Goals for the lecture

you should understand the following concepts

• error decomposition
• bias-variance tradeoff
• PAC learning framework
How to analyze the generalization?

• Key quantity we care in machine learning: the error on the future data points (i.e., the expected error on the whole distribution)

• Divide the analysis of the expected error into steps:
  • What if full information (i.e., infinite data) and full computational power (i.e., can do optimization optimally)?
  • What if finite data but full computational power?
  • What if finite data and finite computational power?

• Example: error decomposition for prediction in supervised learning

Error/risk decomposition

- $h^*$: the optimal function (Bayes classifier)
- $h_{opt}$: the optimal hypothesis on the data distribution
- $\hat{h}_{opt}$: the optimal hypothesis on the training data
- $\hat{h}$: the hypothesis found by the learning algorithm
Error/risk decomposition

\[ \text{err}(\hat{h}) - \text{err}(h^*) = \text{err}(h_{opt}) - \text{err}(h^*) + \text{err}(h_{opt}) - \text{err}(h_{opt}) + \text{err}(\hat{h}) - \text{err}(h_{opt}) \]

Hypothesis class \( H \)
Error/risk decomposition

Approximation error

\[ \text{err}(\hat{h}) - \text{err}(h^*) \]

Estimation error

\[ = \text{err}(h_{opt}) - \text{err}(h^*) \]

Optimization error

\[ + \text{err}(\hat{h}_{opt}) - \text{err}(h_{opt}) \]

\[ + \text{err}(\hat{h}) - \text{err}(\hat{h}_{opt}) \]

“A fundamental theorem of machine learning”
Error/risk decomposition

- approximation error: due to problem modeling (the choice of hypothesis class)

\[ err(\hat{h}) - err(h^*) \]

\[ = err(h_{opt}) - err(h^*) \]

- estimation error: due to finite data

\[ + err(\hat{h}_{opt}) - err(h_{opt}) \]

- optimization error: due to imperfect optimization

\[ + err(\hat{h}) - err(\hat{h}_{opt}) \]
More on estimation error

\[ \text{err}(\hat{h}_{opt}) - \text{err}(h_{opt}) \]

\[ = \text{err}(\hat{h}_{opt}) - \hat{\text{err}}(\hat{h}_{opt}) \]

\[ + \hat{\text{err}}(\hat{h}_{opt}) - \text{err}(h_{opt}) \]

\[ \leq \text{err}(\hat{h}_{opt}) - \hat{\text{err}}(\hat{h}_{opt}) \]

\[ + \hat{\text{err}}(h_{opt}) - \text{err}(h_{opt}) \]

\[ \leq 2 \sup_{h \in H} |\text{err}(h) - \hat{\text{err}}(h)| \]
Another (simpler) decomposition

\[ \text{err}(\hat{h}) = \hat{\text{err}}(\hat{h}) + [\text{err}(\hat{h}) - \hat{\text{err}}(\hat{h})] \]

\[ \leq \hat{\text{err}}(\hat{h}) + \sup_{h \in H} |\text{err}(h) - \hat{\text{err}}(h)| \]

- The training error \( \hat{\text{err}}(\hat{h}) \) is what we can compute
- Need to control the generalization gap
Bias-Variance Tradeoff
Defining bias and variance

• consider the task of learning a regression model given a training set $D = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$

• a natural measure of the error of $f$ is

$$E[(y - f(x; D))^2 | D]$$

where the expectation is taken with respect to the real-world distribution of instances
Defining bias and variance

• further consider a fixed \( x \)
• this can be rewritten as:

\[
E \left[ (y - f(x; D))^2 \mid x, D \right] = E \left[ (y - E[y \mid x])^2 \mid x, D \right] + (f(x; D) - E[y \mid x])^2
\]

- **error of** \( f \) as a predictor of \( y \)
- **noise**: variance of \( y \) given \( x \); doesn't depend on \( D \) or \( f \)
Defining bias and variance

- bias: if on average $f(x; D)$ differs from $E[y | x]$ then $f(x; D)$ is a biased estimator of $E[y | x]$

\[ E_D \left[ (f(x; D) - E[y | x])^2 \right] = \]

\[ (E_D[f(x; D)] - E[y | x])^2 \]

\[ + E_D \left[ (f(x; D) - E_D[f(x; D)])^2 \right] \]

bias

variance

- variance: $f(x; D)$ may be sensitive to $D$ and vary a lot from its expected value
Bias/variance for polynomial interpolation

- the 1st order polynomial has high bias, low variance
- 50th order polynomial has low bias, high variance
- 4th order polynomial represents a good trade-off
Bias/variance trade-off for \( k \)-NN regression

- consider using \( k \)-NN regression to learn a model of this surface in a 2-dimensional feature space
Bias/variance trade-off for k-NN regression

darker pixels correspond to higher values
Bias/variance trade-off

- consider $k$-NN applied to digit recognition
Bias/variance discussion

- predictive error has two controllable components
  - expressive/flexible learners reduce *bias*, but increase *variance*

- for many learners we can trade-off these two components (e.g. via our selection of $k$ in $k$-NN)

- the optimal point in this trade-off depends on the particular problem domain and training set size

- this is not necessarily a strict trade-off; e.g. with ensembles we can often reduce bias and/or variance without increasing the other term
Bias/variance discussion

the bias/variance analysis

• helps explain why simple learners can outperform more complex ones
• helps understand and avoid overfitting
PAC Learning Theory
PAC learning

• Overfitting happens because training error is a poor estimate of generalization error
  → Can we infer something about generalization error from training error?

• Overfitting happens when the learner doesn’t see enough training instances
  → Can we estimate how many instances are enough?
Learning setting

- set of instances $\mathcal{X}$
- set of hypotheses (models) $H$
- set of possible target concepts $C$
- unknown probability distribution $\mathcal{D}$ over instances
Learning setting

• learner is given a set $D$ of training instances $\langle x, c(x) \rangle$ for some target concept $c$ in $C$
  • each instance $x$ is drawn from distribution $\mathcal{D}$
  • class label $c(x)$ is provided for each $x$

• learner outputs hypothesis $h$ modeling $c$
the true error of hypothesis $h$ refers to how often $h$ is wrong on future instances drawn from $\mathcal{D}$

$$\text{error}_\mathcal{D}(h) \equiv P_{x \in \mathcal{D}} \left[ c(x) \neq h(x) \right]$$

instance space $\mathcal{X}$

![Diagram showing instance space $\mathcal{X}$ with hypotheses $c$ and $h$, and positive (+) and negative (-) instances.](image-url)
the *training error* of hypothesis \( h \) refers to how often \( h \) is wrong on instances in the training set \( D \)

\[
error_D(h) \equiv P_{x \in D}[c(x) \neq h(x)] = \frac{\sum_{x \in D} \delta(c(x) \neq h(x))}{|D|}
\]

Can we bound \( error_D(h) \) in terms of \( error_D(h) \)?
What’s successful learning?

To say that our learner $L$ has learned a concept, should we require $\text{error}_D(h) = 0$?

this is not realistic:

• unless we’ve seen every possible instance, there may be multiple hypotheses that are consistent with the training set
• there is some chance our training sample will be unrepresentative
Instead, we’ll require that

- the error of a learned hypothesis $h$ is bounded by some constant $\varepsilon$
- the probability of the learner failing to learn an accurate hypothesis is bounded by a constant $\delta$
Probably Approximately Correct (PAC) learning [Valiant, CACM 1984]

- Consider a class $C$ of possible target concepts defined over a set of instances $\mathcal{X}$ of length $n$, and a learner $L$ using hypothesis space $H$.

- $C$ is PAC learnable by $L$ using $H$ if, for all $c \in C$ distributions $\mathcal{D}$ over $\mathcal{X}$ $\varepsilon$ such that $0 < \varepsilon < 0.5$ $\delta$ such that $0 < \delta < 0.5$

- learner $L$ will, with probability at least $(1-\delta)$, output a hypothesis $h \in H$ such that $error_{\mathcal{D}}(h) \leq \varepsilon$ in time that is polynomial in:
  - $1/\varepsilon$
  - $1/\delta$
  - $n$
  - size($c$)
THANK YOU

Some of the slides in these lectures have been adapted/borrowed from materials developed by Mark Craven, David Page, Jude Shavlik, Tom Mitchell, Nina Balcan, Elad Hazan, Tom Dietterich, and Pedro Domingos.