Machine Learning: Intro

Rodrigo Azuero

University of Pennsylvania

2018

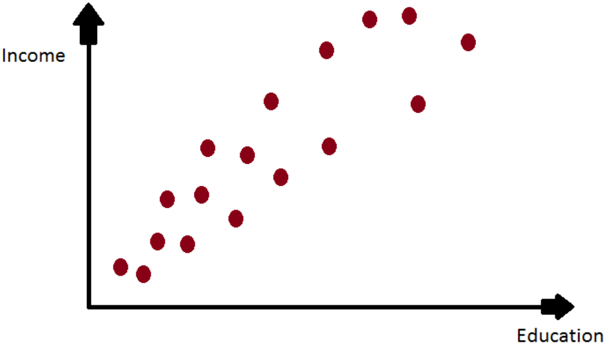
Class 2

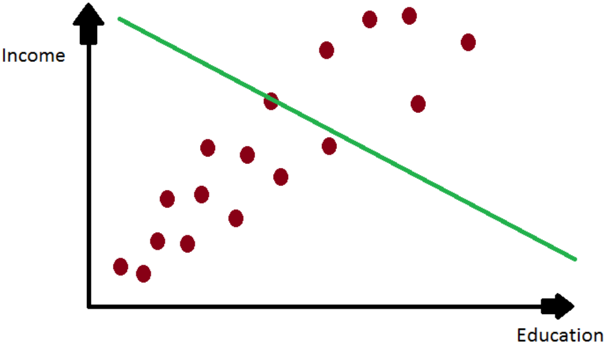
In this session we will cover the following topics

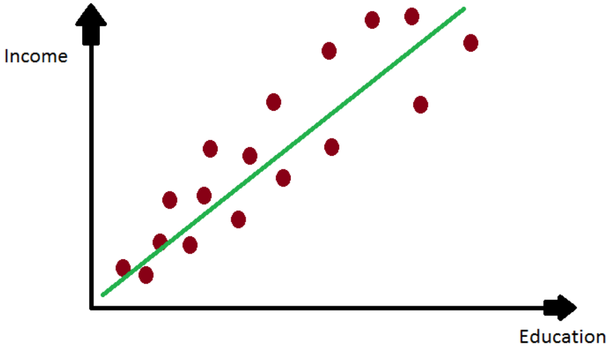
1. Linear regression I
2. Logistic regression I
3. Linear regression II
4. Logistic regression II
5. Bayesian Classification
6. Gradient Descent

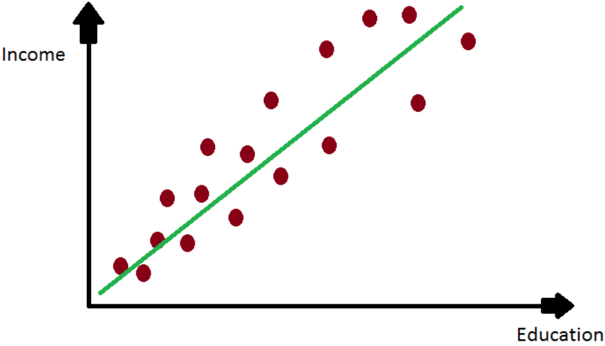
Outline

- Linear regression: our goal is to find a linear relationship between two variables
- Let us consider an example: Income and education.
- We want to predict people's income based on their education

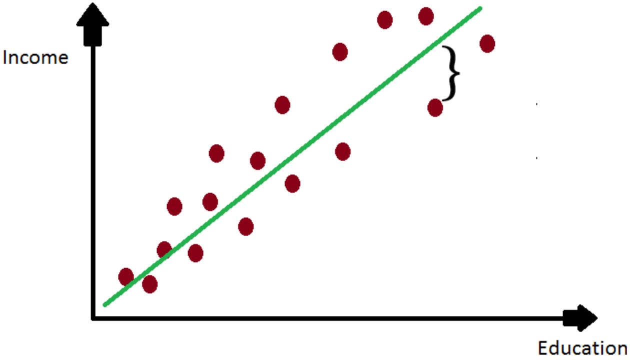

Income
Education


Income
Education


Income
Education



$$\hat{\text{Income}} = \beta_0 + \beta_1 \text{Education}_i$$



$$\hat{\text{Income}}_i = \beta_0 + \beta_1 \text{Education}_i$$

$$\text{Error}_i = \text{Income}_i - \hat{\text{Income}}_i =$$

$$= -\text{Income}_i - (\beta_0 + \beta_1 \text{Education}_i) = 4 - 5$$

$$\hat{\text{Income}} = \beta_0 + \beta_1 \text{Education}_i$$

$$\text{Error}_1 = \text{Income}_1 - \hat{\text{Income}}_1 = 4 - 5 = -1$$

$$\text{Error}_2 = \text{Income}_2 - \hat{\text{Income}}_2 = 5 - 4 = 1$$

- The way to write prediction error for individual i

$$
\begin{aligned}
\text{Error}_i &= \text{Income}_i - \widehat{\text{Income}}_i \\
&= \text{Income}_i - (\beta_0 + \beta_1 \text{Education}_i)
\end{aligned}
$$

- The way to write prediction error for individual i

$$\text{Error}_i = \text{Income}_i - \hat{\text{Income}}_i$$

$$= \text{Income}_i - (\beta_0 + \beta_1 \text{Education}_i)$$

- We want to penalize positive and negative errors, we can square the errors

$$\text{Error}_i^2 = (\text{Income}_i - (\beta_0 + \beta_1 \text{Education}_i))^2$$

- And the sum of squared errors, for all individuals from $i = 1, \ldots n$, is:

$$SSR(\beta_0, \beta_1) = (\text{Income}_1 - (\beta_0 + \beta_1 \text{Education}_1))^2 +$$

$$+ (\text{Income}_2 - (\beta_0 + \beta_1 \text{Education}_2))^2 + \ldots.$$

$$\ldots + (\text{Income}_n - (\beta_0 + \beta_1 \text{Education}_n))^2$$

- And the sum of squared errors, for all individuals from $i = 1, ...n$, is:

$$SSR(\beta_0, \beta_1) = (\text{Income}_1 - (\beta_0 + \beta_1 \text{Education}_1))^2 +$$

$$+ (\text{Income}_2 - (\beta_0 + \beta_1 \text{Education}_2))^2 + ...$$

$$... + (\text{Income}_n - (\beta_0 + \beta_1 \text{Education}_n))^2$$

- Which can be written as:

$$SSR(\beta_0, \beta_1) = \sum_{i=1}^{n} (\text{Error}_i)^2$$

$$SSR(\beta_0, \beta_1) = \sum_{i=1}^{n} (\text{Income}_i - (\beta_0 + \beta_1 \text{Education}_i))^2$$

- In linear regression, we want to find a line that minimizes the sum of squared residuals:

$$(\beta_0^*, \beta_1^*) \rightarrow \text{ minimize } SSR(\beta_0, \beta_1) = \sum_{i=1}^{n} (\beta_0 + \beta_1 \text{Education}_i - \text{Income}_i)^2$$

- Let us go to R and do some Linear Regressions
- We will run a linear regression to predict Wages based on Schooling and Age

$$\hat{\text{Wage}}_i = \beta_0 + \beta_1 \text{age}_i + \beta_2 \text{schooling}_i$$

- Let us go to R and do some Linear Regressions
- We will run a linear regression to predict Wages based on Schooling and Age

$$\hat{\text{Wage}}_i = \beta_0 + \beta_1 \text{age}_i + \beta_2 \text{schooling}_i$$

$$\hat{\text{Wage}}_i = 0.35 + 0.008\text{age}_i + 0.12\text{schooling}_i$$

Outline

Logistic regression: basics

- Let us consider the case of a binary outcome

$$y_i = \begin{cases} 0 \text{ if non-spam} \\ 1 \text{ if spam} \end{cases}$$

- We want to predict probabilities: $P(y_i = 1)$ based on regressors $x_1, ...x_p$
- In linear regression, you obtain predicted probabilities less than zero and greater than one
- Classification problems are rarely linear (e.g. image, sound recognition).

$$P(y_i = 1) = \beta_0 + x_{i,1}\beta_1, ... + x_{i,p}\beta_p$$

Logistic regression: basics II

- In Logistic regression, we assume non-linear function bounded between zero and one:

$$P(y_i = 1) = \frac{1}{1 + e^{-(\beta_0 + x_{i,1}\beta_1, \ldots + x_{i,p}\beta_p)}}$$

$$P(y_i = 0) = 1 - P(y_i = 1)$$

$$= 1 - \frac{1}{1 + e^{-(\beta_0 + x_{i,1}\beta_1, \ldots + x_{i,p}\beta_p)}}$$

- We often refer to $f(z) = \frac{1}{1+e^{-x}}$ as a 'sigmoid' function.

Logistic regression: basics III

- We want to find estimates of $\beta_0, ...\beta_p$ that generate good predictions....

Logistic regression: basics III

- We want to find estimates of $\beta_0, ...\beta_p$ that generate good predictions....
- $P(y_i = 1) = \frac{1}{1+e^{-x_i \beta}}$

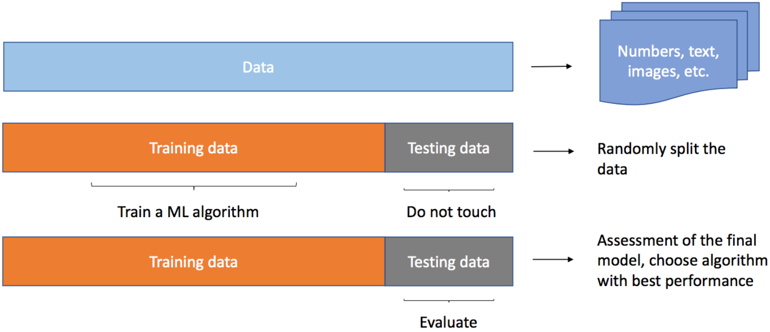
Logistic regression: basics III

- We want to find estimates of $\beta_0, ...\beta_p$ that generate good predictions....
- $P(y_i = 1) = \frac{1}{1+e^{-x_i\beta}}$
- If email is spam $(y_i = 1)$, we want $\frac{1}{1+e^{-x_i\beta}} \to 1$.

Logistic regression: basics III

- We want to find estimates of $\beta_0, ...\beta_p$ that generate good predictions....
- $P(y_i = 1) = \frac{1}{1+e^{-x_i\beta}}$
- If email is spam ($y_i = 1$), we want $\frac{1}{1+e^{-x_i\beta}} \to 1$.
- If email is non-spam ($y_i = 0$), we want $\frac{1}{1+e^{-x_i\beta}} \to 0$.

Supervised Learning workflow



Outline

Linear regression: II

- Predict outcome y_i, for individual i, based on regressors $x_{1,i}, \ldots x_{p,i}$.
- Prediction based on linear transformation:

$$\hat{y}_i = \beta_0 + \beta_1 x_{1,i} + \ldots \beta_p x_{p,i}$$

- Individual error: $e_i = y_i - \hat{y}_i$
- Individual squared error: $e_i^2 = (y_i - \hat{y}_i)^2$
- Sum of squared errors

$$\begin{aligned} SSR(\beta_1, \ldots. \beta_p) &= \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \\ &= \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_{1,i} + \ldots \beta_p x_{p,i}))^2 \end{aligned}$$

Linear regression: notation

- It will be convenient to write it in matrix-vectors:

$$\underbrace{\mathbf{x}_i}_{p\times 1} = \begin{bmatrix} x_{1,i} \\ x_{2,i} \\ \vdots \\ x_{p,i} \end{bmatrix} \; ; \; \underbrace{\beta}_{p\times 1} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix}$$

$$SSR(\beta) = \sum_{i=1}^{n} (y_i - \mathbf{x}_i'\beta)$$

Linear regression: notation II

$$\underbrace{Y}_{n \times 1} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

$$\underbrace{X}_{n \times p} = \begin{bmatrix} x_{1,1} & \cdots & x_{p,1} \\ \vdots & \ldots & \vdots \\ x_{1,n} & \cdots & x_{p,n} \end{bmatrix}$$

$$SSR(\beta) = (Y - X\beta)'(Y - X\beta)$$

Linear regression: goal

- Our goal is to find $\beta^* = \left[\beta_0^*, ...\beta_p^*\right]'$ that minimize the sum of squared errors

$$\beta^* = \arg\min_{\beta}(Y - X\beta)'(Y - X\beta)$$

$$= \arg\min_{\left[\beta_0^*, ...\beta_p^*\right]} \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_{1,i} + ...\beta_p x_{p,i}))^2$$

Linear regression: solution

1. Analytic solution

$$\beta^* = (X'X)^{-1}(X'Y)$$

2. Numerical (computational) solution:
    - Define function $SSR(\beta)$ in R
    - Find β^* as the vector minimizing $SSR(\beta)$

- As we delve deeper into ML algorithms, we will find that most complex methods do not have analytical solutions.
- It will be important, then, to learn how to minimize ***cost*** functions

Outline

Logistic regression: basics

- Let us consider the case of a binary outcome

$$y_i = \begin{cases} 0 \text{ if non-spam} \\ 1 \text{ if spam} \end{cases}$$

- We want to predict probabilities: $P(y_i = 1)$ based on regressors $x_1, \ldots x_p$
- In linear regression, you obtain predicted probabilities less than zero and greater than one
- Classification problems are rarely linear (e.g. image, sound recognition).

$$P(y_i = 1) = \left(\frac{1}{1 + e^{-x_i \beta}} \right)$$

Logistic regression: cost function II

- Let us analyze the following function:

$$g(y_i;\beta) = \left(\frac{1}{1+e^{-x_i\beta}}\right)^{y_i} \times \left(1 - \frac{1}{1+e^{-x_i\beta}}\right)^{1-y_i}$$

Logistic regression: cost function II

- Let us analyze the following function:

$$g(y_i;\beta) = \left(\frac{1}{1+e^{-x_i\beta}}\right)^{y_i} \times \left(1 - \frac{1}{1+e^{-x_i\beta}}\right)^{1-y_i}$$

- What happens to the $g()$ function $y_i = 1$ (spam):

Logistic regression: cost function II

- Let us analyze the following function:

$$g(y_i;\beta) = \left(\frac{1}{1+e^{-x_i\beta}}\right)^{y_i} \times \left(1 - \frac{1}{1+e^{-x_i\beta}}\right)^{1-y_i}$$

- What happens to the $g()$ function $y_i = 1$ (spam):

$$\begin{aligned} g(1;\beta) &= \left(\frac{1}{1+e^{-x_i\beta}}\right)^{1} \times \left(1 - \frac{1}{1+e^{-x_i\beta}}\right)^{1-1} = \\ &= \left(\frac{1}{1+e^{-x_i\beta}}\right)^{1} \times 1 \\ &= \left(\frac{1}{1+e^{-x_i\beta}}\right) \\ &= P(y_i = 1) \rightarrow \text{ Probability of being spam} \end{aligned} \tag{1}$$

Logistic regression: cost function III

If our email is non-spam, our $g()$ function becomes:

$$g(0;\beta) = \left(\frac{1}{1+e^{-x_i\beta}}\right)^{0} \times \left(1 - \frac{1}{1+e^{-x_i\beta}}\right)^{1-0} =$$

$$= \times \left(1 - \frac{1}{1+e^{-x_i\beta}}\right)^{1-0} =$$

$$= \left(1 - \frac{1}{1+e^{-x_i\beta}}\right)$$

$$= P(y_i = 0) \rightarrow \text{ Probability of being non-spam}$$

Logistic regression: cost function IV

- We can interpret function $g(y_i; \beta)$ as the probability of observation i of being y_i.
- We want the function $g(y_i; \beta)$ to be as large as possible (close to one).
- For the whole set of $i = 1, ...n$ observations, the joint probability mass function is:

$$f(y_1, ...y_n; \beta) = \prod_{i=1}^{n} g(y_i; \beta)$$

- We want the parameters β so that the function $f(y_1, ...y_n; \beta)$ to be as large as possible (close to one).

Logistic regression: cost function V

- If we want $f(y_1, ...y_n; \beta)$ to be as large as possible, we want $\ln\left(f(y_1, ...y_n; \beta)\right)$ to be as large as possible.

$$\ln\left(f(y_1, ...y_n; \beta)\right) = \ln\left(\prod_{i=1}^{n} g(y_i; \beta)\right)$$

$$= \sum_{i=1}^{n} \ln g(y_i; \beta)$$

$$= \sum_{i=1}^{n} \ln\left(\left(\frac{1}{1+e^{-x_i\beta}}\right)^{y_i} \times \left(1 - \frac{1}{1+e^{-x_i\beta}}\right)^{1-y_i}\right)$$

$$= \sum_{i=1}^{n} \left(y_i \ln\left(\frac{1}{1+e^{-x_i\beta}}\right) + (1-y_i)\ln\left(1 - \frac{1}{1+e^{-x_i\beta}}\right)\right)$$

Logistic regression: goal

$$= \sum_{i=1}^{n} \left(y_i \ln \left(\frac{1}{1+e^{-x_i\beta}} \right) + (1-y_i) \ln \left(1 - \frac{1}{1+e^{-x_i\beta}} \right) \right)$$

- We want this function to be as large as possible.
- Alternatively, we can define our cost function as the negative, and minimize it.
- Find β^* to minimize:

$$J(\beta; y_1, ...j_n) = -\sum_{i=1}^{n} \left(y_i \ln \left(\frac{1}{1+e^{-x_i\beta}} \right) + (1-y_i) \ln \left(1 - \frac{1}{1+e^{-x_i\beta}} \right) \right)$$

Logistic regression: goal

- Cross entropy cost:

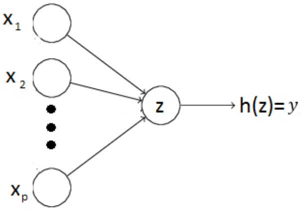
$$J(\beta; y_1, \ldots j_n) = -\sum_{i=1}^{n} \left(y_i \ln \left(\frac{1}{1+e^{-x_i\beta}} \right) + (1-y_i) \ln \left(1 - \frac{1}{1+e^{-x_i\beta}} \right) \right)$$

- (-log-likelihood function)
- Find β that minimizes the cros entropy cost = maximizes (log)-likelihood function

Enough theory,.... let us go to R

Logistic regression: further topics

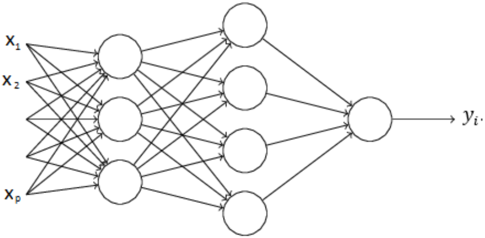
- Congratulations! You have built your first neural network



- $x_1, \ldots x_p$: inputs
- $z = x\beta$: linear transformation
- $h(z) = \frac{1}{1+e^z}$: activation function

Logistic regression: further topics

- General neural network is...



- Composed of various layers.
- Each layer composed of various neurons
- Each neuron is the result of a linear transformation+activation function

Outline

- Let's consider the Spam classification problem:

$$y_i = \begin{cases} 0 \text{ if email } i \text{ is no spam} \\ 1\text{if email } i \text{ is spam} \end{cases} \tag{2}$$

- For each email i we observe the occurrence of various words and characters. We call these features $x_i = \left[x_i^1, x_i^2, ...x_i^p\right]$ where

$$x_i^p = \begin{cases} 0 \text{ if email } i \text{ does not containt feature } p \\ 1 \text{ if email } i \text{ contains feature } p \end{cases} \tag{3}$$

- We are interested in predicting if email is spam or not based on characteristics:
- $f(y_i|x_i)$ probability density function of y_i **conditional** on x_i
- Bayes rule:

$$\underbrace{f(y_i|x_i)}_{posterior} = \frac{\underbrace{f(x_i|y_i)}_{likelihood}\underbrace{f(y_i)}_{prior}}{\underbrace{f(x_i)}_{evidence}} \tag{4}$$

- Bayesian classifier: we are interested in predictions generated from the posterior distibution.

Bayesian classifier

- To estimate the posterior distribution, we need estimates of the likelihood and the prior.
- First, let's see how to estimate the likelihood
- Recall, x_i is a vector.

$$f(x_i|y_i) = f(x_i^1, x_i^2, ...x_i^p|y_i) \tag{5}$$

- Naive bayes classifier assumes independence between x_i^p features.

$$f(x_i^1, x_i^2, ...x_i^p|y_i) = f(x_i^1|y_i) \times f(x_i^2|y_i) \times ... \times f(x_i^p|y_i) \tag{6}$$

- We interpret $f(x_i^1|y_i)$ as the **probability** of observing feature ^1_i if email is in category y_i
- $f(x_i^{53} = 1|y_i = 1)$: probability of email having exclamation point (!) if email is spam.
- $f()$ is the following probability mass function:

$$f(x_i^{53} = 1|y_i = 1) = \theta_{53,1} \quad (7)$$
$$f(x_i^{53} = 0|y_i = 1) = 1 - \theta_{53,1}$$

- We estimate $f(x_i^i|y_i)$ via maximum likelihood.
- In this example: estimating $f(x_i^i|y_i)$ via maximum likelihood $==$ proportion of emails in category y_i that contain feature x_i.

$$\underbrace{f(y_i|x_i)}_{posterior} = \frac{\underbrace{f(x_i|y_i)}_{likelihood}\underbrace{f(y_i)}_{prior}}{\underbrace{f(x_i)}_{evidence}} \tag{8}$$

- We estimated already the likelihood.
- Prior: $f(y_i)$ is simply the proportion of emails that are spam.
- We do not need to estimate the evidence. Regardless of what we choose, our estimates of the posterior will be the same.

Outline

Some review of statistics

Let X be continuous random variables. Let's denote a generic probability density function (pdf) by $f_x()$. Then:

$$
\begin{aligned}
1. P(X \leq b) = \int_{-\infty}^{b} f_x(x)dx = F(b) \\
2. \int_{-\infty}^{\infty} f_x(x)dx = 1 \\
3. f_x(x) \geq 0 \text{ for all x}
\end{aligned}
\tag{9}
$$

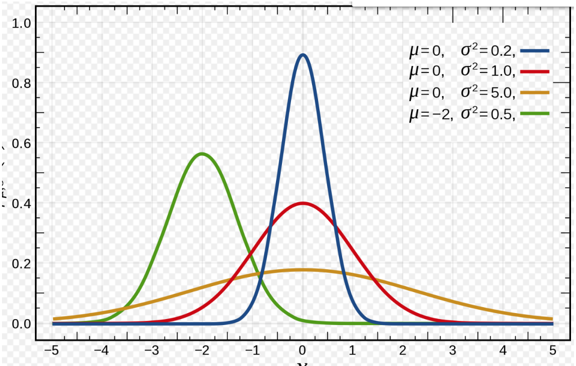
We call $F(.)$ cumulative distribution function

Do not interpret $f(x)$ as probability! $P(X = x) = 0$.

- Due to various results in statistics, the Normal distribution is a widely used function.
- We say X follows a normal distribution with mean μ and variance σ^2 if:

$$X \sim N(\mu, \sigma^2)$$

$$f_x(X) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2} \tag{10}$$



- Let's go back to the wage prediction problem.
- Let's assume log-wage is given by:

$$y_i = \beta_0 + \beta_1 x_{1,i} + \beta_2 x_{2,i} + \varepsilon_i \tag{11}$$

but now, we assume that ε_i follows a normal distribution:

$$\varepsilon_i \sim N(0, \sigma^2) \tag{12}$$

If this is the case, then:

$$y_i \sim N(\beta_0 + \beta_1 x_{1,i} + \beta_2 x_{2,i}, \sigma^2) \tag{13}$$

$$f(y_i; x_i, \beta, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(y_i - (\beta_0 + \beta_1 x_{1,i} + \beta_2 x_{2,i}))^2} \tag{14}$$

Maximum Likelihood Estimation

- The **joint** probability density function of all the data is given by:

$$f(Y;X,\beta,\sigma^2) = f_Y(y_1,....y_n;X,\beta,\sigma^2)$$

- The Likelihood Function is given by:

$$\mathcal{L}(\beta,\sigma^2|Y,X) = f_Y(y_1,....y_n;X,\beta,\sigma^2) \tag{15}$$

- Two random variables are independent iff their joint pdf is the product of their pdf.
- We assume that wages are independent between individuals:

$$f_Y(y_1,....y_n;X,\beta,\sigma^2) = f_y(y_1;X_1,\beta,\sigma^2)\times f_y(y_2;X_2,\beta,\sigma^2)\times$$

$$...\times f_y(y_n;X_n,\beta,\sigma^2)$$

Maximum Likelihood Estimation

- The independence assumption implies that:

$$\mathcal{L}(\beta, \sigma^2 | Y, X) = \prod_{i=1}^{n} f_y(y_i; X_i, \beta, \sigma^2) \tag{16}$$

- We usually work with the log-likelihood function for various reasons...

$$l(\beta, \sigma^2 | Y, X) = \ln\left(\prod_{i=1}^{n} f_y(y_i; X_i, \beta, \sigma^2)\right) \tag{17}$$

$$= \sum_{i=1}^{n} \ln\left(f_y(y_i; X_i, \beta, \sigma^2)\right)$$

- $\left[\hat{\beta}_{MLE}, \hat{\sigma}^2_{MLE}\right] = \arg\max \ln\left(\prod_{i=1}^{n} f_y(y_i; X_i, \beta, \sigma^2)\right)$
- Let's go to R and do some work.

Outline