About Regression

Richard Yi Da Xu
Yida.Xu@uts.edu.au
Wechat: aubedata
www-staff.it.uts.edu.au/~ydxu/

Director of Machine Learning and Data Analytics Lab::GBDTC:: UTS

February 18, 2018

Start with two **classification** type regression:

- Logistic Regression
- Softmax(Multinomial) regression

the **BIG** idea for logistic regression

Assume there is **two classes** that the classifier is capable to classify:

- if data x_1 has label $y_1 = 1$.
- if data x_2 has label $y_2 = 0$
- if data x_3 has label $y_3 = 1$
- if data x_4 has label $y_4 = 0$
- if data x_5 has label $y_5 = 1$
- ...

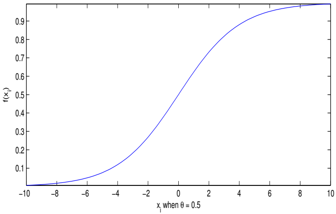
So basically, we want to build a classifier $f(x, \theta)$, such that:

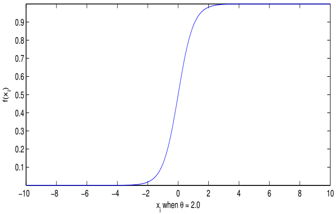
- $(y_1 = 1) \implies f(x_1, \theta)$ should be as close to 1 as possible, e.g., $f(x_1, \theta) = 0.99$
- $(y_2 = 2) \implies f(x_2, \theta)$ should be as close to 0 as possible, e.g., $f(x_2, \theta) = 0.005$
- $(y_3 = 2) \implies f(x_3, \theta)$ should be as close to 1 as possible, e.g., $f(x_3, \theta) = 0.92$
- $(y_4 = 3) \implies f(x_4, \theta)$ should be as close to 0 as possible, e.g., $f(x_4, \theta) = 0.1$
- $(y_5 = 4) \implies f(x_5, \theta)$ should be as close to 1 as possible, e.g., $f(x_5, \theta) = 0.93$
- ...

Let's see the mechanism to achieve this.

- first we need some squashing (activation) function to map values between $(0 \dots 1)$.

$$\sigma(t) = \frac{1}{1+\exp(-t)} = \frac{\exp(t)}{\exp(t)+1} \qquad \rightarrow \qquad \sigma(\mathbf{x}_i^\top \theta) = \frac{1}{1-\exp(\mathbf{x}_i^T \boldsymbol{\theta} + \theta_0)}$$





Properties of sigmoid function

$$1 - \sigma(t) = 1 - \frac{1}{1+\exp(-t)} = \frac{1+\exp(-t)-1}{1+\exp(-t)} = \frac{\exp(-t)}{1+\exp(-t)} = \sigma(-t)$$

$$\frac{d\sigma(t)}{dt} = \frac{d\left(\frac{1}{1+\exp(-t)}\right)}{dt} = \frac{\exp(-t)}{(1+\exp(-t))^2} = \left(\frac{1}{1+\exp(-t)}\right)\left(\frac{\exp(-t)}{1+\exp(-t)}\right)$$

$$= \sigma(t)(1-\sigma(t)) \qquad \text{it's always positive}$$

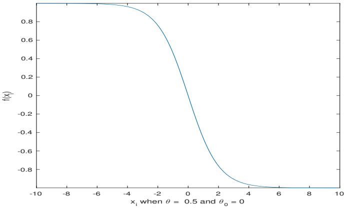
$$\frac{d\sigma(-t)}{dt} = \frac{d\left(\frac{1}{1+\exp(t)}\right)}{dt} = \frac{-\exp(t)}{(1+\exp(t))^2} = -\underbrace{\left(\frac{1}{1+\exp(t)}\right)}_{\sigma(-t) \text{ or } 1-\sigma(t)}\underbrace{\left(\frac{\exp(t)}{1+\exp(t)}\right)}_{\sigma(t)}$$

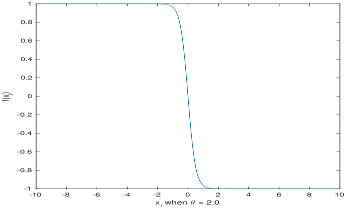
$$= -\sigma(t)(1-\sigma(t)) \qquad \text{it's always negative}$$

Another squashing (activation) tanh function:

- obviously it's inappropriate for logistic regression, as tanh maps values $(-1 \dots 1)$.

$$\tanh(x) = \frac{\exp^x - \exp^{-x}}{\exp^x + \exp^{-x}} = \frac{\exp^{2x} - 1}{\exp^{2x} + 1} = \frac{1 - \exp^{-2x}}{1 + \exp^{-2x}}$$





tanh derivatives

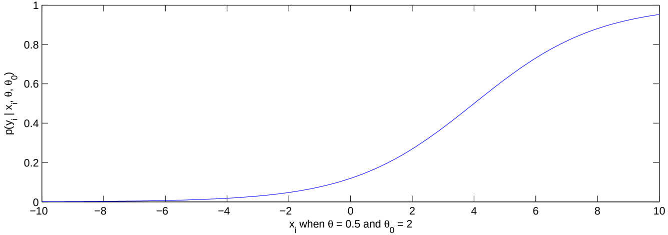
$$\frac{d(f/g)}{dx} = \frac{gf' - fg'}{g^2} \implies \frac{\partial \tanh(x)}{\partial x} = \frac{\partial}{\partial x}\left(\frac{\sinh(x)}{\cosh(x)}\right) = \frac{\cosh \cdot \sinh' - \sinh \cdot \cosh'}{\cosh^2}$$

$$\text{using: } \sinh' = \frac{1}{2}\left(\exp^{x} + \exp^{-x}\right) = \cosh \qquad \cosh' = \frac{1}{2}\left(\exp^{x} - \exp^{-x}\right) = \sinh$$

$$\frac{\partial \tanh(x)}{\partial x} = \frac{\cosh^2 - \sinh^2}{\cosh^2} = 1 - \left(\frac{\sinh}{\cosh}\right)^2 = 1 - \tanh^2$$

Logistic Regression

- Finding a **single** Bernoulli probability: $\Pr(y_i = 1 | \underbrace{x_i^T \boldsymbol{\theta}}_{p})$:



- Write $\{x_i, 1\} \rightarrow x_i$ and $\{\boldsymbol{\theta}, \theta_0\} \rightarrow \boldsymbol{\theta}$:

$$\Pr(\mathbf{Y}|\mathbf{X}, \boldsymbol{\theta}) = \prod_{i=1}^{n} \left[\frac{1}{1 + \exp(-x_i^T \boldsymbol{\theta})} \right]^{y_i} \left[1 - \frac{1}{1 + \exp(-x_i^T \boldsymbol{\theta})} \right]^{1-y_i}$$

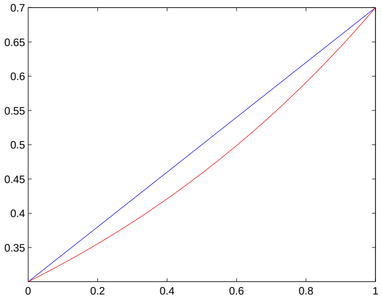
Maximization is performed on the log space:

$$\mathcal{C}(\boldsymbol{\theta}) = -\log[p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\theta})] = -\left(\sum_{i=1}^{n} y_i \log \left[\frac{1}{1 + \exp(-x_i^T \boldsymbol{\theta})} \right] + (1 - y_i) \log \left[1 - \frac{1}{1 + \exp(-x_i^T \boldsymbol{\theta})} \right] \right)$$

Think Bernoulli again

Both functions are the same at end-points $\{0, 1\}$:

$$f_p(x) = (1-p)^{(1-x)} p^x$$
$$f_p(x) = (1-p)(1-x) + px$$



the BIG idea for Softmax

Assume there is four classes that the classifier is capable to classify:

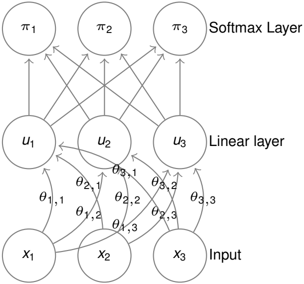
- if data x_1 has label $y_1 = 1$, then y_1 can be written as $[1, 0, 0, 0]^\top$.
- if data x_2 has label $y_2 = 2$, then y_2 can be written as $[0, 1, 0, 0]^\top$.
- if data x_3 has label $y_3 = 2$, then y_3 can be written as $[0, 1, 0, 0]^\top$.
- if data x_4 has label $y_4 = 3$, then y_4 can be written as $[0, 0, 1, 0]^\top$.
- if data x_5 has label $y_5 = 4$, then y_5 can be written as $[0, 0, 0, 1]^\top$.
- $\dots$

So basically, we want to build a classifer $f(x, \theta)$, such that:

- $(y_1 = 1) \implies f(x_1, \theta)$ should be as close to $[1, 0, 0, 0]^\top$ as possible, e.g., $f(x_1, \theta) = [0.99, 0.005, 0.005, 0]^\top$
- $(y_2 = 2) \implies f(x_2, \theta)$ should be as close to $[0, 1, 0, 0]^\top$ as possible, e.g., $f(x_2, \theta) = [0.005, 0.99, 0.005, 0]^\top$
- $(y_3 = 2) \implies f(x_3, \theta)$ should be as close to $[0, 1, 0, 0]^\top$ as possible, e.g., $f(x_3, \theta) = [0.005, 0.99, 0.005, 0]^\top$
- $(y_4 = 3) \implies f(x_4, \theta)$ should be as close to $[0, 0, 1, 0]^\top$ as possible, e.g., $f(x_4, \theta) = [0.005, 0.005, 0.99, 0]^\top$
- $(y_5 = 4) \implies f(x_5, \theta)$ should be as close to $[0, 0, 0, 1]^\top$ as possible, e.g., $f(x_5, \theta) = [0.005, 0.005, 0, 0.99]^\top$
- $\dots$

Let's see the mechanism to achieve this.

Expand to multiple classes: Softmax



$$\boldsymbol{\theta}_1 = \{\theta_{1,1}, \theta_{1,2}, \theta_{1,3}\} \quad \boldsymbol{\theta}_2 = \{\theta_{2,1}, \theta_{2,2}, \theta_{2,3}\} \quad \boldsymbol{\theta}_3 = \{\theta_{3,1}, \theta_{3,2}, \theta_{3,3}\}$$

- Linear Layer

$$\begin{aligned} u_1 &= \mathbf{x}^T \boldsymbol{\theta}_1 = \sum_{i=1}^{3} x_{1,i}\theta_{1,i} \\ u_2 &= \mathbf{x}^T \boldsymbol{\theta}_2 = \sum_{i=1}^{3} x_{2,i}\theta_{2,i} \\ u_3 &= \mathbf{x}^T \boldsymbol{\theta}_3 = \sum_{i=1}^{3} x_{3,i}\theta_{3,i} \end{aligned}$$

- Softmax Layer

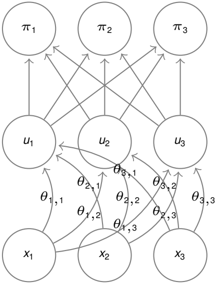
$$\begin{aligned} \pi_1 &\equiv \Pr(y_i = 1 | \mathbf{x}_i, \boldsymbol{\theta}) = \frac{\exp(u_1)}{\sum_{k=1}^{3} \exp(u_k)} = \frac{\exp(\mathbf{x}_i^T \boldsymbol{\theta}_1)}{\sum_{k=1}^{3} \exp(\mathbf{x}_i^T \boldsymbol{\theta}_k)} \\ \pi_2 &\equiv \Pr(y_i = 2 | \mathbf{x}_i, \boldsymbol{\theta}) = \frac{\exp(u_2)}{\sum_{k=1}^{3} \exp(u_k)} = \frac{\exp(\mathbf{x}_i^T \boldsymbol{\theta}_2)}{\sum_{k=1}^{3} \exp(\mathbf{x}_i^T \boldsymbol{\theta}_k)} \\ \pi_3 &\equiv \Pr(y_i = 3 | \mathbf{x}_i, \boldsymbol{\theta}) = \frac{\exp(u_3)}{\sum_{k=1}^{3} \exp(u_k)} = \frac{\exp(\mathbf{x}_i^T \boldsymbol{\theta}_3)}{\sum_{k=1}^{3} \exp(\mathbf{x}_i^T \boldsymbol{\theta}_k)} \end{aligned}$$

- $\pi_1 + \pi_2 + \pi_3 = 1!$
- easily extend them to multiple K arbitary classes

Relationship between 2-class Softmax and Logistic regression

$$
\begin{aligned}
\pi_1 \equiv \pi(y_i = 1 | \mathbf{x}_i, \boldsymbol{\theta}) &= \frac{\exp(\mathbf{x}^T \boldsymbol{\theta}_1)}{\exp(\mathbf{x}^T \boldsymbol{\theta}_1) + \exp(\mathbf{x}^T \boldsymbol{\theta}_2)} \\
&= \frac{1}{1 + \exp\left(\mathbf{x}^T (\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1)\right)} \\
&= \frac{1}{1 + \exp\left(\mathbf{x}^T (-\boldsymbol{\theta})\right)} \\
&= \frac{\exp\left(\mathbf{x}^T \boldsymbol{\theta}\right)}{\exp\left(\mathbf{x}^T \boldsymbol{\theta}\right) + 1}
\end{aligned}
$$

Softmax in our three class example



$$\pi_{1:\text{Data Scientist}} = \frac{\exp(u_1)}{\sum_{l=1}^{3} u_l} = \frac{\exp(\mathbf{x}^T \boldsymbol{\theta}_1)}{\sum_{l=1}^{3} \exp(\mathbf{x}^T \boldsymbol{\theta}_l)}$$

$$\pi_{2:\text{Scholar}} = \frac{\exp(u_2)}{\sum_{l=1}^{3} u_l} = \frac{\exp(\mathbf{x}^T \boldsymbol{\theta}_2)}{\sum_{l=1}^{3} \exp(\mathbf{x}^T \boldsymbol{\theta}_l)}$$

$$\pi_{3:\text{CEO}} = \frac{\exp(u_3)}{\sum_{l=1}^{3} u_l} = \frac{\exp(\mathbf{x}^T \boldsymbol{\theta}_3)}{\sum_{l=1}^{3} \exp(\mathbf{x}^T \boldsymbol{\theta}_l)}$$

	attribute 1	attribute 2	attribute 3	Occupation
Attendee 1	50	64	1.2	Data scientist
Attendee 2	23	23	15	Scholar
Attendee 3	50	80	3.2	Data scientist
...	...	...	...	...
Attendee N	5	90	25	CEO
Attendee N+1	60	43	12	?

So, substitute into the neural network, you hope to get a set of $\boldsymbol{\theta}$, such that:

$$\mathbf{x}^{(1)} = (50, 64, 1.2) \implies (\pi_{1:\text{Data Scientist}} = \mathbf{1}, \pi_{2:\text{Scholar}} = 0, \pi_{3:\text{CEO}} = 0)$$

$$\mathbf{x}^{(2)} = (23, 23, 15) \implies (\pi_{1:\text{Data Scientist}} = 0, \pi_{2:\text{Scholar}} = \mathbf{1}, \pi_{3:\text{CEO}} = 0)$$

$$\mathbf{x}^{(3)} = (50, 80, 3.2) \implies (\pi_{1:\text{Data Scientist}} = \mathbf{1}, \pi_{2:\text{Scholar}} = 0, \pi_{3:\text{CEO}} = 0)$$

$$\dots$$

$$\mathbf{x}^{(N)} = (5, 90, 25) \implies (\pi_{1:\text{Data Scientist}} = 0, \pi_{2:\text{Scholar}} = 0, \pi_{3:\text{CEO}} = \mathbf{1})$$

An unique perfect $\boldsymbol{\theta}$ suitable for all the data obviously do *NOT* exist. We need to find a $\boldsymbol{\theta}$ which minimize the sum of cost.

Minimize cost function example: Softmax

- Logistic regression:

$$p(\mathbf{Y}|\mathbf{X},\boldsymbol{\theta}) = \prod_{i=1}^{N}\left[\frac{1}{1+\exp(-\mathbf{x}_i^T\boldsymbol{\theta})}\right]^{y_i}\left[1-\frac{1}{1+\exp(-\mathbf{x}_i^T\boldsymbol{\theta})}\right]^{1-y_i}$$

- Softmax:

$$p(\mathbf{Y}|\mathbf{X},\boldsymbol{\theta}) = \prod_{i=1}^{N}\prod_{k=1}^{K}\left[\frac{\exp(\mathbf{x}_i^T\boldsymbol{\theta}_k)}{\sum_{l=1}^{K}\exp(\mathbf{x}_i^T\boldsymbol{\theta}_l)}\right]^{y_{i,k}}$$

we write y_i as an indicator vector, e.g., [100], [101]

$$\mathcal{C}(\boldsymbol{\theta}) = -\sum_{i=1}^{N}\sum_{k=1}^{K}\underbrace{y_{i,k}}_{p(x)}\underbrace{\log\left[\frac{\exp(\mathbf{x}_i^T\boldsymbol{\theta}_k)}{\sum_{l=1}^{K}\exp(\mathbf{x}_i^T\boldsymbol{\theta}_l)}\right]}_{\log(q(x))}$$

note the cross entropy form: $H(p,q) = -\sum_x p(x)\log(q(x))$

$$= -\sum_{i=1}^{N}\sum_{k=1}^{K}y_{i,k}\left[\log\exp(\mathbf{x}_i^T\boldsymbol{\theta}_k) - \log\sum_{l=1}^{K}\exp(\mathbf{x}_i^T\boldsymbol{\theta}_l)\right]$$

- What is cross entropy?

Cost function: Cross Entropy

Minimize the cost function: Cross Entropy can be thought of "how much" deviates between the "true" density $p(x)$ and recovered density "q(x)".

$$
\begin{aligned}
H(p,q) &= \mathrm{E}_p[-\log q] \\
&= -\sum_x p(x)\ \log q(x) \\
&= \sum_x \left[\underbrace{-p(x)\ \log q(x) + p(x)\log(p(x))} \underbrace{-p(x)\log(p(x))} \right] \\
&= \sum_x \underbrace{-p(x)\ \log p(x)} + \sum_x p(x)\ \log \frac{p(x)}{q(x)} \\
&= H(p) + D_{\mathrm{KL}}(p\|q),
\end{aligned}
$$

Question If minimising **cross entropy loss** is equivalent of maximising multinomial likelihood estimation, then minimising **square loss** is equivalent of maximising what likelihood estimation?

Softmax or multinomial regression objective function

$$\mathcal{C}(\boldsymbol{\theta}) = -\sum_{i=1}^{N}\sum_{k=1}^{K} y_{i,k}\left[\log\left(\frac{\exp(\mathbf{x}_i^T\boldsymbol{\theta}_k)}{\sum_{l=1}^{K}\exp(\mathbf{x}_i^T\boldsymbol{\theta}_l)}\right)\right]$$

How may we find:

$$\arg\min_{\boldsymbol{\theta}} \mathcal{C}(\boldsymbol{\theta})$$

We solve it using Gradient descend.

Find the derivatives of Multinomial regression

$$\mathcal{C}(\boldsymbol{\theta}) = -\sum_{i=1}^{N}\sum_{k=1}^{K} y_{i,k}\left[\log\left(\frac{\exp(\mathbf{x}_i^T\boldsymbol{\theta}_k)}{\sum_{l=1}^{K}\exp(\mathbf{x}_i^T\boldsymbol{\theta}_l)}\right)\right]$$

Let $Z_k = \mathbf{x}_i^T\boldsymbol{\theta}_k$, we have,

$$\begin{aligned}
\frac{\partial\mathcal{C}(\boldsymbol{\theta})}{\partial Z_k} &= \sum_{i=1}^{N}\frac{\partial\left(-\sum_{k=1}^{K} y_{i,k}\left[\log\left(\frac{\exp(Z_k)}{\sum_{l=1}^{K}\exp(Z_l)}\right)\right]\right)}{\partial Z_k}\\
&= \sum_{i=1}^{N}\left[\frac{\exp(Z_k)}{\sum_{l=1}^{K}\exp(Z_l)} - y_{i,k}\right] \qquad \text{see next page}\\
\Longrightarrow \frac{\partial\mathcal{C}(\boldsymbol{\theta})}{\partial Z} &= \sum_{i=1}^{N}\begin{bmatrix}\frac{\exp(Z_1)}{\sum_{l=1}^{K}\exp(Z_l)} - y_{i,1}\\ \dots\\ \frac{\exp(Z_K)}{\sum_{l=1}^{K}\exp(Z_l)} - y_{i,K}\end{bmatrix} = \sum_{i=1}^{N}\left(\frac{\exp(Z)}{\sum_{l=1}^{K}\exp(Z_l)} - y_i\right)
\end{aligned}$$

If we were to differentiate with respect to $\boldsymbol{\theta}$:

$$\frac{\partial\mathcal{C}(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}} = \frac{\partial\mathcal{C}(\boldsymbol{\theta})}{\partial Z}\frac{\partial Z}{\partial\boldsymbol{\theta}} = \sum_{i=1}^{N}\mathbf{x}_i\left[\frac{\exp(\mathbf{x}_i^T\boldsymbol{\theta})}{\sum_{l=1}^{K}\exp(\mathbf{x}_i^T\boldsymbol{\theta}_l)} - y_i\right]^{\top}$$

Derivative for Softmax regression

$$p_k = \frac{\exp^{z_k}}{\sum_l \exp^{z_l}} \qquad \mathcal{C} = -\sum_k y_k \log p_k, \qquad \text{where } \sum_k y_k = 1$$

when $k = i$:

$$\frac{\partial p_k}{\partial z_k} = \frac{\partial \left(\frac{\exp^{z_k}}{\sum_l \exp^{z_l}}\right)}{\partial z_k} = \frac{\partial \left(\frac{\exp^{z_k}}{\exp^{z_k} + \sum_{l \neq k} \exp^{z_l}}\right)}{\partial \exp(z_k)} \times \frac{\partial \exp^{z_k}}{\partial z_k}$$

We know the identity:

$$\frac{\partial \frac{x}{x+c}}{\partial x} = \frac{\partial x(x+c)^{-1}}{\partial x} = (x+c)^{-1} - x(x+c)^{-2} = \frac{(x+c) - x}{(x+c)^2} = \frac{c}{(x+c)^2}$$

Therefore,

$$\frac{\partial p_k}{\partial z_k} = \frac{\partial \left(\frac{\exp^{z_k}}{\exp^{z_k} + \sum_{l \neq k} \exp^{z_l}}\right)}{\partial \exp(z_k)} \times \frac{\partial \exp^{z_k}}{\partial z_k} = \frac{\sum_{l \neq k} \exp^{z_l}}{\left(\sum_{l=1}^K \exp^{z_l}\right)^2} \times \exp^{z_k}$$

$$= \frac{\sum_{l \neq k} \exp^{z_l}}{\left(\sum_{l=1}^K \exp^{z_l}\right)} \times \frac{\exp^{z_k}}{\left(\sum_{l=1}^K \exp^{z_l}\right)} = p_k(1 - p_k)$$

Derivative for Softmax regression

when $k \neq i$, We know the identity:

$$\frac{\partial \frac{y}{z+c}}{\partial z} = \frac{\partial y(z+c)^{-1}}{\partial z} = -\frac{y}{(z+c)^2}$$

$$\begin{aligned}
\frac{\partial p_k}{\partial z_i} &= \frac{\left(\partial \frac{\exp^{z_k}}{\exp^{z_i} + \sum_{l \neq i} \exp^{z_l}}\right)}{\partial \exp(z_i)} \times \frac{\partial \exp^{z_i}}{\partial z_i} \\
&= -\frac{\exp^{z_k}}{\left(\sum_{l=1}^{K} \exp^{z_l}\right)^2} \times \exp^{z_i} \\
&= -\frac{\exp^{z_i}}{\left(\sum_{l=1}^{K} \exp^{z_l}\right)} \times \frac{\exp^{z_k}}{\left(\sum_{l=1}^{K} \exp^{z_l}\right)} = -p_i p_k
\end{aligned}$$

Derivative for Softmax regression

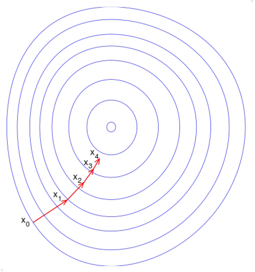
$$\frac{\partial p_k}{\partial z_i} = \begin{cases} p_i(1 - p_i), & i = k \\ -p_i p_k, & i \neq k \end{cases}$$

$$\begin{aligned}
\frac{\partial \mathcal{C}}{\partial z_i} &= -\sum_k y_k \frac{\partial \log p_k}{\partial z_i} \\
&= -\sum_k y_k \frac{1}{p_k} \frac{\partial p_k}{\partial z_i} \\
&= -y_i(1 - p_i) - \sum_{k \neq i} y_k \frac{1}{p_k}(-p_k p_i) \\
&= -y_i(1 - p_i) + \sum_{k \neq i} y_k(p_i) \\
&= -y_i + y_i p_i + \sum_{k \neq i} y_k(p_i) \\
&= p_i \left(\sum_k y_k \right) - y_i \\
&= p_i - y_i
\end{aligned}$$

Gradient Descend for multinomial regression

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \alpha_n \nabla f(\mathbf{x}_n), \;\; n \geq 0$$

$$\mathcal{C}(\boldsymbol{\theta}) = -\sum_{i=1}^{N}\sum_{k=1}^{K} y_{i,k} \log\left[\frac{\exp(\mathbf{x}_i^T\boldsymbol{\theta}_k)}{\sum_{l=1}^{K}\exp(\mathbf{x}_i^T\boldsymbol{\theta}_l)}\right] + \frac{\lambda}{2}\sum_{k=1}^{K}\sum_{j=1}^{m}\theta_{k,j}^2$$



$$\boldsymbol{\theta}^{n+1} = \boldsymbol{\theta}^n - \alpha_n\left(\sum_{i=1}^{N}\mathbf{x}_i\left[\frac{\exp(\mathbf{x}_i^T\boldsymbol{\theta})}{\sum_{l=1}^{K}\exp(\mathbf{x}_i^T\boldsymbol{\theta}_l)} - y_i\right]^\top + \lambda\boldsymbol{\theta}^n\right), \;\; n \geq 0$$

What's this $\frac{\lambda}{2}\sum_{k=1}^{K}\sum_{j=1}^{m}\theta_{k,j}^2$ business? It's the **regulariser** we added.

Linear Regression

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

where

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \qquad \mathbf{X} = \begin{pmatrix} \mathbf{x}_1^{\mathrm{T}} \\ \mathbf{x}_2^{\mathrm{T}} \\ \vdots \\ \mathbf{x}_n^{\mathrm{T}} \end{pmatrix} = \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ x_{21} & \cdots & x_{2p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{pmatrix} \qquad \boldsymbol{\beta} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix} \qquad \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

You have seen it so many times! There is an analytical solution to it:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y} = \left(\sum \mathbf{x}_i \mathbf{x}_i^{\mathrm{T}}\right)^{-1} \left(\sum \mathbf{x}_i y_i\right).$$

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

where

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \qquad \mathbf{X} = \begin{pmatrix} \mathbf{x}_1^{\mathrm{T}} \\ \mathbf{x}_2^{\mathrm{T}} \\ \vdots \\ \mathbf{x}_n^{\mathrm{T}} \end{pmatrix} = \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ x_{21} & \cdots & x_{2p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{pmatrix} \qquad \boldsymbol{\beta} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix} \qquad \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

You have seen it so many times! There is an analytical solution to it:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y} = \left(\sum \mathbf{x}_i \mathbf{x}_i^{\mathrm{T}}\right)^{-1} \left(\sum \mathbf{x}_i y_i\right).$$

R^2

$$R^2 = 1 - \frac{\sum_i \Big(\overbrace{y_i - \hat{y}_i}^{r_i}\Big)^2}{\sum_i (y_i - \bar{y})^2}$$

- the better linear regression fits the data relative to a simple average, the closer the value of R^2 is to 1.
- If the chosen model fits worse than a horizontal line, then R^2 is negative

Polynomial Regression

For each of the data pairs (x_i, y_i):

$$y_i \;=\; a_0 + a_1 x_i + a_2 x_i^2 + \cdots + a_m x_i^m + \varepsilon_i$$

The model can be written as a system of **linear equations**:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^m \\ 1 & x_2 & x_2^2 & \dots & x_2^m \\ 1 & x_3 & x_3^2 & \dots & x_3^m \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^m \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \vdots \\ \varepsilon_n \end{bmatrix},$$

or,

$$\vec{y} = \mathbf{X}\vec{a} + \vec{\varepsilon}.$$

Ordinary least squares estimation is:

$$\widehat{\vec{a}} = (\mathbf{X}^T\mathbf{X})^{-1}\,\mathbf{X}^T\vec{y}.$$

Mixed effect model

Imagine we have the following setting:
A particular company *"develops and markets products for the early detection of target organ damage and management of cardiovascular and renal disease."*. In five years time, this company will have 10000 employees worldwide, HR needs a good model to reward its employees:

- has $N = 10000$ number of employees.
- has $q = 45$ working departments throughout the world.
- each department has N_q number of employees, i.e, $N = \sum_j^q n_j$
- The HR has collected $p = 6$ attributes from each employees, including:
  - utilisation hours
  - yrs of experience
  - salary
  - last year's rating
  - number of awards received
  - market value
- each y_i is the amount of "shared" profit each employee should receive.

Mixed effect model

- In the matrix form:

$$\mathbf{y} = X\beta + Z\gamma + \varepsilon$$

- write down the dimensionality:

$$\overbrace{\mathbf{y}}^{N \times 1} = \overbrace{\underbrace{\mathbf{X}}_{N \times p} \underbrace{\beta}_{p \times 1}}^{N \times 1} + \overbrace{\underbrace{\mathbf{Z}}_{N \times q} \underbrace{\gamma}_{q \times 1}}^{N \times 1} + \overbrace{\varepsilon}^{N \times 1}$$

- substitute numbers into:

$$\overbrace{\mathbf{y}}^{10000x1} = \overbrace{\underbrace{\mathbf{X}}_{10000x6} \underbrace{\beta}_{6 \times 1}}^{10000x1} + \overbrace{\underbrace{\mathbf{Z}}_{10000x45} \underbrace{\gamma}_{45 \times 1}}^{10000x1} + \overbrace{\varepsilon}^{10000x1}$$

$$\mathbf{X} = \begin{bmatrix} \text{utilisation hours} & \text{yrs of experience} & \text{salary} & \text{last year's rating} & \text{number of clients} & \text{market value} \\ 1 & 64.97 & 0 & 1 & 6087 & 4.87 \\ 1 & 53.92 & 0 & 0 & 6700 & 4.68 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 56.07 & 0 & 1 & 6430 & 4.73 \end{bmatrix}$$

- $\mathbf{z}_i$ is a **one-hot** vector, indicating which Department the employee belong to.