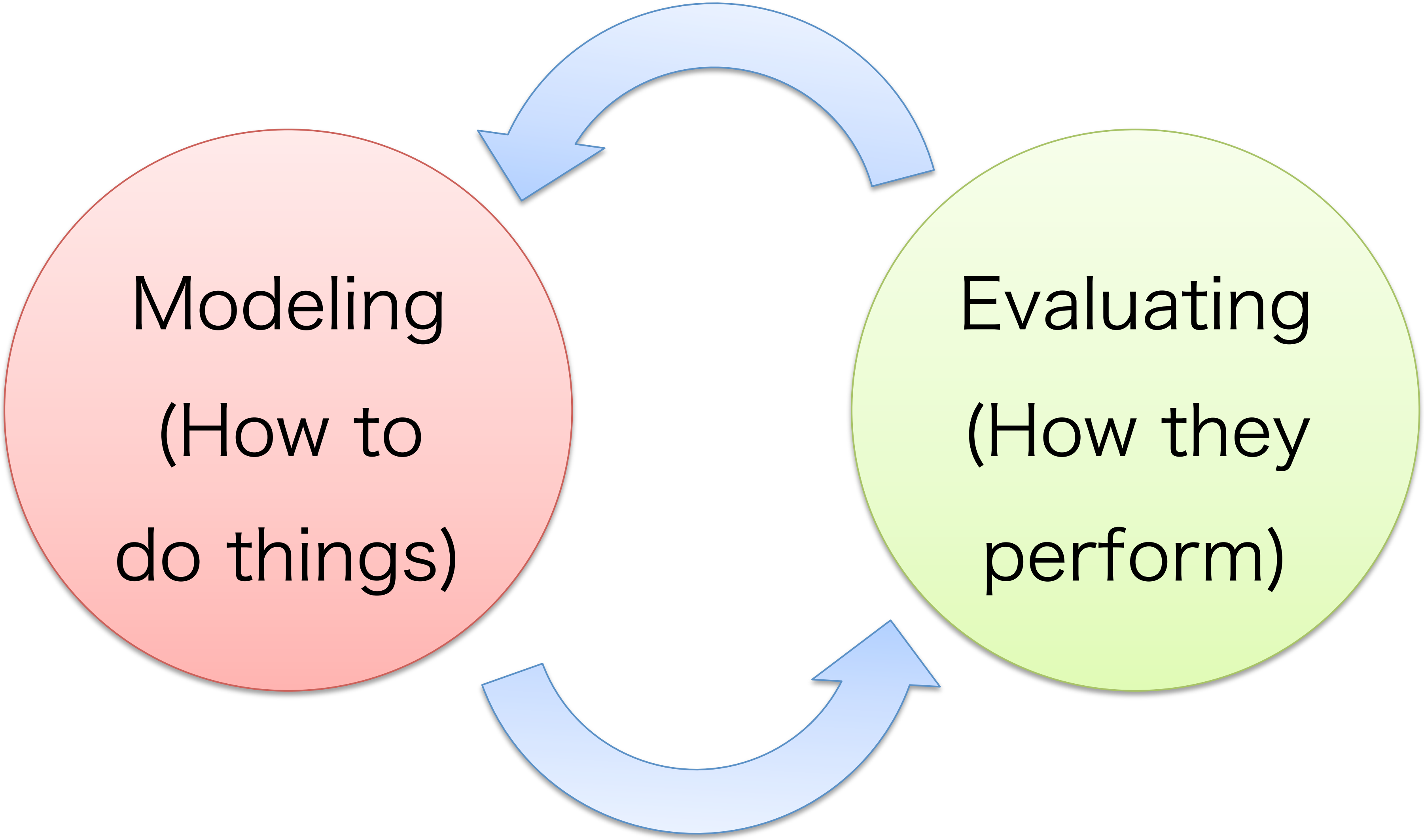
Introduction to the analysis of learning algorithms: ridge regression and lasso

Ryota Tomioka
tomioka@mist.i.u-tokyo.ac.jp
(Univ of Tokyo ➡ TTI Chicago)

Two sides of machine learning



Theory: Why is it hard?

- Mostly because we try to learn too many things at the same time
  - Equality

    $X = Y$ ··· the easiest
  - Inequality

    $X \leq Y$ ··· doable
  - Probabilistic inequality

    $X \leq Y$ with probability p ··· the hardest

In this lecture, I will make separation between them.

The first part: ridge regression

- Can analyze everything using only *equalities (=)*
- Can be considered as a starting point for other (more complex) algorithms
- Curious phase transition phenomena can be observed

The second part: LASSO

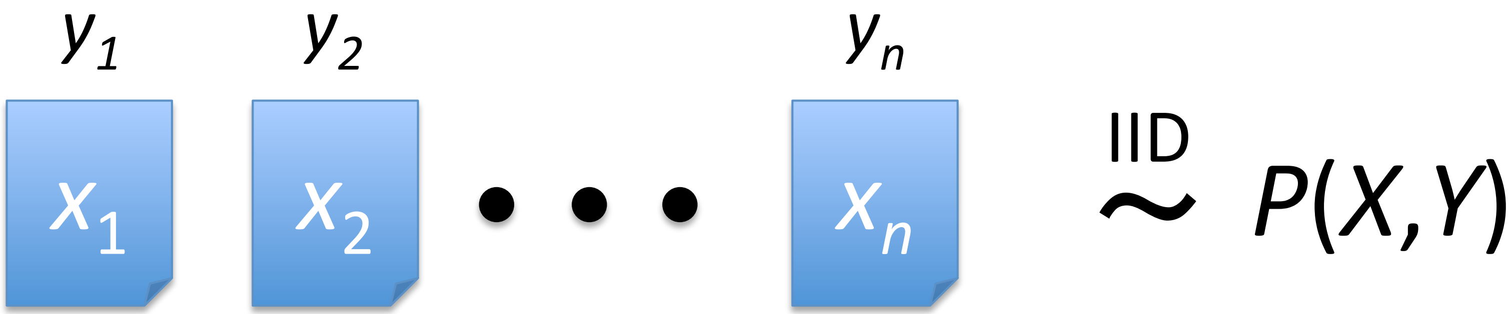
- L1 regularized learning is a convenient way of obtaining sparsity.

- Not only convenient:
  - in many settings O($k$log(p)) samples are enough to learn when the truth is a k-sparse vector in p dimension.
  - enables learning in very high dimension

Ridge Regression

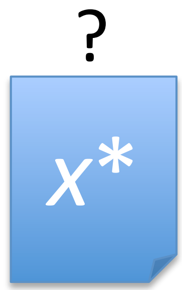
Problem Setting

- Training examples: (x_i, y_i) $(i=1,\cdots, n)$, $x_i \in \mathrm{R}^p$



- Goal
  - Learn a linear function

$$f(x^*) = w^\top x^* \quad (w \in \mathrm{R}^p)$$



that predicts the output y^* for a test point

$$(x^*, y^*) \sim \mathrm{P(X,Y)}$$

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

Ridge Regression

- Solve the minimization problem

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

Training error

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

Target output

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

Design matrix

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

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

$$\boldsymbol{X} = \begin{pmatrix} \text{Size} & \text{\#rooms} & \text{Bathroom} & \text{Sunlight} & \text{Neighborhood} & \text{Train st.} \end{pmatrix}$$

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

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

Solving ridge regression

- Take the gradient, and solve

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

which gives

$$\boldsymbol{w} = \left(\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I}_p\right)^{-1} \boldsymbol{X}^\top \boldsymbol{y}$$

(I_p: p×p identity matrix)

The solution can also be written as (exercise)

$$\boldsymbol{w} = \boldsymbol{X}^\top \left(\boldsymbol{X}\boldsymbol{X}^\top + \lambda \boldsymbol{I}_n\right)^{-1} \boldsymbol{y}$$

Example: polynomial fitting

- Degree (p-1) polynomial model

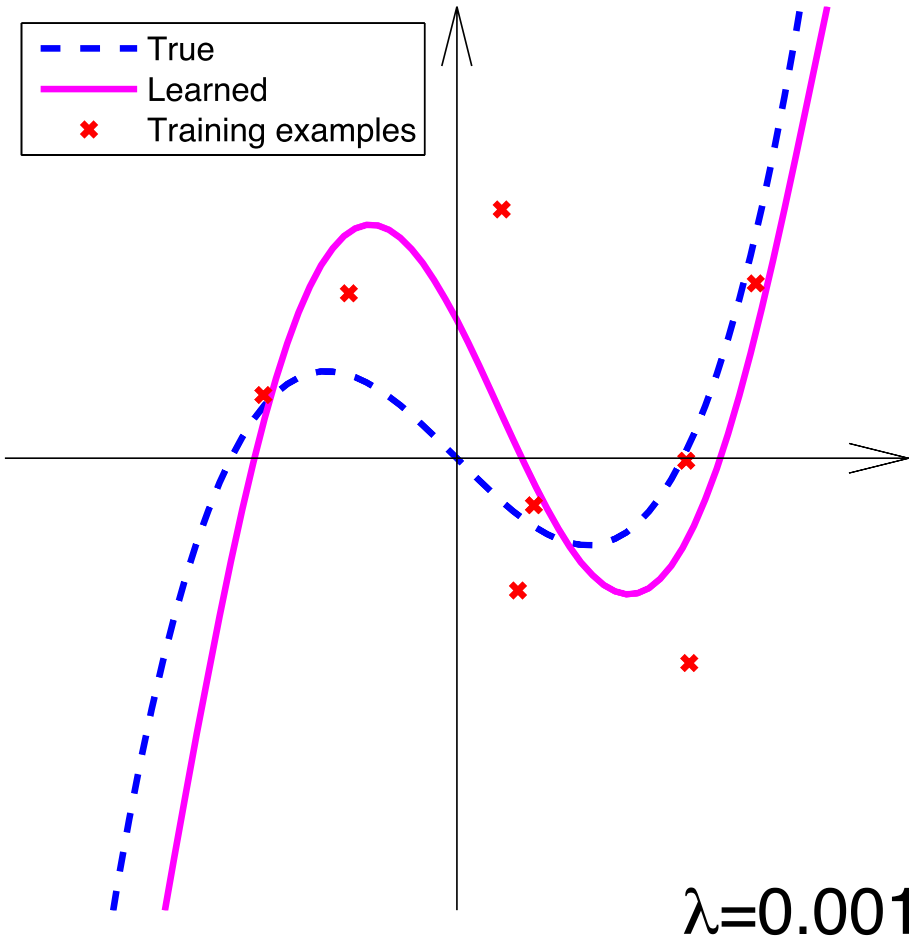
$$y = w_1 x^{p-1} + \cdots + w_{p-1} x + w_p + \text{noise}$$

$$= \begin{pmatrix} x^{p-1} & \cdots & x & 1 \end{pmatrix} \begin{pmatrix} w_1 \\ \vdots \\ w_{p-1} \\ w_p \end{pmatrix} + \text{noise}$$

Design matrix:

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

Example: 5th-order polynomial fitting



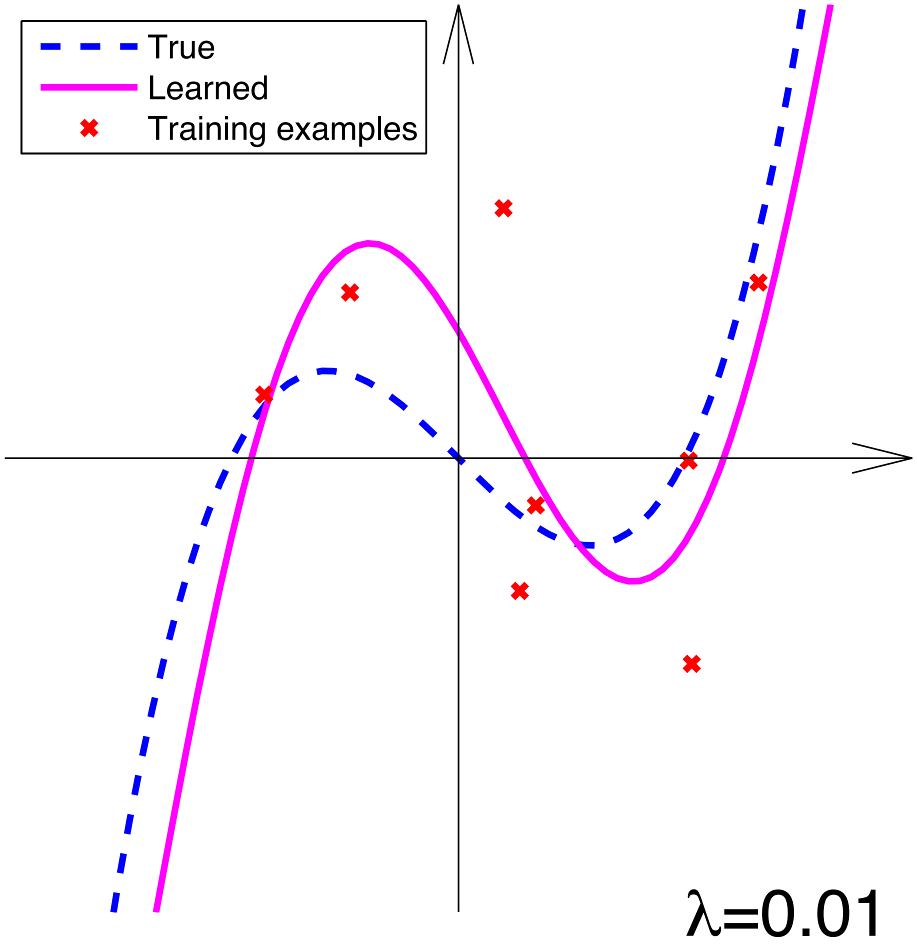
True

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

Learned

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

Example: 5th-order polynomial fitting



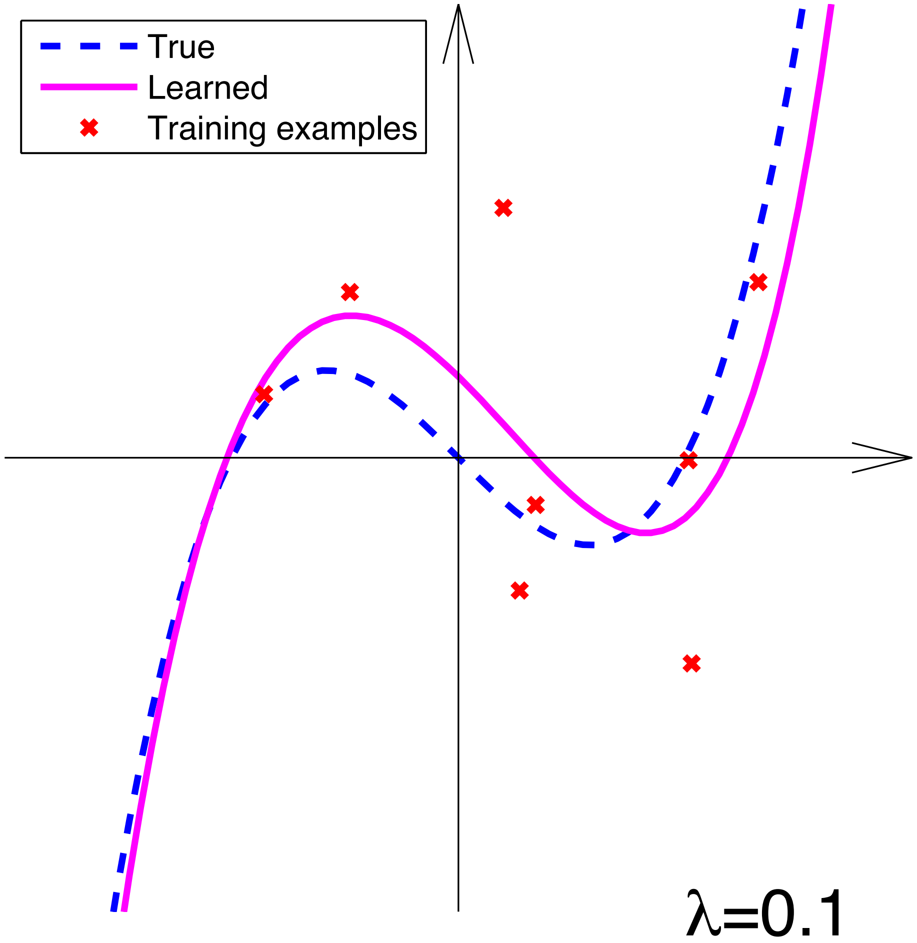
True

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

Learned

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

Example: 5th-order polynomial fitting



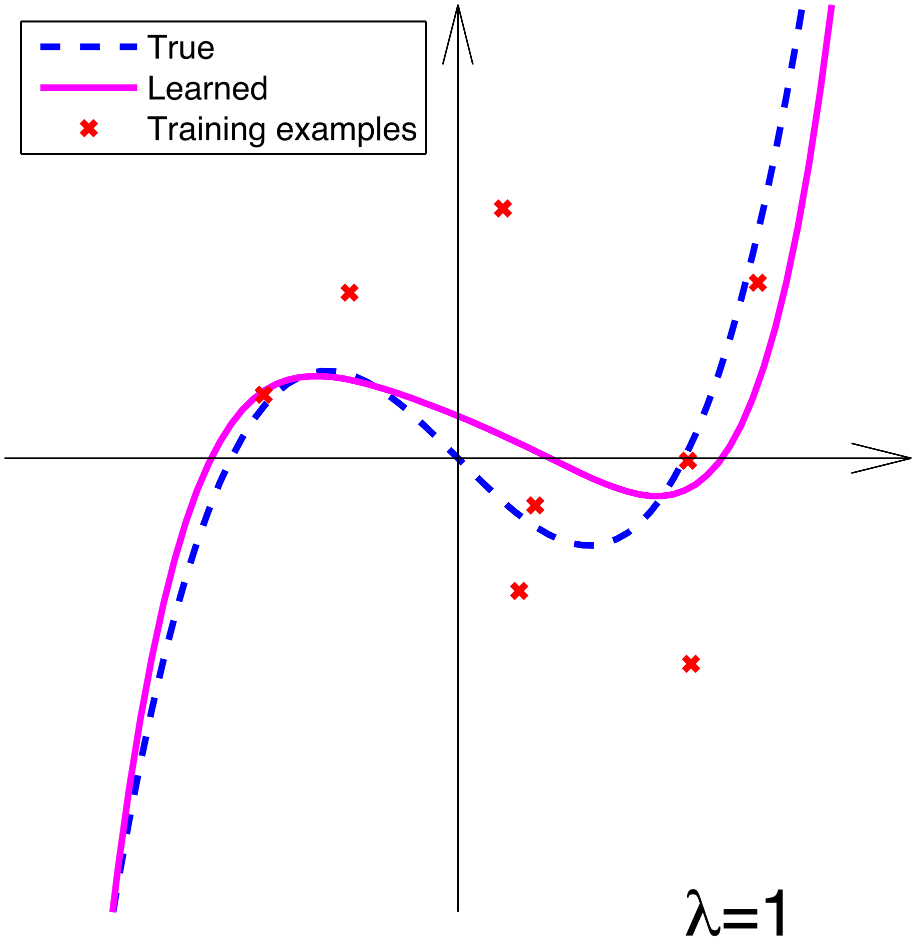
True

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

Learned

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

Example: 5th-order polynomial fitting



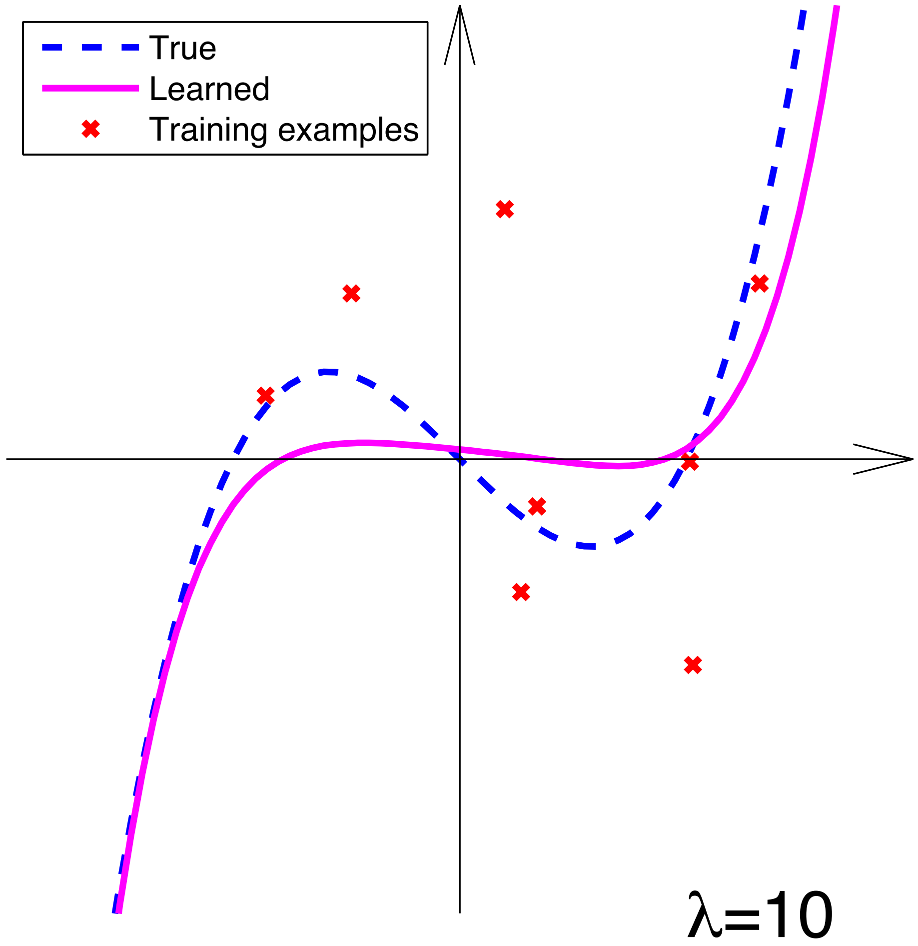
True

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

Learned

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

Example: 5th-order polynomial fitting



True

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

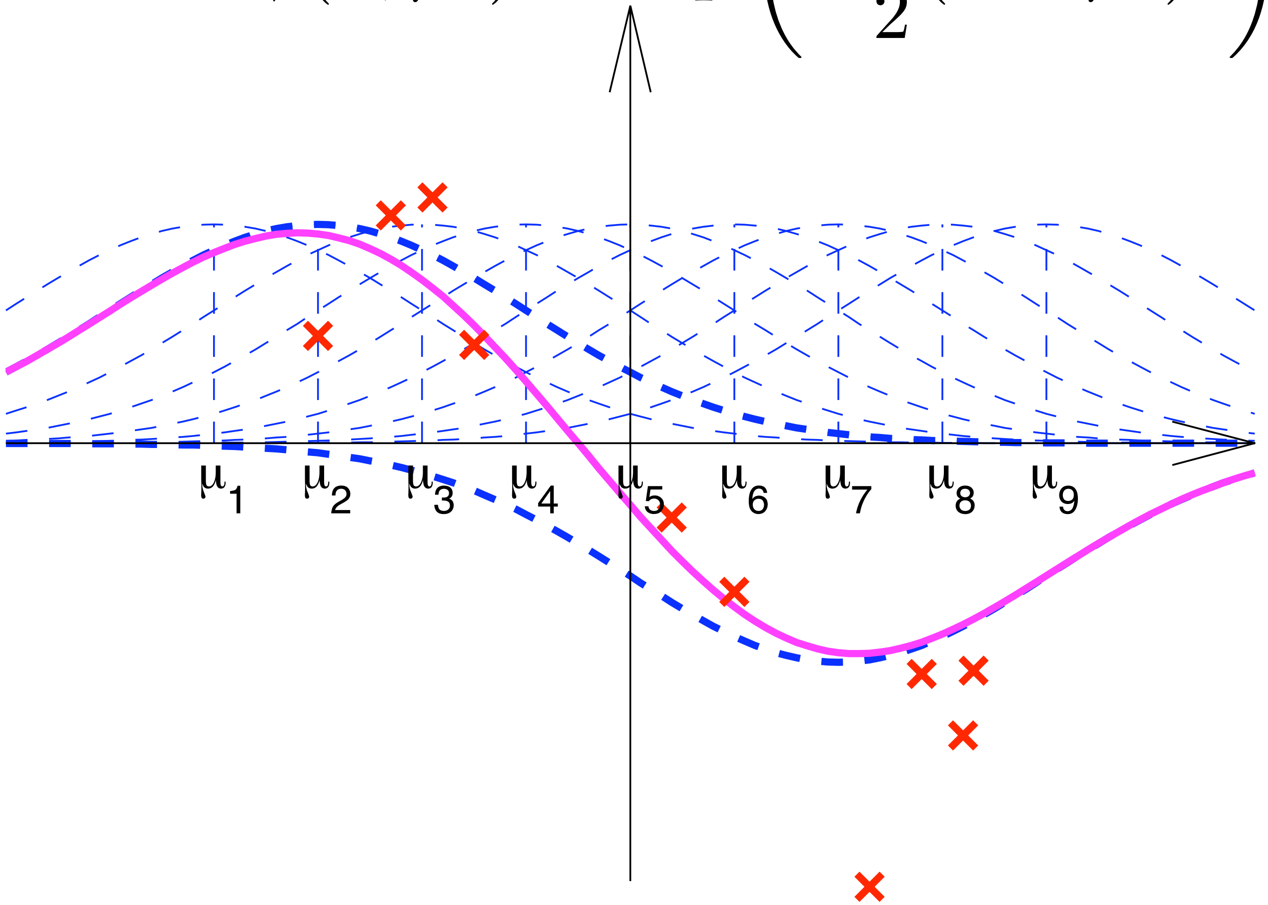
Learned

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

Example: RBF fitting

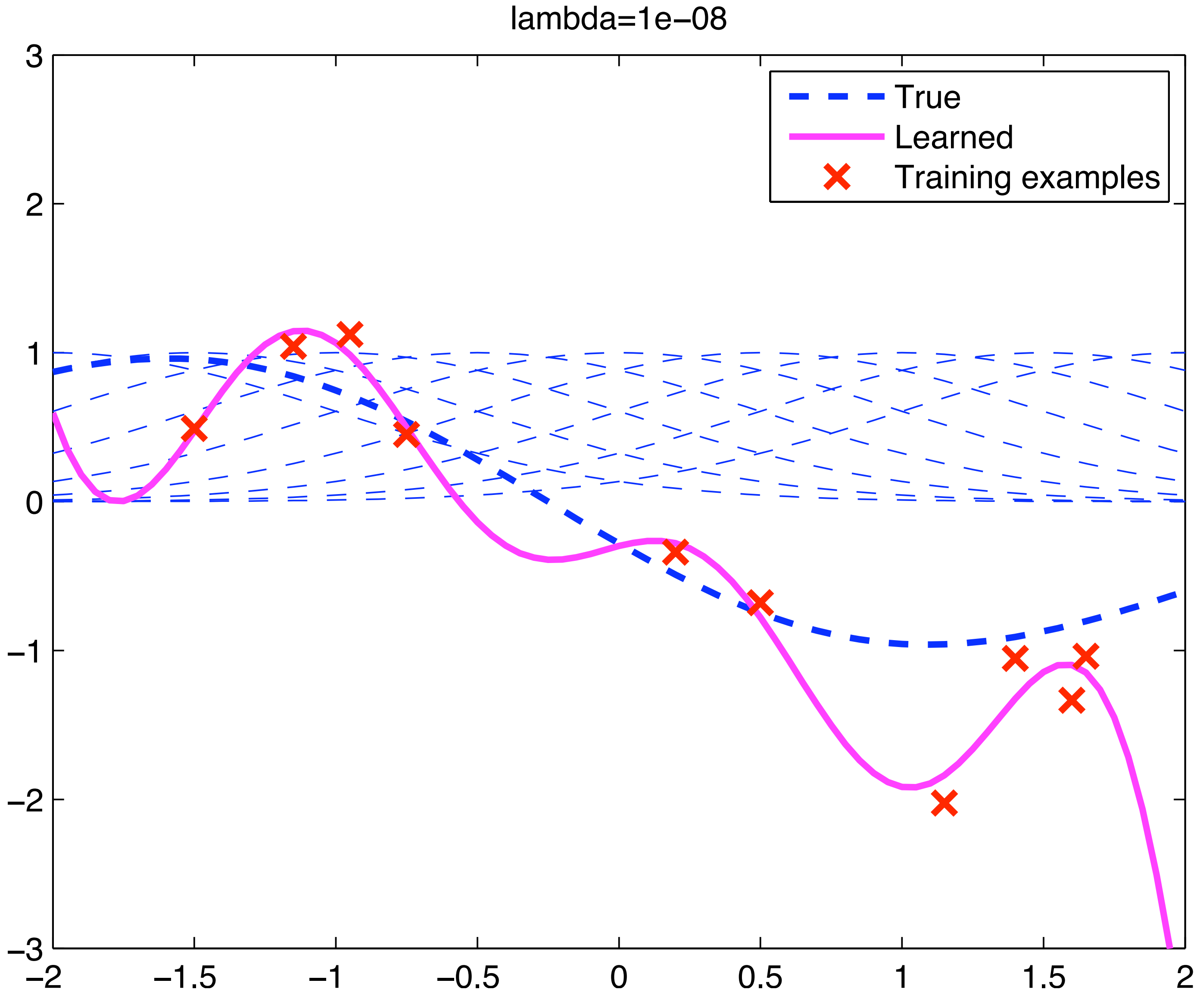
- Gaussian radial basis function (Gaussian-RBF)

$$\phi(x;\mu_c) = \exp\left(-\frac{1}{2}(x-\mu_c)^2\right)$$

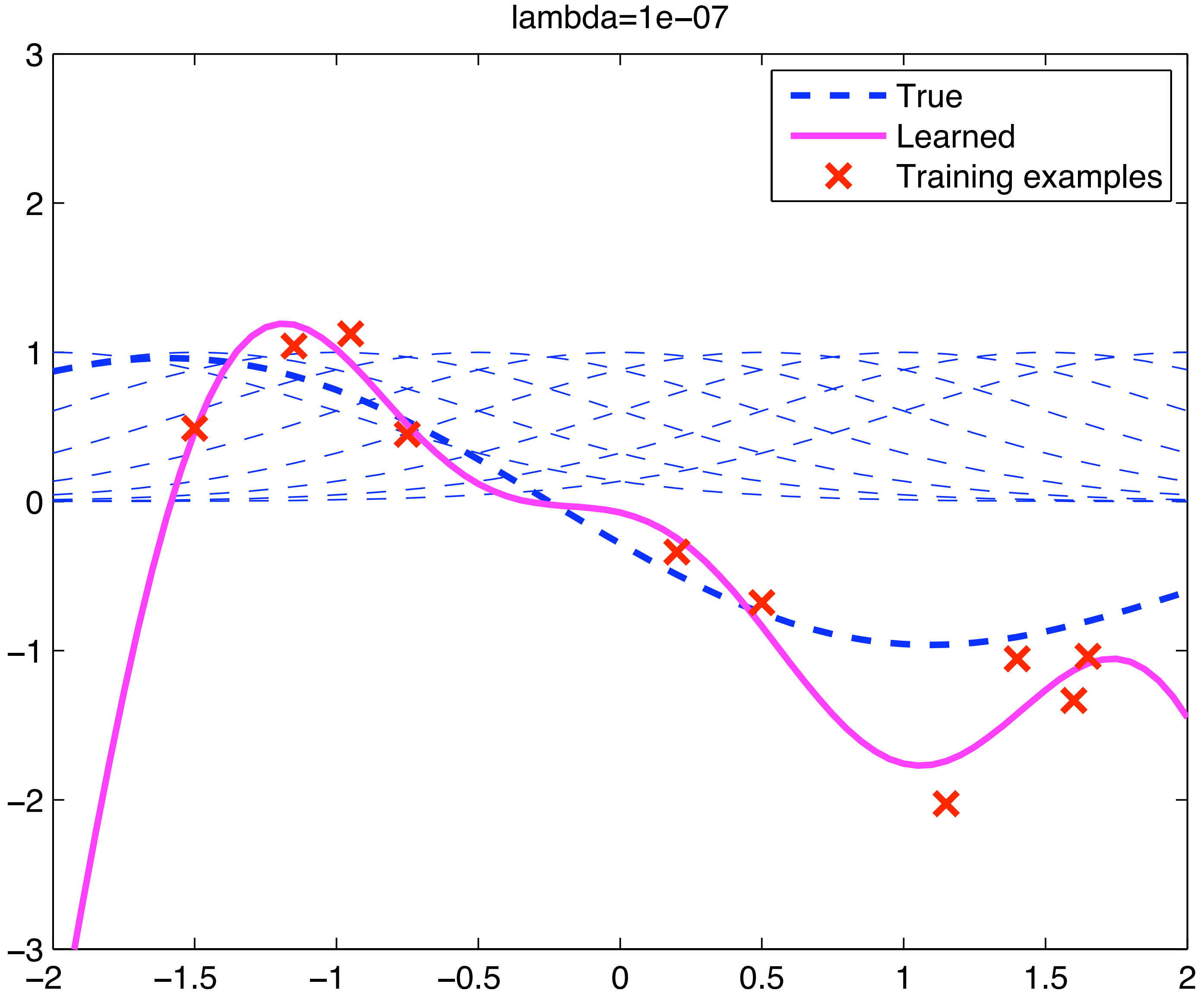


$$\boldsymbol{w}^* = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ 0 \\ 0 \end{pmatrix}$$

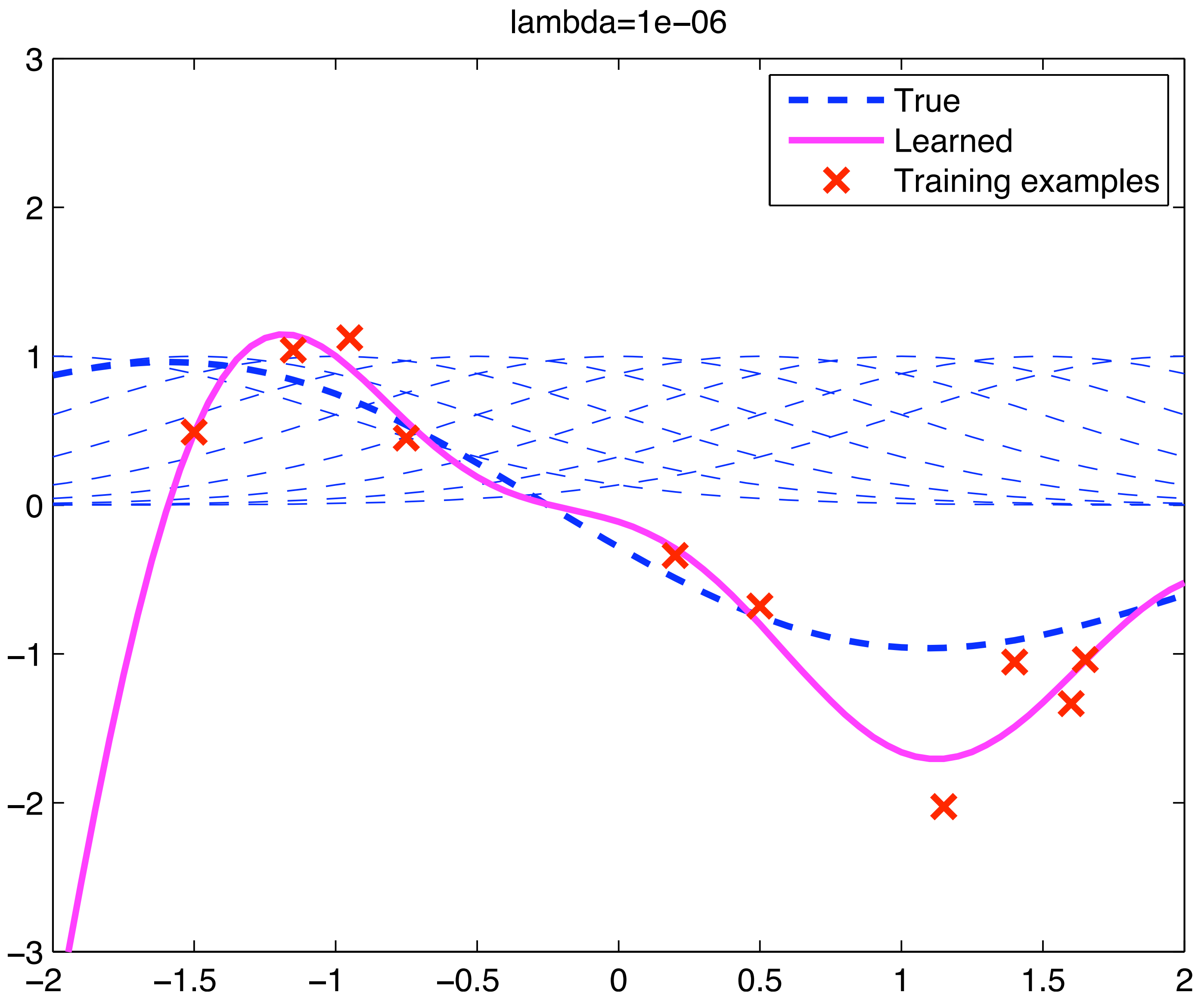
RR-RBF ($\lambda=10^{-8}$)



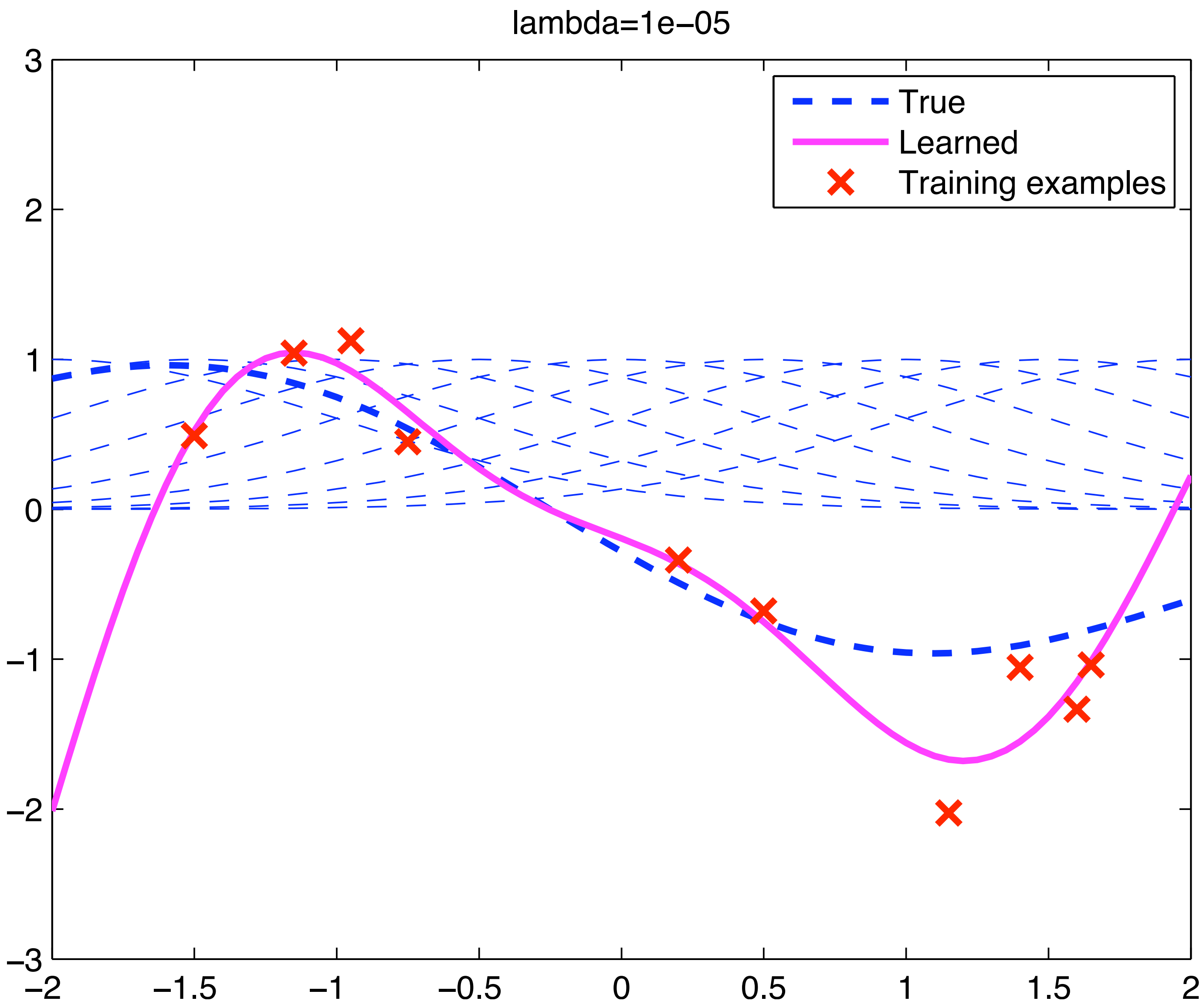
RR-RBF ($\lambda=10^{-7}$)



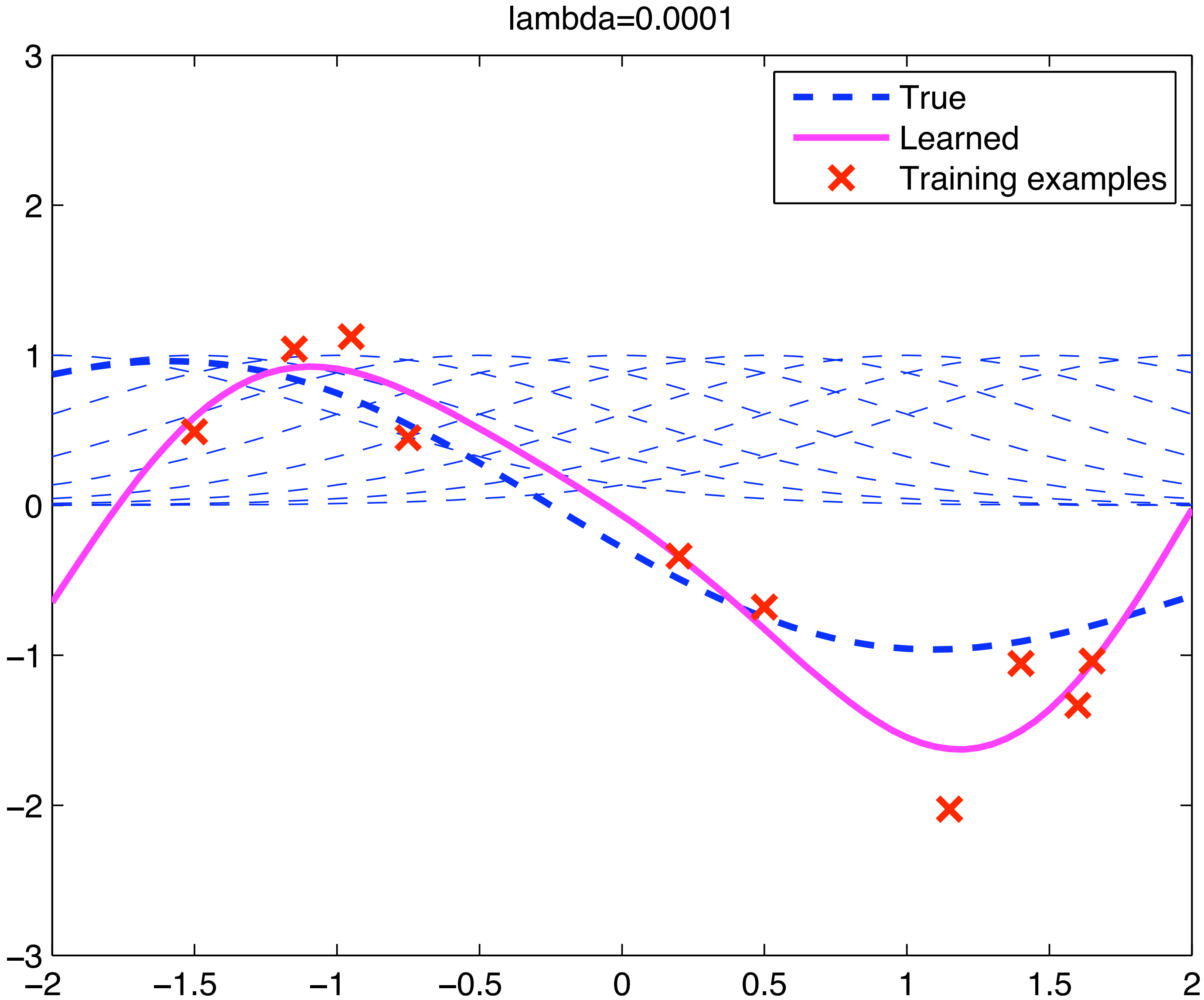
RR-RBF ($\lambda=10^{-6}$)



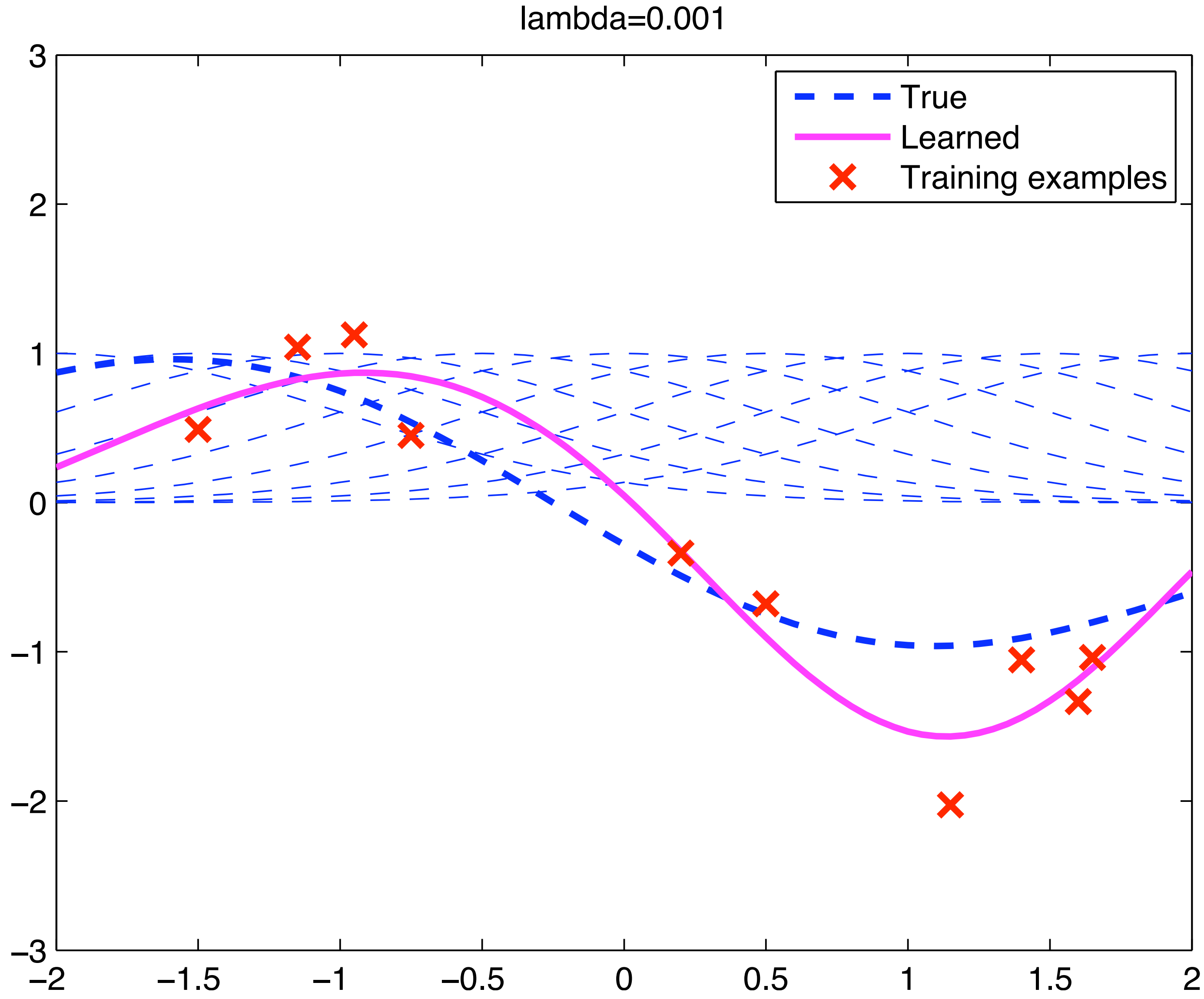
RR-RBF ($\lambda=10^{-5}$)



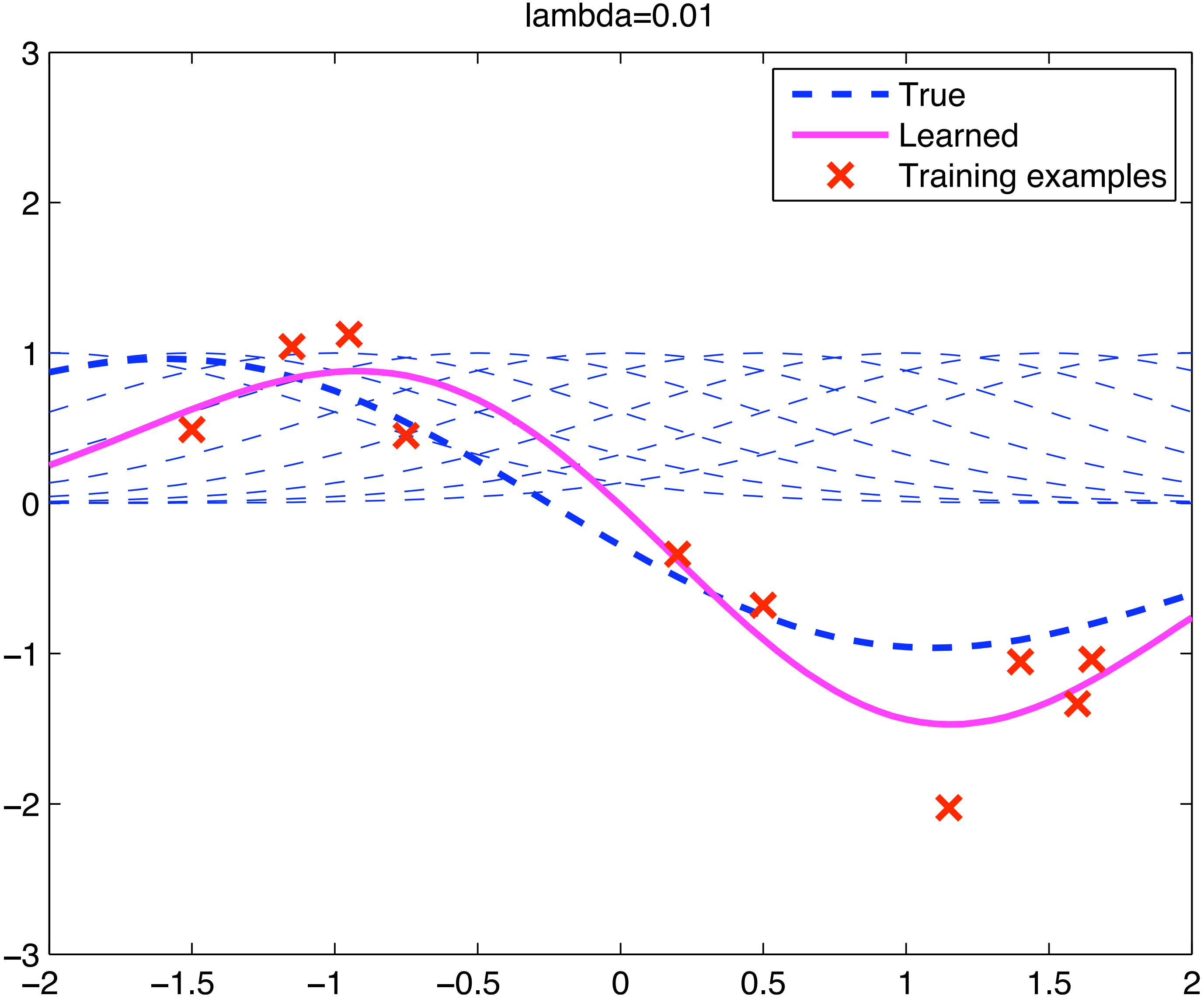
RR-RBF ($\lambda=10^{-4}$)



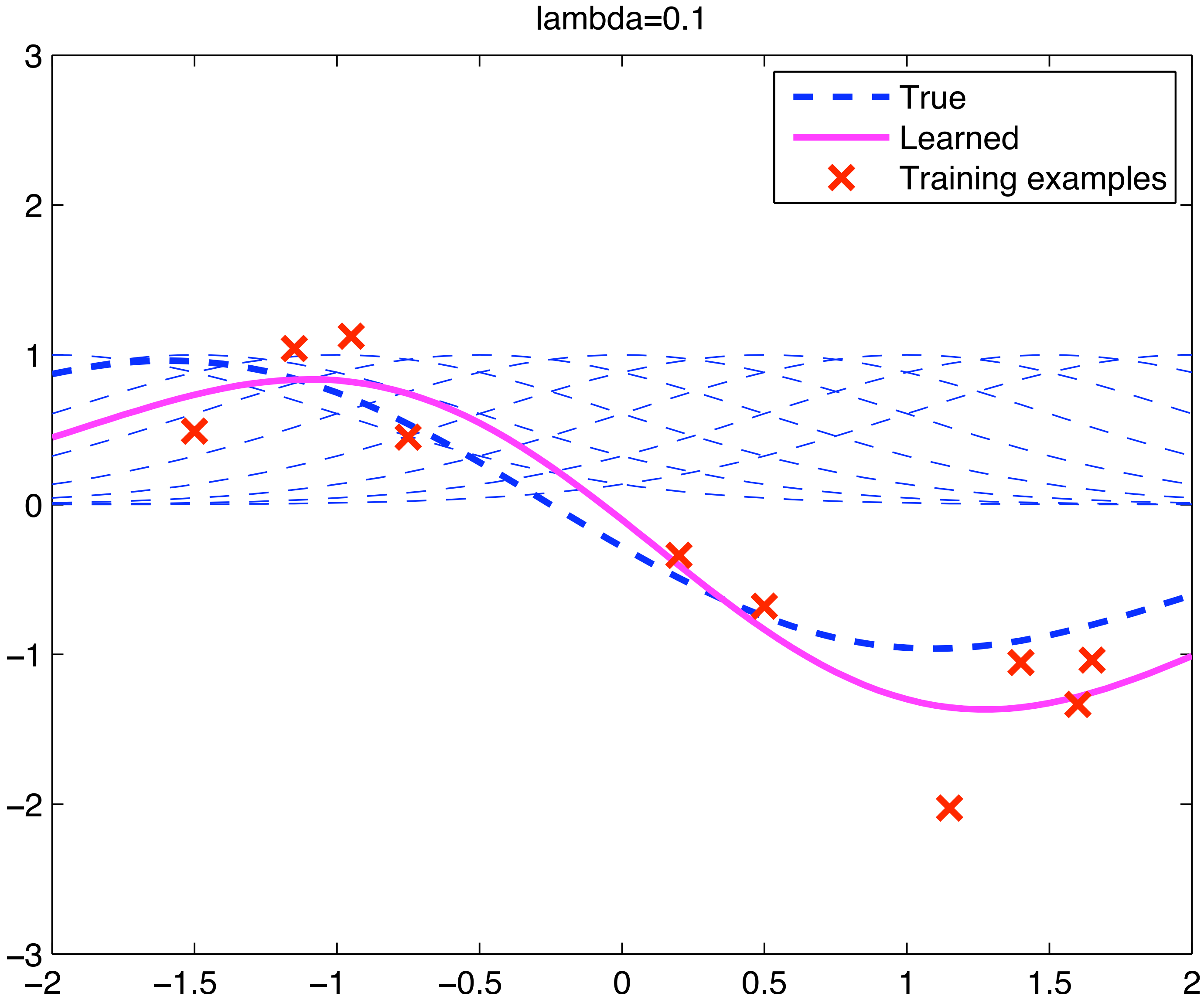
RR-RBF ($\lambda=10^{-3}$)



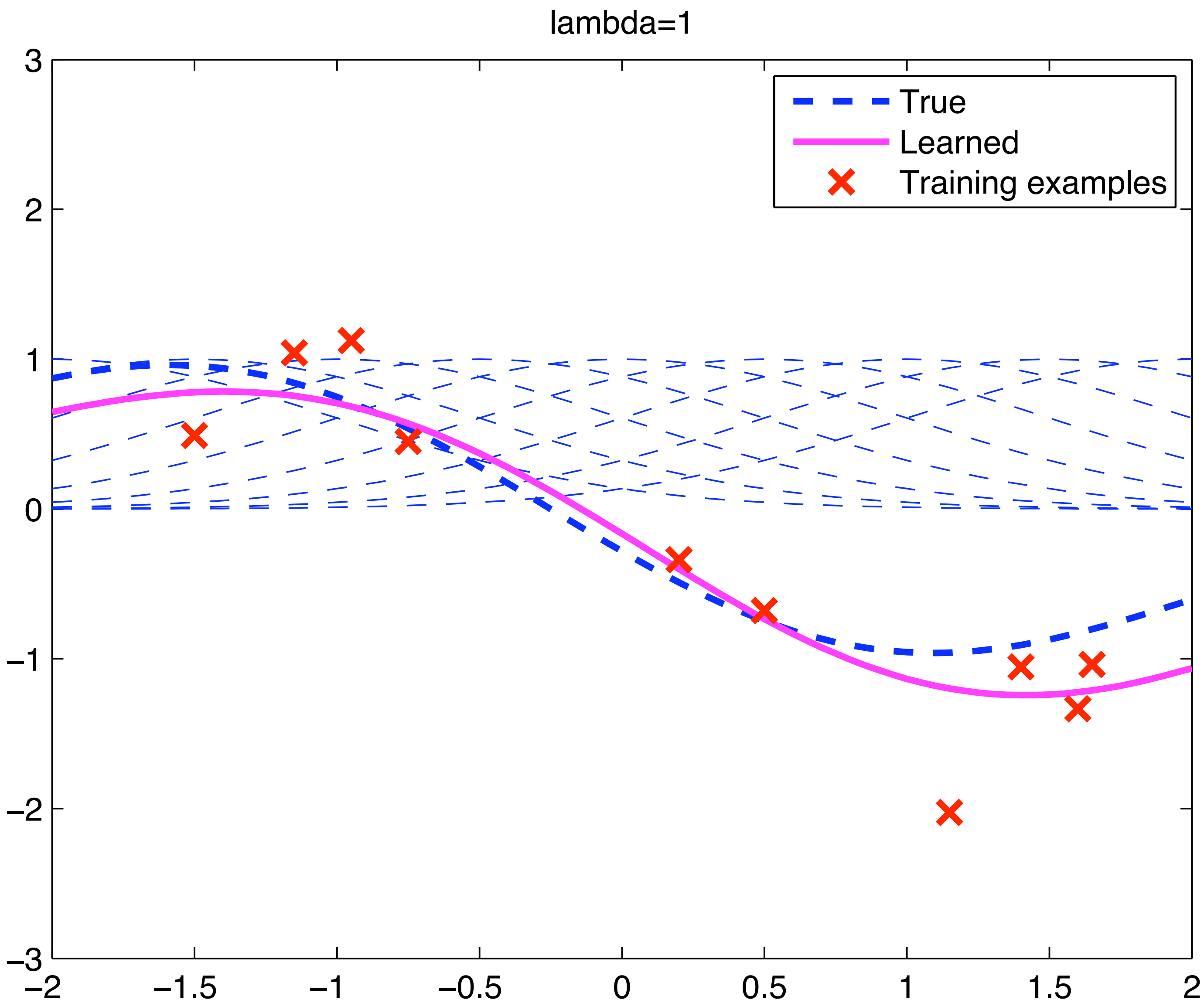
RR-RBF ($\lambda=10^{-2}$)



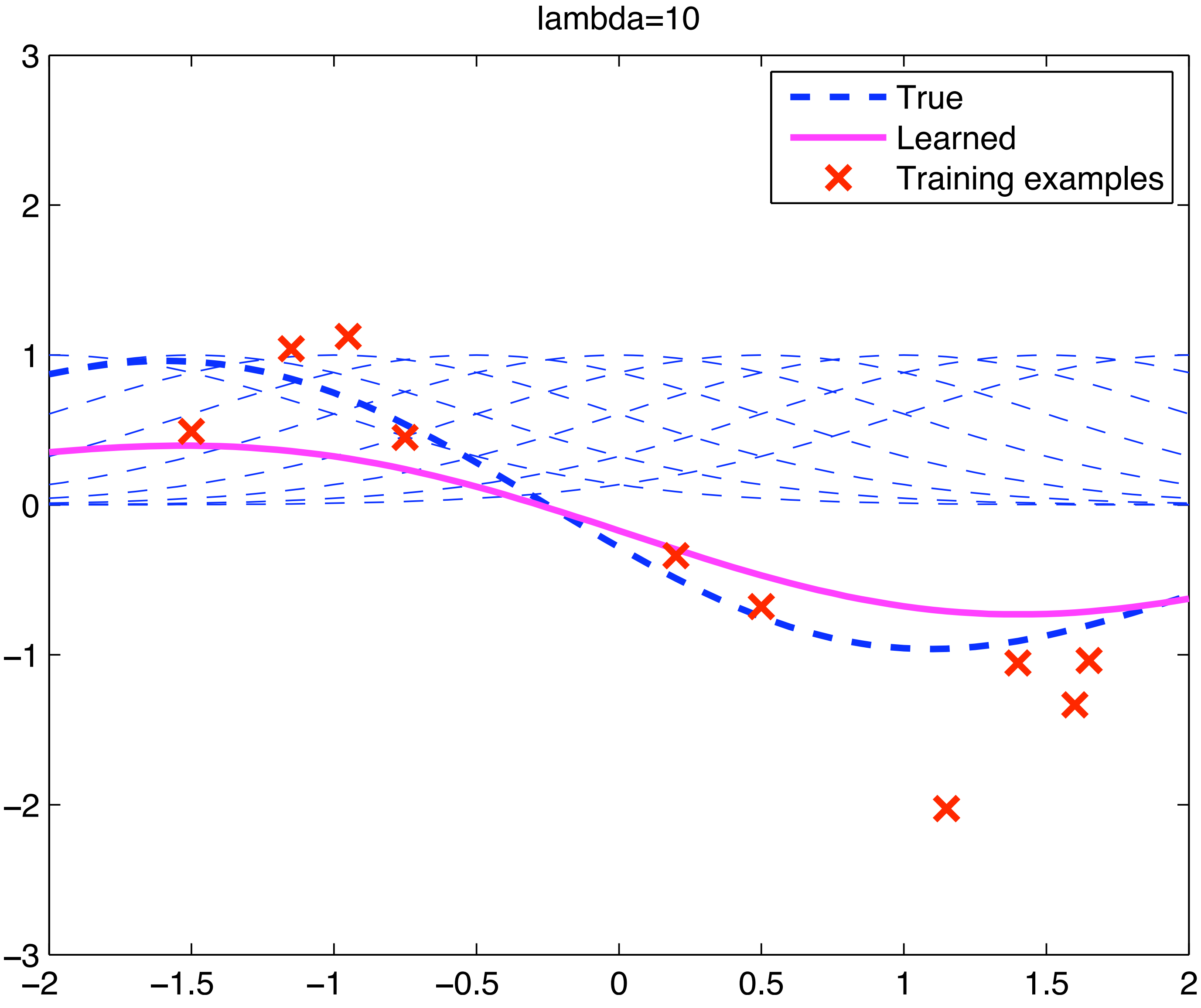
RR-RBF ($\lambda=10^{-1}$)



RR-RBF (λ=1)


lambda=1
True
Learned
Training examples
3
2
1
0
−1
−2
−3
−2
−1.5
−1
−0.5
0
0.5
1
1.5
2

RR-RBF (λ=10)


lambda=10
True
Learned
Training examples
3
2
1
0
−1
−2
−3
−2
−1.5
−1
−0.5
0
0.5
1
1.5
2

Binary classification

- Target y is +1 or -1.

Outputs to be predicted $\boldsymbol{y} = \begin{pmatrix} 1 \\ -1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$

Orange (+1) or lemon (-1)

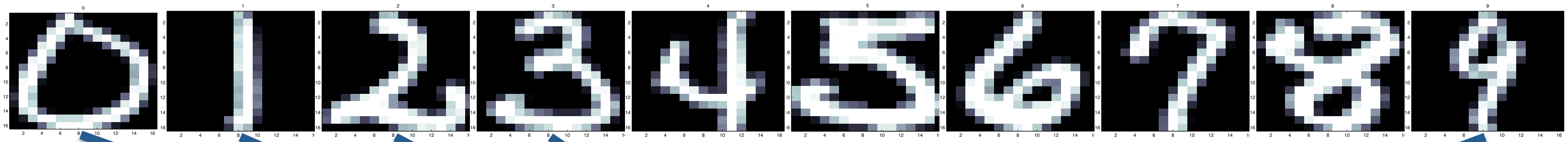
- Just apply ridge regression with +1/-1 targets
  - forget about the Gaussian noise assumption!

Multi-class classification

USPS digits dataset

7291 training samples,
2007 test samples

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

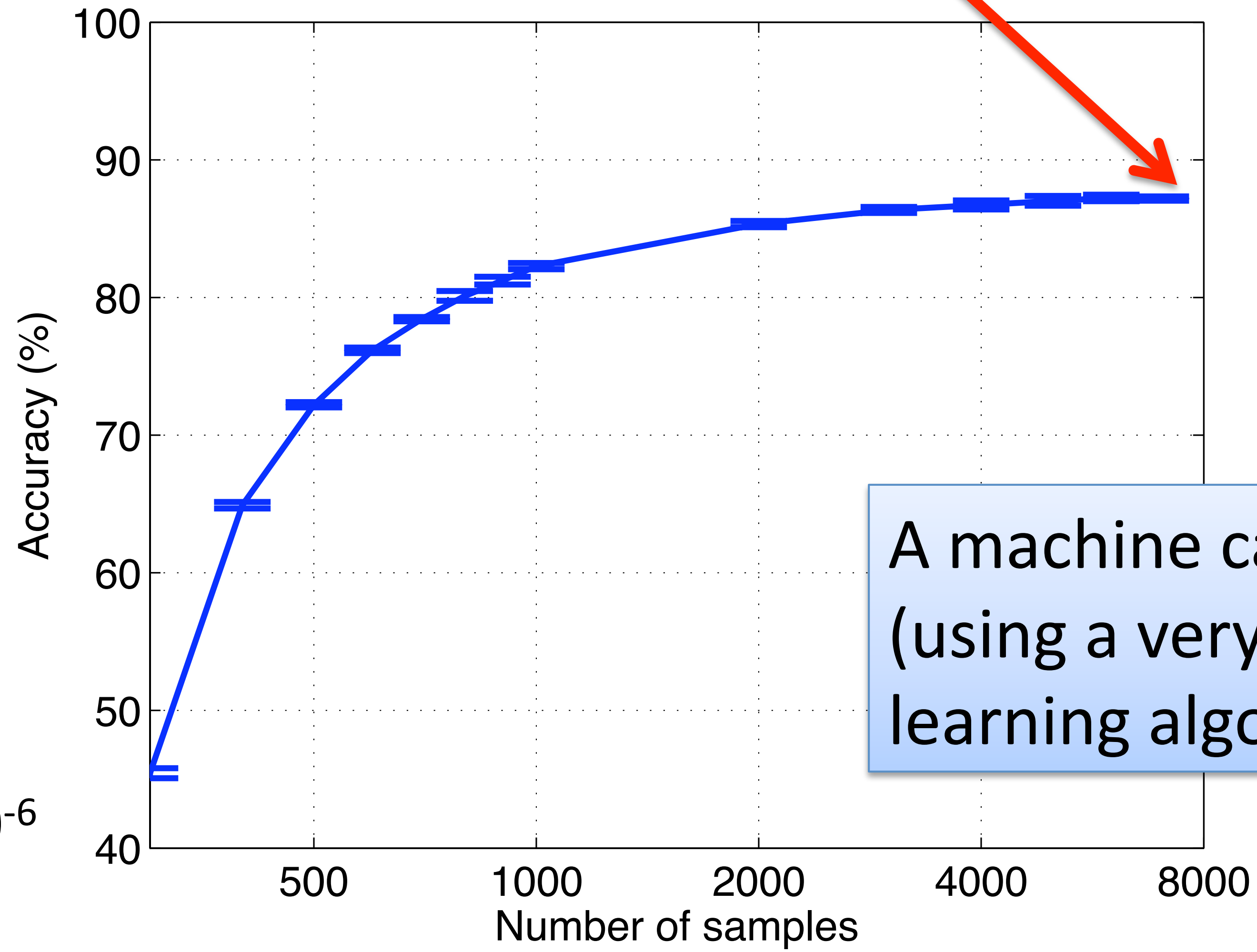


$$\boldsymbol{y} = \begin{pmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & & & & & \vdots \\ 0 & 1 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

Number of samples

USPS dataset

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



A machine can learn!
(using a very simple learning algorithm)

$\lambda=10^{-6}$

Summary (so far)

- Ridge regression (RR) is very simple.
- RR can be coded in one line:

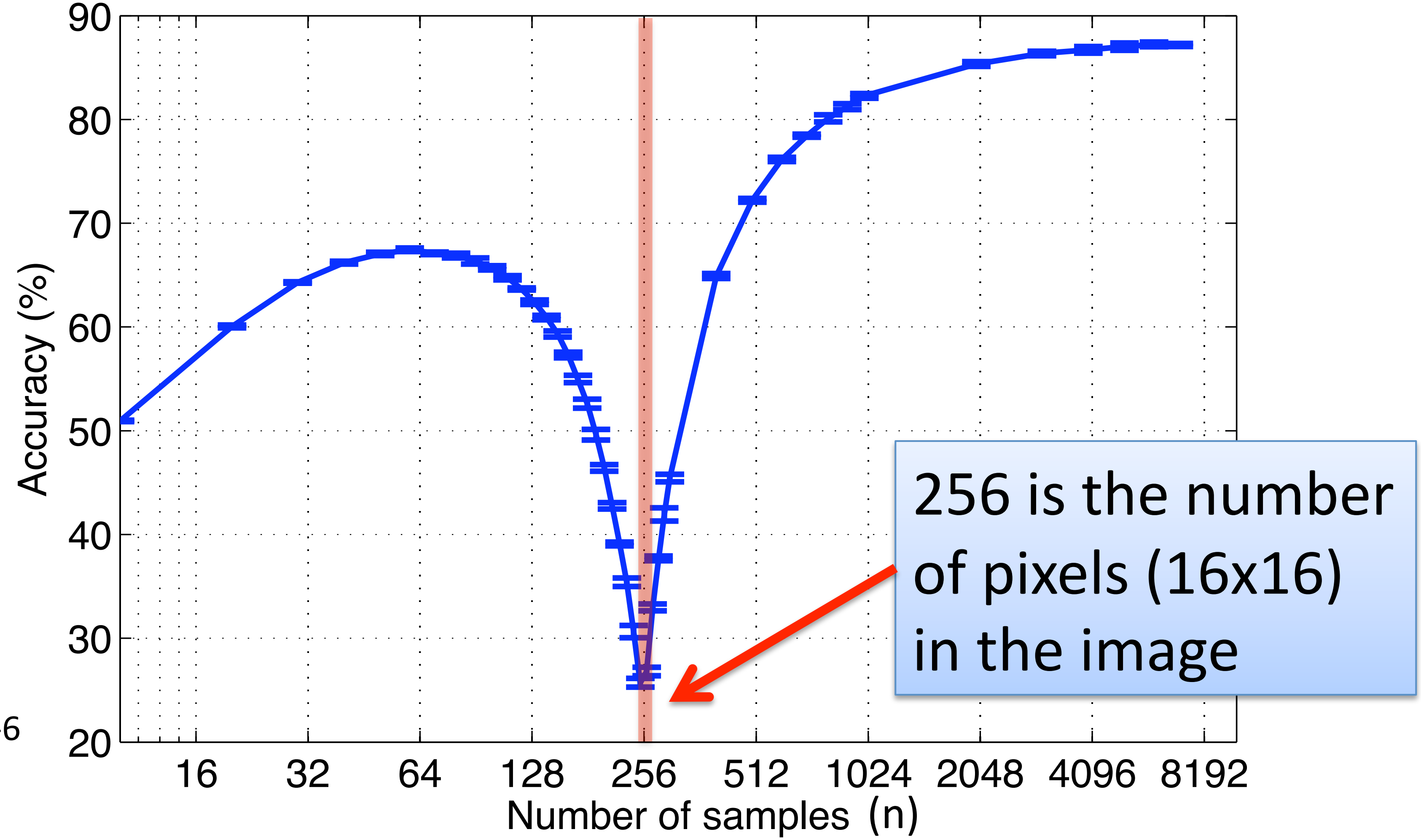
```
W=(X'*X+lambda*eye(n))\(X'*Y);
```

- 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

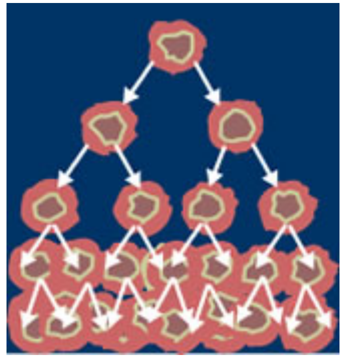
USPS dataset (p=256)
(What I have been hiding)

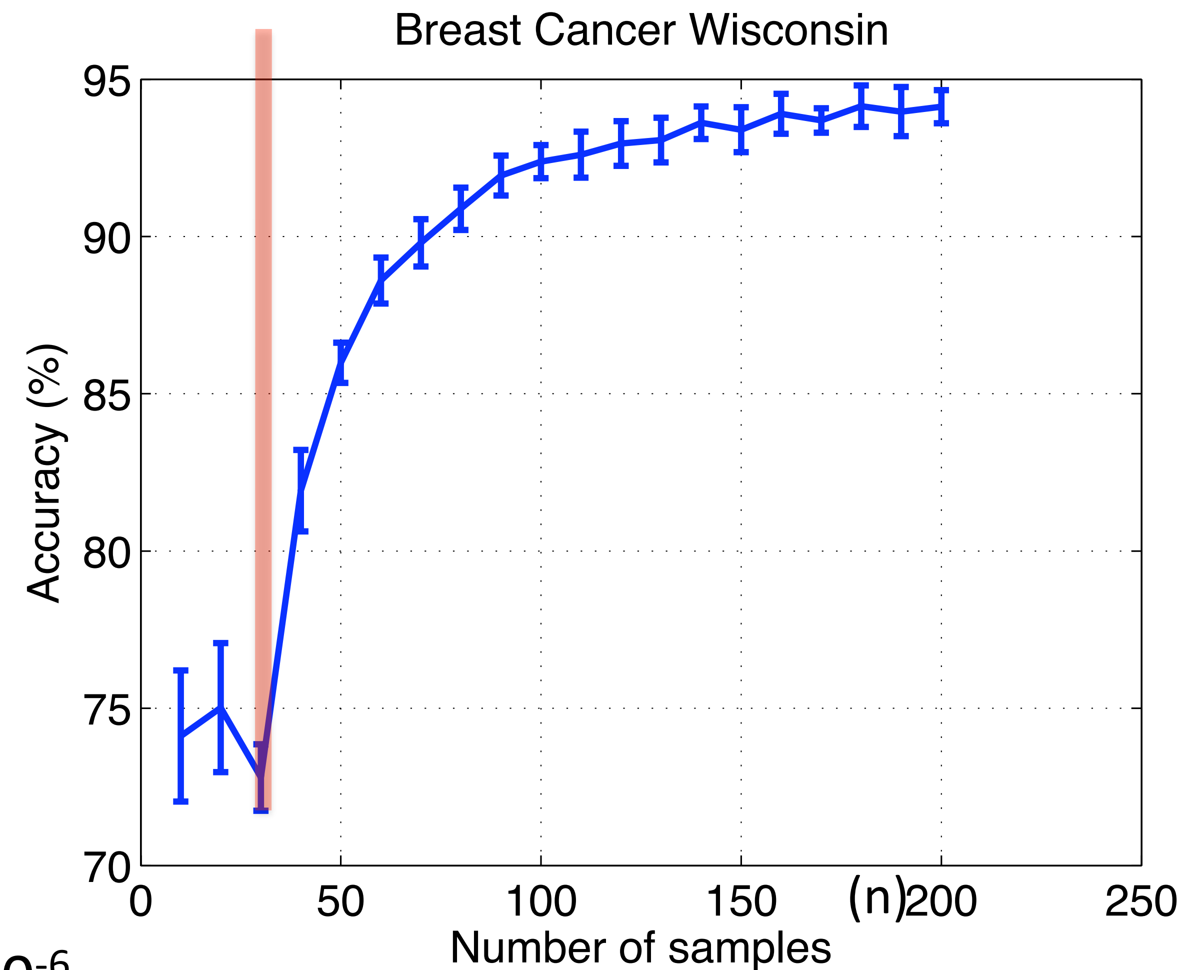
- The more data the less accurate??



$\lambda=10^{-6}$

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



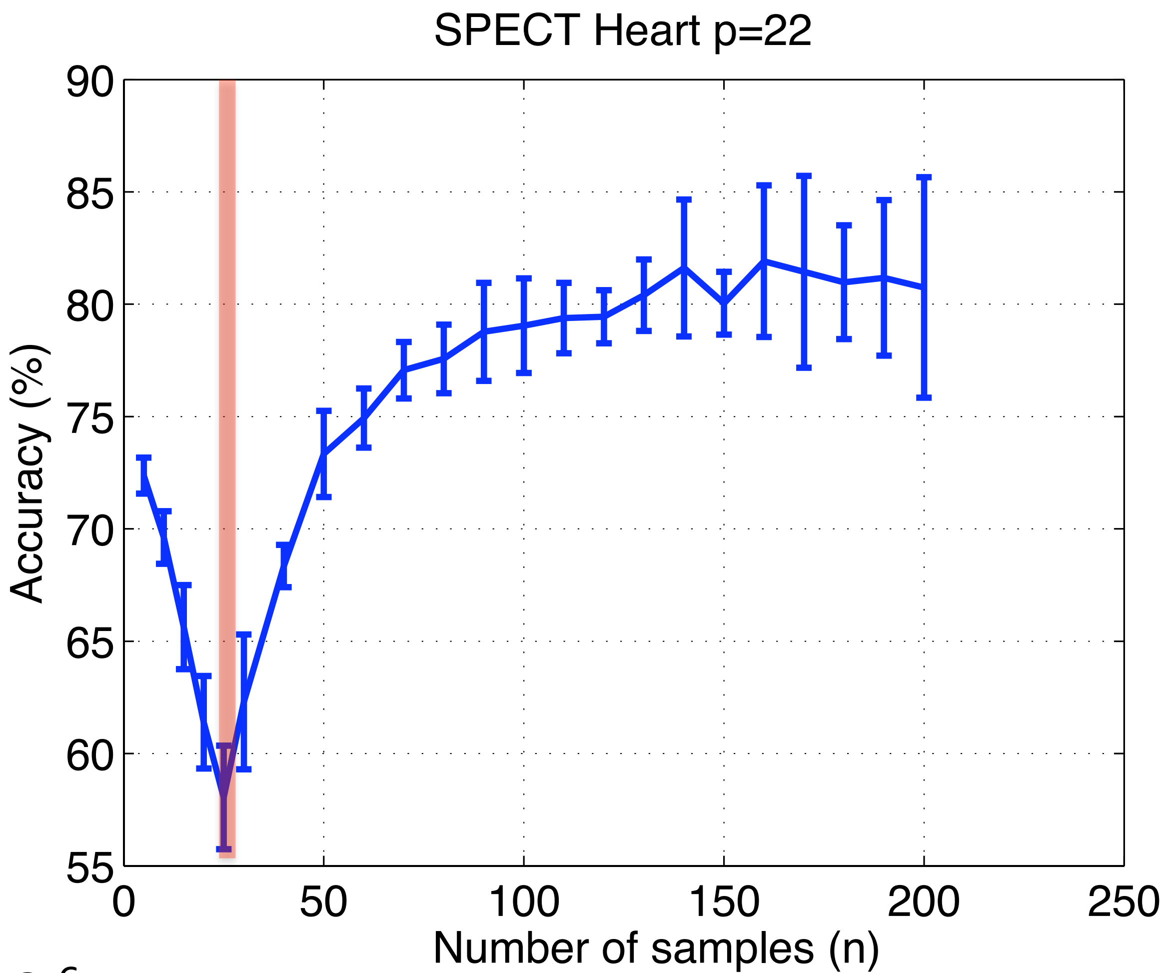


30 real-valued features

- radius
- texture
- perimeter
- area, etc.

$\lambda=10^{-6}$

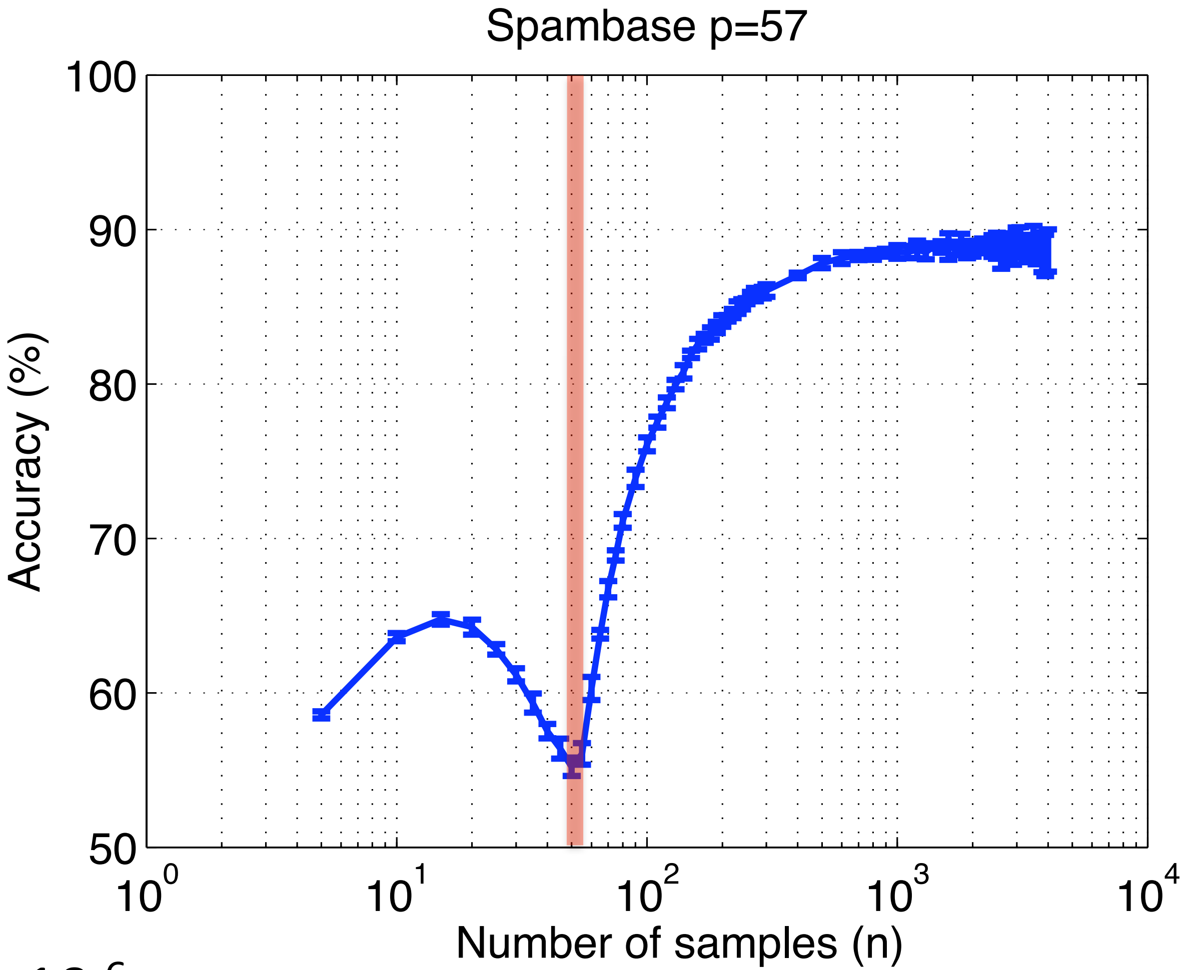
SPECT Heart dataset (p=22)



22 binary features

$\lambda = 10^{-6}$

Spambase dataset (p=57)

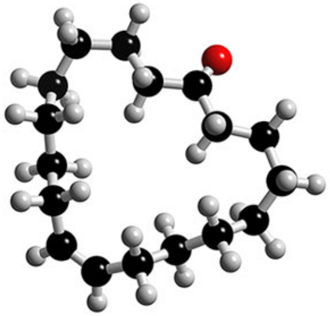


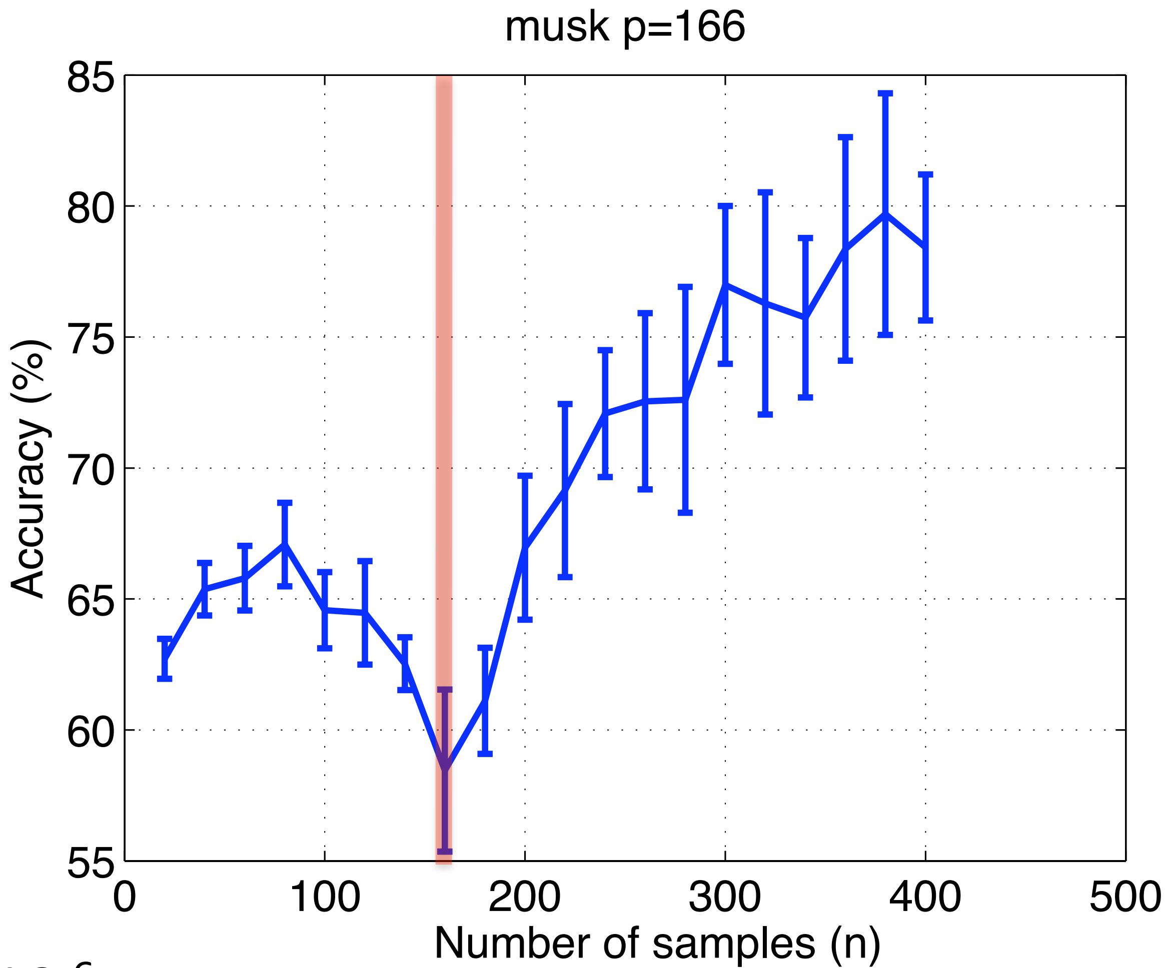
$\lambda=10^{-6}$

55 real-valued features
- word frequency
- character frequency

2 integer-valued feats
- run-length

Musk dataset (p=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

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

- Estimator

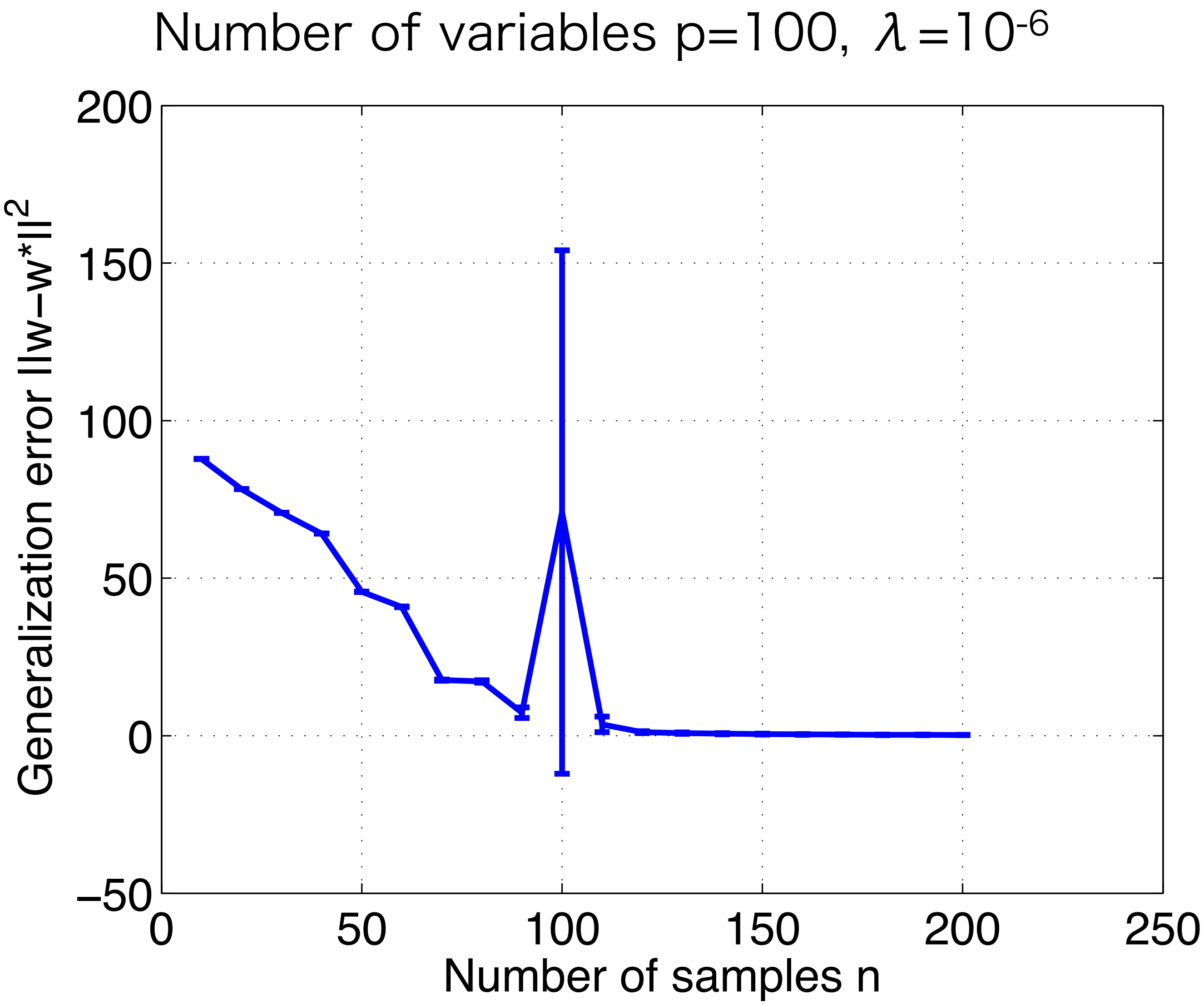
$$\hat{\boldsymbol{w}} = \left(\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I}_p\right)^{-1} \boldsymbol{X}^\top \boldsymbol{y}$$

- Generalization Error

$$\mathbb{E}_{\boldsymbol{\xi}} \left\| \hat{\boldsymbol{w}} - \boldsymbol{w}^* \right\|^2 \qquad \text{expectation over noise}$$

Numerically

Try `exp_ridge_regression.m`



First step

Let's show that

Bias-variance decomposition

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

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

Building blocks:

- linearity of expectation $\mathbb{E}[aX + bY] = a\mathbb{E}[X] + b\mathbb{E}[Y]$
- $\|\boldsymbol{x} + \boldsymbol{y}\|^2 = \|\boldsymbol{x}\|^2 + 2\boldsymbol{x}^\top \boldsymbol{y} + \|\boldsymbol{y}\|^2$

What does it mean?

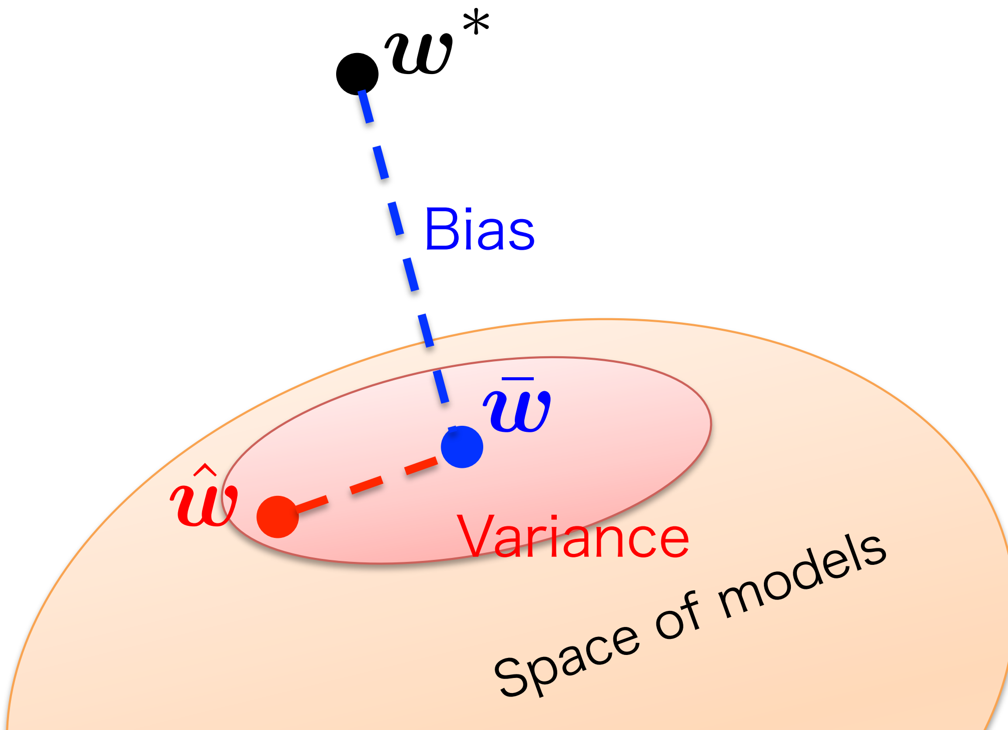
Bias-variance decomposition

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

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

Bias: error coming from the model/design matrix
- under-fitting

Variance: error caused by the noise - over-fitting



For ridge regression,

- Since $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w}^* + \boldsymbol{\xi}$ if $\mathbb{E}\boldsymbol{\xi} = 0$

$$\bar{\boldsymbol{w}} = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I}_p)^{-1} \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{w}^*$$

$$\hat{\boldsymbol{w}} - \bar{\boldsymbol{w}} = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I}_p)^{-1} \boldsymbol{X}^\top \boldsymbol{\xi}$$

Then what is bias? what is variance?

Analyze the bias

Show that

$$\|\bar{\boldsymbol{w}} - \boldsymbol{w}^*\|_2^2 = \sum_{i=1}^{p} \left(\frac{\lambda \boldsymbol{v}_i{}^\top \boldsymbol{w}^*}{s_i^2 + \lambda} \right)^2$$

where $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$:singular-value decomposition

$$\boldsymbol{U}^\top \boldsymbol{U} = \boldsymbol{I}_n,$$

$$\boldsymbol{V}^\top \boldsymbol{V} = \boldsymbol{I}_p,$$

$$\boldsymbol{\Sigma} = \operatorname{diag}(s_1, \ldots, s_m) \qquad (m = \min(n, p))$$

(Define s_i=0 if $i > m$)

Implications

- When n<p, RR is biased (even for $\lambda \to 0$)

$$\|\bar{\boldsymbol{w}} - \boldsymbol{w}^*\|^2 \xrightarrow{\lambda \to 0} \begin{cases} \sum_{i=n+1}^{p} \left({\boldsymbol{v}_i}^\top \boldsymbol{w}^* \right)^2 & (n < p), \\ 0 & (\text{otherwise}). \end{cases}$$

- Bias monotonically decreases with increasing sample size n
- Bias comes from X (n×p) not being able to span the whole feature space

Analyze the variance

- Assume that the noise ξ_i is independent and have identical variance σ^2

(This part depends on what we assume about the noise)

- Then show that

$$\mathbb{E}_{\boldsymbol{\xi}}\|\hat{\boldsymbol{w}} - \bar{\boldsymbol{w}}\|_2^2 = \sigma^2 \mathrm{Tr}\left((\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I}_p)^{-2} \boldsymbol{X}^\top \boldsymbol{X}\right)$$

$$= \sigma^2 \sum_{i=1}^{m} \frac{s_i^2}{(s_i^2 + \lambda)^2}$$

where m=min(n,p)

Building block: $\mathrm{Tr}(\boldsymbol{A}\boldsymbol{B}) = \mathrm{Tr}(\boldsymbol{B}\boldsymbol{A})$

Implications

- Contribution from small singular-values can be large when $\lambda \to 0$

$$\text{Variance} = \sigma^2 \sum_{i=1}^{m} \frac{s_i^2}{(s_i^2 + \lambda)^2} \xrightarrow{\lambda \to 0} \sigma^2 \sum_{i=1}^{m} s_i^{-2}$$

- When does the smallest singular-value hit zero?
  ⇒ around n=p (Marchenko–Pastur)

Marchenko-Pastur distribution

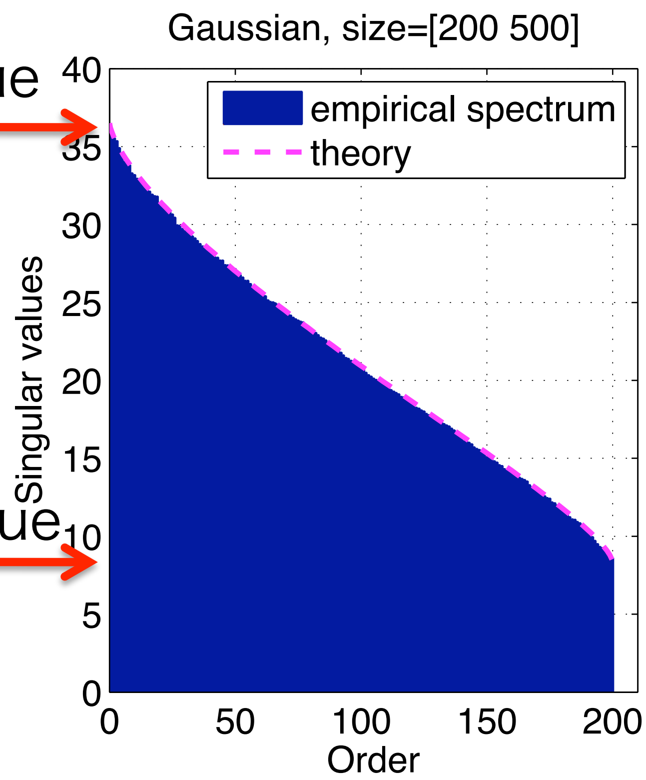
Largest singular-value

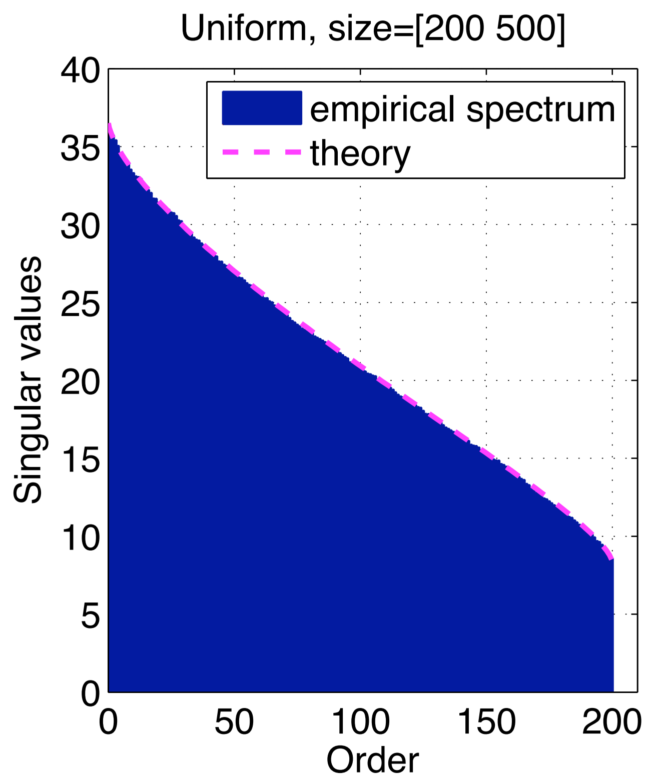
$$\sqrt{n} + \sqrt{p}$$

Smallest singular-value

$$\sqrt{n} - \sqrt{p}$$

(if n > p)





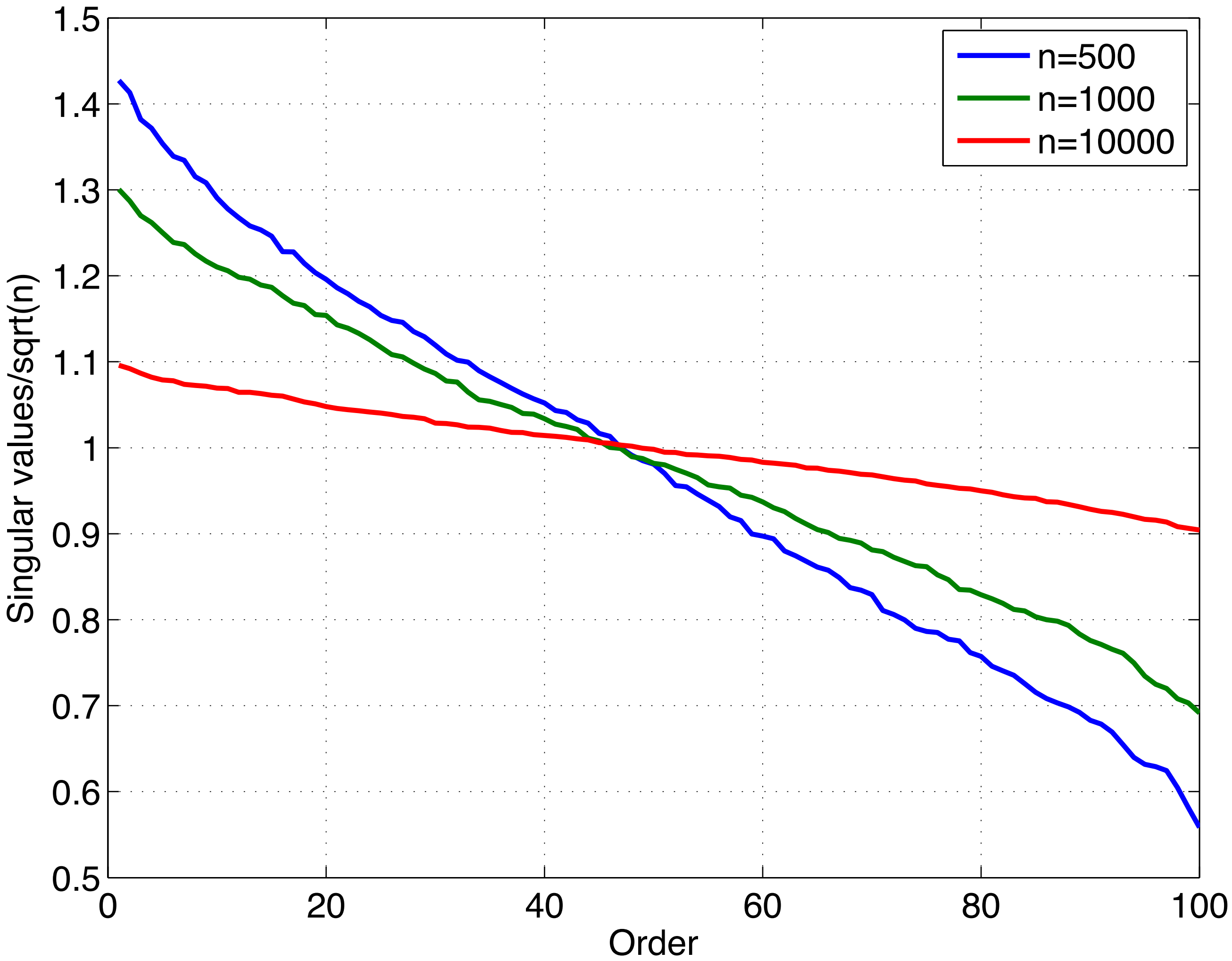
Try `exp_marchenko_pastur.m`

When n >> p

All singular values concentrates around $\sqrt{n}$

$$1 + \sqrt{\frac{p}{n}}$$

$$1 - \sqrt{\frac{p}{n}}$$



When n >> p

- In this regime,

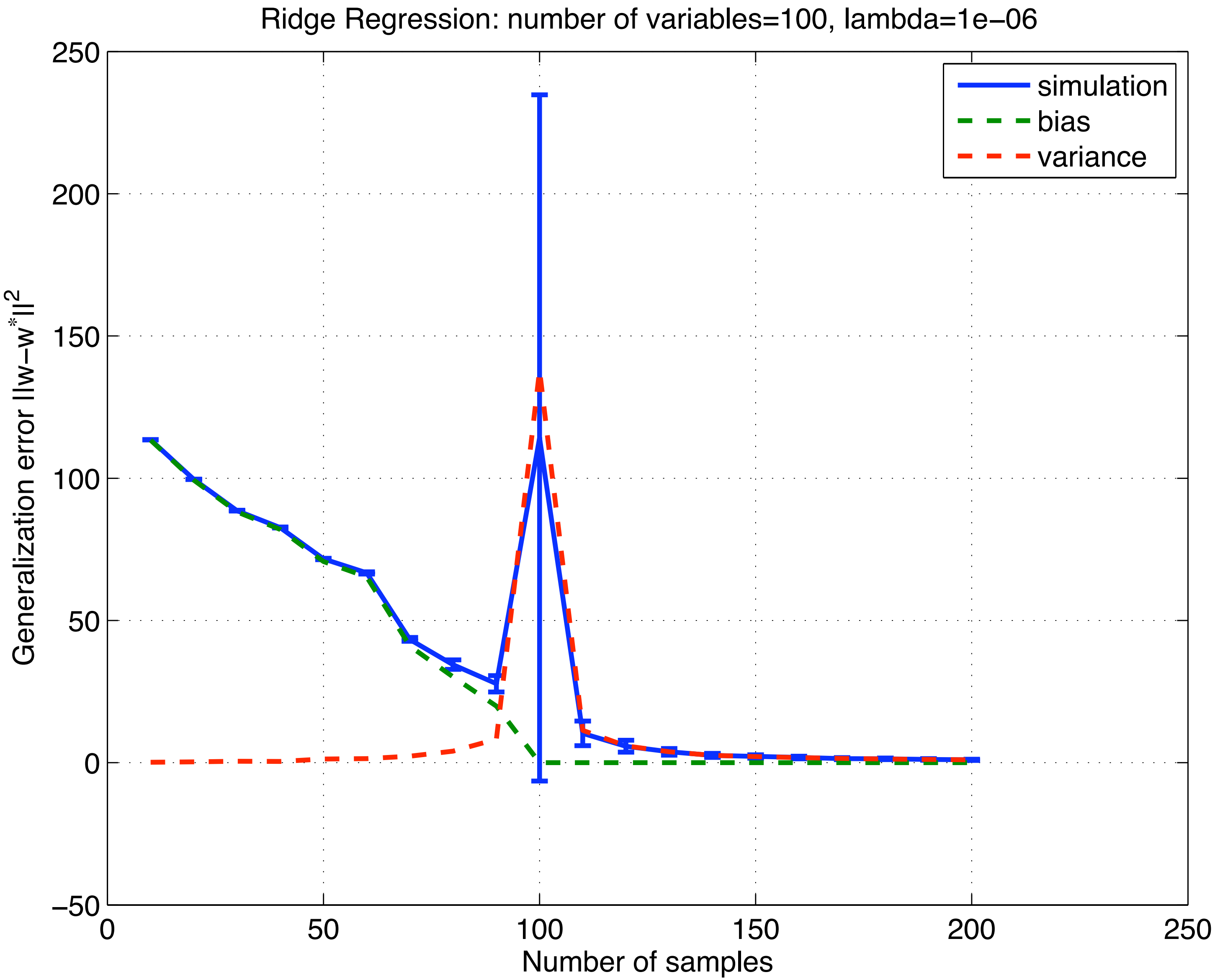
$$\text{Variance} = \sigma^2 \frac{np}{(n+\lambda)^2} \xrightarrow{\lambda \to 0} \sigma^2 \frac{p}{n}$$

The number of samples n we need to get certain error scales linearly with the number of dimension p

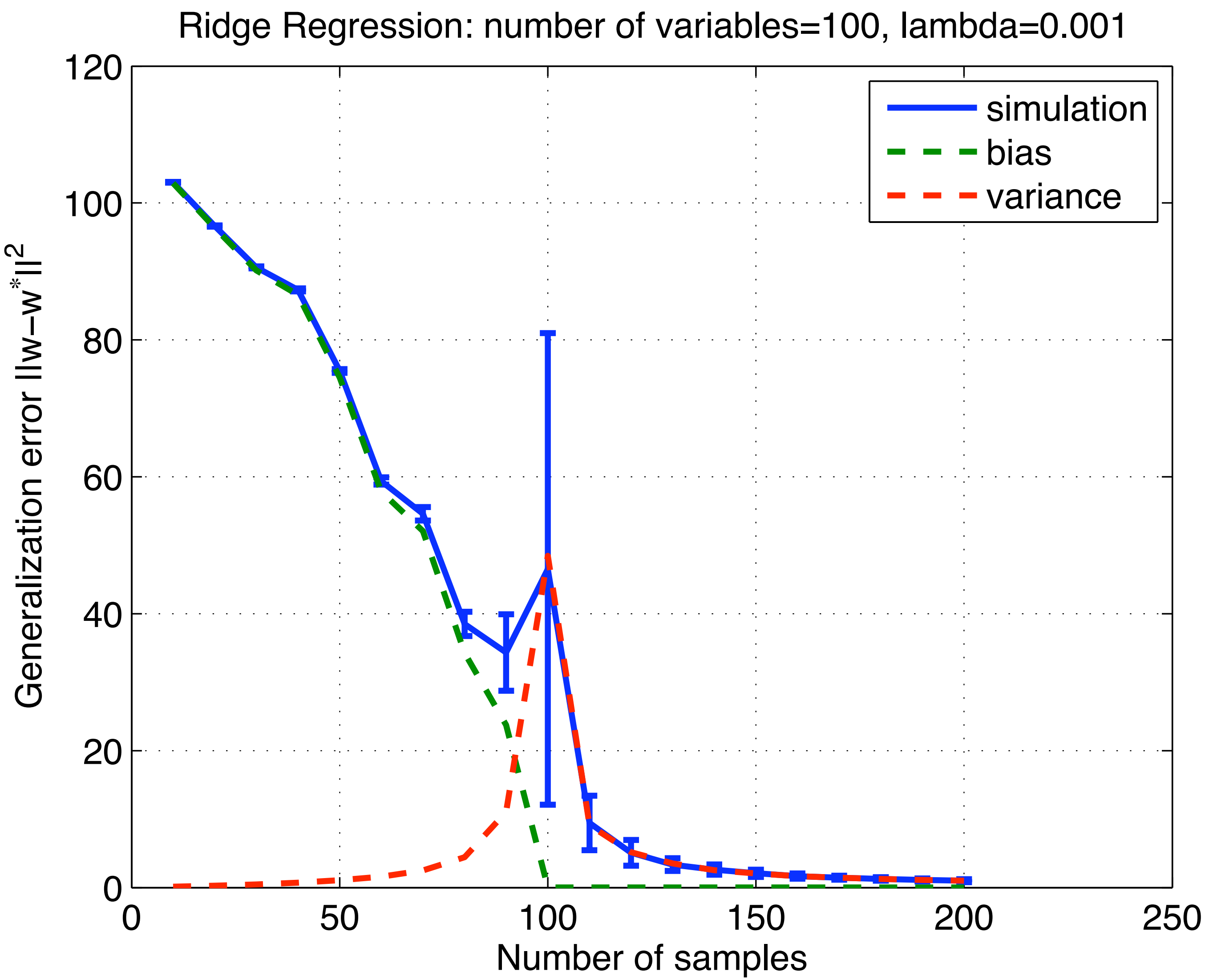
Summary of the analysis

- Bias decreases monotonically with the number of samples
  - bias = 0 for n > p.
- Variance scales like $\sigma^2 \sum_{i=1}^{\min(n,p)} s_i^{-2}$ when λ is small.
  - can be large around n=p

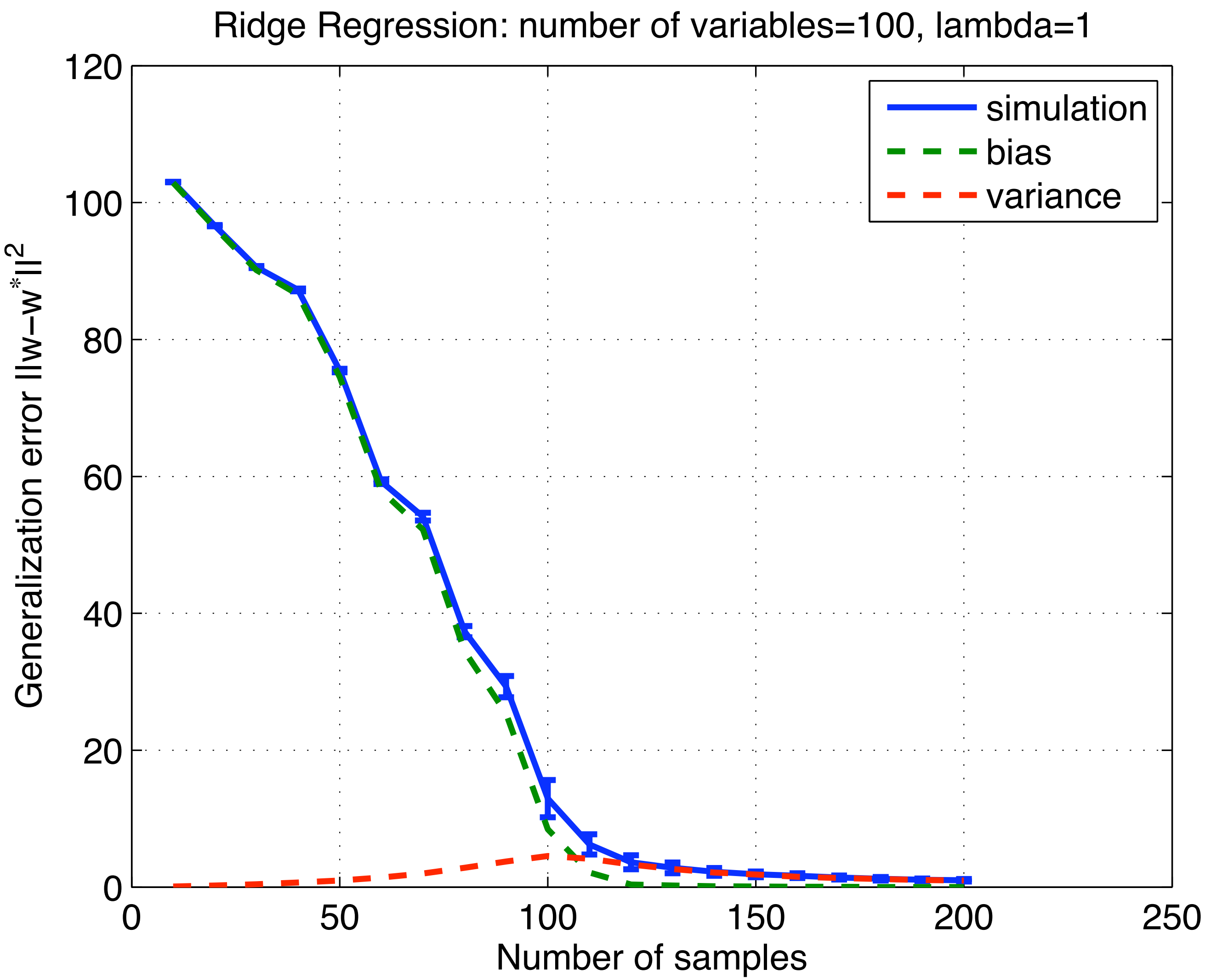
Result ($\lambda=10^{-6}$)



Result (λ=0.001)



Result (λ=1)



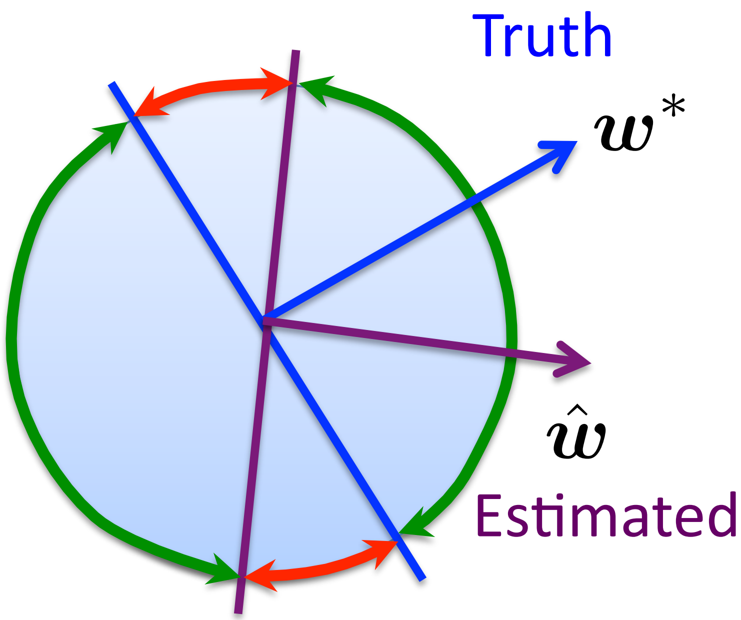
Ridge Regression: number of variables=100, lambda=1
Generalization error ||w−w*||²
Number of samples
simulation
bias
variance
0
20
40
60
80
100
120
0
50
100
150
200
250

How about classification?

- Model
  - Input vector x_i is sampled from standard Gaussian distribution (x_i is a random variable):
    $$x_i \sim \mathcal{N}(0, \boldsymbol{I}_p) \quad (i = 1, \dots, n)$$
  - The true classifier is also a normal random variable: $\boldsymbol{w}^* \sim \mathcal{N}(0, \boldsymbol{I}_p)$
  - Output $\boldsymbol{y} = \mathrm{sign}(\boldsymbol{X}\boldsymbol{w}^*)$

(Not a Gaussian noise!)

- Generalization Error
  $$\epsilon = \frac{1}{\pi} \arccos\left(\frac{\hat{\boldsymbol{w}}^\top \boldsymbol{w}^*}{\|\hat{\boldsymbol{w}}\|\|\boldsymbol{w}^*\|}\right)$$



Analyzing classification

[Opper and Kinzel (1995) Statistical Mechanics of Generalization]

- Let α = n/p and assume that

Number of samples	Number of features	Regularization constant
$n \to \infty,$	$p \to \infty,$	$\lambda \to 0$

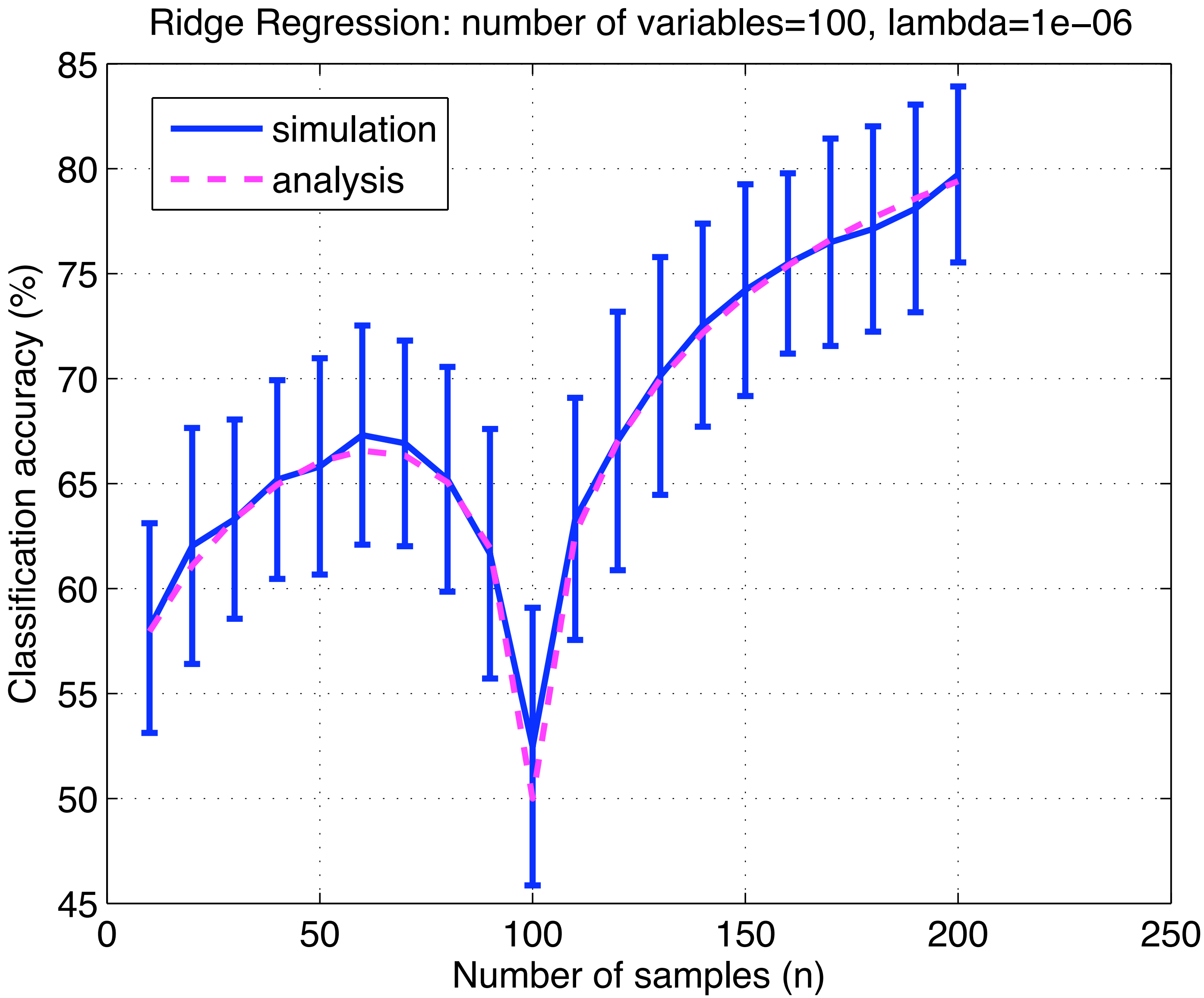
- Analyze the inner product

$$\mathbb{E}\hat{\boldsymbol{w}}^\top \boldsymbol{w}^* = \begin{cases} \sqrt{p}\sqrt{\frac{2}{\pi}\alpha} & (\alpha < 1), \\ \sqrt{p}\sqrt{\frac{2}{\pi}} & (\alpha > 1). \end{cases}$$

- Analyze the norm

$$\mathbb{E}\|\hat{\boldsymbol{w}}\|^2 = \begin{cases} \frac{\alpha(1-\frac{2}{\pi}\alpha)}{1-\alpha} & (\alpha < 1), \\ \frac{\frac{2}{\pi}(\alpha-1)+1-\frac{2}{\pi}}{\alpha-1} & (\alpha > 1). \end{cases} \qquad \mathbb{E}\|w^*\|^2 = p.$$

Analyzing classification (result)

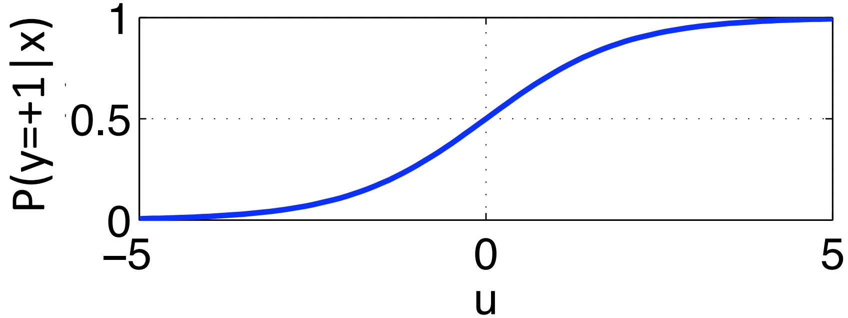


How can we avoid the singularity?

✓Regularization

✓Logistic regression

$$\log \frac{P(y=+1|\boldsymbol{x})}{P(y=-1|\boldsymbol{x})} = \boldsymbol{w}^\top \boldsymbol{x}$$

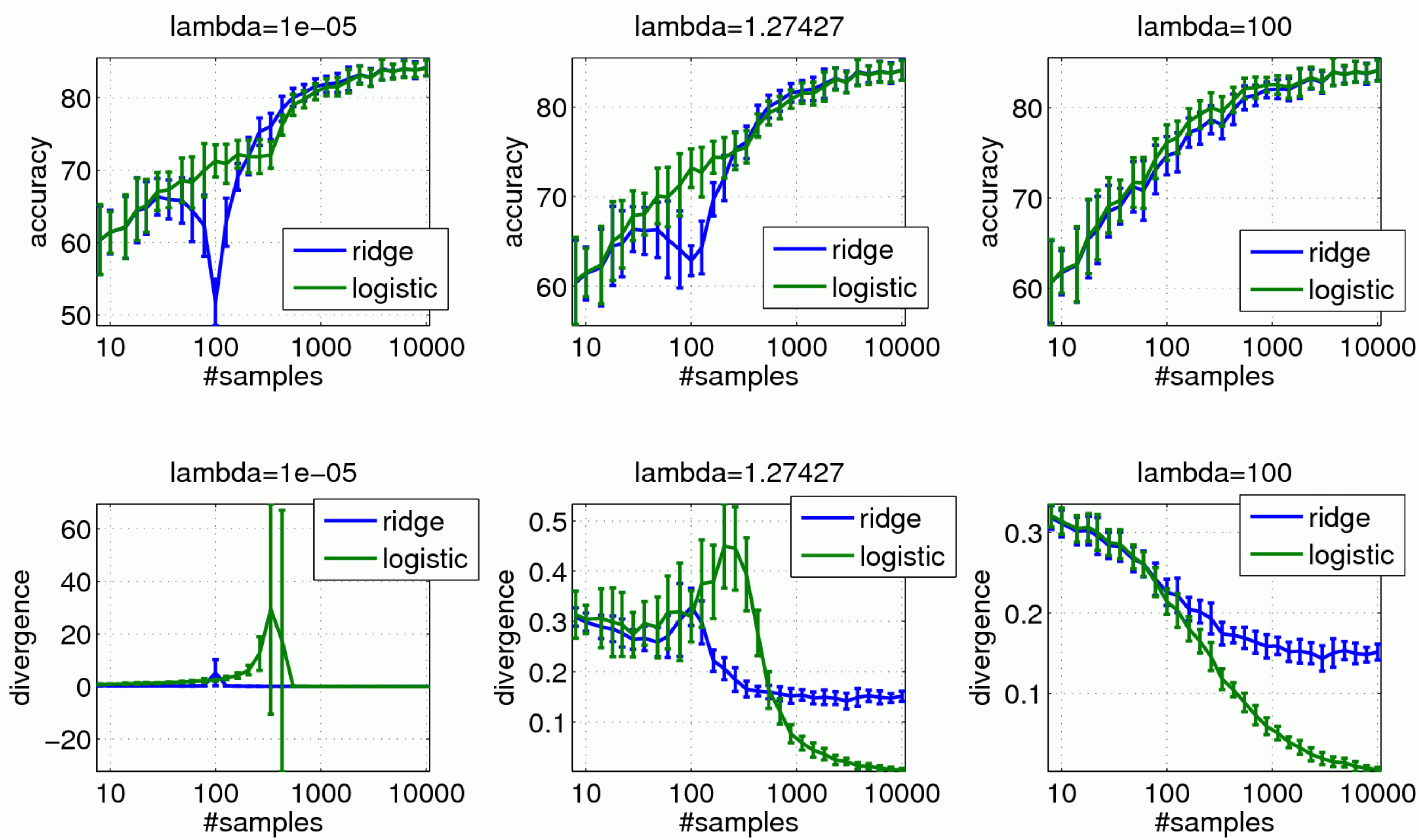


$$\underset{\boldsymbol{w}}{\text{minimize}} \quad \underbrace{\sum_{i=1}^{n} \log(1+\exp(-y_i \boldsymbol{w}^\top \boldsymbol{x}_i))}_{\text{Training error}} + \underbrace{\frac{\lambda}{2}\|\boldsymbol{w}\|^2}_{\text{Regularization term}}$$

Training error

Regularization term
(λ: regularization const.)

Can we avoid singularity?



Summary

- Ridge regression (RR) is very simple and easy to implement.
- RR has wide application, e.g., classification, multi-class classification
- Be careful about the singularity. Adding data does not always help improve performance.
- Analyzing the singularity: predicts the simulated performance quantitatively.
  - Regression setting: variance goes to inifity at n=p.
  - Classification setting: norm $\|\hat{\boldsymbol{w}}\|^2$ goes to inifinity at n=p.

LASSO

This part is heavily based on
"A Unified Framework for High-Dimensional Analysis of M-Estimators with Decomposable Regularizers" by Negahban et al. (2012)

Also I'd like to thank my colleague Taiji Suzuki for suggestions.

What is Lasso?

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}\in\mathbb{R}^p}{\operatorname{argmin}} \left(\frac{1}{2n} \underbrace{\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2}_{\substack{\text{Squared error} \\ \text{(same as RR)}}} + \lambda_n \underbrace{\|\boldsymbol{w}\|_1}_{\substack{\text{L}_1\text{ norm} \\ \text{(promotes sparsity)}}} \right)$$

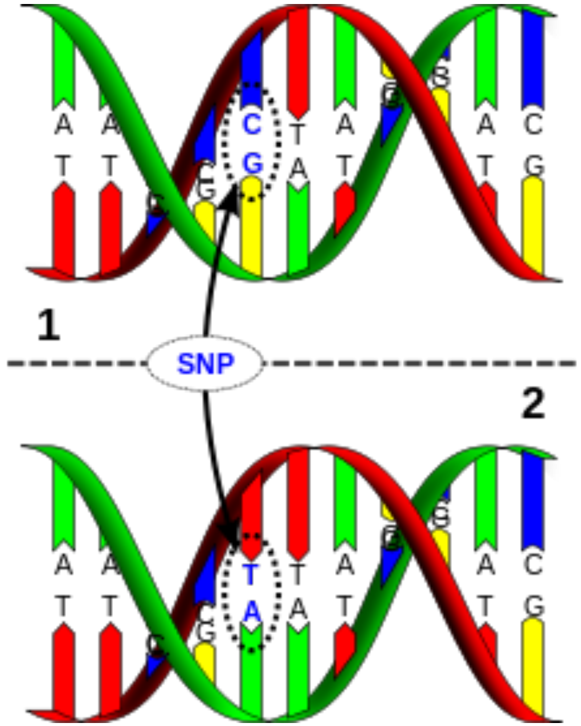
L_1 norm: $$\|\boldsymbol{w}\|_1 = \sum_{j=1}^{p} |w_j|$$

Least Absolute Shrinkage and Selection Operator (Tibshirani 1996)

"Historically, the L_1 estimation methods go back to Galileo (1632) and Laplace (1793)..." (Rudin, Osher, Fatemi 1992)

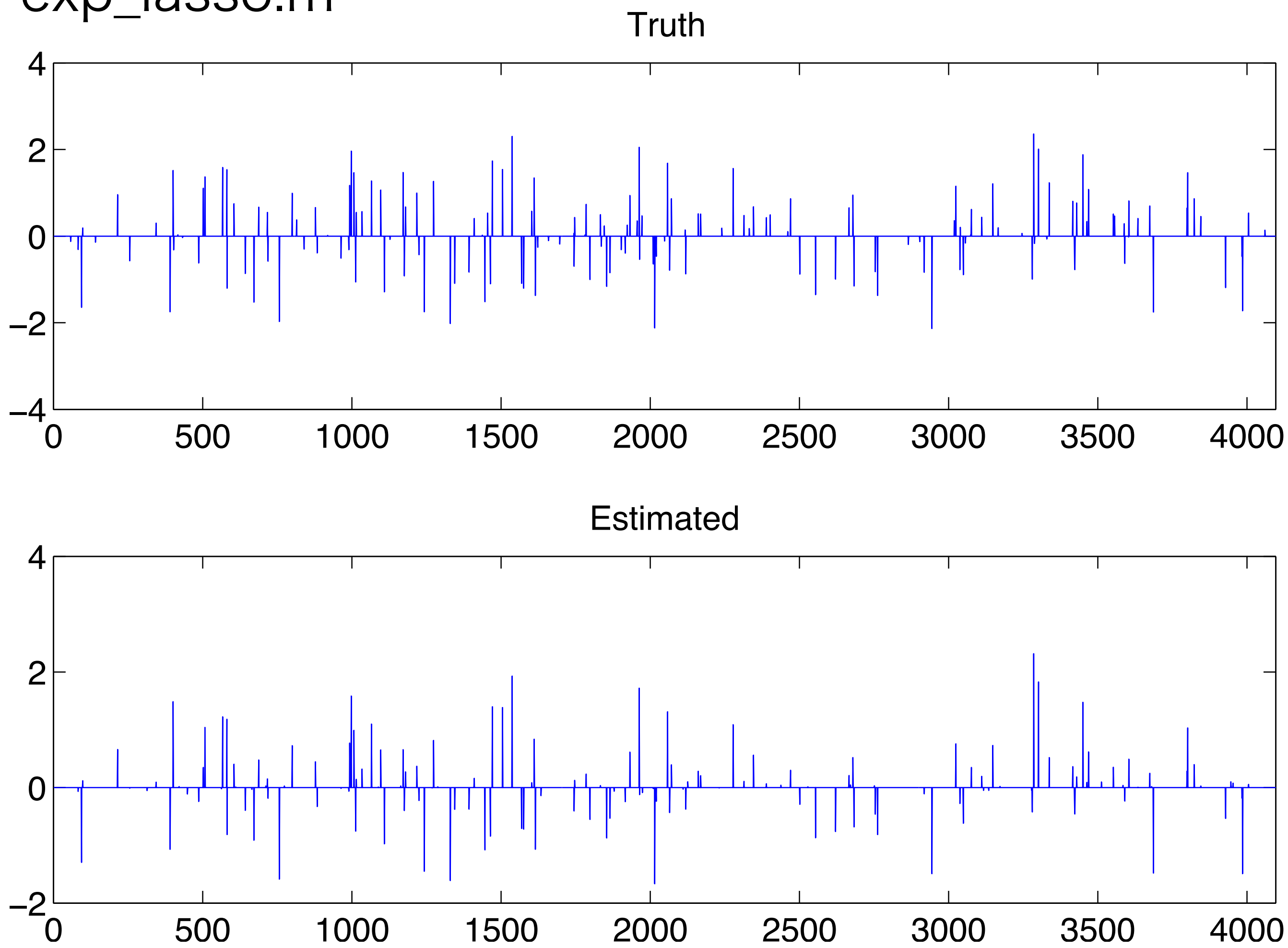
Why sparsity?

- Imagine a classification problem with n << p but many variables are probably irrelevant.
  - How do we select relevant variables?
- L_1 is a basis for more complex structures (e.g., group lasso, low-rank matrices)



Example (n=1024, p=4096)

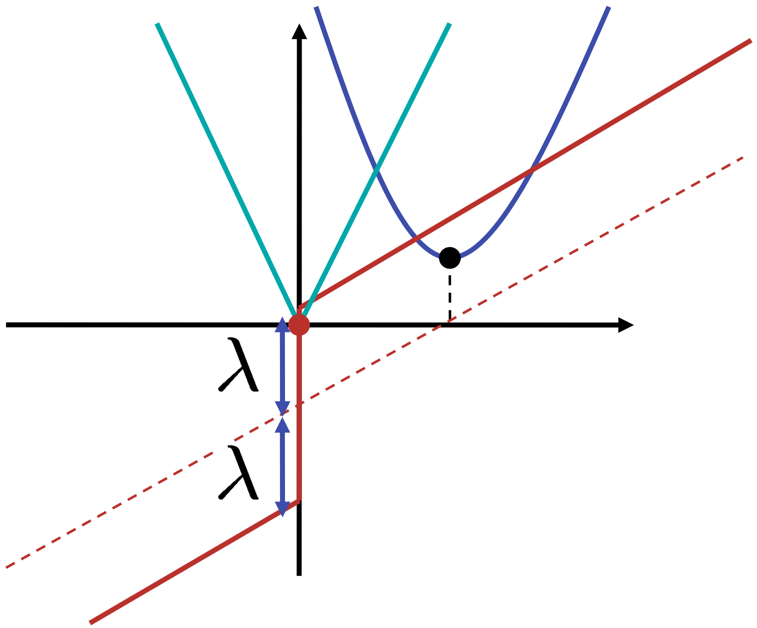
Try exp_lasso.m



Most non-zero coefficients are recovered

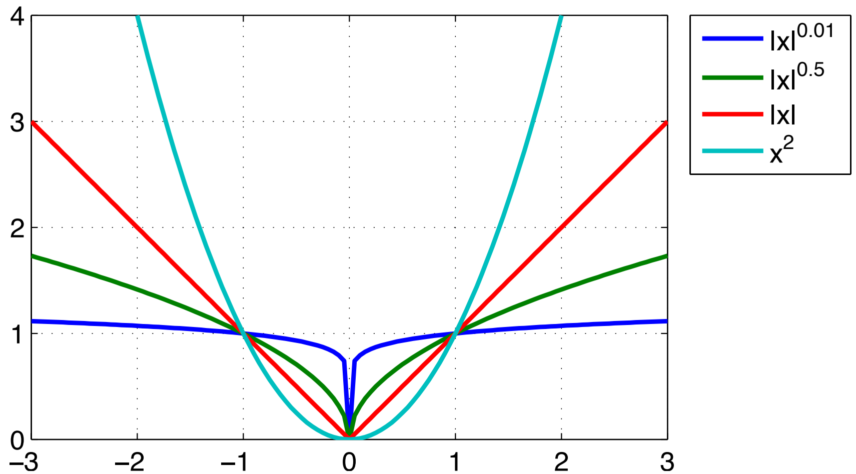
What is special about L_1?

- Induces sparsity at finite λ
  - because of the discontinuity of the gradient at the origin
- Convexity
  - L_1 norm is the tightest convex relaxation (with respect to the L∞ norm)



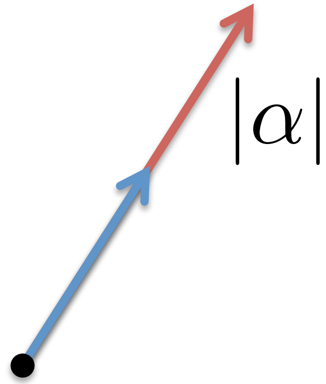
$$\|\boldsymbol{w}\|_q^q = \sum_{j=1}^{p} |w_j|^q$$

$$\xrightarrow{q \to 0} \#\{w_j : |w_j| > 0\}$$



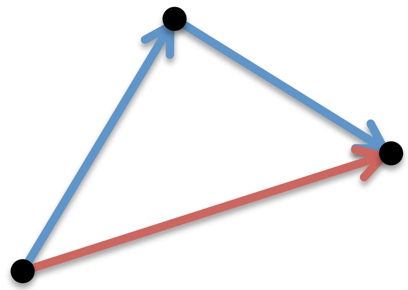
What is a norm?

- Positive homogenous



$$\|\alpha \boldsymbol{x}\| = |\alpha| \cdot \|\boldsymbol{x}\| \quad (\text{for any} \alpha \in \mathbb{R})$$

- Triangle inequality



$$\|\boldsymbol{x} + \boldsymbol{y}\| \leq \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$$

- Zero means zero

$$\|\boldsymbol{x}\| = 0 \quad \Rightarrow \quad \boldsymbol{x} = 0$$

Various norms

Euclidian (L2 norm)

$$\|\boldsymbol{w}\|_2 = \sqrt{\sum\nolimits_{j=1}^{p} w_j^2}$$

L1 norm

$$\|\boldsymbol{w}\|_1 = \sum\nolimits_{j=1}^{p} |w_j|$$

Infinity norm

$$\|\boldsymbol{w}\|_\infty = \max_{j=1,\ldots,p} |w_j|$$

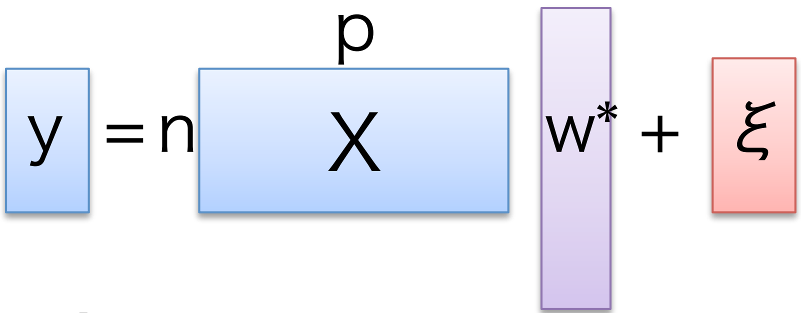
Setup

- Assume the same generative model

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w}^* + \boldsymbol{\xi}$$

$\boldsymbol{w}^*$: truth (*k* sparse)

$\boldsymbol{\xi}$: noise



- Estimator

$$\hat{\boldsymbol{w}} = \operatorname*{argmin}_{\boldsymbol{w}} \left(\frac{1}{2n} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 + \lambda_n \|\boldsymbol{w}\|_1 \right)$$

Theorem (we prove at the end)

There are constants c_1, c_2 such that

$$\|\hat{\boldsymbol{w}} - \boldsymbol{w}^*\|_2^2 \leq c_2 \sigma^2 \frac{k \log p}{n}$$

holds with high probability if

$$n \geq c_1 k \log p \quad \text{and} \quad \lambda_n = 4\sigma R \sqrt{\frac{\log p}{n}}$$

Condition for
the sample size *n*:
- Depends on the sparsity *k*
- Independent of the noise σ

Condition for
the reg. parameter λ_n:
- Independent of the sparsity *k*
- Deepends on the noise σ

A starting point

- $\hat{\boldsymbol{w}}$ minimizes the training objective

$$\frac{1}{2n}\|\boldsymbol{y}-\boldsymbol{X}\hat{\boldsymbol{w}}\|_2^2+\lambda_n\|\hat{\boldsymbol{w}}\|_1 \le \frac{1}{2n}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}^*\|_2^2+\lambda_n\|\boldsymbol{w}^*\|_1$$

Estimated

Truth

- After some manipulations, this implies

$$\frac{1}{2n}\|\boldsymbol{X}(\hat{\boldsymbol{w}}-\boldsymbol{w}^*)\|_2^2 \le \left(\underbrace{\left\|\boldsymbol{X}^\top\boldsymbol{\xi}/n\right\|_\infty}_{\text{Infinity norm } \|\boldsymbol{z}\|_\infty := \max_j |z_j|}+\lambda_n\right)\|\hat{\boldsymbol{w}}-\boldsymbol{w}^*\|_1$$

Proof

Substitute $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w}^* + \boldsymbol{\xi}$ to get

$$\frac{1}{2n}\|\boldsymbol{X}(\boldsymbol{w}^* - \hat{\boldsymbol{w}}) + \boldsymbol{\xi}\|_2^2 + \lambda_n\|\hat{\boldsymbol{w}}\|_1 \leq \frac{1}{2n}\|\boldsymbol{\xi}\|_2^2 + \lambda_n\|\boldsymbol{w}^*\|_1$$

which leads to

$$\frac{1}{2n}\|\boldsymbol{X}(\hat{\boldsymbol{w}} - \boldsymbol{w}^*)\|_2^2 \leq \|\boldsymbol{X}^\top\boldsymbol{\xi}/n\|_\infty\|\hat{\boldsymbol{w}} - \boldsymbol{w}^*\|_1 + \lambda_n(\|\boldsymbol{w}^*\|_1 - \|\hat{\boldsymbol{w}}\|_1)$$

Building blocks:

- Hölder's inequality $\boldsymbol{x}^\top\boldsymbol{y} \leq \|\boldsymbol{x}\|_1\|\boldsymbol{y}\|_\infty$
- Triangle inequality

A closer look

$$\frac{1}{2n}\|\boldsymbol{X}(\hat{\boldsymbol{w}}-\boldsymbol{w}^*)\|_2^2 \leq \left(\|\boldsymbol{X}^\top\boldsymbol{\xi}/n\|_\infty + \lambda_n\right)\|\hat{\boldsymbol{w}}-\boldsymbol{w}^*\|_1$$

Can be bounded as

$$\geq c\|\hat{\boldsymbol{w}}-\boldsymbol{w}^*\|_2^2$$

(explained later)

Can be bounded as

$$\leq 4\sqrt{k}\|\hat{\boldsymbol{w}}-\boldsymbol{w}^*\|_2$$

(explained later)

$$\|\hat{\boldsymbol{w}}-\boldsymbol{w}^*\|_2 \leq \left(\|\boldsymbol{X}^\top\boldsymbol{\xi}/n\|_\infty + \lambda_n\right)\frac{4\sqrt{k}}{c}$$

A closer look

$$\|\hat{\boldsymbol{w}} - \boldsymbol{w}^*\|_2 \leq \left(\| \boldsymbol{X}^\top \boldsymbol{\xi} / n\|_\infty + \lambda_n\right) \frac{4\sqrt{k}}{c}$$

How do we choose the regularization parameter?

- Choosing λ too large $\Rightarrow$ Meaningless bound
- Choosing λ too small $\Rightarrow$ noise term $\| \boldsymbol{X}^\top \boldsymbol{\xi} / n\|_\infty$ will dominate the RHS

Choose $\lambda_n \geq 2 \| \boldsymbol{X}^\top \boldsymbol{\xi} / n\|_\infty$ (why 2? – later)

The consequence

$$\|\hat{\boldsymbol{w}} - \boldsymbol{w}^*\|_2 \leq \frac{6\sqrt{k}\lambda_n}{c}$$

[What we wanted to have:

$$\|\hat{\boldsymbol{w}} - \boldsymbol{w}^*\|_2 \leq c_2\sigma\sqrt{\frac{k\log p}{n}}$$

]

Next step: how do we evaluate $\|\boldsymbol{X}^\top\boldsymbol{\xi}/n\|_\infty$?

⇨ Time for probability theory

Lemma: tail probability of max

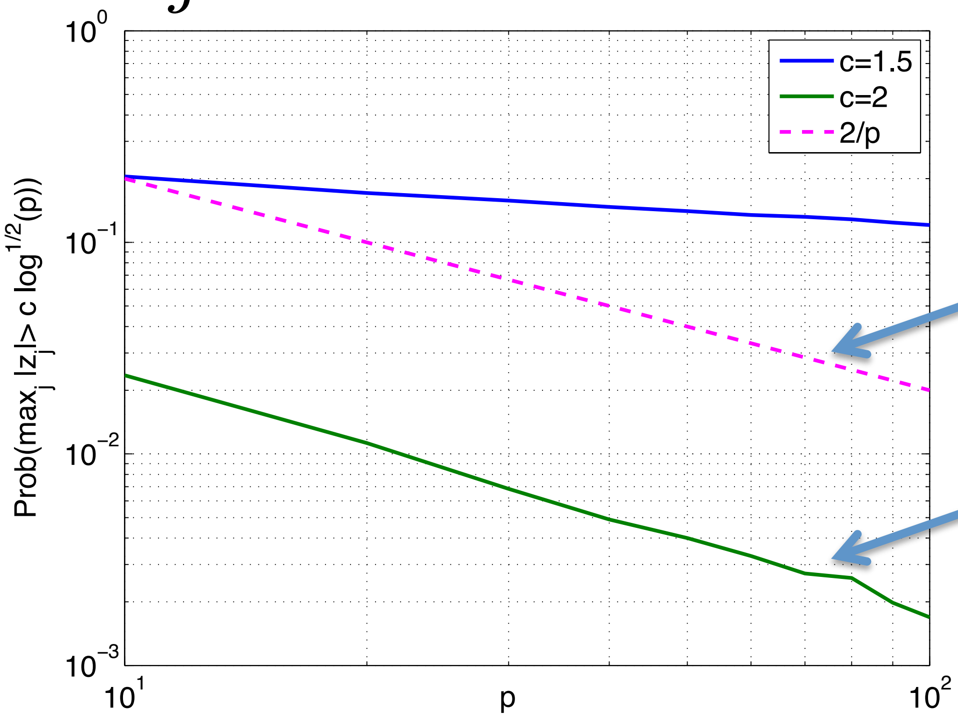
Try `exp_gaussian_max_tail.m`

Gaussian random variables

$$z_j \sim \mathcal{N}(0, \sigma_j^2) \quad (j = 1, \ldots, p)$$

Then $$\Pr(\max_j |z_j| > 2R\sqrt{\log p}) \leq \frac{2}{p}$$

$$R := \max_j \sigma_j$$



Upper bound

Simulation
(100k random
samples)

How large can $\|X^\top \xi / n\|_\infty$ be?

If ξ_i is Gaussian $\xi_i \sim \mathcal{N}(0, \sigma^2)$, we have:

$$\|X^\top \xi / n\|_\infty \leq 2\sigma R \sqrt{\frac{\log p}{n}}$$

where $R := \max_j \frac{\|x_j\|}{\sqrt{n}}$

with prob. greater than 1-2/p (high prob!)

Building blocks:

- Rewrite $\|X^\top \xi\|_\infty = \max_{j=1,\ldots,p} |z_j|, \quad z_j = \sum_{i=1}^{n} x_{ij}\xi_i$
- If ξ_i is Gaussian, z_j is also Gaussian

Summary so far

Choose

$$\lambda_n \geq 4\sigma R\sqrt{\frac{\log p}{n}}$$

Then

$$\|\hat{\boldsymbol{w}} - \boldsymbol{w}^*\|_2 \leq c_2\sigma\sqrt{\frac{k\log p}{n}}$$

with probability at least $1 - \frac{2}{p}$

(probability with respect to the noise ξ)

Two assumptions we used

- Right hand side (easier)

$$\|\hat{\boldsymbol{w}} - \boldsymbol{w}^*\|_1 \leq 4\sqrt{k}\|\hat{\boldsymbol{w}} - \boldsymbol{w}^*\|_2$$

- Left hand side (hard):

$$c\|\hat{\boldsymbol{w}} - \boldsymbol{w}^*\|_2^2 \leq \frac{1}{2n}\|\boldsymbol{X}(\boldsymbol{w}^* - \hat{\boldsymbol{w}})\|_2^2$$

Proof of the right hand side

$$\|\hat{\boldsymbol{w}} - \boldsymbol{w}^*\|_1 \leq 4\sqrt{k}\|\hat{\boldsymbol{w}} - \boldsymbol{w}^*\|_2$$

Compatibility of norms

Fact: for a *k*-sparse vector (exercise)

$$\|\boldsymbol{w}\|_1 \leq \sqrt{k}\|\boldsymbol{w}\|_2$$

(Use $\boldsymbol{x}^\top \boldsymbol{y} \leq \|\boldsymbol{x}\|_2\|\boldsymbol{y}\|_2$)

But $\boldsymbol{\Delta} := \hat{\boldsymbol{w}} - \boldsymbol{w}^*$ is not *k*-sparse.

Decompose it into sparse and non-sparse parts

$$\boldsymbol{\Delta} = \boldsymbol{\Delta}' + \boldsymbol{\Delta}''$$

Decomposability of L_1-norm

L_1 error

$$\|\boldsymbol{\Delta}\|_1 = \|\boldsymbol{\Delta}'\|_1 + \|\boldsymbol{\Delta}''\|_1$$

$$\boldsymbol{\Delta} = \boldsymbol{\Delta}' + \boldsymbol{\Delta}''$$

Sparse part

Non-sparse part

For example

$$\begin{array}{ccccc} \boldsymbol{\Delta} & & \boldsymbol{\Delta}' & & \boldsymbol{\Delta}'' \\ \begin{bmatrix} 1 \\ 3 \\ 1 \\ 0 \\ -2 \\ -1 \end{bmatrix} & = & \begin{bmatrix} 0 \\ 3 \\ 0 \\ 0 \\ 0 \\ -1 \end{bmatrix} & + & \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ -2 \\ 0 \end{bmatrix} \end{array}$$

correct support

$$\mathrm{supp}(\boldsymbol{\Delta}') \subseteq \mathrm{supp}(\boldsymbol{w}^*)$$

$$\mathrm{supp}(\boldsymbol{\Delta}'') \cap \mathrm{supp}(\boldsymbol{w}^*) = \emptyset$$

Bounding the non-sparse part

Triangular inequality

$$\|\boldsymbol{w}^*\|_1 - \|\hat{\boldsymbol{w}}\|_1 \leq \|\hat{\boldsymbol{w}} - \boldsymbol{w}^*\|_1$$
$$(= \|\boldsymbol{\Delta}'\|_1 + \|\boldsymbol{\Delta}''\|_1)$$

Using the decomposability

$$\|\boldsymbol{w}^*\|_1 - \|\hat{\boldsymbol{w}}\|_1 \leq \|\boldsymbol{\Delta}'\|_1 - \|\boldsymbol{\Delta}''\|_1$$

This one is much tighter!

Bounding the non-sparse part

Using the better bound, we get

$$\|\hat{\boldsymbol{w}} - \boldsymbol{w}^*\|_1 \leq 4\sqrt{k}\|\hat{\boldsymbol{w}} - \boldsymbol{w}^*\|_2$$

Building blocks:

- Positivity of a norm $0 \leq \|\boldsymbol{X}(\hat{\boldsymbol{w}} - \boldsymbol{w}^*)\|_2^2$
- Choice of regularization param. $\lambda_n \geq 2\|\boldsymbol{X}^\top \boldsymbol{\xi}/n\|_\infty$
- The bound $\|\boldsymbol{w}^*\|_1 - \|\hat{\boldsymbol{w}}\|_1 \leq \|\boldsymbol{\Delta}'\|_1 - \|\boldsymbol{\Delta}''\|_1$

End of proof.

Proof of the left hand side

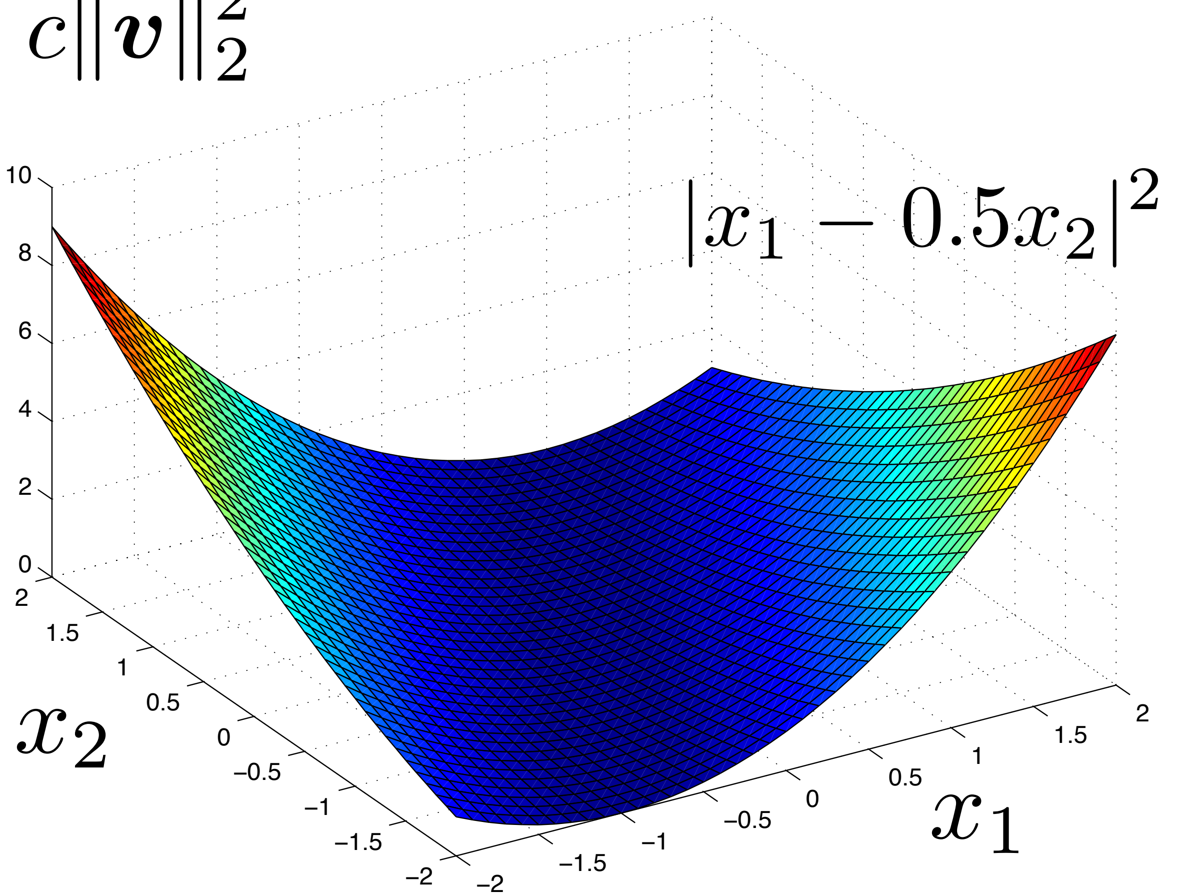
$$c\|\hat{\boldsymbol{w}} - \boldsymbol{w}^*\|_2^2 \leq \frac{1}{2n}\|\boldsymbol{X}(\boldsymbol{w}^* - \hat{\boldsymbol{w}})\|_2^2$$

Lack of strong convexity

- When n < p, we cannot have

$$\frac{1}{2n}\|\boldsymbol{X}\boldsymbol{v}\|_2^2 \geq c\|\boldsymbol{v}\|_2^2$$

in general.



Restricted strong convexity

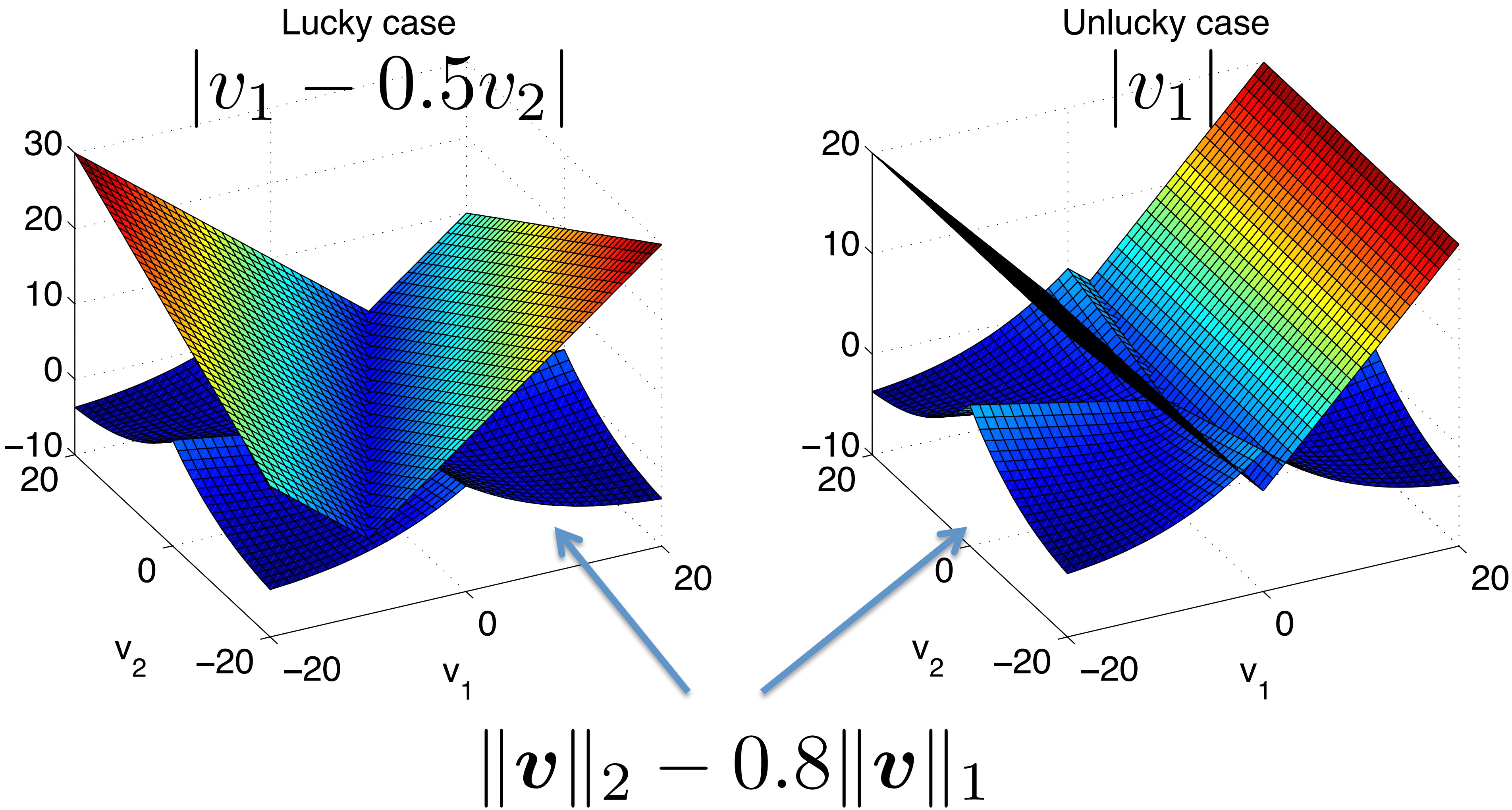
- However we can have

$$\frac{1}{\sqrt{n}}\|\boldsymbol{X}\boldsymbol{v}\|_2 \geq \frac{1}{4}\|\boldsymbol{v}\|_2 - 9\sqrt{\frac{\log p}{n}}\|\boldsymbol{v}\|_1$$

with high probability, when the rows of X are sampled independently from the standard Gaussian distribution.

Note that this is a simplified version of [Raskutti, Wainwright, Yu (2010)]. For correlated X, see the original paper.

Visualizing restricted strong convexity (n=1 and p=2)



Taking sparsity into account

If $n \geq c_1 k \log p$ there is c≥0 s.t.

$$\frac{1}{\sqrt{n}} \|\boldsymbol{X}(\hat{\boldsymbol{w}} - \boldsymbol{w}^*)\|_2 \geq c\|\hat{\boldsymbol{w}} - \boldsymbol{w}^*\|_2$$

where $$c = \frac{1}{4} - \frac{36}{\sqrt{c_1}}$$

Building blocks:

- Use $\|\hat{\boldsymbol{w}} - \boldsymbol{w}^*\|_1 \leq 4\sqrt{k}\|\hat{\boldsymbol{w}} - \boldsymbol{w}^*\|_2$

End of proof.

Theorem (shown again)

There are constants c_1 and c_2 such that

$$\|\hat{\boldsymbol{w}} - \boldsymbol{w}^*\|_2^2 \leq c_2 \sigma^2 \frac{k \log p}{n}$$

holds with high probability, if

$$n \geq c_1 k \log p \quad \text{and} \quad \lambda_n = 4\sigma R \sqrt{\frac{\log p}{n}}$$

Condition for
the sample size n:
- comes from the restricted strong convexity (LHS)

Condition for
the reg. parameter λ_n:
- comes from bounding the noise term $\lambda_n \geq 2\|\boldsymbol{X}^\top \boldsymbol{\xi}/n\|_\infty$

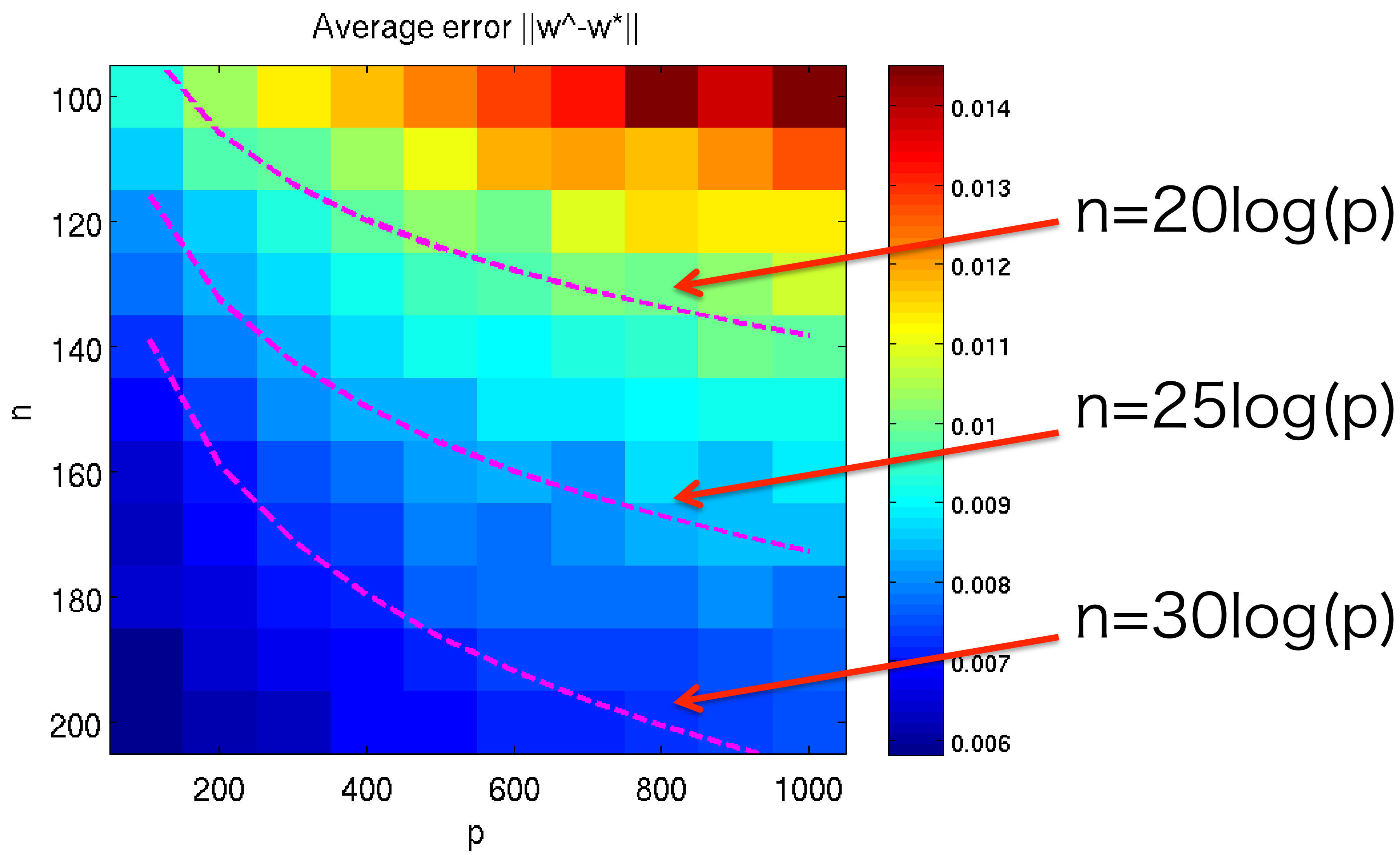
Implications of the bound

- The number of samples we need to achieve certain error is roughly k log(p)
  - Where does the log(p) come from? Max of p Gaussian random variables
  - Why log(p)? because L_∞ norm is dual to L_1 norm
- If *n* is too small, lasso may not work (independent of the noise σ^2)

Simulation

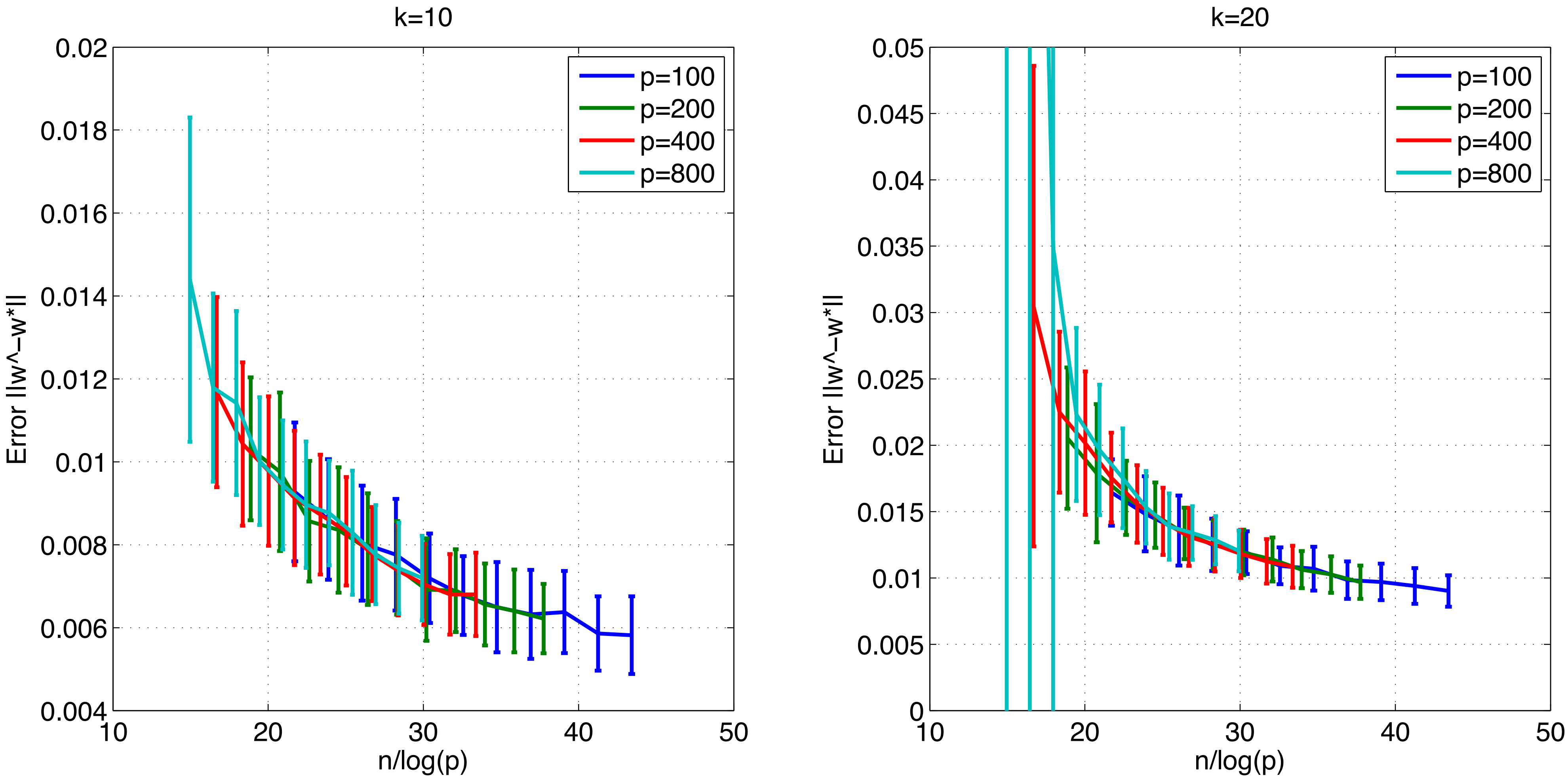
σ=0.01, λ_n=σ sqrt(log(p)/n)

Try `exp_lasso_scaling.m`

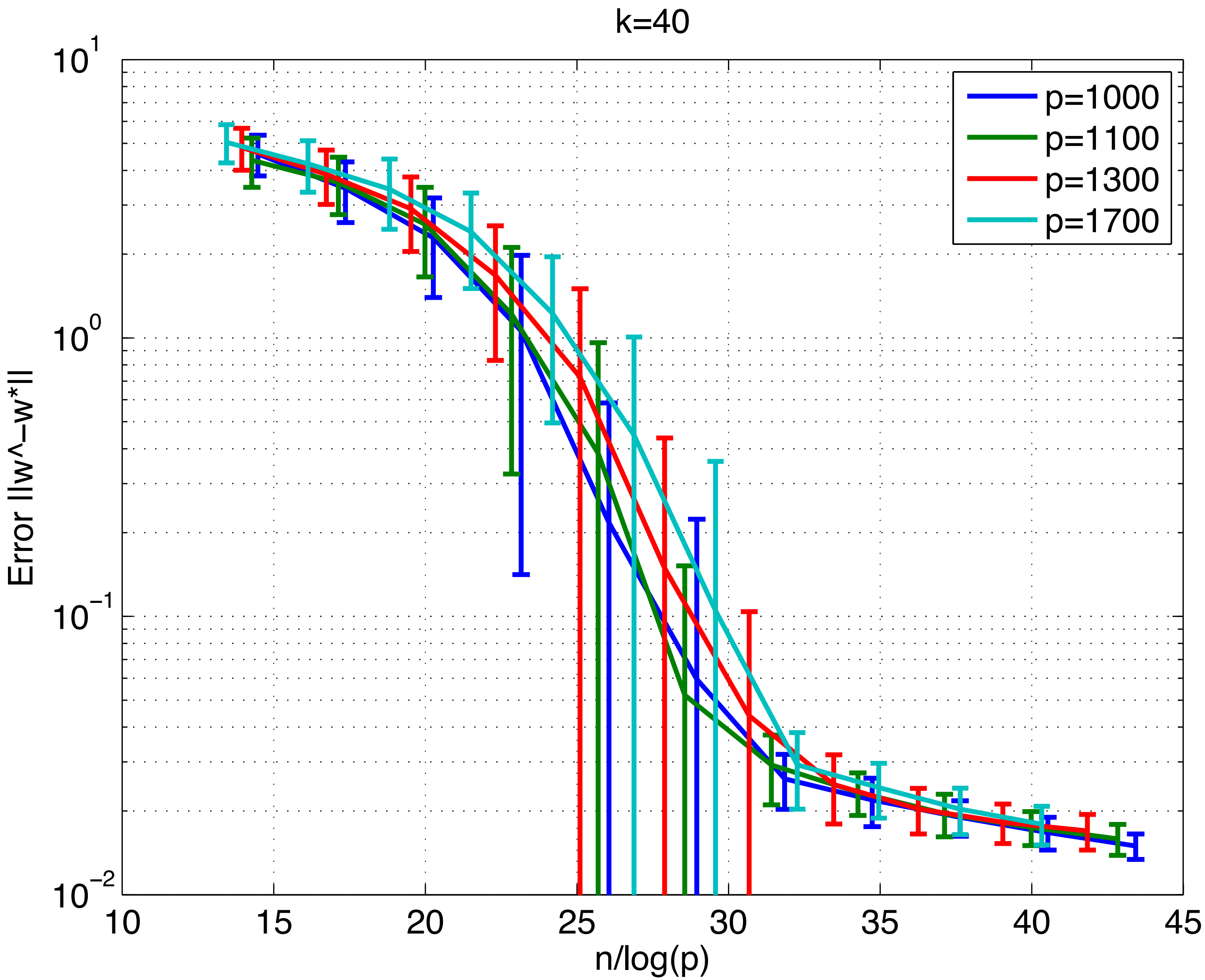


Rescaled

$\sigma = 0.01,\ \lambda_n = \sigma\,\mathrm{sqrt}(\log(p)/n)$



Phase transition!



Conclusion

- Theory lets you understand precisely when the model behaves nicely and when it doesn’t
  - it is (ideally) agnostic to your philosophy (Bayesian or not).
  - can predict the empirical behavior quantitatively and qualitatively.
  - It is doable (and fun).

Bibliography

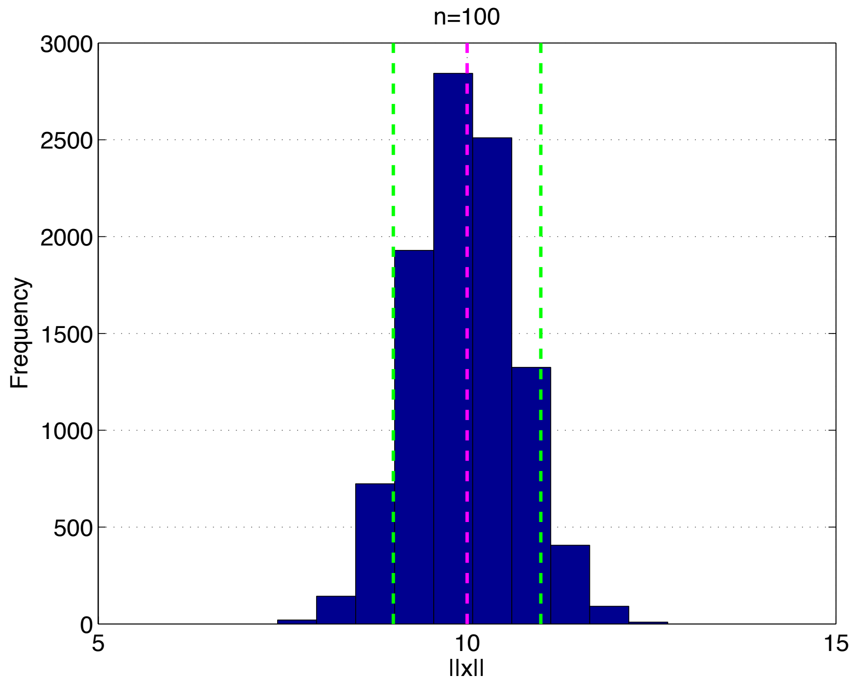
- Bias-variance tradeoff
  - Chapter 7 in Hastie, Tibshirani, & Friedman (2009) "Elements of Statistical Learning."
- Phase transition in ridge regression and SVM
  - Opper & Kinzel (1995) "Statistical Mechanics of Generalization," in "Models of Neural Networks III", Springer.
- Analysis of lasso
  - Negahban et al. (2012) "A Unified Framework for High-Dimensional Analysis of M-Estimators with Decomposable Regularizers". Statistical Science.
  - Raskutti, Wainwright, Yu (2010) "Restricted Eigenvalue Properties for Correlated Gaussian Designs". JMLR.
  - Bühlmann & van de Geer (2011) "Statistics for High-Dimensional Data." Springer.

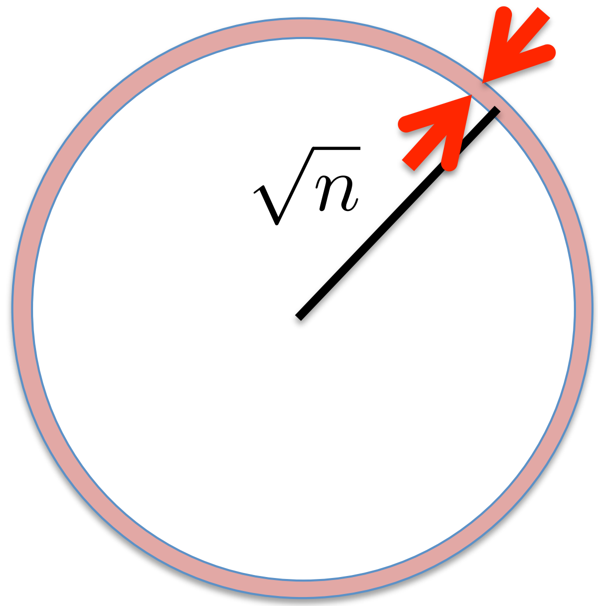
Blessing of dimensionality

Try `exp_concentration.m`

- Norm $\|\boldsymbol{x}\| = (\underbrace{x_1^2 + x_2^2 + \cdots + x_n^2}_{\sim n + \sqrt{n}\sigma\xi})^{1/2}$ (Central limit thm.)

  where $\mathbb{E}[x_i^2] = 1,\ \sigma^2 = \mathrm{Var}[x_i^2],\ \xi \sim \mathcal{N}(0,1)$





Gordon-Slepian (part I)

[Davidson & Szarek 2001]

- $(Y_t)_{t\in T}$, $(Z_t)_{t\in T}$, jointly Gaussian, mean zero for each t, and satisfies

$$\|Y_t - Y_{t'}\|_2 \leq \|Z_t - Z_{t'}\|_2 \quad \text{for } t, t' \in T.$$

Then,

$$\mathbb{E}\max_{t\in T} Y_t \leq \mathbb{E}\max_{t\in T} Z_t$$

GS Lemma for max singular value

Let

$$Y_{(\boldsymbol{u},\boldsymbol{v})} = \boldsymbol{u}^\top \boldsymbol{X} \boldsymbol{v}, \quad Z_{(\boldsymbol{u},\boldsymbol{v})} = \boldsymbol{u}^\top \boldsymbol{g}_1 + \boldsymbol{v}^\top \boldsymbol{g}_2$$

Then,

$$\underbrace{\mathbb{E} \max_{\substack{\|\boldsymbol{u}\|_2 \le 1, \\ \|\boldsymbol{v}\|_2 \le 1}} \boldsymbol{u}^\top \boldsymbol{X} \boldsymbol{v}}_{= \mathbb{E} s_1(\boldsymbol{X})} \le \underbrace{\mathbb{E} \max_{\|\boldsymbol{u}\|_2 \le 1} \boldsymbol{u}^\top \boldsymbol{g}_1}_{= \sqrt{n}} + \underbrace{\mathbb{E} \max_{\|\boldsymbol{v}\|_2 \le 1} \boldsymbol{v}^\top \boldsymbol{g}_2}_{= \sqrt{p}}$$

(for large enough n and p)

Gordon-Slepian (part II)

[Davidson & Szarek 2001]

- $(Y_{(s,t)})_{s\in S, t\in T}$, $(Z_{(s,t)})_{s\in S, t\in T}$, jointly Gaussian, mean zero for each t, and satisfies

$$\text{(i)} \quad \|Y_{(s,t)} - Y_{(s',t')}\|_2 \leq \|Z_{(s,t)} - Z_{(s',t')}\|_2 \quad \text{if } s \neq s'$$

$$\text{(ii)} \quad \|Y_{(s,t)} - Y_{(s,t')}\|_2 \geq \|Z_{(s,t)} - Z_{(s,t')}\|_2 \quad \text{for some } s$$

Then,

$$\mathbb{E} \max_{s\in S} \min_{t\in T} Y_{(s,t)} \leq \mathbb{E} \max_{s\in S} \min_{t\in T} Z_{(s,t)}$$

GS Lemma for min singular value

Let

$$Y_{(\boldsymbol{u},\boldsymbol{v})} = \boldsymbol{u}^\top \boldsymbol{X} \boldsymbol{v}, \quad Z_{(\boldsymbol{u},\boldsymbol{v})} = \boldsymbol{u}^\top \boldsymbol{g}_1 + \boldsymbol{v}^\top \boldsymbol{g}_2$$

Then for n≤p,

$$\underbrace{\mathbb{E} \max_{\|\boldsymbol{u}\|_2 \le 1} \min_{\|\boldsymbol{v}\|_2 \le 1} \boldsymbol{u}^\top \boldsymbol{X} \boldsymbol{v}}_{= -\mathbb{E} s_n(\boldsymbol{X})} \le \underbrace{\mathbb{E} \max_{\|\boldsymbol{u}\|_2 \le 1} \boldsymbol{u}^\top \boldsymbol{g}_1}_{= \sqrt{n}} + \underbrace{\mathbb{E} \min_{\|\boldsymbol{v}\|_2 \le 1} \boldsymbol{v}^\top \boldsymbol{g}_2}_{= -\sqrt{p}}$$

(for large enough n and p)