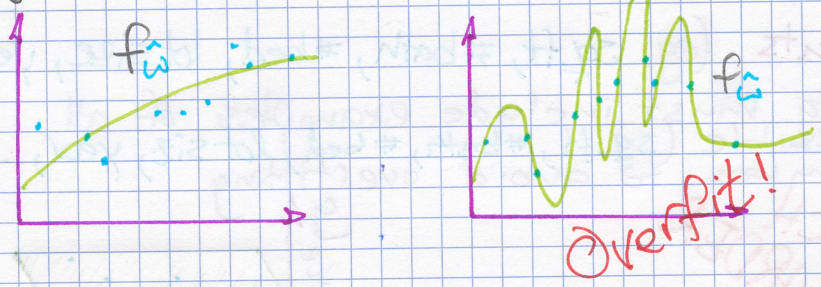
Ridge Regression

Regulating overfitting when using many features

Overfitting of polynomial regression

Flexibility of high-order polynomials

$$y_i = w_0 + w_1 x_i + w_2 x_i^2 + \dots + w_p x_i^p + \varepsilon_i$$



Symptom of overfitting

Often, overfitting associated with very large estimated parameters $\hat{w}$

Overfitting of linear regression models more generically

Overfitting with many features

Not unique to polynomial regression, but also if lots of inputs (D large)

Or generically, lots of features (D large)

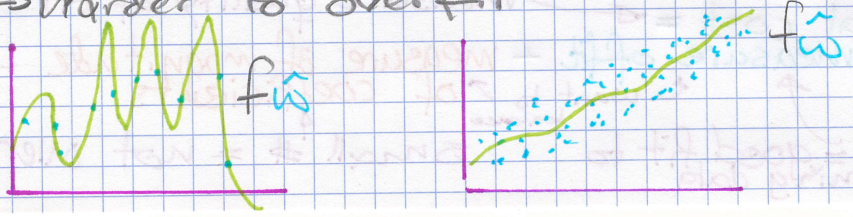
$$y_i = \sum_{j=0}^{D} w_j h_j(x_i) + \varepsilon_i$$

- Square feet
- # bathrooms
- # bedrooms
- Lot size
- Year built
- ...

How does # of observations influence overfitting?

Few observations (N small)
→ rapidly overfit as model complexity increases

Many observations (N very large)
→ harder to overfit



1 input (e.g., sq.ft.)

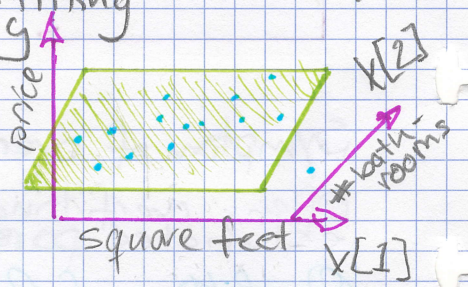
Data must include representative examples of all possible (sq.ft., $) pairs to avoid overfitting

HARD

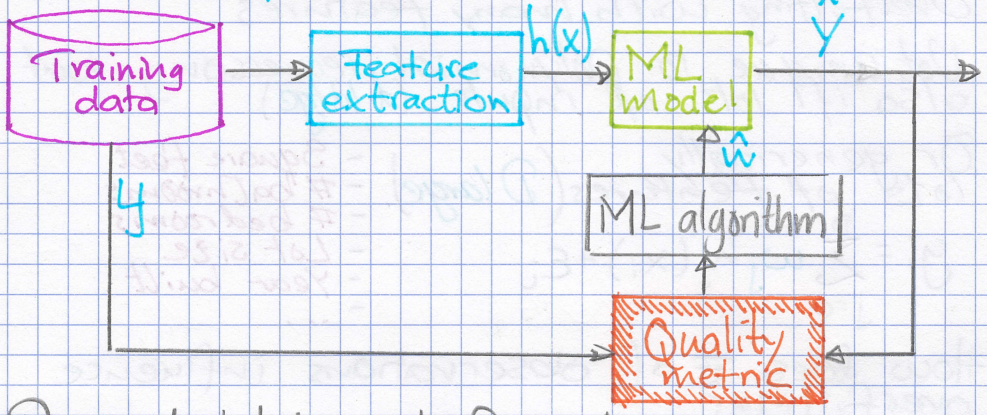
d inputs (eg. sq.ft, #bath, #bed, lot size, year, ...)

Data must include examples of all possible (sq.ft, #bath, #bed, lot size, year, ..., $) combos to avoid overfitting

MUCH HARDER



Adding term to cost-of-fit to prefer small coefficients



Desired total cost format

Want to balance

i. How well function fits data

ii. Magnitude of coefficients

Total cost = measure quality of fit

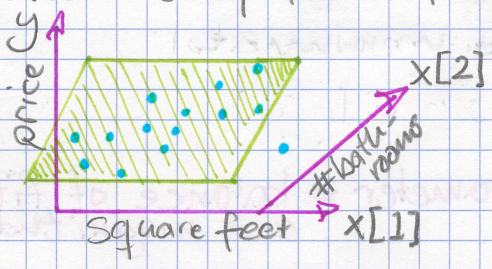
measure of fit + measure of magnitude of coefficients

↑ want to balance

small # = good fit to training data

small # = not overfit

Measure of fit to training data



$$RSS(w) = \sum_{i=1}^{N} \left(y_i - h(x_i)^T w\right)^2$$

$$= \sum_{i=1}^{N} \left(y_i - \hat{y}_i(w)\right)^2$$

$\hat{y}_i(w)$: pred. value using w

small RSS → model fitting training data well

Measure of the magnitude of regression coefficient

What summary # is indicative of size of regression coefficients?

- Sum? $w_0 = 1,527,301$, $w_1 = -1,605,253$
  $w_0 + w_1$ = small #
- Sum of absolute value?

$$|w_0| + |w_1| + \ldots + |w_D| = \sum_{j=0}^{D} |w_j| \Leftrightarrow \|w\|_1$$

L_1 norm

- Sum of squares (L_2 norm)

$$w_0^2 + w_1^2 + \ldots + w_D^2 = \sum_{j=0}^{D} w_j^2 \Leftrightarrow \|w\|_2^2$$

L_2 norm

Total cost =

measure of fit + measure of magnitude of coefficients

$RSS(w)$ + $\|w\|_2^2$

Consider resulting objective

What if $\hat{w}$ selected to minimize

$$RSS(w) + \lambda \|w\|_2^2$$

λ: tuning parameter = balance of fit and magnitude

If $\lambda = 0$:
reduces to minimizing RSS(w), as before
(old solution) $\rightarrow \hat{w}^{LS}$ (least squares)

If $\lambda = \infty$:
For solutions where $\hat{w} \neq 0$, then total cost is ∞
If $\hat{w} = 0$, then total cost = RSS(0)
$\rightarrow$ solution is $\hat{w} = 0$

If λ in between
Then $0 \leq \|\hat{w}\|_2^2 \leq \|\hat{w}^{LS}\|_2^2$

Ridge regression
(a.k.a L_2 regularization)

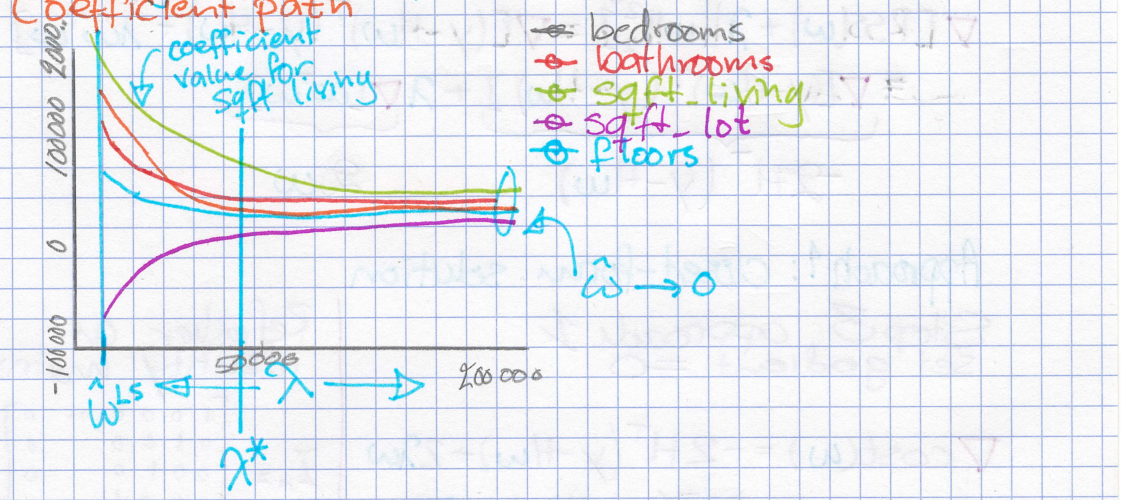
Bias-variance tradeoff

Large λ:
high bias, low variance
(eg, $\hat{w} = 0$ for $\lambda = \infty$)

{In essence, λ controls model complexity

Small λ:
low bias, high variance
(eg, standard least squares (RSS) fit of high-order polynomial for $\lambda = 0$)

The ridge coefficient path



Fitting the ridge regression model (for given λ value)

Step 1: Rewrite total cost in matrix notation

Rewrite magnitude of coefficients in vector notation

$$\|w\|_2^2 = w_0^2 + w_1^2 + w_2^2 + \ldots + w_D^2$$

$$= \begin{bmatrix} w_0 & w_1 & w_2 & \cdots & w_D \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_D \end{bmatrix}$$

$$= w^T w$$

Putting it all together

In matrix form, ridge regression cost is:

$$RSS(w) + \lambda \|w\|_2^2$$

$$= (y - Hw)^T (y - Hw) + \lambda w^T w$$

Gradient of ridge regression cost

$$\nabla[RSS(w) + \lambda\|w\|_2^2] = \nabla[(y-Hw)^T(y-Hw) + \lambda w^T w]$$

$$= \underbrace{\nabla[(y-Hw)^T(y-Hw)]}_{-2H^T(y-Hw)} + \lambda \underbrace{\nabla[w^T w]}_{2w}$$

Approach 1: closed-form solution

Step 3: approach 1
set gradient $= 0$

$$\nabla cost(w) = -2H^T(y-Hw) + 2\lambda w$$
$$= -2H^T(y-Hw) + 2\lambda I w$$

Refresher on identity matrix

$$I_n = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & & 0 \\ 0 & 0 & 1 & & 0 \\ \vdots & & & \ddots & \\ 0 & 0 & 0 & & 1 \end{bmatrix}$$

$Iv = v$

$IA = A$

$A^{-1}A = I$

$AA^{-1} = I$

Ridge closed-form solution:

$$\nabla cost(w) = -2H^T(y-Hw) + 2\lambda I w = 0$$

$$-H^T y + H^T H w + \lambda I \hat{w} = 0$$

$$H^T H \hat{w} + \lambda I \hat{w} = H^T y$$

$$(H^T H + \lambda I)\hat{w} = H^T y$$

$$\hat{w} = (H^T H + \lambda I)^{-1} H^T y$$

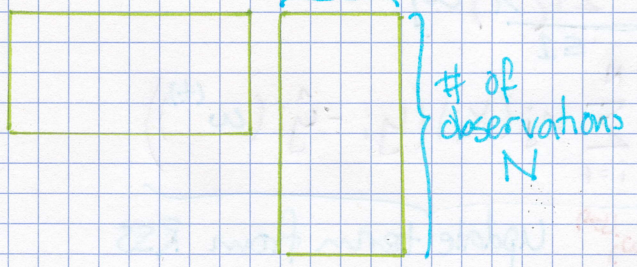
Interpreting ridge closed-form solution

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

If $\lambda = 0$: $\hat{w}^{ridge} = (H^T H)^{-1} H^T y = \hat{w}^{LS}$ ← old solution!

If $\lambda = \infty$: $\hat{w}^{ridge} = 0$ ← because it's like dividing by ∞

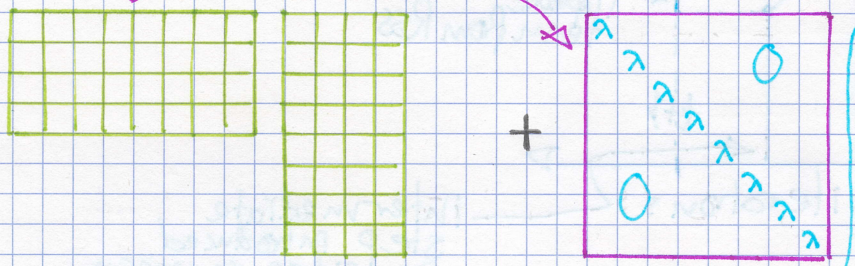
$\hat{w}^{LS} = (H^T H)^{-1} H^T y$



Invertible if
In general, (# linearly independent obs.) $N > D$

Complexity of inverse $O(D^3)$
really important for large D (lots of features)

$\hat{w} = (H^T H + \lambda I)^{-1} H^T y$



Invertible if
Always if $\lambda > 0$
even if $N < D$

Complexity of inverse
$O(D^3)$ big for large D!

"λI is making" → "regular" → "regularized" $H^T H + \lambda I$ more

Approach 2: gradient descent

Step 3, approach 2: Gradient descent

Element-wise ridge regression gradient descent algorithm

$\nabla cost(w) = -2H^T(y - Hw) + 2\lambda w$

Update to j^{th} feature weight:

$$w_j^{(t+1)} \leftarrow w_j^{(t)} - \eta * \left[-2 \sum_{i=1}^{N} h_j(x_i)(y_i - \hat{y}_i(w^{(t)})) + 2\lambda w_j^{(t)} \right]$$

new term, comes from the j^{th} component of $2\lambda w$

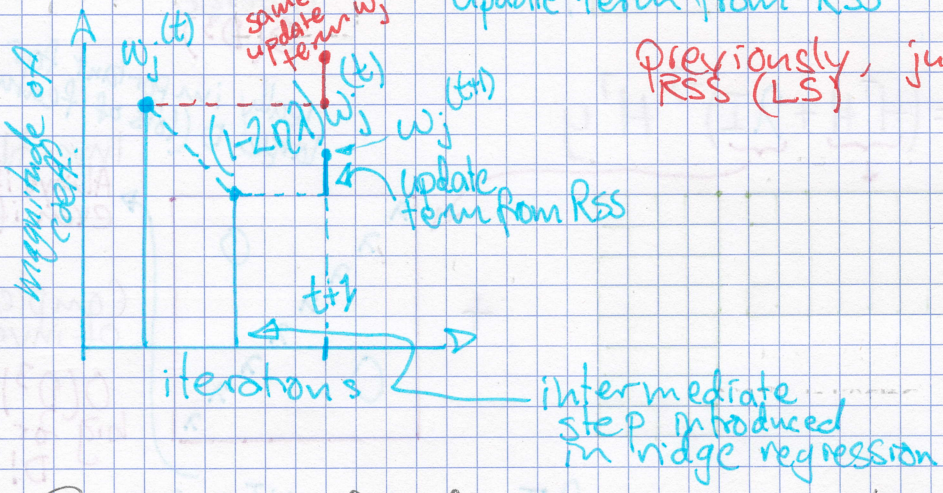
Equivalently: $2\eta\lambda \geq 0$, $\eta > 0$, $\lambda \geq 0$

$$w_j^{(t+1)} \leftarrow (1 - 2\eta\lambda) w_j^{(t)} + 2\eta \sum_{i=1}^{N} h_j(x_i)(y_i - \hat{y}_i(w^{(t)}))$$

($1 - 2\eta\lambda \leq 1$)

Update term from RSS

Previously, just RSS (LS)



Summary of ridge regression algorithm

init $w^{(1)} = 0$ (or randomly, or smarter), $t = 1$

while $\|\nabla RSS(w^{(t)})\| > \varepsilon$

for $j = 0, \ldots, D$

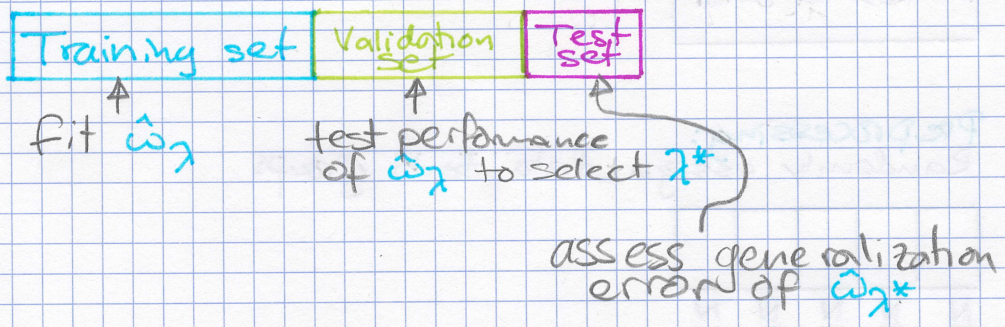
$$partial[j] = -2\sum_{i=1}^{N} h_j(x_i)(y_i - \hat{y}_i(w^{(t)}))$$

$$w_j^{(t+1)} \leftarrow (1 - 2\eta\lambda) w_j^{(t)} - \eta\, partial[j]$$

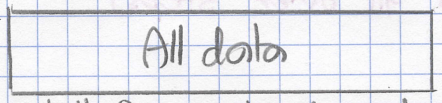
$t \leftarrow t + 1$

How to choose λ

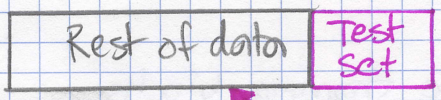
If sufficient amount of data



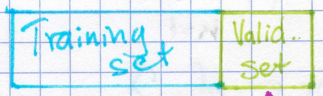
Start with smallish dataset



Still form test set and hold out



use for both training and validation, but not so naively

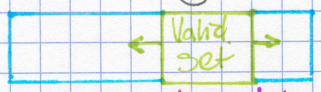


small validation set

Is validation set enough to compare performance of $\hat{w}_\lambda$ across λ values?

NO

Choosing the validation set

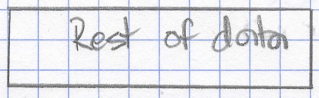


small validation set

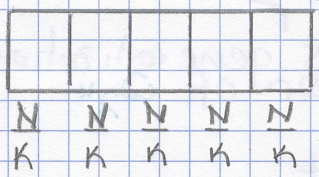
Which subset should I use?

ALL! {average performance over all choices

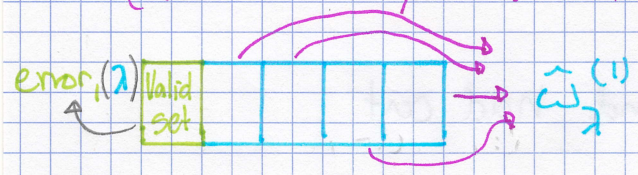
K-fold cross validation



Preprocessing:
Randomly assign data to K groups



(use same split of data for all other steps)



For $k = 1, \ldots, K$
1. Estimate $\hat{w}_\lambda^{(k)}$ on the training blocks
2. Compute error on validation block: $error_k(\lambda)$

Compute average error $CV(\lambda) = \frac{1}{K} \sum_{k=1}^{K} error_k(\lambda)$

Repeat procedure for each choice of λ

Choose λ^* to minimize $CV(\lambda)$

What value of K?
Formally, the best approximation occurs for validation sets of size 1 ($K = N$)

leave-one-out cross validation

Computationally intensive
- requires computing N fits of model per λ

Typically, $K = 5$ or 10

5-fold CV
10-fold CV

How to handle the intercept

Recall multiple regression model

$$y_i = w_0 h_0(x_i) + w_1 h_1(x_i) + \ldots + w_D h_D(x_i) + \varepsilon_i$$

$$= \sum_{j=0}^{D} w_j h_j(x_i) + \varepsilon_i$$

feature 1 = $h_0(x)$ ... often 1 (constant)

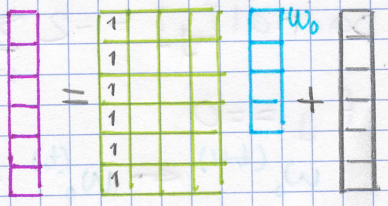
feature 2 = $h_1(x)$ ... e.g., x[1]

feature 3 = $h_2(x)$ ... e.g., x[2]

feature D+1 = $h_D(x)$ ... e.g., x[d]

In matrix notation for N observations

$$y = Hw + \varepsilon$$



Do we penalize intercept?

Standard ridge regression cost

$$RSS(w) + \lambda \|w\|_2^2$$

↑ strength of penalty

Encourages intercept w_0 to also be small

Do we want a small intercept? Conceptually, not indicative of overfitting...

Option 1: Don't penalize intercept

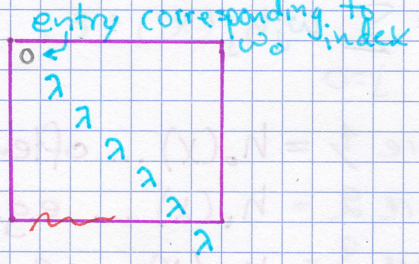
Modified ridge regression cost:

$$RSS(w_0, w_{rest}) + \lambda \|w_{rest}\|_2^2$$

How to implement this in practice?

Option 1: Don't penalize intercept
– Closed-form solution –

$$\hat{w} = (H^T H + \lambda I^{mod})^{-1} H^T y$$



Option 1: Don't penalize intercept
Gradient descent algorithm

while $\|\nabla RSS(w^t)\| > \varepsilon$

for $j = 0, \ldots, D$:

$$partial[j] = -2 \sum_{i=1}^{N} h_j(x_i)(y_i - \hat{y}_i(w^{(t)}))$$

if $j == 0$:

$$w_0^{(t+1)} \leftarrow w_0^{(t)} - \eta\, partial[j]$$ ← old LS update, no shrinkage to w_0

else for all other features

$$w_j^{(t+1)} \leftarrow (1 - 2\eta\lambda) w_j^{(t)} - \eta\, partial[j]$$

$t \leftarrow t + 1$

↑ ridge regression

Option 2: Center data first

If data are first centered about 0, then favoring small intercept not so worrisome

Step 1: Transform y to have 0 mean

Step 2: Run ridge regression as normal (closed-form or gradient algorithms)