NIPS2007, Vancouver, Canada Dec.3-8, 2007

Direct Importance Estimation with Model Selection and Its Application to Covariate Shift Adaptation

Masashi Sugiyama
Tokyo Institute of Technology, Tokyo, Japan
Shinichi Nakajima
Nikon Corporation, Saitama, Japan
Hisashi Kashima
IBM Research, Kanagawa, Japan
Paul von Bünau
Technical University Berlin, Berlin, Germany
Motoaki Kawanabe
Fraunhofer FIRST, Berlin, Germany

Abstract

When training and test samples follow different input distributions (i.e., the situation called *covariate shift*), the maximum likelihood estimator is known to lose its consistency. For regaining consistency, the log-likelihood terms need to be weighted according to the *importance* (i.e., the ratio of test and training input densities). Thus, accurately estimating the importance is one of the key tasks in covariate shift adaptation. A naive approach is to first estimate training and test input densities and then estimate the importance by the ratio of the density estimates. However, since density estimation is a hard problem, this approach tends to perform poorly especially in high dimensional cases. In this paper, we propose a direct importance estimation method that does not require the input density estimates. Our method is equipped with a natural model selection procedure so tuning parameters such as the kernel width can be objectively optimized. This is an advantage over a recently developed method of direct importance estimation. Simulations illustrate the usefulness of our approach.

Common Assumption in Supervised Learning

- In supervised learning, we always assume

Training and test samples are drawn from the same distribution

$$P_{train}(\boldsymbol{x}, y) = P_{test}(\boldsymbol{x}, y)$$

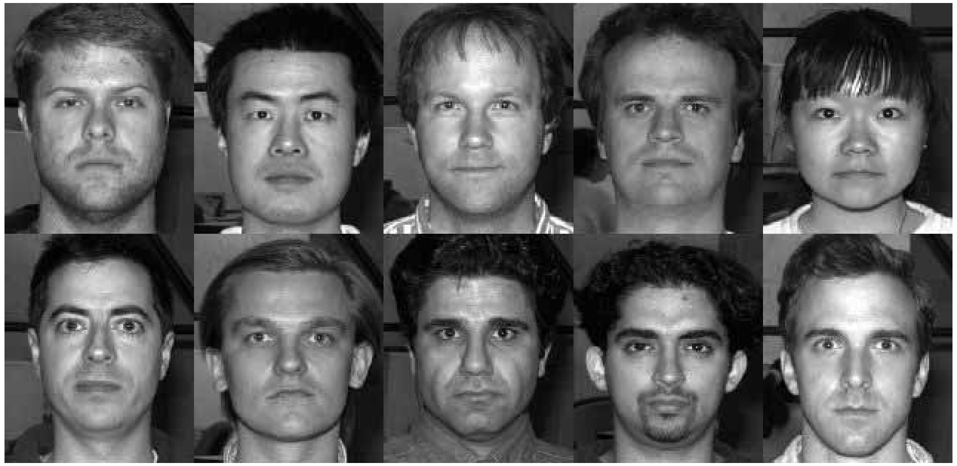
- Is this assumption really true?

Not Always True!

Face Recognition

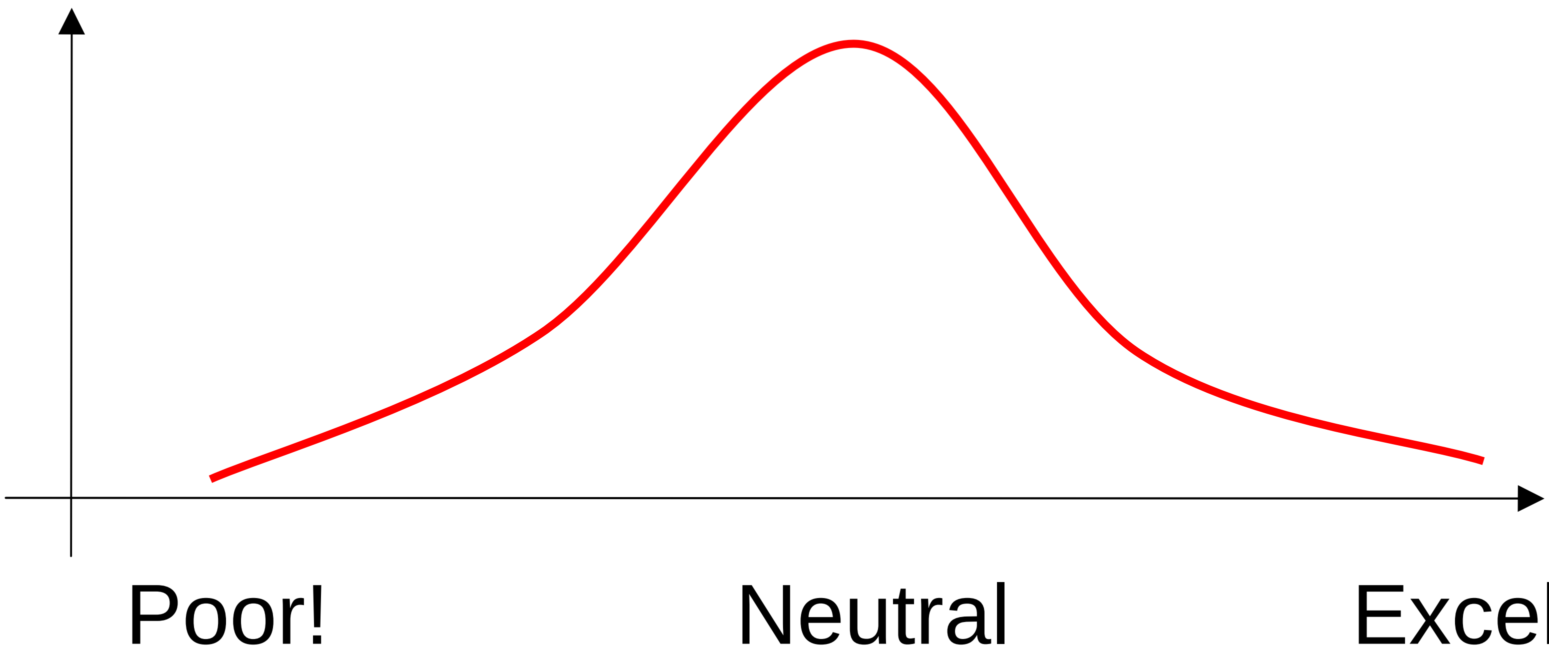
- We tend to collect easy-to-gather samples for training.
  - Training: less women in research labs
  - Test: almost 50-50 in general

The Yale Face Database B



Survey Sampling

- Those who have strong opinions tend to reply to questionnaires.
  - Training: extreme opinions
  - Test: most people are neutral



Brain-Computer Interface

- Sample generation mechanism varies.
  - Input: EEG signals
  - Output: “left” or “right” commands
  - Different mental conditions between training (sleepy…) and test (exciting!) phases may change the EEG signals.

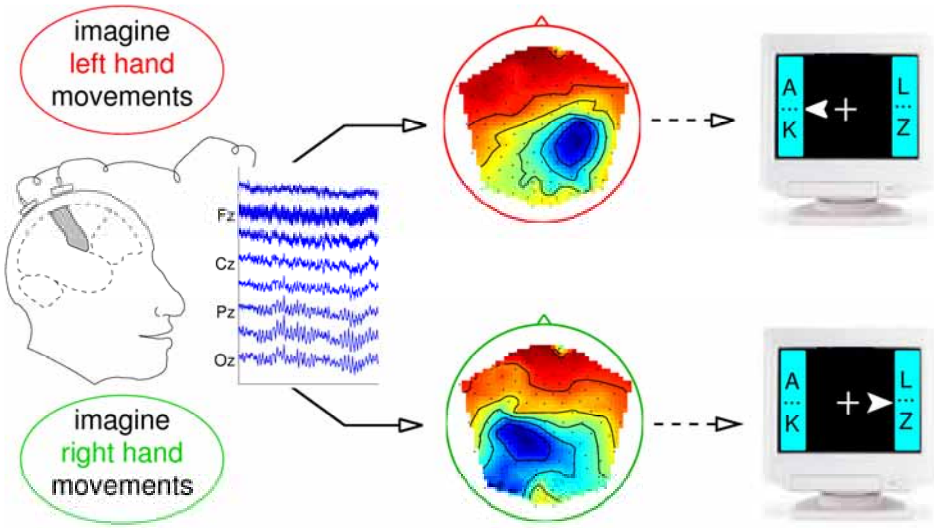
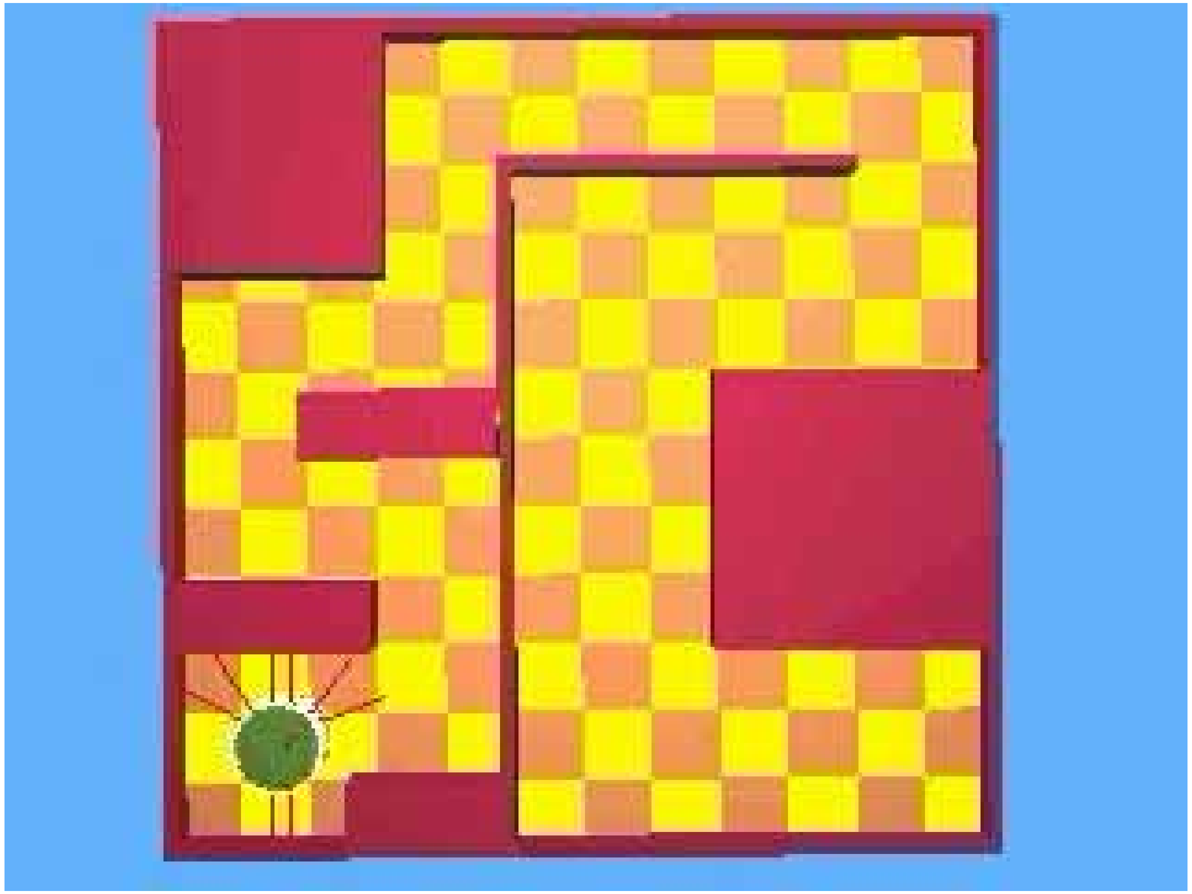


Figure provided by Fraunhofer FIRST, Berlin, Germany

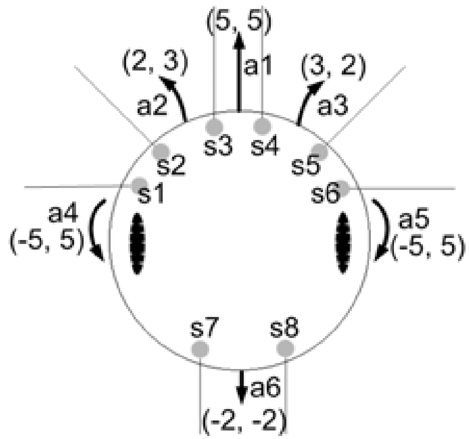
berlin brain computer interface

Robot Control by Reinforcement Learning

- Updating a robot's behavior causes a distribution change.



Khepera Robot



Covariate Shift

- However, no chance for generalization if training and test samples have nothing in common.

$$P_{train}(\boldsymbol{x}, y) \neq P_{test}(\boldsymbol{x}, y)$$

→ We need a (reasonable) constraint

- Covariate shift
  - Input distribution changes:

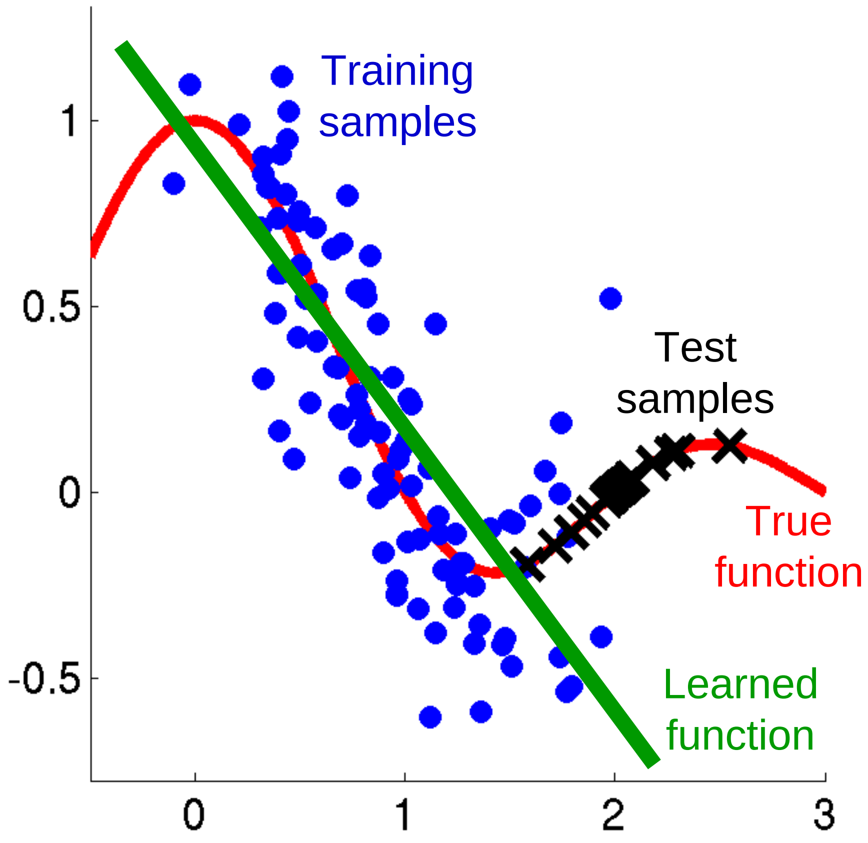
$$P_{train}(\boldsymbol{x}) \neq P_{test}(\boldsymbol{x})$$

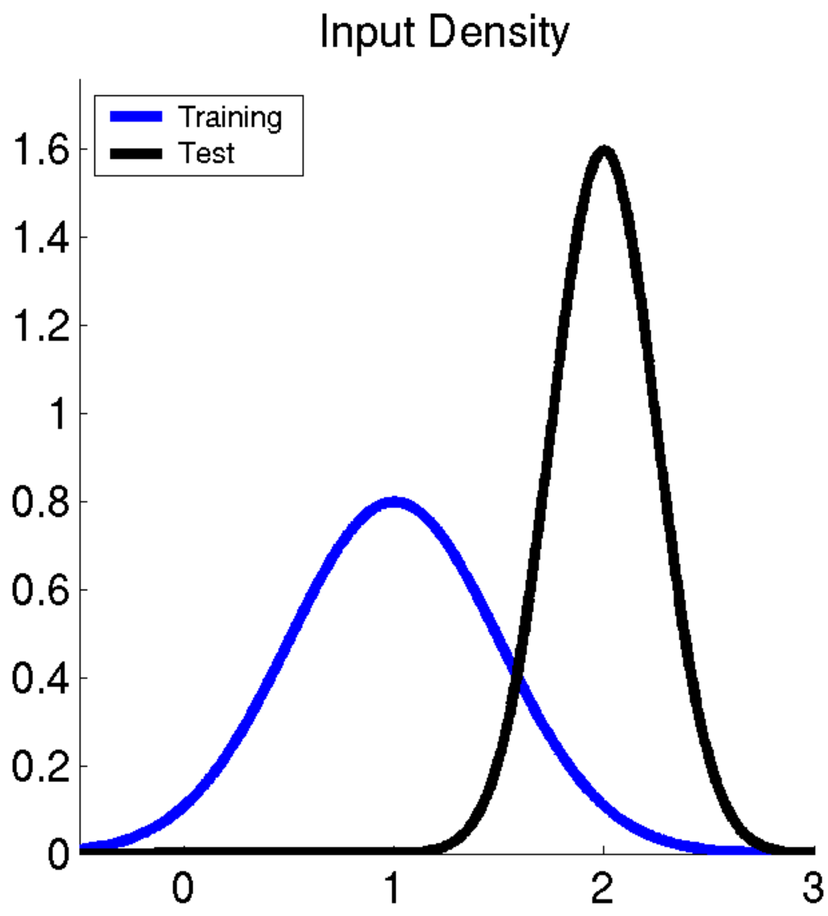
  - Functional relation remains unchanged:

$$P_{train}(y|\boldsymbol{x}) = P_{test}(y|\boldsymbol{x})$$

Illustration of Covariate Shift

(Weak) extrapolation:
Predict output values outside training region





Bias and Variance

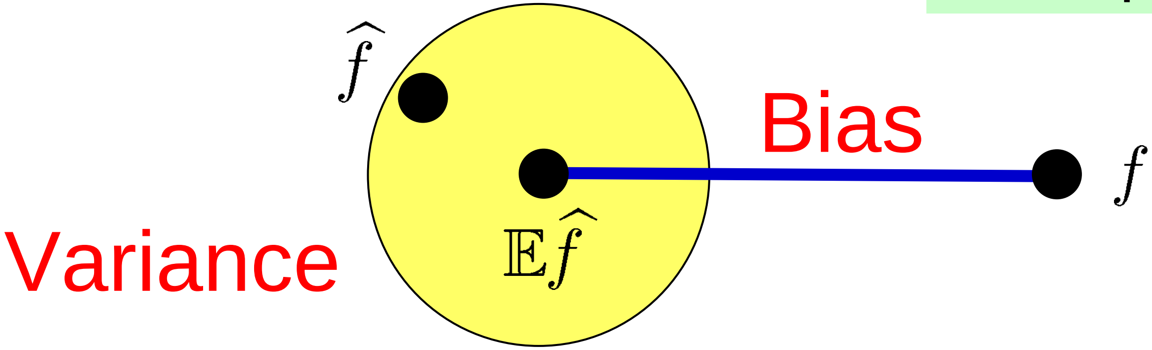
- Generalization error (expected test error):

$$\mathbb{E}\int\left(\widehat{f}(\boldsymbol{x})-f(\boldsymbol{x})\right)^2 p_{test}(\boldsymbol{x})d\boldsymbol{x}$$

$$=\int\left(\mathbb{E}\widehat{f}(\boldsymbol{x})-f(\boldsymbol{x})\right)^2 p_{test}(\boldsymbol{x})d\boldsymbol{x} \qquad \text{Bias}$$

$$+\,\mathbb{E}\int\left(\mathbb{E}\widehat{f}(\boldsymbol{x})-\widehat{f}(\boldsymbol{x})\right)^2 p_{test}(\boldsymbol{x})d\boldsymbol{x} \qquad \text{Variance}$$

$\mathbb{E}$: expectation over samples



Model Specification

- Model is said to be correctly specified if

$$^{\exists}\boldsymbol{\alpha}^*, \ \widehat{f}(\boldsymbol{x};\boldsymbol{\alpha}^*) = f(\boldsymbol{x})$$

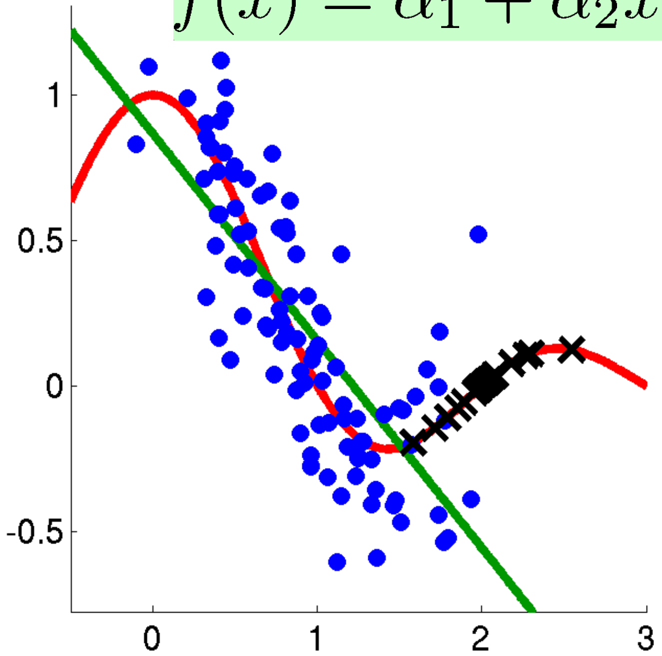
- In practice, our model may not be correct.
- Therefore, we need to explicitly deal with misspecified models!

Ordinary Least-Squares (OLS)

$$\min_{\boldsymbol{\alpha}} \left[\sum_{i=1}^{n_{train}} \left(\widehat{f}(\boldsymbol{x}_i^{train}) - y_i^{train} \right)^2 \right]$$

- If model is correct:
  - OLS minimizes bias asymptotically
- If model is misspecified:
  - OLS does not minimize bias even asymptotically.

$\hat{f}(x) = \alpha_1 + \alpha_2 x$



We want to reduce bias!

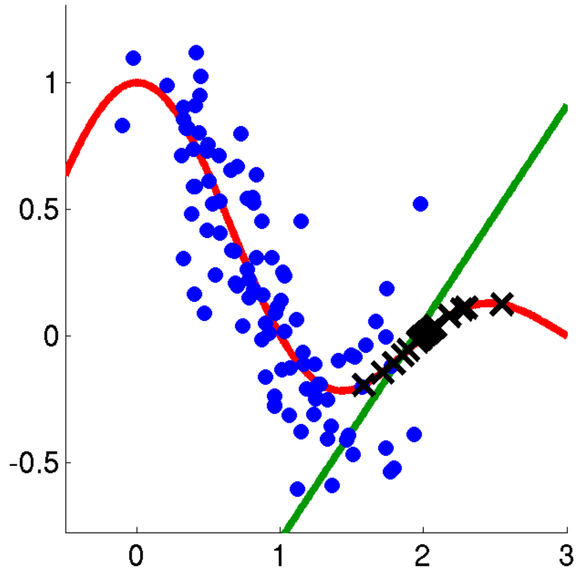
Importance-Weighted LS (IWLS)

$$\min_{\boldsymbol{\alpha}} \left[\sum_{i=1}^{n_{train}} w(\boldsymbol{x}_i^{train}) \left(\widehat{f}(\boldsymbol{x}_i^{train}) - y_i^{train} \right)^2 \right]$$

$$w(\boldsymbol{x}) = \frac{p_{test}(\boldsymbol{x})}{p_{train}(\boldsymbol{x})}$$:Importance

$p_{train}(\boldsymbol{x})$:Assumed strictly positive

- Even for misspedified models, IWLS minimizes bias asymptotically.
- We need to estimate the importance in practice.



Importance Estimation

- Setting: training and test inputs are given

$$\{\boldsymbol{x}_i^{train}\}_{i=1}^{n_{train}} \overset{i.i.d.}{\sim} p_{train}(\boldsymbol{x})$$

$$\{\boldsymbol{x}_j^{test}\}_{j=1}^{n_{test}} \overset{i.i.d.}{\sim} p_{test}(\boldsymbol{x})$$

- Naïve approach: estimate $p_{train}(\boldsymbol{x})$ and $p_{test}(\boldsymbol{x})$ separately, and take the ratio of the density estimates
- Naïve approach does not work well since density estimation is hard in high dimensions.

Modeling Importance Function

- We use a linear model:

$$\widehat{w}(\boldsymbol{x}) = \sum_{\ell=1}^{t} \theta_\ell \phi_\ell(\boldsymbol{x}) \qquad \theta_\ell, \phi_\ell(\boldsymbol{x}) \geq 0$$

- Test density is approximated by

$$\widehat{p}_{test}(\boldsymbol{x}) = \widehat{w}(\boldsymbol{x}) p_{train}(\boldsymbol{x})$$

Learn $\{\theta_\ell\}_{\ell=1}^{t}$ so that $\widehat{p}_{test}(\boldsymbol{x})$ approximates $p_{test}(\boldsymbol{x})$ well.

Kullback-Leibler Divergence

$$\min_{\{\theta_i\}_{i=1}^{t}} KL[p_{test}(\boldsymbol{x})||\widehat{p}_{test}(\boldsymbol{x})]$$

$$\widehat{p}_{test}(\boldsymbol{x}) = \widehat{w}(\boldsymbol{x})p_{train}(\boldsymbol{x})$$

- $KL[p_{test}(\boldsymbol{x})||\widehat{w}(\boldsymbol{x})p_{train}(\boldsymbol{x})]$

$$= \int p_{test}(\boldsymbol{x}) \log \frac{p_{test}(\boldsymbol{x})}{\widehat{w}(\boldsymbol{x})p_{train}(\boldsymbol{x})} d\boldsymbol{x}$$

$$= \int p_{test}(\boldsymbol{x}) \log \frac{p_{test}(\boldsymbol{x})}{p_{train}(\boldsymbol{x})} d\boldsymbol{x}$$ (constant)

$$- \int p_{test}(\boldsymbol{x}) \log \widehat{w}(\boldsymbol{x}) d\boldsymbol{x}$$ (relevant)

Learning Importance Function

- Thus

$$\min_{\{\theta_\ell\}_{\ell=1}^t} KL[\widehat{w}(\boldsymbol{x})p_{train}(\boldsymbol{x})||p_{test}(\boldsymbol{x})]$$

$$\Longleftrightarrow \max_{\{\theta_\ell\}_{\ell=1}^t} \int p_{test}(\boldsymbol{x}) \log \widehat{w}(\boldsymbol{x}) d\boldsymbol{x}$$

(objective function)

- Since $\widehat{p}_{test}(\boldsymbol{x}) = \widehat{w}(\boldsymbol{x})p_{train}(\boldsymbol{x})$ is density,

$$\int \widehat{w}(\boldsymbol{x})p_{train}(\boldsymbol{x})d\boldsymbol{x} = 1$$

(constraint)

KLIEP (Kullback-Leibler Importance Estimation Procedure)

$$\max_{\{\theta_\ell\}_{\ell=1}^{t}} \left[\sum_{j=1}^{n_{test}} \log \widehat{w}(\boldsymbol{x}_j^{test}) \right]$$

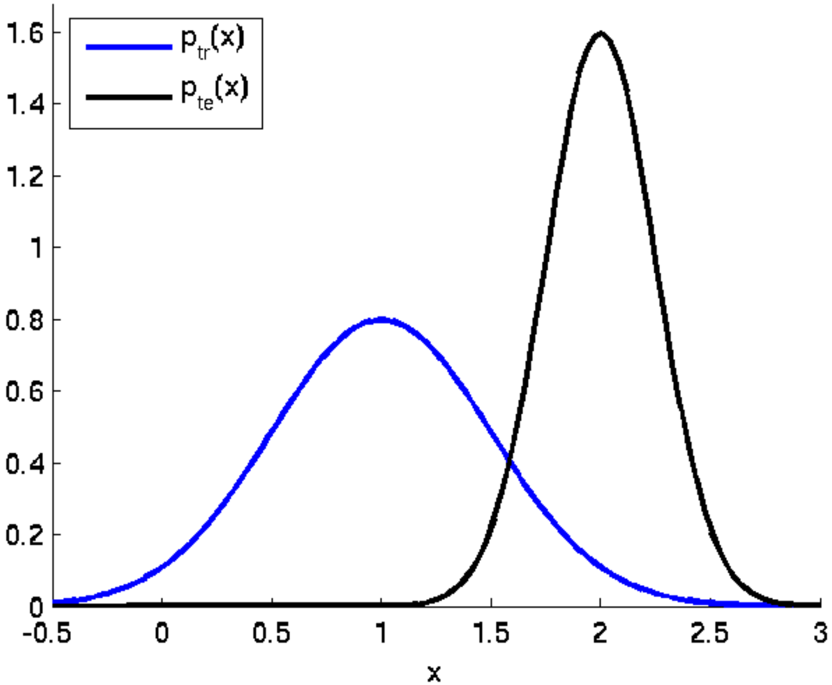
$$\widehat{w}(\boldsymbol{x}) = \sum_{\ell=1}^{t} \theta_\ell \phi_\ell(\boldsymbol{x})$$

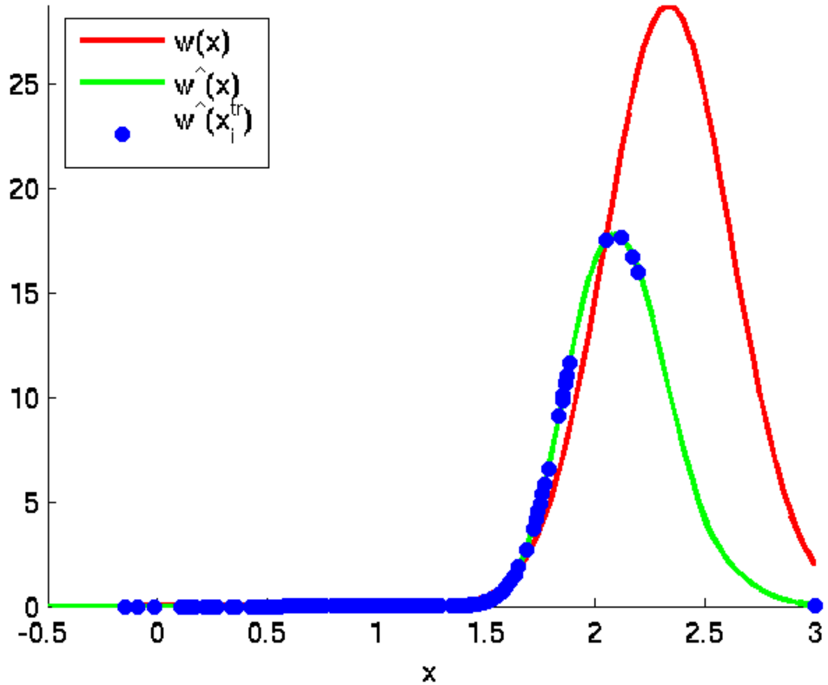
$$\text{subject to} \quad \sum_{i=1}^{n_{train}} \widehat{w}(\boldsymbol{x}_i^{train}) = n_{train}$$

$$\theta_1, \theta_2, \ldots, \theta_t \geq 0$$

- Convexity: unique global solution is available
- Sparse solution: prediction is fast!

Examples





$$\widehat{w}(\boldsymbol{x}) = \sum_{\ell=1}^{n_{test}} \theta_\ell K(\boldsymbol{x}, \boldsymbol{x}_\ell^{test})$$

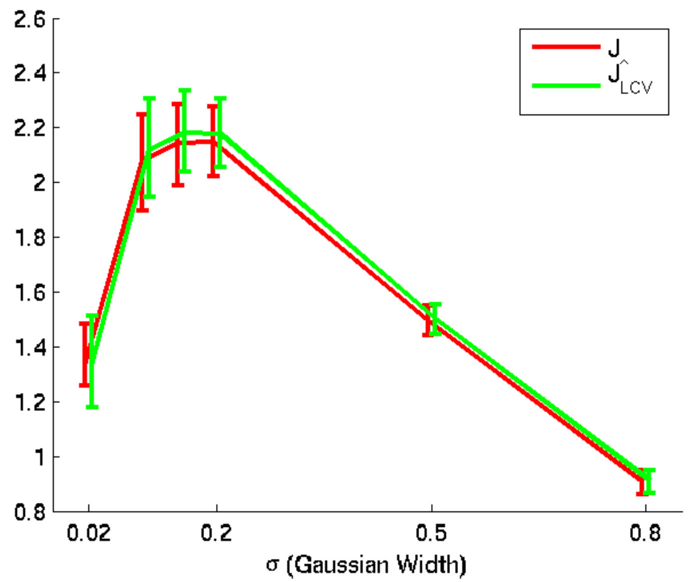
$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\sigma^2}\right)$$

Model Selection of KLIEP

- How to choose tuning parameters (such as Gaussian width)?
- Likelihood cross-validation:
  - Divide test samples $\{\boldsymbol{x}_j^{test}\}_{j=1}^{n_{test}}$ into $\mathcal{X}$ and $\mathcal{X}'$.
  - Learn importance from $\mathcal{X}$.
  - Estimate the likelihood using $\mathcal{X}'$.

$$\frac{1}{|\mathcal{X}'|}\sum_{\boldsymbol{x}'\in\mathcal{X}'}\log\widehat{w}_{\mathcal{X}}(\boldsymbol{x}')$$

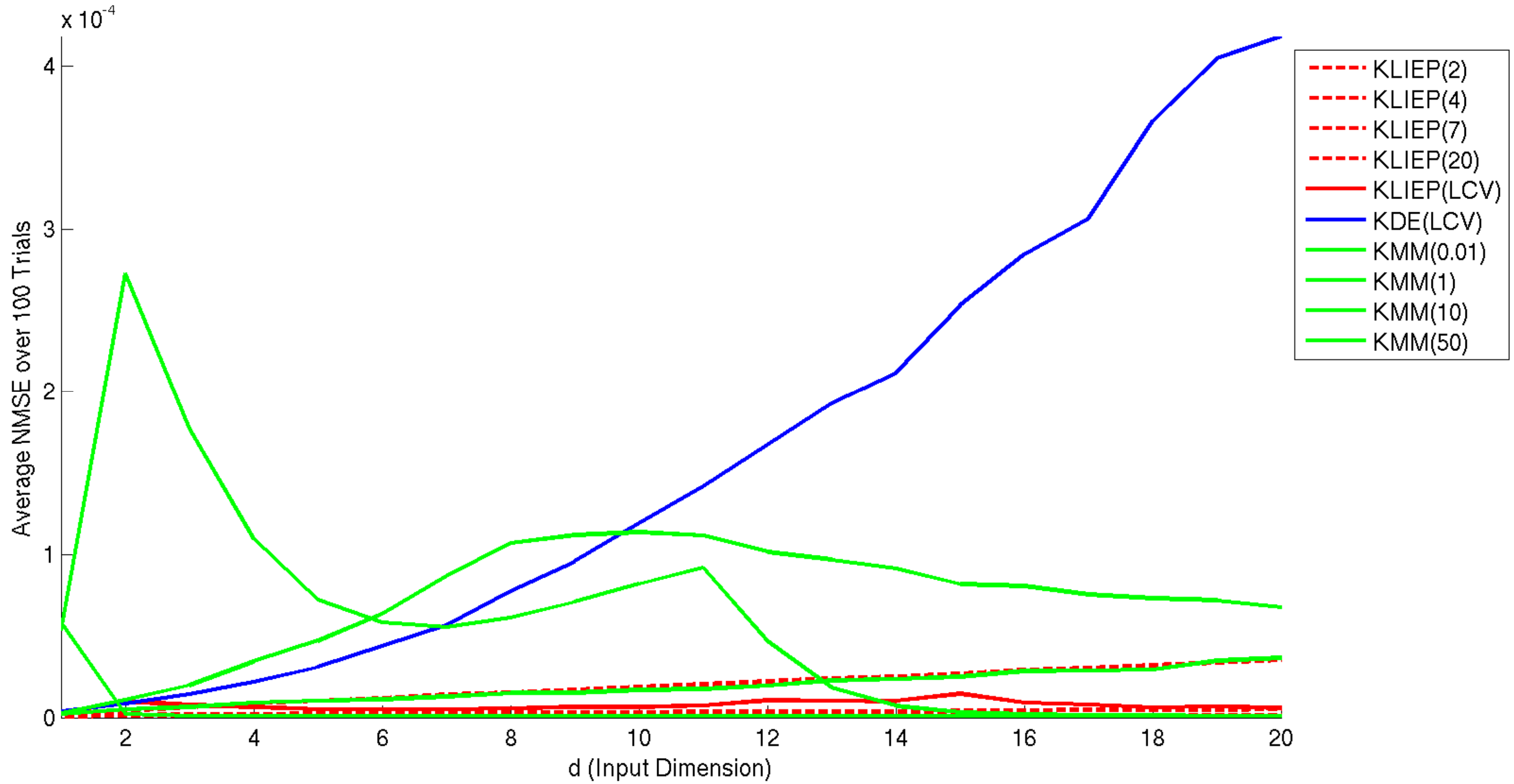
- This gives an unbiased estimate of KL (up to an irrelevant constant).



Illustrative Experiments: Setup

- Kernel density estimator (KDE):
  - Separately estimate training and test input densities.
  - Gaussian kernel width is chosen by likelihood cross-validation.
- Kernel mean matching (KMM): (Huang *et al*., NIPS2006)
  - Direct importance estimation method using universal reproducing kernel Hilbert spaces
  - There is no model selection method for kernel width; we test several different widths.
- Input distributions: standard Gaussian with
  - Training: mean (0,0,…,0)
  - Test: mean (1,0,…,0)

Illustrative Experiments: Results



- KDE: Suffers from the curse of dimensionality
- KMM: Performance depends on kernel width
- KLIEP: Model selection by LCV works well

Regression/Classification

- Goal: given $\{\boldsymbol{x}_i^{train}, y_i^{train}\}_{i=1}^{n_{train}}, \{\boldsymbol{x}_j^{test}\}_{j=1}^{n_{test}}$, predict $\{y_j^{test}\}_{j=1}^{n_{test}}$.
- Gaussian kernel model:

$$\widehat{f}(\boldsymbol{x}) = \sum_{\ell=1}^{50} \alpha_\ell \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{m}_\ell\|^2}{2h^2}\right)$$

$$\{\boldsymbol{m}_\ell\}_{\ell=1}^{50} \subset \{\boldsymbol{x}_j^{test}\}_{j=1}^{n_{test}}$$

- Regularized IWLS:

$$\min_{\boldsymbol{\alpha}} \left[\sum_{i=1}^{n_{train}} \widehat{w}(\boldsymbol{x}_i^{train})\left(\widehat{f}(\boldsymbol{x}_i^{train}) - y_i^{train}\right)^2 + \gamma\|\boldsymbol{\alpha}\|^2\right].$$

- Importance is estimated by KLIEP with LCV.
- h, γ are chosen by importance-weighted cross-validation.

(Sugiyama *et al*., JMLR2007)

Results

Normalized test error
(mean and standard deviation over 100 trials)

	Data	Dim	Uniform	KLIEP(LCV)	KDE(LCV)	KMM(0.01)	KMM(0.3)	KMM(1)
Regression	Kin-8fh	8	1.00(0.34)	0.95(0.31)	1.22(0.52)	1.00(0.34)	0.97(0.34)	1.47(0.24)
	Kin-8fm	8	1.00(0.39)	0.86(0.35)	1.12(0.57)	1.00(0.39)	0.93(0.39)	1.72(0.35)
	Kin-8nh	8	1.00(0.26)	0.99(0.22)	1.09(0.20)	1.00(0.26)	1.00(0.22)	1.15(0.25)
	Kin-8nm	8	1.00(0.30)	0.97(0.25)	1.14(0.26)	1.00(0.30)	1.00(0.29)	1.19(0.22)
	Abalone	7	1.00(0.50)	0.94(0.67)	1.02(0.41)	0.99(0.50)	1.03(0.74)	0.93(0.40)
Classification	Image	18	1.00(0.50)	0.92(0.41)	0.98(0.44)	0.97(0.50)	0.98(0.46)	1.06(0.50)
	Ringnorm	20	1.00(0.04)	0.99(0.06)	0.87(0.04)	1.00(0.04)	0.96(0.07)	0.88(0.06)
	Twonorm	20	1.00(0.58)	0.91(0.52)	1.16(0.71)	0.98(0.50)	0.82(0.52)	0.97(0.62)
	Waveform	21	1.00(0.45)	0.93(0.34)	1.05(0.47)	1.00(0.45)	0.94(0.34)	1.00(0.33)

Red: Best method and comparable ones by Wilcoxon signed rank test at significance level 5%