

Simple Variable Selection

LASSO: Sparse Regression

Sparsity

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

- Vector w is sparse, if many entries are zero

• w

0	1	0	2	0
---	---	---	---	---

• x_i is sparse

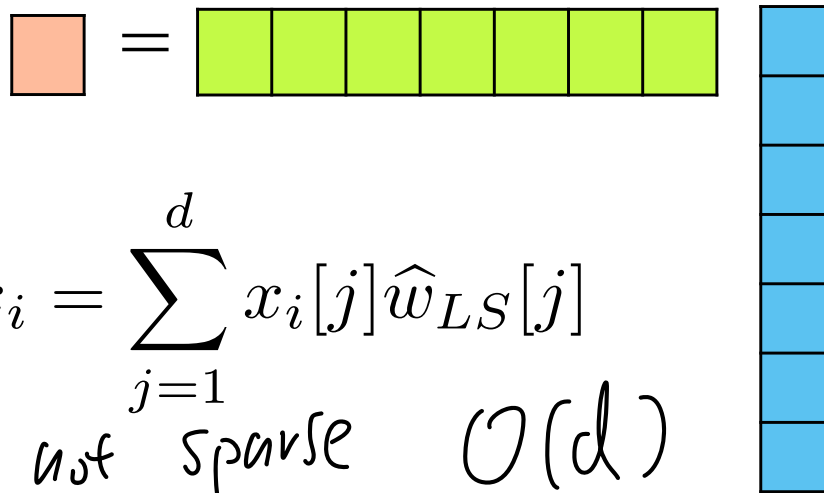
Sparsity

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

- Vector w is sparse, if many entries are zero

– Efficiency: If $\text{size}(w) = 100$ Billion, each prediction is expensive:

- If w is sparse, prediction computation only depends on number of non-zeros



$$\hat{y}_i = \hat{w}_{LS}^T x_i = \sum_{j=1}^d x_i[j] \hat{w}_{LS}[j]$$

\hat{w}_{LS} is not sparse $\mathcal{O}(d)$

if \hat{w}_{LS} is k -sparse:
 only k non-zero entries
 $\mathcal{O}(k)$
 if $k \ll d$

Statistical efficiency
 var: $\frac{d \sigma^2}{n} \rightarrow \frac{k \cdot \sigma^2}{n}$

Sparsity

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

- Vector w is sparse, if many entries are zero

- **Interpretability:** What are the relevant dimension to make a prediction?



w ;

$x =$

Lot size	—	Dishwasher
Single Family	—	Garbage disposal
Year built	—	Microwave
Last sold price	—	Range / Oven
Last sale price/sqft	—	Refrigerator
Finished sqft	—	Washer
Unfinished sqft	—	Dryer
Finished basement sqft	—	Laundry location
# floors	—	Heating type
Flooring types	—	Jetted Tub
Parking type	—	Deck
Parking amount	—	Fenced Yard
Cooling	—	Lawn
Heating	—	Garden
Exterior materials	—	Sprinkler System
Roof type	—	
Structure style	—	

- How do we find “best” subset among all possible?

Finding best subset: Exhaustive

> Try all subsets of size 1, 2, 3, ... and one that minimizes validation error

> Problem?

d features, index: $\{1, \dots, d\}$
for $i=1, \dots, d$
pick i indices

$$i=1, \binom{d}{1} + \binom{d}{2} + \dots +$$

→ approximation

Finding best subset: Greedy

Forward stepwise:

Starting from simple model and iteratively add features most useful to fit

goal:
find k -sparse
 \hat{w}

Forward Greedy T : selected indices from $\{1, \dots, d\}$

- 1: $T \leftarrow \emptyset$
- 2: **For** $j = 1, \dots, k$ **do**
- 3: $j^* \leftarrow \arg \min_{\ell} \min_w \sum_{i=1}^n \left(y_i - \sum_{j \in T \cup \{\ell\}} w[j] \times x_i[j] \right)^2$
 $\ell \in \{1, \dots, d\} \setminus T$
- 4: $T \leftarrow T \cup \{j^*\}$

$O(kd)$, time of regression
 $O(d^2)$

Backward stepwise:

Start with full model and iteratively remove features least useful to fit

$$T = \{1, \dots, d\}$$

Combining forward and backward steps:

In forward algorithm, insert steps to remove features no longer as important

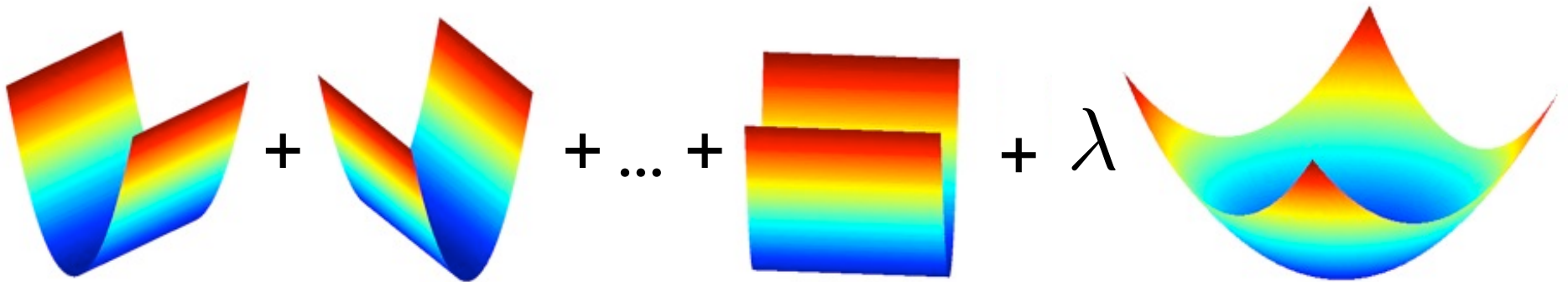
Lots of other variants, too.

Finding best subset: Regularize

Ridge regression makes coefficients small

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

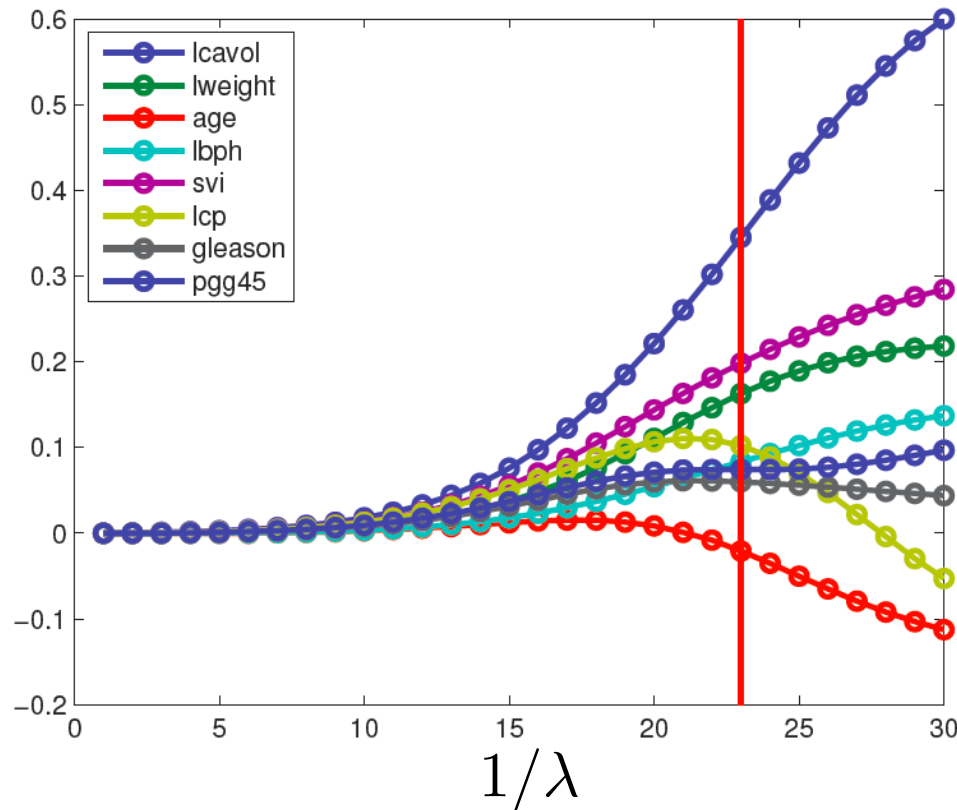
$\lambda \rightarrow \infty$
 $w \rightarrow 0$



Finding best subset: Regularize

Ridge regression makes coefficients small

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



From
Kevin Murphy
textbook

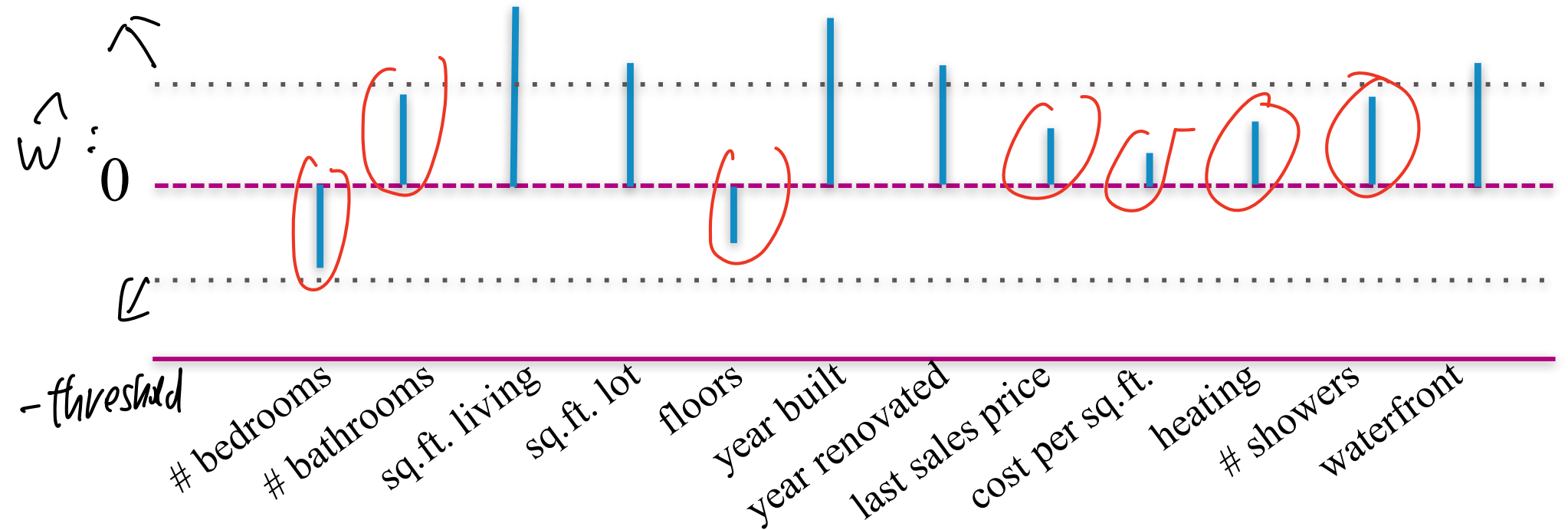
Thresholded Ridge Regression

set $\hat{w}_i \leftarrow 0$ if $|\hat{w}_i| \leq \text{threshold}$

hyperparameter
CV

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

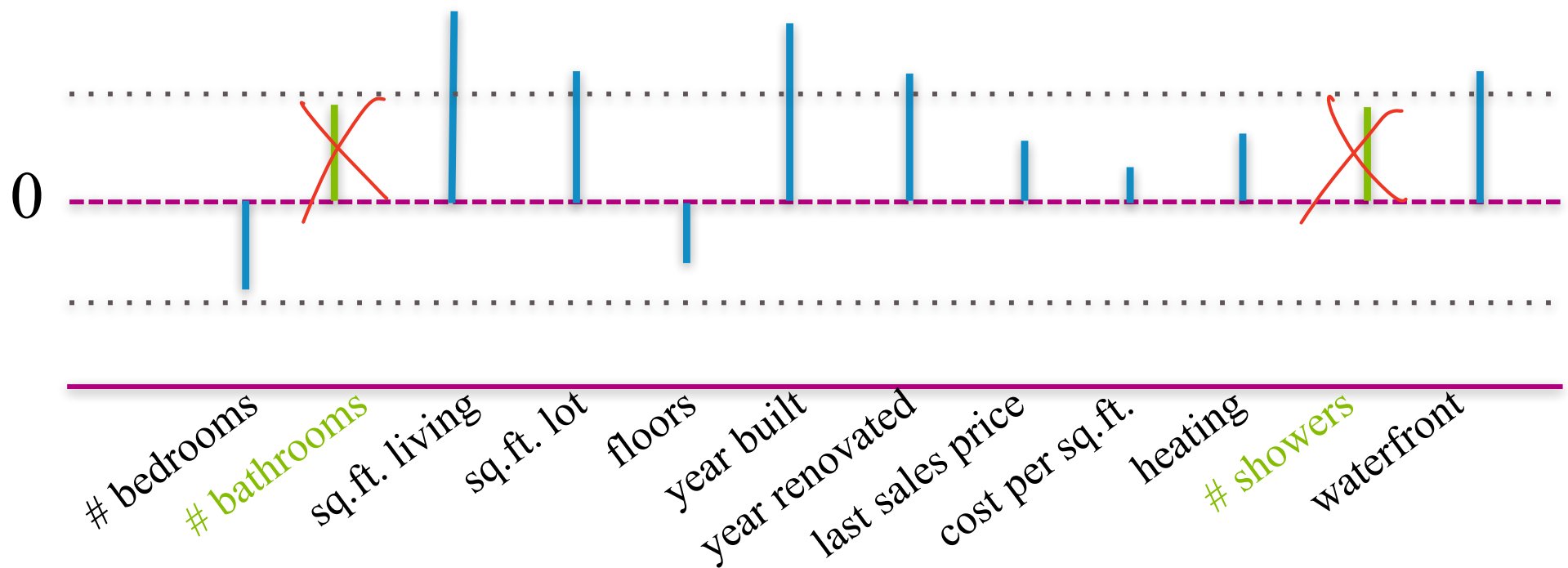
threshold Why don't we just set **small** ridge coefficients to 0?



Thresholded Ridge Regression

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

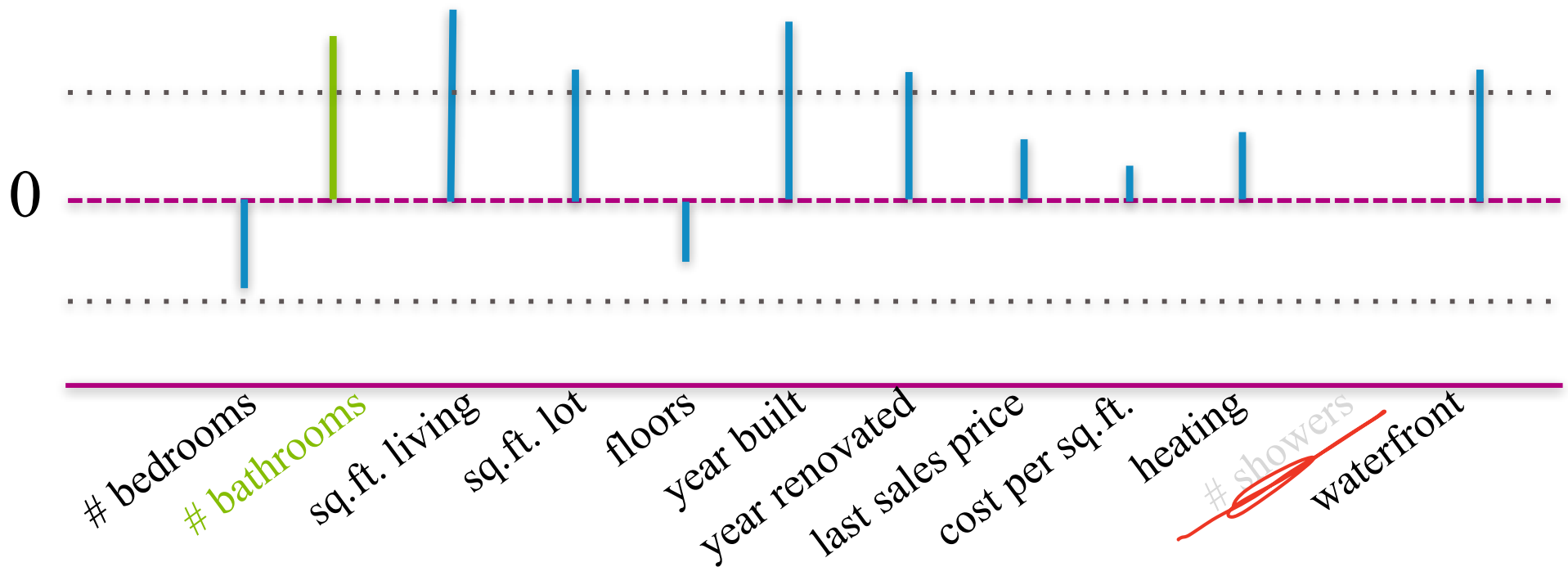
Consider two **related** features (bathrooms, showers)



Thresholded Ridge Regression

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

What if we **didn't** include showers? Weight on bathrooms increases!



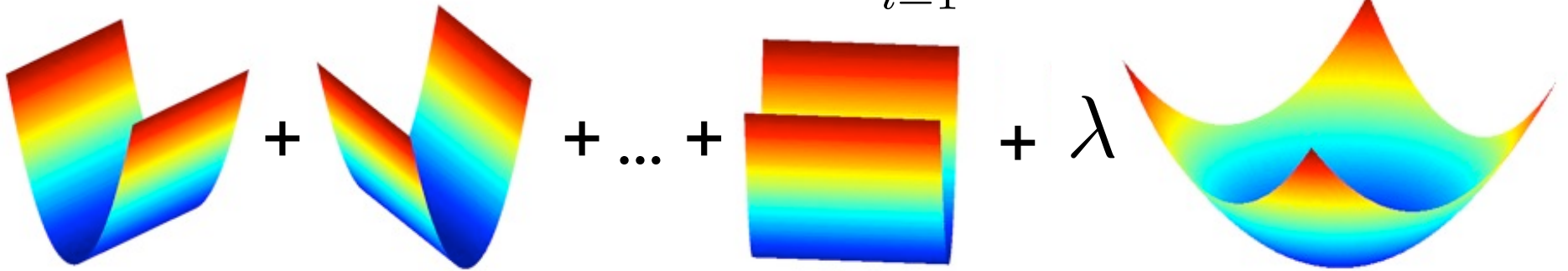
Can another regularizer perform selection automatically?

Recall Ridge Regression

$$\sqrt{\sum_{i=1}^d w_i^2}$$

- Ridge Regression objective:

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



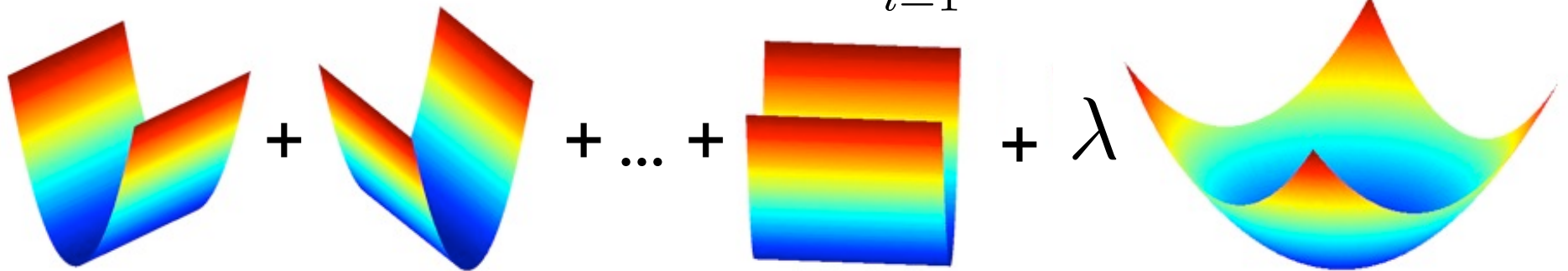
$$\|w\|_p = \left(\sum_{i=1}^d |w_i|^p \right)^{1/p}$$

Ridge vs. Lasso Regression

Least Absolute Shrinkage & Selection Operator

- Ridge Regression objective:

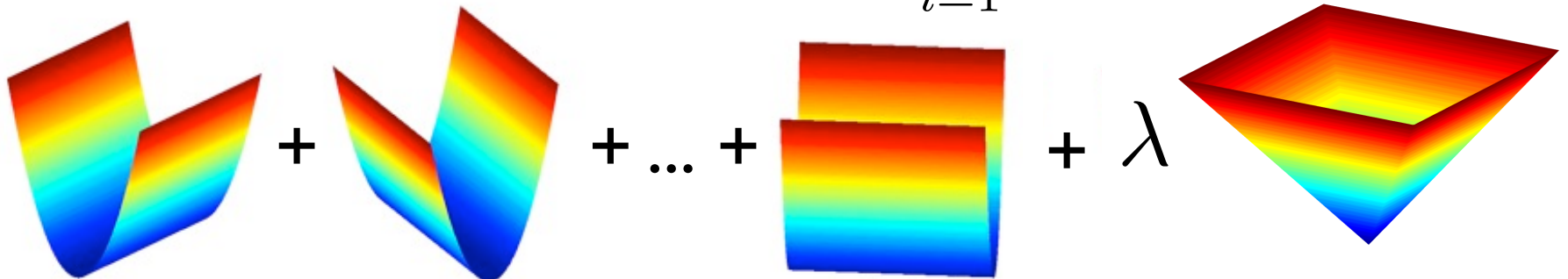
$$\hat{w}_{ridge} = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2 + \lambda \|w\|_2^2$$



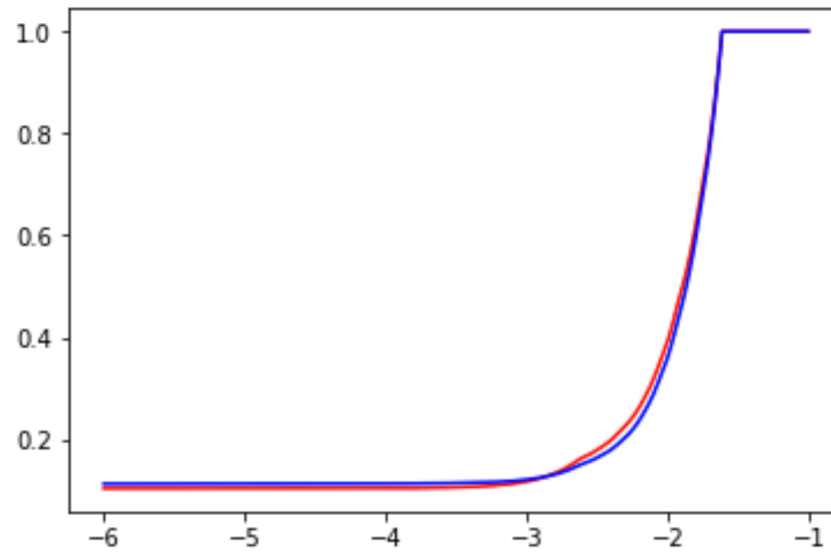
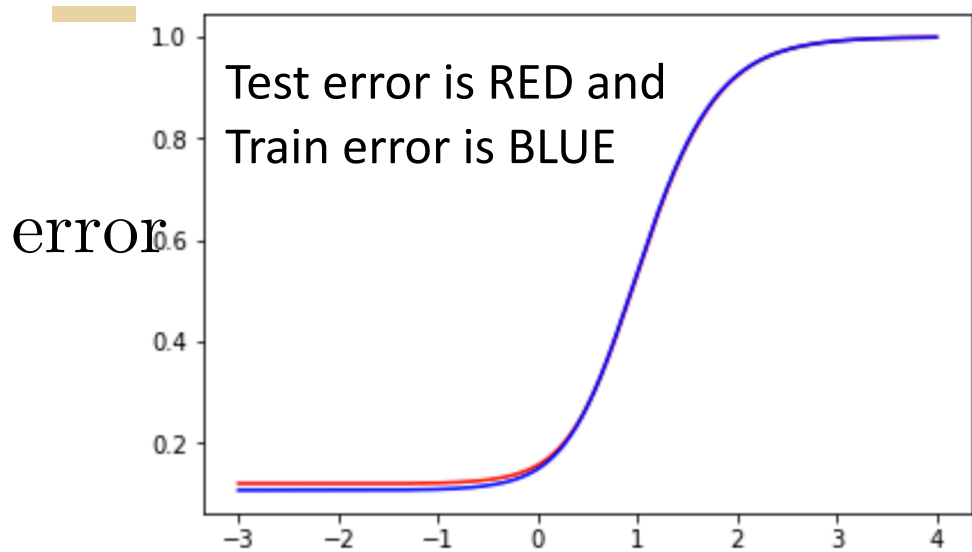
- Lasso objective:

$$\hat{w}_{lasso} = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2 + \lambda \|w\|_1$$

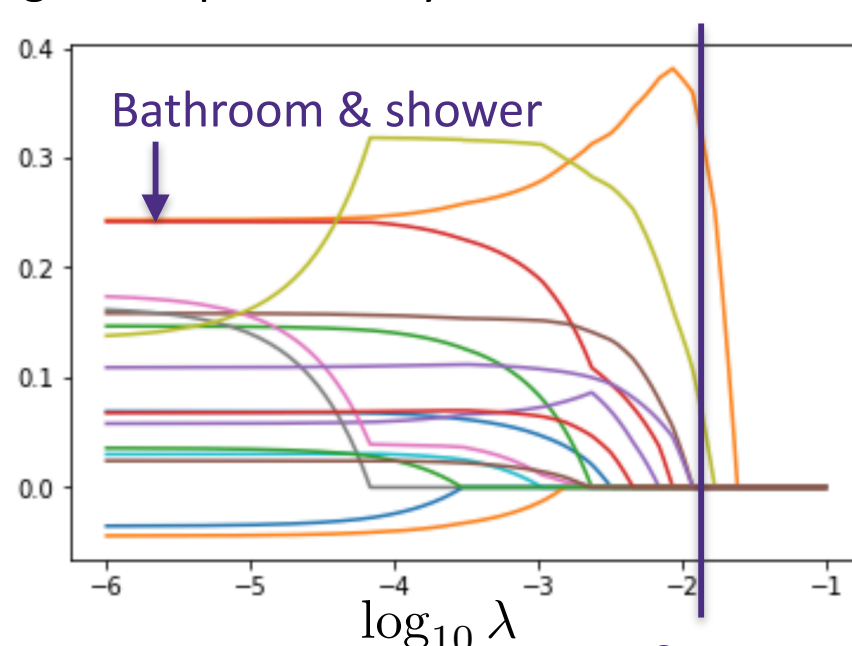
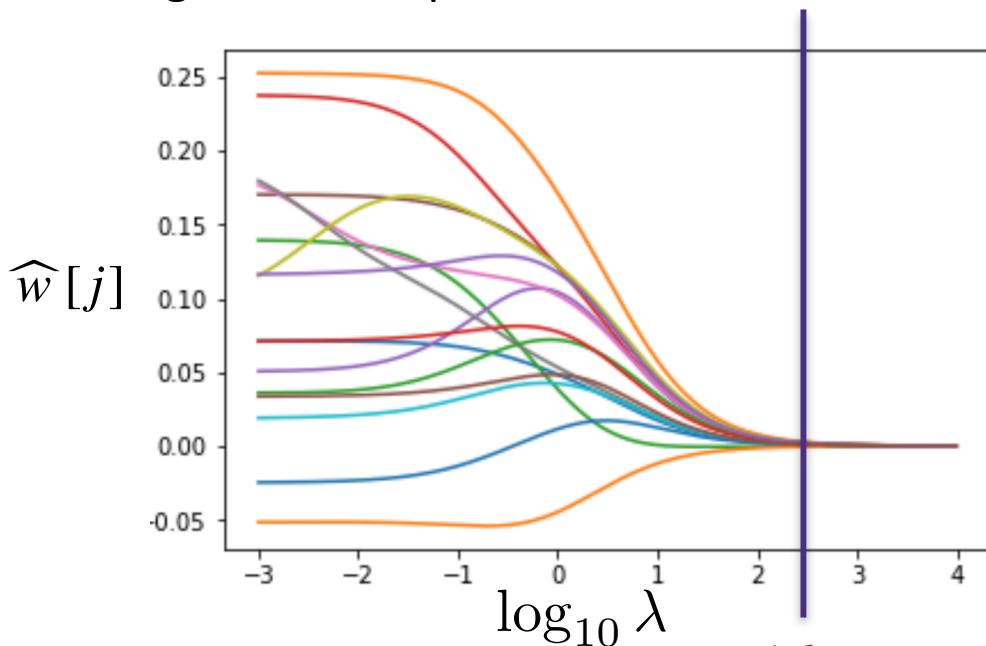
$\rho=1$
↳



Example: house price with 16 features



- Regularization path for Lasso shows that weights drop to exactly zero as λ increases



Ridge regression 16-sparse

Lasso regression 2-sparse

Lasso regression naturally gives sparse features

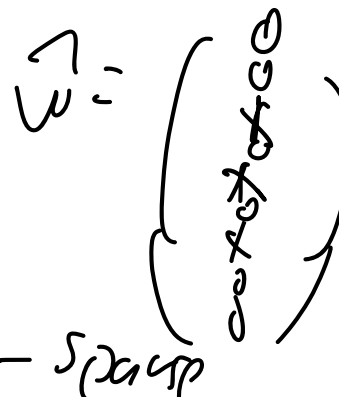
- feature selection with Lasso regression

1. Model selection: choose λ based on cross validation error
2. Feature selection: keep only those features with non-zero (or not-too-small) parameters in w at optimal λ $\rightarrow \hat{w} =$
3. retrain with the sparse model and $\lambda = 0$

why do we need to retrain?

$X \rightarrow X_{\text{selected}} \rightarrow \text{run regression on } X_{\text{selected}}$
dim: $d \rightarrow k$

k -sparse



Example: piecewise-linear fit

$$h_0(x) = 1$$

$$h_i(x) = [x + 1.1 - 0.1i]^+$$

- We use Lasso on the piece-wise linear example

$$\lambda \leftrightarrow R$$

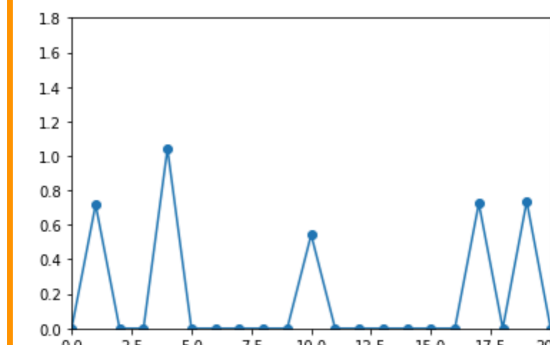
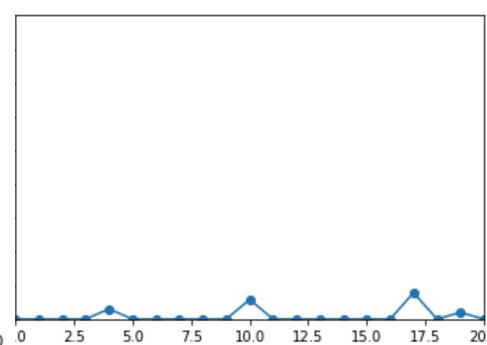
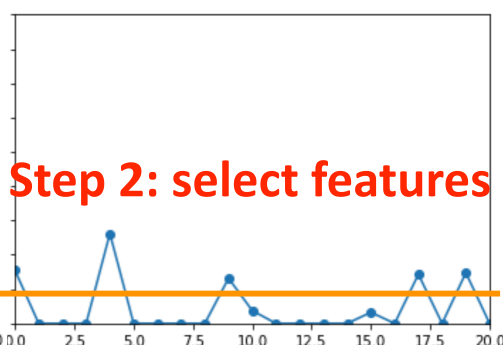
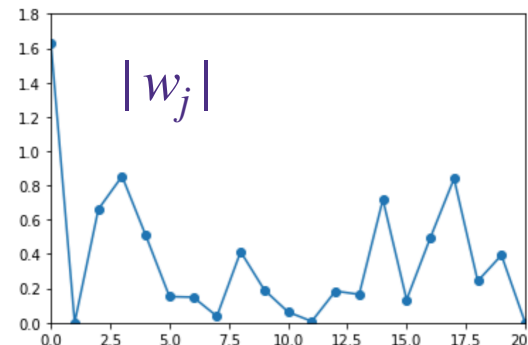
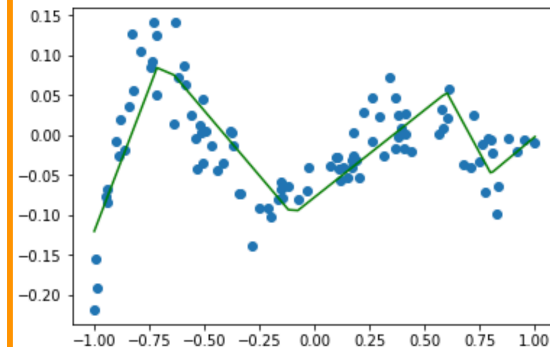
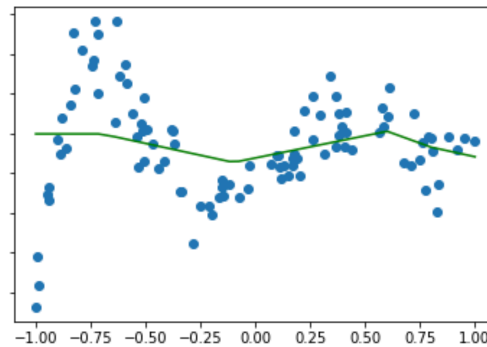
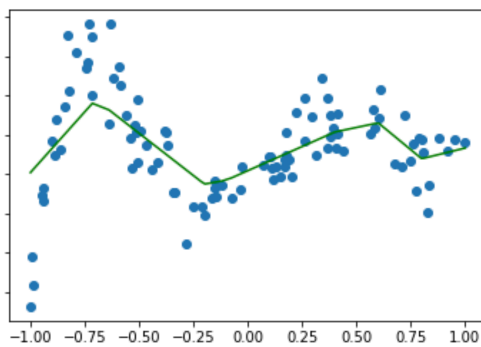
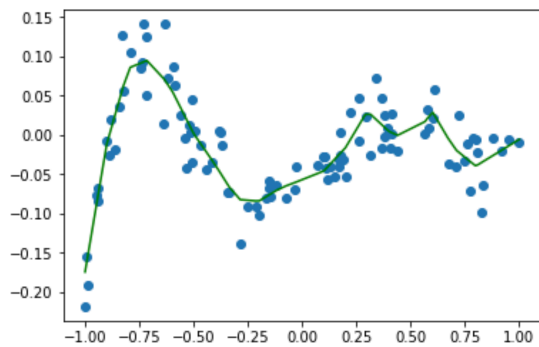
Step 1: find optimal λ^*

$$\text{minimize}_w \mathcal{L}(w) + \lambda \|w\|_1$$

least square loss

Step 3: retrain \hat{w}

$$\text{minimize}_w \mathcal{L}(w)$$



$$\lambda = 10^{-8}$$

$$\lambda = 10^{-4}$$

$$\lambda = 2 \times 10^{-4}$$

$$\lambda = 0$$

- de-biasing (via re-training) is critical!

but only use selected features

Penalized Least Squares

least squares

- Regularized optimization:

$$\hat{w}_r = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2 + \lambda r(w)$$

$$\text{Ridge : } r(w) = \|w\|_2^2$$

$$\text{Lasso : } r(w) = \|w\|_1$$

Penalized Least Squares

- Regularized optimization:

$$\hat{w}_r = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2 + \lambda r(w)$$

Ridge : $r(w) = \|w\|_2^2$

Lasso : $r(w) = \|w\|_1$

Theorem.

- For any $\lambda^* \geq 0$ for which \hat{w}_r achieves the minimum, there exists a $\mu^* \geq 0$ such that the solution of the constrained optimization, \hat{w}_c , is the same as the solution of the regularized optimization, \hat{w}_r , where

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

subject to $\|w\|_2^2 \leq \mu^*$

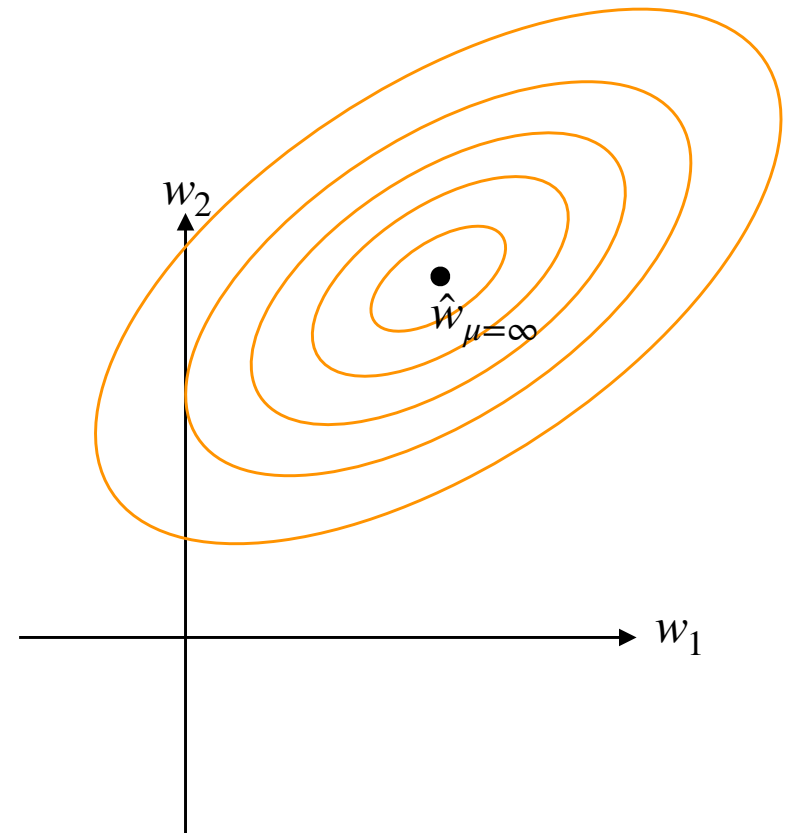
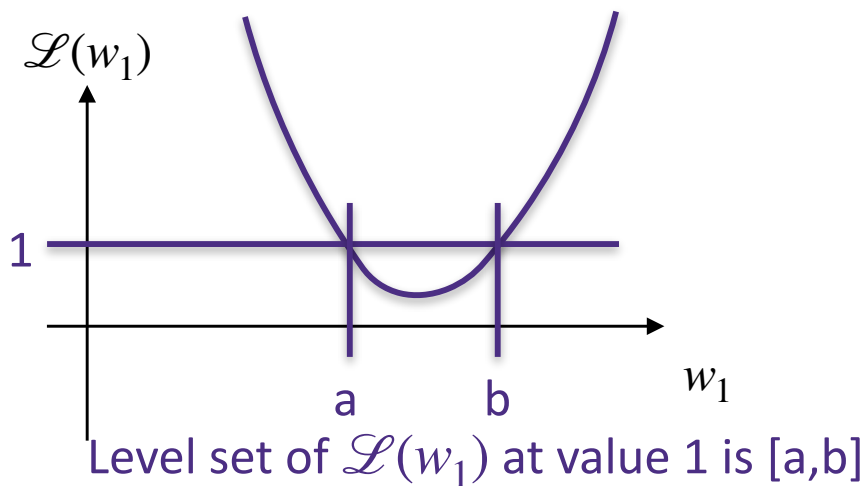
- so there are pairs of (λ, μ) whose optimal solution \hat{w}_r are the same for the regularized optimization and constrained optimization

Why does Lasso give sparse solutions?

$$\left[\begin{array}{l} \text{minimize}_w \sum_{i=1}^n (w^T x_i - y_i)^2 \\ \text{subject to } \|w\|_1 \leq \mu \end{array} \right.$$

- the level set of a function $\mathcal{L}(w_1, w_2)$ is defined as the set of points (w_1, w_2) that have the same function value
- the level set of a quadratic function is an oval
- the center of the oval is the least squares solution $\hat{w}_{\mu=\infty} = \hat{w}_{LS}$

1-D example with quadratic loss



Why does Lasso give sparse solutions?

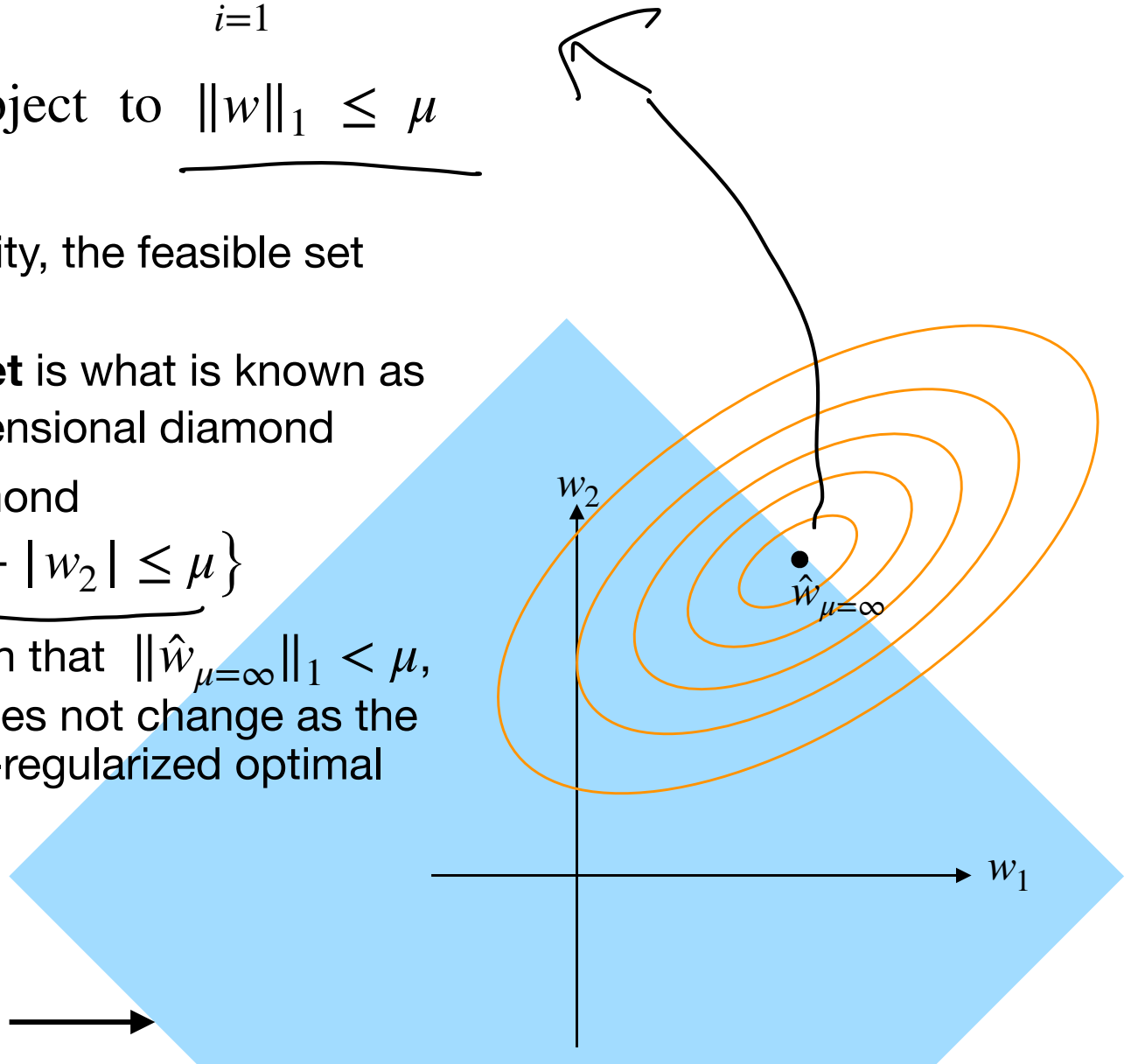
$$\text{minimize}_w \sum_{i=1}^n (w^T x_i - y_i)^2$$

$$\text{subject to } \|w\|_1 \leq \mu$$

- as we decrease μ from infinity, the feasible set becomes smaller
- the shape of the **feasible set** is what is known as L_1 ball, which is a high dimensional diamond
- In 2-dimensions, it is a diamond

$$\{(w_1, w_2) \mid |w_1| + |w_2| \leq \mu\}$$

- when μ is large enough such that $\|\hat{w}_{\mu=\infty}\|_1 < \mu$, then the optimal solution does not change as the feasible set includes the un-regularized optimal solution



feasible set: $\{w \in \mathbb{R}^2 \mid \|w\|_1 \leq \mu\}$ →

Why does Lasso give sparse solutions?

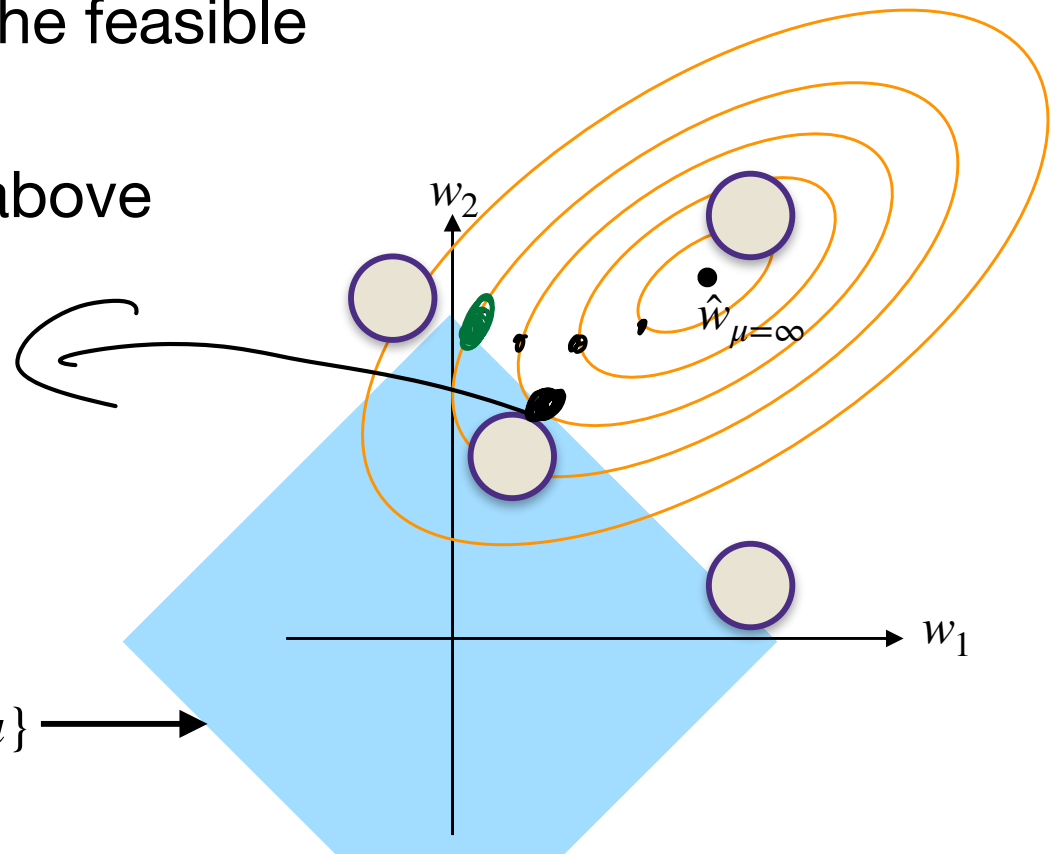
$$\text{minimize}_w \sum_{i=1}^n (w^T x_i - y_i)^2$$

$$\text{subject to } \|w\|_1 \leq \mu$$

- As μ decreases (which is equivalent to increasing regularization λ) the feasible set (blue diamond) shrinks
- The optimal solution of the above optimization is ?

solution

feasible set: $\{w \in \mathbb{R}^2 \mid \|w\|_1 \leq \mu\}$ →



Why does Lasso give sparse solutions?

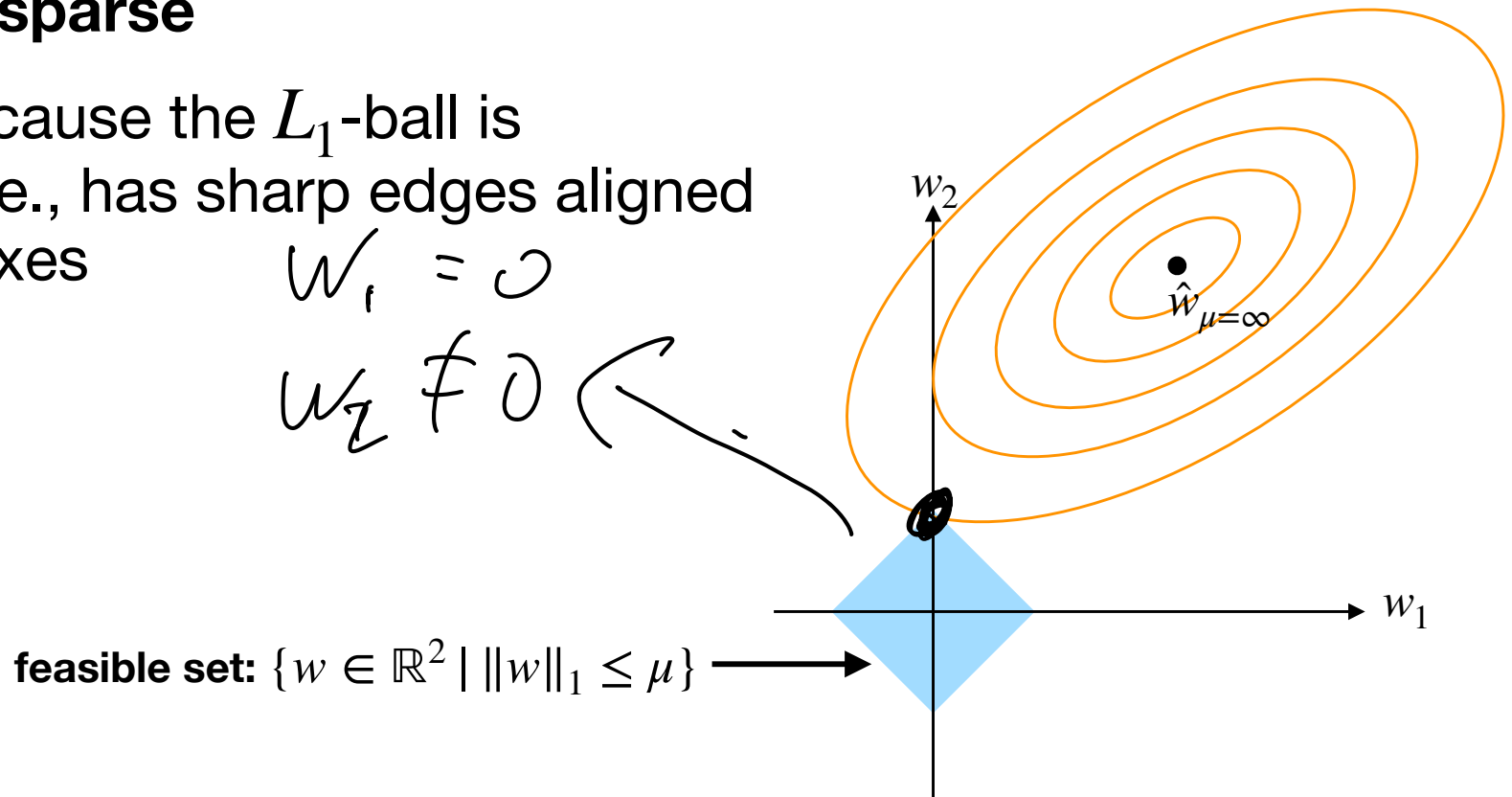
$$\text{minimize}_w \sum_{i=1}^n (w^T x_i - y_i)^2$$

$$\text{subject to } \|w\|_1 \leq \mu$$

- For small enough μ , the optimal solution becomes **sparse**
- This is because the L_1 -ball is “pointy”, i.e., has sharp edges aligned with the axes

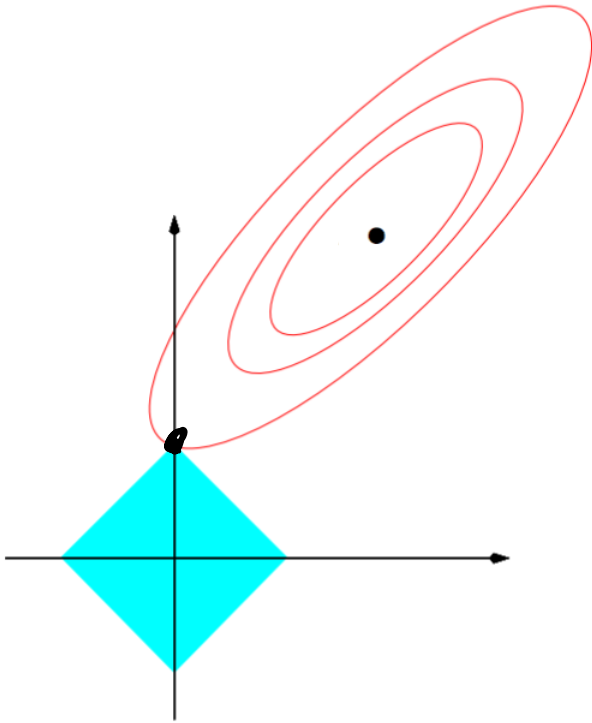
$$w_1 = 0$$

$$w_2 \neq 0$$



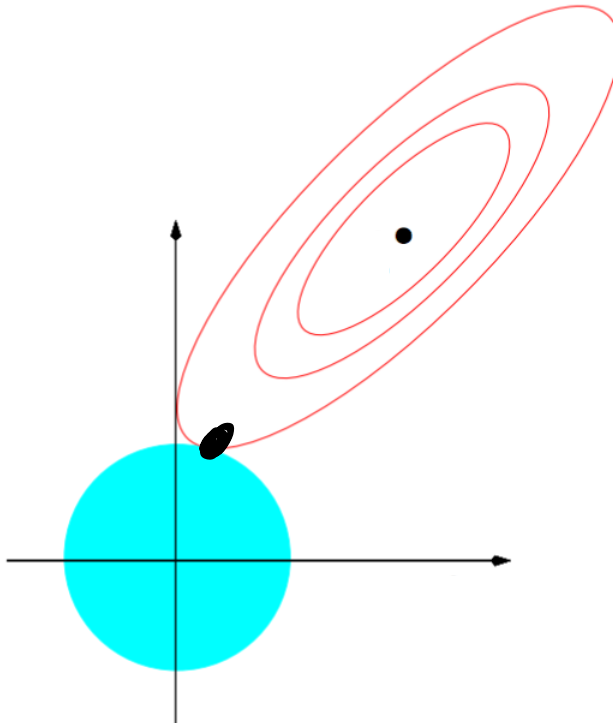
Penalized Least Squares

- Lasso regression finds sparse solutions, as L_1 -ball is “pointy”
- Ridge regression finds dense solutions, as L_2 -ball is “smooth”



$$\text{minimize}_w \sum_{i=1}^n (w^T x_i - y_i)^2$$

$$\text{subject to } \|w\|_1 \leq \mu$$



$$\text{minimize}_w \sum_{i=1}^n (w^T x_i - y_i)^2$$

$$\text{subject to } \underbrace{\|w\|_2^2}_{\text{smooth}} \leq \mu$$