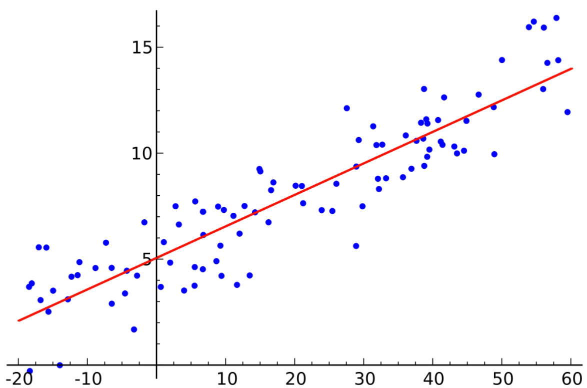
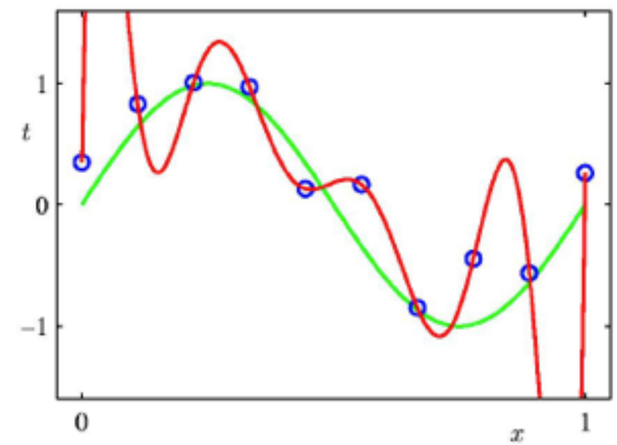
**Tufts COMP 135: Introduction to Machine Learning**
**https://www.cs.tufts.edu/comp/135/2020f/**

# Penalized Linear Regression




Prof. Mike Hughes

*Many slides attributable to:*
*Erik Sudderth (UCI)*
*Finale Doshi-Velez (Harvard)*
*James, Witten, Hastie, Tibshirani (ISL/ESL books)*

# Today's objectives (day 05)

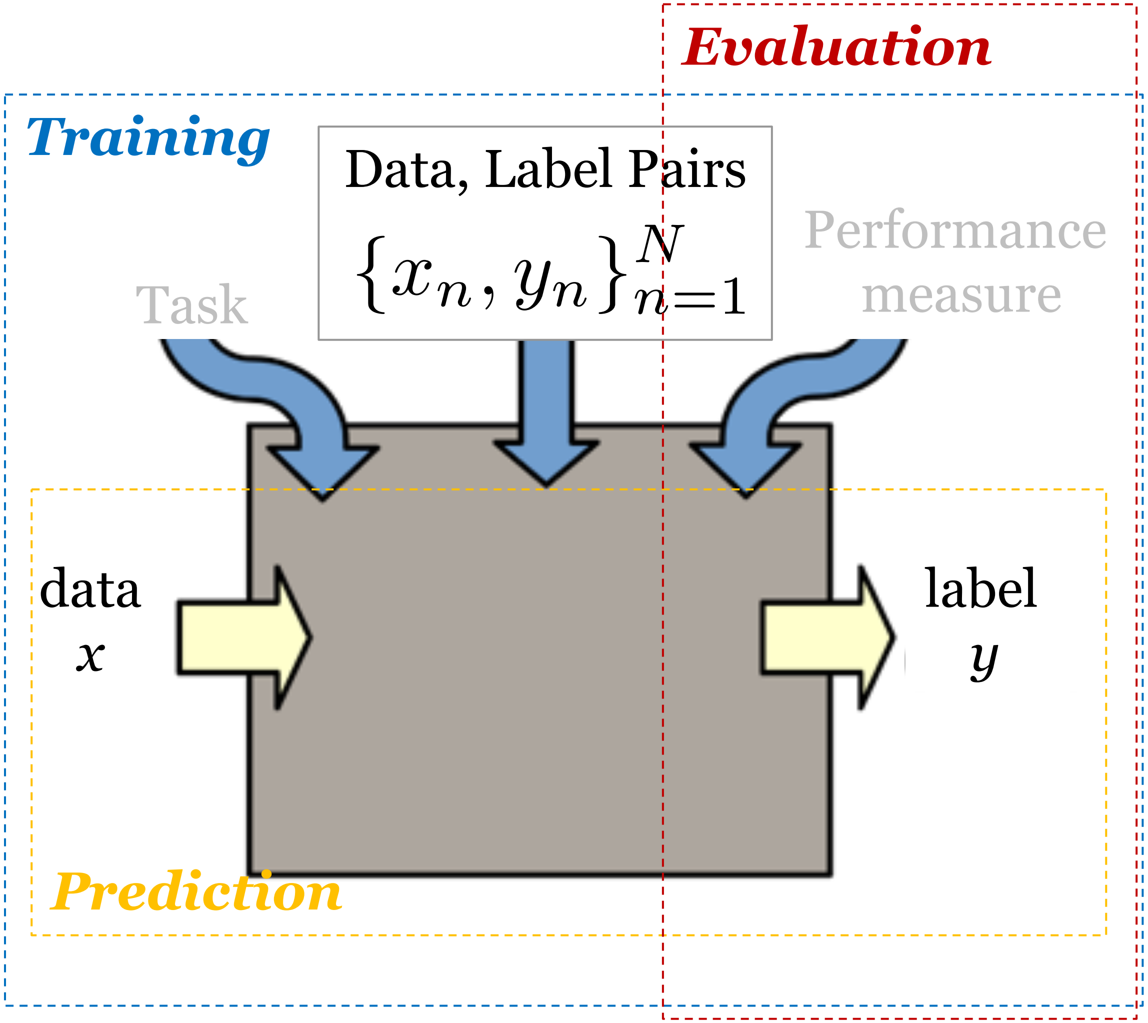
- Recap: Overfitting with high-degree features
- Remedy: Add L2 penalty to the loss ("Ridge")
  - Avoid high magnitude weights
- Remedy: Add L1 penalty to the loss ("Lasso")
  - Avoid high magnitude weights
  - Often, some weights exactly zero (feature selection)

# What will we learn?

Supervised Learning

Unsupervised Learning

Reinforcement Learning



# Task: Regression

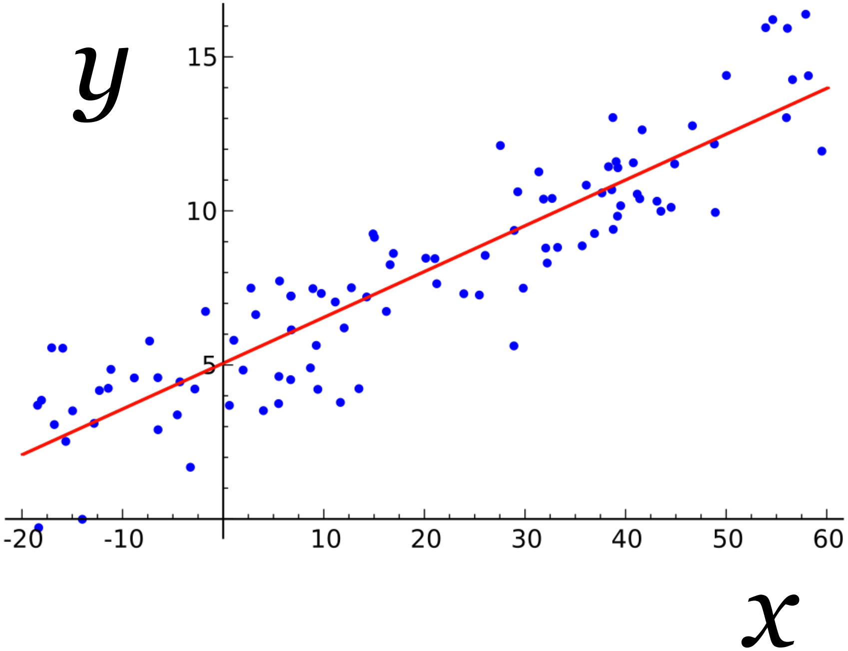
Supervised Learning

regression

Unsupervised Learning

Reinforcement Learning

$y$ is a numeric variable
e.g. sales in $$



# Review: Linear Regression

Optimization problem: "Least Squares"

$$\min_{\theta \in \mathbb{R}^{F+1}} \quad \sum_{n=1}^{N} \left(y_n - \hat{y}(x_n, \theta)\right)^2$$

Exact formula for estimating optimal parameter vector values:

$$\theta := (\tilde{X}^T \tilde{X})^{-1} \tilde{X}^T y$$

$$\tilde{X} = \begin{bmatrix} x_{11} & \dots & x_{1F} & 1 \\ x_{21} & \dots & x_{2F} & 1 \\ & & \dots & \\ x_{N1} & \dots & x_{NF} & 1 \end{bmatrix} \quad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

Can use formula when you observe **at least F+1** examples that are <u>linearly independent</u>
Otherwise, **many theta values** yield lowest possible training error (many linear functions make perfect predictions on the training set)

# Review: Linear Regression with Transformed Features

$$\phi(x_i) = [1 \ \phi_1(x_i) \ \phi_2(x_i) \ldots \phi_{G-1}(x_i)]$$

$$\hat{y}(x_i) = \theta^T \phi(x_i)$$

Optimization problem: "Least Squares"

$$\min_\theta \sum_{n=1}^N (y_n - \theta^T \phi(x_i))^2$$

Exact solution:

$$\theta^* = (\Phi^T \Phi)^{-1} \Phi^T y$$

*G x 1 vector*

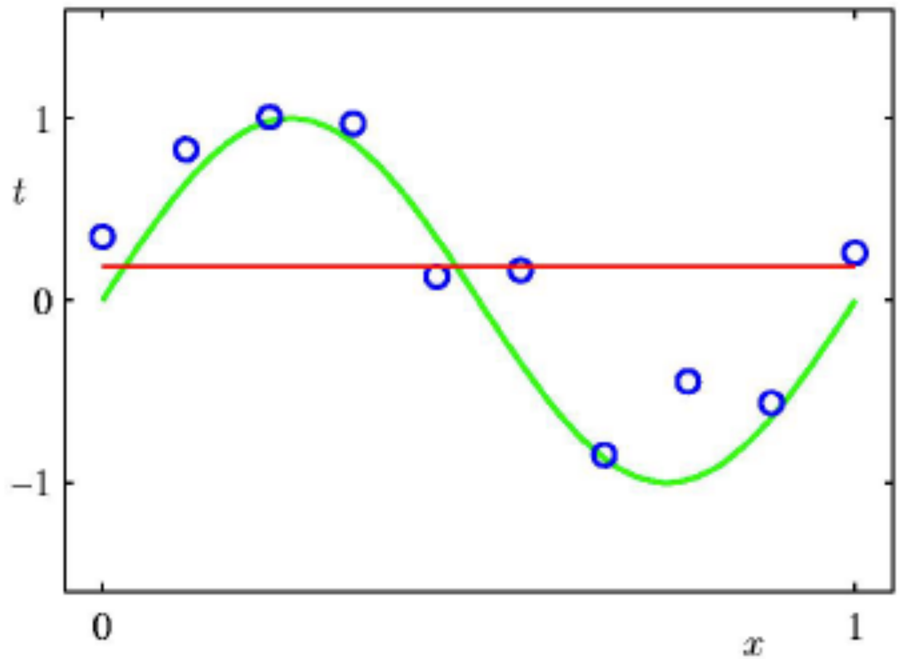
$$\Phi = \begin{bmatrix} 1 & \phi_1(x_1) & \ldots & \phi_{G-1}(x_1) \\ 1 & \phi_1(x_2) & \ldots & \phi_{G-1}(x_2) \\ \vdots & & \ddots & \\ 1 & \phi_1(x_N) & \ldots & \phi_{G-1}(x_N) \end{bmatrix}$$

*N x G matrix*

# 0$^{th}$ degree polynomial features

$$\phi(x_i) = [1]$$

# parameters:
G = 1




--- true function

o training data

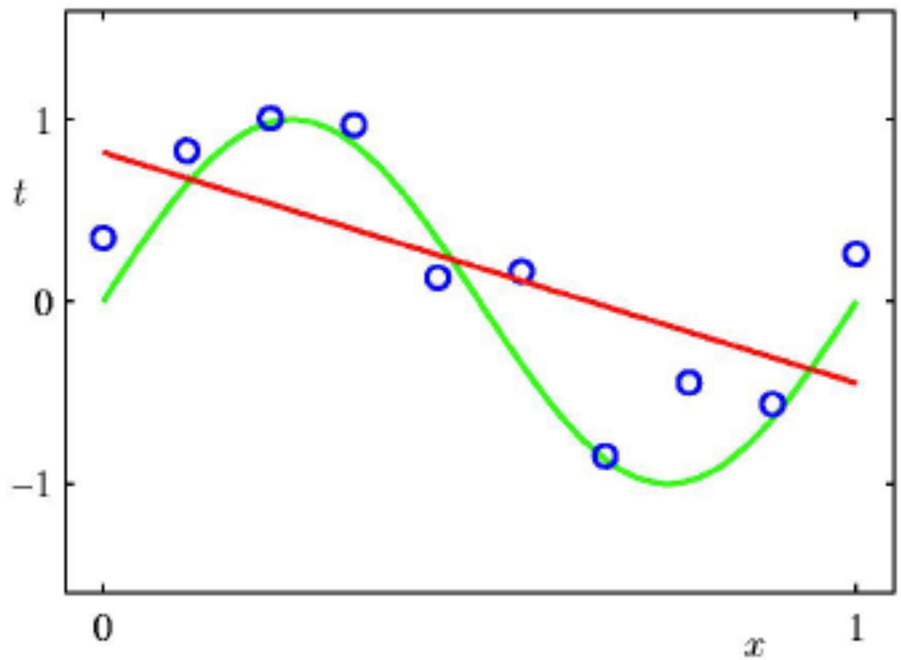
-- predictions from LR using polynomial features

Credit: Slides from course by Prof. Erik Sudderth (UCI)

# $1^{st}$ degree polynomial features

$$\phi(x_i) = [1 \;\; x_{i1}]$$

# parameters:
G = 2




--- true function

o training data

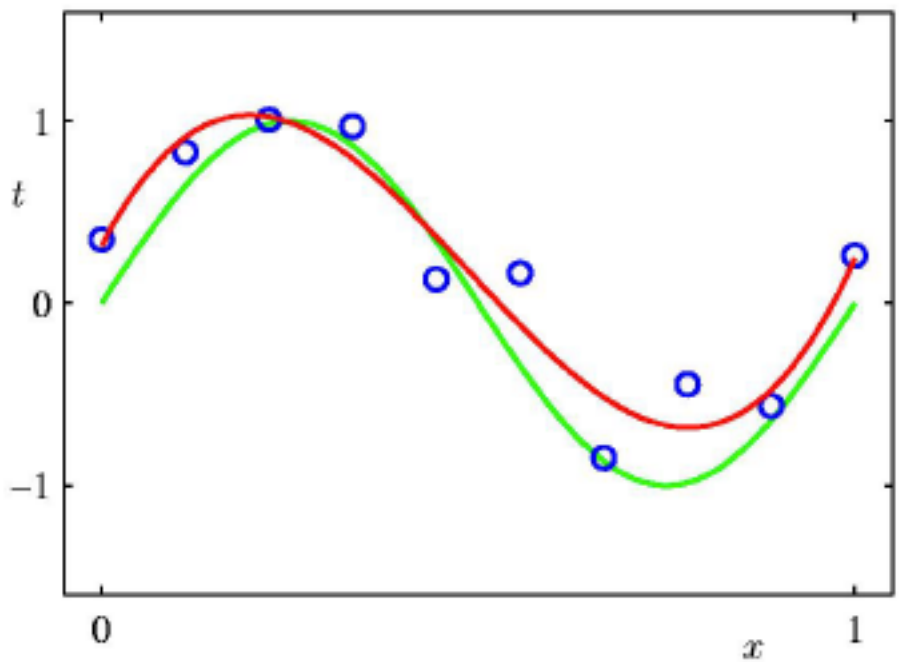
-- predictions from LR using polynomial features

Credit: Slides from course by Prof. Erik Sudderth (UCI)

# 3rd degree polynomial features

$$\phi(x_i) = [1 \quad x_{i1} \quad x_{i1}^2 \quad x_{i1}^3]$$

# parameters:
G = 4




--- true function

o training data

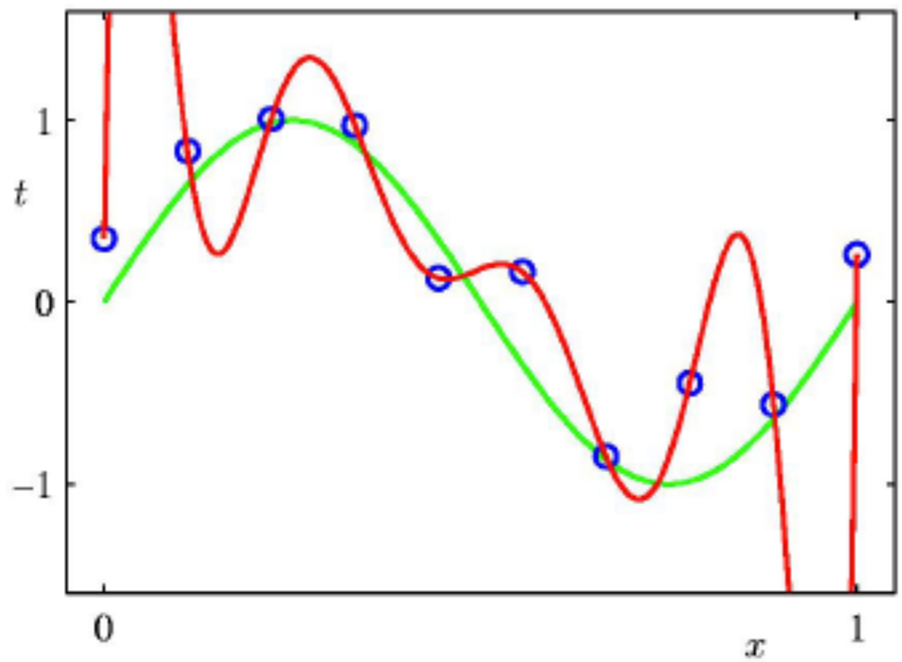
-- predictions from LR using polynomial features

Credit: Slides from course by Prof. Erik Sudderth (UCI)

# $9^{\text{th}}$ degree polynomial features

$$\phi(x_i) = [1 \quad x_{i1} \quad x_{i1}^2 \quad x_{i1}^3 \quad x_{i1}^4 \quad x_{i1}^5 \quad x_{i1}^6 \quad x_{i1}^7 \quad x_{i1}^8 \quad x_{i1}^9]$$

# parameters:
G = 10



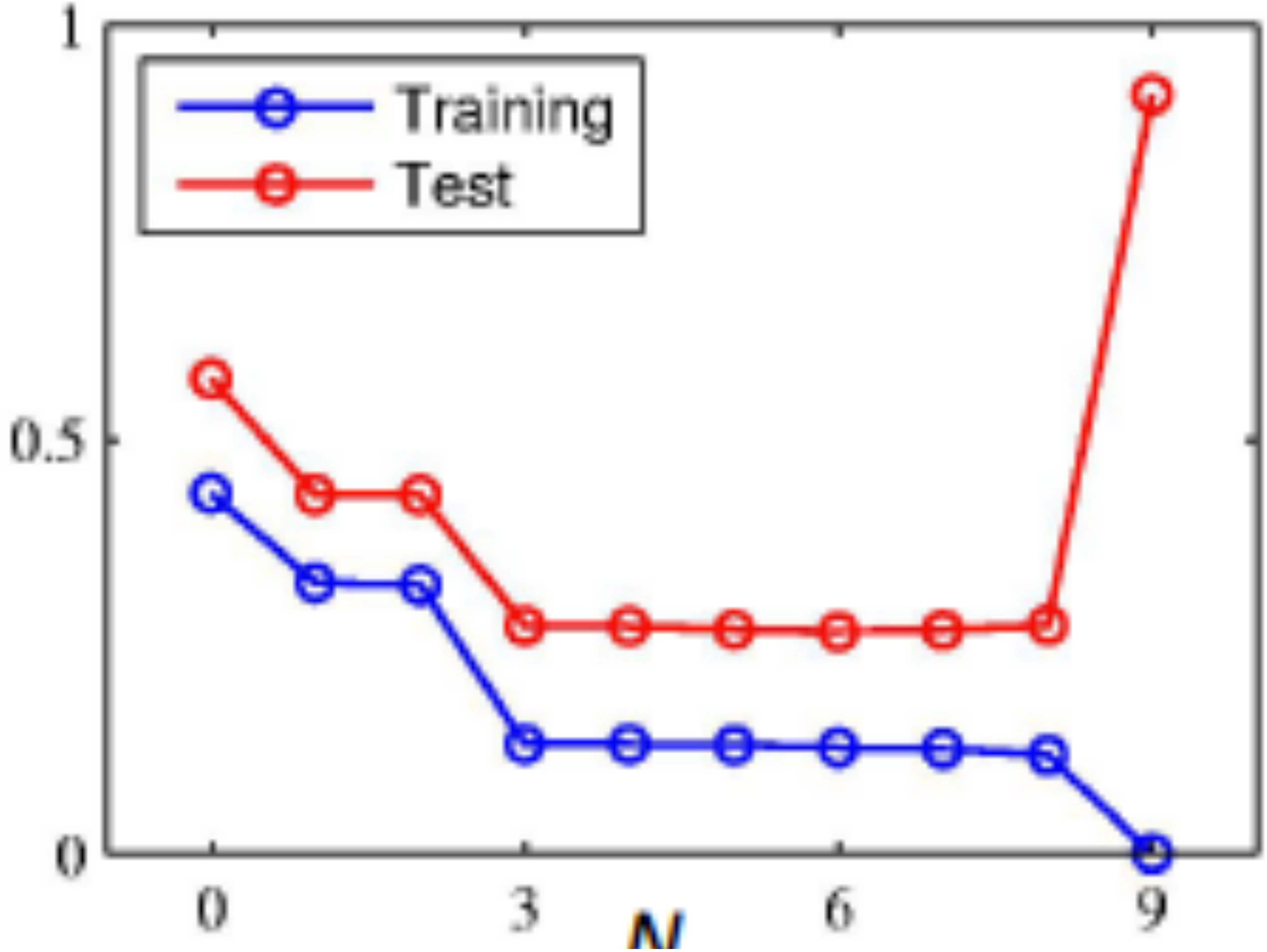

--- true function

o training data

-- predictions from LR using polynomial features

Credit: Slides from course by Prof. Erik Sudderth (UCI)

# Error vs Complexity




mean squared error

polynomial degree

# Weight Values vs Complexity

Estimated Regression Coefficients by Polynomial degree

| 0 | 1 | 3 | 9 |
|---|---|---|---|
| 0.19 | 0.82 | 0.31 | 0.35 |
| | -1.27 | 7.99 | 232.37 |
| | | -25.43 | -5321.83 |
| | | 17.37 | 48568.31 |
| | | | -231639.30 |
| | | | 640042.26 |
| | | | -1061800.52 |
| | | | 1042400.18 |
| | | | -557682.99 |
| | | | 125201.43 |

**WOW! These values are very large.**

Credit: Slides from course by Prof. Erik Sudderth (UCI)

# Idea: Add Penalty Term to Loss

Goal: Avoid finding weights with large magnitude
Result: **Ridge regression**, a method with objective:

$$J(\theta) = \sum_{n=1}^{N} (y_n - \theta^T \phi(x_n))^2 + \alpha \sum_{g=1}^{G} \theta_g^2$$

Penalty term: Sum of squares of entries of theta
= Square of the "L2 norm" of theta vector
Thus, also called "L2-penalized" linear regression

**Hyperparameter**: Penalty strength "alpha" $\alpha \geq 0$

Alpha = 0 recovers original unpenalized Linear Regression
Larger alpha means we prefer smaller magnitude weights

# Rewrite in matrix notation?

N : num. examples
G : num transformed features

$$J(\theta) = \sum_{n=1}^{N} (y_n - \theta^T \phi(x_n))^2 + \alpha \sum_{g=1}^{G} \theta_g^2$$

*Rewriting, this is equivalent to*

Can rewrite sum of squares as an inner product of theta vector with itself

$$J(\theta) = (y - \Phi\theta)^T (y - \Phi\theta) + \alpha \theta^T \theta$$

$$\Phi = \begin{bmatrix} 1 & \phi_1(x_1) & \dots & \phi_{G-1}(x_1) \\ 1 & \phi_1(x_2) & \dots & \phi_{G-1}(x_2) \\ \vdots & & \ddots & \\ 1 & \phi_1(x_N) & \dots & \phi_{G-1}(x_N) \end{bmatrix}$$

N x G

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

N x 1

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_G \end{bmatrix}$$

G x 1

# Estimating weights for L2 penalized linear regression

Optimization problem: "Penalized Least Squares"

$$\min_{\theta} \ (y - \Phi\theta)^T (y - \Phi\theta) + \alpha \theta^T \theta$$

Solution:

$$\theta^* = (\Phi^T \Phi + \alpha \underbrace{I_G}_{\substack{G \times G \\ \text{identity matrix}}})^{-1} \Phi^T y$$

If alpha > 0 , the matrix is **always** invertible!
Always one unique optimal theta vector, provided by this formula

# What happens if we rescale a feature in **unpenalized** LR?

Suppose we changed units on "volume" of the engine feature, from liters to mL

Remember, 1 liter = 1000 mL

*Answer: Just "rescale" the individual weight for that single feature.*
***No other weights will change.***

$x_{:1}$ $x_{:2}$ $y$

**Before**

| vol_in_**L** | hp | mi_per_gal |
|---|---|---|
| 2.1 | 115.0 | 30.0 |
| 2.3 | 150.0 | 25.0 |
| 2.5 | 193.0 | 21.2 |

| | |
|---|---|
| **vol** | **[-51.25 ]** |
| hp | [ 0.15 ] |
| bias | [120.375] |

**After**

| vol_in_**mL** | hp | mi_per_gal |
|---|---|---|
| 2100 | 115.0 | 30.0 |
| 2300 | 150.0 | 25.0 |
| 2500 | 193.0 | 21.2 |

**[ -0.05125 ]**
[ 0.15 ]
[120.375 ]

# What happens if we rescale a feature in penalized LR?

**alpha = 0.01**

Suppose we changed units on "volume" of the engine feature, from liters to mL

Remember, 1 liter = 1000 mL

*Answer: Because all weights contribute to the penalty term,* ***ALL learned weights change***

**Before**

| $x_{:1}$ | $x_{:2}$ | $y$ |
|---|---|---|
| vol_in_**L** | hp | mi_per_gal |
| 2.1 | 115.0 | 30.0 |
| 2.3 | 150.0 | 25.0 |
| 2.5 | 193.0 | 21.2 |

```
[ 18.428 ]
[ -0.199]
[ 13.407]
```

**After**

| vol_in_**mL** | hp | mi_per_gal |
|---|---|---|
| 2100 | 115.0 | 30.0 |
| 2300 | 150.0 | 25.0 |
| 2500 | 193.0 | 21.2 |

```
[  0.027  ]
[ -0.248  ]
[  1.439  ]
```

# Ridge Regression is **more sensitive** to the scale of your features.

Before feeding data into a Ridge regression model, should standardize the scale of all features, so the penalty acts on each feature in more uniform way.

- Rescale each column between 0 and 1
  - sklearn's MinMaxScaler
  - https://scikit-learn.org/stable/modules/preprocessing.html#scaling-features-to-a-range

- Transform each column to have mean 0 and variance 1
  - sklearn's StandardScaler
  - https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html#sklearn.preprocessing.StandardScaler

OR, you can impose your own feature-specific penalties if you want.

# Lasso Regression (L1 penalty)

N : num. examples
G : num transformed features

$$\min_{\theta} \quad (y - \Phi\theta)^T (y - \Phi\theta) + \alpha \sum_{g=1}^{G} |\theta_g|$$

Sum of absolute values of entries (aka the L1 norm of the vector theta)

Like L2 penalty (Ridge), the Lasso objective above encourages small magnitude weights.

*We'll see in lab*: L1 penalty encourages theta to be a **sparse vector**
At modest alpha values, many entries of theta could be **exactly zero**.
Can use for feature selection (only features with non-zero weights matter)

# Today's objectives (day 05)

- Recap: Overfitting with high-degree features
- Remedy: Add L2 penalty to the loss ("Ridge")
  - Avoid high magnitude weights
- Remedy: Add L1 penalty to the loss ("Lasso")
  - Avoid high magnitude weights
  - Often, some weights exactly zero (feature selection)