Machine Learning (CS 181): 4. Model Selection and Regularization

David Parkes and Sasha Rush

Spring 2017

Contents

Contents

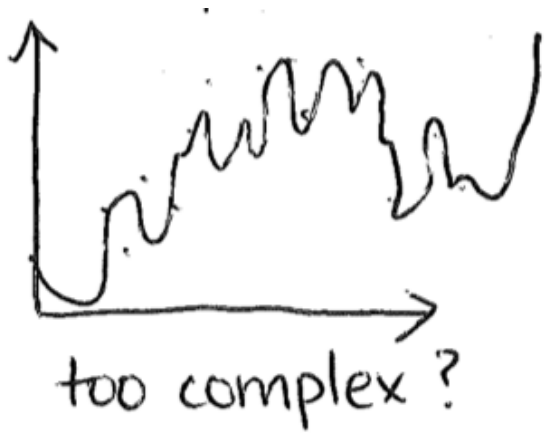
Overview: Model Selection

- Today: non-Bayesian approaches, next lecture Bayesian approaches
- We've already discussed many different models:
  - linear regression with different features ('basis functions')
  - k-nearest neighbors
  - there are many more (e.g. neural nets, decision trees, random forests, SVMs)

What's a good model to use?

Naive approach

- Test each model on data D. Select model with lowest loss. Problem: may overfit the data:



Overfitting can occur when:

- insufficient data (all dogs black, all cats white)
- there is noise in the data
- irrelevant features (e.g., if teacher's socks are red then it will rain)

Example of Spurious Patterns

Suppose 2000 features, each value 0 or 1, at random w.p. 0.5.

True concept: $g(\mathbf{x}) = x_1$. 8 examples. Then:

- Feature x_2 explains the 8 examples with probability $2(1/2)^8 = 1/128$ (via $h(\mathbf{x}) = x_2$ or $h(\mathbf{x}) = 1 - x_2$).
- Prob that none of 1999 spurious features explain data is:

$$1 - \left(\frac{127}{128}\right)^{1999} \approx 1$$

$\Rightarrow$ learner very likely to choose an incorrect model. Be careful about irrelevant features!

Example of Spurious Patterns

Suppose 2000 features, each value 0 or 1, at random w.p. 0.5.

True concept: $g(\mathbf{x}) = x_1$. 8 examples. Then:

- Feature x_2 explains the 8 examples with probability $2(1/2)^8 = 1/128$ (via $h(\mathbf{x}) = x_2$ or $h(\mathbf{x}) = 1 - x_2$).
- Prob that none of 1999 spurious features explain data is:

$$1 - \left(\frac{127}{128}\right)^{1999} \approx 1$$

$\Rightarrow$ learner very likely to choose an incorrect model. Be careful about irrelevant features!

Example of Spurious Patterns

Suppose 2000 features, each value 0 or 1, at random w.p. 0.5.

True concept: $g(\mathbf{x}) = x_1$. 8 examples. Then:

- Feature x_2 explains the 8 examples with probability $2(1/2)^8 = 1/128$ (via $h(\mathbf{x}) = x_2$ or $h(\mathbf{x}) = 1 - x_2$).
- Prob that none of 1999 spurious features explain data is:

$$1 - \left(\frac{127}{128}\right)^{1999} \approx 1$$

$\Rightarrow$ learner very likely to choose an incorrect model. Be careful about irrelevant features!

Example of Spurious Patterns

Suppose 2000 features, each value 0 or 1, at random w.p. 0.5.

True concept: $g(\mathbf{x}) = x_1$. 8 examples. Then:

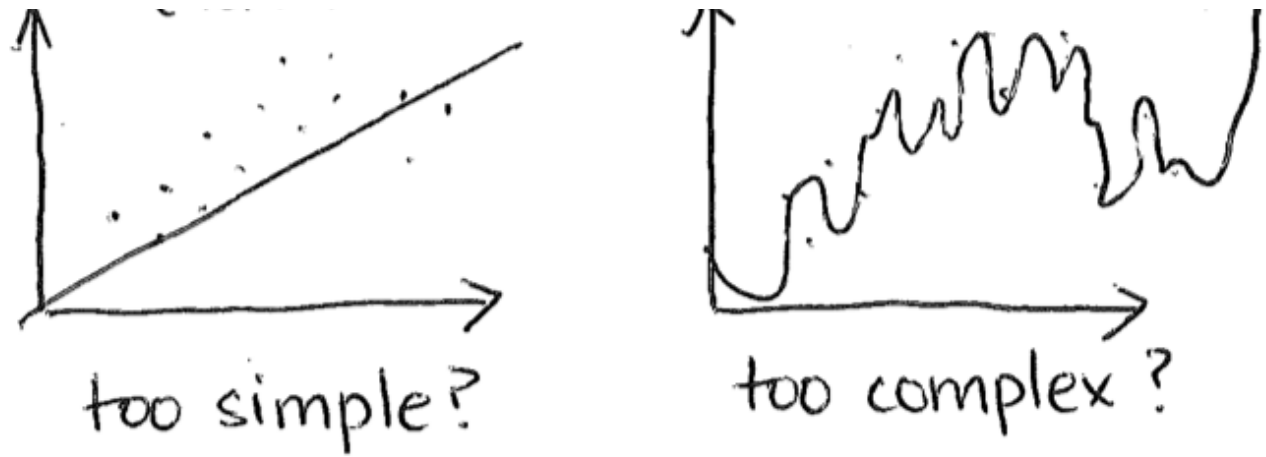
- Feature x_2 explains the 8 examples with probability $2(1/2)^8 = 1/128$ (via $h(\mathbf{x}) = x_2$ or $h(\mathbf{x}) = 1 - x_2$).
- Prob that none of 1999 spurious features explain data is:

$$1 - \left(\frac{127}{128}\right)^{1999} \approx 1$$

$\Rightarrow$ learner very likely to choose an incorrect model. Be careful about irrelevant features!

What's a Good Model?

We want a model that generalizes well, rather than a model that predicts training data well.

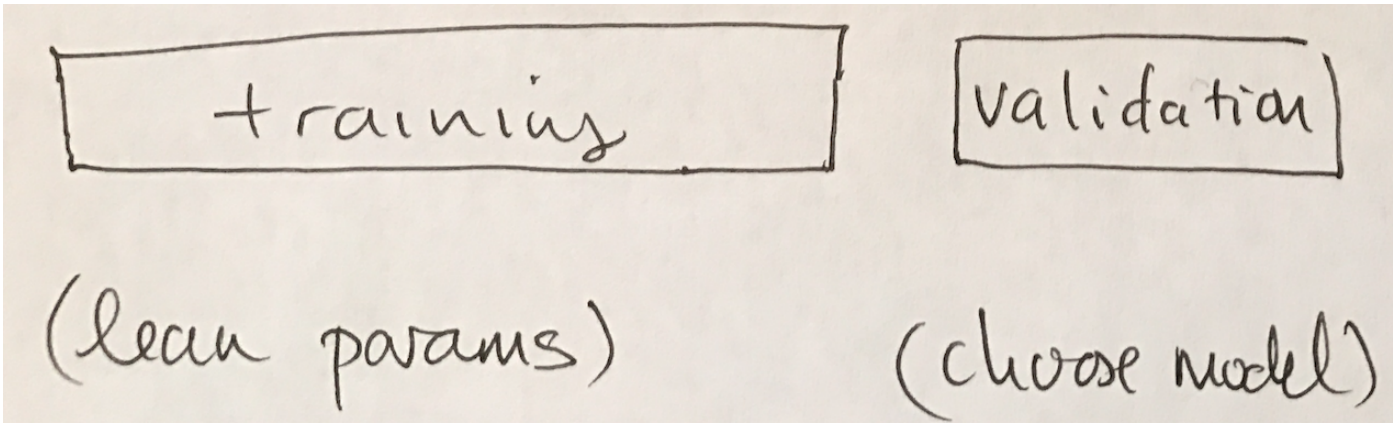


Avoid both overfitting and underfitting. One of the main challenges in machine learning.

Contents

Fix: Use a Validation Set

Divide the data into a training set and a validation set:



- Learn parameters for each model on training set (training error)
- Measure prediction error on validation set (validation error)
- Choose the model with the best validation error.

Reporting the Generalization Error

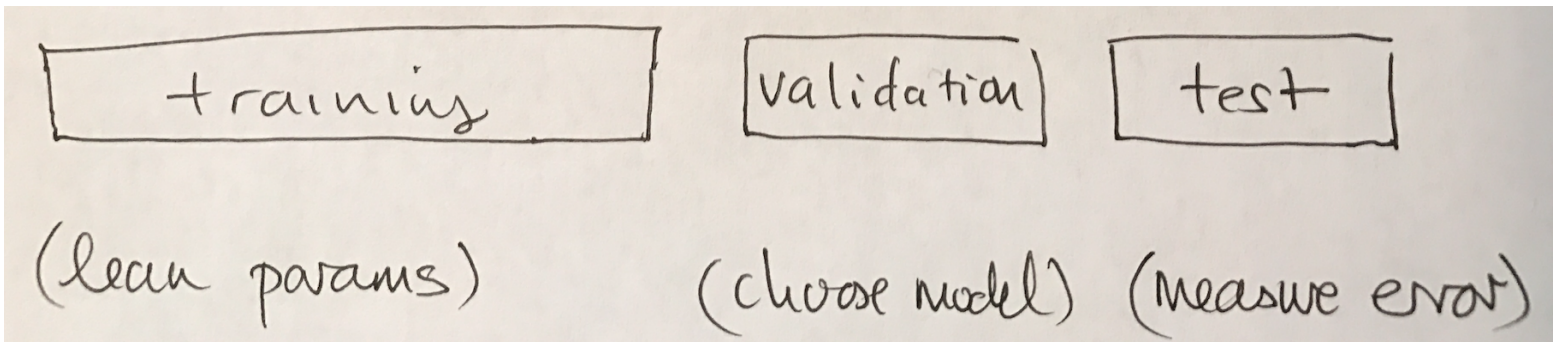
- We care about the prediction accuracy on unseen data. The generalization error.
- The validation error gives a biased estimate of the generalization error of the selected model.
- Why?

Reporting the Generalization Error

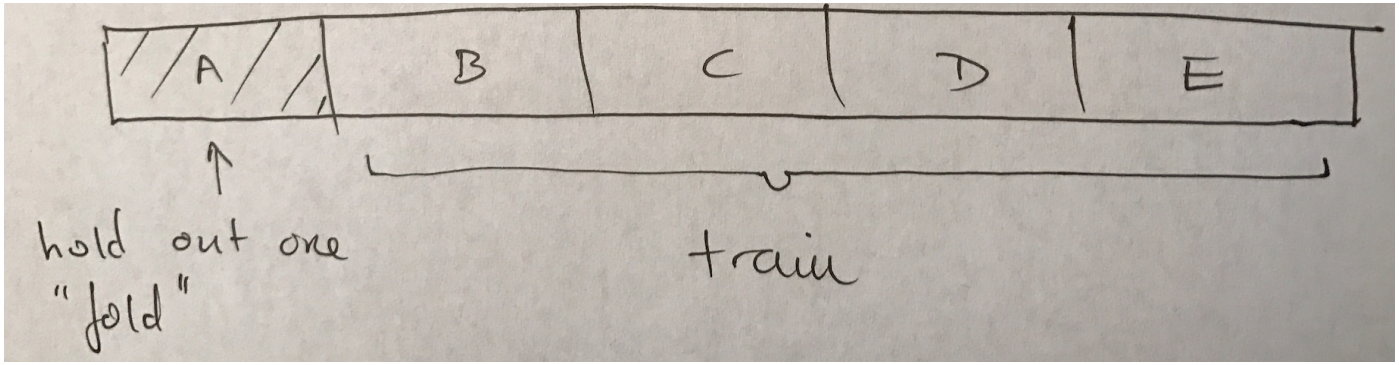
- We care about the prediction accuracy on unseen data. The generalization error.
- The validation error gives a biased estimate of the generalization error of the selected model.
- Peeking: we've used the data on which the error is reported to select the best model.

Reporting the Generalization Error

- We care about the prediction accuracy on unseen data. The generalization error.
- The validation error gives a biased estimate of the generalization error of the selected model.
- Peeking: we used the data on which the error is reported to select the best model.
- Fix: construct a test set. Only use this for measuring test error:

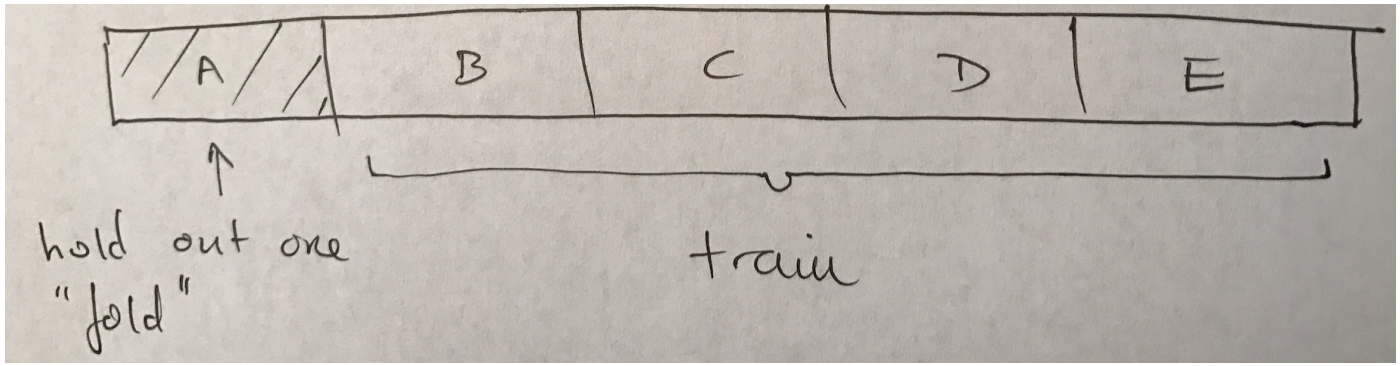


Extension: k-fold Cross Validation



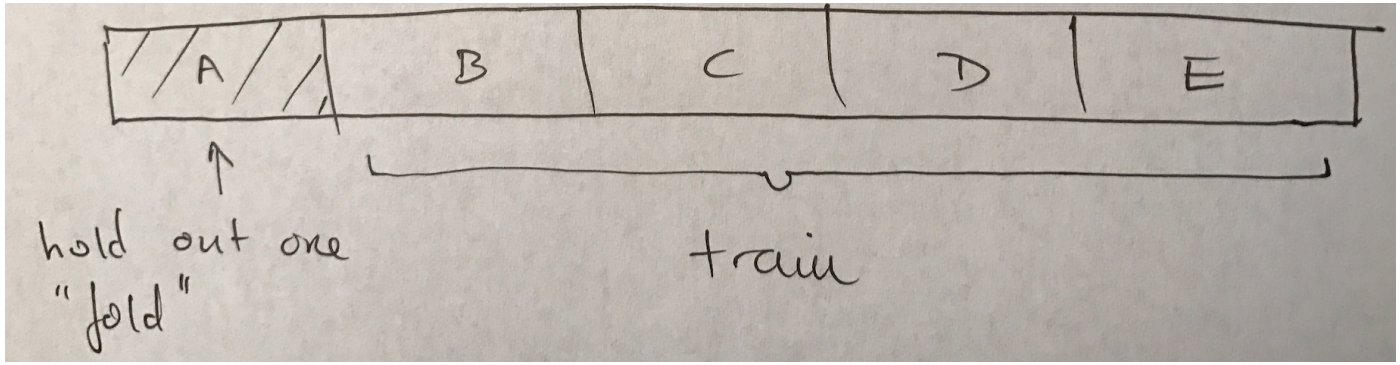
- Repeat, "holding out" one fold at a time (e.g., with $k = 5$ or $k = 10$ 'experiments.')
- e.g., train BCDE / validate A; train ACDE / validate on B; etc. Measure average validation error.
- Standard methodology when insufficient data, even though experiments not independent.
- Variations: leave-one-out CV (repeat n times); random CV (choose n data points at random with replacement).

Extension: k-fold Cross Validation



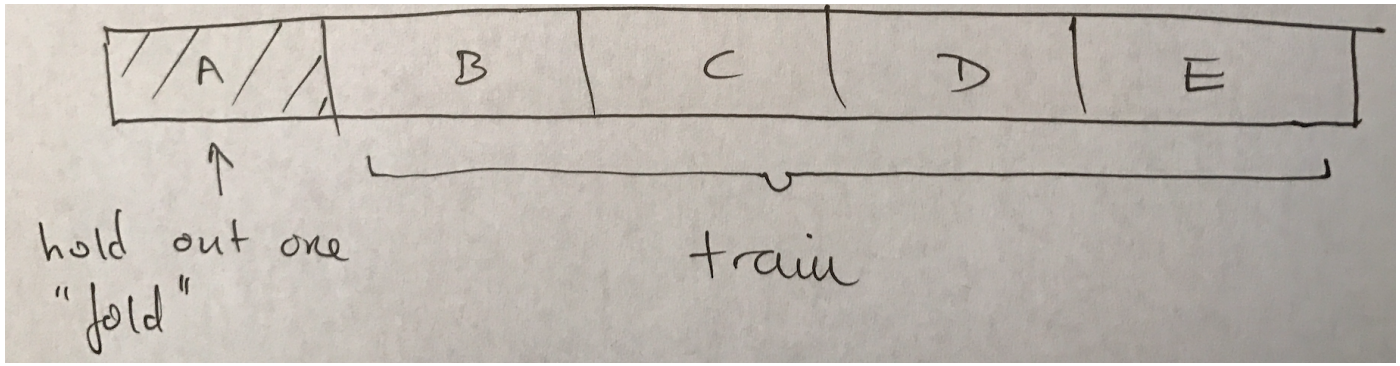
- Repeat, "holding out" one fold at a time (e.g., with $k = 5$ or $k = 10$ 'experiments.')
- e.g., train BCDE / validate A; train ACDE / validate on B; etc. Measure average validation error.
- Standard methodology when insufficient data, even though experiments not independent.
- Variations: leave-one-out CV (repeat n times); random CV (choose n data points at random with replacement).

Extension: k-fold Cross Validation



- Repeat, "holding out" one fold at a time (e.g., with $k = 5$ or $k = 10$ 'experiments.')
- e.g., train BCDE / validate A; train ACDE / validate on B; etc. Measure average validation error.
- Standard methodology when insufficient data, even though experiments not independent.
- Variations: leave-one-out CV (repeat n times); random CV (choose n data points at random with replacement).

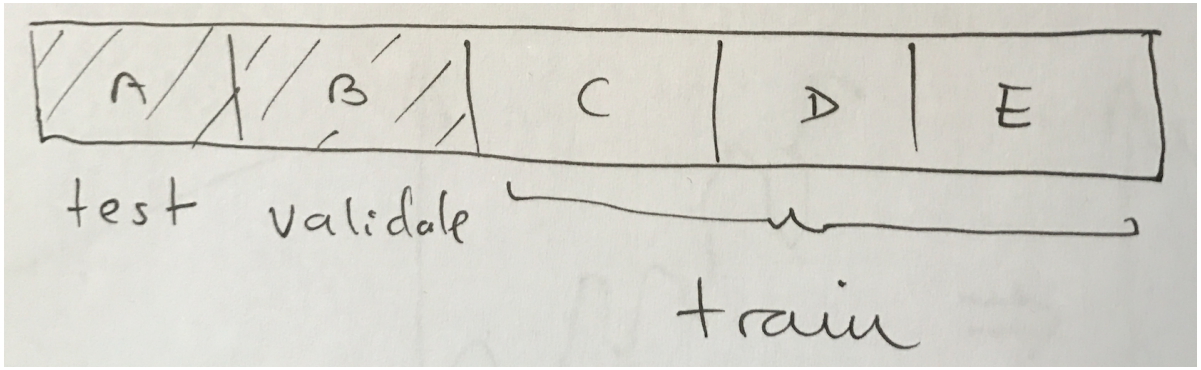
Extension: k-fold Cross Validation



- Repeat, "holding out" one fold at a time (e.g., with $k = 5$ or $k = 10$ 'experiments.')
- e.g., train BCDE / validate A; train ACDE / validate on B; etc. Measure average validation error.
- Standard methodology when insufficient data, even though experiments not independent.
- Variations: leave-one-out CV (repeat n times); random CV (choose n data points at random with replacement).

Extension: Introducing a Test Set into CV

- Hold out 2 folds, one for validation and one for testing.



- With $k = 5$:
  - train CDE, validate B, test A
  - train ADE, validate C, test B
  - ...
  - train BCD, validate A, test E

Choose model based on avg validation error, report avg test error.

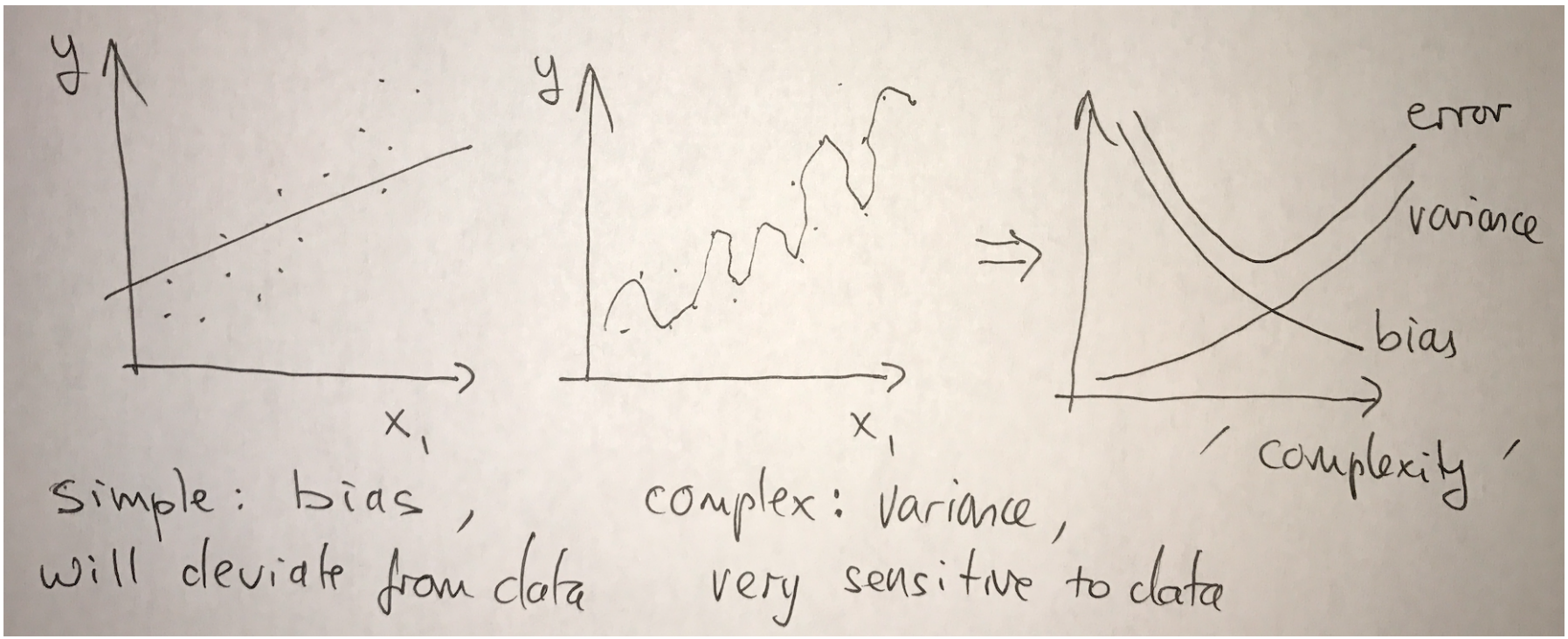
Contents

Understanding Generalization Error

- Why do some models have better or worse generalization error?
- How can we change our training procedure so as not to overfit (i.e., to get good generalization performance without cross-validation?)

Understanding Generalization Error

- Why do some models have better or worse generalization error?
- How can we change our training procedure so as not to overfit (i.e., to get good generalization performance without cross-validation?)
- Answer: make the right bias-variance tradeoff



The Bias-Variance Decomposition

$$\textit{generalization error} = \underbrace{\textit{systematic error}}_{\text{bias}} + \underbrace{\textit{sensitivity of prediction}}_{\text{variance}}$$

- Simple models under-fit: will deviate from data (high bias) but will not be influenced by peculiarities of data (low variance).
- Complex models over-fit: will not deviate systematically from data (low bias) but will be very sensitive to data (high variance).

The right tradeoff between bias and variance depends on the amount of data. More data, can use more complex models.

The Bias-Variance Decomposition

$$\textit{generalization error} = \underbrace{\textit{systematic error}}_{\text{bias}} + \underbrace{\textit{sensitivity of prediction}}_{\text{variance}}$$

- Simple models under-fit: will deviate from data (high bias) but will not be influenced by peculiarities of data (low variance).
- Complex models over-fit: will not deviate systematically from data (low bias) but will be very sensitive to data (high variance).

The right tradeoff between bias and variance depends on the amount of data. More data, can use more complex models.

Bias-Variance: Analysis (1 of 4)

- Define the trained model $h_D : \mathcal{X} \mapsto \mathbb{R}$.
  - Data D is a random variable, sampled $D \sim F^n$ (for distr. F).
  - Leave parameters implicit
- Consider some $\mathbf{x}$. True y is a random variable.

We're interested in the generalization error at $\mathbf{x}$:

$$\mathbb{E}[(y - h_D(\mathbf{x}))^2],$$

where the expectation is taken wrt D and y.

Bias-Variance: Analysis (1 of 4)

- Define the trained model $h_D : \mathcal{X} \mapsto \mathbb{R}$.
  - Data D is a random variable, sampled $D \sim F^n$ (for distr. F).
  - Leave parameters implicit
- Consider some $\mathbf{x}$. True y is a random variable.

We're interested in the generalization error at $\mathbf{x}$:

$$\mathbb{E}[(y - h_D(\mathbf{x}))^2],$$

where the expectation is taken wrt D and y.

Bias-Variance: Analysis (2 of 4)

- Define the true conditional mean, $\overline{y} = \mathbb{E}[y]$.

The generalization error at $\mathbf{x}$ is:

$$\mathbb{E}[(y - h_D(\mathbf{x}))^2] = \mathbb{E}[(y - \overline{y} + \overline{y} - h_D(\mathbf{x}))^2]$$

$$= \underbrace{\mathbb{E}[(y - \overline{y})^2]}_{\text{noise}} + \underbrace{\mathbb{E}[(\overline{y} - h_D(\mathbf{x}))^2]}_{\text{bias+var}} + \underbrace{2\mathbb{E}[(y - \overline{y})(\overline{y} - h_D(\mathbf{x}))]}_{0} \quad (1)$$

The last term can be written as

$$2\mathbb{E}_D[\overline{y} - h_D(\mathbf{x})] \cdot \mathbb{E}_{y|\mathbf{x}}[y - \overline{y}]] = 2\mathbb{E}_D[\overline{y} - h_D(\mathbf{x})] \cdot 0 = 0.$$

Bias-Variance: Analysis (2 of 4)

- Define the true conditional mean, $\overline{y} = \mathbb{E}[y]$.

The generalization error at $\mathbf{x}$ is:

$$\mathbb{E}[(y - h_D(\mathbf{x}))^2] = \mathbb{E}[(y - \overline{y} + \overline{y} - h_D(\mathbf{x}))^2]$$

$$= \underbrace{\mathbb{E}[(y - \overline{y})^2]}_{\text{noise}} + \underbrace{\mathbb{E}[(\overline{y} - h_D(\mathbf{x}))^2]}_{\text{bias+var}} + \underbrace{2\mathbb{E}[(y - \overline{y})(\overline{y} - h_D(\mathbf{x}))]}_{0} \quad (1)$$

The last term can be written as

$$2\mathbb{E}_D[\overline{y} - h_D(\mathbf{x})] \cdot \mathbb{E}_{y|\mathbf{x}}[y - \overline{y}]] = 2\mathbb{E}_D[\overline{y} - h_D(\mathbf{x})] \cdot 0 = 0.$$

Bias-Variance: Analysis (2 of 4)

- Define the true conditional mean, $\overline{y} = \mathbb{E}[y]$.

The generalization error at $\mathbf{x}$ is:

$$\begin{aligned}
&\mathbb{E}[(y - h_D(\mathbf{x}))^2] = \mathbb{E}[(y - \overline{y} + \overline{y} - h_D(\mathbf{x}))^2] \\
&= \underbrace{\mathbb{E}[(y - \overline{y})^2]}_{\text{noise}} + \underbrace{\mathbb{E}[(\overline{y} - h_D(\mathbf{x}))^2]}_{\text{bias+var}} + \underbrace{2\mathbb{E}[(y - \overline{y})(\overline{y} - h_D(\mathbf{x}))]}_{0} \qquad (1)
\end{aligned}$$

The last term can be written as

$$2\mathbb{E}_D[\overline{y} - h_D(\mathbf{x})] \cdot \mathbb{E}_{y|\mathbf{x}}[y - \overline{y}]] = 2\mathbb{E}_D[\overline{y} - h_D(\mathbf{x})] \cdot 0 = 0.$$

Bias-Variance: Analysis (2 of 4)

- Define the true conditional mean, $\overline{y} = \mathbb{E}[y]$.

The generalization error at $\mathbf{x}$ is:

$$\begin{aligned}
&\mathbb{E}[(y - h_D(\mathbf{x}))^2] = \mathbb{E}[(y - \overline{y} + \overline{y} - h_D(\mathbf{x}))^2] \\
&= \underbrace{\mathbb{E}[(y - \overline{y})^2]}_{\text{noise}} + \underbrace{\mathbb{E}[(\overline{y} - h_D(\mathbf{x}))^2]}_{\text{bias+var}} + \underbrace{2\mathbb{E}[(y - \overline{y})(\overline{y} - h_D(\mathbf{x}))]}_{0} \qquad (1)
\end{aligned}$$

The last term can be written as

$$2\mathbb{E}_D[\overline{y} - h_D(\mathbf{x})] \cdot \mathbb{E}_{y|\mathbf{x}}[y - \overline{y}]] = 2\mathbb{E}_D[\overline{y} - h_D(\mathbf{x})] \cdot 0 = 0.$$

Bias-Variance: Analysis (3 of 4)

- Define the prediction mean $\overline{h}(\mathbf{x}) = \mathbb{E}[h_D(\mathbf{x})]$.

Expanding the second term in (1), we have

$$\mathbb{E}[(\overline{y} - h_D(\mathbf{x}))^2] = \mathbb{E}[(\overline{y} - \overline{h}(\mathbf{x}) + \overline{h}(\mathbf{x}) - h_D(\mathbf{x}))^2] =$$
$$\underbrace{(\overline{y} - \overline{h}(\mathbf{x}))^2}_{\text{bias squared}} + \underbrace{\mathbb{E}_D[(\overline{h}(\mathbf{x}) - h_D(\mathbf{x}))^2]}_{\text{variance}} + \underbrace{2\mathbb{E}[(\overline{y} - \overline{h}(\mathbf{x}))(\overline{h}(\mathbf{x}) - h_D(\mathbf{x}))]}_{0} \tag{2}$$

The last term can be written as

$$2(\overline{y} - \overline{h}(\mathbf{x}))\mathbb{E}_D[\overline{h}(\mathbf{x}) - h_D(\mathbf{x})] = 2(\overline{y} - \overline{h}(\mathbf{x}))(0) = 0.$$

Bias-Variance: Analysis (3 of 4)

- Define the prediction mean $\overline{h}(\mathbf{x}) = \mathbb{E}[h_D(\mathbf{x})]$.

Expanding the second term in (1), we have

$$\mathbb{E}[(\overline{y} - h_D(\mathbf{x}))^2] = \mathbb{E}[(\overline{y} - \overline{h}(\mathbf{x}) + \overline{h}(\mathbf{x}) - h_D(\mathbf{x}))^2] =$$

$$\underbrace{(\overline{y} - \overline{h}(\mathbf{x}))^2}_{\text{bias squared}} + \underbrace{\mathbb{E}_D[(\overline{h}(\mathbf{x}) - h_D(\mathbf{x}))^2]}_{\text{variance}} + \underbrace{2\mathbb{E}[(\overline{y} - \overline{h}(\mathbf{x}))(\overline{h}(\mathbf{x}) - h_D(\mathbf{x}))]}_{0} \tag{2}$$

The last term can be written as

$$2(\overline{y} - \overline{h}(\mathbf{x}))\mathbb{E}_D[\overline{h}(\mathbf{x}) - h_D(\mathbf{x})] = 2(\overline{y} - \overline{h}(\mathbf{x}))(0) = 0.$$

Bias-Variance: Analysis (3 of 4)

- Define the prediction mean $\overline{h}(\mathbf{x}) = \mathbb{E}[h_D(\mathbf{x})]$.

Expanding the second term in (1), we have

$$\mathbb{E}[(\overline{y} - h_D(\mathbf{x}))^2] = \mathbb{E}[(\overline{y} - \overline{h}(\mathbf{x}) + \overline{h}(\mathbf{x}) - h_D(\mathbf{x}))^2] =$$

$$\underbrace{(\overline{y} - \overline{h}(\mathbf{x}))^2}_{\text{bias squared}} + \underbrace{\mathbb{E}_D[(\overline{h}(\mathbf{x}) - h_D(\mathbf{x}))^2]}_{\text{variance}} + \underbrace{2\mathbb{E}[(\overline{y} - \overline{h}(\mathbf{x}))(\overline{h}(\mathbf{x}) - h_D(\mathbf{x}))]}_{0} \tag{2}$$

The last term can be written as

$$2(\overline{y} - \overline{h}(\mathbf{x}))\mathbb{E}_D[\overline{h}(\mathbf{x}) - h_D(\mathbf{x})] = 2(\overline{y} - \overline{h}(\mathbf{x}))(0) = 0.$$

Bias-Variance: Analysis (3 of 4)

- Define the prediction mean $\overline{h}(\mathbf{x}) = \mathbb{E}[h_D(\mathbf{x})]$.

Expanding the second term in (1), we have

$$\mathbb{E}[(\overline{y} - h_D(\mathbf{x}))^2] = \mathbb{E}[(\overline{y} - \overline{h}(\mathbf{x}) + \overline{h}(\mathbf{x}) - h_D(\mathbf{x}))^2] =$$
$$\underbrace{(\overline{y} - \overline{h}(\mathbf{x}))^2}_{\text{bias squared}} + \underbrace{\mathbb{E}_D[(\overline{h}(\mathbf{x}) - h_D(\mathbf{x}))^2]}_{\text{variance}} + \underbrace{2\mathbb{E}[(\overline{y} - \overline{h}(\mathbf{x}))(\overline{h}(\mathbf{x}) - h_D(\mathbf{x}))]}_{0} \tag{2}$$

The last term can be written as

$$2(\overline{y} - \overline{h}(\mathbf{x}))\mathbb{E}_D[\overline{h}(\mathbf{x}) - h_D(\mathbf{x})] = 2(\overline{y} - \overline{h}(\mathbf{x}))(0) = 0.$$

Bias-Variance: Analysis (4 of 4)

Substituting (2) back into (1), we have:

$$\mathbb{E}[(y - h_D(\mathbf{x}))^2] =$$
$$\mathbb{E}_{y|\mathbf{x}}[(y - \overline{y})^2] + (\overline{y} - \overline{h}(\mathbf{x}))^2 + \mathbb{E}_D[(\overline{h}(\mathbf{x}) - h_D(\mathbf{x}))^2]$$
$$= \text{noise}(\mathbf{x}) + (\text{bias}(h(\mathbf{x})))^2 + \text{var}_D(h_D(\mathbf{x})).$$

Depends on noise, and (i) systematic error (or bias), and (ii) sensitivity of the predictor to data (or variance.)

Considering the expectation over $\mathbf{x}$, the generalization error is:

$$\mathbb{E}_{\mathbf{x}}\left[\text{noise}(\mathbf{x}) + (\text{bias}(h(\mathbf{x})))^2 + \text{var}_D(h_D(\mathbf{x}))\right]$$

Bias-Variance: Analysis (4 of 4)

Substituting (2) back into (1), we have:

$$\begin{aligned}
&\mathbb{E}[(y - h_D(\mathbf{x}))^2] = \\
&\mathbb{E}_{y|\mathbf{x}}[(y - \overline{y})^2] + (\overline{y} - \overline{h}(\mathbf{x}))^2 + \mathbb{E}_D[(\overline{h}(\mathbf{x}) - h_D(\mathbf{x}))^2] \\
&= \text{noise}(\mathbf{x}) + (\text{bias}(h(\mathbf{x})))^2 + \text{var}_D(h_D(\mathbf{x})).
\end{aligned}$$

Depends on noise, and (i) systematic error (or bias), and (ii) sensitivity of the predictor to data (or variance.)

Considering the expectation over $\mathbf{x}$, the generalization error is:

$$\mathbb{E}_{\mathbf{x}}\left[\text{noise}(\mathbf{x}) + (\text{bias}(h(\mathbf{x})))^2 + \text{var}_D(h_D(\mathbf{x}))\right]$$

Bias-Variance: Analysis (4 of 4)

Substituting (2) back into (1), we have:

$$\begin{aligned}
&\mathbb{E}[(y - h_D(\mathbf{x}))^2] = \\
&\mathbb{E}_{y|\mathbf{x}}[(y - \overline{y})^2] + (\overline{y} - \overline{h}(\mathbf{x}))^2 + \mathbb{E}_D[(\overline{h}(\mathbf{x}) - h_D(\mathbf{x}))^2] \\
&= \text{noise}(\mathbf{x}) + (\text{bias}(h(\mathbf{x})))^2 + \text{var}_D(h_D(\mathbf{x})).
\end{aligned}$$

Depends on noise, and (i) systematic error (or bias), and (ii) sensitivity of the predictor to data (or variance.)

Considering the expectation over $\mathbf{x}$, the generalization error is:

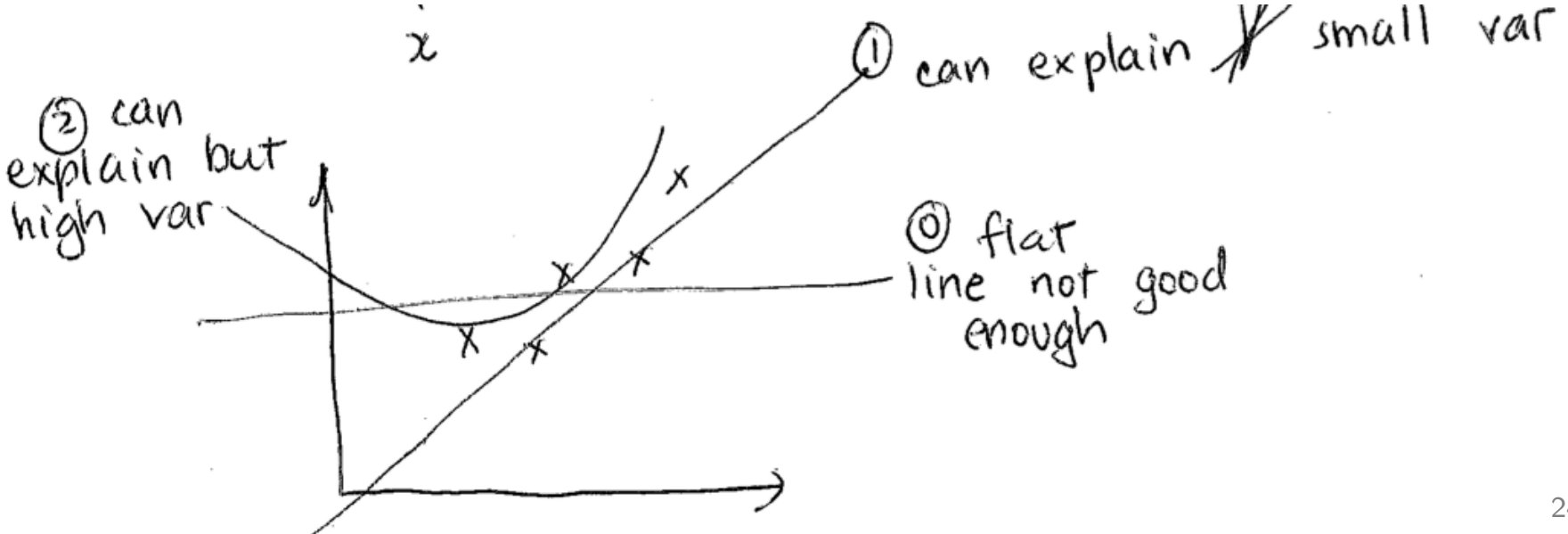
$$\mathbb{E}_{\mathbf{x}}\left[\text{noise}(\mathbf{x}) + (\text{bias}(h(\mathbf{x})))^2 + \text{var}_D(h_D(\mathbf{x}))\right]$$

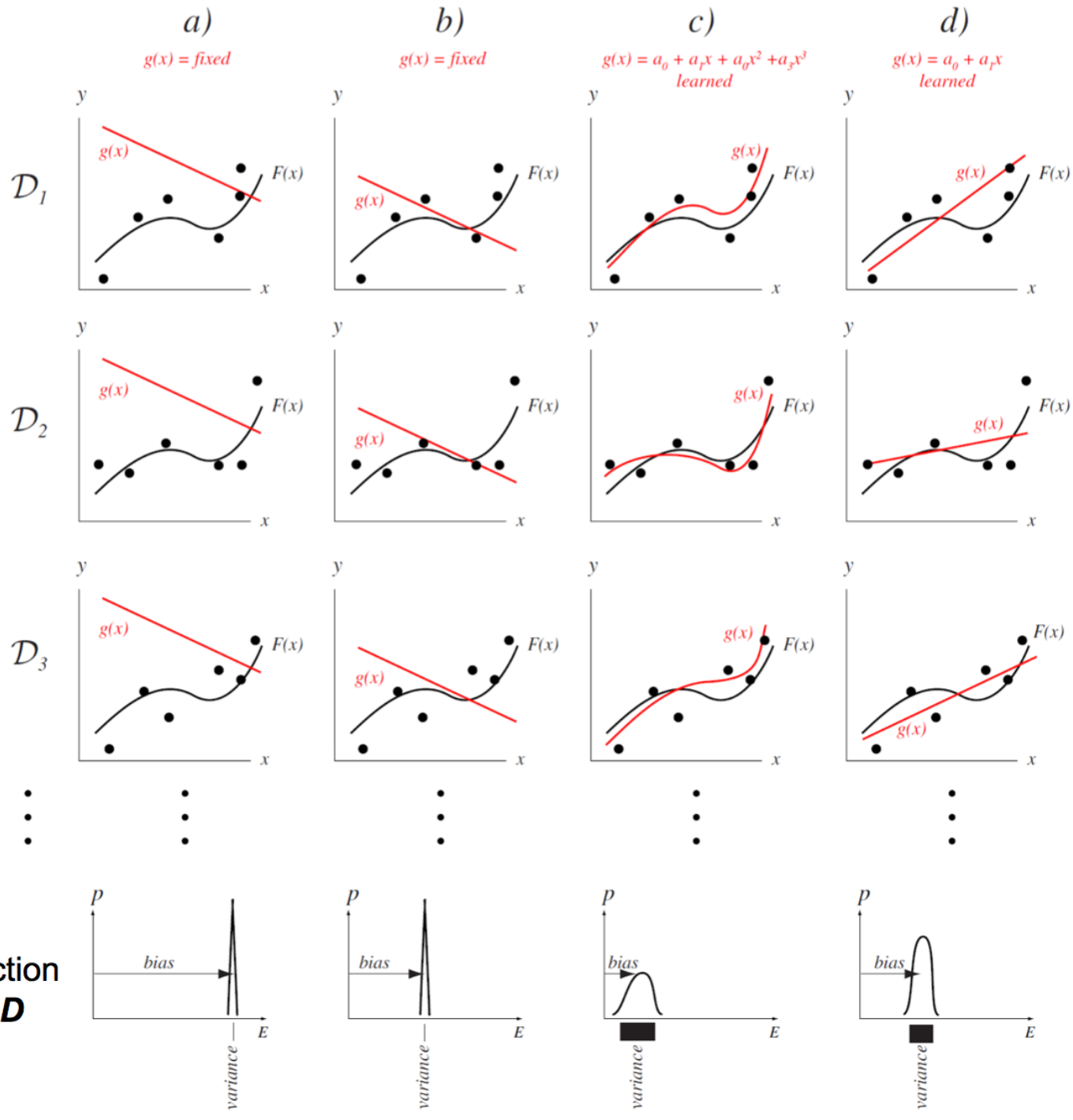
The Bias-Variance Tradeoff

- If model fits the training data perfectly and there is a small amount of data then the variance will be high (overfits!)
- If model is very simple, then the variance will be low but the bias high (underfits!)
- As $n \to \infty$ the variance $\mathbb{E}_D[(\overline{h}(\mathbf{x}) - h_D(\mathbf{x}))^2]$ falls, can use a more complex model.

The Bias-Variance Tradeoff

- If model fits the training data perfectly and there is a small amount of data then the variance will be high (overfits!)
- If model is very simple, then the variance will be low but the bias high (underfits!)
- As $n \to \infty$ the variance $\mathbb{E}_D[(\overline{h}(\mathbf{x}) - h_D(\mathbf{x}))^2]$ falls, can use a more complex model.





probability density function of error wrt ***D***

Contents

Regularization (1 of 4)

How can we control this tradeoff? Recall the training loss for a parametric model:

$$\mathcal{L}(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n}(y_i - h(\mathbf{x}_i; \mathbf{w}))^2.$$

We can give preference to simple models by penalizing complexity. This is regularization. "Simple" is interpreted as not too many parameters that are non-zero, or not too many large parameters.

Regularization captures a preference for simple hypotheses over complex ones, even if they predict the training data less well.

Regularization (1 of 4)

How can we control this tradeoff? Recall the training loss for a parametric model:

$$\mathcal{L}(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n}(y_i - h(\mathbf{x}_i; \mathbf{w}))^2.$$

We can give preference to simple models by penalizing complexity. This is regularization. "Simple" is interpreted as not too many parameters that are non-zero, or not too many large parameters.

Regularization captures a preference for simple hypotheses over complex ones, even if they predict the training data less well.

Regularization (1 of 4)

How can we control this tradeoff? Recall the training loss for a parametric model:

$$\mathcal{L}(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n}(y_i - h(\mathbf{x}_i; \mathbf{w}))^2.$$

We can give preference to simple models by penalizing complexity. This is regularization. "Simple" is interpreted as not too many parameters that are non-zero, or not too many large parameters.

Regularization captures a preference for simple hypotheses over complex ones, even if they predict the training data less well.

Regularization (2 of 4)

Tuning parameter $\lambda \geq 0$.

- Ridge regression (squared ℓ_2 norm)

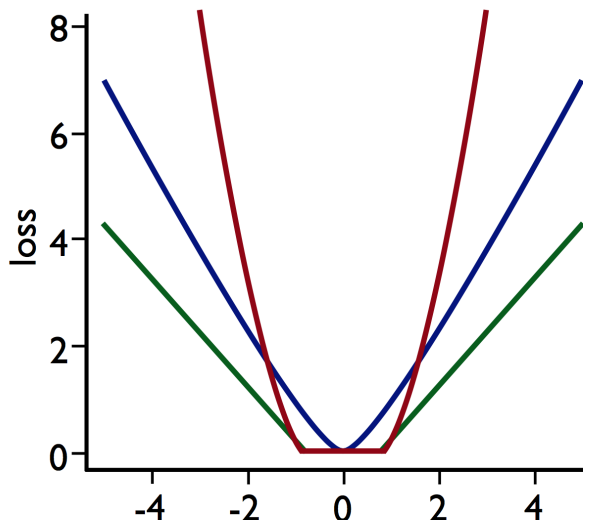
$$\min_{\mathbf{w}} \mathcal{L}(\mathbf{w}) + \lambda ||\mathbf{w}||_2^2$$

- LASSO (ℓ_1 norm)

$$\min_{\mathbf{w}} \mathcal{L}(\mathbf{w}) + \lambda ||\mathbf{w}||_1$$

- Elastic net applies penalty $\lambda_1 ||\mathbf{w}||_1 + \lambda_2 ||\mathbf{w}||_2^2$

Many others! e.g., insensitive close to zero, mix of linear and quad.

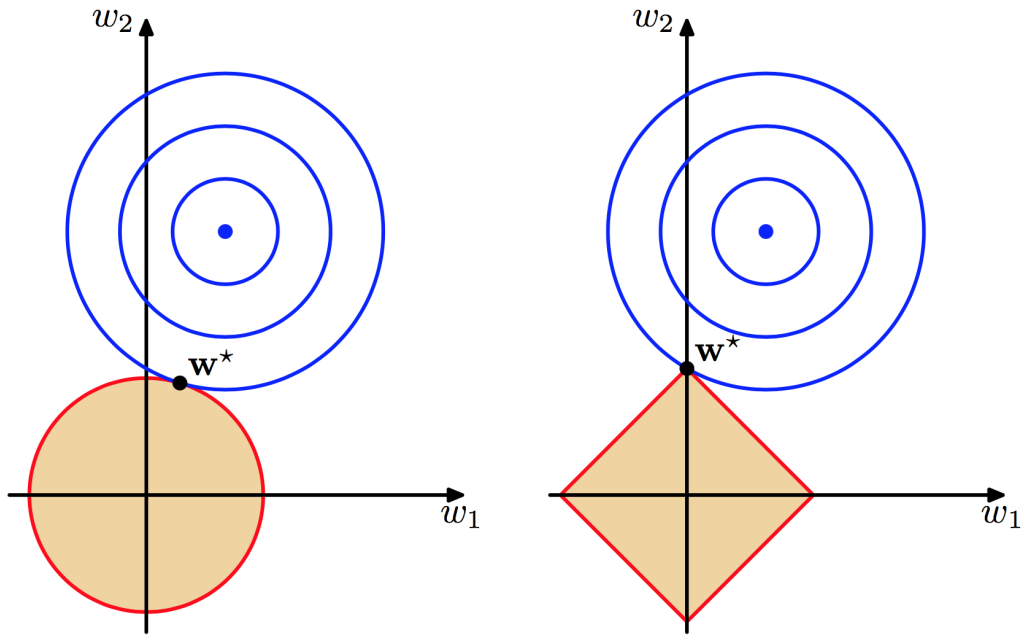


Regularization (3 of 4)

$$\min_{\mathbf{w}} \mathcal{L}(\mathbf{w}) + \lambda ||\mathbf{w}||_1$$

equivalent, via Lagrangian optimization, to

$$\min_{\mathbf{w}} \mathcal{L}(\mathbf{w}) \quad \text{s.t.} \quad ||\mathbf{w}||_1 \leq b, \text{ suitable } b$$



LASSO finds sparse, interpretable solutions (but may underfit).

Regularization (4 of 4)

Computational aspects:

- Ridge regression is convex and differentiable, can solve via matrix inversion Matrix $(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})$ can be expensive to invert if m large. Alternative: gradient descent. We'll discuss gradient descent and variants later in course.
- LASSO regression is not differentiable, and has no closed form. But can solve via 'quadratic programming' techniques from convex optimization. Specialized, iterative algorithms also exist (Efron et al. 2004, Hastie 2005).

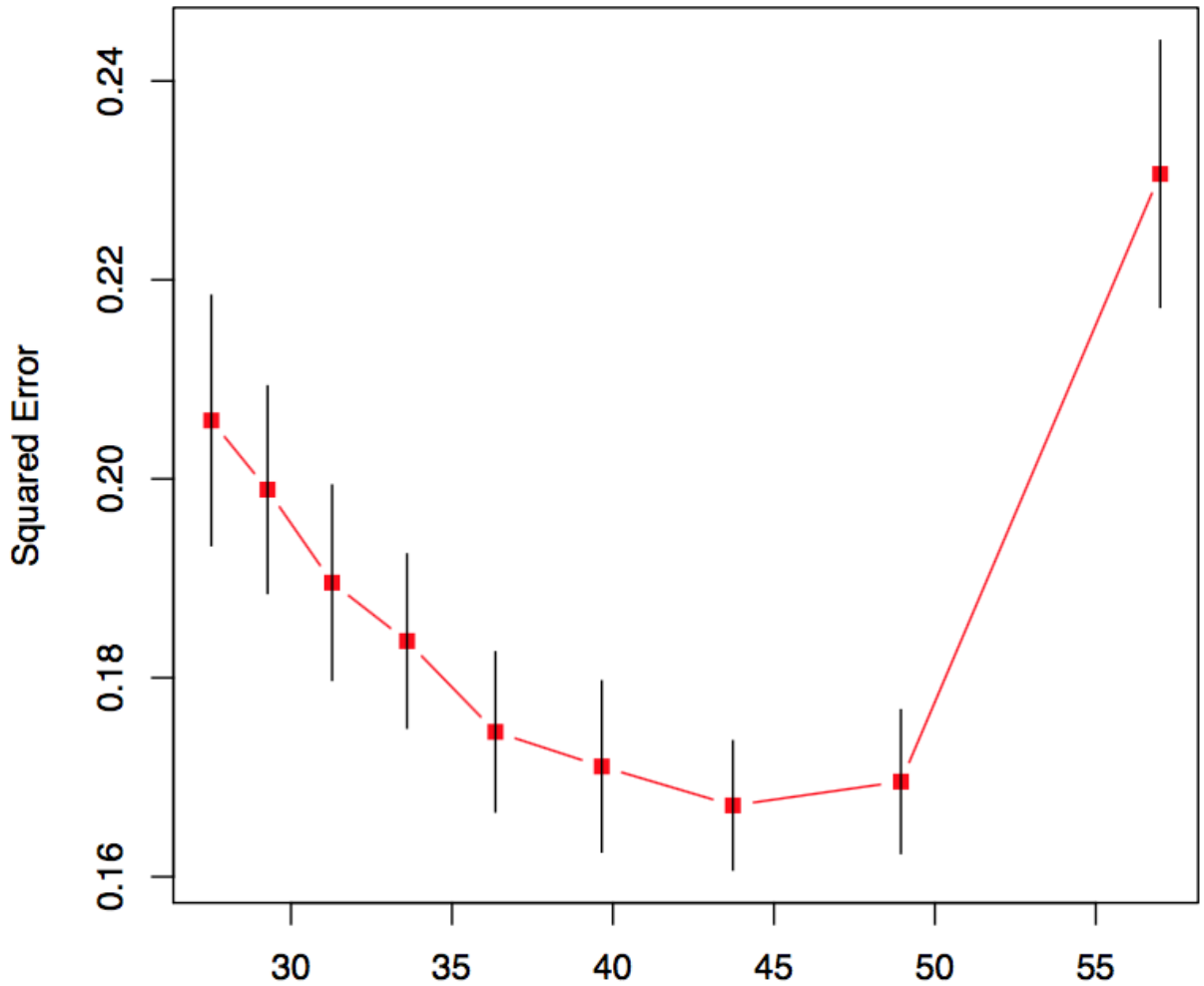
Regularization (4 of 4)

Computational aspects:

- Ridge regression is convex and differentiable, can solve via matrix inversion Matrix $(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})$ can be expensive to invert if m large. Alternative: gradient descent. We'll discuss gradient descent and variants later in course.
- LASSO regression is not differentiable, and has no closed form. But can solve via 'quadratic programming' techniques from convex optimization. Specialized, iterative algorithms also exist (Efron et al. 2004, Hastie 2005).

Using CV to select the Regularization Penalty

Example on spam email classification (x-axis is varying λ, y-axis is validation error):



Contents

Outwitting the Tradeoff: Ensemble methods

- Can we reduce the variance of a learning algorithm without increasing its bias?

Yes! Use a committee: train $r > 1$ hypotheses, and take a majority vote (classification) or average (regression).

The idea:

- Committees don't increase bias because the average performance is equal to the average performance of its members.
- Committees decrease variance because a spurious pattern picked up by one learner may not be found by others, and will be out voted.

Outwitting the Tradeoff: Ensemble methods

- Can we reduce the variance of a learning algorithm without increasing its bias?

Yes! Use a committee: train $r > 1$ hypotheses, and take a majority vote (classification) or average (regression).

The idea:

- Committees don't increase bias because the average performance is equal to the average performance of its members.
- Committees decrease variance because a spurious pattern picked up by one learner may not be found by others, and will be out voted.

Outwitting the Tradeoff: Ensemble methods

- Can we reduce the variance of a learning algorithm without increasing its bias?

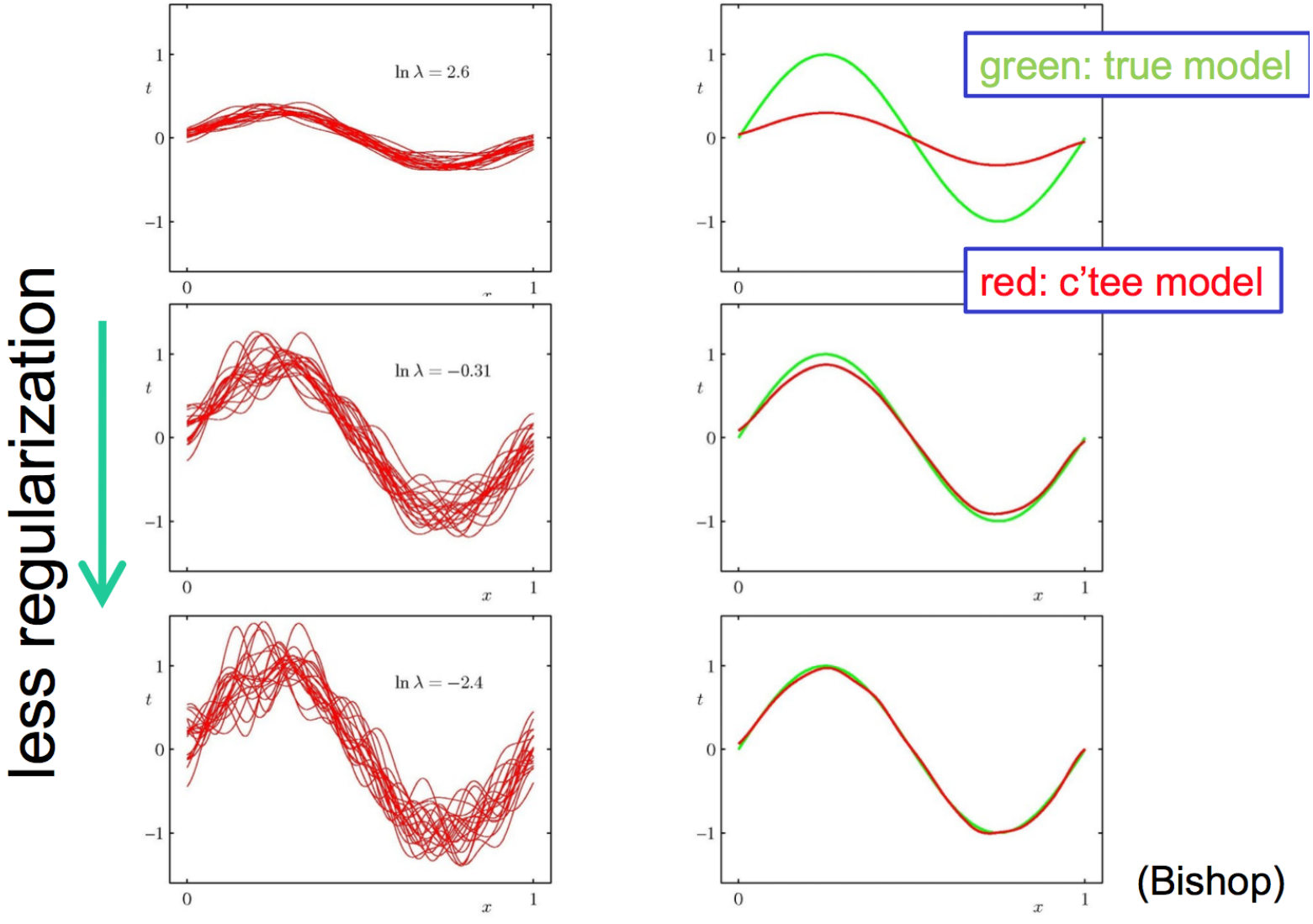
Yes! Use a committee: train $r > 1$ hypotheses, and take a majority vote (classification) or average (regression).

The idea:

- Committees don't increase bias because the average performance is equal to the average performance of its members.
- Committees decrease variance because a spurious pattern picked up by one learner may not be found by others, and will be out voted.

Example: Regression using a Committee

This can work well if the variance of the 'base' learner is high.



Two Ensemble Methods: Bagging and Boosting

Boostrap Aggregation (Bagging):

- Learn $h_1, \ldots, h_r$ hypotheses on r derived training sets (each sampled with replacement from D)
- Classify an example with the majority vote

Boosting:

- Train in sequence, and when train next model give more weight to examples that are not correctly classified so far
- Also weight the votes according to accuracy on the reweighted training set

Relative to simple bagging, boosting makes individual learners more independent, reducing overall variance!

Two Ensemble Methods: Bagging and Boosting

Boostrap Aggregation (Bagging):

- Learn $h_1, \ldots, h_r$ hypotheses on r derived training sets (each sampled with replacement from D)
- Classify an example with the majority vote

Boosting:

- Train in sequence, and when train next model give more weight to examples that are not correctly classified so far
- Also weight the votes according to accuracy on the reweighted training set

Relative to simple bagging, boosting makes individual learners more independent, reducing overall variance!

Two Ensemble Methods: Bagging and Boosting

Boostrap Aggregation (Bagging):

- Learn $h_1, \ldots, h_r$ hypotheses on r derived training sets (each sampled with replacement from D)
- Classify an example with the majority vote

Boosting:

- Train in sequence, and when train next model give more weight to examples that are not correctly classified so far
- Also weight the votes according to accuracy on the reweighted training set

Relative to simple bagging, boosting makes individual learners more independent, reducing overall variance!

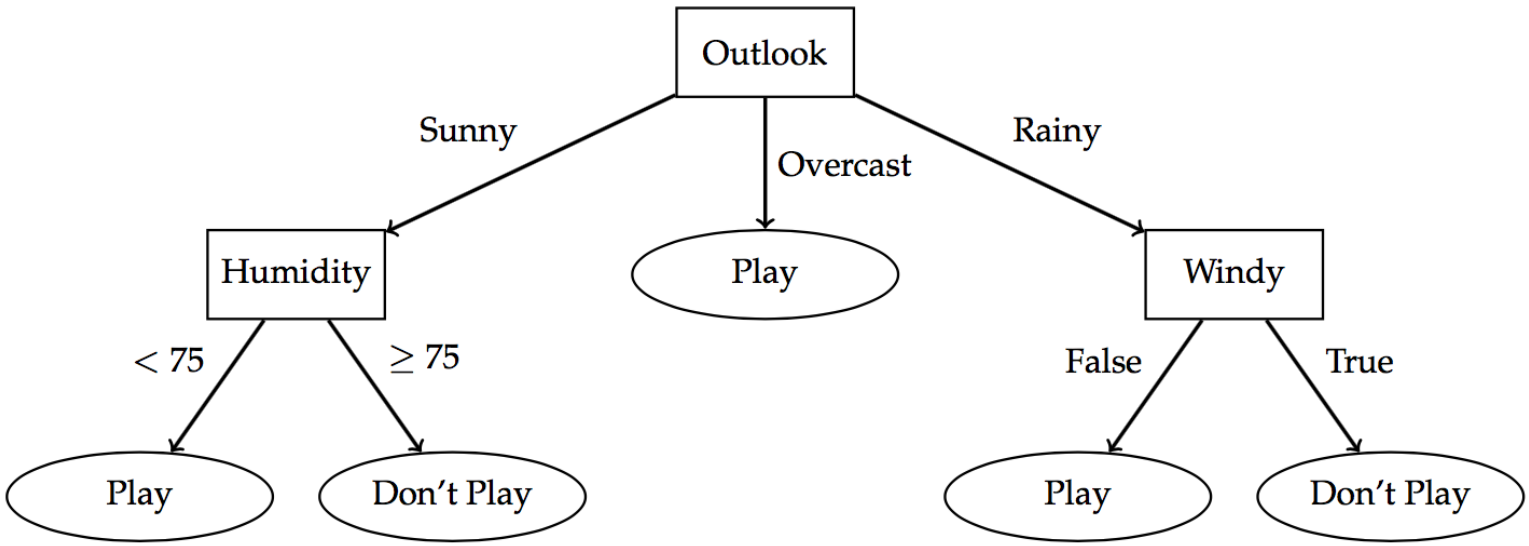
Example: Boosting Decision Trees (1 of 2)

Consider an example of training a decision tree to predict whether or not a group of friends will play golf.

Outlook	Temperature	Humidity	Windy	Play?
sunny	85	85	false	no
sunny	80	90	true	no
overcast	83	78	false	yes
rain	70	96	false	yes
rain	68	80	false	yes
rain	65	70	true	no
overcast	64	65	true	yes
sunny	72	95	false	no
sunny	69	70	false	yes
rain	75	80	false	yes
sunny	75	70	true	yes
overcast	72	90	true	yes
overcast	81	75	false	yes
rain	71	80	true	no

Example: Boosting Decision Trees (1 of 2)

Example of a trained tree (R. Adams):



Decision trees are simple, fast, highly interpretable, and can capture non-linear effects.

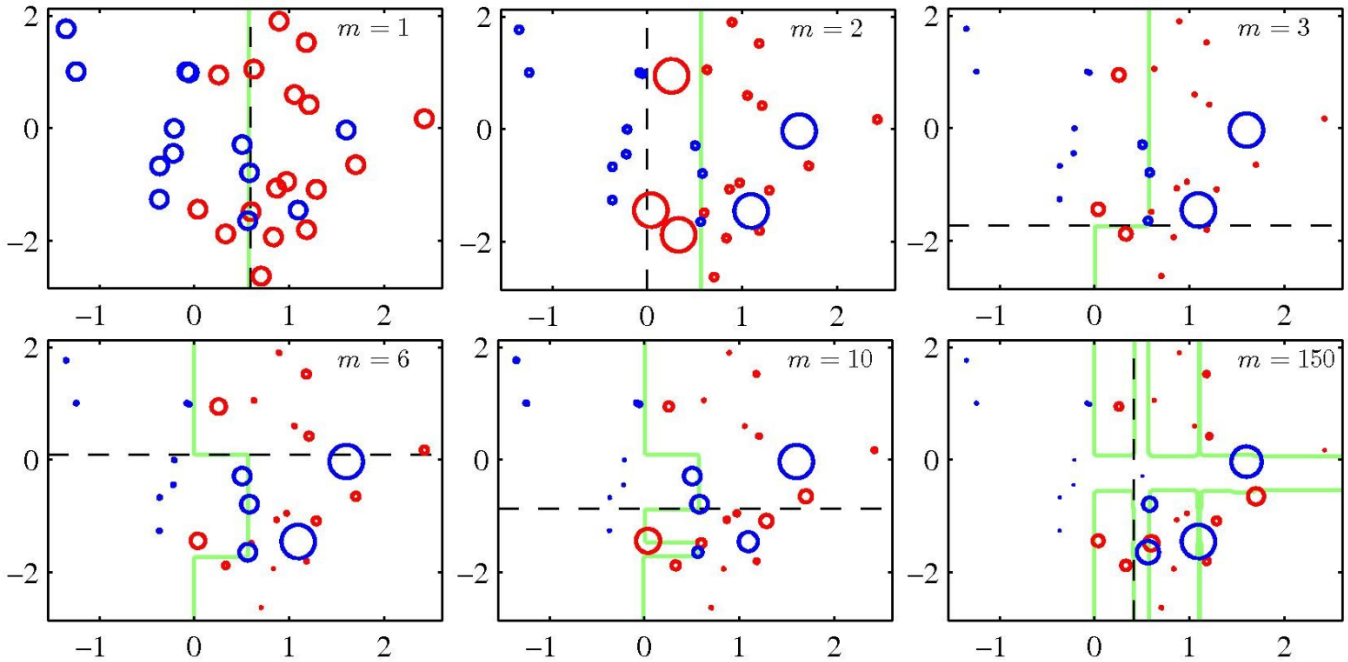
But they tend to have high variance.

Example: Boosting Decision Trees (2 of 2)

A stump is a decision tree with one node. Low variance but high bias.

We can apply boosting to stumps (the "Adaboost" algorithm):

(Bishop)



Base learners are simple linear thresholds on one of the axes (decision stump on continuous attribute). m is # base learners trained so far.

dashed black line- current hypothesis
solid green line- current ensemble
size of circle- current data weight

84

Example: Character recognition

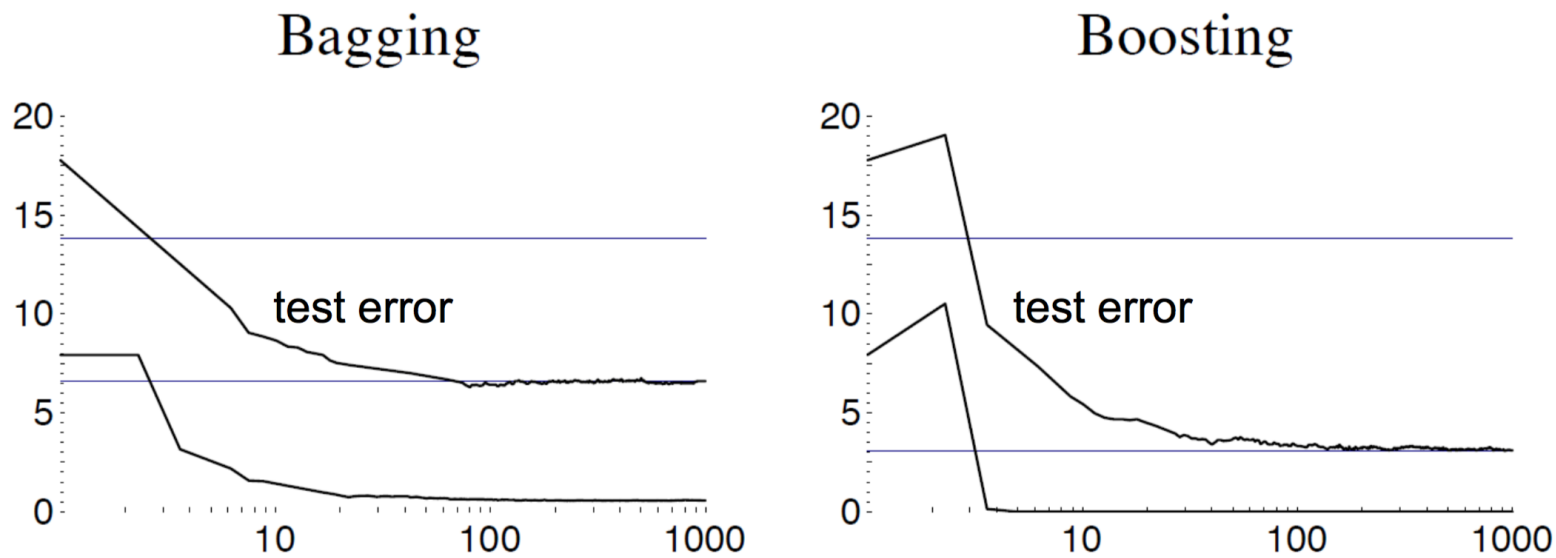
- 20,000 examples
- 16 primitive numerical features (edge counts, mean ‘x-on’, correlation ’x-on, y-on’, etc.)

http://archive.ics.uci.edu/ml/datasets/Letter+Recognition

Bagging vs Boosting on Character Recognition

Number of hypotheses on the x-axis. Boosting reduces test error from 13.8% to 3.1%:

(Shapire et al.)



Improves the performance of many learning algorithms! Support from empirical studies, also theory (improves the margin on training data.)

In Practice: Random Forests

- Bagging applied to randomized decision trees (Ho'95, Breiman'01):
- Repeat:
  - Sample a data set of size n with replacement
  - Train a decision tree (or regression tree), randomly selecting at each split a candidate set of features to split on (e.g. $\sqrt{m}$). This provides higher variance
- The ensemble model uses either the mode prediction (classification) or mean prediction (regression)

RFs are used in many industrial applications.

Summary

- Important for Low Generalization Error
- Cross-validation for model selection
- Bias-Variance tradeoff
- Regularization
- Bagging and boosting