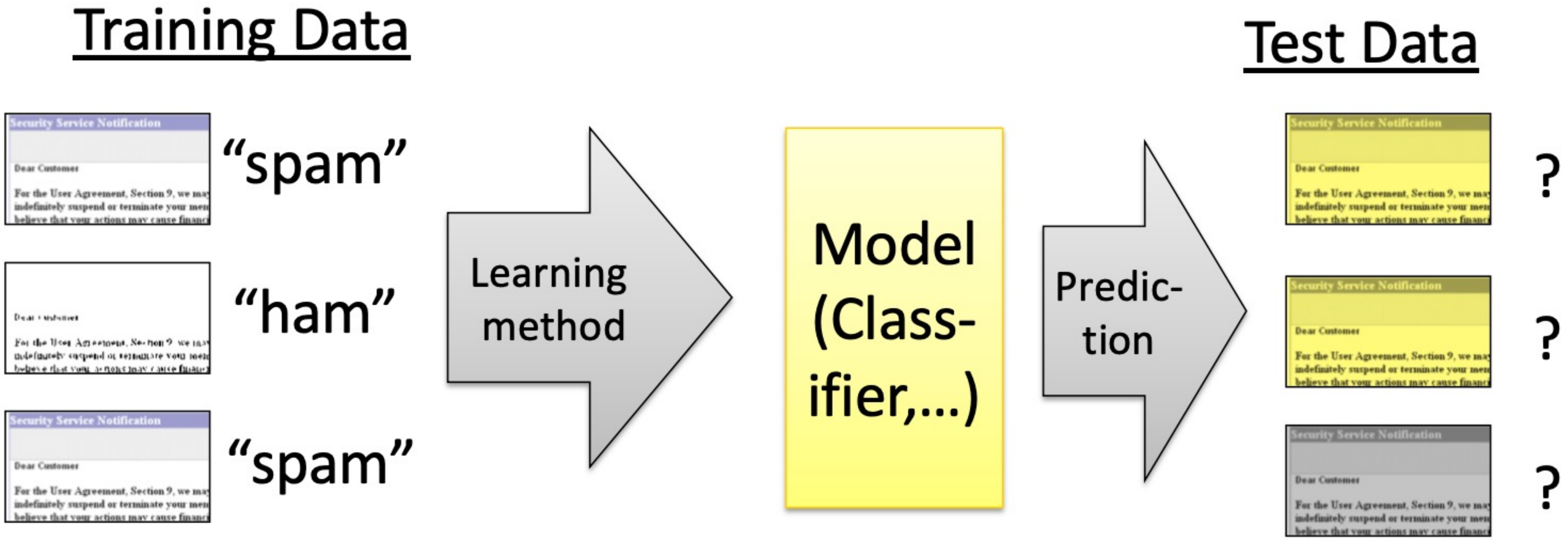
THE UNIVERSITY OF CHICAGO

STAT 37710 / CMSC 35400 / CAAM 37710
Machine Learning

Linear regression: statistical perspective

Cong Ma

Basic supervised learning pipeline

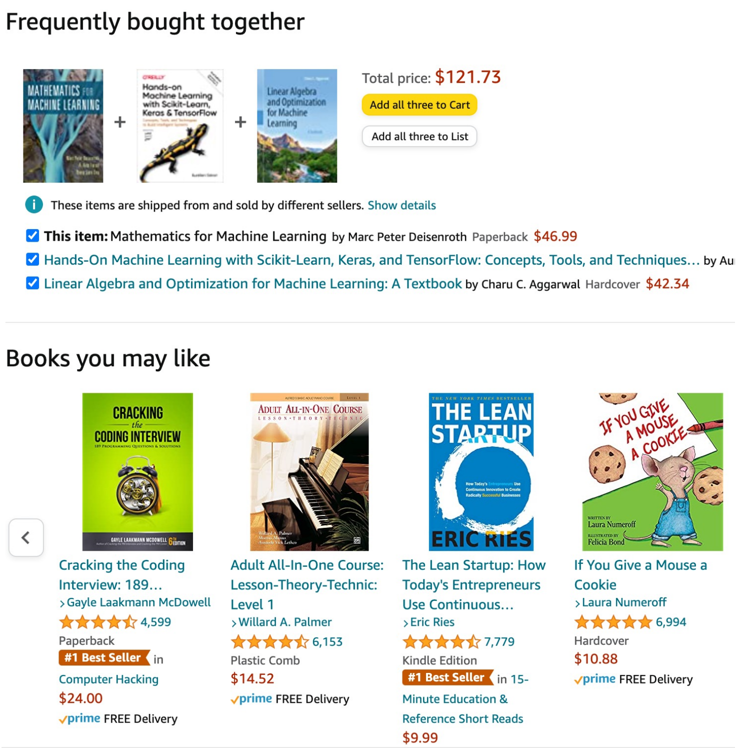


$\mathcal{X}$ $\mathcal{Y}$ $f : \mathcal{X} \rightarrow \mathcal{Y}$ $\mathcal{X}$

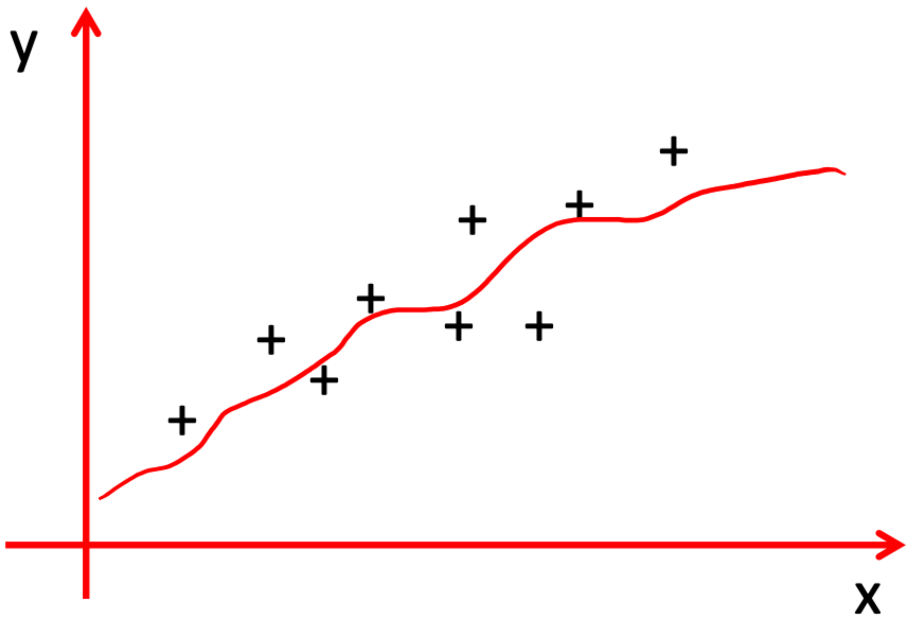
Example: Recommender systems

- **X**: User & article / product features

  Y: Ranking of articles / products to display



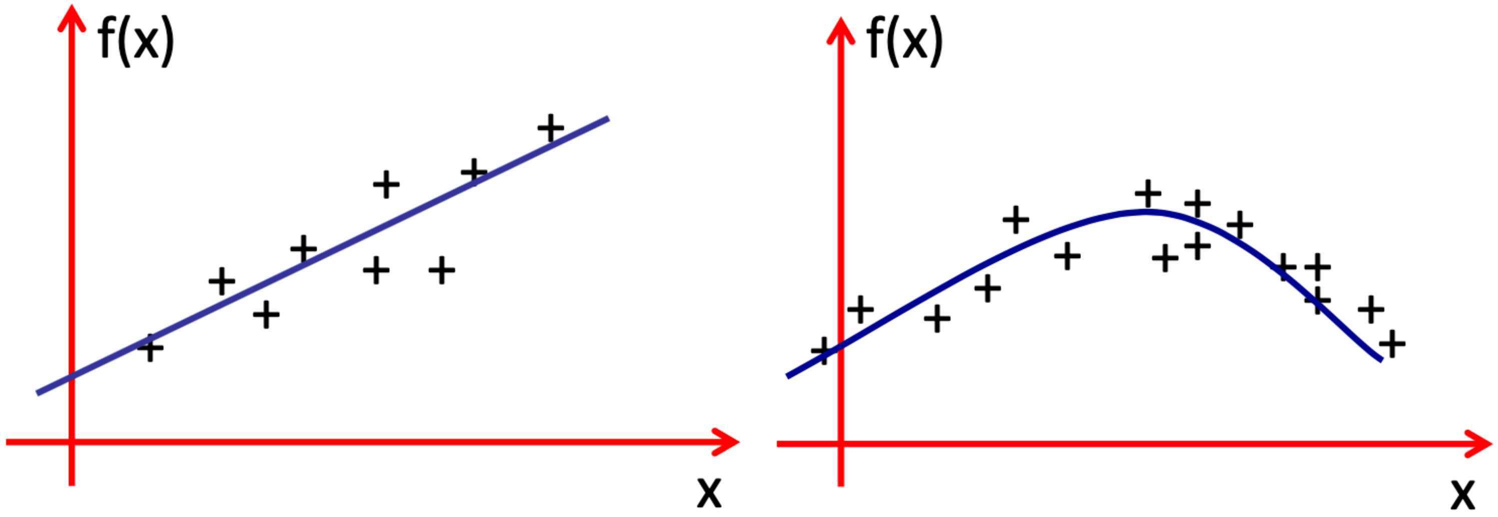
Regression



- **Goal**: learn real valued mapping $f : \mathbb{R}^d \to \mathbb{R}$

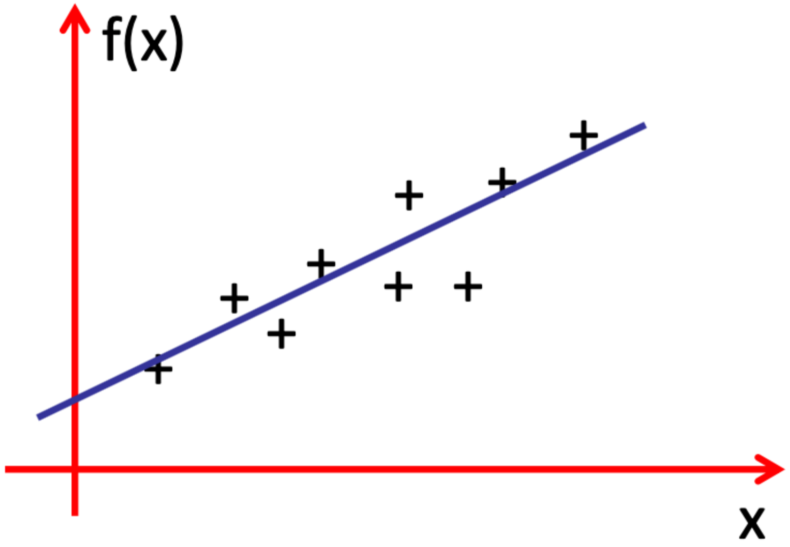
Important choices in regression

- What types of functions f should we consider?



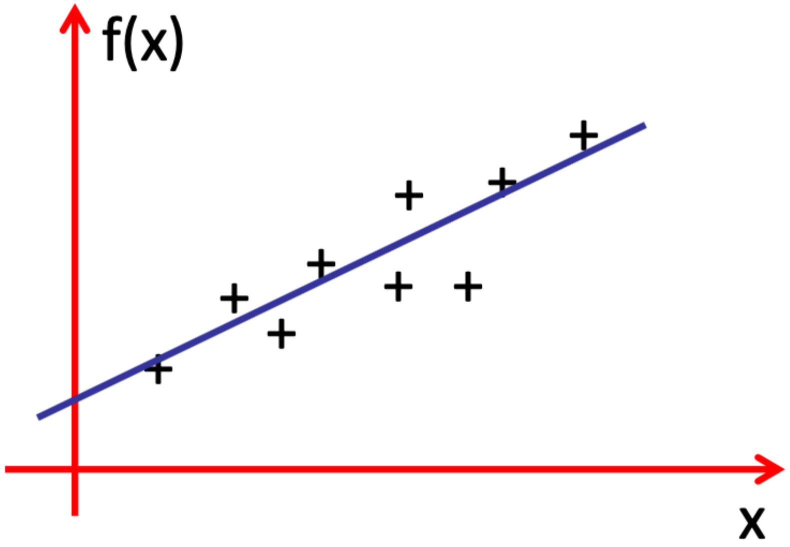
- How should we measure goodness of fit?

Linear regression



Quantifying goodness of fit

$$D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\} \qquad \mathbf{x}_i \in \mathbb{R}^d \qquad y_i \in \mathbb{R}$$



Least-squares linear regression optimization

- Given data set $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ $\mathbf{x}_i \in \mathbb{R}^d$ $y_i \in \mathbb{R}$
- How do we find the optimal weight vector?

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_{i=1}^{n} (y_i - \mathbf{w}^T \mathbf{x}_i)^2$$

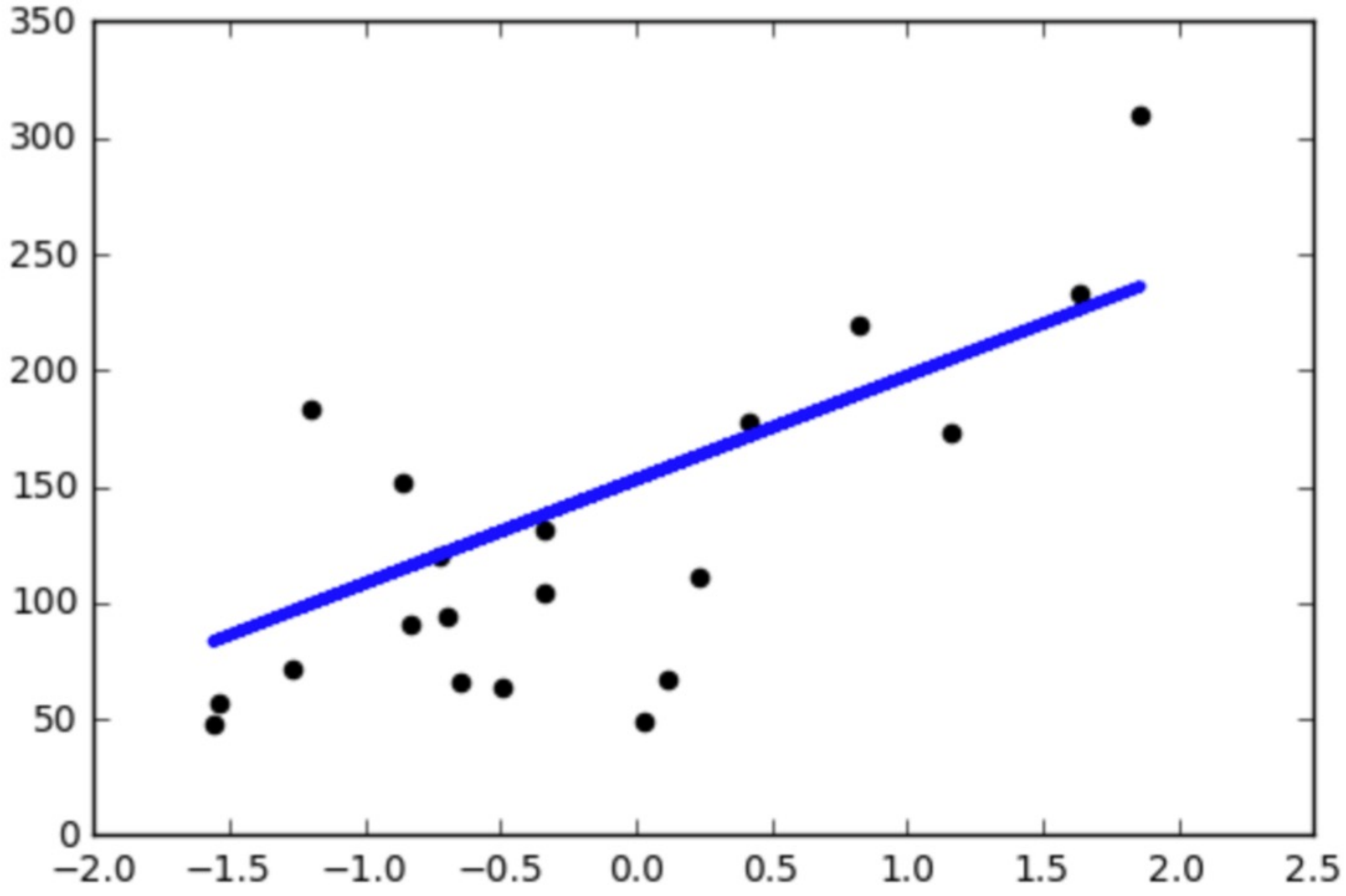
[Legendre 1805, Gauss 1809]

How to solve?

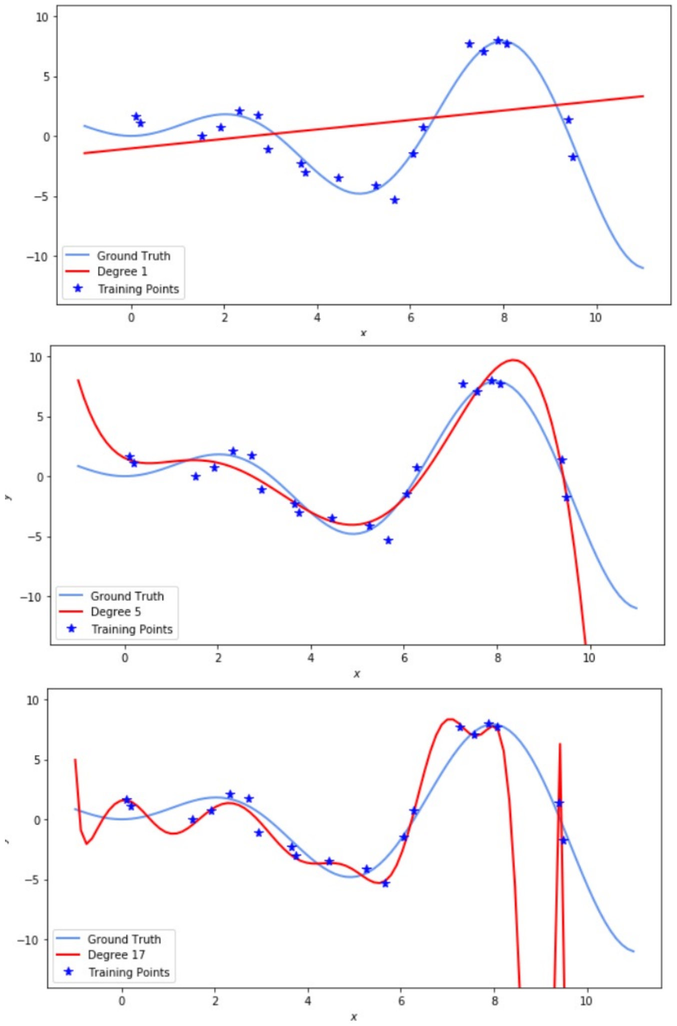
- Example: Scikit Learn

```
# Create linear regression object
regr = linear_model.LinearRegression()

# Train the model using the training set
regr.fit(X_train, Y_train)

# Make predictions on the testing set
Y_pred = regr.predict(X_test)
```

Demo

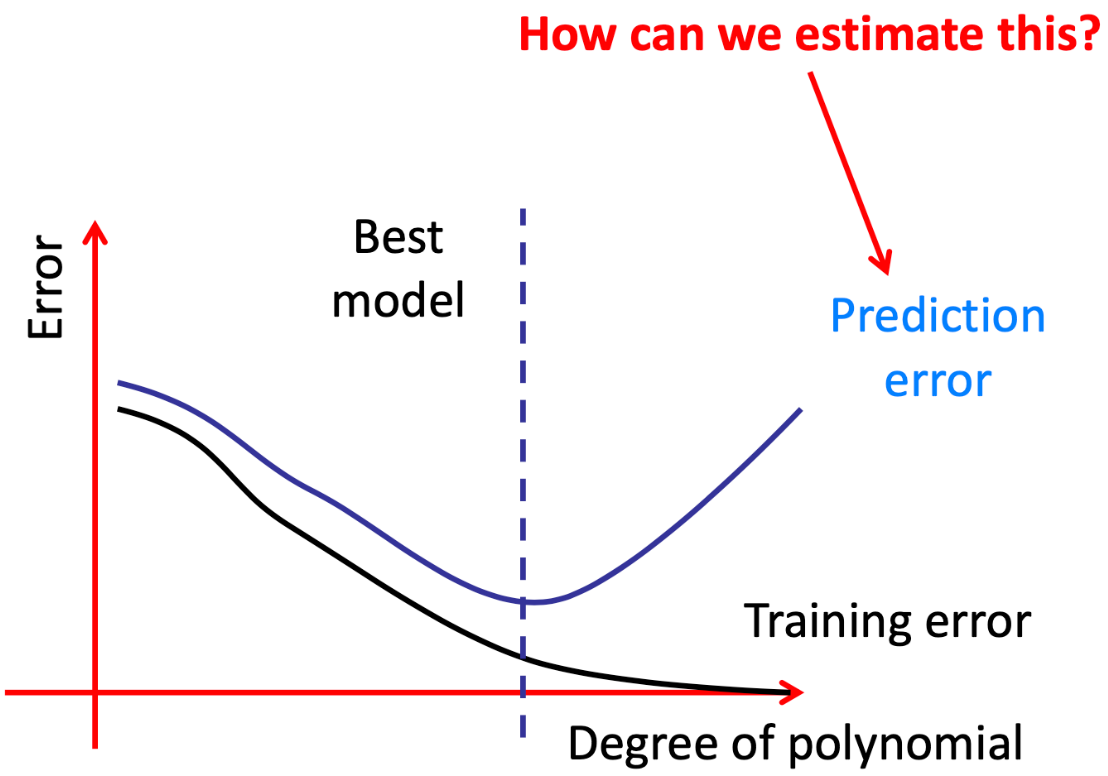


Least-squares regression with polynomials



Underfitting

Overfitting



Regression from a statistical perspective

- Fundamental assumption: Our data set is generated ***independently and identically distributed*** (*iid*) from some unknown distribution *P*

$$(\mathbf{x}_i, y_i) \sim P(\mathbf{X}, Y)$$

- Our goal is to minimize the ***expected error (true risk)*** under *P*

$$R(h) = \int P((x), y)\ell(y; h(\mathbf{x}))d\mathbf{x}dy = \mathbb{E}_{\mathbf{x},y}\left[\ell(y; h(\mathbf{x}))\right]$$

Note on iid assumption

- When is iid assumption invalid?
  - Time series data
  - Spatially correlated data
  - Correlated noise
  - ...

- Often, can still use machine learning, but one has to be careful in interpreting results.
- Most important: Choose train/test to assess the desired generalization

Examples of loss function ℓ for regression

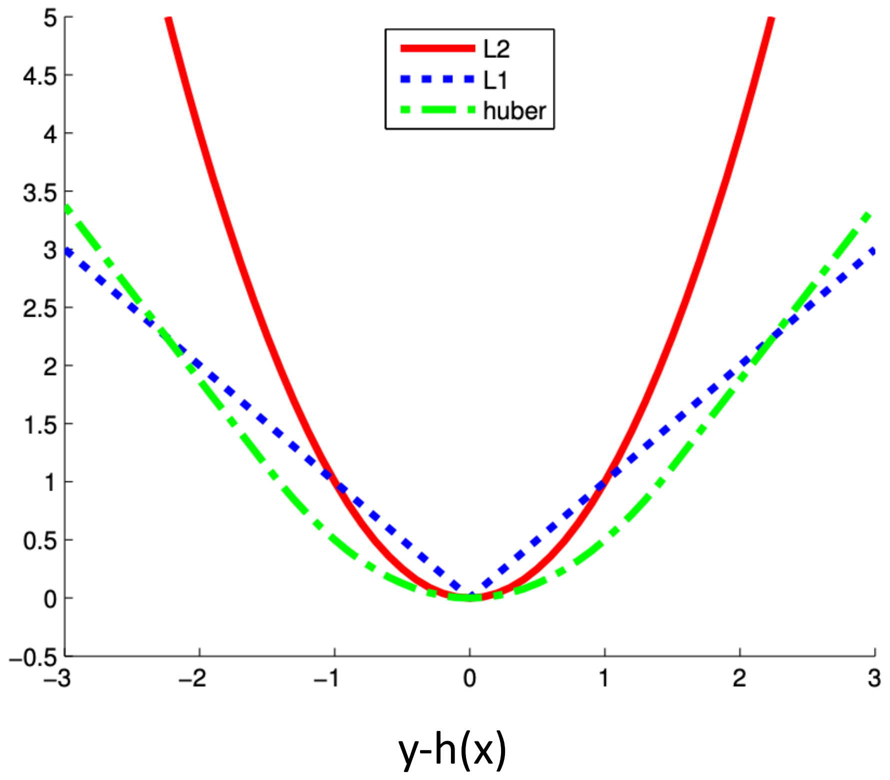
- square loss:

$$\ell(f(x), y) = (y - \mathrm{h}(x))^2$$

- absolute loss

$$\ell(f(x), y) = |y - \mathrm{h}(x)|$$

- huber loss:
  - quadratic for $|y - \mathrm{h}(x)| < \delta$
  - linear for $|y - \mathrm{h}(x)| > \delta$
  - robust and differentiable



Least-squares regression

- In least-squares regression, risk is $R(h) = \mathbb{E}[(y - h(\mathbf{x}))^2]$
- Suppose (unrealistically) we *knew* P(**X**,Y)
  - Which *h* minimizes the risk?
  - For a given x, what is the optimal prediction?

Minimizing the mean squared error (MSE)

- Assuming the data is generated iid according to $(\mathbf{x}_i, y_i) \sim P(\mathbf{X}, Y)$
- The hypothesis h* minimizing $R(h) = \mathbb{E}_{\mathbf{x},y}[(y - h(\mathbf{x}))^2]$ is given by the conditional mean

$$h^*(\mathbf{x}) = \mathbb{E}[Y \mid \mathbf{X} = \mathbf{x}]$$

- This (in practice unattainable) hypothesis is called the

  Bayes' optimal predictor

  for the squared loss (or regression function)

Proof

In practice we have finite data

- Empirical risk minimization
- Can we do it over all possible functions?

$$\hat{h} = \hat{h}_D = \arg\min_{h \in \mathcal{H}} \sum_{(\mathbf{x}, y) \in D} (y - h(\mathbf{x}))^2$$

- For instance, we choose linear function class
- What's the performance of this ERM estimator?

Bias-variance tradeoff

- For least-squares estimation the following holds

$$\overbrace{\mathbb{E}_D \mathbb{E}_{\mathbf{X},Y}\left[\left(Y - \hat{h}(\mathbf{X})\right)^2\right]}^{\text{Expected risk}} = \mathbb{E}_{\mathbf{X}}\left[\underbrace{\mathbb{E}_D \hat{h}_D(\mathbf{X}) - h^*(\mathbf{X})}_{\text{Bias}}\right]^2$$

$$+ \underbrace{\mathbb{E}_{\mathbf{X}}\, \mathbb{E}_D \left[\hat{h}_D(\mathbf{X}) - \mathbb{E}_{D'} \hat{h}_{D'}(\mathbf{X})\right]^2}_{\text{Variance}}$$

$$+ \underbrace{\mathbb{E}_{\mathbf{X},Y}\left[Y - h^*(\mathbf{X})\right]^2}_{\text{Noise}}$$

- Ideally wish to find estimator that simultaneously minimizes bias and variance

Noise in estimation

- Even if we know the Bayes' optimal hypothesis h*, we'd still incur some error due to **noise**

$$\mathbb{E}_{\mathbf{X},Y}[(Y - h^*(\mathbf{X}))^2]$$

- This error is irreducible, i.e., independent of choice of the hypothesis class

Bias in estimation

- ERM estimator depends on training data *D*

$$\hat{h} = \hat{h}_D = \arg\min_{h \in \mathcal{H}} \sum_{(\mathbf{x},y) \in D} (y - h(\mathbf{x}))^2$$

- But training data *D* is itself random (drawn iid from P)
- We might want to choose *H* to have small **bias**
  - (i.e., have small squared error on average)

$$\mathbb{E}_{\mathbf{X}} \left[\mathbb{E}_D \hat{h}_D(\mathbf{X}) - h^*(\mathbf{X}) \right]^2$$

Variance in estimation

- MLE solution depends on training data *D*

$$\hat{h} = \hat{h}_D = \arg\min_{h \in \mathcal{H}} \sum_{(\mathbf{x},y) \in D} (y - h(\mathbf{x}))^2$$

- This estimator is itself random, and has some **variance**

$$\mathbb{E}_{\mathbf{X}} \mathrm{Var}_D \left[\hat{h}_D(\mathbf{X})\right]^2 = \mathbb{E}_{\mathbf{X}} \mathbb{E}_D \left[\hat{h}_D(\mathbf{X}) - \mathbb{E}_{D'} \hat{h}_{D'}(\mathbf{X})\right]^2$$

Proof

Bias and variance in regression

- We have seen that the least-squares solution can overfit
- Thus, trade (a little bit of) bias for a (potentially dramatic) reduction in variance:
  - Regularization (e.g., ridge regression, Lasso, ...)

Summary: Bias Variance Tradeoff

Prediction error = Bias^2 + Variance + Noise

Bias	Excess risk of best model considered compared to minimal achievable risk knowing P(X,Y) (i.e., given infinite data)
Variance	Risk incurred due to estimating model from limited data
Noise	Risk error incurred by optimal model (i.e., irreducible error)

- Trade bias and variance via model selection / regularization

Summary

- Where we are
  - The statistical learning framework: data, model class, loss function
  - Mean squared error (square loss) and bias-variance decomposition

- What's next
  - Given training data and a (parametric) model class $\mathcal{F}$, how to estimate model parameter from observations

References & acknowledgement

- C. Bishop (2006). “Pattern Recognition and Machine Learning”
  - Ch 3.2, “The Bias-Variance Decomposition”

- Deisenroth et al. (2020). “Mathematics for Machine Learning”
  - Ch 8.3 “Parameter Estimation”

- A. Krause, “Introduction to Machine Learning” (ETH Zurich, 2019)