Generalizable Machine Learning via Inductive Modeling of Distribution Shifts

Jiashuo Liu

Department of Computer Science

Tsinghua University

July 3rd, 2025

Joint work with Jose Blanchet, Peng Cui, Jiajin Li, Mihaela van der Schaar

Outline

Overview

Tool 1: Model Stability Evaluation

Tool 2: Risk Region Analysis

Tool 3: Performance Drop Diagnosis

Background

Machine learning algorithms have been widely applied in prediction and decision-making systems.

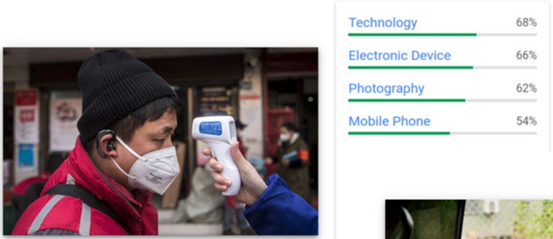
Policy Making

Bank Loans

Medical Diagnosis

Real-World Challenges of AI Systems

Biases exist in AI systems.



Screenshot from 2020-03-31 11-27-22.png

Screenshot from 2020-03-31 11-23-45.png

Real-World Challenges of AI Systems

Hard to generalize.

Owner: “Car kept jamming on the brakes thinking this was a person”

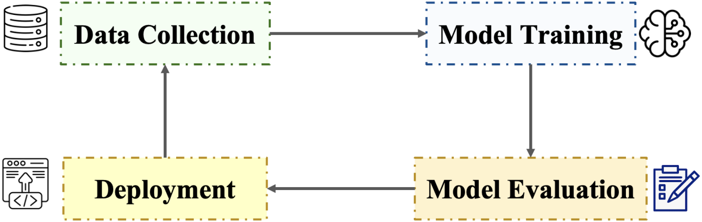
Real-World Challenges of AI Systems

Hard to generalize.

AI Camera Ruins Soccer Game For Fans After Mistaking Referee's Bald Head For Ball

System Level of View of AI

Building a reliable AI stack requires a holistic view.



- Previous: focus on model training
- Now: focus on evaluation & deployment

Invariant Risk Minimization

Assume existence of feature $\Phi(X)$ such that $Y|\Phi(X)$ is invariant across environments. Then, learn this feature.

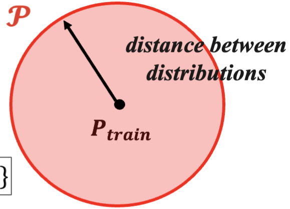
$$\begin{aligned} \min_{\substack{\Phi:\mathcal{X}\to\mathcal{H}\\ w:\mathcal{H}\to\mathcal{Y}}} \quad & \sum_{e\in\mathcal{E}_{\mathrm{tr}}} R^e(w\circ\Phi) \\ \text{subject to} \quad & w \in \arg\min_{\bar{w}:\mathcal{H}\to\mathcal{Y}} R^e(\bar{w}\circ\Phi), \text{ for all } e\in\mathcal{E}_{\mathrm{tr}} \end{aligned}$$

$$\min_{\Phi:\mathcal{X}\to\mathcal{Y}} \sum_{e\in\mathcal{E}_{\mathrm{tr}}} R^e(\Phi) + \lambda\cdot \left\|\nabla_{w|w=1.0} R^e(w\cdot\Phi)\right\|^2, \qquad \text{(IRMv1)}$$

Distributionally Robust Optimization

Empirical Risk Minimization	$\min_{\theta \in \Theta} \mathbb{E}_{Z \sim P_{train}}[\ell(\theta; Z)]$
DRO	$\min_{\theta \in \Theta} \sup_{Q \in \mathcal{P}} \mathbb{E}_{Z \sim Q}[\ell(\theta; Z)]$

$$\boxed{\mathcal{P} = \{Q : Dist(Q, P_{train}) \leq \rho\}}$$



Instead of minimizing loss over training distribution, minimize loss over distributions ***near*** it

Advertisement:)

We made a python package for "Distributionally Robust Optimization".

- 14 DRO formulations and 9 backbone models
- https://python-dro.org

License MIT Downloads 4k

dro : A Python Package for Distributionally Robust Optimization in Machine Learning

Jiashuo Liu†, Tianyu Wang†, Henry Lam, Hongseok Namkoong, Jose Blanchet

† equal contributions (α-β order)

Distributionally Robust Optimization

DRO: A Python Library for Distributionally Robust Optimization in Machine Learning

Jiashuo Liu* LIUJIASHUO77@GMAIL.COM
Tianyu Wang* TW2837@COLUMBIA.EDU
Henry Lam† HENRY.LAM@COLUMBIA.EDU
Hongseok Namkoong† NAMKOONG@GSB.COLUMBIA.EDU
Jose Blanchet† JOSE.BLANCHET@STANFORD.EDU

Abstract

We introduce `dro`, an open-source Python library for distributionally robust optimization (DRO) for regression and classification problems. The library implements 14 DRO formulations and 9 backbone models, enabling 79 distinct DRO methods. Through vectorization and optimization approximation techniques, `dro` reduces runtime by 10x to over 1000x compared to baseline implementations on large-scale datasets. Comprehensive documentation is available at `https://python-dro.org`.

Keywords: distributionally robust optimization, distribution shift, machine learning

Advertisement:)

Table 1: Different DRO methods supported in `dro` package.

	Exact Optimization						Approximate Optimization		
	LAD	OLS	SVM	Logistic	Kernel	Personal	Tree-based	NN	Personal
WDRO	✓	✓	✓	✓	✓	✓		✓	✓
RS-WDRO	✓	✓	✓	✓	✓	✓			
χ^2-DRO	✓	✓	✓	✓	✓	✓	✓	✓	✓
KL-DRO	✓	✓	✓	✓	✓	✓	✓		
Bayesian-DRO	✓	✓	✓	✓					
CVaR-DRO	✓	✓	✓	✓	✓	✓	✓	✓	✓
TV-DRO	✓	✓	✓	✓	✓	✓			
Marginal(-CVaR)-DRO	✓	✓	✓	✓	✓	✓			
Conditional(-CVaR)-DRO	✓	✓	✓	✓	✓	✓			
MMD-DRO	✓	✓	✓	✓		✓			
HR-DRO	✓		✓					✓	
Sinkhorn-DRO	✓	✓	✓	✓				✓	
OutlierRobust(OR)-WDRO	✓		✓						
MOT-DRO	✓		✓						

Previous Philosophy

from **ASSUMPTION** to **Algorithm**

- assume there's a causal structure, and no hidden confounders $\rightarrow$ causal algorithms
- assume the test distribution is near the training distribution $\rightarrow$ distributionally robust optimization methods

However, do those assumptions really hold in practice?
No idea!

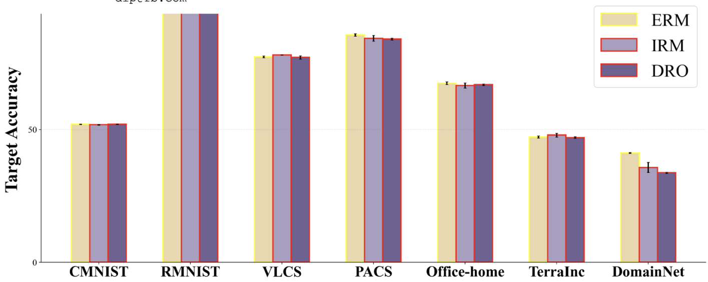
Previous Philosophy

Not really!

IN SEARCH OF LOST DOMAIN GENERALIZATION

Ishaan Gulrajani*
Stanford University
igul222@gmail.com

David Lopez-Paz
Facebook AI Research
dlp@fb.com



Plot generated from Table 4 from Gulrajani, I., & Lopez-Paz, D. (2020, October). In Search of Lost Domain Generalization. In International Conference on Learning Representations.

What we're calling for

from **UNDERSTANDING** to **Algorithm**

- shift the focus from model training to evaluation & deployment
- Evaluation Stage:
  - understand your model's stability under potential shifts → **Model Stability Evaluation**
  - understand your model's underperformed regions within distribution → **Risk Region Analysis**
- Deployment Stage:
  - understand your model's performance drop between distributions→ **Performance Drop Diagnosis**

Better understanding enables more efficient improvements!

Outline

Overview

Tool 1: Model Stability Evaluation

Tool 2: Risk Region Analysis

Tool 3: Performance Drop Diagnosis

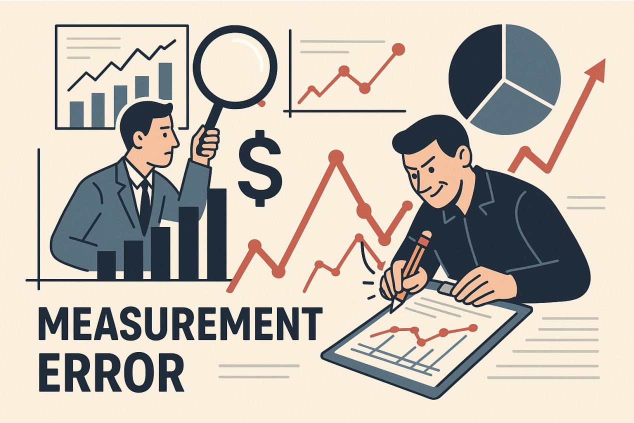
Stability Evaluation

Problem: How do we **evaluate the stability** of a learning model (like neural networks and LLMs) when subjected to **data perturbations**?

Two classes of data perturbations:

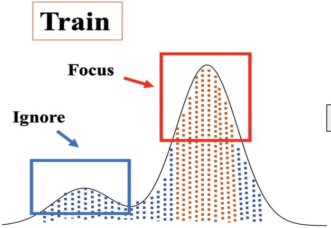
- Data corruptions: changes in the distribution support (i.e., observed data samples).
- Sub-population shifts: perturbation on the probability density or mass function while keeping the same support.

Measurement Error/Noises



Example: Sub-population Shifts

AI Systems can be biased against the minority groups



Amazon scraps secret AI recruiting tool that showed bias against women REUTERS

Preliminary

- OT discrepancy with moment constraints [1]

$$\mathbb{M}_c(\mathbb{Q},\mathbb{P}) = \begin{cases} \inf & \mathbb{E}_\pi[c((Z,W),(\hat{Z},\hat{W}))] \\ \text{s.t.} & \pi \in \mathcal{P}((\mathcal{Z}\times\mathcal{W})^2) \\ & \pi_{(Z,W)} = \mathbb{Q},\ \ \pi_{(\hat{Z},\hat{W})} = \mathbb{P} \\ & \mathbb{E}_\pi[W] = 1 \quad \pi\text{-a.s,} \end{cases}$$

where $\pi_{(Z,W)}$ and $\pi_{(\hat{Z},\hat{W})}$ are the marginal distributions of (Z,W) and $(\hat{Z},\hat{W})$ under π.

- Lift the original sample space $\mathcal{Z}$ to a higher dimensional space $\mathcal{Z}\times\mathcal{W}$ — perturb on a joint (sample, density) space.
- We choose the cost function as:

$$c((z,w),(\hat{z},\hat{w})) = \underbrace{\theta_1\cdot w\cdot(\|x-\hat{x}\|_2^2 + \infty\cdot|y-\hat{y}|)}_{\text{differences between samples}} + \underbrace{\theta_2\cdot(\phi(w)-\phi(\hat{w}))_+}_{\text{differences in probability mass}}.$$

Formulation

Given a learning model f_β and the distribution $\mathbb{P}_0 \in \mathcal{P}(\mathcal{Z})$, we formally introduce the **OT-based stability evaluation criterion** as

$$\mathfrak{R}(\beta, r) = \begin{cases} \inf\limits_{\mathbb{Q} \in \mathcal{P}(\mathcal{Z} \times \mathcal{W})} & \mathbb{M}_c(\mathbb{Q}, \hat{\mathbb{P}}) \\ \text{s.t.} & \underbrace{\mathbb{E}_{\mathbb{Q}}[W \cdot \ell(\beta, Z)]}_{\text{risk under } \mathbb{Q}} \geq \underbrace{r}_{\text{threshold}} \end{cases}. \tag{P}$$

Larger $\mathfrak{R}(\beta, r) \Rightarrow$ More Stable

- Quantify the minimum level of perturbations required for the model's performance to degrade to a predetermined risk threshold.
- $\hat{\mathbb{P}}$: The reference measure selected as $\mathbb{P}_0 \otimes \delta_1$, with δ_1 denoting the Dirac delta function.
- $r > 0$: the *pre-defined* risk threshold (according to policies or ML engineers).
- θ_1, θ_2: Control the relative strength of data corruption and reweighting. When $\theta_1 \to \infty$, the measure degenerates to Namkoong et al. [4].

Illustrations

Projection distance to the distribution set where the model performance falls below a specific threshold.

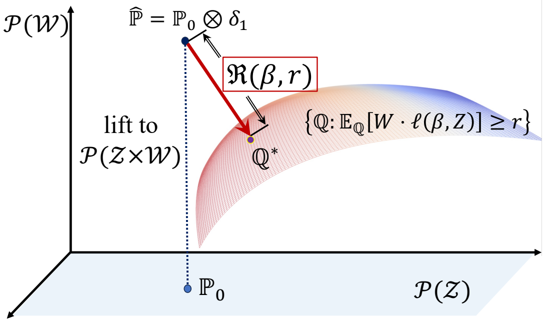


Figure 1: Data distribution projection in the joint (sample, density) space.

Strong Duality

Theorem (Strong duality for problem (P))

Suppose that (i) The set $\mathcal{Z} \times \mathcal{W}$ is compact[a], (ii) $\ell(\beta, \cdot)$ is upper semi-continuous for all β, (iii) the cost function $c : (\mathcal{Z} \times \mathcal{W})^2 \to \mathbb{R}_+$ is continuous; and (iv) the risk level r is less than the worst-case value $\bar{r} := \max_{z \in \mathcal{Z}} \ell(\beta, z)$. Then we have,

$$\mathfrak{R}(\beta, r) = \sup_{h \in \mathbb{R}_+, \alpha \in \mathbb{R}} hr + \alpha + \mathbb{E}_{\hat{\mathbb{P}}}\left[\tilde{\ell}_c^{\alpha,h}(\beta, (\hat{Z}, \hat{W}))\right] \tag{D}$$

where the surrogate function $\tilde{\ell}_c^{\alpha,h}(\beta, (\hat{z}, \hat{w}))$ equals to

$$\min_{(z,w) \in \mathcal{Z} \times \mathcal{W}} c((z, w), (\hat{z}, \hat{w})) + \alpha w - h \cdot w \cdot \ell(\beta, z),$$

for all $\hat{z} \in \mathcal{Z}$ and $\hat{w} \in \mathcal{W}$.

---

[a] When the reference measure $\mathbb{P}_0$ is a discrete measure, some technical conditions (e.g., compactness, (semi)-continuity) can be eliminated.

Dual Reformulation

Theorem (Dual reformulations)

Suppose that $\mathcal{W} = \mathbb{R}_+$. (i) If $\phi(t) = t\log t - t + 1$, then the dual problem (D) admits:

$$\sup_{h\geq 0} hr - \theta_2 \log \mathbb{E}_{\mathbb{P}_0}\left[\exp\left(\frac{\ell_{h,\theta_1}(\hat{Z})}{\theta_2}\right)\right]; \tag{1}$$

(ii) If $\phi(t) = (t-1)^2$, then the dual problem (D) admits:

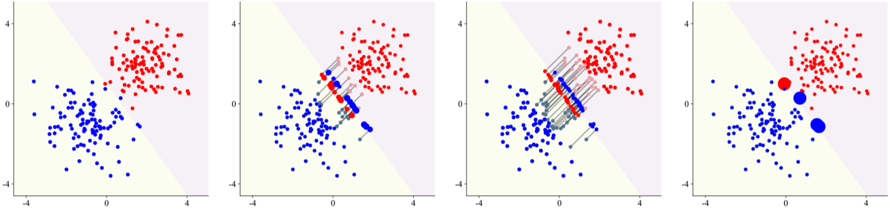
$$\sup_{h\geq 0, \alpha\in\mathbb{R}} hr + \alpha + \theta_2 - \theta_2 \mathbb{E}_{\mathbb{P}_0}\left[\left(\frac{\ell_{h,\theta_1}(\hat{Z}) + \alpha}{2\theta_2} + 1\right)_+^2\right], \tag{2}$$

where the d-transform of $h\cdot\ell(\beta,\cdot)$ with the step size θ_1 is defined as

$$\ell_{h,\theta_1}(\hat{z}) := \max_{z\in\mathcal{Z}} h\cdot\ell(\beta,z) - \theta_1\cdot d(z,\hat{z}).$$

Visualizations on Toy Examples

Visualize the most sensitive distribution $\mathbb{Q}^\star$:



(a) Original Dataset (b) $\theta_1 = 1.0, \theta_2 = 0.25$ (c) $\theta_1 = 0.2, \theta_2 = +\infty$ (d) $\theta_1 = +\infty, \theta_2 = 0.2$

Figure 2: Visualizations on toy examples with *0/1* loss function under different θ_1, θ_2. The original prediction error rate is 1%, and the error rate threshold r is set to 30%. The size of each point is proportional to its sample weight in $\mathbb{Q}^\star$

Usage 1: MLP Stability Analysis

Task: Predict individual's income based on personal features.

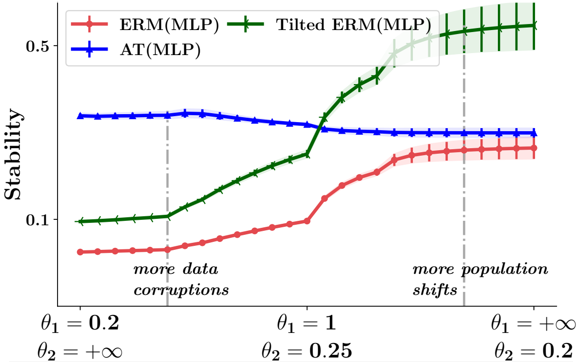
Under evaluation: MLP models optimized via

- Empirical Risk Minimization (ERM)
- Adversarial Training (AT): designed for robustness to data corruptions
- Tilted ERM: designed for robustness to sub-population shifts

Usage 1: MLP Stability Analysis

Insight: A method designed for one class of data perturbation may not be robust against another.

- AT is not stable under sub-population shifts.
- Tilted ERM is not stable under data corruptions.



Usage 2: LLM Stability Analysis

Task: Question-answering (general question & harmful question)

Under evaluation: General LLMs

- Llama-2-chat 7B/13B
- Vicuna 7B/13B
- Mistral 7B
- Deepseek-2 7B
- Qwen-2 7B
- ChatGLM-2 6B

Usage 2: LLM Stability Analysis

Adapt the cost function for LLM:

$$
\begin{aligned}
c((z,w),&(\hat{z},\hat{w})) = \\
&\theta_2 \cdot \underbrace{(\phi(w)-\phi(\hat{w}))_+}_{\text{reweighting distance}} + \\
&\theta_1 \cdot w \cdot \underbrace{\left(\underbrace{\frac{\Phi(x)^T\Phi(\hat{x})}{\|\Phi(x)\|\|\Phi(\hat{x})\|}}_{\text{semantic similarity}} \cdot \underbrace{\max(\frac{\#\text{Token}(x)}{\#\text{Token}(\hat{x})}, \frac{\#\text{Token}(\hat{x})}{\#\text{Token}(x)})\right)}_{\text{token number ratio}}}_{\text{perturbation distance}}.
\end{aligned}
\tag{3}
$$

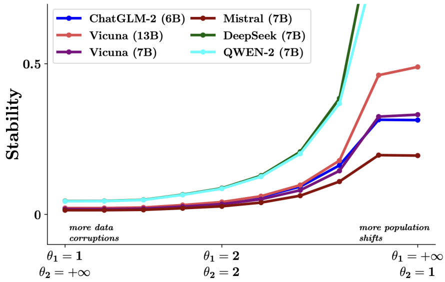
For minimal data perturbation:

- Preserve the semantic meaning
- Ensure the sentence length is similar to the original

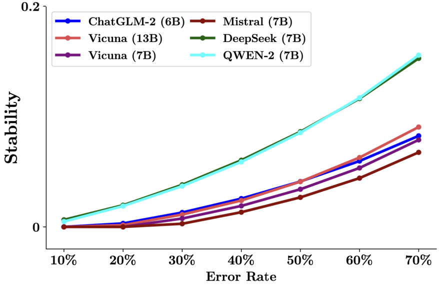
Usage 2: LLM Stability Analysis

Insight: LLM evaluation should not rely on one single metric.

- Ranking of LLMs changes based on different patterns of distribution shifts (θ_1, θ_2), and error rate r.



(a) Varying θ_1, θ_2 on Jailbreak

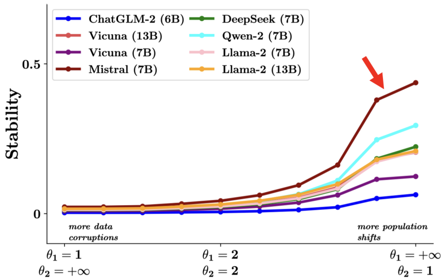


(b) Varying r on Jailbreak

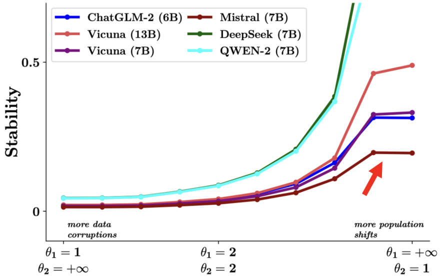
Usage 2: LLM Stability Analysis

Insight: Tradeoff in stability between answering harmless and (not answering) harmful questions.

- Mistral-7B (dark red curve) performs exceptionally well on harmless question answering, but much badly on (not answering) harmful questions.
- Good semantic reasoning ability makes it easier to be cheated by "role-playing" prompts.



(c) Varying θ_1, θ_2 on Alpaca



(a) Varying θ_1, θ_2 on Jailbreak

Usage 3: Feature Stability Analysis

Feature Stability

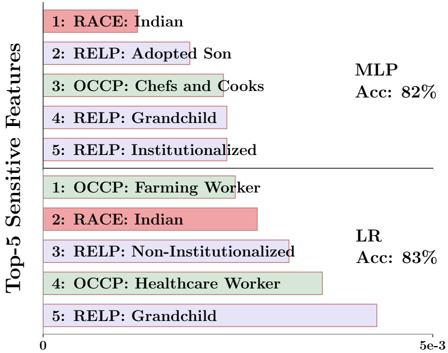
- perturbing on which feature will cause model's performance drop
- providing more fine-grained diagnosis for a prediction model

For i-th feature, choose the cost function as:

$$
\begin{aligned}
c((z,w),&(\hat{z},\hat{w})) = \\
&\theta_2 \cdot (\phi(w) - \phi(\hat{w}))_+ + \\
&\theta_1 \cdot w \cdot \underbrace{(\|z_{(i)} - \hat{z}_{(i)}\|_2^2 + \infty \cdot \|z_{(,-i)} - \hat{z}_{(,-i)}\|_2^2)}_{\text{only allow perturbations on } i\text{-th feature}}.
\end{aligned}
$$

Usage 3: Feature Stability Analysis

Task: predict individual's income based on personal features
Dataset: `ACS Income` [2]



Insight: ERM model focuses too much on the "American Indian" feature, which may introduce potential fairness problem!

Usage 4: "Targeted" Algorithmic Intervention

One simple example:

- AT:

$$\min_{\beta}\left\{\mathbb{E}_{\mathbb{P}_0}[\phi_\gamma(\beta,Z)] := \mathbb{E}_{\mathbb{P}_0}\left[\sup_{z\in\mathcal{Z}}\ell(\beta,Z)-\gamma c(Z,\hat{Z})\right]\right\}, \qquad (4)$$

  where $c(z,\hat{z}) = \|x-\hat{x}\|_2^2 + \infty\cdot|y-\hat{y}|$, and γ is the penalty hyper-parameter.

- Targeted AT:

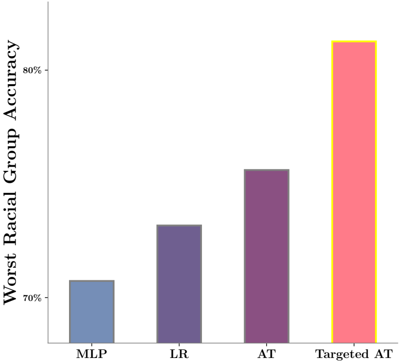
$$\min_{\beta}\left\{\mathbb{E}_{\mathbb{P}_0}[\phi_\gamma(\beta,Z)] = \mathbb{E}_{\mathbb{P}_0}\left[\sup_{z\in\mathcal{Z}}\ell(\beta,Z)-\gamma c(Z,\hat{Z})\right]\right\}. \qquad (5)$$

  $c(z,\hat{z}) = \|z_{(i)}-\hat{z}_{(i)}\|_2^2 + \infty\cdot\|z_{(,-i)}-\hat{z}_{(,-i)}\|_2^2$, where $z_{(i)}$ denotes the target feature of z, $z_{(,-i)}$ denotes all the other features and γ is the penalty hyper-parameter.

Usage 4: "Targeted" Algorithmic Intervention

Insight: Feature stability can motivate refined algorithmic intervention.

- for AT, only perturb the <u>identified</u> sensitive racial feature "American Indian"
- significantly increase the worst racial group accuracy
- align with the empirical findings in WhyShift [3, Section 5]



Takeaways

- A stability measure for ML models (both neural networks and LLMs) based on optimal transport.
- Consider different data perturbations at the same time.
- Help to understand why model fails, and guide targeted algorithmic interventions.

Evaluate $\rightarrow$ Understand $\rightarrow$ Improve

Refer to our papers for more details:

- Jose Blanchet, Peng Cui, Jiajin Li, and Jiashuo Liu (α-β). Stability Evaluation through Distributional Perturbation Analysis. ICML 2024. https://arxiv.org/pdf/2405.03198
- Jiashuo Liu, Jiajin Li, Peng Cui, and Jose Blanchet. Stability Evaluation of Large Language Models via Distributional Perturbation Analysis. NeurIPS 2024 Workshop on Red Teaming GenAI.

Outline

Overview

Tool 1: Model Stability Evaluation

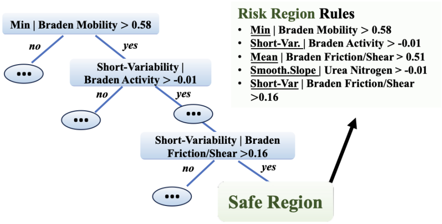
Tool 2: Risk Region Analysis

Tool 3: Performance Drop Diagnosis

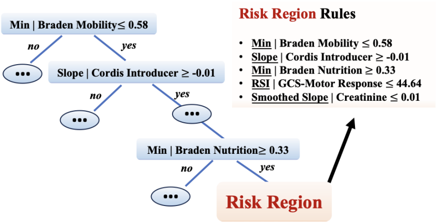
Risk Region Analysis

Problem: Beyond the overall performance, how do we **understand** where our model performs well and where not?

A simple but effective method: fit a decision tree to predict the sample loss from covariates.



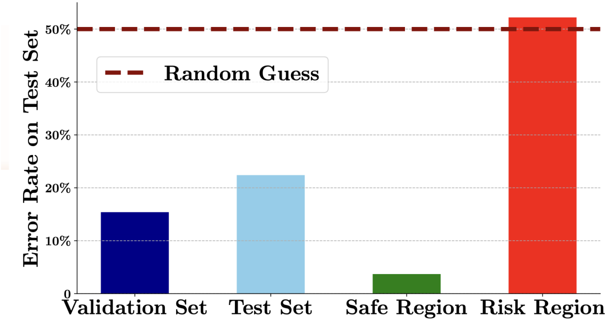
(a) Safe Region.



(b) Risk Region.

Risk Region Analysis

Enable “smart” deployment in practice.



Broader: Error Slice Discovery

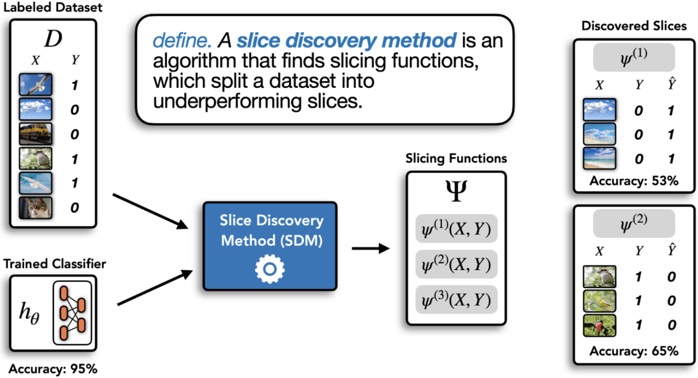


Figure from https://ai.stanford.edu/blog/domino/

Takeaways

- A detailed understanding of model performance enables "smart" model deployment & data collection.

Refer to our papers for more details:

- Jiashuo Liu, Nabeel Seedat, Peng Cui, Mihaela van der Schaar. Going Beyond Static: Understanding Shifts with Time-Series Attribution. ICLR 2025. https://openreview.net/pdf?id=XQlccqJpCC
- Jiashuo Liu, Tianyu Wang, Peng Cui, Hongseok Namkoong. On the Need for a Language Describing Distribution Shifts: Illustrations on Tabular Datasets. NeurIPS 2023. https://arxiv.org/pdf/2307.05284v1 (Two Sigma NeurIPS 2023 Favorite Paper)
- Other papers on error slice discovery.

Outline

Overview

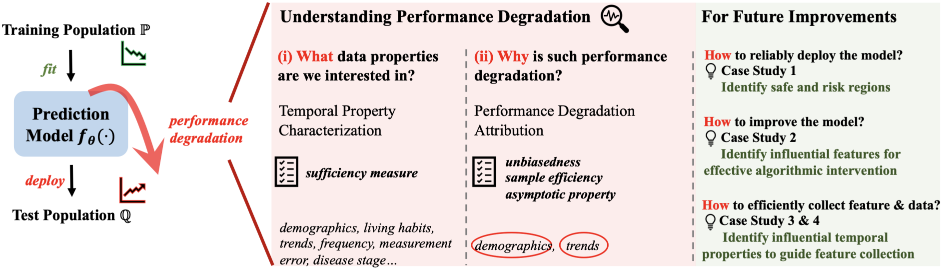
Tool 1: Model Stability Evaluation

Tool 2: Risk Region Analysis

Tool 3: Performance Drop Diagnosis

Problem Setting

When observing performance drop, how to attribute the drop to each time-series feature? Trend changes? Covariance changes?



Time-Series to Static Problem

Extract multiple temporal properties from time-series data.

Temporal Property	Name	Metric	Domain
Global Characteristics	Overall Statistics	Average, Standard Deviation, Max, Min values	Statistics
	Standardized Trend	Equation (16)	Statistics
	Smoothed Trend	Savitzky-Golay Filter (Savitzky & Golay, 1964)	Analytical Chemistry
	Maximum Frequency	Dominant frequency by FFT	Signal Processing
	Signal-to-Noise Ratio	Equation (17)	Signal Processing
Local Dynamics	Short-Term Variability	Equation (19)	Signal Processing
	High-Frequency Energy	Equation (20)	Signal Processing
	Normalized Jitter Index	Equation (21)	Signal Processing
	Relative Strength Index	Equation (22)	Finance
	KPSS Non-Stationary Test	p-Value from KPSS Test	Economics
	Breakout Points	Equation (18) (Bollinger Bands (Bollinger, 1992))	Finance
Structural Changes	Change Points	PELT (Killick et al., 2012)	Statistics
	Trend Variability	Standard deviation of local trends	Statistics
Multivariate Interaction	Covariance Variability	Equation (23)	Finance

Sufficiency measure to evaluate the optimal predictive power:

$$\text{Suff.}(\tilde{X}) = \min_{g\in\mathcal{G}} \mathbb{E}[\text{Loss}(g(\tilde{X}), \ell(f(X), Y))]$$

Attribution

Define the conditional risk as:

$$R_{\mathbb{P}}(\tilde{X}_{-S}) = \mathbb{E}_{\mathbb{P}}[\ell(f(X), Y)|\tilde{X}_{-S}], \quad (6)$$

$$R_{\mathbb{Q}}(\tilde{X}_{-S}) = \mathbb{E}_{\mathbb{Q}}[\ell(f(X), Y)|\tilde{X}_{-S}]. \quad (7)$$

The attribution score is defined as:

$$\text{Attr.}(S) = \mathbb{E}[R_{\mathbb{Q}}(\tilde{X}_{-S}) - R_{\mathbb{P}}(\tilde{X}_{-S})] \quad (8)$$

- similar to average treatment effect estimation
- $\text{Attr.}(\emptyset)$ captures $Y|X$-shifts

Algorithm

Doubly Robust Estimator:

$$\widehat{\mathsf{Attr.}}(S) = \frac{1}{n_P + n_Q}\left(\sum_{i=1}^{n_P+n_Q} \Big(\hat{\mu}_{\mathbb{Q}}(\tilde{X}^i_{-S}) - \hat{\mu}_{\mathbb{P}}(\tilde{X}^i_{-S})\Big) + \sum_{j=1}^{n_Q} \frac{R_{\mathbb{Q}}(\tilde{X}^j_{-S}) - \hat{\mu}_{\mathbb{Q}}(\tilde{X}^j_{-S})}{\pi(\tilde{X}^j_{-S})} - \sum_{i=1}^{n_P} \frac{R_{\mathbb{P}}(\tilde{X}^i_{-S}) - \hat{\mu}_{\mathbb{P}}(\tilde{X}^i_{-S})}{1 - \pi(\tilde{X}^i_{-S})} \right)$$

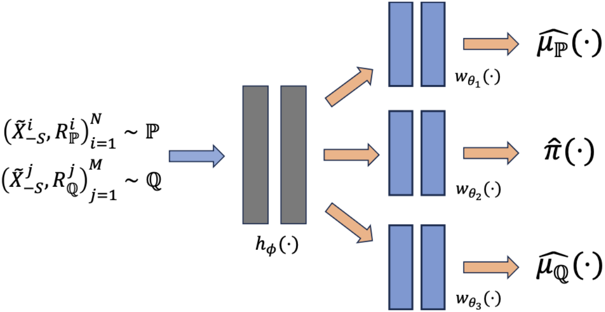
where $\pi(x_{-S}) = \mathsf{Pr}(x_{-S} \text{ from } \mathbb{Q})$.

Theoretical Results:

- Unconfoundedness & Unbiasedness
- Consistency

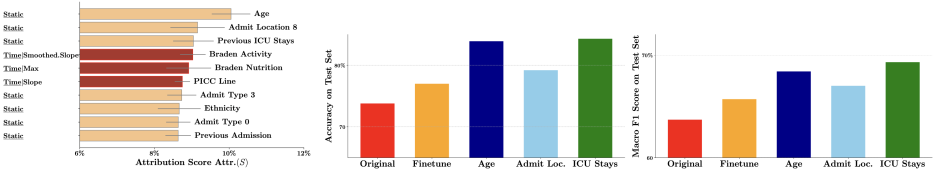
Model Architecture

We use DragonNet for sample efficiency:



Age Shifts in Mortality Risk Prediction

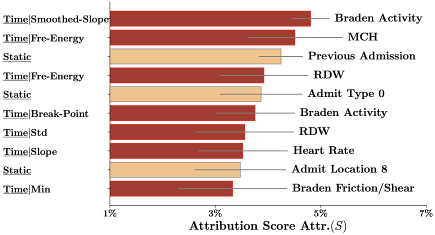
- Correctly attribute the performance drop to age-related features
- Balance the "bins" and retrain the model
- Diagnosis → simple but effective model intervention



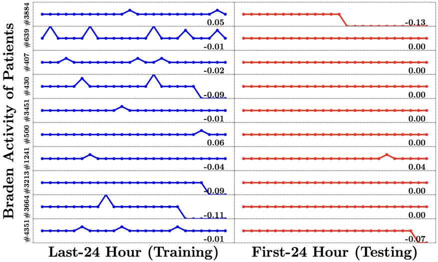
(a) Feature Attribution. (b) Accuracy. (c) Macro-F1 Score.

Preemptive Diagnosis under Temporal Shifts

- Temporal properties are the most important.



(a) Attribution (Case Study 3).

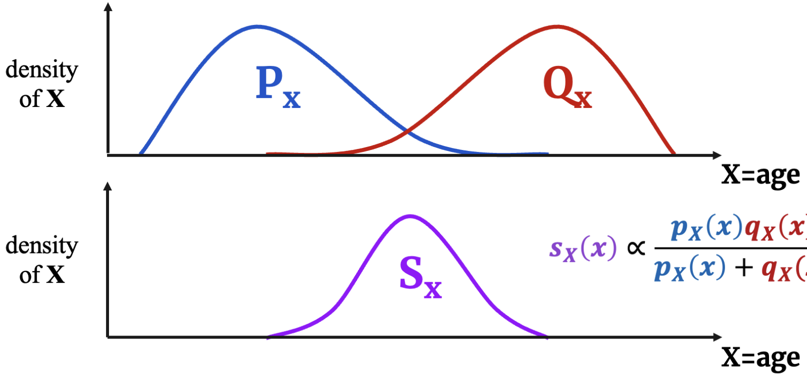


(b) Braden Activity Curve.

Other Forms: X-Shifts vs $Y|X$-Shifts

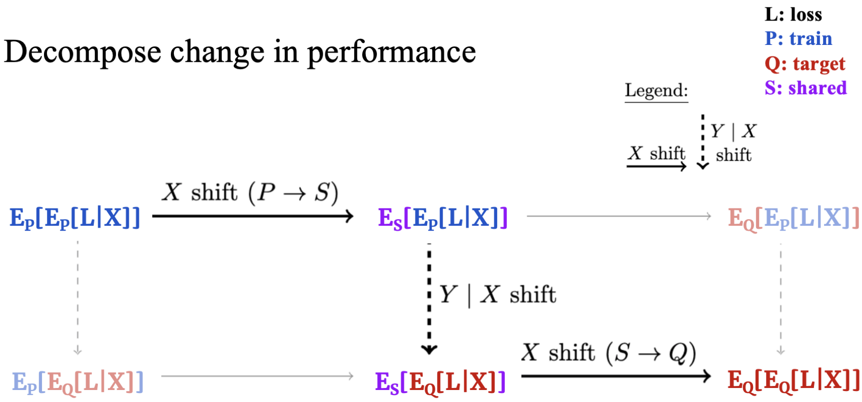
Define **Shared Distribution**

L: loss
P: train
Q: target
S: shared



$$s_X(x) \propto \frac{p_X(x)q_X(x)}{p_X(x)+q_X(x)}$$

Diagnosing Model Performance Under Distribution Shift https://github.com/namkoong-lab/disde https://arxiv.org/abs/2303.02011



Diagnosing Model Performance Under Distribution Shift https://github.com/namkoong-lab/disde https://arxiv.org/abs/2303.02011

Takeaways

- We need an inductive way to deal with distribution shifts.
- Understanding distribution shift is important.
- More methods are needed:)

Refer to our paper for more details:

- Jiashuo Liu, Nabeel Seedat, Peng Cui, Mihaela van der Schaar. Going Beyond Static: Understanding Shifts with Time-Series Attribution. ICLR 2025. https://openreview.net/pdf?id=XQlccqJpCC

Future Works

Toward Self-Robustifying Agents

- Extend current stability evaluation into an online self-monitoring module
- Integrate with a response policy engine (LLMs/RAGs) to suggest/perform actions under distribution shifts
- Form a closed-loop agent system that can evaluate $\rightarrow$ diagnose $\rightarrow$ adjust

References I

[1] Jose Blanchet, Daniel Kuhn, Jiajin Li, and Bahar Taskesen. Unifying distributionally robust optimization via optimal transport theory. *arXiv preprint arXiv:2308.05414*, 2023.

[2] Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. Retiring adult: New datasets for fair machine learning. *Advances in neural information processing systems*, 34:6478–6490, 2021.

[3] Jiashuo Liu, Tianyu Wang, Peng Cui, and Hongseok Namkoong. On the need of a modeling language for distribution shifts: Illustrations on tabular datasets, 2024. URL https://arxiv.org/abs/2307.05284.

[4] Hongseok Namkoong, Yuanzhe Ma, and Peter W Glynn. Minimax optimal estimation of stability under distribution shift. *arXiv preprint arXiv:2212.06338*, 2022.