

Introduction to the analysis of learning algorithms

— Does Bayesianism save you?

Ryota Tomioka

Toyota Technological Institute at Chicago

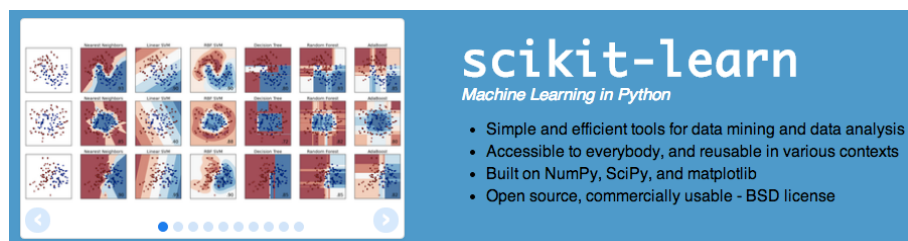
tomioka@ttic.edu

Exercise materials:

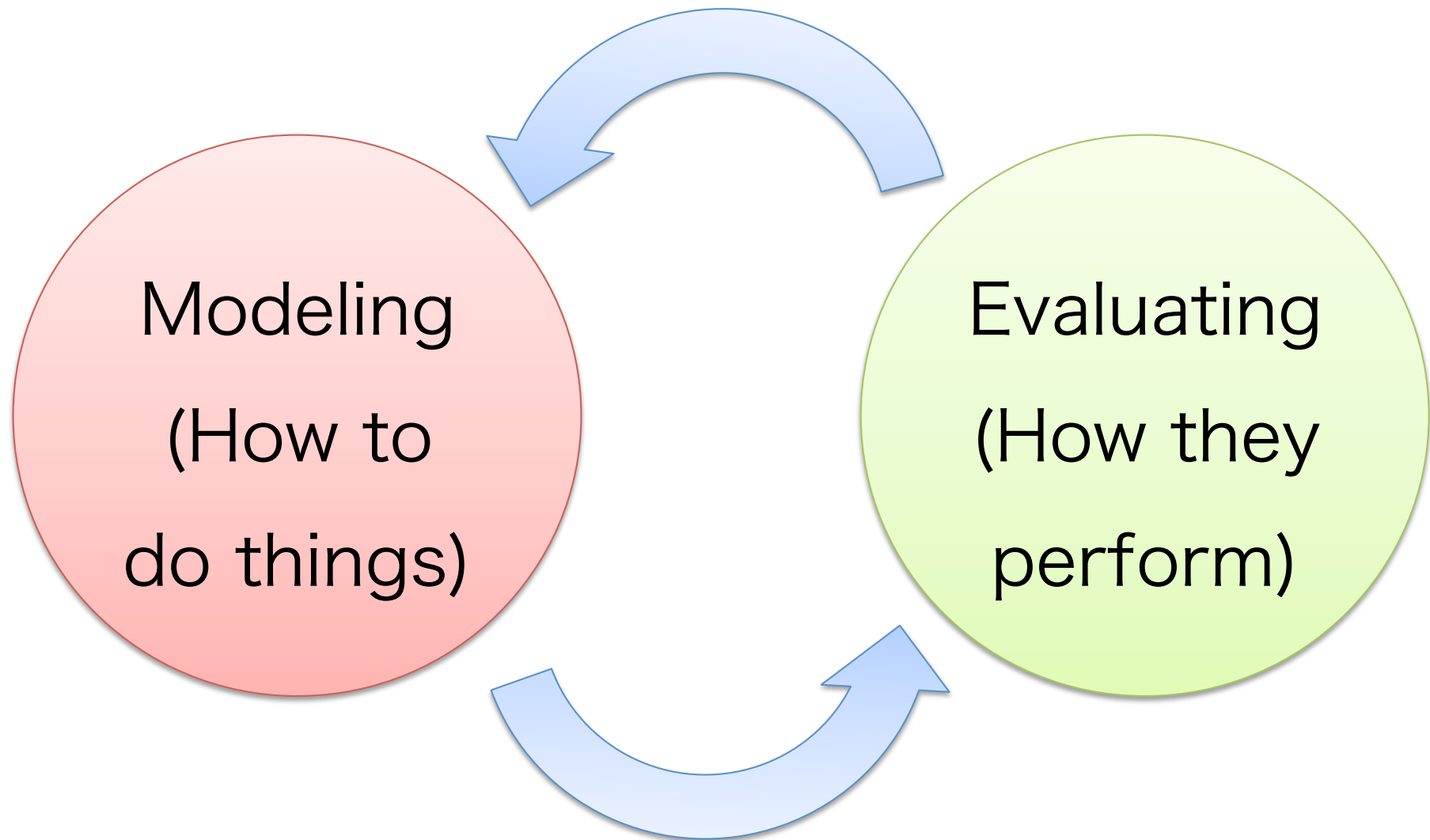
<https://github.com/ryotat/dtuphd14>

Why learn theory?

- Think about your market value
 - Many good algorithms
 - Many good implementations (e.g., scikit-learn, mahout, Stan, infer.net, Church,...



Two sides of machine learning



About this lecture

- I will try to make it as interactive as possible
- If you don't get something, probably I am doing something wrong.
- So, please ask questions
 - It will not only help you but also help others.
 - It will help you stay awake!

Key questions

- Learning
 - the goal is to generalize to a new test example from a limited number of training instances
- What is over-fitting?
- How do we avoid over-fitting?
- Does Bayesian methods avoid over-fitting?

The key is to understand an estimator
as a *random variable*

What we will cover

- First part
 - Ridge regression
 - Bias-variance decomposition
 - Model selection
 - Mallows' C_L
 - Leave-one-out cross validation

Second part

- Bayesian regression
- PAC Bayes theory

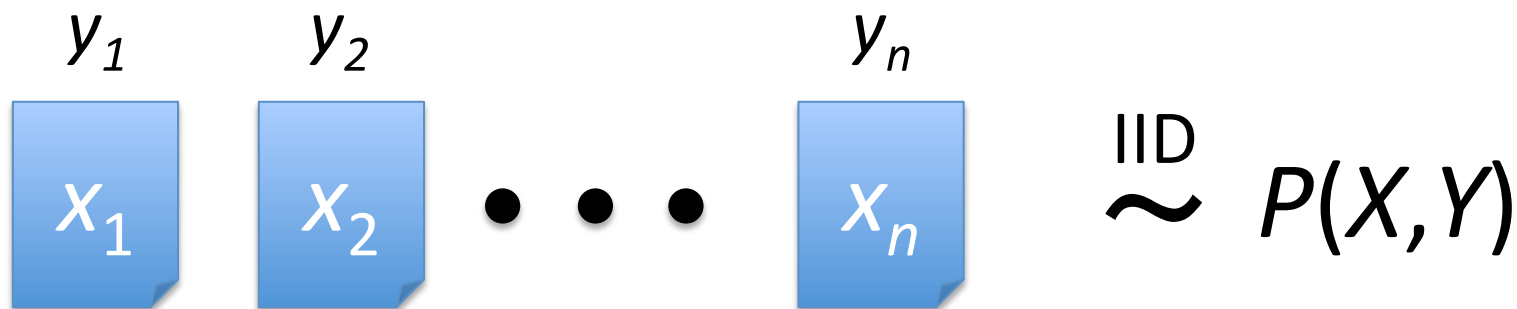
Ridge Regression

Key idea:

Estimator is a random variable

Problem Setting

- Training examples: (x_i, y_i) ($i=1, \dots, n$), $x_i \in \mathbb{R}^d$

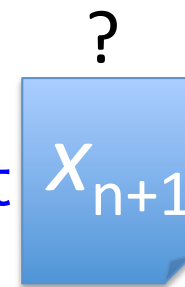


- Goal
 - Learn a linear function

$$f(x) = w^T x \quad (w \in \mathbb{R}^p)$$

that predicts the output y_{n+1} for a **test point** x_{n+1}

$$(x_{n+1}, y_{n+1}) \sim P(X, Y)$$



- Note that the **test point** is not included in the training examples (**We want generalization!**)

Ridge Regression

- Solve the minimization problem

$$\underset{\mathbf{w} \in \mathbb{R}^d}{\text{minimize}} \quad \underbrace{\sum_{i=1}^n (y_i - \mathbf{x}_i^\top \mathbf{w})^2}_{\text{Training error}} + \underbrace{\lambda \|\mathbf{w}\|^2}_{\text{Regularization (ridge) term}} \quad (\lambda: \text{regularization const.})$$

Note: Can be interpreted as a Maximum A Posteriori (MAP) estimation
– Gaussian likelihood with Gaussian prior.

Ridge Regression

- More compactly

$$\underset{\mathbf{w}}{\text{minimize}} \quad \underbrace{\|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2}_{\text{Training error}} + \underbrace{\lambda\|\mathbf{w}\|^2}_{\text{Regularization (ridge) term}}$$

Training error

Regularization (ridge) term
(λ : regularization const.)

Target
output

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

Design
matrix

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_n^\top \end{pmatrix}$$

Note: Can be interpreted as a Maximum A Posteriori (MAP) estimation
– Gaussian likelihood with Gaussian prior.

Designing the design matrix

- Columns of X can be different sources of info
 - e.g., predicting the price of an apartment

$$\mathbf{X} = \left(\begin{array}{c} \text{Size} \\ \text{\#rooms} \\ \text{Bathroom} \\ \text{Sunlight} \\ \text{Neighborhood} \\ \text{Train st.} \end{array} \right)$$

- Columns of X can also be nonlinear
 - e.g., polynomial regression

$$\mathbf{X} = \begin{pmatrix} x_1^p & \cdots & x_1^2 & x_1 & 1 \\ x_2^p & \cdots & x_2^2 & x_2 & 1 \\ \vdots & & & & \vdots \\ x_n^p & \cdots & x_n^2 & x_n & 1 \end{pmatrix}$$

Solving ridge regression

- Take the gradient, and solve

$$-X^{\top} (y - Xw) + \lambda w = 0$$

which gives

$$\hat{w} = (X^{\top} X + \lambda I_d)^{-1} X^{\top} y$$

(I_d : $d \times d$ identity matrix)

The solution can also be written as (exercise)

$$\hat{w} = X^{\top} (X X^{\top} + \lambda I_n)^{-1} y$$

Example: polynomial fitting

- Degree $d-1$ polynomial model

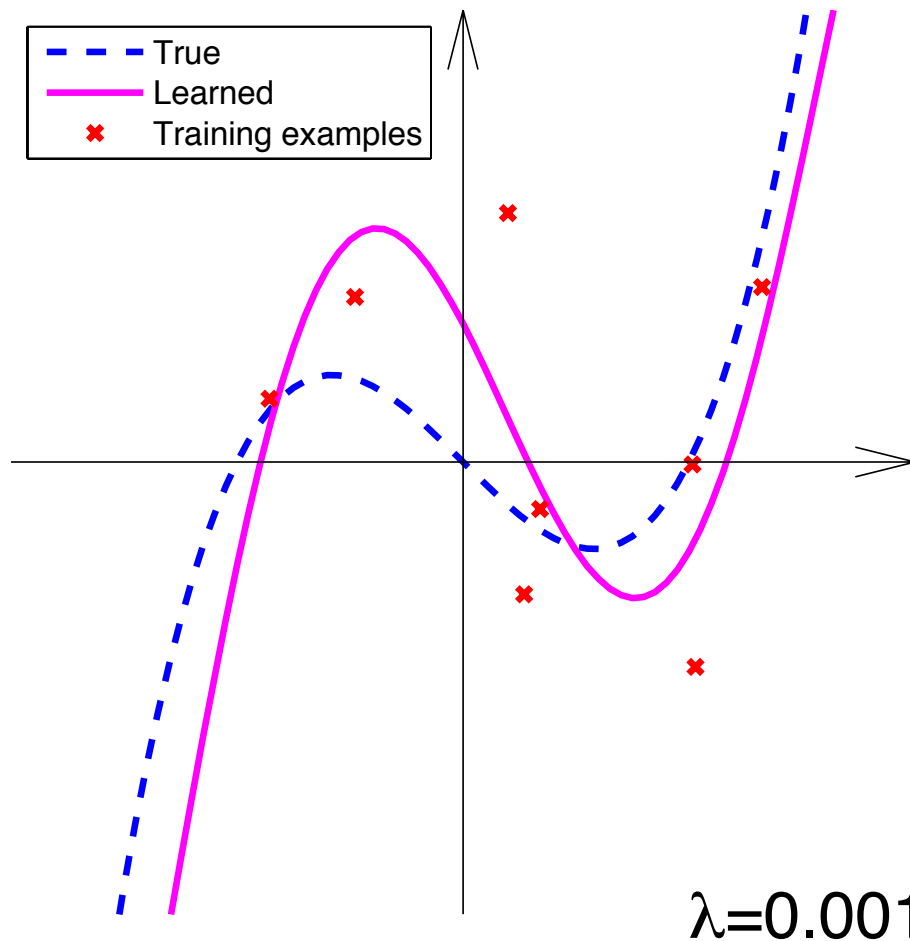
$$y = w_1 x^{d-1} + \dots + w_{d-1} x + w_d + \text{noise}$$

$$= \begin{pmatrix} x^{d-1} & \dots & x & 1 \end{pmatrix} \begin{pmatrix} w_1 \\ \vdots \\ w_{d-1} \\ w_d \end{pmatrix} + \text{noise}$$

Design matrix:

$$\mathbf{X} = \begin{pmatrix} x_1^{d-1} & \dots & x_1^2 & x_1 & 1 \\ x_2^{d-1} & \dots & x_2^2 & x_2 & 1 \\ \vdots & & & & \vdots \\ x_n^{d-1} & \dots & x_n^2 & x_n & 1 \end{pmatrix}$$

Example: 5th-order polynomial fitting



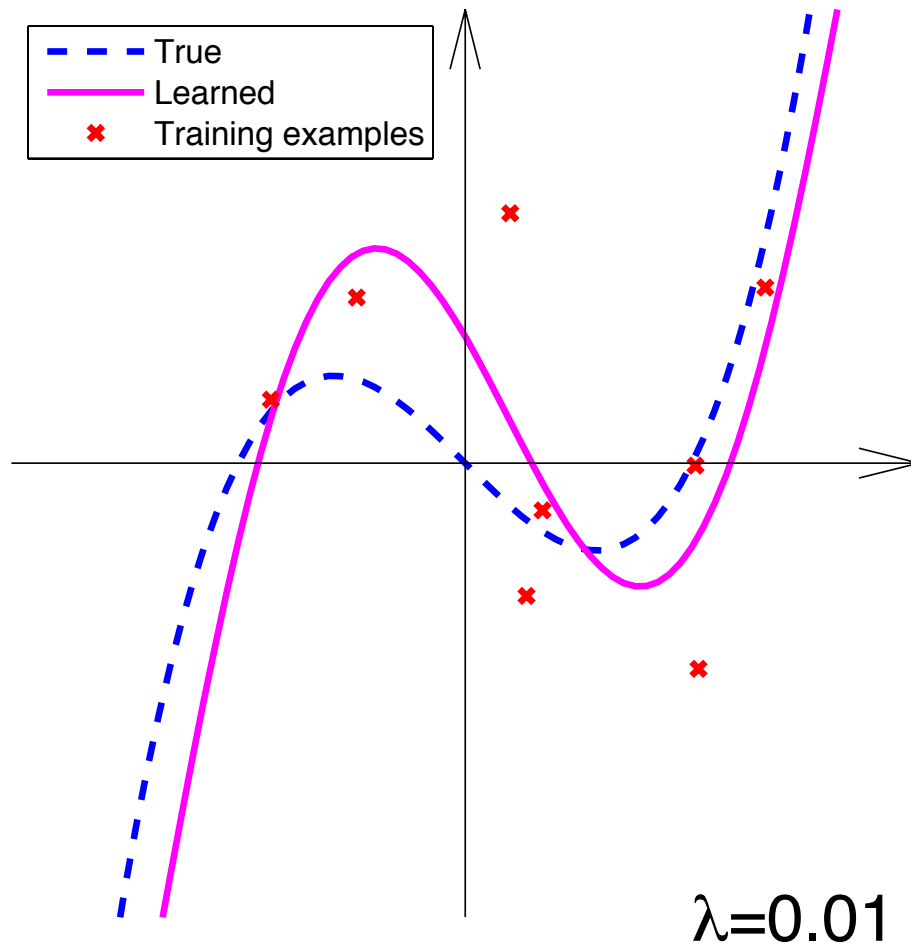
True

$$w^* = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ -1 \\ 0 \end{pmatrix}$$

Learned

$$w = \begin{pmatrix} -0.36 \\ 0.30 \\ 2.32 \\ -1.34 \\ -1.93 \\ 0.61 \end{pmatrix}$$

Example: 5th-order polynomial fitting



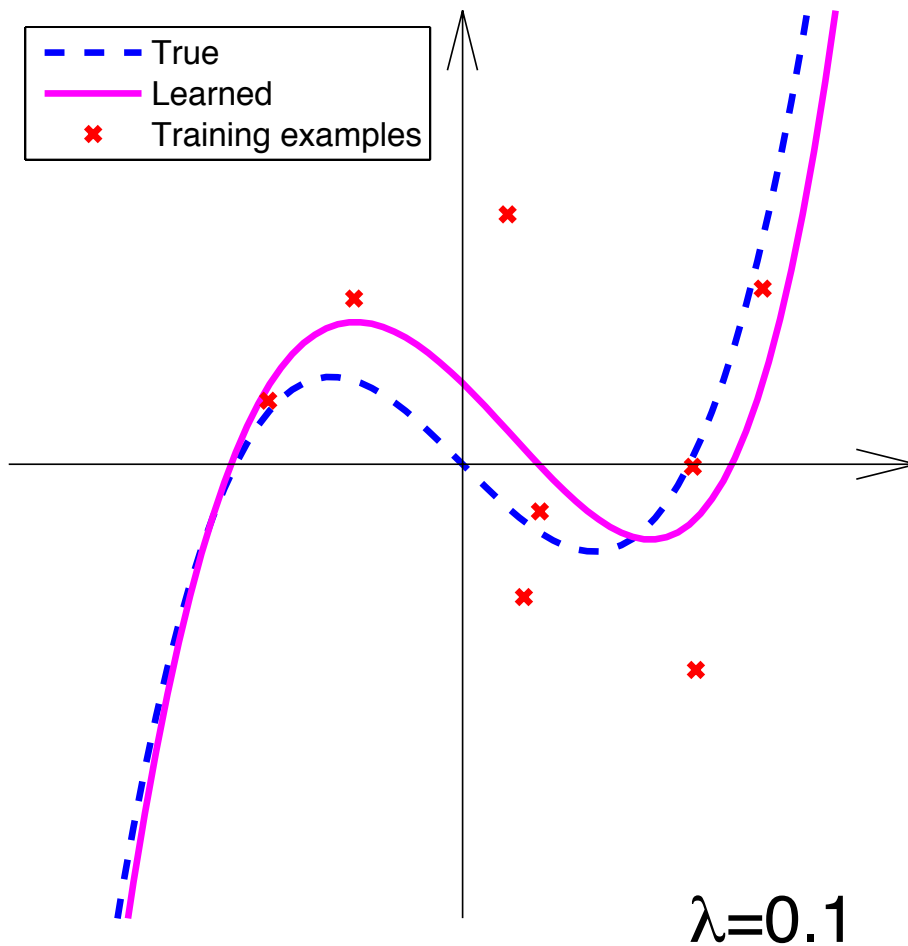
True

$$w^* = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ -1 \\ 0 \end{pmatrix}$$

Learned

$$w = \begin{pmatrix} -0.27 \\ 0.25 \\ 1.99 \\ -1.16 \\ -1.73 \\ 0.56 \end{pmatrix}$$

Example: 5th-order polynomial fitting



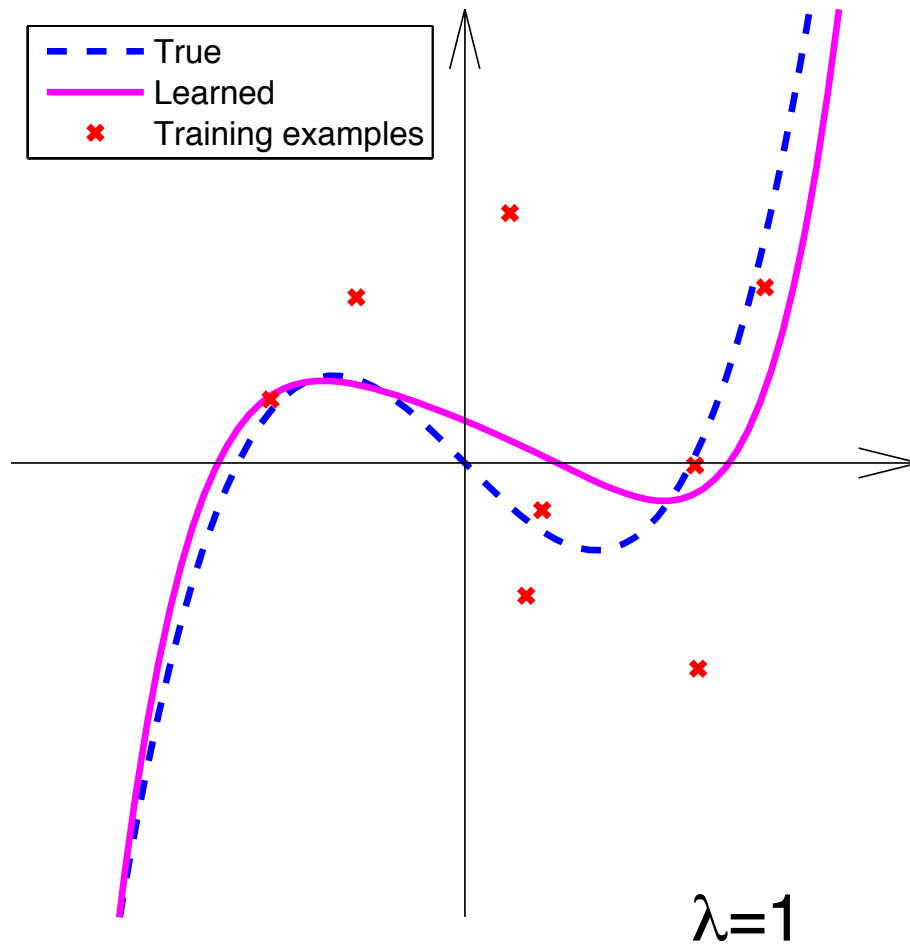
True

$$w^* = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ -1 \\ 0 \end{pmatrix}$$

Learned

$$w = \begin{pmatrix} 0.08 \\ 0.05 \\ 0.74 \\ -0.52 \\ -0.98 \\ 0.36 \end{pmatrix}$$

Example: 5th-order polynomial fitting



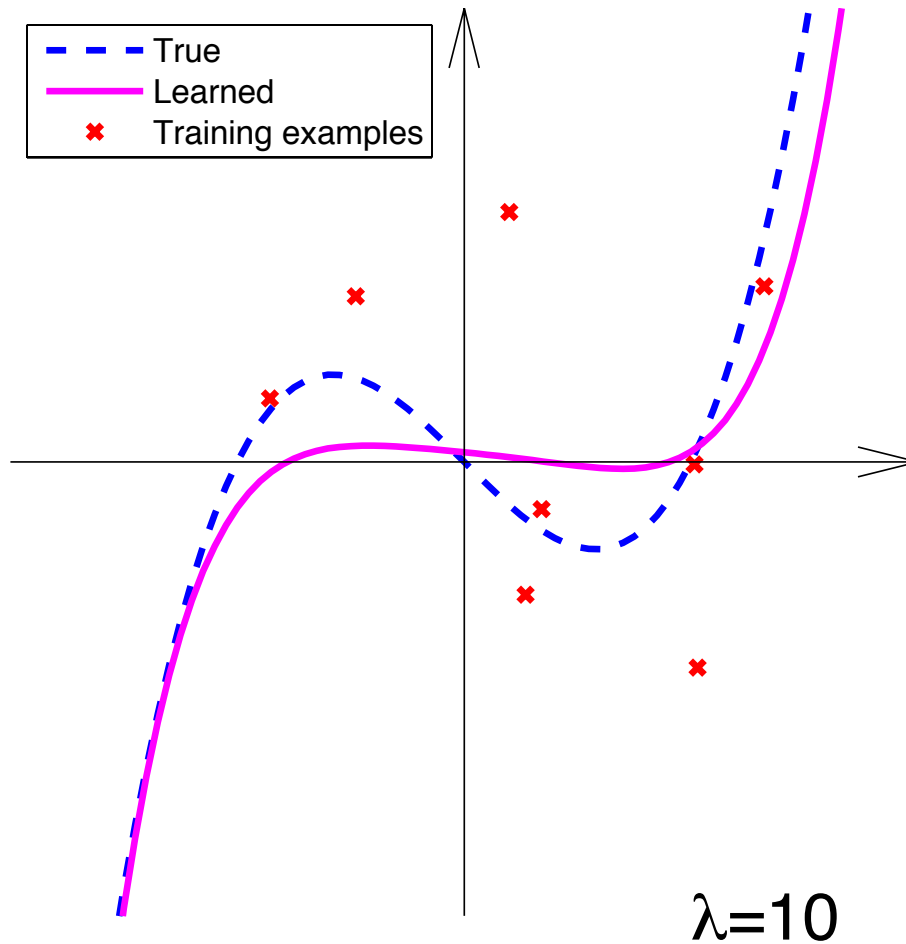
True

$$w^* = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ -1 \\ 0 \end{pmatrix}$$

Learned

$$w = \begin{pmatrix} 0.27 \\ -0.06 \\ -0.01 \\ -0.12 \\ -0.41 \\ 0.19 \end{pmatrix}$$

Example: 5th-order polynomial fitting



True

$$w^* = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ -1 \\ 0 \end{pmatrix}$$

Learned

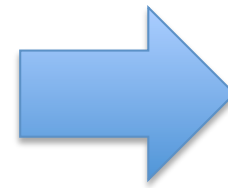
$$w = \begin{pmatrix} 0.22 \\ -0.07 \\ 0.01 \\ -0.05 \\ -0.10 \\ 0.04 \end{pmatrix}$$

Binary classification

- Target y is $+1$ or -1 .

Outputs
to be
predicted

$$\mathbf{y} = \begin{pmatrix} 1 \\ -1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$$



Orange ($+1$)
or lemon (-1)

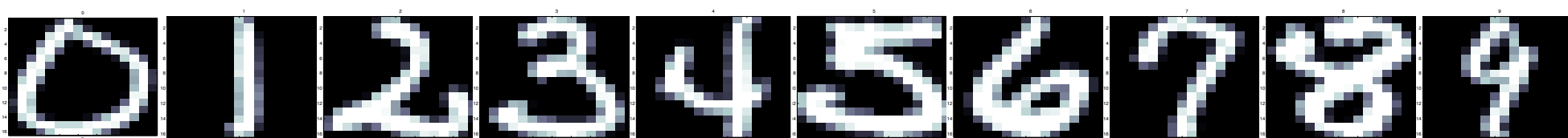
- Just apply ridge regression with $+1/-1$ targets
 - forget about the Gaussian noise assumption!

Multi-class classification

USPS digits dataset

7291 training samples,
2007 test samples

<http://www-stat-class.stanford.edu/~tibs/ElemStatLearn/datasets/zip.info>

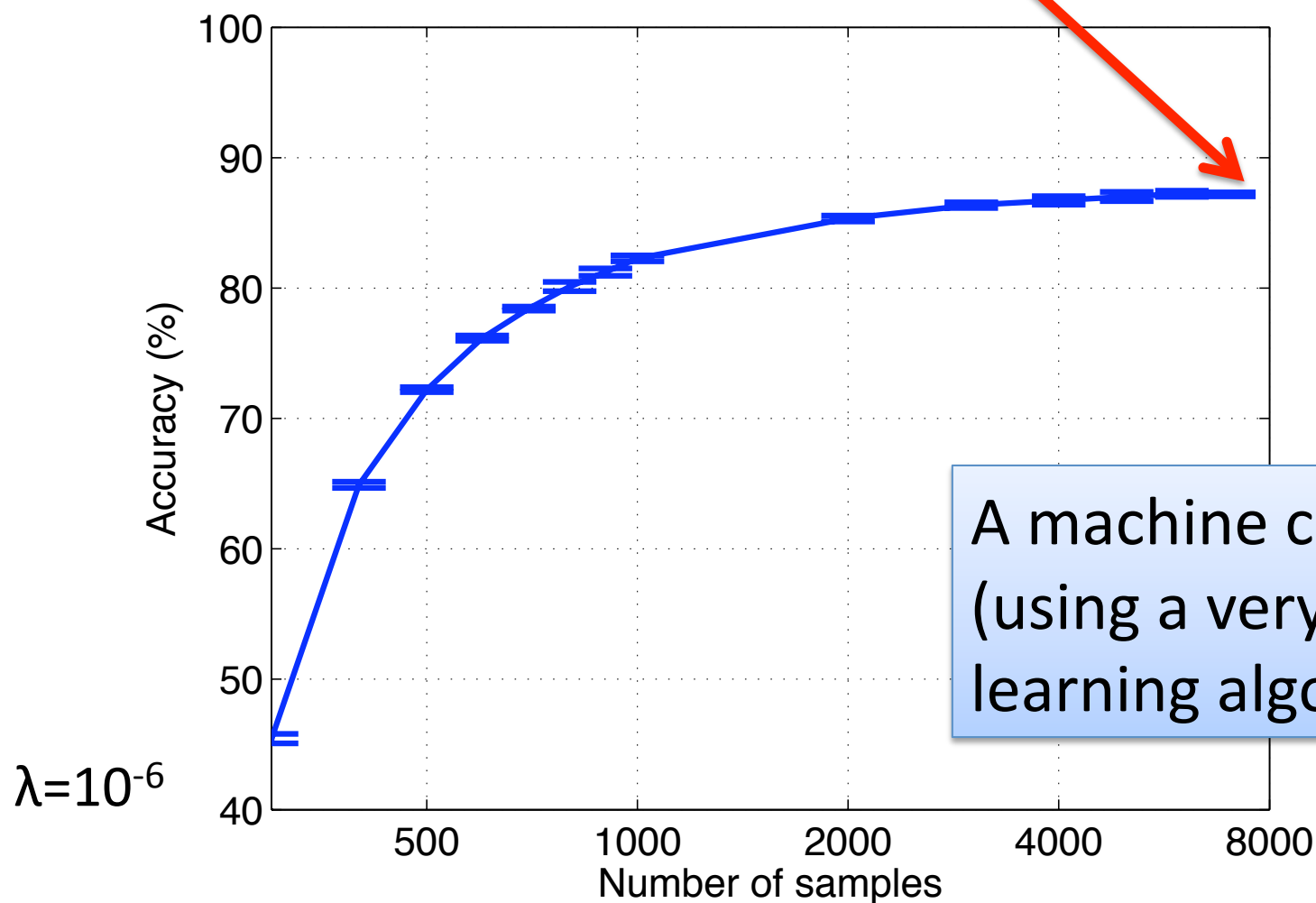


$$\mathbf{y} = \begin{pmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & & & & & \vdots \\ 0 & 1 & 0 & 0 & \dots & 0 \end{pmatrix}$$

Number of samples

USPS dataset

We can obtain 88% accuracy on a held-out test-set using about 7300 training examples



Summary (so far)

- Ridge regression (RR) is very simple.

- RR can be coded in one line:

```
W=(X' *X+lambda*eye(d)) \ (X' *Y);
```

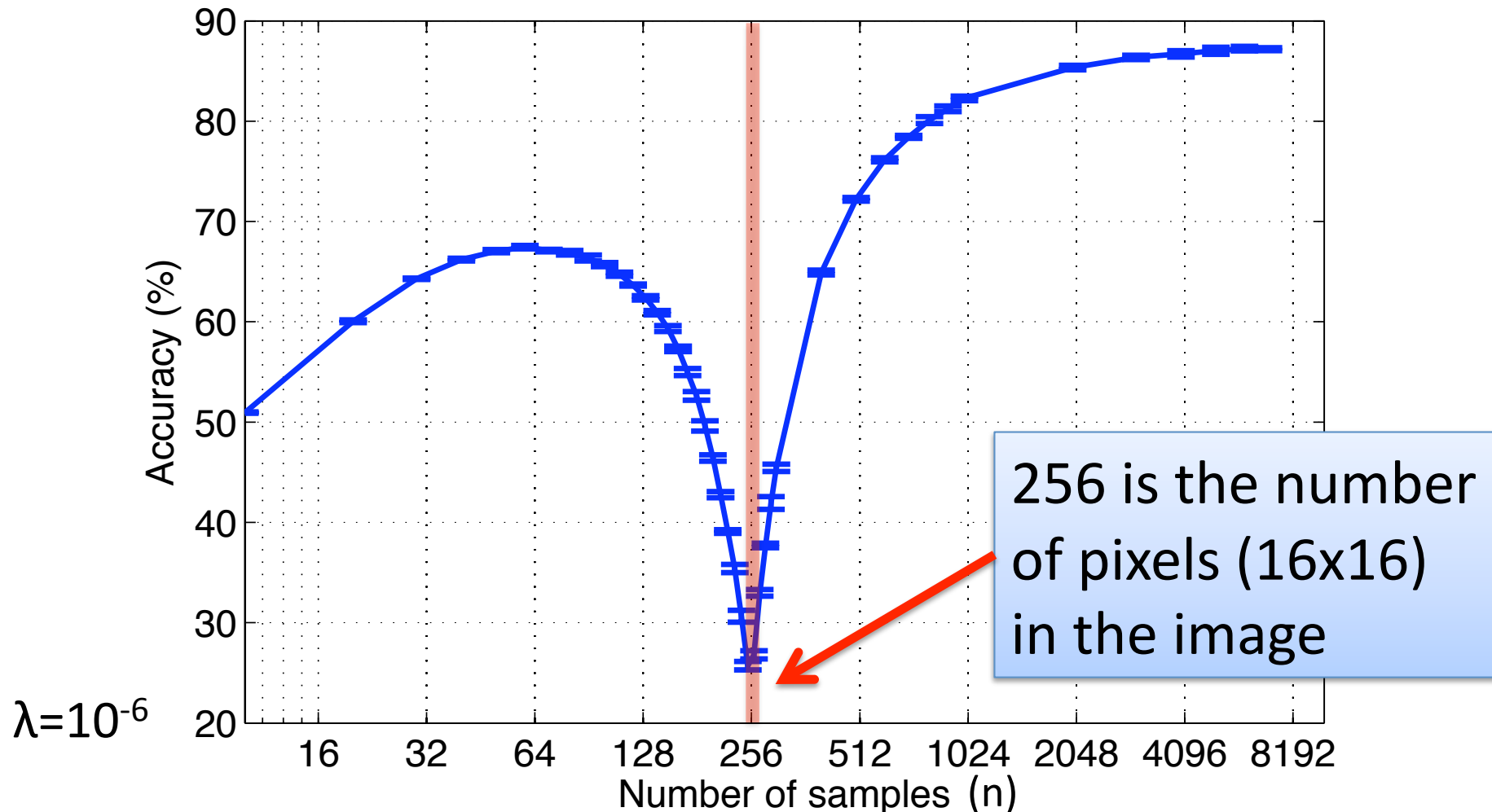
- RR can prevent over-fitting by regularization.
- Classification problem can also be solved by properly defining the output Y.
- Nonlinearities can be handled by using basis functions (polynomial, Gaussian RBF, etc.).

Singularity

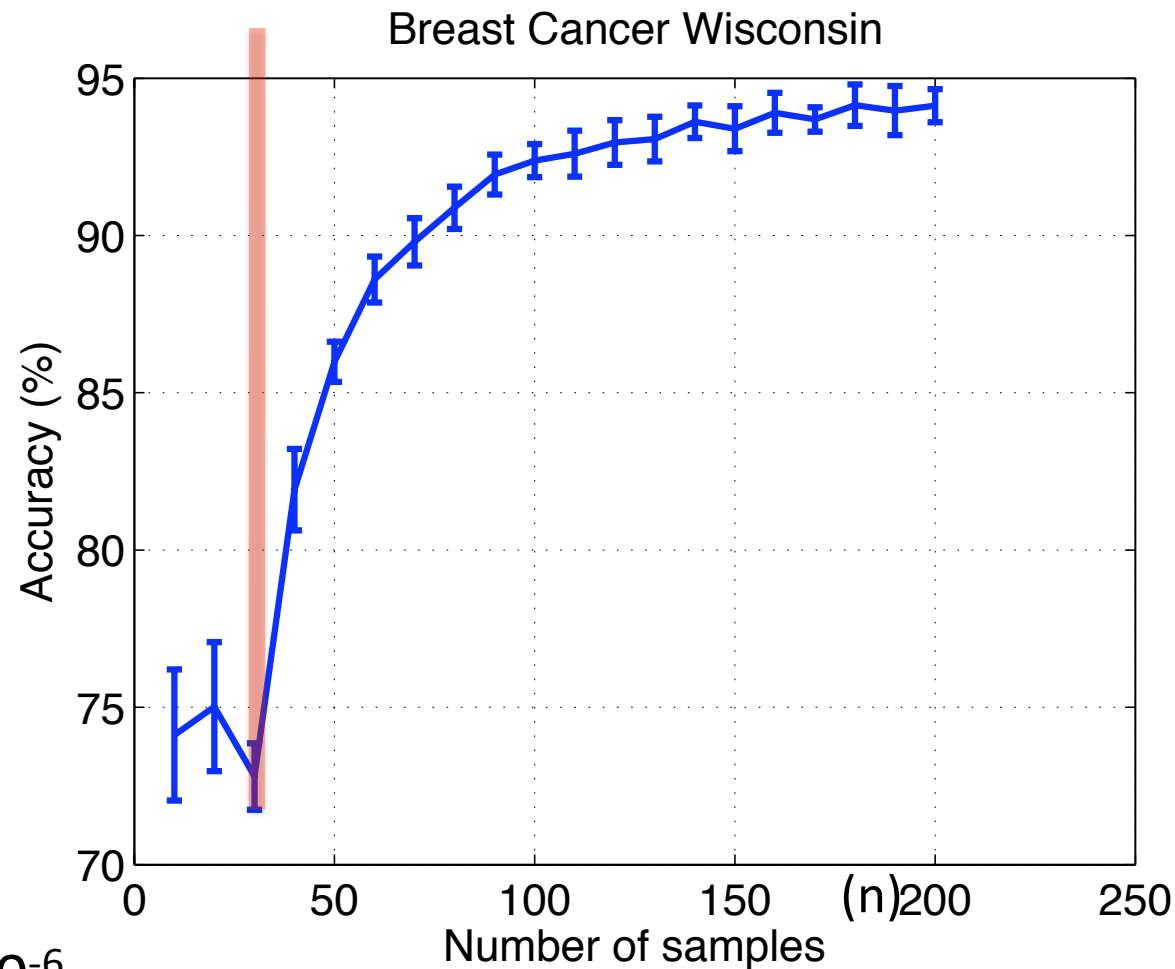
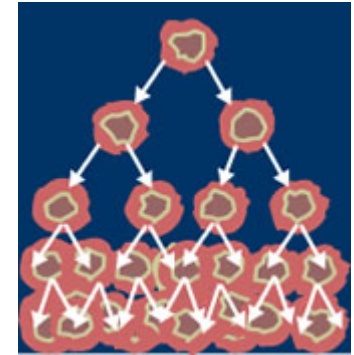
- The dark side of RR**

USPS dataset (d=256) (What I have been hiding)

- The more data the less accurate??



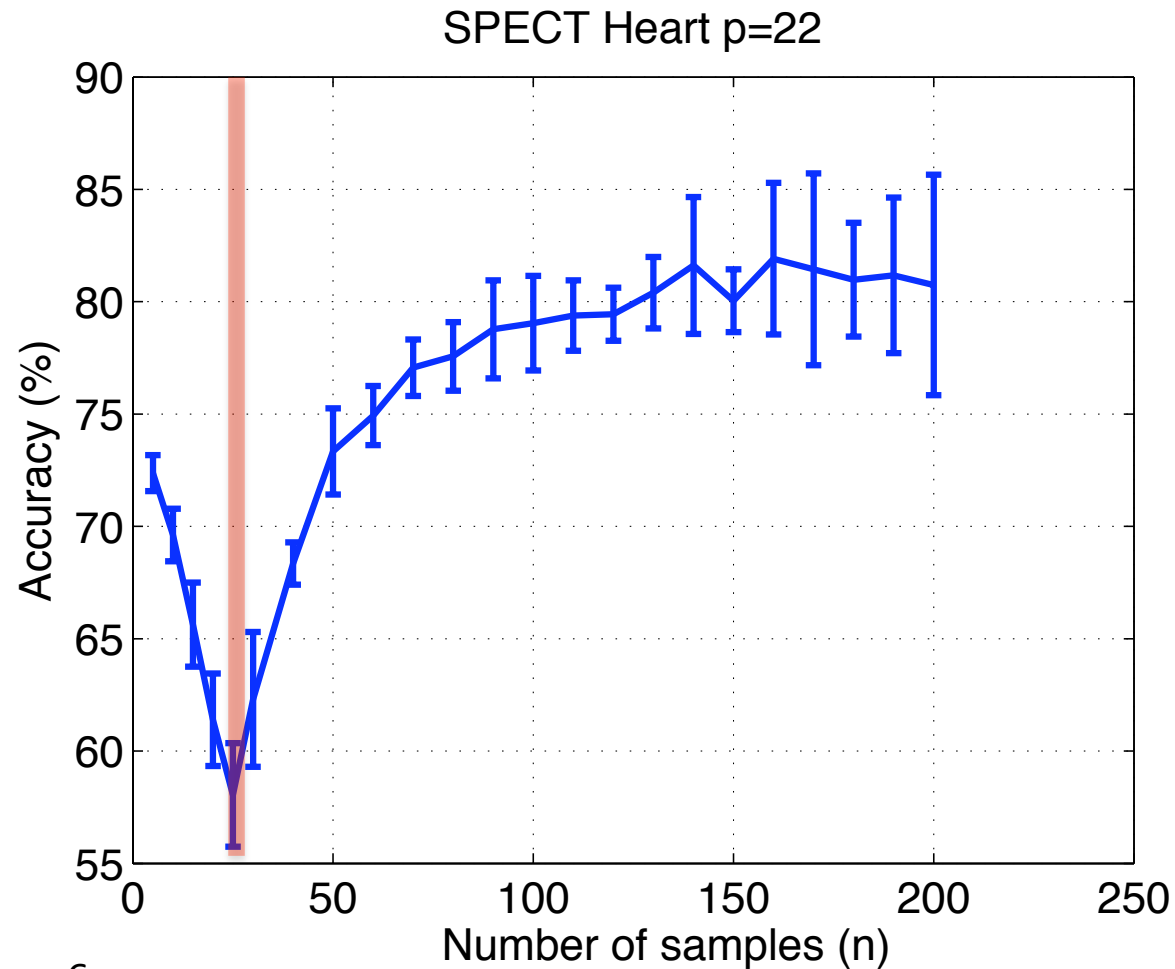
Breast Cancer Wisconsin (diagnostic) dataset (d=30)



30 real-valued features

- radius
- texture
- perimeter
- area, etc.

SPECT Heart dataset (d=22)

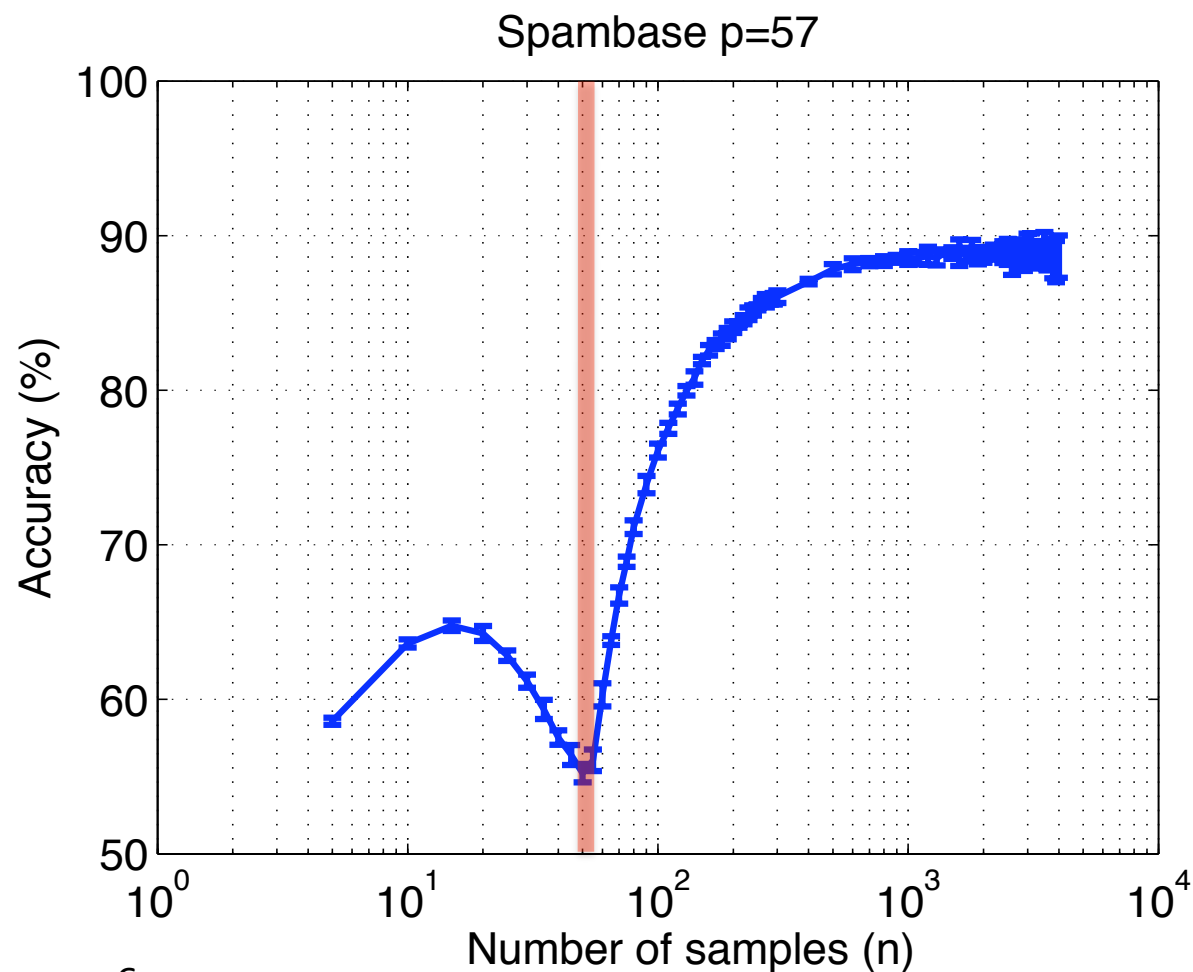


22 binary features

$\lambda=10^{-6}$

Spambase dataset (d=57)

Deleted Items	
From	Subject
CartoonPro...	Get the car of your dreams with CartoonProder help
Totallaptops...	How Old are You Really? - Take the RealAge Test
@ Don't be Lame...	Only way to make it grow[!]
Berryman's	what's up? it's me!
Wandprothes...	Special To TheGarden Member Offer
Accept Credit...	Process Credit Cards for Zero Up Front Cost
James	Your Pharmacy vs.
Quick Cash A...	Get A \$500 Cash Advance
Leviand Denny	breakfast everywhere.
eddy's land	Office OP - \$60
Comp Dept.	Get a complimentary Starbucks Gift Card on us
Guadalupe N...	Pay 10 Attention to the Man Behind the Curtain
Sunset Media	Get ready for Monday OCTOBER 10TH

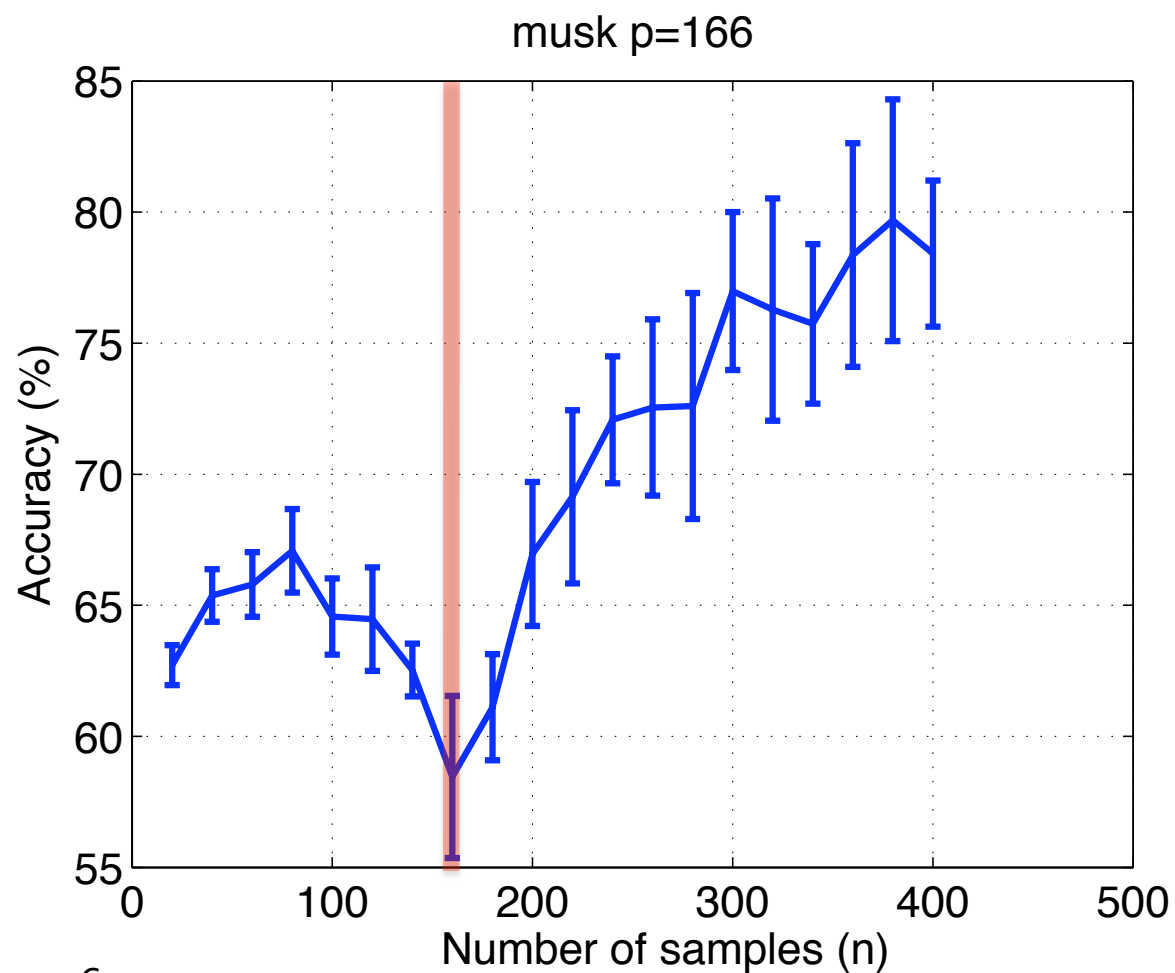
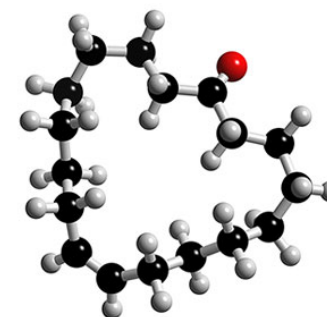


$\lambda=10^{-6}$

55 real-valued features

- word frequency
- character frequency
- 2 integer-valued feats
- run-length

Musk dataset (d=166)



166 real-valued features

$\lambda=10^{-6}$

Singularity

Why does it happen?

How can we avoid it?

Let's analyze the simplest case: regression.

- Model
 - Design matrix X is fixed (X is *not* random)
 - Output

$$\mathbf{y} = X\mathbf{w}^* + \boldsymbol{\xi} \quad \boldsymbol{\xi} : \text{noise}$$

- Estimator

$$\hat{\mathbf{w}} = (X^\top X + \lambda I_d)^{-1} X^\top \mathbf{y}$$

- Estimation Error

$$\text{Err}(\hat{\mathbf{w}}) = \mathbb{E}_{\boldsymbol{\xi}} \|\hat{\mathbf{w}} - \mathbf{w}^*\|^2 \quad \text{expectation over noise}$$

The estimator is a random variable!

Demo

- try `exp_ridgeregression_poly.m`

Estimator as a random variable

- Deriving the generalization error reduces to understanding how the estimator behaves as a random variable.
- Two strategies
 - Worst case
 - Average case – this is what we'll do today

Average case analysis

- Be careful!

$$\underbrace{\mathbb{E}_{\xi} \|\hat{w} - w^*\|^2}_{\text{Average case error}} \neq \underbrace{\|\mathbb{E}_{\xi} \hat{w} - w^*\|^2}_{\text{Error of the averaged estimator}}$$

Average case error
(what we will analyze)

Error of the
averaged estimator

- Which is smaller?

Bias-variance decomposition

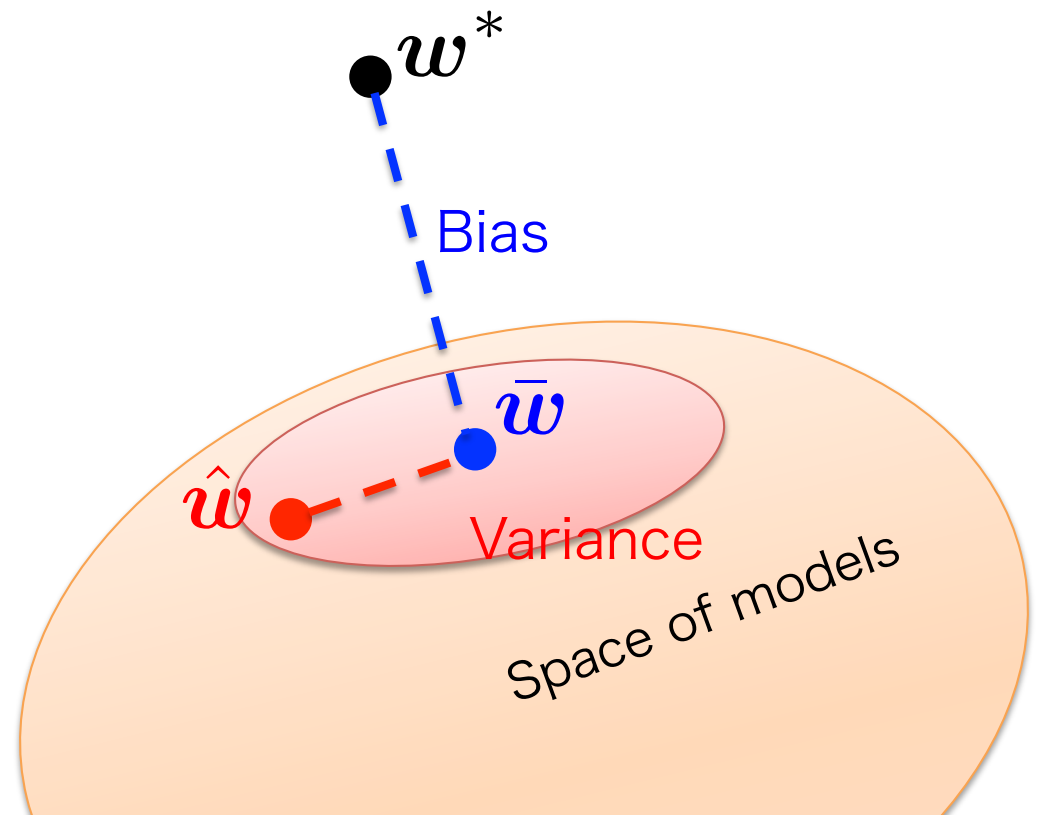
$$\mathbb{E}_{\xi} \|\hat{w} - w^*\|^2 = \underbrace{\mathbb{E}_{\xi} \|\hat{w} - \bar{w}\|^2}_{\text{Variance}} + \underbrace{\|\bar{w} - w^*\|^2}_{\text{Bias}^2}$$

where $\bar{w} = \mathbb{E}_{\xi} \hat{w}$

Bias: error coming from the model/design matrix

- under-fitting

Variance: error caused by the noise - over-fitting



Demo

- Try `exp_ridgeregression_poly.m` again
 - How can we reduce variance?
 - How can we reduce bias²?

For ridge regression,

- Since $\mathbf{y} = \mathbf{X}\mathbf{w}^* + \boldsymbol{\xi}$, if $\mathbb{E}\boldsymbol{\xi} = 0$, $\text{Cov}(\boldsymbol{\xi}) = \sigma^2 \mathbf{I}_n$

$$\mathbb{E}_{\boldsymbol{\xi}}[\hat{\mathbf{w}}] = \left(\hat{\boldsymbol{\Sigma}} + \lambda_n \mathbf{I}_d \right)^{-1} \hat{\boldsymbol{\Sigma}} \mathbf{w}^*$$

$$\text{Cov}(\hat{\mathbf{w}}) = \frac{\sigma^2}{n} \left(\hat{\boldsymbol{\Sigma}} + \lambda_n \mathbf{I}_d \right)^{-1} \hat{\boldsymbol{\Sigma}} \left(\hat{\boldsymbol{\Sigma}} + \lambda_n \mathbf{I}_d \right)^{-1}$$

$$\text{where } \lambda_n := \lambda/n \quad \text{and} \quad \hat{\boldsymbol{\Sigma}} = \frac{1}{n} \mathbf{X}^\top \mathbf{X}$$


Let's see if this is correct...

Exercise

- Analytical exercise:
 - Derive the expressions for both $\mathbb{E}_{\xi}[\hat{\mathbf{w}}]$ and $\text{Cov}(\hat{\mathbf{w}})$
 - Use them to derive bias² and variance.
- Empirical exercise: Plot the ellipse corresponding to the theoretically derived mean and covariance of the ridge regression estimator

- Key function:

`plotEllipse(mu, sigma, color, width, marker_size)`



mean
(2x1 column vec)

covariance
(2x2 matrix)

Bias² and variance from the mean $\mathbb{E}_{\xi}[\hat{\boldsymbol{w}}]$ and covariance $\text{Cov}(\hat{\boldsymbol{w}})$

- Bias²

$$\begin{aligned}\|\bar{\boldsymbol{w}} - \boldsymbol{w}^*\|^2 &= \|\mathbb{E}_{\xi} \hat{\boldsymbol{w}} - \boldsymbol{w}^*\|^2 \\ &= \lambda_n^2 \left\| \left(\hat{\boldsymbol{\Sigma}} + \lambda_n \boldsymbol{I}_d \right)^{-1} \boldsymbol{w}^* \right\|^2\end{aligned}$$

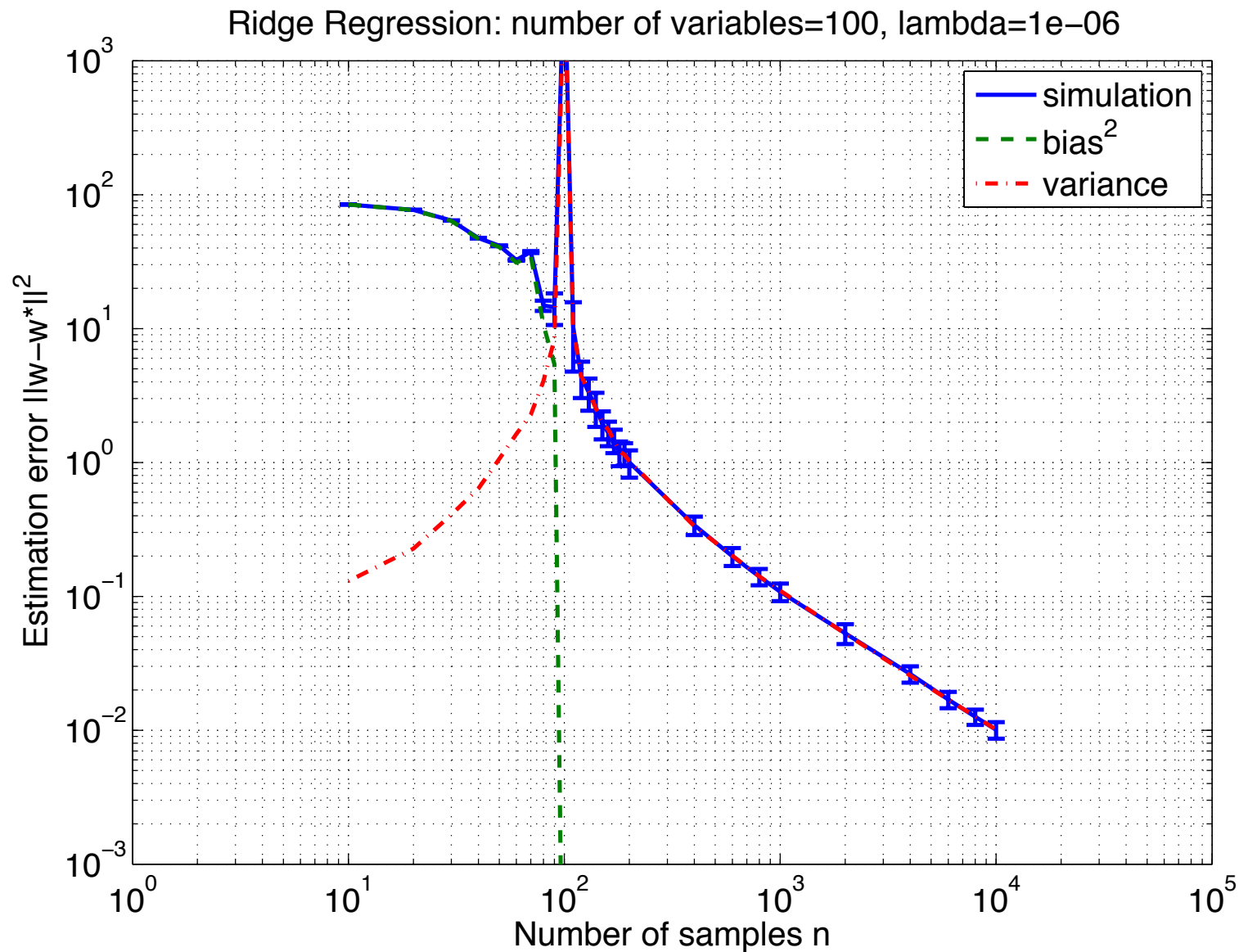
- Variance ($\lambda_n := \lambda/n$)

$$\mathbb{E}_{\xi} \|\hat{\boldsymbol{w}} - \bar{\boldsymbol{w}}\|^2 = \text{Tr}(\text{Cov}(\hat{\boldsymbol{w}}))$$

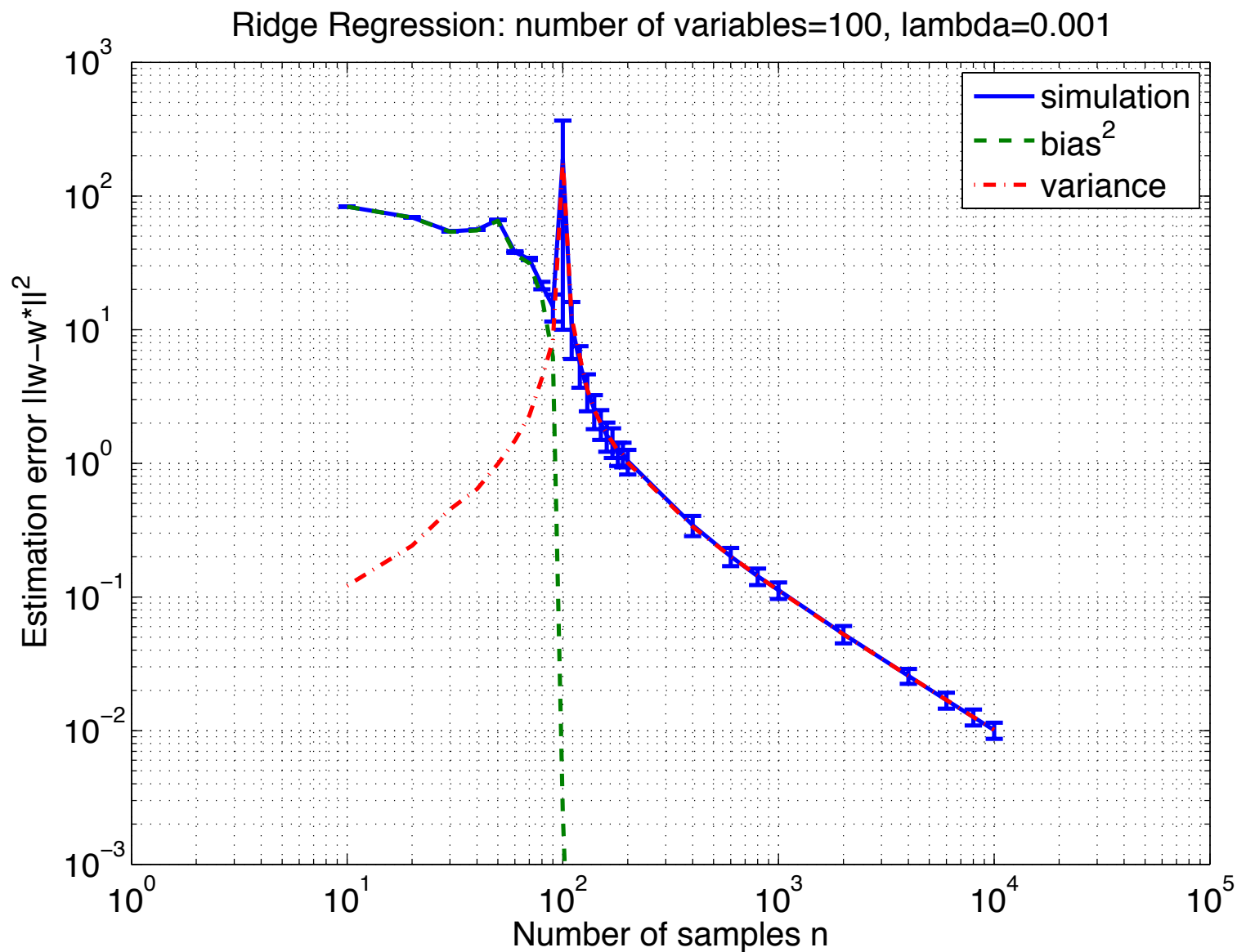
Explaining the singularity

- Bias² is an **increasing function** of λ and bounded by $\|w^*\|^2$
(cannot cause phase transition)
- Variance can be very large when the smallest eigenvalue of $\hat{\Sigma}$ is close to zero
(\Leftrightarrow smallest singular-value of X is close to zero)
- Try sample a random $d \times n$ matrix and see when the smallest singular-value is close to zero.

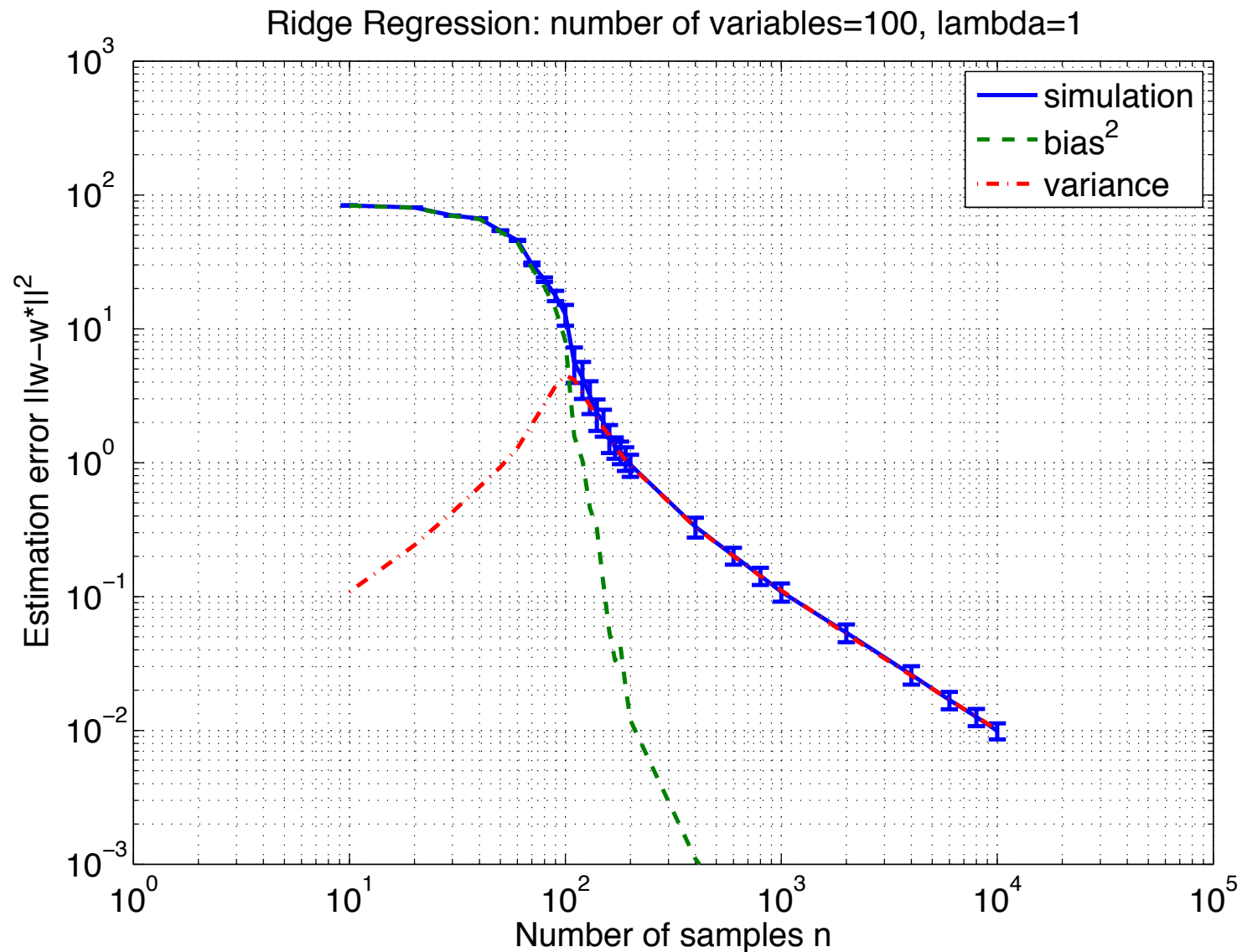
Simulation ($\lambda = 10^{-6}$)



Simulation ($\lambda=0.001$)



Simulation ($\lambda=1$)



Estimation error and generalization error

- So far, we've analyzed the estimation error

$$\mathbb{E}_{\xi} \|\hat{\mathbf{w}} - \mathbf{w}^*\|^2$$

- One might be more interested in analyzing the generalization error

$$\begin{aligned} \text{Gen}(\mathbf{x}) &= \mathbb{E}_{\xi} (\mathbf{x}^{\top} \mathbf{w}^* - \mathbf{x}^{\top} \hat{\mathbf{w}})^2 \\ &= \mathbb{E}_{\xi} \left\{ \mathbf{x}^{\top} (\mathbf{w}^* - \hat{\mathbf{w}}) \right\}^2 \end{aligned}$$

x: Test point

- Try `exp_frequentists_errorbar.m`

Exercise

- Analytical: derive the expression for the generalization error $\text{Gen}(x)$ at an arbitrary point x .

- Hint: use the decomposition

$$\boldsymbol{w}^* - \hat{\boldsymbol{w}} = (\boldsymbol{w}^* - \bar{\boldsymbol{w}}) + (\bar{\boldsymbol{w}} - \hat{\boldsymbol{w}})$$

- Empirical: try `exp_frequentists_errorbar.m` and see
 - when is the error under-estimated?
 - how does it compare to Bayesian posterior?

Generalization error at \mathbf{x}

$$\begin{aligned}\text{Gen}(\mathbf{x}) &= \mathbb{E}_{\boldsymbol{\xi}} \left\{ \mathbf{x}^\top (\mathbf{w}^* - \hat{\mathbf{w}}) \right\}^2 \\ &= \lambda_n^2 \left\{ \mathbf{x}^\top \hat{\boldsymbol{\Sigma}}_{\lambda_n}^{-1} \mathbf{w}^* \right\}^2 + \frac{\sigma^2}{n} \mathbf{x}^\top \hat{\boldsymbol{\Sigma}}_{\lambda_n}^{-1} \hat{\boldsymbol{\Sigma}} \hat{\boldsymbol{\Sigma}}_{\lambda_n}^{-1} \mathbf{x} \\ &\quad \left(\hat{\boldsymbol{\Sigma}}_{\lambda_n} := \hat{\boldsymbol{\Sigma}} + \lambda_n \mathbf{I}_d \right)\end{aligned}$$

- Caution

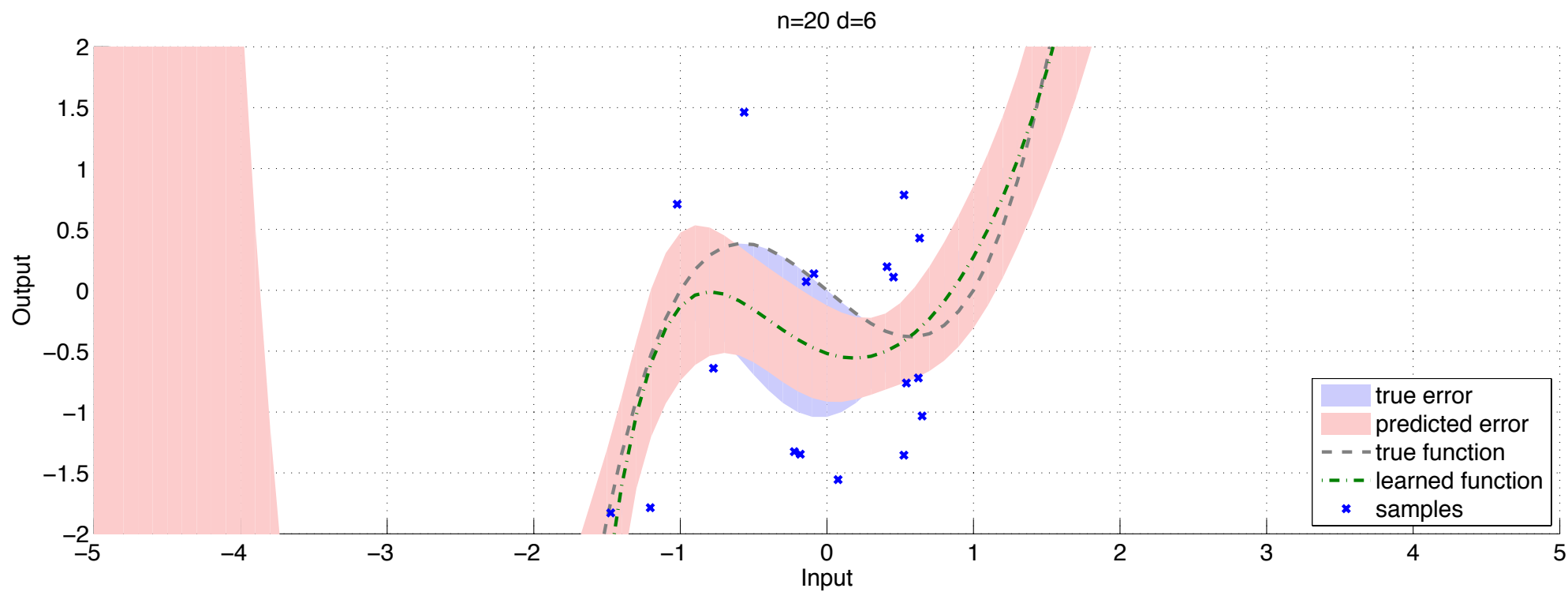
- \mathbf{w}^* is not known!

- worst case

- average case $\mathbb{E}_{\mathbf{w}^*} \left\{ \mathbf{x}^\top \hat{\boldsymbol{\Sigma}}_{\lambda_n}^{-1} \mathbf{w}^* \right\}^2 = \alpha^{-1} \|\hat{\boldsymbol{\Sigma}}_{\lambda_n}^{-1} \mathbf{x}\|^2$

assuming $\mathbb{E}_{\mathbf{w}^*} [\mathbf{w}^* \mathbf{w}^{*\top}] = \alpha^{-1} \mathbf{I}_d$

Frequentists' error-bar



How do we choose λ ?

- Bias² cannot be computed in practice (because we don't know w^*)
- Practical approaches
 - Mallow's C_L
 - Leave-one-out cross validation

Mallows' C_L [Mallows 1973]

- Tells us how the training error is related to bias²

$$\begin{aligned} & \frac{1}{n} \mathbb{E}_{\xi} \|\mathbf{y} - \mathbf{X} \hat{\mathbf{w}}\|^2 + \frac{2\sigma^2}{n} \text{Tr} \left(\hat{\Sigma} (\hat{\Sigma} + \lambda_n \mathbf{I}_d)^{-1} \right) \\ &= \sigma^2 + \underbrace{\mathbb{E}_{\xi} (\hat{\mathbf{w}} - \bar{\mathbf{w}})^{\top} \hat{\Sigma} (\hat{\mathbf{w}} - \bar{\mathbf{w}})}_{\text{Variance}} + \underbrace{(\bar{\mathbf{w}} - \mathbf{w}^*)^{\top} \hat{\Sigma} (\bar{\mathbf{w}} - \mathbf{w}^*)}_{\text{Bias}^2} \end{aligned}$$

$$(\lambda_n := \lambda/n)$$

$\text{Tr} \left(\hat{\Sigma} (\hat{\Sigma} + \lambda_n \mathbf{I}_d)^{-1} \right)$: known as the effective degrees of freedom

Schematically

$$\frac{1}{n} \mathbb{E}_{\xi'} \mathbb{E}_{\xi} \|y' - X \hat{w}\|^2$$

Expected
(fixed design)
generalization
error

$$\frac{2\sigma^2}{n} \text{Tr} \left(\hat{\Sigma} (\hat{\Sigma} + \lambda_n \mathbf{I}_d)^{-1} \right)$$

$$(y' = X w^* + \xi')$$

This is how much
we have overfitted!

$$\frac{1}{n} \mathbb{E}_{\xi} \|y - X \hat{w}\|^2$$

Expected
training error

Leave-one-out cross validation

- Idea: compute an estimator $\hat{\mathbf{w}}_{\setminus i}$ leaving sample (\mathbf{x}_i, y_i) out. Then test it on (\mathbf{x}_i, y_i) .
- It turns out that

$$\sum_{i=1}^n (y_i - \mathbf{x}_i^\top \hat{\mathbf{w}}_{\setminus i})^2 = \sum_{i=1}^n \left(\frac{y_i - \mathbf{x}_i^\top \hat{\mathbf{w}}}{1 - S(i, i)} \right)^2$$

where $\mathbf{S} = \mathbf{X}(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_d)^{-1} \mathbf{X}^\top$

- This can be obtained by solving just one ridge regression problem.

Discussion

- Mallow's C_L is a good approximation of the test error when $\hat{\Sigma} \simeq \Sigma$
 - but it requires the knowledge of σ^2
- Leave-one-out cross validation is an almost unbiased estimator of the generalization error
 - does not require the knowledge of σ^2
 - can be unstable (e.g., $S(i,i)$ close to one)
 - cannot be used for other number of folds (e.g., 10 folds)

Exercise

- Analytical exercise: Derive Mallow's C_L , or LOO-CV, or both.
- Empirical exercise:
 - Try and compare the two strategies on some dataset.
 - compare them to the *cheating strategy*, i.e., choose λ that minimizes the test error
 - also try them on a classification problem.

Further exercise

- Take any model or classifier (logistic regression, L1-regularization, kernel ridge regression, etc)
 - simulate a problem
 - visualize the scattering of the estimated coefficient vector
 - does it look Gaussian?
 - can you see a trade-off between bias and variance?

Summary

- Estimator is a random variable
 - it fluctuates depending on the training examples
 - characterizing the fluctuation is a key to understand its ability
- Training error is an under-estimate of the generalization error
 - systematically biased
 - understanding the bias is a key to derive a model selection criterion

What we did not discuss

- Other loss functions/regularization
 - analysis becomes significantly more challenging because the estimator is not analytically obtained
 - Solution 1: asymptotic second-order expansion. Cf. AIC
 - Solution 2: upper bounding using
$$\text{Objective}(\hat{\boldsymbol{w}}) \leq \text{Objective}(\boldsymbol{w}^*)$$
- Truth (\boldsymbol{w}^*) not contained in the model
 - VC dim, Rademacher complexity, etc. The bound becomes significantly looser.

Bayesian regression

Can we justify why we should
predict with uncertainty?

Bayesian linear regression

- Generative process

Coefficient vector $\boldsymbol{w} \sim \mathcal{N}(0, \alpha^{-1} \boldsymbol{I}_d)$

Noise vector $\boldsymbol{\xi} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I}_n)$

Observation $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w} + \boldsymbol{\xi}$

- Estimator

$$\boldsymbol{w}|\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{C})$$

$$\boldsymbol{\mu} := (\boldsymbol{X}^\top \boldsymbol{X} + \sigma^2 \alpha \boldsymbol{I}_d)^{-1} \boldsymbol{X}^\top \boldsymbol{y}$$

$$\boldsymbol{C} := \sigma^2 (\boldsymbol{X}^\top \boldsymbol{X} + \sigma^2 \alpha \boldsymbol{I}_d)^{-1}$$

Let's visualize it

- Try `exp_bayesian_regression.m`
- Does Bayesian regression get away with over-fitting?

Discussion



S. Kullback



DR. RICHARD A. LEIBLER
R. Leibler

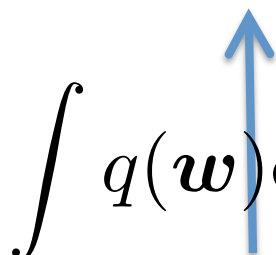
- From a frequentists' point of view, Bayesian posterior is a *distribution-valued estimator*.
- In fact,

$$p(\boldsymbol{w}|\boldsymbol{y}) = \operatorname{argmin}_{q(\boldsymbol{w})} \left\{ \mathbb{E}_{\boldsymbol{w} \sim q(\boldsymbol{w})} [-\log p(\boldsymbol{y}|\boldsymbol{w})] + D(q||p) \right\},$$



Bayesian posterior

subject to



Average log-likelihood

$$\int q(\boldsymbol{w}) d\boldsymbol{w} = 1.$$



Regularization

$p(\boldsymbol{w})$: prior distribution

Predictive distributions

- Bayesian predictive distribution

$$y_{n+1} | \mathbf{x}_{n+1}, \mathbf{y} \sim \mathcal{N}(\mathbf{x}_{n+1}^\top \boldsymbol{\mu}, \sigma^2 + \mathbf{x}^\top \mathbf{C} \mathbf{x})$$

- Plug-in predictive distribution (via RR)

$$y_{n+1} | \mathbf{x}_{n+1}, \mathbf{y} \sim \mathcal{N}(\mathbf{x}_{n+1}^\top \hat{\mathbf{w}}, \sigma^2)$$

Note: $\hat{\mathbf{w}} = \boldsymbol{\mu}$ if $\lambda = \alpha\sigma^2$

\Rightarrow They only differ in the predictive variance!

Evaluating the qualities of predictive distributions

- Kullback-Leibler divergence between the **true** and the **predictive** distributions

$$D(p_{\mathbf{w}^*}(y_{n+1}|\mathbf{x}_{n+1}) \parallel \hat{p}(y_{n+1}|\mathbf{x}_{n+1}))$$
$$= \underbrace{\frac{\{\mathbf{x}_{n+1}^\top (\mathbf{w}^* - \hat{\mathbf{w}})\}^2}{2\sigma_{\text{pred}}^2}}_{\text{Discounted generalization error}} + \underbrace{\frac{1}{2} \left\{ \frac{\sigma^2}{\sigma_{\text{pred}}^2} + \log \left(\frac{\sigma_{\text{pred}}^2}{\sigma^2} \right) - 1 \right\}}_{\text{Penalty for uncertainty}}$$

where

$$p_{\mathbf{w}^*}(y_{n+1}|\mathbf{x}_{n+1}) : y_{n+1}|\mathbf{x}_{n+1} \sim \mathcal{N}(\mathbf{x}_{n+1}^\top \mathbf{w}^*, \sigma^2)$$

$$\hat{p}(y_{n+1}|\mathbf{x}_{n+1}) : y_{n+1}|\mathbf{x}_{n+1} \sim \mathcal{N}(\mathbf{x}_{n+1}^\top \hat{\mathbf{w}}, \sigma_{\text{pred}}^2)$$

Exercise

1. Derive the expression for the KL divergence.
2. Show that the **penalty term** is nonnegative and increasing for $\sigma_{\text{pred}}^2 \geq \sigma^2$.
3. Derive the optimal σ_{pred}^2 that minimizes the KL divergence.

Optimal predictive variance

$$\sigma_{\text{pred}}^{*2} = \sigma^2 + \{\mathbf{x}_{n+1}^\top (\mathbf{w}^* - \hat{\mathbf{w}})\}^2$$

Noise variance + Frequentists' gen. error

Is Bayesian predictive variance optimal?

- In some sense, yes:

$$\mathbb{E}_{\mathbf{w}^* \sim \mathcal{N}(0, \alpha^{-1} \mathbf{I}_d)} \mathbb{E}_{\boldsymbol{\xi}} \left\{ \mathbf{x}_{n+1}^\top (\mathbf{w}^* - \hat{\mathbf{w}}) \right\}^2 = \mathbf{x}^\top \mathbf{C} \mathbf{x}$$

- this assumes that we know the correct noise variance σ^2 and the prior variance α^{-1}
- average over the draw of the true coefficient vector \mathbf{w}^*

Bayes risk [see Haussler & Oppen 1997]

- More generally, Bayesian predictive distribution is the minimizer of the Bayes risk

$$R[q_{\mathbf{y}}] = \mathbb{E}_{\mathbf{w} \sim p(\mathbf{w})} \mathbb{E}_{\mathbf{y} \sim \prod_{i=1}^n p(y_i | \mathbf{w})} [D(p(y_{n+1} | \mathbf{w}) || q_{\mathbf{y}}(y_{n+1}))]$$

Any distribution over y_{n+1} that depends on previous samples y_1, \dots, y_n

Assumes that the truth w comes from the prior, and the samples are drawn from the likelihood $p(y|w)$!

Discussion

- Bayesian predictive distribution minimizes the Bayes risk given the correct prior and correct likelihood.
 - Clearly not satisfying.
- Can we make it independent of the choice of prior/likelihood?
 - PAC Bayes theory

Preliminaries

- Loss function $L(s, \mathbf{w})$
 - assumed to be bounded by L_{\max}
 - e.g., classification error
$$L(s, \mathbf{w}) = \begin{cases} 0 & \text{if } y\mathbf{x}^\top \mathbf{w} \geq 0, \\ 1 & \text{otherwise} \end{cases}$$
- Training Gibbs error $(s = (y, \mathbf{x}), L_{\max} = 1)$

$$\hat{L}(Q) = \frac{1}{n} \sum_{i=1}^n \mathbb{E}_{\mathbf{w} \sim Q(\mathbf{w})} [L(s_i, \mathbf{w})]$$

- Gibbs error (for some “posterior” Q over \mathbf{w})

$$L(Q) = \mathbb{E}_s \mathbb{E}_{\mathbf{w} \sim Q(\mathbf{w})} [L(s, \mathbf{w})]$$

- this is the quantity that we care about

PAC-Bayes training-variance bound

[McAllester 1999, 2013; Catoni 2007]

- Let $\lambda > 1/2$, “prior” $P(w)$ is fixed before seeing the data, “posterior” $Q(w)$ can be any distribution that depends on the data. Then we have

$$\mathbb{E}_S L(Q) \leq \frac{1}{1 - \frac{1}{2\lambda}} \left(\mathbb{E}_S \hat{L}(Q) + \frac{\lambda L_{\max}}{n} \mathbb{E}_S D(Q \| P) \right)$$



Expectation with respect to training examples
(average case)

Note: the worst case version is more commonly presented as PAC Bayes

Discussion

- What is Gibbs error?
 - Error of a prediction made randomly according to the posterior
 - Bayes generalization error \leq Gibbs generalization error
- What is the role of λ ?
 - more or less an artifact in the analysis
 - can be fixed at a large but fixed constant (say $\lambda=10$)
- What is the best prior $P(w)$?
 - $P(w) = E_S[Q(w)]$ minimizes $E_S D(Q(w)|P(w))$
 - $E_S D(Q(w)|E_S[Q(w)])$: measure of variance of the posterior $Q(w)$

Summary

- Bayesian methods are not exempt from overfitting.
- Posterior- and predictive distribution are *random distributions*
- Does it make sense to predict with posterior variance?
 - Only if you measure the quality of the predictive distribution with the KL (or other) divergence.
- PAC-Bayes training-variance bound reflects the variance of the posterior distribution.

Beyond this lecture

- Non-parametric analysis of GP
 - van der Vaart & van Zanten (2011)
“Information Rates of Nonparametric Gaussian Process Methods”

$$\mathbb{E}_S \|\hat{f} - f^*\|_n^2 \leq O \left(n^{-\min(\alpha, \beta)/(2\alpha + d)} \right)$$

for f^* with smoothness parameter β
and posterior mean \hat{f} using Matérn
kernel with parameter α .