Regression: Predicting House Prices

STAT/CSE 416: Intro to Machine Learning
Emily Fox
University of Washington
March 29, 2018

©2018 Emily Fox

How much is my house worth?

$$ ????
Data

\( (x_1 = \text{sq.ft.}, y_1 = \$) \)
\( (x_2 = \text{sq.ft.}, y_2 = \$) \)
\( (x_3 = \text{sq.ft.}, y_3 = \$) \)
\( (x_4 = \text{sq.ft.}, y_4 = \$) \)
\( (x_5 = \text{sq.ft.}, y_5 = \$) \)
\( \vdots \)

Input vs. Output:
- \( y \) is the quantity of interest
- Assume \( y \) can be predicted from \( x \)

Model – How we assume the world works

Regression model:
\[ y_i = f(x_i) + \epsilon_i \]
\[ E[\epsilon_i] = 0 \star \]

\( \underline{\text{Regression model:}} \)

\[ y_i = f(x_i) + \epsilon_i \]
\[ E[\epsilon_i] = 0 \star \]

\( \underline{\text{Expected value}} \)
Use a simple **linear** regression model

Fit a line through the data

\[ y_i = w_0 + w_1 x_i + \varepsilon_i \]

\[ f(x) = w_0 + w_1 x \]

**parameters**

**of model**

**Training Data** 📜 --> **Feature extraction** 📚 --> **ML model** 🤖 --> **ŷ** 📣

\[ \hat{w} = (\hat{w}_0, \hat{w}_1) \]

**ML algorithm** 🤖

**Quality metric** 📃

---

©2018 Emily Fox
"Cost" of using a given line

Residual sum of squares (RSS)

\[
\text{RSS}(w_0, w_1) = \sum_{i=1}^{N} (y_i - [w_0 + w_1 x_i])^2
\]

Find "best" line

Minimize cost over all possible \(w_0, w_1\)

\[
\text{RSS}(w_0, w_1) = \sum_{i=1}^{N} (y_i - [w_0 + w_1 x_i])^2
\]

RSS(\(w_0=1.1, w_1=0.8\)) \#3

RSS(\(w_0=0.98, w_1=0.87\)) \#2

RSS(\(w_0=0.97, w_1=0.85\)) \#1
Minimizing the cost

Minimize function over all possible $w_0, w_1$

$$\min_{w_0, w_1} \sum_{i=1}^{N} (y_i - [w_0 + w_1x_i])^2$$

$\text{RSS}(w_0, w_1)$ is a function of 2 variables
Gradient descent

**Algorithm:**

while not converged

\[ w^{(t+1)} \leftarrow w^{(t)} - \eta \nabla \text{RSS}(w^{(t)}) \]

\( \hat{W} \)

best guess for this dataset

---

The fitted line: use + interpretation
Model vs. fitted line

\[ \hat{y}(x) = \hat{w}_0 + \hat{w}_1 x \]

Regression model:

\[ y_i = w_0 + w_1 x_i + \epsilon_i \]

Estimated parameters:

\[ \hat{w}_0, \hat{w}_1 \]

Seller:
Predicting your house price

\[ \hat{y}(x) = \hat{w}_0 + \hat{w}_1 x \]

Regression model:

\[ y_i = w_0 + w_1 x_i + \epsilon_i \]

Best guess of your house price:

\[ \hat{y}_{\text{house}} = \hat{w}_0 + \hat{w}_1 \text{sq.ft.}_{\text{house}} \]
Buyer:
Predicting size of house

Regression model:
\[ y_i = w_0 + w_1 x_i + \varepsilon_i \]

Best guess of size of house you can afford:
\[ \text{\$in\ bank} = \hat{w}_0 + \hat{w}_1 \text{sq.ft.} \]

A concrete example

\[ \hat{f}(x) = -44850 + 280.76 x \]

Predict \$ of 2,640 sq.ft. house:
\[
-44850 + 280.76 \times 2,640 \\
= \$696,356
\]

Predict sq.ft. of $859,000 sale:
\[
\frac{859000+44850}{280.76} \\
= 3,219 \text{ sq.ft.}
\]
Interpreting the coefficients

\[ \hat{y} = \hat{w}_0 + \hat{w}_1 x \]

Predicted $ of house with sq.ft. = 0 (just land)

Interpreting the coefficients

\[ \hat{y} = \hat{w}_0 + \hat{w}_1 x \]

predicted change in $

predicted change in $ per unit change in sq.ft.

1 sq. ft.
Interpreting the coefficients

\[ \hat{y} = \hat{w}_0 + \hat{w}_1 x \]

**Warning:** magnitude depends on units of both features and observations

A concrete example

\[ \hat{f}(x) = -$44,850 + 280.76 ($/sq.ft.) x \]

Predict $ of 2,640 sq.ft. house:

\[-$44,850 + 280.76 ($/sq.ft.) \times 2,640 \text{ sq.ft.} = $696,356\]

Predict sq.ft. of $859,000 sale:

\[ ($859,000 + $44,850) / 280.76 ($/sq.ft.) = 3,219 \text{ sq.ft.} \]
A concrete example

\[ f(x) = -44,850 + 280.76 \text{ ($/sq.ft.$) x} \]

But what if:
- House was measured in square meters?
- Price was measured in RMB?

Adding higher order effects
You show your friend your analysis.

Dude, it’s not a linear relationship!
What about a quadratic function?

$$f_w(x) = w_0 + w_1 x + w_2 x^2$$

Dude, it’s not a linear relationship!
Even higher order polynomial

\[ f(x) = w_0 + w_1 x + w_2 x^2 + \ldots + w_p x^p \]

Polynomial regression

Model:

\[ y_i = w_0 + w_1 x_i + w_2 x_i^2 + \ldots + w_p x_i^p + \varepsilon_i \]

treat as different features

feature 1 = 1 (constant)  \quad parameter 1 = w_0

feature 2 = x  \quad parameter 2 = w_1

feature 3 = x^2  \quad parameter 3 = w_2

...  

feature p+1 = x^p  \quad parameter p+1 = w_p
Other functions of one input

Motivating application: Detrending time series

\[ y_i = \text{\$ of } i^{th} \text{ house sale} \]
\[ t_i = \text{month of } i^{th} \text{ house sale} \]

House sales recorded monthly
Trends over time

On average, house prices tend to increase with time.

Seasonality

Most houses listed in summer
- Good houses sell quickly

Few homes listed in Nov./Dec.
- Transactions often leftover inventory or special circumstances
An example detrending

Model:
\[ y_i = w_0 + w_1 t_i + w_2 \sin\left(\frac{2\pi t_i}{12} - \Phi\right) + \varepsilon_i \]

Linear trend
Seasonal component = Sinusoid with period 12 (resets annually)
Unknown phase/shift

Trigonometric identity: \( \sin(a-b) = \sin(a)\cos(b) - \cos(a)\sin(b) \)
\[ \Rightarrow \sin\left(\frac{2\pi t_i}{12} - \Phi\right) = \sin\left(\frac{2\pi t_i}{12}\right)\cos(\Phi) - \cos\left(\frac{2\pi t_i}{12}\right)\sin(\Phi) \]

An example detrending

Equivalently,
\[ y_i = w_0 + w_1 t_i + w_2 \sin\left(\frac{2\pi t_i}{12}\right) + w_3 \cos\left(\frac{2\pi t_i}{12}\right) + \varepsilon_i \]

feature 1 = 1 (constant)
feature 2 = \( t \)
feature 3 = \( \sin(2\pi t / 12) \)
feature 4 = \( \cos(2\pi t / 12) \)
Detrended housing data

Fit polynomial trend and sinusoidal seasonal component

5th order polynomial + sin/cos basis

Zoom in...

Fit polynomial trend and sinusoidal seasonal component

5th order polynomial + sin/cos basis
Other examples of seasonality

- Motion capture data
- Weather modeling (e.g., temperature, rainfall)
- Demand forecasting (e.g., jacket purchases)
- Flu monitoring

Generic basis expansion

Model:
\[ y_i = \sum_{j=0}^{D} w_j h_j(x_i) + \epsilon_i \]

- \( y_i \) is the observed value at time \( i \)
- \( w_j \) is the \( j^{th} \) regression coefficient or weight
- \( h_j(x_i) \) is the \( j^{th} \) feature
Generic basis expansion

Model:
\[ y_i = \sum_{j=0}^{D} w_j h_j(x_i) + \varepsilon_i \]

\[ y_i = w_0 h_0(x_i) + w_1 h_1(x_i) + ... + w_D h_D(x_i) + \varepsilon_i \]

*feature 1 = \(h_0(x)\)...often 1 (constant)*
*feature 2 = \(h_1(x)\)... e.g., \(x\)*
*feature 3 = \(h_2(x)\)... e.g., \(x^2\) or \(\sin(2\pi x/12)\) or \(\log(x)\)*
...
*feature D+1 = \(h_D(x)\)... e.g., \(x^p\)*

Adding other inputs
Predictions just based on house size

only 1 bathroom! Not same as my 3 bathrooms

Add more inputs

\[ f(x) = w_0 + w_1 \text{sq.ft.} + w_2 \#\text{bath} \]
Many possible inputs

- Square feet
- # bathrooms
- # bedrooms
- Lot size
- Year built
- ...

General notation

Output: $y$ \(\text{scalar}\)

Inputs: $x = (x[1], x[2], ..., x[d])$

\(d\)-dim vector

Notational conventions:
- $x[j] = j^{th}$ input \((\text{scalar})\)
- $h_j(x) = j^{th}$ feature \((\text{scalar})\)
- $x_i = \text{input of } i^{th} \text{ data point} \,(\text{vector})$
- $x_i[j] = j^{th}$ input of $i^{th}$ data point \((\text{scalar})\)
**Generic linear regression model**

Model:

\[ y_i = w_0 h_0(x_i) + w_1 h_1(x_i) + \ldots + w_D h_D(x_i) + \varepsilon_i \]

\[ = \sum_{j=0}^{D} w_j h_j(x_i) + \varepsilon_i \]

- feature 1 = \( h_0(x) \) ... e.g., 1
- feature 2 = \( h_1(x) \) ... e.g., \( x[1] = \text{sq. ft.} \)
- feature 3 = \( h_2(x) \) ... e.g., \( x[2] = \# \text{bath} \)
  - or, \( \log(x[7]) \cdot x[2] = \log(\# \text{bed}) \cdot \# \text{bath} \)

...  
- feature \( D+1 = h_D(x) \) ... some other function of \( x[1], \ldots, x[d] \)

**Inputs vs. features**

Hi ML expert, here is a data table to analyze.
More on notation

# observations \((x_i, y_i) : N\)
# inputs \(x[j] : d\)
# features \(h_j(x) : D\)

RSS for multiple regression

\[
\text{RSS}(w) = \sum_{i=1}^{N} (y_i - \sum_{j=0}^{D} w_j h_j(x_i))^2
\]

- price ($)
- square feet (sq.ft.)
- # bathrooms
- \(x[1]\)
- \(x[2]\)
How many features to use?

- More on this soon!

A compact representation
Representing function using vectors

\[ f(x) = w_0 h_0(x) + w_1 h_1(x) + \ldots + w_D h_D(x) = \sum_{j=0}^{D} w_j h_j(x) \]

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 5 & 3 & 0 & 0 & 1 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

\[
\begin{bmatrix}
3 & 0 & 0 & 0 & 2 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0
\end{bmatrix}
\]

As an algorithm...

\[ f(x) = w_0 h_0(x) + w_1 h_1(x) + \ldots + w_D h_D(x) = \sum_{j=0}^{D} w_j h_j(x) \]

Algorithm:

for \( j = 0, \ldots, D \)  
\[ f(x)_{\tilde{x}} = w_j h_j(x) \]
Compact notation

\[ f(x) = w_0 h_0(x) + w_1 h_1(x) + \ldots + w_D h_D(x) = \sum_{j=0}^{D} w_j h_j(x) \]

\[
\begin{array}{cccccccc}
1 & 0 & 0 & 0 & 5 & 3 & 0 & 0 \\
3 & 0 & 0 & 0 & 2 & 0 & 0 & 1
\end{array}
\]

\[
\begin{array}{cccccccc}
1 & 0 & 0 & 0 & 5 & 3 & 0 & 0 \\
3 & 0 & 0 & 0 & 2 & 0 & 0 & 1
\end{array}
\]

= \begin{array}{c} h(x_i) \\ \text{feat. vec.} \end{array}

\[
\begin{array}{cccccccc}
1 & 0 & 0 & 0 & 5 & 3 & 0 & 0 \\
3 & 0 & 0 & 0 & 2 & 0 & 0 & 1
\end{array}
\]

= \begin{array}{c} w \\ \text{weights vec.} \end{array}

\[
\begin{array}{cccccccc}
1 & 0 & 0 & 0 & 5 & 3 & 0 & 0 \\
3 & 0 & 0 & 0 & 2 & 0 & 0 & 1
\end{array}
\]

= \begin{array}{c} \mathbf{w}^\top h(x_i) \end{array}

\[
\begin{array}{cccccccc}
1 & 0 & 0 & 0 & 5 & 3 & 0 & 0 \\
3 & 0 & 0 & 0 & 2 & 0 & 0 & 1
\end{array}
\]

= \begin{array}{c} \mathbf{w}^\top h(x_i) \end{array}

\[
\begin{array}{cccccccc}
1 & 0 & 0 & 0 & 5 & 3 & 0 & 0 \\
3 & 0 & 0 & 0 & 2 & 0 & 0 & 1
\end{array}
\]