Constructing novel nonparametric estimators for information-theoretic measure estimation.

AI and Quantum Information Applications
in Fundamental Physics workshop
Konjiam Resort
2023. 2. 14 (Tue.)

Yung-Kyun Noh

Hanyang University &

Korea Institute for Advanced Study

From Neil Lawrence's Blog

Role of Deep Structure

- Review process in NeurIPS is conducted through blindly.

Reviewer question:

I. Introduction
1) Limited number of papers have investigated deep networks from a theoretical perspective?? I disagree with this statement, many papers from Bengio's group do just this.

...

Author rebuttal:

We are aware of the theoretical work from Bengio's lab. What we meant is that, despite those earlier contributions, we feel that many questions remain open. Much of the available theoretical work is either too narrow (showing the efficiency of deep networks for very restricted classes of functions) or focusses on models that are not as widely used in practice. Our work looks at a family of models that is presently widely used in industry and academia. Furthermore, it shows the benefits of depth for the representation of a rather large family of functions (with certain types of symmetries or invariances).

Contents

- Modification of objective functions for producing a better prediction accuracy
  - Identifying additional b-jets
  - Autoencoder for anomaly detection

- Nonparametric estimation of *f*-divergences
  - Methods of constructing estimators
  - Feature selection
  - Metric learning

Eur. Phys. J. Plus (2022) 137:870
https://doi.org/10.1140/epjp/s13360-022-03024-8

THE EUROPEAN PHYSICAL JOURNAL PLUS

Regular Article

Learning to increase matching efficiency in identifying additional b-jets in the $t\bar{t}b\bar{b}$ process

Cheongjae Jang[1], **Sang-Kyun Ko**[2], **Jieun Choi**[3], **Jongwon Lim**[3], **Yung-Kyun Noh**[1,2,4], **Tae Jeong Kim**[3,a]

[1] A.I. Institute, Hanyang University, Seoul 04763, Republic of Korea
[2] Department of Computer Science, Hanyang University, Seoul 04763, Republic of Korea
[3] Department of Physics, Hanyang University, Seoul 04763, Republic of Korea
[4] Korea Institute for Advanced Study, Seoul 02455, Republic of Korea

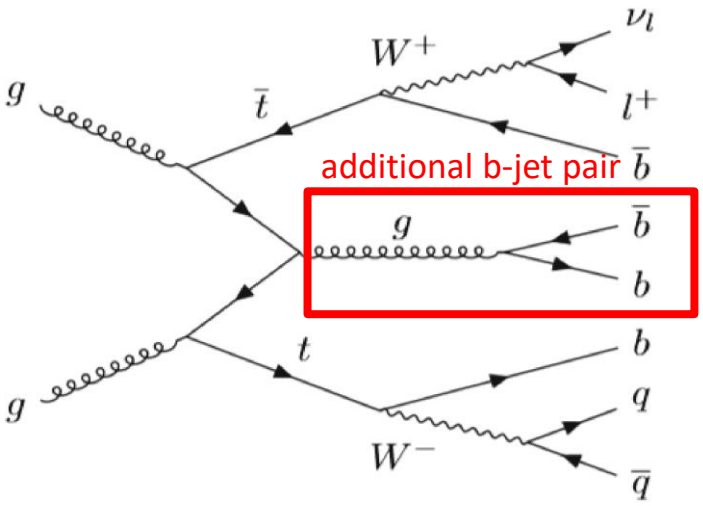
Received: 6 July 2021 / Accepted: 29 June 2022

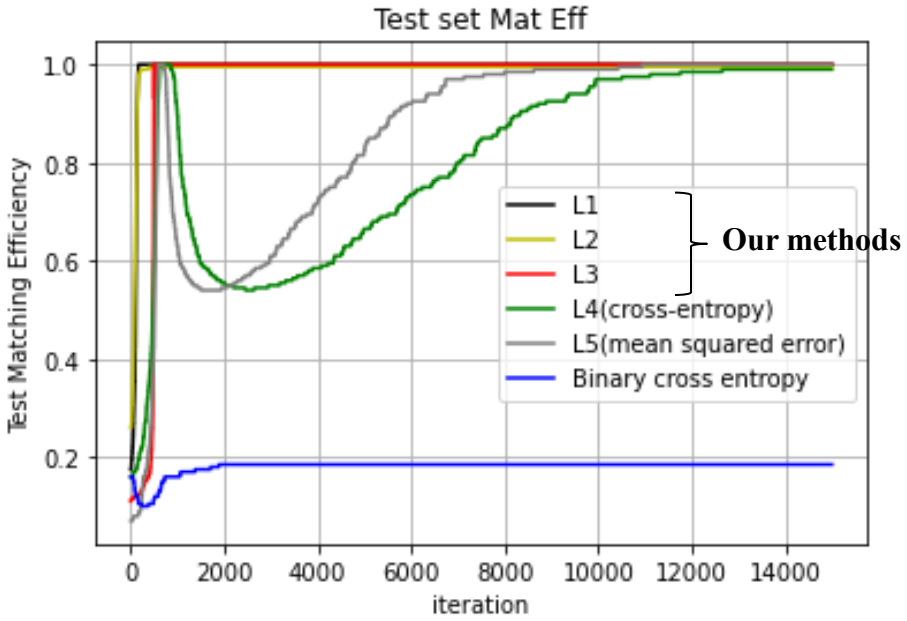
Dr. Cheongjae Jang

Abstract The $t\bar{t}H(b\bar{b})$ process is an essential channel in revealing the Higgs boson properties; however, its final state has an irreducible background from the $t\bar{t}b\bar{b}$ process, which produces a top quark pair in association with a b quark pair. Therefore, understanding the $t\bar{t}b\bar{b}$ process is crucial for improving the sensitivity of a search for the $t\bar{t}H(b\bar{b})$ process. To this end, when measuring the differential cross section of the $t\bar{t}b\bar{b}$ process, we need to distinguish the b-jets originating from top quark decays and additional b-jets originating from gluon splitting. In this paper, we train deep neural networks that identify the additional b-jets in the $t\bar{t}b\bar{b}$ events under the supervision of a simulated $t\bar{t}b\bar{b}$ event data set in which true additional b-jets are indicated. By exploiting the special structure of the $t\bar{t}b\bar{b}$ event data, several loss functions are proposed and minimized to directly increase matching efficiency, i.e., the accuracy of identifying additional b-jets. We show that, via a proof-of-concept experiment using synthetic data, our method can be more advantageous for improving matching efficiency than the deep learning-based binary classification approach presented in [1]. Based on simulated $t\bar{t}b\bar{b}$ event data in the lepton+jets channel from pp collision at $\sqrt{s} = 13$ TeV, we then verify that our method can identify additional b-jets more accurately: compared with the approach in [1], the matching efficiency improves from 62.1% to 64.5% and from 59.9% to 61.7% for the leading order and the next-to-leading order simulations, respectively.

Advanced Study

Finding Additional b-jet Using Matching Efficiency





Identify **additional b-jets** in the $t\bar{t}b\bar{b}$ events by optimizing **Matching Efficiency**

$$\frac{1}{N}\sum_{i=1}^{N}\delta(y_i,\widehat{y}(M_i)) \qquad M_i \in \mathbb{R}^{c_i \times F}$$

← an event

Structure of Data

- One event includes one additional b-jet

b-jet pair 6_{M1}	0
b-jet pair 2_{M1}	0
b–jet pair 5_{M1}	1
b–jet pair 1_{M1}	0
⋮	

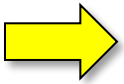
M_1

F features

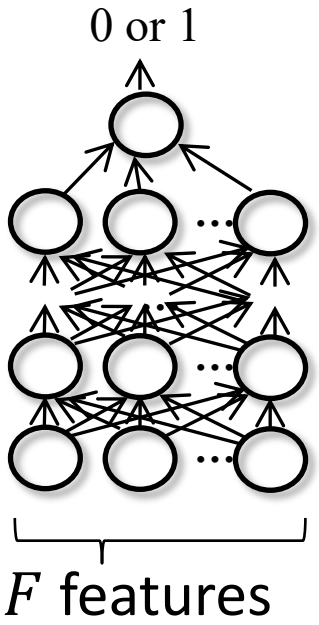
b-jet pair 4_{M2}	0
b-jet pair 1_{M2}	0
b–jet pair 6_{M2}	0
b–jet pair 2_{M2}	1
⋮	

M_2

⋮

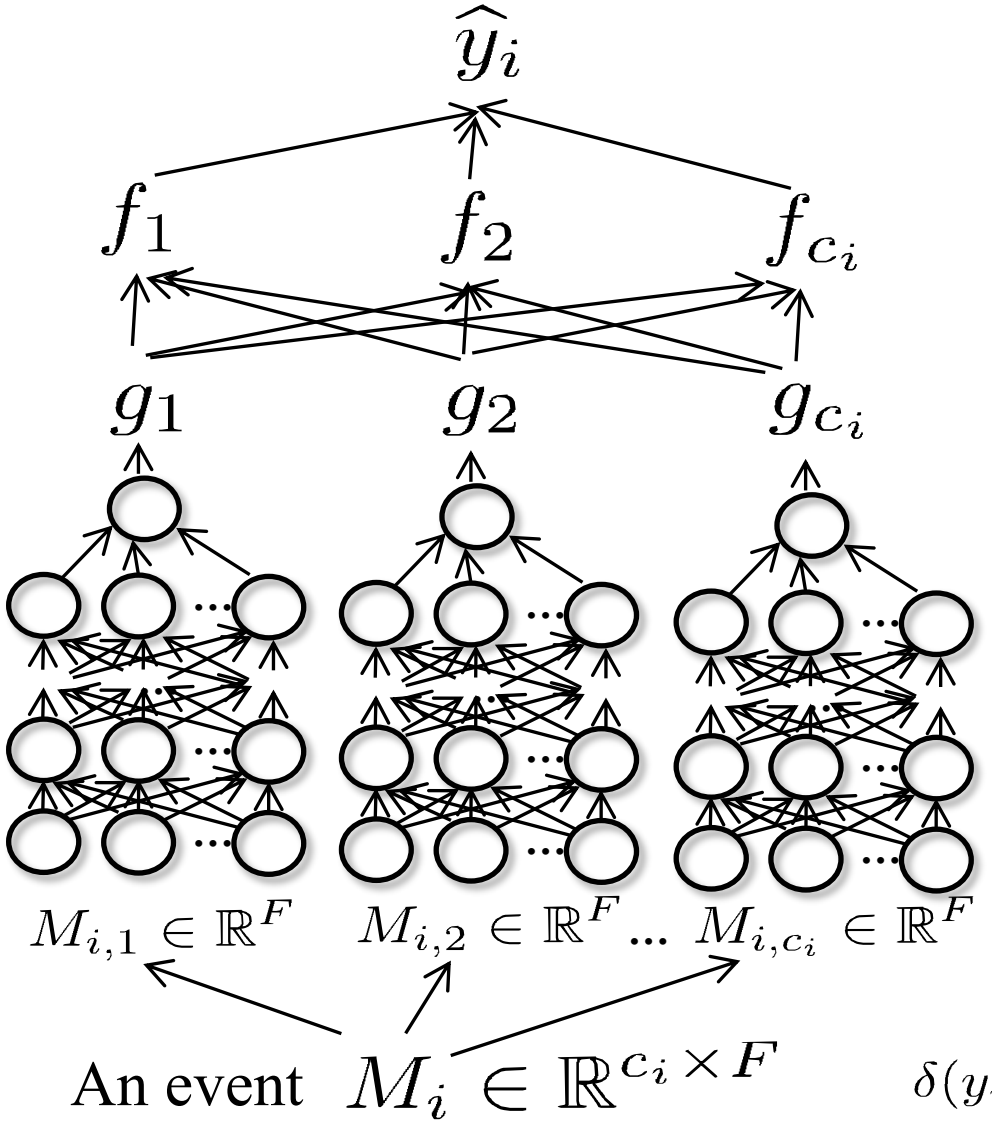


b-jet pair 6_{M1}	0
b-jet pair 4_{M2}	0
b-jet pair 2_{M1}	0
b-jet pair 1_{M2}	0
b–jet pair 5_{M1}	1
b–jet pair 1_{M1}	0
b–jet pair 2_{M2}	1
b–jet pair 6_{M2}	0
b–jet pair 4_{M1}	0
b–jet pair 3_{M1}	0
b–jet pair 5_{M2}	0
b–jet pair 3_{M2}	0
⋮	



"No information about the structure of data"

Identifying Additional b-jets



- An event:

$$M_i \in \mathbb{R}^{c_i \times F}$$

c_i: # of b-jet pairs in event

F: # of features in b-jet pair

- Data

$$\mathcal{D} = \{M_i, y_i\}_{i=1}^{N_E}$$

$$y_i \in \{1, \ldots, c_i\}$$

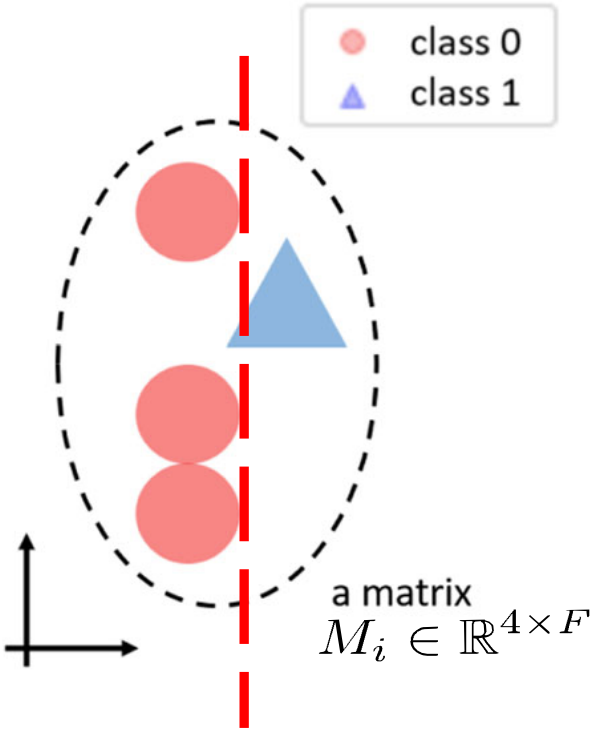
- Prediction

$$f_j(M_i) = \frac{\exp(g_j(M_i))}{\sum_{k=1}^{c_i} \exp(g_k(M_i))}$$

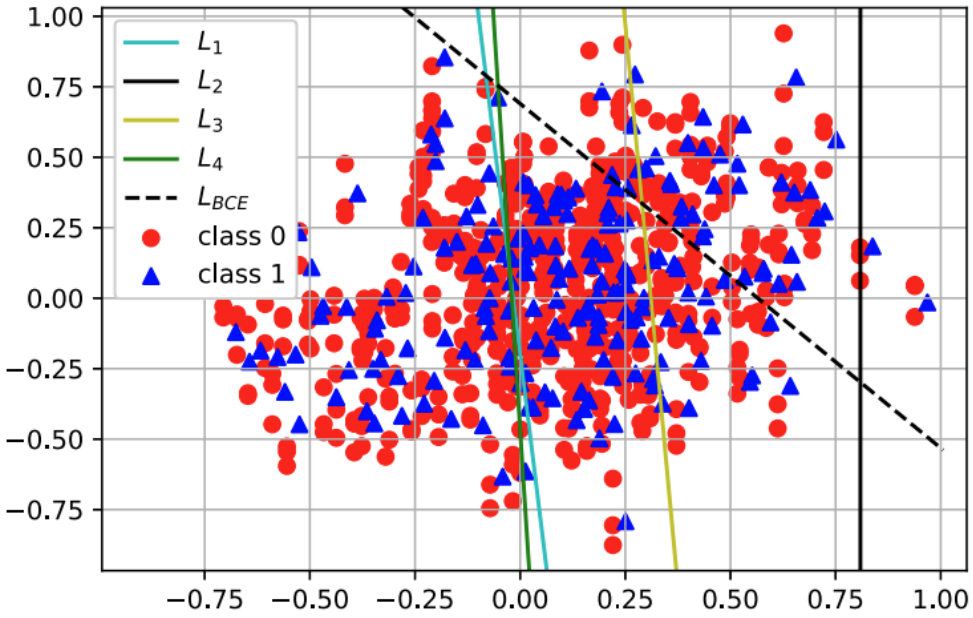
$$\delta(y_i, \widehat{y}(M_i)) = f_{y_i}(M_i) - \sum_{j \neq y_i} f_j(M_i)$$

Structure of Data

- One event includes one additional jet



(a) An example matrix



(b) Trained models

Autoencoding Under Normalization Constraints

Sangwoong Yoon [1] **Yung-Kyun Noh** [2 3] **Frank C. Park** [1 4]

Abstract

Likelihood is a standard estimate for outlier detection. The specific role of the normalization constraint is to ensure that the out-of-distribution (OOD) regime has a small likelihood when samples are learned using maximum likelihood. Because autoencoders do not possess such a process of normalization, they often fail to recognize outliers even when they are obviously OOD. We propose the Normalized Autoencoder (NAE), a normalized probabilistic model constructed from an autoencoder. The probability density of NAE is defined using the reconstruction error of an autoencoder, which is differently defined in the conventional energy-based model. In our model, normalization is enforced by suppressing the reconstruction of negative samples, significantly improving the outlier detection performance. Our experimental results confirm the efficacy of NAE, both in detecting outliers and in generating in-distribution samples.

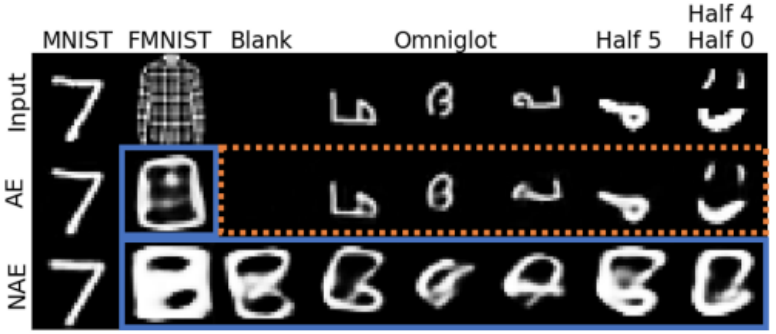


Figure 1. Examples of reconstructed outliers. The last two rows show the reconstructions from a conventional autoencoder (AE) and NAE. Both autoencoders are trained on MNIST, and other inputs are outliers. The architecture of the two autoencoders is identical. Successful detection of an outlier is highlighted with blue solid rectangles, while detection failures due to the reconstruction of outliers are denoted with an orange dotted rectangle. Note that AE is not the identity mapping, as it fails to reconstruct the shirt.

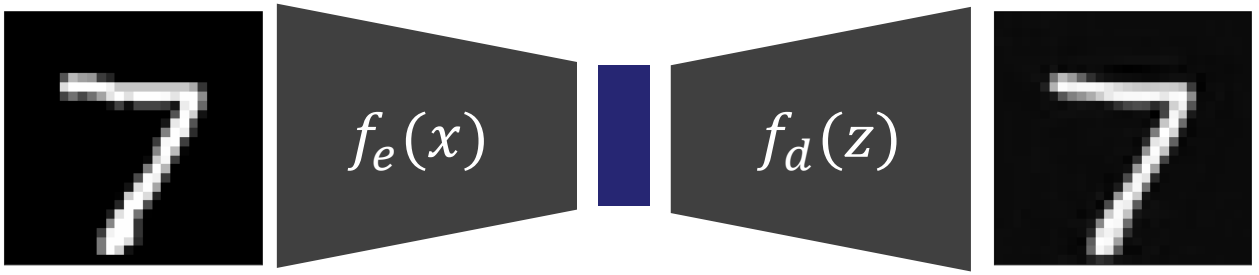
outliers consistently across a wide range of experimental settings (Lyudchik, 2016; Tong et al., 2019; Zong et al., 2018; Gong et al., 2019). We name this phenomenon *outlier reconstruction*. Figure 1 shows examples of some outliers

Autoencoder and Outlier Detection

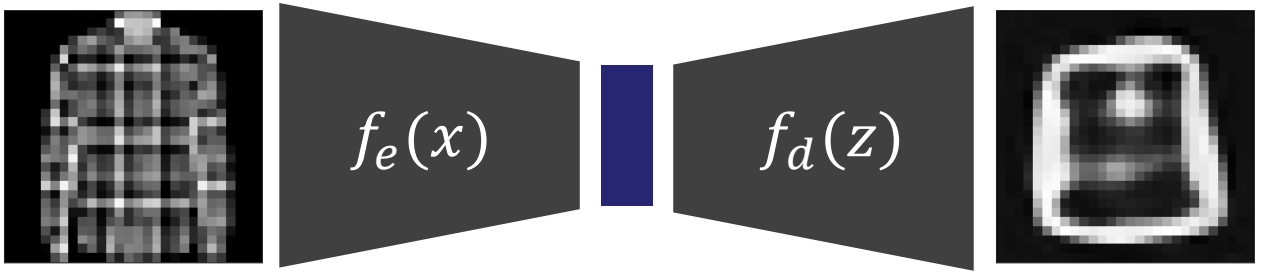
Reconstruction Error

$$l_\theta(x) = \|x - \tilde{x}\|^2$$

x $\qquad\qquad$ $\tilde{x} = f_d(f_e(x))$



Small $l_\theta(x)$ "Inlier"



Large $l_\theta(x)$ "Outlier"

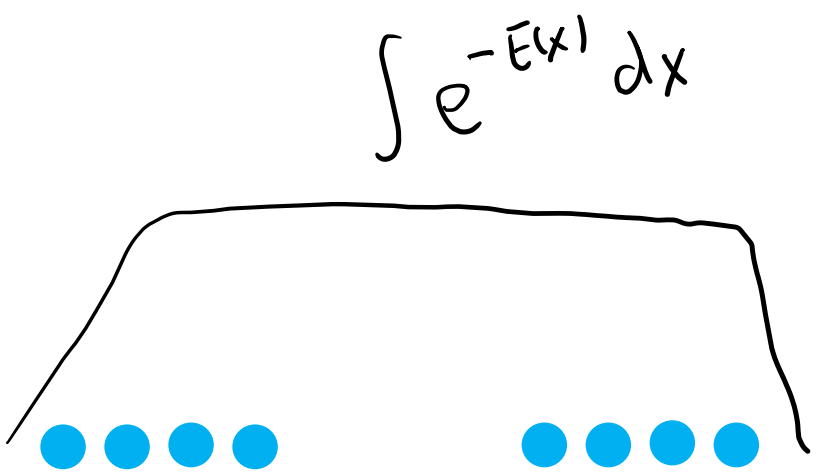
$$p_\theta(x) = \frac{1}{\Omega_\theta} e^{-l_\theta(x)}, \qquad \Omega_\theta = \int e^{-l_\theta(x)} dx$$

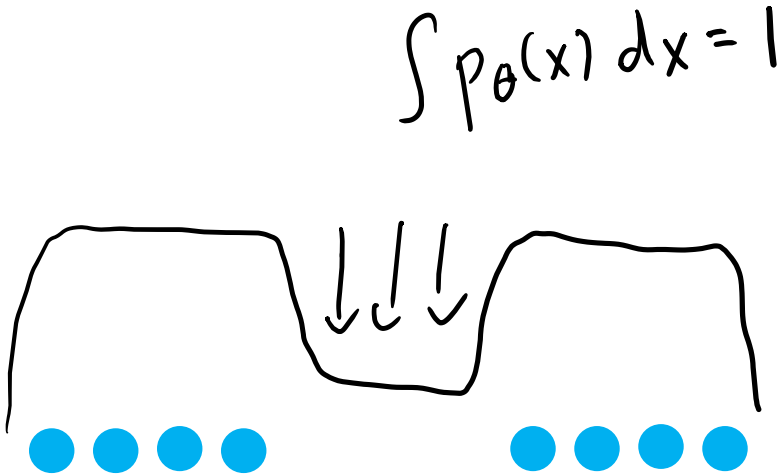
$$\min_\theta \mathbb{E}_{x\sim p(x)}[l_\theta(x)] + \log \Omega_\theta$$

Sangwoong Yoon

$$p_\theta(x) = \frac{1}{\Omega_\theta} e^{-l_\theta(x)}, \qquad \Omega_\theta = \int e^{-l_\theta(x)} dx$$

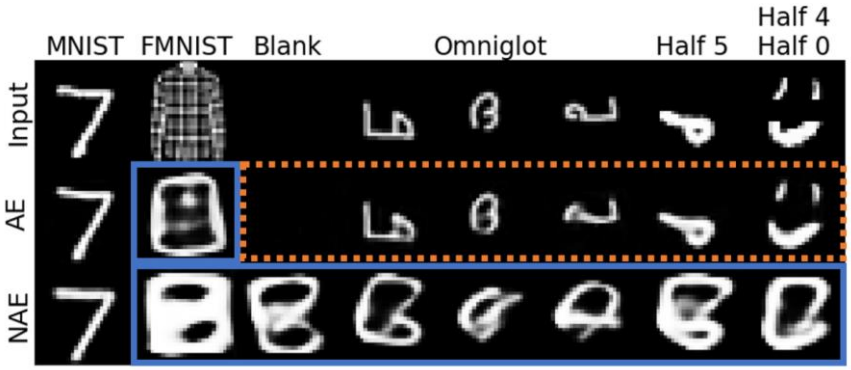
$$\min_\theta \mathbb{E}_{x\sim p(x)}[l_\theta(x)] + \log \Omega_\theta$$





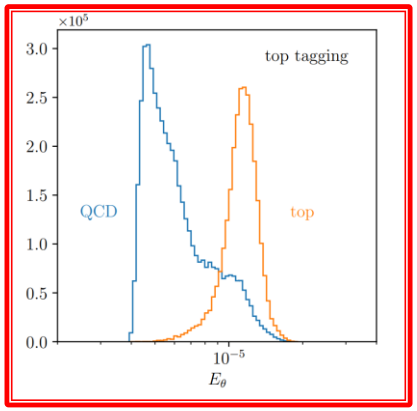
Autoencoding Under Normalization (NAE)

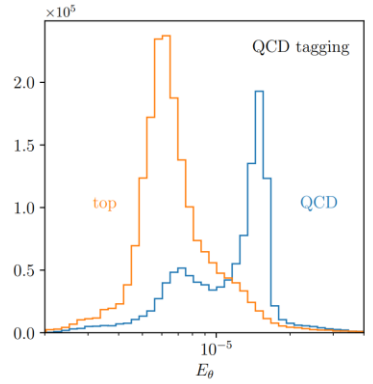
- Preventing the reconstruction of the outliers in autoencoders



Yoon, S., Noh, Y.-K., and Park, F.C. (2021), Autoencoding Under Normalization Constraints, *International Conference on Machine Learning (ICML)*

Among the background events (QCD), find anomalous jet (top jet)

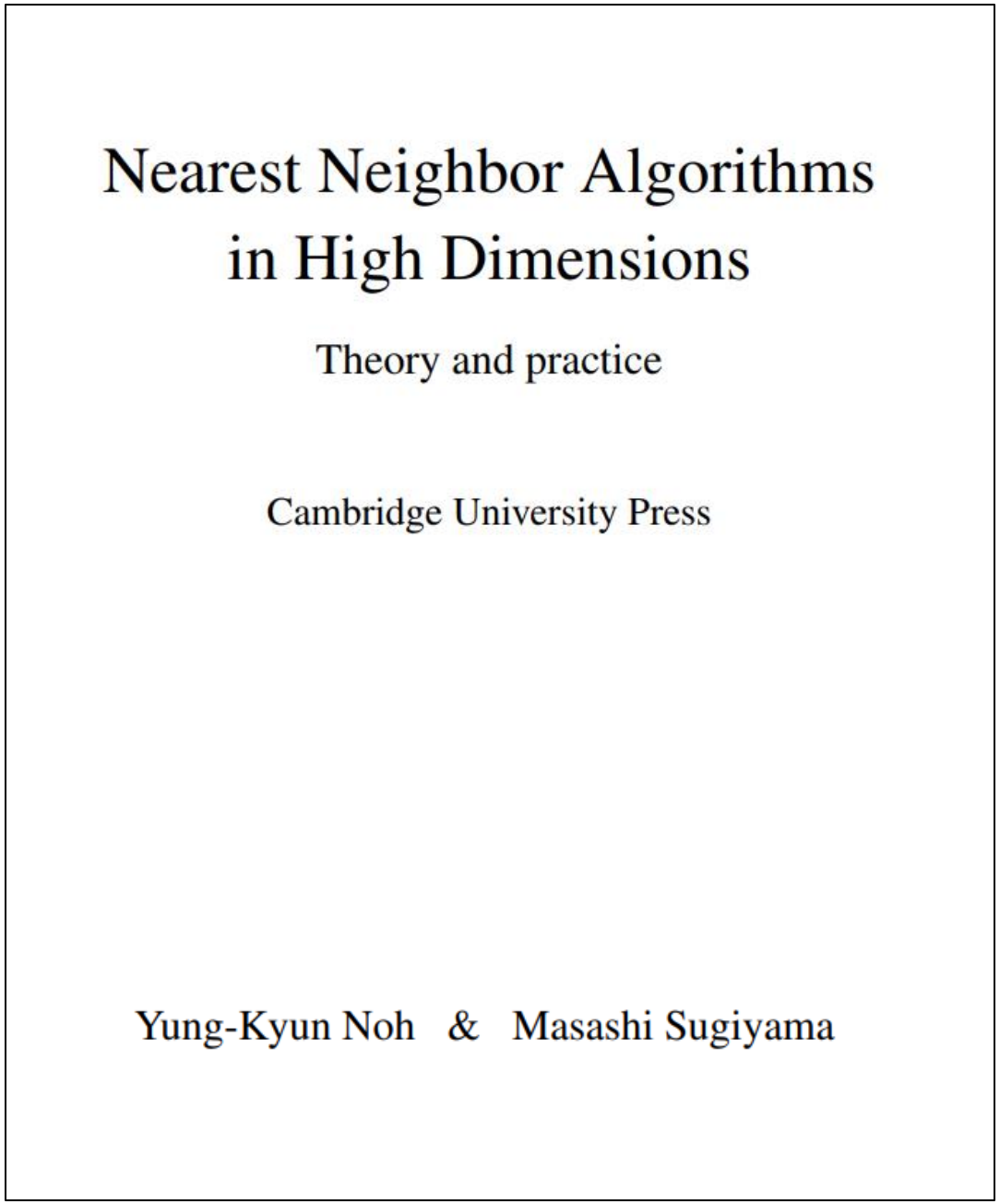




Signal	NAE AUC	NAE $\epsilon_B^{-1}(\epsilon_S = 0.2)$	AE [1] AUC	DVAE [6] AUC
top (AE)	0.875	68	0.89	0.87
top (NAE)	0.91	80		
QCD (AE)	0.579	12	–	0.75
QCD (NAE)	0.89	350		

Barry Dillon et al. (2022) A Normalized Autoencoder for LHC Triggers, *arXiv 2206.14225 [hep-ph]* [Tilman Plehn group]

f-divergences


Nearest Neighbor Algorithms
in High Dimensions
Theory and practice
Cambridge University Press
Yung-Kyun Noh & Masashi Sugiyama

$$i_1^*, \ldots, i_d^* = \arg \max_{i_1,\ldots,i_d \in \{1,\ldots,D\}} \widehat{J}_{JS}(x_{i_1}, \ldots, x_{i_d}; y)$$

Metric Learning

Feature Selection

$$\mathbf{z} = L^\top \mathbf{x}, \quad \mathbf{x}, \mathbf{z} \in \mathbb{R}^D$$

$$d = \sqrt{(\mathbf{z}_1 - \mathbf{z}_2)^2}$$

$$= \sqrt{(\mathbf{x}_1 - \mathbf{x}_2)^\top A (\mathbf{x}_1 - \mathbf{x}_2)}$$

$$A = LL^\top$$

f-divergences

Regularization, Objective function

$$\theta^* = \arg\min_\theta \widehat{L}(\theta; X, Y) - \lambda \widehat{MI}(Y; m|X; \theta)$$

$$\theta^* = \arg\min_\theta \widehat{L}(\theta; X, Y) - \lambda \widehat{H}(Y|X; \theta)$$

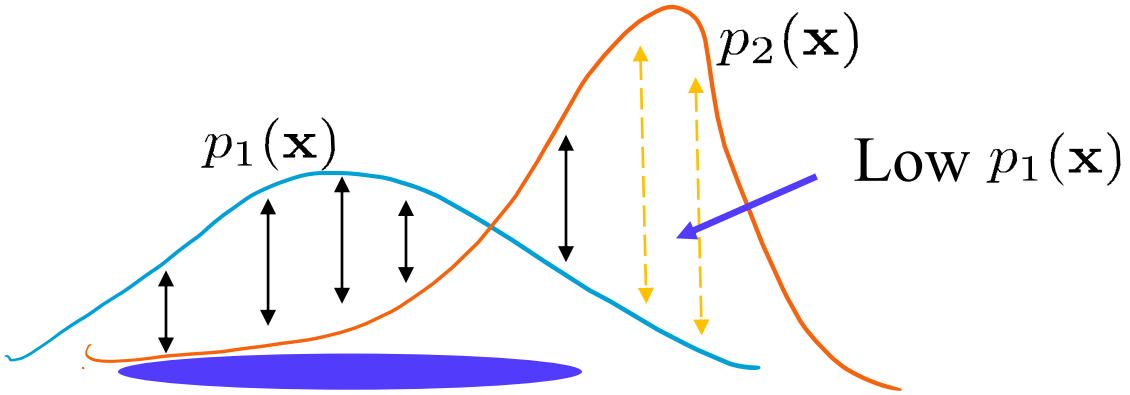
$$\psi* = \arg\min_\psi \widehat{D_{\mathbf{KL}}}(q_\psi(\theta), p(\theta|Data))$$

f-divergences

$$D_f(p_1(\mathbf{x}), p_2(\mathbf{x})) = \int p_1(\mathbf{x}) f\left(\frac{p_2(\mathbf{x})}{p_1(\mathbf{x})}\right) d\mathbf{x}$$

$$D_f(p_1(\mathbf{x}), p_2(\mathbf{x})) = \int p_1(\mathbf{x}) f\left(\frac{p_2(\mathbf{x})}{p_1(\mathbf{x})}\right) d\mathbf{x}$$

f-divergences



The functional definition looks asymmetric, but it contains symmetric functions

$$D_f(p_1(\mathbf{x}), p_2(\mathbf{x})) = \int p_1(\mathbf{x}) f\left(\frac{p_2(\mathbf{x})}{p_1(\mathbf{x})}\right) d\mathbf{x}$$

f-divergences

KL-divergence

$$D_{KL}(p_1(\mathbf{x}), p_2(\mathbf{x})) = -\int p_1(\mathbf{x}) \log\left(\frac{p_2(\mathbf{x})}{p_1(\mathbf{x})}\right) d\mathbf{x}$$

$$f(t) = -\log t$$

The functional definition looks asymmetric,
but it contains symmetric functions

$$D_f(p_1(\mathbf{x}), p_2(\mathbf{x})) = \int p_1(\mathbf{x}) f\left(\frac{p_2(\mathbf{x})}{p_1(\mathbf{x})}\right) d\mathbf{x}$$

f-divergences

JS-divergence

$$p_i(\mathbf{x}) = p(\mathbf{x}|y=i)$$

$$D_{JS}(p_1(\mathbf{x}), p_2(\mathbf{x})) = -\sum_{y\in\{1,2\}} \int p(\mathbf{x}, y) \log\left(\frac{p(\mathbf{x})P(y)}{p(\mathbf{x}, y)}\right) d\mathbf{x}$$

$$= \frac{1}{2}\left(D_{KL}\left(p_1(\mathbf{x}), \frac{p_1(\mathbf{x}) + p_2(\mathbf{x})}{2}\right) + D_{KL}\left(p_2(\mathbf{x}), \frac{p_1(\mathbf{x}) + p_2(\mathbf{x})}{2}\right)\right)$$

$$f(t) = -\frac{1}{2}\log\frac{1+t}{2} - \frac{t}{2}\log\frac{1+t}{2t}$$

$$D_f(p_1(\mathbf{x}), p_2(\mathbf{x})) = \int p_1(\mathbf{x}) f\left(\frac{p_2(\mathbf{x})}{p_1(\mathbf{x})}\right) d\mathbf{x}$$

f-divergences

Condition:

$f(.)$: convex

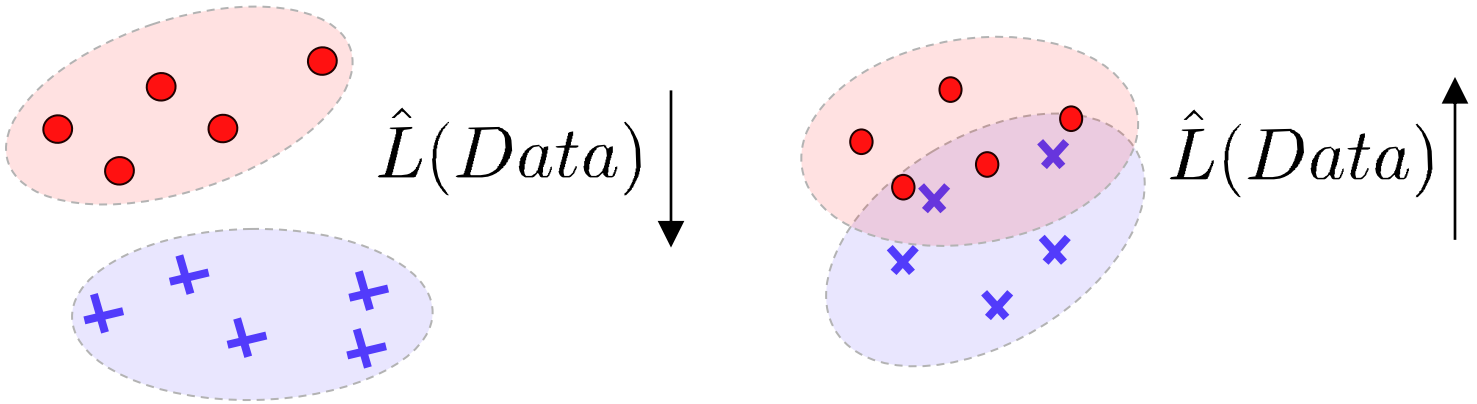
$D_f(p_1(\mathbf{x}), p_2(\mathbf{x}))$

is minimized when

$p_1(\mathbf{x}) = p_2(\mathbf{x})$ for all $\mathbf{x}$

$$D_f(p_1(\mathbf{x}), p_2(\mathbf{x})) = \int p_1(\mathbf{x}) f\left(\frac{p_2(\mathbf{x})}{p_1(\mathbf{x})}\right) d\mathbf{x}$$

f-divergences



Similar to Loss, but Not the Same

- Invariant to the coordinate transformation once the dimensionality is conserved.

$$\mathbf{z} = T(\mathbf{x}) \ , \quad \mathbf{z}, \mathbf{x} \in \mathbb{R}^D$$

$$\int p_1(\mathbf{x}) f\left(\frac{p_2(\mathbf{x})}{p_1(\mathbf{x})}\right) d\mathbf{x} = \int p_1(\mathbf{z}) \cdot j(\mathbf{x}) f\left(\frac{p_2(\mathbf{z}) \cdot j(\mathbf{x})}{p_1(\mathbf{z}) \cdot j(\mathbf{x})}\right) \frac{d\mathbf{z}}{j(\mathbf{x})}$$

$$= \int p_1(\mathbf{z}) f\left(\frac{p_2(\mathbf{z})}{p_1(\mathbf{z})}\right) d\mathbf{z} \qquad j(\mathbf{x}) = \left|\frac{dT}{d\mathbf{x}}\Big|_{\mathbf{x}}\right|$$

- Eliminate the property obtained by coordinate transformation (← Different from Loss)
- It must capture the difference in terms of the information between underlying densities, not of the choice of the coordinate.

Role of Loss

- $\hat{L}(Data)$: Empirical loss
  - Consider two separate roles
    - (1) Find the true posterior for given features
    - (2) Improve the features (or representations)

- Optimizing *f*-divergence encompasses the result of the first role of optimizing $\hat{L}(Data)$

- Use *f*-divergence in case we are interested in (2) without performing (1).

Loss function

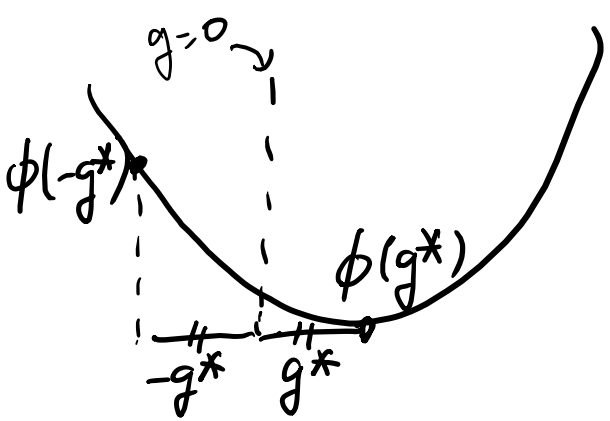
$$\widehat{L}(g) = \frac{1}{N}\sum_{i=1}^{N}\phi(y_i, g(\mathbf{x}_i))$$

Margin-based

$\phi(y, g(x)) = \phi(yg(x))$

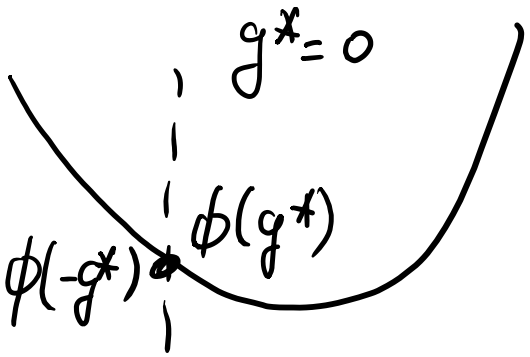
$y \in \{-1, 1\}$

- Optimal g with convex ϕ



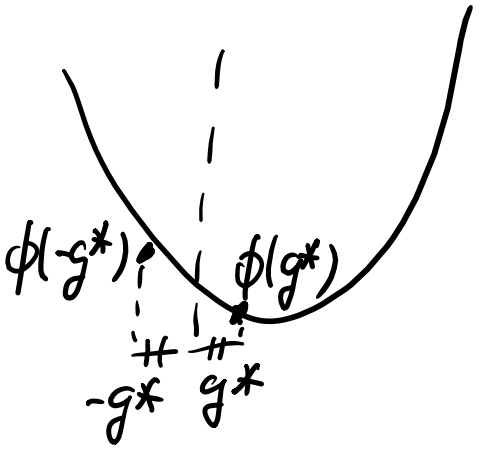
$P(y=1|\mathbf{x}) = 1$

$P(y=-1|\mathbf{x}) = 0$



$P(y=1|\mathbf{x}) = 0$

$P(y=-1|\mathbf{x}) = 1$



$P(y=1|\mathbf{x}) = 0.3$

$P(y=-1|\mathbf{x}) = 0.7$

Ex) Logistic Loss (Cross Entropy)

$$\phi(\alpha) = \log(1+\exp(-\alpha)) \qquad \alpha = y\,g(x)$$

Expectation at x:

$$E[L(g(x))] = P(y=1|x)\,\phi(g(x)) + P(y=-1|x)\,\phi(-g(x))$$

$$= \frac{p_1}{p_1+p_{-1}}\log(1+\exp(-g)) + \frac{p_{-1}}{p_1+p_{-1}}\log(1+\exp(g))$$

$$\frac{d}{dg}E[L] = 0 \quad \Rightarrow \quad g^* = \log\frac{p_{-1}}{p_1}$$

Plug-in

$$E[L] = E_{y_i}\left[\frac{1}{N}\sum_{i=1}^{N} \log\left(1+\exp\left(-y_i g^*(x_i)\right)\right)\right]$$

$$= E_{y_i}\left[\frac{1}{N}\sum_{i=1}^{N} \log\left(1+\exp\left(-y_i \log\frac{p_1(x_i)}{p_{-1}(x_i)}\right)\right)\right]$$

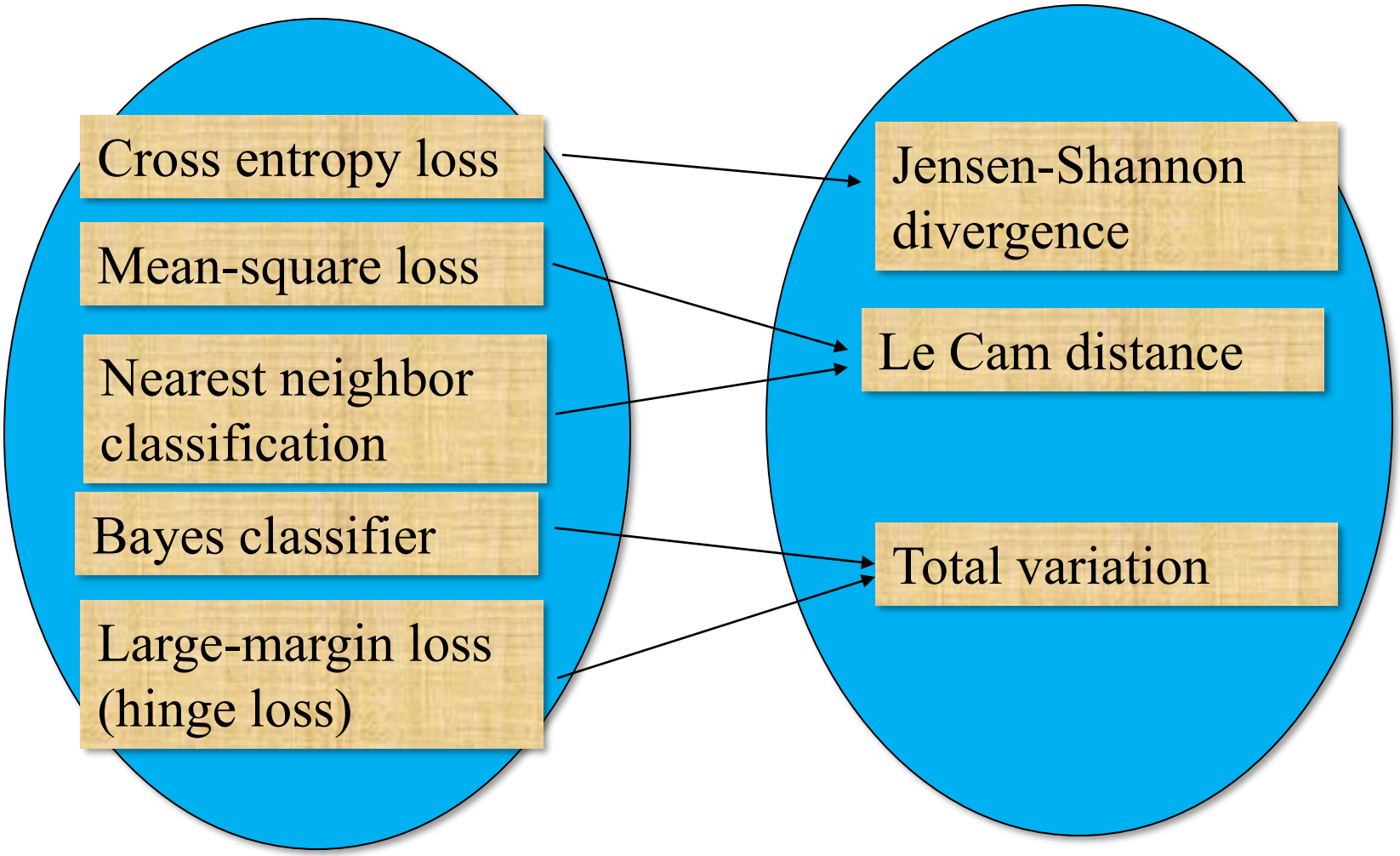
$$\approx \frac{1}{N}\sum_{i=1}^{N} \frac{p_1}{p_1+p_{-1}} \log\left(1+\frac{p_{-1}}{p_1}\right) + \frac{p_{-1}}{p_1+p_{-1}} \log\left(1+\frac{p_1}{p_{-1}}\right)$$

($\frac{p_{-1}}{p_1}$: $y_i = 1$; $\frac{p_1}{p_{-1}}$: $y_i = -1$)

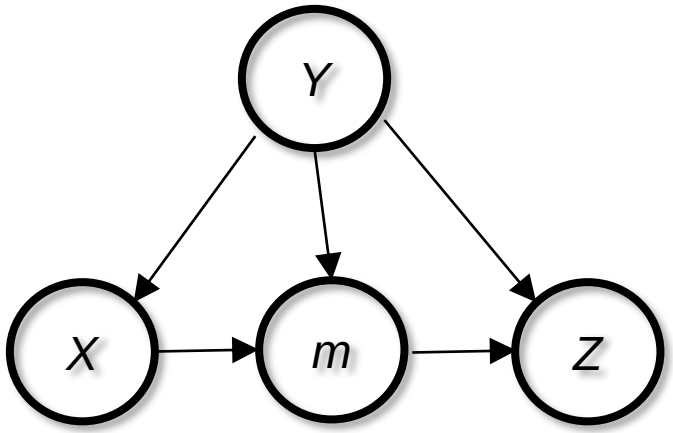
$$= -\frac{1}{2} KL\left(p_1 \,\|\, \frac{p_1+p_{-1}}{2}\right) - \frac{1}{2} KL\left(p_{-1} \,\|\, \frac{p_1+p_{-1}}{2}\right) + \log 2$$

- We can generalize that a loss function with an optimal prediction function is related to its corresponding *f*-divergence. (Many to one relationship)

Loss functions – *f*-divergences



We know *m* should not be a relevant feature



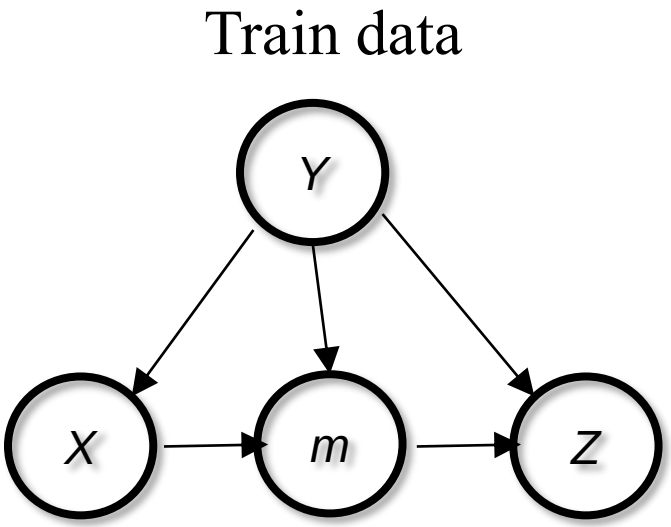
Y: discrete
X, m, Z: continuous

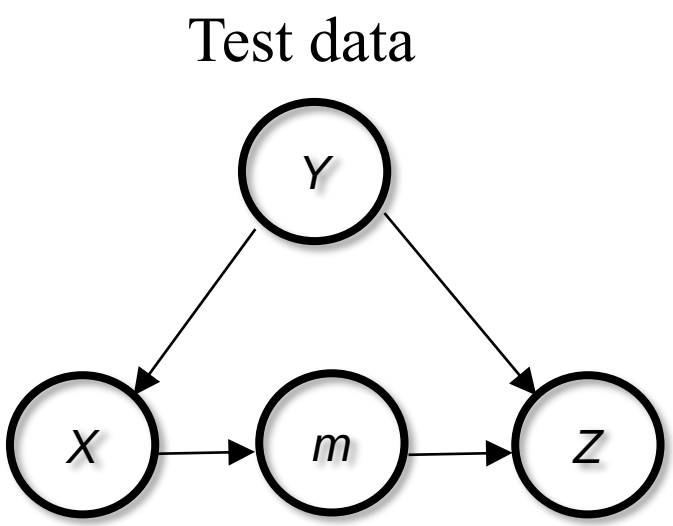
We do not want to include *m* into the set of relevant features (for various reasons).

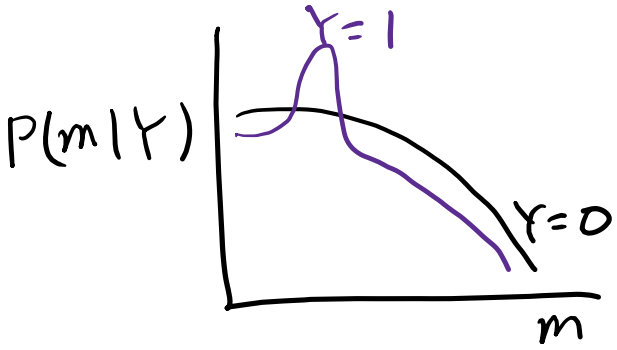
Various reasons:

- moral reasons
- prohibited by law
- Real data will not have the effect from *m*.
- We want to eliminate the effect of one variable (e.g. medicine)

We know *m* should not be a relevant feature





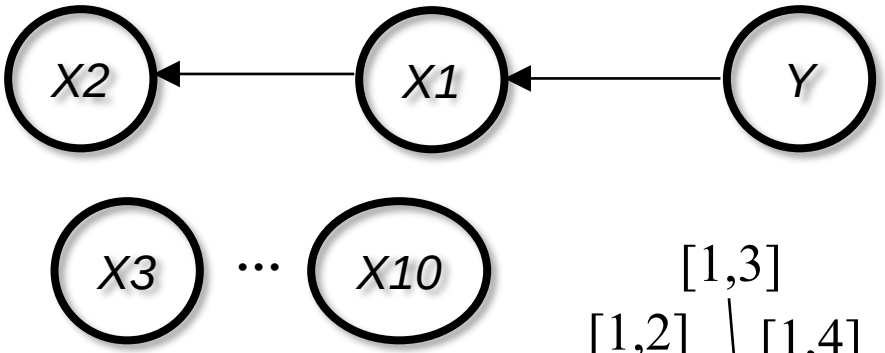


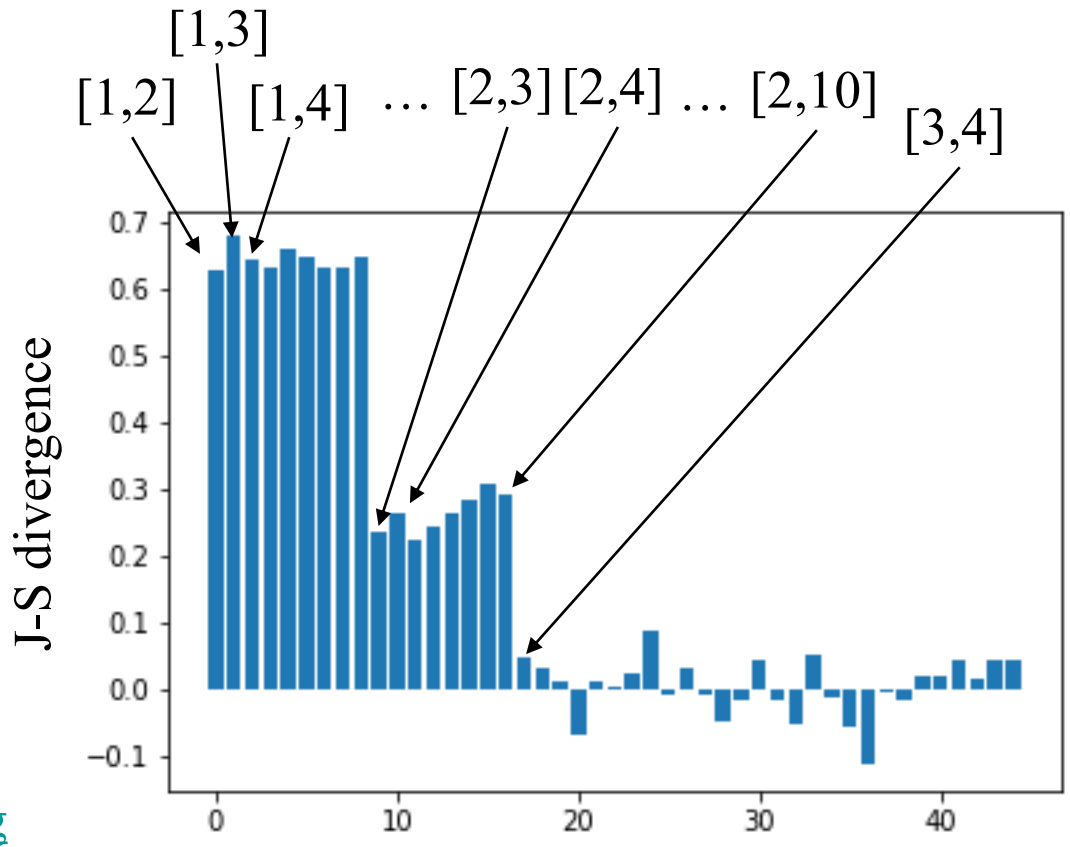
In simulation $Y \not\perp m \mid X$

In real $Y \perp\!\!\!\perp m \mid X$

Extraction of *m* is not enough.
Eliminate the effect of *m* from Z is necessary.

Select Two Features





https://github.com/nohyung

Nearest Neighbor Density Functional Estimation From Inverse Laplace Transform

J. Jon Ryu, *Student Member, IEEE*, Shouvik Ganguly, *Member, IEEE*, Young-Han Kim, *Fellow, IEEE*, Yung-Kyun Noh, *Member, IEEE*, and Daniel D. Lee, *Fellow, IEEE*

A new approach to L_2-consistent estimation of a general functional using k-nearest neighbor distances is proposed where the functional under consideration is in the form of the expectation of some function f of the densities at each point. The estimator is designed to be asymptotically unbiased, using the convergence of the normalized volume of a k-nearest neighbor ball to a Gamma distribution in the large-sample limit, and naturally involves the inverse Laplace transform of a scaled version of the function f. Some instantiations of the proposed estimator recover existing k-nearest neighbor based estimators of Shannon and Rényi entropies and Kullback–Leibler and Rényi divergences, and discover new consistent estimators for many other functionals such as logarithmic entropies and divergences. The L_2-consistency of the proposed estimator is established for a broad class of densities for general functionals, and the convergence rate in mean squared error is established as a function of the sample size for smooth, bounded densities.

***Index Terms*—Density functional estimation, information measure, nearest neighbor, inverse Laplace transform.**

I. Introduction

THIS paper studies the problem of estimating an entropy functional of the form

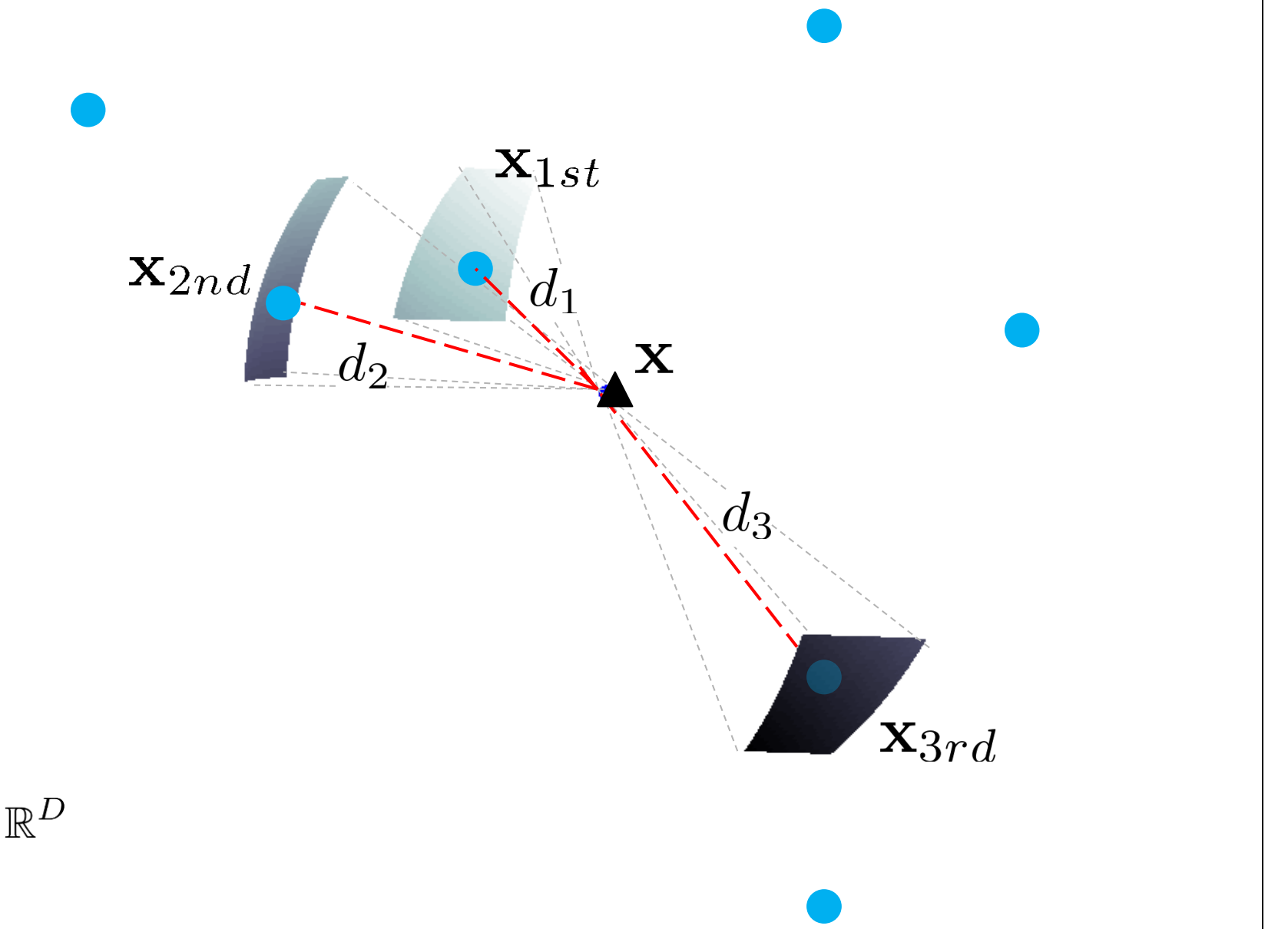
where $f\colon \mathbb{R}_+ \to \mathbb{R}$ is a given function and p is a probability density over $\mathbb{R}^d$. Table I lists examples of f and the corresponding functional T_f. The goal is to estimate $T_f(p)$ based on independent and identically distributed (i.i.d.) samples $\mathbf{X}_{1:m} = (\mathbf{X}_1, \ldots, \mathbf{X}_m)$ from p by forming an estimator $\hat{T}_f^m(\mathbf{X}_{1:m})$ that converges to $T_f(p)$ in L_2 as the sample size m grows to infinity, that is,

$$\lim_{m\to\infty} \mathbb{E}\big[\big(\hat{T}_f^m(\mathbf{X}_{1:m}) - T_f(p)\big)^2\big] = 0.$$

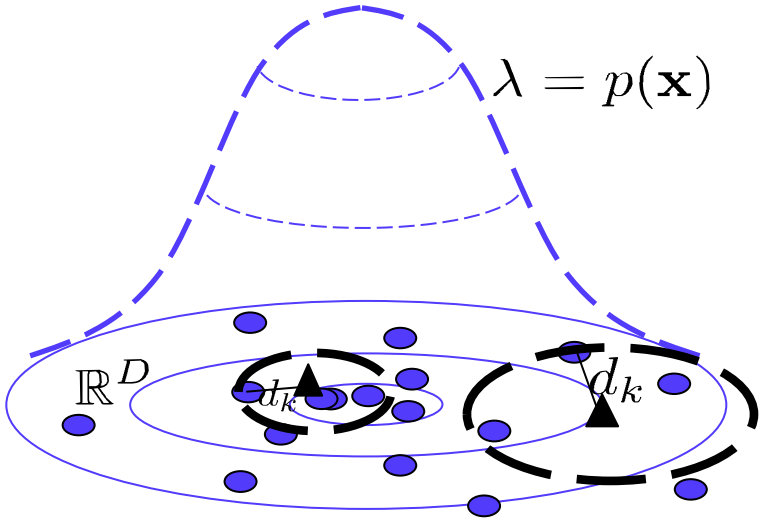
More generally, let $f\colon \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}$ and consider a divergence functional

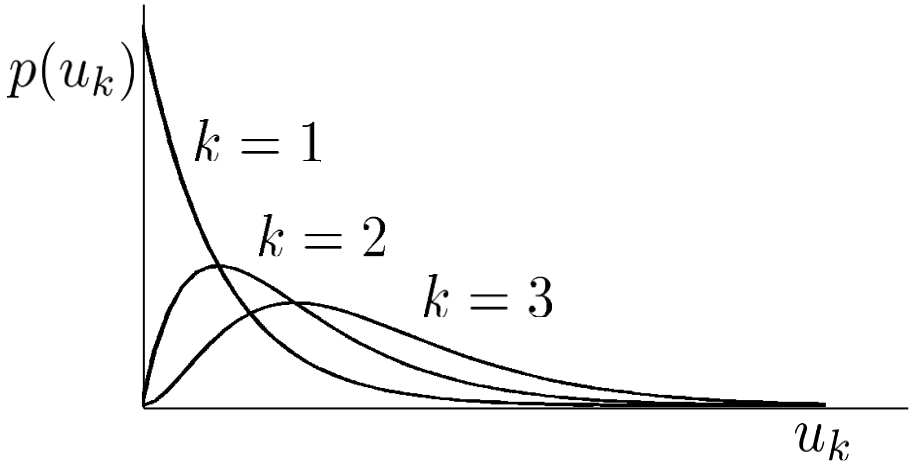
$$T_f(p,q) := \mathbb{E}_{\mathbf{X}\sim p}[f(p(\mathbf{X}), q(\mathbf{X}))] = \int f(p(\mathbf{x}), q(\mathbf{x}))p(\mathbf{x})\,\mathrm{d}\mathbf{x}$$

of a pair of probability densities p and q over $\mathbb{R}^d$. Table II lists examples of f and the corresponding T_f. In this case, the main problem is to construct an estimator $\hat{T}_f^{m,n}(\mathbf{X}_{1:m}, \mathbf{Y}_{1:n})$ based on i.i.d. samples $\mathbf{X}_{1:m}$ from p and $\mathbf{Y}_{1:n}$ from q, independent of each other, such that


$\mathbf{x}_{1st}$
$\mathbf{x}_{2nd}$
d_1
d_2
$\mathbf{x}$
d_3
$\mathbf{x}_{3rd}$
$\mathbb{R}^D$

Density Function for Nearest Neighbor Distances





Gamma (Erlang) function of order k

$N \to \infty$,

$$p(u^{(k)}|\lambda) = \frac{\lambda^k}{\Gamma(k)} \exp\left(-\lambda u^{(k)}\right) (u^{(k)})^{k-1}$$

$(\lambda = p(\mathbf{x}))$

Volume of sphere

$$u^{(k)} = N\gamma d_k^D, \quad \gamma = \frac{\pi^{\frac{D}{2}}}{\Gamma(\frac{D}{2}+1)}$$

Karl W. Pettis et al. (1979) TPAMI
Hertz, P. (1909) Mathematische Annalen

Construction of the Estimator

$$D_f(p_1(\mathbf{x}), p_2(\mathbf{x})) = \int p_1(\mathbf{x}) f\left(\frac{p_2(\mathbf{x})}{p_1(\mathbf{x})}\right) d\mathbf{x}$$

$$\widehat{D_f}(p_1(\mathbf{x}), p_2(\mathbf{x})) = \frac{1}{N} \sum_{\mathbf{x}_i \sim p_1(\mathbf{x})} \phi(u_1^{(k_1)}(\mathbf{x}_i), u_2^{(k_2)}(\mathbf{x}_i))$$

classes

Let $\mathbb{E}_{u_1^{(k_1)}, u_2^{(k_2)}}[\phi(\mathbf{x})] = f\left(\frac{p_2(\mathbf{x})}{p_1(\mathbf{x})}\right)$

Example – How to Build an Estimator

- Kullback-Leibler Estimator

$$D_{\mathrm{KL}}(p_1(\mathbf{x}), p_2(\mathbf{x})) = -\int p_1(\mathbf{x}) \log\left(\frac{p_2(\mathbf{x})}{p_1(\mathbf{x})}\right) d\mathbf{x}$$

$$\mathbb{E}_{u_1^{(k)}, u_2^{(k)}}[\phi] =$$

$$\int_0^\infty \int_0^\infty \frac{p_1^k}{\Gamma(k)} \exp(-p_1 u_1^{(k)}) {u_1^{(k)}}^{k-1} \frac{p_2^k}{\Gamma(k)} \exp(-p_2 u_2^{(k)}) {u_2^{(k)}}^{k-1} \underline{\phi(u_1^{(k)}, u_2^{(k)})} du_1^{(k)} du_2^{(k)}$$

$$= \frac{p_1^k p_2^k}{\Gamma(k)^2} \mathcal{L}_{p_1}\left[\mathcal{L}_{p_2}\left[\underline{\phi(u_1^{(k)}, u_2^{(k)})} {u_1^{(k)}}^{k-1} {u_2^{(k)}}^{k-1}\right]\right] = -\log\left(\frac{p_2}{p_1}\right)$$

Laplace transform: $\mathcal{L}_s[f(t)] = \int_0^\infty f(t) \exp(-st) dt$

Laplace Transform

$$u_1 = u_1^{(k_1)}, u_2 = u_2^{(k_2)}$$

$$\mathcal{L}_{p_1}\left[\mathcal{L}_{p_2}\left[\phi(u_1,u_2){u_1}^{k_1-1}{u_2}^{k_2-1}\right]\right] = -\frac{\Gamma(k_1)\Gamma(k_2)}{p_1^{k_1}p_2^{k_2}}\log\left(\frac{p_2}{p_1}\right)$$

- Perform the inverse Laplace transform of $-\frac{\Gamma(k_1)\Gamma(k_2)}{p_1^{k_1}p_2^{k_2}}\log\left(\frac{p_2}{p_1}\right)$ with respect to p_1 and p_2, then multiply $\frac{1}{u_1^{k_1-1}u_2^{k_2-1}}$ to obtain $\phi(u_1,u_2)$.

- Use the Laplace Transforms

$$\mathcal{L}_s[t^n\log t] = \Gamma(n+1)s^{-(n+1)}(\psi(n+1)-\log s), \quad n>-1$$
$$\mathcal{L}_s[t^n] = \Gamma(n+1)s^{-(n+1)}, \quad n>-1$$

$$\phi(u_1, u_2) = \log u_1 - \log u_2 - \psi(k_1) + \psi(k_2)$$

$$\mathbb{E}_{u_1,u_2}\phi(u_1, u_2) = -\log \frac{p_2}{p_1}$$

- Convergence?
  - It is practically working to check whether the variance (expectation of the square) diverge or not.

$$Var\left[\phi(u_1, u_2)^2\right] =$$

$$\mathbb{E}_{u_1,u_2}\left[\phi(u_1, u_2)^2\right] - \mathbb{E}_{u_1,u_2}\left[\phi(u_1, u_2)\right]^2 < \infty$$

$$\mathbb{E}_{u_1,u_2}\left[\phi(u_1,u_2)^2\right] < \infty$$

$$\Longleftrightarrow$$

$$\mathcal{L}_{p_1}\mathcal{L}_{p_2}\left[\phi(u_1,u_2)^2 u_1^{k_1-1}u_2^{k_2-1}\right] < \infty$$

$$\Longleftrightarrow$$

$$\mathcal{L}_{p_1}\left[u_1^{k_1-1}(\log u_1)^2\right] = (-1)^{k_1-1}\frac{d^{k_1-1}}{dp_1^{k_1-1}}\frac{1}{p_1}\left((\log p_1 + C)^2 + \frac{1}{6}\pi^2\right) < \infty$$

$$\mathcal{L}_{p_2}\left[u_2^{k_2-1}(\log u_2)^2\right] < \infty$$

Kozachenko-Leonenko estimator

$$\widehat{D_f}(p_1(\mathbf{x}), p_2(\mathbf{x})) = \frac{1}{N} \sum_{\mathbf{x}_i \sim p_1(\mathbf{x})} \phi(u_1^{(k_1)}(\mathbf{x}_i), u_2^{(k_2)}(\mathbf{x}_i))$$

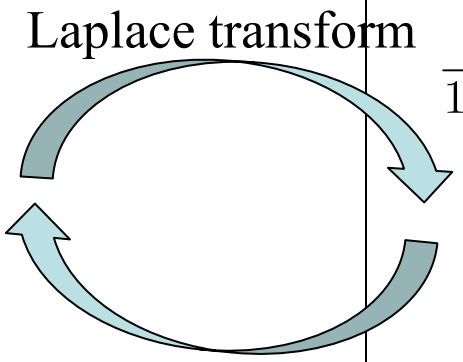
$$\phi(u_1^{(k_1)}, u_2^{(k_2)}) = \log u_1^{(k_1)} - \log u_2^{(k_2)} - \psi(k_1) + \psi(k_2)$$

L. F. Kozachenko and N. N. Leonenko (1987) Problemy Peredachi Informatsii
N. Leonenko, L. Pronzato, &V. Savani, (2008) Annals of Statistics
B. Poczos and J. Schneider (2011) AISTATS

– For the analysis with finite *N*, see

D. Lombardi and S. Pant (2016) Phys. Rev. E
A. Kraskov, H. Stögbauer, and P. Grassberger (2004) Phys. Rev. E

$D_f(p_1(\mathbf{x}), p_2(\mathbf{x}))$	Estimator $\phi(u_1, u_2)$	$f(t)$
$\frac{1}{\alpha-1}\left(\int p_1^{(1-\alpha)} p_2^{\alpha} d\mathbf{x} - 1\right)$ $(\alpha \neq 1)$	$\frac{1}{\alpha-1}\left(\frac{\Gamma(k_1)\Gamma(k_2)}{\Gamma(\alpha+k_1)\Gamma(k_2-\alpha)}\left(\frac{u_1}{u_2}\right)^{\alpha} - \frac{\Gamma(k_1)\Gamma(k_2)}{\Gamma(k_1+1)\Gamma(k_2-1)}\frac{u_1}{u_2}\right)$	$\frac{t^{\alpha}-t}{\alpha-1}$
$-\int p_1 \log\left(\frac{p_2}{p_1}\right) d\mathbf{x}$	$\log u_1^{(k_1)} - \log u_2^{(k_2)} - \psi(k_1) + \psi(k_2)$ $\psi(.)$: digamma	$-\log t$
$1 - \int \sqrt{p_1 p_2}\, d\mathbf{x}$	$1 - \frac{1}{\Gamma(1.5)\Gamma(2.5)}\sqrt{\frac{v_1^{(2)}}{u_2^{(2)}}}$	$1-\sqrt{t}$
$1 - \int \frac{p_1 p_2}{p_1+p_2} d\mathbf{x}$	$\mathbb{1}(u_1^{(1)} < u_2^{(1)})$	$\frac{1}{1+t}$
$\vdots$	$\vdots$	$\vdots$

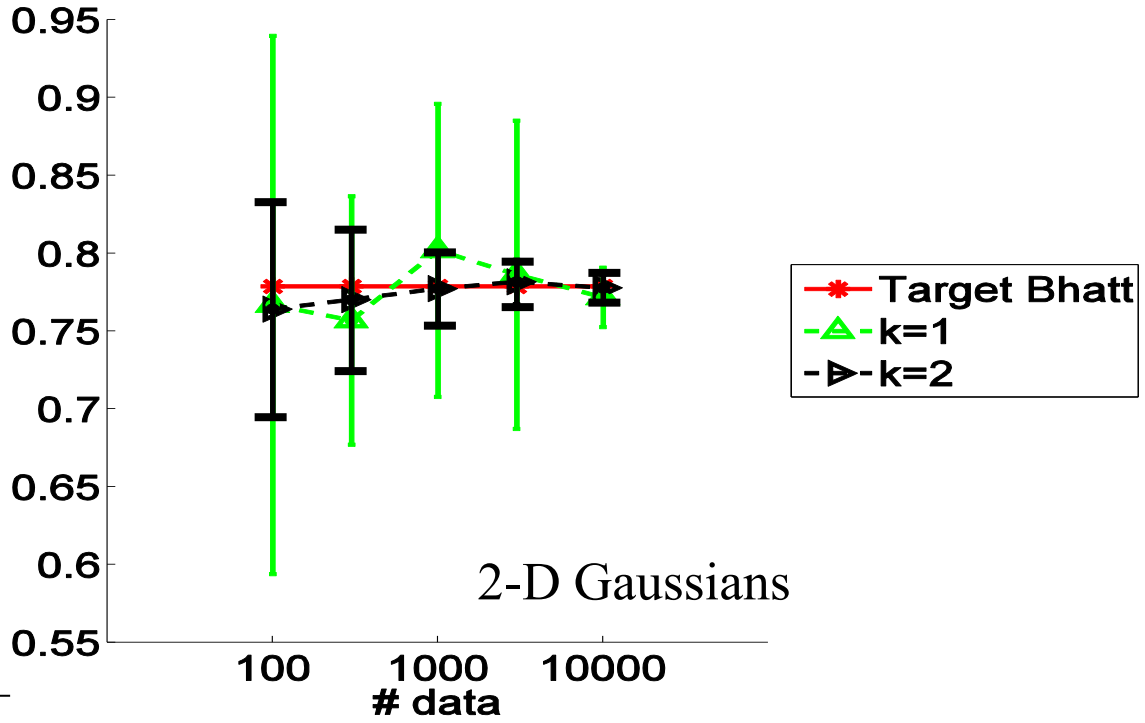


Estimation of Bhattacharyya Coefficient

k_1=k_2=1 ➔ $\phi(u_1, u_2) = \frac{1}{\Gamma(1.5)\Gamma(0.5)}\sqrt{\frac{u_1}{u_2}}$ Condition for k_2 is not satisfied

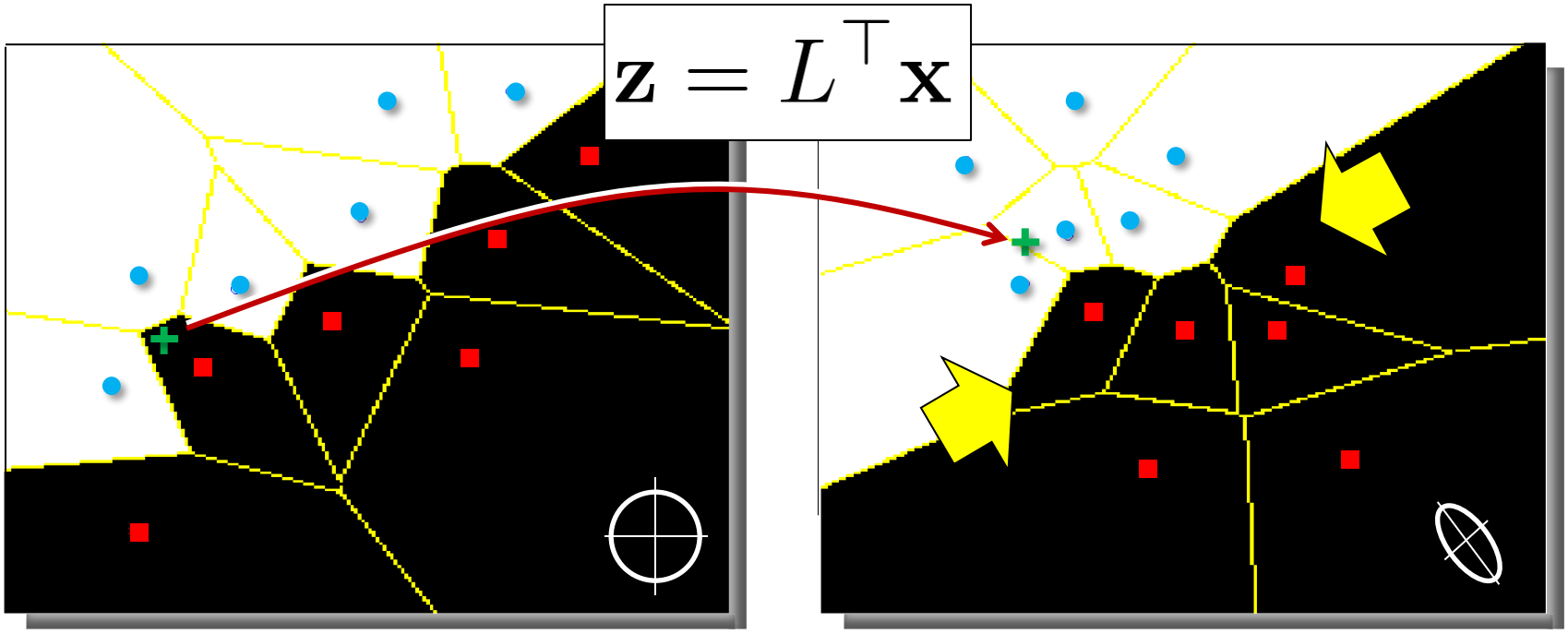
k_1=k_2=2 ➔ $\phi(u_1, u_2) = \frac{1}{\Gamma(2.5)\Gamma(1.5)}\sqrt{\frac{u_1}{u_2}}$

- For two 2-D Gaussian data:



Metric Dependency of Nearest Neighbors

- Different metric changes class belongings



Classified as red *Classified as blue*

Mahalanobis-type distance:

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T A (\mathbf{x}_i - \mathbf{x}_j)}, \quad A \succ 0$$

Biased Estimators

Nearest neighbor classification

$$E_{NN} \to \int \frac{p_1(\mathbf{x})p_2(\mathbf{x})}{p_1(\mathbf{x})+p_2(\mathbf{x})} d\mathbf{x}$$

$$+ \frac{1}{4D} \int \frac{\mathbb{E}_{d_N}\left[d_N^2|\mathbf{x}\right]}{(p_1+p_2)^2} [p_1^2\nabla^2 p_2 + p_2^2\nabla^2 p_1 - p_1 p_2(\nabla^2 p_1 + \nabla^2 p_2)] d\mathbf{x}$$

[Noh et al. TPAMI 2018]

Nadaraya-Watson regression

$$\mathbb{E}\left[\widehat{y}(\mathbf{x}) - \mathbb{E}[y|\mathbf{x}]\right] \to h^2 \left(\frac{\nabla^\top p(\mathbf{x}) \nabla \mathbb{E}[y|\mathbf{x}]}{p(\mathbf{x})} + \frac{\nabla^2 \mathbb{E}[y|\mathbf{x}]}{2} \right) + o(h^4)$$

[Noh et al. NeurIPS 2017]

Nonparametric KL-divergence estimation

$$\sum_{\mathbf{x}\sim p_1} \log \frac{N_2 d_2^D}{N_1 d_1^D} \to -\int p_1 \log \frac{p_2}{p_1} d\mathbf{x}$$

$$+ \frac{1}{2D\gamma^{\frac{2}{D}}} \int \left[(N_1 p_1)^{-\frac{2}{D}} \frac{\nabla^2 p_1}{p_1} - (N_2 p_2)^{-\frac{2}{D}} \frac{\nabla^2 p_2}{p_2} \right] d\mathbf{x}$$

[Noh et al. Neural Computation 2018]

Metric Dependency of Laplacian

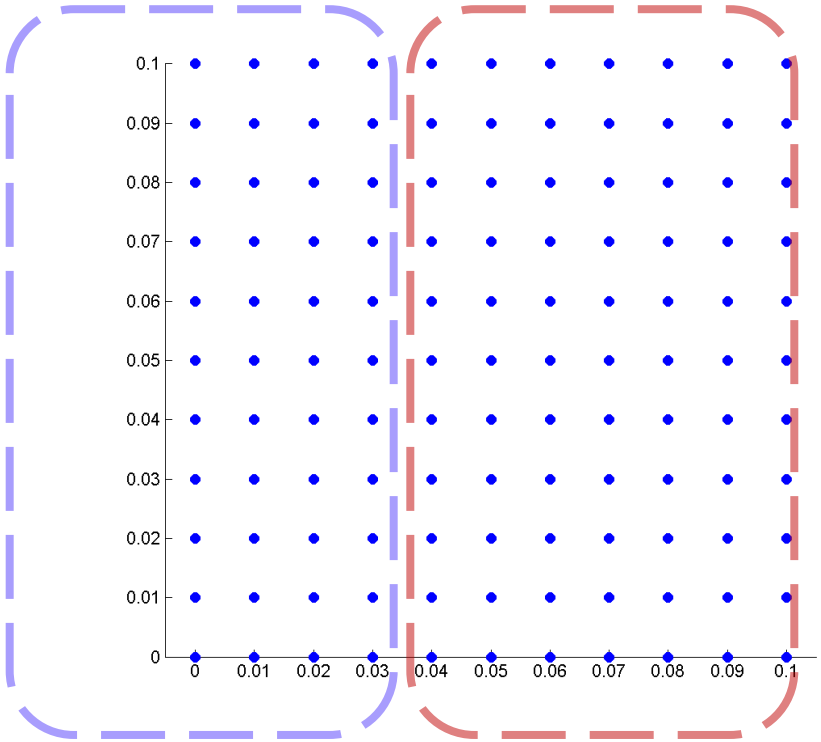
$$\mathbf{z} = L^\top \mathbf{x} \qquad \mathbf{x}, \mathbf{z} \in \mathbb{R}^D$$

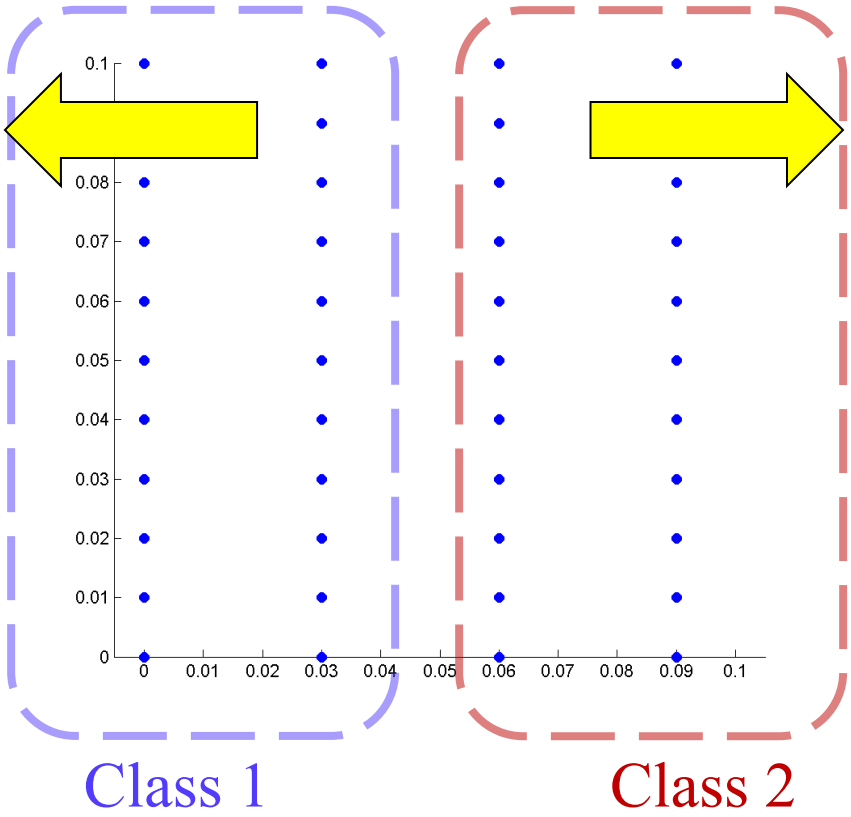
$$\nabla^2 p(\mathbf{z}) = \mathrm{tr}[\nabla\nabla p(\mathbf{z})]$$

$$= \mathrm{tr}[A^{-1}\nabla\nabla p(\mathbf{x})]$$

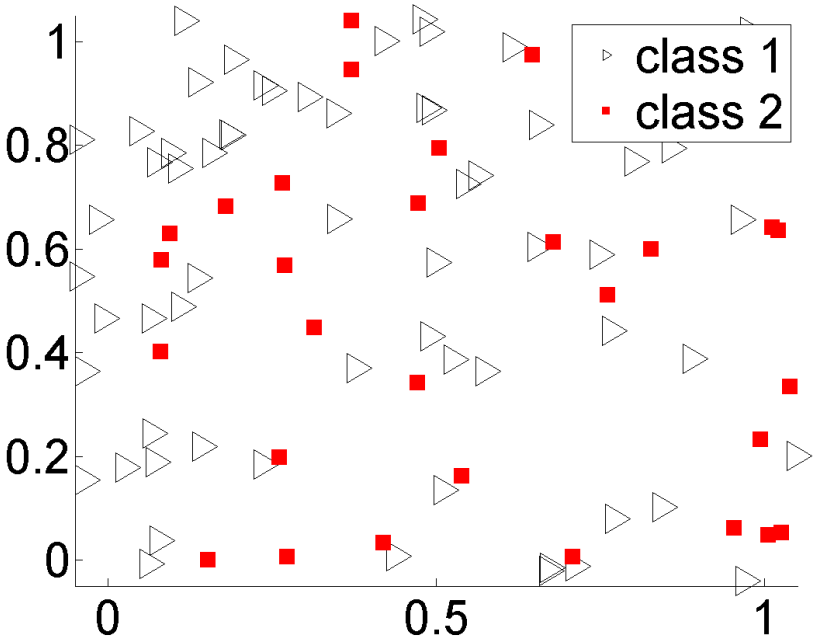
$$(A \text{ s.t. } A = LL^\top)$$

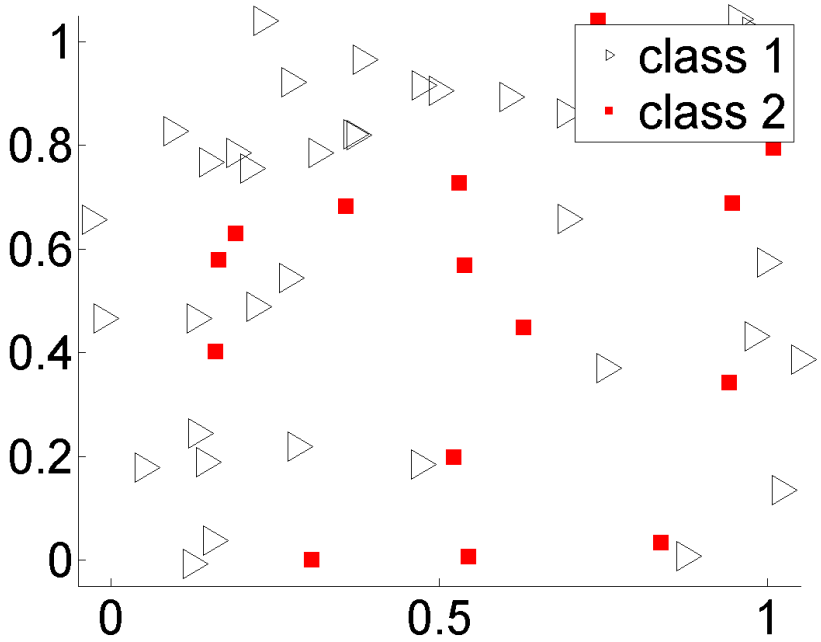
Conventional Idea of Metric Learning

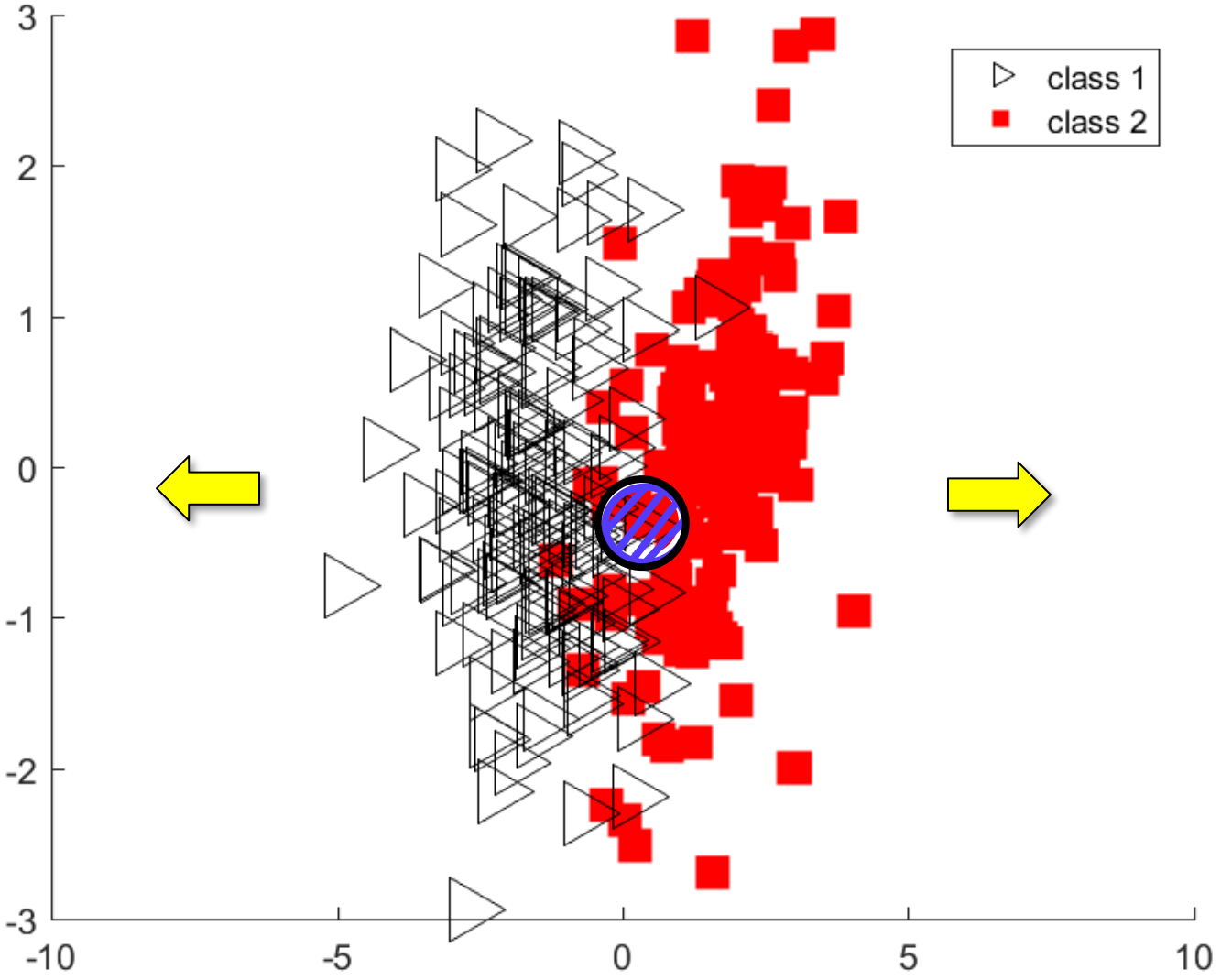




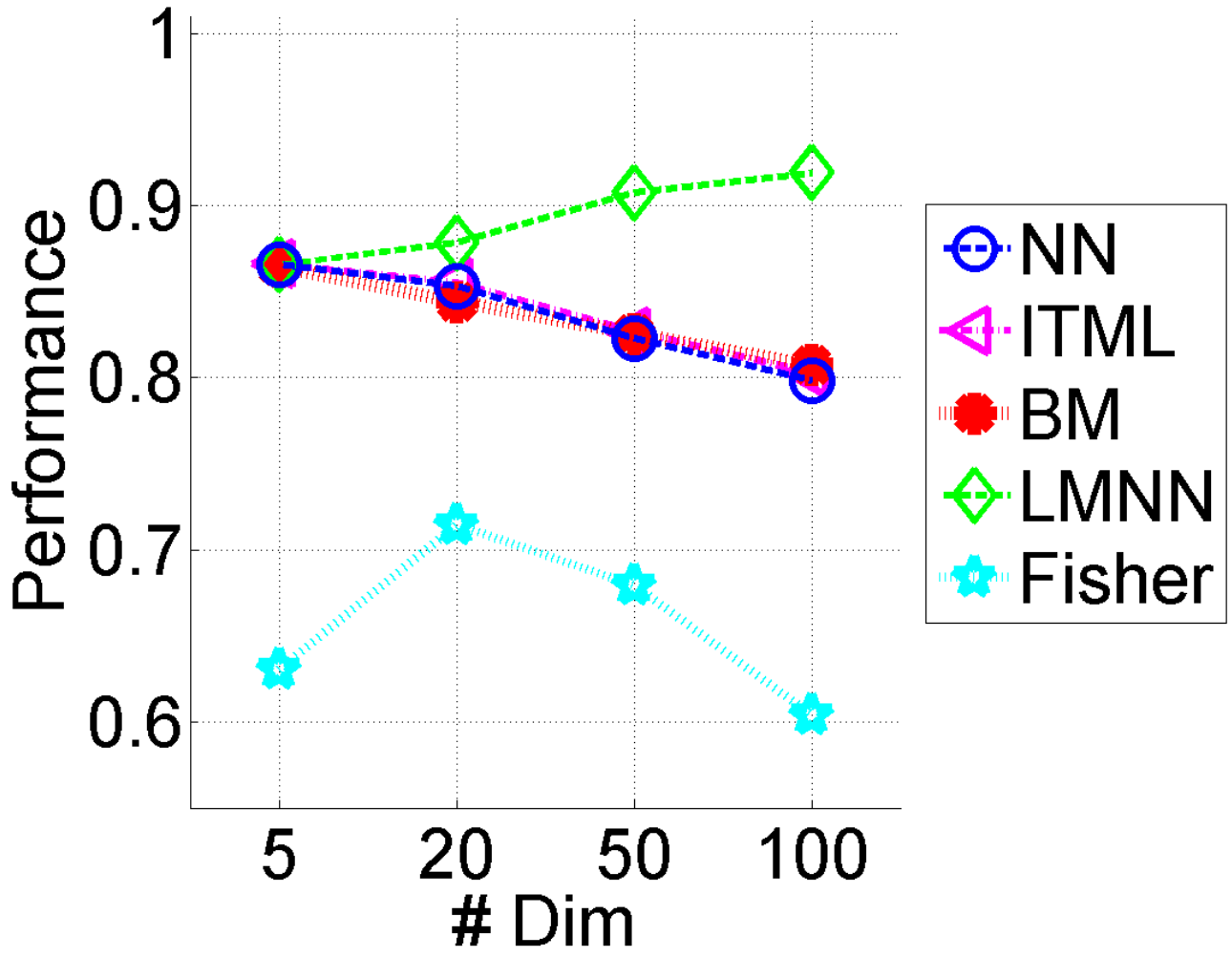
Many Data Situation with Overlap



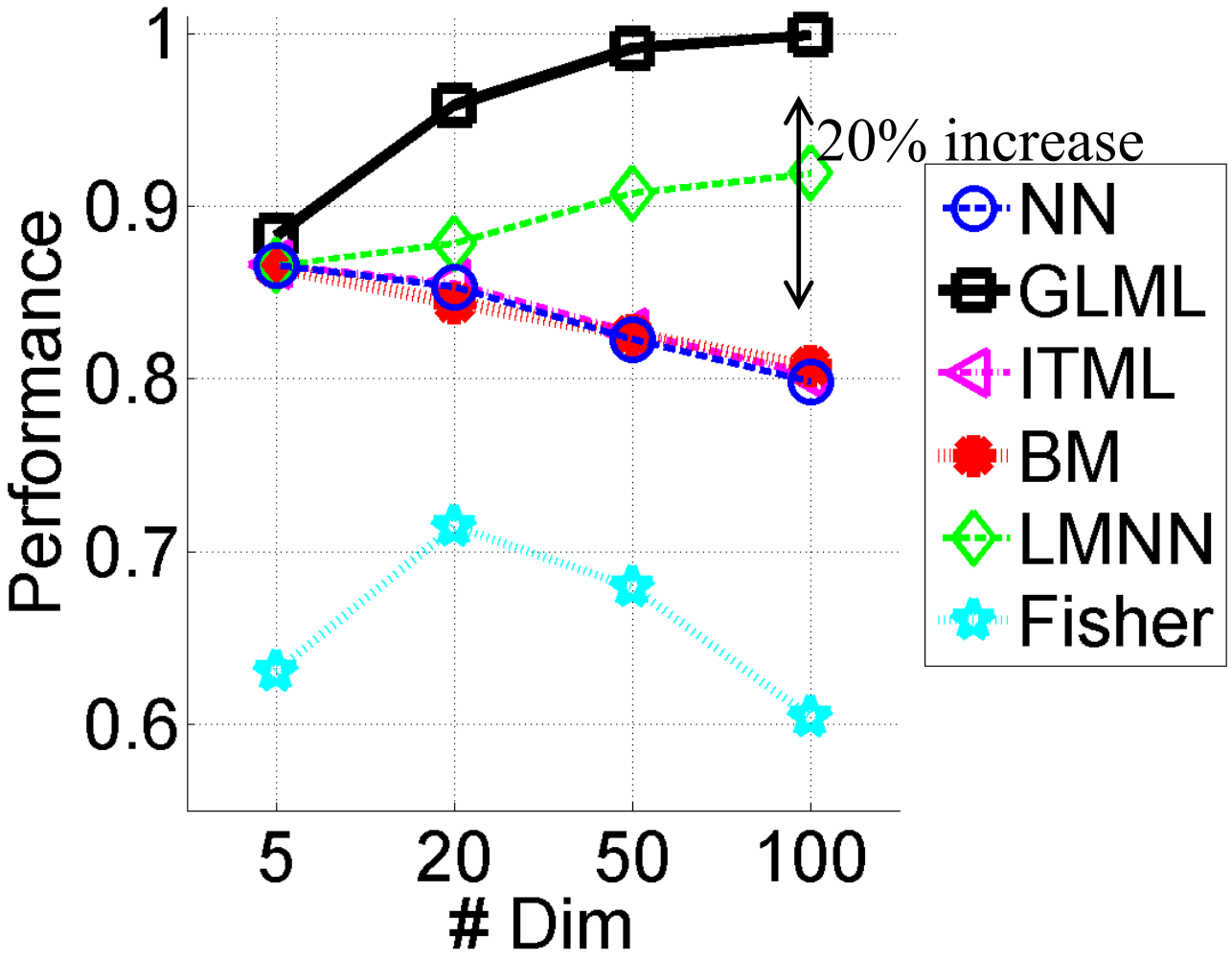



class 1
class 2
3
2
1
0
-1
-2
-3
-10
-5
0
5
10

Conventional Metric Learning

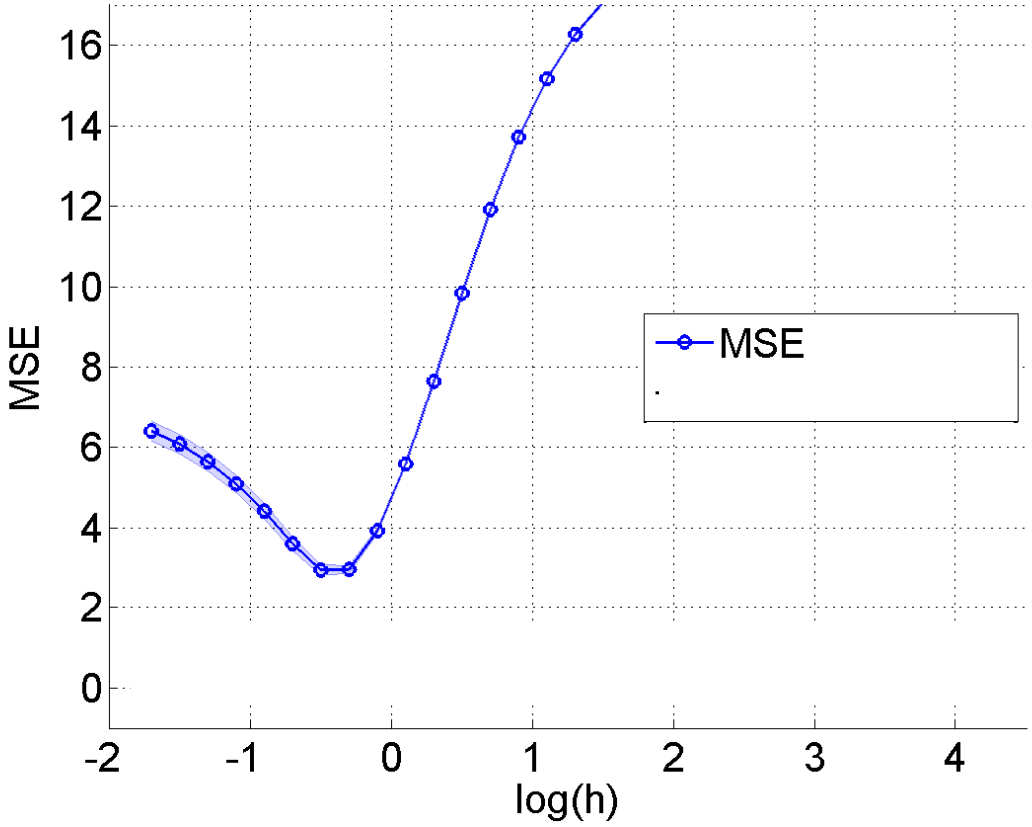


Generative Local Metric Learning (GLML)



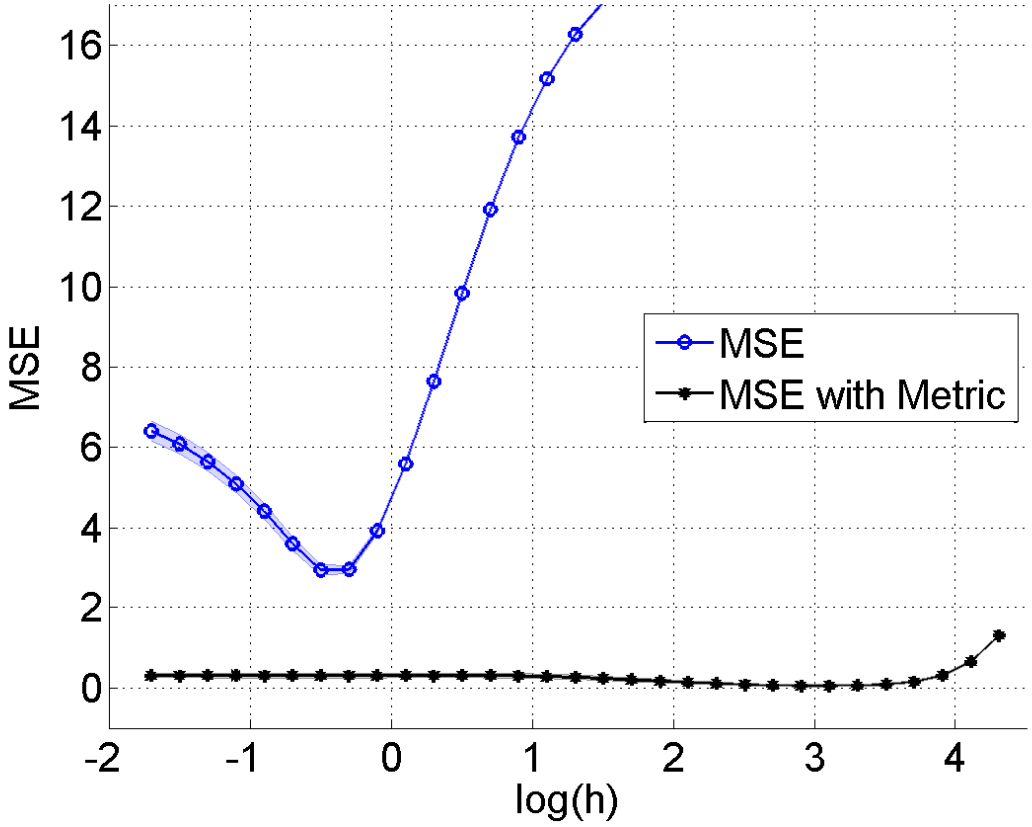
For x & y Jointly Gaussian

- Learned metric is not sensitive to the bandwidth

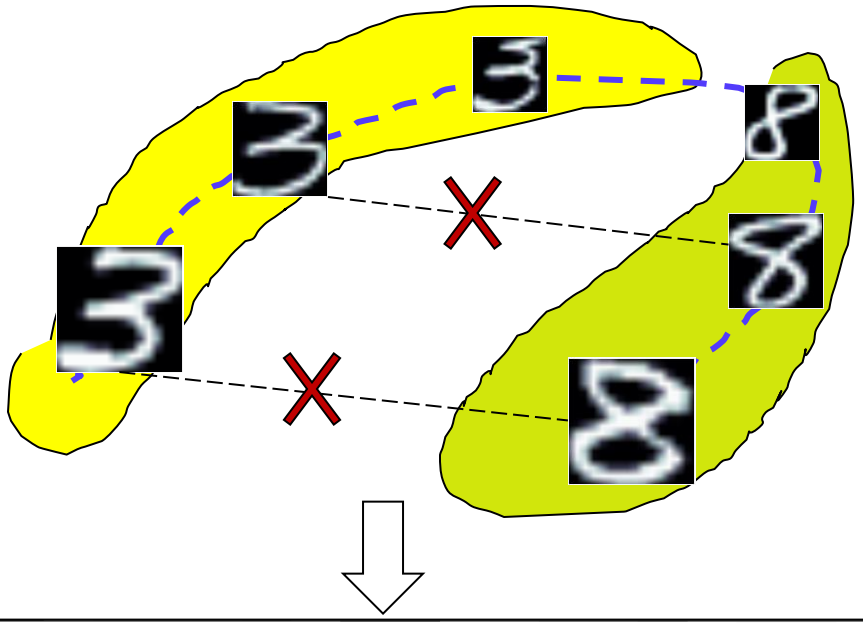


For x & y Jointly Gaussian

- Learned metric is not sensitive to the bandwidth

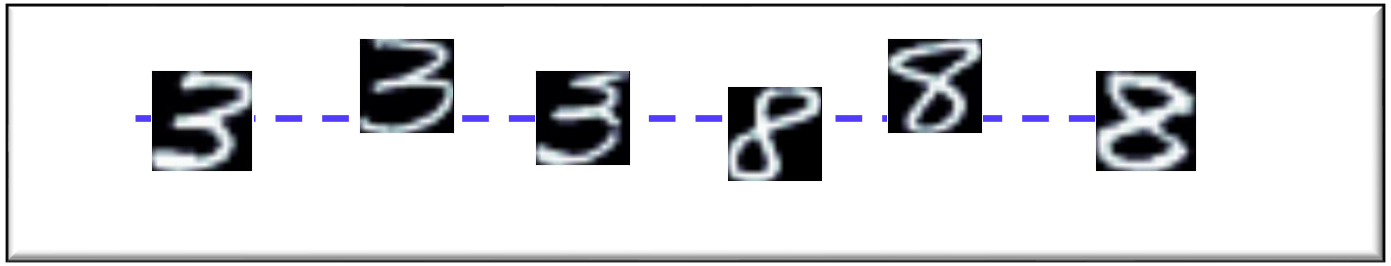


Manifold Embedding (Isomap)



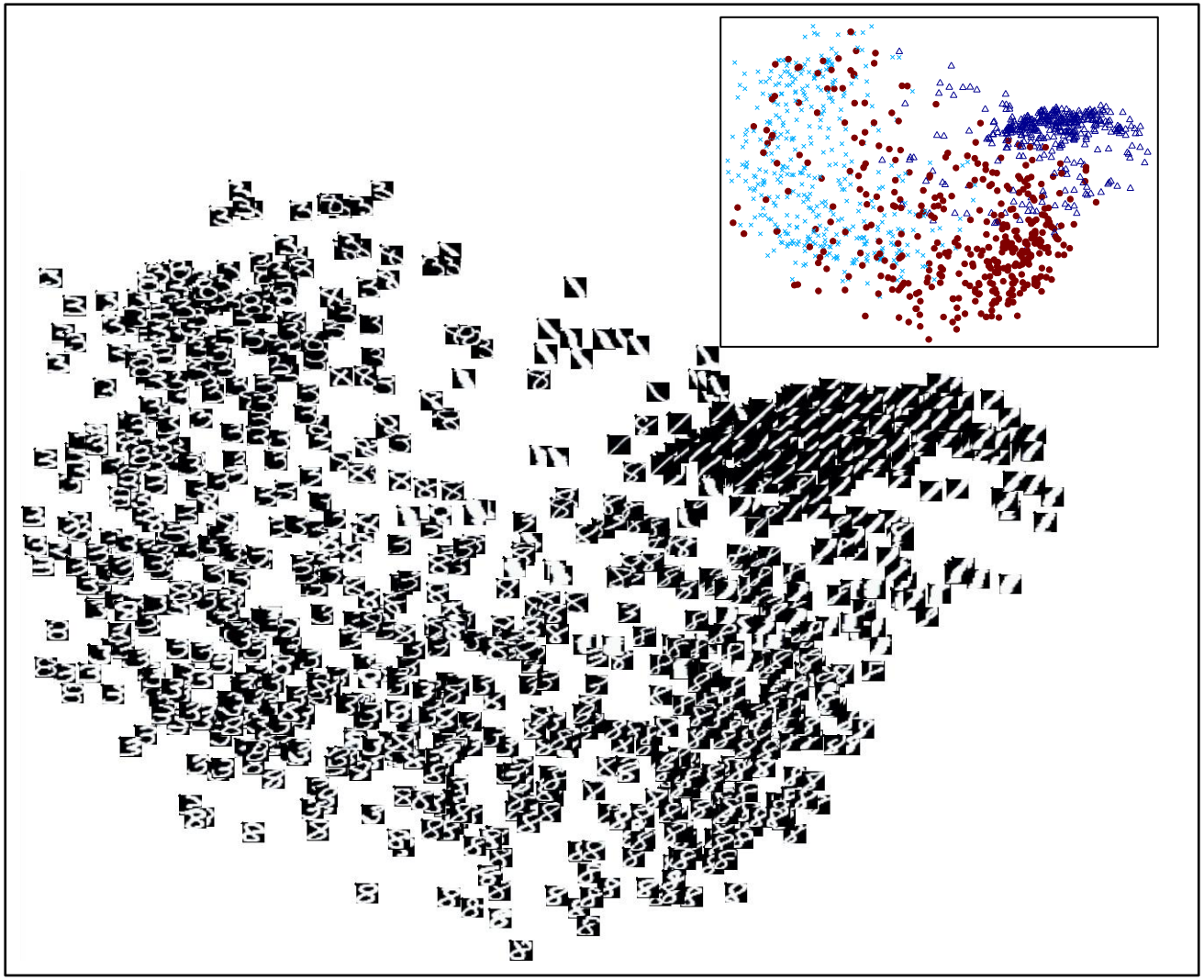
Use Dijkstra algorithm to calculate the manifold distance from nearest neighbor distance

➔ MDS using manifold distance

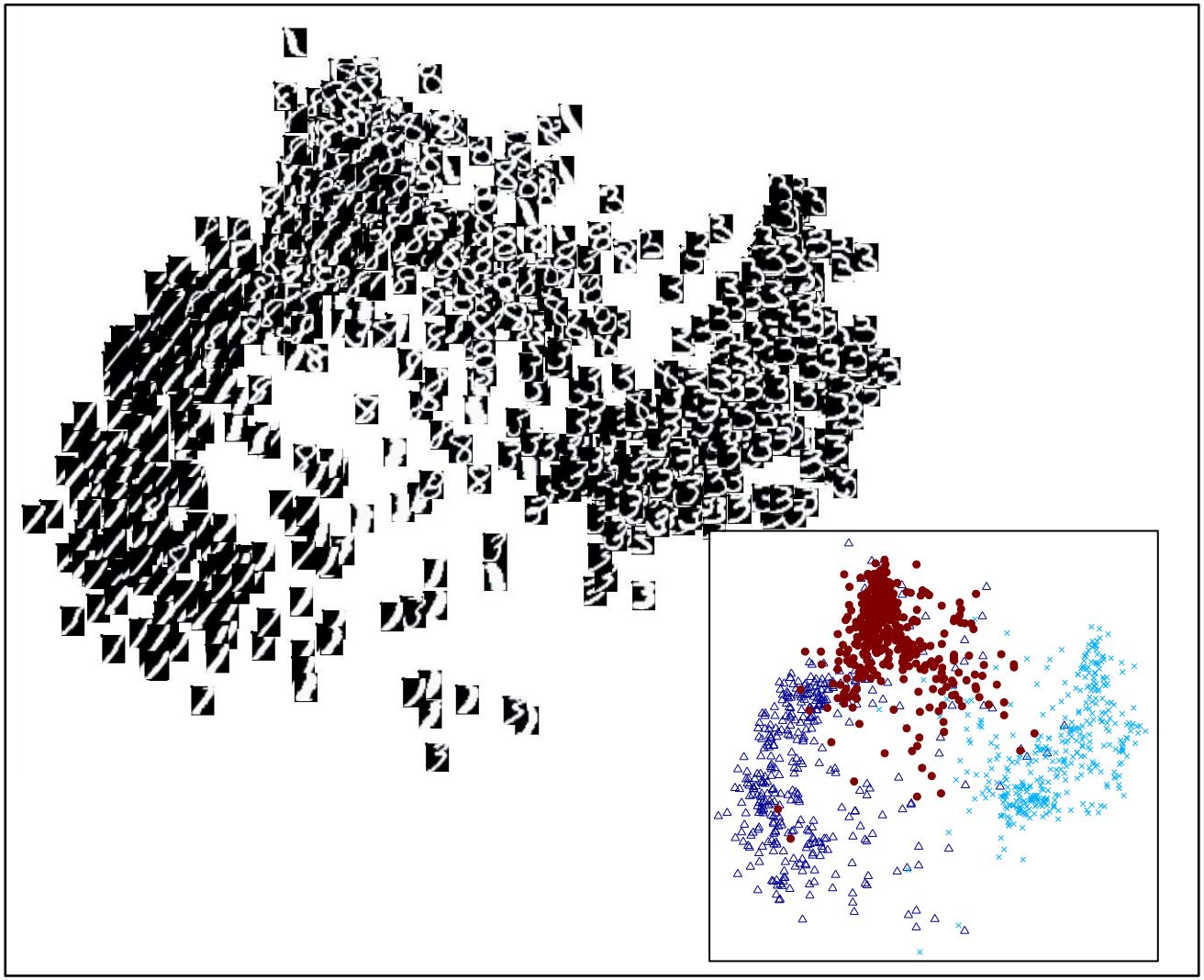


(X)

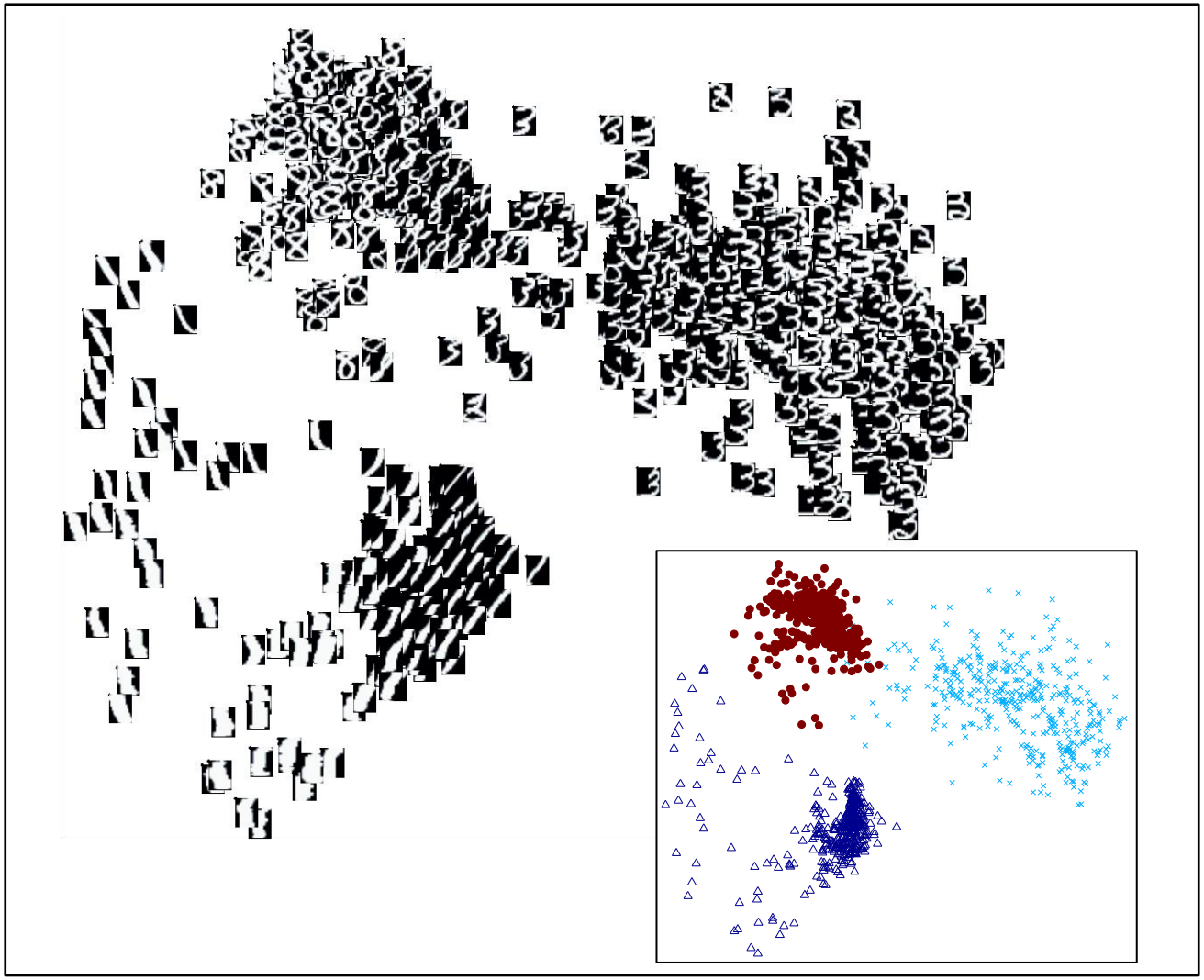
Manifold Embedding (Isomap)



Isomap with LMNN Metric



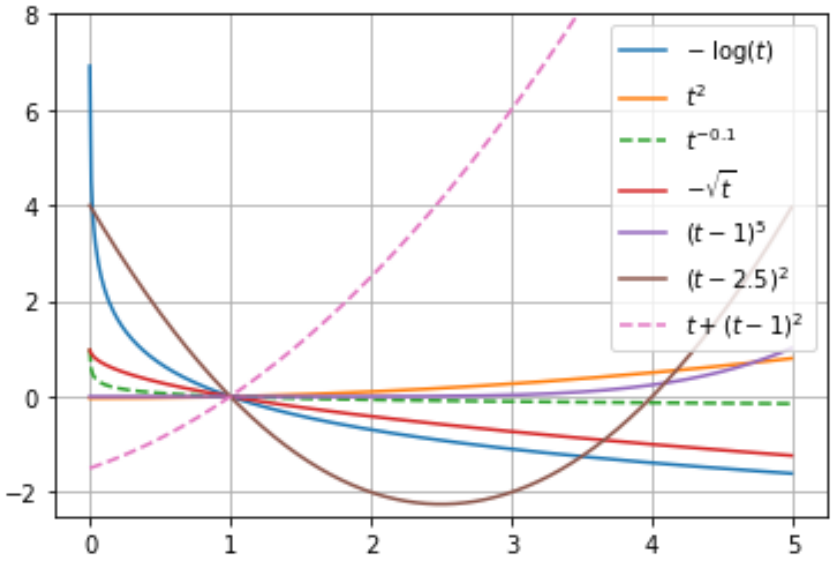
Isomap with GLM Metric



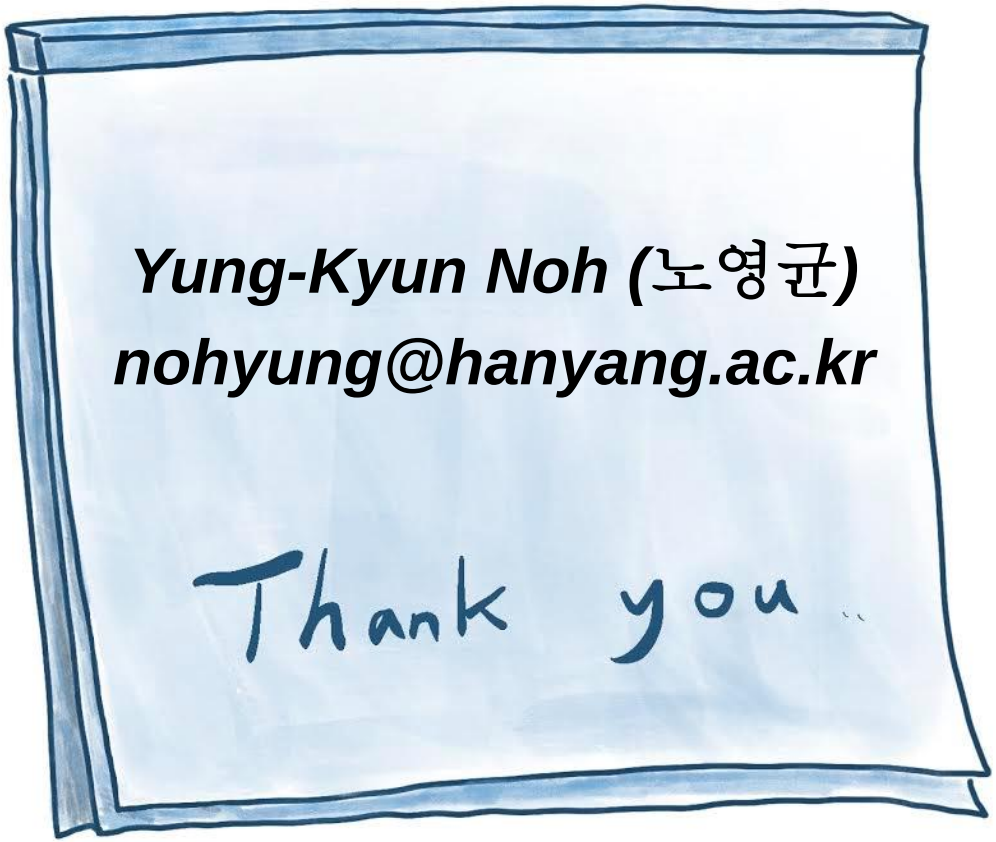
(Summary) Use Diversity of the Measures

- Optimize *f*-divergence for finding a good set of features (or representations).

- Use a variety of *f*-divergences



- Each *f*-divergence has its associated loss function


Yung-Kyun Noh (노영균)
nohyung@hanyang.ac.kr
Thank you