Least Squares Data Fitting

Stephen Boyd

EE103
Stanford University

October 31, 2017
Outline

Least squares model fitting

Validation

Feature engineering

Least squares model fitting
Setup

- We believe a scalar $y$ and an $n$-vector $x$ are related by model
  \[ y \approx f(x) \]
- $x$ is called the independent variable
- $y$ is called the outcome or response variable
- $f : \mathbb{R}^n \rightarrow \mathbb{R}$ gives the relation between $x$ and $y$
- Often $x$ is a feature vector, and $y$ is something we want to predict
- We don’t know $f$, which gives the ‘true’ relationship between $x$ and $y$
Data

- we are given some data

\[ x^{(1)}, \ldots, x^{(N)}, \quad y^{(1)}, \ldots, y^{(N)} \]

also called observations, examples, samples, or measurements

- \( x^{(i)}, y^{(i)} \) is \( i \)th data pair

- \( x^{(i)}_j \) is the \( j \)th component of \( i \)th data point \( x^{(i)} \)
choose model \( \hat{f} : \mathbb{R}^n \rightarrow \mathbb{R} \), a guess or approximation of \( f \)

linear in the parameters model form:

\[
\hat{f}(x) = \theta_1 f_1(x) + \cdots + \theta_p f_p(x)
\]

\( f_i : \mathbb{R}^n \rightarrow \mathbb{R} \) are basis functions that we choose

\( \theta_i \) are model parameters that we choose

\( \hat{y}^{(i)} = \hat{f}(x^{(i)}) \) is (the model’s) prediction of \( y^{(i)} \)

we’d like \( \hat{y}^{(i)} \approx y^{(i)} \), i.e., model is consistent with observed data
Least squares data fitting

- prediction error or residual is \( r_i = y^{(i)} - \hat{y}^{(i)} \)
- least squares data fitting: choose model parameters \( \theta_i \) to minimize RMS prediction error on data set

\[
\left( \frac{(r^{(1)})^2 + \cdots + (r^{(N)})^2}{N} \right)^{1/2}
\]

- this can be formulated (and solved) as a least squares problem
Least squares data fitting

- express $y^{(i)}$, $\hat{y}^{(i)}$, and $r^{(i)}$ as $N$-vectors
  - $y^d = (y^{(1)}, \ldots, y^{(N)})$ is vector of outcomes
  - $\hat{y}^d = (\hat{y}^{(1)}, \ldots, \hat{y}^{(N)})$ is vector of predictions
  - $r^d = (r^{(1)}, \ldots, r^{(N)})$ is vector of residuals

- $\text{rms}(r^d)$ is RMS prediction error

- define $N \times p$ matrix $A$, $A_{ij} = f_j(x^{(i)})$, so $\hat{y}^d = A\theta$

- least squares data fitting: choose $\theta$ to minimize

  \[ ||r^d||^2 = ||y^d - \hat{y}^d||^2 = ||y^d - A\theta||^2 = ||A\theta - y^d||^2 \]

- $\hat{\theta} = (A^TA)^{-1}A^Ty$ (if columns of $A$ are independent)

- $||A\hat{\theta} - y||^2/N$ is minimum mean-square (fitting) error
Fitting a constant model

- simplest possible model: \( p = 1, \ f_1(x) = 1 \), so model \( \hat{f}(x) = \theta_1 \) is a constant function
- \( A = 1 \), so
  \[
  \hat{\theta}_1 = (1^T 1)^{-1} 1^T y^d = (1/N) 1^T y^d = \text{avg}(y^d)
  \]
- the mean of \( y^{(1)}, \ldots, y^{(N)} \) is the least squares fit by a constant
- MMSE is \( \text{std}(y^d)^2 \); RMS error is \( \text{std}(y^d) \)
- more sophisticated models are judged against the constant model
Fitting univariate functions

- when $n = 1$, we seek to approximate a function $f : \mathbb{R} \rightarrow \mathbb{R}$
- we can plot the data $(x_i, y_i)$ and the model function $\hat{y} = \hat{f}(x)$
Straight-line fit

- \( p = 2 \), with \( f_1(x) = 1, f_2(x) = x \)
- model has form \( \hat{f}(x) = \theta_1 + \theta_2 x \)
- matrix \( A \) has form

\[
A = \begin{bmatrix}
1 & x^{(1)} \\
1 & x^{(2)} \\
\vdots & \vdots \\
1 & x^{(N)}
\end{bmatrix}
\]

- can work out \( \hat{\theta}_1 \) and \( \hat{\theta}_2 \) explicitly:

\[
\hat{f}(x) = \text{avg}(y^d) + \rho \frac{\text{std}(y^d)}{\text{std}(x^d)}(x - \text{avg}(x^d))
\]

where \( x^d = (x^{(1)}, \ldots, x^{(N)}) \)
Example

Least squares model fitting
Asset $\alpha$ and $\beta$

- $x$ is return of whole market, $y$ is return of a particular asset
- write straight-line model as

$$\hat{y} = (r^{rf} + \alpha) + \beta(x - \mu^{mkt})$$

- $\mu^{mkt}$ is the average market return
- $r^{rf}$ is the risk-free interest rate
- several other slightly different definitions are used

- called asset ‘$\alpha$’ and ‘$\beta$’, widely used
Time series trend

- $y^{(i)}$ is value of quantity at time $x^{(i)} = i$
- $\hat{y}^{(i)} = \hat{\theta}_1 + \hat{\theta}_2 i, \quad i = 1, \ldots, N$, is called trend line
- $y^d - \hat{y}^d$ is called de-trended time series
- $\hat{\theta}_2$ is trend coefficient
World petroleum consumption

Least squares model fitting
World petroleum consumption, de-trended

Least squares model fitting
Polynomial fit

- $f_i(x) = x^{i-1}, \quad i = 1, \ldots, p$
- model is a polynomial of degree less than $p$

$$\hat{f}(x) = \theta_1 + \theta_2 x + \cdots + \theta_p x^{p-1}$$

(here $x^i$ means scalar $x$ to $i$th power; $x^{(i)}$ is $i$th data point)

- $A$ is Vandermonde matrix

$$A = \begin{bmatrix}
1 & x^{(1)} & \cdots & (x^{(1)})^{p-1} \\
1 & x^{(2)} & \cdots & (x^{(2)})^{p-1} \\
\vdots & \vdots & \ddots & \vdots \\
1 & x^{(N)} & \cdots & (x^{(N)})^{p-1}
\end{bmatrix}$$

Least squares model fitting
Example

$N = 100$ data points

Least squares model fitting
Regression as general data fitting

- regression model is affine function \( \hat{y} = \hat{f}(x) = x^T \beta + v \)
- fits general fitting form with basis functions

\[
f_1(x) = 1, \quad f_i(x) = x_{i-1}, \quad i = 2, \ldots, n + 1
\]

so model is

\[
\hat{y} = \theta_1 + \theta_2 x_1 + \cdots + \theta_{n+1} x_n = x^T \theta_{2:n} + \theta_1
\]

- \( \beta = \theta_{2:n+1} \), \( v = \theta_1 \)
General data fitting as regression

- general fitting model: \( \hat{f}(x) = \theta_1 f_1(x) + \cdots + \theta_p f_p(x) \)
- common assumption: \( f_1(x) = 1 \)
- same as regression model: \( \hat{f}(\tilde{x}) = \tilde{x}^T \beta + v \), with
  - \( \tilde{x} = (f_2(x), \ldots, f_p(x)) \) are ‘transformed features’
  - \( v = \theta_1, \beta = \theta_{2:p} \)
Auto-regressive time series model

- time series $z_1, z_2, \ldots$
- auto-regressive (AR) prediction model:
  \[ \hat{z}_{t+1} = \theta_1 z_t + \cdots + \theta_M z_{t-M+1}, \quad t = M, M + 1, \ldots \]
- $M$ is memory of model
- $\hat{z}_{t+1}$ is prediction of next value, based on previous $M$ values
- we'll choose $\beta$ to minimize sum of squares of prediction errors,
  \[ (\hat{z}_{M+1} - z_{M+1})^2 + \cdots + (\hat{z}_T - z_T)^2 \]
- put in general form with
  \[ y^{(i)} = z_{M+i}, \quad x^{(i)} = (z_{M+i-1}, \ldots, z_i), \quad i = 1, \ldots, T - M \]
Example

- hourly temperature at LAX in May 2016, length 744
- average is 61.76°F, standard deviation 3.05°F
- predictor $\hat{z}_{t+1} = z_t$ gives RMS error 1.16°F
- predictor $\hat{z}_{t+1} = z_{t-23}$ gives RMS error 1.73°F
- AR model with $M = 8$ gives RMS error 0.98°F
solid line shows one-hour ahead predictions from AR model, first 5 days
Outline

Least squares model fitting

Validation

Feature engineering
Generalization

basic idea:

▶ goal of model is *not* to predict outcome for the given data
▶ instead it is to *predict the outcome on new, unseen data*

▶ a model that makes reasonable predictions on new, unseen data has *generalization ability, or generalizes*
▶ a model that makes poor predictions on new, unseen data is said to suffer from *over-fit*
Validation

a simple and effective method to guess if a model will generalize

- split original data into a training set and a test set
- typical splits: 80%/20%, 90%/10%
- build (‘train’) model on training data set
- then check the model’s predictions on the test data set
- (can also compare RMS prediction error on train and test data)
- if they are similar, we can guess the model will generalize
Validation

- can be used to choose among different candidate models, *e.g.*
  - polynomials of different degrees
  - regression models with different sets of regressors
- we’d use one with low, or lowest, test error
Example

- polynomials fit using *training set* of 100 points
- plots below show performance with *test set* of 100 points
Example

- suggests degree 4, 5, or 6 are reasonable choices
Cross validation

to carry out cross validation:

▶ divide data into 10 *folds*
▶ for $i = 1, \ldots, 10$, build (train) model using all folds except $i$
▶ test model on data in fold $i$

interpreting cross validation results:

▶ if test RMS errors are much larger than train RMS errors, model is over-fit
▶ if test and train RMS errors are similar and consistent, we can guess the model will have a similar RMS error on future data
Example

- house price, regression fit with $x = (\text{area/1000 ft.}^2, \text{bedrooms})$
- 774 sales, divided into 5 folds of 155 sales each
- fit 5 regression models, removing each fold

<table>
<thead>
<tr>
<th>Fold</th>
<th>Model parameters</th>
<th>RMS error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$v$</td>
<td>$\beta_1$</td>
</tr>
<tr>
<td>1</td>
<td>60.65</td>
<td>143.36</td>
</tr>
<tr>
<td>2</td>
<td>54.00</td>
<td>151.11</td>
</tr>
<tr>
<td>3</td>
<td>49.06</td>
<td>157.75</td>
</tr>
<tr>
<td>4</td>
<td>47.96</td>
<td>142.65</td>
</tr>
<tr>
<td>5</td>
<td>60.24</td>
<td>150.13</td>
</tr>
</tbody>
</table>
Outline

Least squares model fitting

Validation

Feature engineering
Feature engineering

- start with original or base feature $n$-vector $x$
- choose basis functions $f_1, \ldots, f_p$ to create ‘mapped’ feature $p$-vector
  \[ (f_1(x), \ldots, f_p(x)) \]
- now fit linear in parameters model with mapped features
  \[ \hat{y} = \theta_1 f_1(x) + \cdots + \theta_p f_p(x) \]
- check the model using validation
Transforming features

- **standardizing features**: replace $x_i$ with $(x_i - b_i)/a_i$
  - $b_i \approx$ mean value of the feature across the data
  - $a_i \approx$ standard deviation of the feature across the data
  
  new features are called *z*-scores

- **log transform**: if $x_i$ is nonnegative and spans a wide range, replace it with $\log(1 + x_i)$

- **hi and lo features**: create new features given by
  
  $$
  \max\{x_1 - b, 0\}, \quad \min\{x_1 - a, 0\}
  $$

  (called hi and lo versions of original feature $x_i$)
Example

- house price prediction
- start with base features
  - \( x_1 \) is area of house (in 1000ft.\(^2\))
  - \( x_2 \) is number of bedrooms
  - \( x_3 \) is 1 for condo, 0 for house
  - \( x_4 \) is zip code of address (62 values)
- we’ll use \( p = 8 \) basis functions:
  - \( f_1(x) = 1, f_2(x) = x_1, f_3(x) = \max\{x_1 - 1.5, 0\} \)
  - \( f_4(x) = x_2, f_5(x) = x_3 \)
  - \( f_6(x), f_7(x), f_8(x) \) are Boolean functions of \( x_4 \) which encode 4 groups of nearby zipcodes (i.e., neighborhood)
- five fold model validation
### Example

<table>
<thead>
<tr>
<th>Fold</th>
<th>$\theta_1$</th>
<th>$\theta_2$</th>
<th>$\theta_3$</th>
<th>$\theta_4$</th>
<th>$\theta_5$</th>
<th>$\theta_6$</th>
<th>$\theta_7$</th>
<th>$\theta_8$</th>
<th>RMS error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>122.35</td>
<td>166.87</td>
<td>−39.27</td>
<td>−16.31</td>
<td>−23.97</td>
<td>−100.42</td>
<td>−106.66</td>
<td>−25.98</td>
<td>67.29</td>
</tr>
<tr>
<td>2</td>
<td>100.95</td>
<td>186.65</td>
<td>−55.80</td>
<td>−18.66</td>
<td>−14.81</td>
<td>−99.10</td>
<td>−109.62</td>
<td>−17.94</td>
<td>67.83</td>
</tr>
<tr>
<td>3</td>
<td>133.61</td>
<td>167.15</td>
<td>−23.62</td>
<td>−18.66</td>
<td>−14.71</td>
<td>−109.32</td>
<td>−114.41</td>
<td>−28.46</td>
<td>69.70</td>
</tr>
<tr>
<td>4</td>
<td>108.43</td>
<td>171.21</td>
<td>−41.25</td>
<td>−15.42</td>
<td>−17.68</td>
<td>−94.17</td>
<td>−103.63</td>
<td>−29.83</td>
<td>65.58</td>
</tr>
<tr>
<td>5</td>
<td>114.45</td>
<td>185.69</td>
<td>−52.71</td>
<td>−20.87</td>
<td>−23.26</td>
<td>−102.84</td>
<td>−110.46</td>
<td>−23.43</td>
<td>70.69</td>
</tr>
</tbody>
</table>