Active Learning with Model Selection in Linear Regression

Masashi Sugiyama    Neil Rubens

Department of Computer Science
Tokyo Institute of Technology, Japan
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

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

\[ x_i \overset{i.i.d.}{\sim} p_{train}(x) \]
\[ y_i = f(x_i) + \epsilon_i \]
\[ \epsilon_i \overset{i.i.d.}{\sim} \text{mean 0, variance } \sigma^2 \]

Goal: Learn \( f(x) \) from \( \{(x_i, y_i)\}_{i=1}^{n} \)
Linear Regression Model

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

- $\alpha_i$ : Parameter
- $\varphi_i(\mathbf{x})$ : Basis function

We do NOT assume our model is correct. ($f(\mathbf{x})$ is not necessarily included in the model).
Error Metric

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

\[
\left( \hat{f}(t) - f(t) \right)^2
\]

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

Learn \( \alpha \) so that generalization error is minimized

\[
\hat{f}(\mathbf{x}) = \sum_{i=1}^{b} \alpha_i \varphi_i(\mathbf{x})
\]

\( \alpha = (\alpha_1, \alpha_2, \ldots, \alpha_b)^T \)
**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

\[ x_i, t \sim i.i.d. P_{\text{train}}(x) \]


**Generalization error**

\[ G = \int \left( \hat{f}(x) - f(x) \right)^2 p_{\text{train}}(x) dx \]
Covariate Shift

Test and training input points follow different distributions.

\[ x_i \overset{i.i.d.}{\sim} p_{\text{train}}(x) \quad t \sim p_{\text{test}}(t) \quad p_{\text{train}}(x) \neq p_{\text{test}}(x) \]

Generalization error

\[ G = \int \left( \hat{f}(x) - f(x) \right)^2 p_{\text{test}}(x) \, dx \]
Example of Covariate Shift

(Weak) extrapolation:
Predict output values outside training region
Parameter Learning: Ordinary Least-Squares under Covariate Shift

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

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

OLS is not consistent
Law of Large Numbers

- Sample average converges to the population mean:

\[
\frac{1}{n} \sum_{i=1}^{n} A(x_i) \longrightarrow \int A(x) p_{\text{train}}(x) \, dx
\]

\[x_i \overset{i.i.d.}{\sim} p_{\text{train}}(x)\]

- We want to estimate the expectation over test input points from training input points \(\{x_i\}_{i=1}^{n}\).

\[
\int A(x) p_{\text{test}}(x) \, dx \quad t \sim p_{\text{test}}(x)
\]
Importance-Weighted Average

- **Importance**: Ratio of test and training input densities

\[
\frac{p_{\text{test}}(x)}{p_{\text{train}}(x)}
\]

- Importance-weighted average:

\[
\frac{1}{n} \sum_{i=1}^{n} \frac{p_{\text{test}}(x_i)}{p_{\text{train}}(x_i)} A(x_i) \rightarrow \int \frac{p_{\text{test}}(x)}{p_{\text{train}}(x)} A(x)p_{\text{train}}(x) dx
\]

\[
t \sim p_{\text{test}}(x)
\]

\[
x_i \text{ i.i.d. } p_{\text{train}}(x)
\]  

(cf. importance sampling)
Importance-Weighted LS for Covariate Shift

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

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

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

$$G = \int \left( \hat{f}(\mathbf{x}) - f(\mathbf{x}) \right)^2 p_{test}(\mathbf{x}) d\mathbf{x} = \| \hat{f} - f \|^2$$
Generalization Error Estimation

Generalization error is not accessible since the target function $f(x)$ is unknown.

Instead, we use a generalization error estimate.

\[ G = \| \hat{f} - f \|^2 \]
Assumption

We use **linear parameter learning**:

\[ \hat{\alpha} = Ly \]

\[ L : \text{matrix independent of training output noise} \]

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

\[ y = (y_1, y_2, \ldots, y_n)^\top \]

E.g., importance-weighted least-squares

\[ L = (X^\top DX)^{-1} X^\top D \]

\[ X_{i,j} = \varphi_j(x_i) \]

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

\[ D_{i,j} = \text{diag} \left( \frac{p_{\text{test}}(x_1)}{p_{\text{train}}(x_1)}, \ldots, \frac{p_{\text{test}}(x_n)}{p_{\text{train}}(x_n)} \right) \]
Estimating Generalization Error

\[ G = \| \hat{f} - f \|^2 \]
\[ = \| \hat{f} \|^2 + \| f \|^2 - 2 \langle \hat{f}, f \rangle \]

Accessible

Constant (ignored)

Estimated

\[ f = g + r \]
\[ g(x) = \sum_{i=1}^{b} \alpha_i^* \varphi_i(x) \]
\[ \langle r, \varphi_i \rangle = 0 \]

\[ \langle \hat{f}, f \rangle = \langle \hat{f}, g \rangle = \hat{\alpha}^\top U \alpha^* \]

\[ \hat{f}(x) = \sum_{i=1}^{b} \hat{\alpha}_i \varphi_i(x) \]
\[ U_{i,j} = \langle \varphi_i, \varphi_j \rangle \]
Subspace Information Criterion

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

\[ \hat{\alpha}^\top U \alpha^* \]

**Idea:** Replace \( \alpha^* \) by a linear unbiased estimator \( \tilde{\alpha} \)

\[ \tilde{\alpha} = \tilde{L} y \]

Since \( \tilde{\alpha} \) and \( \hat{\alpha} \) are estimated from the same sample \( y \), it causes a bias:

\[ \mathbb{E}_\epsilon[\hat{\alpha}^\top U \alpha^* - \hat{\alpha}^\top U \tilde{\alpha}] = \sigma^2 \text{tr}(U L \tilde{L}^\top) \]

\( \mathbb{E}_\epsilon \) : expectation over noise

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

Sugiyama & Müller (Statistics & Decisions 2005)

\[
\text{IWSIC}[L] = y^\top L^\top ULy - 2y^\top \tilde{L}^\top ULy + 2\tilde{\sigma}^2 \text{tr}(ULL\tilde{L}^\top)
\]

- \(U_{i,j} = \langle \varphi_i, \varphi_j \rangle\)
- \(\tilde{\alpha} = Ly\)
- \(\tilde{\alpha} = Ly\)
- \(\tilde{X} : X \text{ for largest model}\)
- \(\tilde{\sigma}^2 = \|Gy\|^2 / \text{tr}(G)\)
- \(G = I - \tilde{X}(\tilde{X}^\top \tilde{X})^{-1} \tilde{X}^\top\)
- \(D_{i,j} = \text{diag}\left(\frac{p_{\text{test}}(x_1)}{p_{\text{train}}(x_1)}, \ldots, \frac{p_{\text{test}}(x_n)}{p_{\text{train}}(x_n)}\right)\)

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

\[
\mathbb{E}_\varepsilon(\text{IWSIC} - G - C) = \mathcal{O}_p(\delta n^{-1/2})
\]

- \(\delta : \text{model error} \ (= \|r\|)\)
- \(\mathbb{E}_\varepsilon : \text{expectation over noise}\)
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.

\[ \mathbb{E}_\epsilon (\text{IWSIC} - G - C) = O_p(\delta n^{-1/2}) \]

\( \delta \): model error
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

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

Polynomial of order 3

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

- 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:

\[ G = \int \left( \hat{f}(\mathbf{x}) - f(\mathbf{x}) \right)^2 p_{test}(\mathbf{x}) \, d\mathbf{x} \]
Batch active learning: optimize location of all training inputs $\{x_i\}_{i=1}^n$ 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 $p_{\text{train}}(x)$ and draw training inputs from it.
Generalization error is not accessible since the target function \( f(x) \) is unknown.

Instead, we use a generalization error estimate.

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

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

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

\[
\hat{\alpha} = Ly
\]

\[
L = (X^\top D X)^{-1} X^\top D
\]

\[
X_{i,j} = \varphi_j(x_i)
\]

\[
D_{i,j} = \text{diag} \left( \frac{p_{\text{test}}(x_1)}{p_{\text{train}}(x_1)}, \ldots, \frac{p_{\text{test}}(x_n)}{p_{\text{train}}(x_n)} \right)
\]

\[
y = (y_1, y_2, \ldots, y_n)^\top
\]
$$\mathbb{E}_\epsilon G = \mathbb{E}_\epsilon \| \hat{f} - f \|^2 = \delta^2 + B + V$$

- **Model error:**
  $$\delta = \| f - g \|$$

- **Bias:**
  $$B = \| \mathbb{E}_\epsilon \hat{f} - g \|^2$$

- **Variance:**
  $$V = \mathbb{E}_\epsilon \| \mathbb{E}_\epsilon \hat{f} - \hat{f} \|^2$$

$\mathbb{E}_\epsilon$: expectation over noise
We want to estimate $\mathbb{E}_\varepsilon G$ without using $\{y_i\}_{i=1}^n$.

- Model error: constant and can be ignored
  $$\delta = \| f - g \|$$

- Variance: computable up to scaling factor $\sigma^2$
  $$V = \mathbb{E}_\varepsilon \| \mathbb{E}_\varepsilon \hat{f} - \hat{f} \|^2 = \sigma^2 \text{tr}(U L L^\top) = O_p(n^{-1})$$

- Bias: hard to estimate, but can be safely ignored if $\delta = o(1)$
  $$B = \| \mathbb{E}_\varepsilon \hat{f} - g \|^2 = O_p(\delta^2 n^{-1})$$
ALICE

Sugiyama (JMLR 2006)

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

\[
\text{ALICE}[\rho_{\text{train}}] = \text{tr}(ULL^\top)
\]

\[
U_{i,j} = \langle \varphi_i, \varphi_j \rangle
\]

\[
L = (X^\top DX)^{-1}X^\top D
\]

\[
X_{i,j} = \varphi_j(x_i)
\]

\[
D_{i,j} = \frac{p_{\text{test}}(x_i)}{p_{\text{train}}(x_i)} \delta_{i,j}
\]

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

\[
\sigma^2 \text{ALICE} - G + \delta^2 = O_p(n^{-1})
\]
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 $p_{\text{train}}(x)$
  \[
  \min_{p_{\text{train}}} G(p_{\text{train}})
  \]

- **ALMS**: optimize both $\mathcal{M}$ and $p_{\text{train}}(x)$
  \[
  \min_{\mathcal{M}, p_{\text{train}}} G(\mathcal{M}, p_{\text{train}})
  \]

\[
G = \int \left( \hat{f}(x) - f(x) \right)^2 p_{\text{test}}(x) dx
\]
Suppose there exist the common optimal training input density for all model candidates.

\[ p^*_{\text{train}} = \arg\min_{p_{\text{train}}} G(\mathcal{M}, p_{\text{train}}) \text{ for all } \mathcal{M} \]

Then using \( p^*_{\text{train}} \) 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.

Sugiyama & Ogawa (IEICE Trans. 2003)
Can we simply employ existing MS and AL methods for simultaneously optimizing $\mathcal{M}$ and $p_{\text{train}}(\mathbf{x})$?

**AL/MS dilemma:**
- MS methods require to fix $p_{\text{train}}(\mathbf{x})$.
- 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.

\[
\begin{align*}
\text{Start} & \\
\text{Choose an initial model } M^{(i)} & \\
\text{Choose the next training input point } x_{i+1} & \\
\text{Gather output value } y_{i+1} \text{ at } x_{i+1} & \\
\text{Choose a model } M^{(i+1)} & \\
i = i + 1 & \\
i \geq n? & \\
\text{No} & \\
\text{Yes} & \\
\text{End} &
\end{align*}
\]
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.

![Diagram of the Batch Approach process]

1. Start
2. Choose an initial model $M^{(0)}$
3. Choose all training input points $\{x_i\}_{i=1}^{n}$
4. Gather all output values $\{y_i\}_{i=1}^{n}$ at $\{x_i\}_{i=1}^{n}$
5. Choose the final model $M^{(n)}$
6. End

The choice of models:
- Optimal
- Poor

The number of training samples:
1, 2, ..., n
We need to choose an initial model before observing training samples $\{(x_i, y_i)\}_{i=1}^{n}$.

- IWSIC can not be computed without $\{(x_i, y_i)\}_{i=1}^{n}$.
- ALICE can be computed without $\{(x_i, y_i)\}_{i=1}^{n}$, 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.
Choose training input density for all models:

\[
\min_{p_{\text{train}}} \sum_{\mathcal{M}} G(\mathcal{M}, p_{\text{train}})
\]

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

\[
\text{Start}
\]

Choose all training input points \(\{x_i\}_{i=1}^n\) for ensemble of all models

Gather all output values \(\{y_i\}_{i=1}^n\) at \(\{x_i\}_{i=1}^n\)

Choose the final model \(M^{(n)}\)

\[
\text{End}
\]

Sugiyama & Rubens (2007)
## Simulation Results

All methods outperform passive.

- **Ensemble method works the best!**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Passive</th>
<th>Sequential</th>
<th>Batch</th>
<th>Ensemble</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bank-8fm</td>
<td>1.00(1.22)</td>
<td>0.59(0.85)</td>
<td>0.46(0.25)</td>
<td>0.45(0.28)</td>
</tr>
<tr>
<td>Bank-8fh</td>
<td>1.00(0.42)</td>
<td>0.53(0.22)</td>
<td>0.46(0.18)</td>
<td>0.44(0.11)</td>
</tr>
<tr>
<td>Bank-8nm</td>
<td>1.00(0.76)</td>
<td>0.63(0.19)</td>
<td>0.58(0.21)</td>
<td>0.56(0.10)</td>
</tr>
<tr>
<td>Bank-8nh</td>
<td>1.00(0.28)</td>
<td>0.61(0.19)</td>
<td>0.53(0.14)</td>
<td>0.51(0.11)</td>
</tr>
<tr>
<td>Pumadyn-8fm</td>
<td>1.00(0.22)</td>
<td>0.83(0.36)</td>
<td>0.92(0.68)</td>
<td>0.91(0.73)</td>
</tr>
<tr>
<td>Pumadyn-8fh</td>
<td>1.00(0.17)</td>
<td>0.80(0.17)</td>
<td>0.76(0.22)</td>
<td>0.71(0.19)</td>
</tr>
<tr>
<td>Pumadyn-8nm</td>
<td>1.00(0.18)</td>
<td>0.86(0.15)</td>
<td>0.85(0.20)</td>
<td>0.81(0.18)</td>
</tr>
<tr>
<td>Pumadyn-8nh</td>
<td>1.00(0.19)</td>
<td>0.85(0.14)</td>
<td>0.81(0.17)</td>
<td>0.77(0.15)</td>
</tr>
</tbody>
</table>
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.
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)
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 (JMLR 2006)
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.