Supervised Learning Setting

- Data consists of input-output pairs
- Inputs (also covariates, independent variables, predictors, features)
- Output (also variates, dependent variable, targets, labels)
Outline

**Supervised Learning Setting**
- Data consists of *input-out put* pairs
- Inputs (also covariates, independent variables, predictors, features)
- Output (also variates, dependent variable, targets, labels)

**Goals**
- Understand the supervised learning setting
- Understand linear regression (aka *least squares*)
- Derivation of the least squares estimate
Why study linear regression?

- “Least squares” is at least 200 years old (Legendre, Gauss)
- Francis Galton: Regression to mediocrity (1886)
Why study linear regression?

- “Least squares” is at least 200 years old (Legendre, Gauss)
- Francis Galton: Regression to mediocrity (1886)
- Often real processes can be **approximated** by linear models
- More complicated models require understanding linear regression
Why study linear regression?

- “Least squares” is at least 200 years old (Legendre, Gauss)
- Francis Galton: Regression to mediocrity (1886)

- Often real processes can be **approximated** by linear models
- More complicated models require understanding linear regression

- Closed form analytic solutions can be obtained
- Many **key notions** of machine learning can be introduced
A toy example

Want to predict commute time into city centre

What variables would be useful?
- Distance to city centre
- Day of the week

Data

<table>
<thead>
<tr>
<th>dist (km)</th>
<th>day</th>
<th>commute time (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.7</td>
<td>fri</td>
<td>25</td>
</tr>
<tr>
<td>4.1</td>
<td>mon</td>
<td>33</td>
</tr>
<tr>
<td>1.0</td>
<td>sun</td>
<td>15</td>
</tr>
<tr>
<td>5.2</td>
<td>tue</td>
<td>45</td>
</tr>
<tr>
<td>2.8</td>
<td>sat</td>
<td>22</td>
</tr>
</tbody>
</table>
Suppose the input is a vector \( x \in \mathbb{R}^n \) and the output is \( y \in \mathbb{R} \).

We have data \( \langle x_i, y_i \rangle_{i=1}^m \)

**Notation:** dimension \( n \), size of dataset \( m \), column vectors

**Linear model:**

\[
y = w_0 + x_1w_1 + \cdots + x_nw_n + \text{noise}
\]
Linear model:

\[ y = w_0 + x_1 w_1 + \cdots + x_n w_n + \text{noise} \]
Linear Models

Linear model:

\[ y = w_0 + x_1 w_1 + \cdots + x_n w_n + \text{noise} \]

Input encoding: mon-sun has to be converted to a number

- Monday: 0, Tuesday: 1, ..., Sunday: 6
- 0 if weekend, 1 if weekday
Linear Models

Linear model:

\[ y = w_0 + x_1 w_1 + \cdots + x_n w_n + \text{noise} \]

Input encoding: mon-sun has to be converted to a number
- monday: 0, tuesday: 1, \ldots , sunday: 6
- 0 if weekend, 1 if weekday

Assume \( x_1 = 1 \) for all data. So model can be succinctly represented as

\[ y = x \cdot w + \text{noise} = \hat{y} + \text{noise} \]
Learning Linear Models

Data: \( \langle (x_i, y_i) \rangle_{i=1}^m \)

Model parameter \( w \)
Learning Linear Models

Data: $\langle (x_i, y_i) \rangle_{i=1}^{m}$

Model parameter $w$

Training phase: (learning/estimation of $w$)
Learning Linear Models

Data: $\langle (x_i, y_i) \rangle_{i=1}^m$

Model parameter $w$

Training phase: (learning/estimation of $w$)

Testing phase: (predict $\hat{y}_{m+1} = x_{m+1} \cdot w$)
\[ \hat{y}(x) = w_0 + x w_1 \]

\[ L(w) = L(w_0, w_1) = \sum_{i=1}^{m} (\hat{y}_i - y_i)^2 = \sum_{i=1}^{m} (w_0 + x_i w_1 - y_i)^2 \]
\[ \hat{y}(x) = w_0 + xw_1 \]

\[ L(w) = L(w_0, w_1) = \sum_{i=1}^{m} (\hat{y}_i - y_i)^2 = \sum_{i=1}^{m} (w_0 + x_iw_1 - y_i)^2 \]
Linear Regression: General Case

Recall that the linear model is

\[ \hat{y}_i = \sum_{j=1}^{n} x_{ij} w_j \]

where we assume that \( x_{i1} = 1 \) for all \( x_i \). So \( w_1 \) is the bias or offset term.
Linear Regression: General Case

Recall that the linear model is

\[ \hat{y}_i = \sum_{j=1}^{n} x_{ij} w_j \]

where we assume that \( x_{i1} = 1 \) for all \( x_i \). So \( w_1 \) is the bias or offset term.

Expressing everything in matrix notation

\[ \hat{y} = Xw \]

Here we have \( \hat{y} \in \mathbb{R}^{m \times 1} \), \( X \in \mathbb{R}^{m \times n} \) and \( w \in \mathbb{R}^{n \times 1} \).
Back to toy example

<table>
<thead>
<tr>
<th>dist (km)</th>
<th>weekday?</th>
<th>commute time (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.7</td>
<td>1 (fri)</td>
<td>25</td>
</tr>
<tr>
<td>4.1</td>
<td>1 (mon)</td>
<td>33</td>
</tr>
<tr>
<td>1.0</td>
<td>0 (sun)</td>
<td>15</td>
</tr>
<tr>
<td>5.2</td>
<td>1 (tue)</td>
<td>45</td>
</tr>
<tr>
<td>2.8</td>
<td>0 (sat)</td>
<td>22</td>
</tr>
</tbody>
</table>

We have $m = 5$, $n = 3$ and so we get

$$
y = \begin{bmatrix} 25 \\ 33 \\ 15 \\ 45 \\ 22 \end{bmatrix}, \quad X = \begin{bmatrix} 1 & 2.7 & 1 \\ 1 & 4.1 & 1 \\ 1 & 1.0 & 0 \\ 1 & 5.2 & 1 \\ 1 & 2.8 & 0 \end{bmatrix}, \quad w = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}
$$
Back to toy example

<table>
<thead>
<tr>
<th>dist (km)</th>
<th>weekday?</th>
<th>commute time (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.7</td>
<td>1 (fri)</td>
<td>25</td>
</tr>
<tr>
<td>4.1</td>
<td>1 (mon)</td>
<td>33</td>
</tr>
<tr>
<td>1.0</td>
<td>0 (sun)</td>
<td>15</td>
</tr>
<tr>
<td>5.2</td>
<td>1 (tue)</td>
<td>45</td>
</tr>
<tr>
<td>2.8</td>
<td>0 (sat)</td>
<td>22</td>
</tr>
</tbody>
</table>

We have \( m = 5, n = 3 \) and so we get

\[
y = \begin{bmatrix} 25 \\ 33 \\ 15 \\ 45 \\ 22 \end{bmatrix}, \quad X = \begin{bmatrix} 1 & 2.7 & 1 \\ 1 & 4.1 & 1 \\ 1 & 1.0 & 0 \\ 1 & 5.2 & 1 \\ 1 & 2.8 & 0 \end{bmatrix}, \quad w = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}
\]

Suppose we get \( w = [6, 6.5, 2]^T \). Then our predictions would be

\[
\hat{y} = \begin{bmatrix} 25.55 \\ 34.65 \\ 12.5 \\ 41.8 \\ 24.2 \end{bmatrix}
\]
Linear Prediction

Suppose someone lives 4.8km from city centre, how long would they take to get in on a Wednesday?

We can write $x_{\text{new}} = [1, 4.8, 1]^T$ and then compute

$$\hat{y}_{\text{new}} = \begin{bmatrix} 1 & 4.8 & 1 \end{bmatrix} \begin{bmatrix} 6 \\ 6.5 \\ 2 \end{bmatrix} = 39.2 \text{ minutes}$$
Suppose someone lives 4.8km from city centre, how long would they take to get in on a Wednesday?

We can write $x_{\text{new}} = [1, 4.8, 1]^T$ and then compute

$$
\hat{y}_{\text{new}} = \begin{bmatrix} 1 & 4.8 & 1 \end{bmatrix} \begin{bmatrix} 6 \\ 6.5 \\ 2 \end{bmatrix} = 39.2 \text{ minutes}
$$

Interpreting regression coefficients
Minimizing the Squared Error

\[ L(w) = \sum_{i=1}^{m} (x_i^T w - y_i)^2 = (Xw - y)^T (Xw - y) \]
Minimizing the Squared Error

\[ L(w) = \sum_{i=1}^{m} (x_i^T w - y_i)^2 = (Xw - y)^T (Xw - y) \]
Finding Optimal Solutions using Calculus

\[ L(w) = \sum_{i=1}^{m} (x_i^T w - y_i)^2 = (Xw - y)^T (Xw - y) \]
Differentiating Matrix Expressions

\[ L(w) = \sum_{i=1}^{m} (x_i^T w - y_i)^2 = (Xw - y)^T (Xw - y) \]

Rules
(i) \( \nabla_w c^T w = c \)
(ii) \( \nabla_w w^T A w = A w + A^T w (= 2A w \text{ for symmetric } A) \)
Solution to Linear Regression

\[ L(w) = \sum_{i=1}^{m} (x_i^T w - y_i)^2 = (Xw - y)^T (Xw - y) \]

\[ w = (X^T X)^{-1} X^T y \]
Solution to Linear Regression

\[ L(w) = \sum_{i=1}^{m} (x_i^T w - y_i)^2 = (Xw - y)^T (Xw - y) \]

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

What if we had one-hot encoding for days?

- 7 binary features, exactly one of them is 1
- \( x_1 = 1, x_2 = \text{dist}, x_3 = \text{mon?}, x_4 = \text{tue?}, \ldots, x_9 = \text{sun?} \)
Linear Regression as Solving Noisy Linear Systems

Without noise: $Xw = y$

Linear regression finds $\hat{y}$ that makes system $Xw = \hat{y}$ feasible

And $\hat{y}$ is closest to $y$ in Euclidean distance
Other Loss Functions

\[ L(w) = \sum_{i=1}^{m} |x_i^T w - y_i| \]
Loss Functions

- Consider $\ell : \mathbb{R}^+ \to \mathbb{R}^+$ satisfying
  1. $\ell(0) = 0$
  2. $\ell$ is non-decreasing

- $L(w) = \sum_{i=1}^{m} \ell(|x_i^T \cdot w - y_i|)$
Loss Functions

- Consider $\ell : \mathbb{R}^+ \to \mathbb{R}^+$ satisfying
  1. $\ell(0) = 0$
  2. $\ell$ is non-decreasing

- $L(w) = \sum_{i=1}^{m} \ell(|x_i^T \cdot w - y_i|)$

\[
\ell(z) = |z|^2 \\
\ell(z) = |z|^4 \\
\ell(z) = |z|^{10} \\
\ell(z) = |z| \\
\ell(z) = \sqrt{|z|} \\
\ell(z) = |z|^{0.01}
\]
Optimization with different loss functions

\[ L(w) = \sum |\hat{y}_i - y_i|^2 \]

\[ L(w) = \sum |\hat{y}_i - y_i| \]

\[ L(w) = \sum \sqrt{|\hat{y}_i - y_i|} \]

\[ L(w) = \sum |\hat{y}_i - y_i|^{0.1} \]
Probabilistic Modelling

- Linear Model: \( y = x^T w^* + \text{noise} \) (for some \( w^* \))

- \( \mathbb{E}[y \mid x, w^*] = x^T w^* \)

- Data \( \langle (x_i, y_i) \rangle_{i=1}^m \)

- Unbiased estimator \( \hat{w} = (X^T X)^{-1} X^T y \)
Next Time

- Maximum likelihood estimation
- Make sure you are familiar with the Gaussian distribution
- Non-linearity using basis expansion
- What to do when you have more features than data?