Overview

Recall module is roughly split into four parts:

1. **Random events**: counting, events, axioms of probability, Bayes, independence

2. **Random variables**: discrete RVs, mean and variance, correlation, conditional expectation

   **Mid-term**

3. **Inequalities and laws of large numbers**: Markov, Chebyshev, Chernoff bounds, sample mean, weak law of large numbers, central limit theorem, confidence intervals, bootstrapping

4. **Statistical models**: continuous random variables, logistic regression, least squares
Overview

- Randomized Response
- Classification
- Logistic Regression
- Parameter Estimation
- Maximum Likelihood Estimate
Example

Suppose we carry out a poll and ask people in Dublin whether they have paid the water charge. The twist on our previous setup is that people may respond incorrectly.

- Pick one person at random. Ask them to toss a coin. If it comes up heads they answer truthfully, otherwise they answer “no”.
- Called randomised response\(^1\), used when asking sensitive questions to provide respondents with plausible deniability.

\(^1\)https://en.wikipedia.org/wiki/Randomized_response
**Example**

Our **statistical model** is:

- Let random variable $\Theta = 1$ ("yes") if they have paid and 0 ("no") otherwise.
- Suppose we ask them $n$ times and record the responses.
- Let $V_i = 1$ when coin is heads at the $i$th question, and 0 otherwise, $P(V_i = 1) = 1/2$. Respond with true answer $\theta$ if $V_i = 1$ else “no”.
- We observe $Y = \sum_{i=1}^{n} \theta V_i$. Suppose from our knowledge of the Dublin population, we also know $P(\Theta = 1) = 0.8$. We want to estimate $\theta$, the value of RV $\Theta$ for this person.
Example

- Bayes Rule:

\[ P(\Theta = 0 | Y = y) = \frac{P(Y = y | \Theta = 0)P(\Theta = 0)}{P(Y = y)} \]

\[ P(\Theta = 1 | Y = y) = \frac{P(Y = y | \Theta = 1)P(\Theta = 1)}{P(Y = y)} \]

- \( P(Y = 0 | \Theta = 0) = 1 \) (always reply “no” if truth is “no”).
- \( P(Y = y | \Theta = 1) = P(\sum_{i=1}^{n} V_i = y) = \binom{n}{y} 0.5^y (1 - 0.5)^{n-y} = \binom{n}{y} 0.5^n \)
- \( P(\Theta = 1) = 0.8 \) and so \( P(\Theta = 0) = 0.2 \)

\[ P(\Theta = 0 | Y = y) = \begin{cases} \frac{0.2}{P(Y=y)} & y = 0 \\ 0 & y \neq 0 \end{cases} \]

\[ P(\Theta = 1 | Y = y) = \frac{\binom{n}{y} 0.5^n \times 0.8}{P(Y = y)} \]
Example

\[ P(\Theta = 0|Y = y) = \begin{cases} 
\frac{0.2}{P(Y=y)} & y = 0 \\
0 & y \neq 0
\end{cases} \]

\[ P(\Theta = 1|Y = y) = \frac{\binom{n}{y} 0.5^n \times 0.8}{P(Y = y)} \]

- Now what? If \( P(\Theta = 1|Y = y) > P(\Theta = 0|Y = y) \) then we’ll take \( \Theta = 1 \) as our estimate, else \( \Theta = 0 \) as our estimate.
- Notice that we can compare \( P(\Theta = 1|Y = y) \) and \( P(\Theta = 0|Y = y) \) even when we don’t know \( P(Y = y) \). It’s just a normalising constant, not important.
- When \( y \neq 0 \) then \( P(\Theta = 0|Y = y) = 0 \) and we always estimate \( \Theta = 1 \).
- When \( y = 0 \) and \( \binom{n}{y} 0.5^n \times 0.8 < 0.2 \) we estimate \( \Theta = 0 \).
Example

- This approach is called **randomized response**. Common way to ask sensitive questions in surveys.
- What we’re doing here is an “attack” on the process, trying to figure out what the true answer of an individual was.
- Can see from analysis its important that a question is only asked once if want protection.
- What if we know that a group usually answer the same way? Asking several members of that group is almost the same as asking one person multiple times.
- Could also use this to make an accurate **prediction** of answer of other people in this group (without actually asking them any question).
Classification

More generally,

• Suppose we have a collection of objects and each has an unknown label associated with it e.g. likes marmite or doesn’t

• For a subset of the objects we observe the label plus some other properties e.g. location, nationality (features, explanatory variables, independent variables). This is our training data.

• We are willing to make a number of assumptions, our model.

• We now want to build a classifier that predicts the label of a new object drawn from the collection.

Examples:

• Based on the text within an email, predict whether it is spam or not

• Given the contents of my shopping basket, predict whether I’m vegetarian or not

• Given where I live in Dublin, predict which political party I’ll vote for.
Classification: Logistic Regression

- Label $Y$ only takes values 0 or 1. Real-valued vector $\vec{X}$ of $m$ observed features $X^{(1)}, X^{(2)}, \ldots, X^{(m)}$

- In **Logistic regression** our statistical model is that:

$$P(Y = 1|\Theta = \vec{\theta}, \vec{X} = \vec{x}) = \frac{1}{1 + \exp(-z)} \text{ with } z = \sum_{i=1}^{m} \theta^{(i)} x^{(i)}$$

$$P(Y = 0|\Theta = \vec{\theta}, \vec{X} = \vec{x}) = 1 - P(Y = 1|\Theta = \vec{\theta}, \vec{X} = \vec{x}) = \frac{\exp(-z)}{1 + \exp(-z)}$$

- Model has $m$ parameters $\theta^{(1)}, \theta^{(2)} \ldots, \theta^{(m)}$. We gather these together into a vector $\vec{\theta}$

- Will streamline notation for $P(Y = 1|\Theta = \vec{\theta}, \vec{X} = \vec{x})$ to $P(Y = 1|\vec{\theta}, \vec{x})$. 
Classification: Logistic Regression

- $P(Y = 1|\theta, \vec{x})$ changes smoothly with $\vec{x}$
- Want to try to learn to predict when $Y = 1$ and $Y = 0$ given a value of $\vec{x}$. 

![Graph of Logistic Regression](image-url)
Linear Separability

- Can also plot $P(Y = 1|\vec{\theta}, \vec{x})$ against $\vec{x}$ rather than $z$.
- Example with vector $\vec{x} = [1, x^{(0)}, x^{(1)}]$
Linear Separability

- In general, $\sum_{i=1}^{m} \theta(i)x^{(i)} = 0$ is called a **linear** equation. It defines a plane in $m$-dimensions.
- Logistic regression thresholds $z$ and predicts $Y = 1$ when $z > 0$ and $Y = 0$ when $z < 0$.
- So we can think of logistic regression as trying to fit a plane that separates the $Y = 1$ data from the $Y = 0$ data.
- We call such data “linearly separable”. 
Linear Separability

- Not all data is linearly separable e.g.

- Close links between logistic regression and neural networks.
Parameter Estimation

- **Training data is RV $D$.** Consists of $n$ observations $d = \{(\vec{x}_1, y_1), \cdots, (\vec{x}_n, y_n)\}$
- **Recall Bayes Rule**
  \[
P(\Theta = \hat{\theta} | D = d) = \frac{P(D = d | \Theta = \hat{\theta}) P(\Theta = \hat{\theta})}{P(D = d)}
  \]
  *posterior*  *likelihood*  *prior*
- **Likelihood.** Probability of seeing the data $d$ given model with parameter $\Theta = \hat{\theta}$
- **Prior.** Before seeing any data what is our belief about the model i.e. what is probability of parameter values $\Theta$.
- **Posterior.** After seeing the data, what is our belief about probability of parameter values $\Theta$ now that we have seen the data.
- **Maximum A Posteriori (MAP) estimate of $\hat{\theta}$** is value that maximises $P(\Theta = \hat{\theta} | D = d)$
Parameter Estimation

- Likelihood is:

\[
P(D = d | \Theta = \tilde{\theta}) = \prod_{k=1}^{n} P(Y = y_k | \tilde{\theta}, \bar{x}_k) = \prod_{k=1}^{n} \left( \frac{1}{1 + \exp(-z_k)} \right)^{y_k} \left( \frac{\exp(-z_k)}{1 + \exp(-z_k)} \right)^{1-y_k}
\]

with \( z_k = \sum_{i=1}^{m} \theta^{(i)} x^{(i)}_k \).

- Prior \( P(\Theta = \tilde{\theta}) \). If \( \tilde{\theta} \) discrete valued then we can use any prior we like. But usually allow \( \tilde{\theta} \) to be continuous valued in Logistic regression.

- For now let’s consider **Maximum Likelihood** estimate of \( \tilde{\theta} \), the value which maximises \( P(D | \tilde{\theta}) \).
Maximum Likelihood Estimate

Example: have two pairs of observations \((x_1 = 1, y_1 = 1)\) and \((x_2 = -1, y_2 = 0)\), one feature \(x_k\) and \(z_k = \theta x_k\),

\[
P(D = d|\Theta = \vec{\theta}) = p_1^{y_1}(1 - p_1)^{1-y_1} \times p_2^{y_2}(1 - p_2)^{1-y_2}
\]

with

\[
p_1 = \frac{1}{1 + \exp(-z_1)} = \frac{1}{1 + \exp(-\theta x_1)}, p_2 = \frac{1}{1 + \exp(-z_2)} = \frac{1}{1 + \exp(-\theta x_2)}
\]

That is,

\[
P(D = d|\Theta = \vec{\theta}) = p_1 \times (1 - p_2)
\]

with

\[
p_1 = \frac{1}{1 + \exp(-\theta)}, p_2 = \frac{1}{1 + \exp(+\theta)}
\]
Maximum Likelihood Estimate

- Maximising $\log P(D = d | \Theta = \hat{\theta})$ is the same as maximising $P(D = d | \Theta = \hat{\theta})$ (why ?)

- $\log P(D = d | \Theta = \hat{\theta})$ is referred to as the log-likelihood.

- Compute derivative of log-likelihood with respect to $\theta^{(i)}$:

$$\sum_{k=1}^{n} \left( y_k - \frac{1}{1 + \exp(-z_k)} \right) x_k^{(i)}$$

(Remember $\frac{d \log(x)}{dx} = \frac{1}{x}, \frac{d \exp(x)}{dx} = \exp(x), \frac{d}{dx} \frac{1}{x} = -\frac{1}{x^2}$ and chain rule $\frac{df(z(x))}{dx} = \frac{df}{dz} \frac{dz}{dx}$)
Maximum Likelihood Estimate

Example: have two pairs of observations \((x_1 = 1, y_1 = 1)\) and \((x_2 = 0, y_2 = 0)\), one feature \(x_k\) and \(z_k = \theta x_k\),

\[
P(D = d|\Theta = \vec{\theta}) = p_1 \times (1 - p_2), \quad p_1 = \frac{1}{1 + \exp(-\theta)}, \quad p_2 = \frac{1}{1 + \exp(+\theta)}
\]

Derivative of \(\log P(D = d|\Theta = \vec{\theta})\) is

\[
(y_1 - p_1)x_1 + (y_2 - p_2)x_2 = (1 - p_1) + (0 - p_2) \times 0 = 1 - p_1
\]
Maximum Likelihood Estimate

- Compute derivative of log-likelihood with respect to $\theta^{(i)}$:

$$
\sum_{k=1}^{n} \left( y_k - \frac{1}{1 + \exp(-z_k)} \right) x_k^{(i)}
$$

- Would like to set derivative equal to 0 to find ML estimate of $\vec{\theta}$, but hard to do this.

- Instead solve numerically using iteration:

$$
\theta_j^{(i)} = \theta_j^{(i)} + \alpha \sum_{k=1}^{n} \left( y_k - \frac{1}{1 + \exp(-z_{k,j})} \right) x_k^{(i)} \text{ with } z_{k,j} = \sum_{i=1}^{m} \theta_j^{(i)} x_k^{(i)}
$$
Maximum Likelihood Estimate

- log-likelihood is concave, has a single maximum
- iteration climbs uphill until it reaches the maximum
Maximum Likelihood Estimate

```python
alpha = 0.01;  
[N,m] = size(X);  
theta = zeros(1,m);  
for l = 1:10000,  
    grad = zeros(1,m);  
    for k = 1:N,  
        Z = 0;  
        for i = 1:m  
            Z = Z + theta(i)*X(k,i);  
        end  
        for i = 1:m  
            grad(i) = grad(i) + ...  
                (Y(k) - 1/(1+exp(-Z)))*X(k,i);  
        end  
    end  
    for i = 1:m  
        theta(i) = theta(i) + alpha*grad(i);  
    end  
end
```
Example

- Two inputs \( \vec{x} = [1, x] \) so \( x = \theta(0) + \theta(1)x \). First input is fixed and means \( \theta(0) \) captures offset, \( \theta(1) \) captures slope.
- As we increase “noise” on \( Y \) then parameter \( \theta(1) \) changes to broaden curve reflecting greater uncertainty.