Reliable Decision Making Through the Lens of Statistics and Optimization

Speaker: Jie Wang

Date: January 27, 2023

Committee members

- Dr. Yao Xie
  (Industrial and Systems Engineering, Georgia Tech)
- Dr. Alexander Shapiro
  (Industrial and Systems Engineering, Georgia Tech)
- Dr. Guanghui Lan
  (Industrial and Systems Engineering, Georgia Tech)
- Dr. Xin Chen
  (Industrial and Systems Engineering, Georgia Tech)
- Dr. Rui Gao
  (Department of Information, Risk and Operations Management, UT Austin)

Table of Contents

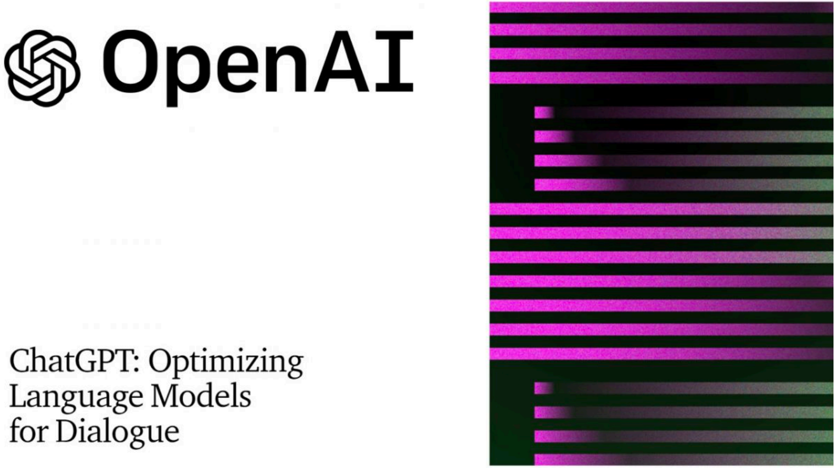
Table of Contents

Question: How to Compare Two Samples

- **Given**: Samples from unknown distributions P and Q.
- **Goal**: Do P and Q differ?

Two-sample Test is Fundamental in Practice

- Scientific Discovery: single-cell data, drug effectiveness, social science;
- Goodness-of-fit tests;

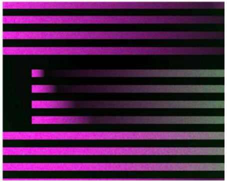


Two-sample Test is Fundamental in Practice

- Scientific Discovery: single-cell data, drug effectiveness, social science;
- Goodness-of-fit tests;

OpenAI

ChatGPT: Optimizing Language Models for Dialogue



ChatGPT detector could help spot cheaters using AI to write essays

A tool called GPTZero can identify whether text was produced by a chatbot, which could help teachers tell if students are getting AI to help with their homework

This article has been viewed 3115 times in the last 24 hours.

TECHNOLOGY 17 January 2023

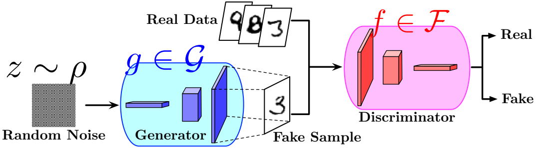
By Alex Wilkins

Two-sample Test is Fundamental in Practice

- Scientific Discovery: single-cell data, drug effectiveness, social science;
- Goodness-of-fit tests;
- Model critics for machine learning models.

LSUN Dataset (Bedroom)

Output from Generative Adversarial Network



Architecture for Generative Adversarial Network

$$\min_{g \in \mathcal{G}} \max_{f \in \mathcal{F}} \ \mathbb{E}_{z \sim \rho}[f(g(z))] - \mathbb{E}_{x \sim \mu}[f(x)]$$

[Chun-Liang et al. NIPS 2017]

A Brief on Stochastic Optimization (SO)

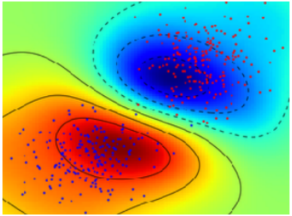
Risk : $\mathcal{R}(\theta;\mathbb{P}) = \mathbb{E}_{\mathbb{P}}[f_\theta(z)]$

Optimal Risk : $\mathcal{R}(\Theta;\mathbb{P}) = \inf_{\theta\in\Theta} \mathbb{E}_{\mathbb{P}}[f_\theta(z)]$

Applications

Supply Chain Mgmt.

Portfolio Mgmt.



Machine Learning

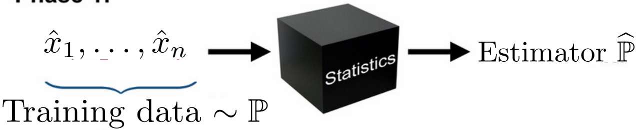
Another Question: How to Find Decision of SO under Distributional Uncertainty?

- Available Information:

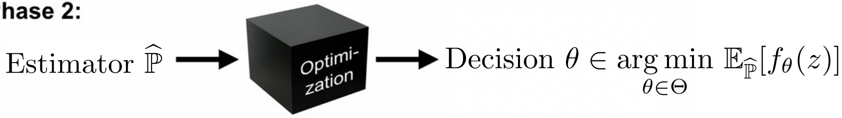
$$\begin{aligned} &\text{Structual :} && \mathbb{P} \text{ is supported on } \Omega \subseteq \mathbb{R}^d \\ &\text{Statistical :} && \hat{x}_1, \ldots, \hat{x}_n \sim \mathbb{P} \end{aligned}$$

- Existing Solution:

Phase 1:



Phase 2:



On the Robustness of State-of-the-art Models

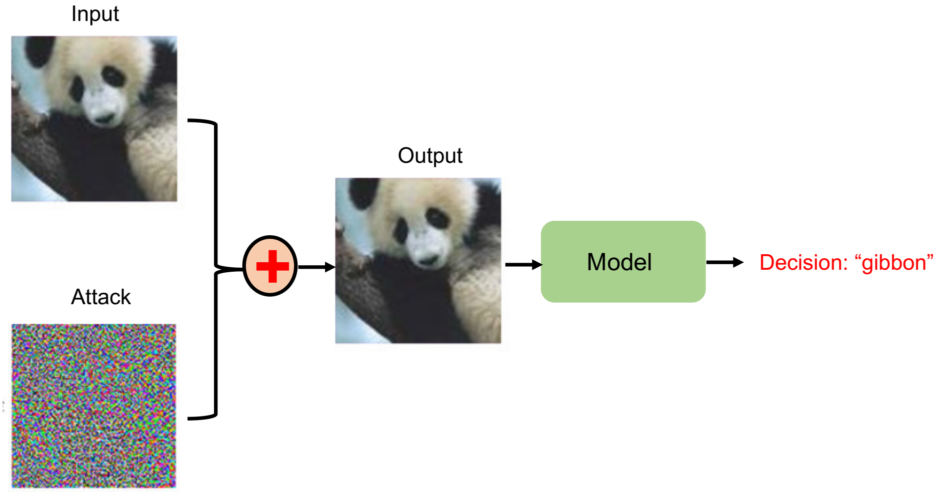
Existing models **generalize poorly** to new environments [Beery et al. ECCV2018]:

(A) **Cow: 0.99**, Pasture: 0.99, Grass: 0.99, No Person: 0.98, Mammal: 0.98

(B) No Person: 0.99, Water: 0.98, Beach: 0.97, Outdoors: 0.97, Seashore: 0.97

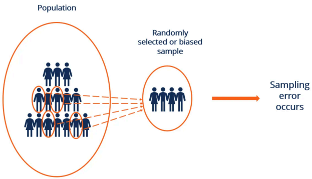
(C) No Person: 0.97, **Mammal: 0.96**, Water: 0.94, Beach: 0.94, Two: 0.94

On the Robustness of State-of-the-art Models

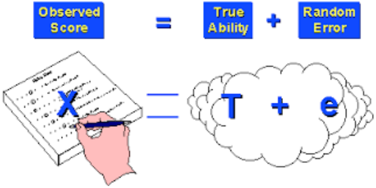


[Goodfellow et al. 2015]

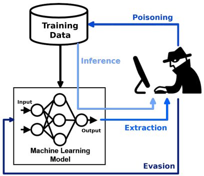
Challenges for Decision-Making Under Uncertainty



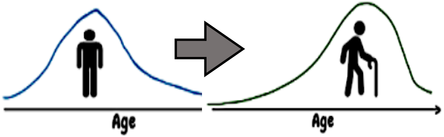
(a) Insufficient Sample Size



(b) Measurement Error



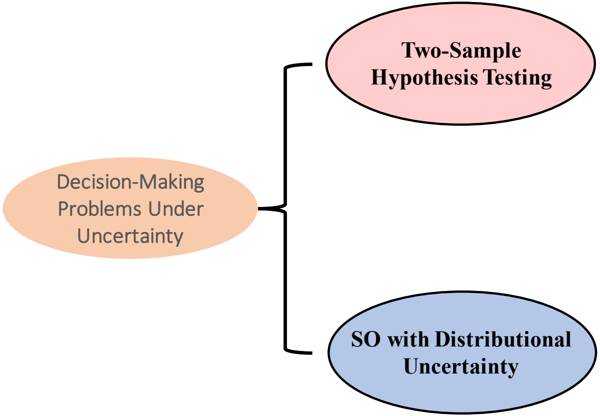
(c) Adversarial Data Perturbation



(d) Model Mis-specification

(e) High Dimensionality

Our Methodology



- **Non-parametric** testing by **nonlinear dimensionality reduction**.
- Computation by **Riemannian optimization algorithms**.
- **Non-asymptotic uncertainty quantification** for proposed testing statistic.

- **Distributionally Robust Optimization** framework with **Sinkhorn Distance**.
- Computationally tractable by **stochastic approximation** with **inexact gradient oracles**.
- **Absolutely Continuous** worst-case distribution expression.

Table of Contents

Problem Setup

- Given two independent sample sets:

$$X = \{x_1, \ldots, x_n\} \sim P, Y = \{y_1, \ldots, y_m\} \sim Q,$$

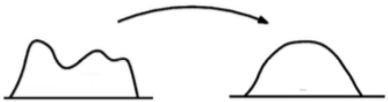
- A test $T: \ (X, Y) \mapsto \{d_0, d_1\}$ decide:

$$\mathcal{H}_0: \ P = Q, \qquad \mathcal{H}_1: P \neq Q.$$

- Risk functions:

$$\text{Type-I Error}: \ \mathbb{P}_{x^n \sim \mu, y^m \sim \nu}\Big(T(x^n, y^m) = d_1\Big), \quad \text{under } \mathcal{H}_0,$$
$$\text{Type-II Error}: \ \mathbb{P}_{x^n \sim \mu, y^m \sim \nu}\Big(T(x^n, y^m) = d_0\Big), \quad \text{under } \mathcal{H}_1.$$

Wasserstein Distance



$$W(\mathbb{P},\mathbb{Q}) := \min_{\gamma\in\mathcal{P}(\Omega^2)} \Big\{ \mathbb{E}_{(\omega,\omega')\sim\gamma}[\|\omega-\omega'\|] : \\ \gamma \text{ has marginal distributions } \mathbb{P} \text{ and } \mathbb{Q} \Big\}.$$

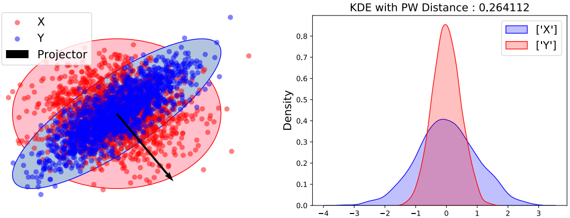
- **Cheapest cost** of transporting probability mass from one distribution to another.
- Advantages:
  - Geometric properties;
  - Flexibility: non-overlapping support, discrete and continuous.

Projected Wasserstein Distance

- Design projected Wasserstein distance for testing[1]:

$$\mathcal{PW}(P, Q) = \max_{\mathcal{A} \in \mathbb{V}_d} W(\mathcal{A}\#P, \mathcal{A}\#Q).$$

- Find linear projector $\mathcal{A} \in \mathbb{V}_d = \{\mathcal{A} : \ \mathcal{A}(z) = A^{\mathrm{T}} z, A^{\mathrm{T}} A = I_d\}$ for which the Wasserstein distance between the projected distributions is as large as possible.



(a) Scatter plots (b) Projected samples

[1] **Jie Wang**, Rui Gao, and Yao Xie. "Two-sample Test using Projected Wasserstein Distance". In: *2021 IEEE International Symposium on Information Theory (ISIT)*. 2021, pp. 3320–3325.

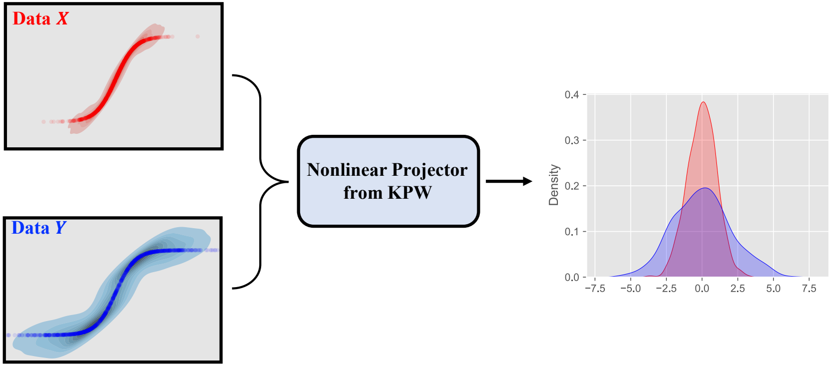
Kernel Projected Wasserstein Distance

- Develop kernel projected Wasserstein distance for testing:

$$KPW(P, Q) = \max_{f \in \mathcal{F}} W\left(f\#P,\ f\#Q\right)$$

$$\text{where} \quad \mathcal{F} = \{f \in \mathcal{H} : \ \|f\|_{\mathcal{H}} \leq 1\}.$$

where $\mathcal{H}$ is a $\mathbb{R}^d$-valued RKHS.



Reproducing Kernel Hilbert Space (RKHS)

Scalar-valued RKHS	Vector-valued RKHS
$K: \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ is P.S.D. if $$\sum_{i,j} y_i K(x_i, x_j) y_j \geq 0,\ \forall x_i \in \mathbb{R}^D, y_i \in \mathbb{R}.$$	$K: \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}^{d \times d}$ is P.S.D. if $$\sum_{i,j} \langle y_i, K(x_i, x_j) y_j \rangle \geq 0,\ \forall x_i \in \mathbb{R}^D, y_i \in \mathbb{R}^d.$$
Reproducing Property: $$f(x) = \langle f, K_x \rangle_{\mathcal{H}_K}, \quad \forall f \in \mathcal{H}_K,$$ where the kernel section $$K_x(x') \triangleq K(x', x), \quad \forall x' \in \mathbb{R}^D.$$	Reproducing Property: $$\langle f(x), y \rangle = \langle f, K_x y \rangle_{\mathcal{H}_K} \quad \forall f \in \mathcal{H}_K,$$ where the kernel section $$(K_x y)(x') \triangleq K(x', x) y, \quad \forall x' \in \mathbb{R}^D, y \in \mathbb{R}^d.$$

KPW Test

- Develop kernel projected Wasserstein distance for testing:

$$KPW(P,Q) = \max_{f \in \mathcal{F}} W\left(f\#P,\ f\#Q\right)$$

$$\text{where} \quad \mathcal{F} = \{f \in \mathcal{H} :\ \|f\|_{\mathcal{H}} \leq 1\}.$$

where $\mathcal{H}$ is a $\mathbb{R}^d$-valued RKHS.

(I) Compute nonlinear projector in training dataset;

(II) Perform permutation test in testing dataset.

- Three-fold Contributions:
  - Computing KPW Distance
  - Finite-sample Guarantee
  - Numerical Simulation

Challenges for Computing KPW Distance

Computing KPW distance is equivalent to:

$$KPW(P_n, Q_m) = \max_{f\in\mathcal{H}:\ \|f\|_{\mathcal{H}}^2 \le 1} \left\{ \min_{\pi\in\Gamma} \sum_{i,j} \pi_{i,j} \|f(x_i) - f(y_j)\|^2 \right\}.$$

- Infinite-dimensional optimization – Develop a representer theorem:

> There exists an optimal solution $\hat{f}$ with
>
> $$\hat{f}(z) = \sum_{i=1}^{n} K(z, x_i) a_{x,i} - \sum_{j=1}^{m} K(z, y_j) a_{y,j}, \quad z \in \mathbb{R}^D,$$
>
> where $a_{x,i}, a_{y,j} \in \mathbb{R}^d$ are coefficients.

- Non-convex problem – Focus on finding stationary point.

Reformulation of KPW Distance (I)

$$KPW(P_n, Q_m) = \max_{f \in \mathcal{H}:\ \|f\|_{\mathcal{H}}^2 \le 1} \left\{ \min_{\pi \in \Gamma} \sum_{i,j} \pi_{i,j} \|f(x_i) - f(y_j)\|^2 \right\}.$$

- Step 1: Substituting the form of representer theorem:

$$\hat{f}(z) = \sum_{i=1}^{n} K(z, x_i) a_{x,i} - \sum_{j=1}^{m} K(z, y_j) a_{y,j}$$

$$= \underbrace{\begin{pmatrix} K(z,x_1) & K(z,x_2) & \cdots & K(z,x_n) & -K(z,y_1) & -K(z,y_2) & \cdots & -K(z,y_m) \end{pmatrix}}_{G(z; x^n, y^m)} \underbrace{\begin{pmatrix} a_{x,1} \\ \vdots \\ a_{x,n} \\ a_{y,1} \\ \vdots \\ a_{y,m} \end{pmatrix}}_{\omega}$$

Reformulation of KPW Distance (II)

- Step 1: Substituting the form of representer theorem:

$$G \triangleq \begin{pmatrix} G(x_1; x^n, y^m) \\ \vdots \\ G(x_n; x^n, y^m) \\ -G(y_1; x^n, y^m) \\ \vdots \\ -G(y_n; x^n, y^m) \end{pmatrix} \qquad \omega \triangleq \begin{pmatrix} a_{x,1} \\ \vdots \\ a_{x,n} \\ a_{y,1} \\ \vdots \\ a_{y,m} \end{pmatrix}$$

$$KPW(P_n, Q_m) = \max_{\omega} \left\{ \min_{\pi \in \Gamma} \sum_{i,j} \pi_{i,j} c_{i,j} : \ \omega^T G \omega \leq 1 \right\}$$

$$c_{i,j} = \|\hat{f}(x_i) - \hat{f}(y_j)\|_2^2$$

Reformulation of KPW Distance (III)

- Step 1: Substituting the form of representer theorem:

$$\max_{\omega} \left\{ \min_{\pi \in \Gamma} \sum_{i,j} \pi_{i,j} c_{i,j} \,:\, \omega^{\mathrm{T}} G \omega \leq 1 \right\}.$$

- Step 2: Take $G^{-1} = UU^{\mathrm{T}}$ and $s = U^{-1}\omega$, we have

$$\max_{s \in \mathbb{S}^{d(n+m)-1}} \left\{ \min_{\pi \in \Gamma} \sum_{i,j} \pi_{i,j} c_{i,j} \right\}, \quad \text{where } \mathbb{S}^{d(n+m)-1} \text{ is a sphere.}$$

- Step 3: Adding entropic regularization and reformulate by duality:

$$\max_{s \in \mathbb{S}^{d(n+m)-1}} \left\{ \min_{\pi \in \Gamma} \sum_{i,j} \pi_{i,j} c_{i,j} - \eta H(\pi) \right\}$$

$$= \min_{u,v,s} \left\{ F(u,v,s) \,:\, s \in \mathbb{S}^{d(n+m)-1}, u \in \mathbb{R}^n, v \in \mathbb{R}^m \right\}.$$

Algorithm: Riemannian Block Coordinate Descent

$$\min_{u,v,s} \left\{ F(u,v,s): \; s \in \mathbb{S}^{d(n+m)-1}, u \in \mathbb{R}^n, v \in \mathbb{R}^m \right\}$$

Develop a Riemannian BCD method:

$$\begin{aligned}
u^{t+1} &= \min_{u \in \mathbb{R}^n} F(u, v^t, s^t), \\
v^{t+1} &= \min_{v \in \mathbb{R}^m} F(u^{t+1}, v, s^t), \\
\zeta^{t+1} &= \nabla_s F(u^{t+1}, v^{t+1}, s^t), \\
\xi^{t+1} &= \mathcal{P}_{s^t}\big(\zeta^{t+1}\big), \\
s^{t+1} &= \mathsf{Retr}_{s^t}\big(-\tau \xi^{t+1}\big),
\end{aligned}$$

where $\mathcal{P}_s(\cdot)$ and $\mathsf{Retr}_s(\cdot)$ denote the projection and retraction on sphere.

Convergence Analysis for Computing KPW Distance

We say that $(\hat{u}, \hat{v}, \hat{s})$ is a (ϵ_1, ϵ_2)-stationary point if

$$\begin{aligned} \|\mathsf{Grad}_s F(\hat{u}, \hat{v}, \hat{s})\| &\le \epsilon_1, \\ F(\hat{u}, \hat{v}, \hat{s}) - \min_{u,v} F(u, v, \hat{s}) &\le \epsilon_2. \end{aligned}$$

Proposed method returns an (ϵ_1, ϵ_2)-stationary point within

- iteration number:

$$\mathcal{O}\left(\log(mn) \cdot \left[\frac{1}{\epsilon_2^3} + \frac{1}{\epsilon_1^2 \epsilon_2}\right]\right),$$

- Arithmetic operations:

$$\mathcal{O}\left(N^3 d^3 \log(N) \cdot \left[\frac{1}{\epsilon_2^3} + \frac{1}{\epsilon_1^2 \epsilon_2}\right]\right),$$

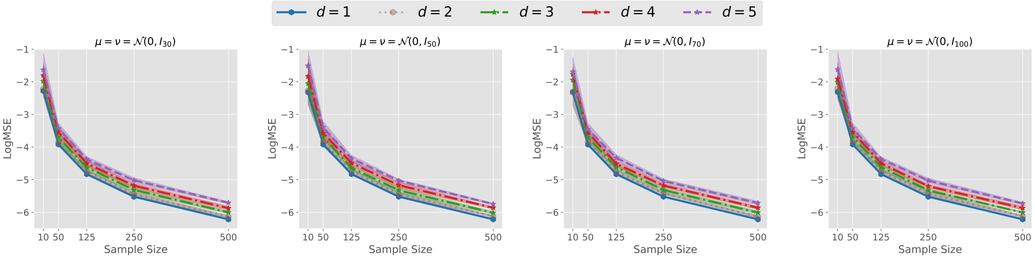
- Storage requirement: $\mathcal{O}(d^2 N^2)$.

Uncertainty Quantification on Testing Statistic

- Cost function $c(x, y) = \|x - y\|_2^p$ with $p \in [1, \infty)$;
- Take $N = n \wedge m$ and $P = Q$, then with high probability,

$$KPW(P_n, Q_m)^{1/p} = \widetilde{O}(N^{-1/(2p)\vee d}).$$

- When taking characteristic kernels, $KPW(P, Q) > 0$ if and only if $P \neq Q$.

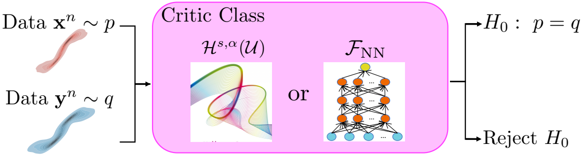


Manifold Two-Sample Test with Neural Networks [2]

- **Assumption:** Data generating distributions p and q supported on $\mathcal{U}$, a d-dimensional manifold embedded in $\mathbb{R}^D$.
- **Testing Statistics:**

$$d_{\mathcal{H}^{s,\beta}(\mathcal{U})}(p,q) = \sup_{f\in\mathcal{H}^{s,\beta}(\mathcal{U})} \mathbb{E}_p[f(\mathbf{x})] - \mathbb{E}_q[f(\mathbf{x})]$$

$$d_{\mathcal{F}_{\mathrm{NN}}(R,\kappa,L,t,K)}(p,q) = \sup_{f\in\mathcal{F}_{\mathrm{NN}}(R,\kappa,L,t,K)} \mathbb{E}_p[f(\mathbf{x})] - \mathbb{E}_q[f(\mathbf{x})].$$



- **Main Result:** Type-II risk of the NN IPM test is of $O(n^{-(s+\beta)/d})$.

[2] **Jie Wang**, Minshuo Chen, Tuo Zhao, Wenjing Liao, and Yao Xie. "A Manifold Two-Sample Test Study: Integral Probability Metric with Neural Networks". In: arXiv preprint arXiv: 2205.02043 (2022).

Numerical Experiments

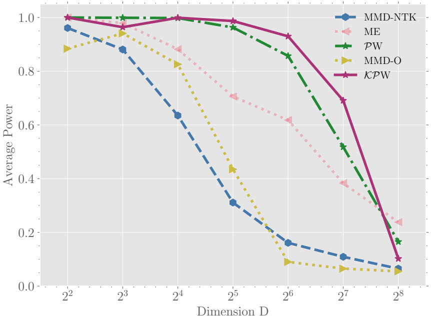
- Design of matrix-valued kernel:

$$K(x, x') = k(x, x') \cdot P,$$
$$P = (1-\rho)\mathbf{1}\mathbf{1}^{\mathrm{T}} + \rho I_d, \quad \text{with } \rho \in [0, 1].$$

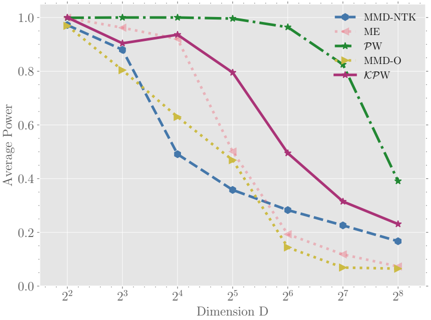
where $k(\cdot, \cdot)$ denotes a scalar-valued kernel function and $P \in \mathbb{S}_{+}^{d\times d}$.

Methodology in Literature	**Advantages**
Projected Wasserstein test (PW) [Wang et al. 2021]	Find linear subspace to separate data
Gaussian MMD test with optimized bandwidth [Gretton et al. 2012]	Powerful non-parametric test
MMD test with neural network (NTK-MMD) [Cheng and Xie, 2021]	Computationally efficient
Mean embedding test (ME) [Jitkrittum et al. 2016]	Powerful non-parametric test

Testing Power for Synthetic Datasets (Gaussian)

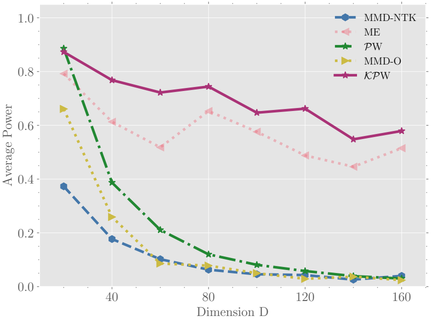


(a) Covariance-shifted (Diagonal) Gaussian

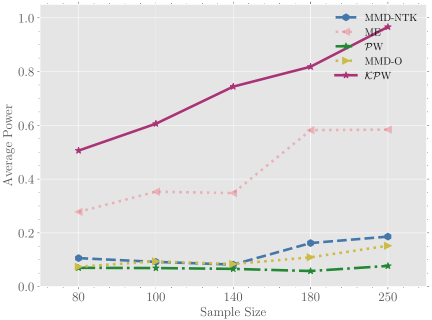


(b) Covariance-shifted (Non-diagonal) Gaussian

Testing Power for Synthetic Datasets (Gaussian Mixture)



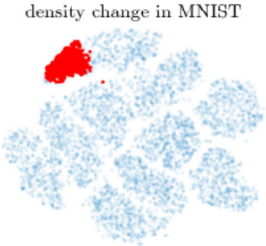
(a) Power v.s. Dimension



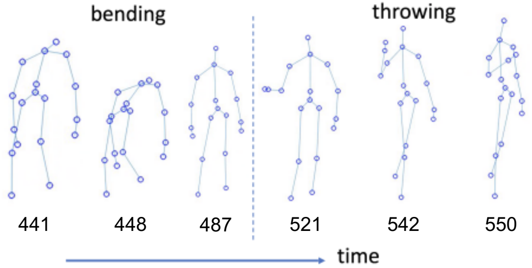
(b) Power v.s. Sample Size

Test for MNIST Digits

N	MMD-NTK	MMD-O	ME	PW	KPW
200	0.639±0.029	**0.696±0.006**	0.298±0.031	0.302±0.033	0.663±0.015
250	0.763±0.010	0.781±0.002	0.472±0.017	0.369±0.030	**0.785±0.014**
300	0.813±0.016	0.869±0.002	0.630±0.025	0.524±0.023	**0.928±0.001**
400	0.881±0.013	0.956±0.003	0.779±0.020	0.591±0.044	**0.978±0.000**
500	0.950±0.002	0.988±0.000	0.927±0.006	0.782±0.040	**1.000±0.000**
Avg.	0.809	0.858	0.621	0.513	**0.870**



Online Human Activity Detection



User	MMD-NTK	MMD-O	ME	PW	KPW
1	36	73	82	47	**33**
2	8	7	97	9	**1**
3	15	13	27	**2**	20
4	22	83	69	16	**12**
Mean	20.25	44.0	68.8	18.50	**16.5**
Std	**12.0**	39.5	30.1	19.8	13.5

This Part is Based on ...

[1] **Jie Wang**, Rui Gao, and Yao Xie. “Two-Sample Test with Kernel Projected Wasserstein Distance”. In: *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*. Vol. 151. Accepted as Oral Presentation (acceptance rate 2.6%). 2022, pp. 8022–8055

[2] **Jie Wang**, Rui Gao, and Yao Xie. “Two-sample Test using Projected Wasserstein Distance”. In: *2021 IEEE International Symposium on Information Theory (ISIT)*. 2021, pp. 3320–3325

[3] **Jie Wang**, Minshuo Chen, Tuo Zhao, Wenjing Liao, and Yao Xie. “A Manifold Two-Sample Test Study: Integral Probability Metric with Neural Networks”. In: *arXiv preprint arXiv:2205.02043* (2022). Submitted to Information and Inference: a Journal of the IMA

Table of Contents

Decision-Making Under Uncertainty

Risk : $\mathcal{R}(\theta;\mathbb{P}) = \mathbb{E}_{\mathbb{P}}[f_\theta(z)]$

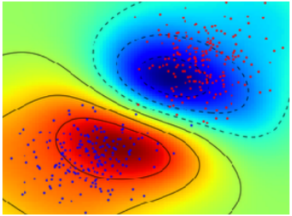
Optimal Risk : $\mathcal{R}(\Theta;\mathbb{P}) = \inf_{\theta\in\Theta} \mathbb{E}_{\mathbb{P}}[f_\theta(z)]$

DRO Model : $\mathcal{R}_{\mathsf{DRO}}(\Theta;\mathcal{P}) = \inf_{\theta\in\Theta}\sup_{\mathbb{P}\in\mathcal{P}} \mathbb{E}_{\mathbb{P}}[f_\theta(z)]$

Applications

Supply Chain Mgmt.

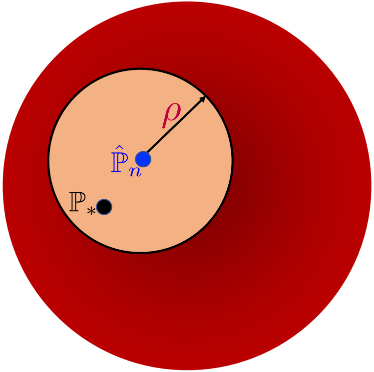
Portfolio Mgmt.



Machine Learning

Wasserstein DRO

Definition: $\mathcal{P} = \{\mathbb{P} : \ W(\mathbb{P}, \hat{\mathbb{P}}_n) \leq \rho\}$.



Contain each $\mathbb{P}$ such that $W(\mathbb{P}, \hat{\mathbb{P}}_n) \leq \rho$

Worst-case risk : $\sup_{\mathbb{P} \in \mathcal{P}} \mathbb{E}_{\mathbb{P}}[f_\theta(z)]$

Robust Optimal Risk : $\inf_{\theta \in \Theta} \sup_{\mathbb{P} \in \mathcal{P}} \mathbb{E}_{\mathbb{P}}[f_\theta(z)]$

Limitations of Wasserstein DRO

- Worst-case distribution is discrete:

  For WDRO with n-point nominal distribution, the worst-case distribution is supported on $n+1$ points [Gao and Kleywegt 2016].

- Some cases the same performance as SAA.

  E.g., 1-Wasserstein DRO formulation for newsvendor problem [Esfahani and Kuhn, 2017].

- Tractability only for limited scenarios [Esfahani and Kuhn 2017, Sinha et al. 2018, Jose et al. 2022].

Tractability of Wasserstein DRO

$$\inf_{\theta\in\Theta}\left\{\sup_{\mathbb{P}:\ W(\mathbb{P},\widehat{\mathbb{P}})\le\rho}\mathbb{E}_{z\sim\mathbb{P}}[f_\theta(z)]\right\}$$

$$=\inf_{\theta\in\Theta,\lambda\ge 0}\left\{\lambda\rho+\int\sup_{z\in\mathcal{Z}}\left[f_\theta(z)-\lambda c(x,z)\right]\,\mathrm{d}\widehat{\mathbb{P}}(x)\right\}$$

Reference(s)	Loss function $f_\theta(z)$	Cost function	Nominal distribution $\widehat{\mathbb{P}}$	Support $\mathcal{Z}$
[Zhang et al. 2021]	General	General	General	Discrete and finite set
[Sinha et al. 2018]	$z\mapsto f_\theta(z)-\lambda^* c(x,z)$ is strongly concave[i]	General	General	General
[Esfahani and Kuhn, 2017]	Piecewise concave in z	Norm function	Empirical distribution	Polytope
[Shafieezadeh-Abadeh et al. 2015]	Generalized linear model in (z,θ)	Norm function	Empirical distribution	Whole Euclidean space[ii]
[Jose et al. 2022]	Generalized linear model in (z,θ)	Squared norm function	General	Whole Euclidean space[ii]

Tractability of Wasserstein DRO

$$\inf_{\theta\in\Theta} \left\{ \sup_{\mathbb{P}:\ W(\mathbb{P},\widehat{\mathbb{P}})\le\rho} \mathbb{E}_{z\sim\mathbb{P}}[f_\theta(z)] \right\}$$

$$= \inf_{\theta\in\Theta,\lambda\ge 0} \left\{ \lambda\rho + \int \sup_{z\in\mathcal{Z}} \left[f_\theta(z) - \lambda c(x,z) \right] \mathrm{d}\widehat{\mathbb{P}}(x) \right\}$$

- Example on the hardness of Wasserstein DRO:

Multi-class Classification: $f_B(z) = -\vec{y}^{\mathrm{T}} B^{\mathrm{T}} x + \log\left(1^{\mathrm{T}} e^{B^{\mathrm{T}} x}\right), \quad z = (x, \vec{y}).$

  - x: feature vector in the ball $\{x:\ \|x\|_\infty \le 1\}$.
  - $\vec{y}$: one-hot label vector in $\{0,1\}^C$.
  - randomness only on feature vector.

Remark: Inner supremum problem is a high-dimensional maximization over non-concave function.

Numerical Simulation Results

Multi-class classification:

$$\min_{B} \max_{\mathbb{P} \in \mathbb{B}_{\rho,\epsilon}(\widehat{\mathbb{P}})} \mathbb{E}_{(x,\vec{y}) \sim \mathbb{P}} \left[-\vec{y}^{\mathrm{T}} B^{\mathrm{T}} x + \log \left(1^{\mathrm{T}} e^{B^{\mathrm{T}} x} \right) \right],$$

where the probability support for feature x is a bounded polytope.

Mis-classification rate:

Dataset	SAA	KL-DRO	1-WDRO	1-SDRO	2-WDRO	2-SDRO
MNIST	$.075 \pm .002$	$.067 \pm .002$	$.037 \pm .003$	$\mathbf{.035 \pm .002}$	$.047 \pm .003$	$\mathbf{.041 \pm .002}$
IRIS	$.396 \pm .024$	$.351 \pm .015$	$.321 \pm .021$	$\mathbf{.308 \pm .021}$	$.378 \pm .023$	$\mathbf{.342 \pm .022}$
wine	$.089 \pm .010$	$.086 \pm .010$	$.082 \pm .005$	$\mathbf{.077 \pm .005}$	$.078 \pm .005$	$\mathbf{.076 \pm .005}$
vowel	$.481 \pm .012$	$.478 \pm .011$	$\boxed{.492 \pm .012}$	$\mathbf{.456 \pm .011}$	$.476 \pm .012$	$\mathbf{.443 \pm .012}$
vehicle	$.379 \pm .007$	$.368 \pm .007$	$\boxed{.481 \pm .014}$	$\mathbf{.343 \pm .006}$	$\boxed{.434 \pm .009}$	$\mathbf{.349 \pm .007}$
svmguide4	$.427 \pm .009$	$.418 \pm .009$	$\boxed{.430 \pm .010}$	$\mathbf{.417 \pm .009}$	$.425 \pm .009$	$\mathbf{.393 \pm .011}$

Sinkhorn Distance

- Sinkhorn Distance:

$$W_\epsilon(\mathbb{P},\mathbb{Q}) = \inf_{\gamma\in\Gamma(\mathbb{P},\mathbb{Q})} \left\{\mathbb{E}_{(X,Y)\sim\gamma}[c(X,Y)] + \epsilon H(\gamma \mid \mathbb{P}\otimes\nu)\right\}.$$

  Remark: Sinkhorn distance does not satisfy definition of “distance function”.

- Relative Entropy between γ and $\mathbb{P}\otimes\nu$:

$$H(\gamma \mid \mathbb{P}\otimes\nu) = \int \log\left(\frac{\mathrm{d}\gamma(x,y)}{\mathrm{d}\mathbb{P}(x)\,\mathrm{d}\nu(y)}\right)\mathrm{d}\gamma(x,y).$$

Historical Review:

- Originally proposed by [Wilson’ 62].
- Convergence of algorithm for the first time by [Sinkhorn’64].
- Operation complexity analysis and GPGPU parallel by [Cuturi’13].

Sinkhorn Distance

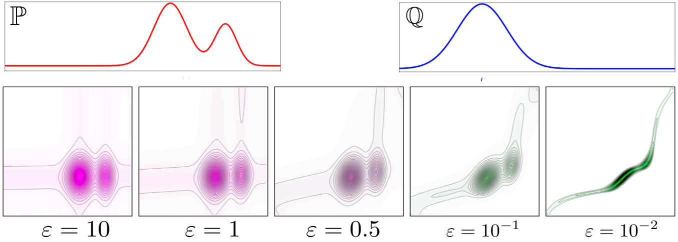
- Sinkhorn Distance:

$$W_\epsilon(\mathbb{P},\mathbb{Q}) = \inf_{\gamma\in\Gamma(\mathbb{P},\mathbb{Q})} \left\{\mathbb{E}_{(X,Y)\sim\gamma}[c(X,Y)] + \epsilon H(\gamma \mid \mathbb{P}\otimes\nu)\right\}.$$

Remark: Sinkhorn distance does not satisfy definition of "distance function".

- Relative Entropy between γ and $\mathbb{P}\otimes\nu$:

$$H(\gamma \mid \mathbb{P}\otimes\nu) = \int \log\left(\frac{\mathrm{d}\gamma(x,y)}{\mathrm{d}\mathbb{P}(x)\,\mathrm{d}\nu(y)}\right)\mathrm{d}\gamma(x,y).$$



Highlights of Sinkhorn Distance

Probability distance between distributions in $\mathbb{R}^d$ using n samples:

	MMD	Wasserstein	Sinkhorn
Computation	$O(n)$	$\tilde{O}(n^3)$	$\tilde{O}(n^2)$ [Altschuler, Niles-Weed, and Rigollet 2017]
Sample Complexity	$O(n^{-1/2})$	$O(n^{-1/d})$	$O(e^{\kappa/\epsilon} n^{-1/2} \epsilon^{-\lfloor d/2 \rfloor})$[3][Genevay et al. 2019]

- Fast algorithms for implementation;
- Sharp sample complexity rate;
- Encourage stochastic optimal transport (helpful in some applications, e.g., domain adaptation [Courty, Flamary, and Tuia 2014]).

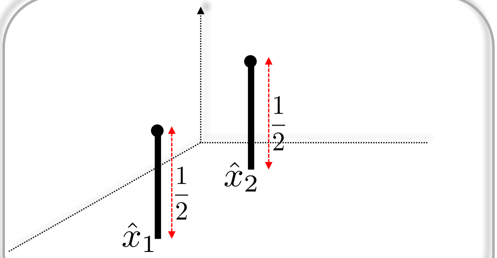
[3] κ is a smoothness parameter of the data distribution.

Main Framework

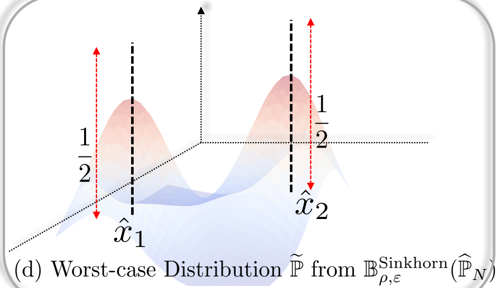
- Sinkhorn DRO:

$$\inf_{\theta} \sup_{\mathbb{P}\in\mathbb{B}_{\rho,\epsilon}(\widehat{\mathbb{P}})} \mathbb{E}_{z\sim\mathbb{P}}[f_\theta(z)],$$

where $\mathbb{B}_{\rho,\epsilon}(\widehat{\mathbb{P}}) = \left\{\mathbb{P}: \ W_\epsilon(\widehat{\mathbb{P}},\mathbb{P}) \le \rho\right\}.$



(a) Empirical Distribution $\widehat{\mathbb{P}}_N$



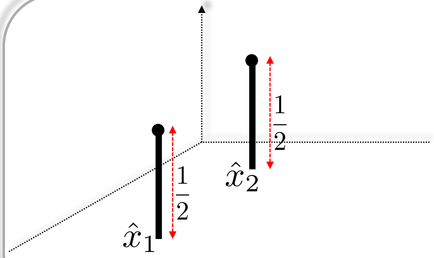
(d) Worst-case Distribution $\widetilde{\mathbb{P}}$ from $\mathbb{B}_{\rho,\varepsilon}^{\text{Sinkhorn}}(\widehat{\mathbb{P}}_N)$.

General DRO Models

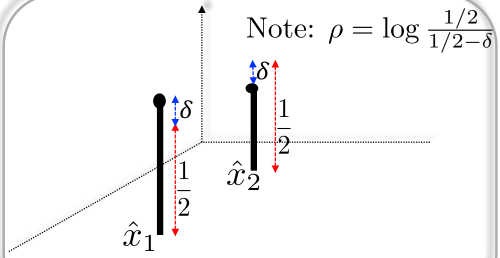
- KL-DRO:

$$\inf_{\theta} \sup_{\mathbb{P} \in \mathbb{B}_{\rho}^{\mathsf{KL}}(\widehat{\mathbb{P}})} \mathbb{E}_{z \sim \mathbb{P}}[f_\theta(z)],$$

$$\text{where } \mathbb{B}_{\rho}^{\mathsf{KL}}(\widehat{\mathbb{P}}) = \{\mathbb{P} : \ D_{\mathsf{KL}}(\widehat{\mathbb{P}}, \mathbb{P}) \le \rho\}.$$



(a) Empirical Distribution $\widehat{\mathbb{P}}_N$



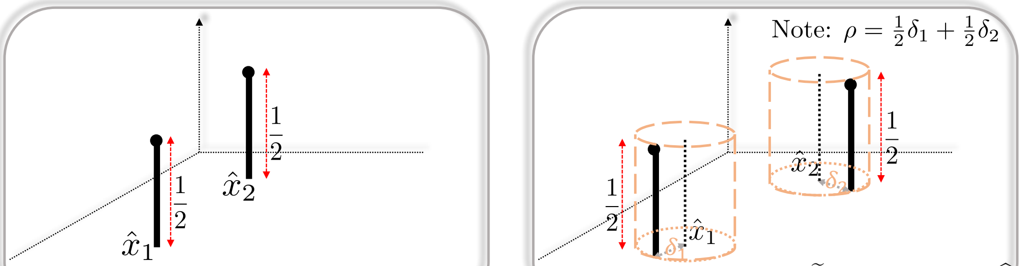
(b) Worst-case Distribution $\widetilde{\mathbb{P}}$ from $\mathbb{B}_{\rho}^{\mathrm{KL}}(\widehat{\mathbb{P}}_N)$.

General DRO Models

- Wasserstein-DRO:

$$\inf_{\theta} \sup_{\mathbb{P} \in \mathbb{B}_{\rho}^{\text{Wasserstein}}(\widehat{\mathbb{P}})} \mathbb{E}_{z \sim \mathbb{P}}[f_{\theta}(z)],$$

where $\mathbb{B}_{\rho}^{\text{Wasserstein}}(\widehat{\mathbb{P}}) = \{\mathbb{P} : \ W(\widehat{\mathbb{P}}, \mathbb{P}) \leq \rho\}$.



(a) Empirical Distribution $\widehat{\mathbb{P}}_N$

(c) Worst-case Distribution $\widetilde{\mathbb{P}}$ from $\mathbb{B}_{\rho}^{\text{Wasserstein}}(\widehat{\mathbb{P}}_N)$.

Ongoing Outline

- Sinkhorn DRO:

$$\inf_{\theta} \sup_{\mathbb{P}\in\mathbb{B}_{\rho,\epsilon}(\widehat{\mathbb{P}})} \mathbb{E}_{z\sim\mathbb{P}}[f_\theta(z)],$$

where $\mathbb{B}_{\rho,\epsilon}(\widehat{\mathbb{P}}) = \Big\{\mathbb{P}:\ W_\epsilon(\widehat{\mathbb{P}},\mathbb{P}) \le \rho\Big\}.$

  - Duality Formulation for Sinkhorn DRO
  - First-order Optimization Algorithm
  - Properties and Numerical Results

Tractable Formulation

Under mild conditions, the primal

$$V_{\mathrm{P}} = \sup_{\mathbb{P}\in\mathbb{B}_{\rho,\epsilon}(\widehat{\mathbb{P}})} \mathbb{E}_{z\sim\mathbb{P}}[f(z)], \quad \text{where } \mathbb{B}_{\rho,\epsilon}(\widehat{\mathbb{P}}) = \{\mathbb{P} : \ W_\epsilon(\widehat{\mathbb{P}}, \mathbb{P}) \le \rho\}.$$

(Sinkhorn DRO)

has the **strong dual reformulation**:

$$V_{\mathrm{D}} = \inf_{\lambda>0} \ \lambda\overline{\rho} + \lambda\epsilon \int_\Omega \log\left(\mathbb{E}_{\mathbb{Q}_x}\left[e^{f(z)/(\lambda\epsilon)}\right]\right) \mathrm{d}\widehat{\mathbb{P}}(x),$$

where

$$\overline{\rho} = \rho + \epsilon \int_\Omega \log\left(\int_\Omega e^{-c(x,z)/\epsilon}\,\mathrm{d}\nu(z)\right) \mathrm{d}\widehat{\mathbb{P}}(x),$$

$$\mathrm{d}\mathbb{Q}_x(z) = \frac{e^{-c(x,z)/\epsilon}}{\int_\Omega e^{-c(x,u)/\epsilon}\,\mathrm{d}\nu(u)}\,\mathrm{d}\nu(z).$$

Interpretation of Worst-case Distribution

$$\tilde{\mathbb{P}} = \arg\max_{\mathbb{P}} \ \left\{ \mathbb{E}_{z\sim\mathbb{P}}[f(z)] : \ W_\epsilon(\widehat{\mathbb{P}}, \mathbb{P}) \le \rho \right\}$$

- For each $x \in \mathsf{supp}(\widehat{\mathbb{P}})$, optimal transport maps it to a (conditional) distribution γ_x such that

$$\frac{\mathrm{d}\gamma_x(z)}{\mathrm{d}\nu(z)} = \alpha_x \cdot \exp\left(\big(f(z) - \lambda^* c(x,z)\big)/(\lambda^*\epsilon) \right),$$

  where α_x is the normalizing constant.
- Closed-form expression on $\tilde{\mathbb{P}}$:

$$\frac{\mathrm{d}\tilde{\mathbb{P}}(z)}{\mathrm{d}\nu(z)} = \int \alpha_x \cdot \exp\left(\big(f(z) - \lambda^* c(x,z)\big)/(\lambda^*\epsilon) \right) \mathrm{d}\widehat{\mathbb{P}}(x).$$

Worst-case distribution $\tilde{\mathbb{P}}$ support on whole space, while W-DRO is discrete.

Toy Example: Newsvendor

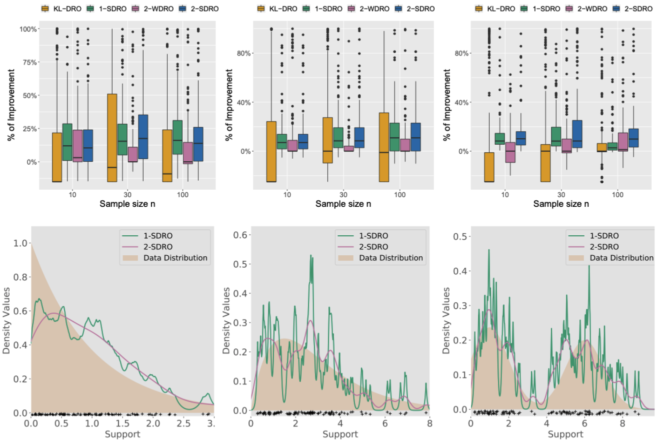
Newsvendor problem: (β: Demand); ($u \min\{\beta, z\}$: Earning); ($k\beta$: Loss).

$$\min_{\beta} \ \mathbb{E}_{\mathbb{P}_*}\left[k\beta - u\min\{\beta, z\}\right], \quad k = 5, u = 7.$$

Performance and Visualization

Newsvendor problem:

$$\min_{\beta} \; \mathbb{E}_{\mathbb{P}_*}\left[k\beta - u\min\{\beta,\zeta\}\right], \quad k=5, u=7.$$



Connection of Sinkhorn DRO with Wasserstein DRO

When $\epsilon \to 0$, the dual objective of Sinkhorn DRO converges into

$$\lambda\rho + \int \text{ess-sup}_{\nu} \left(f(\cdot) - \lambda c(x,\cdot)\right) \mathrm{d}\widehat{\mathbb{P}}(x).$$

When supp$(\nu) = \Omega$,

Sinkhorn DRO

$$\sup_{\mathbb{P}} \quad \mathbb{E}_{z\sim\mathbb{P}}[f(z)] \qquad \text{s.t.} \quad W_\epsilon(\hat{\mathbb{P}}, \mathbb{P}) \le \rho$$

$\xrightarrow{\text{Take } \epsilon \to 0}$

Wasserstein DRO

$$\sup_{\mathbb{P}} \quad \mathbb{E}_{z\sim\mathbb{P}}[f(z)] \qquad \text{s.t.} \quad W(\hat{\mathbb{P}}, \mathbb{P}) \le \rho$$

Connection of Sinkhorn DRO with KL DRO

Upper bound of Sinkhorn DRO:

$$V_{\mathrm{D}} \triangleq \inf_{\lambda>0} \ \lambda\overline{\rho} + \lambda\epsilon \int_{\Omega} \log\left(\mathbb{E}_{\mathbb{Q}_x}\left[e^{f(y)/(\lambda\epsilon)}\right]\right) \mathrm{d}\widehat{\mathbb{P}}(x)$$
$$\leq \inf_{\lambda>0} \ \lambda\overline{\rho} + \lambda\epsilon \log\left(\mathbb{E}_{\mathbb{P}^0}\left[e^{f(y)/(\lambda\epsilon)}\right]\right)$$

$\mathbb{P}^0$: kernel density estimate based on $\widehat{\mathbb{P}}$:

$$\mathrm{d}\mathbb{P}^0(z) = \int_x \mathrm{d}\mathbb{Q}_x(z)\,\mathrm{d}\widehat{\mathbb{P}}(x).$$

Sinkhorn DRO

$$\sup_{\mathbb{P}} \ \mathbb{E}_{z\sim\mathbb{P}}[f(z)] \quad \text{s.t.} \quad W_\epsilon(\hat{\mathbb{P}}, \mathbb{P}) \leq \rho$$

$\xrightarrow{\text{Take } \mathbb{P}^0 \text{ as the KDE estimate of } \hat{\mathbb{P}}}$

KL DRO

$$\sup_{\mathbb{P}} \ \mathbb{E}_{z\sim\mathbb{P}}[f(z)] \quad \text{s.t.} \quad D_{\mathrm{KL}}(\mathbb{P}\|\mathbb{P}^0) \leq \bar{\rho}/\epsilon$$

Connection of Sinkhorn DRO with SAA

When $\overline{\rho} = 0$, Sinkhorn becomes SAA:

$$V_{\mathrm{P}} = \mathbb{E}_{z\sim\mathbb{P}^0}[f(z)]$$

$\mathbb{P}^0$: kernel density estimate based on $\widehat{\mathbb{P}}$:

$$\mathrm{d}\mathbb{P}^0(z) = \int_x \mathrm{d}\mathbb{Q}_x(z)\,\mathrm{d}\widehat{\mathbb{P}}(x).$$

Sinkhorn DRO

$$\sup_{\mathbb{P}} \quad \mathbb{E}_{z\sim\mathbb{P}}[f(z)]$$
$$\text{s.t.} \quad W_\epsilon(\hat{\mathbb{P}}, \mathbb{P}) \le \rho$$

$\xrightarrow{\text{Take } \bar{\rho} = 0}$

SAA

$$\mathbb{E}_{z\sim\mathbb{P}^0}[f(z)]$$

Choice of Hyper-parameters $(\epsilon, \overline{\rho})$

Sinkhorn DRO

$$\min_{\theta} \sup_{\mathbb{P}} \quad \mathbb{E}_{z\sim\mathbb{P}}[f_\theta(z)] \quad \text{s.t.} \quad \mathcal{W}_\epsilon(\hat{\mathbb{P}}, \mathbb{P}) \le \rho$$

$$\xrightarrow{\text{Take } \bar{\rho} = 0}$$

SAA

$$\min_{\theta} \quad \int \mathbb{E}_{z\sim\mathbb{Q}_x}[f_\theta(z)]d\hat{\mathbb{P}}(x)$$

- First choose ϵ to optimize the hold-out performance for

$$\operatorname{argmin}_\theta \int \mathbb{E}_{z\sim\mathbb{Q}_x}[f(z)]\,\mathrm{d}\widehat{\mathbb{P}}(x), \quad \mathrm{d}\mathbb{Q}_x(z) \propto e^{-c(\hat{x}_i, z)/\epsilon}\,\mathrm{d}\nu(z).$$

- For fixed ϵ, choose $\overline{\rho}$ to optimize the hold-out performance for

$$\operatorname{argmin}_\theta \sup_{\mathbb{P}} \left\{ \mathbb{E}_{z\sim\mathbb{P}}[f_\theta(z)] : \quad W_\epsilon(\widehat{\mathbb{P}}, \mathbb{P}) \le \rho \right\}.$$

Optimization Algorithm for Sinkhorn DRO

- Based on strong duality,

$$\min_{\theta\in\Theta}\sup_{\mathbb{P}}\left\{\mathbb{E}_{z\sim\mathbb{P}}[f_\theta(z)]:\quad W_\epsilon(\widehat{\mathbb{P}},\mathbb{P})\le\rho\right\}$$

$$=\min_{\lambda\ge 0}\left\{\lambda\overline{\rho}+\underbrace{\min_{\theta\in\Theta}\ \mathbb{E}_{x\sim\widehat{\mathbb{P}}}\left[\lambda\epsilon\log\left(\mathbb{E}_{z\sim\mathbb{Q}_x}\left[e^{f_\theta(z)/(\lambda\epsilon)}\right]\right)\right]}_{V(\lambda)}\right\}$$

- Solve the Monte-Carlo approximated formulation [Shapiro, Dentcheva, and RuszczyÅĎski 2014]:

$$V(\lambda)\approx\min_{\theta\in\Theta}\ \frac{1}{n}\sum_{i=1}^{n}\lambda\epsilon\log\left(\frac{1}{m}\sum_{j=1}^{m}e^{f_\theta(z_{i,j})/(\lambda\epsilon)}\right),$$

where $\{\hat{x}_i\}_{i=1}^n\sim\widehat{\mathbb{P}}$ and $\{z_{i,j}\}_{j=1}^m$ are i.i.d. samples generated from $\mathbb{Q}_{\hat{x}_i}$.

- **Cons:** It requires $\tilde{O}(\delta^{-3})$ samples to obtain δ-optimal solution.

Optimization Algorithm for Sinkhorn DRO

- Based on strong duality,

$$\min_{\theta\in\Theta}\sup_{\mathbb{P}} \left\{\mathbb{E}_{z\sim\mathbb{P}}[f_\theta(z)] : \quad W_\epsilon(\widehat{\mathbb{P}},\mathbb{P}) \le \rho\right\}$$
$$= \min_{\lambda\ge 0} \left\{ \lambda\overline{\rho} + \underbrace{\min_{\theta\in\Theta}\ \mathbb{E}_{x\sim\widehat{\mathbb{P}}}\left[\lambda\epsilon\log\left(\mathbb{E}_{z\sim\mathbb{Q}_x}\left[e^{f_\theta(z)/(\lambda\epsilon)}\right]\right)\right]}_{V(\lambda)} \right\}$$

- Solve the Monte-Carlo approximated formulation [Shapiro, Dentcheva, and RuszczyÅĎski 2014]:

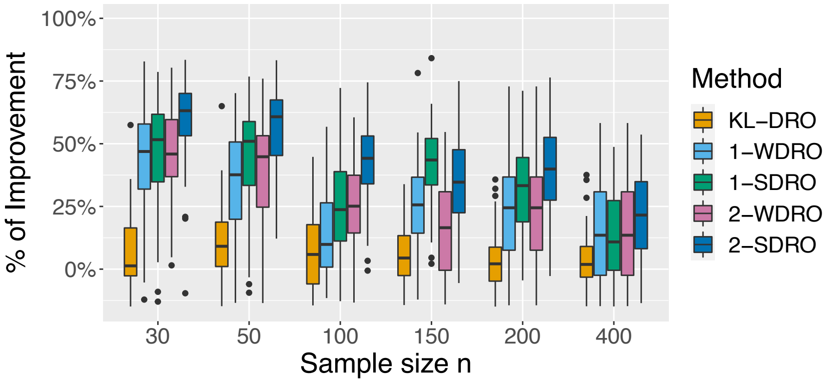
$$V(\lambda) \approx \min_{\theta\in\Theta} \frac{1}{n}\sum_{i=1}^{n}\lambda\epsilon\log\left(\frac{1}{m}\sum_{j=1}^{m} e^{f_\theta(z_{i,j})/(\lambda\epsilon)}\right),$$

where $\{\hat{x}_i\}_{i=1}^n \sim \widehat{\mathbb{P}}$ and $\{z_{i,j}\}_{j=1}^m$ are i.i.d. samples generated from $\mathbb{Q}_{\hat{x}_i}$.
- **Cons:** It requires $\tilde{O}(\delta^{-3})$ samples to obtain δ-optimal solution.

Numerical Results

Portfolio Optimization:

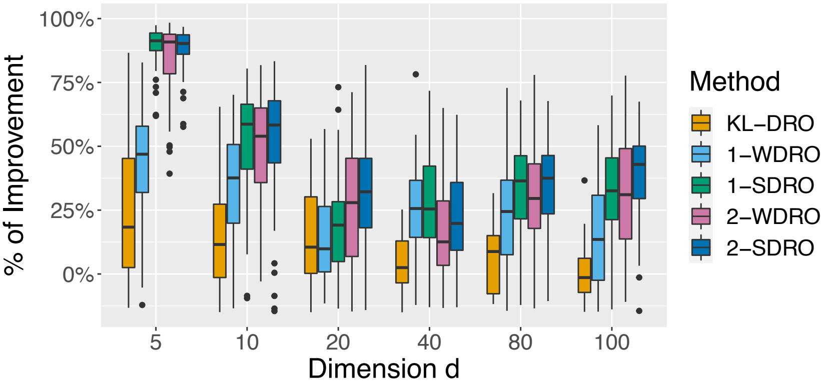
$$\begin{aligned} \inf_{x} \quad & \mathbb{E}_{\mathbb{P}_*}\left[-\langle x, \zeta\rangle\right] + \varrho \cdot \mathbb{P}_*\text{-CVaR}_\alpha(-\langle x, \zeta\rangle) \\ \text{s.t.} \quad & x \in \mathcal{X} = \{x \in \mathbb{R}_+^D : \; x^{\mathrm{T}} 1 = 1\}. \end{aligned}$$



Numerical Results

Portfolio Optimization:

$$\begin{aligned} \inf_{x} \quad & \mathbb{E}_{\mathbb{P}_*}\left[-\langle x, \zeta\rangle\right] + \varrho \cdot \mathbb{P}_*\text{-CVaR}_{\alpha}(-\langle x, \zeta\rangle) \\ \text{s.t.} \quad & x \in \mathcal{X} = \{x \in \mathbb{R}_+^D : \ x^{\mathrm{T}} 1 = 1\}. \end{aligned}$$

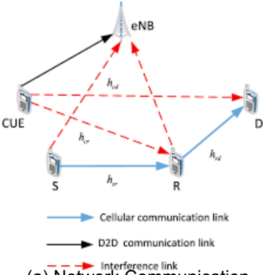


Take Home Message

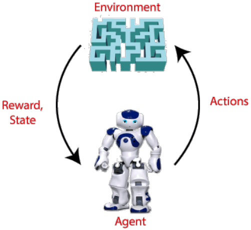
Sinkhorn DRO is a great notion of DRO models:

- Inherit geometric properties from optimal transport;
- Absolutely continuous worst-case distribution thanks to entropic regularization;
- Improve the out-of-sample performance of Wasserstein DRO;
- Optimization by **Monte Carlo approximation** and **first order method**;
- **Sample-size independent** complexity rate with mild assumptions.

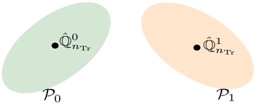
Applications of DRO



(a) Network Communication
[Jie et al. ISIT-21]

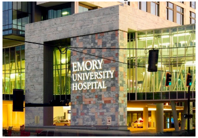


(b) Reinforcement Learning
[Jie et al. Operations Research]



$$\inf_{\text{Detector } T} \sup_{\mathbb{P}_k \in \mathcal{P}_k} \{\text{Type-I Error} + \text{Type-II Error}\}$$

(c) Hypothesis Testing
[Jie et al. ISIT-22]



Source (with Labels)

Target (No Labels)

(d) Domain Adapation for Health Care
[Jie et al. ML4H-22]

This Part is Based on ...

[1] **Jie Wang**, Rui Gao, and Yao Xie. "Sinkhorn Distributionally Robust Optimization". In: *arXiv preprint arXiv:2109.11926* (2022). To be submitted to Operations Research

- Winner of 2022 INFORMS Best Poster Award;
- Winner of Best Student Poster Award (Honorable Mention) at Georgia Statistics Day 2022.

[2] **Jie Wang** and Zhiyuan Jia and Hoover H. F. Yin and Shenghao Yang. "Small-sample inferred adaptive recoding for batched network coding". In: *2021 IEEE International Symposium on Information Theory (ISIT)*. 2021, pp. 1427–1432

[3] **Jie Wang**, Rui Gao, and Hongyuan Zha. "Reliable off-policy evaluation for reinforcement learning". In: *Operations Research* (2022)

[4] **Jie Wang** and Yao Xie. "A Data-Driven Approach to Robust Hypothesis Testing Using Sinkhorn Uncertainty Sets". In: *2022 IEEE International Symposium on Information Theory (ISIT)*. 2022, pp. 3315–3320

[5] **Jie Wang** and Ronald Moore and Yao Xie and Rishikesan Kamaleswaran. "Improving Sepsis Prediction Model Generalization With Optimal Transport". In: *Machine Learning for Health*. 2022, pp. 474–488

Table of Contents

Kernel Two-Sample Testing

- Maximum Mean Discrepancy:

$$\mathsf{MMD}(\mu, \nu; K) \triangleq \sup_{f \in \mathcal{H}_K, \|f\|_{\mathcal{H}_K} \leq 1} \left\{ \mathbb{E}_\mu[f] - \mathbb{E}_\nu[f] \right\}.$$

  Sample-based reformulation:

$$\mathrm{MMD}^2(\mu, \nu; K) \quad = \quad \mathbb{E}_{x,x' \sim \mu}[K(x, x')] \; + \; \mathbb{E}_{y,y' \sim \nu}[K(y, y')] \; - \; \mathbb{E}_{x \sim \mu, y \sim \nu}[K(x, y)].$$

- Data-driven estimate:

$$S^2(\mathbf{x}^n, \mathbf{y}^m; K) \triangleq \frac{1}{n^2} \sum_{i,j \in [n]} K_{i,j}^{x,x} + \frac{1}{m^2} \sum_{i,j \in [m]} K_{i,j}^{y,y} - \frac{2}{mn} \sum_{i \in [n], j \in [m]} K_{i,j}^{x,y}.$$

Statistical Analysis of DRO

- How to optimally choose the Uncertainty size and entropic regularization value:

$$\inf_{\theta} \sup_{\mathbb{P}\in\mathbb{B}_{\rho,\epsilon}(\widehat{\mathbb{P}})} \mathbb{E}_{z\sim\mathbb{P}}[f_\theta(z)],$$

$$\text{where } \mathbb{B}_{\rho,\epsilon}(\widehat{\mathbb{P}}) = \{\mathbb{P}: \ W_\epsilon(\widehat{\mathbb{P}},\mathbb{P}) \le \rho\}.$$

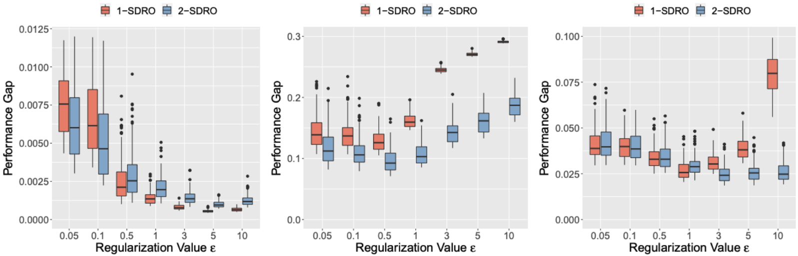
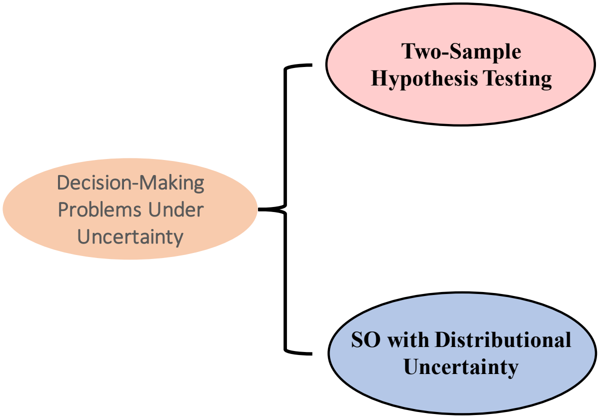


Figure EC.3 Performance of Sinkhorn DRO models for portfolio problem versus different choices of regularization values ϵ. For those fix figures from left to right, from top to bottom, we specify the problem parameters (sample size n and data dimension d) as $(30,30), (100,30), (400,30), (100,5), (100,20), (100,100)$, respectively.

Conclusion



- **Non-parametric** testing by **nonlinear dimensionality reduction**.
- Computation by **Riemannian optimization algorithms**.
- **Non-asymptotic uncertainty quantification** for proposed testing statistic.
- Testing by **feature selection**.
- **Manifold** Two-sample Testing.

- **Distributionally Robust Optimization** framework with **Sinkhorn Distance**;
- Computationally tractable by **stochastic approximation** with **inexact gradient oracles**;
- **Absolutely Continuous** worst-case distribution expression.
- **Statistical analysis**?

Table of Contents

SMD for Stochastic Optimization

- Consider S-smooth and convex optimization problem

$$\begin{array}{ll} \text{Minimize} & \mathbb{E}[f_\theta(z)] \\ \text{s.t.} & \theta \in \Theta \subseteq \mathbb{R}^d. \end{array}$$

- Stochastic Mirror Descent: iteratively,
  - Step 1: generate unbiased gradient estimator $G(\theta_t, \xi^t)$ with $\mathbb{V}\mathrm{ar}\big[G(\theta, \xi)\big] \leq \sigma^2$
  - Step 2: perform

$$\theta_{t+1} = \textbf{Proximal}_{\theta_t}\big(\gamma G(\theta_t, \xi^t)\big).$$

  Estimator of solution: randomly selected from $\{\theta_t\}_{t=1}^T$.
- For step size $\gamma = \sqrt{2V(\theta_1, \theta^*)/(S\sigma^2 T)}$,

$$\mathbb{E}\big[F(\widehat{\theta}_{1:T}) - F(\theta^*)\big] \leq \sqrt{\frac{2S\sigma^2 V(\theta_1, \theta^*)}{T}}.$$

Bias-Variance Trade-off for SMD

- Consider S-smooth and convex optimization problem

$$\begin{array}{ll} \text{Minimize} & F(\theta) \\ \text{s.t.} & \theta \in \Theta \subseteq \mathbb{R}^d. \end{array}$$

- Stochastic Mirror Descent: iteratively,
  - Step 1: generate random vector $v(\theta_t)$ with

$$\mathbb{E}[v(\theta_t)] = \nabla \overline{F}(\theta_t), \quad \Delta_F := \sup_{\theta \in \Theta} |\overline{F}(\theta) - F(\theta)|, \quad \mathbb{V}\mathrm{ar}\,[v(\theta_t)] \leq \sigma^2.$$

  - Step 2: perform

$$\theta_{t+1} = \textbf{Proximal}_{\theta_t}\big(\gamma v(\theta_t)\big).$$

$\widehat{\theta}_{1:T}$: estimator randomly selected from $\{\theta_t\}_{t=1}^T$.

- For step size $\gamma = \sqrt{2V(\theta_1, \overline{\theta}^*)/(S\sigma^2 T)}$,

$$\mathbb{E}\big[F(\widehat{\theta}_{1:T}) - F(\theta^*)\big] \leq 2\Delta_F + \sqrt{\frac{2S\sigma^2 V(\theta_1, \theta^*)}{T}}.$$

Optimization Algorithm for Sinkhorn DRO: Biased Gradient Update

- Goal: to solve the optimization

$$\min_{\theta\in\Theta} \left\{F(\theta) := \mathbb{E}_{x\sim\widehat{\mathbb{P}}}\left[\lambda\epsilon\log\left(\mathbb{E}_{z\sim\mathbb{Q}_x}\left[e^{f_\theta(z)/(\lambda\epsilon)}\right]\right)\right]\right\}.$$

- Biased gradient update: for each iteration t,
  - Construct a gradient estimate of $F(\theta_t)$, denoted as $v(\theta_t)$;
  - Update $\theta_{t+1} = \textbf{Proximal}_{\theta_t}\left(\gamma_t v(\theta_t)\right)$.

  Estimator of solution: randomly selected from (or average over) $\{\theta_t\}_{t=1}^T$.

  Remark: optimally pick gradient estimator to balance bias-variance trade-off [Hu, Chen and He 2021].

Optimization Algorithm for Sinkhorn DRO: Biased Gradient Update

- Goal: to solve the optimization

$$\min_{\theta\in\Theta} \left\{ F(\theta) := \mathbb{E}_{x\sim\widehat{\mathbb{P}}} \left[\lambda\epsilon \log \left(\mathbb{E}_{z\sim\mathbb{Q}_x} \left[e^{f_\theta(z)/(\lambda\epsilon)} \right] \right) \right] \right\}.$$

- Biased gradient update: for each iteration t,
  - Construct a gradient estimate of $F(\theta_t)$, denoted as $v(\theta_t)$;
  - Update $\theta_{t+1} = \textbf{Proximal}_{\theta_t}\left(\gamma_t v(\theta_t)\right)$.

  Estimator of solution: randomly selected from (or average over) $\{\theta_t\}_{t=1}^T$.

  Remark: optimally pick gradient estimator to balance bias-variance trade-off [Hu, Chen and He 2021].

Optimization Algorithm for Sinkhorn DRO: Biased Gradient Update

- Goal: to solve the optimization

$$\min_{\theta\in\Theta} \left\{F(\theta) := \mathbb{E}_{x\sim\widehat{\mathbb{P}}}\left[\lambda\epsilon \log\left(\mathbb{E}_{z\sim\mathbb{Q}_x}\left[e^{f_\theta(z)/(\lambda\epsilon)}\right]\right)\right]\right\}.$$

- Biased gradient update: for each iteration t,
  - Construct a gradient estimate of $F(\theta_t)$, denoted as $v(\theta_t)$;
  - Update $\theta_{t+1} = \textbf{Proximal}_{\theta_t}\big(\gamma_t v(\theta_t)\big)$.

Estimator of solution: randomly selected from (or average over) $\{\theta_t\}_{t=1}^T$.

Remark: optimally pick gradient estimator to balance bias-variance trade-off [Hu, Chen and He 2021].

Estimators	Convex Nonsmooth	Convex Smooth	Nonconvex Smooth
Vanilla SGD	$O(\delta^{-3})$	$O(\delta^{-3})$	$O(\delta^{-6})$
V-MLMC	N/A	$\tilde{O}(\delta^{-2})$	$\tilde{O}(\delta^{-4})$
RT-MLMC	N/A	$\tilde{O}(\delta^{-2})$	$\tilde{O}(\delta^{-4})$

Configuration of Gradient Estimators

- Goal: to solve the optimization

$$\min_{\theta\in\Theta} \left\{F(\theta) := \mathbb{E}_{x\sim\widehat{\mathbb{P}}} \left[\lambda\epsilon \log\left(\mathbb{E}_{z\sim\mathbb{Q}_x}\left[e^{f_\theta(z)/(\lambda\epsilon)}\right]\right)\right]\right\}.$$

- Construct a sequence of approximation functions $\{F^\ell(\theta)\}_{\ell\geq 0}$ instead, where

$$F^\ell(\theta) = \mathbb{E}_{x^\ell\sim\widehat{\mathbb{P}}}\, \mathbb{E}_{\{z_j^\ell\}_{j\in[2^\ell]}|x^\ell} \left[\lambda\epsilon \log\left(\frac{1}{2^\ell}\sum_{j\in[2^\ell]} \exp\left(\frac{f_\theta(z_j^\ell)}{\lambda\epsilon}\right)\right)\right].$$

Remark: generating unbiased gradient estimate of $F^\ell(\theta)$ is easy!

Naive Gradient Estimators

- **Objective**: generate gradient estimate of

$$F^L(\theta) = \mathbb{E}_{x^L \sim \widehat{\mathbb{P}}} \mathbb{E}_{\{z_j^L\}_{j\in[2^L]}|x^L} \left[\lambda\epsilon \log\left(\frac{1}{2^L} \sum_{j\in[2^L]} \exp\left(\frac{f_\theta(z_j^L)}{\lambda\epsilon} \right) \right) \right].$$

- **Oracle:** sample random parameters $\zeta^L \triangleq \{z_j\}_{j\in[2^L]}$ and compute

$$g^L(\theta, \zeta^L) \triangleq \nabla_\theta \left\{ \lambda\epsilon \log\left(\frac{1}{2^L} \sum_{j\in[2^L]} \exp\left(\frac{f_\theta(z_j^L)}{\lambda\epsilon} \right) \right) \right\}.$$

- **L-SGD Estimator:** at point θ, query oracle for n_L^{o} times to obtain $\{g^L(\theta, \zeta_i^L)\}_{i=1}^{n_L^{\mathrm{o}}}$ and construct

$$v^{\text{L-SGD}}(\theta) = \frac{1}{n_L^{\mathrm{o}}} \sum_{i=1}^{n_L^{\mathrm{o}}} g^L(\theta, \zeta_i^L).$$

Two-Level Monte-Carlo Gradient Estimators

- **Objective**: generate gradient estimate of

$$F^L(\theta) = \mathbb{E}_{x^L \sim \widehat{\mathbb{P}}} \, \mathbb{E}_{\{z_j^L\}_{j\in[2^L]} | x^L} \left[\lambda\epsilon \log \left(\frac{1}{2^L} \sum_{j\in[2^L]} \exp\left(\frac{f_\theta(z_j^L)}{\lambda\epsilon} \right) \right) \right].$$

  Decomposition:

$$\nabla_\theta F^{L-1}(\theta) + \left[\nabla_\theta \big(F^L(\theta) - F^{L-1}(\theta) \big) \right].$$

- **Two-step procedure:**
  1. n_{L-1} queries of oracles for estimating $\nabla_\theta F^{L-1}(\theta)$;
  2. n_L queries of oracles for estimating $\nabla_\theta \big(F^L(\theta) - F^{L-1}(\theta) \big)$.
- **Observation:**
  - Generating an oracle for estimating $\nabla_\theta F^{L-1}(\theta)$ is of $O(2^{L-1})$;
  - Generating an oracle for estimating $\nabla_\theta F^L(\theta)$ is of $O(2^L)$;

Two-Level Monte-Carlo Gradient Estimators

- **Objective**: generate gradient estimate of

$$F^L(\theta) = \mathbb{E}_{x^L \sim \widehat{\mathbb{P}}} \mathbb{E}_{\{z_j^L\}_{j\in[2^L]}|x^L} \left[\lambda\epsilon \log\left(\frac{1}{2^L} \sum_{j\in[2^L]} \exp\left(\frac{f_\theta(z_j^L)}{\lambda\epsilon} \right) \right) \right].$$

  Decomposition:

$$\nabla_\theta F^{L-1}(\theta) + \left[\nabla_\theta \big(F^L(\theta) - F^{L-1}(\theta) \big) \right].$$

- **Two-step procedure:**
  1. n_{L-1} queries of oracles for estimating $\nabla_\theta F^{L-1}(\theta)$;
  2. n_L queries of oracles for estimating $\nabla_\theta \big(F^L(\theta) - F^{L-1}(\theta) \big)$.
- **Advantages:**
  - n_{L-1} is large, but each oracle query is quite efficient.
  - Since $F^L(\theta) \approx F^{L-1}(\theta)$, $\nabla_\theta \big(F^L(\theta) - F^{L-1}(\theta) \big)$ has small variance.
    Thereby n_L is small.

List of Multi-Level Monte-Carlo Gradient Estimators

- Denote

$$U_{n_1:n_2}(\theta, \zeta^\ell) = \lambda\epsilon \log\left(\frac{1}{n_2 - n_1 + 1}\sum_{j\in[n_1:n_2]} \exp\left(\frac{f_\theta(z_j^\ell)}{\lambda\epsilon}\right)\right).$$

Define

$$\begin{aligned} g^\ell(\theta, \zeta^\ell) &= \nabla_\theta U_{1:2^\ell}(\theta, \zeta^\ell), \\ G^\ell(\theta, \zeta^\ell) &= \nabla_\theta\left[U_{1:2^\ell}(\theta, \zeta^\ell) - \frac{1}{2}U_{1:2^{\ell-1}}(\theta, \zeta^\ell) - \frac{1}{2}U_{2^{\ell-1}+1:2^\ell}(\theta, \zeta^\ell)\right]. \end{aligned}$$

List of Multi-Level Monte-Carlo Gradient Estimators

- Denote
$$U_{n_1:n_2}(\theta, \zeta^\ell) = \lambda\epsilon \log\left(\frac{1}{n_2 - n_1 + 1} \sum_{j\in[n_1:n_2]} \exp\left(\frac{f_\theta(z_j^\ell)}{\lambda\epsilon}\right)\right).$$
Define
$$g^\ell(\theta, \zeta^\ell) = \nabla_\theta U_{1:2^\ell}(\theta, \zeta^\ell),$$
$$G^\ell(\theta, \zeta^\ell) = \nabla_\theta \left[U_{1:2^\ell}(\theta, \zeta^\ell) - \frac{1}{2}U_{1:2^{\ell-1}}(\theta, \zeta^\ell) - \frac{1}{2}U_{2^{\ell-1}+1:2^\ell}(\theta, \zeta^\ell)\right].$$
- **L-SGD Estimator:** at point θ, query oracle for n_L^{o} times to obtain $\{g^L(\theta, \zeta_i^L)\}_{i=1}^{n_L^{\text{o}}}$ and construct
$$v^{\text{L-SGD}}(\theta) = \frac{1}{n_L^{\text{o}}} \sum_{i=1}^{n_L^{\text{o}}} g^L(\theta, \zeta_i^L).$$

List of Multi-Level Monte-Carlo Gradient Estimators

- Denote

$$U_{n_1:n_2}(\theta,\zeta^\ell) = \lambda\epsilon \log\left(\frac{1}{n_2-n_1+1}\sum_{j\in[n_1:n_2]}\exp\left(\frac{f_\theta(z_j^\ell)}{\lambda\epsilon}\right)\right).$$

  Define

$$\begin{aligned} g^\ell(\theta,\zeta^\ell) &= \nabla_\theta U_{1:2^\ell}(\theta,\zeta^\ell), \\ G^\ell(\theta,\zeta^\ell) &= \nabla_\theta\left[U_{1:2^\ell}(\theta,\zeta^\ell) - \frac{1}{2}U_{1:2^{\ell-1}}(\theta,\zeta^\ell) - \frac{1}{2}U_{2^{\ell-1}+1:2^\ell}(\theta,\zeta^\ell)\right]. \end{aligned}$$

- **V-MLMC Estimator:** at point θ, for each ℓ we query oracle for n_ℓ times to obtain $\{G^\ell(\theta,\zeta_i^\ell)\}_{i=1}^{n_\ell}$ and construct

$$v^{\text{V-MLMC}}(\theta) = \sum_{\ell=0}^{L}\frac{1}{n_\ell}\sum_{i=1}^{n_\ell}G^\ell(\theta,\zeta_i^\ell).$$

List of Multi-Level Monte-Carlo Gradient Estimators

- Denote

$$U_{n_1:n_2}(\theta, \zeta^\ell) = \lambda\epsilon \log \left(\frac{1}{n_2 - n_1 + 1} \sum_{j \in [n_1:n_2]} \exp\left(\frac{f_\theta(z_j^\ell)}{\lambda\epsilon} \right) \right).$$

  Define

$$\begin{aligned} g^\ell(\theta, \zeta^\ell) &= \nabla_\theta U_{1:2^\ell}(\theta, \zeta^\ell), \\ G^\ell(\theta, \zeta^\ell) &= \nabla_\theta \left[U_{1:2^\ell}(\theta, \zeta^\ell) - \frac{1}{2} U_{1:2^{\ell-1}}(\theta, \zeta^\ell) - \frac{1}{2} U_{2^{\ell-1}+1:2^\ell}(\theta, \zeta^\ell) \right]. \end{aligned}$$

- **RT-MLMC Estimator:** at point θ, we sample *random levels* for n_L^{o} times, denoted as $\iota_1, \ldots, \iota_{n_L^{\text{o}}}$, following distribution $Q_{\mathsf{RT}} = \{q_\ell\}_{\ell=0}^L$ with $\mathbb{P}(\iota = \ell) = q_\ell$. Then construct

$$v^{\text{RT-MLMC}}(\theta) = \frac{1}{n_L^{\text{o}}} \sum_{i=1}^{n_L^{\text{o}}} \frac{1}{q_{\iota_i}} G^{\iota_i}(\theta, \zeta^{\iota_i}).$$

List of Multi-Level Monte-Carlo Gradient Estimators

- Denote

$$U_{n_1:n_2}(\theta, \zeta^\ell) = \lambda\epsilon \log\left(\frac{1}{n_2 - n_1 + 1}\sum_{j\in[n_1:n_2]} \exp\left(\frac{f_\theta(z_j^\ell)}{\lambda\epsilon}\right)\right).$$

  Define

$$g^\ell(\theta, \zeta^\ell) = \nabla_\theta U_{1:2^\ell}(\theta, \zeta^\ell),$$
$$G^\ell(\theta, \zeta^\ell) = \nabla_\theta\left[U_{1:2^\ell}(\theta, \zeta^\ell) - \frac{1}{2}U_{1:2^{\ell-1}}(\theta, \zeta^\ell) - \frac{1}{2}U_{2^{\ell-1}+1:2^\ell}(\theta, \zeta^\ell)\right].$$

- **Highlight of MLMC Estimators:** Cost reduced significantly, with variance reduction.