Today

• (Ch 13) Regression
  • The regression problem
  • Training a linear regression model using least squares
  • Evaluating a model using the R-squared metric

Next lecture

• (Ch 13) Regression
  • Outliers, overfitting and regularization
  • Nearest neighbors regression
A charming house minutes from Apple HQ

Source: zillow.com
Wait ... is that a reasonable price?

10341 N Portal Ave  
Cupertino, CA 95014

4 beds • 3 baths • 2,621 sqft

Extensive Luxury Remodel, Fantastic Price Per Square Foot of $1,101.87!

Facts and Features

<table>
<thead>
<tr>
<th>Type</th>
<th>Year Built</th>
<th>Heating</th>
<th>Parking</th>
<th>Lot</th>
<th>Price/sqft</th>
<th>Saves</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Family</td>
<td>1910</td>
<td>Forced air</td>
<td>5 spaces</td>
<td>0.25 acres</td>
<td>$1,064</td>
<td>29</td>
</tr>
</tbody>
</table>

Source: zillow.com
Can we use data to predict the sale price?

Source: julianalee.com/cupertino/cupertino-home-sales.htm
The regression problem

- Given a set of feature vectors $x_i$ where each has a numerical label $y_i$, we want to train a model that can map unlabeled vectors to numerical values.

- We can think of regression as fitting a line (or curve or hyperplane, etc.) to data.

- Regression is like classification except that the prediction target is a number, not a class label (and that changes everything).
Some terminology

• Suppose the dataset \{ (x, y) \} consists of N labeled items \( (x_i, y_i) \)

• If we represent the dataset as a table
  • The \( d \) columns representing \{ x \} are called explanatory variables \( x^{(j)} \)
  • The numerical column \( y \) is called the dependent variable

\[\begin{array}{ccc}
  x^{(1)} & x^{(2)} & y \\
  1 & 3 & 0 \\
  2 & 3 & 2 \\
  3 & 6 & 5 \\
\end{array}\]
Linear model

• We begin by modeling \( y \) as a linear function of \( x^{(j)} \) plus randomness

\[
y = x^{(1)} \beta_1 + x^{(2)} \beta_2 + \cdots + x^{(d)} \beta_d + \xi
\]

where \( \xi \) is a zero-mean random variable that represents model error

• In vector notation

\[
y = x^T \beta + \xi
\]

where \( \beta \) is the \( d \)-dimensional vector of coefficients that we train
Each data item gives an equation ...

Model: \( y = \mathbf{x}^T \boldsymbol{\beta} + \xi = \mathbf{x}^{(1)} \beta_1 + \mathbf{x}^{(2)} \beta_2 + \xi \)

Training data

<table>
<thead>
<tr>
<th>( \mathbf{x}^{(1)} )</th>
<th>( \mathbf{x}^{(2)} )</th>
<th>( y )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>5</td>
</tr>
</tbody>
</table>
... which together form a matrix equation
Training the model means choosing $\beta$

- Given a training dataset $\{(x, y)\}$, we want to fit a model $y = x^T \beta + \xi$

- Define $y = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$ and $X = \begin{bmatrix} x_1^T \\ \vdots \\ x_N^T \end{bmatrix}$ and $e = \begin{bmatrix} \xi_1 \\ \vdots \\ \xi_N \end{bmatrix}$

- To train the model, we must choose $\beta$ that makes $e$ small in the matrix equation

$$y = X\beta + e$$
Training using least squares

• In the least squares method, we aim to minimize $||e||^2$

$$||e||^2 = ||y - X\beta||^2 = (y - X\beta)^T(y - X\beta)$$

• Differentiating and setting to zero (and skipping some matrix calculus) gives

$$X^TX\beta - X^Ty = 0$$

• If $X^TX$ is invertible, the least squares estimate of the coefficients is

$$\hat{\beta} = (X^TX)^{-1}X^Ty$$
Training using least squares example

Model: \( y = x^{(1)} \beta_1 + x^{(2)} \beta_2 + \xi = x^T \beta + \xi \)

Training data

<table>
<thead>
<tr>
<th>(x^{(1)})</th>
<th>(x^{(2)})</th>
<th>(y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>5</td>
</tr>
</tbody>
</table>
Prediction

• If we train the model with coefficients $\beta$, we can predict $y_0^p$ from $x_0$

\[ y_0^p = x_0^T \beta \]

• In the model $y = x^{(1)} \beta_1 + x^{(2)} \beta_2 + \xi$ with $\beta = \begin{bmatrix} 2 \\ -1/3 \end{bmatrix}$

  • the prediction for $x_0 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ is $y_0^p = \quad$

  • the prediction for $x_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ is $y_0^p = \quad$
A linear model with constant offset

• The problem with the model $y = x^{(1)} \beta_1 + x^{(2)} \beta_2 + \xi$ is that it always predicts $y_0^p = 0$ if the input feature vector $x_0 = [0, 0]^T$.

• Let’s add a constant offset $\beta_0$ to the model

$$y = \beta_0 + x^{(1)} \beta_1 + x^{(2)} \beta_2 + \xi$$
Training and prediction with constant offset

Model: \[ y = \beta_0 + x^{(1)} \beta_1 + x^{(2)} \beta_2 + \xi = x^T \beta + \xi \]

Training data

<table>
<thead>
<tr>
<th>x^{(1)}</th>
<th>x^{(2)}</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>5</td>
</tr>
</tbody>
</table>
Evaluating models using R-squared

• The least squares estimate satisfies this property (proven in book)

\[ \text{var}(\{y_i\}) = \text{var}(\{x_i^T \beta\}) + \text{var}(\{\xi_i\}) \]

• This property gives us an evaluation metric called R squared

\[ R^2 = \frac{\text{var}(\{x_i^T \hat{\beta}\})}{\text{var}(\{y_i\})} \]

• We have \( 0 \leq R^2 \leq 1 \) with a larger value meaning a better fit
R-squared examples

Chirp frequency vs temperature in crickets

Heart rate vs temperature in humans
Comparing our example models

\[ y = x^{(1)} \beta_1 + x^{(2)} \beta_2 + \xi \]

\[ y = \beta_0 + x^{(1)} \beta_1 + x^{(2)} \beta_2 + \xi \]

\[
\begin{array}{cccc}
  x^{(1)} & x^{(2)} & y & x^T \hat{\beta} \\
  1 & 3 & 0 & 1 \\
  2 & 3 & 2 & 3 \\
  3 & 6 & 5 & 4 \\
\end{array}
\]

\[
\hat{\beta} = \begin{bmatrix} 2 \\ -1/3 \end{bmatrix}
\]

\[
\begin{array}{cccc}
  1 & x^{(1)} & x^{(2)} & y & x^T \hat{\beta} \\
  1 & 1 & 3 & 0 & 0 \\
  1 & 2 & 3 & 2 & 2 \\
  1 & 3 & 6 & 5 & 5 \\
\end{array}
\]

\[
\hat{\beta} = \begin{bmatrix} -3 \\ 2 \\ 1/3 \end{bmatrix}
\]