# Do not distribute course material

https://www.cs.princeton.edu/courses/archive/spr09/cos513/scribe/lecture10.pdf

http://www.stat.cmu.edu/~ryantibs/advmethods/notes/highdim.pdf

http://timroughgarden.org/s17/l/l6.pdf

# Regularization Preventing overfitting

A way to prefer some model/hypothesis over others in our class based on some idea of what is the preferred model

ence…

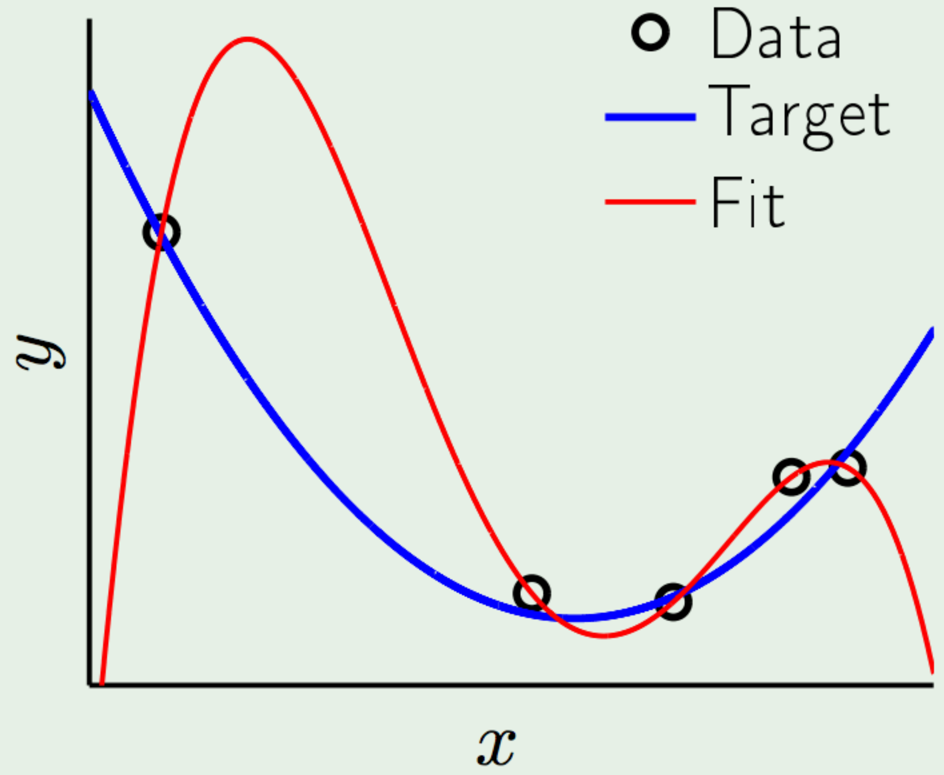
How can we reduce the out of sample error by preferring some solutions in our hypothesis class

$$E_{in}(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - \hat{y})^2 = \frac{1}{N}RSS(\mathbf{w})$$

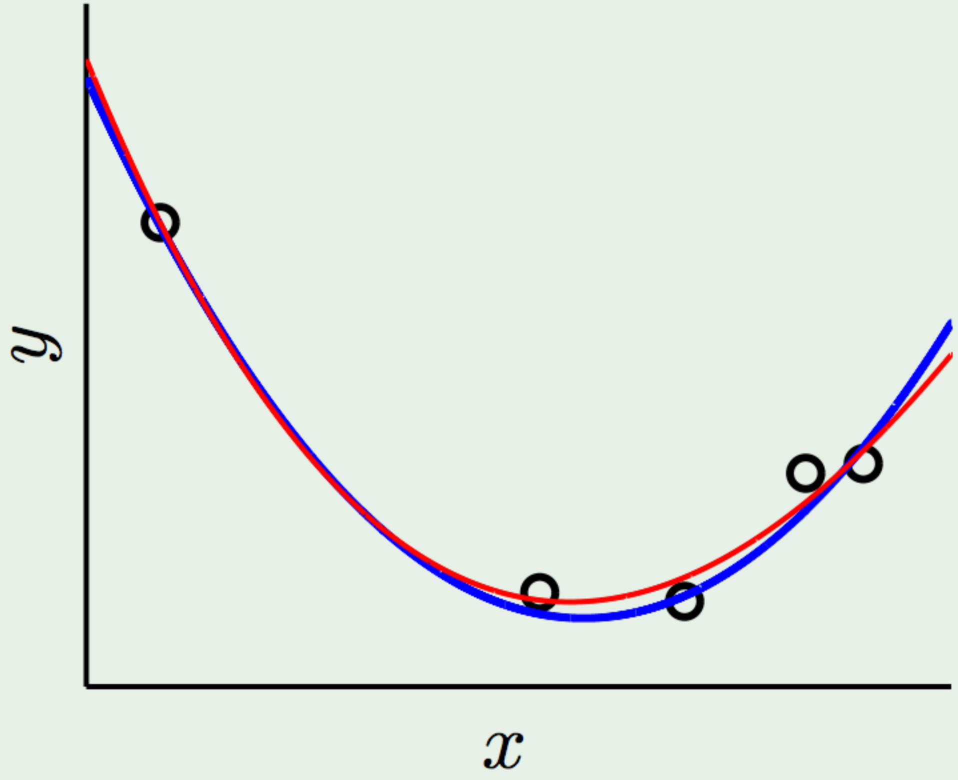
**Occam's razor** (Latin: *novacula Occami*)

“entities should not be multiplied beyond necessity”, sometimes inaccurately paraphrased as "the simplest explanation is usually the best one.”

# Putting the brakes

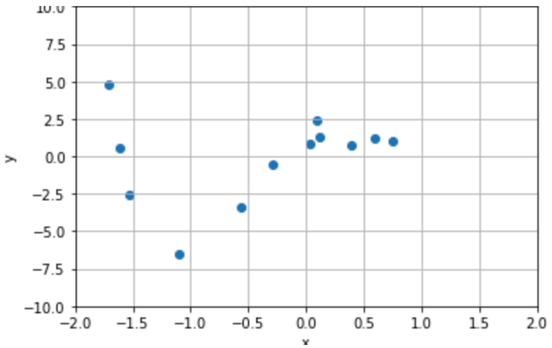


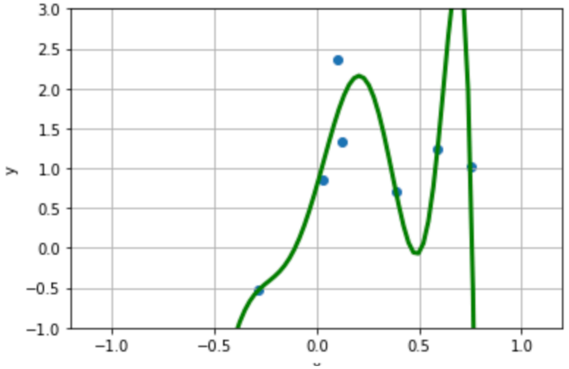

free fit




restrained fit

# Example:





Poor Generalization!

$\mathbf{w}^T$=[0.8 9 9 -76 -176 208 551 36 -479 -328 -66]

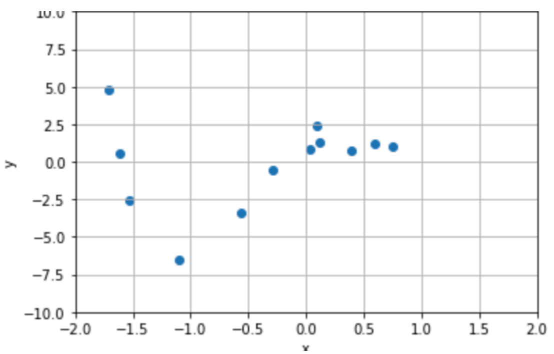
Observations:
Notice that the amount of overfitting depended on the order of the model and how many examples we have.
Our hypothesis that overfit had large coefficients. How could we keep the coefficients small?

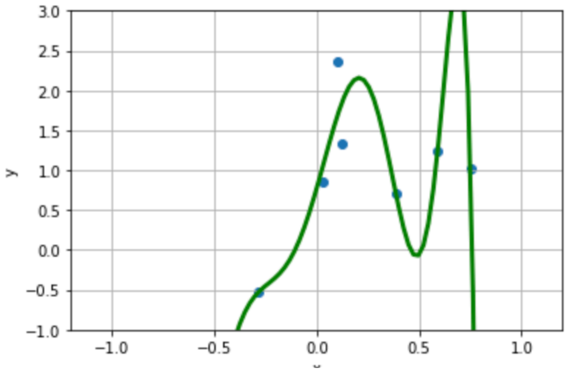
We will need to balance between how well we fit the data (the in sample error) and how much we restrict the size of our coefficients (that we are using to prevent overfitting)

$$\underbrace{E_{\text{in}}(\mathbf{w})}_{\text{fit}} + \underbrace{\text{penalty for large } \mathbf{w}}_{\text{restrict the size of our coefficients}}$$

NYU TANDON SCHOOL OF ENGINEERING

# Example:




$\mathbf{w}^T$=[0.8 9 9 -76 -176 208 551

**Poor Generalization!**

If $w_j = 551$ then a small change to the value of the $j$th feature makes a huge change in $\hat{y}$

Observations:
Notice that the amount of overfitting depended on the order of the model and how many examples we have.
Our hypothesis that overfit had large coefficients. How could we keep the coefficients small?

We will need to balance between how well we fit the data (the in sample error) and how much we restrict the size of our coefficients (that we are using to prevent overfitting)

$$\underbrace{E_{\text{in}}(\mathbf{w})}_{\text{fit}} + \underbrace{\text{penalty for large } \mathbf{w}}_{\text{restrict the size of our coefficients}}$$

# Preventing Overfitting

- We prefer to have smaller coefficients, or a smaller number of parameters.
  - How do we choose smaller coefficients?
  - How do we choose which parameters are important?
- Want to reduce variance (and possibly increase bias)
- To do this we can change our *objective function*!

$$E_{aug}(\mathbf{w}) = E_{in}(\mathbf{w}) + \text{penalty for complex models}$$

What function should we use?

sum of coefficients?

$$\cancel{w_0} + w_1 + w_2 + \cdots + w_d$$

sum of absolute value of coefficients?

$$|\cancel{w_0}| + |w_1| + |w_2| + \cdots + |w_d|$$

Sum of squares of coefficients?

$$\cancel{w_0^2} + w_1^2 + w_2^2 + \cdots + w_d^2$$

$$E_{lasso}(\mathbf{w}) = E_{in}(\mathbf{w}) + \text{penalty on } \left(|\cancel{w_0}| + |w_1| + \cdots + |w_d|\right)$$

We will explore this penalty - LASSO regression

$$E_{ridge}(\mathbf{w}) = E_{in}(\mathbf{w}) + \text{penalty on } (\cancel{w_0^2} + w_1^2 + \cdots + w_d^2)$$

We will explore this penalty - Ridge Regression

- Tuning parameter $\lambda$ to balance fit and number of parameters

$$E_{lasso}(\mathbf{w}) = E_{in}(\mathbf{w}) + \lambda\left(|\cancel{w_0}| + |w_1| + \cdots + |w_d|\right)$$

$$E_{ridge}(\mathbf{w}) = E_{in}(\mathbf{w}) + \lambda(\cancel{w_0^2} + w_1^2 + \cdots + w_d^2)$$

$\lambda$ is called the tuning parameter.

$\lambda$ determines the amount regularization

# Preventing Overfitting

d = # features. Note we don't want to restrict $w_0$. Many approaches to this issue are possible. We will leave $w_0$ out of the penalty term. Note: Scaling the features is suggested

sum of coefficients?

$$\cancel{w_0} + w_1 + w_2 + \cdots + w_d$$

sum of absolute value of coefficients?

$$|\cancel{w_0}| + |w_1| + |w_2| + \cdots + |w_d|$$

Sum of squares of coefficients?

$$\cancel{w_0^2} + w_1^2 + w_2^2 + \cdots + w_d^2$$

- We prefer to have smaller coefficients, or a smaller number of parameters.
  - How do we choose smaller coefficients?
  - How do we choose which parameters are important?
- Want to reduce variance (and possibly increase bias)
- To do this we can change our *objective function*!

$$\mathrm{E}_{\mathrm{aug}}(\mathbf{w}) = \mathrm{E}_{\mathrm{in}}(\mathbf{w}) + \text{penalty for complex models}$$

This is a technique that is used in when learning many other models

$$\mathrm{E}_{\mathrm{lasso}}(\mathbf{w}) = \mathrm{E}_{\mathrm{in}}(\mathbf{w}) + \begin{matrix}\text{penalty}\\ \text{on}\end{matrix} \left(|\cancel{w_0}| + |w_1| + \cdots + |w_d|\right)$$

We will explore this penalty - LASSO regression

$$\mathrm{E}_{\mathrm{ridge}}(\mathbf{w}) = \mathrm{E}_{\mathrm{in}}(\mathbf{w}) + \begin{matrix}\text{penalty}\\ \text{on}\end{matrix} (\cancel{w_0^2} + w_1^2 + \cdots + w_d^2)$$

We will explore this penalty - Ridge Regression

- Tuning parameter $\lambda$ to balance fit and number of par

$$\mathrm{E}_{\mathrm{lasso}}(\mathbf{w}) = \mathrm{E}_{\mathrm{in}}(\mathbf{w}) + \lambda\left(|\cancel{w_0}| + |w_1| + \cdots + |w_d|\right)$$

$$\mathrm{E}_{\mathrm{ridge}}(\mathbf{w}) = \mathrm{E}_{\mathrm{in}}(\mathbf{w}) + \lambda(\cancel{w_0^2} + w_1^2 + \cdots + w_d^2)$$

Choosing the right $\lambda$ is also part of model selection

We will focus on choosing the right tuning parameter for accuracy (not interpretation)

zation

# *Ridge Regression*

## $L_2$ regularization

$$\mathrm{E}_{\text{ridge}}(\mathbf{w}) = E_{\text{in}}(\mathbf{w}) + \lambda(\cancel{w_0^2} + w_1^2 + w_2^2 + \cdots + w_d^2)$$

The size of a vector is referred to as the norm of the vector. What is the "size". It depends
The L2 norm of a vector is the square root of the sum of the squared vector values

$$\mathbf{v}^T = [1,2,3]$$

$$\|\mathbf{v}\|_2 = \sqrt{1^2 + 2^2 + 3^2}$$

# Ridge Regression
# $L_2$ regularization

- Tuning parameter $\lambda$ to balance fit and number of parameters

$$E_{\text{ridge}} = E_{\text{in}}(\mathbf{w}) + \text{penalty for complex models}$$

$$E_{\text{ridge}}(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \mathbf{w}^T\mathbf{x}_i)^2 + \lambda(w_1^2 + w_2^2 + \cdots + w_d^2)$$

- $\lambda$ controls the model complexity
  - Large $\lambda$
    - high bias, low variance
  - small $\lambda$
    - low bias, high variance

If $\lambda$=0 then

$$\mathbf{w}_{\text{ridge}} = \mathbf{w}_{\text{lin}}$$

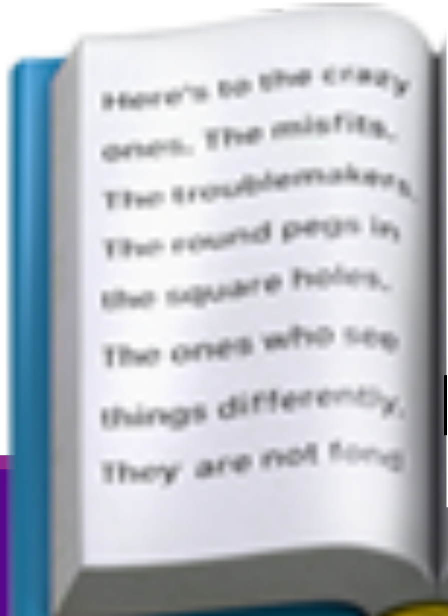
$\mathbf{w}_{\text{lin}}$ = the best parameters for least squares cost

If $\lambda$ very large then

$$w_i \sim 0 \text{ for all i}$$

If $\lambda$ is a constant then

$$0 \leq \|\mathbf{w}_{\text{ridge}}\|_2^2 \leq \|\mathbf{w}_{\text{lin}}\|_2^2$$



p-norms:

$L_1$-norm

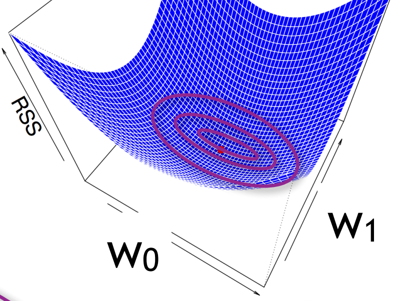
$$\|\mathbf{w}\|_1 = \sum_{i=0}^{d}|w_i|$$

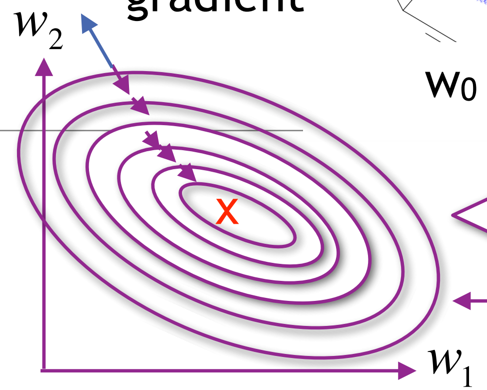
$L_2$-norm

$$\|\mathbf{w}\|_2 = \sqrt{\sum_{i=0}^{d}(w_i)^2}$$

# Geometric Intuition

Level curve/contour line/ Isoline of $||X\mathbf{w}-\mathbf{y}||^2 = c$




contour plot

$E_{\text{in}}(\mathbf{w}) = c$

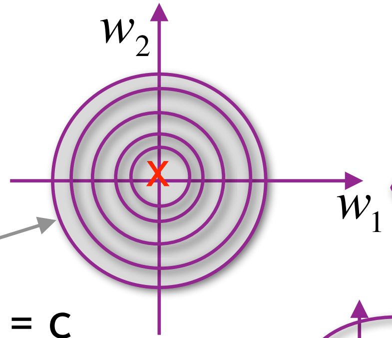
Ellipse

- Looking at the contour plot of RSS

$$\mathrm{E}_{\text{in}}(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{n}(y^{(i)} - \hat{y}^{(i)})^2 = \frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - \mathbf{w}^T\mathbf{x})^2$$

- Looking at the contour plot of the $L_2$ norm

$$\left\|\mathbf{w}_{1:d}\right\|_2^2 = \sum_{i=1}^{d} w_i^2$$

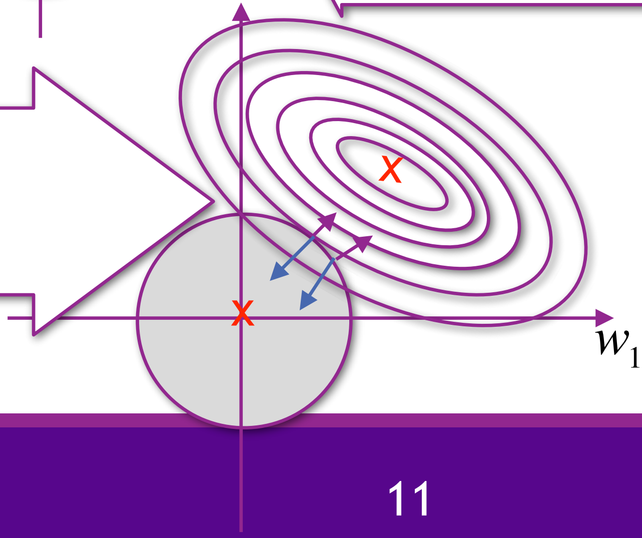



sphere (hypersphere in higher dimensions)

Level curve/contour line/ Isoline of $||\mathbf{w}_{1:d}||^2 = c$

- Looking at $\mathrm{E}_{\text{ridge}}(\mathbf{w}) = \mathrm{E}_{\text{in}}(\mathbf{w}) + \lambda\left\|\mathbf{w}_{1:d}\right\|_2^2$

For each $\lambda$ there is a point which minimizes cost($\mathbf{w}$)



# Remember how in Multiple linear regression we found **w** to minimize $E_{in}$

$$J(\mathbf{w}) = \frac{1}{2N}\sum_{i=1}^{N}((\mathbf{w}^T x^{(i)}) - y^{(i)})^2 \quad = \frac{1}{2N} RSS(\mathbf{w})$$

# Remember: Multiple linear regression
# Closed form solution

$$J(\mathbf{w}) = \frac{1}{2N}\sum_{i=1}^{N}((\mathbf{w}^T x^{(i)}) - y^{(i)})^2 \quad = \frac{1}{2N} RSS(\mathbf{w})$$

An alternative objective function

$$E_{in}(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - \hat{y})^2$$
$$= \frac{1}{N} RSS(\mathbf{w})$$

Goal find $\mathbf{w}_{\text{lin}}$ such that $\nabla J(\mathbf{w}) = \mathbf{0}$ (same $\mathbf{w}$ that makes RSS($\mathbf{w}$) $= \mathbf{0}$ and $\nabla E_{\text{in}}(\mathbf{w}) = \mathbf{0}$

$$\nabla J(\mathbf{w}) = \frac{1}{N} X^T (X\mathbf{w} - \mathbf{y}) = \frac{1}{N}(X^T X\mathbf{w} - X^T \mathbf{y})$$

Setting $\nabla J(\mathbf{w}) = \frac{1}{N}(X^T X\mathbf{w} - X^T \mathbf{y}) = \mathbf{0}$

Results in: $X^T X\mathbf{w} = X^T \mathbf{y}$

Thus $\mathbf{w}_{\text{lin}} = \underbrace{(X^T X)^{-1} X^T}_{\text{pseudoinverse left inverse}} \mathbf{y}$

We added 'lin' to $\mathbf{w}$ to specify it was linear regression

$X^T X$ is a $d \times d$ matrix
$(X^T X)^{-1}$ is a $d \times d$ matrix
$(X^T X)^{-1} X^T$ is a $d \times N$ matrix
$(X^T X)^{-1} X^T \mathbf{y}$ is $d \times 1$

$$\nabla J(\mathbf{w}) = \frac{1}{N}(X^T X\mathbf{w} - X^T \mathbf{y})$$

$$\nabla E_{\text{in}}(\mathbf{w}) = \frac{2}{N}(X^T X\mathbf{w} - X^T \mathbf{y})$$

# Finding $\mathbf{w}$ to minimize $E_{ridge}$

$$E_{\text{ridge}}(\mathbf{w}) = E_{\text{in}}(\mathbf{w}) + \lambda(w_1^2 + w_2^2 + \cdots + w_d^2)$$

Goal find $\mathbf{w}_{\text{ridge}}$ such that $\nabla E_{\text{ridge}}(\mathbf{w}) = \mathbf{0}$

$$E_{in}(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - \hat{y})^2$$

$$J(\mathbf{w}) = \frac{1}{2N}\sum_{i=1}^{N}(y^{(i)} - \hat{y})^2$$

$$\|\mathbf{v}\|^2 = \mathbf{v}^T\mathbf{v} = v_1^2 + v_2^2 + \cdots + v_d^2$$

$$\nabla E_{\text{ridge}}(\mathbf{w}) = \nabla E_{\text{in}}(\mathbf{w}) + \nabla\lambda(\mathbf{w}_{1:d})^T\mathbf{w}_{1:d}$$

$$\nabla E_{\text{in}}(\mathbf{w}) = \frac{2}{N}(X^T X\mathbf{w} - X^T\mathbf{y})$$

$$(\mathbf{w}_{1:d})^T\mathbf{w}_{1:d} = (w_1^2 + w_2^2 + \cdots + w_d^2)$$

$$\frac{\partial(\mathbf{w}_{1:d})^T\mathbf{w}_{1:d}}{\partial w_j} = 2w_j$$

$$\nabla(\mathbf{w}_{1:d})^T\mathbf{w}_{1:d} = 2\mathbf{w}_{1:d}$$

$$\nabla\lambda(\mathbf{w}_{1:d})^T\mathbf{w}_{1:d} = 2\lambda\mathbf{w}_{1:d}$$

Something is wrong! **Pair share**

Using this form so I can rewrite the derivative of $E_{ridge}$ and the dimension still match

$$2\lambda\grave{\mathbf{w}} = 2\lambda\begin{bmatrix}0\\ w_1\\ \vdots\\ w_d\end{bmatrix} = 2\lambda\begin{bmatrix}0 & 0 & \cdots & 0\\ 0 & 1 & \cdots & 0\\ \vdots & \vdots & & \vdots\\ 0 & 0 & \cdots & 1\end{bmatrix}\begin{bmatrix}w_0\\ w_1\\ \vdots\\ w_d\end{bmatrix} = 2\lambda\grave{\mathrm{I}}\mathbf{w}$$

$$\nabla E_{\text{ridge}}(\mathbf{w}) = \frac{2}{N}(X^T X\mathbf{w} - X^T\mathbf{y}) + 2\lambda\,\mathrm{I}\,\grave{\mathbf{w}}$$

# Finding **w** to minimize E_ridge

$$\mathrm{E}_{\text{ridge}}(\mathbf{w}) = \mathrm{E}_{\text{in}}(\mathbf{w}) + \lambda(w_1^2 + w_2^2 + \cdots + w_d^2)$$

---

Goal find $\mathbf{w}_{\text{ridge}}$ such that $\nabla E_{\text{ridge}}(\mathbf{w}) = \mathbf{0}$

Setting $\nabla E_{\text{ridge}}(\mathbf{w}) = \frac{2}{N}(X^T X\mathbf{w} - X^T \mathbf{y}) + 2\lambda\, \mathrm{I}'\, \mathbf{w} \quad = \mathbf{0}$

Results in:

$$X^T X\mathbf{w} + N\lambda\, \mathrm{I}'\, \mathbf{w} = X^T \mathbf{y}$$

$$(X^T X + N\lambda\, \mathrm{I}')\mathbf{w} = X^T \mathbf{y}$$

$$\mathbf{w} = (X^T X + N\lambda\, \mathrm{I}')^{-1} X^T \mathbf{y}$$

Thus $\mathbf{w}_{\text{ridge}} = \underbrace{(X^T X + N\lambda\, \mathrm{I}')^{-1} X^T \mathbf{y}}$

Closed form solution!

Please note that we could have written the regularization as $\lambda/\mathrm{N}\ \mathbf{w}^T\mathbf{w}$ since the need for regularization decreases as the number of training examples increases.

In this case we are minimizing $\mathrm{Ein}(\mathrm{w}) + \lambda/\mathrm{N}\ \mathrm{w}^T\mathrm{w}$

and the closed form solution becomes $(X^TX + \lambda \mathrm{I}')^{-1}X^T\mathbf{y}$

$$\mathbf{w}_{\text{lin}} = \left(X^T X\right)^{-1} X^T \mathbf{y}$$

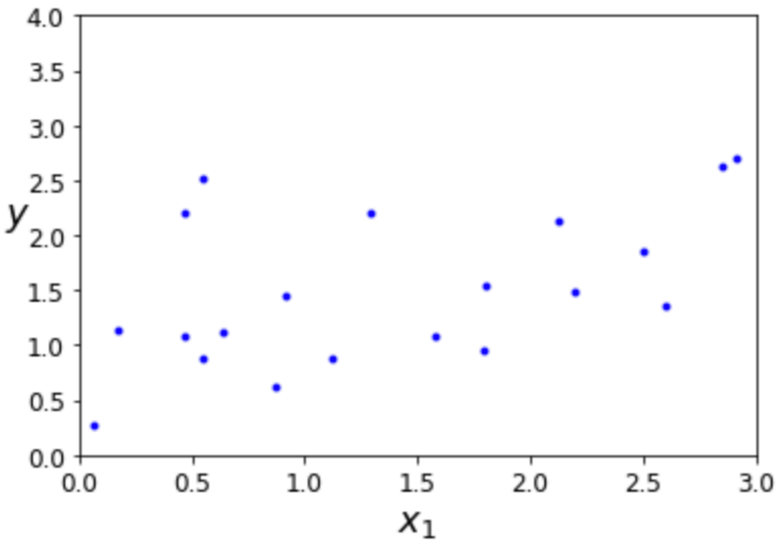
# Example

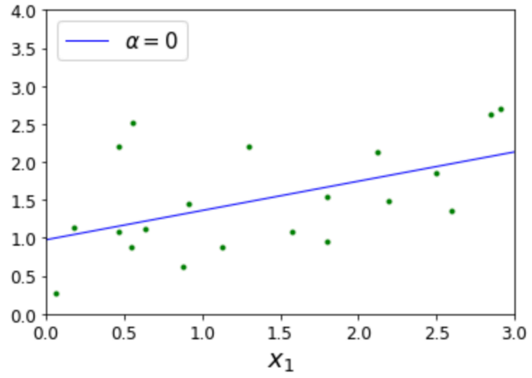
X

```
[[1.    1.12]
 [1.    2.85]
 [1.    2.2 ]
 [1.    1.8 ]
 [1.    0.47]
 [1.    0.47]
 [1.    0.17]
 [1.    2.6 ]
 [1.    1.8 ]
 [1.    2.12]
 [1.    0.06]
 [1.    2.91]
 [1.    2.5 ]
 [1.    0.64]
 [1.    0.55]
 [1.    0.55]
 [1.    0.91]
 [1.    1.57]
 [1.    1.3 ]
 [1.    0.87]]
```

y

```
[[0.89]
 [2.64]
 [1.49]
 [0.96]
 [2.21]
 [1.08]
 [1.13]
 [1.35]
 [1.54]
 [2.14]
 [0.26]
 [2.71]
 [1.85]
 [1.12]
 [0.87]
 [2.51]
 [1.45]
 [1.08]
 [2.2 ]
 [0.62]]
```

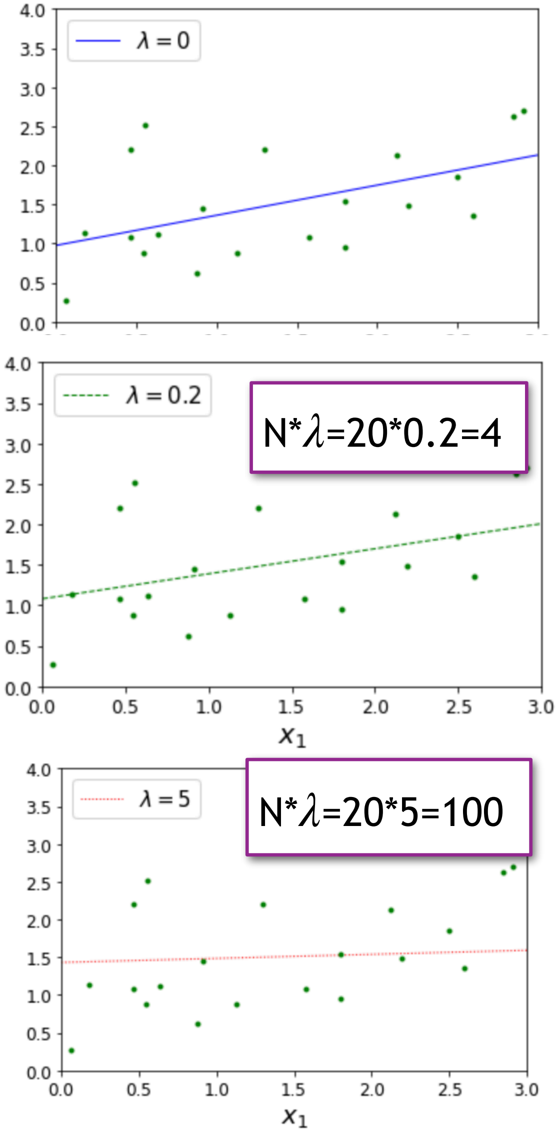



$$\begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = \left[ \begin{bmatrix} 1 & 1.12 \\ 1 & 2.85 \\ 1 & 2.2 \\ 1 & 1.8 \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots \\ 1.12 & 2.85 & 2.2 & 1.8 & \cdots \end{bmatrix} \right]^{-1} \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots \\ 1.12 & 2.85 & 2.2 & 1.8 & \cdots \end{bmatrix} \begin{bmatrix} 0.89 \\ 2.64 \\ 1.49 \\ 0.96 \\ \vdots \end{bmatrix} = \begin{bmatrix} 0.98 \\ 0.39 \end{bmatrix}$$

$$\mathbf{w}_{\text{ridge}} = (X^T X + N\lambda\,\mathrm{I}')^{-1} X^T \mathbf{y}$$

# Example



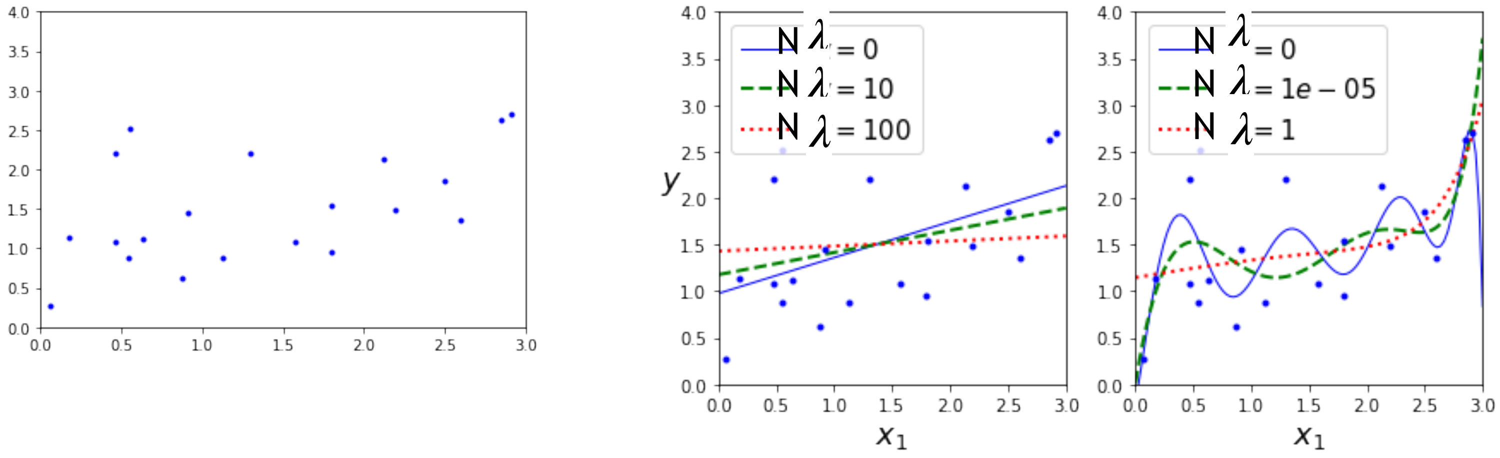

$$\begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = \left[ \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots \\ 1.12 & 2.85 & 2.2 & 1.8 & \cdots \end{bmatrix} \begin{bmatrix} 1 & 1.12 \\ 1 & 2.85 \\ 1 & 2.2 \\ 1 & 1.8 \\ \vdots & \vdots \end{bmatrix} \right]^{-1} \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots \\ 1.12 & 2.85 & 2.2 & 1.8 & \cdots \end{bmatrix} \begin{bmatrix} 0.89 \\ 2.64 \\ 1.49 \\ 0.96 \\ \vdots \end{bmatrix} = \begin{bmatrix} 0.98 \\ 0.39 \end{bmatrix}$$

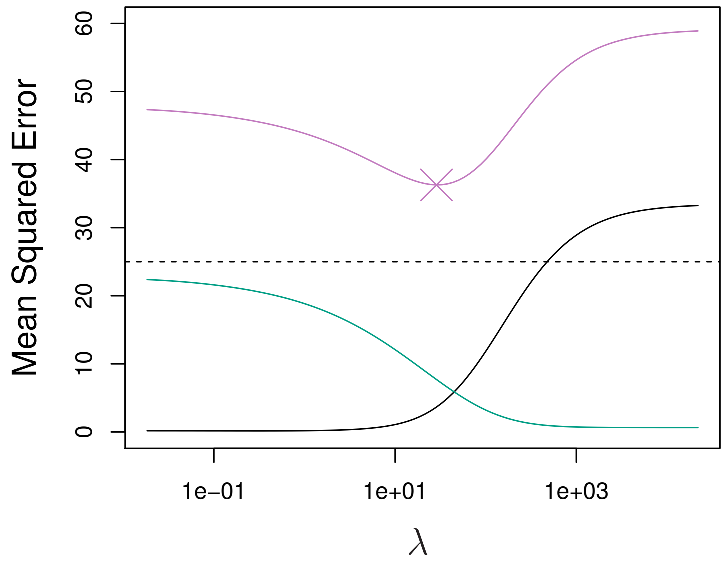
$$\begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = \left[ \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots \\ 1.12 & 2.85 & 2.2 & 1.8 & \cdots \end{bmatrix} \begin{bmatrix} 1 & 1.12 \\ 1 & 2.85 \\ 1 & 2.2 \\ 1 & 1.8 \\ \vdots & \vdots \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 4 \end{bmatrix} \right]^{-1} \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots \\ 1.12 & 2.85 & 2.2 & 1.8 & \cdots \end{bmatrix} \begin{bmatrix} 0.89 \\ 2.64 \\ 1.49 \\ 0.96 \\ \vdots \end{bmatrix} = \begin{bmatrix} 1.08 \\ 0.31 \end{bmatrix}$$

$$\begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = \left[ \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots \\ 1.12 & 2.85 & 2.2 & 1.8 & \cdots \end{bmatrix} \begin{bmatrix} 1 & 1.12 \\ 1 & 2.85 \\ 1 & 2.2 \\ 1 & 1.8 \\ \vdots & \vdots \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 100 \end{bmatrix} \right]^{-1} \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots \\ 1.12 & 2.85 & 2.2 & 1.8 & \cdots \end{bmatrix} \begin{bmatrix} 0.89 \\ 2.64 \\ 1.49 \\ 0.96 \\ \vdots \end{bmatrix} = \begin{bmatrix} 1.43 \\ 0.05 \end{bmatrix}$$

# Sklearn Regularization

The true function is linear f(x) = 1 + 0.5 * X + noise




Example from page 129-131 in Machine Learning with Scikit-Learn and TensorFlow

Figure 6.5 from ISLR
The data was synthetic data
Purple - test MSE
Green - variance
Black - bias (aka bias$^2$)