

Warm up

Homework due tonight! 11:59 PM

Let $X \sim \mathcal{N}(\mu, \Sigma)$ where $X \in \mathbb{R}^d$

1. Let $Y = AX + b$. For what $\tilde{\mu}, \tilde{\Sigma}$ is $Y \sim \mathcal{N}(\tilde{\mu}, \tilde{\Sigma})$
2. Suppose I can generate independent Gaussians $Z \sim \mathcal{N}(0, 1)$ (e.g., `numpy.random.randn`). How can I use this to generate X ?
3. What is $\mathbb{E}[X^T \Sigma^{-1} X]$?



Bias-Variance Tradeoff

Machine Learning – CSE546

Kevin Jamieson

University of Washington

Oct 4, 2018

Statistical Learning

$$P_{XY}(X = x, Y = y)$$

Goal: Predict Y given X

Find function η that minimizes

$$\mathbb{E}_{XY}[(Y - \eta(X))^2]$$

Statistical Learning

$$P_{XY}(X = x, Y = y)$$

Goal: Predict Y given X

Find function η that minimizes

$$\mathbb{E}_{XY}[(Y - \eta(X))^2] = \mathbb{E}_X \left[\mathbb{E}_{Y|X}[(Y - \eta(x))^2 | X = x] \right]$$

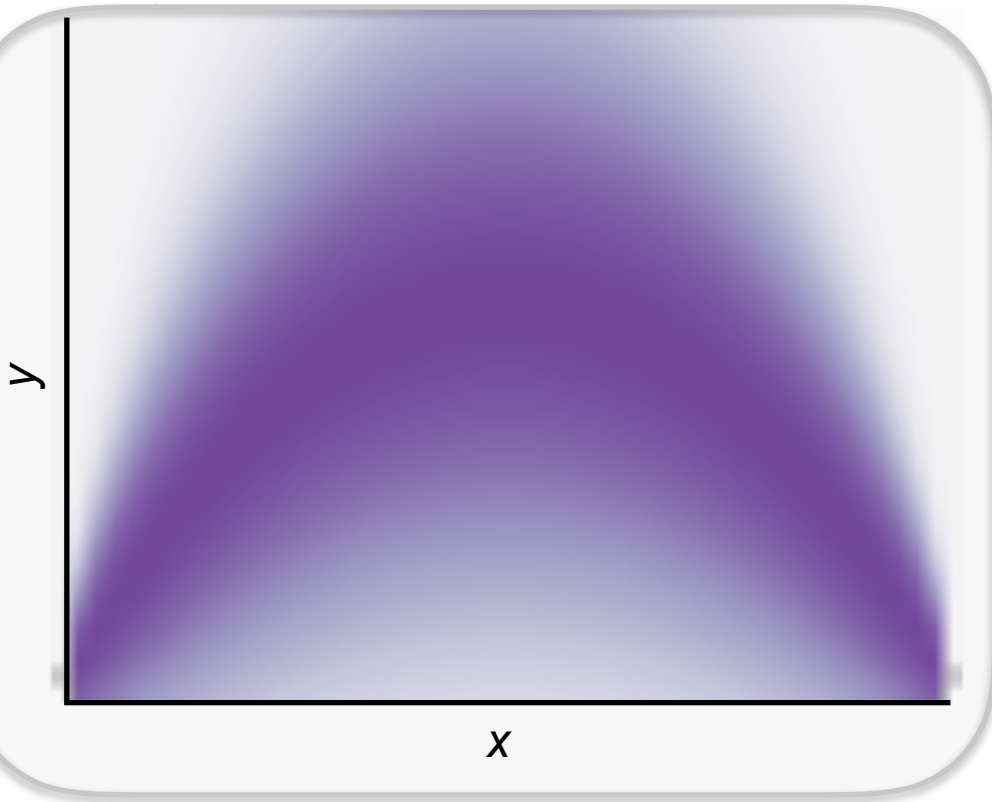
$$\eta(x) = \arg \min_c \mathbb{E}_{Y|X}[(Y - c)^2 | X = x] = \mathbb{E}_{Y|X}[Y | X = x]$$

Under LS loss, optimal predictor: $\eta(x) = \mathbb{E}_{Y|X}[Y | X = x]$

Statistical Learning

$$\mathbb{E}_{XY}[(Y - \eta(X))^2]$$

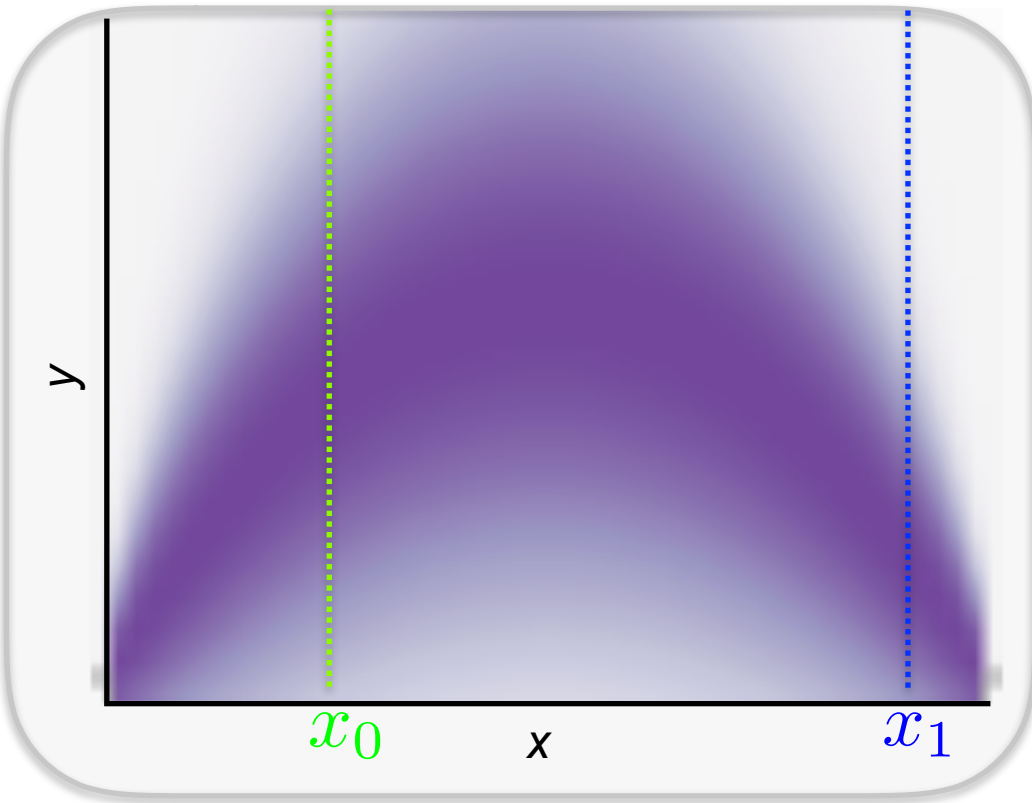
$$P_{XY}(X = x, Y = y)$$



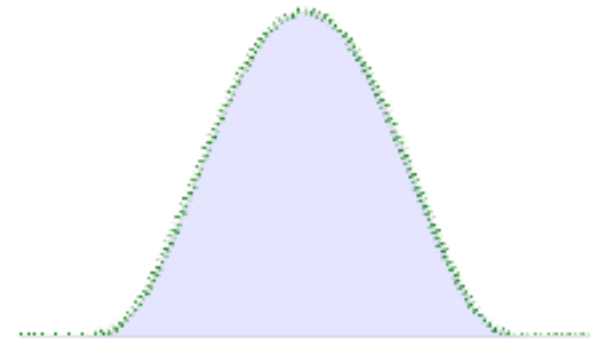
Statistical Learning

$$\mathbb{E}_{XY}[(Y - \eta(X))^2]$$

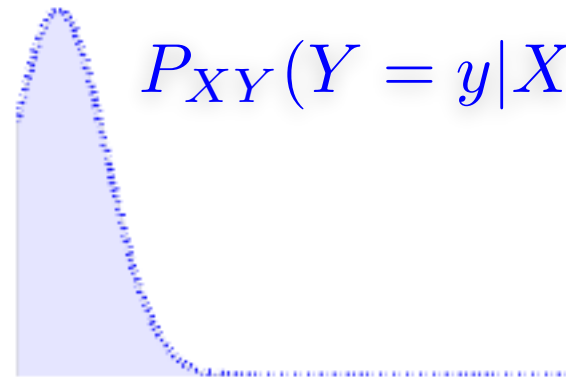
$$P_{XY}(X = x, Y = y)$$



$$P_{XY}(Y = y|X = x_0)$$



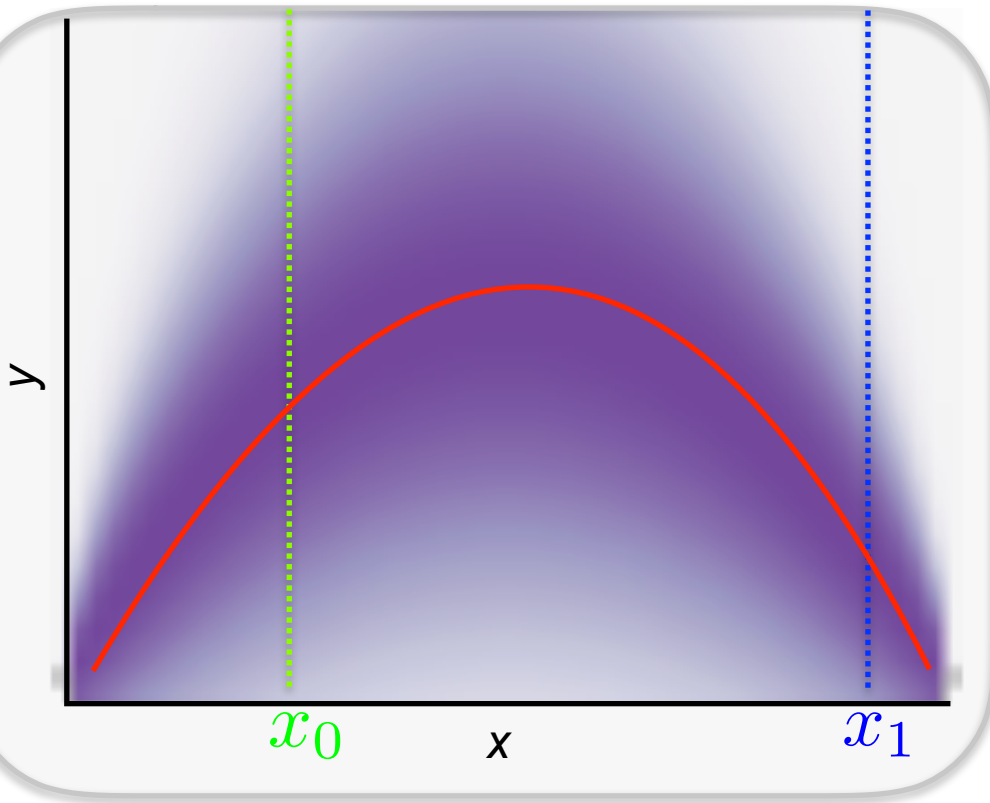
$$P_{XY}(Y = y|X = x_1)$$



Statistical Learning

$$\mathbb{E}_{XY}[(Y - \eta(X))^2]$$

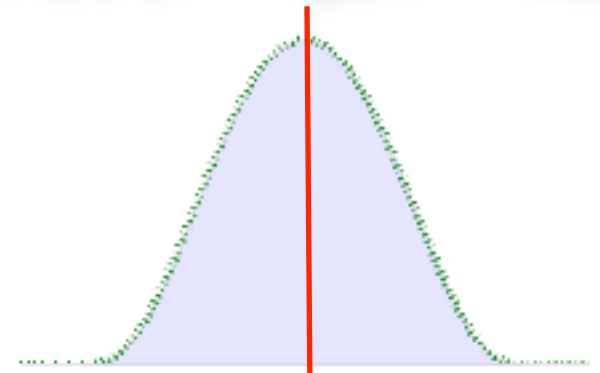
$$P_{XY}(X = x, Y = y)$$



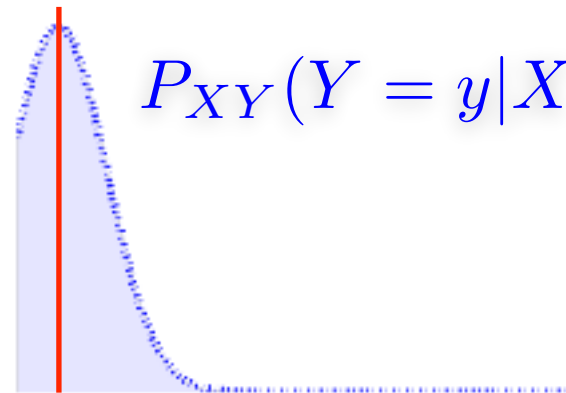
Ideally, we want to find:

$$\eta(x) = \mathbb{E}_{Y|X}[Y|X = x]$$

$$P_{XY}(Y = y|X = x_0)$$



$$P_{XY}(Y = y|X = x_1)$$

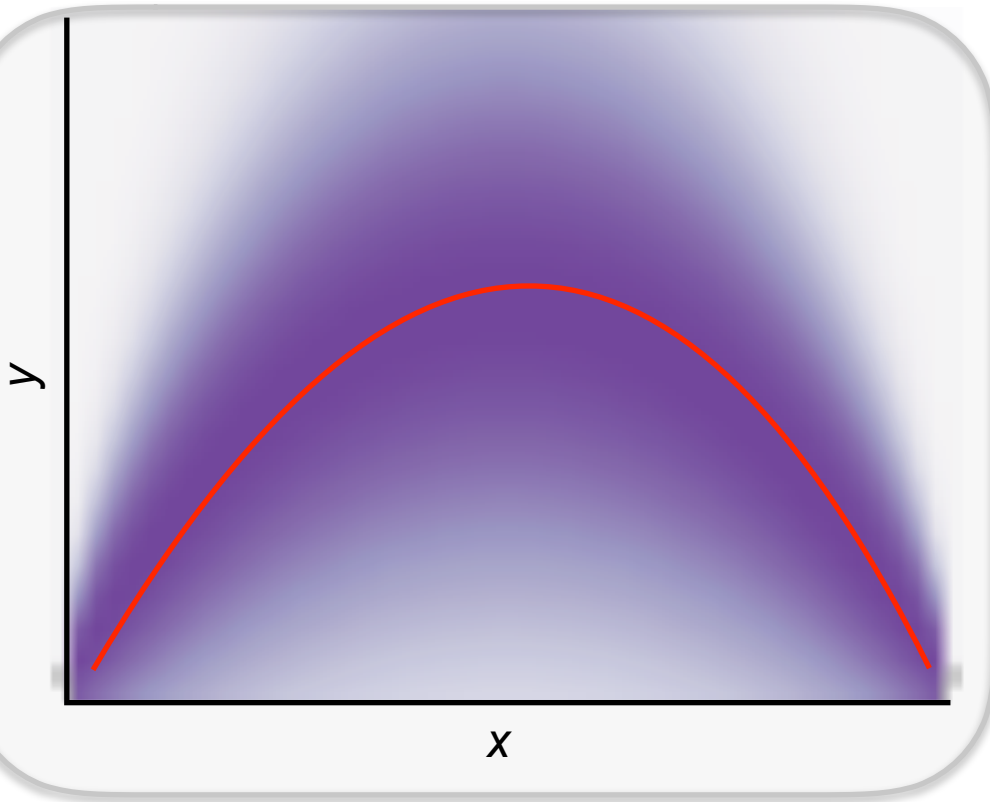


Statistical Learning

$$P_{XY}(X = x, Y = y)$$

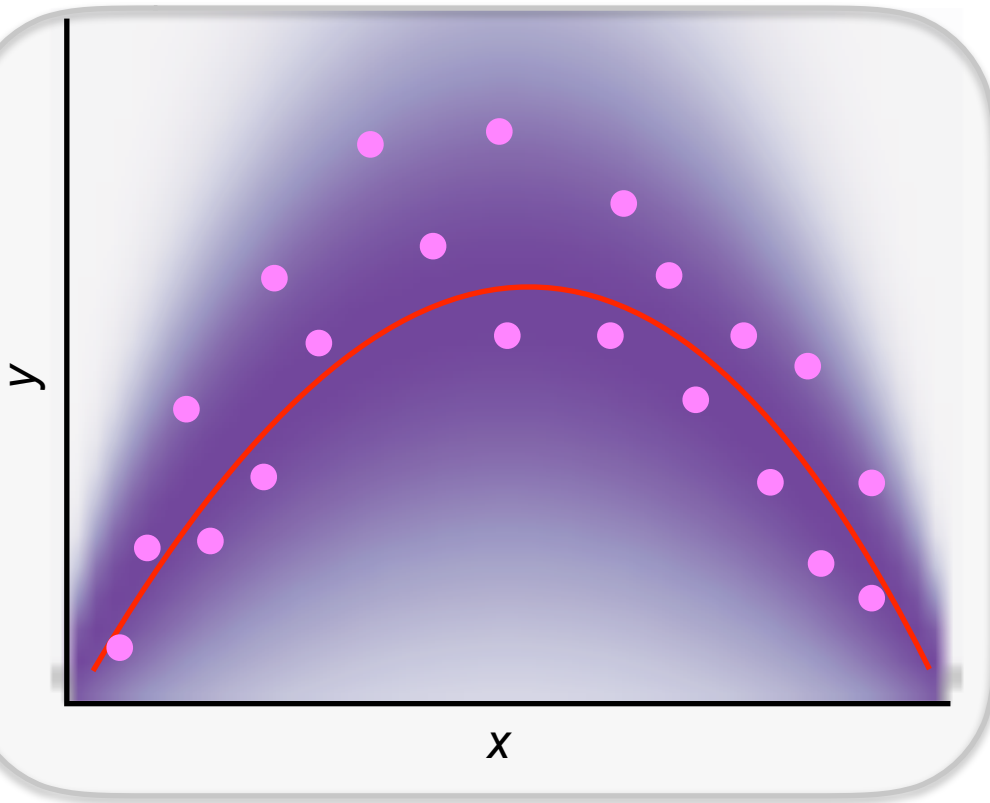
Ideally, we want to find:

$$\eta(x) = \mathbb{E}_{Y|X}[Y|X = x]$$



Statistical Learning

$$P_{XY}(X = x, Y = y)$$



Ideally, we want to find:

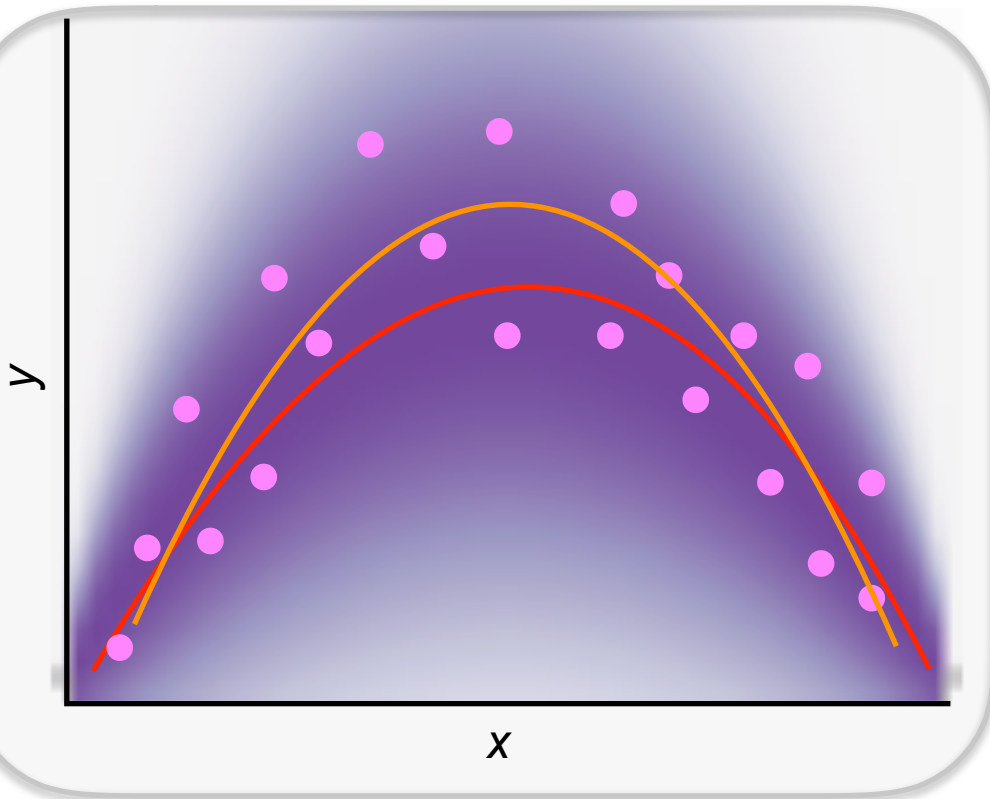
$$\eta(x) = \mathbb{E}_{Y|X}[Y|X = x]$$

But we only have samples:

$$(x_i, y_i) \stackrel{i.i.d.}{\sim} P_{XY} \quad \text{for } i = 1, \dots, n$$

Statistical Learning

$$P_{XY}(X = x, Y = y)$$



Ideally, we want to find:

$$\eta(x) = \mathbb{E}_{Y|X}[Y|X = x]$$

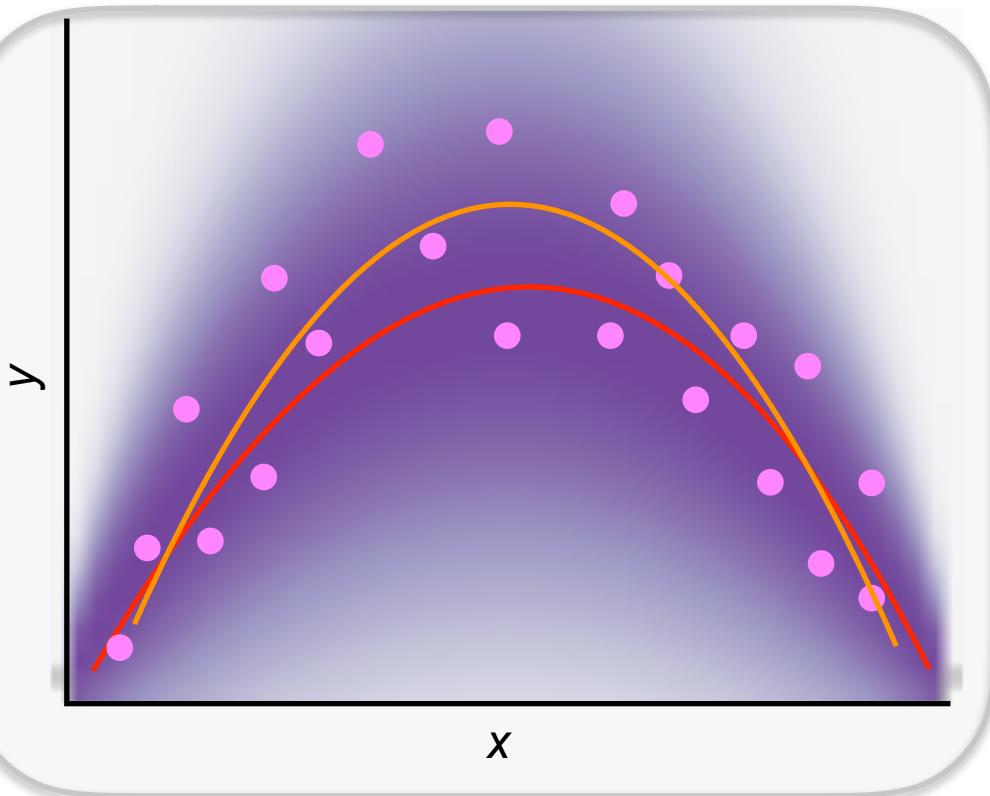
But we only have samples:
 $(x_i, y_i) \stackrel{i.i.d.}{\sim} P_{XY}$ for $i = 1, \dots, n$

and are restricted to a
function class (e.g., linear)
so we compute:

$$\hat{f} = \arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2$$

Statistical Learning

$$P_{XY}(X = x, Y = y)$$



Ideally, we want to find:

$$\eta(x) = \mathbb{E}_{Y|X}[Y|X = x]$$

But we only have samples:
 $(x_i, y_i) \stackrel{i.i.d.}{\sim} P_{XY}$ for $i = 1, \dots, n$

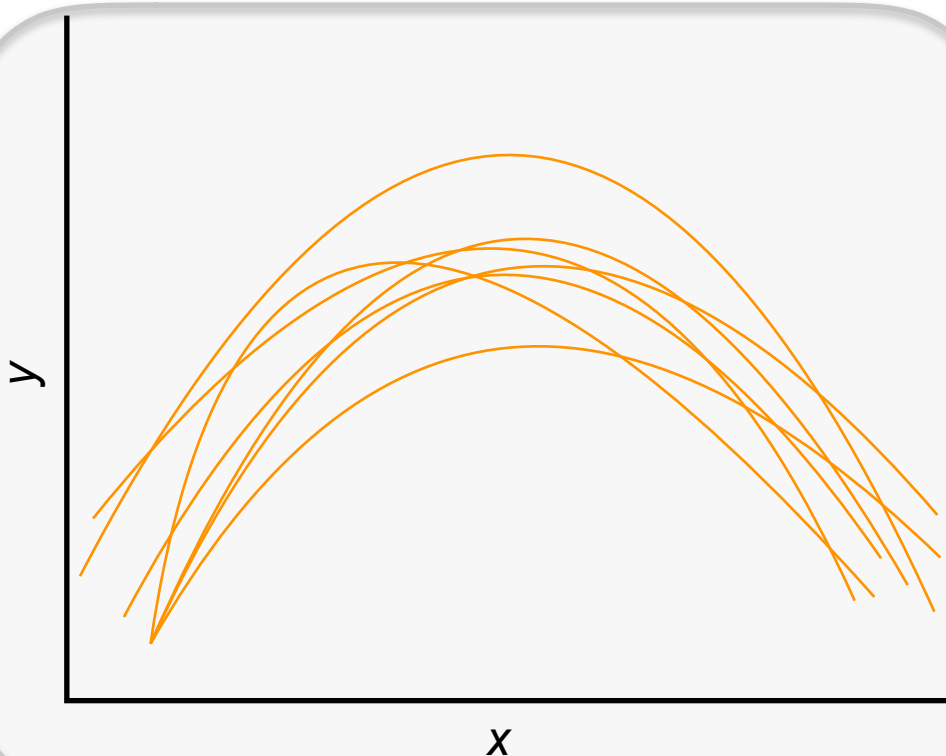
and are restricted to a
function class (e.g., linear)
so we compute:

$$\hat{f} = \arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2$$

We care about future predictions: $\mathbb{E}_{XY}[(Y - \hat{f}(X))^2]$

Statistical Learning

$$P_{XY}(X = x, Y = y)$$



Ideally, we want to find:

$$\eta(x) = \mathbb{E}_{Y|X}[Y|X = x]$$

But we only have samples:
 $(x_i, y_i) \stackrel{i.i.d.}{\sim} P_{XY}$ for $i = 1, \dots, n$

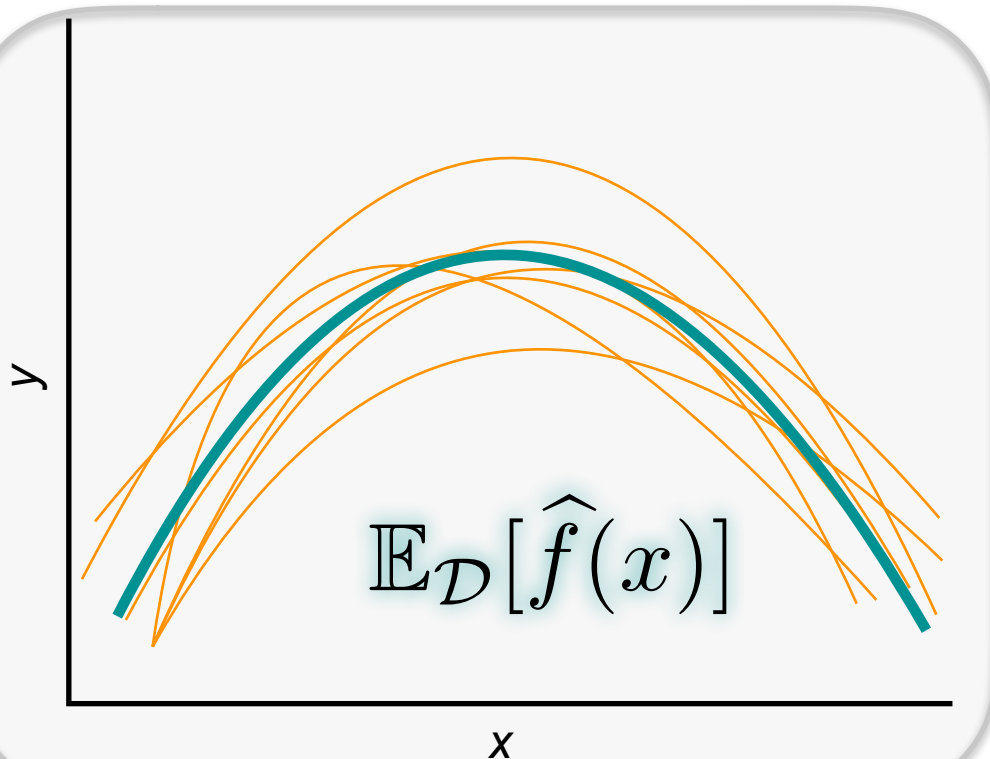
and are restricted to a function class (e.g., linear) so we compute:

$$\hat{f} = \arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2$$

Each draw $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ results in different \hat{f}

Statistical Learning

$$P_{XY}(X = x, Y = y)$$



Ideally, we want to find:

$$\eta(x) = \mathbb{E}_{Y|X}[Y|X = x]$$

But we only have samples:
 $(x_i, y_i) \stackrel{i.i.d.}{\sim} P_{XY}$ for $i = 1, \dots, n$

and are restricted to a function class (e.g., linear) so we compute:

$$\hat{f} = \arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2$$

Each draw $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ results in different \hat{f}

Bias-Variance Tradeoff

$$\eta(x) = \mathbb{E}_{Y|X}[Y|X = x] \qquad \hat{f} = \arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2$$

$$\mathbb{E}_{Y|X}[\mathbb{E}_{\mathcal{D}}[(Y - \hat{f}_{\mathcal{D}}(x))^2]|X = x] = \mathbb{E}_{Y|X}[\mathbb{E}_{\mathcal{D}}[(Y - \eta(x) + \eta(x) - \hat{f}_{\mathcal{D}}(x))^2]|X = x]$$

Bias-Variance Tradeoff

$$\eta(x) = \mathbb{E}_{Y|X}[Y|X = x] \quad \hat{f} = \arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2$$

$$\begin{aligned} \mathbb{E}_{Y|X}[\mathbb{E}_{\mathcal{D}}[(Y - \hat{f}_{\mathcal{D}}(x))^2]|X = x] &= \mathbb{E}_{Y|X}[\mathbb{E}_{\mathcal{D}}[(Y - \eta(x) + \eta(x) - \hat{f}_{\mathcal{D}}(x))^2]|X = x] \\ &= \mathbb{E}_{Y|X} \left[\mathbb{E}_{\mathcal{D}}[(Y - \eta(x))^2 + 2(Y - \eta(x))(\eta(x) - \hat{f}_{\mathcal{D}}(x)) \right. \\ &\quad \left. + (\eta(x) - \hat{f}_{\mathcal{D}}(x))^2] | X = x \right] \\ &= \underbrace{\mathbb{E}_{Y|X}[(Y - \eta(x))^2 | X = x]}_{\text{irreducible error}} + \underbrace{\mathbb{E}_{\mathcal{D}}[(\eta(x) - \hat{f}_{\mathcal{D}}(x))^2]}_{\text{learning error}} \end{aligned}$$

irreducible error

Caused by stochastic
label noise

learning error

Caused by either using too “simple”
of a model or not enough
data to learn the model accurately

Bias-Variance Tradeoff

$$\eta(x) = \mathbb{E}_{Y|X}[Y|X = x]$$

$$\hat{f} = \arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2$$

$$\underline{\mathbb{E}_{\mathcal{D}}[(\eta(x) - \hat{f}_{\mathcal{D}}(x))^2]} = \mathbb{E}_{\mathcal{D}}[(\eta(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] + \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - \hat{f}_{\mathcal{D}}(x))^2]$$

Bias-Variance Tradeoff

$$\eta(x) = \mathbb{E}_{Y|X}[Y|X = x]$$

$$\hat{f} = \arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2$$

$$\begin{aligned} \underline{\mathbb{E}_{\mathcal{D}}[(\eta(x) - \hat{f}_{\mathcal{D}}(x))^2]} &= \mathbb{E}_{\mathcal{D}}[(\eta(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] + \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - \hat{f}_{\mathcal{D}}(x))^2] \\ &= \mathbb{E}_{\mathcal{D}}[(\eta(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)])^2 + 2(\eta(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)])(\mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - \hat{f}_{\mathcal{D}}(x)) \\ &\quad + (\mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - \hat{f}_{\mathcal{D}}(x))^2] \\ &= \underline{(\eta(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)])^2} + \underline{\mathbb{E}_{\mathcal{D}}[(\mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - \hat{f}_{\mathcal{D}}(x))^2]} \\ &\quad \text{biased squared} \qquad \qquad \qquad \text{variance} \end{aligned}$$

Bias-Variance Tradeoff

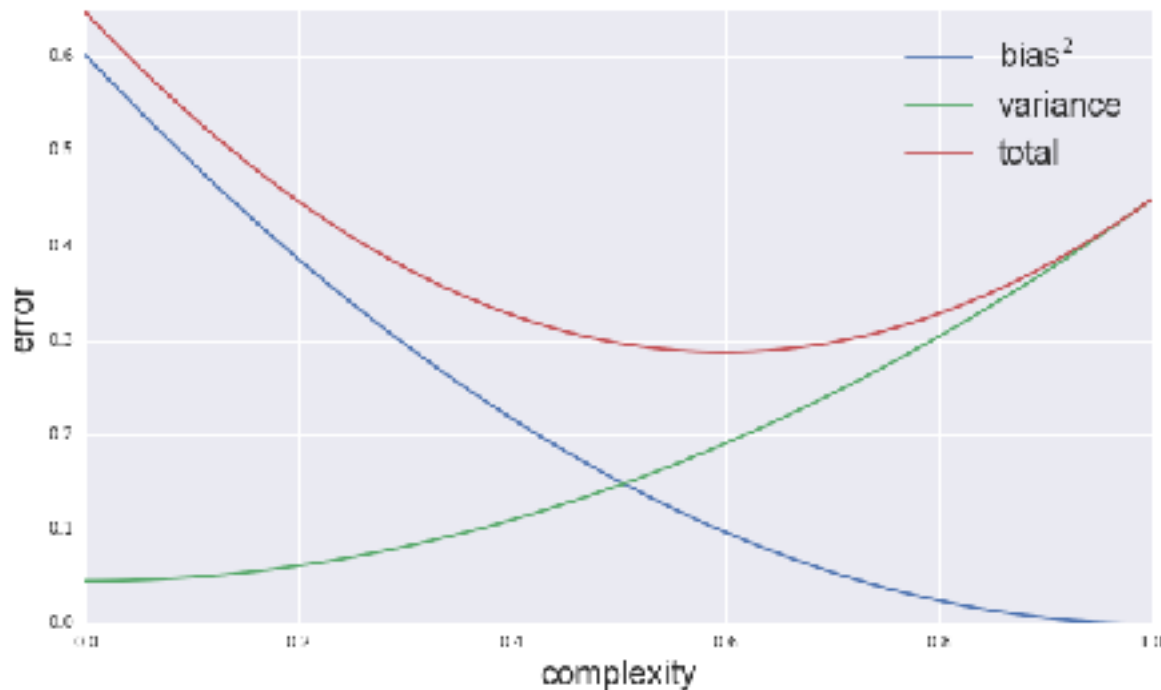
$$\mathbb{E}_{Y|X}[\mathbb{E}_{\mathcal{D}}[(Y - \hat{f}_{\mathcal{D}}(x))^2] | X = x] = \underbrace{\mathbb{E}_{Y|X}[(Y - \eta(x))^2 | X = x]}$$

irreducible error

$$+ \underbrace{(\eta(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)])^2}_{\text{biased squared}} + \underbrace{\mathbb{E}_{\mathcal{D}}[(\mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - \hat{f}_{\mathcal{D}}(x))^2]}_{\text{variance}}$$

biased squared

variance



Example: Linear LS $Y = \mathbf{X}w + \epsilon$

if $y_i = x_i^T w + \epsilon_i$ and $\epsilon_i \stackrel{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$

$$\hat{w}_{MLE} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y} = w + (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \epsilon$$

$$\eta(x) = \mathbb{E}_{Y|X} [Y | X = x]$$

$$\hat{f}_{\mathcal{D}}(x) =$$

Example: Linear LS $\mathbf{Y} = \mathbf{X}w + \epsilon$

if $y_i = x_i^T w + \epsilon_i$ and $\epsilon_i \stackrel{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$

$$\hat{w}_{MLE} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y} = w + (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \epsilon$$

$$\eta(x) = \mathbb{E}_{Y|X}[Y|X = x]$$

$$\hat{f}_{\mathcal{D}}(x) = \hat{w}^T x = w^T x + \epsilon^T \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} x$$

$$\mathbb{E}_{XY}[(Y - \eta(x))^2 | X = x] = \sigma^2$$

irreducible error

$$(\eta(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)])^2 = 0$$

biased squared

Example: Linear LS $Y = Xw + \epsilon$

if $y_i = x_i^T w + \epsilon_i$ and $\epsilon_i \stackrel{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$

$$\hat{w}_{MLE} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y} = w + (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \epsilon$$

$$\hat{f}_{\mathcal{D}}(x) = \hat{w}^T x = w^T x + \epsilon^T \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} x$$

$$\mathbb{E}_{\mathcal{D}} [(\mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - \hat{f}_{\mathcal{D}}(x))^2] =$$

variance

Example: Linear LS $Y = Xw + \epsilon$

if $y_i = x_i^T w + \epsilon_i$ and $\epsilon_i \stackrel{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$

$$\hat{w}_{MLE} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y} = w + (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \epsilon$$

$$\hat{f}_{\mathcal{D}}(x) = \hat{w}^T x = w^T x + \epsilon^T \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} x$$

$$\begin{aligned} \underbrace{\mathbb{E}_{\mathcal{D}}[(\mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - \hat{f}_{\mathcal{D}}(x))^2]}_{\text{variance}} &= \mathbb{E}_{\mathcal{D}}[x^T (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \epsilon \epsilon^T \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} x] \\ &= \mathbb{E}_{\mathcal{D}}[\sigma^2 x^T (\mathbf{X}^T \mathbf{X})^{-1} x] \\ &= \sigma^2 \mathbb{E}_{\mathcal{D}}[\text{Trace}((\mathbf{X}^T \mathbf{X})^{-1} x x^T)] \end{aligned}$$

$$\mathbf{X}^T \mathbf{X} = \sum_{i=1}^n x_i x_i^T \xrightarrow{n \text{ large}} n \Sigma \quad \Sigma = \mathbb{E}[X X^T], \quad X \sim P_X$$

$$\mathbb{E}_{X=x}[\mathbb{E}_{\mathcal{D}}[(\mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - \hat{f}_{\mathcal{D}}(x))^2]] = \frac{\sigma^2}{n} \mathbb{E}_X[\text{Trace}(\Sigma^{-1} X X^T)] = \frac{d\sigma^2}{n}$$

Example: Linear LS $\mathbf{Y} = \mathbf{X}w + \epsilon$

if $y_i = x_i^T w + \epsilon_i$ and $\epsilon_i \stackrel{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$

$$\hat{w}_{MLE} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y} = w + (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \epsilon$$

$$\eta(x) = \mathbb{E}_{Y|X}[Y|X = x]$$

$$\hat{f}_{\mathcal{D}}(x) = \hat{w}^T x = w^T x + \epsilon^T \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} x$$

$$\underbrace{\mathbb{E}_{XY}[(Y - \eta(x))^2 | X = x]}_{\text{irreducible error}} = \sigma^2 \qquad \underbrace{(\eta(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)])^2}_{\text{biased squared}} = 0$$

irreducible error

biased squared

$$\mathbb{E}_{X=x} \left[\underbrace{\mathbb{E}_{\mathcal{D}}[(\mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - \hat{f}_{\mathcal{D}}(x))^2]}_{\text{variance}} \right] = \frac{d\sigma^2}{n}$$

variance



Overfitting

Machine Learning – CSE546

Kevin Jamieson

University of Washington

Oct 4, 2018

Bias-Variance Tradeoff



- Choice of hypothesis class introduces learning bias
 - More complex class \rightarrow less bias
 - More complex class \rightarrow more variance
- But in practice??

Bias-Variance Tradeoff

- Choice of hypothesis class introduces learning bias
 - More complex class → less bias
 - More complex class → more variance
- But in practice??
- Before we saw how increasing the feature space can increase the complexity of the learned estimator:

$$\mathcal{F}_1 \subset \mathcal{F}_2 \subset \mathcal{F}_3 \subset \dots$$

$$\hat{f}_{\mathcal{D}}^{(k)} = \arg \min_{f \in \mathcal{F}_k} \frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - f(x_i))^2$$

Complexity grows as k grows

Training set error as a function of model complexity

$$\mathcal{F}_1 \subset \mathcal{F}_2 \subset \mathcal{F}_3 \subset \dots \quad \mathcal{D} \stackrel{i.i.d.}{\sim} P_{XY}$$

$$\hat{f}_{\mathcal{D}}^{(k)} = \arg \min_{f \in \mathcal{F}_k} \frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - f(x_i))^2$$

TRAIN error:

$$\frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - \hat{f}_{\mathcal{D}}^{(k)}(x_i))^2$$

TRUE error:

$$\mathbb{E}_{XY} [(Y - \hat{f}_{\mathcal{D}}^{(k)}(X))^2]$$

Training set error as a function of model complexity

$$\mathcal{F}_1 \subset \mathcal{F}_2 \subset \mathcal{F}_3 \subset \dots \quad \mathcal{D} \stackrel{i.i.d.}{\sim} P_{XY}$$

$$\hat{f}_{\mathcal{D}}^{(k)} = \arg \min_{f \in \mathcal{F}_k} \frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - f(x_i))^2$$

TRAIN error:

$$\frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - \hat{f}_{\mathcal{D}}^{(k)}(x_i))^2$$

TRUE error:

$$\mathbb{E}_{XY} [(Y - \hat{f}_{\mathcal{D}}^{(k)}(X))^2]$$

TEST error:

$$\mathcal{T} \stackrel{i.i.d.}{\sim} P_{XY}$$

$$\frac{1}{|\mathcal{T}|} \sum_{(x_i, y_i) \in \mathcal{T}} (y_i - \hat{f}_{\mathcal{D}}^{(k)}(x_i))^2$$

Important: $\mathcal{D} \cap \mathcal{T} = \emptyset$

Complexity (k)

Training set error as a function of model complexity

$$\mathcal{F}_1 \subset \mathcal{F}_2 \subset \mathcal{F}_3 \subset \dots \quad \mathcal{D} \stackrel{i.i.d.}{\sim} P_{XY}$$

$$\hat{f}_{\mathcal{D}}^{(k)} = \arg \min_{f \in \mathcal{F}_k} \frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - f(x_i))^2$$

TRAIN error:

$$\frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - \hat{f}_{\mathcal{D}}^{(k)}(x_i))^2$$

TRUE error:

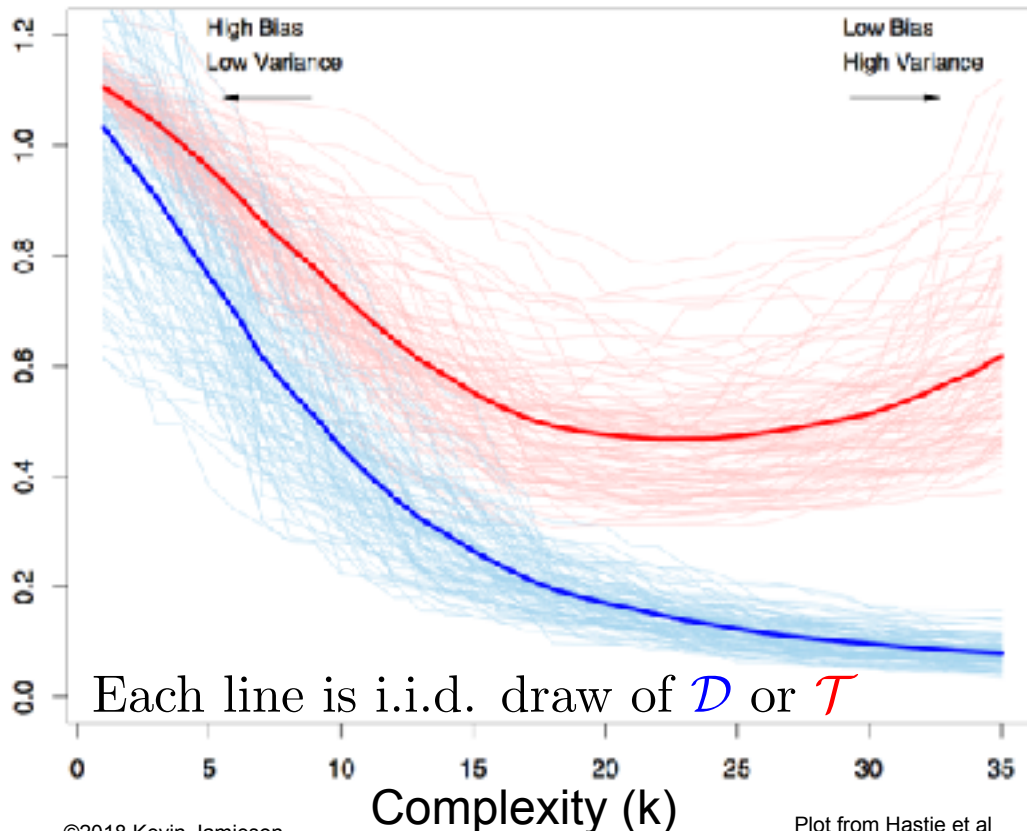
$$\mathbb{E}_{XY} [(Y - \hat{f}_{\mathcal{D}}^{(k)}(X))^2]$$

TEST error:

$$\mathcal{T} \stackrel{i.i.d.}{\sim} P_{XY}$$

$$\frac{1}{|\mathcal{T}|} \sum_{(x_i, y_i) \in \mathcal{T}} (y_i - \hat{f}_{\mathcal{D}}^{(k)}(x_i))^2$$

Important: $\mathcal{D} \cap \mathcal{T} = \emptyset$



Training set error as a function of model complexity

$$\mathcal{F}_1 \subset \mathcal{F}_2 \subset \mathcal{F}_3 \subset \dots \quad \mathcal{D} \stackrel{i.i.d.}{\sim} P_{XY}$$
$$\hat{f}_{\mathcal{D}}^{(k)} = \arg \min_{f \in \mathcal{F}_k} \frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - f(x_i))^2$$

TRAIN error is **optimistically biased** because it is evaluated on the data it trained on. **TEST error** is **unbiased** only if \mathcal{T} is never used to train the model or even pick the complexity k .

TRAIN error:

$$\frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - \hat{f}_{\mathcal{D}}^{(k)}(x_i))^2$$

TRUE error:

$$\mathbb{E}_{XY} [(Y - \hat{f}_{\mathcal{D}}^{(k)}(X))^2]$$

TEST error:

$$\mathcal{T} \stackrel{i.i.d.}{\sim} P_{XY}$$
$$\frac{1}{|\mathcal{T}|} \sum_{(x_i, y_i) \in \mathcal{T}} (y_i - \hat{f}_{\mathcal{D}}^{(k)}(x_i))^2$$

Important: $\mathcal{D} \cap \mathcal{T} = \emptyset$

Test set error

- Given a dataset, **randomly** split it into two parts:
 - Training data: \mathcal{D}
 - Test data: \mathcal{T}
- Use **training data** to learn predictor
 - e.g., $\frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - \hat{f}_{\mathcal{D}}^{(k)}(x_i))^2$
 - use **training data** to pick complexity k
- Use **test data** to report predicted performance

$$\text{Important: } \mathcal{D} \cap \mathcal{T} = \emptyset$$

$$\frac{1}{|\mathcal{T}|} \sum_{(x_i, y_i) \in \mathcal{T}} (y_i - \hat{f}_{\mathcal{D}}^{(k)}(x_i))^2$$

How many points do I use for training/testing?

- Very hard question to answer!
 - Too few training points, learned model is bad
 - Too few test points, you never know if you reached a good solution
- Bounds, such as Hoeffding's inequality can help:

$$P(|\hat{\theta} - \theta^*| \geq \epsilon) \leq 2e^{-2N\epsilon^2}$$

- More on this later the quarter, but still hard to answer
- Typically:
 - If you have a reasonable amount of data 90/10 splits are common
 - If you have little data, then you need to get fancy (e.g., bootstrapping)



Regularization

Machine Learning – CSE546

Kevin Jamieson

University of Washington

October 4, 2016

Regularization in Linear Regression

Recall Least Squares: $\hat{w}_{LS} = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2$

$$= \arg \min_w (\mathbf{y} - \mathbf{X}w)^T (\mathbf{y} - \mathbf{X}w)$$

when $(\mathbf{X}^T \mathbf{X})^{-1}$ exists.... $= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$

Regularization in Linear Regression

Recall Least Squares: $\hat{w}_{LS} = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2$

$$= \arg \min_w (\mathbf{y} - \mathbf{X}w)^T (\mathbf{y} - \mathbf{X}w)$$

In general:

$$= \arg \min_w w^T (\mathbf{X}^T \mathbf{X}) w - 2y^T \mathbf{X}w$$

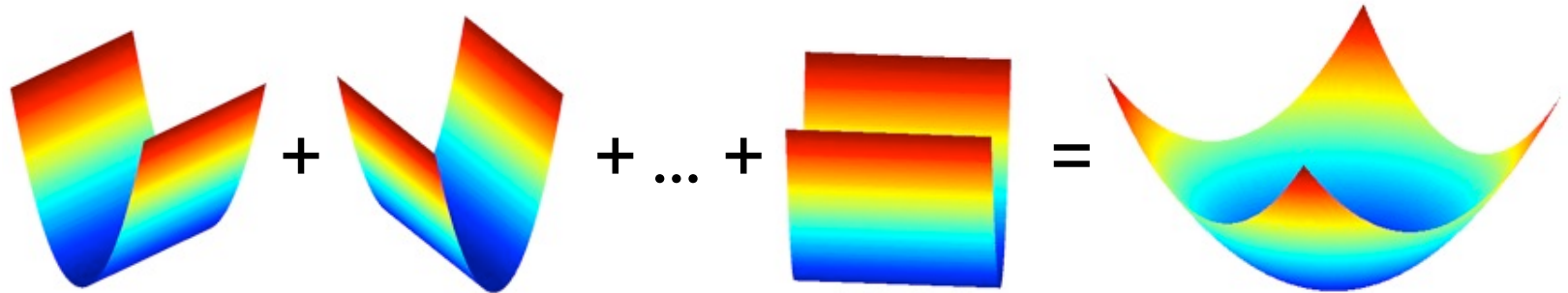
Regularization in Linear Regression

Recall Least Squares: $\hat{w}_{LS} = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2$

$$= \arg \min_w (\mathbf{y} - \mathbf{X}w)^T (\mathbf{y} - \mathbf{X}w)$$

In general:

$$= \arg \min_w w^T (\mathbf{X}^T \mathbf{X})w - 2\mathbf{y}^T \mathbf{X}w$$



$$(y_1 - x_1^T w)^2 + (y_2 - x_2^T w)^2 + \dots + (y_n - x_n^T w)^2 = \sum_{i=1}^n (y_i - x_i^T w)^2$$

What if $x_i \in \mathbb{R}^d$ and $d > n$?

Regularization in Linear Regression

Recall Least Squares: $\hat{w}_{LS} = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2$

When $x_i \in \mathbb{R}^d$ and $d > n$ the objective function is flat in some directions:



Regularization in Linear Regression

Recall Least Squares: $\hat{w}_{LS} = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2$

When $x_i \in \mathbb{R}^d$ and $d > n$ the objective function is flat in some directions:

Implies optimal solution is *underconstrained* and unstable due to lack of curvature:

- small changes in training data result in large changes in solution
- often the *magnitudes* of w are “very large”

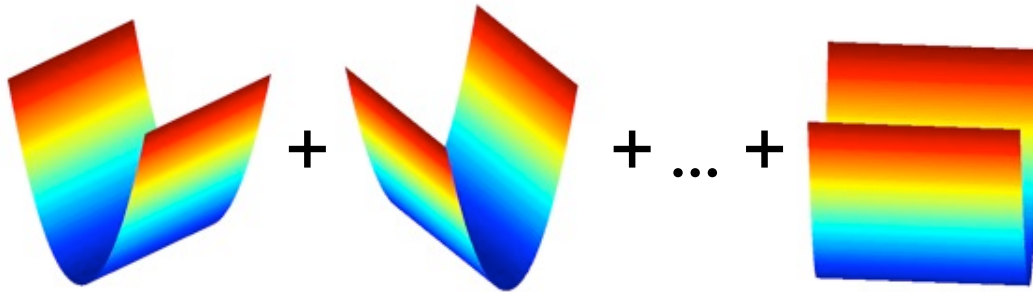


Regularization imposes “simpler” solutions by a “complexity” penalty

Ridge Regression

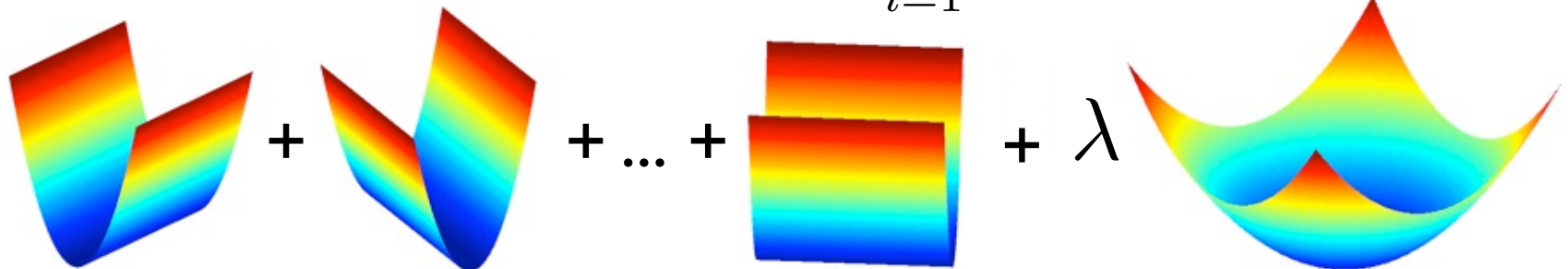
- Old Least squares objective:

$$\hat{w}_{LS} = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2$$



- Ridge Regression objective:

$$\hat{w}_{ridge} = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2 + \lambda \|w\|_2^2$$



Minimizing the Ridge Regression Objective

$$\hat{w}_{ridge} = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2 + \lambda \|w\|_2^2$$

Shrinkage Properties

$$\epsilon \sim \mathcal{N}(0, \sigma^2 I)$$

$$\hat{w}_{ridge} = (\mathbf{X}^T \mathbf{X} + \lambda I)^{-1} \mathbf{X}^T \mathbf{y}$$

- **Assume:** $\mathbf{X}^T \mathbf{X} = nI$ **and** $\mathbf{y} = \mathbf{X}w + \epsilon$

Shrinkage Properties

$$\epsilon \sim \mathcal{N}(0, \sigma^2 I)$$

$$\hat{w}_{ridge} = (\mathbf{X}^T \mathbf{X} + \lambda I)^{-1} \mathbf{X}^T \mathbf{y}$$

- **Assume:** $\mathbf{X}^T \mathbf{X} = nI$ and $\mathbf{y} = \mathbf{X}w + \epsilon$

$$\begin{aligned} \hat{w}_{ridge} &= (\mathbf{X}^T \mathbf{X} + \lambda I)^{-1} \mathbf{X}^T (\mathbf{X}w + \epsilon) \\ &= \frac{n}{n + \lambda} w + \frac{1}{n + \lambda} \mathbf{X}^T \epsilon \end{aligned}$$

$$\mathbb{E} \|\hat{w}_{ridge} - w\|^2 = \frac{\lambda^2}{(n + \lambda)^2} \|w\|^2 + \frac{dn\sigma^2}{(n + \lambda)^2} \quad \lambda^* = \frac{d\sigma^2}{\|w\|^2}$$

Ridge Regression: Effect of Regularization

$$\hat{w}_{ridge} = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2 + \lambda \|w\|_2^2$$

- Solution is indexed by the regularization parameter λ
- Larger λ
- Smaller λ
- As $\lambda \rightarrow 0$
- As $\lambda \rightarrow \infty$

Ridge Regression: Effect of Regularization

$\mathcal{D} \stackrel{i.i.d.}{\sim} P_{XY}$

$$\hat{w}_{\mathcal{D},ridge}^{(\lambda)} = \arg \min_w \frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - x_i^T w)^2 + \lambda \|w\|_2^2$$

TRAIN error:

$$\frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - x_i^T \hat{w}_{\mathcal{D},ridge}^{(\lambda)})^2$$

TRUE error:

$$\mathbb{E}[(Y - X^T \hat{w}_{\mathcal{D},ridge}^{(\lambda)})^2]$$

TEST error:

$\mathcal{T} \stackrel{i.i.d.}{\sim} P_{XY}$

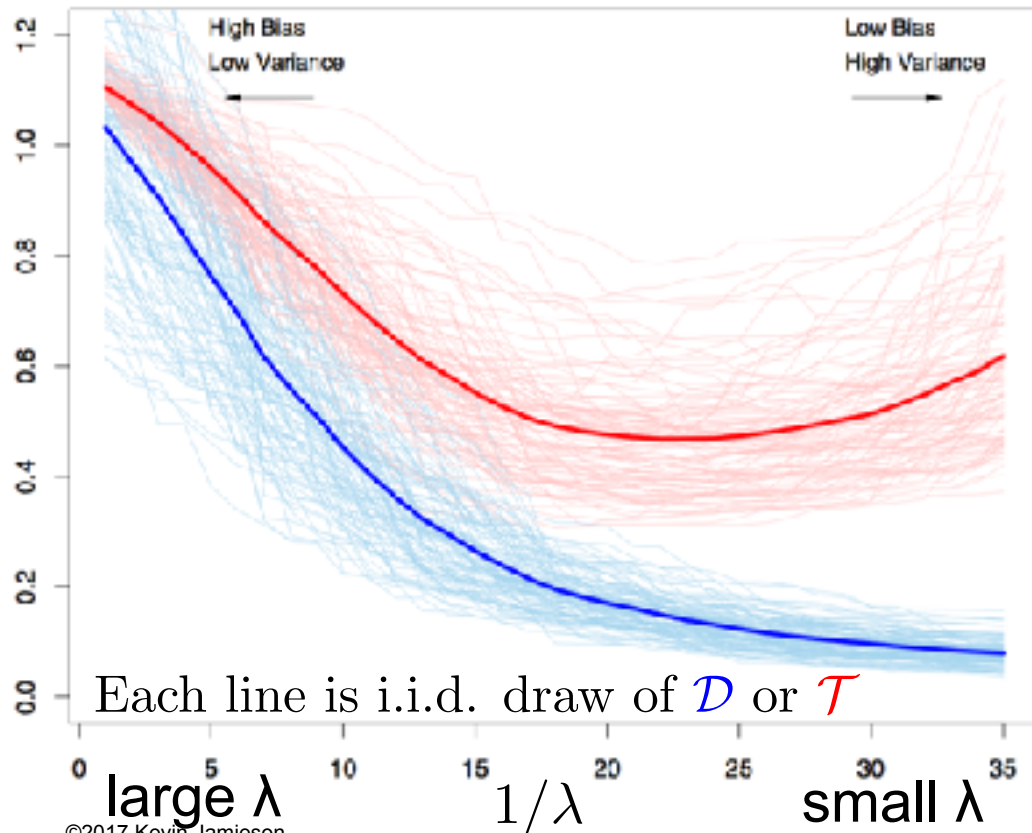
$$\frac{1}{|\mathcal{T}|} \sum_{(x_i, y_i) \in \mathcal{T}} (y_i - x_i^T \hat{w}_{\mathcal{D},ridge}^{(\lambda)})^2$$

Important: $\mathcal{D} \cap \mathcal{T} = \emptyset$

Ridge Regression: Effect of Regularization

$$\mathcal{D} \stackrel{i.i.d.}{\sim} P_{XY}$$

$$\hat{w}_{\mathcal{D},ridge}^{(\lambda)} = \arg \min_w \frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - x_i^T w)^2 + \lambda \|w\|_2^2$$



TRAIN error:

$$\frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - x_i^T \hat{w}_{\mathcal{D},ridge}^{(\lambda)})^2$$

TRUE error:

$$\mathbb{E}[(Y - X^T \hat{w}_{\mathcal{D},ridge}^{(\lambda)})^2]$$

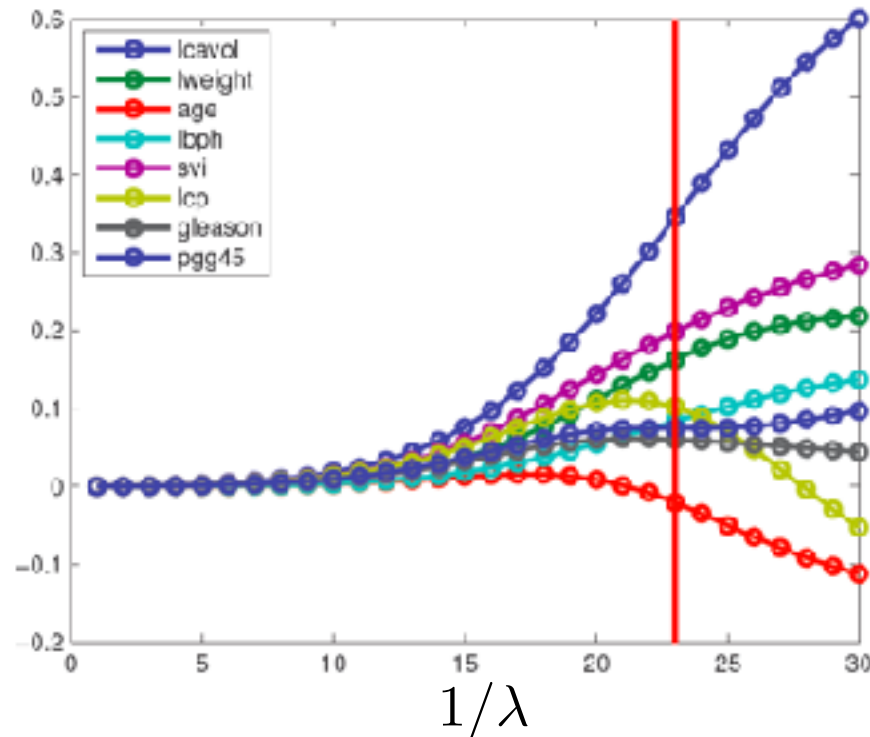
TEST error:

$$\mathcal{T} \stackrel{i.i.d.}{\sim} P_{XY}$$

$$\frac{1}{|\mathcal{T}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - x_i^T \hat{w}_{\mathcal{D},ridge}^{(\lambda)})^2$$

Important: $\mathcal{D} \cap \mathcal{T} = \emptyset$

Ridge Coefficient Path



From
Kevin Murphy
textbook

- Typical approach: select λ using cross validation, up next

What you need to know...

- Regularization
 - Penalizes for complex models
- Ridge regression
 - L_2 penalized least-squares regression
 - Regularization parameter trades off model complexity with training error



Cross-Validation

Machine Learning – CSE546

Kevin Jamieson

University of Washington

October 4, 2016

How... How... How????????

- *How do we pick the regularization constant λ ...*
- *How do we pick the number of basis functions...*
- We could use the test data, but...

(LOO) Leave-one-out cross validation

- Consider a **validation set with 1 example**:
 - D – training data
 - $D \setminus j$ – training data with j th data point $(\mathbf{x}_j, \mathbf{y}_j)$ moved to validation set
- **Learn classifier $f_{D \setminus j}$ with $D \setminus j$ dataset**
- **Estimate true error** as squared error on predicting \mathbf{y}_j :
 - Unbiased estimate of error_{true} $(f_{D \setminus j})!$

□

(LOO) Leave-one-out cross validation

- Consider a **validation set with 1 example**:
 - D – training data
 - $D \setminus j$ – training data with j th data point $(\mathbf{x}_j, \mathbf{y}_j)$ moved to validation set
- **Learn classifier $f_{D \setminus j}$ with $D \setminus j$ dataset**
- **Estimate true error** as squared error on predicting \mathbf{y}_j :
 - Unbiased estimate of error_{true}($f_{D \setminus j}$)!
- **LOO cross validation**: Average over all data points j :
 - For each data point you leave out, learn a new classifier $f_{D \setminus j}$
 - Estimate error as:

$$\text{error}_{LOO} = \frac{1}{n} \sum_{j=1}^n (y_j - f_{D \setminus j}(x_j))^2$$

LOO cross validation is (almost) unbiased estimate of true error of h_D !

- When computing **LOOCV error**, we only use **$N-1$ data points**
 - So it's not estimate of true error of learning with N data points
 - Usually pessimistic, though – learning with less data typically gives worse answer
- **LOO is almost unbiased! Use LOO error for model selection!!!**
 - **E.g., picking λ**

Computational cost of LOO

- Suppose you have 100,000 data points
- You implemented a great version of your learning algorithm
 - Learns in only 1 second
- Computing LOO will take about 1 day!!!
 -

Use k -fold cross validation

- Randomly divide training data into k equal parts

- D_1, \dots, D_k

- For each i

- Learn classifier $f_{D \setminus D_i}$ using data point not in D_i

- Estimate error of $f_{D \setminus D_i}$ on validation set D_i :

$$\text{error}_{D_i} = \frac{1}{|D_i|} \sum_{(x_j, y_j) \in D_i} (y_j - f_{D \setminus D_i}(x_j))^2$$



Use k -fold cross validation

- Randomly divide training data into k equal parts

- D_1, \dots, D_k

- For each i

- Learn classifier $f_{D \setminus D_i}$ using data point not in D_i

- Estimate error of $f_{D \setminus D_i}$ on validation set D_i :

$$\text{error}_{D_i} = \frac{1}{|D_i|} \sum_{(x_j, y_j) \in D_i} (y_j - f_{D \setminus D_i}(x_j))^2$$



- k -fold cross validation error is average over data splits:

$$\text{error}_{k\text{-fold}} = \frac{1}{k} \sum_{i=1}^k \text{error}_{D_i}$$

- k -fold cross validation properties:

- Much faster to compute than LOO

- More (pessimistically) biased – using much less data, only $n(k-1)/k$

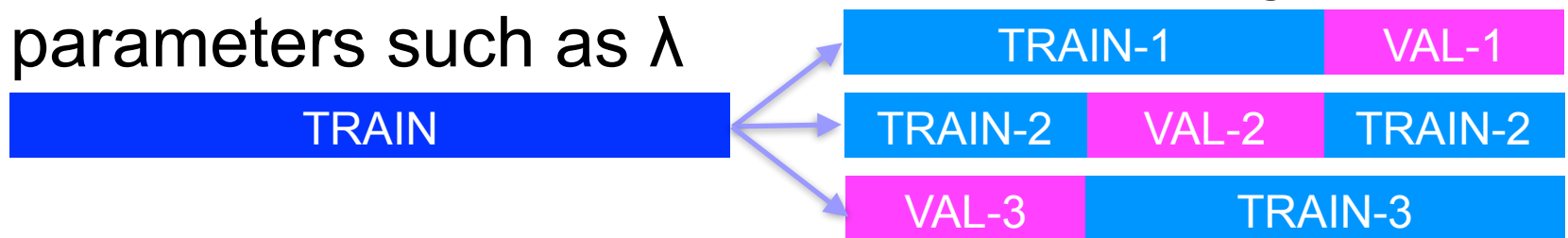
- Usually, $k = 10$

Recap

- Given a dataset, begin by splitting into



- Model selection:** Use k-fold cross-validation on **TRAIN** to train predictor and choose magic parameters such as λ



- Model assessment:** Use **TEST** to assess the accuracy of the model you output
 - Never ever ever ever ever train or choose parameters based on the test data

Example

- Given 10,000-dimensional data and n examples, we pick a subset of 50 dimensions that have the highest correlation with labels in the training set:

50 indices j that have largest $\frac{|\sum_{i=1}^n x_{i,j} y_i|}{\sqrt{\sum_{i=1}^n x_{i,j}^2}}$

- After picking our 50 features, we then use CV to train ridge regression with regularization λ
- What's wrong with this procedure?

Recap

- Learning is...
 - Collect some data
 - E.g., housing info and sale price
 - Randomly split dataset into TRAIN, VAL, and TEST
 - E.g., 80%, 10%, and 10%, respectively
 - Choose a hypothesis class or model
 - E.g., linear with non-linear transformations
 - Choose a loss function
 - E.g., least squares with ridge regression penalty on TRAIN
 - Choose an optimization procedure
 - E.g., set derivative to zero to obtain estimator, cross-validation on VAL to pick num. features and amount of regularization
 - Justifying the accuracy of the estimate
 - E.g., report TEST error with Bootstrap confidence interval