Introduction to Learning Theory Part 1

CS 760@UW-Madison

Goals for the lecture

you should understand the following concepts

- error decomposition
- bias-variance tradeoff
- PAC learning framework

Error Decomposition

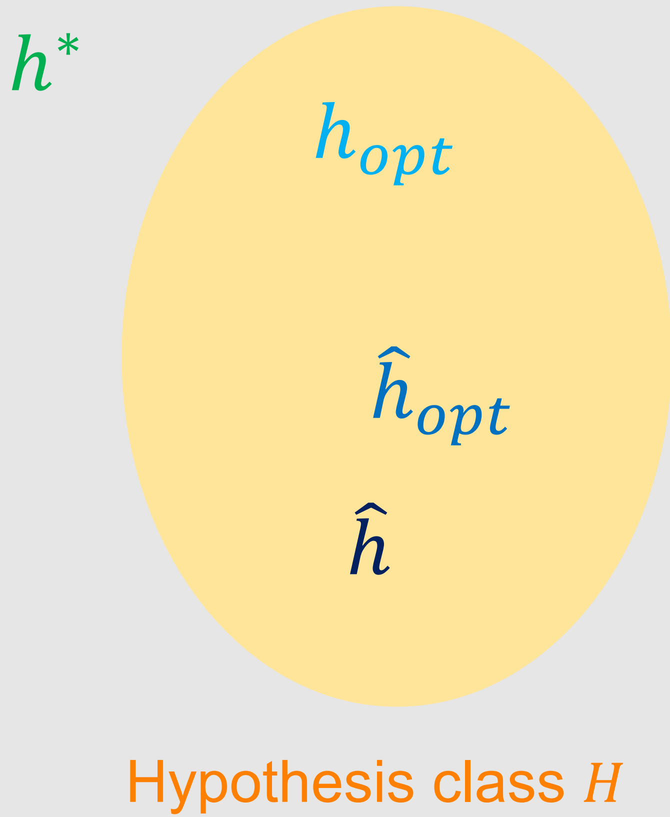
How to analyze the generalization?

- 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

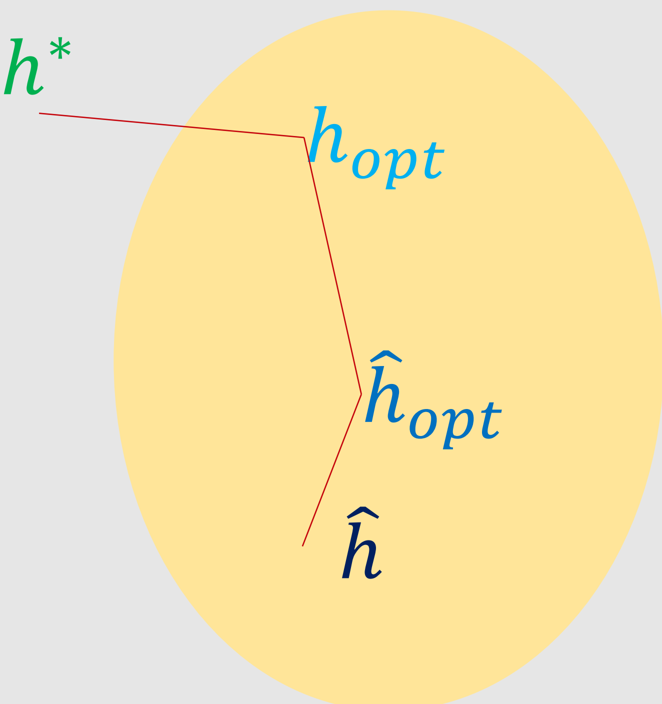
  Bottou, Léon, and Olivier Bousquet. "The tradeoffs of large scale learning." *Advances in neural information processing systems*. 2008.

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



Hypothesis class H

$$
\begin{aligned}
& err(\hat{h}) - err(h^*) \\
&= err(h_{opt}) - err(h^*) \\
&+ err(\hat{h}_{opt}) - err(h_{opt}) \\
&+ err(\hat{h}) - err(\hat{h}_{opt})
\end{aligned}
$$

Error/risk decomposition

$$err(\hat{h}) - err(h^*)$$

Approximation error

$$= err(h_{opt}) - err(h^*)$$

Estimation error

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

Optimization error

$$+ err(\hat{h}) - err(\hat{h}_{opt})$$

"A fundamental theorem of machine learning"

Error/risk decomposition

- approximation error: due to problem modeling (the choice of hypothesis class)
- estimation error: due to finite data
- optimization error: due to imperfect optimization

$$err(\hat{h}) - err(h^*)$$

$$= err(h_{opt}) - err(h^*)$$

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

$$+ err(\hat{h}) - err(\hat{h}_{opt})$$

More on estimation error

$$err(\hat{h}_{opt}) - err(h_{opt})$$

$$= err(\hat{h}_{opt}) - \widehat{err}(\hat{h}_{opt})$$

$$+ \widehat{err}(\hat{h}_{opt}) - err(h_{opt})$$

$$\leq err(\hat{h}_{opt}) - \widehat{err}(\hat{h}_{opt})$$

$$+ \widehat{err}(h_{opt}) - err(h_{opt})$$

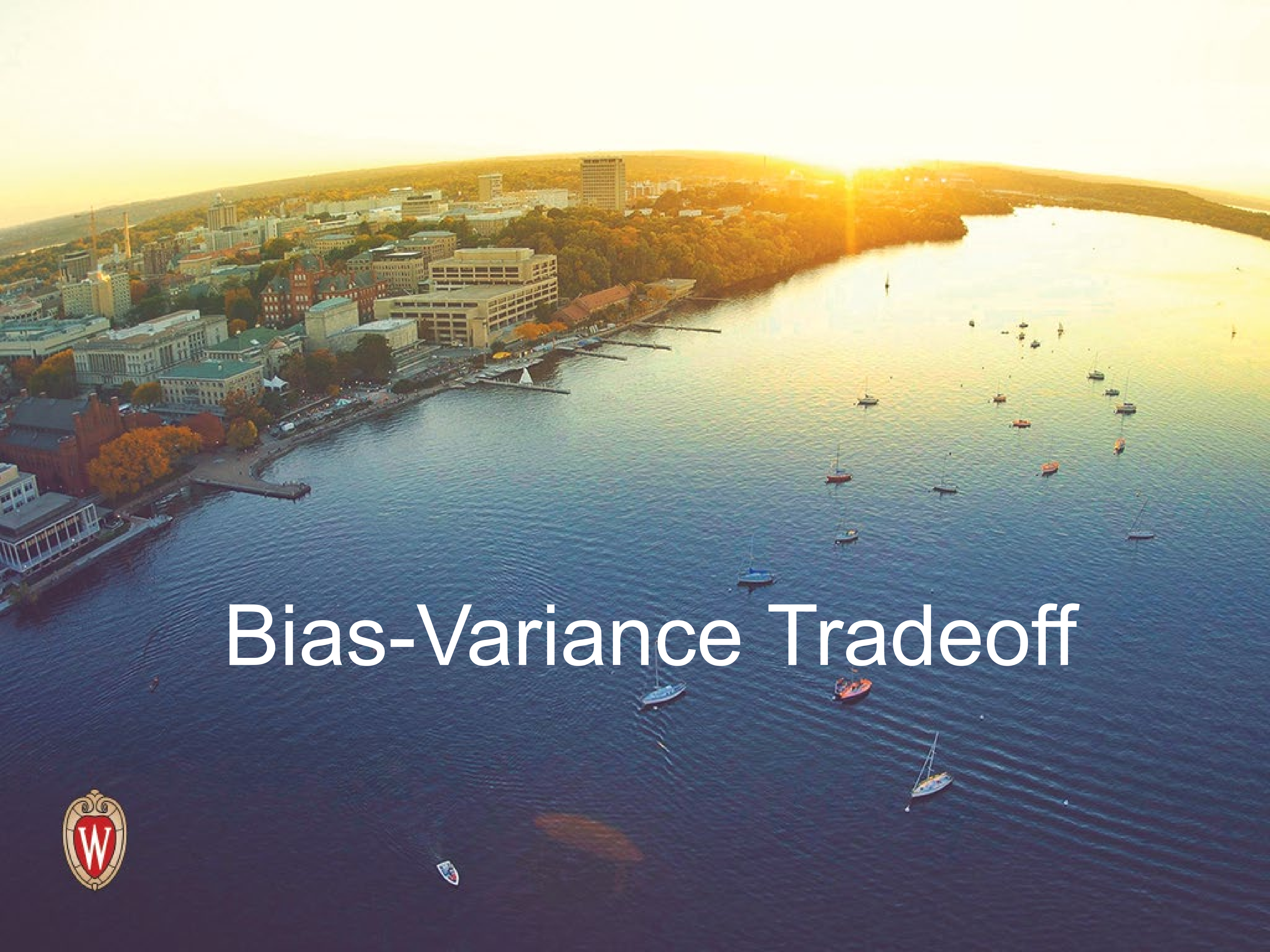
$$\leq 2 \sup_{h \in H} |err(h) - \widehat{err}(h)|$$

Another (simpler) decomposition

$$err(\hat{h}) = \widehat{err}(\hat{h}) + \underbrace{\left[err(\hat{h}) - \widehat{err}(\hat{h})\right]}_{\text{Generalization gap}}$$

$$\leq \widehat{err}(\hat{h}) + \sup_{h \in H} |err(h) - \widehat{err}(h)|$$

- The training error $\widehat{err}(\hat{h})$ 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 $f(\boldsymbol{x};\, D)$ given a training set $D = \left\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\right\}$

indicates the dependency of model on D

- a natural measure of the error of f is

$$E\left[(y - f(\mathbf{x};\ D))^2 | D\right]$$

where the expectation is taken with respect to the real-world distribution of instances

Defining bias and variance

- further consider a fixed $\mathbf{x}$
- this can be rewritten as:

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

<u>error</u> of f as a predictor of y

<u>noise</u>: variance of y given $\boldsymbol{x}$; doesn't depend on D or f

Defining bias and variance

- now consider the expectation (over different data sets D) for the second term

$$E_D\left[\left(f(\boldsymbol{x}; D) - E[y \mid \boldsymbol{x}]\right)^2\right] =$$

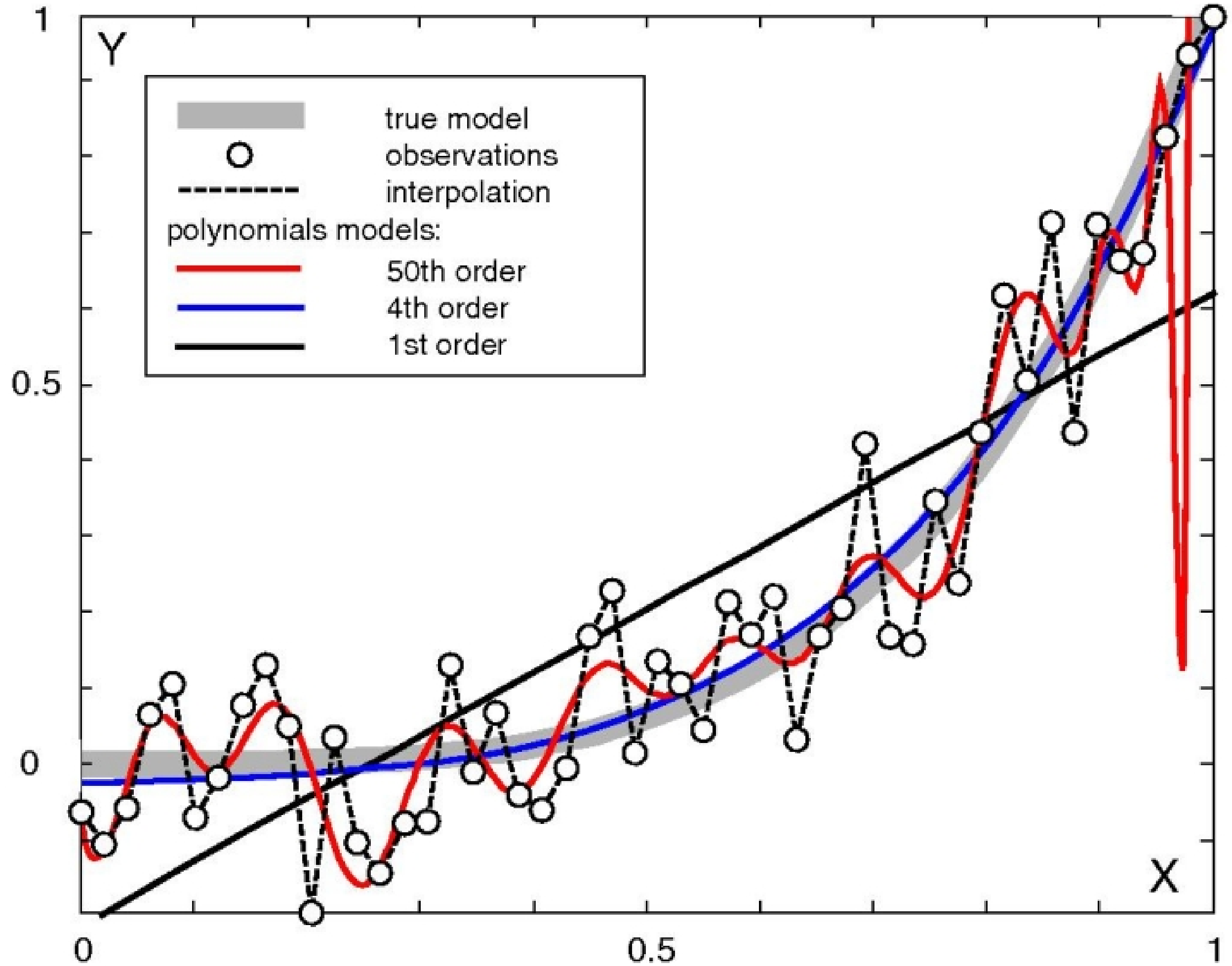
$$\left(E_D[f(\boldsymbol{x}; D)] - E[y \mid \boldsymbol{x}]\right)^2 \quad \text{bias}$$

$$+\, E_D\left[\left(f(\boldsymbol{x}; D) - E_D[f(\boldsymbol{x}; D)]\right)^2\right] \quad \text{variance}$$

- bias: if on average $f(\boldsymbol{x}; D)$ differs from $E[y \mid \boldsymbol{x}]$ then $f(\boldsymbol{x}; D)$ is a biased estimator of $E[y \mid \boldsymbol{x}]$
- variance: $f(\boldsymbol{x}; D)$ may be sensitive to D and vary a lot from its expected value

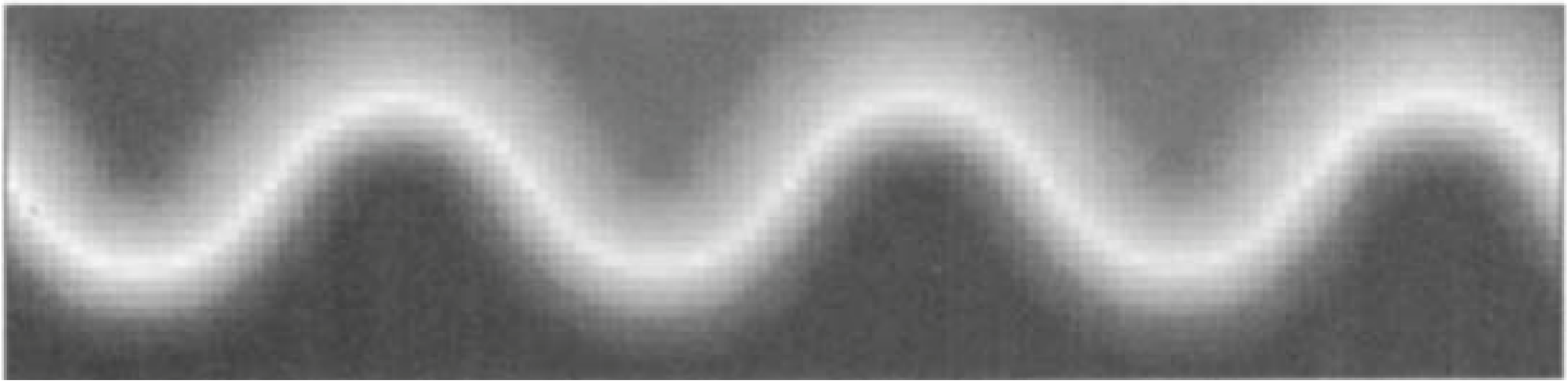
Bias/variance for polynomial interpolation

- 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



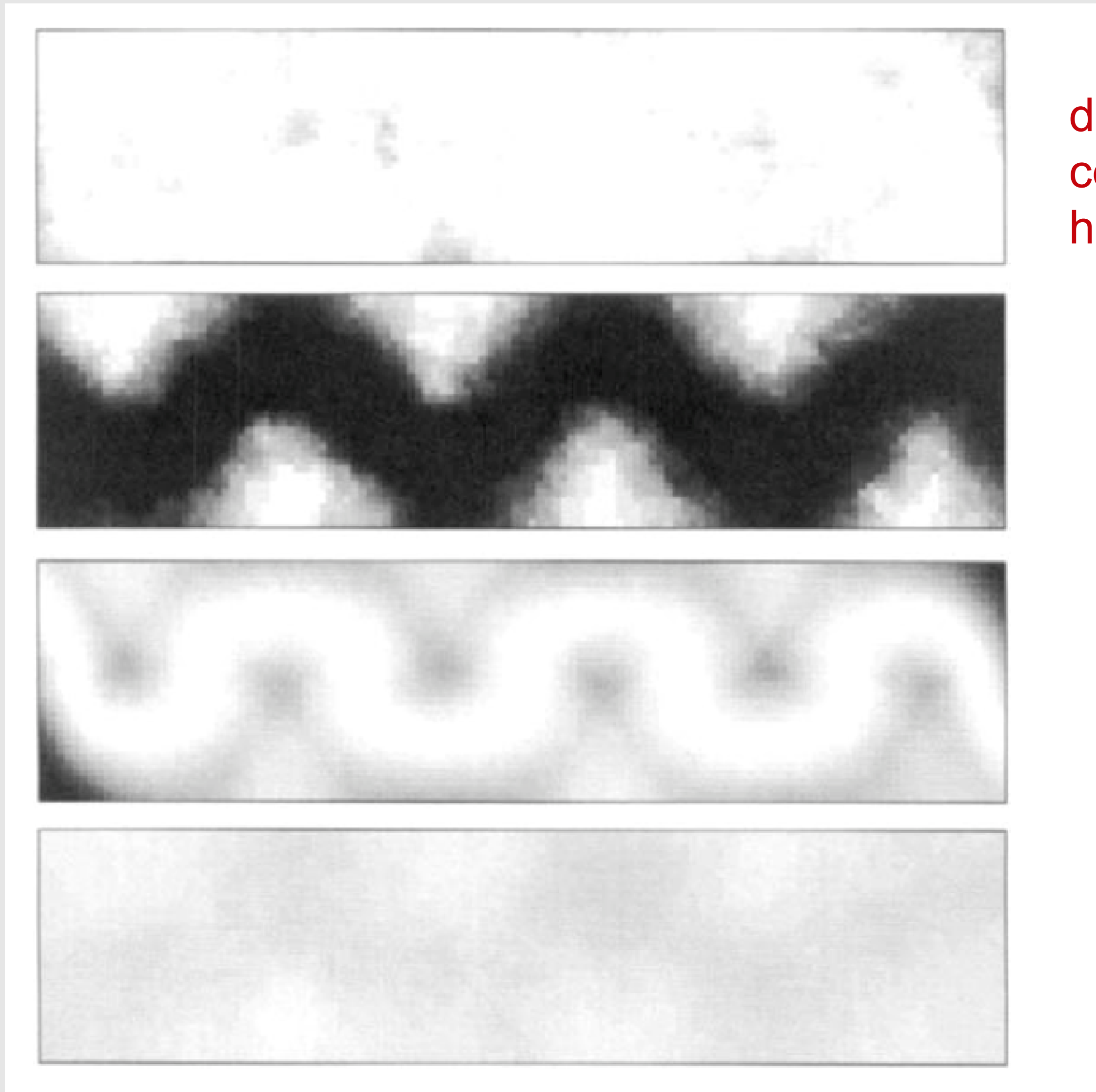
Bias/variance trade-off for k-NN regression

bias for 1-NN

variance for 1-NN

bias for 10-NN

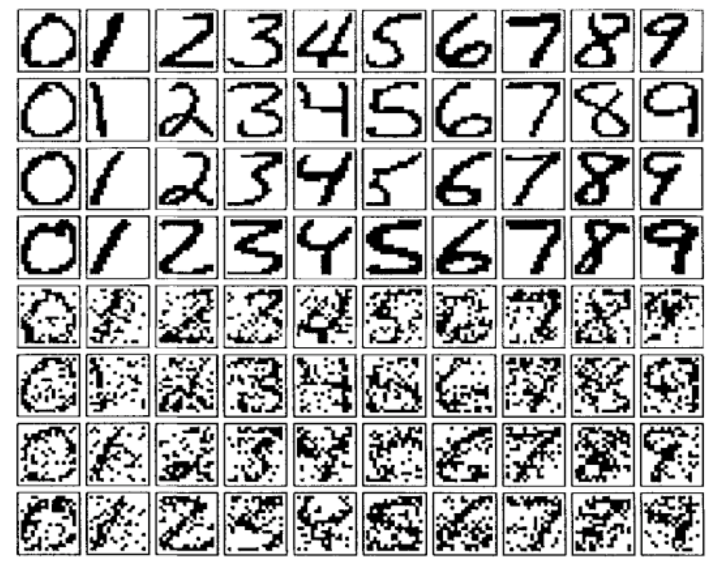
variance for 10-NN

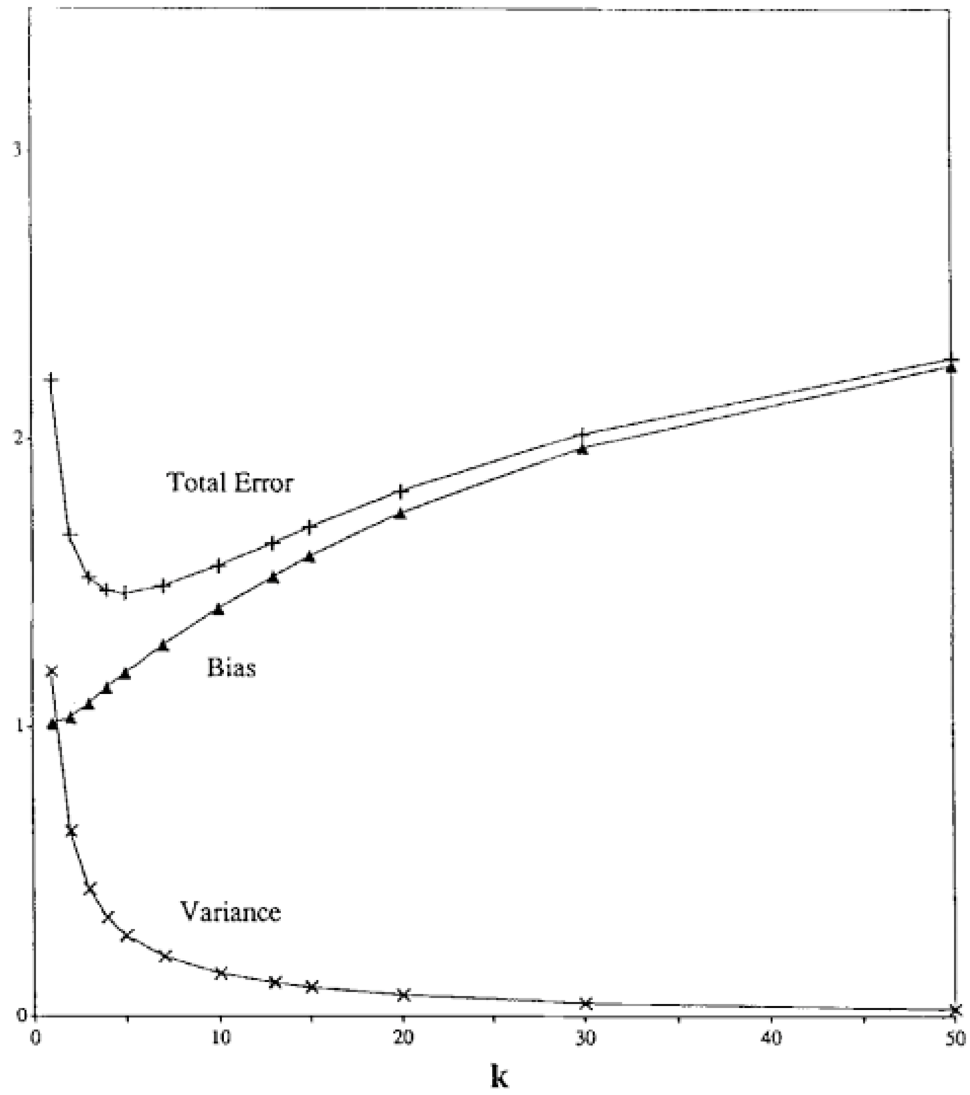


darker pixels correspond to higher values

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)

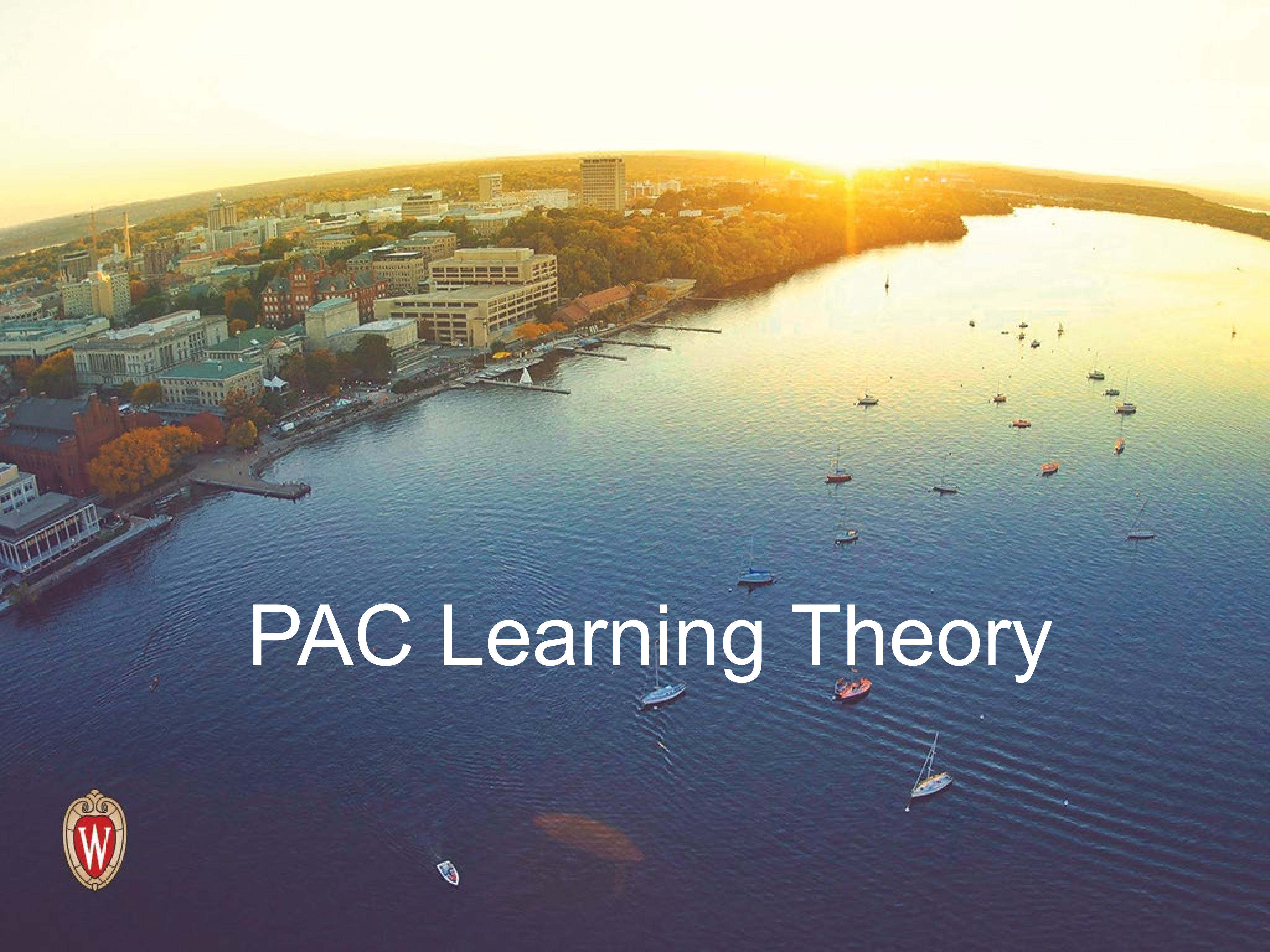
- the optimal point in this trade-off depends on the particular problem domain and training set size

- 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



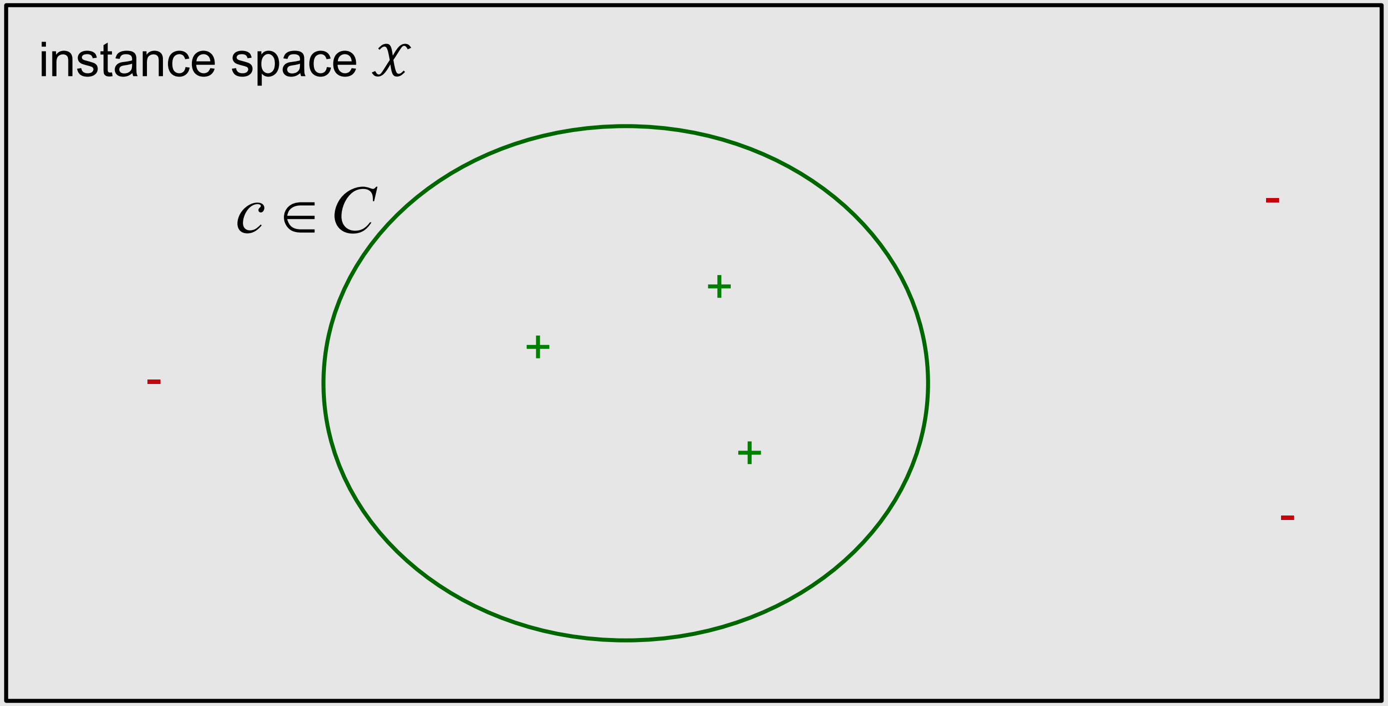
PAC Learning Theory

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

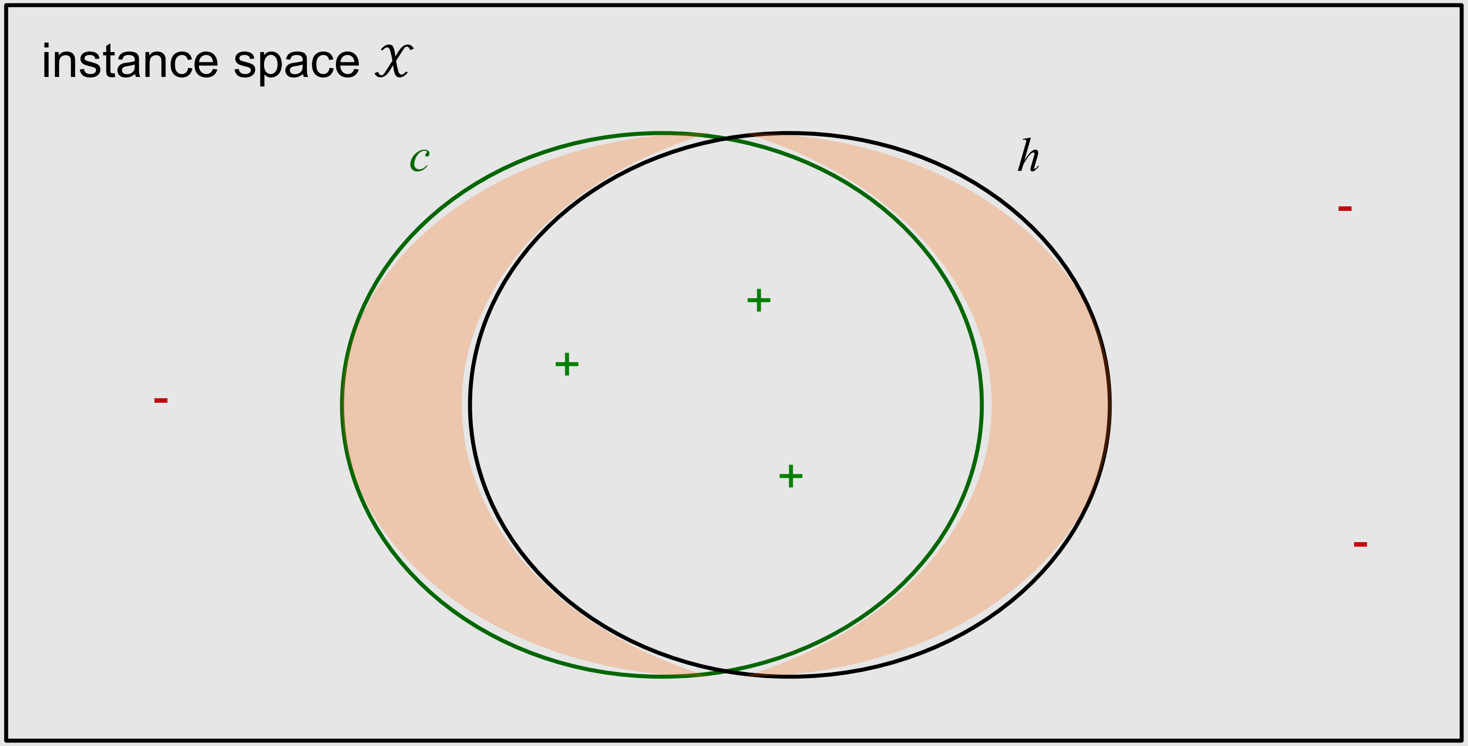
- learner is given a set D of training instances $\langle\, \boldsymbol{x}, c(\boldsymbol{x}) \,\rangle$ for some target concept c in C
  - each instance $\boldsymbol{x}$ is drawn from distribution $\mathcal{D}$
  - class label $c(\boldsymbol{x})$ is provided for each $\boldsymbol{x}$

- learner outputs hypothesis h modeling c

True error of a hypothesis

the *true error* of hypothesis h refers to how often h is wrong on future instances drawn from $\mathcal{D}$

$$error_{\mathcal{D}}(h) \equiv P_{x \in \mathcal{D}}\left[c(\boldsymbol{x}) \neq h(\boldsymbol{x})\right]$$



Training error of a hypothesis

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 $error_{\mathcal{D}}(h)$ in terms of $error_{\mathrm{D}}(h)$?

What's successful learning?

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

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

Probably approximately correct learning?

Instead, we'll require that

- the error of a learned hypothesis h is bounded by some constant ε
- the probability of the learner failing to learn an accurate hypothesis is bounded by a constant δ

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 $\mathcal{X}$

  ε such that $0 < \varepsilon < 0.5$

  δ such that $0 < \delta < 0.5$

- learner L will, with probability at least $(1\text{-}\delta)$, 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.