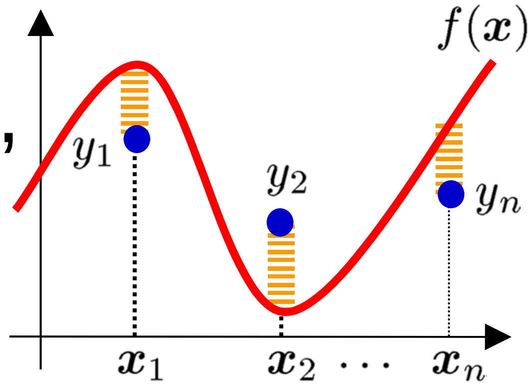
Pattern Information Processing: Covariate Shift Adaptation

Masashi Sugiyama
(Department of Computer Science)

Contact: W8E-505
sugi@cs.titech.ac.jp
http://sugiyama-www.cs.titech.ac.jp/~sugi/

Common Assumption in Supervised Learning

- Goal of supervised learning: From training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$, predict outputs of unseen test samples



- 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!

- Less women in face dataset than reality.
- More criticisms in survey sampling than reality.
- Sample generation mechanism varies over time.

The Yale Face Database B

Covariate Shift

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

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

- Covariate shift:
  - Input distribution changes

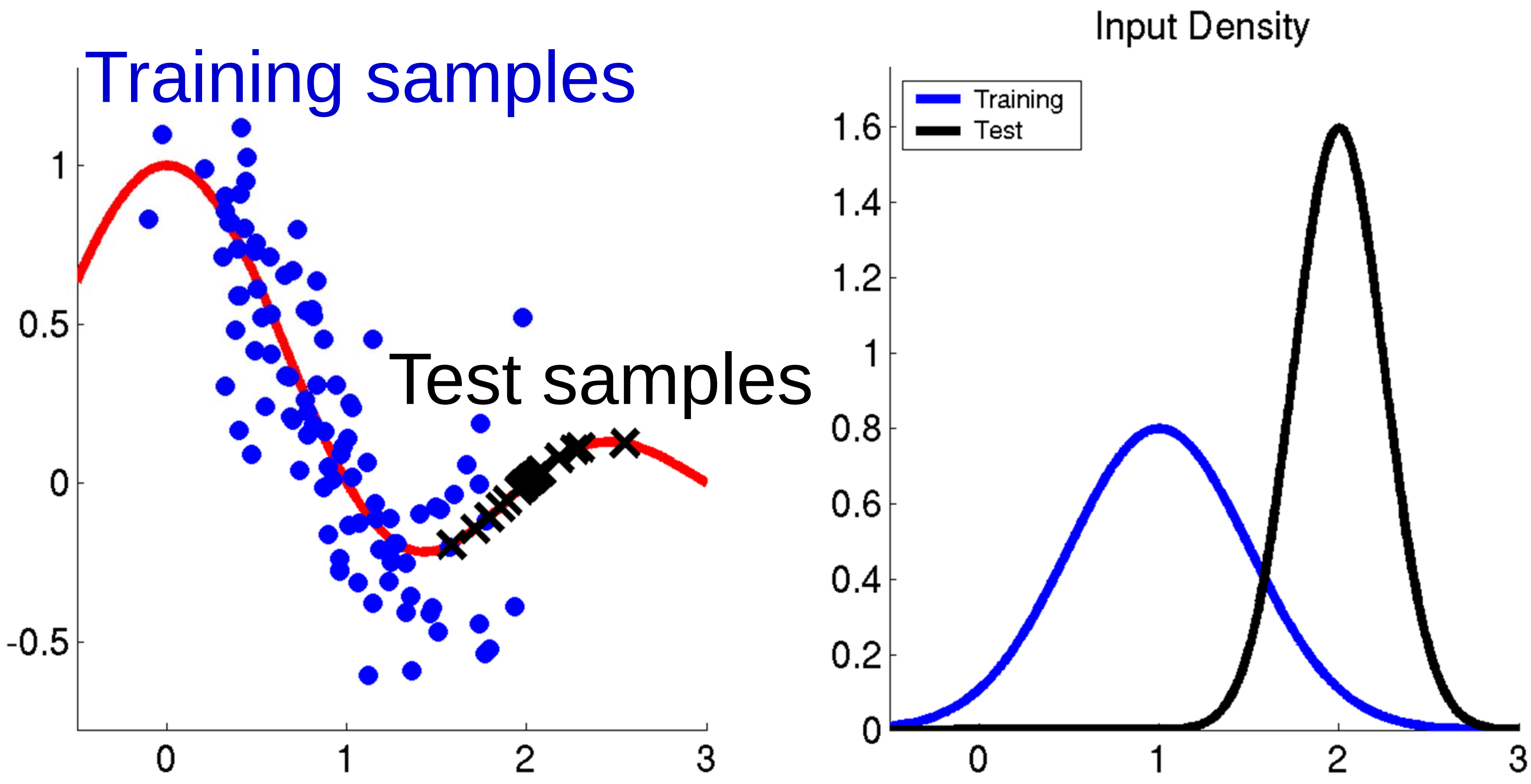
$$P_{train}(\boldsymbol{x}) \neq P_{test}(\boldsymbol{x})$$

  - Functional relation remains unchanged

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

Examples of Covariate Shift

(Weak) extrapolation:
Predict output values outside training region

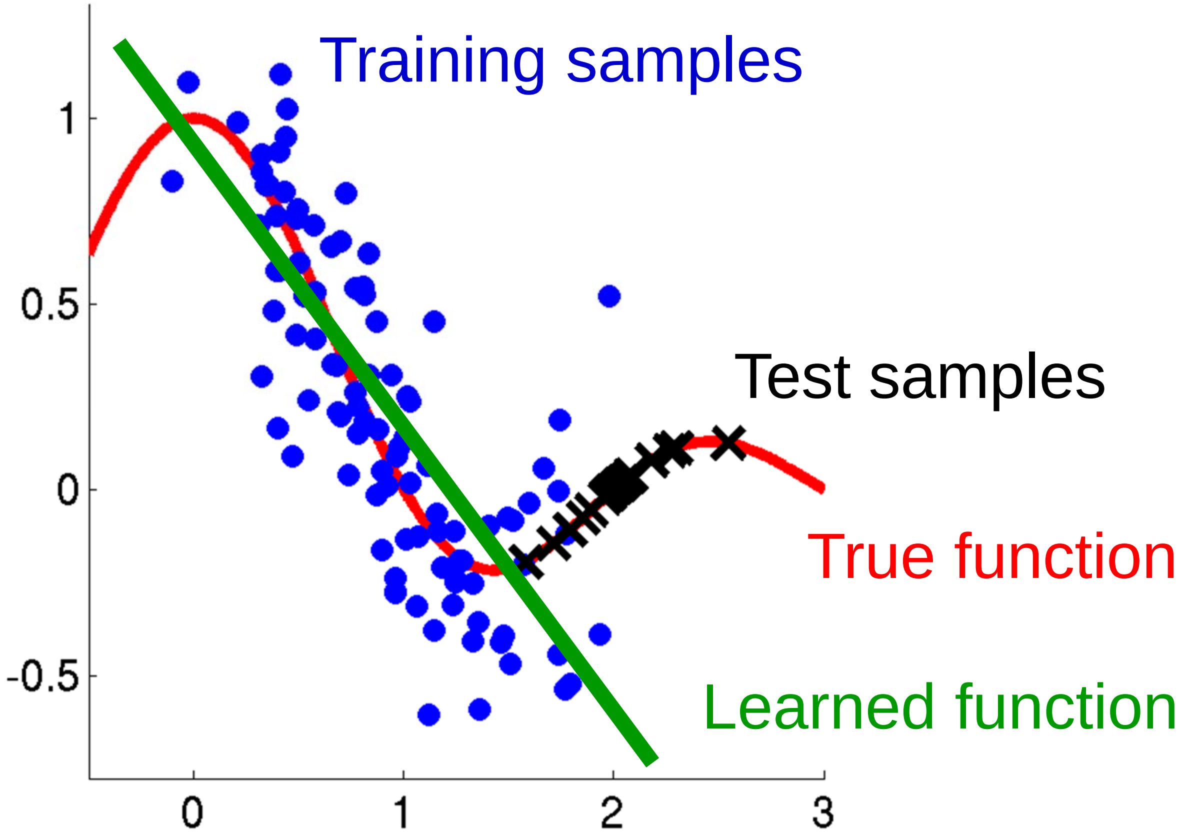


Organization

1. Linear regression under covariate shift
2. Parameter learning
3. Importance estimation
4. Model selection

Covariate Shift

- To illustrate the effect of covariate shift, let’s focus on linear extrapolation

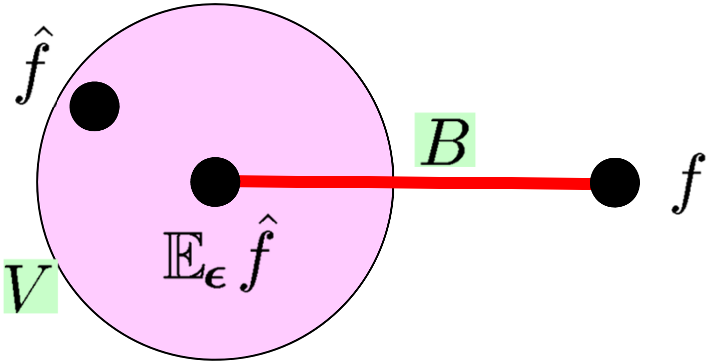


Generalization Error = Bias + Variance

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

$$= \int \left(\mathbb{E}_\epsilon \hat{f}(\boldsymbol{x}) - f(\boldsymbol{x})\right)^2 p_{test}(\boldsymbol{x}) d\boldsymbol{x}$$ Bias

$$+\mathbb{E}_\epsilon \int \left(\mathbb{E}_\epsilon \hat{f}(\boldsymbol{x}) - \hat{f}(\boldsymbol{x})\right)^2 p_{test}(\boldsymbol{x}) d\boldsymbol{x}$$ Variance



$\mathbb{E}_\epsilon$: expectation over noise

Model Specification

- Model is said to be correctly specified if

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

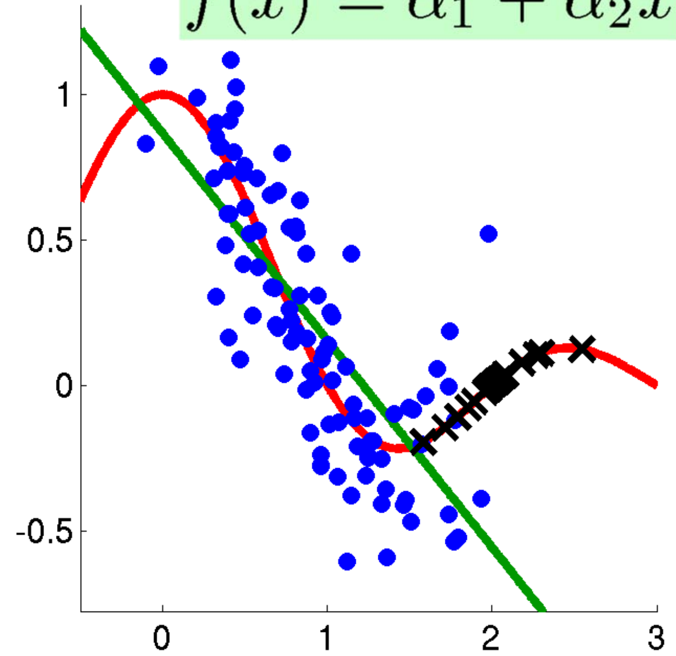
- In practice, our model may not be correct.
- Therefore, we need a theory for misspecified models!

Ordinary Least-Squares

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

- 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!

Law of Large Numbers

- Sample average converges to the population mean:

$$\frac{1}{n}\sum_{i=1}^{n} A(\boldsymbol{x}_i) \longrightarrow \int A(\boldsymbol{x}) p_{train}(\boldsymbol{x}) d\boldsymbol{x}$$

$$\boldsymbol{x}_i \overset{i.i.d.}{\sim} p_{train}(\boldsymbol{x})$$

- We want to estimate the expectation over test input points only using training input points $\{\boldsymbol{x}_i\}_{i=1}^{n}$.

$$\int A(\boldsymbol{t}) p_{test}(\boldsymbol{t}) d\boldsymbol{t} \qquad \boldsymbol{t} \sim p_{test}(\boldsymbol{x})$$

Key Trick: Importance-Weighted Average

- Importance: Ratio of test and training input densities

$$\frac{p_{test}(\boldsymbol{x})}{p_{train}(\boldsymbol{x})}$$

- Importance-weighted average:

$$\frac{1}{n}\sum_{i=1}^{n}\frac{p_{test}(\boldsymbol{x}_i)}{p_{train}(\boldsymbol{x}_i)}A(\boldsymbol{x}_i) \longrightarrow \int \frac{p_{test}(\boldsymbol{x})}{p_{train}(\boldsymbol{x})}A(\boldsymbol{x})p_{train}(\boldsymbol{x})d\boldsymbol{x}$$

$$\boldsymbol{x}_i \overset{i.i.d.}{\sim} p_{train}(\boldsymbol{x}) \qquad = \int A(\boldsymbol{x})p_{test}(\boldsymbol{x})d\boldsymbol{x}$$

$$\boldsymbol{t} \sim p_{test}(\boldsymbol{x})$$

(cf. importance sampling)

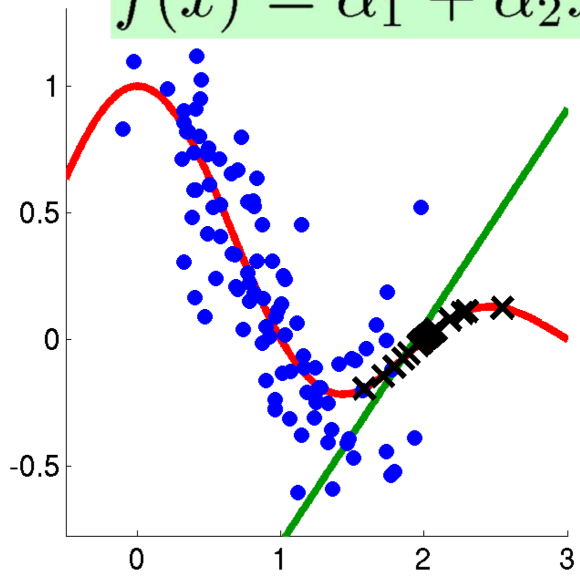
Importance-Weighted LS

$$\min_{\boldsymbol{\alpha}} \left[\sum_{i=1}^{n} \frac{p_{test}(\boldsymbol{x}_i)}{p_{train}(\boldsymbol{x}_i)} \left(\hat{f}(\boldsymbol{x}_i) - y_i \right)^2 \right]$$

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

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

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



Organization

1. Linear regression under covariate shift
2. Parameter learning
3. Importance estimation
4. Model selection

Importance Estimation

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

- Assumption: We have training inputs $\{\boldsymbol{x}_i^{train}\}_{i=1}^{n_{train}}$ and test inputs $\{\boldsymbol{x}_i^{test}\}_{i=1}^{n_{test}}$.
- Naïve approach: Estimate $p_{train}(\boldsymbol{x})$ and $p_{test}(\boldsymbol{x})$ separately, and take the ratio of the density estimates
- This does not work well since density estimation is hard in high dimensions.

Modeling Importance Function

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

- We use a linear model:

$$\widehat{w}(\boldsymbol{x}) = \sum_{i=1}^{t} \theta_i \phi_i(\boldsymbol{x}) \qquad \theta_i, \phi_i(\boldsymbol{x}) \geq 0$$

- Test density is approximated by

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

- Idea: Learn $\{\theta_i\}_{i=1}^{t}$ so that $\widehat{p}_{test}(\boldsymbol{x})$ well approximates $p_{test}(\boldsymbol{x})$.

Kullback-Leibler Divergence

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

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

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

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

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

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

Learning Importance Function

- Thus

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

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

(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)

$$\max_{\{\theta_i\}_{i=1}^{t}} \left[\sum_{i=1}^{n_{test}} \log \widehat{w}(\boldsymbol{x}_i^{test}) \right]$$

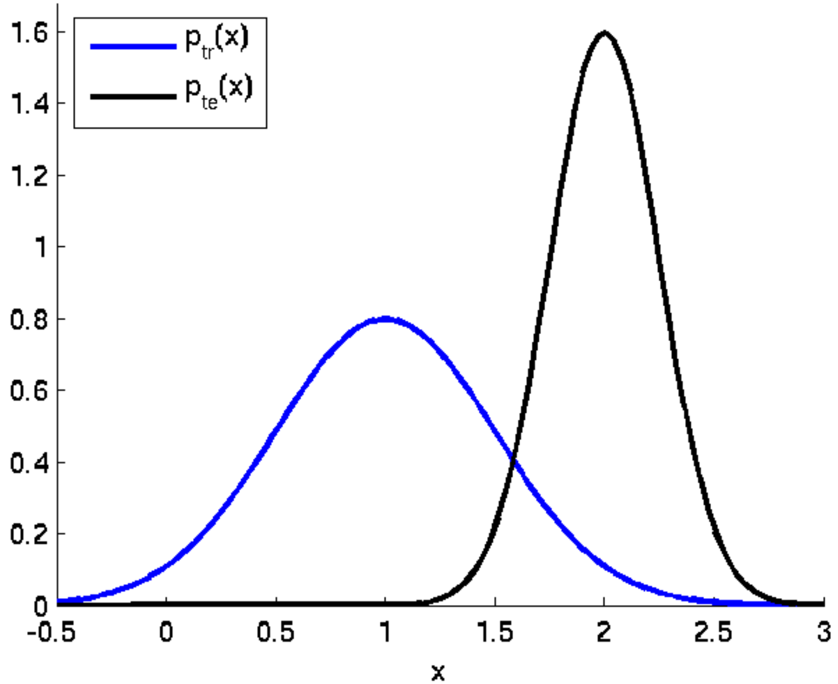
$$\widehat{w}(\boldsymbol{x}) = \sum_{i=1}^{t} \theta_i \phi_i(\boldsymbol{x})$$

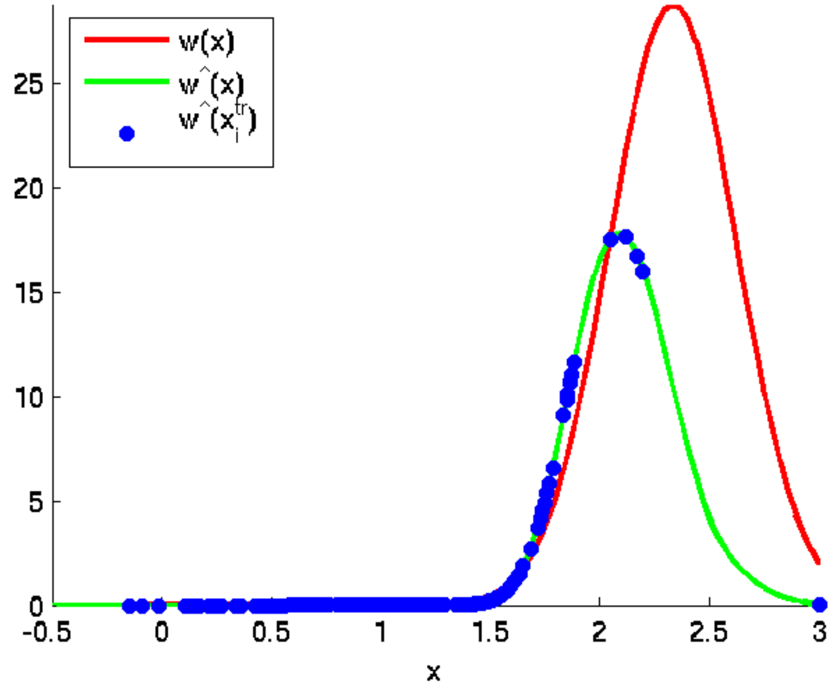
$$\text{subject to} \quad \sum_{i=1}^{n_{train}} \widehat{w}(\boldsymbol{x}_i^{train}) = n_{train}$$

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

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

Examples





$$\widehat{w}(\boldsymbol{x}) = \sum_{i=1}^{n_{test}} \theta_i K(\boldsymbol{x}, \boldsymbol{x}_i^{test})$$

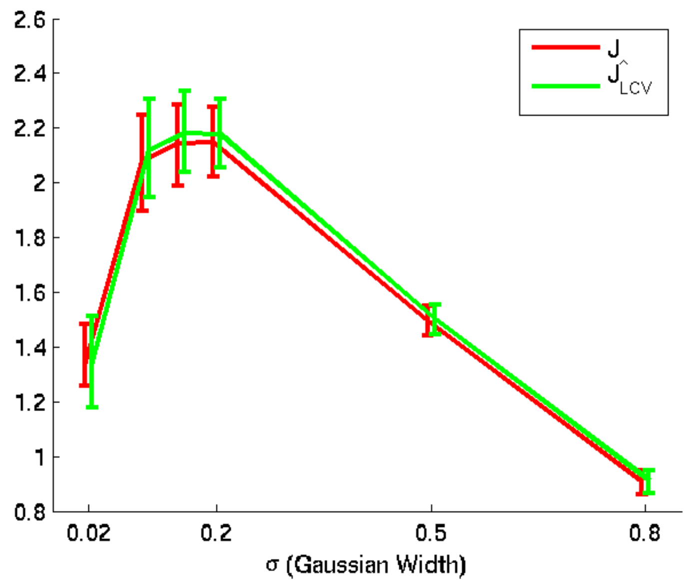
$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\sigma^2}\right)$$

Model Selection of KLIEP

- How to choose tuning parameters (such as Gaussian width)?
- Likelihood cross-validation:
  - Divide test samples $\{\boldsymbol{x}_i^{test}\}_{i=1}^{n_{test}}$ 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).



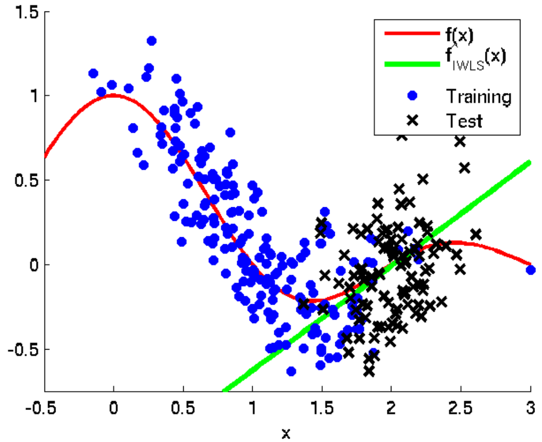
Organization

1. Linear regression under covariate shift
2. Parameter learning
3. Importance estimation
4. Model selection

Model Selection

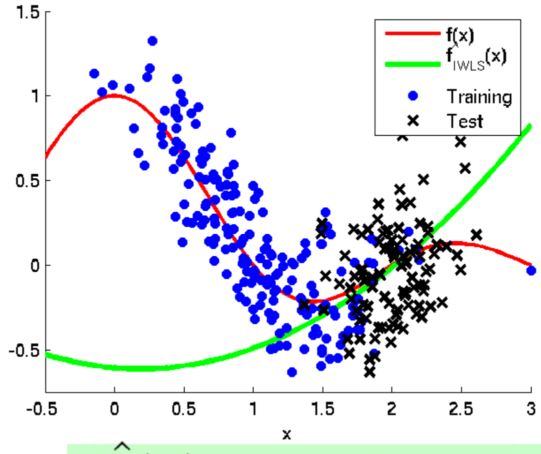
- Choice of models is crucial:

Polynomial of order 1



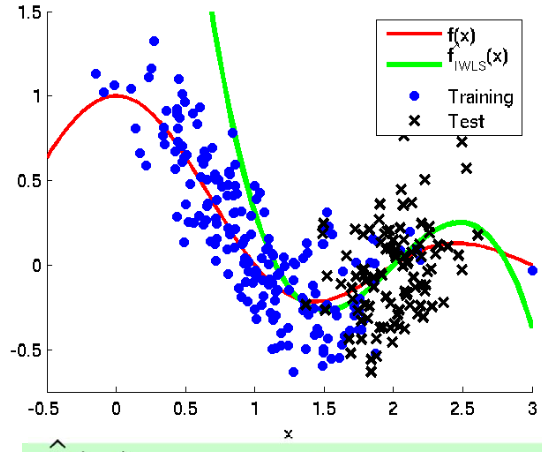
$$\hat{f}(x) = \alpha_1 + \alpha_2 x$$

Polynomial of order 2



$$\hat{f}(x) = \alpha_1 + \alpha_2 x + \alpha_3 x^2$$

Polynomial of order 3



$$\hat{f}(x) = \alpha_1 + \alpha_2 x + \alpha_3 x^2 + \alpha_4 x^3$$

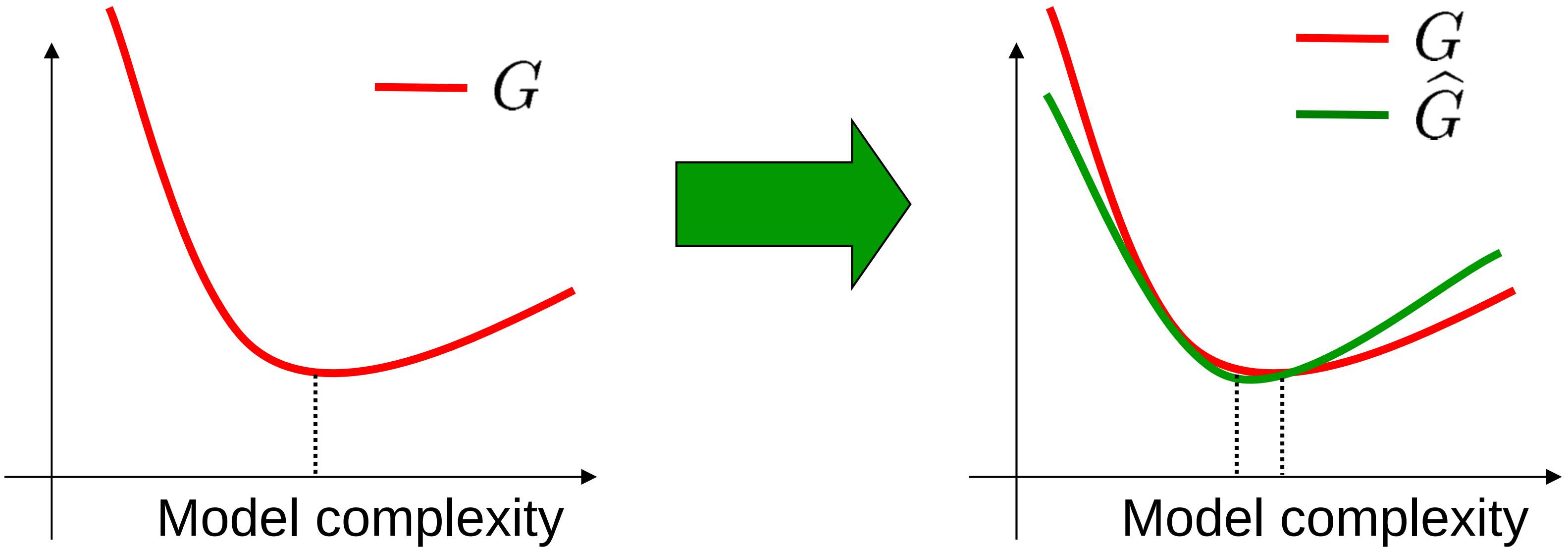
- We want to determine the model so that generalization error is minimized:

$$G = \int \left(\hat{f}(\boldsymbol{x}) - f(\boldsymbol{x}) \right)^2 p_{test}(\boldsymbol{x}) d\boldsymbol{x}$$

Generalization Error Estimation

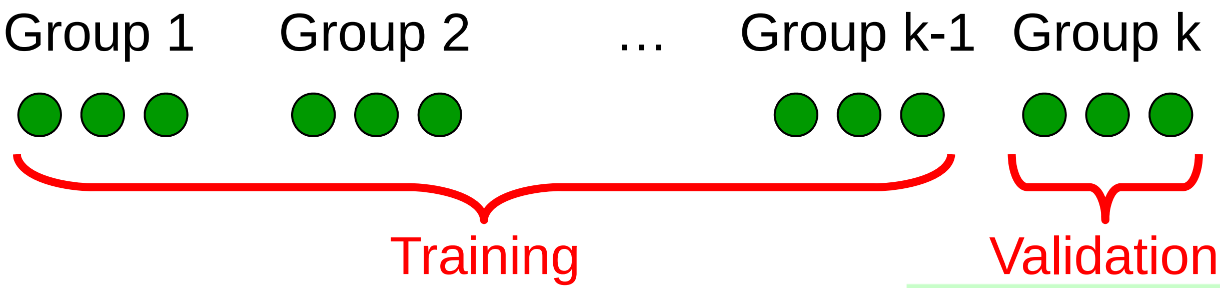
$$G = \int \left(\hat{f}(\boldsymbol{x}) - f(\boldsymbol{x})\right)^2 p_{test}(\boldsymbol{x}) d\boldsymbol{x}$$

- Generalization error is not accessible since the target function $f(\boldsymbol{x})$ is unknown.
- Instead, we use a generalization error estimate.



Cross-Validation

- Divide training samples into k groups.
- Train a learning machine with $k-1$ groups.
- Validate the trained machine using the rest.
- Repeat this for all combinations and output the mean validation error.



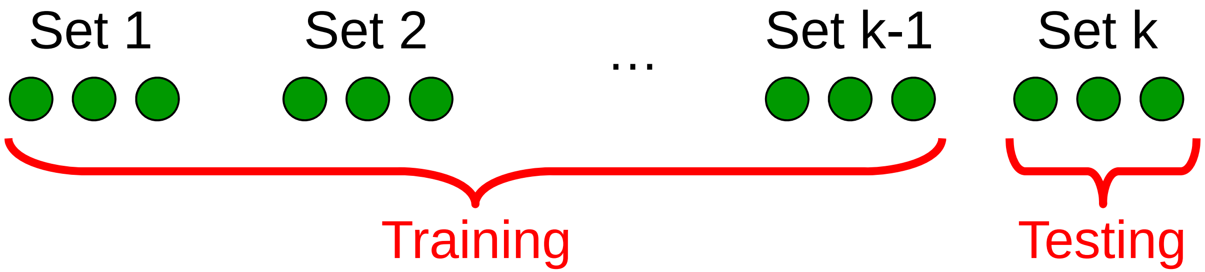
$$\widehat{f}(\boldsymbol{x})$$

$$\left(\widehat{f}(\boldsymbol{x}_t) - y_t\right)^2$$

- CV is almost unbiased without covariate shift.
- But, CV is heavily biased under covariate shift!

Importance-Weighted CV (IWCV)

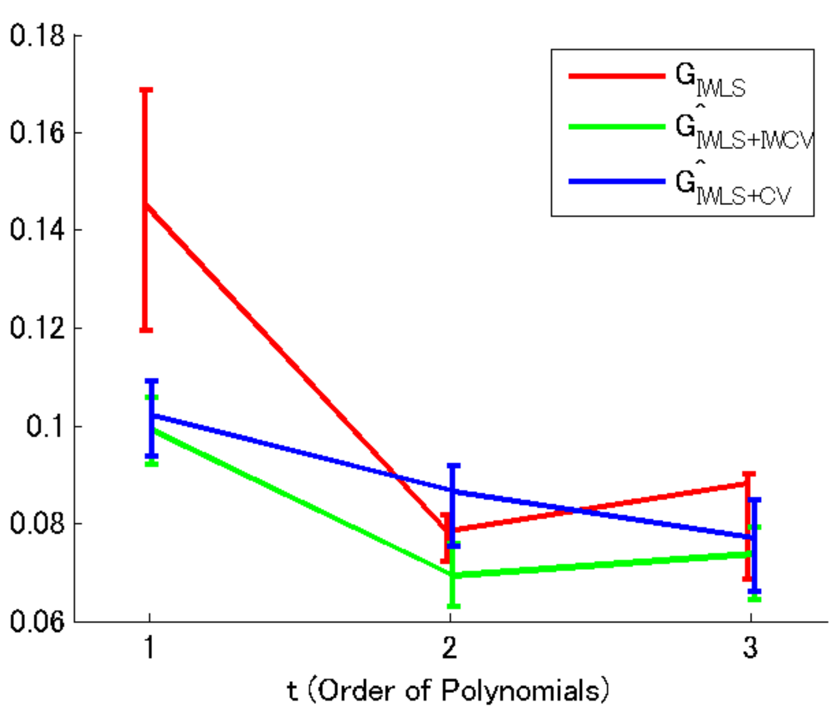
- When testing the classifier in CV process, we also importance-weight the test error.

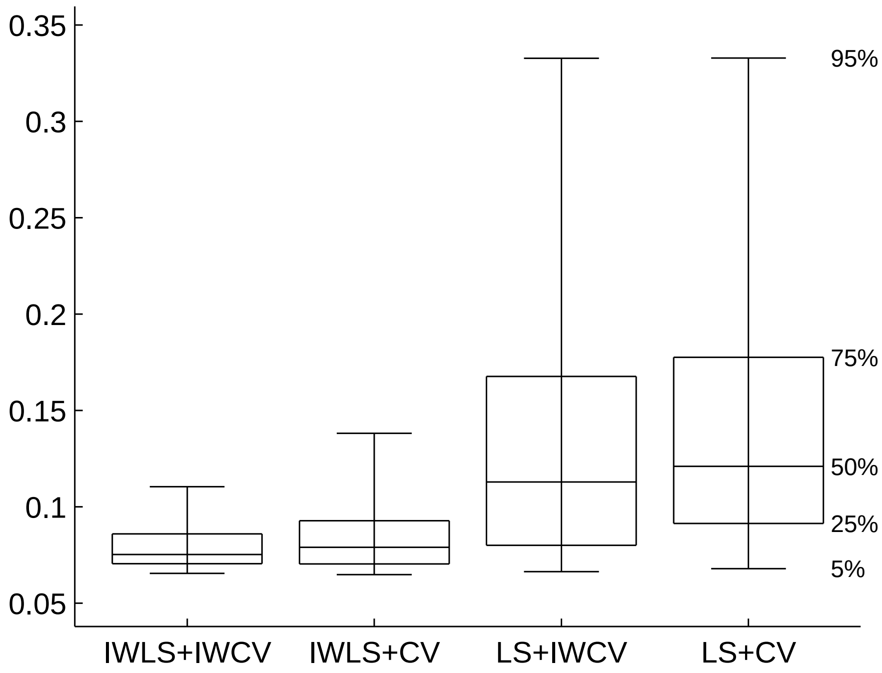


$$\widehat{f}(\boldsymbol{x}) \qquad \frac{p_{test}(\boldsymbol{x}_t)}{p_{train}(\boldsymbol{x}_t)}\left(\widehat{f}(\boldsymbol{x}_t) - y_t\right)^2$$

IWCV gives almost unbiased estimates of generalization error even under covariate shift

Example of IWCV



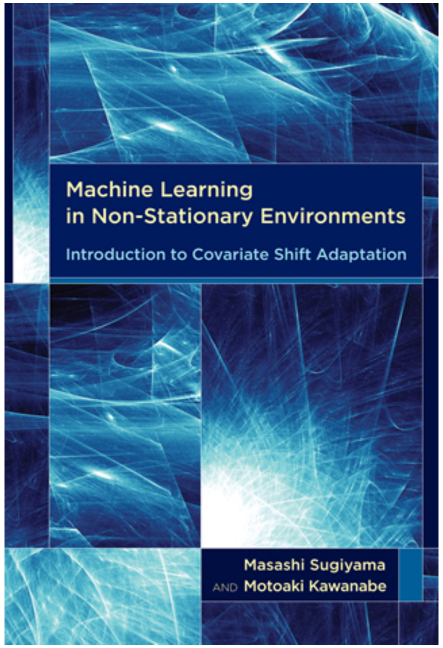


- IWCV gives better estimates of generalization error than CV.
- Model selection by IWCV outperforms CV!

Summary

- Covariate shift: input distribution varies but functional relation remains unchanged
- Importance weighting for adaptation.
  - IW least-squares: consistent
  - KLIEP: direct importance estimation
  - IW cross-validation: unbiased
- Further reading:

  Sugiyama & Kawanabe
  Machine Learning
  in Non-Stationary Environments,
  MIT Press, 2012



Notification of Final Assignment

1. Apply supervised learning techniques to your data set and analyze it.
2. Write your opinion about this course.

- Final report deadline: Aug 3rd (Fri.)
- E-mail submission is also accepted!

sugi@cs.titech.ac.jp