\Orchestrating a brighter world NEC

AAAI 2021

Measuring Dependence with Matrix-based Entropy Functional

Shujian Yu[1], Francesco Alesiani[1], Xi Yu[2], Robert Jenssen[3], Jose C. Principe[2]

[1]NEC Laboratories Europe GmbH

[2]University of Florida

[3]UiT – The Arctic University of Norway

Contact: Shujian.Yu@neclab.eu

Motivation

Statistical Dependence is Everywhere

- Statistics and Machine Learning
- Biology and Neuroscience
- Signal Processing …

Commonly used Dependence Measures

- Pearson correlation coefficient
  - $\rho_{X,Y} = \mathbb{E}[(X-\mu_X)(Y-\mu_Y)]/\sigma_X\sigma_Y$
- Mutual information
  - $I_{X,Y} = \int_{\mathcal{X}} \int_{\mathcal{Y}} p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right) dxdy$
- Mutual information neural estimator (MINE)[1] …

But …

- Only applicable to two random variables
- Only applicable to scalar variables (rather than random vectors)
- Not interpretable
- Hard to estimate
- Not differentiable

[1] Luis Gonzalo Belghazi, etc. "Mutual information neural estimation." In *International Conference on Machine Learning*, pp. 531-540. 2018.

Motivation

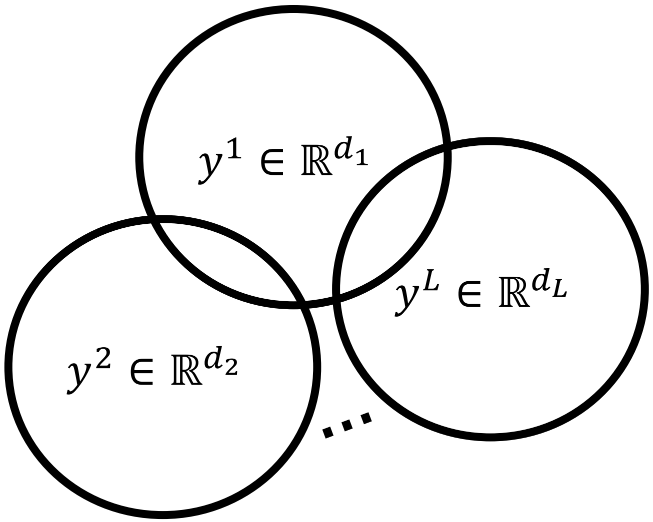
Our Target – A New Dependence Measure

- Applicable to more than two random variables
- Applicable to random vectors
- Interpretable (always between 0 and 1)
  - 0 indicates independence between any pairwise variables
  - 1 indicates functional dependence, e.g., $y^1 = y^2 = \cdots = y^L$
- Easy to estimate
  - Avoid density estimation
  - Avoid model training
- Differentiable
  - Applicable as a loss function to train deep neural networks

Motivation

Problem Setup

- Given L $(L \geq 2)$ variables, $y^1 \in \mathbb{R}^{d_1}, y^2 \in \mathbb{R}^{d_2}, \ldots, y^L \in \mathbb{R}^{d_L}$, the total amount of dependence between $y^1, y^2, \ldots, y^L$?
- Two extreme cases
  - $L = 2 \rightarrow$ random vector associations, e.g., MINE, HSIC[2]
  - $L > 2 \,\&\, \forall_i d_i = 1 \rightarrow$ multivariate correlation analysis, e.g., Multivariate Spearman's ρ[3]
- Our target: no restrictions on L and d_i $(i = 1,2, \ldots, L)$



$$D(y^1; y^2; \ldots, y^L) = ?$$

[2] Gretton, Arthur, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. "Measuring statistical dependence with Hilbert-Schmidt norms." In *International conference on algorithmic learning theory*, pp. 63-77. Springer, Berlin, Heidelberg, 2005.

[3] Schmid, Friedrich, and Rafael Schmidt. "Multivariate extensions of Spearman's rho and related statistics." *Statistics & probability letters* 77, no. 4 (2007): 407-416.

Motivation

General Idea and Preliminary Knowledge

- If variables are pairwise independent, then

$$P(y^1, y^2, \dots, y^L) = \prod_{i=1}^{L} P(y^i)$$

- Measure the difference between $P(y^1, y^2, \dots, y^L)$ to $\prod_{i=1}^{L} P(y^i)$, i.e., $D\left(P(y^1, y^2, \dots, y^L) \parallel \prod_{i=1}^{L} P(y^i)\right)$

- Two examples
  - $D_{\mathrm{KL}}\left(P(y^1, y^2, \dots, y^L) \parallel \prod_{i=1}^{L} P(y^i)\right) = \sum_{i=1}^{L} H(y^i) - H(y^1, y^2, \dots, y^L)$
  - $D_{\mathrm{MMD}}\left(P\left(f(y^1), f(y^2), \dots, f(y^L)\right) \parallel \prod_{i=1}^{L} P\left(f(y^i)\right)\right)$, f is a transform (e.g., the probability integral transform)[4]

[4] Póczos, Barnabás, Zoubin Ghahramani, and Jeff Schneider. "Copula-based kernel dependency measures." In *International Conference on Machine Learning*, pp. 1635-1642, 2012.

Motivation

General Idea and Preliminary Knowledge

- If variables are pairwise independent, then
$$P(y^1, y^2, \dots, y^L) = \prod_{i=1}^{L} P(y^i)$$

- Measure the difference between $P(y^1, y^2, \dots, y^L)$ to $\prod_{i=1}^{L} P(y^i)$, i.e., $D\left(P(y^1, y^2, \dots, y^L) \parallel \prod_{i=1}^{L} P(y^i)\right)$

- Two examples
  - $D_{\mathrm{KL}}\left(P(y^1, y^2, \dots, y^L) \parallel \prod_{i=1}^{L} P(y^i)\right) = \sum_{i=1}^{L} H(y^i) - H(y^1, y^2, \dots, y^L)$
  - $D_{\mathrm{MMD}}\left(P\left(f(y^1), f(y^2), \dots, f(y^L)\right) \parallel \prod_{i=1}^{L} P\left(f(y^i)\right)\right)$, f is a transform (e.g., the probability integral transform)[4]

[4] Póczos, Barnabás, Zoubin Ghahramani, and Jeff Schneider. "Copula-based kernel dependency measures." In *International Conference on Machine Learning*, pp. 1635-1642, 2012.

Motivation

- Q1: Is $\sum_{i=1}^{L} H(y^i) - H(y^1, y^2, \dots, y^L)$ the only expression (in mathematical sense)?

- Q2: How to estimate $H(y^i)$ and $H(y^1, y^2, \dots, y^L)$?

- Q3: How to become interpretable and differentiable?

Matrix-based Dependence Measure

- Q1: Is $\sum_{i=1}^{L} H(y^i) - H(y^1, y^2, \dots, y^L)$ the only expression (in mathematical sense)?
  - No !

- Shearer's inequality[5]
  - $H(y^1, y^2, \dots, y^L) \leq \frac{1}{k}\sum_{\mathcal{S}\in\varphi} H(y^i, i \in \mathcal{S})$
    - φ refers to the family of all subsets of $[L]$ with the property that every member of $[L]$ lies in at least k members of φ
    - There are at least $(L-1)$ potential mathematical formulas to quantify the dependence of $y^1, y^2, \dots, y^L$.
  - Two special and simplest cases
    - If $\varphi = \binom{L}{1}$, $H\left(y^1, y^2, \dots, y^L\right) \leq \sum_{i=1}^{L} H(y^i)$,
      - Total correlation $T\left(y^1, y^2, \dots, y^L\right) = \sum_{i=1}^{L} H(y^i) - H(y^1, y^2, \dots, y^L)$.
    - If $\varphi = \binom{L}{L-1}$, $H\left(y^1, y^2, \dots, y^L\right) \leq \frac{1}{L-1}\left[\sum_{i=1}^{L} H(y^1, \dots, y^{i-1}, y^i, \dots, y^L)\right]$,
      - Dual total correlation $D\left(y^1, y^2, \dots, y^L\right) = \sum_{i=1}^{L} H(y^1, \dots, y^{i-1}, y^i, \dots, y^L) - (L-1)H(y^1, y^2, \dots, y^L)$.

[5] Chung, Fan RK, Ronald L. Graham, Peter Frankl, and James B. Shearer. "Some intersection theorems for ordered sets and graphs." *Journal of Combinatorial Theory, Series A* 43, no. 1 (1986): 23-37.

Matrix-based Dependence Measure

Q2: How to estimate $H(y^i)$ and $H(y^1, y^2, \ldots, y^L)$?

Matrix-based Rényi's α-entropy functional[6]

- Entropy of variable y (i.e., $H(y)$) is estimated as:
  - Given $Y = \{y_1, y_2, \ldots, y_N\}$ and a kernel Gram matrix $(K)_{ij} = \kappa(y_i, y_j)$:
  - $\mathbf{S}_\alpha(A) = \frac{1}{1-\alpha} \log[\sum_{i=1}^{N} \lambda_i(A)^\alpha]$, with
    - $A = K / \mathrm{tr}(K)$
    - $\lambda_i(A)$ denotes the i-th eigenvalue of A
  - Measure entropy or uncertainty of data on the eigenspectrum of a Gram matrix in kernel space
  - Independent to dimension of y
  - Avoid density estimation and model training
  - Only two hyper-parameters (α and σ)
    - $\alpha = 1.01$
    - σ (kernel size): many heuristic rules in kernel learning

[6] Shujian Yu, Luis Gonzalo Sanchez Giraldo, Robert Jenssen, and Jose C. Principe. "Multivariate Extension of Matrix-based Renyi's α-order Entropy Functional." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 2960 - 2966, 2020.

Matrix-based Dependence Measure

- Q2: How to estimate $H(y^i)$ and $H(y^1, y^2, \dots, y^L)$?

- Matrix-based Rényi's α-entropy functional[6]
  - Joint entropy of L variables $y^1, y^2, \dots, y^L$(i.e., $H(y^1, y^2, \dots, y^L)$) is estimated as:
    - Given a collection of N samples $\{\boldsymbol{y}_i = (y_i^1, y_i^2, \dots, y_i^L)\}_{i=1}^N$, with $y^1 \in \mathbb{R}^{d_1}$, $y^2 \in \mathbb{R}^{d_2}$, …, $y^L \in \mathbb{R}^{d_L}$
    - Evaluate a kernel Gram matrix for each of the L variables, that is $(A^1)_{ij} = \kappa(y_i^1, y_j^1)$, $(A^2)_{ij} = \kappa(y_i^2, y_j^2)$, …, $(A^L)_{ij} = \kappa(y_i^L, y_j^L)$
    - $\mathbf{S}_\alpha(A^1, A^2, \dots, A^L) = \mathbf{S}_\alpha\left(\frac{A^1 \circ A^2 \circ \cdots \circ A^L}{\mathrm{tr}(A^1 \circ A^2 \circ \cdots \circ A^L)}\right)$, with
      - "∘" denotes the Hadamard product
    - Avoid density estimation and model training
    - Only two hyper-parameters (α and σ)

[6] Shujian Yu, Luis Gonzalo Sanchez Giraldo, Robert Jenssen, and Jose C. Principe. "Multivariate Extension of Matrix-based Renyi's α-order Entropy Functional." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 2960 - 2966, 2020.

Matrix-based Dependence Measure

Q3: How to become interpretable and differentiable?

Interpretability

- Normalization to $[0,1]$ by upper bound
  - $T^*(\boldsymbol{y}) = \frac{\left[\sum_{i=1}^{L} H(y^i)\right] - H\left(y^1, y^2, \ldots, y^L\right)}{\left[\sum_{i=1}^{L} H(y^i)\right] - \max_i H(y^i)}$
  - $D^*(\boldsymbol{y}) = \frac{\left[\sum_{i=1}^{L} H(y^{[L]\setminus i})\right] - (L-1) H\left(y^1, y^2, \ldots, y^L\right)}{H(y^1, y^2, \ldots, y^L)}$
  - $T^*(\boldsymbol{y})$ and $D^*(\boldsymbol{y})$ reduces to 0 iff $y^1, y^2, \ldots, y^L$ are independent

- Additional notes on normalization
  - When do we need normalization?
  - Influence on quantitative performance (e.g., deep neural networks generalization error)
    - *Normalization depends on application and priority on interpretability*
  - When $L = 2$, $T^*(\boldsymbol{y}) = D^*(\boldsymbol{y})$
    - $I^*(\boldsymbol{y}) = \frac{H(y^1) + H(y^2) - H(y^1, y^2)}{\min_i H(y^i)}$: *normalized mutual information*
    - $I^*(\boldsymbol{y}) = \frac{H(y^1) + H(y^2) - H(y^1, y^2)}{\max_i H(y^i)}$: *an alternative form (usually performs better practically)*

Matrix-based Dependence Measure

- Q3: How to become interpretable and differentiable?

- Differentiability
  - Analytical gradient of matrix-based Rényi's α-entropy functional
  - Automatically differentiable with PyTorch (recommend) and Tensorflow

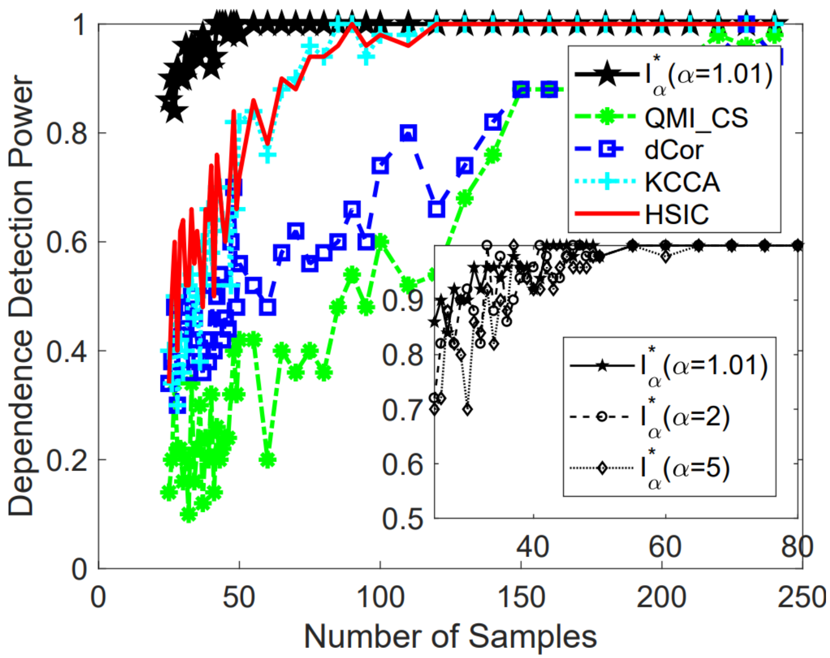
$$\frac{\partial S_\alpha(A)}{\partial A} = \frac{\alpha}{(1-\alpha)} \frac{A^{\alpha-1}}{\operatorname{tr}(A^\alpha)},$$

$$\frac{\partial S_\alpha(A,B)}{\partial A} = \frac{\alpha}{(1-\alpha)} \left[\frac{(A \circ B)^{\alpha-1} \circ B}{\operatorname{tr}(A \circ B)^\alpha} - \frac{I \circ B}{\operatorname{tr}(A \circ B)} \right]$$

$$\frac{\partial I_\alpha(A;B)}{\partial A} = \frac{\partial S_\alpha(A)}{\partial A} + \frac{\partial S_\alpha(A,B)}{\partial A}$$

Ability in detecting (nonlinear) dependence:

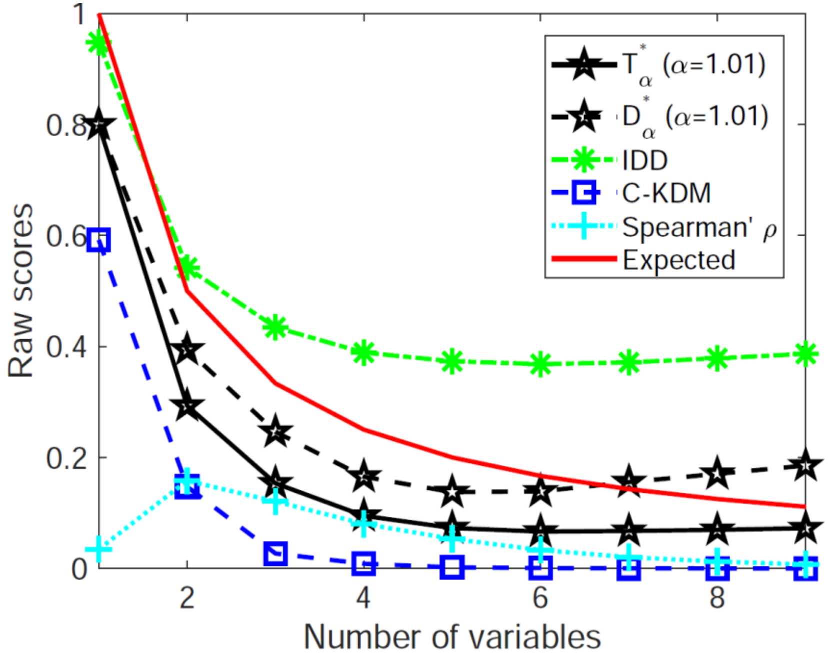
- Example 1.
- $y^1 \sim \mathcal{N}(0, \boldsymbol{I})$
- Each sample in y^2 is generated as $y_i^2 = y_i^1 \varepsilon_i$, ε_i is independent normal variable.
- Non-monotonic dependence between y^1 and y^2.



Matrix-based Dependence Measure

Ability in detecting (nonlinear) dependence:

- Example 2.
- $y^1 = \left(\frac{1}{L-1}\sum_{i=2}^{L} y^i\right)^2$
- $y^2, y^3, \dots, y^L$ are uniformly and independently distributed.
- Decaying dependence with the increase of L.



Applications

Applications

- Gene regulatory network inference
- Robust machine learning under covariate shift and non-Gaussian noise
- Deep deterministic information bottleneck
- …

Applications

Gene Regulatory Network Inference

- Reconstruct gene regulatory network from gene expression data
- Evaluate pairwise dependence on g_i and g_j with $D(g_i; g_j)$
- Only consider undirected graph (dependence is symmetric)

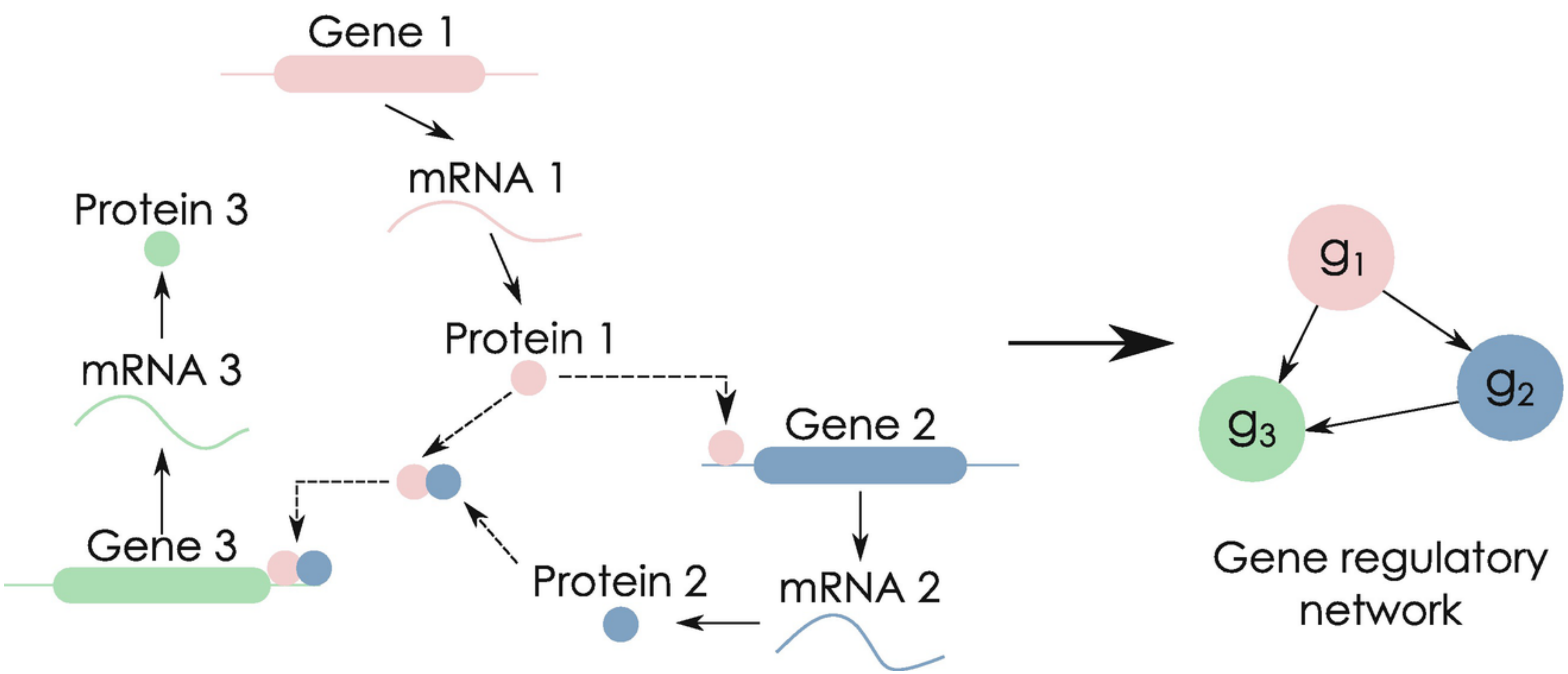


Figure credit to: Sanguinetti, Guido. "Gene regulatory network inference: an introductory survey." In *Gene Regulatory Networks*, pp. 1-23. Humana Press, New York, NY, 2019.

Applications

Gene Regulatory Network Inference

- Reconstruct gene regulatory network from gene expression data
- Evaluate pairwise dependence on g_i and g_j with $D(g_i; g_j)$
- Only consider undirected graph (dependence is symmetric)

Data set	ρ	MI (bin)	MI (KSG)	MIG	I_α^*
Network 1	0.62	0.59	0.74	<u>0.75</u>	**0.78**
Network 2	0.52	0.58	<u>0.76</u>	0.74	**0.87**
Network 3	0.44	0.61	<u>0.83</u>	0.76	**0.84**
Network 4	0.45	0.60	**0.75**	**0.75**	**0.75**
Network 5	0.38	0.61	0.88	<u>0.89</u>	**0.97**

GRN inference results (AUC score) on DREAM4 challenge. The first and second best performances are in bold and underlined, respectively.

Robust Machine Learning under Covariate Shift

- Classification in Distribution Shift

Training Data

Test Data

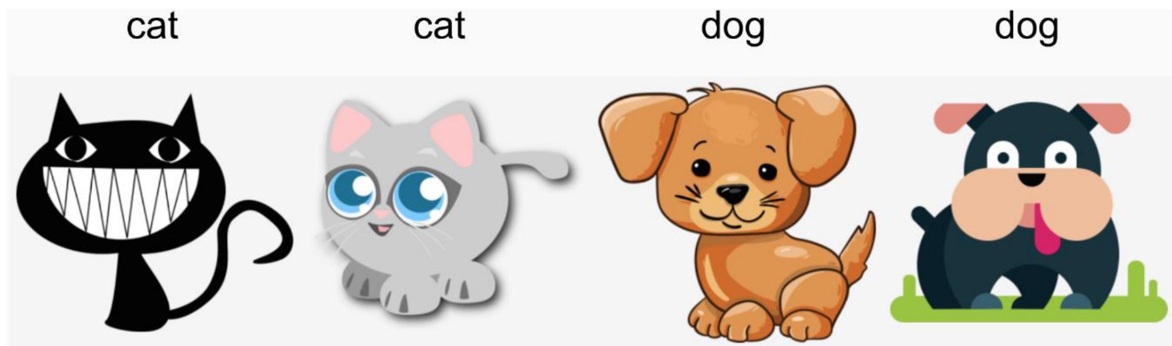
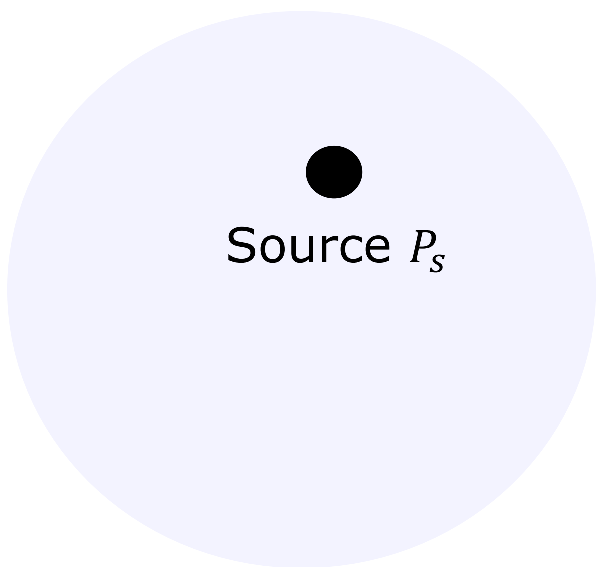


Figure credit to: https://d2l.ai/chapter_multilayer-perceptrons/environment.html

Robust Machine Learning under Covariate Shift

- Classification in Distribution Shift
- How to learn models which are robust to a-priori unknown changes in test distribution?
  - *Source distribution* $P_s(\boldsymbol{x}, y)$
  - *Target distribution* $P_t(\boldsymbol{x}, y) \in Q$



Applications

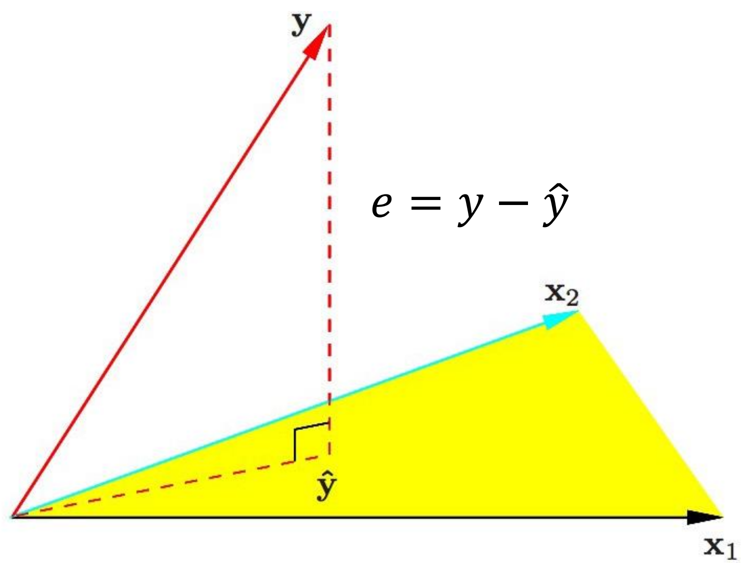
Robust Machine Learning under Covariate Shift

- Classification in Distribution Shift
- How to learn models which are robust to a-priori unknown changes in test distribution?
  - *Source distribution* $P_s(\boldsymbol{x}, y)$
  - *Target distribution* $P_t(\boldsymbol{x}, y) \in Q$
- Covariate Shift
  - $P_s(y|\boldsymbol{x}) = P_t(y|\boldsymbol{x}),\ P_s(\boldsymbol{x}) \neq P_t(\boldsymbol{x})$
  - We want a single model f that works well on all possible $P_t(\boldsymbol{x}, y) \in Q$
  - We cannot use any labeled or unlabeled samples data from $P_t(\boldsymbol{x}, y)$

Applications

Robust Machine Learning under Covariate Shift

- Our Approach
  - Find f such that $y - f(\boldsymbol{x}) \perp \boldsymbol{x}$: the prediction residual is independent to input instances
  - A model is robust against covariate shift iff $y - f(\boldsymbol{x}) \perp \boldsymbol{x}$ [7]
  - $y - f(\boldsymbol{x}) \perp \boldsymbol{x}$ also encourages model is robust against noise in labels: $\tilde{y} = y + e$
- $f^* = \arg\min_f D(y - f(\boldsymbol{x}); \boldsymbol{x})$
- Matrix-based Independence Criterion (MIC)



[7] Greenfeld, Daniel, and Uri Shalit. "Robust learning with the hilbert-schmidt independence criterion." In *International Conference on Machine Learning*, pp. 3759-3768. PMLR, 2020.

Robust Machine Learning under Covariate Shift

- Matrix-based Independence Criterion (MIC)

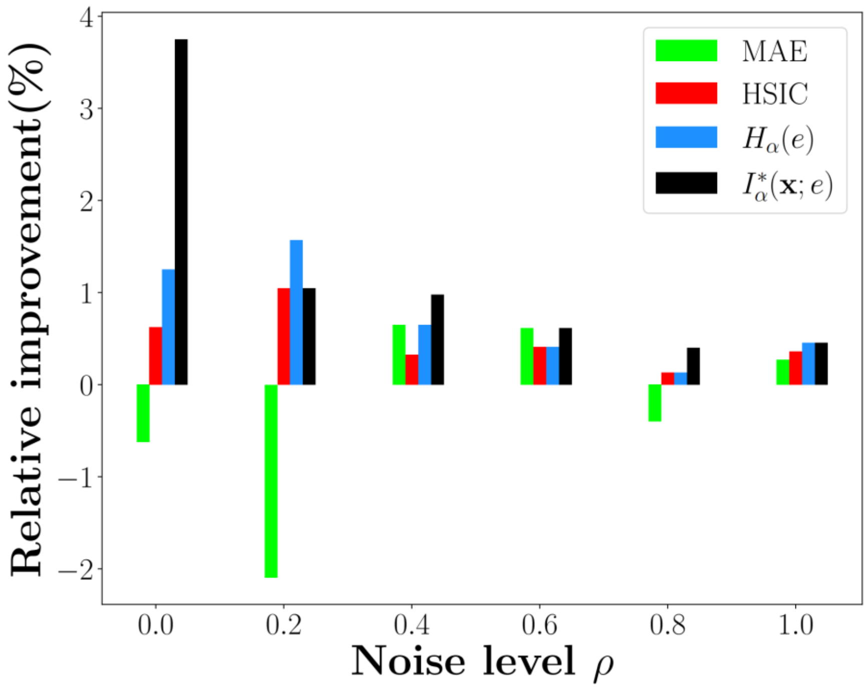
Algorithm 1 *Learning with matrix-based independence criterion (MIC)*

1: **Input:** samples $(\mathbf{x}_i, y_i)_{i=1}^n$, kernel size σ_x for input $\mathbf{x}$ and σ_e for error e, Rényi's entropy order α, mini-batch size m.
2: Initialize neural network parameter θ.
3: **Repeat:**
4: Sample mini-batch $(\mathbf{x}_i, y_i)_{i=1}^m$
5: Evaluate the error for each instances in mini-batch $e_i = y_i - f_\theta(x_i)$
6: Compute the (normalized) Gram matrices of size $m \times m$ for $\{\mathbf{x}_i\}_{i=1}^m$ and $\{e_i\}_{i=1}^m$ (denote them $A_\mathbf{x}$ and A_e, respectively).
7: Compute the normalized Rényi's α-entropy mutual information (i.e., $I_\alpha^*(\mathbf{x}, e)$) based on A_x and A_e with Eq. (6).
8: Update $\theta \leftarrow \text{Optimize}(I_\alpha^*(\mathbf{x}; e))$.
9: **Until** convergence.
10: Compute the estimated source bias: $b \leftarrow \frac{1}{n}\sum_{i=1}^n y_i - \frac{1}{n}\sum_{i=1}^n f_\theta(\mathbf{x}_i)$
11: **Outputs** $f(\mathbf{x}) = f_\theta(\mathbf{x}) + b$.

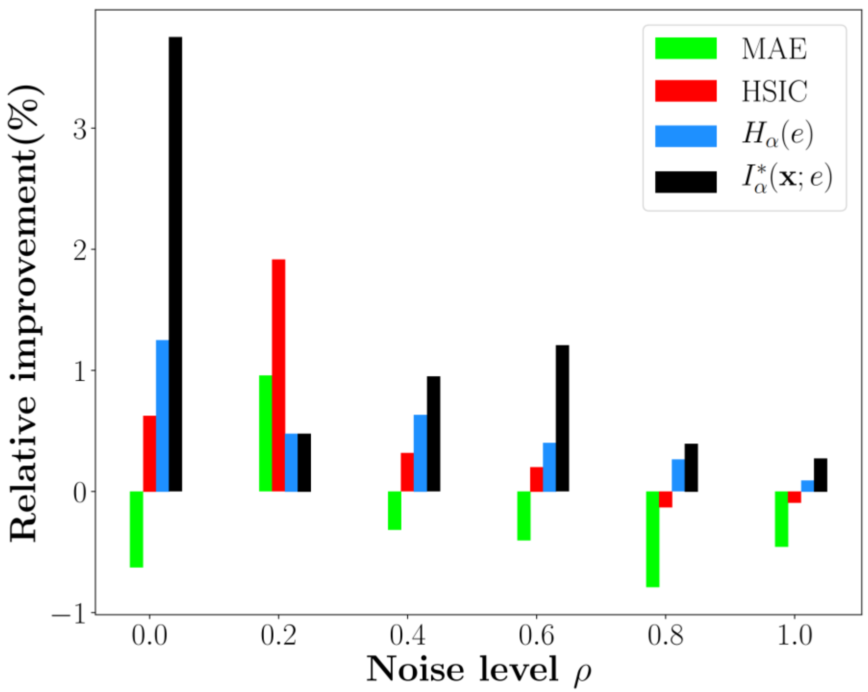
Applications

Robust Machine Learning under Covariate Shift

- Predict the number of hourly bike rentals in Porto (a Kaggle challenge[8])
- First 3 seasons as training data, last season as test data



Laplacian noise



Gaussian noise

[8] https://www.kaggle.com/c/bike-sharing-demand/

Applications

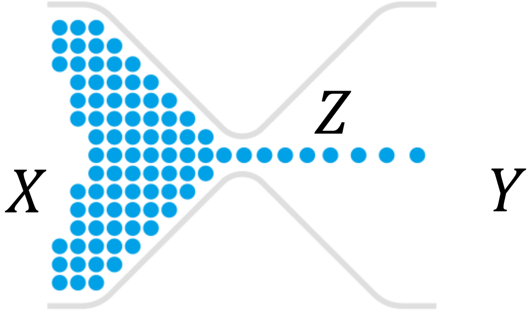
Representation Learning with Information Bottleneck

- Given input X and task Y, learn a useful representation Z or $p(z|x)$
- Information bottleneck principle[9]

$$\max_{p(z|x)} I(Z;Y) \quad \text{s.t.}, I(Z;X) \leq \alpha$$

$$\max_{p(z|x)} I(Z;Y) - \beta I(Z;X)$$

- Z is the trade-off between *sufficiency* and *minimality*
  - Sufficiency
    - Z contains all information regarding Y that can be obtained also from X
  - Minimality
    - Z contains only relevant information regarding Y, but least information from X



A representation Z that is maximally expressive about Y while being maximally compressive about X

[9] Tishby, Naftali, Fernando C. Pereira, and William Bialek. "The information bottleneck method.“ In *Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing*, pp. 368-377, 2000.

Applications

Representation Learning with Information Bottleneck

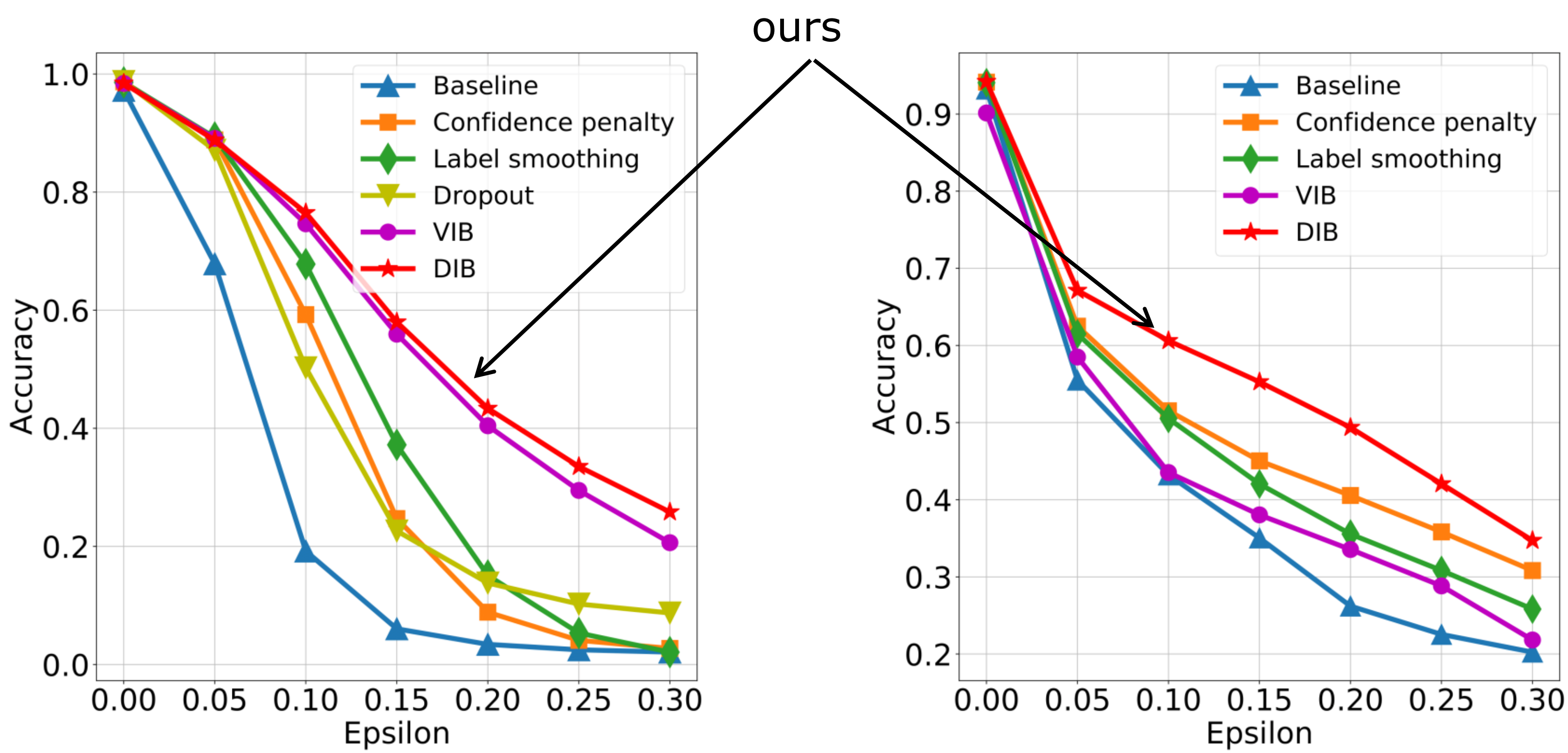
- Neural Network parameterization of IB
  - $J = \max_{\theta} I(Z;Y) - \beta I(Z;X)$
- An alternative formulation of $\max_{\theta} I(Z;Y)$ in deep learning
  - $\max_{\theta} I(Z;Y) \Leftrightarrow \min_{\theta} H(\hat{Y};Y)$, the cross-entropy loss

- Deep Information Bottleneck
  - $J = \min_{\theta} H(\hat{Y};Y) + \beta I(Z;X)$
  - How to estimate $I(Z;X)$?
    - Neural Mutual Information Estimator (MINE)
    - Variational lower bound
    - Matrix-based Dependence Measure (MDM)

Applications

Representation Learning with Information Bottleneck

- Evaluation on generalization and robustness
  - $J = \min_{\theta} I(\hat{Y}; Y) + \beta I(Z; X)$



(a) MLP on MNIST

(b) VGG-16 on CIFAR-10

Conclusions

New Dependence Measure based on Rényi's α-entropy

- Easy to estimate
  - Avoid density estimation
  - Avoid model training
- Statistically more powerful than most of existing ones (e.g., HSIC)
- Automatically differentiable (deep neural networks training)
  - Robust learning under covariate shift
  - Deep information bottleneck
- Applicable to different scenarios (insensitive to # variables and *dim*. of variables) and problems (e.g., bioinformatics, neuroscience, economics, etc.)

- Limitation
  - Given L variables and N samples, complexity is nearly $\mathcal{O}(LN^2) + \mathcal{O}(N^3)$.
  - Taking subsamples can significantly reduce complexity with negligible performance loss (more discussion in supplementary material).

Conclusions

New Dependence Measure based on Rényi's α-entropy

- Resource and contact information

GitHub

WeChat

\Orchestrating a brighter world

NEC