Introduction to Statistical Machine Learning

Xiang Zhou

School of Data Science
Department of Mathematics
City University of Hong Kong

Statistical Learning: Bias-Variance Trade-off

Bias-Variance Decomposition / Trade-off

Let f be a given deterministic function (ground truth) $\mathcal{X} \to \mathcal{Y}$.
Assume that the response r.v. Y is defined by the additive error statistical model:

$$Y = f(X) + \varepsilon$$

where the r.v. ε is independent of X and has mean 0 and variance σ_ε^2.
Assume this model generates the training dataset
$\mathtt{D} = \{(x_i, y_i) : 1 \leq i \leq N\}$, i.e., $y_i = f(x_i) + \varepsilon_i$ and iid $\varepsilon_i \sim \varepsilon$ Assume one learns a function based on $\mathtt{D}$, still denoted by $\hat{f}_{\mathtt{D}}(\cdot)$. [1]

[1] it does not have to be the one in previous regression part associated with square loss. For example, $\hat{f}_{\mathtt{D}}$ can be the k-NN model.

- The question is the prediction performance of this function $\hat{f}_{\mathtt{D}}$ on a new data (x_0, y_0) where $x_0 \in \mathcal{X}$ is arbitrary and $y_0 = f(x_0) + \varepsilon_0$ with the new measurement error ε_0 being independent of $\mathtt{D}$ and with distributed the same as ε.
- The subtlety here is that $\mathtt{D}$ is random *per se*. $\mathcal{E}(\hat{f}_{\mathtt{D}})$ is also uncertain [1]. So, the average w.r.t. to $\mathtt{D}$ is more relevant [2]

$$\mathbb{E}_{\mathtt{D}}\, \mathcal{E}(\hat{f}_{\mathtt{D}})$$

- The population risk for the trained model is

$$\mathcal{E}(\hat{f}_{\mathtt{D}}) = \mathbb{E}_{X,Y}\, \ell(Y, \hat{f}_{\mathtt{D}}(X)) = \mathbb{E}_X\, \mathbb{E}_{Y|X} \left[\ell(Y, \hat{f}_{\mathtt{D}}(X)) | X \right].$$

  It suffices to consider on a generic test point $x_0 \in \mathcal{X}$. [3]

$$\mathbb{E}\left[\ell(Y, \hat{f}_{\mathtt{D}}(X = x_0)) | X = x_0 \right]$$

  This is the expected predicted error (EPR).

---

[1] different training data give rise to different $\hat{f}_{\mathtt{D}}$

[2] compare to the upper bound in learning theory before

[3] x_0 is non-random. The expectation is w.r.t. the training data $\mathtt{D}$ and $Y|X = x_0$, i.e., the measurement noise ε

Decomposition of expected prediction error

for mean square loss in regression

The **expected prediction error** for the means square loss at x_0 is

$$\mathbb{E}[(Y - \hat{f}_{\mathtt{D}}(X))^2 | X = x_0] = \mathbb{E}_{\varepsilon_0, \mathtt{D}}(f(x_0) + \varepsilon_0 - \hat{f}_{\mathtt{D}}(x_0))^2$$

$$= \sigma_\varepsilon^2 + \mathbb{E}_{\mathtt{D}} \left(f(x_0) - \hat{f}_{\mathtt{D}}(x_0) \right)^2 \quad \because \mathbb{E}(\varepsilon_0) = 0, \text{ and } \mathtt{D} \text{ indp.} \varepsilon_0$$

$$= \sigma_\varepsilon^2 + \mathbb{E}_{\mathtt{D}} \left(f(x_0) - \mathbb{E}_{\mathtt{D}} \hat{f}_{\mathtt{D}}(x_0) + \mathbb{E}_{\mathtt{D}} \hat{f}_{\mathtt{D}}(x_0) - \hat{f}_{\mathtt{D}}(x_0) \right)^2$$

$$= \sigma_\varepsilon^2 + \underbrace{\left(f(x_0) - \mathbb{E}_{\mathtt{D}} \hat{f}_{\mathtt{D}}(x_0) \right)^2}_{\text{Bias}^2} + \underbrace{\text{Var}_{\mathtt{D}} \left(\hat{f}_{\mathtt{D}}(x_0) \right)}_{\text{Variance}}$$

Here the subscripts emphasize the random elements to take expectation.
$\sigma_\varepsilon^2 = \text{Var}(\varepsilon_0)$ is the irreducible uncertainty of the new measurement error.

Definition

$\hat{f}_{\mathtt{D}}$ is called **unbiased** *if the functions $\mathbb{E}_{\mathtt{D}} \hat{f}_{\mathtt{D}}(x)$ and $f(x)$ are equal.*

For linear regression:

$$\text{Var}(\hat{f}_{\mathtt{D}}(x_0)) \sim p\sigma_\varepsilon^2/N$$

where $p = \dim(\mathcal{X})$.

- low bias: large model space, low training error, overfitting, bad generalization ability (high variance);
- low variance: rigid model space, insensitive to the perturbation of the dataset used in fitting; good extrapolation on the new data from the same distribution.
- BAD news[1] : it is almost impossible to decrease the bias and variance terms simultaneously!
- Criteria for model assessment or variable selection: **good trade-off between the bias and variance**

[1] Good news: it is possible and in fact it might be very common for deep learning

scikit-learn : Underfitting vs. Overfitting

Fit a polynomial function by (ordinary) linear regression
https://scikit-learn.org/stable/auto_examples/model_selection/plot_underfitting_overfitting.html

conceptual diagram

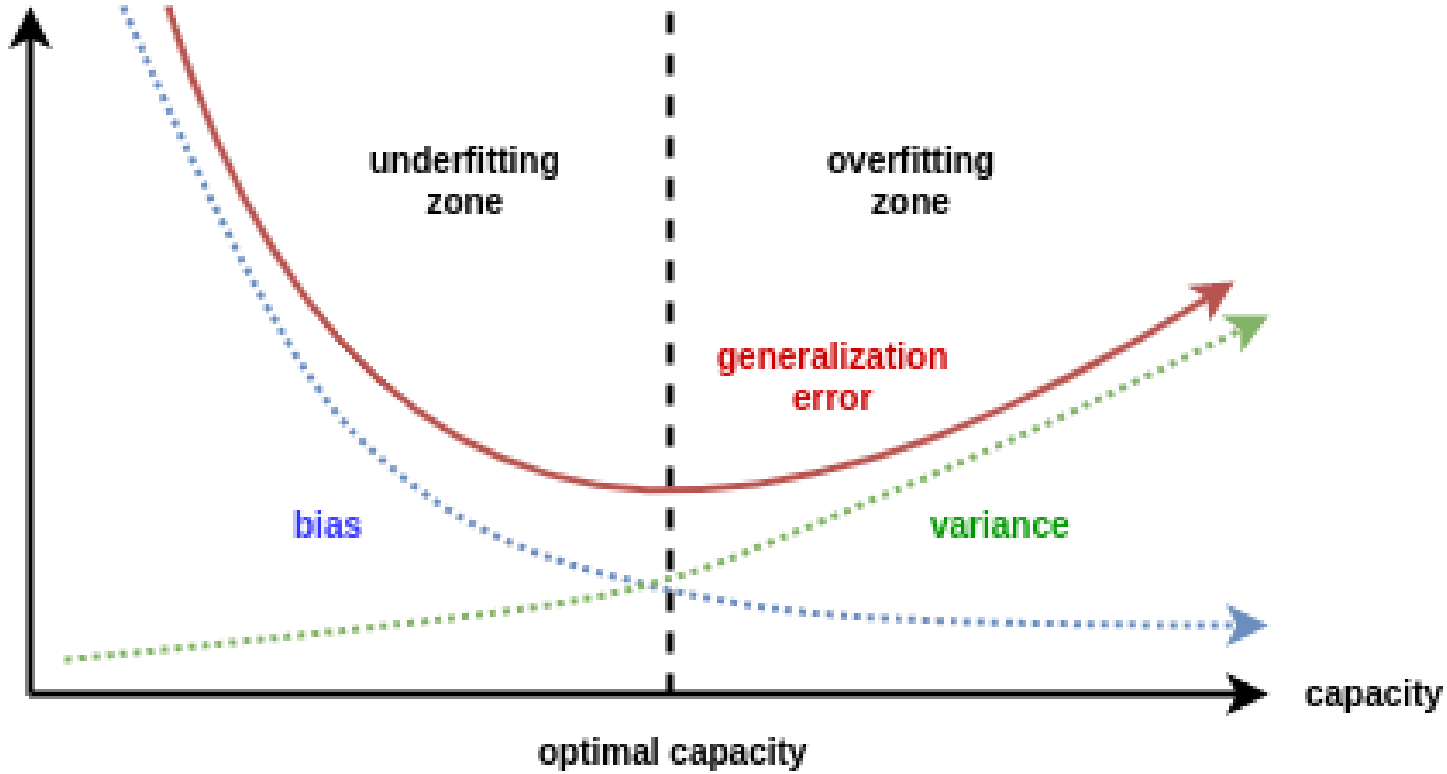

Figure: Bias-Variance Tradeoff as a Function of Model Capacity/Complexity

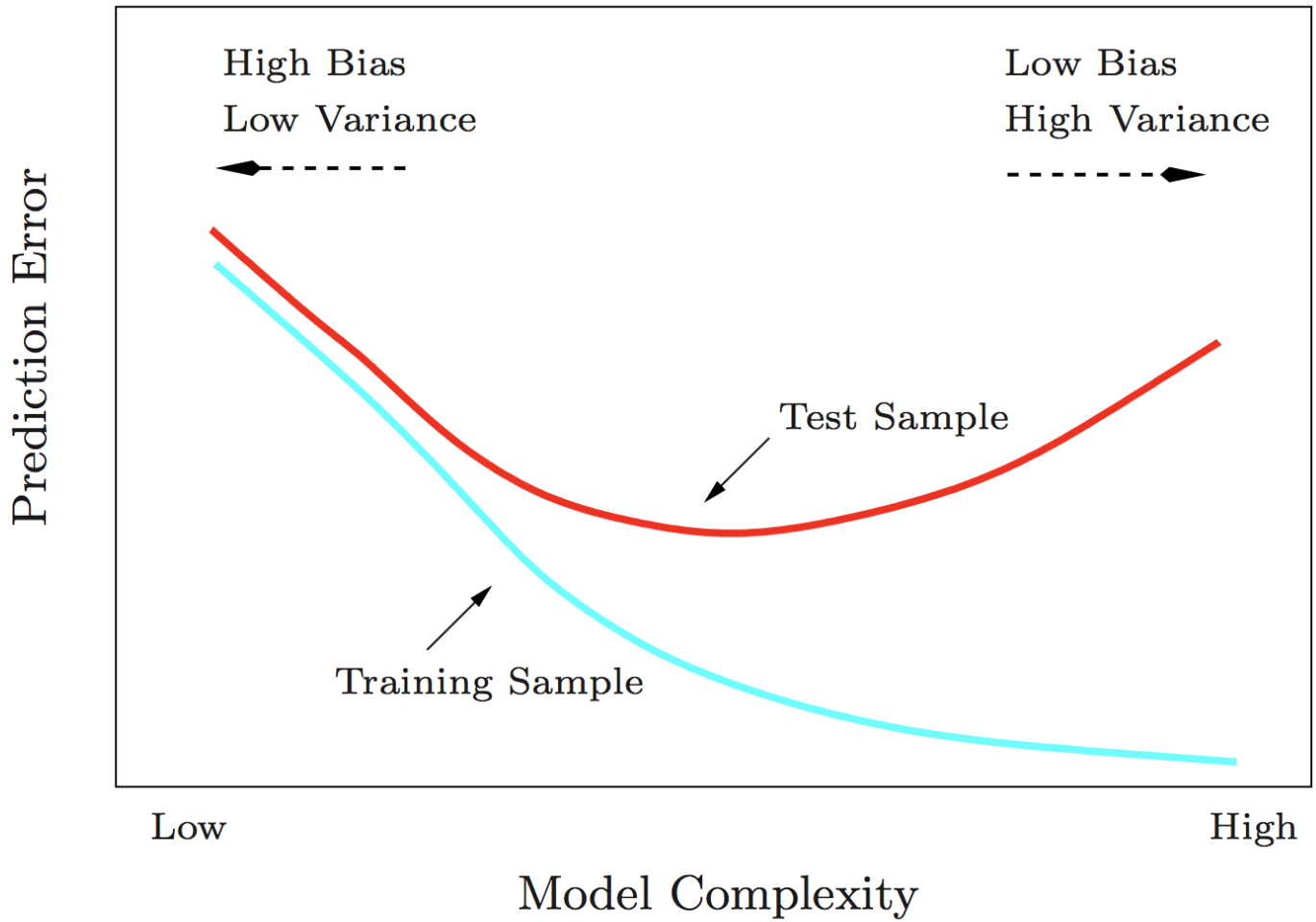

FIGURE 2.11. *Test and training error as a function of model complexity.*

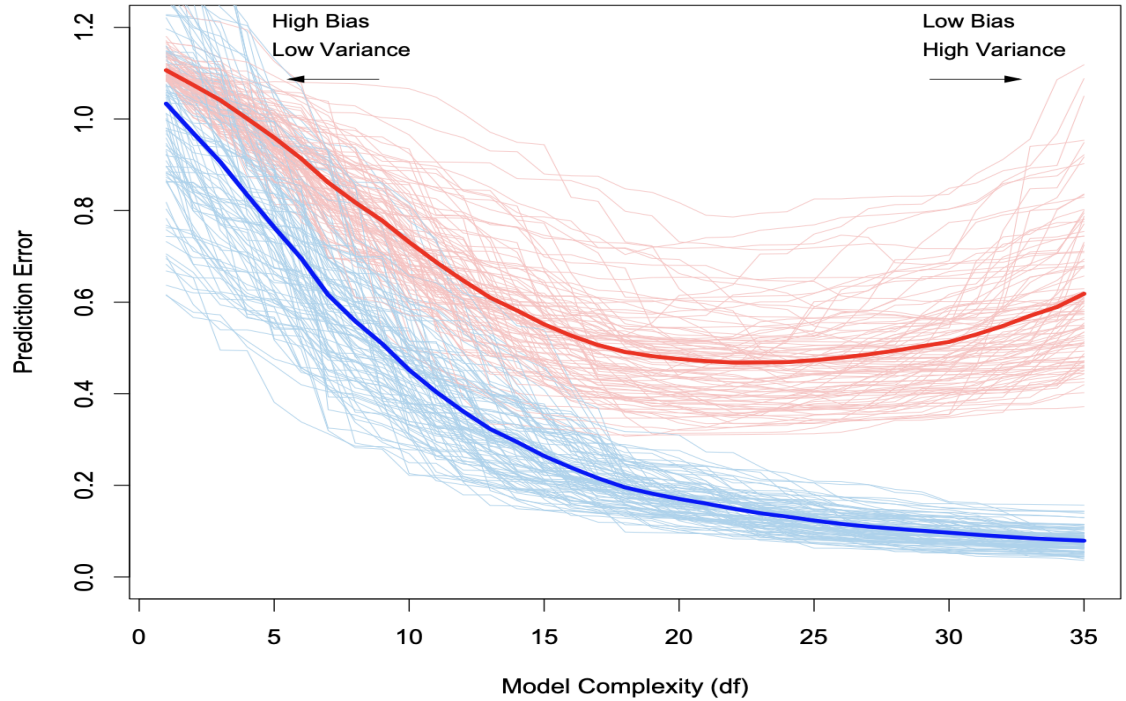

FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\text{err}}$*, while the light red curves show the conditional test error* $\text{Err}_{\mathcal{T}}$ *for* 100 *training sets of size* 50 *each, as the model complexity is increased. The solid curves show the expected test error* Err *and the expected training error* $\text{E}[\overline{\text{err}}]$*.*

capacity/complexity

- capacity is the most important concept in Vapnik's theory , related to VC dimension;
- "complexity" may be used exchangeable with "capacity".
- but they are NOT equal to the number of parameter in the model. Sometimes, they are called "effective number of parameters" like in ridge/lasso regression.

Application of Bias-Variance decomposition to k-NN

Consider the k-nearest-neighbor regression fit to the data $\mathtt{D} = \{(x_i, y_i)\}$ arising from the additive model $Y = f(X) + \varepsilon$:

$$\hat{f}^k_{\mathtt{D}}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i = \frac{1}{k} \sum_{x_i \in N_k(x)} (f(x_i) + \varepsilon_i).$$

Assume that the input design points $\{x_i\}$ is deterministic. Then the expectation w.r.t. $\mathtt{D}$ is only for the measurement errors ε_i:

- The bias is $f(x_0) - \mathbb{E}_{\mathtt{D}}[\hat{f}^k_{\mathtt{D}}(x_0)] = f(x_0) - \frac{1}{k} \sum_{x_i \in N_k(x_0)} f(x_i)$;
- The variance is $\mathrm{Var}_{\mathtt{D}}(\hat{f}^k_{\mathtt{D}}(x_0)) = \frac{1}{k}\sigma^2_\varepsilon$

Model Assessment: practical techniques

Typical objectives:

1. Choose a set of tuning parameter (hyperparameter) used in the model (such as k in k-NN)
2. Estimate the prediction performance (test error) of a given model

Remarks:

- For both objectives, the best approach is to run the procedure on an independent test set, if one is available.
- If possible, one should use different test data for (1) and (2) above: a *validation set* for objective (1) and a *test set* for objective (2).
- Often there is insufficient data to create a separate validation or test set. In this case, *Cross-Validation* is useful.

K-fold cross validation

Denote the hyper-parameter by λ. K-fold cross validation is the most popular method for estimating a tuning parameter λ.

Divide the dataset (of size N) into K subsets: $\mathcal{A}_1, \ldots, \mathcal{A}_K$ ($K = 2, 5, 10$ or N)

- For each $k = 1, \ldots, K$, fit the model with parameter λ to $\{\mathcal{A}_1, \ldots, \mathcal{A}_{k-1}, \mathcal{A}_{k+1}, \ldots, \mathcal{A}_K\}$ giving $f_\lambda^{-k}(\cdot)$, and compute its prediction error on $\mathcal{A}_k$:

$$E_k(\lambda) = \sum_{x_i \in \mathcal{A}_k} \ell(y_i, f_\lambda^{-k}(x_i)).$$

- The average of these K values $E_k(\lambda)$ give the cross-validation error (per sample)

$$CV(\lambda) := \frac{1}{K} \sum_{k=1}^{K} E_k(\lambda).$$

- Choose the optimal λ^* yielding the smallest $CV(\lambda)$.

K-fold cross validation

- Cross-validation is often abbreviated as CV.
- In the subset selection procedure, λ is the subset size
- $f^{-k}(\lambda)$ is the best model of size λ, found from the training set that leaves out the k-th part of the data
- $E_k(\lambda)$ is its estimated test error on the k-th part.
- Using K-fold CV, the K test error estimates are averaged to give the final CV estimated test error.
- The output is the model associated with λ^*, typically, computed by using all N data.

scikit-learn : model evaluation

In practice, various "score" are used for evaluation of various models. See details at

https://scikit-learn.org/stable/modules/model_evaluation.html

This long list is simply to answer one single question;

'What criterion/score in practice to judge your model is good".

`scikit-learn` : cross-validation with linear models to select LASSO penalty α

https://scikit-learn.org/stable/auto_examples/exercises/plot_cv_diabetes.html

Bootstrap

- Bootstrap works by sampling N times with replacement from the training set to form a “bootstrap" data set. Then model is estimated on the bootstrap data set, and predictions are made on the original training set.
- This process is repeated many times and the results are averaged.
- *Bootstrap is most useful for estimating standard errors of predictions.*
- Can also use modified versions of the bootstrap to estimate prediction error.