NIPS2005 Dec 5-8, 2005

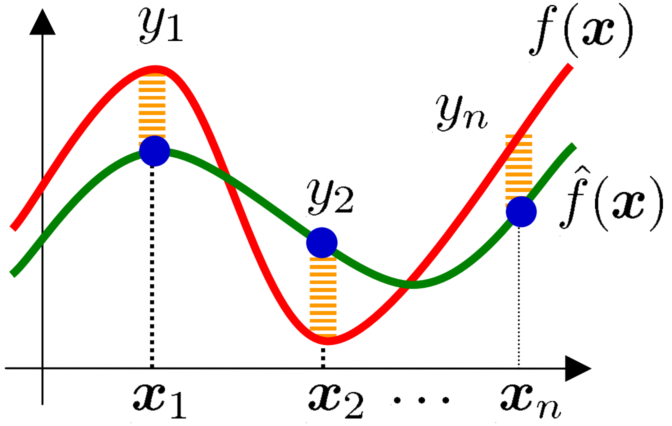
Active Learning for Misspecified Models

Masashi Sugiyama
Tokyo Institute of Technology

Abstract

The goal of active learning is to determine the locations of training input points so that the generalization error is minimized. We discuss the problem of active learning in linear regression scenarios. Traditional active learning methods using least-squares learning often assume that the model used for learning is correctly specified. In many practical situations, however, this assumption may not be fulfilled. Recently, active learning methods using ``importance''-weighted least-squares learning have been proposed, which are shown to be robust against misspecification of models. In this paper, we propose a new active learning method also using the weighted least-squares learning, which we call ALICE (Active Learning using the Importance-weighted least-squares learning based on Conditional Expectation of the generalization error). An important difference from existing methods is that we predict the conditional expectation of the generalization error given training input points, while existing methods predict the full expectation of the generalization error. By this difference, the training input design can be fine-tuned depending on the realization of training input points. Theoretically, we prove that the proposed active learning criterion is a more accurate predictor of the single-trial generalization error than the existing criterion in some sense. Numerical studies with toy and benchmark data sets show that the proposed method compares favorably to existing methods.

Linear Regression Problem



- Learning target:

$$f(\boldsymbol{x}) : \mathbb{R}^d \rightarrow \mathbb{R}$$

- Training examples:

$$\{\boldsymbol{x}_i, y_i\}_{i=1}^n \qquad y_i = f(\boldsymbol{x}_i) + \epsilon_i$$

$\{\epsilon_i\}_{i=1}^n$:iid, mean 0, variance σ^2

- 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

Generalization Error

- Goal of regression：
  Obtain a learned function $\hat{f}(\boldsymbol{x})$ that minimizes generalization error (expected error for unseen test input points).

- When test input points follows a density $q(\boldsymbol{x})$, generalization error is

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

Distribution of Training Input Points

- Passive learning (ordinary setting):
  Draw training input points from the same density as test input points

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

- Active learning (experimental design):
  Learner design training input density $p(\boldsymbol{x})$

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

$$p(\boldsymbol{x}) \neq q(\boldsymbol{x})$$

Active Learning

- Goal: Design training input density $p(\boldsymbol{x})$ so that the generalization error G is minimized

$$G(p) = \|\hat{f}_p - f\|^2$$

$$\hat{f}_p(\boldsymbol{x}) = \sum_{i=1}^{b} \alpha_i \varphi_i(\boldsymbol{x}) \qquad \boldsymbol{x}_i \overset{i.i.d.}{\sim} p(\boldsymbol{x})$$

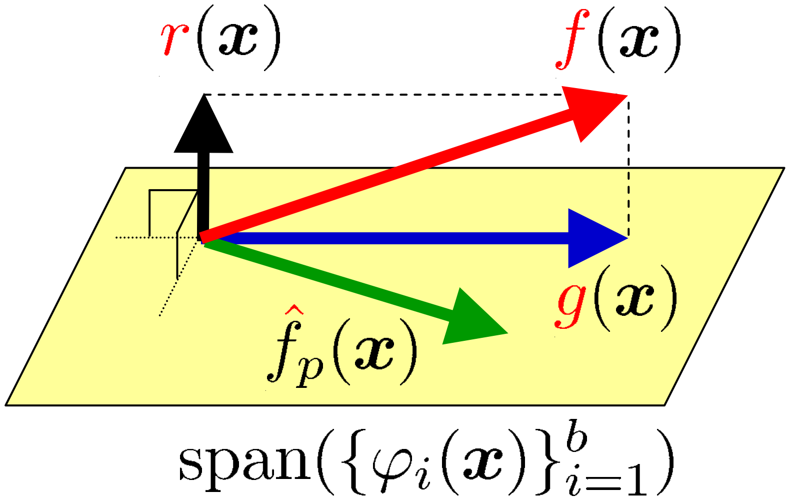
Need to predict generalization error before observing output values!

Decomposition of Learning Target

$$f(\boldsymbol{x}) = g(\boldsymbol{x}) + r(\boldsymbol{x})$$

- Linear part: $g(\boldsymbol{x}) = \sum_{i=1}^{b} \alpha_i^* \varphi_i(\boldsymbol{x})$
- Residual: $r(\boldsymbol{x}) \perp g(\boldsymbol{x})$

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


$r(\boldsymbol{x})$
$f(\boldsymbol{x})$
$g(\boldsymbol{x})$
$\hat{f}_p(\boldsymbol{x})$
$\mathrm{span}(\{\varphi_i(\boldsymbol{x})\}_{i=1}^{b})$

Bias/Variance Decomposition

$$\mathbb{E}_{\boldsymbol{\epsilon}} G(p) = C + B(p) + V(p)$$

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

$$G(p) = \|\hat{f}_p - f\|^2$$

- Model error (constant):

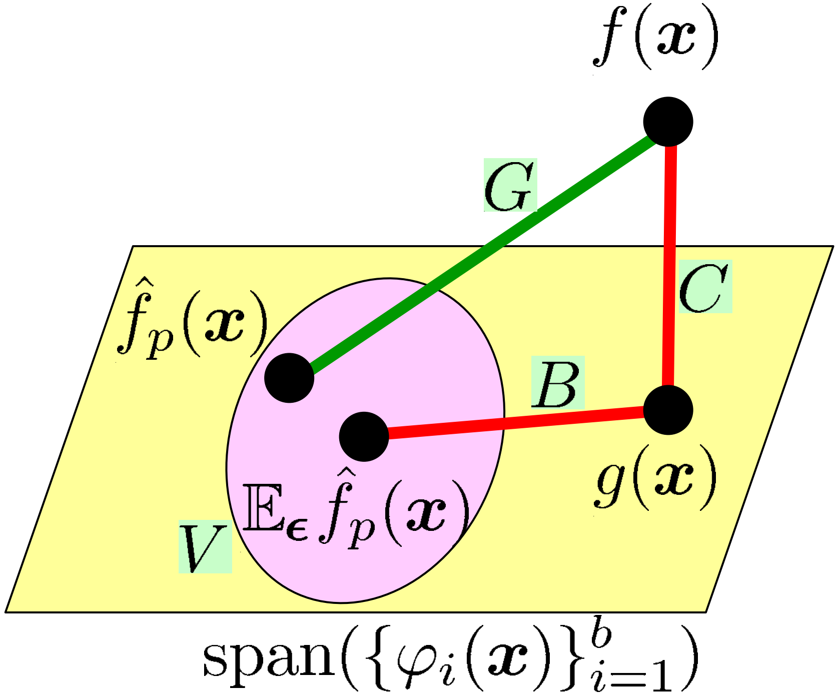
$$C = \|r\|^2$$

- Bias:

$$B(p) = \|\mathbb{E}_{\boldsymbol{\epsilon}} \hat{f}_p - g\|^2$$

- Variance:

$$V(p) = \mathbb{E}_{\boldsymbol{\epsilon}} \|\mathbb{E}_{\boldsymbol{\epsilon}} \hat{f}_p - \hat{f}_p\|^2$$


$f(\boldsymbol{x})$
G
C
$\hat{f}_p(\boldsymbol{x})$
B
$g(\boldsymbol{x})$
$\mathbb{E}_{\boldsymbol{\epsilon}} \hat{f}_p(\boldsymbol{x})$
V
$\mathrm{span}(\{\varphi_i(\boldsymbol{x})\}_{i=1}^{b})$

Ordinary Least Squares

- Ordinary least squares (OLS):

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

- Solution:

$$\hat{\boldsymbol{\alpha}}_O = \boldsymbol{L}_O \boldsymbol{y}$$

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

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

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

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

Active Learning with OLS

$$\mathbb{E}_{\boldsymbol{\epsilon}} G_O(p) = C + B_O(p) + V_O(p)$$

- Want to find $p(\boldsymbol{x})$ s.t. $\underset{p}{\operatorname{argmin}} \mathbb{E}_{\boldsymbol{\epsilon}}[G_O(p)]$
- However, bias is hard to predict
- Variance:

$$V_O(p) = \sigma^2 \mathrm{tr}(\boldsymbol{U}\boldsymbol{L}_O\boldsymbol{L}_O^\top)$$

$$\boldsymbol{U}_{i,j} = \langle \varphi_i, \varphi_j \rangle_{L_2(q)}$$
$$\boldsymbol{L}_O = (\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top$$
$$\boldsymbol{X}_{i,j} = \varphi_j(\boldsymbol{x}_i)$$

- Variance-only active learning (OLS-Based):

$$\min_{p(\boldsymbol{x})} J_O(p) \quad J_O(p) = \mathrm{tr}(\boldsymbol{U}\boldsymbol{L}_O\boldsymbol{L}_O^\top)$$

(e.g., Fedorov, 1972; Cohn, Ghahramani & Jordan, 1996; Fukumizu, 2000)

Can We Ignore Bias?

$$\delta = \|r\|$$

- If model is correct, bias is zero.

$$\delta = 0 \quad \Rightarrow \quad B_O(p) = 0$$

- If model is misspecified, bias is not zero even asymptotically

$$\delta \neq 0 \quad \Rightarrow \quad \lim_{n \to \infty} B_O \neq 0$$

- Solutions:
  - Take bias into account
  - Use learning method with smaller bias

Importance-Weighted LS

(Shimodaira, 2000)

- Importance-weighted LS (WLS):

$$\min_{\boldsymbol{\alpha}} \left[\sum_{i=1}^{n} \frac{q(\boldsymbol{x}_i)}{p(\boldsymbol{x}_i)} \left(\hat{f}(\boldsymbol{x}_i) - y_i \right)^2 \right] \qquad \hat{f}(\boldsymbol{x}) = \sum_{i=1}^{b} \alpha_i \varphi_i(\boldsymbol{x})$$

- Solution:

$$\hat{\boldsymbol{\alpha}}_W = \boldsymbol{L}_W \boldsymbol{y}$$

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

$$\text{cf. } \boldsymbol{L}_O = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top$$

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

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

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

$$\boldsymbol{D} = \mathrm{diag}\left(\frac{q(\boldsymbol{x}_i)}{p(\boldsymbol{x}_i)} \right)$$

Bias

$\delta = \|r\|$

Model	OLS	WLS
Correct $\delta = 0$	$B_O = 0$	$B_W = 0$
Misspecified $\delta \neq 0$	$B_O \not\to 0$	$B_W \to 0$

- Bias of WLS asymptotically vanishes!

Proposed Active Learning Method using WLS (ALICE)

- Variance:

$$V_W(p) = \sigma^2 \mathrm{tr}(\boldsymbol{U}\boldsymbol{L}_W\boldsymbol{L}_W^\top)$$

$$\boldsymbol{U}_{i,j} = \langle \varphi_i, \varphi_j \rangle_{L_2(q)} \qquad \boldsymbol{X}_{i,j} = \varphi_j(\boldsymbol{x}_i)$$

$$\boldsymbol{L}_W = (\boldsymbol{X}^\top \boldsymbol{D}\boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{D} \qquad \boldsymbol{D} = \mathrm{diag}\left(\frac{q(\boldsymbol{x}_i)}{p(\boldsymbol{x}_i)}\right)$$

- Variance-only active learning with WLS: ALICE

$$\min_{p(\boldsymbol{x})} J(p) \qquad J(p) = \mathrm{tr}(\boldsymbol{U}\boldsymbol{L}_W\boldsymbol{L}_W^\top)$$

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

Comparison (1): OLS

- Condition for being able to ignore bias when model is misspecified:
  - OLS-based: $\delta = o(n^{-1/2})$
  - ALICE: $\delta = o(1)$

$$\delta = \|r\|$$

⇒ ALICE is more general

- When model is correct, bias is zero for both OLS and WLS. However, OLS has smaller variance than WLS (cf. BLUE)

⇒ OLS-based is better

Comparison (2-1): Wiens

- ALICE: Minimize conditional variance given training input points

$$\min_{p(\boldsymbol{x})} J(p) \qquad J(p) = \mathrm{tr}(\boldsymbol{U}\boldsymbol{L}_W\boldsymbol{L}_W^\top)$$

- Wiens(2000): Minimize full expectation of variance

$$\min_{p(\boldsymbol{x})} J_W(p) \qquad J_W(p) = \mathrm{tr}(\boldsymbol{U}^{-1}\boldsymbol{T})$$

$$\boldsymbol{T}_{i,j} = \int \varphi_i(\boldsymbol{x})\varphi_j(\boldsymbol{x})\frac{q(\boldsymbol{x})^2}{p(\boldsymbol{x})}d\boldsymbol{x}$$

- Note: $J_W(p)$ does not depend on $\{\boldsymbol{x}_i\}_{i=1}^n$

Comparison (2-2): Wiens

$\delta = \|r\|$

- Accuracy of generalization error prediction:

When $\delta = o(n^{-\frac{1}{4}})$, if terms of $o(n^{-3})$ are ignored

$$\mathbb{E}_{\boldsymbol{\epsilon}}(\sigma^2 J_W - G_W)^2 \geq \mathbb{E}_{\boldsymbol{\epsilon}}(\sigma^2 J - G_W)^2$$

ALICE is more accurate predictor of single-trial generalization error G_W !

Comparison (2-3): Wiens

- Wiens(2000): Can analytical give optimal training input density

$$p_W^*(\boldsymbol{x}) = \frac{\widehat{h}(\boldsymbol{x})}{\int \widehat{h}(\boldsymbol{x}) d\boldsymbol{x}}$$

$$\widehat{h}(\boldsymbol{x}) = q(\boldsymbol{x}) \sqrt{\sum_{i,j=1}^{b} U_{i,j}^{-1} \varphi_i(\boldsymbol{x}) \varphi_j(\boldsymbol{x})}$$

- ALICE: Naïvely chooses best one from several candidate densities

Comparison (3-1): K&S

- Kanamori & Shimodaira (2003): Minimizes generalization error without ignoring bias by choosing training input points in 2 stages
- 1st stage: Predict generalization error using randomly chosen training input points
- 2nd stage: Design training input density so that predicted generalization error is minimized

Comparison (3-2): K&S

- Kanamori & Shimodaira (2003): Needs randomly gathered samples in 1st stage
- Learner can’t design all training input points
- Condition for being able to ignore bias when model is misspecified:
  - K&S: $\delta = O(1)$
  - ALICE: $\delta = o(1)$

$$\delta = \|r\|$$

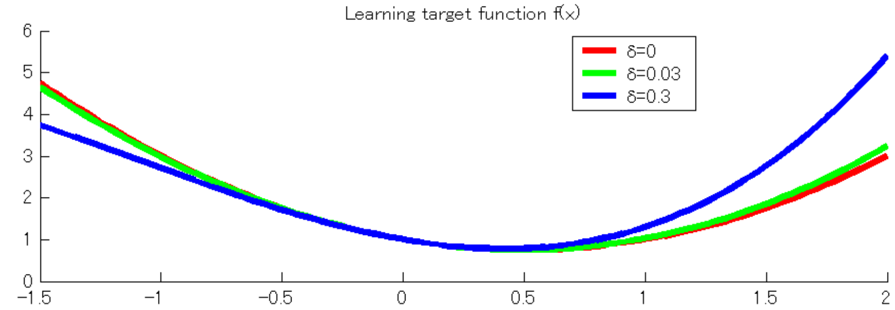
- K&S is more general, when model is totally misspecified. But learning is meaningless in this case.

Simulations (Toy)

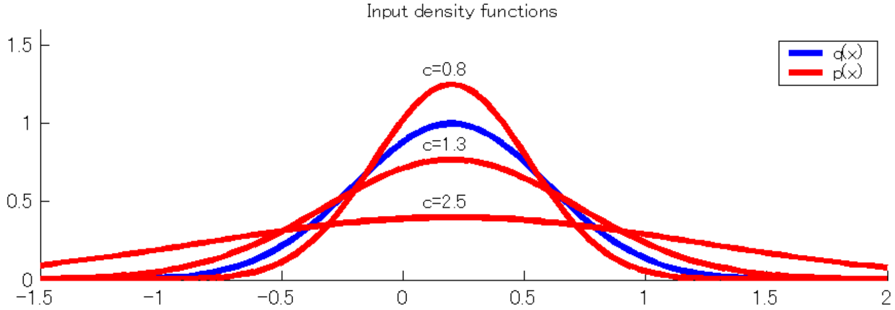
$\delta = 0, 0.03, 0.3$

- Learning target: $f(x) = 1 - x + x^2 + \delta x^3$
- Model: $\hat{f}(x) = \alpha_1 + \alpha_2 x + \alpha_3 x^2$
- Test input density: $\mathcal{N}(0.2, (0.4)^2)$
- Training input density: $\mathcal{N}(0.2, (0.4c)^2)$

$c = 0.8, 0.9, 1.0, \ldots, 2.5$



Learning target function f(x)



Input density functions

Obtained Generalization Error

Mean±Std (1000 trials) T-test (95%)

	$\delta = 0$	$\delta = 0.03$	$\delta = 0.3$
ALICE	2.07±1.90	2.09±1.90	4.28±2.02
Wiens	2.42±2.14	2.43±2.15	4.58±2.28
Wiens*	2.37±2.06	2.40±2.06	4.61±2.21
K&S	2.96±2.95	2.98±2.97	5.47±3.42
OLS-based	1.45±1.82	2.56±2.24	113±63.7
Passive	3.10±2.61	3.13±2.61	5.75±3.09

- When model is correct, OLS-based works well. But when model is misspecified, it is very poor.
- ALICE works well in all cases.

Simulations (DELVE)

- Test input density is unknown: Approximated by Gaussian

$$\mathcal{N}(\widehat{\boldsymbol{\mu}}_{MLE}, \widehat{\gamma}^2_{MLE})$$

- Training input density is chosen from

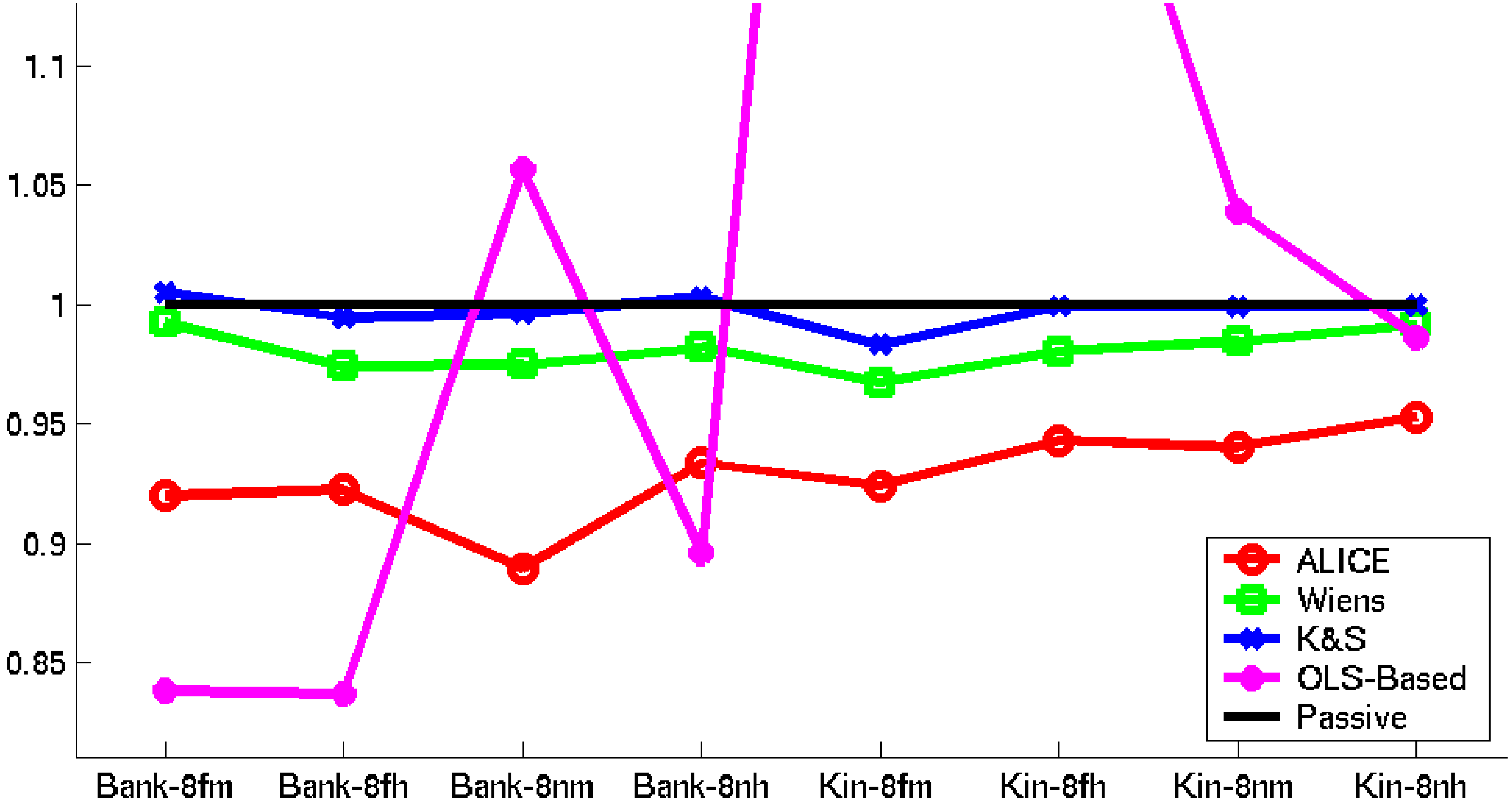
$$\mathcal{N}(\widehat{\boldsymbol{\mu}}_{MLE}, (c\widehat{\gamma}_{MLE})^2) \qquad c = 0.7, 0.75, 0.8, \ldots, 2.4$$

- Can not gather samples at arbitrary locations since only 8192 samples are available

→ Choose samples closest to desired locations

Obtained Test Errors

Mean over 100 trials (normalized by passive)



- OLS-based is sometimes good, but unstable.
- ALICE performs well and stable.