Inductive Learning

Anurag Nagar

Recap

- Inductive learning - generalize from a limited set of training data
- Training data is labeled
- You would like to estimate true separating function f
- Attributes of data (i.e. features) are important

Learning from data

- Given: a set of labeled training examples: <x, f(x)>
  Global f(x) is unknown to us
  Distribution of x is unknown to us
- Find: An <u>approximation</u> of f(x)

Appropriate situations

- **Credit risk assessment**

  $\mathbf{x}$: Properties of customer and proposed purchase.

  $f(\mathbf{x})$: Approve purchase or not.

- **Disease diagnosis**

  $\mathbf{x}$: Properties of patient (symptoms, lab tests)

  $f(\mathbf{x})$: Disease (or maybe, recommended therapy)

- **Face recognition**

  $\mathbf{x}$: Bitmap picture of person's face

  $f(\mathbf{x})$: Name of the person.

Learning

- Improving with experience (E) at some task (T) with respect to some performance measure (P).
- Experience = Training data
  Task = Any classification task (for this class, at least)
  Performance Measure = Error value
  -> difference between true value and predicted value.

Model Representation

What are you given in supervised learning?

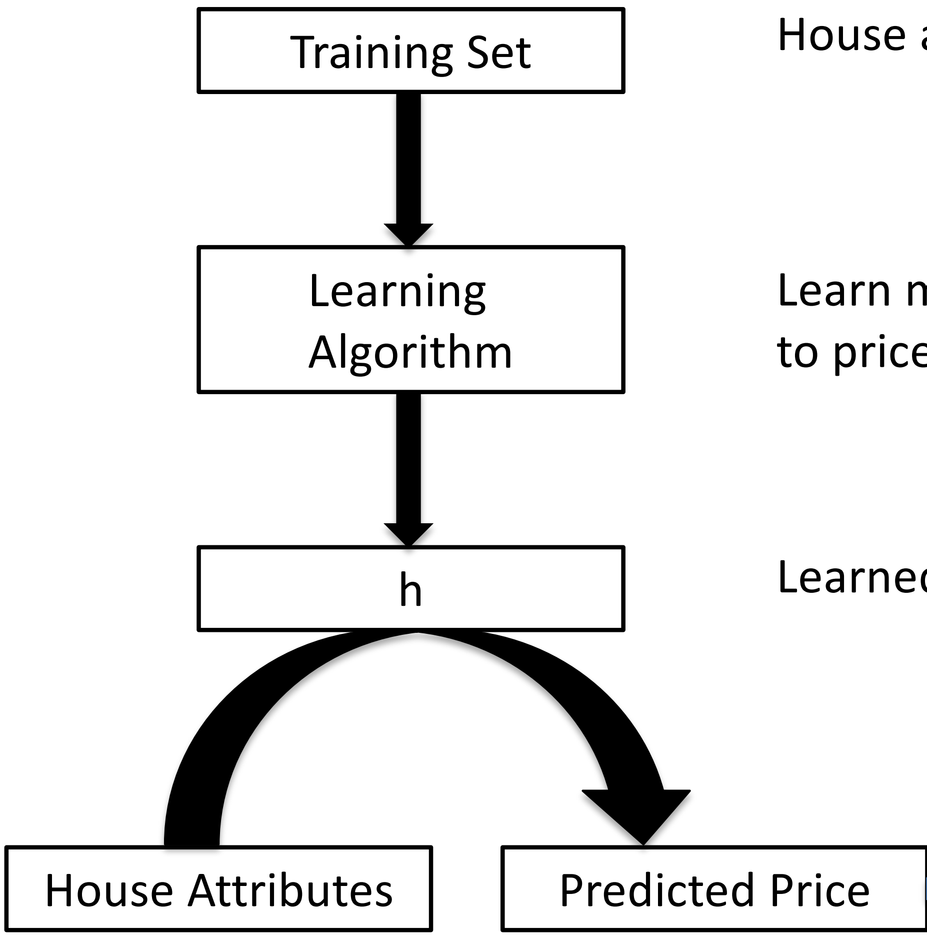
A set of training examples and their labels $(x^{(i)}, y^{(i)})$

** It is assumed $y^{(i)}$ is generated by a true function f(x) **

What do you do with the training data?

Feed it to a learning algorithm that learns a function h, that is an approximation to f

Learning Process



House attributes + Price

Learn mapping from attributes to price

Learned function

Error Metric

use it as a guide
After each training instance, refine h so that value of error metric goes down.

Model Representation

How do you know if h is good?

We measure the error (overall) by using h

Example:

Error = | f(x) – h(x) | or

Error = 1/(2m)* $(f(x) – h(x))^2$

Think:

Is more training data

good?

Always?

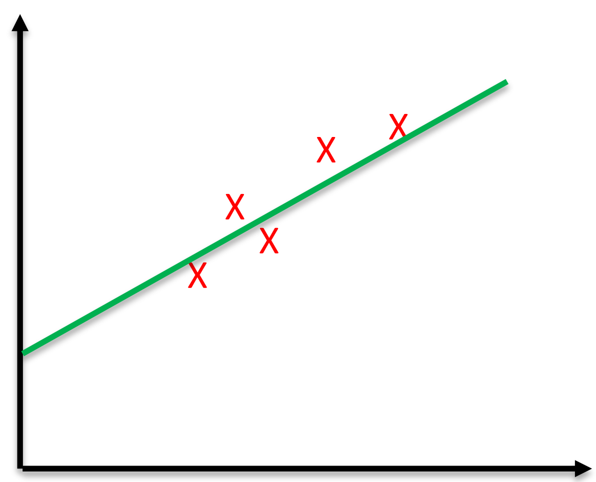
LINEAR REGRESSION

Learning a linear function

- Suppose we want to learn a function of the form:

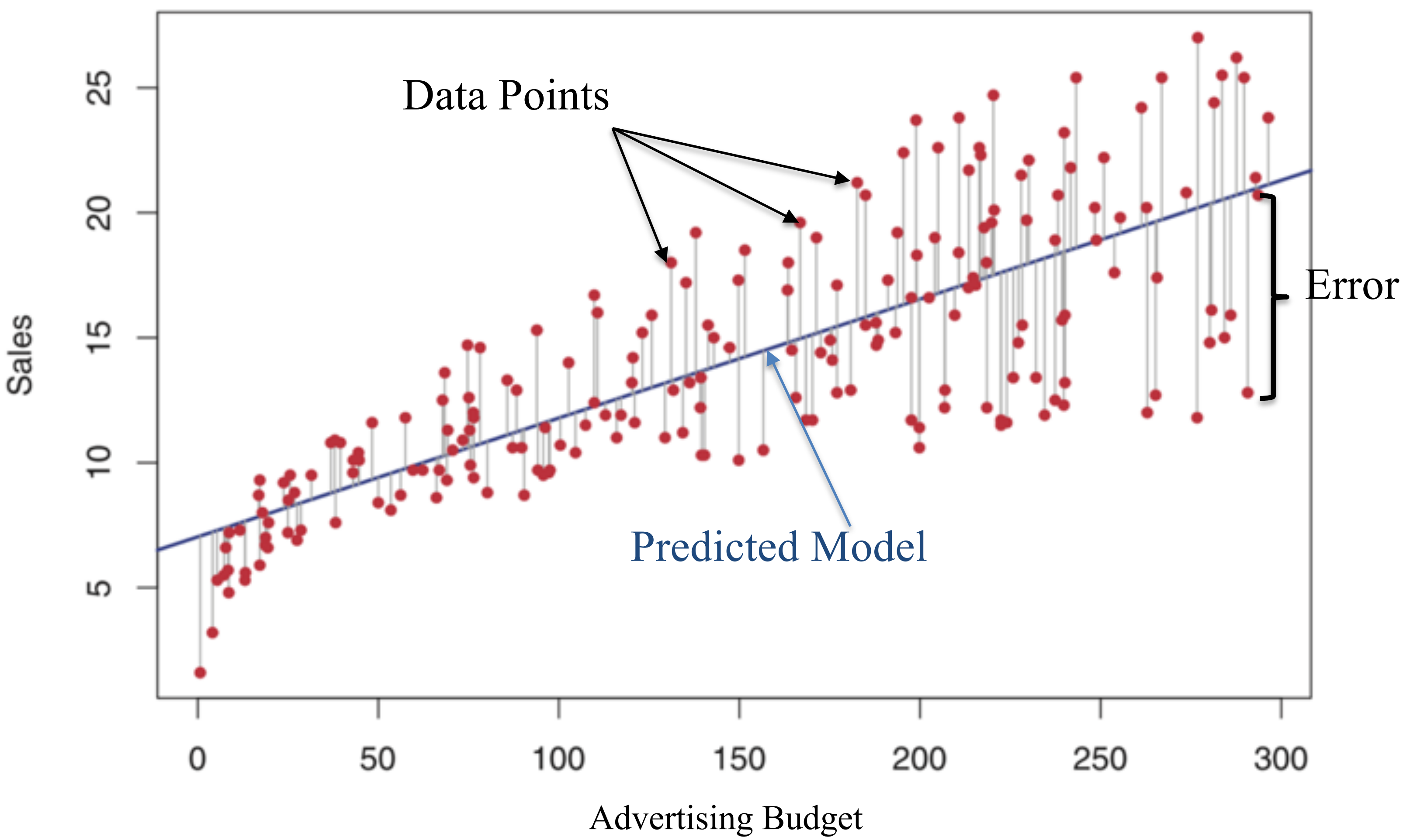
$$h_\theta(x) = \theta_0 + \theta_1 x$$

to represent house price. Let's say it's a one-D problem and only independent attribute is house size x.



Linear Regression

- Linear Regression – find best model that fits a continuous (i.e. real valued) output



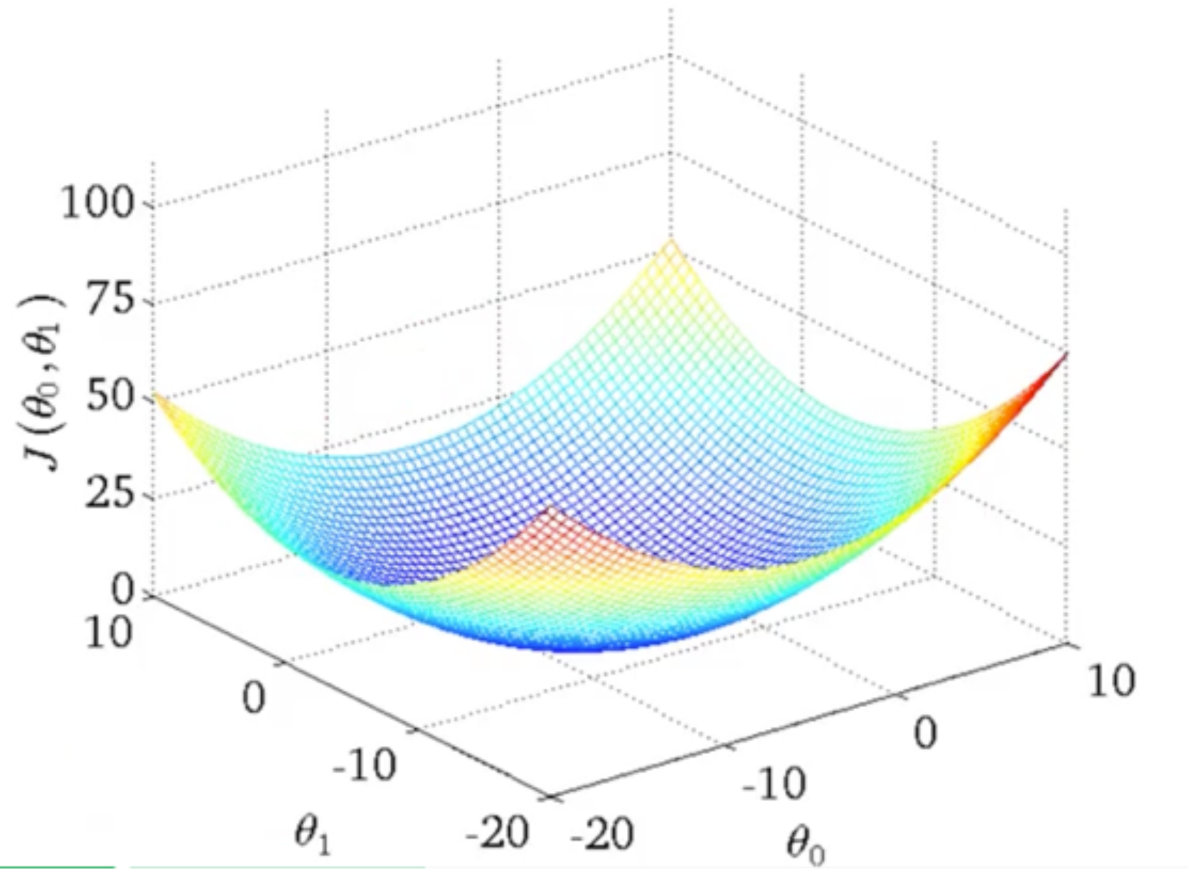
Error Function

- Our aim can be stated as:
  Choose parameters $\theta_0 \; and \; \theta_1$ such that our hypothesis $h_\theta(x)$ is as close to y for our training examples.
- Mathematically, choose parameters such that the following is minimized (called error or cost function). m is the number of training instances

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

Error function

- How does J vary wrt the parameters
- Contour plot
- We are looking for the minima
- How do we get there?



Gradient Descent

- Given a function J of parameters Θ, how do we find its minimum or maximum.
- Gradient Descent is a very powerful and popular algorithm.
- Widely used in machine learning
- In many cases, analytical solution is not possible, so we have to randomly take steps in search of minimum.

Gradient Descent

- Aim: We have a function $J(\theta_0, \theta_1)$, and we want

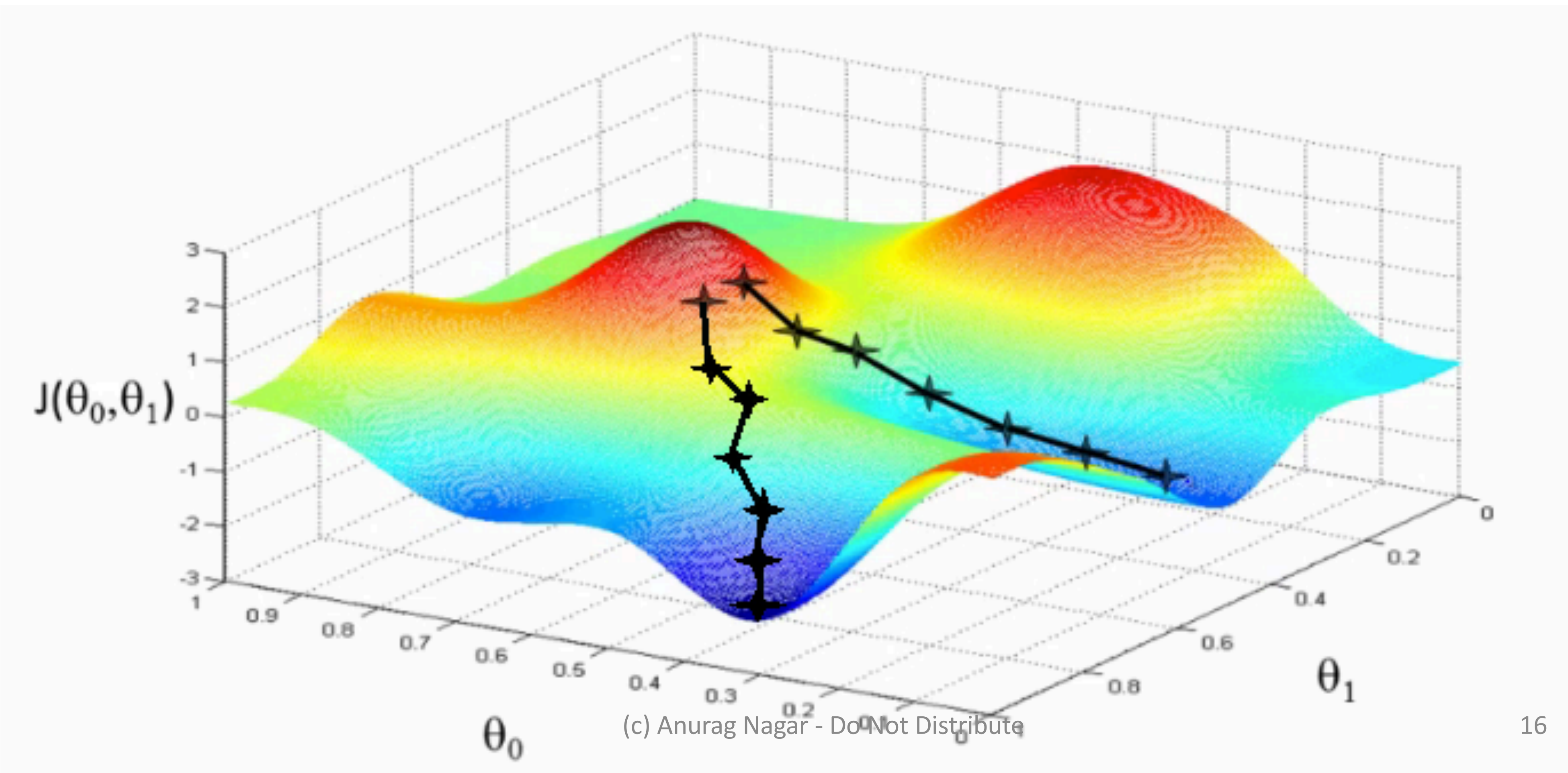
$$\underset{\theta_0 \, \theta_1}{\operatorname{argmin}} \; J(\theta_0, \theta_1)$$

STEPS:

- Start with some random values
- Keep changing these values such that you achieve a reduction in J

Gradient Descent

- Imagine a man at a random point on the mountains.
- He needs to reach the city by walking randomly



Gradient descent algorithm

$$\text{repeat until convergence } \{$$
$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \quad (\text{for } j = 0 \text{ and } j = 1)$$
$$\}$$

Correct: Simultaneous update

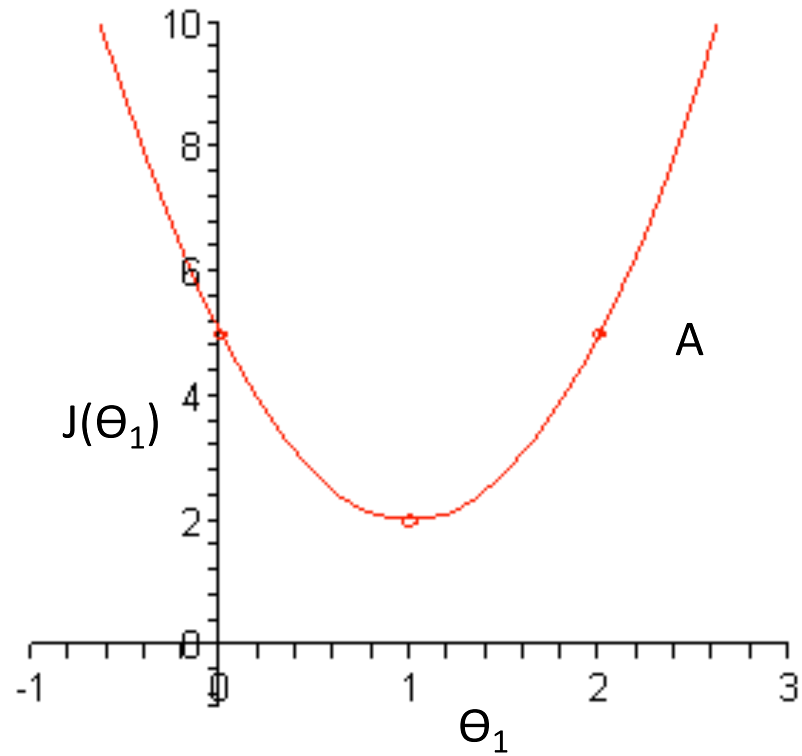
$$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$
$$\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$
$$\theta_0 := \text{temp0}$$
$$\theta_1 := \text{temp1}$$

α is called the learning rate
Intuition: It is how big a step you are taking.

Illustration

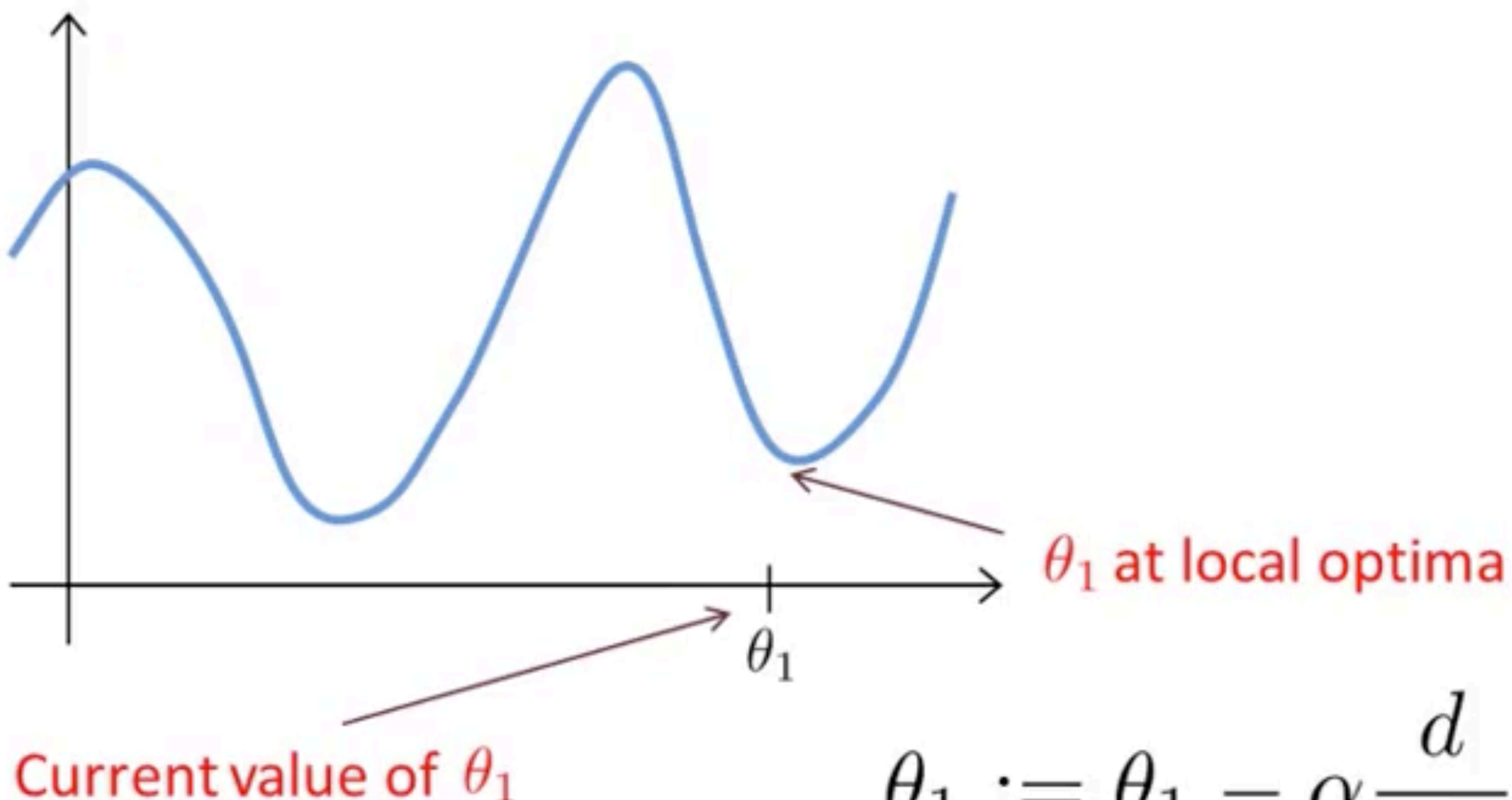
- In the curve on the right, imagine you are at point A
- The slope there is positive
- Update rule:

$$\theta_1 = \theta_1 - \alpha \frac{\partial J}{\partial \theta_1}$$



since $\frac{\partial J}{\partial \theta_1}$ is positive and α is always positive, we would move towards left.

Local Minima can be a problem



$$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$$

Gradient Descent for Linear Regression

Square meters	Bedrooms	Floors	Age of building (years)	Price in 1000€
x1	*x2*	*x3*	*x4*	*y*
200	5	1	45	460
131	3	2	40	232
142	3	2	30	315
756	2	1	36	178
…	…	…	…	…

- **Notation**

n – number of features (here $n = 4$)

$x^{(i)}$ – input features of ith training example

$x_j^{(i)}$ – feature j in ith training example

$$x^{(3)} = \begin{bmatrix} 142 \\ 3 \\ 2 \\ 40 \end{bmatrix}$$

$$x_1^{(4)} = 756$$

Gradient Descent for Linear Regression

Hypothesis representation

- $h_\theta(x_1, \ldots, x_n) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$
- **More compact**

$$\mathrm{x} = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{n+1}, \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix} \in \mathbb{R}^{n+1}$$ **with definition** $\mathrm{x}_0 := 1$

$$h_\theta(x) = \begin{bmatrix} \theta_0 & \theta_1 & \ldots & \theta_n \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}$$

$$= \theta^T x$$

Gradient Descent for Linear Regression

Gradient descent for multiple variables

- **Generalized cost function** $J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$
- **Generalized gradient descent**

$$
\begin{array}{l}
\textbf{while} \ \text{not converged:} \\
\quad \textbf{for} \ \text{all } j: \\
\qquad tmp_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \\
\quad \theta := \begin{bmatrix} tmp_0 \\ \vdots \\ tmp_n \end{bmatrix}
\end{array}
$$

Gradient Descent for Linear Regression

Partial derivative of cost function for multiple variables

- **Calculating the partial derivative**

$$
\begin{aligned}
\frac{\partial}{\partial\theta_j} J(\theta) &= \frac{\partial}{\partial\theta_j} \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 \\
&= \frac{\partial}{\partial\theta_j} \frac{1}{2m} \sum_{i=1}^{m} \left((\theta_0 x_0^{(i)} + \dots + \theta_n x_n^{(i)}) - y^{(i)}\right)^2 \\
&= \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right) x_j^{(i)}
\end{aligned}
$$

Gradient Descent for Linear Regression

Gradient descent for multiple variables

- **Simplified gradient descent**

$$
\begin{aligned}
&\textbf{while not } \text{converged:}\\
&\quad \textbf{for } \text{all } j\text{:}\\
&\quad\quad tmp_j := \theta_j - \alpha \tfrac{1}{m} \textstyle\sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) x_j^{(i)}\\
&\quad \theta := \begin{bmatrix} tmp_0 \\ \vdots \\ tmp_n \end{bmatrix}
\end{aligned}
$$

Regression Evaluation Metrics

- Suppose we propose a linear model:

$$Y = \beta_0 + \beta_1 X + \epsilon$$

  where ϵ represents the error.

- The coefficients β_0 and β_1 need to be estimated from the data (using gradient descent or other computational techniques).
- Let's suppose our estimates are $\widehat{\beta_0}$ and $\widehat{\beta_1}$, then the predicted value would be:

$$\hat{y} = \widehat{\beta_0} + \widehat{\beta_1} X$$

Regression Evaluation Metrics

- $e_i = \widehat{y_i} - y_i$ represents the residual or error for the ith data point.
- Residual sum of square (RSS) is defined as:

$$RSS = e_1^2 + e_2^2 + \cdots + e_n^2$$

- By minimizing the RSS, we can arrive at the estimates $\widehat{\beta_0}$ and $\widehat{\beta_1}$

Another evaluation metric

- We would like to check what fraction of data variance is explained by the model.
- R^2 statistic measures this:

$$R^2 = 1 - \frac{RSS}{TSS}$$

where RSS is the residual sum of squares (defined earlier) and TSS is the total sum of squares: $TSS = \sum_{i=1}^{n}(y_i - \bar{y})^2$

Regression Packages

- For Python, see https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.linregress.html

- For R see, http://r-statistics.co/Linear-Regression.html

Practice Question

- Consider the problem of predicting the number of A grades that a student at UTD will obtain in second year of M.S. based on the number of A grades obtained in the first year of M.S. course.

  Below is the data:

x	y
3	2
1	2
0	1
4	3

x represents the number of A grades in 1st year
y represents the number of A grades in 2nd year

You decide to use a hypothesis of the form $h_\theta(x) = \theta_0 + \theta_1 x$ where θ_0=0 and θ_1=1. Find the value of the squared error?