

Lecture 5

Model Selection

.

Outline

- ▶ Model Selection
- ▶ Occam's Razor
 - ▶ Quantifying Model Complexity
- ▶ Popper's Prediction Strength
 - ▶ Cross-Validation
- ▶ Dataset Bias and The 'Clever Hans' Effect
- ▶ Bias-Variance Analysis

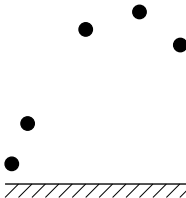
Learning a Model of the Data

Assuming some observations from some unknown true data distribution $p(x, y)$, we would like to find a model θ such that some distance $D(p, p_\theta)$ is minimized. For regression tasks, we can consider the simpler objective:

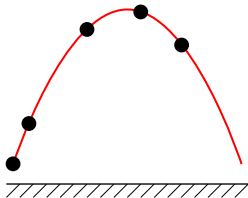
$$\min_{\theta} \int (\mathbb{E}_{\hat{p}}[y|x] - f_{\theta}(x))^2 \cdot \hat{p}(x) \cdot dx$$

where \hat{p} is some guess of the true p , e.g. the empirical distribution.

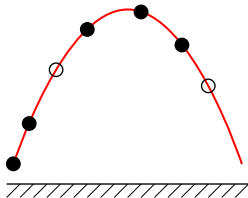
1. observations



2. fit a model



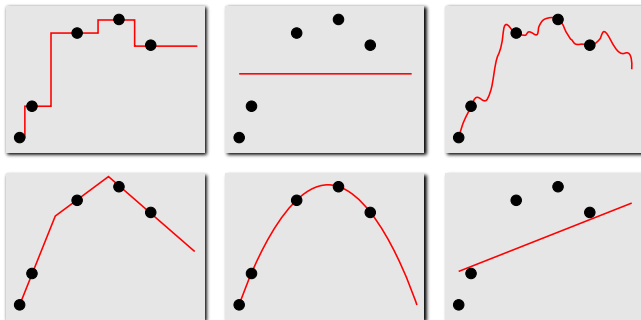
3. make predictions



Model Selection

Questions:

1. Among models that correctly predict the data, which one should be retained?
2. Should we always choose a model that perfectly fits the data?



Occam's Razor

William of Ockham (1287–1347)

“Entia non sunt multiplicanda praeter necessitatem”

English translation

“Entities must not be multiplied beyond necessity.”



Interpreting Occam's Razor

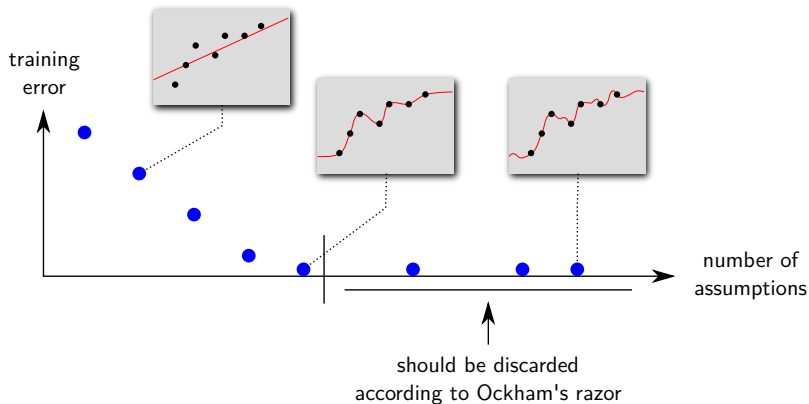
What *advantages* do we expect from a model based on *few assumptions*?

- ▶ If two models correctly predict the data, the one that makes fewer assumptions should be preferred because simplicity is *desirable in itself*.
- ▶ If two models correctly predict the data, the one that makes fewer assumptions should be preferred because it is likely to have *lower generalization error*.

Further reading:

Domingos (1998) Occam's two Razors: The Sharp and the Blunt.

Occam's Razor for Model Selection



How to Quantify “Few Assumptions”?

Many possibilities:

- ▶ **Number of free parameters of the model**
- ▶ Minimum description length (MDL)
- ▶ \vdots
- ▶ **Size of function class (structural risk minimization)**
 - ▶ VC-Dimension (next week)

Number of Parameters of the Model

Constant classifier

$$g(\mathbf{x}) = \underbrace{C}_1$$

$\mathcal{O}(1)$ parameters

Nearest mean classifier

$$g(\mathbf{x}) = \mathbf{x}^\top \underbrace{(\mathbf{m}_1 - \mathbf{m}_2)}_{2d} + \underbrace{C}_1$$

$\mathcal{O}(d)$ parameters

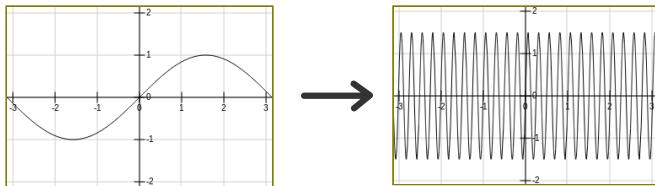
Fisher vector classifier (on top of k PCA components):

$$g(\mathbf{x}) = \underbrace{\text{PCA}(\mathbf{x})^\top}_{k \cdot d} \underbrace{S_W^{-1}}_{k^2} \underbrace{(\mathbf{m}_1 - \mathbf{m}_2)}_{2k} + \underbrace{C}_1 \quad \mathcal{O}(k \cdot d) \text{ parameters}$$

Number of Parameters of the Model

Counter-example

- ▶ The two-parameters model $g(x) = a \sin(\omega x)$ can fit almost *any* finite dataset in \mathbb{R} , by setting very large values for ω .



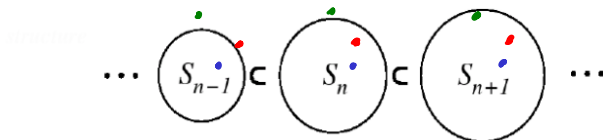
- ▶ However, it is also clear that the model will not generalize well.

By only counting the number of parameters in the model, we have not specified the range of values the parameter ω is allowed to take in practice.

Structural Risk Minimization

Vapnik and Chervonenkis (1974)

Idea: Structure your space of solutions into a nesting of increasingly larger regions. If two solutions fit the data, prefer the solution that belongs to the smaller region.



Example:

Assuming \mathbb{R}^d is the whole solution space for θ , build a sequence of real numbers $C_1 < C_2 < \dots < C_N$, and create a nested collection (S_1, S_2, \dots, S_N) of regions, where


$$S_n = \{\theta \in \mathbb{R}^d: \|\theta\|^2 \leq C_n\}$$

Connection to Regularization

Example (cont.): We optimize for multiple C_n the objective

$$\min_{\theta} \underbrace{\sum_{i=1}^N \ell(y_i, f_{\theta}(\mathbf{x}_i))}_{\mathcal{E}_{\text{empirical}}} \quad \text{s.t. } \|\theta\|^2 < C_n$$

and discard solutions with index n larger than necessary to fit the data.
This objective can be equivalently formulated as:

$$\min_{\theta} \left[\underbrace{\sum_{i=1}^N \ell(y_i, f_{\theta}(\mathbf{x}_i))}_{\mathcal{E}_{\text{empirical}}} + \underbrace{\lambda_n \|\theta\|^2}_{\mathcal{E}_{\text{reg}}} \right]$$


with appropriate $(\lambda_n)_n$. This objective is known in various contexts as L_2 regularization, ridge regression, large margin, weight decay, etc.

From Occam's Razor to Popper

Occam's Razor

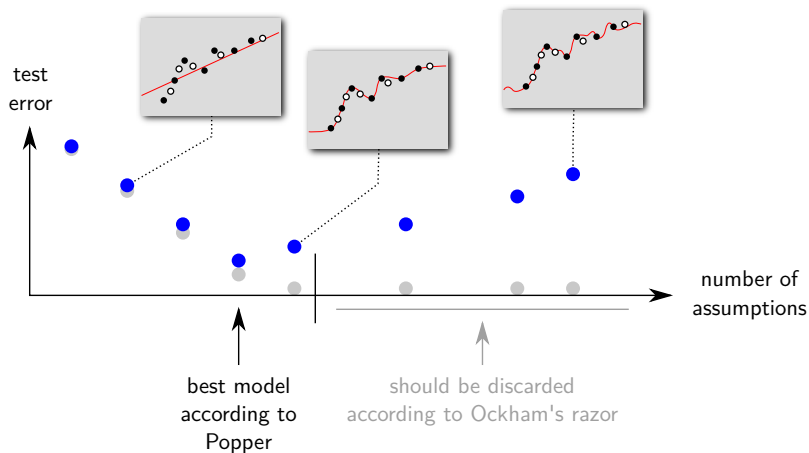
"Entities must not be multiplied beyond necessity."

Falsifiability/prediction strength (S. Hawking, after K. Popper)

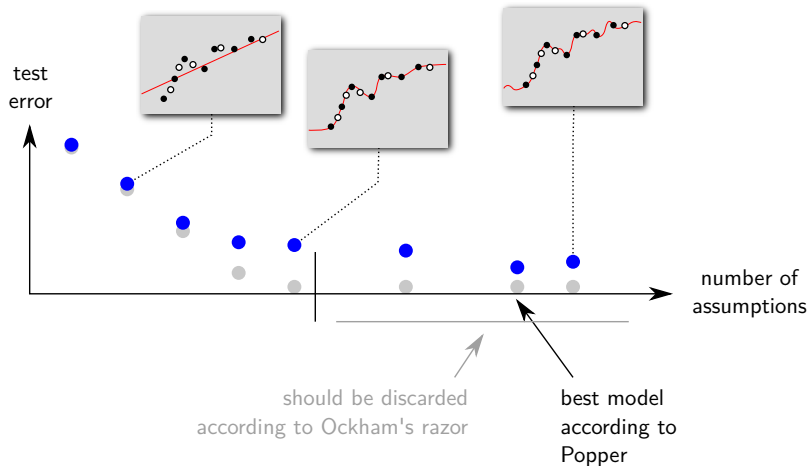
"[a good model] must accurately describe a large class of observations on the basis of a model that contains only a few arbitrary elements, and it must make definite predictions about the results of future observations."

In other words, the model with lowest generalization error is preferable.

From Occam's Razor to Popper

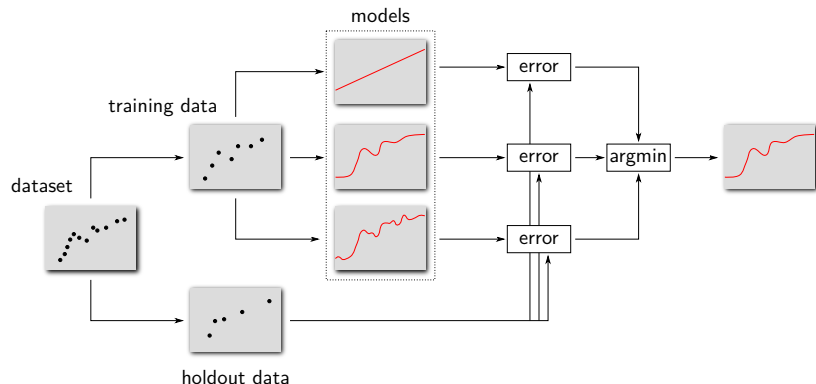


From Occam's Razor to Popper



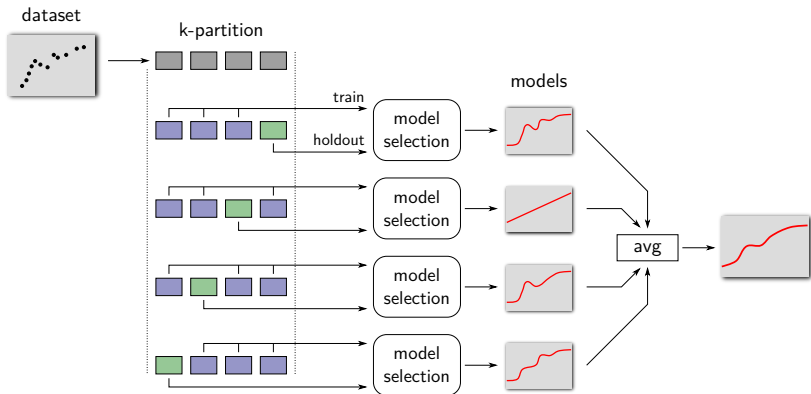
The Holdout Selection Procedure

Idea: Predict out-of-sample error by splitting the data randomly in two parts (one for training, and one for estimating the error of the model).



Cross-Validation (k -Fold Procedure)

To improve error estimates without consuming too much data, the processes of error estimation can be improved by computing an average over different splits:



The Cross-Validation Procedure

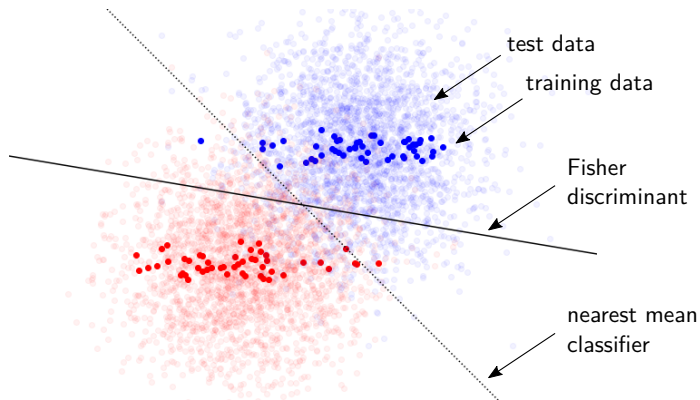
Advantages:

- ▶ The model can now be selected directly based on simulated future observations.

Limitations:

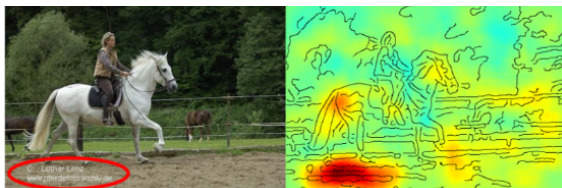
- ▶ For a small number of folds k , the training data is reduced significantly, which may lead to less accurate models. For k large, the procedure becomes computationally costly.
- ▶ This technique assumes that the available data is representative of the future observations (not always true!).

The Problem of Dataset Bias



This effect can lead cross-validation procedure to not work well, even when we have enough training data.

The Problem of Dataset Bias



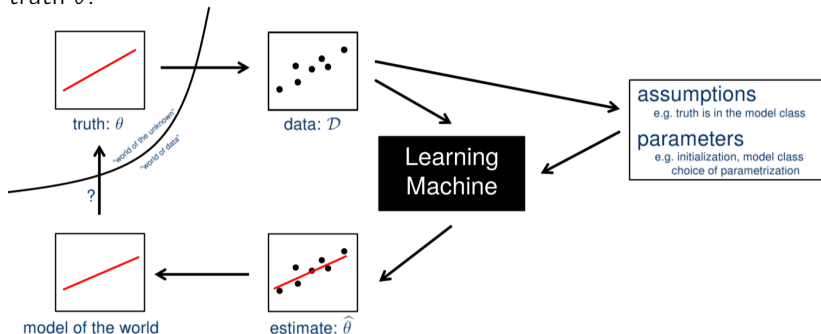
Observation: The classifier has exploited a spurious correlation between images of the class horse and the presence of a copyright tag in the left bottom corner of the horse images. Here, cross-validation doesn't help here because the spurious correlation would also be present in the validation set.

Further reading: Lapuschkin et al. (2019) Unmasking Clever Hans predictors and assessing what machines really learn

Part II. Bias-Variance Analysis of ML Models

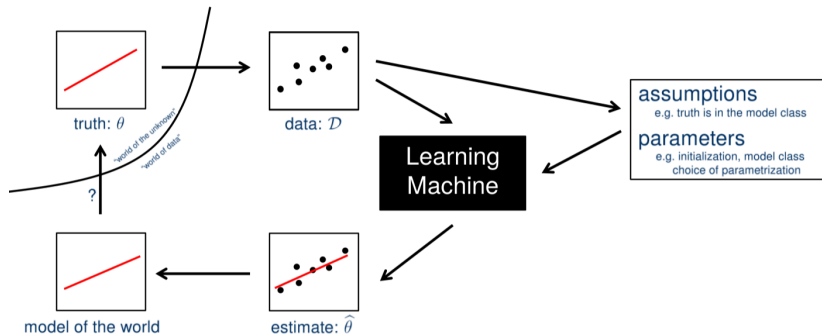
Machine Learning Models

Machine learning models are learned from the data to approximate some truth θ .



A learning machine can be abstracted as a function that maps a dataset \mathcal{D} to an estimator $\hat{\theta}$ of the truth θ .

ML Models and Prediction Error



A good learning machine is one that produces an estimator $\hat{\theta}$ close to the truth θ . Closeness to the truth can be measured by some error function, e.g. the square error:

$$\text{Error}(\hat{\theta}) = (\theta - \hat{\theta})^2.$$

Bias, Variance, and MSE of an Estimator

Parametric estimation:

- ▶ θ is a value in \mathbb{R}^h
- ▶ $\hat{\theta}$ is a function of the data $\mathcal{D} = \{X_1, \dots, X_N\}$, where X_i are random variables producing the data points.

Statistics of the estimator:

$$\text{Bias}(\hat{\theta}) = \mathbb{E}[\hat{\theta} - \theta] \quad (\text{measures expected deviation of the mean})$$

$$\text{Var}(\hat{\theta}) = \mathbb{E}[(\hat{\theta} - \mathbb{E}[\hat{\theta}])^2] \quad (\text{measures scatter around estimator of mean})$$

$$\text{MSE}(\hat{\theta}) = \mathbb{E}[(\hat{\theta} - \theta)^2] \quad (\text{measures prediction error})$$

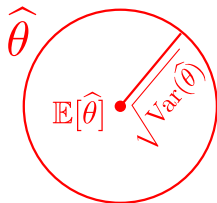
Note: for $\theta \in \mathbb{R}^h$, we use the notation $\theta^2 = \theta^\top \theta$.

Visualizing Bias and Variance

True parameter

θ •

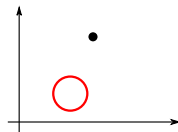
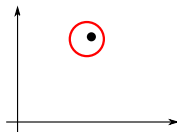
Parameter estimator



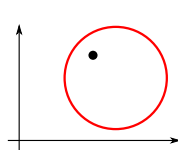
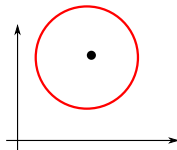
low bias

high bias

low variance



high variance



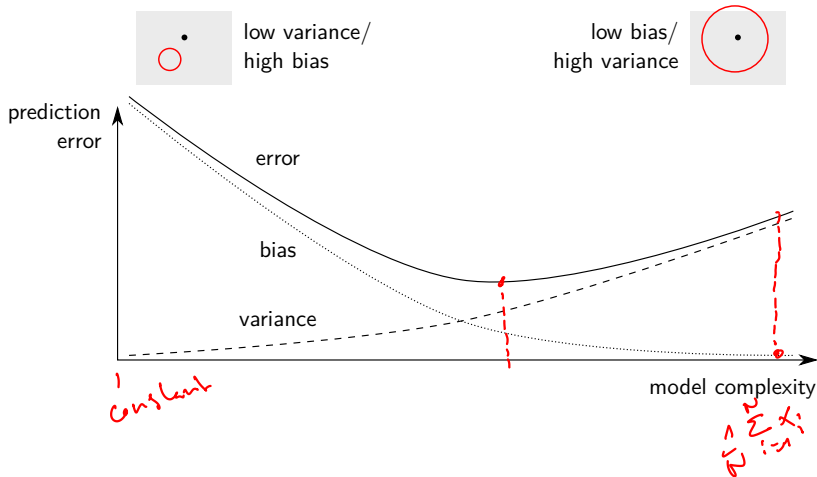
Bias-Variance Decomposition

$$\text{Bias}(\hat{\theta}) = \mathbb{E}[\hat{\theta} - \theta], \quad \text{Var}(\hat{\theta}) = \mathbb{E}[(\hat{\theta} - \mathbb{E}[\hat{\theta}])^2], \quad \text{MSE}(\hat{\theta}) = \mathbb{E}[(\hat{\theta} - \theta)^2]$$

We can show that $\text{MSE}(\hat{\theta}) = \text{Bias}(\hat{\theta})^2 + \text{Var}(\hat{\theta})$.

$$\begin{aligned} \text{MSE}(\hat{\theta}) &= \mathbb{E}[(\hat{\theta} - \mathbb{E}[\hat{\theta}] + \mathbb{E}[\hat{\theta}] - \theta)^2] \\ &= \mathbb{E}[\underbrace{(\hat{\theta} - \mathbb{E}[\hat{\theta}])^2}_{\text{Var}(\hat{\theta})}] + \cancel{\mathbb{E}[(\mathbb{E}[\hat{\theta}] - \theta)^2]} + \mathbb{E}[(\hat{\theta} - \mathbb{E}[\hat{\theta}])(\mathbb{E}[\hat{\theta}] - \theta)] \\ &\quad + \underbrace{\text{Bias}(\hat{\theta})^2}_{\text{Bias}(\hat{\theta})^2} + \underbrace{2(\mathbb{E}[\hat{\theta}] - \mathbb{E}[\mathbb{E}[\hat{\theta}]])}_0 \dots \end{aligned}$$

Visualizing Bias and Variance



Example: Parameters of a Gaussian

parametric estimation:

θ is a value in \mathbb{C}^n (e.g. $\theta = (\mu, \Sigma)$ for Gaussians)

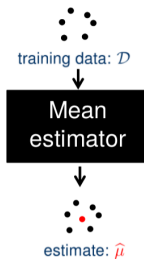
$\hat{\theta}$ is function in the data $\mathcal{D} = \{X_1, \dots, X_N\}$
(X_i are random variables giving back data points)

e.g. mean estimator

$$\hat{\mu} = \frac{1}{N} \sum_{i=1}^N X_i$$

covariance estimator

$$\hat{\Sigma} = \frac{1}{N-1} (X_i - \hat{\mu})(X_i - \hat{\mu})^\top$$



Example: Parameters of a Gaussian

The 'natural' estimator of mean $\hat{\mu} = \frac{1}{N} \sum_{i=1}^N X_i$ decomposes to

$$\boxed{\text{Bias}(\hat{\mu}) = 0} \quad \text{and} \quad \boxed{\text{Var}(\hat{\mu}) = \sigma^2/N}.$$

$$\begin{aligned} \text{bias}(\hat{\mu}) &= E[\hat{\mu} - \mu] = E\left[\frac{1}{N} \sum_{i=1}^N X_i\right] - \mu \\ &= \frac{1}{N} \sum_{i=1}^N \underbrace{E[X_i]}_{\mu} - \mu = \mu - \mu = 0 \\ \text{Var}(\hat{\mu}) &= \text{Var}\left(\frac{1}{N} \sum_{i=1}^N X_i\right) = \frac{1}{N^2} \text{Var}\left(\sum_{i=1}^N X_i\right) \\ &= \frac{1}{N^2} \sum_{i=1}^N \underbrace{\text{Var}(X_i)}_{\sigma^2} = \frac{1}{N} \cdot \sigma^2 \end{aligned}$$

The James-Stein Estimator

$$\hat{\mu} = \frac{1}{N} \sum_{i=1}^N X_i$$

“natural” estimator

$$\text{Bias}(\hat{\mu}) = 0$$

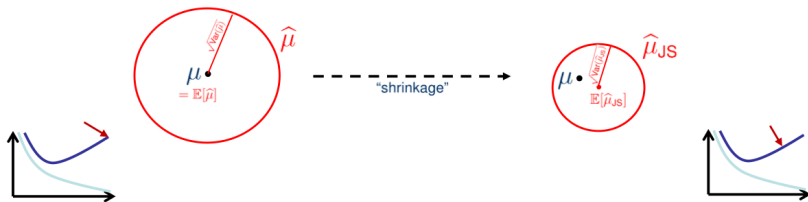
$$\text{MSE}(\hat{\mu}) = \text{Var}(\hat{\mu}) = \frac{\sigma^2}{N}$$

$$\hat{\mu}_{\text{JS}} = \hat{\mu} - \frac{(n-2)\sigma^2}{\hat{\mu}^2} \hat{\mu}$$

James-Stein estimator

$$\text{Bias}(\hat{\mu}_{\text{JS}}) > 0$$

$$\text{MSE}(\hat{\mu}_{\text{JS}}) < \text{MSE}(\hat{\mu})$$



Estimator of Functions

supervised learning:

training data \mathcal{D} is X_1, \dots, X_N with labels Y_1, \dots, Y_N
(e.g. in regression, $X_i \in \mathbb{R}^n, Y_i \in \mathbb{R}$)

parameter θ “is” a generative function $f = f_\theta$:

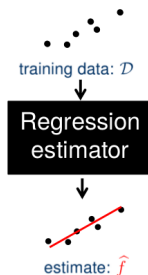
$$Y_i = f(X_i) + \varepsilon_i$$

ε_i is error with $\mathbb{E}[\varepsilon_i] = 0$

Learning Machine learns approximation $\hat{f} = f_{\hat{\theta}}$
such that $Y_i \approx \hat{f}(X_i)$

Example (Linear Regression):

$$f(x) = \beta^\top x + \alpha, \quad \theta = (\alpha, \beta)$$



Bias-Variance Analysis of the Function Estimator (locally)

supervised learning:

training data \mathcal{D} is X_1, \dots, X_N with labels Y_1, \dots, Y_N
(e.g. in regression, $X_i \in \mathbb{R}^n, Y_i \in \mathbb{R}$)

parameter θ “is” a generative function $f = f_\theta$:

$$Y_i = f(X_i) + \varepsilon_i$$

$$\text{bias of } \hat{f} \text{ at } X_i: \quad \text{Bias}(\hat{f}|X_i) = \mathbb{E}_Y[\hat{f}(X_i) - f(X_i)]$$

$$\text{variance of } \hat{f} \text{ at } X_i: \quad \text{Var}(\hat{f}|X_i) = \mathbb{E}_Y \left[(\hat{f}(X_i) - \mathbb{E}_Y[\hat{f}(X_i)])^2 \right]$$

$$\text{MSE of } \hat{f} \text{ at } X_i: \quad \text{MSE}(\hat{f}|X_i) = \mathbb{E}_Y \left[(\hat{f}(X_i) - Y_i)^2 \right]$$

$$\textbf{Proposition: } \text{MSE}(\hat{f}|X_i) = \text{Var}(\varepsilon_i) + \text{Bias}(\hat{f}|X_i)^2 + \text{Var}(\hat{f}|X_i)$$

Summary

- ▶ **Occam's Razor:** Given two models with the same training error, the simpler one should be preferred.
- ▶ **Popper's View:** How to make sure that a model predicts well? By testing it on out-of-sample data. Out-of-sample data can be simulated by applying a k-fold cross-validation procedure.
- ▶ **Bias-Variance Decomposition:** The error of a predictive model can be decomposed into bias and variance. Best models often results from some tradeoff between the two terms.