Example: Advertising Data

- Data taken from An Introduction to Statistical Learning with Applications in R (http://www-bcf.usc.edu/~gareth/ISL/data.html)
- Data consists of the advertising budgets for three media (TV, radio and newspapers) and the overall sales in 200 different markets.

<table>
<thead>
<tr>
<th></th>
<th>TV</th>
<th>Radio</th>
<th>Newspaper</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>230.1</td>
<td>37.8</td>
<td>69.2</td>
<td>22.1</td>
</tr>
<tr>
<td></td>
<td>44.5</td>
<td>39.3</td>
<td>45.1</td>
<td>10.4</td>
</tr>
<tr>
<td></td>
<td>17.2</td>
<td>45.9</td>
<td>69.3</td>
<td>9.3</td>
</tr>
<tr>
<td></td>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
</tbody>
</table>

...
Suppose we want to predict sales in a new area?
Predict sales when the TV advertising budget is increased to 350?
... Draw a line that fits through the data points
Recall Notation

Training data:

<table>
<thead>
<tr>
<th>TV (x)</th>
<th>Sales (y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>230.1</td>
<td>22.1</td>
</tr>
<tr>
<td>44.5</td>
<td>10.4</td>
</tr>
<tr>
<td>17.2</td>
<td>9.3</td>
</tr>
</tbody>
</table>

- $m$ = number of training examples
- $x$ = “input” variable/features
- $y$ = “output” variable/“target” variable

- $(x^{(i)}, y^{(i)})$ the $i$th training example
- $x^{(1)} = 230.1$, $y^{(1)} = 22.1$, $x^{(2)} = 44.5$, $y^{(2)} = 10.4$
Linear Model

- Prediction: $y = h_\theta(x) = \theta_0 + \theta_1 x$
- $\theta_0, \theta_1$ are (unknown) parameters

\[ \begin{align*}
\theta_0 &= 15, \quad \theta_1 = 0 \\
\theta_0 &= 0, \quad \theta_1 = 0.1 \\
\theta_0 &= 15, \quad \theta_1 = 0.1
\end{align*} \]
Cost Function: How to choose model parameters $\theta$?

- Prediction: $y = h_\theta(x) = \theta_0 + \theta_1 x$
- Idea: Choose $\theta_0$ and $\theta_1$ so that $h_\theta(x^{(i)})$ is close to $y^{(i)}$ for each of our training examples $(x^{(i)}, y^{(i)}), i = 1, \ldots, m$.
- Least squares case: select the values for $\theta_0$ and $\theta_1$ that minimise cost function:

$$J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$
Example: Advertising Data
Linear Regression with Multiple Variables

Advertising example:

<table>
<thead>
<tr>
<th>TV</th>
<th>Radio</th>
<th>Newspaper</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>$x_2$</td>
<td>$x_3$</td>
<td>$y$</td>
</tr>
<tr>
<td>230.1</td>
<td>37.8</td>
<td>69.2</td>
<td>22.1</td>
</tr>
<tr>
<td>44.5</td>
<td>39.3</td>
<td>45.1</td>
<td>10.4</td>
</tr>
<tr>
<td>17.2</td>
<td>45.9</td>
<td>69.3</td>
<td>9.3</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

- $n$=number of features (3 in this example)
- $(x^{(i)}, y^{(i)})$ the $i$th training example e.g.

$$x^{(1)} = [230.1, 37.8, 69.2]^T = \begin{bmatrix} 230.1 \\ 37.8 \\ 69.2 \end{bmatrix}$$

- $x_j^{(i)}$ is feature $j$ in the $i$th training example, e.g. $x_2^{(1)} = 37.8$
Linear Regression with Multiple Variables

Hypothesis: \( h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 \)
e.g. \( h_\theta(x) = 15 + 0.1 x_1 - 5 x_2 + 10 x_3 \)

\( Sales \) \( TV \) \( Radio \) \( Newspaper \)

More generally, when have \( n \) features:

- For convenience, define \( x_0 = 1 \)
i.e. \( x_0^{(1)} = 1, x_0^{(2)} = 1 \) etc

- Feature vector \( x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \)

- Parameter vector \( \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix} \)

- \( h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n = \theta^T x \)
Linear Regression with Multiple Variables

- Hypothesis: $h_\theta(x) = \theta^T x$
- Parameters: $\theta$
- Cost Function: $J(\theta) = J(\theta) = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$
- Learn Parameters: Select $\theta$ that minimises $J(\theta)$. E.g. can find $\theta$ using gradient descent.
Gradient Descent with Multiple Variables

For $J(\theta) = \sum_{i=1}^{m} (h_{\theta}(x^{(i)}) - y^{(i)})^2$ with $h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n$:

- $\frac{\partial}{\partial \theta_0} J(\theta) = \frac{2}{m} \sum_{i=1}^{m} (h_{\theta}(x^{(i)}) - y^{(i)})$
- $\frac{\partial}{\partial \theta_1} J(\theta) = \frac{2}{m} \sum_{i=1}^{m} (h_{\theta}(x^{(i)}) - y^{(i)}) x_1^{(i)}$
- $\frac{\partial}{\partial \theta_j} J(\theta) = \frac{2}{m} \sum_{i=1}^{m} (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)}$

So gradient descent algorithm is:

- Start with some $\theta$
- Repeat:
  
  for $j=0$ to $n$ \{ $\text{temp}_j := \theta_j - \frac{2\alpha}{m} \sum_{i=1}^{m} (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)}$ \}
  
  for $j=0$ to $n$ \{ $\theta_j := \text{temp}_j$ \}