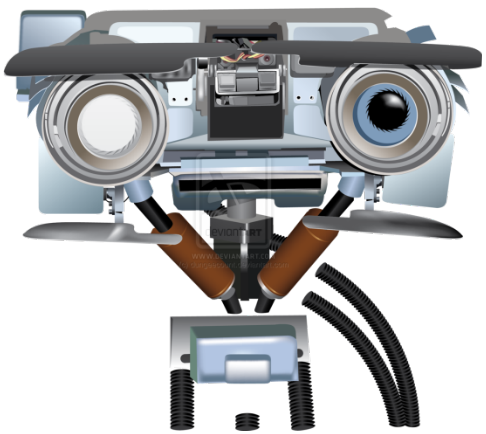
Today

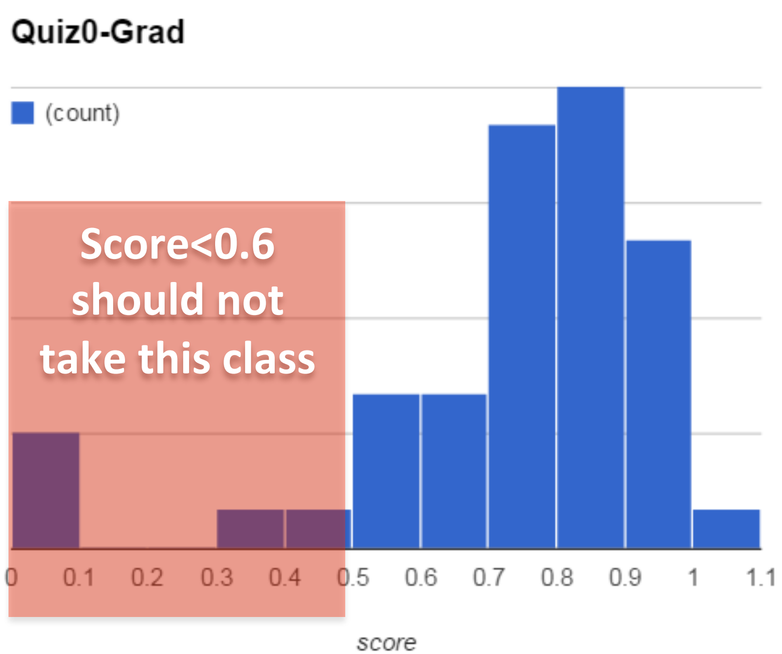
- Application spotlight: Games
- Supervised Learning: Linear Regression

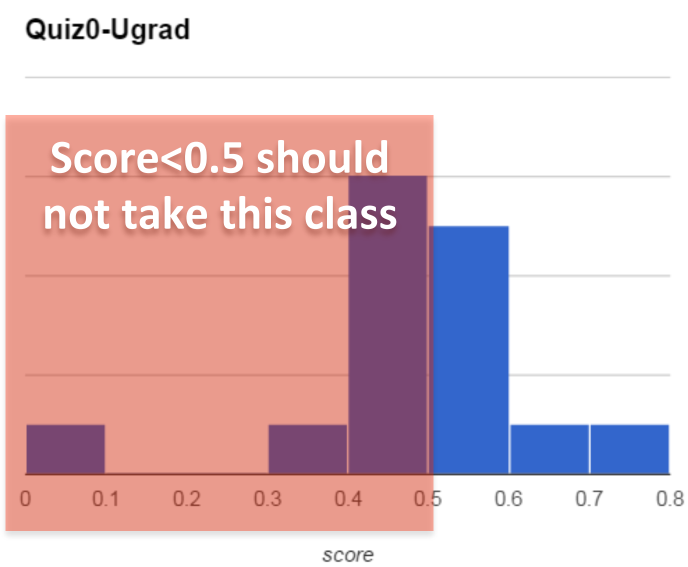


Announcements

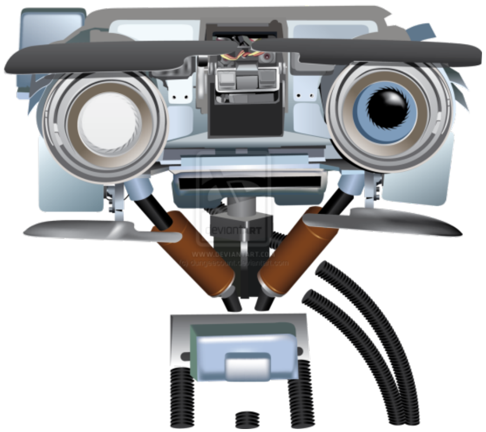
- Pset 1 released, due in one week
- Quiz0 graded

Quiz0 Grades





- If your score is below threshold (60% for grad, 50% for undergrad), you should not be taking this class
- Drop before deadline **Monday Feb 2**



Application Spotlight

AI playing games

Poker playing bot can beat any human

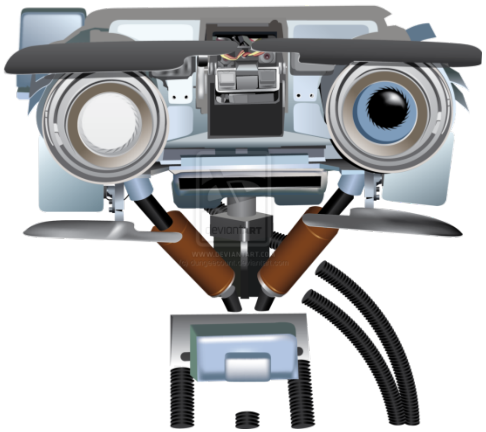
- no human expert help, only Texas Hold’em rules
- computer played against itself, playing more games than played by entire human race
- optimal strategy found using 5,000 CPUs

NPR Article, Science paper

AI playing games

- computers also beat humans at chess, checkers, backgammon and Jeopardy!
- Different, specialized strategy for each
- Far behind human brain, which is an all-purpose game playing “machine”

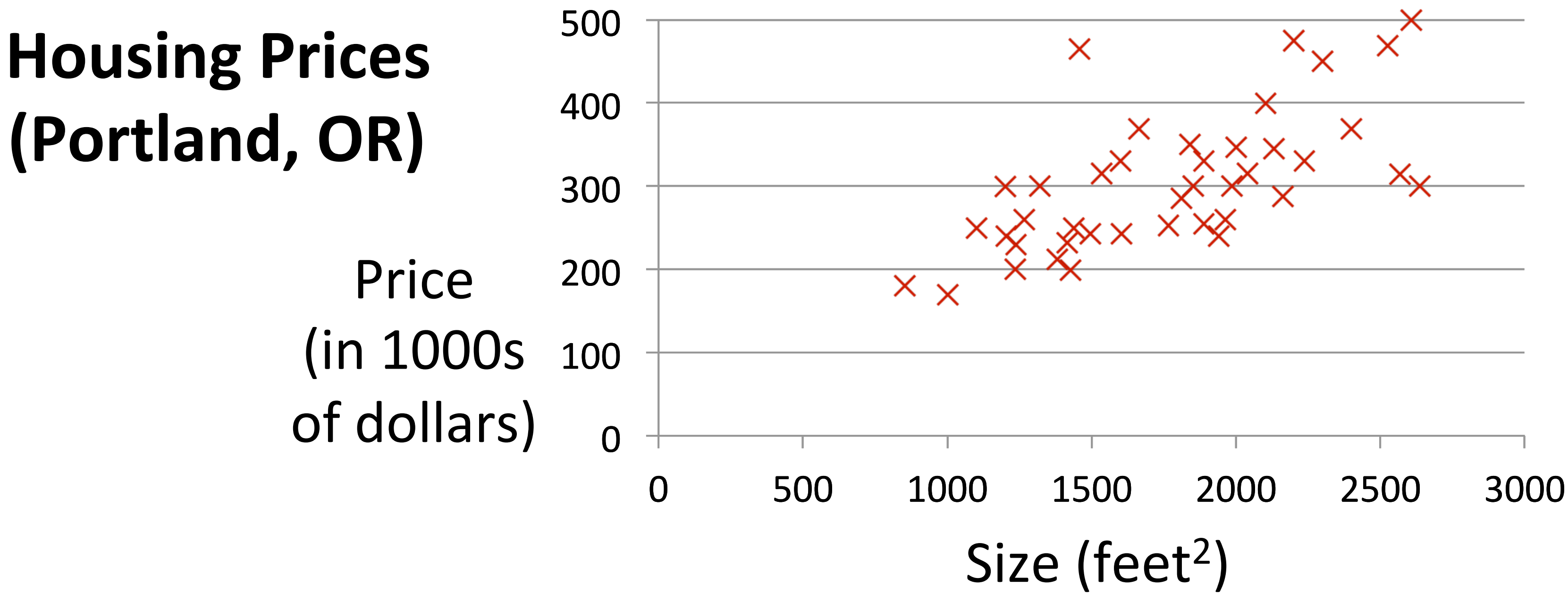
[IBM's Watson playing Jeopardy!]



Linear Regression

Kate Saenko

Example: house price prediction



<u>Supervised Learning</u>

Given the "right answer" for each example in the data.

<u>Regression Problem</u>

Predict real-valued output

Slide credit: Andrew Ng

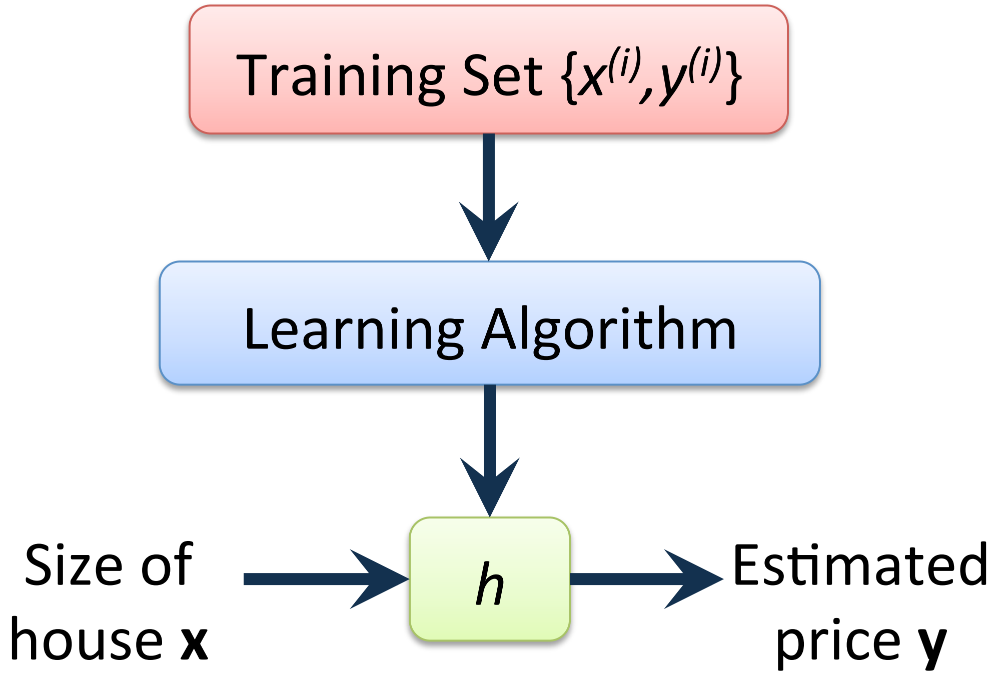
Training set

Size in feet^2 (x)	**Price ($) in 1000's (y)**
2104	460
1416	232
1534	315
852	178
…	…

Notation:

m = Number of training examples
$x^{(i)}$ = "input" variable / features
$y^{(i)}$ = "output" variable / "target" variable

Hypothesis

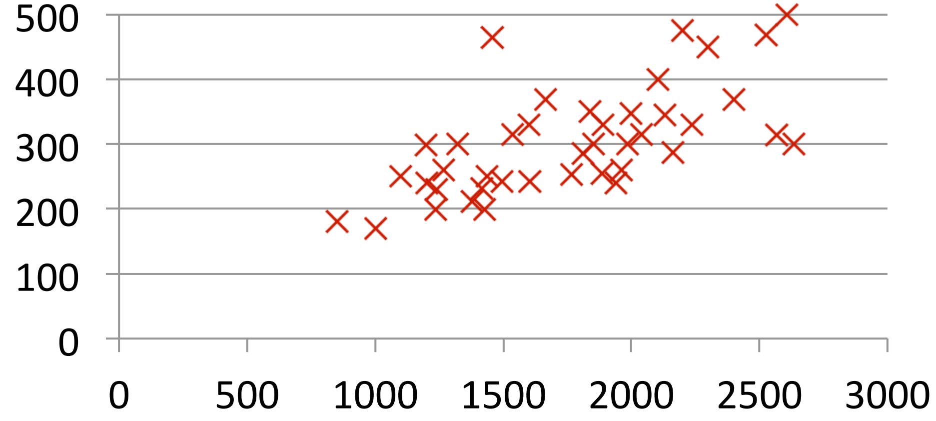


Training Set {$x^{(i)}$,$y^{(i)}$}
Learning Algorithm
Size of
house **x**
h
Estimated
price **y**

What should *h* be?

Linear hypothesis:

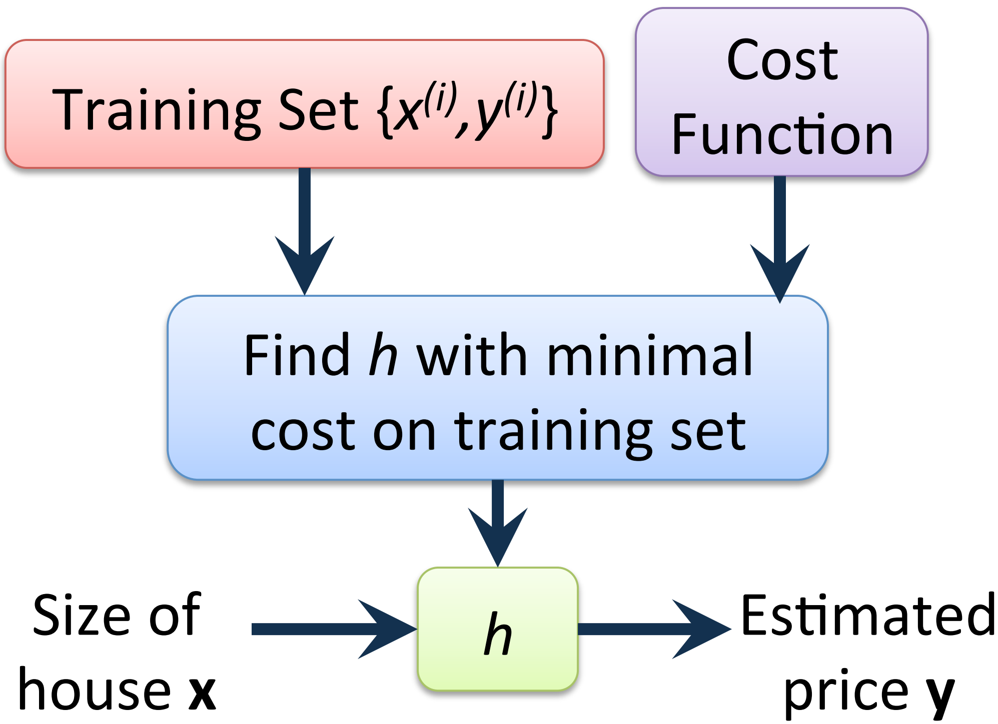
$$h_\theta(x) = \theta_0 + \theta_1 x$$



θ_i's: Parameters

How to choose θ_i's ?

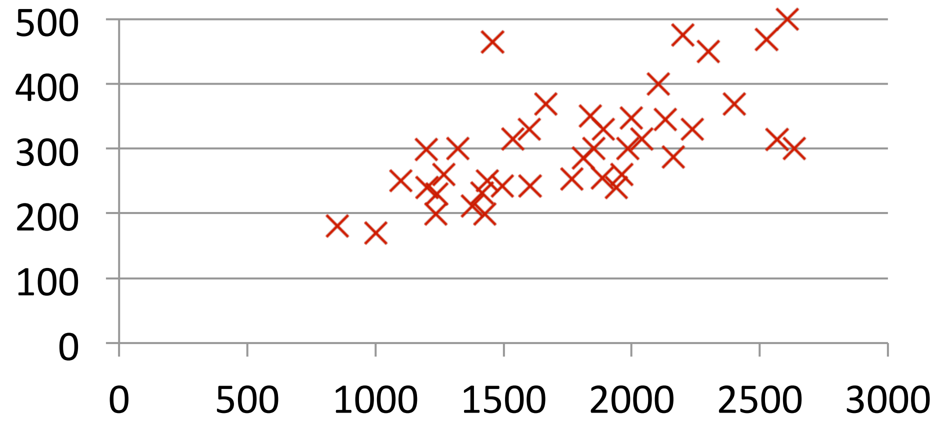
Cost function



"Sum of squared differences" or SSD cost function

Hypothesis:

$$h_\theta(x) = \theta_0 + \theta_1 x$$



θ_i's: Parameters

Cost Function:

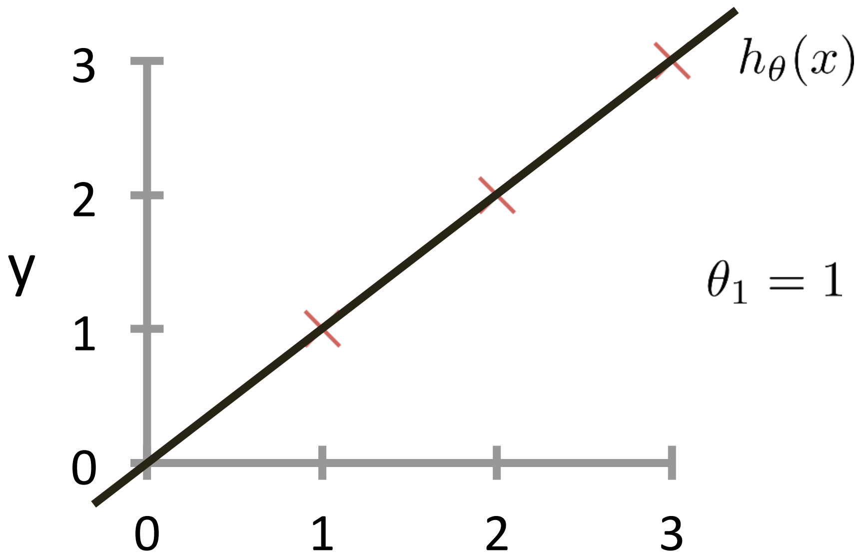
$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

Cost function intuition

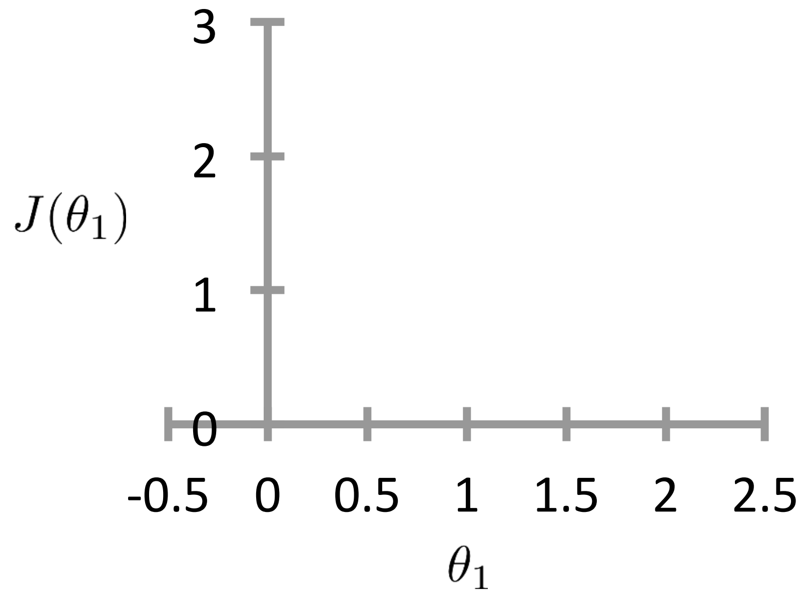
$h_\theta(x)$

(for fixed θ_1, this is a function of x)



$J(\theta_1)$

(function of the parameter θ_1)

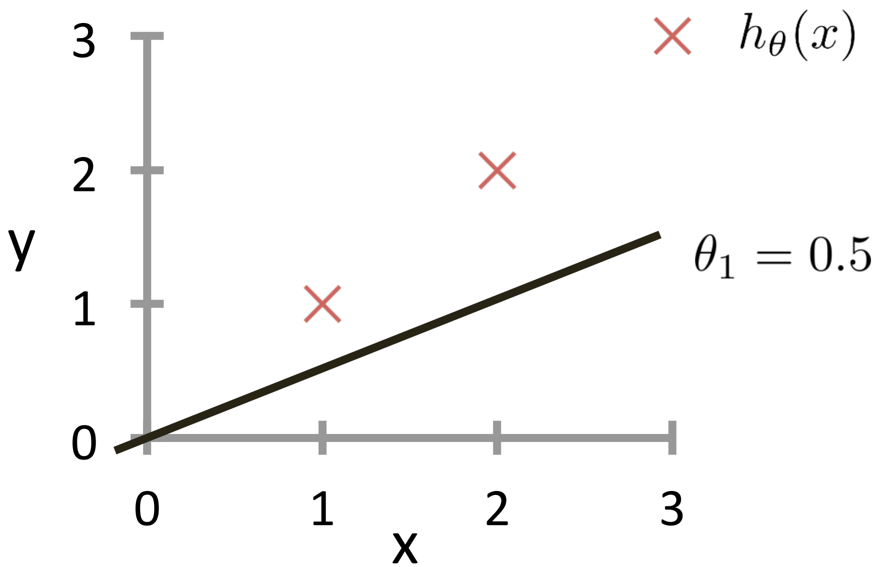


$$J(\theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Cost function intuition

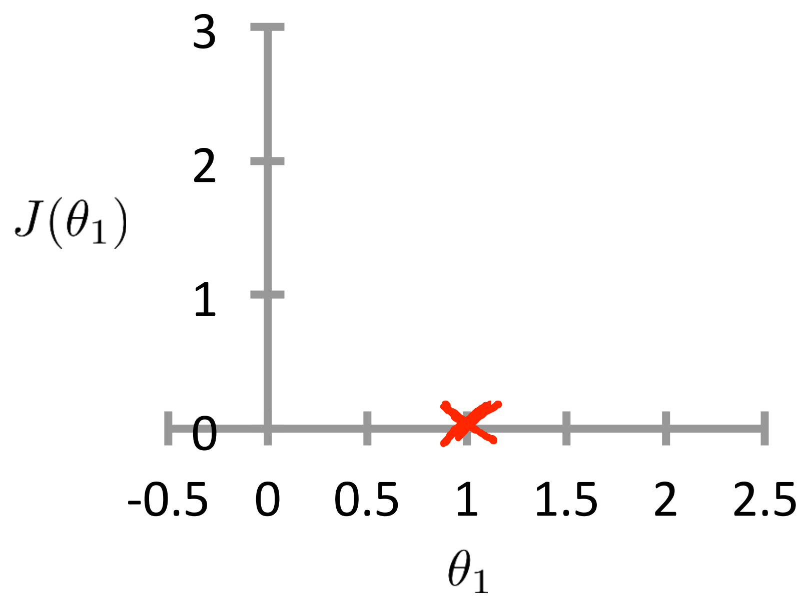
$h_\theta(x)$

(for fixed θ_1, this is a function of x)



$J(\theta_1)$

(function of the parameter θ_1)

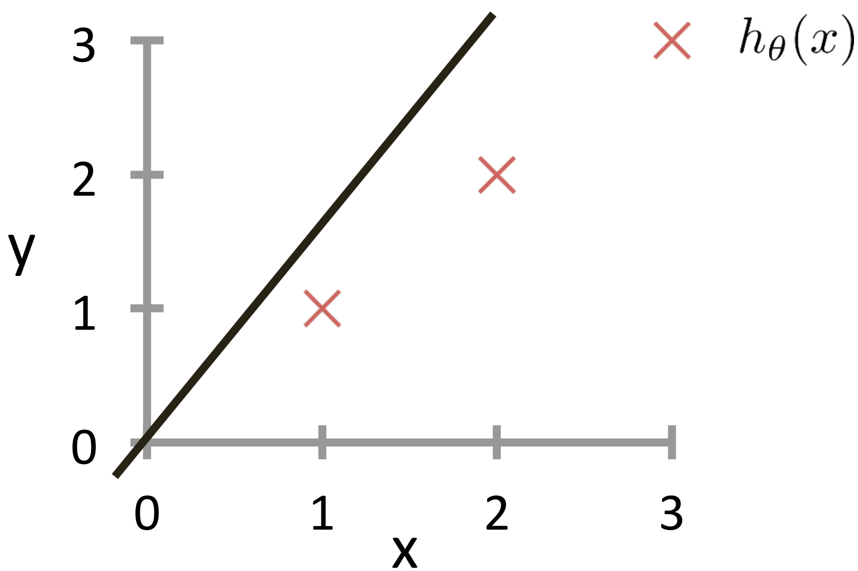


$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Cost function intuition

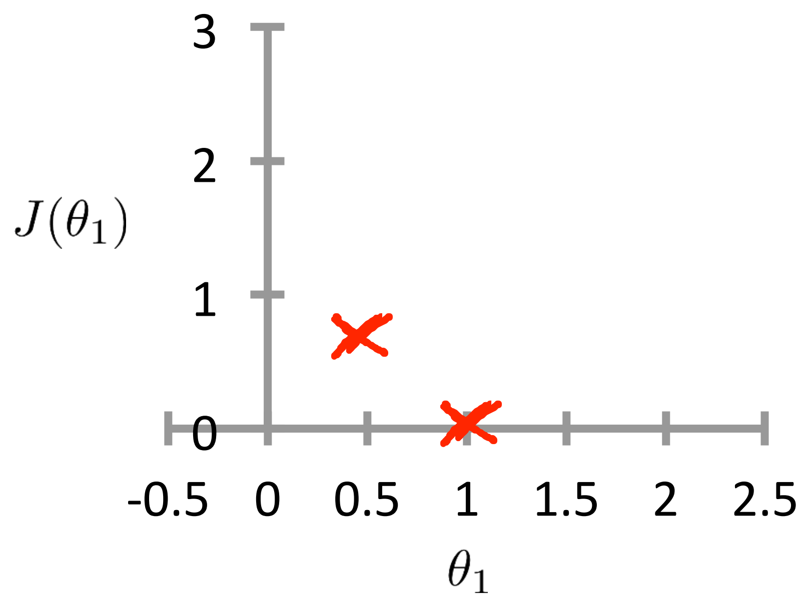
$h_\theta(x)$

(for fixed θ_1, this is a function of x)



$J(\theta_1)$

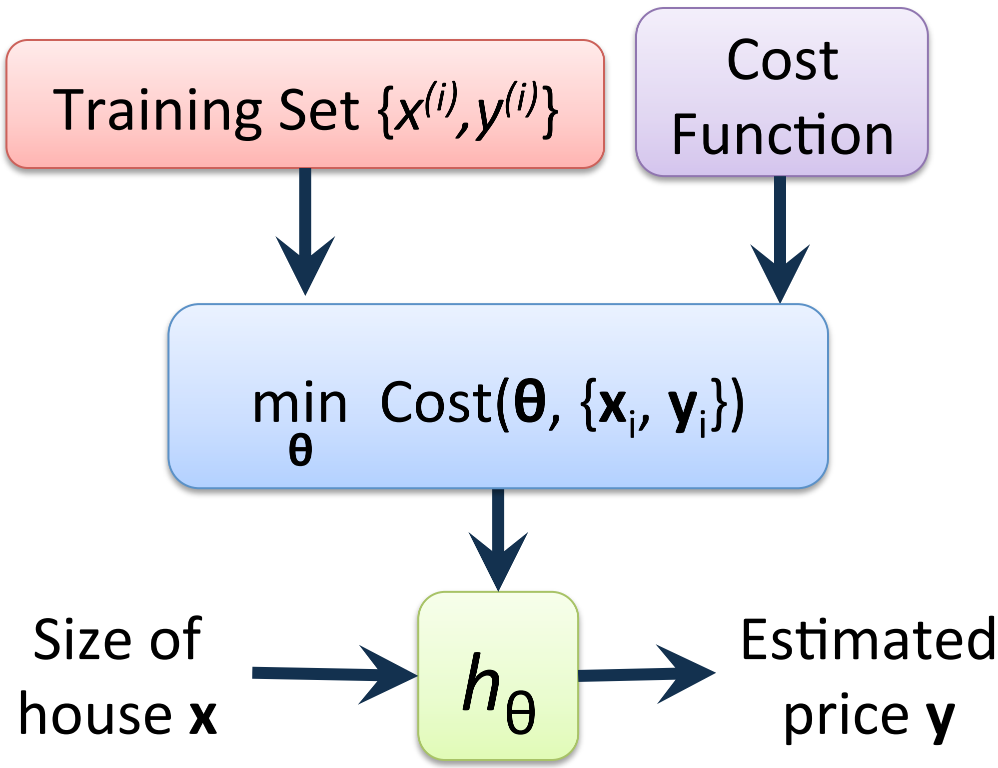
(function of the parameter θ_1)



$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Choose θ_1 with minimum cost

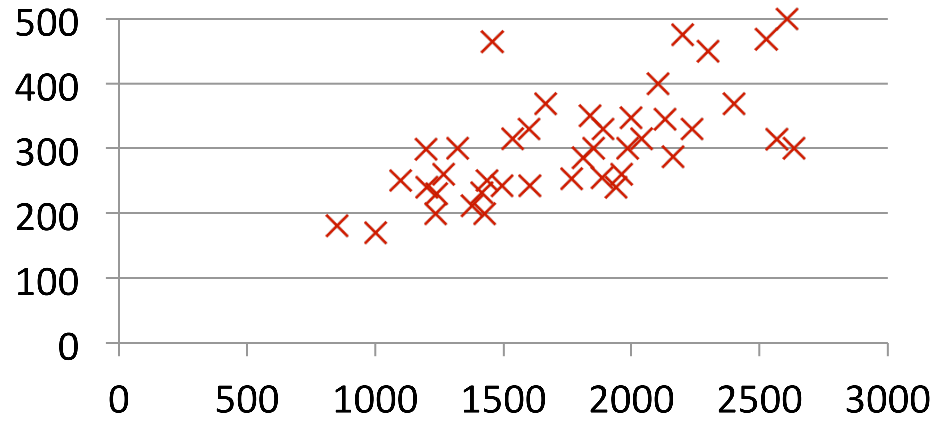
Supervised Learning



2-dimensional θ

Hypothesis:

$$h_\theta(x) = \theta_0 + \theta_1 x$$



θ_i's: Parameters

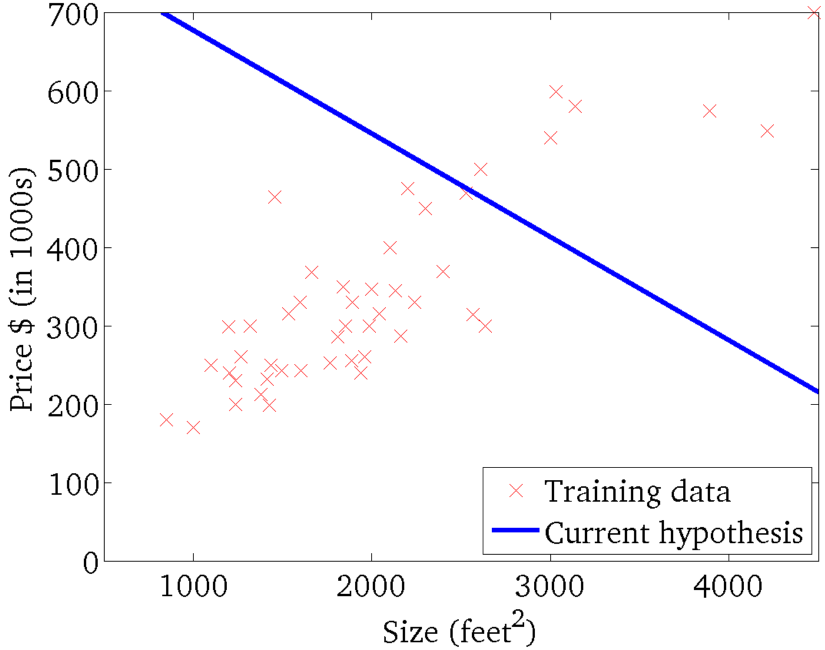
Cost Function:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Plotting cost for 2-dimensional θ

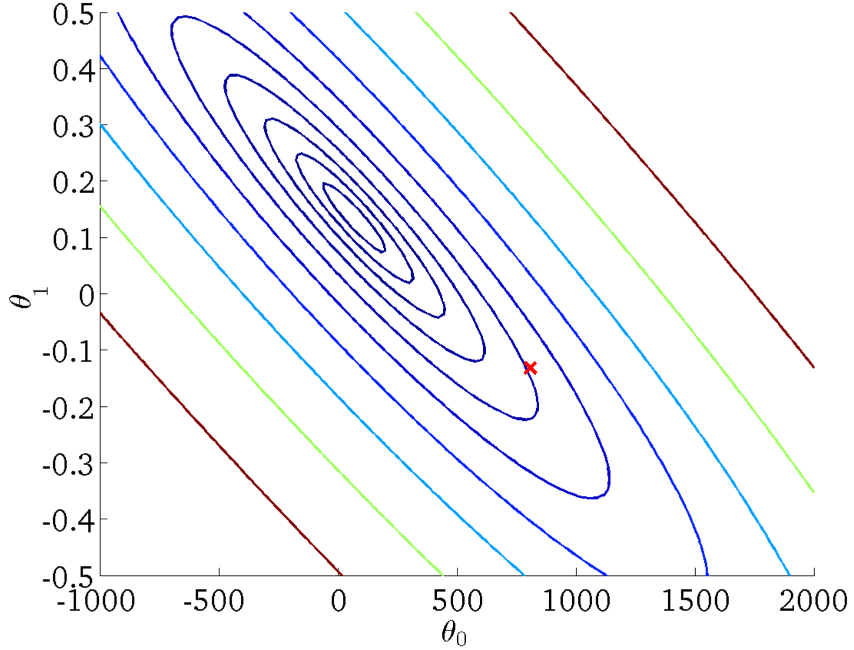
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

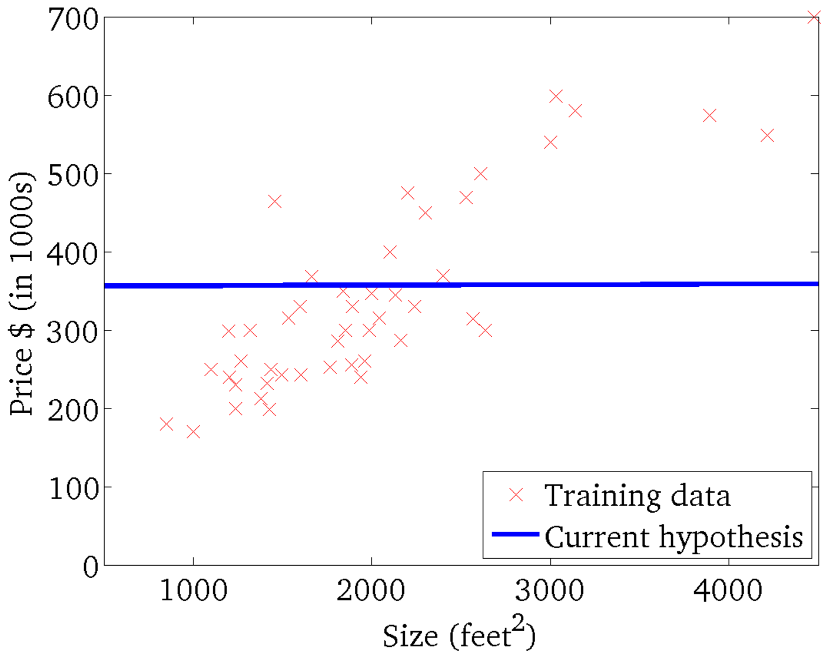
(function of the parameters θ_0, θ_1)



Plotting cost for 2-dimensional θ

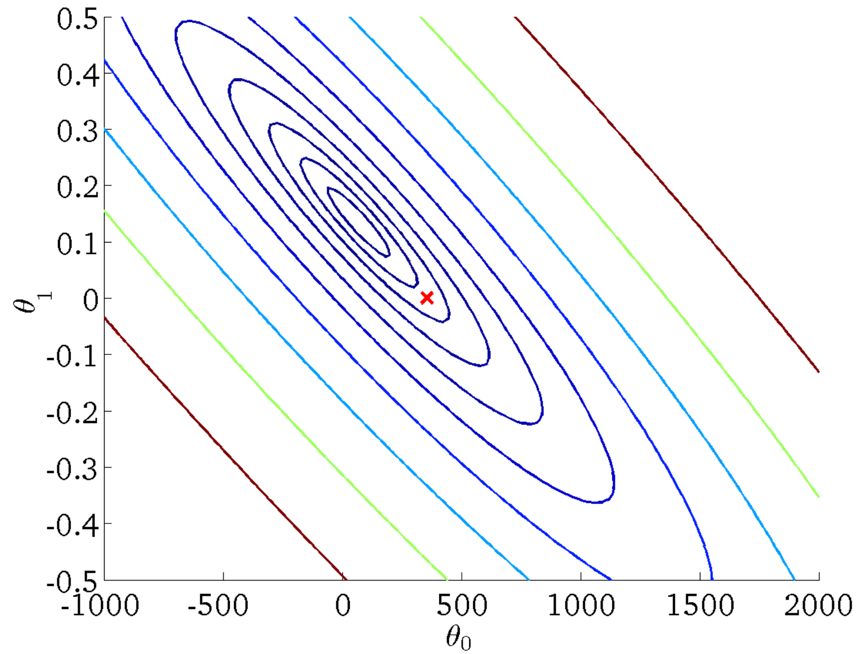
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



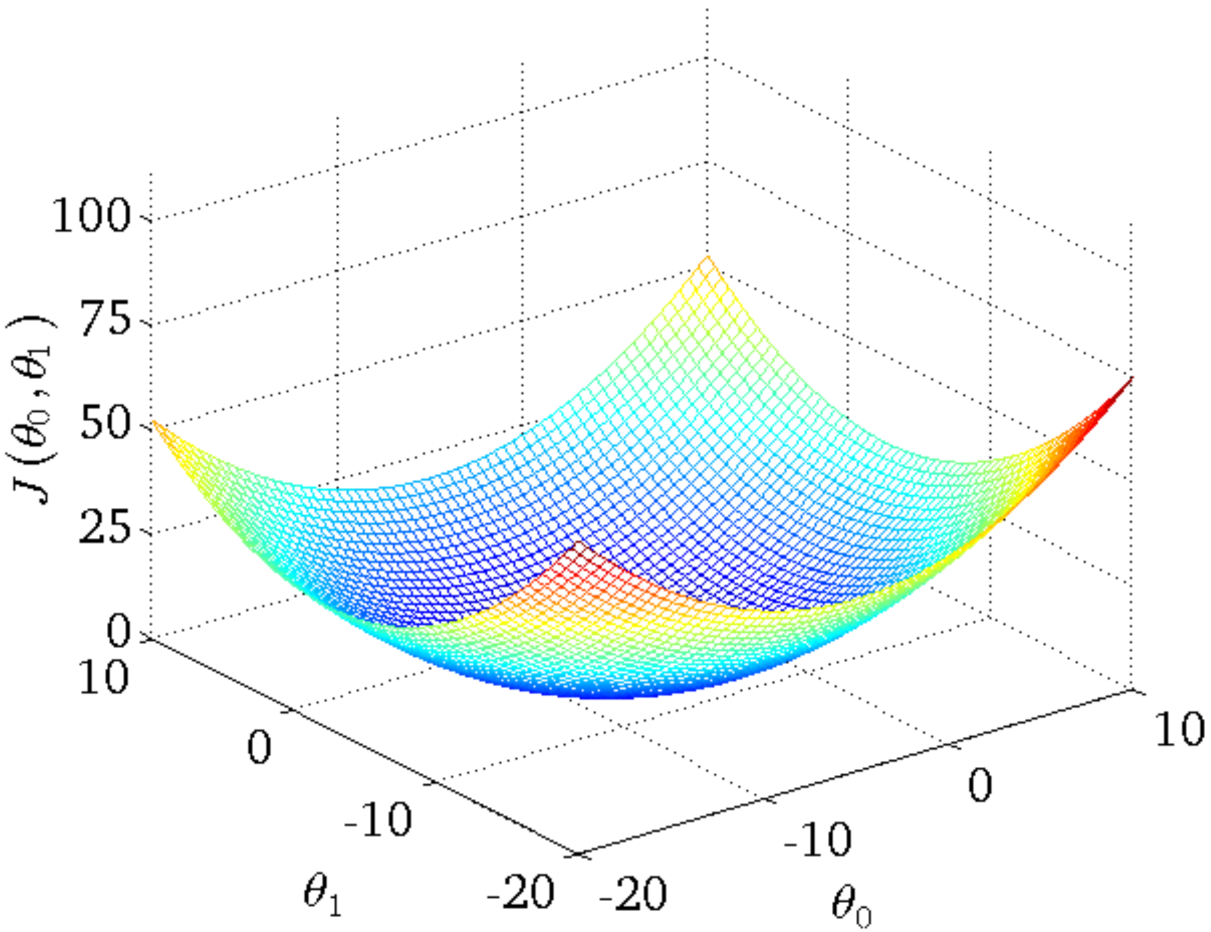
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)

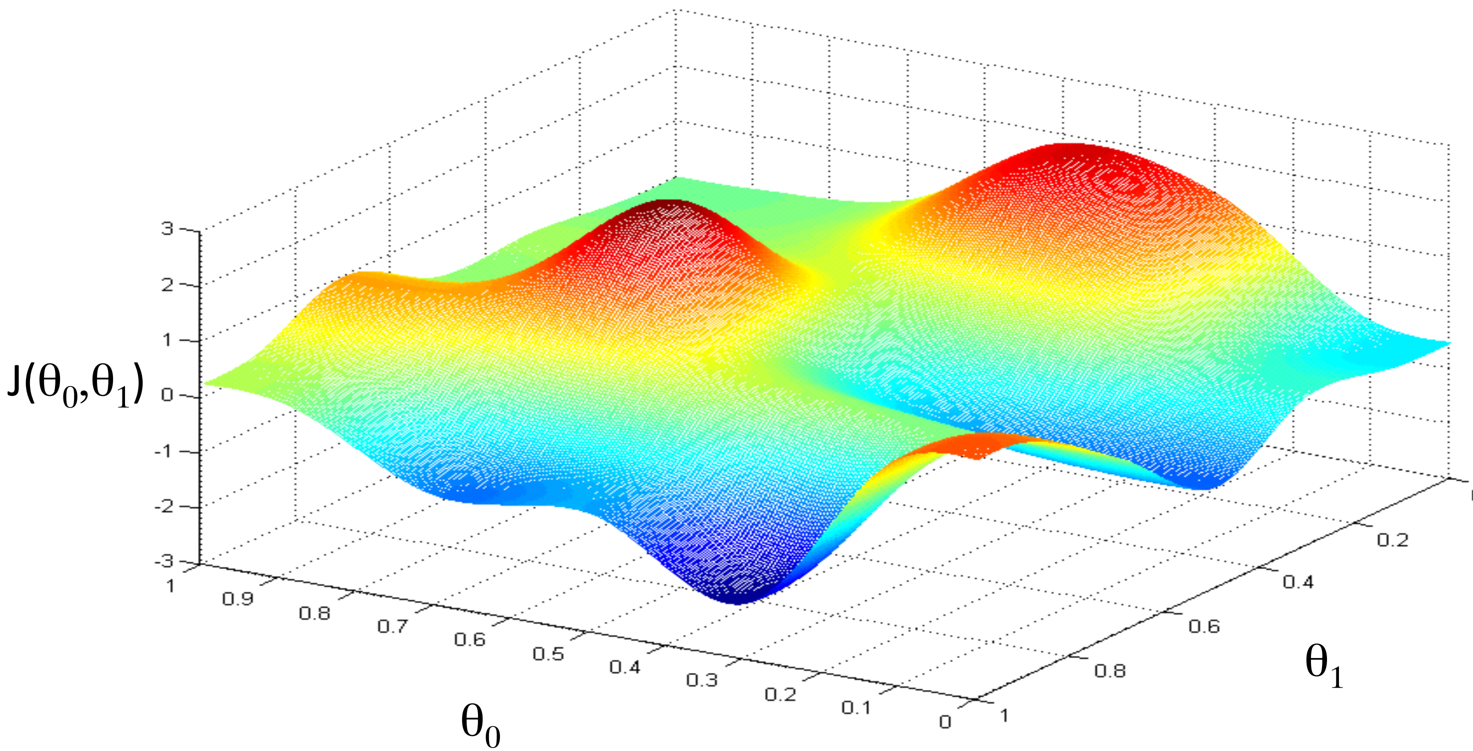


Note, squared loss cost is convex in parameters

SSD loss is convex



Non-convex cost function



Multidimensional inputs

Size (feet^2)	Number of bedrooms	Number of floors	Age of home (years)	Price ($1000)
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
…	…	…	…	…

Notation:

n = number of features

$x^{(i)}$ = input (features) of i^{th} training example.

$x_j^{(i)}$ = value of feature j in i^{th} training example.

Multivariate Linear Regression

Hypothesis:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

For convenience of notation, define $x_0 = 1$.

θ_i 's: Parameters

Cost Function:

$$J(\theta_0, \theta_1, \ldots, \theta_n) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

Goal: $\underset{\theta_0, \theta_1, \ldots, \theta_n}{\text{minimize}} \; J(\theta_0, \theta_1, \ldots, \theta_n)$

Two potential solutions

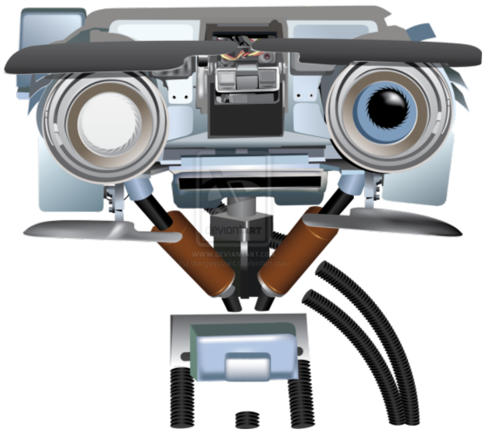
$$\min_{\theta} J(\theta; x_1, y_1, \ldots, x_m, y_m)$$

Gradient descent (or other iterative algorithm)

- Start with a guess for θ
- Change θ to decrease J(θ)
- Until reach minimum

Direct minimization

- Take derivative, set to zero
- Sufficient condition for minima



Gradient Descent

Gradient Descent Algorithm

Set $\theta = 0$

Repeat {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

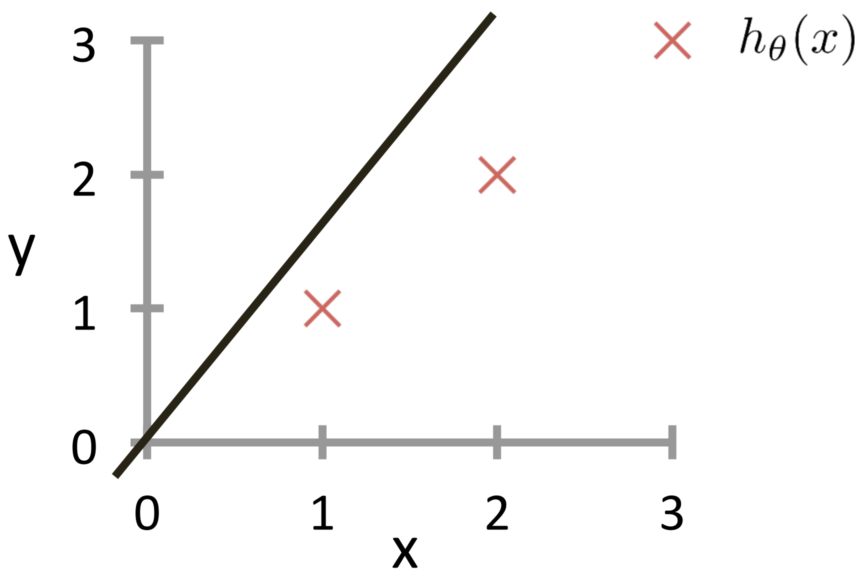
simultaneously for all $j = 0, \ldots, n$

} until convergence

Gradient Descent: Intuition

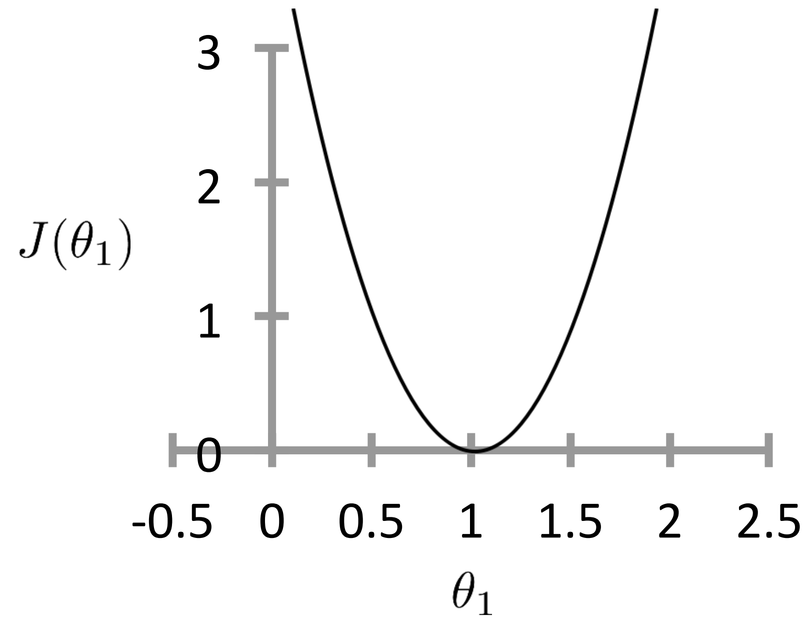
$h_\theta(x)$

(for fixed θ_1, this is a function of x)



$J(\theta_1)$

(function of the parameter θ_1)



$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

Gradient for Least Squares Cost

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

$$\frac{\partial}{\partial\theta_j}J(\theta) \quad =$$

Gradient for Least Squares Cost

For one example

$$\begin{aligned}
\frac{\partial}{\partial \theta_j} J(\theta) &= \frac{\partial}{\partial \theta_j} \frac{1}{2} \left(h_\theta(x) - y\right)^2 \\
&= 2 \cdot \frac{1}{2} \left(h_\theta(x) - y\right) \cdot \frac{\partial}{\partial \theta_j} \left(h_\theta(x) - y\right) \\
&= \left(h_\theta(x) - y\right) \cdot \frac{\partial}{\partial \theta_j} \left(\sum_{i=0}^{n} \theta_i x_i - y \right) \\
&= \left(h_\theta(x) - y\right) x_j
\end{aligned}$$

Gradient Descent Algorithm

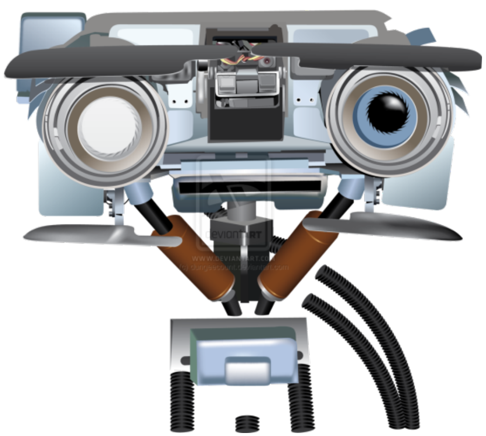
Set $\theta = 0$

Repeat {

$$\theta_j := \theta_j + \alpha \sum_{i=1}^{m} \left(y^{(i)} - h_\theta(x^{(i)})\right) x_j^{(i)}$$

simultaneously for all $j = 0, \ldots, n$

} until convergence



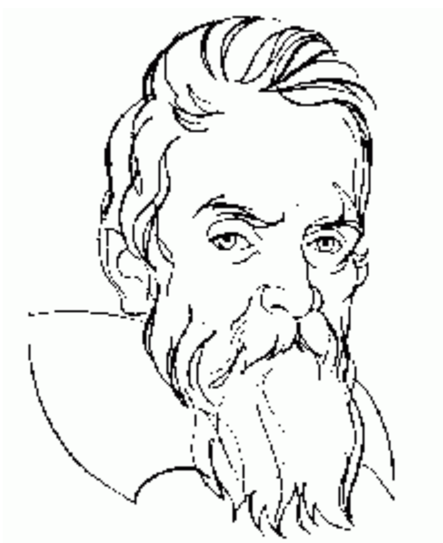
Probabilistic Solution

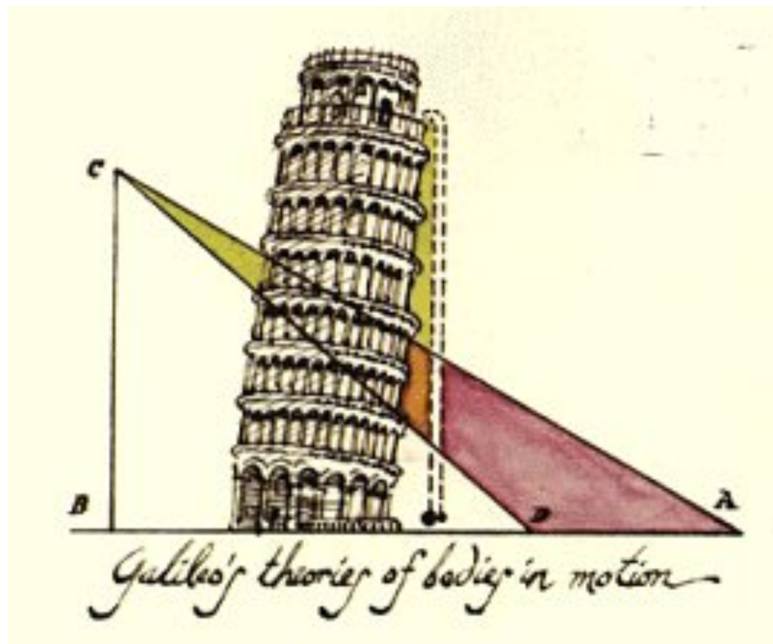
Maximum Likelihood

So far, we have treated observations as noiseless

An alternate view:

- data (x,y) are generated by unknown process
- however, we only observe a noisy version
- how can we model this uncertainty?





Learning models from data

- Many statistical models
  - Parametric: decision trees, linear and logistic regression, etc
  - Non-parametric: k-Nearest Neighbors, etc
- How to specify parameters in these models?
  - Manually: cumbersome and does not scale up
  - Automatically: estimate from data!
- Parameter estimation techniques
  - Maximum likelihood estimation (frequentist)
  - Bayesian inference

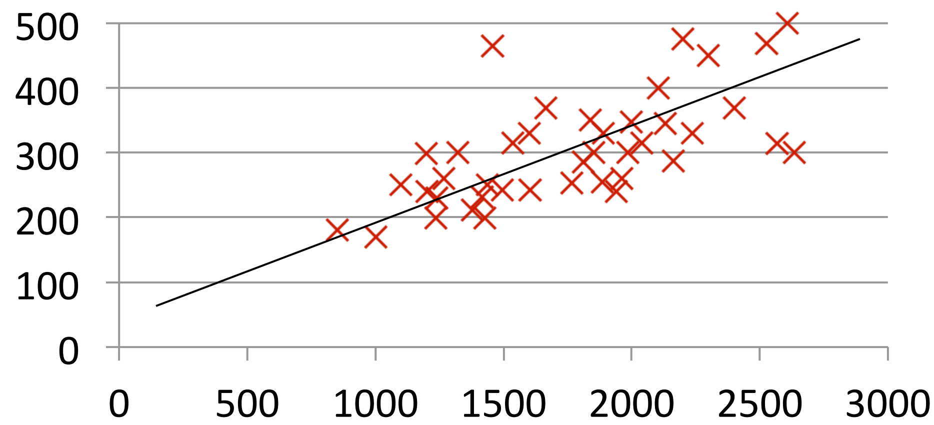
How to model uncertainty in data?

Hypothesis:

$h_{\theta}(x)=\theta^{T} x$

θ: parameters

$D=(x^{(i)},y^{(i)})$: data



Probability of: ???

Goal: maximize above probability

Maximum Likelihood: Example

- Intuitive example: Estimate a coin toss

  I have seen 3 flips of heads, 2 flips of tails, what is the chance of head (or tail) of my next flip?

- Model:

  Each flip is a Bernoulli random variable X

  X can take only two values: 1 (head), 0 (tail)

  $$p(X=1)=\theta, \quad p(X=0)=1-\theta$$

- θ is a parameter to be identified from data

Maximum Likelihood: Example

- 5 (independent) trials

$X_1 = 1$ $X_2 = 0$ $X_3 = 1$ $X_4 = 1$ $X_5 = 0$

- Likelihood of all 5 observations:

$$p(D|\theta) = p(X_1, \ldots, X_5 | \theta) = \theta^3 (1-\theta)^2$$

- Intuition

ML chooses θ such that likelihood is maximized

Maximum Likelihood: Example

- 5 (independent) trials

$X_1 = 1$ $X_2 = 0$ $X_3 = 1$ $X_4 = 1$ $X_5 = 0$

- Likelihood of all 5 observations:

$$p(D|\theta)=p(X_1,\ldots,X_5|\theta)=\theta^3(1-\theta)^2$$

- Solution (left as exercise)

$$\theta_{ML}=3/(3+2)$$

i.e. fraction of heads in total number of trials

Maximum Likelihood

More generally, assume

$$X \sim p(X|\theta)$$

Observations

$$D = \{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$$

Maximum likelihood estimate

Likelihood

$$\mathcal{L}(D) = \prod_{i=1}^{m} p(x^{(i)}|\theta)$$

Log likelihood

$$\theta_{ML} = \arg\max_{\theta} \mathcal{L}(D)$$

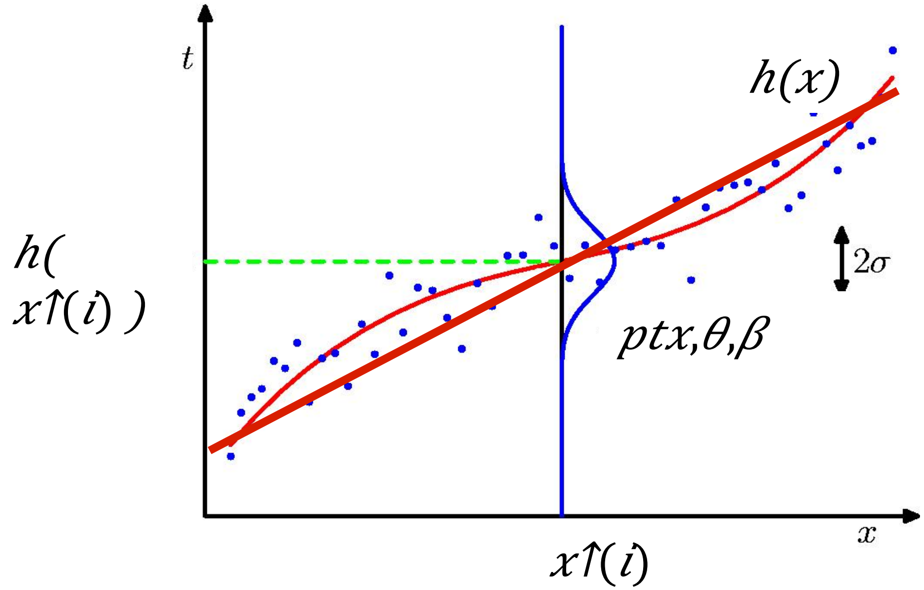
$$= \arg\max_{\theta} \sum_{i=1}^{m} \log p(x^{(i)}|\theta)$$

Assume:

$t = y + \epsilon$

Noise $\epsilon \sim N(\epsilon \mid 0, \beta^{-1})$,

where $\beta = 1/\sigma^2$



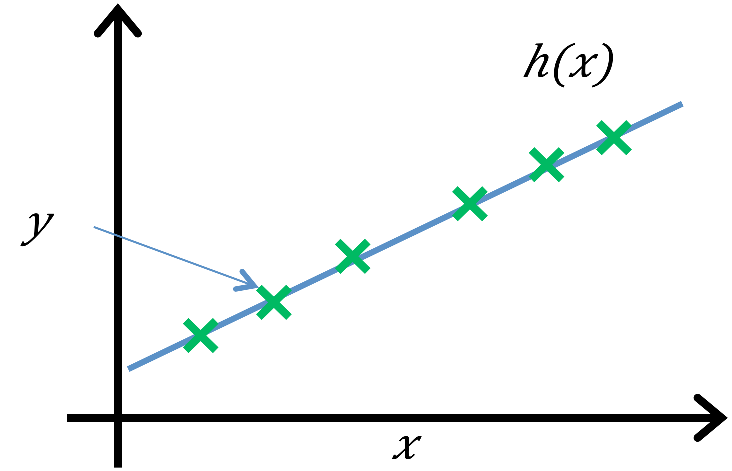
we don't get to see y, only t

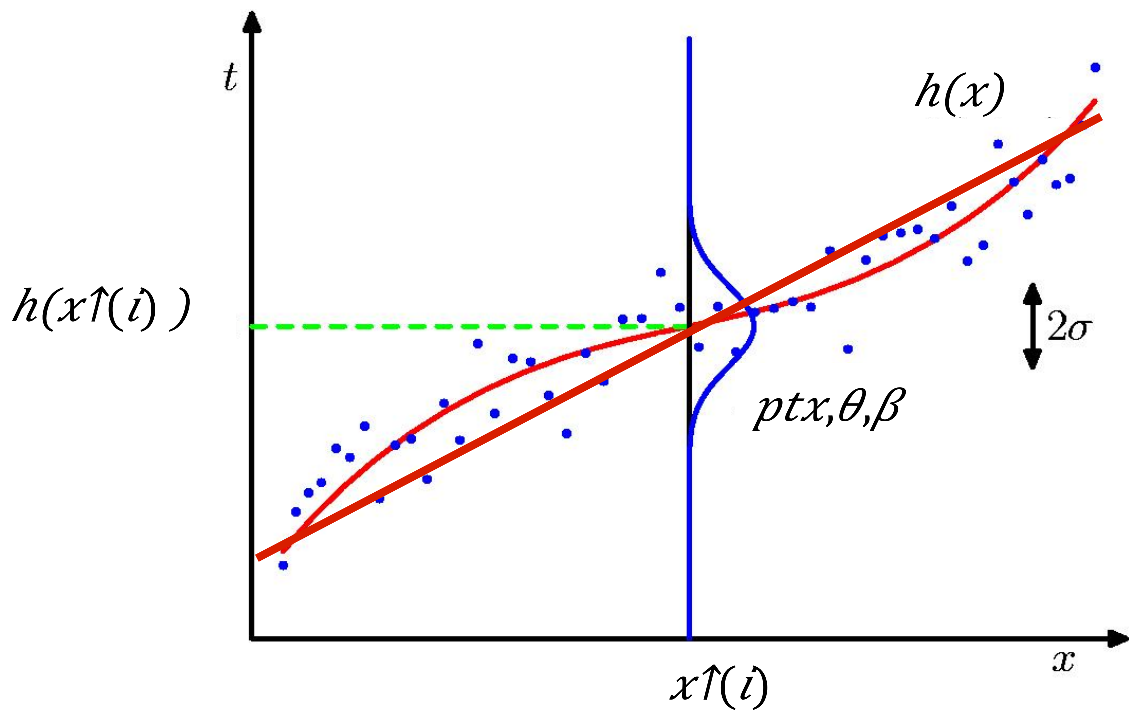
Assume:

$t = y + \epsilon$

Noise $\epsilon \sim N(\epsilon \mid 0, \beta^{-1})$, where $\beta = 1/\sigma^2$

$p(t \mid x, \theta, \beta) = N(t \mid h(x), \beta^{-1})$



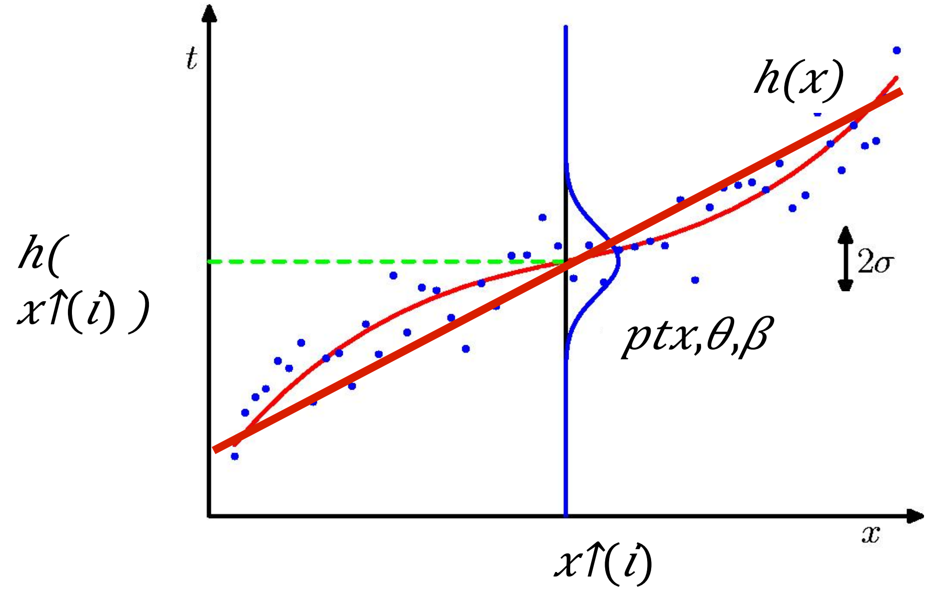


Assume:

$t = y + \epsilon$

Noise $\epsilon \sim N(\epsilon \mid 0, \beta^{-1})$,

where $\beta = 1/\sigma^2$



$p(t \mid x, \theta, \beta) = N(t \mid h(x), \beta^{-1})$

$\boldsymbol{t} = \{t^{(i)}\}$, $\mathbf{x} = \{x^{(i)}\}$, assume $t^{(i)}$ are i.i.d

i.i.d assumption: if x_i are independent r.v.s, then

$p(x_1, x_2, \ldots, x_m) = \mathrm{p}(x_1)\, \mathrm{p}(x_2) \ldots \mathrm{p}(x_m)$

$p(\boldsymbol{t} \mid \boldsymbol{x}, \theta, \beta) = \prod_{i=1}^{m} N(t^{(i)} \mid h(x^{(i)}), \beta^{-1})$

Likelihood Function

Likelihood function:

$$p\boldsymbol{t}\boldsymbol{x},\theta,\beta = \prod_{i=1}^{m} N(t^{(i)} \,|\, h(x^{(i)}), \beta^{-1})$$

Maximum likelihood solution:

$$\theta_{ML} = \operatorname{argmax}_{\theta}\, p\boldsymbol{t}\boldsymbol{x},\theta,\beta$$

$$\beta_{ML} = \operatorname{argmax}_{\beta}\, p\boldsymbol{t}\boldsymbol{x},\theta,\beta$$

Want to maximize

$$p\boldsymbol{t}\boldsymbol{x},\theta,\beta = \prod_{i=1}^{m} N(t^{(i)} \mid h(x^{(i)}), \beta^{-1})$$

Easier to maximize log()

$$\ln p\boldsymbol{t}\boldsymbol{x},\theta,\beta = -\beta/2 \sum_{i=1}^{m} (h(x^{(i)}) - t^{(i)})^{2} + m/2 \ln\beta - m/2 \ln(2\pi)$$

Want to maximize w.r.t. θ

$$\ln p \boldsymbol{t} \boldsymbol{x}, \theta, \beta = -\beta/2 \sum_{i=1}^{m} \left(h(x^{(i)}) - t^{(i)}\right)^2 + m/2 \ln \beta - m/2 \ln(2\pi)$$

… but this is same as minimizing sum-of-squares cost[1]

$$1/2m \sum_{i=1}^{m} \left(h(x^{(i)}) - t^{(i)}\right)^2$$

… which is the same as our SSE cost from before:

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

[1]multiply by $-1/m\beta$, changing max to min, omit last two terms (don't depend on θ)

Probabilistic Motivation for SSE

- Under the Gaussian noise assumption, maximizing the probability of the data points is the same as minimizing a sum-of-squares cost function

- Also known as least squares method

- Not only for linear hypotheses!
  - But linear least squares has closed-form solution

Bayesian vs. Frequentist

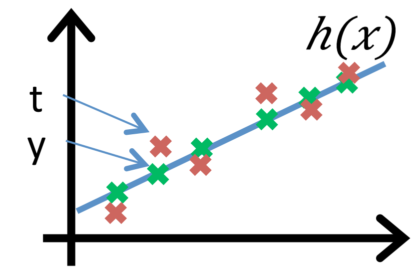
Frequentist: maximize likelihood

$$pDmodel = pD\theta$$

Bayesian: treat θ as random variable, e.g. maximize posterior

$$p\theta D = pD\theta\, p(\theta)/p(D)$$

Summary: Max Likelihood for LR



Assume linear model with Gaussian observation noise

Likelihood function:

$$p(\mathbf{t}|\mathbf{x},\theta,\beta)=\prod_{i=1}^{m}\mathcal{N}(t^{(i)}\,|\,h(x^{(i)}),\beta^{-1})$$

Maximum likelihood solution:

$$\theta_{ML}=\operatorname{argmax}_{\theta}\, p(\mathbf{t}|\mathbf{x},\theta,\beta)$$

use Gradient Descent, or
closed form solution
(will see shortly)

$$=\operatorname{argmin}_{\theta}\,\frac{1}{2m}\sum_{i=1}^{m}(h(x^{(i)})-t^{(i)})^2$$

(same as minimizing SSE)

$$\beta_{ML}=\operatorname{argmax}_{\beta}\, p(\mathbf{t}|\mathbf{x},\theta,\beta)$$

(left as exercise)

Next Lecture

- Matrix calculus
- Direct solution: normal equations