UNIVERSITY AT ALBANY
STATE UNIVERSITY OF NEW YORK

CSI 436/536 (Fall 2024)

# Machine Learning

## Lecture 14: Error Decomposition

Chong Liu

Assistant Professor of Computer Science

Oct 31, 2024

# Today

- Generalization error by bias-variance decomposition
  - Understand the problem of overfitting
- Learning risk decomposition
  - Introduction to learning theory

# 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

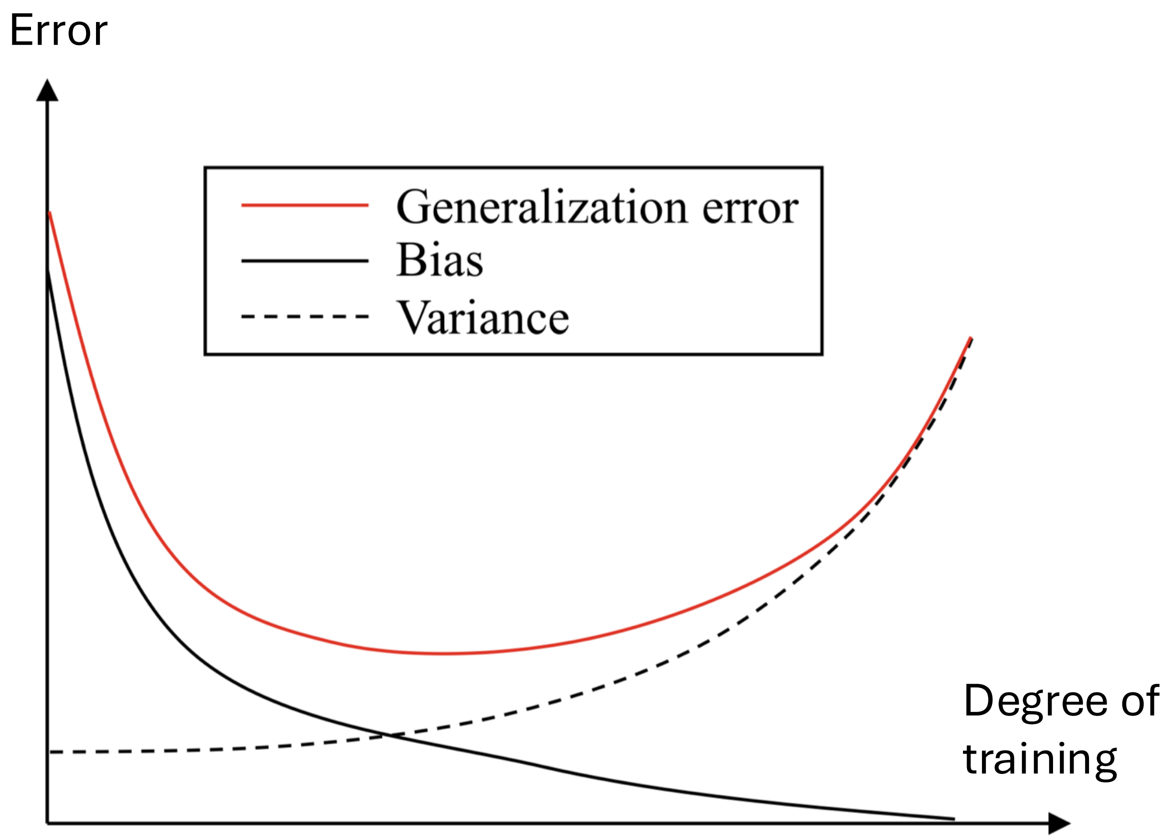
- Definitions:
  - Feature: $x$
  - Label: $y = f(x) + \epsilon$
    - Label generating function: $f(x)$
    - Noise: $\epsilon, E[\epsilon] = 0, Var[\epsilon] = \sigma^2$
  - Prediction: $\hat{y}$
- Bias:
  - $|f(x) - E[\hat{y}]|$
- Variance:
  - $E[(\hat{y} - E[\hat{y}])^2]$
- Generalization error: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

- 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:
  - Decision tree: number of depth
  - Neural network: number of rounds
- Less training:
  - Model fitting is so weak, high bias
- 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

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

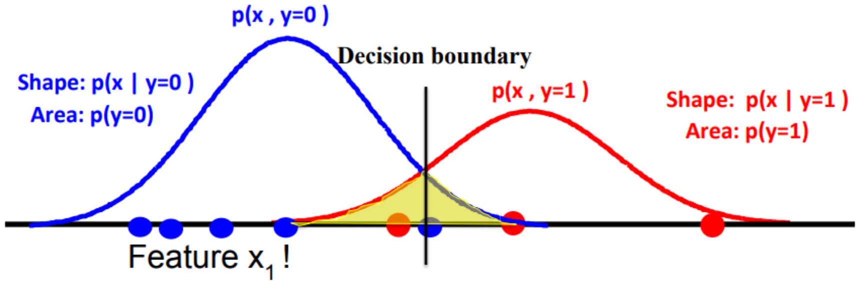
- (Population) Risk function

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

# 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

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

$$\leq \underbrace{\boxed{\mathbb{E}[\hat{R}(\hat{h}) - \hat{R}(h_{\text{ERM}})]}}_{\text{Optimization error}} + \underbrace{\boxed{R(h^*) - R(h_{\text{Bayes}})}}_{\text{Approximation error}} + \underbrace{\boxed{\mathbb{E}[R(\hat{h}) - \hat{R}(\hat{h})]}}_{\text{Generalization error}}$$

# Machine learning can be viewed as a collection of techniques in minimizing the three types of errors

| | **Optimization error** | **Generalization Error** | **Approximation Error** |
|---|---|---|---|
| Definition | $\hat{R}(\hat{h}) - \hat{R}(h_{\mathrm{ERM}})$ | $R(\hat{h}) - \hat{R}(\hat{h})$ | $R(h^*) - R(h_{\mathrm{Bayes}})$ |
| Challenges | • Finding ERM for some loss functions is NP-Hard.<br>• Efficiency isn't enough. Need to be scalable. | • We do not observe Risk!<br>• Don't have infinite data.<br>• Large generalization error ⇔ Overfitting | • Don't know data distribution.<br>• No knowledge of Bayes optimal classifier.<br>• Large approx. error ⇔ Underfitting! |
| **What we have learned to address these challenges?** | | | |

# Machine learning can be viewed as a collection of techniques in minimizing the three types of errors

| | Optimization error | Generalization Error | Approximation Error |
|---|---|---|---|
| Definition | $\hat{R}(\hat{h}) - \hat{R}(h_{\text{ERM}})$ | $R(\hat{h}) - \hat{R}(\hat{h})$ | $R(h^*) - R(h_{\text{Bayes}})$ |
| Challenges | • Finding ERM for some loss functions is NP-Hard.<br>• Efficiency isn't enough. Need to be scalable. | • We do not observe Risk!<br>• Don't have infinite data.<br>• Large generalization error ⇔ Overfitting | • Don't know data distribution.<br>• No knowledge of Bayes optimal classifier.<br>• Large approx. error ⇔ Underfitting! |
| **What we have learned to address these challenges?** | "Just-relax" Surrogate loss, Gradient Descent, SGD | | |

# Machine learning can be viewed as a collection of techniques in minimizing the three types of errors

| | Optimization error | Generalization Error | Approximation Error |
|---|---|---|---|
| Definition | $\hat{R}(\hat{h}) - \hat{R}(h_{\mathrm{ERM}})$ | $R(\hat{h}) - \hat{R}(\hat{h})$ | $R(h^*) - R(h_{\mathrm{Bayes}})$ |
| Challenges | • Finding ERM for some loss functions is NP-Hard.<br>• Efficiency isn't enough. Need to be scalable. | • We do not observe Risk!<br>• Don't have infinite data.<br>• Large generalization error ⇔ Overfitting | • Don't know data distribution.<br>• No knowledge of Bayes optimal classifier.<br>• Large approx. error ⇔ Underfitting! |
| **What we have learned to address these challenges?** | "Just-relax" Surrogate loss, Gradient Descent, SGD | Holdout, Cross-Validation<br>Regularization<br>Statistical learning theory *(not covered)* | |

# Machine learning can be viewed as a collection of techniques in minimizing the three types of errors

| | Optimization error | Generalization Error | Approximation Error |
|---|---|---|---|
| Definition | $\hat{R}(\hat{h}) - \hat{R}(h_{\text{ERM}})$ | $R(\hat{h}) - \hat{R}(\hat{h})$ | $R(h^*) - R(h_{\text{Bayes}})$ |
| Challenges | • Finding ERM for some loss functions is NP-Hard.<br>• Efficiency isn't enough. Need to be scalable. | • We do not observe Risk!<br>• Don't have infinite data.<br>• Large generalization error ⬄ Overfitting | • Don't know data distribution.<br>• No knowledge of Bayes optimal classifier.<br>• Large approx. error ⬄ Underfitting! |
| **What we have learned to address these challenges?** | "Just-relax" Surrogate loss, Gradient Descent, SGD | Holdout, Cross-Validation<br>Regularization<br>Statistical learning theory<br>*(not covered)* | Better features<br>More flexible decision boundaries<br>Better probabilistic models |

# Machine learning can be viewed as a collection of techniques in minimizing the three types of errors

| | Optimization error | Generalization Error | Approximation Error |
|---|---|---|---|
| Definition | $\hat{R}(\hat{h}) - \hat{R}(h_{\mathrm{ERM}})$ | $R(\hat{h}) - \hat{R}(\hat{h})$ | $R(h^*) - R(h_{\mathrm{Bayes}})$ |
| Challenges | • Finding ERM for some loss functions is NP-Hard.<br>• Efficiency isn't enough. Need to be scalable. | • We do not observe Risk!<br>• Don't have infinite data.<br>• Large generalization error ⇔ Overfitting | • Don't know data distribution.<br>• No knowledge of Bayes optimal classifier.<br>• Large approx. error ⇔ Underfitting! |
| **What we have learned to address these challenges?** | "Just-relax" Surrogate loss, Gradient Descent, SGD | Holdout, Cross-Validation<br>Regularization<br>Statistical learning theory<br>*(not covered)* | Better features<br>More flexible decision boundaries<br>Better probabilistic models<br><br>**But how to minimize approx. error automatically?** |

# Machine learning can be viewed as a collection of techniques in minimizing the three types of errors

| | Optimization error | Generalization Error | Approximation Error |
|---|---|---|---|
| Definition | $\hat{R}(\hat{h}) - \hat{R}(h_{\text{ERM}})$ | $R(\hat{h}) - \hat{R}(\hat{h})$ | $R(h^*) - R(h_{\text{Bayes}})$ |
| Challenges | • Finding ERM for some loss functions is NP-Hard.<br>• Efficiency isn't enough. Need to be scalable. | • We do not observe Risk!<br>• Don't have infinite data.<br>• Large generalization error ⇔ Overfitting | • Don't know data distribution.<br>• No knowledge of Bayes optimal classifier.<br>• Large approx. error ⇔ Underfitting! |
| **What we have learned to address these challenges?** | "Just-relax" Surrogate loss, Gradient Descent, SGD | Holdout, Cross-Validation<br>Regularization<br>Statistical learning theory *(not covered)* | Better features<br>More flexible decision boundaries<br>Better probabilistic models<br><br>**But how to minimize approx. error automatically?** |

Often there is a tradeoff.
More **flexible** hypothesis class => smaller approximation error
but larger generalization error (more overfitting) and sometimes harder optimization

# Three main approaches for expanding the hypothesis class (systematically minimizing the approx. error)

- Kernel methods (lift features to higher-dimensional space)
  - e.g., adding polynomial expansion, add interaction terms
  - Other nonlinear transformation of the original features

- Boosting and Bagging (Ensemble learning)
  - Combine many weak learners (e.g., decision trees with depth 3) into a strong learner (e.g., by majority voting…)

- Deep Learning
  - Train large neural networks using SGD
  - Learn feature representation and classification jointly.