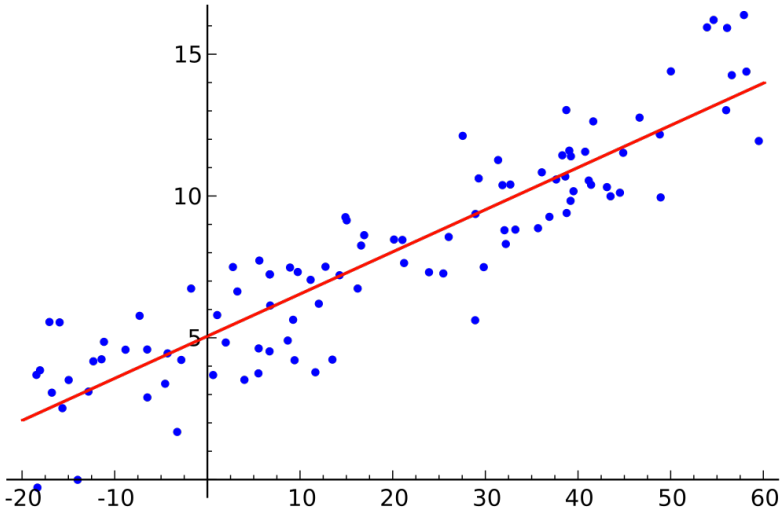
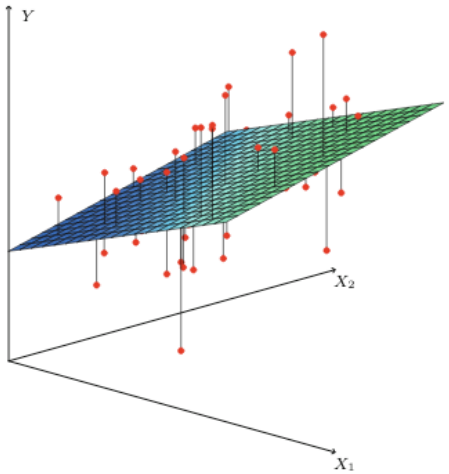
Tufts COMP 135: Introduction to Machine Learning
https://www.cs.tufts.edu/comp/135/2019s/

Linear Regression





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

Prof. Mike Hughes

Objectives for Today (day 03)

- Training "least squares" linear regression
  - Simplest case: 1-dim. features without intercept
  - Simple case: 1-dim. features with intercept
  - General case: Many features with intercept

- Concepts (algebraic and graphical view)
  - Where do formulas come from?
  - When are optimal solutions unique?

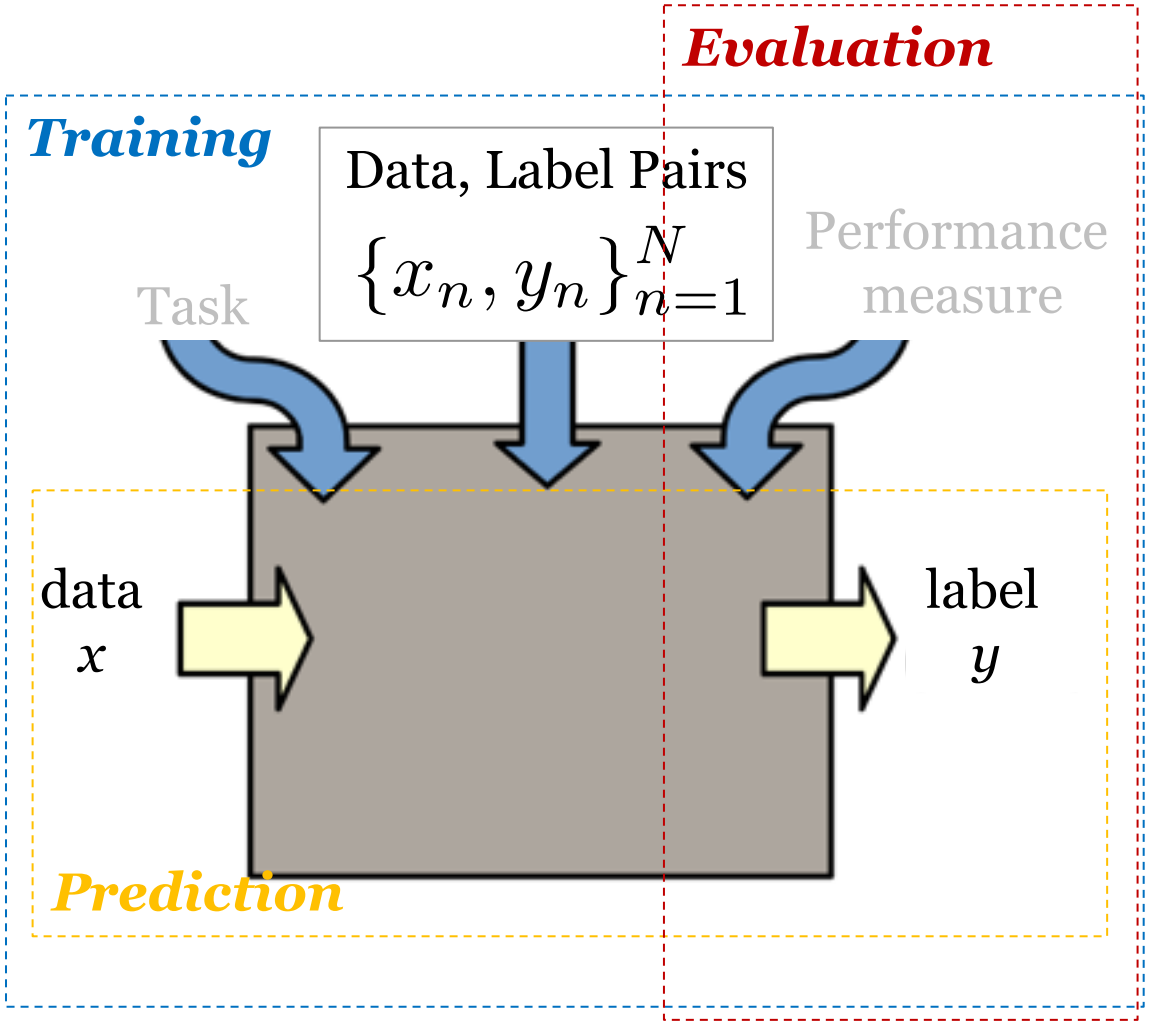
- Programming:
  - How to solve linear systems in Python
    - Hint: use **np.linalg.solve**; avoid **np.linalg.inv**

What will we learn?

Supervised Learning

Unsupervised Learning

Reinforcement Learning



Task: Regression

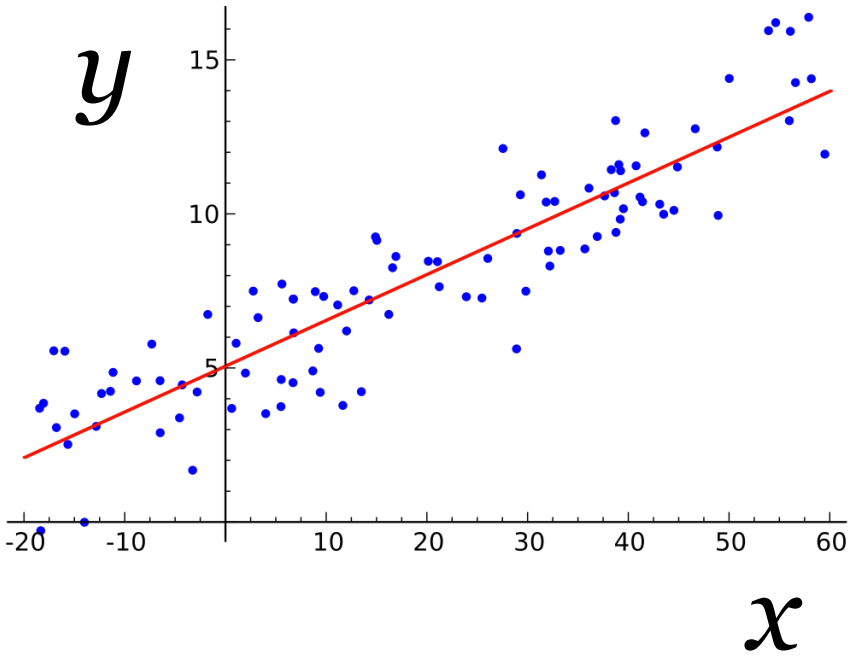
Supervised Learning

regression

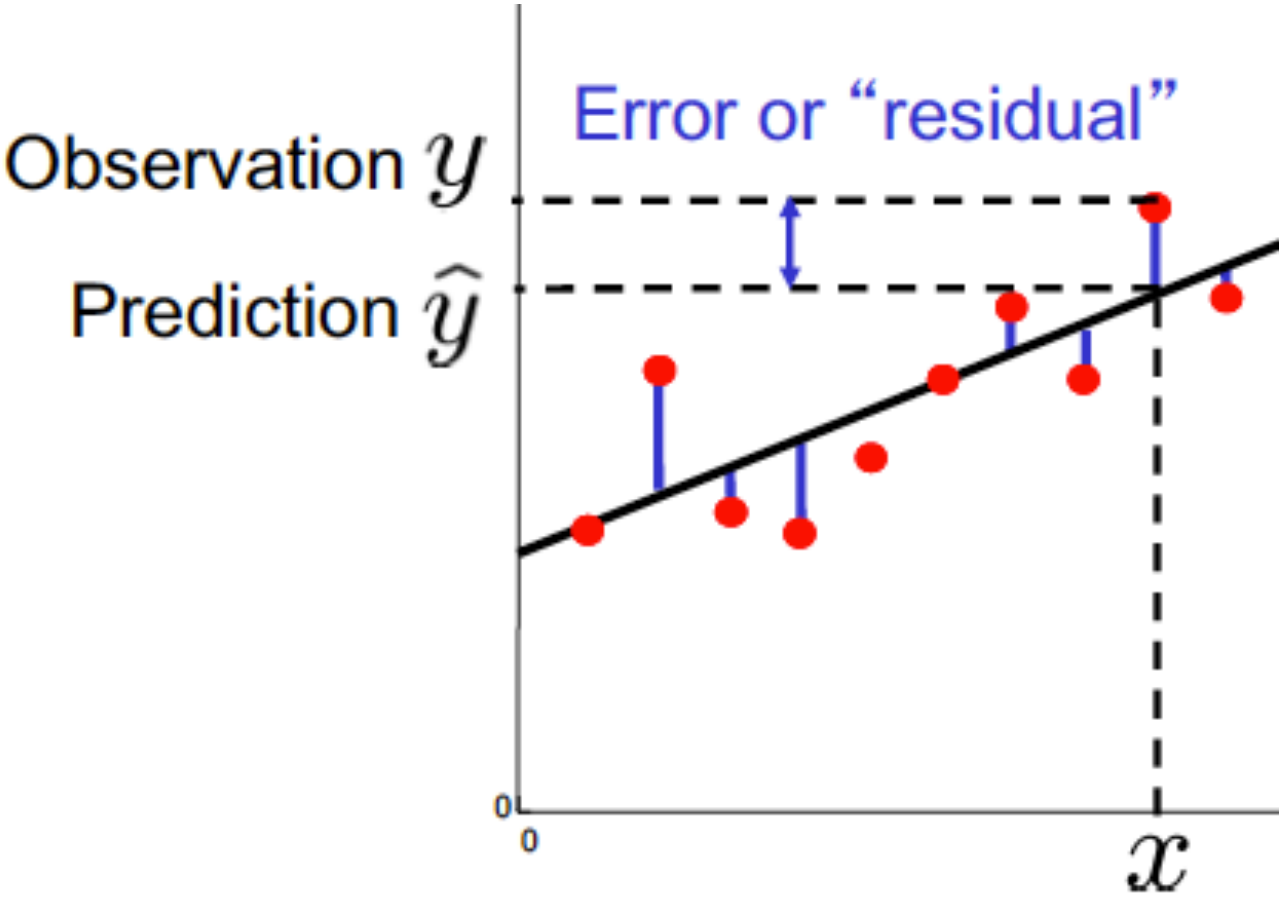
Unsupervised Learning

Reinforcement Learning

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



Visualizing errors



Evaluation Metrics for Regression

- mean squared error $\frac{1}{N} \sum_{n=1}^{N} (y_n - \hat{y}_n)^2$

- mean absolute error $\frac{1}{N} \sum_{n=1}^{N} |y_n - \hat{y}_n|$

Today, we'll focus on mean squared error (MSE). Mean squared error is **smooth everywhere**. Good analytical properties and widely studied. Thus, it is a common choice.

NB: Many applications, absolute error (or other error metrics) may be more suitable, if computational or analytical convenience was not the chief concern.

Linear Regression
1-dim features, no bias

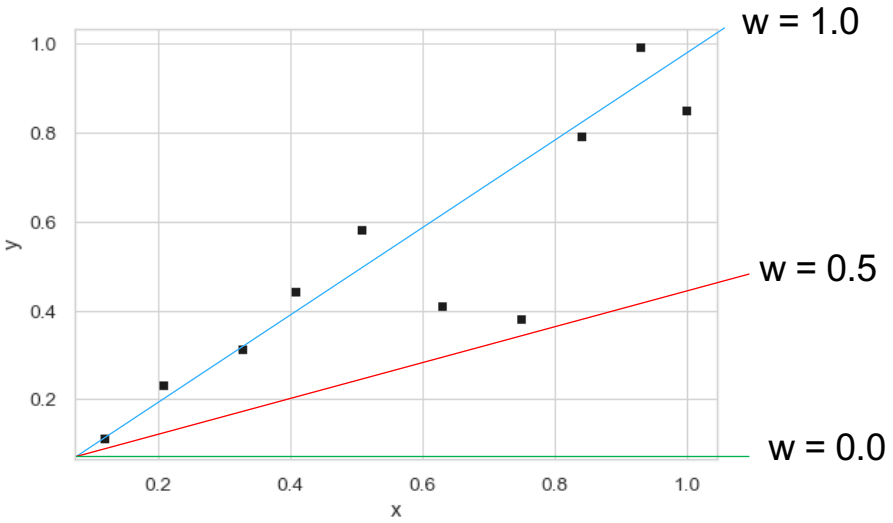
Parameters:

weight scalar w

Prediction:

$$\hat{y}(x_i) \triangleq w \cdot x_{i1}$$

Graphical interpretation: Pick a line with slope w that *goes through the origin*



Training:

Input: training set of N observed examples of features x and responses y
Output: value of w that minimizes **mean squared error** on training set.

Training for 1-dim, no-bias LR

Training objective: minimize squared error ("least squares" estimation)

$$\min_{w \in \mathbb{R}} \sum_{n=1}^{N} (y_n - \hat{y}(x_n, w))^2$$

Formula for parameters that minimize the objective:

$$w^* = \frac{\sum_{n=1}^{N} y_n x_n}{\sum_{n=1}^{N} x_n^2}$$

When can you use this formula?

When you observe **at least 1** example with **non-zero** features

Otherwise, *all possible w values* will be perfect (zero training error)

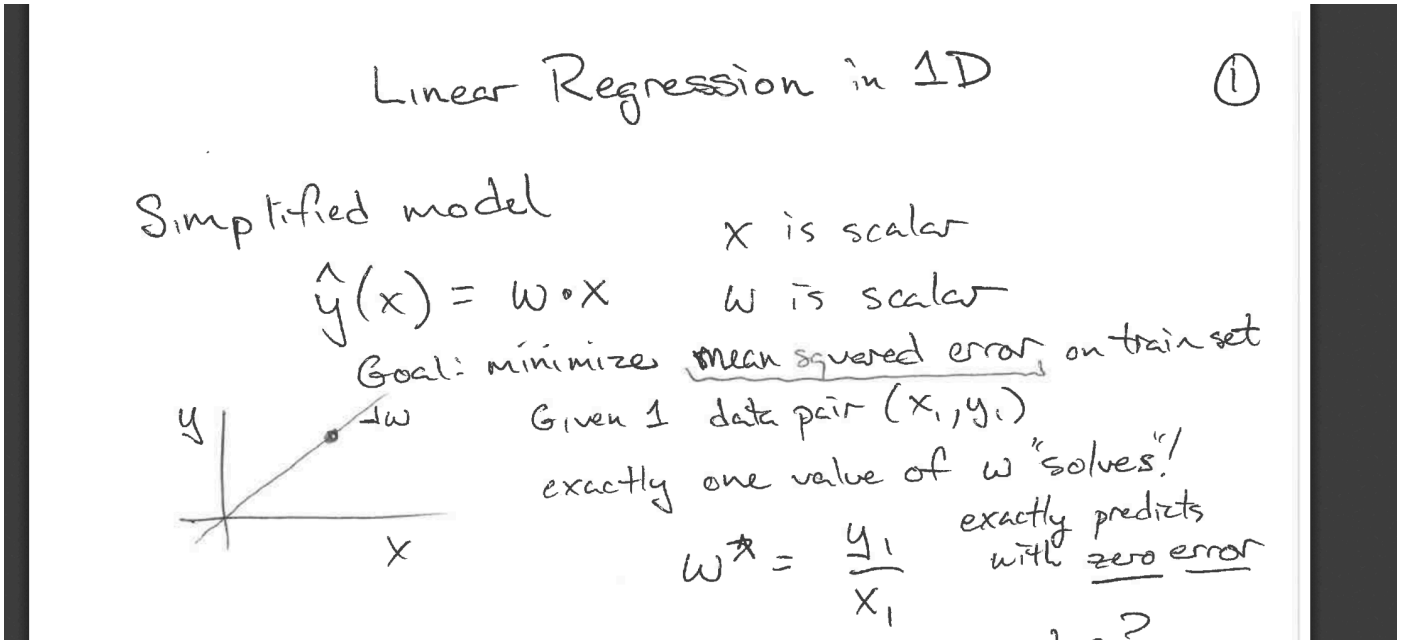
Why? all lines in our hypothesis space go through origin.

How to derive the formula (see notes):

1. Compute gradient of objective, as a function of w
2. Set gradient equal to zero and solve for w

For details, see derivation notes

https://www.cs.tufts.edu/comp/135/2020f/notes/day03_linear_regression.pdf



Linear Regression
1-dim features, with bias

Parameters:

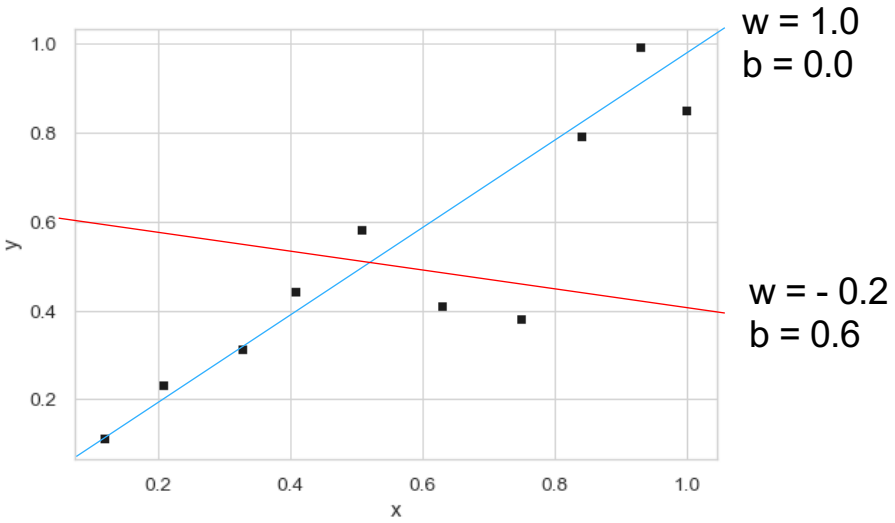
weight scalar w

bias scalar b

Prediction:

$$\hat{y}(x_i) \triangleq w \cdot x_{i1} + b$$

Graphical interpretation:
Predict along line with slope w and intercept b



Training:

Input: training set of N observed examples of features *x* and responses *y*
Output: values of *w* and *b* that minimize **mean squared error** on training set.

Training for 1-dim, with-bias LR

Training objective: minimize squared error ("least squares" estimation)

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

Formula for parameters that minimize the objective:

$$w = \frac{\sum_{n=1}^{N}(x_n - \bar{x})(y_n - \bar{y})}{\sum_{n=1}^{N}(x_n - \bar{x})^2}$$

$$b = \bar{y} - w\bar{x}$$

$$\bar{x} = \text{mean}(x_1, \ldots x_N)$$

$$\bar{y} = \text{mean}(y_1, \ldots y_N)$$

When can you use this formula?

When you observe **at least 2** examples with distinct 1-dim. features

Otherwise, **many** *w, b* will be perfect (lowest possible training error)

Why? many lines in our hypothesis space go through one point

How to derive the formula (see notes):

1. Compute gradient of objective wrt w, as a function of w and b
2. Compute gradient of objective wrt b, as a function of w and b
3. Set (1) and (2) equal to zero and solve for w and b (2 equations, 2 unknowns)

Linear Regression
F-dim features, with bias

Parameters:

weight vector $w = [w_1, w_2, \ldots w_F]$

bias scalar b

Prediction:

$$\hat{y}(x_i) \triangleq \sum_{f=1}^{F} w_f x_{if} + b$$

Graphical interpretation:
Predict along one plane in F+1-dim. space

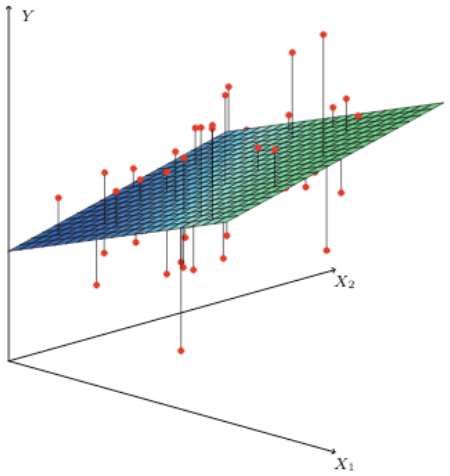


FIGURE 3.4. *In a three-dimensional setting, with two predictors and one response, the least squares regression line becomes a plane. The plane is chosen to minimize the sum of the squared vertical distances between each observation (shown in red) and the plane.*

Training:

Input: training set of N observed examples of features *x* and responses *y*
Output: values of *w* and *b* that minimize **mean squared error** on training set.

Training for F-dim, with-bias LR

Training objective: minimize squared error ("least squares" estimation)

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

Formula for parameters that minimize the objective:

$$[w_1 \ldots w_F \ b]^T = (\tilde{X}^T \tilde{X})^{-1} \tilde{X}^T y$$

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

When can you use this formula?

When you observe **at least F+1** examples that are linearly independent

Otherwise, *infinitely many w, b* will yield lowest possible training error

How to derive the formula (see notes):

1. Compute gradient of objective wrt each entry of w, and wrt scalar b (F+1 total expressions)
2. Set all gradients equal to zero and solve for w and b (F+1 equations, F+1 unknowns)

More compact notation

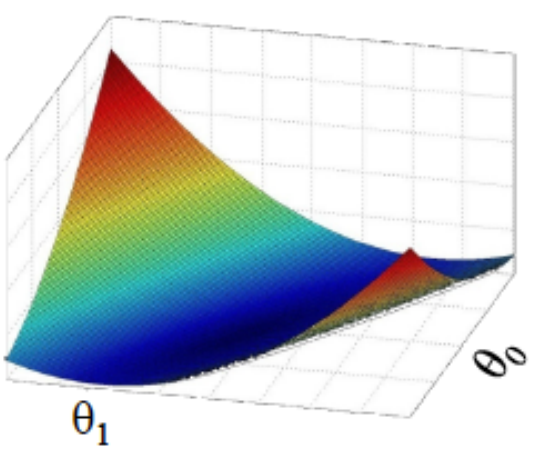
$$\theta = [b \; w_1 \; w_2 \ldots w_F]$$

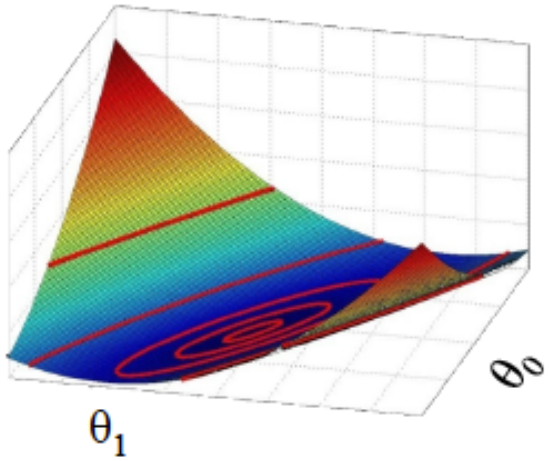
$$\tilde{x}_n = [1 \; x_{n1} \; x_{n2} \ldots x_{nF}]$$

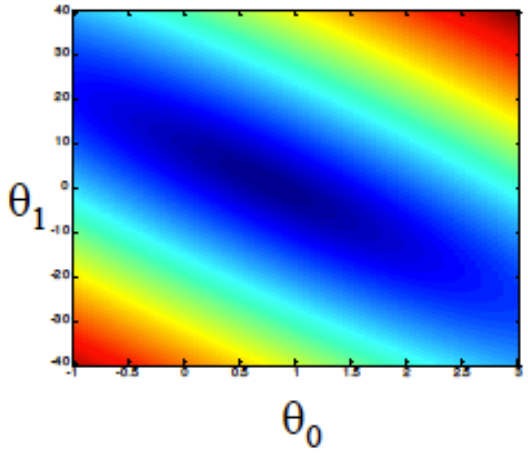
$$\hat{y}(x_n, \theta) = \theta^T \tilde{x}_n$$

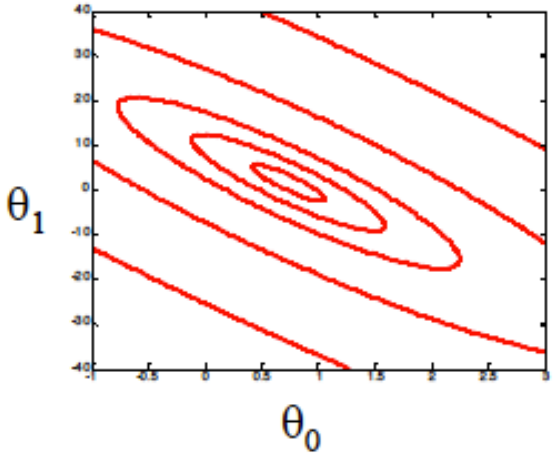
$$J(\theta) \triangleq \sum_{n=1}^{N} (y_n - \hat{y}(x_n, \theta))^2$$

Visualizing the cost function









"Level set" contours : all points with same function value

Breakout!

- Do the day03 lab!

- Ask questions in Live Q&A on Piazza