SDM2008 Apr. 24-26, 2008

Active Learning with Model Selection in Linear Regression

Masashi Sugiyama Neil Rubens

Department of Computer Science
Tokyo Institute of Technology, Japan

Abstract

Optimally designing the location of training input points (active learning) and choosing the best model (model selection) are two important components of supervised learning and have been studied extensively. However, these two issues seem to have been investigated separately as two independent problems. If training input points and models are simultaneously optimized, the generalization performance would be further improved. In this paper, we propose a new approach called active learning for solving the problems of active learning and model selection at the same time. We demonstrate by numerical experiments that the proposed method compares favorably with alternative approaches such as iteratively performing active learning and model selection in a sequential manner.

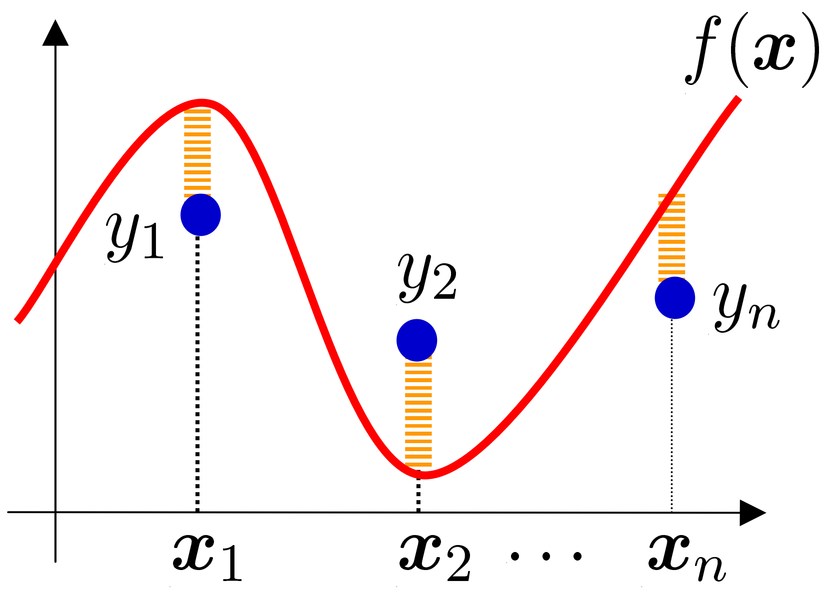
Regression Problem

- $f(\boldsymbol{x})$: Learning target function
- $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$: Training samples

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

$$y_i = f(\boldsymbol{x}_i) + \epsilon_i$$

$$\epsilon_i \overset{i.i.d.}{\sim} \text{mean } 0, \ \text{variance } \sigma^2$$



Goal: Learn $f(\boldsymbol{x})$ from $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$

Linear Regression Model

$$\hat{f}(\boldsymbol{x}) = \sum_{i=1}^{b} \alpha_i \varphi_i(\boldsymbol{x})$$

α_i :Parameter

$\varphi_i(\boldsymbol{x})$:Basis function

- We do NOT assume our model is correct.

($f(\boldsymbol{x})$ is not necessarily included in the model).

Error Metric

- $\boldsymbol{t}$:Test input point (not included in training set)
- Test error: Prediction error at $\boldsymbol{t}$

$$\left(\hat{f}(\boldsymbol{t}) - f(\boldsymbol{t})\right)^2$$

- Generalization error: Expected test error over all test input points

Learn $\boldsymbol{\alpha}$ so that generalization error is minimized

$$\hat{f}(\boldsymbol{x}) = \sum_{i=1}^{b} \alpha_i \varphi_i(\boldsymbol{x})$$

$$\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_b)^\top$$

Common Assumption

- A common assumption in most supervised learning methods proposed so far:

Test input points follow the same distribution as the training input points

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

e.g. standard text books such as Wahba (1990), Bishop (1995,2006), Vapnik (1998), Hastie *et al.* (2001), Schölkopf & Smola (2002)

Generalization error

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

Covariate Shift

Shimodaira (JSPI 2000)

- Test and training input points follow different distributions.

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

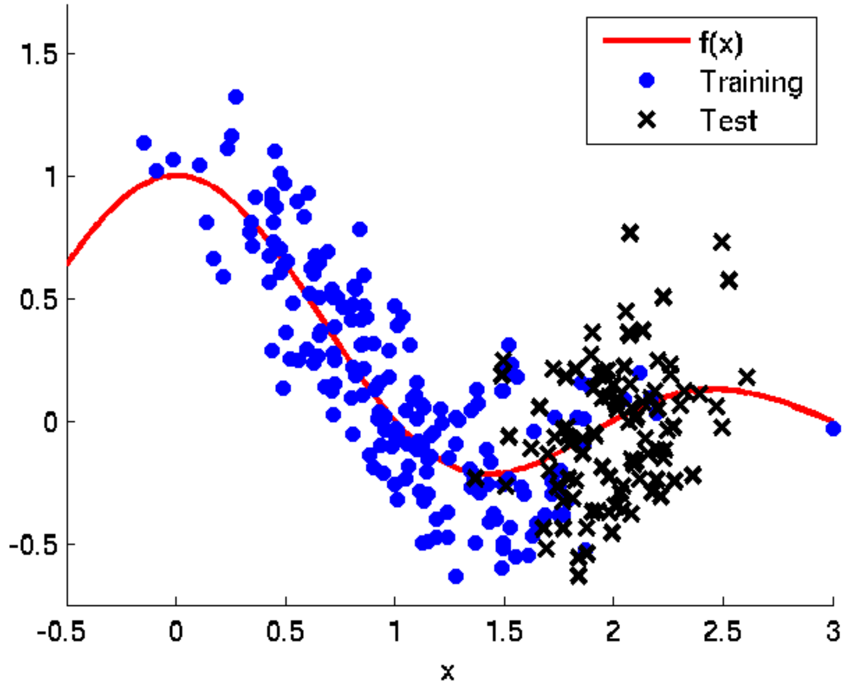
$$p_{train}(\boldsymbol{x}) \neq p_{test}(\boldsymbol{x})$$

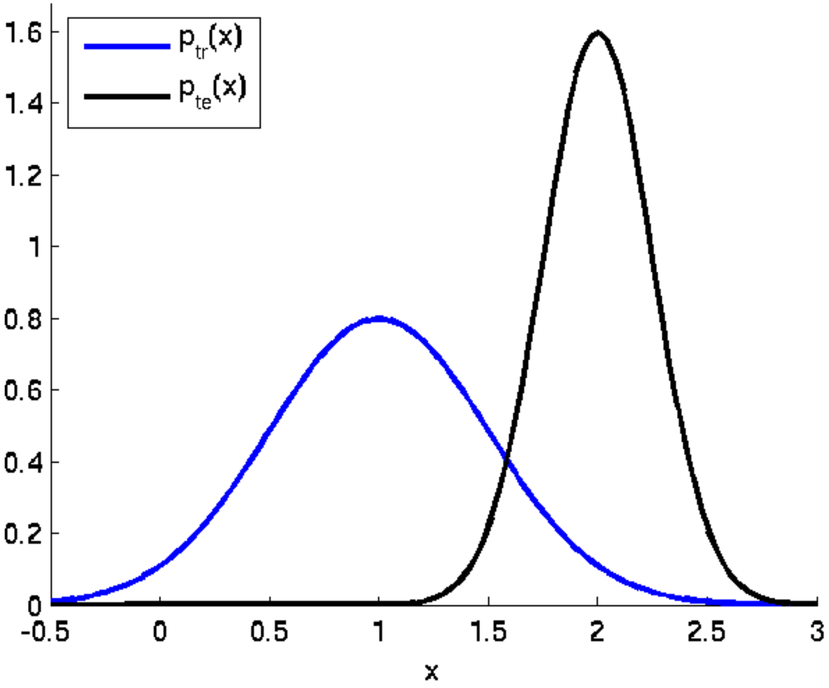
Generalization error

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

Example of Covariate Shift

(Weak) extrapolation:
Predict output values outside training region

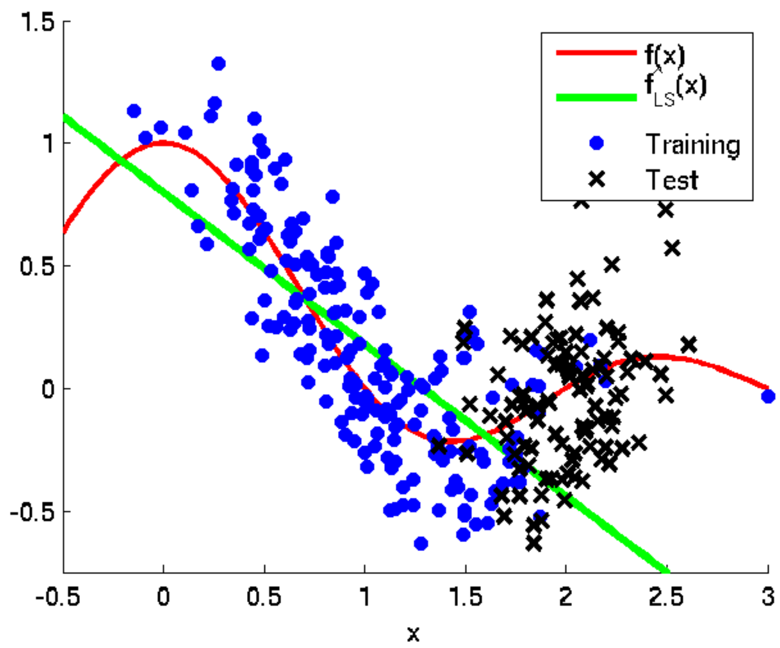




Parameter Learning: Ordinary Least-Squares under Covariate Shift

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

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



OLS is not consistent

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 from training input points $\{\boldsymbol{x}_i\}_{i=1}^{n}$.

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

Importance-Weighted Average

- Importance: Ratio of test and training input densities

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

- Importance-weighted average:

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

$$= \int A(\boldsymbol{x})p_{test}(\boldsymbol{x})d\boldsymbol{x}$$

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

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

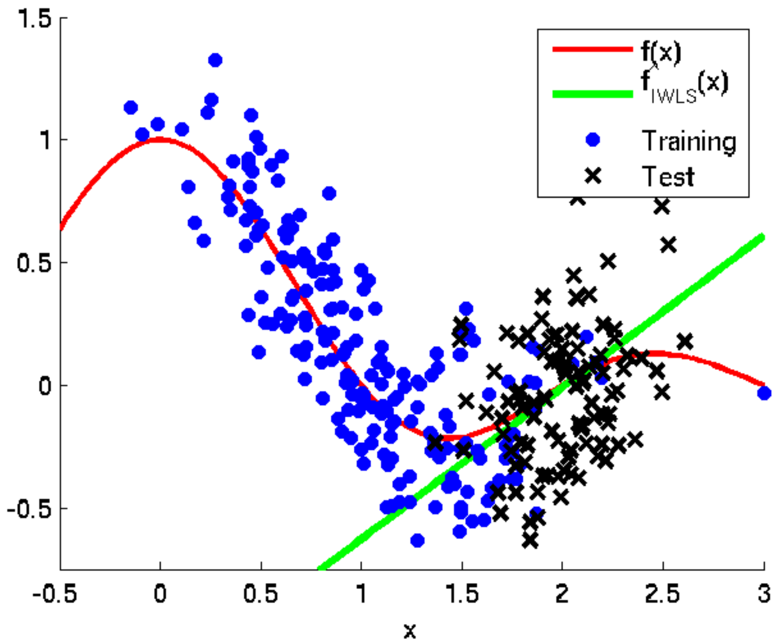
(cf. importance sampling)

Importance-Weighted LS for Covariate Shift

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

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

IWLS is consistent

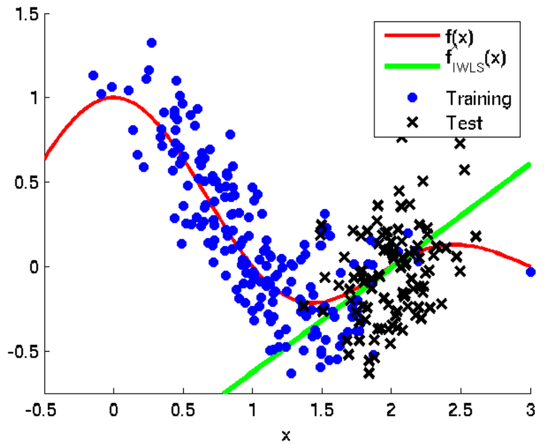


- Importance can be estimated efficiently, e.g., by KLIEP. Sugiyama *et al*. (2007)

Model Selection

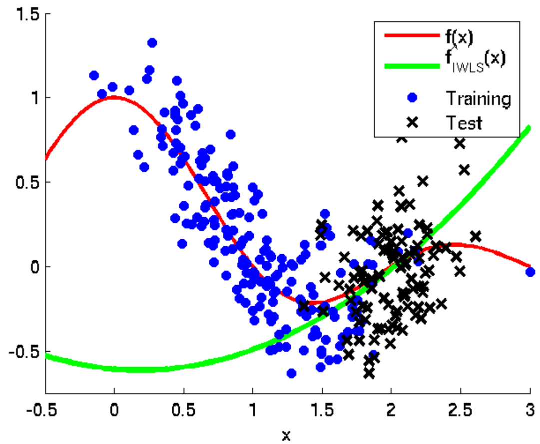
- Choice of models is crucial:

Polynomial of order 1



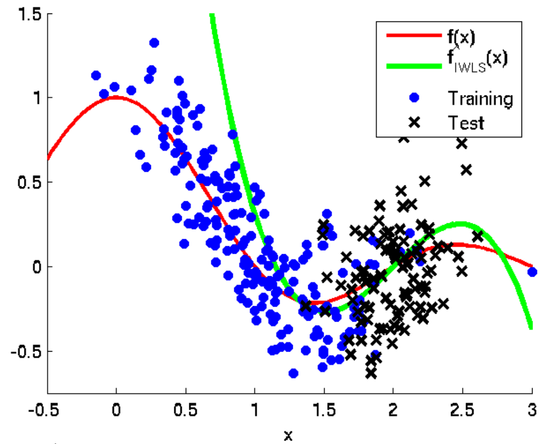
$$\hat{f}(x) = \alpha_1 + \alpha_2 x$$

Polynomial of order 2



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

Polynomial of order 3



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

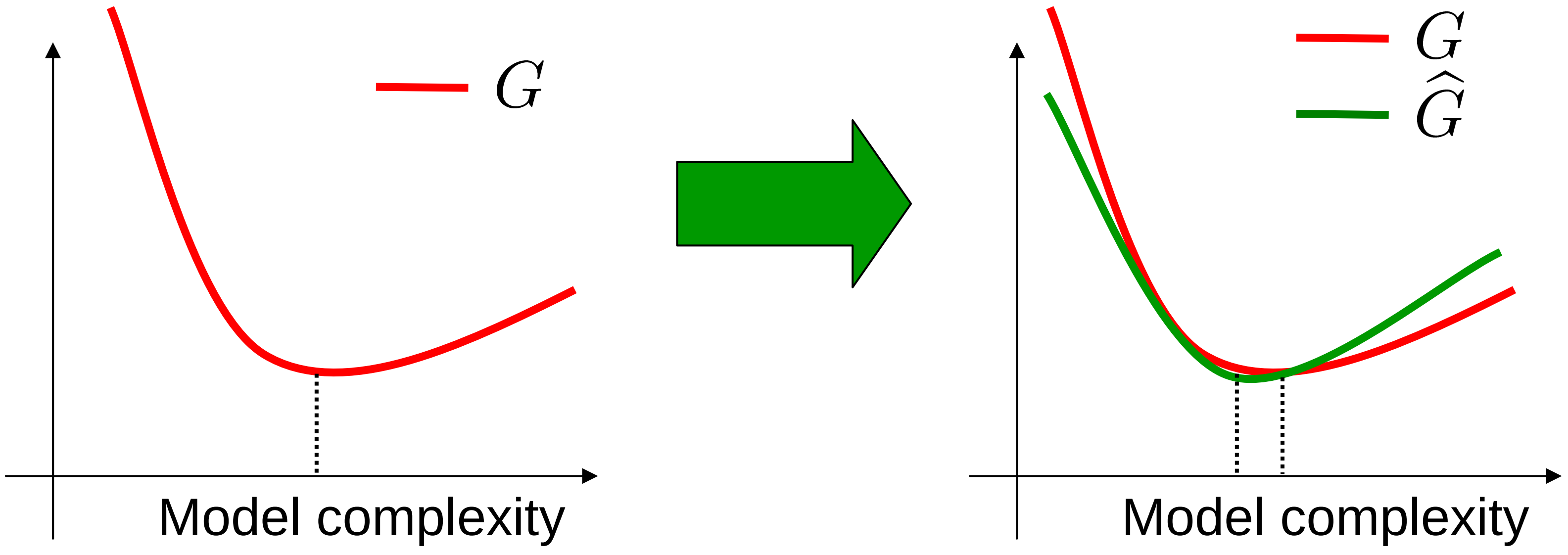
- We want to determine the model so that generalization error is minimized:

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

Generalization Error Estimation

$$G = \|\widehat{f} - f\|^2$$

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



Assumption

- We use linear parameter learning:

$$\widehat{\boldsymbol{\alpha}} = \boldsymbol{L}\boldsymbol{y}$$

$\boldsymbol{L}$: matrix independent of training output noise

$$\hat{f}(\boldsymbol{x}) = \sum_{i=1}^{b} \alpha_i \varphi_i(\boldsymbol{x})$$

$$\boldsymbol{y} = (y_1, y_2, \ldots, y_n)^\top$$

E.g., importance-weighted least-squares

$$\boldsymbol{L} = (\boldsymbol{X}^\top \boldsymbol{D} \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{D}$$

$$\boldsymbol{X}_{i,j} = \varphi_j(\boldsymbol{x}_i)$$

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

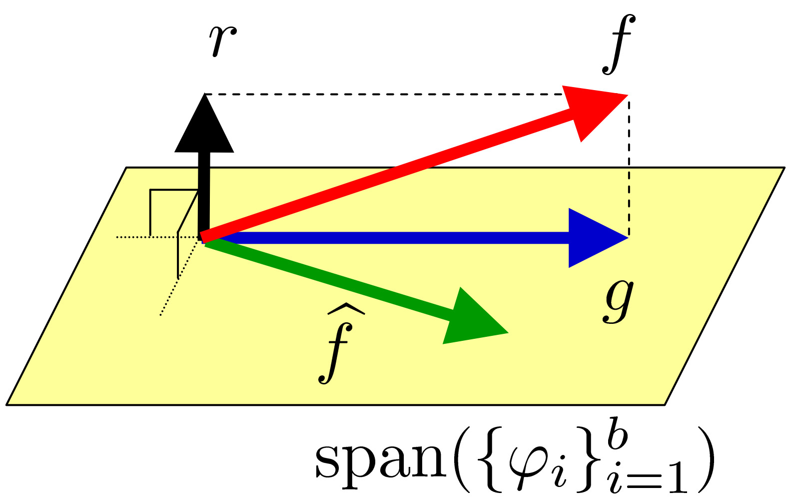
$$\boldsymbol{D}_{i,j} = \mathrm{diag}\left(\frac{p_{test}(\boldsymbol{x}_1)}{p_{train}(\boldsymbol{x}_1)}, \ldots, \frac{p_{test}(\boldsymbol{x}_n)}{p_{train}(\boldsymbol{x}_n)} \right)$$

Estimating Generalization Error

$$G = \|\widehat{f} - f\|^2$$

$$= \|\widehat{f}\|^2 \quad + \|f\|^2 \quad - 2\langle \widehat{f}, f\rangle$$

Accessible — Constant (ignored) — Estimated



$$f = g + r$$

$$g(\boldsymbol{x}) = \sum_{i=1}^{b} \alpha_i^* \varphi_i(\boldsymbol{x})$$

$$\langle r, \varphi_i \rangle = 0$$

$$\langle \widehat{f}, f\rangle = \langle \widehat{f}, g\rangle = \widehat{\boldsymbol{\alpha}}^\top \boldsymbol{U} \boldsymbol{\alpha}^*$$

$$\widehat{f}(\boldsymbol{x}) = \sum_{i=1}^{b} \widehat{\alpha}_i \varphi_i(\boldsymbol{x}) \qquad \boldsymbol{U}_{i,j} = \langle \varphi_i, \varphi_j \rangle$$

Subspace Information Criterion

Sugiyama & Ogawa (Neural Comp. 2001)
Sugiyama & Müller (JMLR 2002)

$$\widehat{\boldsymbol{\alpha}}^\top \boldsymbol{U}\boldsymbol{\alpha}^*$$

- Idea: Replace $\boldsymbol{\alpha}^*$ by a linear unbiased estimator $\widetilde{\boldsymbol{\alpha}}$

$$\widetilde{\boldsymbol{\alpha}} = \widetilde{\boldsymbol{L}}\boldsymbol{y}$$

- Since $\widetilde{\boldsymbol{\alpha}}$ and $\widehat{\boldsymbol{\alpha}}$ are estimated from the same sample $\boldsymbol{y}$, it causes a bias:

$$\widehat{\boldsymbol{\alpha}} = \boldsymbol{L}\boldsymbol{y}$$

$$\mathbb{E}_{\boldsymbol{\epsilon}}[\widehat{\boldsymbol{\alpha}}^\top \boldsymbol{U}\boldsymbol{\alpha}^* - \widehat{\boldsymbol{\alpha}}^\top \boldsymbol{U}\widetilde{\boldsymbol{\alpha}}] = \sigma^2 \mathrm{tr}(\boldsymbol{U}\boldsymbol{L}\widetilde{\boldsymbol{L}}^\top)$$

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

- Bias correction results in a generalization error estimator (named SIC).

Importance-Weighted SIC

Sugiyama & Müller (Statistics & Decisions 2005)

$$\mathrm{IWSIC}[\boldsymbol{L}] = \boldsymbol{y}^\top \boldsymbol{L}^\top \boldsymbol{U} \boldsymbol{L} \boldsymbol{y} - 2\boldsymbol{y}^\top \widetilde{\boldsymbol{L}}^\top \boldsymbol{U} \boldsymbol{L} \boldsymbol{y} + 2\widetilde{\sigma}^2 \mathrm{tr}(\boldsymbol{U} \boldsymbol{L} \widetilde{\boldsymbol{L}}^\top)$$

$$\boldsymbol{U}_{i,j} = \langle \varphi_i, \varphi_j \rangle \qquad \widetilde{\boldsymbol{L}} = (\widetilde{\boldsymbol{X}}^\top \boldsymbol{D} \widetilde{\boldsymbol{X}})^{-1} \widetilde{\boldsymbol{X}}^\top \boldsymbol{D} \qquad \boldsymbol{X}_{i,j} = \varphi_j(\boldsymbol{x}_i)$$

$\widehat{\boldsymbol{\alpha}} = \boldsymbol{L}\boldsymbol{y}$ $\qquad$ $\widetilde{\boldsymbol{X}}$: $\boldsymbol{X}$ for largest model $\qquad$ $\widetilde{\sigma}^2 = \|\boldsymbol{G}\boldsymbol{y}\|^2 / \mathrm{tr}(\boldsymbol{G})$

$$\boldsymbol{G} = \boldsymbol{I} - \widetilde{\boldsymbol{X}}(\widetilde{\boldsymbol{X}}^\top \widetilde{\boldsymbol{X}})^{-1} \widetilde{\boldsymbol{X}}^\top \qquad \boldsymbol{D}_{i,j} = \mathrm{diag}\left(\frac{p_{test}(\boldsymbol{x}_1)}{p_{train}(\boldsymbol{x}_1)}, \ldots, \frac{p_{test}(\boldsymbol{x}_n)}{p_{train}(\boldsymbol{x}_n)} \right)$$

- IWSIC is asymptotically unbiased (up to relevant terms):

$$\mathbb{E}_{\boldsymbol{\epsilon}}(\mathrm{IWSIC} - G - C) = \mathcal{O}_p(\delta n^{-1/2})$$

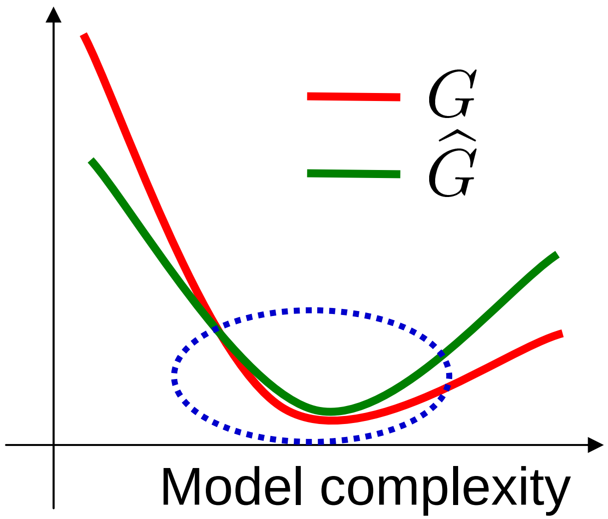
δ : model error $(= \|r\|)$
$\mathbb{E}_{\boldsymbol{\epsilon}}$: expectation over noise

Accuracy and Model Error

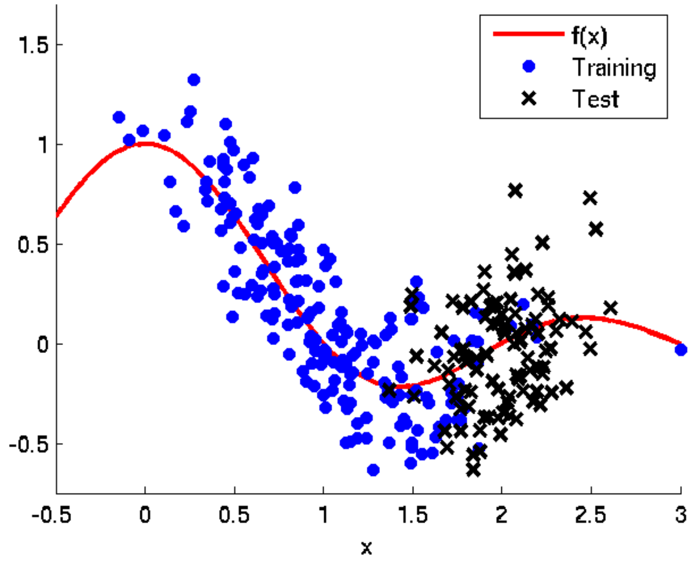
- Model selection: choose the most promising model from candidates
- Easy to distinguish too simple models from good ones by a rough gen. error estimator.
- Therefore, our real interest is to find an excellent model from good models.
- IWSIC is useful in this respect since it is more accurate for better models.

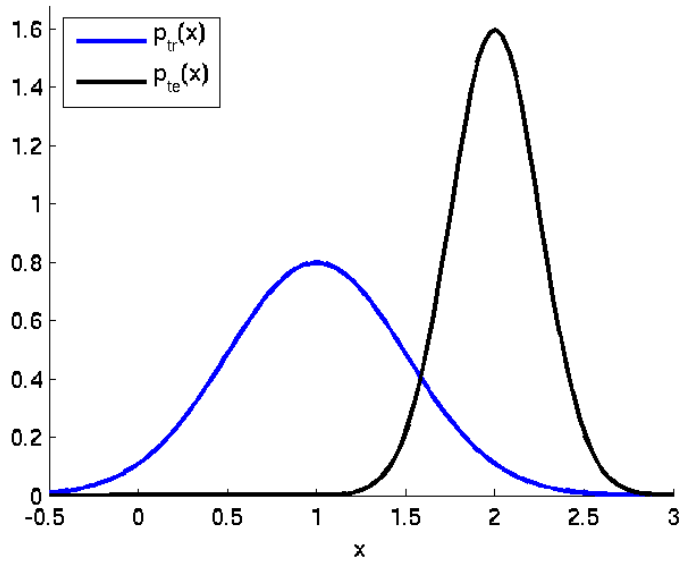
$$\mathbb{E}_{\epsilon}(\mathrm{IWSIC} - G - C) = \mathcal{O}_p(\delta n^{-1/2})$$

δ : model error



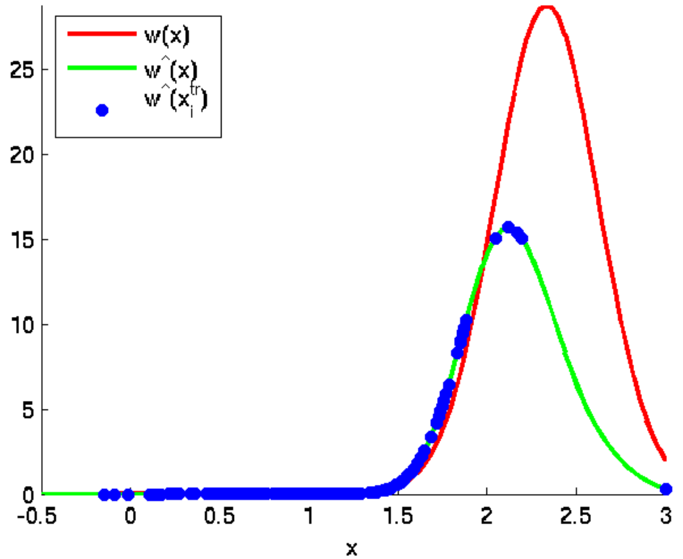
Numerical Examples





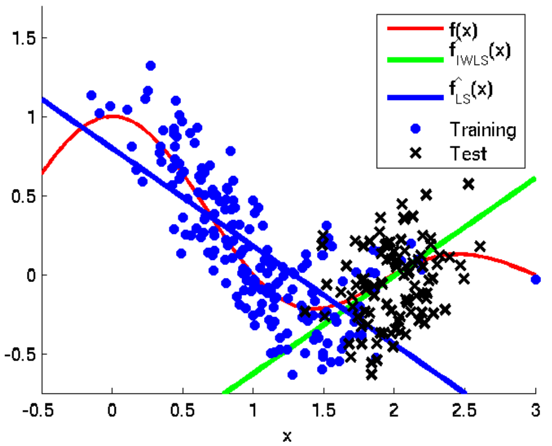
- Importance is estimated by KLIEP with automatic model selection (no tuning parameters remains).

Sugiyama *et al*. (2007)



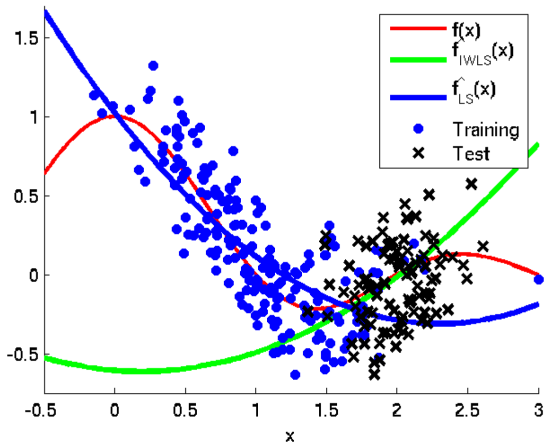
Numerical Examples (cont.)

Polynomial of order 1



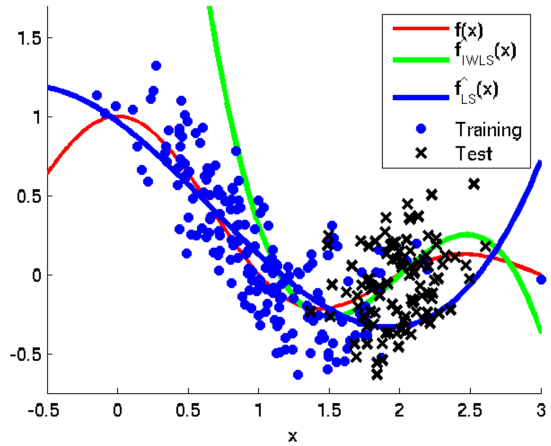
$$\hat{f}(x) = \alpha_1 + \alpha_2 x$$

Polynomial of order 2

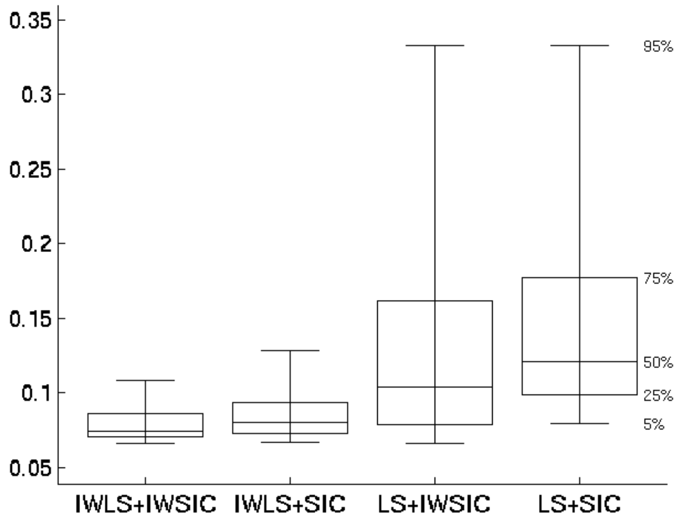


$$\widehat{f}(x) = \alpha_1 + \alpha_2 x + \alpha_3 x^2$$

Polynomial of order 3



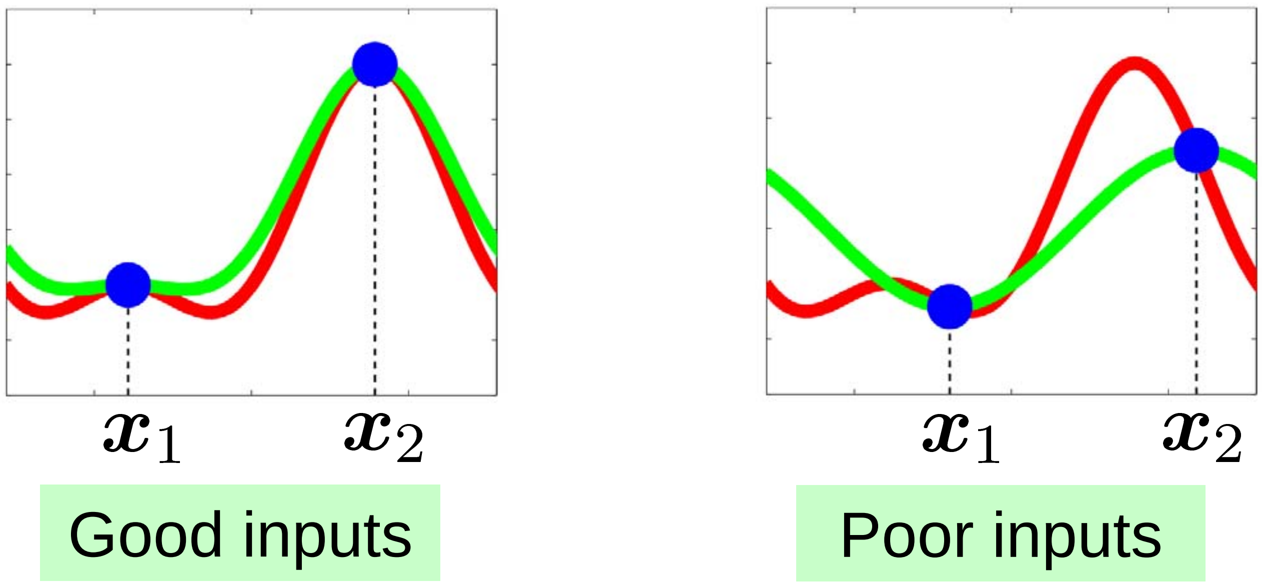
$$\widehat{f}(x) = \alpha_1 + \alpha_2 x + \alpha_3 x^2 + \alpha_4 x^3$$



- IWLS+IWSIC works better than others.

Active Learning

- Choice of training input location is crucial:



- We want to determine training input location so that generalization error is minimized:

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

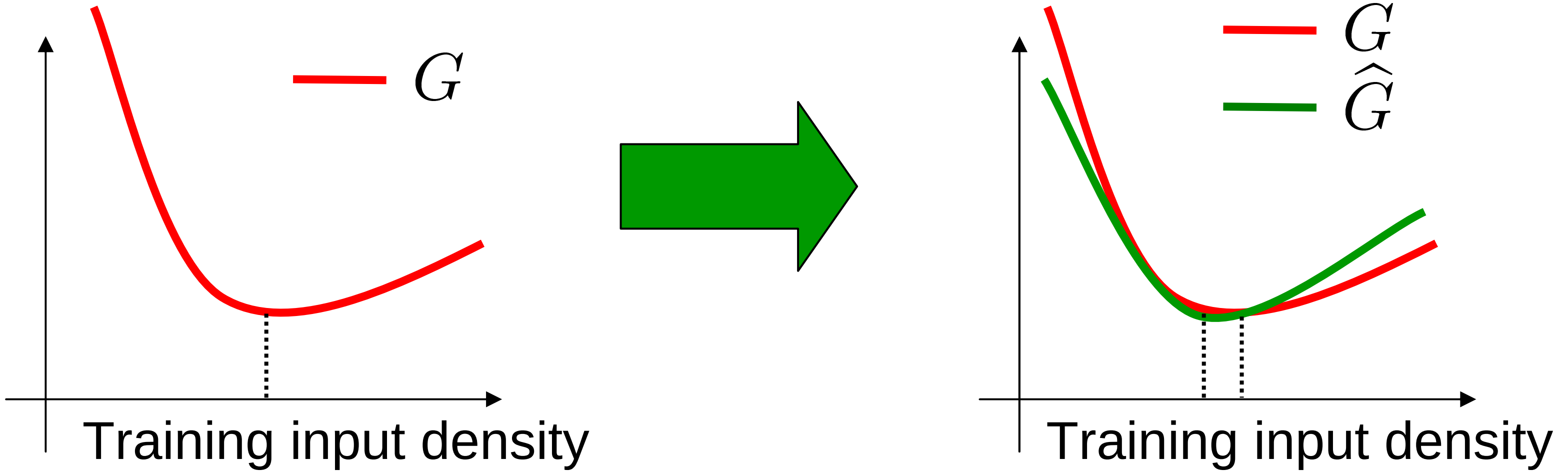
Batch Active Learning

- Batch active learning: optimize location of all training inputs $\{\boldsymbol{x}_i\}_{i=1}^{n}$ in the beginning.
- However, this is computationally hard since n points are simultaneously optimized
- Incremental approach: optimize inputs one by one, which is popular but greedy optimal.
- We optimize training input density $p_{train}(\boldsymbol{x})$ and draw training inputs from it.

Generalization Error Estimation

$$G = \|\widehat{f} - f\|^2$$

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



- Similar to model selection, but horizontal axis is different (model or training input density).

Remarks

- We need to estimate generalization error before observing training outputs $\{y_i\}_{i=1}^n$.
- Thus generalization error estimation in active learning would be harder than model selection.

- We design training input density by ourselves.
- Thus covariate shift always occurs in active learning.

Assumption

- We use importance-weighted least-squares:

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

$$\hat{f}(\boldsymbol{x}) = \sum_{i=1}^{b} \alpha_i \varphi_i(\boldsymbol{x})$$

$$\widehat{\boldsymbol{\alpha}} = \boldsymbol{L}\boldsymbol{y}$$

$$\boldsymbol{L} = (\boldsymbol{X}^\top \boldsymbol{D} \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{D}$$

$$\boldsymbol{D}_{i,j} = \mathrm{diag}\left(\frac{p_{test}(\boldsymbol{x}_1)}{p_{train}(\boldsymbol{x}_1)}, \ldots, \frac{p_{test}(\boldsymbol{x}_n)}{p_{train}(\boldsymbol{x}_n)} \right)$$

$$\boldsymbol{X}_{i,j} = \varphi_j(\boldsymbol{x}_i)$$

$$\boldsymbol{y} = (y_1, y_2, \ldots, y_n)^\top$$

Bias/Variance Decomposition

$$\mathbb{E}_{\boldsymbol{\epsilon}} G = \mathbb{E}_{\boldsymbol{\epsilon}} \|\widehat{f} - f\|^2 = \delta^2 + B + V$$

- Model error:

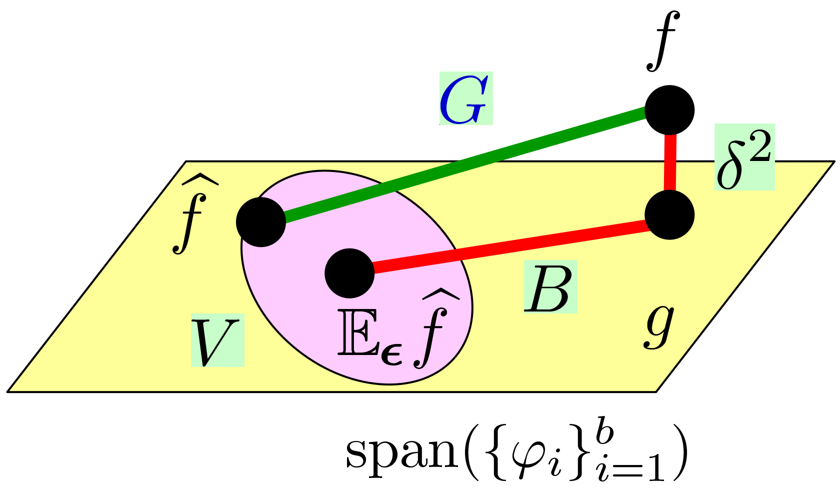
$$\delta = \|f - g\|$$

- Bias:

$$B = \|\mathbb{E}_{\boldsymbol{\epsilon}} \widehat{f} - g\|^2$$

- Variance:

$$V = \mathbb{E}_{\boldsymbol{\epsilon}} \|\mathbb{E}_{\boldsymbol{\epsilon}} \widehat{f} - \widehat{f}\|^2$$



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

Bias/Variance of IWLS for Approximately Correct Models

$$\mathbb{E}_{\boldsymbol{\epsilon}} G = \mathbb{E}_{\boldsymbol{\epsilon}} \|\widehat{f} - f\|^2 = \delta^2 + B + V$$

- We want to estimate $\mathbb{E}_{\boldsymbol{\epsilon}} G$ without using $\{y_i\}_{i=1}^n$.
  - Model error: constant and can be ignored

    $$\delta = \|f - g\|$$

  - Variance: computable up to scaling factor σ^2:

    $$V = \mathbb{E}_{\boldsymbol{\epsilon}} \|\mathbb{E}_{\boldsymbol{\epsilon}} \widehat{f} - \widehat{f}\|^2 = \sigma^2 \mathrm{tr}(\boldsymbol{U}\boldsymbol{L}\boldsymbol{L}^\top) = \mathcal{O}_p(n^{-1})$$

  - Bias: hard to estimate, but can be safely ignored if $\delta = o(1)$:

    $$B = \|\mathbb{E}_{\boldsymbol{\epsilon}} \widehat{f} - g\|^2 = \mathcal{O}_p(\delta^2 n^{-1})$$

ALICE

Sugiyama (JMLR 2006)

Active Learning using Importance-weighted least-squares based on Conditional Expectation of generalization error

$$\mathrm{ALICE}[p_{train}] = \mathrm{tr}(\boldsymbol{U}\boldsymbol{L}\boldsymbol{L}^\top)$$

$$\boldsymbol{U}_{i,j} = \langle \varphi_i, \varphi_j \rangle$$

$$\boldsymbol{L} = (\boldsymbol{X}^\top \boldsymbol{D} \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{D}$$

$$\boldsymbol{X}_{i,j} = \varphi_j(\boldsymbol{x}_i)$$

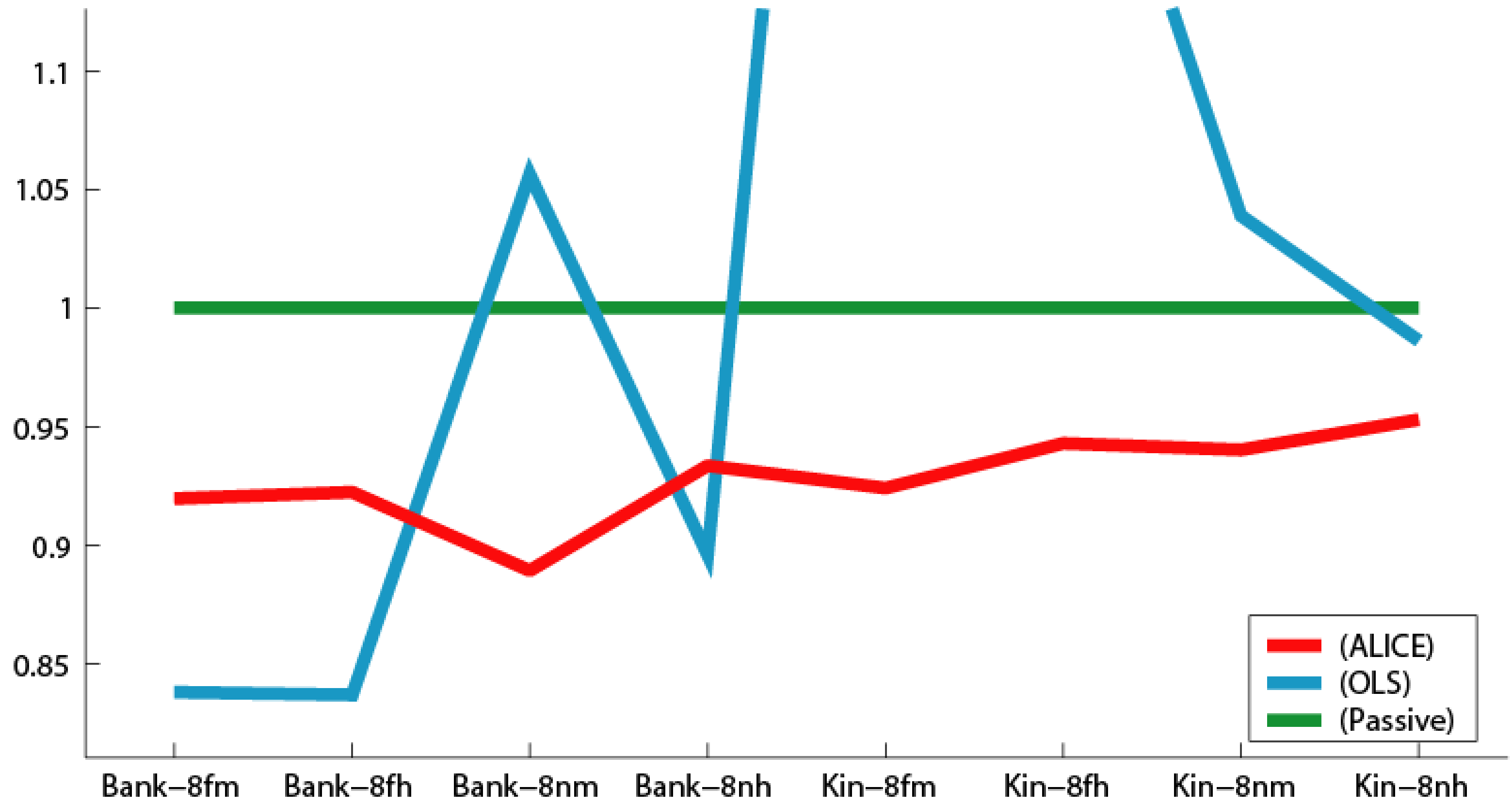
$$\boldsymbol{D}_{i,j} = \frac{p_{test}(\boldsymbol{x}_i)}{p_{train}(\boldsymbol{x}_i)} \delta_{i,j}$$

- ALICE is consistent (up to relevant terms) for approximately correct models with $\delta = o(1)$:

$$\sigma^2 \mathrm{ALICE} - G + \delta^2 = \mathcal{O}_p(n^{-1})$$

Simulation Results

Mean over 100 trials (normalized by passive)



- OLS-based is sometimes good, but unstable.

Cohn *et al*. (JAIR 1996), Fukumizu (IEEE-TNN 2000)

- ALICE works well in a stable manner.

Active Learning with Model Selection (ALMS)

- MS: optimize model $\mathcal{M}$

$$\min_{\mathcal{M}} G(\mathcal{M})$$

- AL: optimize training input density $p_{train}(\boldsymbol{x})$

$$\min_{p_{train}} G(p_{train})$$

- ALMS: optimize both $\mathcal{M}$ and $p_{train}(\boldsymbol{x})$

$$\min_{\mathcal{M}, p_{train}} G(\mathcal{M}, p_{train})$$

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

Optimal Solution

Sugiyama & Ogawa (IEICE Trans. 2003)

- Suppose there exist the common optimal training input density for all model candidates.

$$p^*_{train} = \mathop{\mathrm{argmin}}_{p_{train}} G(\mathcal{M}, p_{train}) \text{ for all } \mathcal{M}$$

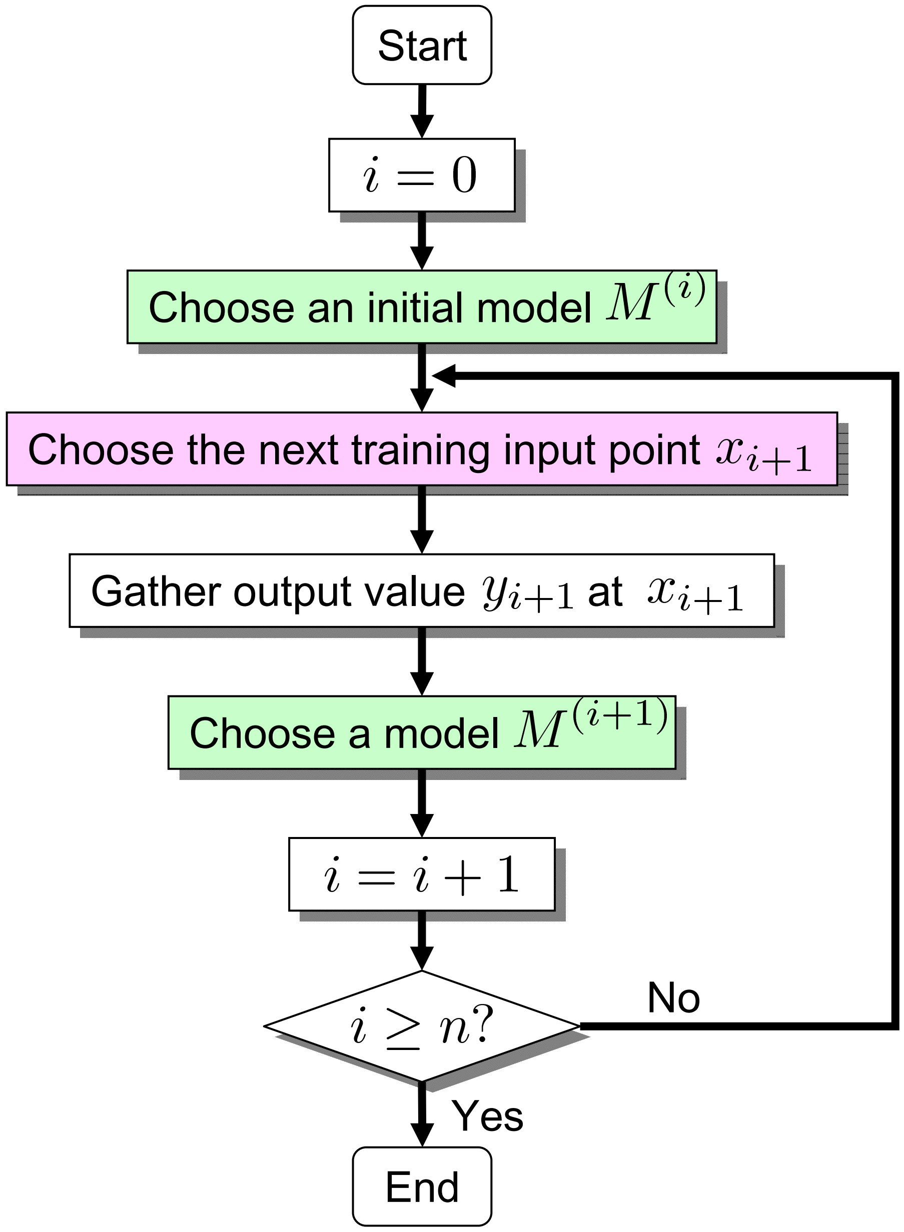
- Then using p^*_{train} and choose a model by an existing MS method is optimal.
- This scenario can be realized for correct trigonometric polynomial models.
- However, not possible for general models.

AL/MS Dilemma

- Can we simply employ existing MS and AL methods for simultaneously optimizing $\mathcal{M}$ and $p_{train}(\boldsymbol{x})$?
- AL/MS dilemma:
  - MS methods require to fix $p_{train}(\boldsymbol{x})$.
  - AL methods require to fix $\mathcal{M}$.
- Batch ALMS can not be solved by simply combining existing MS and AL methods.

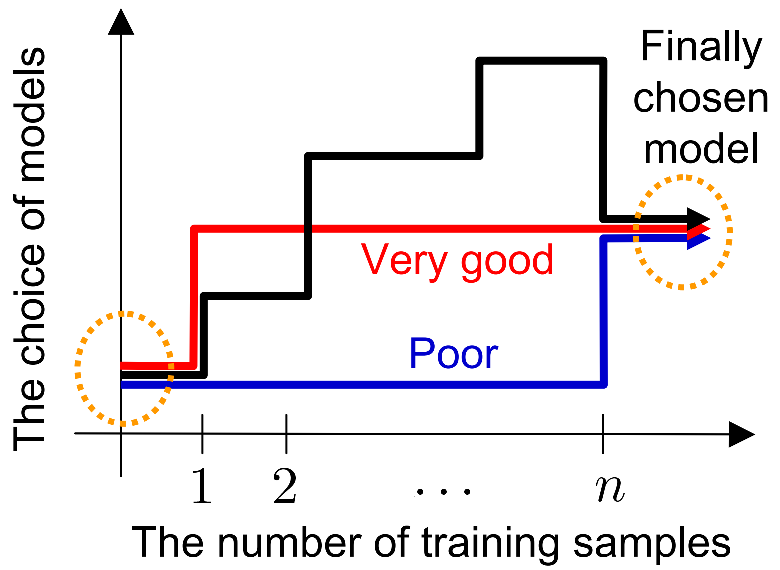
Sequential Approach

- Iteratively choose
  - a training input point (or a small portion)
  - a model
- This is commonly used in practice.



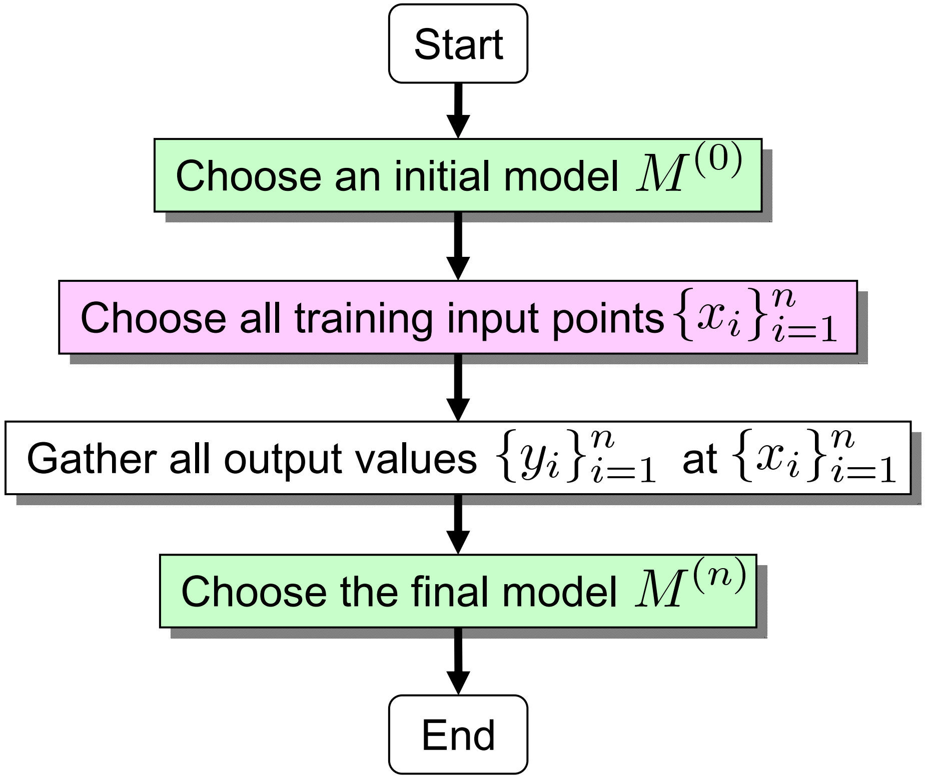
Model Drift

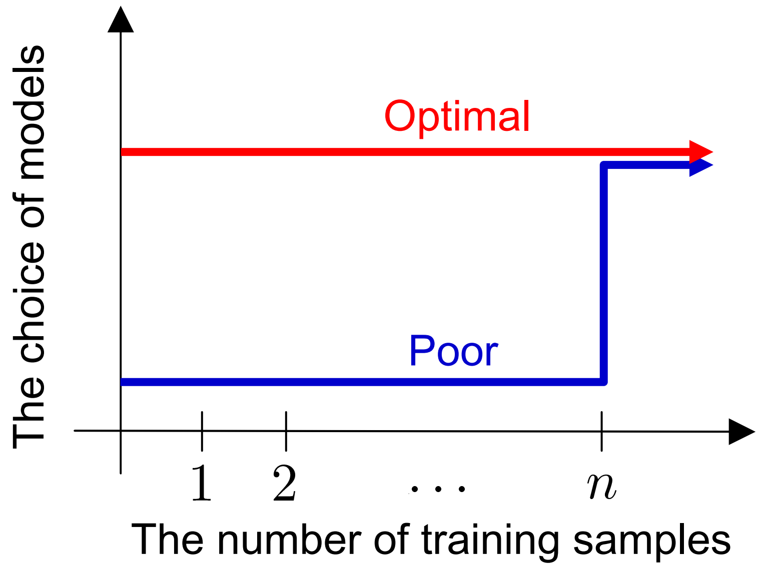
- However, sequential approach is not effective.
  - Target model varies through learning process.
  - Good training input density depends heavily on the target model.
  - Training input points determined in early stages could be poor for finally chosen model.
  - AL overfits to target models.



Batch Approach

- Perform batch AL for an initially chosen model.
- This does not suffer from model drift.





Difficulty in Initial Model Choice

- We need to choose an initial model before observing training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$.
  - IWSIC can not be computed without $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$.
  - ALICE can be computed without $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, but the simplest model is always chosen since it is a variance estimator.
- In practice, we may have to determine the initial model randomly.
- Therefore, batch approach is not reliable.

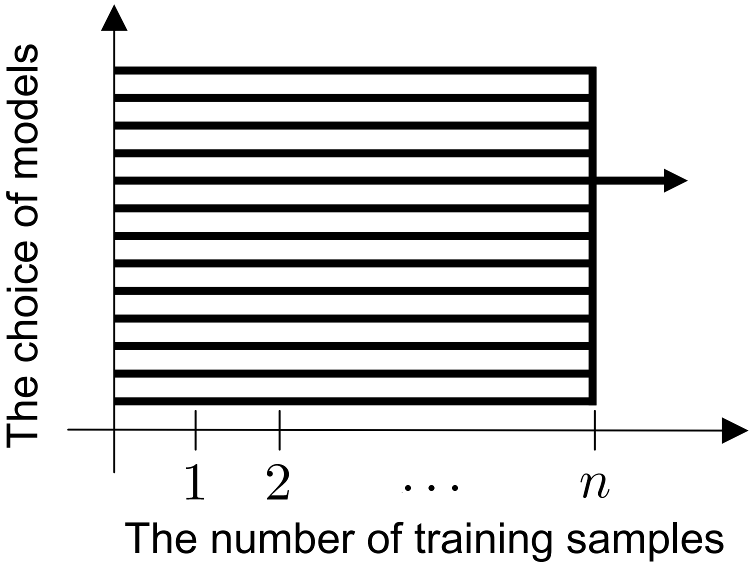
Ensemble Active Learning

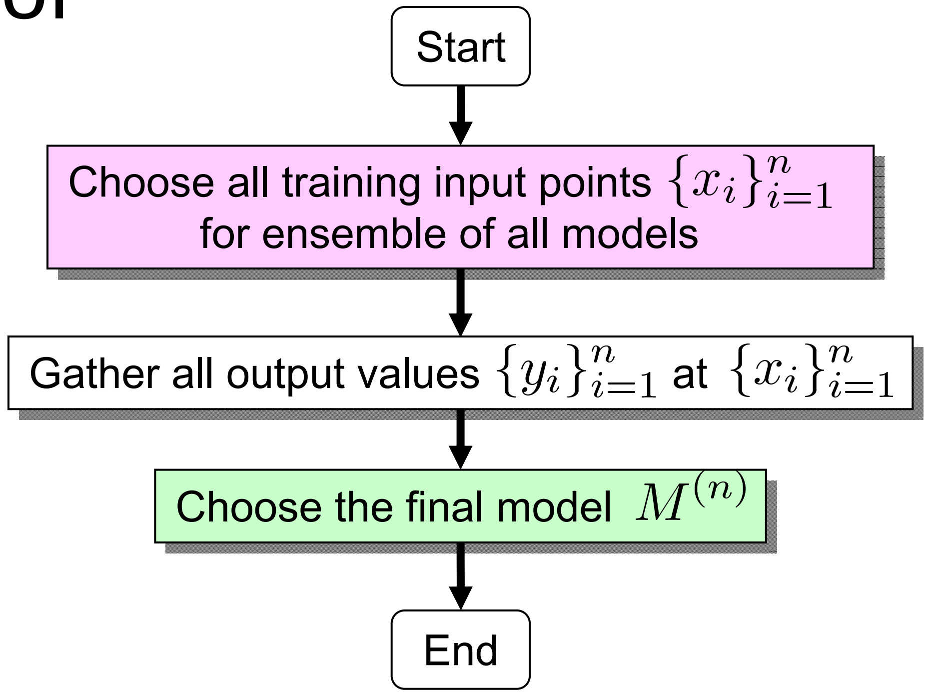
Sugiyama & Rubens (2007)

- Choose training input density for all models:

$$\min_{p_{train}} \left[\sum_{\mathcal{M}} G(\mathcal{M}, p_{train}) \right]$$

- This reduces the risk of overfitting to a single (inferior) model.





Simulation Results

Dataset	Passive	Sequential	Batch	Ensemble
Bank-8fm	1.00(1.22)	0.59(0.85)	0.46(0.25)	0.45(0.28)
Bank-8fh	1.00(0.42)	0.53(0.22)	0.46(0.18)	0.44(0.11)
Bank-8nm	1.00(0.76)	0.63(0.19)	0.58(0.21)	0.56(0.10)
Bank-8nh	1.00(0.28)	0.61(0.19)	0.53(0.14)	0.51(0.11)
Pumadyn-8fm	1.00(0.22)	0.83(0.36)	0.92(0.68)	0.91(0.73)
Pumadyn-8fh	1.00(0.17)	0.80(0.17)	0.76(0.22)	0.71(0.19)
Pumadyn-8nm	1.00(0.18)	0.86(0.15)	0.85(0.20)	0.81(0.18)
Pumadyn-8nh	1.00(0.19)	0.85(0.14)	0.81(0.17)	0.77(0.15)

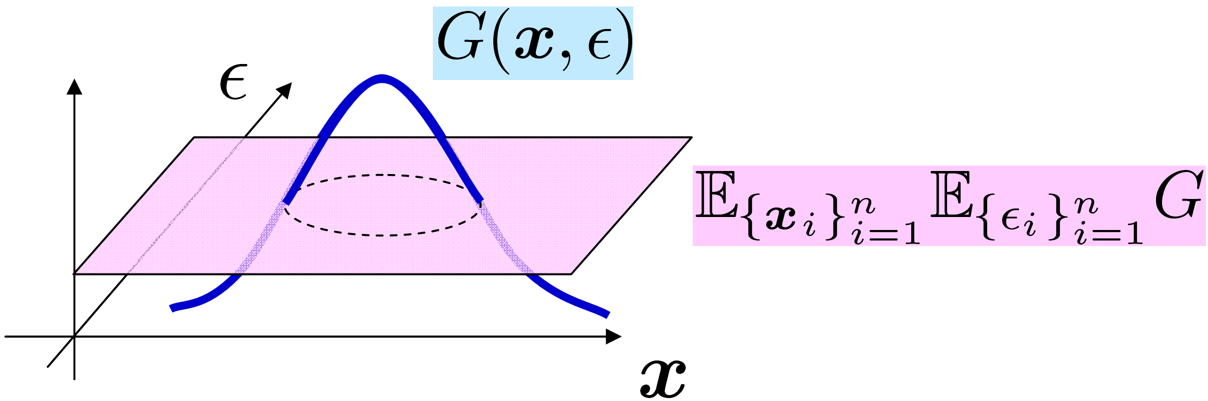
- All methods outperform passive.
- Ensemble method works the best!

Conclusions

- We have proposed
  - SIC for model selection
  - ALICE for active learning
  - Ensemble active learning for active learning with model selection
- Key issues of these methods are:
  - Input-dependence of generalization error estimation.
  - Approximate correctness of models.

Data-Independent Approach

- Evaluation of generalization error is in terms of average over both training inputs and noise.



- Model selection:

  Akaike information criterion (Akaike, IEEE-AC 1974)

  Cross validation

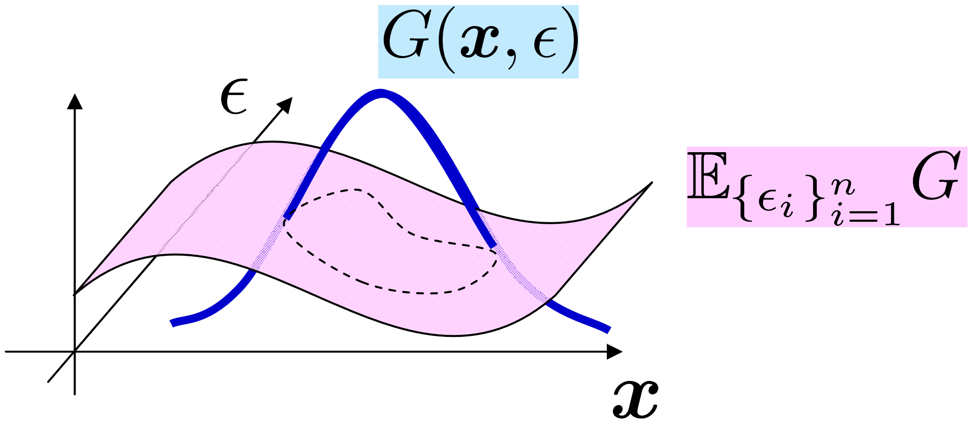
- Active learning:

  Wiens (JSPI 2000)

  Kanamori & Shimodaira (JSPI 2003)

Input-Dependent Approach

- Evaluation of generalization error is in terms of average over only noise (with fixed inputs).



- Input-dependent approach (such as SIC and ALICE) is provably more accurate than data-independent approach.

Sugiyama & Ogawa (Neural Comp. 2001)
Sugiyama & Müller (JMLR 2002, Stat. & Dec. 2005)
Sugiyama (JMLR 2006)

Approximate Correctness of Models

- Our model can never be correct in practice.
- However, our models may not be that bad.
- Learning with approximately correct models is practically important:
- SIC and ALICE are provably more accurate than other approaches for approximately correct models.