Linear Regression: Model and Algorithms

CSE 446: Machine Learning
Emily Fox
University of Washington
January 9, 2017

Linear regression: The model
How much is my house worth?

I want to list my house for sale

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:

\[y_i = f(x_i) + \epsilon_i\]

\[E[\epsilon_i] = 0\]

Expected relationship between x and y
Model – How we assume the world works

“Essentially, all models are wrong, but some are useful.”
George Box, 1987.

Simple linear regression model

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

\[ f(x) = w_0 + w_1 x \]
Simple linear regression model

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

parameters:
regression coefficients

What about a quadratic function?

\[ f(x) = w_0 + w_1 x + w_2 x^2 \]
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 \text{parameter 1} = w_0
feature 2 = x  \quad \text{parameter 2} = w_1
feature 3 = x^2  \quad \text{parameter 3} = w_2
\ldots
feature p+1 = x^p  \quad \text{parameter p+1} = w_p
Generic basis expansion

Model:
\[ y_i = w_0 h_0(x_i) + w_1 h_1(x_i) + \ldots + w_D h_D(x_i) + \epsilon_i \]
\[ = \sum_{j=0}^{D} w_j h_j(x_i) + \epsilon_i \]

- 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 \( \sin(2\pi x / 12) \)
- ...
- feature D+1 = \( h_D(x) \)...e.g., \( x^p \)
Predictions just based on house size

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

Add more inputs
Many possible inputs

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

General notation

Output: \( y \) \( \text{scalar} \)
Inputs: \( x = (x[1], x[2], \ldots, x[d]) \) \( \text{d-dim vector} \)

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

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

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

- 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, \( \log(x[7]) \times x[2] = \log(\# \text{bed}) \times \# \text{bath} \)
- ... feature D+1 = \( h_D(x) \) ... some other function of \( x[1], \ldots, x[d] \)
Step 1:
Rewrite the regression model

Rewrite in matrix notation

For observation $i$

$$y_i = \sum_{j=0}^{D} w_j h_j(x_i) + \epsilon_i$$

$$Y_i = \begin{bmatrix} w_0 & w_1 & \cdots & w_D \end{bmatrix} = \begin{bmatrix} h_0(x_i) \\ h_1(x_i) \\ \vdots \\ h_D(x_i) \end{bmatrix} + \epsilon_i$$

$$= \omega^T h(x_i) + \epsilon_i = h^T(x_i) \omega + \epsilon_i$$
Rewrite in matrix notation

For all observations together

\[
\begin{array}{c}
\mathbf{w} \\
\mathbf{x}_1 \\
\vdots \\
\mathbf{x}_N
\end{array} 
+ \begin{array}{c}
\mathbf{b} \\
\mathbf{e}_1 \\
\vdots \\
\mathbf{e}_N
\end{array} = \mathbf{y}
\implies \mathbf{y} = \mathbf{Hw} + \mathbf{e}
\]

Step 2: Compute the cost
"Cost" of using a given line

Residual sum of squares (RSS)

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

RSS for multiple regression

\[
\text{RSS}(w) = \sum_{i=1}^{N} (y_i - h(x_i) w)^2 = (y - Hw)^T(y - Hw)
\]

Show this at home!
Step 3:
Take the gradient

\[ \nabla \text{RSS}(w) = \nabla [(y - Hw)^T(y - Hw)] \]
\[ = - 2H^T(y - Hw) \]

Why? By analogy to 1D case:
\[ \frac{d}{dw} (y-hw)(y-hw) = \frac{d}{dw} (y-hw)^2 = 2 \cdot (y-hw)(-h) \]
\[ = -2h \cdot (y-hw) \]
Step 4, Approach 1:
Set the gradient $= 0$

Closed-form solution

$$\nabla \text{RSS}(w) = -2H^T(y - Hw) = 0$$

Solve for $w$:

$$-\frac{1}{2}H^Ty + \frac{1}{2}H^TH\hat{w} = 0$$

$$H^TH\hat{w} = H^Ty$$

$$\hat{w} = (H^TH)^{-1}H^Ty$$
Closed-form solution

\[ \hat{w} = (H^T H)^{-1} H^T y \]

Invertible if:

- In most cases: \( N > D \)

Complexity of inverse:

\[ O(D^3) \]

Step 4, Approach 2: Gradient descent
Gradient descent

\[
\text{while not converged}
\]

\[
\mathbf{w}^{(t+1)} \leftarrow \mathbf{w}^{(t)} - \eta \nabla \text{RSS} \left( \mathbf{w}^{(t)} \right) - 2\mathbf{H}^T (\mathbf{y} - \mathbf{H}\mathbf{w})
\]

Interpreting elementwise

Update to \( j \text{-th feature weight:} \)

\[
\mathbf{w}_j^{(t+1)} \leftarrow \mathbf{w}_j^{(t)} + 2\eta \sum_{i=1}^{N} h_j(x_i)(y_i - \hat{y}_i(\mathbf{w}^{(t)}))
\]

If underestimated, impact of \# baths \((\hat{w}_j^{(t)})\text{ is too small}), then \((y_i - \hat{y}_i(\mathbf{w}^{(t)}))\text{ on average weighted by } \# \text{ baths will be positive}\)

\[
\Rightarrow \mathbf{w}_j^{(t+1)} > \mathbf{w}_j^{(t)} \quad \text{(increase)}
\]
Summary of gradient descent for multiple regression

init \( w^{(1)} = 0 \) (or randomly, or smartly), \( t = 1 \)

while \( \| \nabla \text{RSS}(w^{(t)}) \| > \varepsilon \) 

for \( j = 0, \ldots, D \)

partial\([j]\) = \(-2\sum_{i=1}^{N} h_j(x_i)(y_i - \hat{y}_i(w^{(t)}))\)

\( w_j^{(t+1)} \leftarrow w_j^{(t)} - \eta \text{ partial}\([j]\)\)

\( t \leftarrow t + 1 \)

Why min RSS?
Assuming Gaussian noise

Model for $\varepsilon_i$:

\[ \varepsilon_i \sim \mathcal{N}(0, \sigma^2) \]

Gaussian

Implied distribution on $y_i$:

\[ y_i = f(x_i) + \varepsilon_i = h^T(x_i) \omega + \varepsilon_i \]

\[ y_i \sim \mathcal{N}(h^T(x_i) \omega, \sigma^2) \]

Maximum likelihood estimate of params

Maximize log-likelihood wrt $\omega$

\[
\begin{align*}
\ln p(D \mid \omega, \sigma) &= \ln \left( \frac{1}{\sigma \sqrt{2\pi}} \right)^N \prod_{i=1}^{N} e^{-\frac{(y_i - \sum_{j} w_j h_j(x_i))^2}{2\sigma^2}} \\
&= \arg \max_{\omega} N \ln \left( \frac{1}{\sigma \sqrt{2\pi}} \right)^N - \frac{1}{2\sigma^2} \sum_{i=1}^{N} (y_i - \sum_{j} w_j h_j(x_i))^2 - 2 \ln \sigma - \text{const}
\end{align*}
\]

\[
\begin{align*}
&= \arg \min_{\omega} \sum_{i=1}^{N} (y_i - \sum_{j} w_j h_j(x_i))^2 \quad \text{MLE for Gauss error model} \\
&= \min \text{ RSS}!! \quad \text{LS line}
\end{align*}
\]
Interpreting the coefficients -
Simple linear regression

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

price ($) vs. square feet (sq.ft.)

predicted change in $ for 1 sq. ft.
Interpreting the coefficients - Two linear features

\[ \hat{y} = \hat{w}_0 + \hat{w}_1 x[1] + \hat{w}_2 x[2] \]

For fixed \# sq.ft.!
Interpreting the coefficients - Multiple linear features

\[ \hat{y} = \hat{w}_0 + \hat{w}_1 x[1] + \ldots + \hat{w}_j x[j] + \ldots + \hat{w}_d x[d] \]

- Can't hold other features fixed!
Recap of concepts

What you can do now...

- Describe polynomial regression
- Write a regression model using multiple inputs or features thereof
- Cast both polynomial regression and regression with multiple inputs as regression with multiple features
- Calculate a goodness-of-fit metric (e.g., RSS)
- Estimate model parameters of a general multiple regression model to minimize RSS:
  - In closed form
  - Using an iterative gradient descent algorithm
- Interpret the coefficients of a non-featurized multiple regression fit
- Exploit the estimated model to form predictions