UNIVERSITY AT ALBANY
STATE UNIVERSITY OF NEW YORK

CSI 436/536 (Spring 2025)

Machine Learning

Lecture 13: Error Decomposition

Chong Liu

Department of Computer Science

Mar 31, 2025

Today

- Naïve Bayes model
- Generalization error by bias-variance decomposition
  - Understand the problem of overfitting
- Learning risk decomposition
  - Understand machine learning

Recap: Direct modeling $p(x|y)$ is hard

Binary vectors, 2^3 rows + binary output $Y \in \{0,1\}$

x_1	x_2	x_3
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

1. At least 8 data points are needed to determine $p(x|y)$.

2. At least 2^d data points are needed if there are d binary features.

Naïve Bayes assumption

- Given the class label y, the features are conditional independent of each other.
  - $P(x|y) = \prod_j p(x_j|y)$
  - x_j is the j-th feature, not j-th data point!

- How to understand?
  - All features independently affect the label.

Model $p(x|y)$ with naïve Bayes assumption

Binary vectors, 2^3 rows + binary output $Y \in \{0,1\}$

x_1	x_2	x_3
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Discussion: How many data points are needed to determine $p(x|y)$ given binary output and d binary features?

$$2d$$

Why? When we use the first feature, we ignore all other features...

Naïve Bayes model

- Goal of probabilistic model:
  - $h(x) = \text{argmax}_y P(y|x)$
  - $h(x) = \text{argmax}_y \frac{P(y)P(x|y)}{P(x)}$
  - $h(x) = \text{argmax}_y P(y)P(x|y)$
  - $h_{\text{Naïve\_Bayes}}(x) = \text{argmax}_y P(y) \prod_{j=1}^{d} P(x_j|y)$

- Bayes rule
  - $P(y|x) = \frac{P(y)P(x|y)}{P(x)}$
- Naïve Bayes assumption
  - $P(x|y) = \prod_j p(x_j|y)$

In-class exercise: Naïve Bayes model for email filter

Email ID	Contains "buy"	Contains "cheap"	Contains "free"	Label
1	Yes	Yes	No	Spam
2	No	No	Yes	Not Spam
3	Yes	No	Yes	Spam
4	No	Yes	No	Not Spam
5	Yes	Yes	Yes	Spam
6	No	No	No	Not Spam

- Naïve Bayes model: $h_{\text{Naïve\_Bayes}}(x) = \text{argmax}_y P(y) \prod_{j=1}^{d} P(x_j|y)$
- Step 1: calculate $P(y)$
- Step 2: calculate $P(x_j|y)$
- Step 3: prediction on [Yes, No, No]

Checkpoint: Probabilistic models

- Probabilistic / Generative / Bayesian models are very powerful and interpretable method for modeling the world.
  - Customize ML models for your applications
  - Explicitly model the dependence
- Key task: Modeling $P(x|y)$
  - Maximum likelihood estimation
    - Data are identically and independently distributed (i.i.d.)
  - Naïve Bayes model
    - Features are independent given the label

So far, we have learned a lot of ML algorithms

- The key goal of ML algorithms is to
  - Minimize the generalization error
  - We want to train a learning algorithm that works well on test data

- The ultimate goal of ML algorithms is to
  - Learn the best hypothesis!

- What are the factors that systematically affect learning process?

Bias-variance decomposition

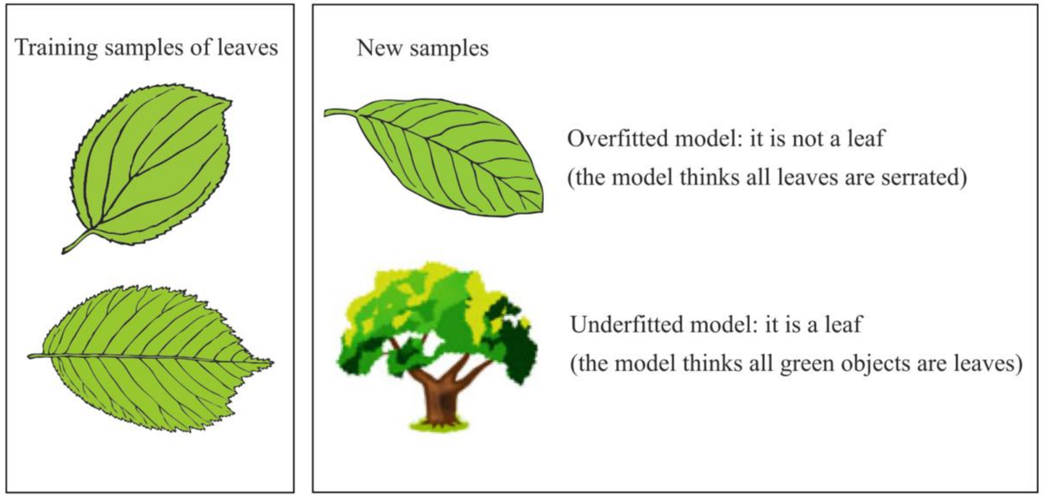
- Definitions:
  - Feature: x
  - Label: $y = f(x) + \epsilon$
    - Label generating function: $f(x)$
    - Noise: $\epsilon, E[\epsilon] = 0, Var[\epsilon] = \sigma^2$
  - Prediction: $\hat{y}$
- Bias:
  - $|f(x) - E[\hat{y}]|$
- Variance:
  - $E[(\hat{y} - E[\hat{y}])^2]$
- Generalization error:
  - $E[(y - \hat{y})^2]$

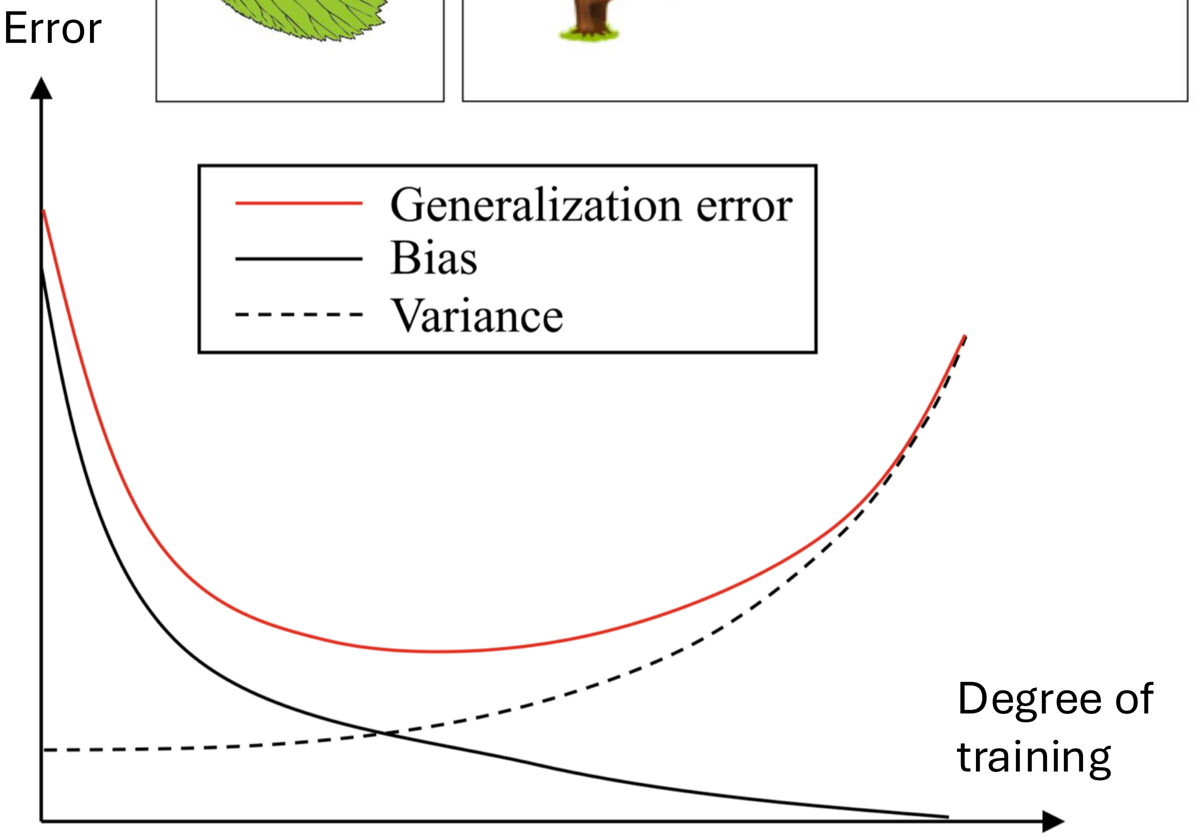
- Generalization error decomposition:
  - $E[(y - \hat{y})^2] = Variance + bias^2 + \sigma^2$

  - Bias:
    - fitting of learning algorithm
  - Variance:
    - effect of given dataset
  - Noise:
    - difficulty of the learning problem

Bias-variance trade-off

- Generalization error decomposition:
  - $E[(y-\hat{y})^2] = Variance + bias^2 + \sigma^2$
- How to control the degree of training:
  - SGD: number of rounds
- Less training:
  - Model fitting is so weak, high bias
  - Underfitted
- Too much training:
  - Learned a lot of details of data, high variance
  - Overfitting





Loss, Empirical Risk, and Risk

- Loss function

$$\ell(h, (x, y))$$

- Empirical Risk function

$$\hat{R}(h, \mathrm{Data}) = \frac{1}{n} \sum_{i=1}^{n} \ell(h, (x_i, y_i))$$

- (Population) Risk function

$$R(h, \mathcal{D}) = \mathbb{E}_{\mathcal{D}}[\ell(h, (x_i, y_i))]$$

Bayes optimal classifier, optimal classifier within the hypothesis class, Empirical Risk Minimizer

- Bayes Optimal classifier: $h_{\text{Bayes}} = \arg\min_{h} R(h)$
  - For 0-1 loss, the Bayes optimal classifier is

$$h_{\text{Bayes}} = \arg\max_{y} p(y|x) = \arg\max_{y} p(x|y)p(y)$$

- Optimal (within hypothesis class) classifier $h^* = \arg\min_{h \in \mathcal{H}} R(h)$

- ERM Classifier $h_{\text{ERM}} = \arg\min_{h \in \mathcal{H}} \hat{R}(h)$

- My classifier $\hat{h} = \text{My\_Learning\_Algorithm}(\text{Data})$

Risk Decomposition

$$\mathbb{E}[R(\hat{h})] - R(h_{\text{Bayes}})$$

$$\leq \boxed{\mathbb{E}[\hat{R}(\hat{h}) - \hat{R}(h_{\text{ERM}})]} + \boxed{R(h^*) - R(h_{\text{Bayes}})} + \boxed{\mathbb{E}[R(\hat{h}) - \hat{R}(\hat{h})]}$$

Optimization error

Approximation error

Generalization error