ML&MEA (2024)
Lecture 6 - Logistic Regression

Quanying Liu

BME, SUSTech

2024.3.12

Content

Recap Lecture 4

- Bayesian view: The parameters β are not constant, which have their prior distributions $P(\beta)$.
  Now we assume $\beta \sim \mathcal{N}(0, \sigma_0^2)$, it becomes ridge regression.

Recap Lecture 4

- Bayesian view: The parameters β are not constant, which have their prior distributions $P(\beta)$.
  Now we assume $\beta \sim \mathcal{N}(0, \sigma_0^2)$, it becomes ridge regression.
- Probabilistic model for ridge regression:

$$
\begin{aligned}
y &= \beta^T \mathbf{x} + \epsilon, \\
\epsilon &\sim \mathcal{N}\left(0, \sigma^2\right), \\
\beta &\sim \mathcal{N}(0, \sigma_0^2).
\end{aligned}
$$

Recap Lecture 4

- Bayesian view: The parameters β are not constant, which have their prior distributions $P(\beta)$.
  Now we assume $\beta \sim \mathcal{N}(0, \sigma_0^2)$, it becomes ridge regression.
- Probabilistic model for ridge regression:

$$
\begin{aligned}
y &= \beta^T \mathbf{x} + \epsilon, \\
\epsilon &\sim \mathcal{N}\left(0, \sigma^2\right), \\
\beta &\sim \mathcal{N}(0, \sigma_0^2).
\end{aligned}
$$

- Likelihood:

$$
P(y \mid \beta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{ -\frac{\left(y - \beta^T \mathbf{x}\right)^2}{2\sigma^2} \right\}
$$

Recap Lecture 4

- Bayesian view: The parameters β are not constant, which have their prior distributions $P(\beta)$.
  Now we assume $\beta \sim \mathcal{N}(0, \sigma_0^2)$, it becomes ridge regression.
- Probabilistic model for ridge regression:

$$y = \beta^T \mathbf{x} + \epsilon,$$
$$\epsilon \sim \mathcal{N}\left(0, \sigma^2\right),$$
$$\beta \sim \mathcal{N}(0, \sigma_0^2).$$

- Likelihood:

$$P\left(y \mid \beta\right) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{\left(y - \beta^T \mathbf{x}\right)^2}{2\sigma^2}\right\}$$

- Bayes' theorem:

$$P(\beta|\mathbf{y}) = \frac{P(\mathbf{y}|\beta)P(\beta)}{P(\mathbf{y})}$$

Recap Lecture 4

- Let's derive the MAP:

$$
\begin{aligned}
\hat{\beta}_{MAP} &= \arg\max_{\beta} \log P(\mathbf{y}|\mathbf{X};\beta) + \log p(\beta) \\
&= \arg\max_{\beta} \sum_{i=1}^{N} \log\left(\frac{1}{\sigma\sqrt{2\pi}}\right) + \log\left(\frac{1}{\sigma_0\sqrt{2\pi}}\right) \\
&\quad - \left[\sum_{i=1}^{N} \frac{(y^{(i)} - \beta^T \mathbf{x}^{(i)})^2}{2\sigma^2} + \frac{\|\beta\|^2}{2\sigma_0^2}\right] \\
&= \arg\min_{\beta} \sum_{i=1}^{N} \left(y^{(i)} - \beta^T \mathbf{x}^{(i)}\right)^2 + \frac{\sigma^2}{\sigma_0^2}\|\beta\|^2 \\
&= \arg\min_{\beta} \|\mathbf{y} - \mathbf{X}\beta\|^2 + \frac{\sigma^2}{\sigma_0^2}\|\beta\|^2
\end{aligned}
$$

What is logistic regression (逻辑回归)?

Logistic regression is a supervised problem, a two-class classification (**not** regression) task.

逻辑回归，不是回归模型，是分类模型！

- **Data**: $\{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{n}$
  $\boldsymbol{x}^{(i)} \in \mathbb{R}^{p\times 1}$ are features, $y^{(i)} \in \{0,1\}$ is label.

What is logistic regression (逻辑回归)?

Logistic regression is a supervised problem, a two-class classification (**not** regression) task.

逻辑回归，不是回归模型，是分类模型！

- **Data**: $\{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{n}$
  $\boldsymbol{x}^{(i)} \in \mathbb{R}^{p\times 1}$ are features, $y^{(i)} \in \{0, 1\}$ is label.
- **Model**: We could approach the classification problem ignoring the fact that y is discrete-valued, and use our old linear regression algorithm to try to predict y given $\boldsymbol{x}$.

$$
\begin{aligned}
h(\mathbf{x}) &= \sigma(f(\mathbf{x})) \\
&= \sigma(\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p) = \sigma(\beta^T \mathbf{x}),
\end{aligned} \tag{1}
$$

where $\sigma()$ is a *sigmoid function* (see Eq. (2)).
Parameters: $\beta = [\beta_0\ \beta_1 \ldots \beta_p]^T$

Sigmoid function $\sigma(z)$

- Sigmoid function(or logistic function):

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{2}$$

Sigmoid function $\sigma(z)$

- Sigmoid function(or logistic function):

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{2}$$

- The function $\sigma : \mathbb{R} \mapsto (0, 1)$; in logistic regression, $\beta^T \mathbf{x} \mapsto p$, where p indicates a probability.

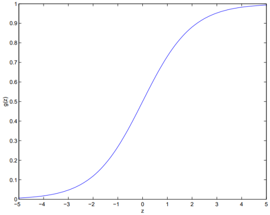


图 1: Sigmoid function (or logistic function)

Derivation of Sigmoid function, $\sigma'(z)$

- The sigmoid function: $\sigma(z) = \frac{1}{1+e^{-z}}$

Derivation of Sigmoid function, $\sigma'(z)$

- The sigmoid function: $\sigma(z) = \frac{1}{1+e^{-z}}$
- Now, let's derive the the derivative of Sigmoid function:

Derivation of Sigmoid function, $\sigma'(z)$

- The sigmoid function: $\sigma(z) = \frac{1}{1+e^{-z}}$
- Now, let's derive the the derivative of Sigmoid function:

$$
\begin{aligned}
\sigma'(z) &= \frac{d}{dz}\frac{1}{1+e^{-z}} \\
&= \frac{1}{(1+e^{-z})^2}\left(0+e^{-z}\right) \\
&= \frac{1}{(1+e^{-z})}\frac{e^{-z}}{(1+e^{-z})} \\
&= \frac{1}{(1+e^{-z})}\left(1-\frac{1}{(1+e^{-z})}\right) \\
&= \sigma(z)(1-\sigma(z))
\end{aligned}
\tag{3}
$$

Probability of $y = 0$, and $y = 1$

- Recall Bernoulli distribution: $Ber(x|\theta) = \theta^{x}(1-\theta)^{1-x}$

Probability of $y = 0$, and $y = 1$

- Recall Bernoulli distribution: $Ber(x|\theta) = \theta^x(1-\theta)^{1-x}$
- For logistic regression, given a sample $\mathbf{x}$, we can calculate probability of $y = 0$ and $y = 1$, respectively:

$$\text{Case 1:} \quad p_1 = \mathrm{P}(y = 1 \mid \mathbf{x}; \beta) = h_\beta(\mathbf{x}) = \frac{1}{1 + e^{-\beta^T \mathbf{x}}}$$

$$\text{Case 2:} \quad p_0 = \mathrm{P}(y = 0 \mid \mathbf{x}; \beta) = 1 - h_\beta(\mathbf{x}) = \frac{e^{-\beta^T \mathbf{x}}}{1 + e^{-\beta^T \mathbf{x}}}$$

Probability of $y = 0$, and $y = 1$

- Recall Bernoulli distribution: $Ber(x|\theta) = \theta^x(1-\theta)^{1-x}$
- For logistic regression, given a sample $\mathbf{x}$, we can calculate probability of $y = 0$ and $y = 1$, respectively:

$$\text{Case 1: } p_1 = \mathrm{P}(y = 1 \mid \mathbf{x}; \beta) = h_\beta(\mathbf{x}) = \frac{1}{1 + e^{-\beta^T \mathbf{x}}}$$

$$\text{Case 2: } p_0 = \mathrm{P}(y = 0 \mid \mathbf{x}; \beta) = 1 - h_\beta(\mathbf{x}) = \frac{e^{-\beta^T \mathbf{x}}}{1 + e^{-\beta^T \mathbf{x}}}$$

- We get the **likelihood** for logistic regression:

$$\begin{aligned} p(y \mid \mathbf{x}; \beta) &= p_0^{1-y} p_1^y \\ &= (1 - h_\beta(\mathbf{x}))^{1-y} (h_\beta(\mathbf{x}))^y \end{aligned} \tag{4}$$

Probability of $y = 0$, and $y = 1$

- Recall Bernoulli distribution: $Ber(x|\theta) = \theta^x(1-\theta)^{1-x}$
- For logistic regression, given a sample $\mathbf{x}$, we can calculate probability of $y = 0$ and $y = 1$, respectively:

$$\text{Case 1:} \quad p_1 = \mathrm{P}(y = 1 \mid \mathbf{x}; \beta) = h_\beta(\mathbf{x}) = \frac{1}{1 + e^{-\beta^T \mathbf{x}}}$$

$$\text{Case 2:} \quad p_0 = \mathrm{P}(y = 0 \mid \mathbf{x}; \beta) = 1 - h_\beta(\mathbf{x}) = \frac{e^{-\beta^T \mathbf{x}}}{1 + e^{-\beta^T \mathbf{x}}}$$

- We get the **likelihood** for logistic regression:

$$\begin{aligned} p(y \mid \mathbf{x}; \beta) &= p_0^{1-y} p_1^y \\ &= (1 - h_\beta(\mathbf{x}))^{1-y} (h_\beta(\mathbf{x}))^y \end{aligned} \tag{4}$$

- Q: Can you calculate the **log odds**: $LO \triangleq \log \frac{p(y=1|x)}{p(y=0|x)}$?

Interpret logistic regression

- The **likelihood** for logistic regression:

$$p(y \mid \mathbf{x}; \beta) = p_0^{1-y} p_1^{y} = (h_\beta(\mathbf{x}))^{y} (1 - h_\beta(\mathbf{x}))^{1-y}$$

Interpret logistic regression

- The **likelihood** for logistic regression:

$$p(y \mid \mathbf{x}; \beta) = p_0^{1-y} p_1^{y} = (h_\beta(\mathbf{x}))^y \left(1 - h_\beta(\mathbf{x})\right)^{1-y}$$

- Q: Can you calculate the **log odds**: $LO \triangleq \log \frac{p(y=1|x)}{p(y=0|x)}$?

Interpret logistic regression

- The **likelihood** for logistic regression:

$$p(y \mid \mathbf{x}; \beta) = p_0^{1-y} p_1^{y} = (h_\beta(\mathbf{x}))^y \,(1 - h_\beta(\mathbf{x}))^{1-y}$$

- Q: Can you calculate the **log odds**: $LO \triangleq \log \frac{p(y=1|x)}{p(y=0|x)}$?

$$\log \frac{p(y=1|x)}{p(y=0|x)} = \log \frac{\frac{1}{1+e^{-\beta^T \boldsymbol{x}}}}{\frac{e^{-\beta^T \boldsymbol{x}}}{1+e^{-\beta^T \boldsymbol{x}}}} = \log \frac{1}{e^{-\beta^T \boldsymbol{x}}} = \beta^T \boldsymbol{x} \qquad (5)$$

Interpret logistic regression

- The **likelihood** for logistic regression:

$$p(y \mid \mathbf{x}; \beta) = p_0^{1-y} p_1^{y} = (h_\beta(\mathbf{x}))^y \, (1 - h_\beta(\mathbf{x}))^{1-y}$$

- Q: Can you calculate the **log odds**: $LO \triangleq \log \frac{p(y=1|x)}{p(y=0|x)}$?

$$\log \frac{p(y=1|x)}{p(y=0|x)} = \log \frac{\frac{1}{1+e^{-\beta^T \boldsymbol{x}}}}{\frac{e^{-\beta^T \boldsymbol{x}}}{1+e^{-\beta^T \boldsymbol{x}}}} = \log \frac{1}{e^{-\beta^T \boldsymbol{x}}} = \beta^T \boldsymbol{x} \qquad (5)$$

- Example: The feature vector: x_1 is the times to miss homework; x_2 is the number of attendance in course. The goal is to predict the probability that you will fail the ML course (*i.e.*, $p(y=1|\boldsymbol{x}, \beta)$). We use logistic regression to model it, and we estimate the parameters as $\beta = [2, -1]^T$.

Interpret logistic regression

- The **likelihood** for logistic regression:

$$p(y \mid \mathbf{x}; \beta) = p_0^{1-y} p_1^{y} = (h_\beta(\mathbf{x}))^y \, (1 - h_\beta(\mathbf{x}))^{1-y}$$

- Q: Can you calculate the **log odds**: $LO \triangleq \log \frac{p(y=1|x)}{p(y=0|x)}$?

$$\log \frac{p(y=1|x)}{p(y=0|x)} = \log \frac{\frac{1}{1+e^{-\beta^T \boldsymbol{x}}}}{\frac{e^{-\beta^T \boldsymbol{x}}}{1+e^{-\beta^T \boldsymbol{x}}}} = \log \frac{1}{e^{-\beta^T \boldsymbol{x}}} = \beta^T \boldsymbol{x} \qquad (5)$$

- Example: The feature vector: x_1 is the times to miss homework; x_2 is the number of attendance in course. The goal is to predict the probability that you will fail the ML course (*i.e.*, $p(y=1|\boldsymbol{x}, \beta)$). We use logistic regression to model it, and we estimate the parameters as $\beta = [2, -1]^T$.
- Interpret: every time you miss a homework, the risk to fail the course increases by a factor of e^2.

Estimating parameters using MLE

- The entire **training data**: $\{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$
  Label: $y^{(i)} \in \{0, 1\}$
  Model: $h(\mathbf{x}) = \sigma(\beta^T \mathbf{x})$
  Parameters: $\beta = [\beta_1, \beta_2, \ldots, \beta_p]^T$

Estimating parameters using MLE

- The entire **training data**: $\{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$
  Label: $y^{(i)} \in \{0, 1\}$
  Model: $h(\mathbf{x}) = \sigma(\beta^T \mathbf{x})$
  Parameters: $\beta = [\beta_1, \beta_2, \ldots, \beta_p]^T$
- Q: How to estimate parameters β?

Estimating parameters using MLE

- The entire **training data**: $\{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$
  Label: $y^{(i)} \in \{0, 1\}$
  Model: $h(\mathbf{x}) = \sigma(\beta^T \mathbf{x})$
  Parameters: $\beta = [\beta_1, \beta_2, \ldots, \beta_p]^T$
- Q: How to estimate parameters β?
- Using **Maximum Likelihood Estimate** (MLE) to estimate β
  For the entire training data, the **likelihood** is:

$$\begin{aligned} p(y \mid \mathbf{X}; \beta) &= \prod_{i=1}^{n} p\left(y^{(i)} \mid \mathbf{x}^{(i)}; \beta\right) \\ &= \prod_{i=1}^{n} \left(h_\beta\left(\mathbf{x}^{(i)}\right)\right)^{y^{(i)}} \left(1 - h_\beta\left(\mathbf{x}^{(i)}\right)\right)^{1-y^{(i)}} \end{aligned} \tag{6}$$

Log-likelihood

- For the entire training data, the **likelihood** is:

$$p(y \mid \mathbf{X}; \beta) = \prod_{i=1}^{N} \left(h_\beta\left(\mathbf{x}^{(i)}\right)\right)^{y^{(i)}} \left(1 - h_\beta\left(\mathbf{x}^{(i)}\right)\right)^{1-y^{(i)}} \tag{7}$$

Log-likelihood

- For the entire training data, the **likelihood** is:

$$p(y \mid \mathbf{X}; \beta) = \prod_{i=1}^{N} \left(h_\beta\left(\mathbf{x}^{(i)}\right)\right)^{y^{(i)}} \left(1 - h_\beta\left(\mathbf{x}^{(i)}\right)\right)^{1-y^{(i)}} \tag{7}$$

- Let us derive the **log-likelihood**:

$$\begin{aligned} \log p(y \mid \mathbf{X}; \beta) = \sum_{i=1}^{N} & y^{(i)} \log h_\beta\left(\mathbf{x}^{(i)}\right) \\ & + \left(1 - y^{(i)}\right) \log\left(1 - h_\beta\left(\mathbf{x}^{(i)}\right)\right) \end{aligned} \tag{8}$$

Cross-entropy loss function

The definition of Cross entropy:

$$H(p,q) \triangleq -\sum_{x} p(x)\log(q(x))$$

Recall Eq. (8), the log-likelihood is

$$\log p(y|\mathbf{X};\beta) = \sum_{i=1}^{N} y^{(i)} \log h_{\beta}\left(\mathbf{x}^{(i)}\right) + \left(1 - y^{(i)}\right)\log\left(1 - h_{\beta}\left(\mathbf{x}^{(i)}\right)\right)$$

We notice that $H(p,q) = -\log p(y|\mathbf{X};\beta)$, if we set:
distribution p(x): $p(x=1) = y^{(i)}$, $p(x=0) = 1 - y^{(i)}$
distribution q(x): $q(x=1) = h_{\beta}(x^{(i)})$, $q(x=0) = 1 - h_{\beta}(x^{(i)})$
So, MLE is the same as minimizing the cross-entropy loss.

Estimating parameters $\hat{\beta}$

- Estimating parameters by minimizing the cross-entropy loss function, *i.e.*, $\mathcal{L}(\beta)$,

$$\begin{aligned}\hat{\beta} &= \arg\min_{\beta} \mathcal{L}(\beta) \\ &= \arg\min_{\beta} -\sum_{i=1}^{N} y^{(i)} \log h_{\beta}\left(\mathbf{x}^{(i)}\right) + \left(1 - y^{(i)}\right) \log\left(1 - h_{\beta}\left(\mathbf{x}^{(i)}\right)\right)\end{aligned} \tag{9}$$

Recall the Gradient Descent algorithm:

$$\beta \leftarrow \beta - \eta \nabla \mathcal{L}(\beta)$$

The key is to derive the gradient of cross-entropy loss, $\nabla \mathcal{L}(\beta)$.

Compute the derivation of log-likelihood

Let us start by working with just one training data, *e.g.*, $(\mathbf{x}, y)$:

$$
\begin{aligned}
&\nabla \mathcal{L}(\beta)|_{(\mathbf{x},y)} \\
&= -\left[y \frac{1}{\sigma\left(\beta^T \mathbf{x}\right)} - (1-y) \frac{1}{1-\sigma\left(\beta^T \mathbf{x}\right)}\right] \frac{\partial}{\partial \beta} \sigma\left(\beta^T \mathbf{x}\right) \\
&= -[\cdots] \sigma\left(\beta^T \mathbf{x}\right)\left(1-\sigma\left(\beta^T \mathbf{x}\right)\right) \frac{\partial(\beta^T \mathbf{x})}{\partial \beta} \\
&= -[\cdots] \sigma\left(\beta^T \mathbf{x}\right)\left(1-\sigma\left(\beta^T \mathbf{x}\right)\right) \mathbf{x} \\
&= -\left(y\left(1-\sigma\left(\beta^T \mathbf{x}\right)\right)-(1-y) \sigma\left(\beta^T \mathbf{x}\right)\right) \mathbf{x} \\
&= -\left(y-\sigma(\beta^T \mathbf{x})\right) \mathbf{x} \\
&= -\left(y-h_\beta(\mathbf{x})\right) \mathbf{x}
\end{aligned}
\tag{10}
$$

Algorithm for Logistic regression with cross-entropy loss

Given a data pair $(\mathbf{x}, y)$, the gradient of cross-entropy loss:

$$\nabla \mathcal{L}(\beta)|_{(\mathbf{x},y)} = -\left(y - h_\beta(\mathbf{x})\right)\mathbf{x}$$

The entire training data: $(\mathbf{X}, \mathbf{y})$

1. Initiate β

Algorithm for Logistic regression with cross-entropy loss

Given a data pair $(\mathbf{x}, y)$, the gradient of cross-entropy loss:

$$\nabla \mathcal{L}(\beta)|_{(\mathbf{x},y)} = -\left(y - h_\beta(\mathbf{x})\right)\mathbf{x}$$

The entire training data: $(\mathbf{X}, \mathbf{y})$

1. Initiate β
2. Calculate $h_\beta(\mathbf{x}^{(i)}) = \sigma\left(\beta^T \mathbf{x}^{(i)}\right)$ for for each $\mathbf{x}^{(i)}$

Algorithm for Logistic regression with cross-entropy loss

Given a data pair $(\mathbf{x}, y)$, the gradient of cross-entropy loss:

$$\nabla \mathcal{L}(\beta)|_{(\mathbf{x},y)} = -\left(y - h_\beta(\mathbf{x})\right)\mathbf{x}$$

The entire training data: $(\mathbf{X}, \mathbf{y})$

1. Initiate β
2. Calculate $h_\beta(\mathbf{x}^{(i)}) = \sigma\left(\beta^T \mathbf{x}^{(i)}\right)$ for for each $\mathbf{x}^{(i)}$
3. Calculate $\nabla \mathcal{L}(\beta)|_{(\mathbf{x}^{(i)},y^{(i)})} = -\left(y^{(i)} - h_\beta(\mathbf{x}^{(i)})\right)\mathbf{x}^{(i)}$ for each data pair $(\mathbf{x}^{(i)}, y^{(i)})$

Algorithm for Logistic regression with cross-entropy loss

Given a data pair $(\mathbf{x}, y)$, the gradient of cross-entropy loss:

$$\nabla\mathcal{L}(\beta)|_{(\mathbf{x},y)} = -\left(y - h_\beta(\mathbf{x})\right)\mathbf{x}$$

The entire training data: $(\mathbf{X}, \mathbf{y})$

1. Initiate β
2. Calculate $h_\beta(\mathbf{x}^{(i)}) = \sigma\left(\beta^T\mathbf{x}^{(i)}\right)$ for for each $\mathbf{x}^{(i)}$
3. Calculate $\nabla\mathcal{L}(\beta)|_{(\mathbf{x}^{(i)},y^{(i)})} = -\left(y^{(i)} - h_\beta(\mathbf{x}^{(i)})\right)\mathbf{x}^{(i)}$ for each data pair $(\mathbf{x}^{(i)}, y^{(i)})$
4. Calculate $\nabla\mathcal{L} = \sum_{i=1}^{i=n}\nabla\mathcal{L}(\beta)|_{(\mathbf{x}^{(i)};y^{(i)})}$

Algorithm for Logistic regression with cross-entropy loss

Given a data pair $(\mathbf{x}, y)$, the gradient of cross-entropy loss:

$$\nabla\mathcal{L}(\beta)|_{(\mathbf{x},y)} = -\left(y - h_\beta(\mathbf{x})\right)\mathbf{x}$$

The entire training data: $(\mathbf{X}, \mathbf{y})$

1. Initiate β
2. Calculate $h_\beta(\mathbf{x}^{(i)}) = \sigma\left(\beta^T\mathbf{x}^{(i)}\right)$ for for each $\mathbf{x}^{(i)}$
3. Calculate $\nabla\mathcal{L}(\beta)|_{(\mathbf{x}^{(i)},y^{(i)})} = -\left(y^{(i)} - h_\beta(\mathbf{x}^{(i)})\right)\mathbf{x}^{(i)}$ for each data pair $(\mathbf{x}^{(i)}, y^{(i)})$
4. Calculate $\nabla\mathcal{L} = \sum_{i=1}^{i=n}\nabla\mathcal{L}(\beta)|_{(\mathbf{x}^{(i)};y^{(i)})}$
5. Update β: $\quad \beta \leftarrow \beta + \eta\nabla\mathcal{L}$

Logistic regression with MSE loss

- **Training data**: $\{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{n}$

Logistic regression with MSE loss

- **Training data**: $\{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{n}$
- **Label**: $y^{(i)} \in \{0, 1\}$

Logistic regression with MSE loss

- **Training data**: $\{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{n}$
- **Label**: $y^{(i)} \in \{0, 1\}$
- **Model**: $h(\mathbf{x}) = \sigma(\beta^T \mathbf{x})$

Logistic regression with MSE loss

- **Training data**: $\{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{n}$
- **Label**: $y^{(i)} \in \{0, 1\}$
- **Model**: $h(\mathbf{x}) = \sigma(\beta^T \mathbf{x})$
- **Parameters**: $\beta = [\beta_0, \beta_1, \ldots, \beta_p]^T$

Logistic regression with MSE loss

- **Training data**: $\{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{n}$
- **Label**: $y^{(i)} \in \{0, 1\}$
- **Model**: $h(\mathbf{x}) = \sigma(\beta^T \mathbf{x})$
- **Parameters**: $\beta = [\beta_0, \beta_1, \ldots, \beta_p]^T$
- The mean-squared-error (MSE) loss for logistic regression:

$$\mathcal{L}(\beta) = \sum_{i=1}^{n} \left(\sigma(\beta^T \mathbf{x}^{(i)}) - y^{(i)} \right)^2 \tag{11}$$

Logistic regression with MSE loss

- **Training data**: $\{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{n}$
- **Label**: $y^{(i)} \in \{0, 1\}$
- **Model**: $h(\mathbf{x}) = \sigma(\beta^T \mathbf{x})$
- **Parameters**: $\beta = [\beta_0, \beta_1, \ldots, \beta_p]^T$
- The mean-squared-error (MSE) loss for logistic regression:

$$\mathcal{L}(\beta) = \sum_{i=1}^{n} \left(\sigma(\beta^T \mathbf{x}^{(i)}) - y^{(i)} \right)^2 \tag{11}$$

- Given a data pair $(\mathbf{x}, y)$, the **gradient** of MSE loss:

Logistic regression with MSE loss

- **Training data**: $\{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{n}$
- **Label**: $y^{(i)} \in \{0, 1\}$
- **Model**: $h(\mathbf{x}) = \sigma(\beta^T \mathbf{x})$
- **Parameters**: $\beta = [\beta_0, \beta_1, \ldots, \beta_p]^T$
- The mean-squared-error (MSE) loss for logistic regression:

$$\mathcal{L}(\beta) = \sum_{i=1}^{n} \left(\sigma(\beta^T \mathbf{x}^{(i)}) - y^{(i)}\right)^2 \tag{11}$$

- Given a data pair $(\mathbf{x}, y)$, the **gradient** of MSE loss:

$$\begin{aligned}\frac{\partial \left(\sigma(\beta^T \mathbf{x}) - y\right)^2}{\partial \beta} &= 2\left(\sigma(\beta^T \mathbf{x}) - y\right) \frac{\partial \sigma(\beta^T \mathbf{x})}{\partial \beta} \\ &= 2\left(\sigma(\beta^T \mathbf{x}) - y\right) \sigma(\beta^T \mathbf{x}) \left(1 - \sigma(\beta^T \mathbf{x})\right) \mathbf{x}\end{aligned}$$

Intuition on the gradient of MSE loss

Given a data pair $(\mathbf{x}, y)$, the **gradient** of MSE loss:

$$\nabla\mathcal{L}(\beta) = \frac{\partial\left(\sigma(\beta^T\mathbf{x}) - y\right)^2}{\partial\beta} = 2\left(\sigma(\beta^T\mathbf{x}) - y\right)\sigma(\beta^T\mathbf{x})\left(1 - \sigma(\beta^T\mathbf{x})\right)\mathbf{x}$$

Intuitions on the gradient of MSE loss:

- For the case $y = 1$

$$\text{If } \sigma(\beta^T\mathbf{x}) \approx 1 \text{ (close to target)} \longrightarrow \nabla\mathcal{L}(\beta) \approx 0$$
$$\text{If } \sigma(\beta^T\mathbf{x}) \approx 0 \text{ (far from target)} \longrightarrow \nabla\mathcal{L}(\beta) \approx 0$$

Intuition on the gradient of MSE loss

Given a data pair $(\mathbf{x}, y)$, the **gradient** of MSE loss:

$$\nabla\mathcal{L}(\beta) = \frac{\partial\left(\sigma(\beta^T\mathbf{x}) - y\right)^2}{\partial\beta} = 2\left(\sigma(\beta^T\mathbf{x}) - y\right)\sigma(\beta^T\mathbf{x})\left(1 - \sigma(\beta^T\mathbf{x})\right)\mathbf{x}$$

Intuitions on the gradient of MSE loss:

- For the case $y = 1$

  If $\sigma(\beta^T\mathbf{x}) \approx 1$ (close to target) $\longrightarrow \nabla\mathcal{L}(\beta) \approx 0$
  If $\sigma(\beta^T\mathbf{x}) \approx 0$ (far from target) $\longrightarrow \nabla\mathcal{L}(\beta) \approx 0$

- For the case $y = 0$

  If $\sigma(\beta^T\mathbf{x}) \approx 1$ (far from target) $\longrightarrow \nabla\mathcal{L}(\beta) \approx 0$
  If $\sigma(\beta^T\mathbf{x}) \approx 0$ (close to target) $\longrightarrow \nabla\mathcal{L}(\beta) \approx 0$

Intuition on the gradient of MSE loss

Given a data pair $(\mathbf{x}, y)$, the **gradient** of MSE loss:

$$\nabla \mathcal{L}(\beta) = \frac{\partial \left(\sigma(\beta^T \mathbf{x}) - y\right)^2}{\partial \beta} = 2\left(\sigma(\beta^T \mathbf{x}) - y\right) \sigma(\beta^T \mathbf{x}) \left(1 - \sigma(\beta^T \mathbf{x})\right) \mathbf{x}$$

Intuitions on the gradient of MSE loss:

- For the case $y = 1$

  If $\sigma(\beta^T \mathbf{x}) \approx 1$ (close to target) $\longrightarrow \nabla \mathcal{L}(\beta) \approx 0$
  If $\sigma(\beta^T \mathbf{x}) \approx 0$ (far from target) $\longrightarrow \nabla \mathcal{L}(\beta) \approx 0$

- For the case $y = 0$

  If $\sigma(\beta^T \mathbf{x}) \approx 1$ (far from target) $\longrightarrow \nabla \mathcal{L}(\beta) \approx 0$
  If $\sigma(\beta^T \mathbf{x}) \approx 0$ (close to target) $\longrightarrow \nabla \mathcal{L}(\beta) \approx 0$

- It seems the gradient of MSE loss does not guide the learning in any case.

Visualization of different loss functions

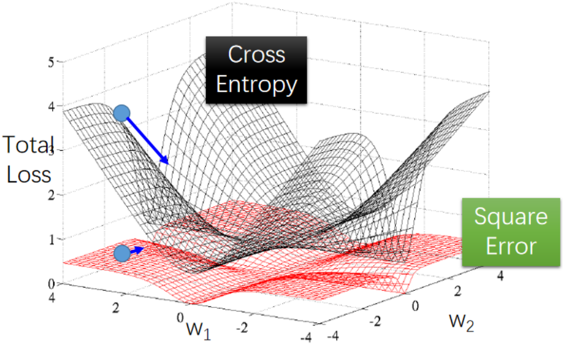


图 2: loss visualization

When we use MSE loss, it is very slow to update parameters.

Advantages of logistic regression

- Q: Please take some examples to use logistic regression.

Advantages of logistic regression

- Q: Please take some examples to use logistic regression.
  1. **Model the decision making**
     $y = 1$ for attending the course; $y = 0$ for skipping the course.
     $\boldsymbol{x}$ is the feature vector (*e.g.*, interest, mood, attraction, ...)
     β is weight of each feature (effect of features on a decision).

Advantages of logistic regression

- Q: Please take some examples to use logistic regression.
  1. **Model the decision making**
     $y = 1$ for attending the course; $y = 0$ for skipping the course.
     $\boldsymbol{x}$ is the feature vector (*e.g.*, interest, mood, attraction, ...)
     β is weight of each feature (effect of features on a decision).
  2. **Model the result of a TUMOR test**
     $y = 1$ for positive (阳性); $y = 0$ for negative (阴性).
     $\boldsymbol{x} = [x_1, x_2, \dots]^T$ are features (*e.g.*, smoking, gene, ...)
     $\beta = [\beta_1, \beta_2, \dots]^T$ are weights

Advantages of logistic regression

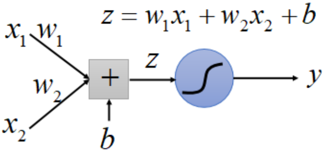
- Q: Please take some examples to use logistic regression.
  1. **Model the decision making**
     $y = 1$ for attending the course; $y = 0$ for skipping the course.
     $\boldsymbol{x}$ is the feature vector (*e.g.*, interest, mood, attraction, ...)
     β is weight of each feature (effect of features on a decision).
  2. **Model the result of a TUMOR test**
     $y = 1$ for positive (阳性); $y = 0$ for negative (阴性).
     $\boldsymbol{x} = [x_1, x_2, \dots]^T$ are features (*e.g.*, smoking, gene, ...)
     $\beta = [\beta_1, \beta_2, \dots]^T$ are weights
- Recall Eq. (5), the log odd $LO = \beta^T \boldsymbol{x}$.
  Interpret our TUMOR model: increase x_1, the risk of being **positive** increases by a factor of e^{β_1}, relative to being **negative**.

Advantages of logistic regression

- Q: Please take some examples to use logistic regression.
  1. **Model the decision making**
     $y = 1$ for attending the course; $y = 0$ for skipping the course.
     $\boldsymbol{x}$ is the feature vector (*e.g.*, interest, mood, attraction, ...)
     β is weight of each feature (effect of features on a decision).
  2. **Model the result of a TUMOR test**
     $y = 1$ for positive (阳性); $y = 0$ for negative (阴性).
     $\boldsymbol{x} = [x_1, x_2, \ldots]^T$ are features (*e.g.*, smoking, gene, ...)
     $\beta = [\beta_1, \beta_2, \ldots]^T$ are weights
- Recall Eq. (5), the log odd $LO = \beta^T \boldsymbol{x}$.
  Interpret our TUMOR model: increase x_1, the risk of being **positive** increases by a factor of e^{β_1}, relative to being **negative**.
- **Advantages** of LR: simple, easy to train, interpretable, ...

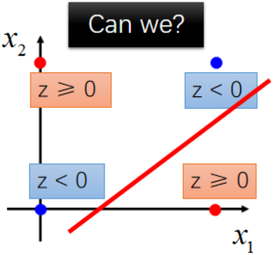
Limitation of logistic regression

The two classes are not separable with a sigmoid function on a linear transformation $\boldsymbol{w}^T\boldsymbol{x}$.



$$\begin{cases} Class1 & y \geq 0.5 \quad (z \geq 0) \\ Class2 & y < 0.5 \quad (z < 0) \end{cases}$$

Input Feature		Label
x_1	x_2	
0	0	Class 2
0	1	Class 1
1	0	Class 1
1	1	Class 2



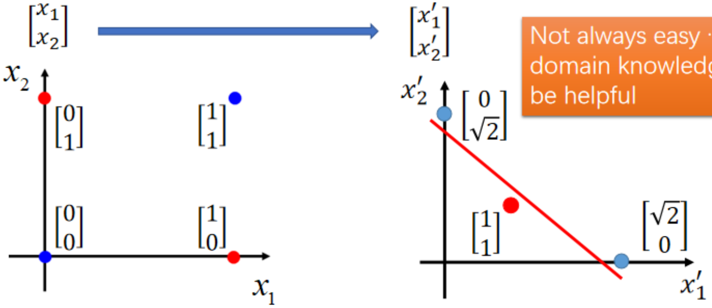
Courtesy of HUNG-YI LEE

Possible solution: feature transformation

- ***Feature transformation***

x_1': distance to $\begin{bmatrix}0\\0\end{bmatrix}$

x_2': distance to $\begin{bmatrix}1\\1\end{bmatrix}$

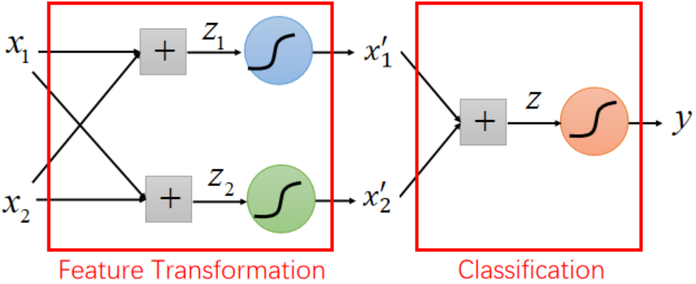


Not always easy … domain knowledge can be helpful

Courtesy of HUNG-YI LEE

Cascading logistic regression models

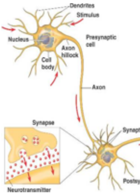
- Cascading logistic regression models

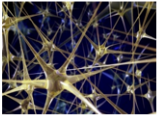


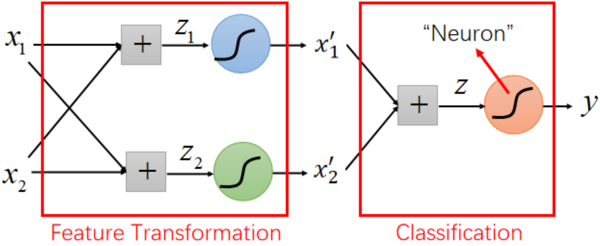
It becomes a neural network!

Deep Learning!

All the parameters of the logistic regressions are jointly learned.







Neural Network

Cross-entropy loss & its gradient

Given a single training data $(\mathbf{x}, y)$, the cross-entropy loss is:

$$\mathcal{L}(\beta) = H(y, h_\beta(x)) = -[y \log h_\beta(x) + (1-y)(1 - \log h_\beta(x))]$$

The gradient of cross-entropy loss is:

$$\begin{aligned}
\nabla\mathcal{L}(\beta) &= -\left[y\frac{1}{\sigma\left(\beta^T\mathbf{x}\right)} - (1-y)\frac{1}{1-\sigma\left(\beta^T\mathbf{x}\right)}\right]\frac{\partial}{\partial\beta}\sigma\left(\beta^T\mathbf{x}\right) \\
&= -\left[\cdots\right]\sigma\left(\beta^T\mathbf{x}\right)\left(1-\sigma\left(\beta^T\mathbf{x}\right)\right)\frac{\partial(\beta^T\mathbf{x})}{\partial\beta} \\
&= -\left[\cdots\right]\sigma\left(\beta^T\mathbf{x}\right)\left(1-\sigma\left(\beta^T\mathbf{x}\right)\right)\mathbf{x} \\
&= -\left(y\left(1-\sigma\left(\beta^T\mathbf{x}\right)\right) - (1-y)\sigma\left(\beta^T\mathbf{x}\right)\right)\mathbf{x} \\
&= -\left(y - h_\beta(\mathbf{x})\right)\mathbf{x}
\end{aligned}$$

GD for logistic regression

Model: $y = h_\beta(\mathbf{x}) = \sigma\left(\beta^T \mathbf{x}\right)$
Gradient of cross-entropy loss: $\nabla \mathcal{L}(\beta) = -\left(y - h_\beta(\mathbf{x})\right)\mathbf{x}$

The entire training data: $(\mathbf{X}, \mathbf{y})$

1. Initiate β
2. Calculate $h_\beta(\mathbf{x}^{(i)}) = \sigma\left(\beta^T \mathbf{x}^{(i)}\right)$ for for each $\mathbf{x}^{(i)}$
3. Calculate $\nabla \mathcal{L}(\beta)|_{(\mathbf{x}^{(i)}, y^{(i)})} = -\left(y^{(i)} - h_\beta(\mathbf{x}^{(i)})\right)\mathbf{x}^{(i)}$ for each data pair $(\mathbf{x}^{(i)}, y^{(i)})$
4. Calculate $\nabla \mathcal{L} = \sum_{i=1}^{i=n} \nabla \mathcal{L}(\beta)|_{(\mathbf{x}^{(i)}; y^{(i)})}$
5. Update β: $\quad \beta \leftarrow \beta + \eta \nabla \mathcal{L}$

For logistic regression, MLE = minimizing cross-entropy loss