Linear Regression

Linear regression assumes that the expected value of the output given an input, $E[y|x]$, is linear.

Simplest case: $\text{Out}(x) = wx$ for some unknown $w$.

Given the data, we can estimate $w$. 

<table>
<thead>
<tr>
<th>inputs</th>
<th>outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1 = 1$</td>
<td>$y_1 = 1$</td>
</tr>
<tr>
<td>$x_2 = 3$</td>
<td>$y_2 = 2.2$</td>
</tr>
<tr>
<td>$x_3 = 2$</td>
<td>$y_3 = 2$</td>
</tr>
<tr>
<td>$x_4 = 1.5$</td>
<td>$y_4 = 1.9$</td>
</tr>
<tr>
<td>$x_5 = 4$</td>
<td>$y_5 = 3.1$</td>
</tr>
</tbody>
</table>
1-parameter linear regression

Assume that the data is formed by

\[ y_i = wx_i + \text{noise}_i \]

where...

- the noise signals are independent
- the noise has a normal distribution with mean 0 and unknown variance \( \sigma^2 \)

\( P(y|w,x) \) has a normal distribution with

- mean \( wx \)
- variance \( \sigma^2 \)

Bayesian Linear Regression

\[ P(y|w,x) = \text{Normal} \ (\text{mean} \ wx, \ \text{var} \ \sigma^2) \]

We have a set of datapoints \((x_1,y_1), (x_2,y_2), \ldots, (x_R,y_R)\) which are EVIDENCE about \( w \).

We want to infer \( w \) from the data.

\[ P(w|x_1, x_2, x_3, \ldots, x_R, y_1, y_2, \ldots, y_R) \]

- You can use BAYES rule to work out a posterior distribution for \( w \) given the data.
- Or you could do Maximum Likelihood Estimation
Maximum likelihood estimation of \( w \)

Asks the question:
“For which value of \( w \) is this data most likely to have happened?”

\[ \Rightarrow \]

For what \( w \) is

\[ P(y_{1}, y_{2}, \ldots y_{R} | x_{1}, x_{2}, \ldots x_{R}, w) \]

maximized?

\[ \Rightarrow \]

For what \( w \) is

\[ \prod_{i=1}^{n} P(y_{i} | w, x_{i}) \]

maximized?

For what \( w \) is

\[ \prod_{i=1}^{R} P(y_{i} | w, x_{i}) \]

maximized?

For what \( w \) is

\[ \prod_{i=1}^{R} \exp\left(-\frac{1}{2} \left(\frac{y_{i} - wx_{i}}{\sigma}\right)^{2}\right) \]

maximized?

For what \( w \) is

\[ \sum_{i=1}^{R} - \frac{1}{2} \left(\frac{y_{i} - wx_{i}}{\sigma}\right)^{2} \]

maximized?

For what \( w \) is

\[ \sum_{i=1}^{R} \left(\frac{y_{i} - wx_{i}}{\sigma}\right)^{2} \]

minimized?
Linear Regression

The maximum likelihood \( w \) is the one that minimizes sum-of-squares of residuals.

\[
E(w) = \sum_i (y_i - wx_i)^2
\]

\[
= \sum_i y_i^2 - 2\sum_i x_i y_i w + \left( \sum x_i^2 \right) w^2
\]

We want to minimize a quadratic function of \( w \).

---

Linear Regression

Easy to show the sum of squares is minimized when

\[
w = \frac{\sum x_i y_i}{\sum x_i^2}
\]

The maximum likelihood model is \( \text{Out}(x) = wx \)

We can use it for prediction.
**Linear Regression**

Easy to show the sum of squares is minimized when

\[
W = \frac{\sum x_i y_i}{\sum x_i^2}
\]

The maximum likelihood model is

\[
\text{Out}(x) = wx
\]

We can use it for prediction.

**Note:** In Bayesian stats you’d have ended up with a prob dist of \( w \) — and predictions would have given a prob dist of expected output.

Often useful to know your confidence. Max likelihood can give some kinds of confidence too.

---

**Multivariate Regression**

What if the inputs are vectors?

Dataset has form

\[
\begin{align*}
x_1 & \quad y_1 \\
x_2 & \quad y_2 \\
x_3 & \quad y_3 \\
\vdots & \quad \vdots \\
x_R & \quad y_R
\end{align*}
\]

2-d input example
Multivariate Regression

Write matrix $X$ and $Y$ thus:

\[
X = \begin{bmatrix}
\ldots x_1 \ldots \\
\ldots x_2 \ldots \\
\vdots \\
\ldots x_R \ldots \\
\end{bmatrix}
= \begin{bmatrix}
x_{11} & x_{12} & \cdots & x_{1m} \\
x_{21} & x_{22} & \cdots & x_{2m} \\
\vdots & \vdots & \ddots & \vdots \\
x_{R1} & x_{R2} & \cdots & x_{Rm} \\
\end{bmatrix}
Y = \begin{bmatrix}
y_1 \\
y_2 \\
\vdots \\
y_R \\
\end{bmatrix}
\]

(There are $R$ datapoints. Each input has $m$ components)

The linear regression model assumes a vector $w$ such that

\[
\text{Out}(x) = w^T x = w_1 x[1] + w_2 x[2] + \ldots w_m x[m]
\]

The max. likelihood $w$ is

\[
w = (X^T X)^{-1} (X^T Y)
\]

Multivariate Regression (con’t)

The max. likelihood $w$ is $w = (X^T X)^{-1} (X^T Y)$

$X^T X$ is an $m \times m$ matrix: $i,j$’th elt is $\sum_{k=1}^{R} x_{ki} x_{kj}$

$X^T Y$ is an $m$-element vector: $i$’th elt is $\sum_{k=1}^{R} x_{ki} y_k$
What about a constant term?

We may expect linear data that does not go through the origin.

Statisticians and Neural Net Folks all agree on a simple obvious hack.

Can you guess??

The constant term

• The trick is to create a fake input “$X_0$” that always takes the value 1

Before:

\[ Y = w_1X_1 + w_2X_2 \]

...has to be a poor model

After:

\[ Y = w_0X_0 + w_1X_1 + w_2X_2 \]

\[ = w_0 + w_1X_1 + w_2X_2 \]

...has a fine constant term

In this example, You should be able to see the MLE $w_0$, $w_1$, and $w_2$ by inspection
What about higher-order terms?

Maybe we suspect a higher-order polynomial function like
\[ y = w_0 + w_1 x + w_2 x^2 + w_3 x^3 \]
would fit the data better.

In that case, we can simply perform multivariate linear regression using additional dimensions for all higher-order terms.

---

Higher-order terms

<table>
<thead>
<tr>
<th>Linear Fit</th>
<th>Quadratic Fit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 X Y</td>
<td>1 X X^2 Y</td>
</tr>
<tr>
<td>1 1 2</td>
<td>1 1 1 2</td>
</tr>
<tr>
<td>1 2 5</td>
<td>1 2 4 5</td>
</tr>
<tr>
<td>1 3 10</td>
<td>1 3 9 10</td>
</tr>
<tr>
<td>1 5 26</td>
<td>1 5 25 26</td>
</tr>
</tbody>
</table>
Maximum Likelihood Nonlinear Regression

Assume correct function is $y = f(x, w)$, where $f$ is any function of the input $x$ parameterized by $w$, and observations are corrupted by additive Gaussian noise (with some fixed variance $\sigma^2$).

For example, $f$ could be the function computed by a multilayer neural network whose weights are $w$.

As before, we would like to determine for what $w$

$$P(y_1, y_2, ..., y_R | x_1, x_2, x_3, ..., x_R, w)$$

is maximized.

And just as before, this translates into:
For what \( \mathbf{w} \) is
\[
\prod_{i=1}^{R} P(y_i | \mathbf{w}, \mathbf{x}_i) \text{ maximized?}
\]

For what \( \mathbf{w} \) is
\[
\prod_{i=1}^{R} \exp\left(-\frac{1}{2} \left( \frac{||y_i - f(\mathbf{x}_i, \mathbf{w})||}{\sigma} \right)^2 \right) \text{ maximized?}
\]

For what \( \mathbf{w} \) is
\[
\sum_{i=1}^{R} -\frac{1}{2} \left( \frac{||y_i - f(\mathbf{x}_i, \mathbf{w})||}{\sigma} \right)^2 \text{ maximized?}
\]

For what \( \mathbf{w} \) is
\[
\sum_{i=1}^{R} \left( ||y_i - f(\mathbf{x}_i, \mathbf{w})|| \right)^2 \text{ minimized?}
\]

- So, for example, with the usual squared-error measure, backpropagation can be viewed as a technique for searching for a maximum-likelihood fit of a neural network to a given set of training data.
- This applies when neural networks are used for regression, assuming additive Gaussian noise.
- What about for classification?
Maximum Likelihood Probability Estimation

- Consider a 2-class classification problem, and assume that the probability that an instance $x$ is classified as positive has the functional form $y = f(x, w)$.
- Then it can be shown that the correct criterion to optimize to generate ML estimates of the probability of belonging to the + class is *not* squared error.

Maximum Cross-Entropy

- Instead the following *cross-entropy* measure should be maximized:
  $$\sum_{i=1}^{R} \left( y_i \log f(x_i, w) + (1 - y_i) \log(1 - f(x_i, w)) \right)$$
- In a multilayer neural network, the gradient computation for this measure still follows the backpropagation process.