Chapter 1 – Predictive Modelling

Fabien Petit

Department of Economics, University of Barcelona
Centre for Education Policy and Equalising Opportunities, UCL

Predictive Modelling

This lecture introduces (supervised) machine learning models and their connection with traditional econometrics

- We provide a conceptual framework to understand the differences and similarities between the two approaches
- There is a substantial overlap between the two disciplines, as both are grounded in statistics
- The views of the two disciplines are voluntarily contrasted and do not reflect the diversity of methods

Function Approximation

Function Approximation

Consider an unknown function of interest F that maps some input array x_i to a response y_i given some error ϵ_i

$$y_i = F(x_i) + \epsilon_i \quad (1)$$
$$i = 1, \ldots, N$$

where N is the number of observable population (x_i, y_i) pairs generated by the function

- Many tasks can be formulated this way (e.g., estimating one aspect or the entire data-generating process)
- The input may be pixel intensities in an image or characters in a text mapped to a probability

Function Approximation

To illustrate the different approaches, let's simulate a data-generating process or target function

Target Function

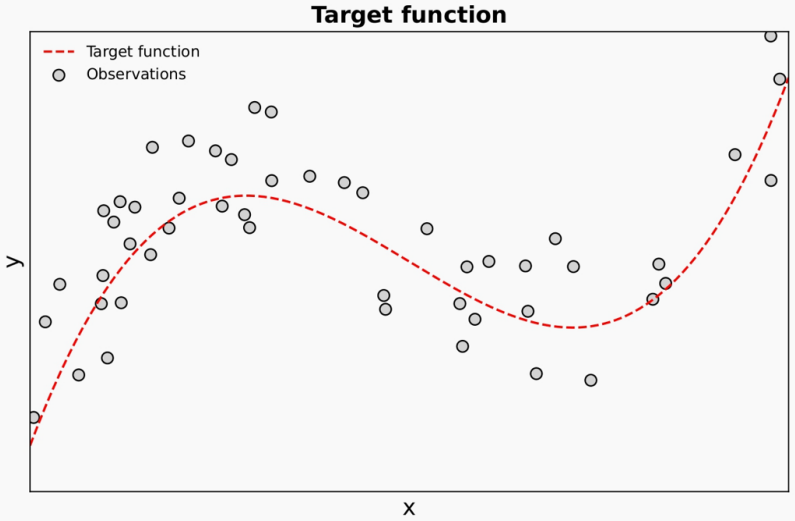
```
import numpy as np
from numpy import random

# Function
def f(x): return x**3 - x * 5

# Data
random.seed(0)
n = 50
s = np.random.choice([True, False], n, p=[.5, .5])
x = random.uniform(-3, 3, n)
e = random.normal(0., 3, n)
y = f(x) + e
```

In practice, we do not know the target function; otherwise, statistical modelling would not be required

Function Approximation



Function Approximation

In practice, there are two reasons why we may want to compute an estimate $\widehat{F}$ for this function

1. To interpret the relationship between x_i and $y_i \rightarrow$ focus on estimated parameters $\widehat{\beta}$ and **explanatory power**
2. To predict accurately y_i using $x_i \rightarrow$ focus on estimated response $\widehat{y}_i$ and **predictive power** (out-of-sample)

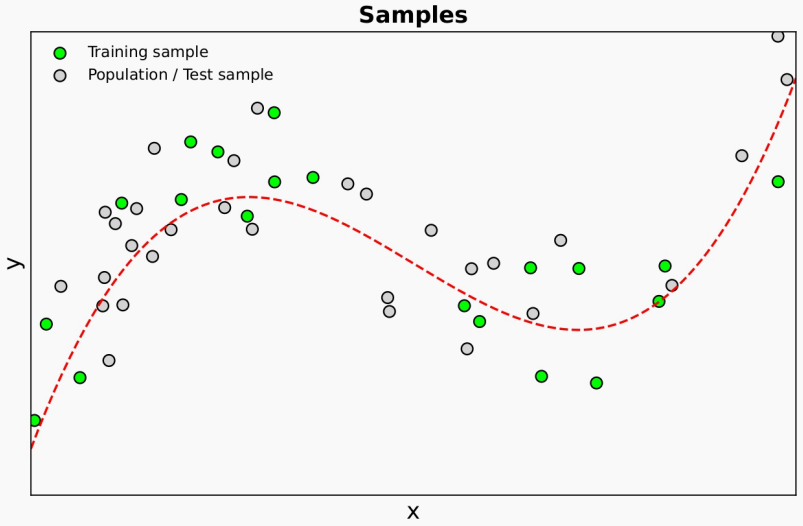
Parametric econometrics is used for the former, while the latter is best solved using supervised learning

Function Approximation

Regardless of the application, we only observe a random sample representative of the population

- A random sample means that every observation has an equal probability of being sampled
- This implies that the sample's characteristics are distributed similarly to those of the population
- Unrepresentative or insufficient samples cause models to produce biased and/or inefficient estimates

Function Approximation



Parameter Estimation

Parameter Estimation

There is an infinity of functions passing through the observed data points and function approximation is unsolvable

$$y_i = f(x_i, \beta) + \epsilon_i \qquad (2)$$
$$i = 1, \ldots, n$$

where f is an empirical model, β a set of parameters and n the number of observed sample (x_i, y_i) pairs

- We impose constraints on the problem by restricting the search space to a parametric family of functions
- The model structure encodes those constraints and depends on the application and prior knowledge

Parameter Estimation

Empirical Model

```
import pasty
from statsmodels import api
class Spline:
def ___init__(self, d:int, df:int):
    self.d = d
    self.df = df
    self.model = None
def transform(self, x:np.ndarray, d:int, df:int) -> np.ndarray:
    X = patsy.dmatrix(f'bs(x, degree={d}, df={df})', {'x': x})
    return X
def fit(self, x:np.ndarray, y:np.ndarray) -> None:
    X = self.transform(x, d=self.d, df=self.df)
    self.model = api.GLM(y, X).fit()
def predict(self, x:np.ndarray) -> np.ndarray:
    X = self.transform(x, d=self.d, df=self.df)
    yh = self.model.predict(X)
    return yh
```

Parameter Estimation

$$\widehat{\beta} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \mathcal{L}\left(y_i, f\left(x_i, \beta\right)\right) \tag{3}$$
$$i = 1, \ldots, n$$

where $\mathcal{L}$ is a loss function, a measure of distance between the observed and the estimated response

- The form of the loss function depends on the distribution of the response (e.g., continuous, categorical)
- The optimization may have a closed-form solution or require iterative routines (e.g., gradient-based)

Inference

Inference

To compute interpretable parameter estimates, the model is often additive and linear in the parameters

- The additivity allows parameters to be interpreted separately (i.e., partialling out, ceteris paribus)
- The linearity ensures that parameters have a simple interpretation (e.g., average unit-increase)
- Transformations are restricted to those producing interpretable parameters (e.g., logs, interactions)

Inference

Importantly, the model is derived from either theory or intuition and requires strong prior assumptions

- The researcher defines what inputs enter the model and their functional relationship with the response
- Non-linearities and interactions are defined by transforming the inputs before estimating the model
- This approach is sensible for low-dimensional problems with a well-established theoretical background

Inference

Under restrictive conditions on the form of F and the conditional distribution of the population error

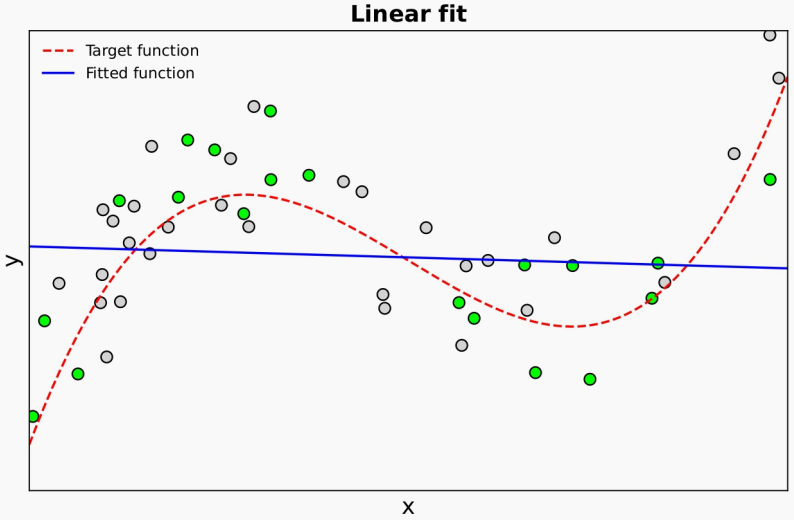
- Parametric econometric models deliver interpretable and meaningful average parameter estimates
- Statistical inference provides confidence intervals to establish generality beyond the sample used
- The optimization has a closed-form solution or can be solved efficiently (e.g, Newton-Raphson)

Inference

Linear models are usually not suited for predictions, except when the target function is linear

- Focus on unbiasedness (under linearity) and does not trade off bias and variance (more on this)
- When the target function is non-linear, a linear model provides stable but biased predictions
- Cannot approximate complex functions, high-dimensional with unknown non-linearities and interactions

Inference



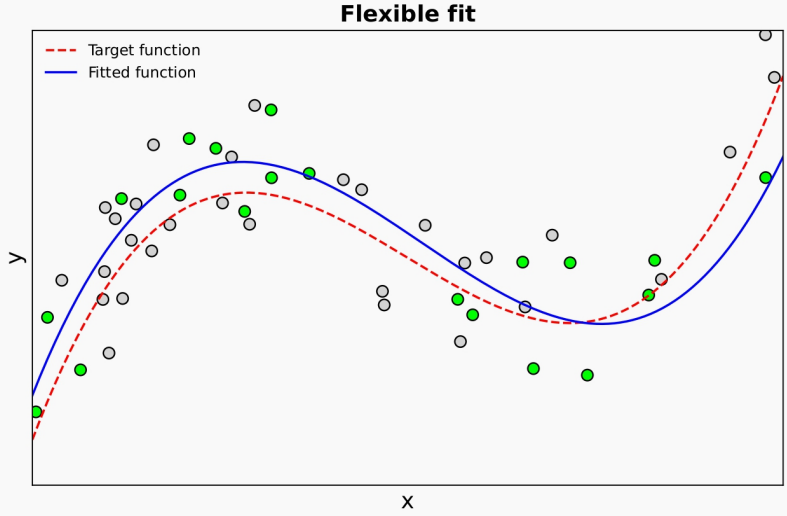
Prediction

Prediction

For predictive tasks, parameter interpretability is not required and more flexible methods become available

- Supervised learning models make many fewer assumptions about the target function
- The models use the sample data and numerous parameters to learn about their shape
- They uncover key non-linearities and interactions that were not defined in advance (i.e., functional form)

Prediction



Prediction – Motivating Example

- We predict the probability that aerial images contain a house (e.g., logistic)
- An image is represented as a multi-dimensional array of pixel intensities
- This observation contains $1024 \times 1024 \times 3 = 3.14$ million "variables"

Prediction

The target function is high-dimensional and contains numerous unknown interactions and non-linearities

- Interactions: objects are depicted by particular spatial patterns of pixels with specific intensities
- Non-linearities: e.g., no simple mapping between the color intensities and the probability
- We have no idea about this function and cannot possibly devise a sensible parametric model

Prediction

A more sensible approach is to build a model directly from the sample data using supervised learning

- Other models naturally handle the high dimensions and uncover non-linearities and interactions
- The estimated function should be general enough to predict accurately out-of-sample observations
- When the estimated model approximates the target function with accuracy, it "learns" the function

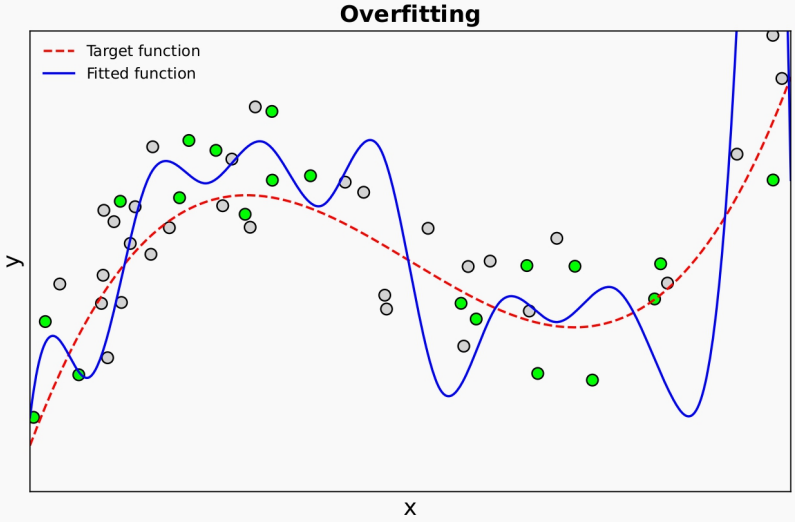
Regularization

Regularization

For predictive applications, the model must be assessed using observations **outside the (training) sample**

- Flexible methods approximate not only the target function but also the observational error (i.e., overfitting)
- The **training sample** loss provides a biased estimate of predictive performance (i.e., tends toward 0)
- The model is assessed on another random sample that has not been used for training, that is, the **test sample**

Regularization



Regularization

During optimisation, we minimise the training sample error, but we really care about the **test sample error**

- The empirical model must be flexible enough to approximate complex functions (i.e., low bias)
- However, the estimated function must also generalize to out-of-sample observations (i.e., low variance)
- This **fundamental trade-off** is addressed by adding a regularization term to the loss function

Regularization

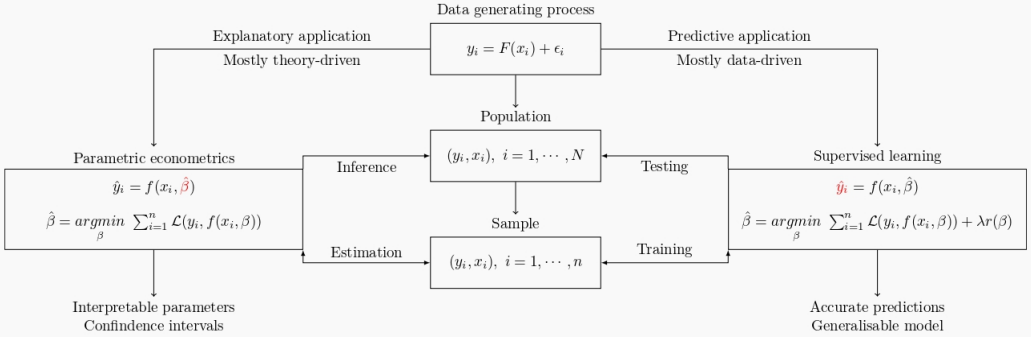
A regularization term is added to the loss to encourage more general models. Equation (3) becomes

$$\widehat{\beta} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \mathcal{L}\left(y_i, f\left(x_i, \beta\right)\right) + \lambda r(\beta) \tag{4}$$
$$i = 1, \ldots, n$$

where r is a regularization function that increases with large parameter values and λ is the regularization parameter

- Large parameter values allow the estimated function to change its behavior over a small space
- The relative strength of the two counteracting effects is governed by the tuning parameter λ

Regularization – Inference and Predictive Methods



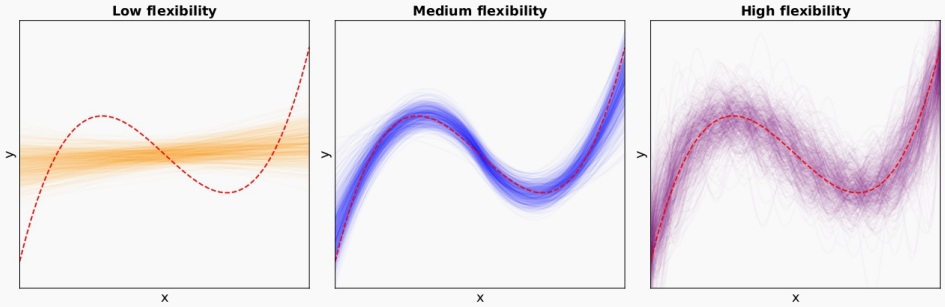
Bias and Variance

Bias and Variance

To produce accurate predictions, our model must minimize **simultaneously** prediction bias and variance

- These statistics refer to the distribution of $\widehat{y}$ when the same model is estimated on different random samples
- Estimators can be considered as random variables with the estimated response being their realization
- For predictions, we consider the integrated effect of bias and variance over the entire function space

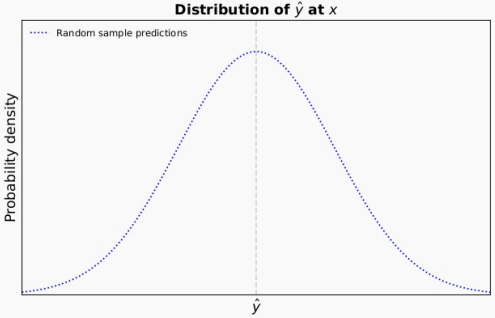
Bias and Variance – Bootstrapped Predictions for Different Estimators



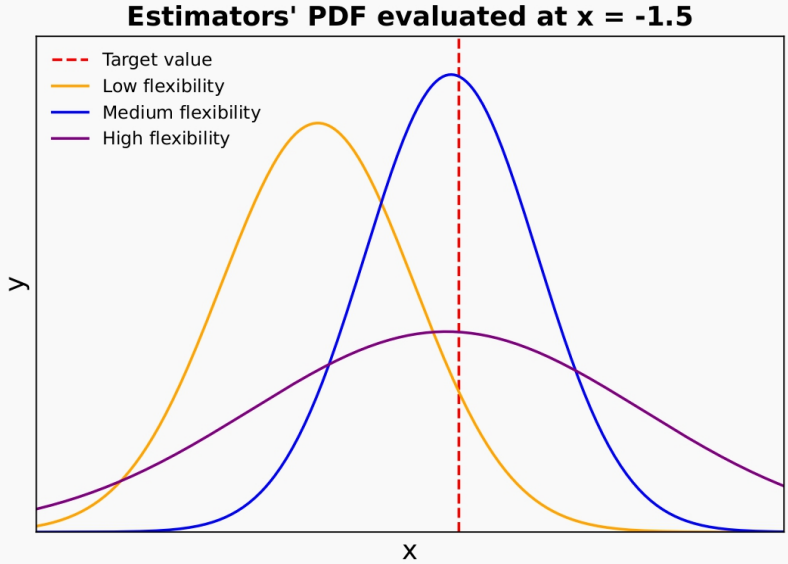
- **Bias** is the tendency of an estimator to systematically produce an overestimate or an underestimate
- **Variance** is the amount by which the estimates change when fitting the model on another random sample

Bias and Variance

- The shape of the probability density function gives important information about the estimator's properties
- The first moment measures the central tendency (**bias**)
- The second moment measures the dispersion (**variance**)
- Integrating over a range of the PDF provides the probability that the estimate falls within that range



Bias and Variance



Bias and Variance

To derive formally the **bias-variance trade-off**, consider the mean squared error loss function

$$\mathcal{L}_{mse}\left(y_i, \widehat{y}_i\right) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - f\left(x_i, \widehat{\beta}\right)\right)^2 \tag{5}$$

In predictive applications, the error is typically expressed at the observation level, rather than the sample level

- The squared term ensures that positive and negative estimated errors do not cancel out
- Observations whose predicted response lies far from the true response are given more weight

Bias and Variance

The expected training mean-squared error can be expressed as the sum of **reducible** and **irreducible error**

$$E\left[(y-\widehat{f})^2\right] = \underbrace{(f-\widehat{f})^2}_{\text{reducible error}} + \underbrace{\text{Var}[\epsilon]}_{\text{irreducible error}} \tag{6}$$

where $f = F(x_i), \widehat{f} = f\left(x_i, \widehat{\beta}\right)$, and the subscript i is dropped to simplify notation

- We can only minimize **reducible error**, the distance between the target and the estimated function
- Even with a perfect estimate for F, the training error is non-zero because ϵ_i cannot be predicted using x_i

Bias and Variance

Note that f is constant and $\widehat{f}$ is assumed to be fixed. If the error is truly random $E(\epsilon) = 0$

$$
\begin{aligned}
E\left[(y - \widehat{f})^2\right] &= E\left[(f + \epsilon - \widehat{f})^2\right] \\
&= E\left[(f - \widehat{f} + \epsilon)^2\right] \\
&= E\left[(f - \widehat{f})^2 + 2\epsilon(f - \widehat{f}) + \epsilon^2\right] \\
&= E\left[(f - \widehat{f})^2\right] + E[2\epsilon(f - \widehat{f})] + E\left[\epsilon^2\right] \\
&= E\left[(f - \widehat{f})^2\right] + 2\underline{E[\epsilon]}E[f - \widehat{f}] + E\left[\epsilon^2\right] \\
E\left[(y - \widehat{f})^2\right] &= \underbrace{(f - \widehat{f})^2}_{\text{reducible error}} + \underbrace{\text{Var}[\epsilon]}_{\text{irreducible error}}
\end{aligned}
$$

Bias and Variance

The **reducible error** in Equation (6) can be further decomposed as the sum of bias (squared) and variance

$$E\left[(y-\hat{f})^2\right] = \underbrace{(E[\hat{f}]-f)^2}_{\text{Bias}^2} + \underbrace{E\left[(\hat{f}-E[\hat{f}])^2\right]}_{\text{Variance}} + \text{Var}[\epsilon] \tag{7}$$

- Minimizing the test sample error implies minimizing both the bias and variance on the training sample
- Increased regularization can be seen as reducing the estimator's variance at the cost of introducing bias

Bias and Variance

$$
\begin{aligned}
E\left[(y-\hat{f})^2\right] &= E\left[(f+\epsilon-\hat{f})^2\right] \\
&= E\left[(f+\epsilon-\hat{f}+E[\hat{f}]-E[\hat{f}])^2\right] \\
&= E\left[(f-E[\hat{f}])+\epsilon+(E[\hat{f}]-\hat{f})^2\right] \\
&= E\left[(f-E[\hat{f}])^2\right]+E\left(\epsilon^2\right)+E\left[(E[\hat{f}]-\hat{f})^2\right] \\
&+ 2E[(f-E[\hat{f}])\epsilon]+2E[\epsilon(E[\hat{f}]-\hat{f})] \\
&+ 2E[(f-E[\hat{f}])(E[\hat{f}]-\hat{f})]
\end{aligned}
$$

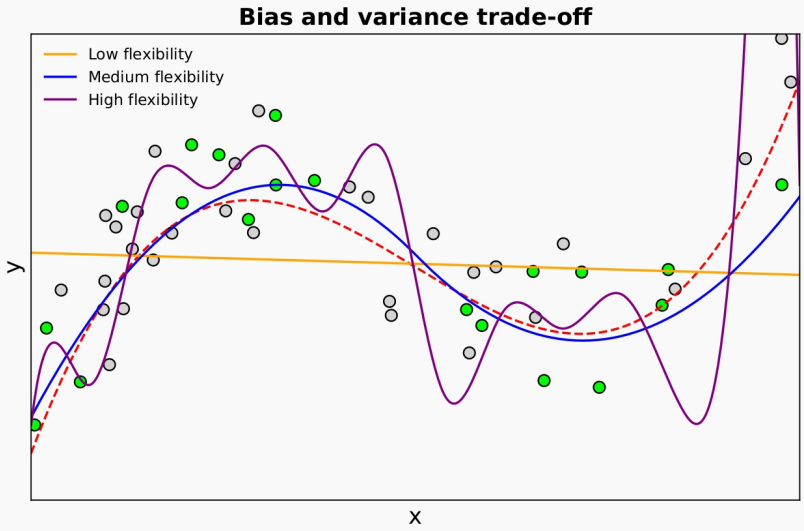
Bias and Variance

$$
\begin{aligned}
&= (f - E[\hat{f}])^2 + E\left(\epsilon^2\right) + E\left[(E[\hat{f}] - \hat{f})^2\right] \\
&+ 2(f - E[\hat{f}])\underline{E[\epsilon]} \\
&+ 2\underline{E[\epsilon]}E[E[\hat{f}] - \hat{f}] \\
&+ 2E[(f - E[\hat{f}])\underline{(E[\hat{f}] - \hat{f})}] \\
E\left[(y - \hat{f})^2\right] &= \text{Bias}^2[\hat{f}] + \text{Var}[\hat{f}] + \text{Var}[\epsilon]
\end{aligned}
$$

Note that f is constant and $\hat{f}$ is assumed to be fixed so $E(\hat{f} - E(\hat{f})) = 0$. If the error is truly random $E(\epsilon) = 0$

Bias-Variance Trade-off

Bias-Variance Trade-off



Bias-Variance Trade-off

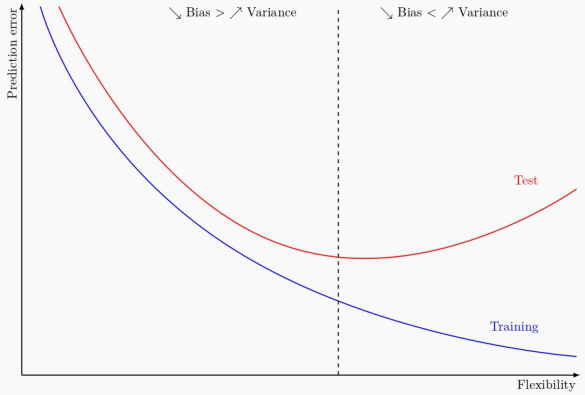
As we add **flexibility** to the model (i.e., more parameters), the **bias decreases** and the **variance increases**

Table: Bootstrapped Bias and Variance

Flexibility	MSE	Bias^2	Var.	Δ MSE	Δ Bias^2	Δ Var.
Low	18.10	16.87	1.23			
Medium	2.53	0.90	1.63	-15.57	-15.97	0.40
High	6.87	0.52	6.36	4.34	-0.38	4.73

The challenge of regularization is to find the model that strikes the right balance between bias and variance

Bias-Variance Trade-off – Training and Test Sample Error

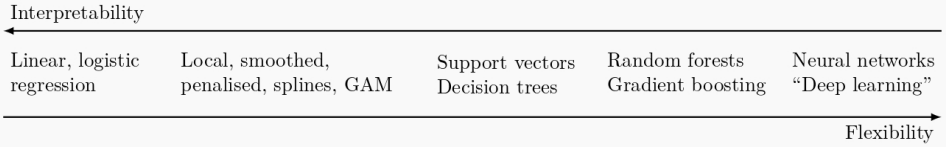


Before the dashed line, the bias decreases faster than the variance increases, and the test error decreases. After that point, additional flexibility has little impact on the bias but increases variance, and the error increases

Bias-Variance Trade-off

Predictive models can be understood as striking a different balance between interpretability and flexibility

- Linear models are interpretable but lack the flexibility to approximate complex non-linear functions
- Flexibility means that a model with different sets of parameters can produce similar predictions



Re-Sampling

Re-Sampling

The statistical properties (e.g., finite sample, asymptotic) of flexible models can often not be derived analytically

- We can repeatedly draw subsamples from a larger sample (proxy for the population) and compute statistics
- The distribution of these statistics (e.g., predictions) gives information about the model's statistical properties
- Useful techniques include cross-validation to compare models and bootstrap to discover their properties

Re-Sampling – Cross-Validation

A predictive model is estimated on a training sample and assessed on a test sample

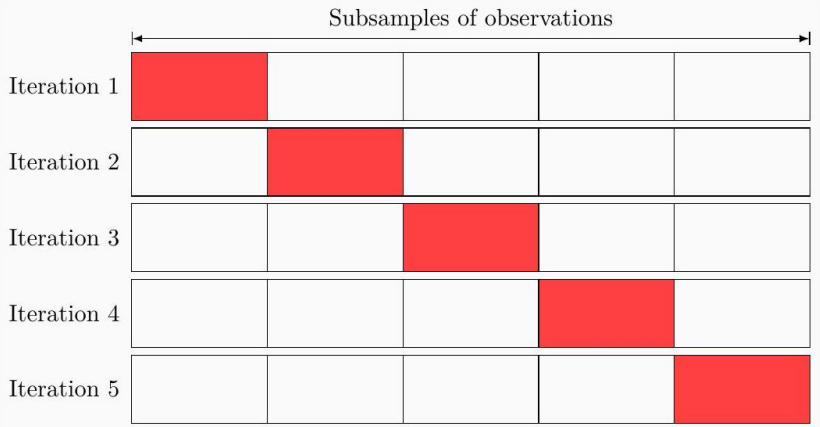
- There is a trade-off with the partition of the observed data into the training and the test sample
- The best model is estimated using all the observed data, but the performance estimate would be biased
- An unbiased estimate requires a sufficiently large test sample, which could be used to estimate a better model

Re-Sampling – Cross-Validation

Cross-validation (Stone 1974) is a resampling procedure that addresses both these issues simultaneously

- The observations are randomly partitioned into k distinct groups without replacement
- The model is repeatedly estimated on $k-1$ partitions and evaluated on the remaining partition
- There are k iterations, so that each observation is used $k-1$ times for training and once for testing

Re-Sampling – Cross-Validation



Re-Sampling – Cross-Validation

The cross-validated mean squared error is calculated by averaging the MSE on the k test partitions

$$\mathcal{L}_{mse}^{cv}(k) = \frac{1}{k} \sum_{1}^{k} \left[\frac{1}{n_t} \sum_{i=1}^{n_t} \left(y_i - \hat{f}\left(x_i\right) \right)^2 \right],$$

where $i = 1, \ldots, n_t$ are the observations of the test fold

- The value of k is a trade-off between the computational cost and the bias of the estimator
- "Leave-one-out" cross-validation involves k folds. $k = \{5, 10\}$ provide reasonably good estimates

Re-Sampling

K-Fold Cross Validation

```
model = Spline(d=3, df=3)
k     = 10

# Splitting folds
folds = np.split(random.permutation(n), k)
folds = [~np.isin(np.arange(n), fold) for fold in folds]
eh_cv = np.zeros(k)

# K-fold CV
for i, s in enumerate(folds) :
    model.fit(x[s], y[s])
    eh_cv[i] = np.mean((y[~s] - model.predict(x[~s]))**2)
np.mean(eh_cv)
```

Re-Sampling – Bootstrap

Bootstrap (Efron and Tibshirani 1994) is used to estimate standard errors for the estimated parameters and response

- Non-parametric methods do not provide naturally standard errors for the estimates
- We want to draw additional samples from the population, yet we have only a single one
- An estimate can be computed by sampling repeatedly with replacement (to ensure independence across samples) observations from that sample

Re-Sampling – Bootstrap

For instance, standard error of an estimated parameter $\hat{\beta}$ can be estimated via bootstrap using

$$se(\hat{\beta}) = \sqrt{\frac{\sum_{b=1}^{B} \left(\hat{\beta}_b - \overline{\hat{\beta}}\right)^2}{B-1}}$$

$$\overline{\hat{\beta}} = \sum_{b=1}^{B} \frac{\hat{\beta}_b}{B}$$

where B is the number of bootstraps

- There is no agreed-upon convention concerning the size of the subsample and the number of bootstraps
- The bias vs. computation trade-off also applies to bootstrap methods (more is better)

Re-Sampling – Bootstrap

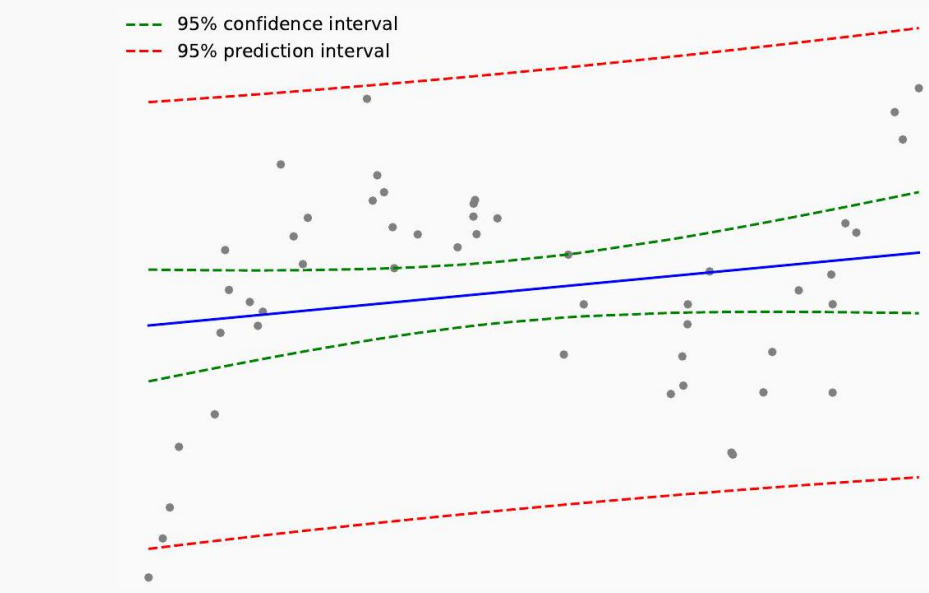
Bootstrap

```
nb = 100
ns = 40
samples = [random.choice(np.arange(ns), ns) for _ in range(nb)]

bh_bs = np.zeros((nb, 2))
for i, sample in enumerate(samples):
    bh_bs[i] = model.fit(x[sample], y[sample])

se_bs = np.sum((bh_bs - np.mean(bh_bs, axis=0))**2, axis=0)
se_bs = np.sqrt(se_bs / (nb-1))
se_bs.round(4)
# array([0.7249, 0.4965])
```

Bootstrap – Confidence and Prediction Intervals



Summary

Summary

Supervised learning and econometrics can be seen as function approximation methods. You may use the former when

- Prediction accuracy is more important than the interpretability of estimated parameters
- No functional form is suggested by theory, and no sensible parametric model can be written
- The target function is non-linear, potentially high-dimensional, and there are many observations

Summary

Every statistical model can be seen as striking a different balance between prediction bias and variance

- This trade-off can also be interpreted as a compromise between flexibility and interpretability
- The model choice depends on the research problem and the nature of the target function
- Resampling methods provide additional information about the statistical properties of a model

References

- Stone, Mervyn (1974). "Cross-validatory choice and assessment of statistical predictions". *Journal of the Royal Statistical Society* 36.2, pp. 111-147 (cit. on p. 51).
- Harrison, David J. and Daniel L. Rubinfeld (1978). "Hedonic housing prices and the demand for clean air". *Journal of Environmental Economics and Management* 5.1, pp. 81-102 (cit. on p. 68).
- Efron, Bradley and Robert Tibshirani (1994). An introduction to the bootstrap. CRC Press (cit. on p. 55).
- Breiman, Leo (2001). "Statistical modeling: The two cultures". In: Statistical Science 16.3, pp. 199-231.
- Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). The elements of statistical learning. Springer.
- Mullainathan, Sendhil and Jann Spiess (2017). "Machine learning: An applied econometric approach". *Journal of Economic Perspective* 31.2, pp. 87-106.

Appendix – Random Variables

Estimators can be considered as random variables with the estimated response being their realization

- A random variable is a series of realizations (i.e., estimates) that take real values with a probability
- With a sample, we observe one realization, one value out of the set of possible values (i.e., random samples)
- These values and the associated probabilities are represented as a probability density function

Appendix

Various predictive models for a hedonic price model on the Boston dataset (Harrison and Rubinfeld 1978)

Model	R^2 train	R^2 test
Least-squares	74.14	72.20
GAM	87.63	83.41
Random forest	97.76	88.26
Neural networks	98.86	86.36