theta_{t+1} = \theta_t - \delta \cdot d\theta$ with:

$$d\theta = -\frac{1}{N^U}\sum_{i=1}^{N^U} M^U\left[X_i^U, f_\theta(X_i^U)\right] \cdot \nabla_\theta f_\theta(X_i^U) + \frac{1}{N^D}\sum_{i=1}^{N^D} M^D\left[X_i^D, f_\theta(X_i^D)\right] \cdot \nabla_\theta f_\theta(X_i^D)$$

i.i.d. samples: $X^U \sim \mathbb{P}^U$, $X^D \sim \mathbb{P}^D$

PSO Estimation Framework

Inputs:
$\mathbb{P}^U$ and $\mathbb{P}^D$: *up* and *down* densities
M^U and M^D : *magnitude* functions
θ : initial parameters of model $f_\theta \in \mathcal{F}$
δ : learning rate
Outputs: f_{θ^*} : PSO solution that satisfies *balance state* in Eq. (2)
begin:
  while *Not converged* **do**
    Obtain samples $\{X_i^U\}_{i=1}^{N^U}$ from $\mathbb{P}^U$
    Obtain samples $\{X_i^D\}_{i=1}^{N^D}$ from $\mathbb{P}^D$
    Calculate $d\theta$ via Eq. (1)
    $\theta = \theta - \delta \cdot d\theta$
  end
  $\theta^* = \theta$
end

Algorithm 1: PSO estimation algorithm. Sample batches can be either identical or different for all iterations, which corresponds to GD and stochastic GD respectively.

Perform a standard GD via the defined $d\theta$

- Claim: convergence is at $\mathbb{P}^U(X) \cdot M^U[X, f^*(X)] = \mathbb{P}^D(X) \cdot M^D[X, f^*(X)]$

PSO Derivation - Euler-Lagrange Equation

- Consider a general-form PSO loss:

$$L_{PSO}(f) = -\mathop{\mathbb{E}}_{X \sim \mathbb{P}^U} \widetilde{M}^U [X, f(X)] + \mathop{\mathbb{E}}_{X \sim \mathbb{P}^D} \widetilde{M}^D [X, f(X)]$$

antiderivative of M^U

$$M^U[X, s] = \frac{\partial \widetilde{M}^U(X,s)}{\partial s}$$

antiderivative of M^D

$$M^D[X, s] = \frac{\partial \widetilde{M}^D(X,s)}{\partial s}$$

- According to Euler-Lagrange equation of $L_{PSO}(f)$, optima $f^* = \arg\min_{f \in \mathcal{F}} L_{PSO}(f)$ satisfies the variational equilibrium:

$$\mathbb{P}^U(X) \cdot M^U [X, f^*(X)] = \mathbb{P}^D(X) \cdot M^D [X, f^*(X)]$$

PSO Derivation - Optimization

- Solve optimization over $f_\theta \in \mathcal{F}$: $\min_{f_\theta \in \mathcal{F}} L_{PSO}(f_\theta)$

- Loss gradient w.r.t. θ:

define variational equilibrium

$$\nabla_\theta L_{PSO}(f_\theta) = - \mathop{\mathbb{E}}_{X \sim \mathbb{P}^U} M^U [X, f_\theta(X)] \cdot \nabla_\theta f_\theta(X) + \mathop{\mathbb{E}}_{X \sim \mathbb{P}^D} M^D [X, f_\theta(X)] \cdot \nabla_\theta f_\theta(X)$$

define metric over function space

- Approximated by $d\theta$:

$$d\theta = -\frac{1}{N^U} \sum_{i=1}^{N^U} M^U [X_i^U, f_\theta(X_i^U)] \cdot \nabla_\theta f_\theta(X_i^U) + \frac{1}{N^D} \sum_{i=1}^{N^D} M^D [X_i^D, f_\theta(X_i^D)] \cdot \nabla_\theta f_\theta(X_i^D)$$

PSO Derivation - Balance State

- Stationary solution at (Euler-Lagrange Eq. of loss $L_{PSO}(f_\theta)$):

$$\mathbb{P}^U(X) \cdot M^U[X, f^*(X)] = \mathbb{P}^D(X) \cdot M^D[X, f^*(X)]$$

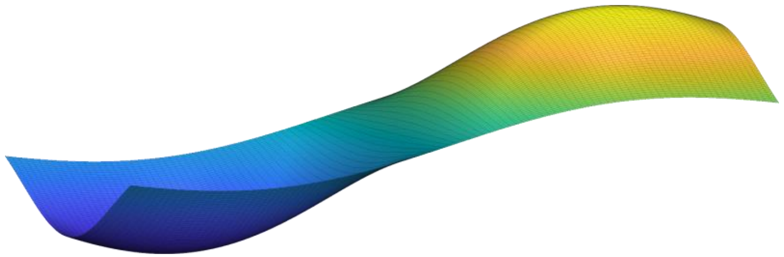
  - Choice of $\{M^U, M^D\}$ controls convergence f^*
  - Knowledge of antiderivatives $\{\widetilde{M}^U, \widetilde{M}^D\}$ is not necessary
  - Can be used for (ratio) density estimation, but not only
  - *Magnitudes* must satisfy some minor “sufficient” conditions

Contents Outline

Physical System Perspective – Model Kernel

- Model f_θ as a representation of the surface:

$$\boxed{f_\theta(X) : \mathbb{R}^n \rightarrow \mathbb{R}}$$



- Examples of the function space $\mathcal{F}$:
  - NNs – fully-connected, CNN, ResNet, etc.
  - RKHS – $f_\theta(X) = \phi(X)^T \cdot \theta$ defined via reproducing kernel $k(X, X') = \phi(X)^T \cdot \phi(X')$

- Important property of $\mathcal{F}$ – the model kernel: $g_\theta(X, X') \triangleq \nabla_\theta f_\theta(X)^T \cdot \nabla_\theta f_\theta(X')$
  - Responsible for interpolation/extrapolation during GD
  - NNs – a.k.a. Neural Tangent Kernel (NTK) [6]
  - RKHS – $g_\theta(X, X') \equiv k(X, X')$

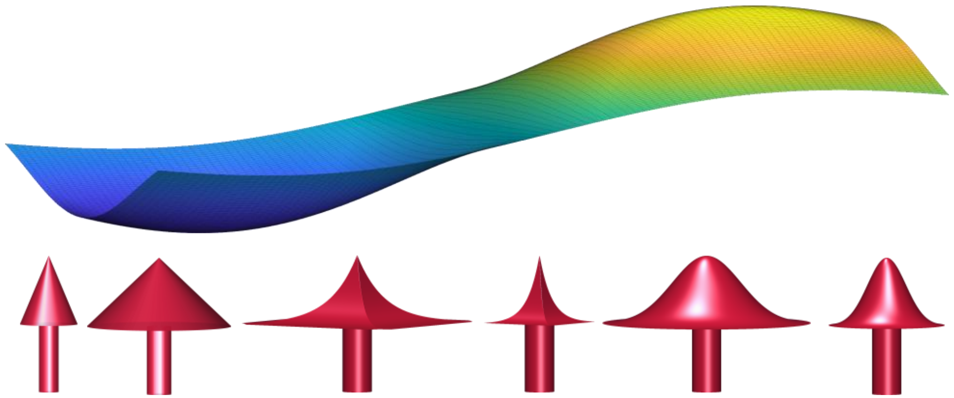
Physical System Perspective – Model Kernel

- Consider update $\theta_{t+1} = \theta_t + \nabla_\theta f_{\theta_t}(X)$. Then:

$$f_{\theta_{t+1}}(X') - f_{\theta_t}(X') \approx \nabla_\theta f_{\theta_t}(X')^T \cdot \nabla_\theta f_{\theta_t}(X) \triangleq g_{\theta_t}(X, X')$$

first-order Taylor approximation for NNs, identity for RKHS

- When we "push"/optimize at X, our model f_θ at any other X' changes according to $g_\theta(X, X')$, approximately



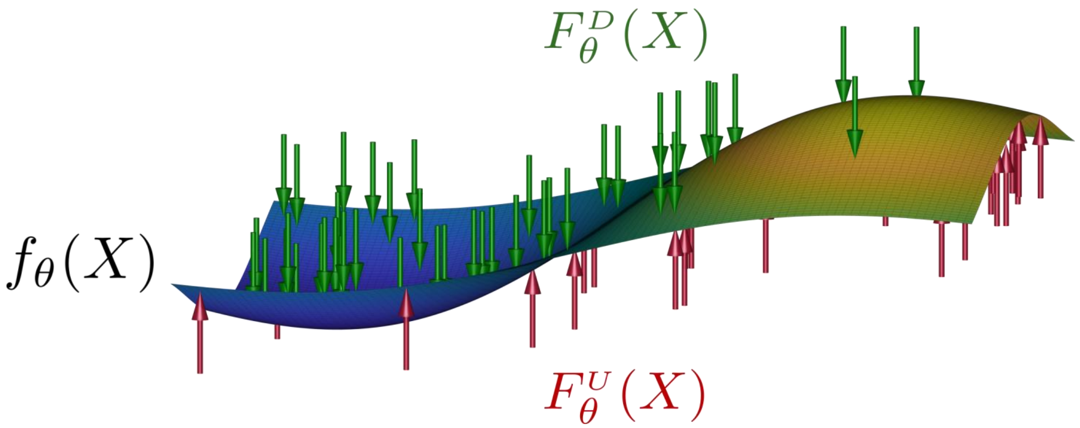
- Intuitively, $g_\theta(X, X')$ can be viewed as the shape of a pushing "wand":
- Can we use this "wand" to sculpt f_θ to any desired shape?

Physical System Perspective

- Consider again PSO update:

$$d\theta = -\frac{1}{N^U}\sum_{i=1}^{N^U} M^U\left[X_i^U, f_\theta(X_i^U)\right]\cdot\nabla_\theta f_\theta(X_i^U) + \frac{1}{N^D}\sum_{i=1}^{N^D} M^D\left[X_i^D, f_\theta(X_i^D)\right]\cdot\nabla_\theta f_\theta(X_i^D)$$

- We push *up* at $X_i^U \sim \mathbb{P}^U$ with
  force *magnified* by $M^U\left[X_i^U, f_\theta(X_i^U)\right]$
- We push *down* at $X_i^D \sim \mathbb{P}^D$ with
  force *magnified* by $M^D\left[X_i^D, f_\theta(X_i^D)\right]$



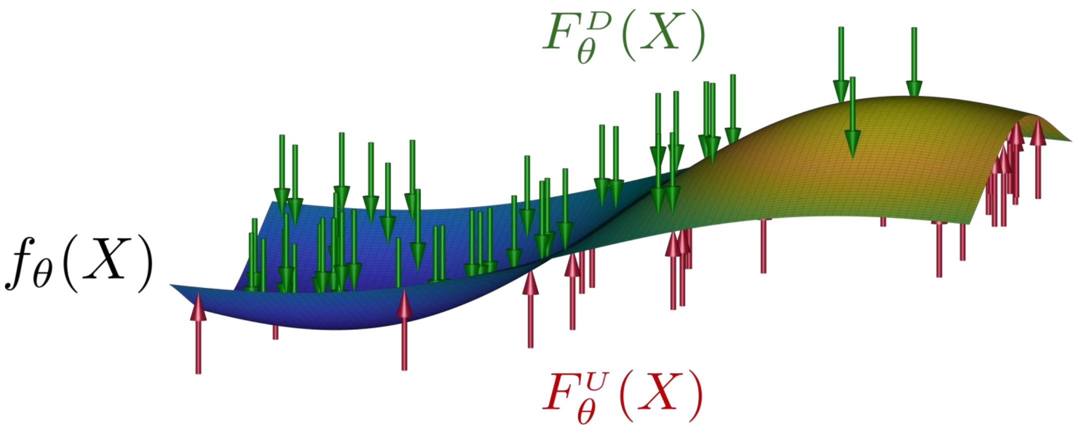
- $g_\theta(X, X')$ serves as sort of a sculpture tool set

Physical System Perspective

- In asymptotic regime $\min(N^U, N^D) \to \infty$ and when the bandwidth of g_θ goes to zero, the point-wise *up* and *down* averaged forces at any X can be defined as:

$$F_\theta^U(X) \triangleq \mathbb{P}^U(X) \cdot M^U[X, f_\theta(X)], \quad F_\theta^D(X) \triangleq \mathbb{P}^D(X) \cdot M^D[X, f_\theta(X)]$$

- Yields a dynamical system:



- PSO Equilibrium (variational equilibrium) at:

$$F_\theta^U(X) = F_\theta^D(X)$$

- Actual GD equilibrium **strongly** depends on g_θ, N^U and N^D!

Contents Outline

Simple Example – Apply PSO Equilibrium for Inference

- Consider PSO estimator (also known as uLSIF [7]) with *magnitudes*:

$$M^U\left[X, f(X)\right] = 1, \quad M^D\left[X, f(X)\right] = f(X)$$

- Solving PSO balance state:

$$\frac{\mathbb{P}^U(X)}{\mathbb{P}^D(X)} = \frac{M^D\left[X, f^*(X)\right]}{M^U\left[X, f^*(X)\right]} = \frac{f^*(X)}{1} \quad \Rightarrow \quad f^*(X) = \frac{\mathbb{P}^U(X)}{\mathbb{P}^D(X)}$$

- We got a method that infers a density ratio from data $\{X_i^U\}_{i=1}^{N^U}$ and $\{X_i^D\}_{i=1}^{N^D}$

Simple Example (Part 2)

- Consider PSO estimator with *magnitudes* (denoted as DeepPDF [2]):

$$M^U[X, f(X)] = \mathbb{P}^D(X), \qquad M^D[X, f(X)] = f(X)$$

where $\mathbb{P}^D$ is a known auxiliary distribution (e.g. Uniform, Gaussian)

- Solving PSO balance state: $$\frac{\mathbb{P}^U(X)}{\mathbb{P}^D(X)} = \frac{M^D[X, f^*(X)]}{M^U[X, f^*(X)]} = \frac{f^*(X)}{\mathbb{P}^D(X)} \quad \Rightarrow \quad f^*(X) = \mathbb{P}^U(X)$$

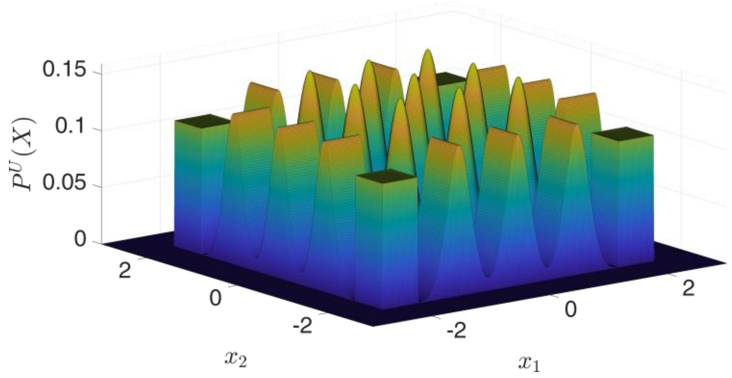
- We got a new method for density estimation

[2] ***D. Kopitkov****, V. Indelman, "Deep PDF: Probabilistic Surface Optimization and Density Estimation", 2018, <arXiv>*

DeepPDF - Demonstration

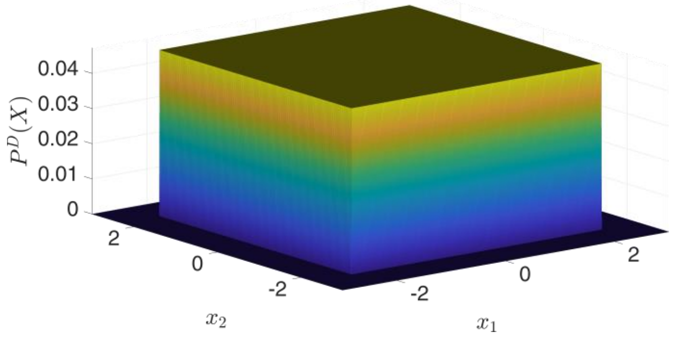
- DeepPDF *magnitudes*: $M^U\left[X, f(X)\right] = \mathbb{P}^D(X)$, $M^D\left[X, f(X)\right] = f(X)$
- Densities:

$\mathbb{P}^U$



learned

$\mathbb{P}^D$



known

- Given samples $\{X_i^U\}_{i=1}^{N^U}$ and $\{X_i^D\}_{i=1}^{N^D}$ from $\mathbb{P}^U(X)$ and $\mathbb{P}^D(X)$, we "push" $f_\theta(X)$ to have a shape of $\mathbb{P}^U(X)$, see online <demo1, demo2>

PSO Convergence – More General View

- Define a ratio $R(X, s) : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$:

$$R[X, s] = \frac{M^D[X,s]}{M^U[X,s]}$$

inversion

- Define PSO convergence $T(X, z) : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$:

$$f^*(X) = T\left[X, \frac{\mathbb{P}^U(X)}{\mathbb{P}^D(X)}\right]$$

- Then, R and T are inverses, $R \equiv T^{-1}$

Inverse Functions h and h^{-1}: $\forall z : h^{-1}[X, h[X, z]] = z$ and $\forall s : h[X, h^{-1}[X, s]] = s$.

- PSO instance for any target $T\left[X, \frac{\mathbb{P}^U(X)}{\mathbb{P}^D(X)}\right]$ can be constructed by:

1 finding its inverse R, 2 finding *magnitudes* $\{M^U, M^D\}$ whose ratio is $R \equiv \frac{M^D}{M^U}$

Example: Construct New PSO Methods for Log-density

- Let's invent new PSO methods to approximate $\log \mathbb{P}^U(X)$
- The corresponding PSO convergence $T\left[X, \frac{\mathbb{P}^U(X)}{\mathbb{P}^D(X)}\right]$ is described by:

$$T(X, z) = \log \mathbb{P}^D(X) + \log z$$

- Its inverse is:

$$R(X, s) = \frac{\exp s}{\mathbb{P}^D(X)}$$

1 *inverse w.r.t. second argument*

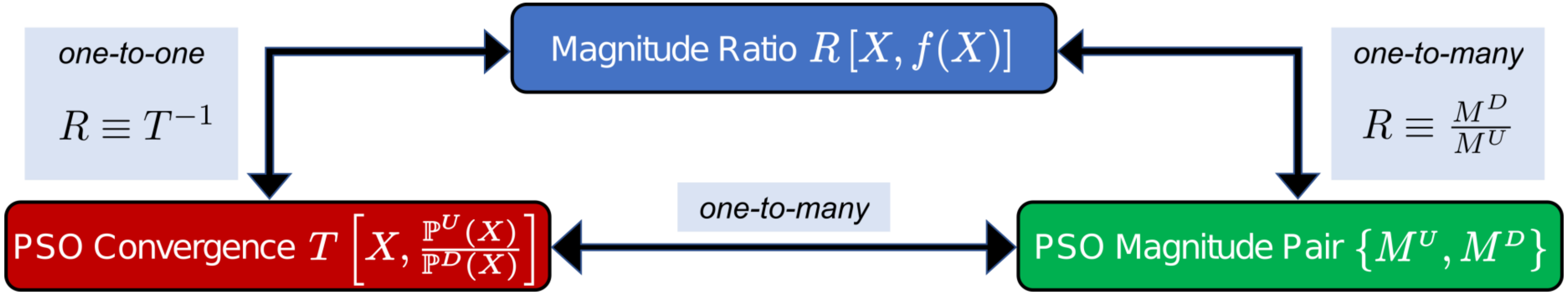
- Then, any PSO instance with $\{M^U, M^D\}$ satisfying below criteria (+ some "sufficient" conditions) will produce the required convergence:

$$\frac{M^D(X, f(X))}{M^U(X, f(X))} = \frac{\exp[f(X)]}{\mathbb{P}^D(X)}$$

2 *propose magnitudes with the required ratio*

Inverse Relation $R \equiv T^{-1}$

- One-to-one relationship – knowing one we can identify other
- Antiderivatives of R and T are related via Legendre transformation (i.e. they are convex conjugate of one another)
- Reminds relation between Lagrangian and Hamiltonian mechanics, opens a bridge between control theory and learning theory
- Infinitely many pairs $\{M^U, M^D\}$ produce the same ratio R. Which should we choose?



Bounding PSO Magnitudes

- Consider any $\{M^U, M^D\}$ with the corresponding convergence T
- Then, a new pair has the same convergence:

$$M^U_{bounded}[X,s] = \frac{M^U[X,s]}{|M^U[X,s]| + |M^D[X,s]|}, \quad M^D_{bounded}[X,s] = \frac{M^D[X,s]}{|M^U[X,s]| + |M^D[X,s]|}$$

- New pair is bounded to [-1, 1]
- Bounded magnitudes are typically more numerically stable during the optimization
- Turns the objective function to be Lipschitz continuous
- Other norms can also be used
- Most of the popular losses have bounded *magnitudes* (NCE, Logistic loss, Cross-entropy)

PSO Instances - Summary so far..

- Single algorithm to infer numerous statistical modalities - in a similar manner we can learn $\mathbb{P}^U(X)$, $\frac{\mathbb{P}^U(X)}{\mathbb{P}^D(X)}$ or any function of it, $T\left[X, \frac{\mathbb{P}^U(X)}{\mathbb{P}^D(X)}\right]$
- Simple and intuitive
- Virtual force equilibrium – surprising and easy to understand
- We can mechanically recover **almost** all existing objective functions for density estimation (e.g. MLE, Noise Contrastive Estimation, Importance Sampling, etc.)
- Recovery of various statistical divergencies
- Cross-entropy and critic losses of most GANs
- Conditional density estimation by applying Bayes theorem

Contents Outline

Model GD Dynamics

- So far, we considered variational equilibrium. Now we shall focus on understanding GD behavior.
- First-order dynamics of f_θ (t is iteration index):

$$f_{t+1} \approx f_t - \delta \cdot G_t F(f_t)$$

we are still in an asymptotic regime:
$\min(N^U, N^D) \to \infty$

- Euler-Lagrange Eq. (steepest direction in a function space):

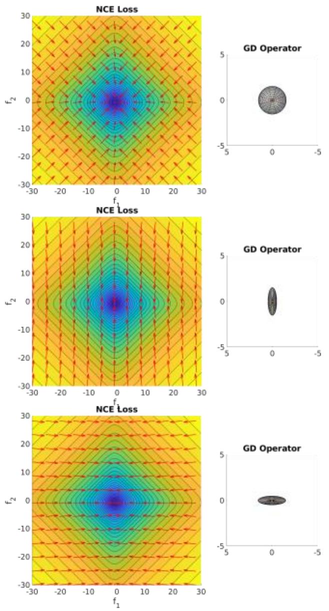
$$[Fu](\cdot) = -\mathbb{P}^U(\cdot) \cdot M^U [\cdot, u(\cdot)] + \mathbb{P}^D(\cdot) \cdot M^D [\cdot, u(\cdot)]$$

- GD operator (integral operator w.r.t. model kernel):

$$[G_t u](\cdot) = \int g_t(\cdot, X) u(X) dX$$

- How G_t affects the inference? New equilibrium: $F(f_\infty) \equiv 0$ $\Rightarrow$ $G_\infty F(f_\infty) \equiv 0$

Role of GD Operator in $f_{t+1} \approx f_t - \delta \cdot G_t F(f_t)$

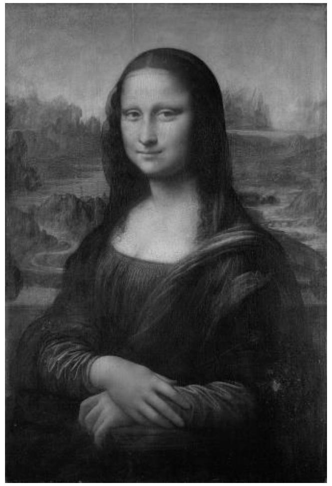


- G_t is a metric over a function space $\mathcal{F}$
- Eigenvalues/eigenfunctions of $g_t(X, X')$ define which directions are easy/fast to go to, and in which directions movement is **too** slow
- Alignment between $F(f_t)$ and eigenfunctions with largest eigenvalues decides the optimization outcome
- Kernel alignment methods are very popular in RKHS literature [9,10]
- NNs perform such alignment during the optimization!

Contents Outline

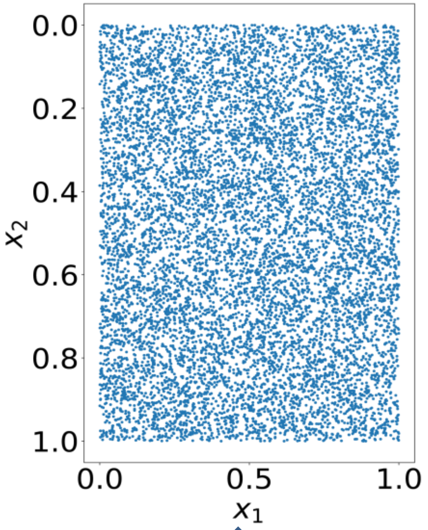
NN Model Kernel Alignment

Target f^*



Learned f_θ

Samples

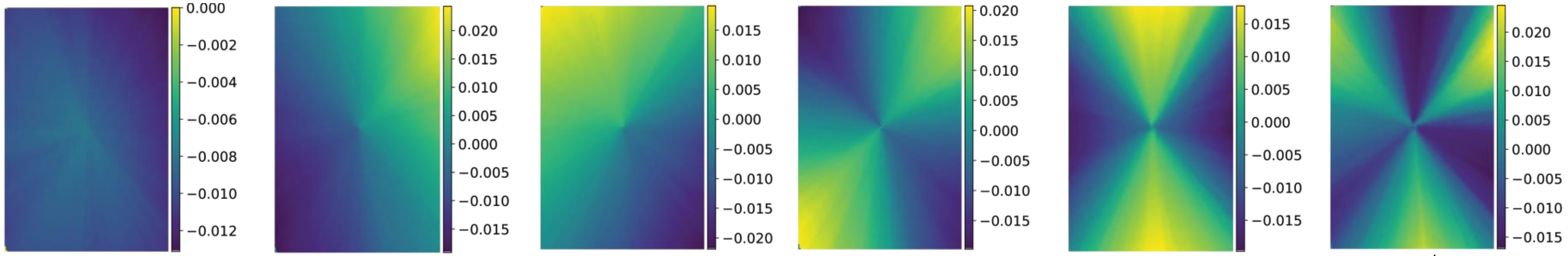


- Consider a 2D regression task:
- Setup:
  - 10000 samples X^i, Y^i
  - Least-Squares loss
  - GD for 600000 steps
- Goal: investigate how $g_t(X, X')$, its eigenvalues $\{\lambda_i^t\}_{i=1}^N$ and eigenfunctions $\{\overline{\upsilon}_i^t\}_{i=1}^N$ change along the GD optimization

NN Model Kernel Alignment

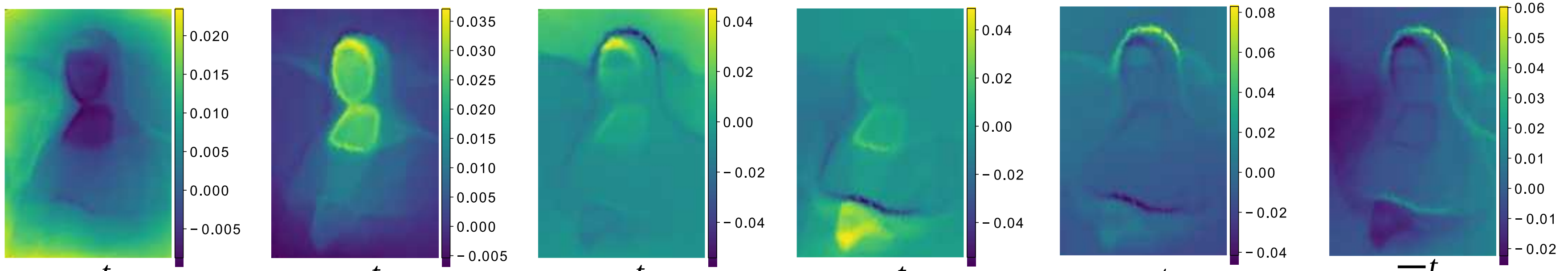
- First **top** eigenfunctions for Leaky-Relu FC NN with 6 layers at
  - $t = 0$:

lower frequency — higher frequency



$\bar{\upsilon}_1^t$ $\bar{\upsilon}_2^t$ $\bar{\upsilon}_3^t$ $\bar{\upsilon}_4^t$ $\bar{\upsilon}_5^t$ $\bar{\upsilon}_6^t$

  - $t = 20000$:

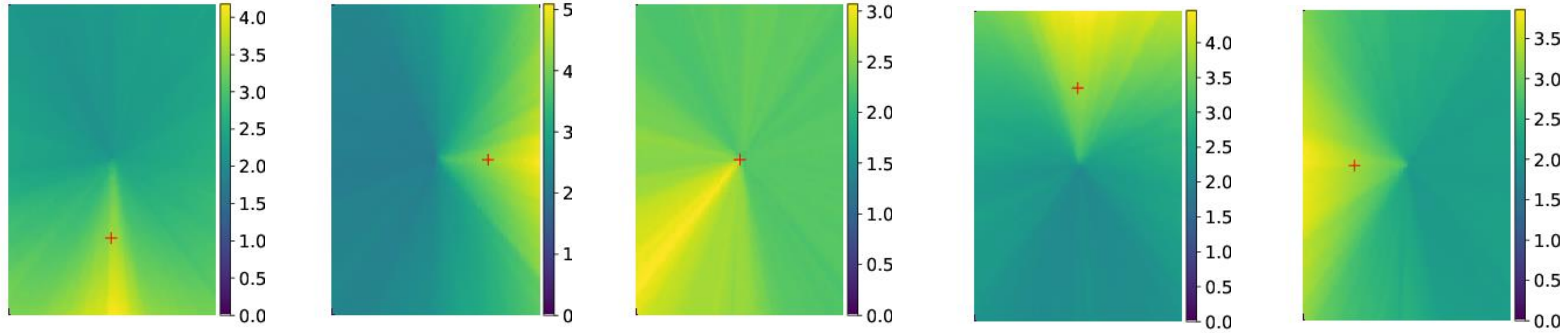


$\bar{\upsilon}_1^t$ $\bar{\upsilon}_2^t$ $\bar{\upsilon}_3^t$ $\bar{\upsilon}_4^t$ $\bar{\upsilon}_5^t$ $\bar{\upsilon}_6^t$

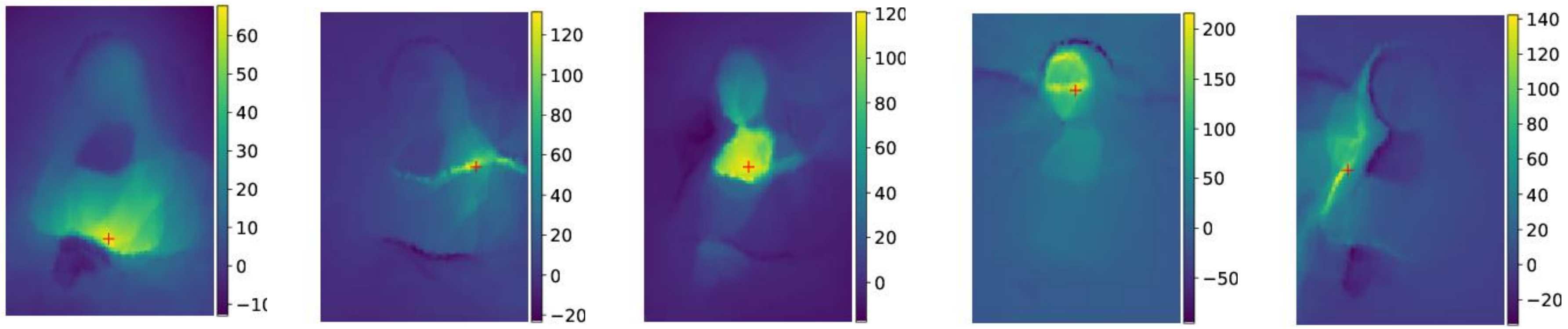
TECHNION Israel Institute of Technology

NN Model Kernel Alignment

- $g_t(X, X')$, where X marked by +, for Leaky-Relu FC NN with 6 layers at
  - $t = 0$:

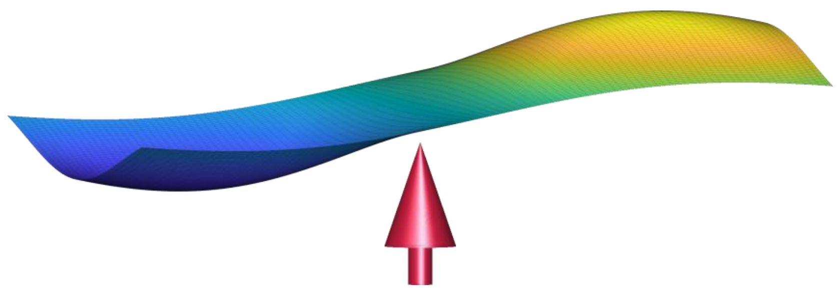


  - $t = 50000$:



Experiment Outcome

- Strong evidence that **top** eigenfunctions of $g_t(X, X')$ align towards f^*
  - In other words, both $g_t(X, X')$ and f_t converge to f^*
- Increases movement speed into direction f^* within space $\mathcal{F}$
- Intuitively, our pushing stick obtains a shape that aligns well with the surface



- Deeper NNs have higher alignment, which also explains their performance superiority
- Beyond GD and L2 loss, similar behavior was also observed for SGD, Adam and unsupervised PSO learning losses (see [3])

*[3] **D. Kopitkov**, V. Indelman, "Neural Spectrum Alignment: Empirical Study", International Conference on Artificial Neural Networks (ICANN) 2020, <arXiv>*

Summary

- Conclusions:
  - Proposed PSO framework allows to learn (almost) any target $T\left[X, \frac{\mathbb{P}^U(X)}{\mathbb{P}^D(X)}\right]$
  - Strong intuition allows to use PSO force concepts for various numerous applications
  - Actual equilibrium strongly depends on properties of the model kernel
  - In NNs $g_\theta(X, X')$ aligns itself with the target function (for currently unknown reasons)
- Future work:
  - Robust statistics – which PSO instance is better? What is optimal? How it is related to the kernel?
  - Convergence rates? Generalization error? Impact of $g_\theta(X, X')$ in small dataset setting?
  - Design a NN architecture to control properties of $g_\theta(X, X')$
  - Better regularization of models in high-dimensional small dataset setting

References

- *[1]* ***D. Kopitkov****, V. Indelman, "General Probabilistic Surface Optimization and Log Density Estimation", 2020, <arXiv>*
- *[2]* ***D. Kopitkov****, V. Indelman, "Deep PDF: Probabilistic Surface Optimization and Density Estimation", 2018, <arXiv>*
- *[3]* ***D. Kopitkov****, V. Indelman, "Neural Spectrum Alignment: Empirical Study", International Conference on Artificial Neural Networks (ICANN) 2020, <arXiv>*
- *[4] XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. IEEE Transactions on Information Theory, 56(11):5847-5861, 2010.*
- *[5] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In Advances in Neural Information Processing Systems, pages 271-279, 2016.*
- *[6] Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In Advances in Neural Information Processing Systems (NIPS), pages 8571-8580, 2018.*
- *[7] Takafumi Kanamori, Shohei Hido, and Masashi Sugiyama. A least-squares approach to direct importance estimation. Journal of Machine Learning Research, 10(Jul):1391-1445,2009.*
- *[8] Michael Gutmann and Aapo Hyvarinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pages 297-304, 2010.*
- *[9] Mehmet Gonen and Ethem Alpaydın. Multiple kernel learning algorithms. Journal of machine learning research, 12(Jul):2211–2268, 2011.*
- *[10] Tinghua Wang, Dongyan Zhao, and Shengfeng Tian. An overview of kernel alignment and its applications. Artificial Intelligence Review, 43(2):179–192, 2015.*

Thanks For Listening

Questions?

Extra Material

Convoluted PSO Equilibrium

- Variational PSO balance state $F(f_\infty) \equiv 0$ leads to PSO force equality:

$$\mathbb{P}^U(X) \cdot M^U\left[X, f_\infty(X)\right] = \mathbb{P}^D(X) \cdot M^D\left[X, f_\infty(X)\right]$$

- GD balance state $G_\infty F(f_\infty) \equiv 0$ **changed**! Convoluted with $g_\infty(X, X')$:

$$\int g_\infty(X,X') \cdot \mathbb{P}^U(X') \cdot M^U\left[X', f_\infty(X')\right] dX' = \int g_\infty(X,X') \cdot \mathbb{P}^D(X') \cdot M^D\left[X', f_\infty(X')\right] dX'$$

- Both equilibriums are identical iff G_∞ is an injective (invertible) operator
- Typically, at the optimization end $F(f_\infty)$ is zero only along $g_\infty(X, X')$'s top eigenfunctions
- Hence, a bias from the model kernel is introduced into the solution f_∞

Various Equilibriums

- Variational equilibrium (infinite datasets, infinitely flexible surface):

$$\mathbb{P}^U(X) \cdot M^U[X, f_\infty(X)] = \mathbb{P}^D(X) \cdot M^D[X, f_\infty(X)]$$

- Data-infinite GD equilibrium (infinite datasets, optimization via GD):

properties of $g_\infty(X, X')$

$$\int g_\infty(X, X') \cdot \mathbb{P}^U(X') \cdot M^U[X', f_\infty(X')]\, dX' = \int g_\infty(X, X') \cdot \mathbb{P}^D(X') \cdot M^D[X', f_\infty(X')]\, dX'$$

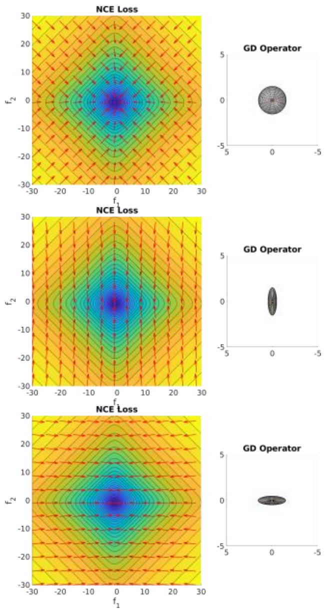
CLT and statistical concentration

- Data-finite GD equilibrium (finite datasets, optimization via GD):

$$\frac{1}{N^U} \sum_{i=1}^{N^U} M^U[X_i^U, f_\infty(X_i^U)] \cdot g_\infty(X, X_i^U) = \frac{1}{N^D} \sum_{i=1}^{N^D} M^D[X_i^D, f_\infty(X_i^D)] \cdot g_\infty(X, X_i^D)$$

- How about mini-batch SGD? Momentum? Adam?...

Role of GD Operator - Additional Aspects



- Spectrum of $g_t(X, X')$ can be considered as an implicit distribution over elements in $\mathcal{F}$ (typical in Gaussian Process literature)
- G_t is constant for RKHS but time-dependent for NNs
- Bandwidth of $g_t(X, X')$ defines if we can move in a direction of high-frequency/"not smooth" functions
- This bandwidth and the eigenvalue decay of G_t are equivalent in some sense, they both represent how "easy" it is to learn functions with many small details/high frequency