Errors and Cross Validation: Basics

Wei Li

Syracuse University

Spring 2024

OVERVIEW

Prediction Errors and MSE

Problem of prediction

- (X, Y) has a joint distribution given by $P(X, Y)$
- $Y = f(X) + \epsilon$
- $E(\epsilon|X) = 0$ and $Var(\epsilon|X) = \sigma^2$.

We are agnostic about the form of f and is only interested in getting a good prediction for Y using some model for f.

Linear regression prediction

To predict at some arbitrary point x using a simple linear regression, take an approximation $f(x) \approx \beta_0 + \beta_1 x$

$$\begin{aligned}\hat{f}(x) &= \hat{\alpha} + \hat{\beta} x \\ &= \cdots \\ &= \sum_{i=1}^{n} \frac{1}{n} \left(1 + \frac{(x - \bar{x})(x_i - \bar{x})}{\hat{\sigma}_X^2}\right) y_i\end{aligned}$$

where $\hat{\sigma}_X^2 = \sum_i (x_i - \bar{x})^2 / n$.

Note that it takes the form

$$\hat{f}(x) = \sum_{i=1}^{n} w(x, x_i) \cdot y_i = \mathbf{w}(x)\mathbf{y}$$

We call a regression prediction of this form a **linear smoother** (linear in y).

Polynomial regression

With polynomial regression

$$f(x) \approx \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \ldots + \beta_d x^d$$

In general, approximate $f(x)$ using a series of approximating functions, $f(x) \approx \sum_j \beta_j b_j(x)$, where $\{b_j : j = 1, \ldots, M\}$ is some basis functions $(M < n)$.

Let $\mathbf{B}$ denote the matrix whose (i, j)-th element is given by $b_j(x_i)$. Then the estimate for β is given by

$$\hat{\beta} = \left(\mathbf{B}^T \mathbf{B}\right)^{-1} \mathbf{B}^T \mathbf{y}$$

Then $\hat{f}(x) = b(x)^T \hat{\beta} = b(x)^T \left(\mathbf{B}^T \mathbf{B}\right)^{-1} \mathbf{B}^T \mathbf{y}$,
$b(x) = (b_1(x), \ldots, b_M(x))^T$.

k-nearest-neighbors (KNN) regression

With

$$w(x, x_i) = 1(x_i \in N_k(x))/k,$$

$$\hat{f}(x) = \frac{1}{k} \sum_{i \in N_k(x)} y_i = \sum_i w(x, x_i)\, y_i,$$

where $N_k(x)$ gives the k nearest neighbors of x.

*memory-based model (lazy learner or examplar-based learner)

Prediction Errors and MSE

Suppose the (X, Y) has a joint distribution given by $P(X, Y)$, which can be represented as

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

with $E(\epsilon|X) = 0$ and $Var(\epsilon|X) = \sigma^2$.

Denote random sample (i.i.d.) $\boldsymbol{\tau} = \{(\boldsymbol{x}_i, y_i), i = 1, \cdots, n\}$.

Using $\boldsymbol{\tau}$, we can obtain some prediction rule $\cdot \mapsto \hat{f}(\cdot)$ (depends on $\boldsymbol{\tau}$):

- linear regression, $\cdot \mapsto \hat{f}(\cdot) = \hat{\beta}^T(\cdot)$,
- polynomial regression, $\cdot \mapsto \hat{f}(\cdot) = \hat{\beta}^T(\cdot)$,
- knn regression, $\cdot \mapsto \hat{f}(\cdot) = \frac{1}{k}\sum_{i \in N_k(\cdot)} y_i$.

The prediction rule for Y given $X = x$ is $\hat{y} = \hat{f}(x)$.

*let's focus on regression in this lesson

How to obtain a $\hat{f}$ in general?

- ERM (empirical risk minimization)
- MLE (model-based approach)

ERM:

$$\hat{f} \in \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} L(Y_i, f(X_i))$$

- $L(\cdot, \cdot)$: a loss function
  - what loss function? depending on the nature of the problem.
- $\mathcal{F}$: a class of candidate functions
  - larger or smaller class? more subtle.

Goal of prediction

Let (X_0, Y_0) be an arbitrary test point (future observation), satisfying

$$Y_0 = f(X_0) + \epsilon_0, \qquad (X_0, Y_0) \sim P(X, Y)$$

and independent of $\boldsymbol{\tau}$.

What is the right criterion for assessing how $\hat{f}$ perform?

Expected prediction (test) error (EPE) for $\hat{f}$:

$$\text{EPE} = \text{E}[L(Y_0, \hat{f}(X_0))]$$

where expectation is taken over the training set $\boldsymbol{\tau} = \{X_i, Y_i\}_{i=1}^{n}$ and some test point (X_0, Y_0).

A basic fact:

$$\text{E}[L(Y_0, \hat{f}(X_0))] \geq \inf_{f} \text{E}[L(Y_0, f(X_0))]$$

Decomposition of EPE

Consider the squared error loss

$$\text{EPE} = E_{X_0}\left(E_{(\boldsymbol{\tau},Y)|X_0}\left[\hat{Y}_0 - Y_0\right]^2 \Big| X_0\right)$$

Expected Prediction/test Error (EPE) of $\hat{f}$ at $\boldsymbol{x}_0$:

$$\begin{aligned}\text{EPE}\left(\boldsymbol{x}_0\right) &= E_{Y_0|X_0=\boldsymbol{x}_0} E_{\boldsymbol{\tau}}\left[\hat{Y}_0 - Y_0\right]^2 = E_{Y_0|X_0=\boldsymbol{x}_0} E_{\boldsymbol{\tau}}\left[\hat{f}\left(\boldsymbol{x}_0\right) - Y_0\right]^2 \\ &= \sigma^2 + \mathrm{E}_{\boldsymbol{\tau}}\left[\hat{f}\left(\boldsymbol{x}_0\right) - f\left(\boldsymbol{x}_0\right)\right]^2\end{aligned}$$

Decomposition of MSE

- **Mean Squared Error** (MSE) for $\hat{f}$ at $\boldsymbol{x}_0$:

$$\mathrm{MSE}\left(\boldsymbol{x}_0\right) = \mathrm{E}_{\boldsymbol{\tau}}\left[\hat{Y}_0 - f\left(\boldsymbol{x}_0\right)\right]^2 = \mathrm{E}_{\boldsymbol{\tau}}\left[\hat{f}\left(\boldsymbol{x}_0\right) - f\left(\boldsymbol{x}_0\right)\right]^2$$

We can decompose the $\mathrm{MSE}\left(\boldsymbol{x}_0\right)$ as

$$\begin{aligned}\mathrm{MSE}\left(\boldsymbol{x}_0\right) &= E_{\boldsymbol{\tau}}\left[\hat{f}\left(\boldsymbol{x}_0\right) - f\left(\boldsymbol{x}_0\right)\right]^2 \\ &= \mathrm{Var}_{\boldsymbol{\tau}}\left[\hat{f}\left(\boldsymbol{x}_0\right)\right] + \left[E_{\boldsymbol{\tau}}\left(\hat{f}\left(\boldsymbol{x}_0\right) - f\left(\boldsymbol{x}_0\right)\right)\right]^2 \\ &= \mathrm{Var}_{\boldsymbol{\tau}}\left[\hat{f}\left(\boldsymbol{x}_0\right)\right] + \mathrm{Bias}^2\left[\hat{f}\left(\boldsymbol{x}_0\right)\right]\end{aligned}$$

$$\text{EPE}(\boldsymbol{x}_0) = \sigma^2 + \text{Var}_{\boldsymbol{\tau}}\left[\hat{f}\left(\boldsymbol{x}_0\right)\right] + \text{Bias}^2\left[\hat{f}\left(\boldsymbol{x}_0\right)\right]$$

Remark:

- The noise and pure stochastic nature of the error (irreducible)
- The variation from estimating our $\hat{f}$ (from limited sample size)
- The bias from estimating our model for f
  - average error from estimation
  - model mis-specification error (intrinsic error from model)

$$\text{Bias}\left[\hat{f}\left(\boldsymbol{x}_0\right)\right] = E_{\boldsymbol{\tau}}\left(\hat{f}\left(\boldsymbol{x}_0\right) - f^*\left(X_0\right)\right) + f^*\left(\boldsymbol{x}_0\right) - f\left(\boldsymbol{x}_0\right)$$

where $f^* = \arg\min_{\tilde{f}\in\mathcal{F}} E(f(X) - \tilde{f}(X))^2$.

Bias-Variance Tradeoff

Bias-Variance Tradeoff

The goal is to achieve small EPE is to minimize both the variance and bias by using some good estimator for f by considering a good class $\mathcal{F}$.

It turns out, you cannot achieve both goals to the best possible extent.

Consider the k-nearest-neighbor regression:

$$\hat{f}(x) = \frac{1}{k} \sum_{i \in N_k(x)} y_i$$

where $N_k(x)$ gives the k nearest neighbors of x.

What are the tradeoffs at either end of k?

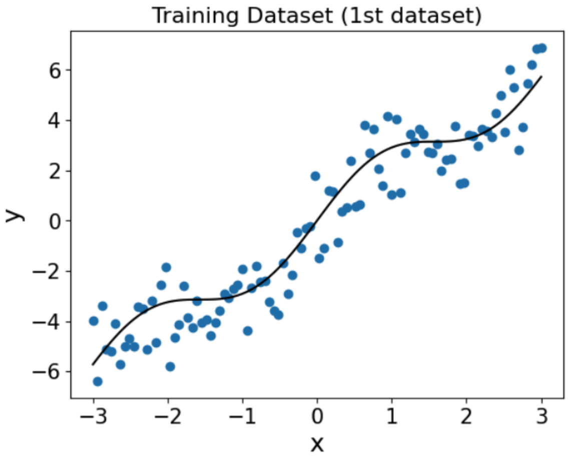
$$EPE(\boldsymbol{x}_0) = \sigma^2 + \operatorname{Var}\left(\frac{1}{k} \sum_{i \in N_k(\boldsymbol{x}_0)} Y_i\right) + \left[f\left(x_0\right) - E\left(\frac{1}{k} \sum_{i \in N_k(\boldsymbol{x}_0)} Y_i\right)\right]^2$$

For large k, we may miss important details, due to averaging over too many y_i values, called **underfitting**.

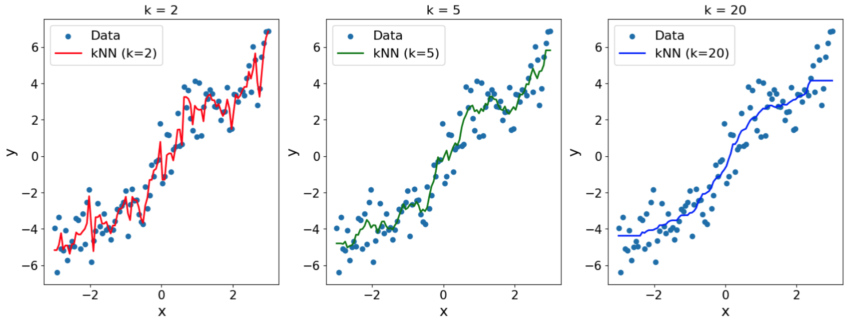
- bad performance on both training and test datasets
- high bias and low variance regime

For small k, we risk picking up noisy features of the sample that don't really have to do with the true regression function $f(\boldsymbol{x})$, called **overfitting**.

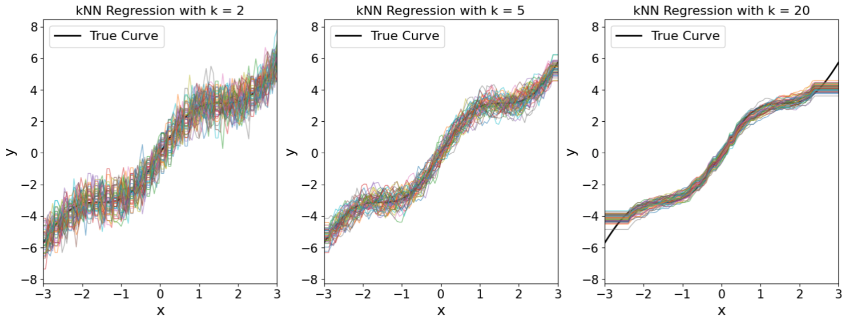
- overly-good performance on training dataset, but worse performance on test dataset
- low bias and high variance regime



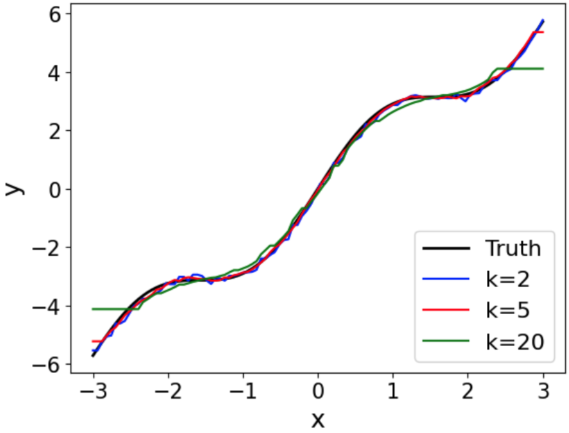
One training dataset and the true function



Fitted curve by KNN regression (one training dataset)



Fitted curves by KNN regression (50 training datasets)



Fitted curves by KNN regression averaged over training datasets

(Overall) expected test error (expected loss/risk) of $\hat{f}$:

$$\text{EPE} = \text{Err} = E_{(Y_0, X_0, \boldsymbol{\tau})}\left[\left(Y_0 - \hat{f}(X_0)\right)^2\right]$$

A good estimate is the **expected test error**

$$E\left[\frac{1}{n'}\sum_{i=1}^{n'}\left(Y_{0i} - \hat{f}(X_{0i})\right)^2\right], \text{ where } (X_{0i}, Y_{0i})_{i=1}^{n'} \text{ is the new sample}$$

The expectation here is taken over both training and test samples.

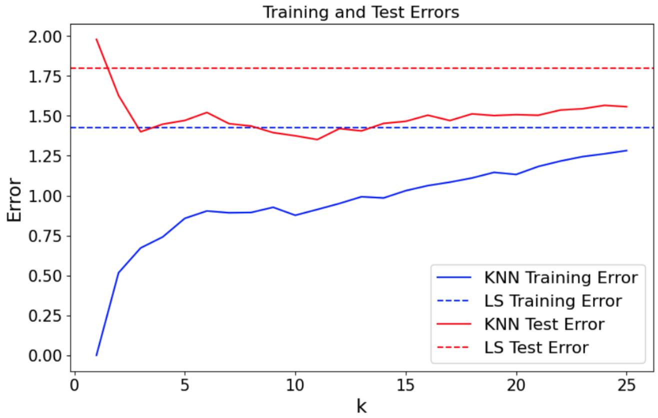
Compared with **expected value of the sample training error**

$$E\left[\frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{f}(X_i)\right)^2\right].$$

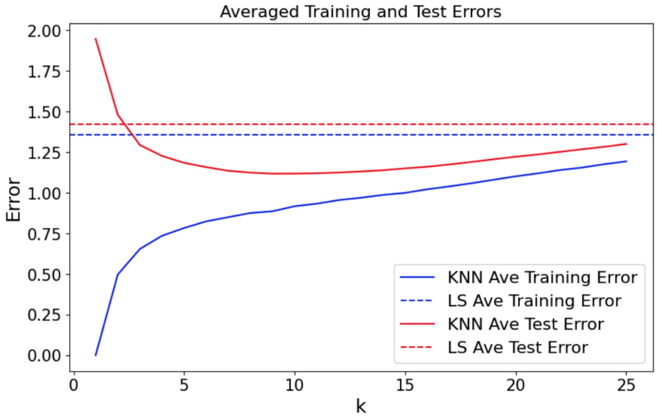
The expectation here is taken over training samples.

It can be shown that expected training error is no larger than expected test error

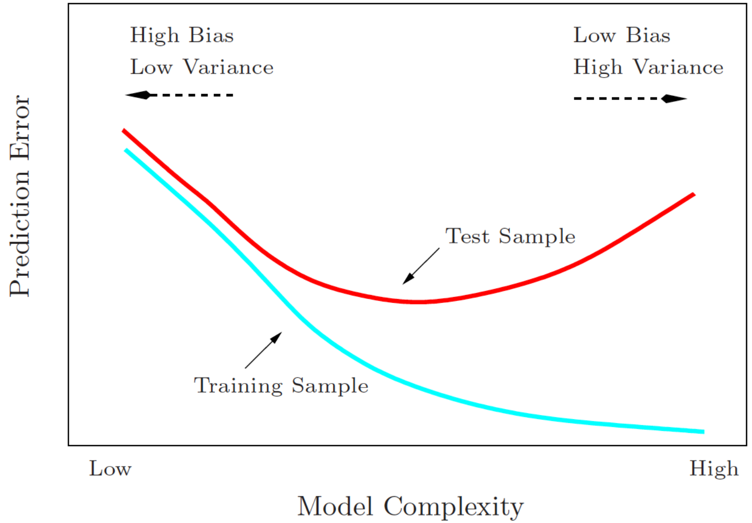
$$E_{\boldsymbol{\tau}}\left[\frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{f}(X_i)\right)^2\right] \leq E_{\boldsymbol{\tau},Y_0X_0}\left[\frac{1}{n'}\sum_{i=1}^{n'}\left(Y_{0i} - \hat{f}(X_{0i})\right)^2\right].$$



Training error and test error (one training data and test data)



Training error and test error averaged over multiple training data and test data


High Bias
Low Variance
Low Bias
High Variance
Prediction Error
Test Sample
Training Sample
Low
High
Model Complexity

ESL: Fig 2.11

Bias-variance tradeoff

As the model complexity increases

- The training error decreases
- The test error first decreases then increases (U-shape)

If the training error is low, then a persistent high gap between test error and training error is an indication of overfitting (dominant variance).

If both training and test errors are high and their gap is small, this is an indication of underfitting (dominant bias).

Double Descent

In the deep learning literature, it has been observed a phenomenon about the test curve–called "double descent".

It is observed and conjectured that after the model complexity surpasses a certain threshold ("interpolating or overparametrized regime"), the variance from estimating over-parametrized models may start to decrease significantly, which may then lower the test curve. This is mostly an empirical finding, and a rigorous theoretical explanation is not yet available.

For most of the models we discuss here, we are in the "classical regime" where we do not attempt to interpolate the data nor attempt to overparametrize the model. The benefit for doing this is not clear in view of the double descent phenomenon.

ref. Belkin et al. PNAS 2019.

Remark

- how to regularize the level of complexity?
  - penalization, pruning…
  - (complex models) early stopping, selective samples (dropout), model architecture, algorithms…
- the bias-variance relation presumes that the training sample size is fixed
  - (for complex problem) larger sample size in general requires more complex model.
  - for a fixed level of complexity, may study how the training sample size affects the performance (**learning curves**)
- the bias-variance decomposition above clearly pertains to the choice of loss function
  - for classification, we shall see that the bias and variance is related differently.

more in future lessons…

Curse of dimensionality

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

we are completely agnostic about f and model f using flexible approximation.

- KNN regression
- polynomial regression with (unknown) increasing degree
- …(more later)

These are examples of so-called **nonparametric regressions** as opposed to **parametric regressions** where the form f and capacity is assumed to be exact.

- advantage: flexibility
- disadvantage: “curse of dimensionality”

More on this later…

Estimation of Test Errors

Validation-set approach

Without tuning:

- randomly divide the available set of samples into two parts: a training set and a validation or hold-out set.
- the model is fit on the training set, and the fitted model is used to predict the outcome for the observations in the validation set.

With tuning: Randomly divide the dataset into three parts:

- training set: to fit the models
- validation (tuning) set: to estimate the prediction error for model selection
- test set: to assess the generalization error of the final chosen model

The typical proportions are respectively: $50\%, 25\%, 25\%$.

Cross-validation

Hold-out each point (y_i, x_i) from our training set in turn, fit $\hat{f}^{-(i)}$ on all points except this one (i.e., $(y_j, x_j), j \neq i$), record the squared error on (x_i, y_i) and average the results. This yields the so-called **leave-one-out cross validation** (LOOCV) estimate of test error

$$CV(\hat{f}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}^{-(i)}(x_i))^2$$

where $\hat{f}^{-(i)}$ is fitted using all training data except the i-th observation.

More generally, we can split up our training set into K divisions or folds, for some number K: K**-fold validation**.

- Let the K parts be $C_1, C_2, \dots C_K$, where C_k denotes the indices of the observations in part k. There are n_k observations in part k : if n is a multiple of K, then $n_k = n/K$
- Compute

$$CV(\hat{f}) = \frac{1}{K}\sum_{k=1}^{K} \mathrm{CV}_k$$

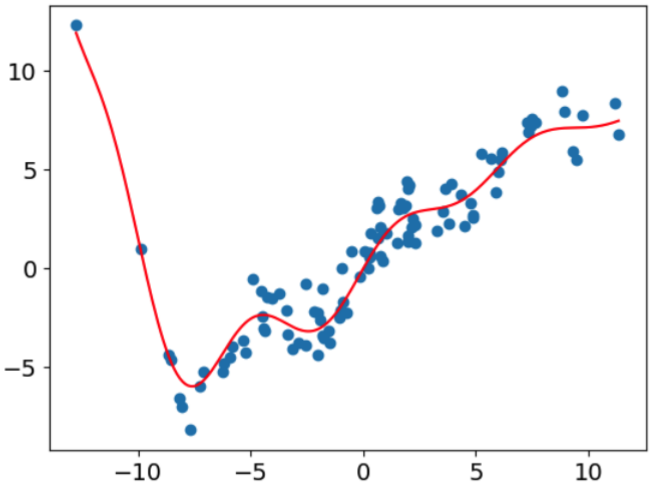
  where $\mathrm{CV}_k := \mathrm{CV}_k(\hat{f}^{-(k)}) = \sum_{i \in C_k}(y_i - \hat{f}^{-(k)}(x_i))^2/n_k$, and $\hat{f}^{-(k)}(x_i)$ is the fit for observation i, obtained from the data with part k removed.

Common choice of K: $K = 5$ or $K = 10$, or $K = n$.

Remark

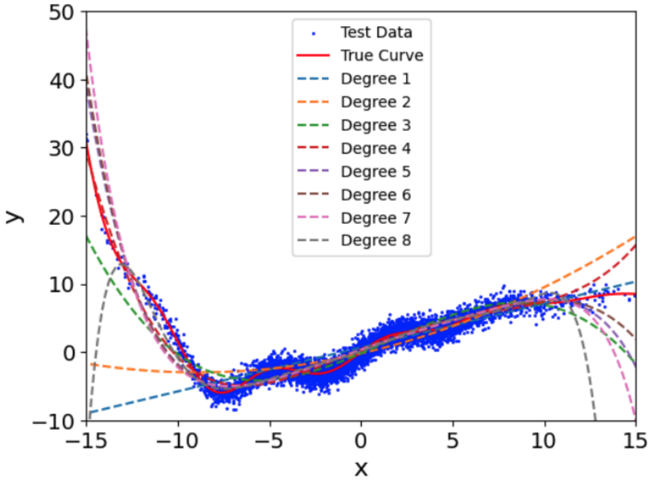
- If each training set only uses a part of the sample, it tends to overestimate test error. Compared with validation approach and general K-fold CV, LOOCV has less bias.
- However, LOOCV has higher variance than does K-fold CV ($K < n$), because the n training sets are so similar to one another.
- Note also LOOCV does not involve random splitting. One may further average K-fold CV.

Example: polynomial regression



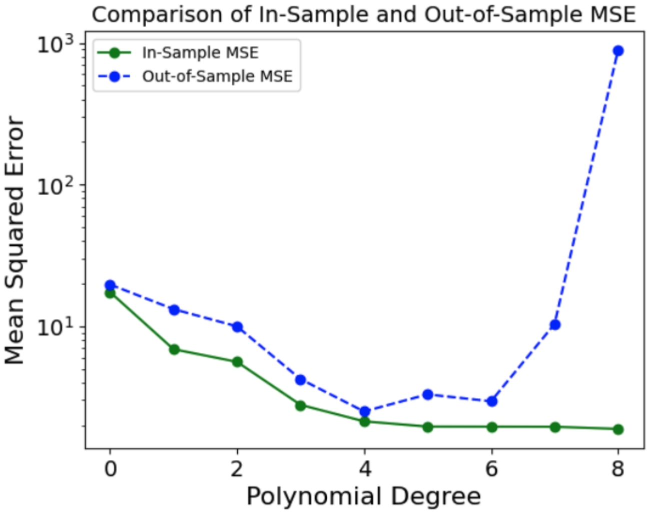
The (training) dataset and the true curve

Example: polynomial regression

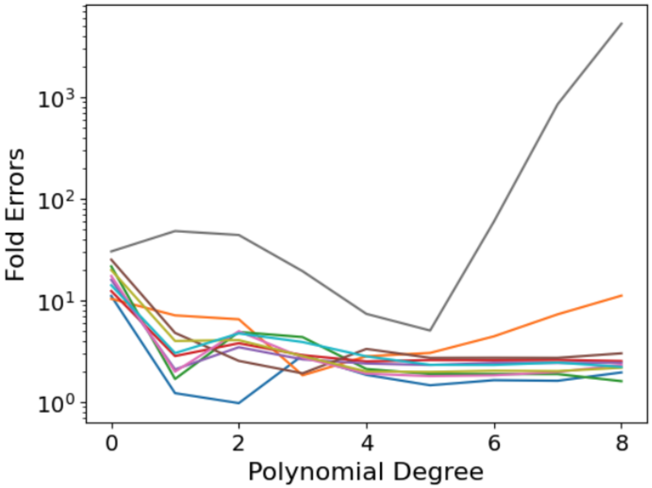


Fitted polynomial curves based on training data, and test data (blue dots)

Example: polynomial regression

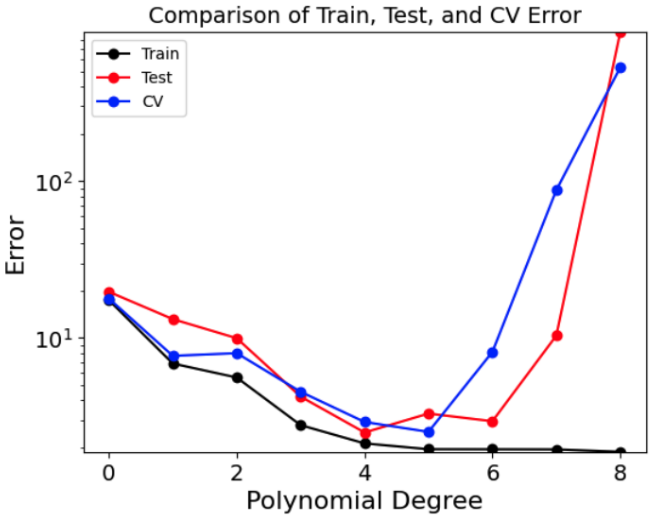


Example: polynomial regression



10 folds CV estimates

Example: polynomial regression



Comparison of training error, test error and CV error (estimate)

CV standard error

$$\mathrm{Var}(\mathrm{CV}(\hat{f})) = \mathrm{Var}\left(\frac{1}{K}\sum_{k=1}^{K}\mathrm{CV}_k\right) \approx \frac{1}{K}\,\mathrm{Var}\left(\mathrm{CV}_1\right)$$

This approximation is valid for small K(e.g., , $K = 5$ or 10) but not really for big K (e.g., $K = n$).

For small K(e.g., $K = 5$ or 10), the sample variance of the cross-validation error estimate is

$$\frac{1}{K}\,\mathrm{S}^2\left\{\mathrm{CV}_1, \ldots \mathrm{CV}_K\right\}$$

Choose the optimal α

Suppose $\hat{f} = \hat{f}_\alpha$ depends on some tuning parameter α.

To find the best parameter $\hat{\alpha}$ by minimizing $CV(\hat{f}_\alpha)$:

let $\alpha \in \{\alpha_1, \ldots, \alpha_m\}$, within a set of candidate values for α, obtain all cross validation errors

$$CV(\hat{f}_\alpha) = \frac{1}{K} \sum_{k=1}^{K} CV_k(\hat{f}_\alpha^{-(k)})$$

Choose

$$\hat{\alpha} = \arg \min_{\alpha \in \{\alpha_1, \ldots, \alpha_m\}} CV(\hat{f}_\alpha)$$

one-standard error rule

Occam's Razor suggests that among competing models, the simplest model should be chosen:

- generalization
- parsimony
- efficiency

With standard errors of $CV(\hat{f}_\alpha)$:

$$SE(\alpha) = \sqrt{\mathrm{Var}(CV(\hat{f}_\alpha))},$$

choose the α among $\{\alpha : CV(\hat{f}_\alpha) \leq CV(\hat{f}_{\hat{\alpha}}) + SE(\hat{\alpha})\}$ that gives the most parsimonious model (less complex model).

Performance of $\hat{f}_{\hat{\alpha}}$?

Suppose we are interested in the generalization error of the tuned model $\hat{f}_{\hat{\alpha}}$ where $\hat{\alpha}$ is chosen by CV.

To estimate this error, a naive estimate is

$$\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}_{\hat{\alpha}}^{-\kappa(i)}(x_i)\right)^2,$$

$\kappa(\cdot) : \{1, \ldots, n\} \to \{1, \ldots, K\}$ index function.

Correct approaches (later on):

- honest cross validation (nested cross validation)
- 0.632 and 0.632+estimator (bootstrap approach)

The correct way to use CV

Cross-validation must be applied to the entire sequence of modeling steps. Apply CV to the outermost loop to avoid the leakage of information about the validation/test data into the training process (overfitting).

- samples must be "left out" before any selection or filtering steps are applied.
- exception: perform unsupervised tools that do not involve using information about the response variables.

e.g.:

Outer Loop (CV): split the dataset into training and validation sets

- For each fold:
  - feature selection + model training: on the training set of each CV split, perform feature selection and estimation (possibly involving another CV loop).
  - validation: evaluate the model on the validation set of the CV split.
- Average the validation performance over all folds

The Implication for the Linear Model

The Implication for the Linear Model

If *true* relationship is approximately linear, then Ordinary Least Squares has low bias. But for variance:

- When n $\gg p$, OLS tends to have low variance, and hence perform well on test observations.
- When $n < p$, then there is no longer OLS estimate, and the variance of these estimates are infinite. May be able to increase bias slightly, but decrease variance substantially via
  - regularization/shrinkage
  - features selection
  - dimension reduction
- When n is not much larger than p, then the least squares fit can have high variance and may result in over fitting and poor prediction on future observations.

The **key message**: It is possible to trade a little bias with the large reduction in variance, thus achieving higher prediction accuracy.

Model interpretability

Even it is true that $Y_i = \beta^T X_i + \epsilon_i$ where $E[\epsilon_i \mid X_i] = 0$ and $\mathrm{Var}(\epsilon_i \mid X_i) = \sigma^2$,

- When we have a large number of variables X in the model there will generally be many that have little or no effect on Y
- Leaving these variables in the model makes it harder to see the "big picture", i.e., the effect of the "important variables"
- The model would be easier to interpret by removing (i.e. setting the coefficients to zero) the unimportant variables. This can be accomplished via variable selection, or feature selection.

Caveats

- All of the standard theory of statistical inference you have learned so far presumes that you have a model which was fixed in advance of seeing the data.
- If you use the data to select the model, and conduct estimation and inference etc., the standard theory becomes invalid, and it will no longer give you correct p-values for hypothesis tests, confidence sets for parameters, etc.
- A more savvy approach is to take into account the model-selection procedure, to adjust for the hypothesis testing, confidence interval: **post-selection inference**.