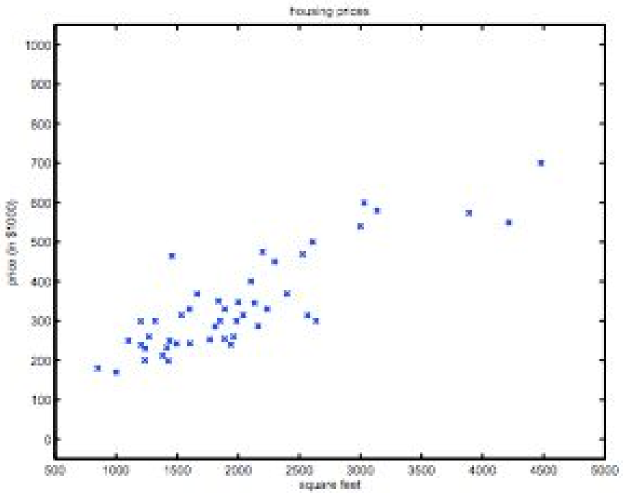
Linear Regression

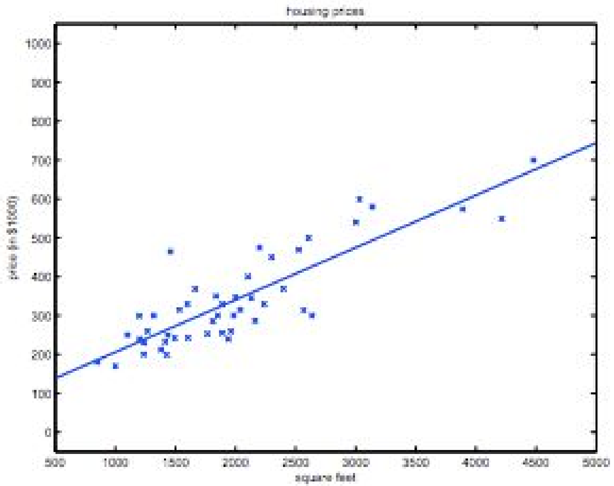
GA DAT3

Agenda

1. Overview
2. Cost Function
3. Gradient Descent
4. Normal Equation
5. Probabilistic Interpretation
6. Locally Weighted Linear Regression

Linear Regression Overview





Supervised Learning

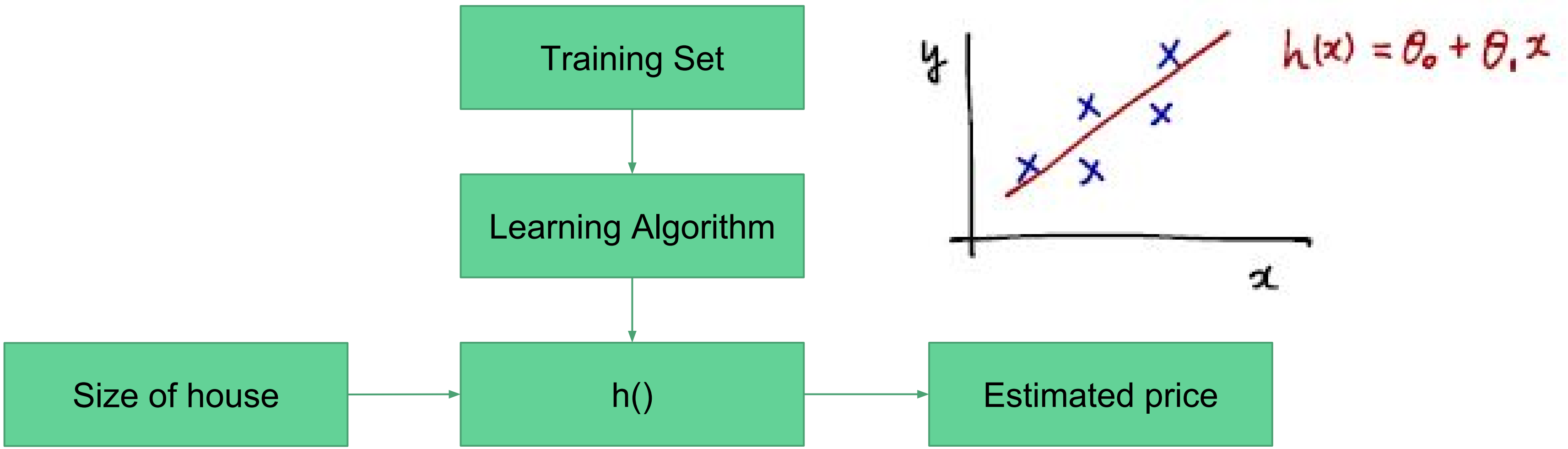
Regression: predict real-value output
Classification: predict discrete value output

Training set of housing prices (Portland, OR)

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's = "input" variable / features
y's = "output" variable / "target" variable



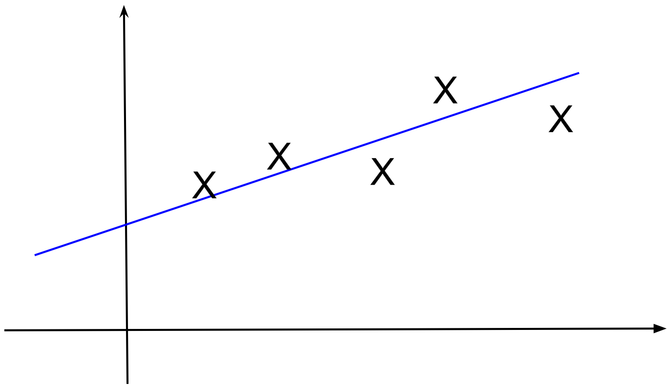
Linear regression with one variable **:**
Univariate linear regression

Cost Function : Squared Error Loss

Minimize:

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

WHY?



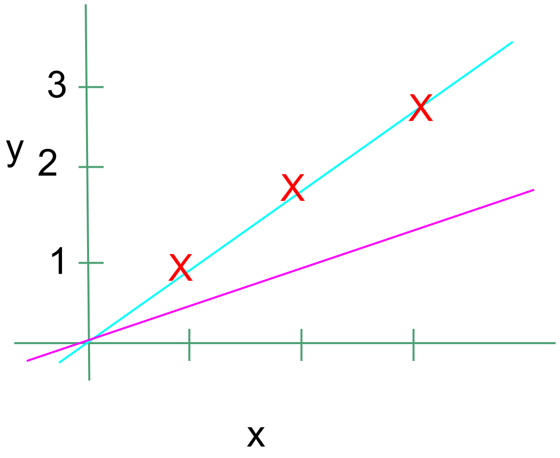
General Multivariate Form

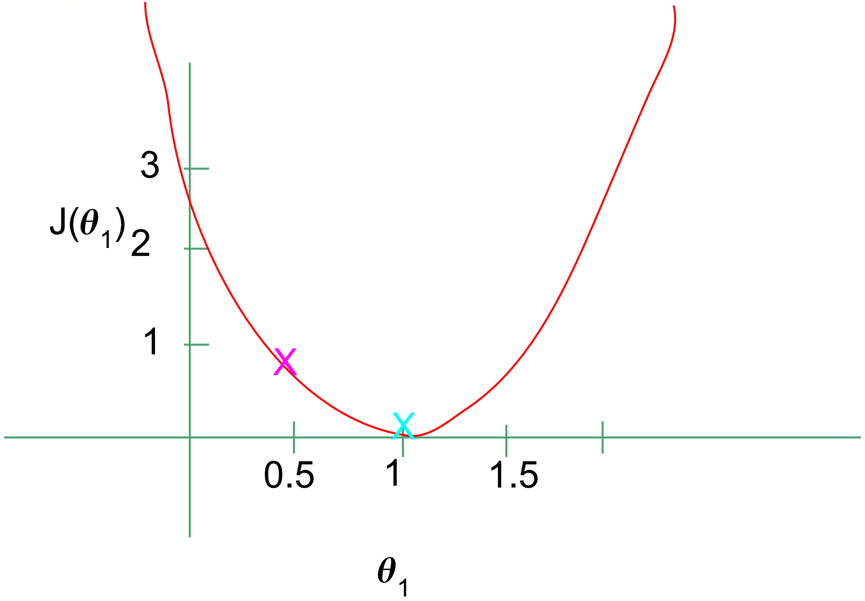
$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 \boxed{\text{+ ... + theta(n)x(n)}}$$

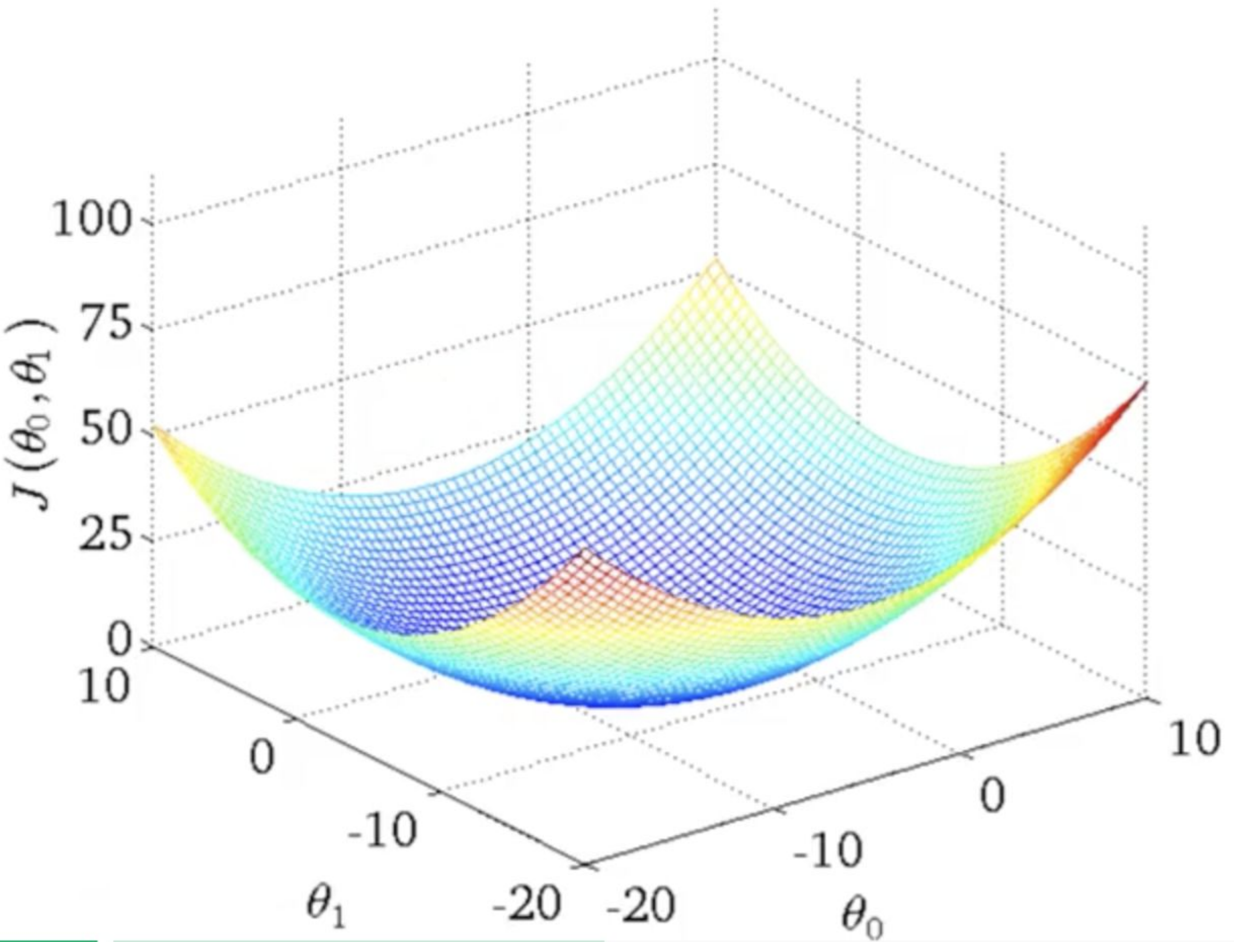
$$h(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x,$$

Cost Function : Squared Error Loss

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







100
75
50
25
0
$J(\theta_0, \theta_1)$
10
0
-10
-20
θ_1
-20
-10
0
10
θ_0

Gradient Descent

Update Rule:

Learning rate

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

Repeat until convergence {

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

}

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

original value

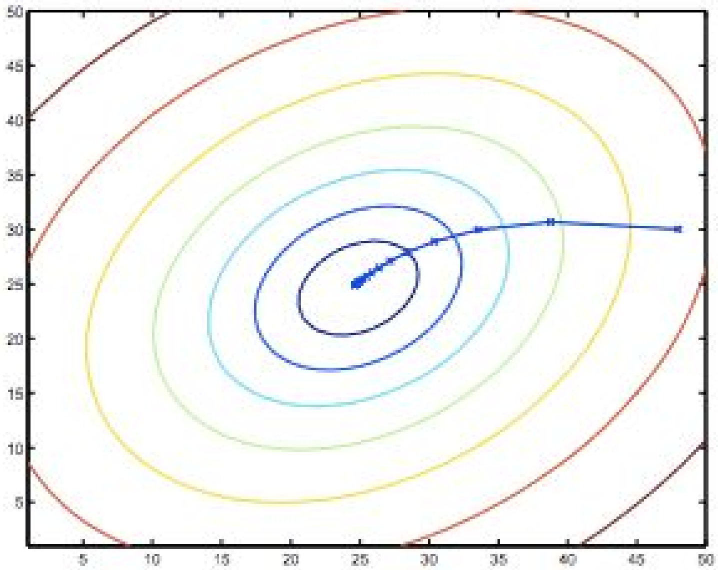
expected value

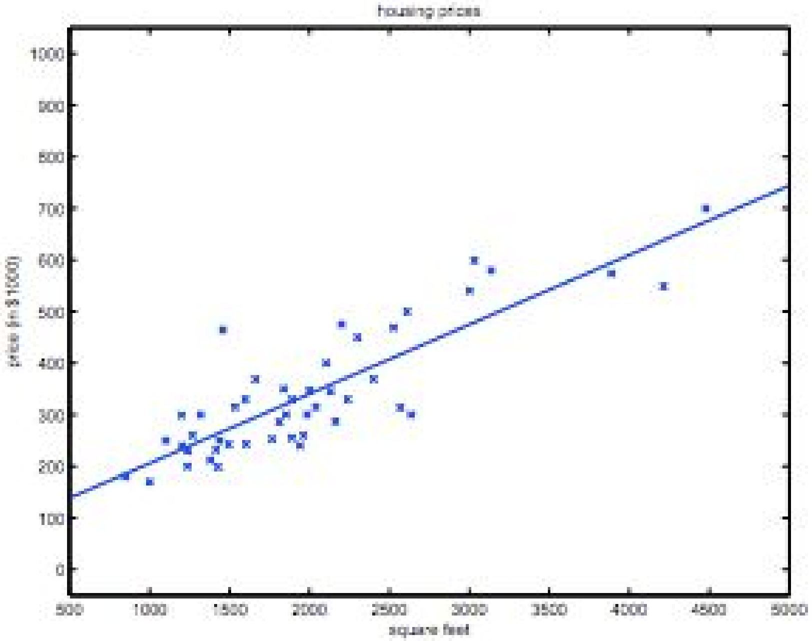
For sufficiently small α, θ should decrease on every iteration.
But if α is too small, gradient descent can be slow to converge.'\

if alpha too big, might overshoot global min.
if alpha too small, takes too much time.

learning rate can be made smaller after every iteration, see stochastic descent in later slide

What are the axes?




housing prices
price (in $1000)
square feet
0
100
200
300
400
500
600
700
800
900
1000
500
1000
1500
2000
2500
3000
3500
4000
4500
5000

Batch vs Stochastic Descent

will be covered at later session when dealing with very large data sets

for theta:

    for each example:

        ...

for each example:

    for theta:

        ...

Normal Equations

$$
\begin{aligned}
\nabla_\theta J(\theta) &= \nabla_\theta \frac{1}{2}(X\theta - \vec{y})^T(X\theta - \vec{y}) \\
&= \frac{1}{2}\nabla_\theta \left(\theta^T X^T X\theta - \theta^T X^T \vec{y} - \vec{y}^T X\theta + \vec{y}^T \vec{y}\right) \\
&= \frac{1}{2}\nabla_\theta \operatorname{tr}\left(\theta^T X^T X\theta - \theta^T X^T \vec{y} - \vec{y}^T X\theta + \vec{y}^T \vec{y}\right) \\
&= \frac{1}{2}\nabla_\theta \left(\operatorname{tr}\theta^T X^T X\theta - 2\operatorname{tr}\vec{y}^T X\theta\right) \\
&= \frac{1}{2}\left(X^T X\theta + X^T X\theta - 2X^T \vec{y}\right) \\
&= X^T X\theta - X^T \vec{y}
\end{aligned}
$$

$$\theta = \boxed{(X^T X)^{-1}} X^T \vec{y}.$$

works only if we can get this. if we cannot, then we use a pseudo inverse matrix

Gradient Descent vs Normal Equation

Gradient Descent	Normal Equation
Need to choose **α**	No need to choose **α**
Need many iterationa	Don’t need to iterate
Work well even with large n	Need to compute $(X^TX)^{-1}$
	Slow if n is large

n = number of features
n too large = in 1000s

Probabilistic Interpretation

$$y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)}$$

$$p(\epsilon^{(i)}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\epsilon^{(i)})^2}{2\sigma^2}\right).$$

We can minimise **cost**
or
maximise likelihood

max likelihood of a point on the regressed line that is the same as observed point

What's the likelihood?

Likelihood

This is a **conditional probability density**

$$L(\theta) = \prod_{i=1}^{m} p(y^{(i)} \mid x^{(i)}; \theta)$$

$$= \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)$$

Can you write this as a sum?

Likelihood - For you advanced folks

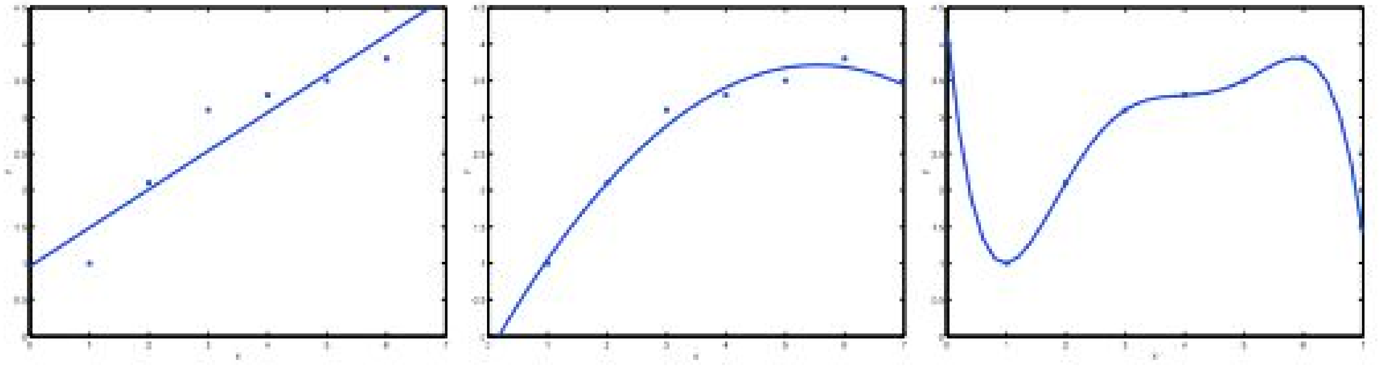
$$
\begin{aligned}
\ell(\theta) &= \log L(\theta) \\
&= \log \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right) \\
&= \sum_{i=1}^{m} \log \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right) \\
&= m \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{\sigma^2} \cdot \frac{1}{2} \sum_{i=1}^{m} (y^{(i)} - \theta^T x^{(i)})^2.
\end{aligned}
$$

Hence, maximizing $\ell(\theta)$ gives the same answer as minimizing

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

which is the cost function

Locally Weighted Linear Regression



classic tradeoff

Locally Weighted Linear Regression

Instead of doing:

1. Fit θ to minimize $\sum_i (y^{(i)} - \theta^T x^{(i)})^2$.
2. Output $\theta^T x$.

We do:

1. Fit θ to minimize $\sum_i w^{(i)} (y^{(i)} - \theta^T x^{(i)})^2$
2. Output $\theta^T x$.

introduce weight

Locally Weighted Linear Regression

Good examples of weights?

$$w^{(i)} = \exp\left(-\frac{(x^{(i)} - x)^2}{2\tau^2}\right)$$

Locally Weighted Linear Regression

Parametric vs Non-Parametric

What was k-Nearest Neighbours?

http://learning.cis.upenn.edu/cis520_fall2009/index.php?n=Lectures.LocalLearning#toc8

Q??