Inference and Estimation Using Nearest Neighbors

2019 The Second Korea-Japan Machine Learning Workshop
2019. 2. 22 (Fri.)

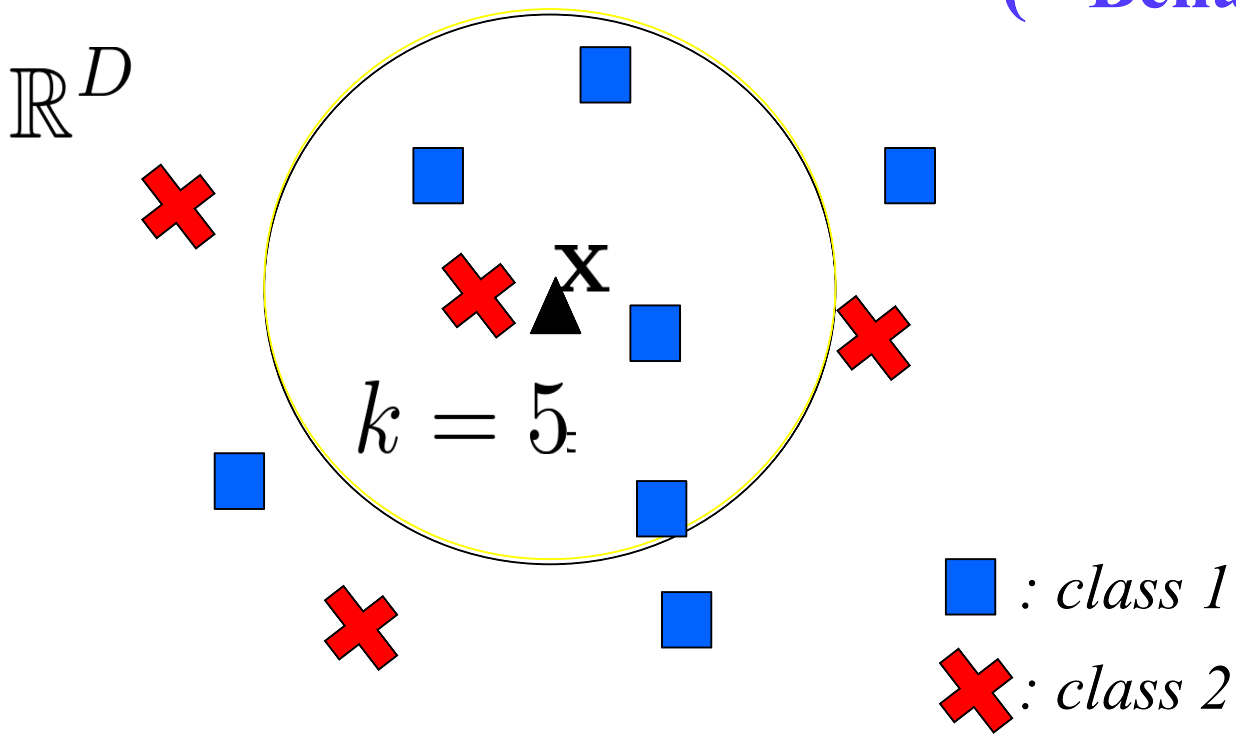
Yung-Kyun Noh

*Seoul National University → **Hanyang University***

Nearest Neighbors

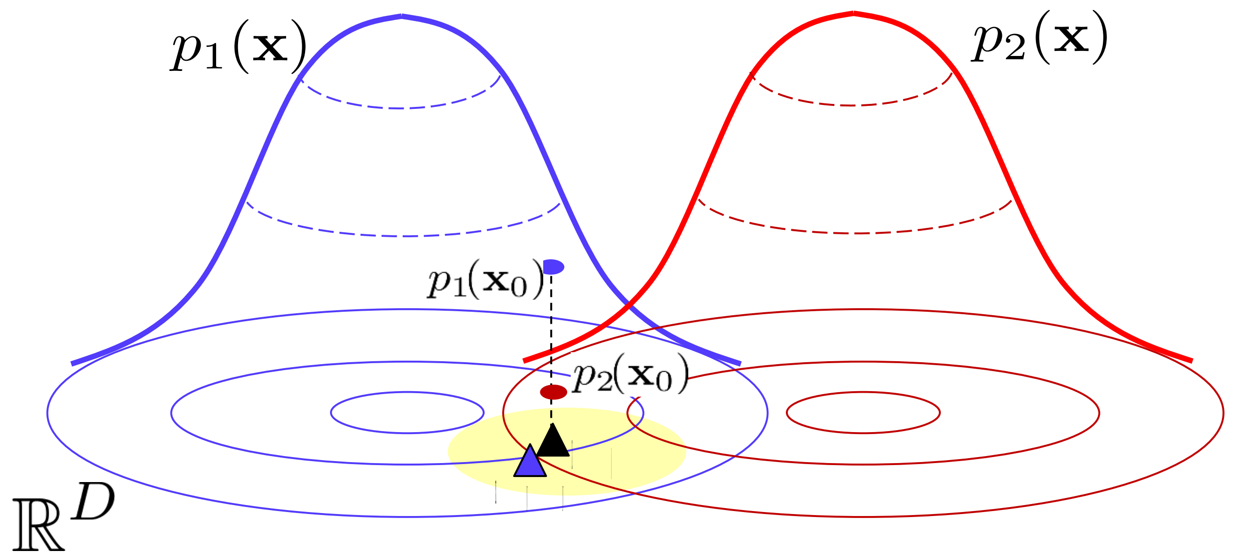
- Similar data share similar properties **(= Labels?)**

  (= Behavior)



$\mathbf{x}_{NN} \to \mathbf{x}$, uniformly with increasing N.

In the limit, $p_1(\mathbf{x}) = p_1(\mathbf{x}_{NN})$ & $p_2(\mathbf{x}) = p_2(\mathbf{x}_{NN})$



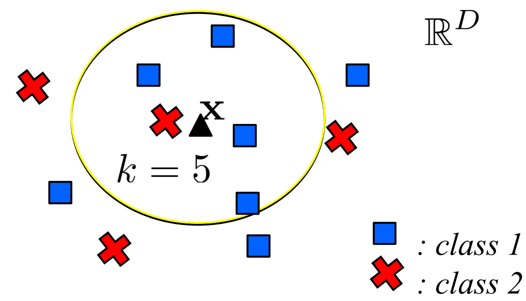
For classification as an example:

$$\epsilon_{(k=1)} = \int \frac{p_1(\mathbf{x})p_2(\mathbf{x})}{p_1(\mathbf{x}) + p_2(\mathbf{x})} d\mathbf{x} \leq 2E_{Bayes}(1 - E_{Bayes})$$

[T. Cover and P. Hart, *IEEE TIT*, 1967]

Applications of Using Nearest Neighbors

- Prediction using *k*-Nearest Neighbor Information
  - *k*-Nearest Neighbor Classification
  - *k*-Nearest Neighbor Regression



- Estimation using *k*-Nearest Neighbor Information

[Leonenko, N., Pronzato, L., & Savani, V., 2008]

$$KL(p_1(\mathbf{x}) || p_2(\mathbf{x})) = -\int p_1(\mathbf{x}) \log \frac{p_2(\mathbf{x})}{p_1(\mathbf{x})} d\mathbf{x}$$

$$\widehat{KL}(p_1(\mathbf{x}) || p_2(\mathbf{x})) = \frac{1}{N_1} \sum_{\mathbf{x} \sim p_1} \log \frac{u_2(\mathbf{x})}{u_1(\mathbf{x})}$$

$$u_c = N_c d_c^D$$

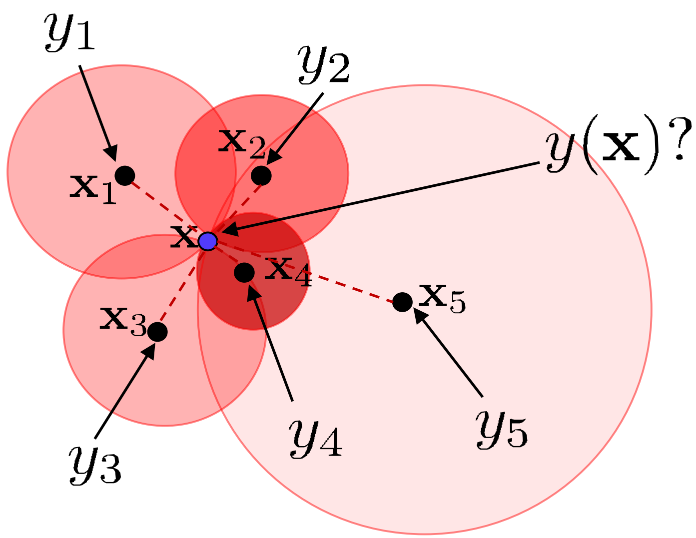
d_c is a distance to the nearest neighbor in class c from $\mathbf{x} \in \mathbb{R}^D$.

Similar Formulations

- Nadaraya-Watson estimator for kernel classification/regression

$$\widehat{y}_N(\mathbf{x}) = \frac{\sum_{i=1}^{N} K(\mathbf{x}_i, \mathbf{x}) y_i}{\sum_{i=1}^{N} K(\mathbf{x}_i, \mathbf{x})}$$

$$\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^{N}$$



$$\mathbf{x}_i, \mathbf{x} \in \mathbb{R}^D$$

Kernel weight with respect to the distance

$$K(\mathbf{x}_i, \mathbf{x}) = K\left(\frac{||\mathbf{x}_i - \mathbf{x}||}{h}\right)$$

bandwidth

Bias Analysis

- *k*-Nearest Neighbor Classification

[R. R. Snapp et al. *The Annals of Statistics*, 1998]
[Y.-K. Noh et al. *IEEE TPAMI*, 2018]

$$E_{NN} \cong \int \frac{p_1(\mathbf{x})p_2(\mathbf{x})}{p_1(\mathbf{x})+p_2(\mathbf{x})}d\mathbf{x} \qquad \cdots ①$$

$$+\frac{1}{4D}\int \frac{\mathbb{E}_{d_N}\left[d_N^2|\mathbf{x}\right]}{(p_1+p_2)^2}\boxed{\left[p_1^2\nabla^2 p_2+p_2^2\nabla^2 p_1-p_1p_2(\nabla^2 p_1+\nabla^2 p_2)\right]}d\mathbf{x} \qquad \cdots ②$$

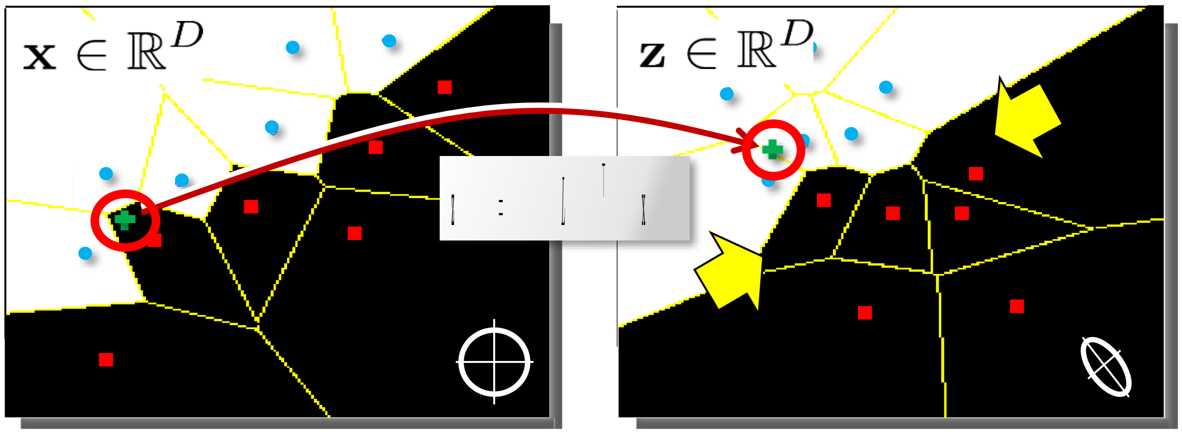
①: Asymptotic *NN* Error

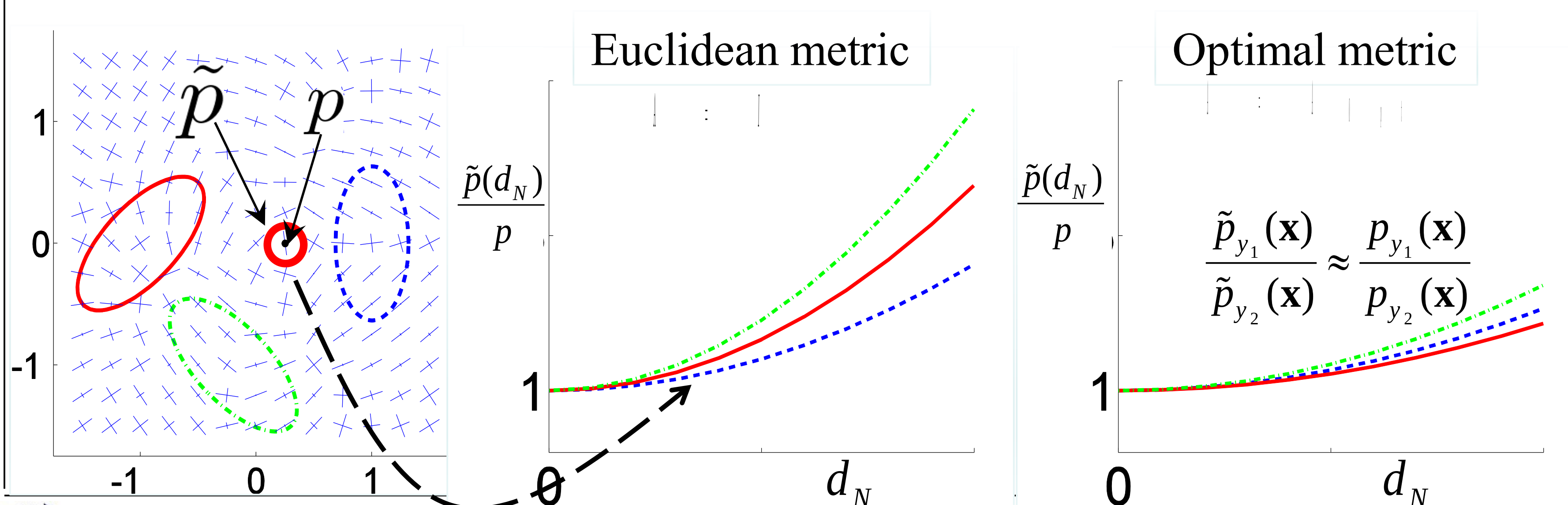
②: Residual due to *Finite Sampling* .

Change of Metric

[Y.-K. Noh et al. *IEEE TPAMI*, 2018]

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

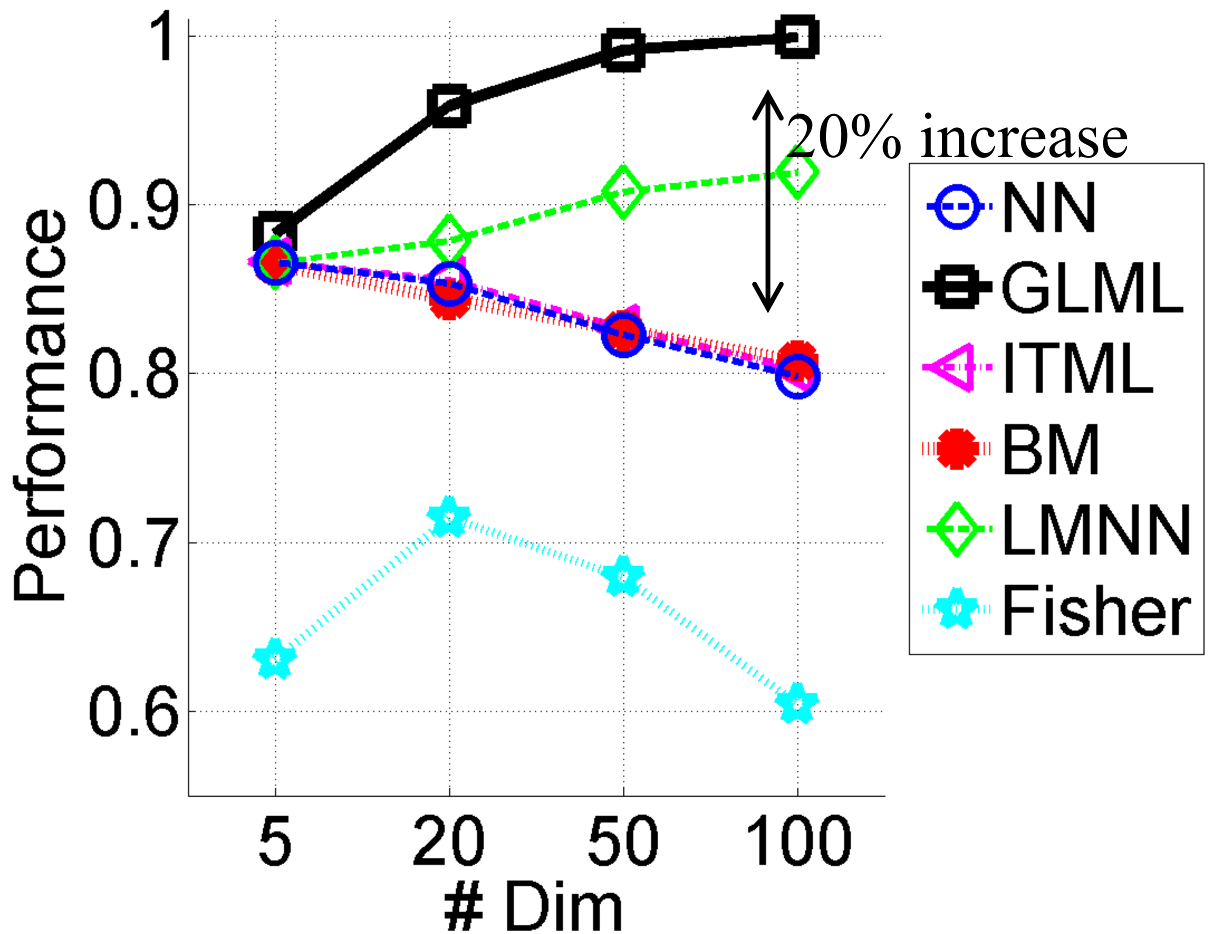




Nearest Neighbor Classification with Metric

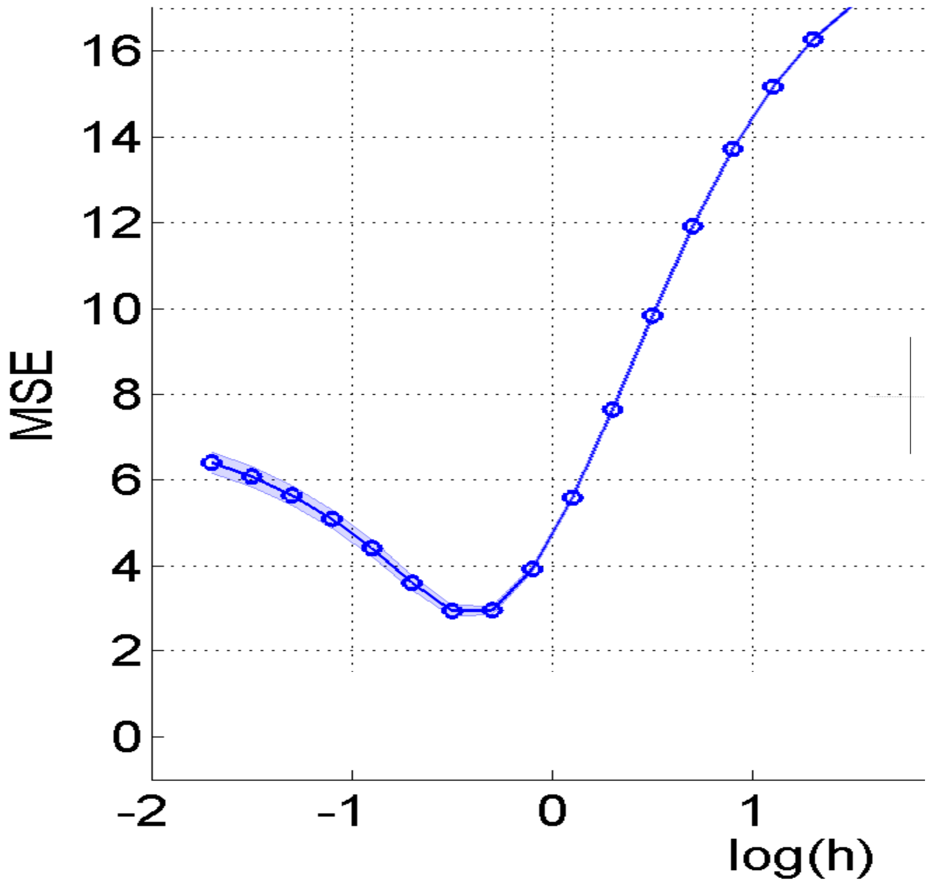
$\nabla^2 p_1, \nabla^2 p_2, \; p_1, \; p_2$ ← Obtain from generative models

[Y.-K. Noh et al. *IEEE TPAMI*, 2018]



Bandwidth and Nadaraya-Watson Regression

$$\widehat{y}_N(\mathbf{x}) = \frac{\sum_{i=1}^{N} K_h(\mathbf{x}_i, \mathbf{x}) y_i}{\sum_{j=1}^{N} K_h(\mathbf{x}_j, \mathbf{x})}$$



Bias Analysis

- *k*-Nearest Neighbor Classification

$$\lim_{N\to\infty} \widehat{y}_N(\mathbf{x}) = \mathbb{E}_{p(y|\mathbf{x})}[y] \qquad (h \to 0)$$

→ Minimizes mean square error (MSE)

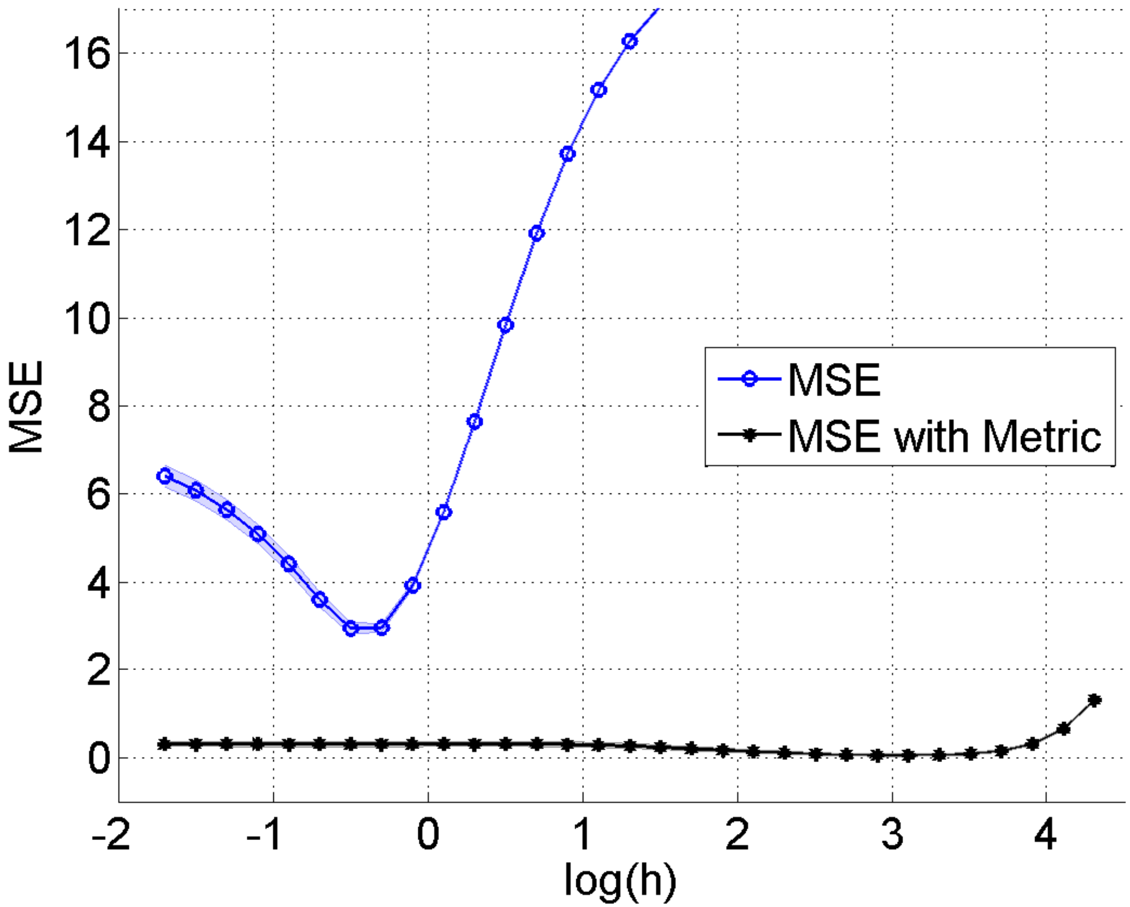
→ Metric independent asymptotic property

- Bias

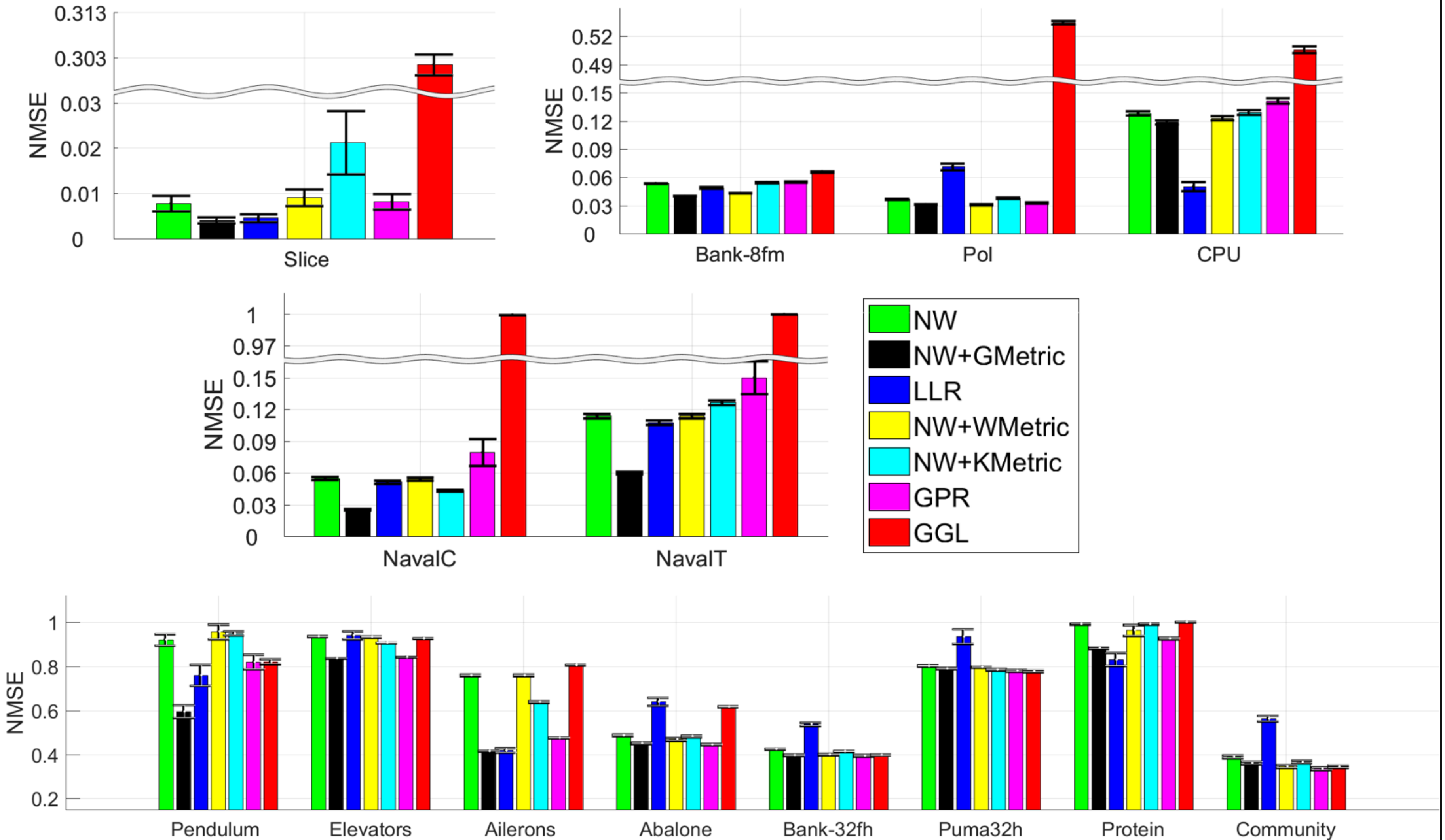
$$\mathbb{E}\left[\widehat{y}(\mathbf{x}) - y(\mathbf{x})\right] = h^2 \left(\frac{\nabla^\top p(\mathbf{x}) \nabla y(\mathbf{x})}{p(\mathbf{x})} + \frac{\nabla^2 y(\mathbf{x})}{2} \right) + o(h^4)$$

For x & y Jointly Gaussian

- Learned metric is not sensitive to the bandwidth



[Y.-K. Noh, et al., *NeurIPS*, 2017]



[Y.-K. Noh, et al., *NeurIPS*, 2017]

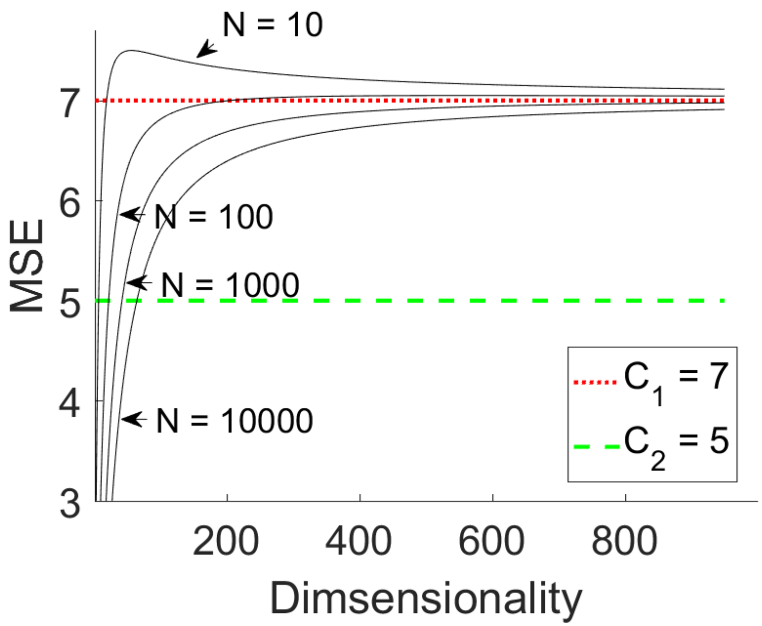
Variance Reduction is Not Critical in High-Dimensions

[Y.-K. Noh, et al., *NeurIPS*, 2017]

$$f(h) = h^4 C_1 + \frac{1}{Nh^D} C_2.$$

$$h_* = N^{-\frac{1}{D+4}} \left(\frac{D \cdot C_2}{4 \cdot C_1} \right)^{\frac{1}{D+4}}$$

$$\lim_{D \to \infty} f(h_*) = C_1$$



Proposition

Reducing the variance is not important in a high dimensional space once the bias is minimized and the bandwidth selection is followed.

Information-theoretic Measure Estimation

$$u_c = N_c d_c^D$$

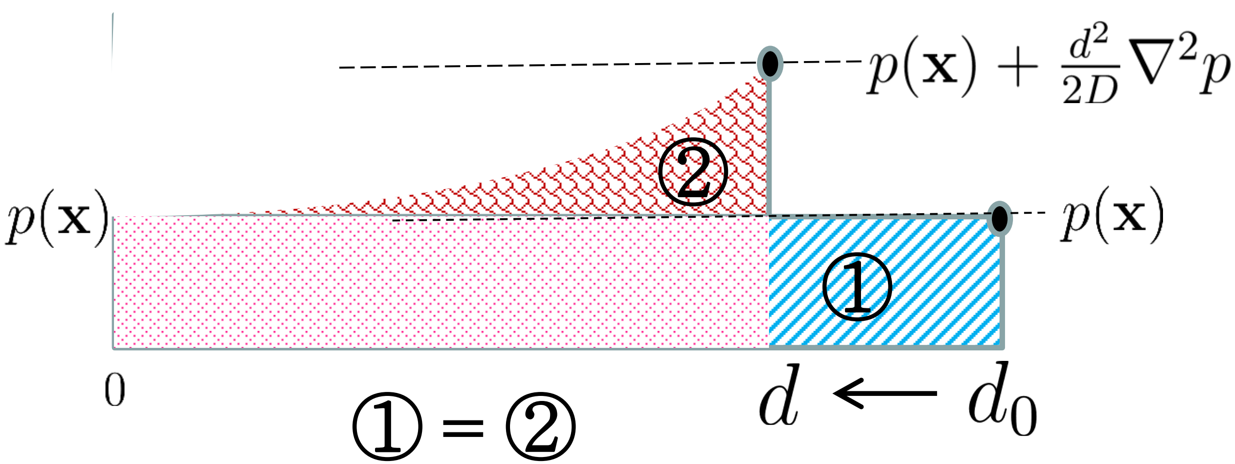
d_c is a distance to the nearest neighbor in class c from $\mathbf{x}$.

Metric invariant

Metric dependent

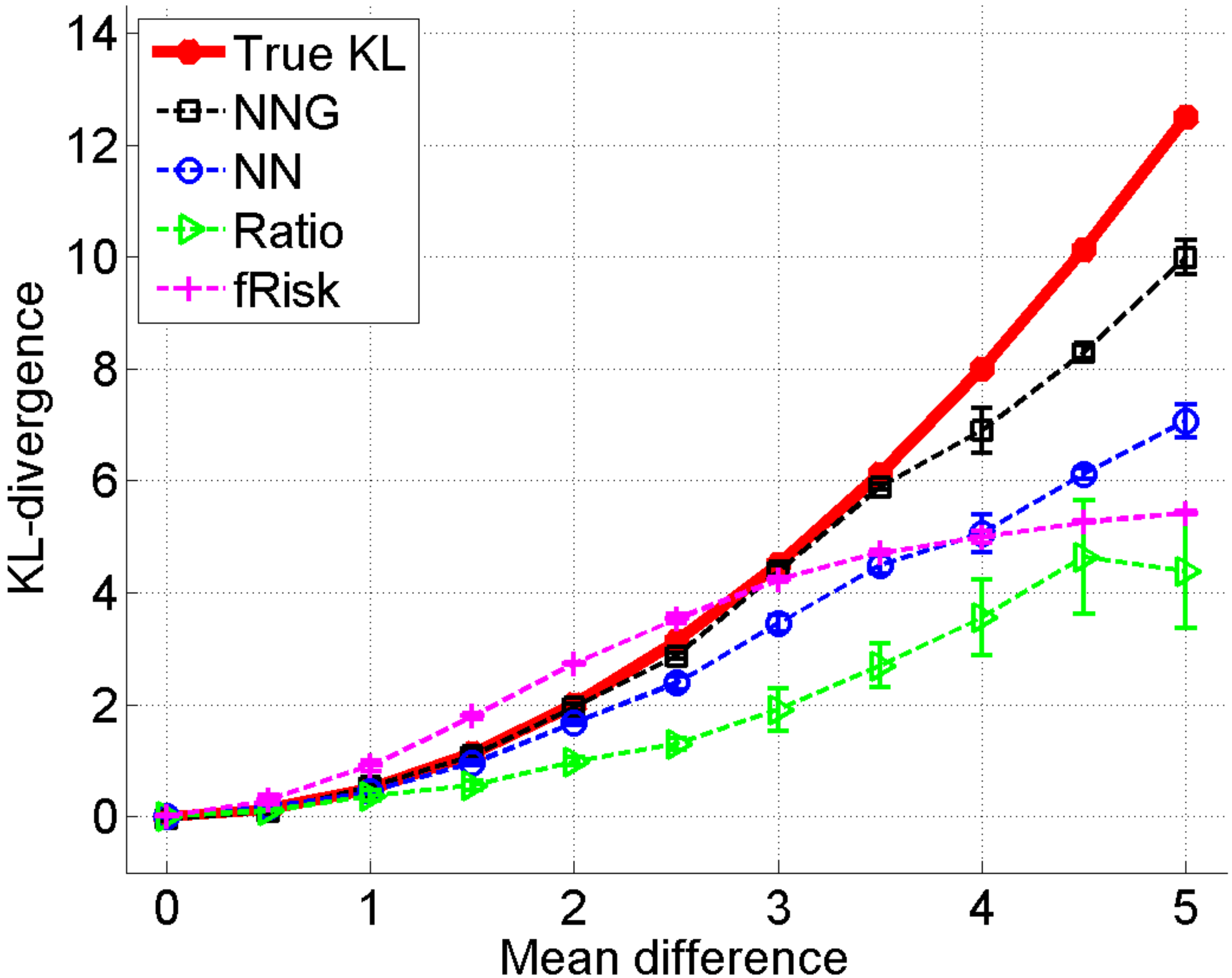
$$\frac{1}{N_1}\sum_{\mathbf{x}\sim p_1}\log\frac{u_2}{u_1} \rightarrow \boxed{-\int p_1 \log\frac{p_2}{p_1}d\mathbf{x}}$$

$$\boxed{+\frac{1}{2D\gamma^{\frac{2}{D}}}\int\left[(N_1p_1)^{-\frac{2}{D}}\frac{\nabla^2 p_1}{p_1} - (N_2p_2)^{-\frac{2}{D}}\frac{\nabla^2 p_2}{p_2}\right]d\mathbf{x}}$$



Increase the KL-Divergence of Two Gaussians and its Estimation

[Y.-K. Noh, et al., *NeCo*, 2018]



MAKING GENERAL ESTIMATORS FOR F-DIVERGENCES

Estimation of the General *f*-Divergences

- Shannon Entropy Estimation

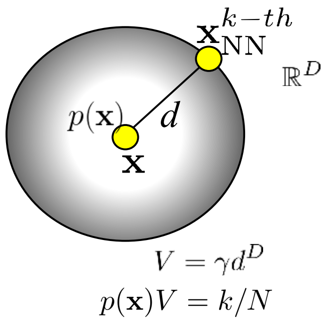
[D. Lombardi and S. Pant, *Phys. Rev. E*, 2016]
[A. Kraskov, H. Stögbauer, and P. Grassberger, *Phys. Rev. E*, 2004]

$$\widehat{H}(X) = \psi(N) - \psi(k) + \log(\gamma) + \frac{1}{N}\sum_{i=1}^{N} \log d(\mathbf{x}_i)^D$$

$$\gamma = \frac{\pi^{\frac{D}{2}}}{\Gamma(1 + D/2)}, \qquad \psi(t) = \Gamma(t)^{-1}\frac{d\Gamma(t)}{dt}$$

Note that $p(\mathbf{x})V = k/N$

In this case, $p(\mathbf{x}) = \frac{k}{VN} = \frac{k}{(\gamma d(\mathbf{x})^D) \cdot N}$



Density Estimator and Entropy Estimator

- Loftsgaarden and Quesenberry (1965)

$$\widehat{p}(\mathbf{x}) = \frac{k}{\gamma N d(\mathbf{x})^D}$$

- Shannon Entropy Estimator

$$\begin{aligned}\widehat{H}(X) &= \psi(N) - \psi(k) + \log(\gamma) + \frac{1}{N}\sum_{i=1}^{N} \log d(\mathbf{x}_i)^D \\ &= -\frac{1}{N}\sum_{i=1}^{N} \log \widehat{p}(\mathbf{x}_i) + (\psi(N) - \log(N)) - (\psi(k) - \log(k))\end{aligned}$$

Historical Remarks of Making Plug-in Estimators

[N. Leonenko, L. Pronzato, &V. Savani, *Annals of Statistics*, 2008]
[B. Poczos and J. Schneider, *AISTATS*, 2011]

Shannon entropy

$$\widehat{H}(X) = -\frac{1}{N}\sum_{i=1}^{N}\log \widehat{p}(\mathbf{x}_i) - \psi(k)$$

Plug-in and correction

$$\to -\int p(\mathbf{x})\log p(\mathbf{x})d\mathbf{x}$$

Rényi and Tsallis entropies

$$\widehat{H}_q(X) = \frac{1}{N}\frac{\Gamma(k)}{\Gamma(k+1-q)}\sum_{i=1}^{N}\left(\widehat{p}(\mathbf{x}_i)\right)^{1-q}$$

Plug-in and correction

$$\to -\int p(\mathbf{x})^q d\mathbf{x}$$

[Moon, K. & Hero, A., 2014] considers the general *f*-divergence plug-in estimator

Plug-in Nearest Neighbor f-divergence Estimator

- Kullback-Leibler Divergence

$$\frac{1}{N}\sum_{i=1}^{N}\log\widehat{p}(\mathbf{x}_i)-\log\widehat{q}(\mathbf{x}_i) \Rightarrow -\int p(\mathbf{x})\log\left(\frac{q(\mathbf{x})}{p(\mathbf{x})}\right)d\mathbf{x}$$

- Tsallis-alpha Divergence

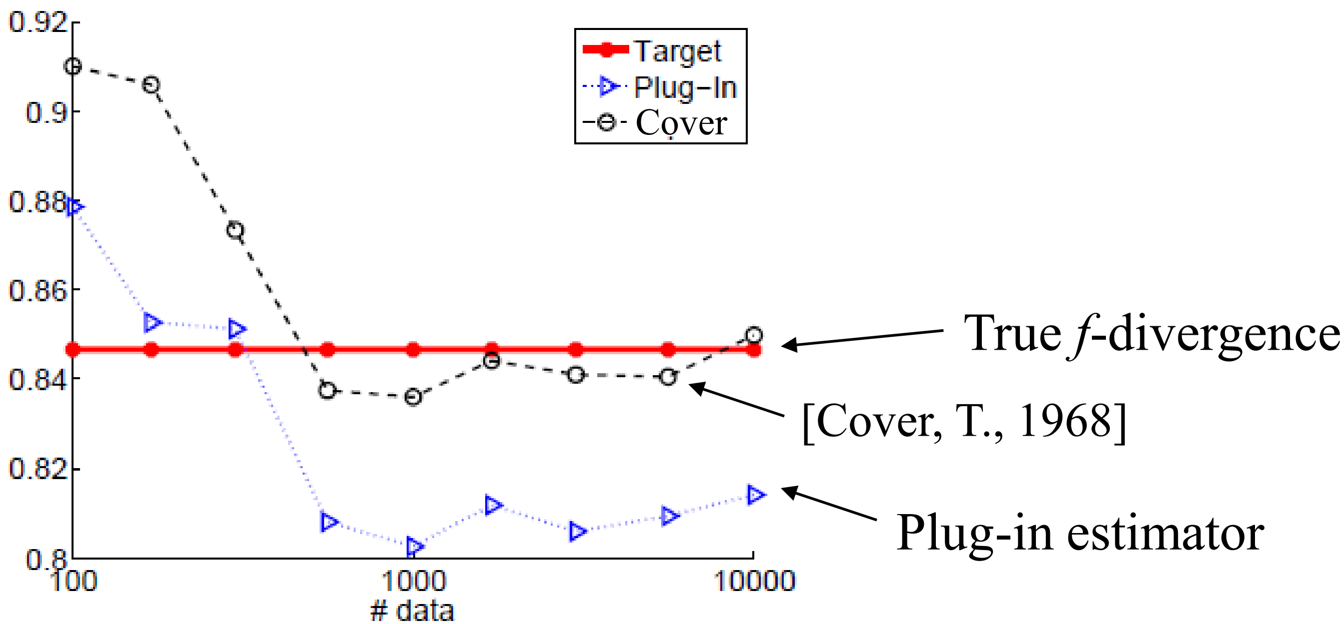
$$\frac{1}{\alpha-1}\left(\frac{1}{N}\frac{\Gamma(k)^2}{\Gamma(k-\alpha+1)\Gamma(k+\alpha-1)}\sum_{i=1}^{N}\left(\frac{\widehat{q}(\mathbf{x}_i)}{\widehat{p}(\mathbf{x}_i)}\right)^{1-\alpha}-1\right)$$

$$\Rightarrow \frac{1}{\alpha-1}\left(\int\left(\frac{p(\mathbf{x})}{q(\mathbf{x})}\right)^{\alpha}p(\mathbf{x})d\mathbf{x}-1\right)$$

Plug-in methods do not work for general *f*-divergences

[Cover, T., 1968]

$$\frac{1}{N_p}\left(\sum_{i=1}^{N_p}\mathbf{1}(d_p > d_q)\right) \longrightarrow \int \frac{p(\mathbf{x})q(\mathbf{x})}{p(\mathbf{x})+q(\mathbf{x})}d\mathbf{x}$$



(b) Nonparametric estimator performance

[Noh, Y.-K. Ph.D. thesis, 2011]

Obtaining the General *f*-Divergence Estimator

$$X_{1:m} \sim p(\mathbf{x}), \quad Y_{1:n} \sim q(\mathbf{x})$$

$$\widehat{T}(X_{1:m}, Y_{1:n}) = \frac{1}{m} \sum_{i=1}^{m} \phi(u(\mathbf{x}_i), v(\mathbf{x}_i)) \qquad \begin{aligned} u &= m\gamma d_p^D \\ v &= n\gamma d_q^D \end{aligned}$$

$$\rightarrow \int p(\mathbf{x}) f\left(\frac{p(\mathbf{x})}{q(\mathbf{x})}\right) d\mathbf{x}$$

$$\phi(u(\mathbf{x}_i), v(\mathbf{x}_i)) = \frac{(k-1)!(l-1)!}{u^{k-1} v^{l-1}} \mathcal{L}^{-1}_{(u,v)} \left[\frac{f(s,t)}{s^k t^l} \right]$$

Inverse Laplace Transform

arXiv:1805.08342

Nearest neighbor density functional estimation based on inverse Laplace transform

Shouvik Ganguly[†] [*] **Jongha Ryu**[†] [*] **Young-Han Kim**[†] **Yung-Kyun Noh**[‡] **Daniel D. Lee**[§]
[†] University of California, San Diego [‡] Seoul National University [§] University of Pennsylvania
[†] {shgangul,jongharyu,yhk}@ucsd.edu,[‡]nohyung@snu.ac.kr,[§]ddlee@seas.upenn.edu

Summary

- Asymptotically, nearest neighbor methods are very nice. (*In terms of Theory!!*)
- With finite samples, bias treatment using geometry change can improve the conventional nonparametric methods significantly (in high-dimensional space).
- General and systematic way of obtaining *f*-divergence using nearest neighbor information.

THANK YOU

Yung-Kyun Noh

nohyung@snu.ac.kr