Linear Regression (continued)

Readings:
Bishop, 3.1

Matt Gormley
Lecture 6
September 19, 2016
Probabilistic Interpretation of LMS

• Let us assume that the target variable and the inputs are related by the equation:

\[ y_i = \theta^T x_i + \varepsilon_i \]

where \( \varepsilon \) is an error term of unmodeled effects or noise.

• Now assume that \( \varepsilon \) follows a Gaussian \( \mathcal{N}(0, \sigma) \), then we have:

\[
p(y_i \mid x_i ; \theta) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left( - \frac{(y_i - \theta^T x_i)^2}{2\sigma^2} \right)
\]

• By independence assumption:

\[
L(\theta) = \prod_{i=1}^{n} p(y_i \mid x_i ; \theta) = \left( \frac{1}{\sqrt{2\pi\sigma}} \right)^n \exp\left( - \frac{\sum_{i=1}^{n} (y_i - \theta^T x_i)^2}{2\sigma^2} \right)
\]
Probabilistic Interpretation of LMS, cont.

• Hence the log-likelihood is:

\[
l(\theta) = n \log \frac{1}{\sqrt{2\pi\sigma}} - \frac{1}{\sigma^2} \frac{1}{2} \sum_{i=1}^{n} (y_i - \theta^T x_i)^2
\]

• Do you recognize the last term?

\[
J(\theta) = \frac{1}{2} \sum_{i=1}^{n} (x_i^T \theta - y_i)^2
\]

Yes it is:

• Thus under independence assumption, LMS is equivalent to MLE of \( \theta \)!
Non-Linear basis function

- So far we only used the observed values $x_1, x_2, ...$
- However, linear regression can be applied in the same way to functions of these values
  - Eg: to add a term $w \cdot x_1 \cdot x_2$ add a new variable $z = x_1 \cdot x_2$ so each example becomes: $x_1, x_2, ... z$
- As long as these functions can be directly computed from the observed values the parameters are still linear in the data and the problem remains a multi-variate linear regression problem

$$y = w_0 + w_1 x_1^2 + \ldots + w_k x_k^2 + \epsilon$$
Non-linear basis functions

- What type of functions can we use?
- A few common examples:

  - Polynomial: $\phi_j(x) = x^j$ for $j=0 \ldots n$
  
  - Gaussian: $\phi_j(x) = \frac{(x - \mu_j)}{2\sigma_j^2}$
  
  - Sigmoid: $\phi_j(x) = \frac{1}{1 + \exp(-s_j x)}$

  - Logs: $\phi_j(x) = \log(x + 1)$

Any function of the input values can be used. The solution for the parameters of the regression remains the same.
General linear regression problem

• Using our new notations for the basis function linear regression can be written as

\[ y = \sum_{j=0}^{n} w_j \phi_j(x) \]

• Where \( \phi_j(x) \) can be either \( x_j \) for multivariate regression or one of the non-linear basis functions we defined

• … and \( \phi_0(x) = 1 \) for the intercept term
An example: polynomial basis vectors on a small dataset

– From Bishop Ch 1
$0^{th}$ Order Polynomial

\[ M = 0 \]

$n=10$
1\textsuperscript{st} Order Polynomial

\[ M = 1 \]
$3^{rd}$ Order Polynomial

$M = 3$
9th Order Polynomial

$M = 9$
Over-fitting

Root-Mean-Square (RMS) Error: \( E_{\text{RMS}} = \sqrt{2E(w^*)/N} \)
## Polynomial Coefficients

<table>
<thead>
<tr>
<th></th>
<th>$M = 0$</th>
<th>$M = 1$</th>
<th>$M = 3$</th>
<th>$M = 9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_0^*$</td>
<td>0.19</td>
<td>0.82</td>
<td>0.31</td>
<td>0.35</td>
</tr>
<tr>
<td>$w_1^*$</td>
<td>-1.27</td>
<td>7.99</td>
<td></td>
<td>232.37</td>
</tr>
<tr>
<td>$w_2^*$</td>
<td></td>
<td>-25.43</td>
<td>-5321.83</td>
<td></td>
</tr>
<tr>
<td>$w_3^*$</td>
<td></td>
<td>17.37</td>
<td>48568.31</td>
<td></td>
</tr>
<tr>
<td>$w_4^*$</td>
<td></td>
<td></td>
<td>-231639.30</td>
<td></td>
</tr>
<tr>
<td>$w_5^*$</td>
<td></td>
<td></td>
<td>640042.26</td>
<td></td>
</tr>
<tr>
<td>$w_6^*$</td>
<td></td>
<td></td>
<td></td>
<td>-1061800.52</td>
</tr>
<tr>
<td>$w_7^*$</td>
<td></td>
<td></td>
<td></td>
<td>1042400.18</td>
</tr>
<tr>
<td>$w_8^*$</td>
<td></td>
<td></td>
<td></td>
<td>-557682.99</td>
</tr>
<tr>
<td>$w_9^*$</td>
<td></td>
<td></td>
<td></td>
<td>125201.43</td>
</tr>
</tbody>
</table>
Regularization

Penalize large coefficient values

\[ J_{x,y}(w) = \frac{1}{2} \sum_i \left( y^i - \sum_j w_j \phi_j(x^i) \right)^2 - \frac{\lambda}{2} \|w\|^2 \]
Regularization:

\[ \ln \lambda = +0.18 \]
Polynomial Coefficients

<table>
<thead>
<tr>
<th></th>
<th>none</th>
<th>exp(18)</th>
<th>huge</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_0^*$</td>
<td>0.35</td>
<td>0.35</td>
<td>0.13</td>
</tr>
<tr>
<td>$w_1^*$</td>
<td>232.37</td>
<td>4.74</td>
<td>-0.05</td>
</tr>
<tr>
<td>$w_2^*$</td>
<td>-5321.83</td>
<td>-0.77</td>
<td>-0.06</td>
</tr>
<tr>
<td>$w_3^*$</td>
<td>48568.31</td>
<td>-31.97</td>
<td>-0.05</td>
</tr>
<tr>
<td>$w_4^*$</td>
<td>-231639.30</td>
<td>-3.89</td>
<td>-0.03</td>
</tr>
<tr>
<td>$w_5^*$</td>
<td>640042.26</td>
<td>55.28</td>
<td>-0.02</td>
</tr>
<tr>
<td>$w_6^*$</td>
<td>-1061800.52</td>
<td>41.32</td>
<td>-0.01</td>
</tr>
<tr>
<td>$w_7^*$</td>
<td>1042400.18</td>
<td>-45.95</td>
<td>-0.00</td>
</tr>
<tr>
<td>$w_8^*$</td>
<td>-557682.99</td>
<td>-91.53</td>
<td>0.00</td>
</tr>
<tr>
<td>$w_9^*$</td>
<td>125201.43</td>
<td>72.68</td>
<td>0.01</td>
</tr>
</tbody>
</table>
Over Regularization:
Example: Stock Prices

• Suppose we wish to predict Google’s stock price at time t+1

• What features should we use? (putting all computational concerns aside)
  – Stock prices of all other stocks at times t, t-1, t-2, ..., t - k
  – Mentions of Google with positive / negative sentiment words in all newspapers and social media outlets

• Do we believe that all of these features are going to be useful?
Ridge Regression

• Adds an **L2 regularizer** to Linear Regression

\[ J_{RR}(\theta) = J(\theta) + \lambda \| \theta \|_2^2 \]

\[ = \frac{1}{2} \sum_{i=1}^{N} (\theta^T x(i) - y(i))^2 + \lambda \sum_{k=1}^{K} \theta_k^2 \]

• Bayesian interpretation: MAP estimation with a **Gaussian prior** on the parameters

\[ \theta^{MAP} = \arg\max_{\theta} \sum_{i=1}^{N} \log p_{\theta}(y(i)|x(i)) + \log p(\theta) \]

\[ = \arg\max_{\theta} J_{RR}(\theta) \]

where

\[ p(\theta) \sim \mathcal{N}(0, \frac{1}{\lambda}) \]
LASSO

- Adds an \textbf{L1 regularizer} to Linear Regression

\[
J_{\text{LASSO}}(\theta) = J(\theta) + \lambda \|\theta\|_1
\]

= \frac{1}{2} \sum_{i=1}^{N} (\theta^T x^{(i)} - y^{(i)})^2 + \lambda \sum_{k=1}^{K} |\theta_k|

- Bayesian interpretation: MAP estimation with a \textbf{Laplace prior} on the parameters

\[
\theta^{\text{MAP}} = \arg\max_{\theta} \sum_{i=1}^{N} \log p_\theta(y^{(i)}|x^{(i)}) + \log p(\theta)
\]

= \arg\max_{\theta} J_{\text{LASSO}}(\theta)

where \( p(\theta) \sim \text{Laplace}(0, f(\lambda)) \)
Ridge Regression vs Lasso

\[
\min_{\beta} (X\beta - Y)^T (X\beta - Y) + \lambda \text{pen}(\beta) = \min_{\beta} J(\beta) + \lambda \text{pen}(\beta)
\]

Ridge Regression:
\[
\text{pen}(\beta) = \|\beta\|_2^2
\]

Lasso:
\[
\text{pen}(\beta) = \|\beta\|_1
\]

Lasso (l1 penalty) results in sparse solutions – vector with more zero coordinates
Good for high-dimensional problems – don’t have to store all coordinates!
Data Set Size:

$N = 100$

9th Order Polynomial