Logistics:

- HW0 graded, for regrade request submit it through GradeScope within 7 days from release of grade.
- HW1 due Tuesday Jan 25th midnight

Lecture 9: Simple variable selection: LASSO for sparse regression

feature

- Yet another hyper-parameter/family of model classes, but with a special property
  - # of features in polynomial regression
  - Regularization coefficient λ for ridge regression
  - Regularization coefficient λ for LASSO

Sparsity

$$\widehat{w}_{LS} = \arg\min_{w} \sum_{i=1}^{n} \left(y_i - x_i^T w\right)^2$$

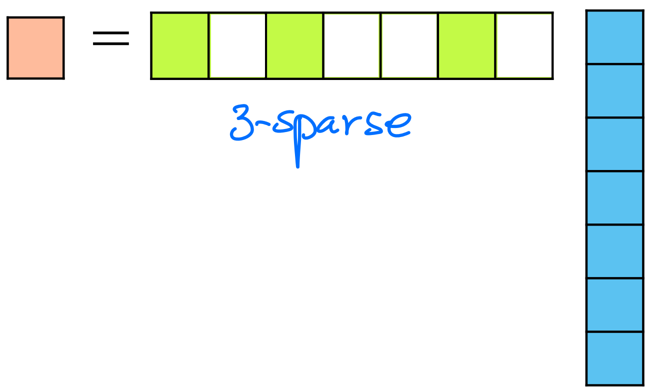
- Vector w is **sparse**, if many entries are zero
    - A vector w is said to be k-sparse if at most k entries are non-zero
    - We are interested in k-sparse w with $k \ll d$
    - Why do we prefer sparse vector w in practice?

Sparsity

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

- Vector w is **sparse**, if many entries are zero
  - **Efficiency**: If size(w) = 100 Billion, each prediction $w^T x$ is expensive:
    - If w is sparse, prediction computation only depends on number of non-zeros in w

$$\widehat{y}_i = \widehat{w}_{\mathrm{LS}}^T x_i$$



$$= \sum_{j=1}^{d} \widehat{w}_{\mathrm{LS}}[j] \times x_i[j] = \sum_{j: w_{\mathrm{LS}}[j] \neq 0} \widehat{w}_{\mathrm{LS}}[j] \times x_i[j]$$

Computational complexity decreases from $2d$ to $2k$ for k-sparse $\widehat{w}_{\mathrm{LS}}$

Sparsity

$$\widehat{w}_{LS} = \arg\min_{w} \sum_{i=1}^{n} \left(y_i - x_i^T w\right)^2$$

- Vector w is **sparse**, if many entries are zero
  - **Interpretability**: What are the relevant features to make a prediction?

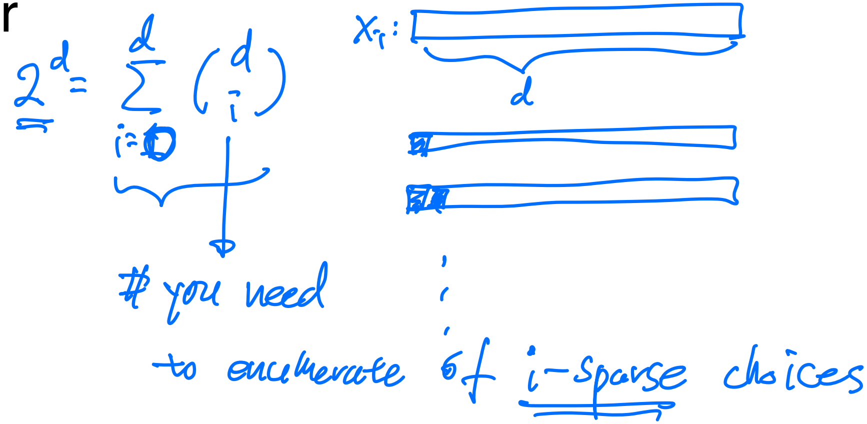
Lot size
Single Family
Year built
Last sold price
Last sale price/sqft
Finished sqft
Unfinished sqft
Finished basement sqft
floors
Flooring types
Parking type
Parking amount
Cooling
Heating
Exterior materials
Roof type
Structure style

Dishwasher
Garbage disposal
Microwave
Range / Oven
Refrigerator
Washer
Dryer
Laundry location
Heating type
Jetted Tub
Deck
Fenced Yard
Lawn
Garden
Sprinkler System

- How do we find "best" subset of features useful in predicting the price among all possible combinations?

Finding best subset of features that explain the outcome/label: Exhaustive

- Try all subsets of size 1, 2, 3, … and one that minimizes validation error
  - Problem?
  - Any Ideas?



Finding best subset: Greedy

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

Forward Greedy

1: $T \leftarrow \emptyset$

2: **For** $j = 1, \ldots, k$ **do**

3: $$j^* \leftarrow \arg\min_{\ell} \min_{w} \sum_{i=1}^{n} \left(y_i - \sum_{j \in T \cup \{\ell\}} w[j] \times x_i[j] \right)^2$$

4: $T \leftarrow T \cup \{j^*\}$

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

Combining forward and backward steps:
In forward algorithm, insert steps to remove features no longer as important

Lots of other variants, too.

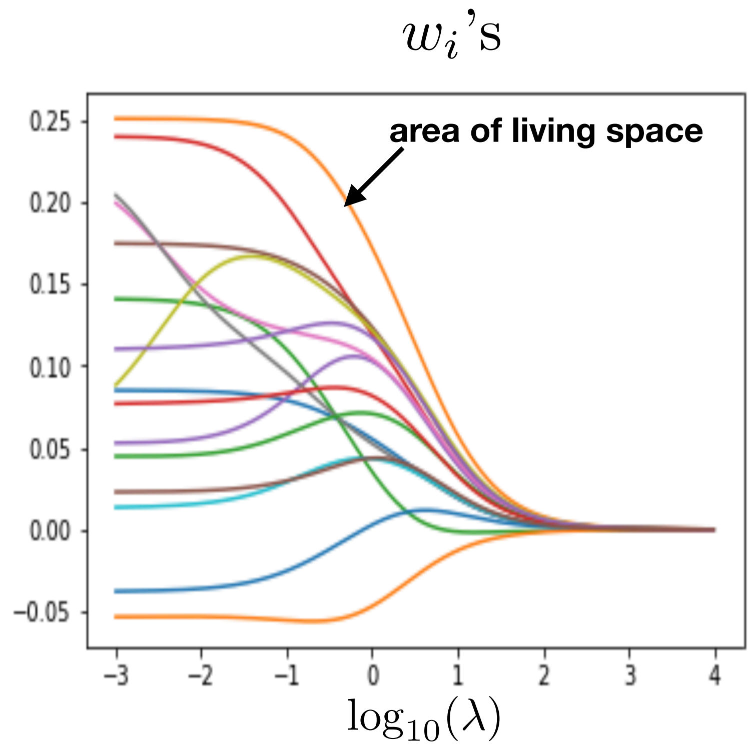
Finding best subset: Regularize

Principled way to get sparsity

Recall that Ridge regression makes coefficients small

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

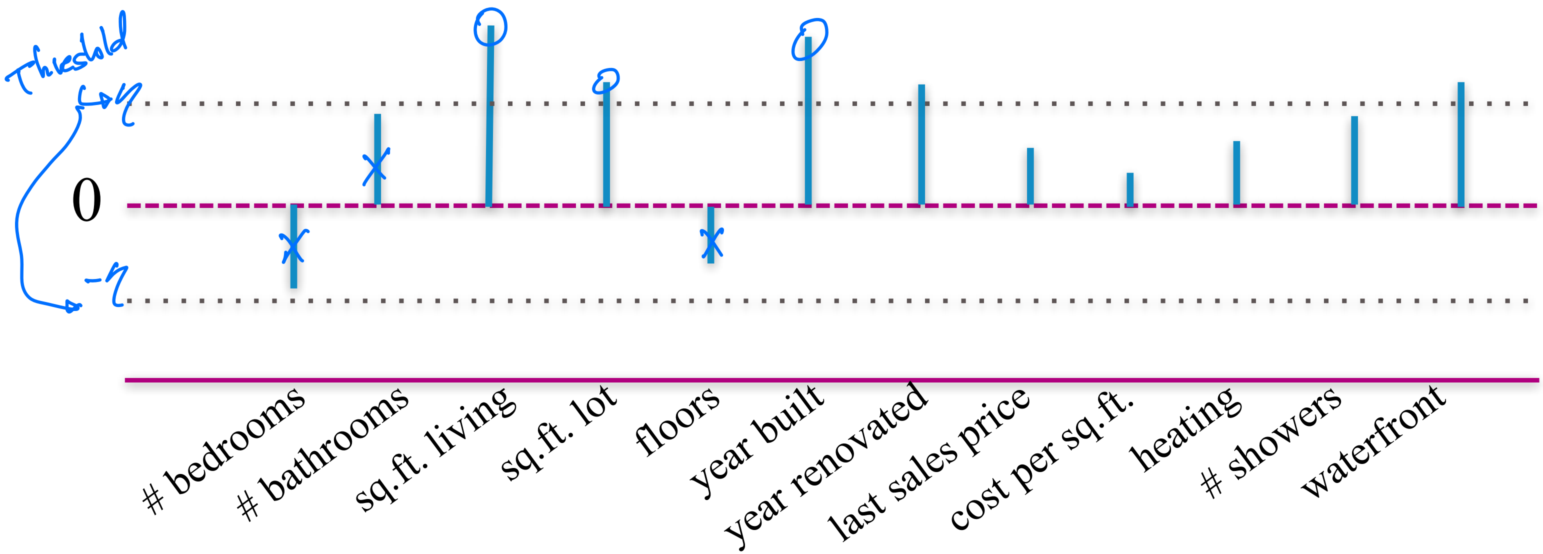
$\sum_{j=1}^{d} w[j]^2$



Thresholded Ridge Regression

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

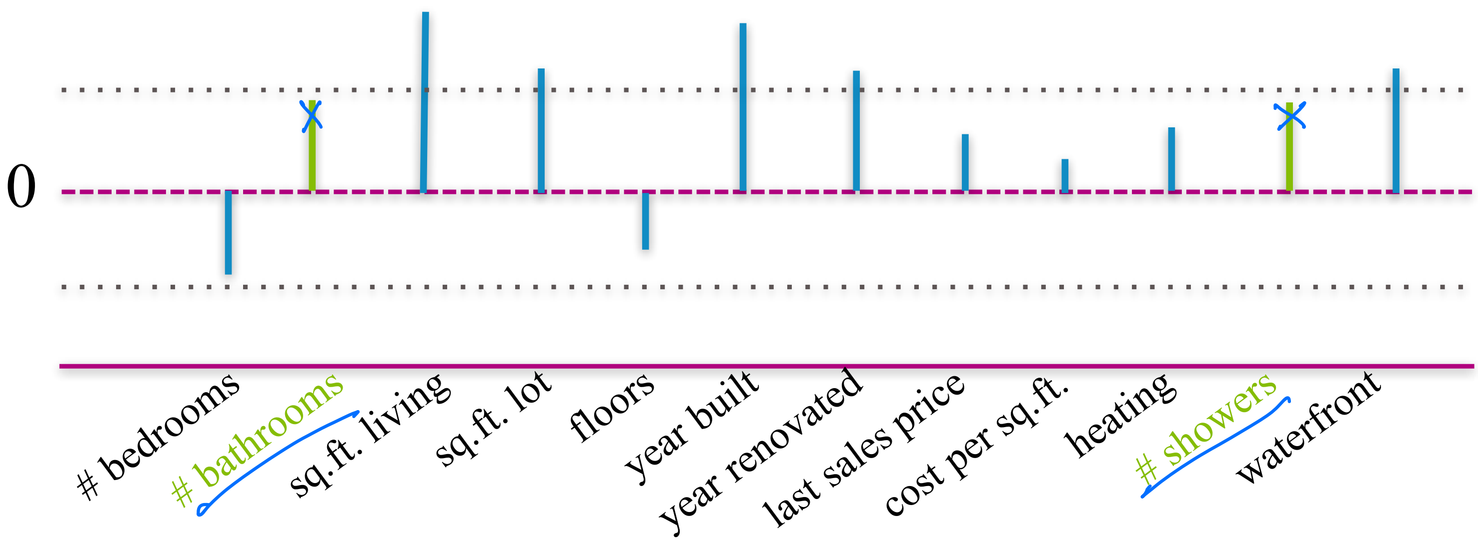
- Why don't we just set **small** ridge coefficients to 0?
  - Any issues?



Thresholded Ridge Regression

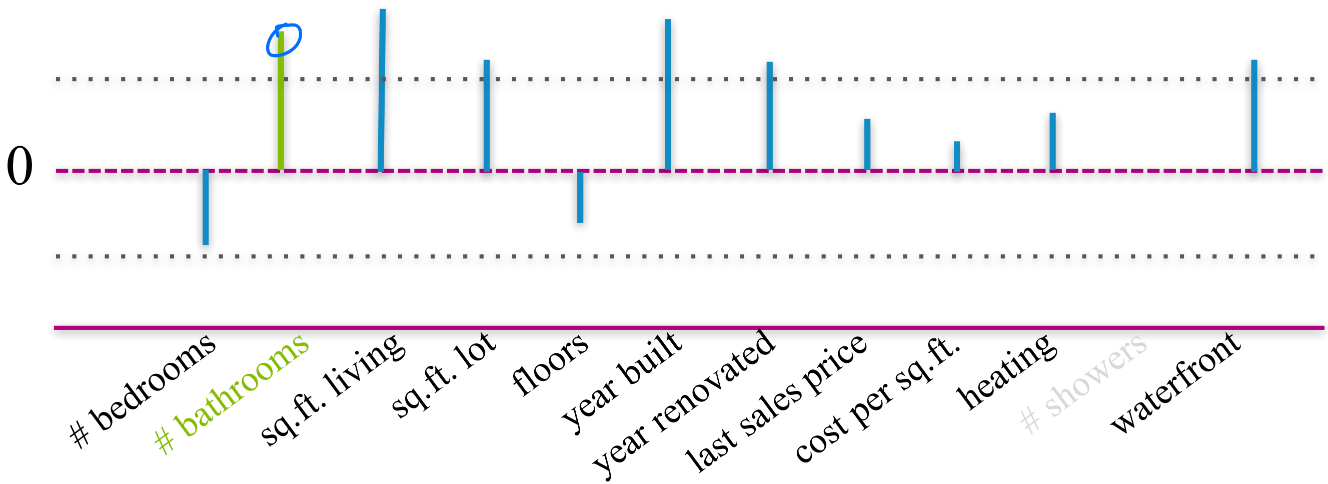
$$\widehat{w}_{ridge} = \arg\min_{w} \sum_{i=1}^{n} \left(y_i - x_i^T w\right)^2 + \lambda ||w||_2^2$$

- Consider two related features (bathrooms, showers)
- Consider $\widehat{w}[\text{bath}] = 1$ and $\widehat{w}[\text{shower}] = 1$, and $\to \lambda \cdot (1^2 + 1^2) = 2\lambda$
  $\widehat{w}[\text{bath}] = 2$ and $\widehat{w}[\text{shower}] = 0$, $\to \lambda(2^2 + 0) = 4\lambda$
  which one does ridge regression choose?
  (assuming #bathroom=#showers in every house)



Thresholded Ridge Regression

- Consider two related features (bathrooms, showers)
- Issue with thresholded ridge regression is that
  ridge regression prefers balanced weights between similar features
- What if we **didn't** include showers? Weight on bathrooms increases, and it should have been selected.
- We want a feature selection scheme that selects one of (#bathroom) or (#showers) automatically,
  using the fact that if you delete #showers #bathroom is an important feature



- There is a better regularizer for sparse regression,
  that can perform the feature selection automatically.

Ridge vs. Lasso Regression

- Recall Ridge Regression objective:

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

- sensitivity of a model w is measured in squared ℓ_2 norm $\|w\|_2^2$
- A principled method to get sparse model is **Lasso** with regularized objective:

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

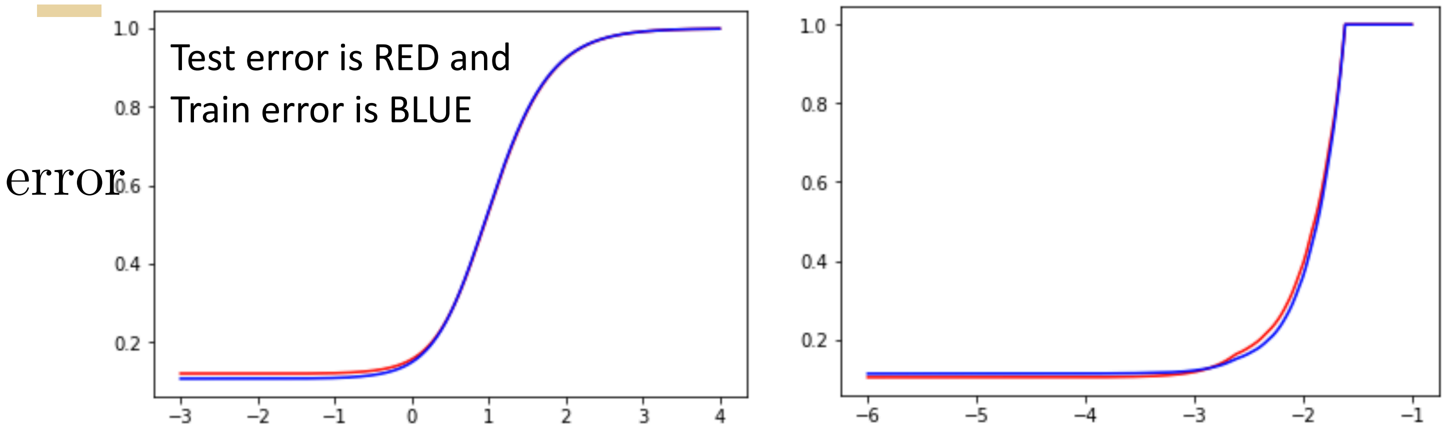
- sensitivity of a model w is measured in ℓ_1 norm:

$$\|w\|_1 = \sum_{j=1}^{d} \left| w[j] \right|$$

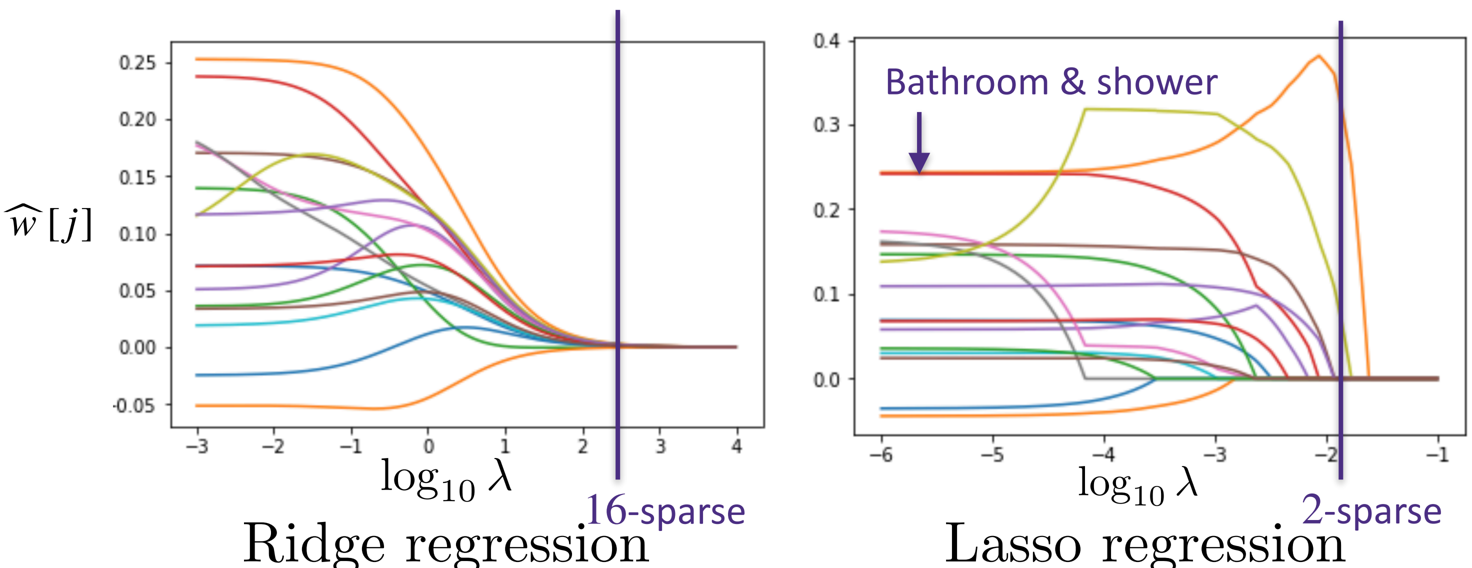
ℓ_p-norm of a vector $w \in \mathbb{R}^d$ is

$$\|w\|_p \triangleq \left(\sum_{j=1}^{d} |w[j]|^p \right)^{1/p}$$

Example: house price with 16 features



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



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
    3. **retrain** with the sparse model and $\lambda = 0$

    why do we need to retrain?

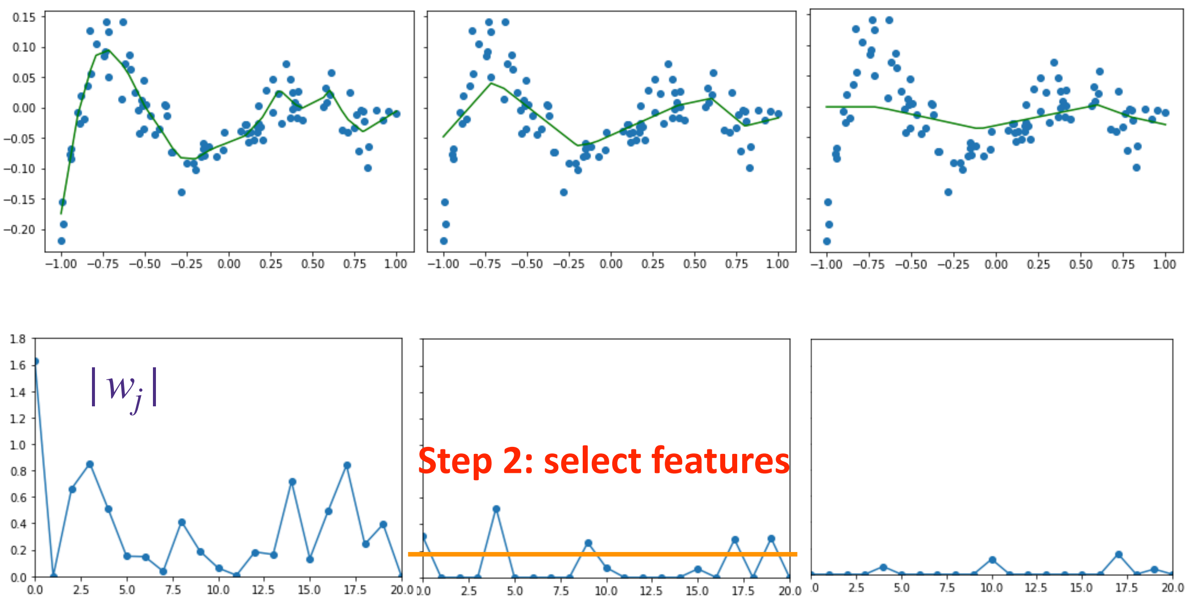
Example: piecewise-linear fit

- We use Lasso on the piece-wise linear example

$$h_0(x) = 1$$
$$h_i(x) = [x + 1.1 - 0.1i]^+$$

Step 1: find optimal $\lambda*$

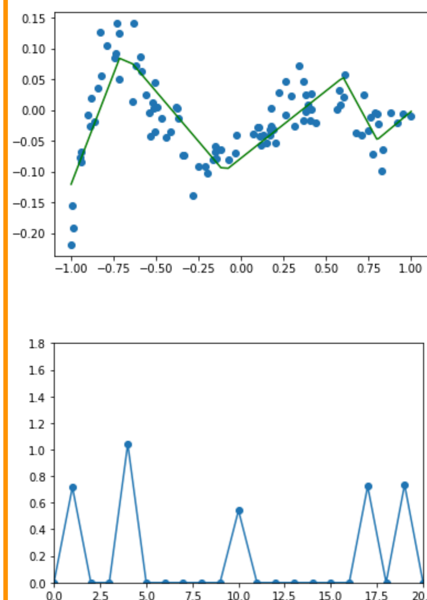
$$\text{minimize}_w \; \mathscr{L}(w) + \lambda \|w\|_1$$



$\lambda = 10^{-8}$ $\lambda = 10^{-4}$ $\lambda = 2 \times 10^{-4}$

Step 3: retrain

$$\text{minimize}_w \; \mathscr{L}(w)$$



$\lambda = 0$

but only use selected features

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

Penalized Least Squares

$$\text{Ridge} : r(w) = ||w||_2^2 \qquad \text{Lasso} : r(w) = ||w||_1$$

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

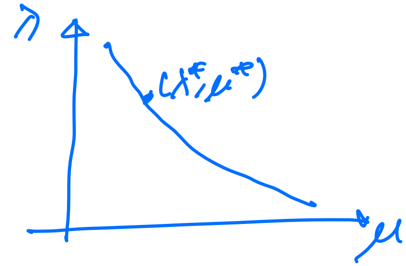
Penalized Least Squares

- Regularized optimization:

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

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

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



- 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, $\widehat{w}_c$, is the same as the solution of the regularized optimization , $\widehat{w}_r$, where

$$\widehat{w}_C = \arg\min_w \sum_{i=1}^{n} (y_i - x_i^T w)^2 \qquad \text{subject to } \; r(w) \leq \mu^*$$

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

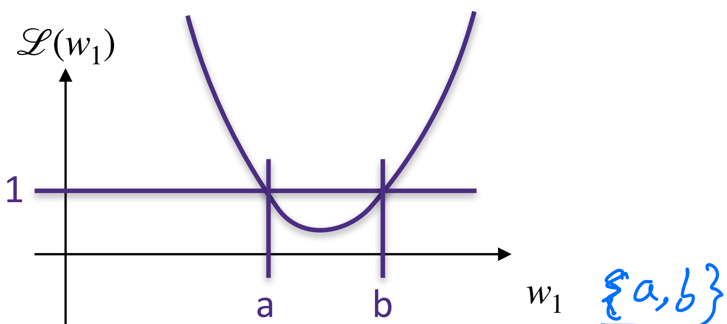
Why does Lasso give sparse solutions?

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

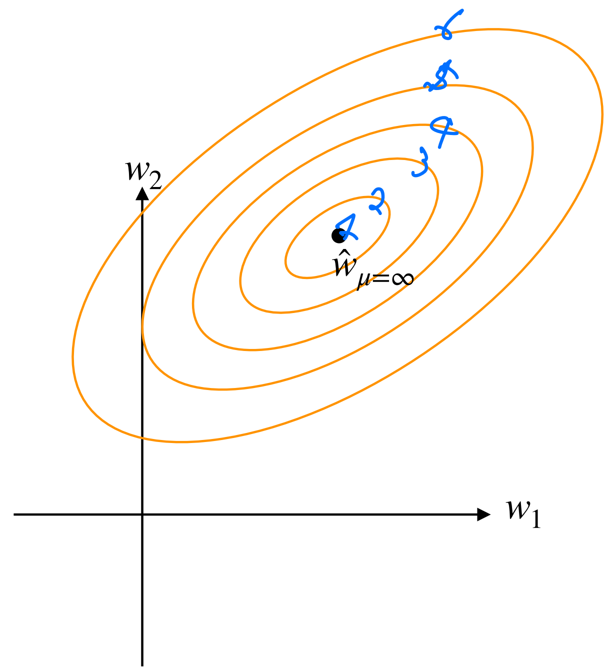
$$\text{subject to} \quad \|w\|_1 \;\leq\; \mu$$

- the **level set** of a function $\mathscr{L}(w_1, w_2)$ 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 $\hat{w}_{\mu=\infty} = \hat{w}_{\text{LS}}$

1-D example with quadratic loss



Level set of $\mathscr{L}(w_1)$ at value 1 is [a,b]



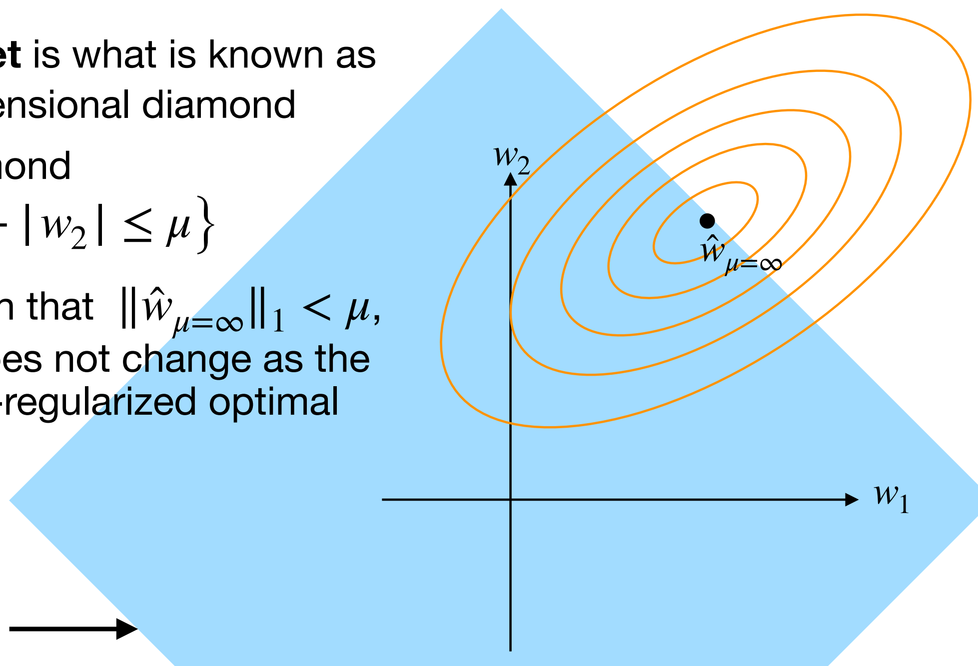
Why does Lasso give sparse solutions?

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

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

$|w_1| + |w_2| \leq \mu$

- 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
$$\left\{ (w_1, w_2) \,\middle|\, |w_1| + |w_2| \leq \mu \right\}$$
- 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



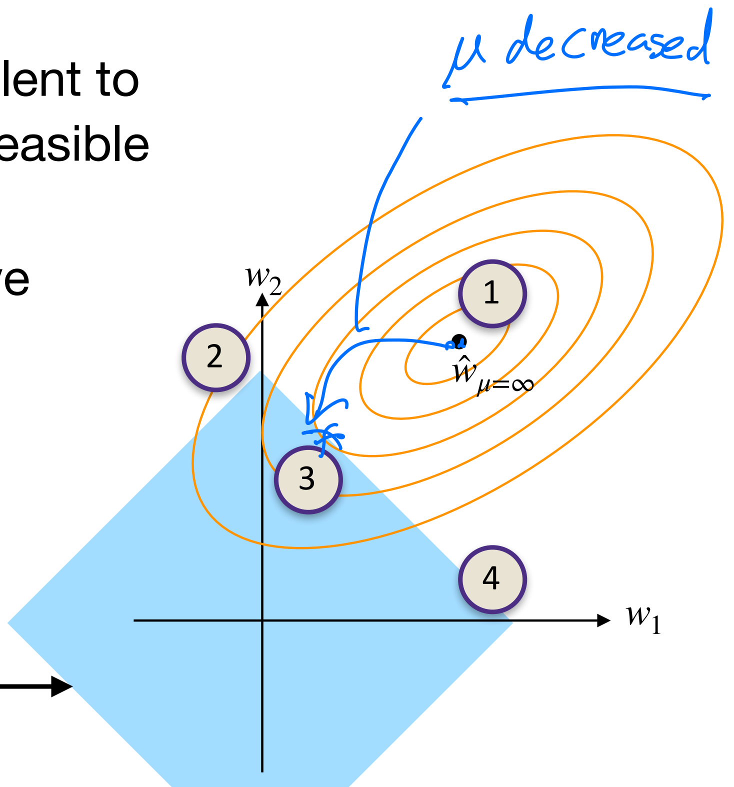
feasible set: $\{w \in \mathbb{R}^2 \mid \|w\|_1 \leq \mu\}$ →

Why does Lasso give sparse solutions?

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

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

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



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

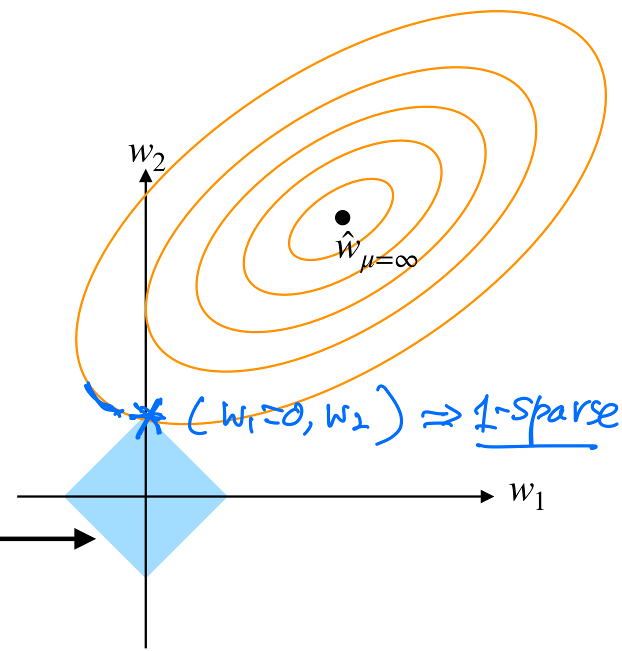
Why does Lasso give sparse solutions?

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

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

decreasing μ ⇕ increasing λ in Lasso

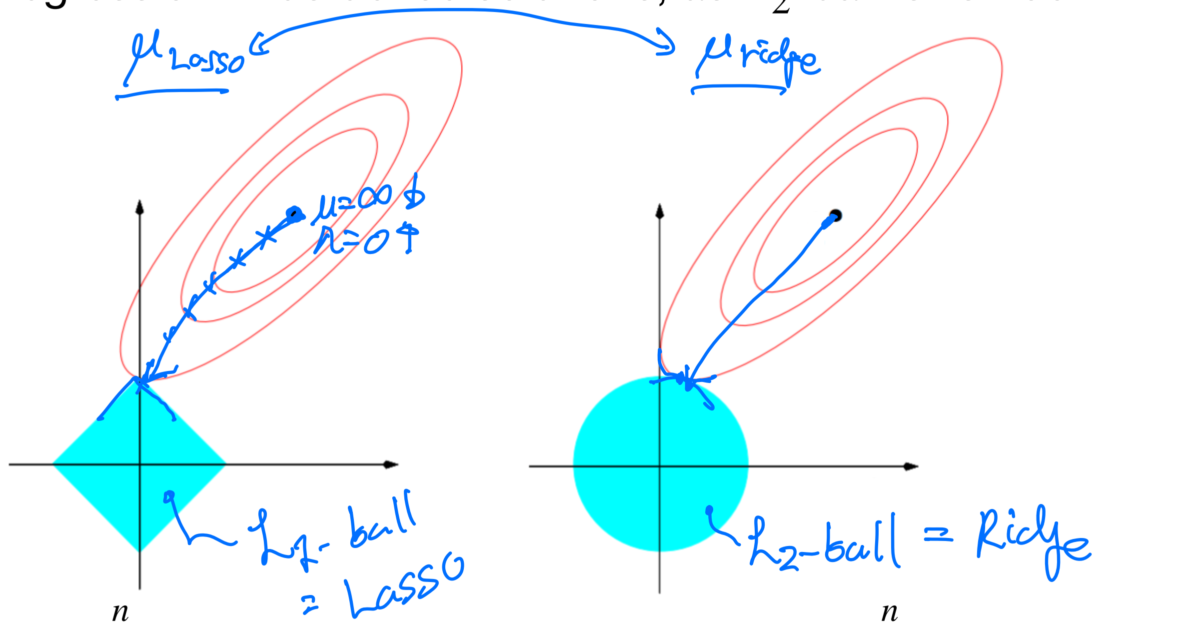
- For small enough μ, the optimal solution becomes **sparse**
- This is because the L_1-ball is "pointy",i.e., has sharp edges aligned with the axes



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

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"



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

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

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

$$\text{subject to} \quad \|w\|_2^2 \ \leq \ \mu$$

Questions?

weight decay

Ridge better when you have

littlie time.

Lasso is slower. using Optimization