Classification

1.求偏导数 $g_\theta(y, \hat{y})$

2.gradient-based update

$\theta^{(t+1)} = \theta^{(t)} - \lambda g_{\theta^{(t)}}(y, \hat{y})$ minimise

$\theta^{(t+1)} = \theta^{(t)} + \lambda g_{\theta^{(t)}}(y, \hat{y})$ maximum

3. gradient optimisation stops = min of loss function

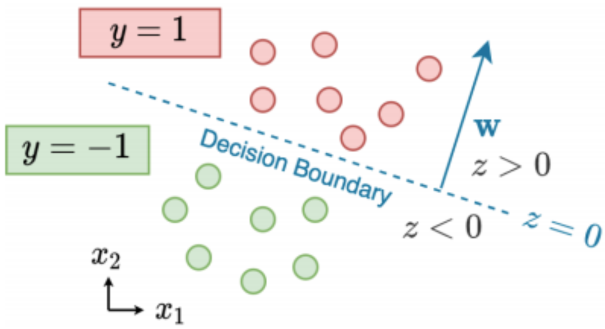
$$g_\theta(y, \hat{y}) = \left. \frac{\partial L_\theta(y, \hat{y})}{\partial \theta} \right|_{\theta = \hat{\theta}} = 0$$

Linear Classifier

Linear regression：z = $h_\theta(x) = \mathbf{w}^t\mathbf{x} + w_0 = w_1x_1 + w_2x_2 + \cdots + w_px_p + w_0$

Perceptrons 感知器

$$\hat{y} = \begin{cases} -1, & z < 0, \\ +1, & z > 0. \end{cases} \qquad \hat{y} = \operatorname{sign}(z)$$



步骤:
1. Compute z = wtx + w0,
2. If z > 0, set y = 1, else set y = −1, then update. Roll back to step1

Loss function

$$L_\mathbf{w}(y, \hat{y}) = \sum_{i=1}^{n} l_\mathbf{w}(y_i, \hat{y}_i) = \sum_{i \in \mathcal{M}} -y_i \cdot (\mathbf{w}^t\mathbf{x}_i + w_0)$$

Update in SGD

Differentiate the loss function with respect to each parameter:

LF 求导

$$\frac{\partial L_\mathbf{w}(y, \hat{y})}{\partial w_j} = \sum_{i \in \mathcal{M}} -y_i \cdot x_{ij},$$

$$\frac{\partial L_\mathbf{w}(y, \hat{y})}{\partial w_0} = \sum_{i \in \mathcal{M}} -y_i,$$

The general SGD update is given by:

$$w_j \leftarrow w_j - \lambda \frac{\partial l(\mathbf{w})}{\partial w_j}$$

This leads to updates of the form:

$$w_j^{(t+1)} = w_j^{(t)} + \lambda_j y_i x_{ij} \mathbb{I}(y_i \neq \hat{y}_i),$$
$$w_0^{(t+1)} = w_0^{(t)} + \lambda_j y_i \mathbb{I}(y_i \neq \hat{y}_i),$$

Thus,

$$w_j \leftarrow w_j + \lambda[y_i - h(\mathbf{w}, \mathbf{x}_i)]x_i^j$$

If we set the learning rate $\lambda_j = 1$ then:

$$
\begin{aligned}
w_j^{(t+1)} &= w_j^{(t)} + x_{ij}, \;\; y_i = 1, \hat{y}_i = -1, \\
w_j^{(t+1)} &= w_j^{(t)} - x_{ij}, \;\; y_i = -1, \hat{y}_i = 1, \\
w_0^{(t+1)} &= w_0^{(t)} + 1, \;\; y_i = 1, \hat{y}_i = -1, \\
w_0^{(t+1)} &= w_0^{(t)} - 1, \;\; y_i = -1, \hat{y}_i = 1.
\end{aligned}
$$

This implies across all inputs:

$$
\begin{aligned}
\mathbf{w}^{(t+1)} &= \mathbf{w}^{(t)} + \mathbf{x}_i, \;\; y_i = 1, \hat{y}_i = -1, \\
\mathbf{w}^{(t+1)} &= \mathbf{w}^{(t)} - \mathbf{x}_i, \;\; y_i = -1, \hat{y}_i = 1,
\end{aligned}
$$

Decision boundary

$$0 = \mathbf{w}^t\mathbf{x} + w_0.$$

0就是决策边界

For two inputs:

$$0 = w_1x_1 + w_2x_2 + w_0$$

which we can rearrange to give

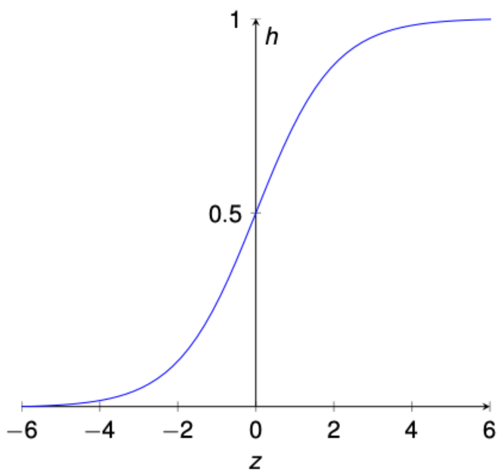
$$x_2 = -\frac{1}{w_2}(w_1x_1 + w_0) = -\left(\frac{w_1}{w_2}\right)x_1 + -\frac{w_0}{w_2}$$

Logistic Regression

1. Compute z = wtx + w0,
2. Apply the logistic function to z to get h = Logistic(z), 也叫$sigmoid$
3. Generate a random number between 0 and 1 uniformly, r ~ Uniform(0, 1)
4. If r < h, set y = 1, else set y = 0.

The **logistic function** maps the real line to (0, 1] via the transformation:

$$h = \frac{1}{1 + \exp(-z)}$$



Note that as:

- $z \to \infty$: $h \to 1$,
- $z \to -\infty$: $h \to 0$,
- $z = 0 : h = 0.5$

h>=0.5 y=1
h<0.5 y=0

h=0.5 是角色边界(d)

MLE 概率:

$$p(y_i|\mathbf{x}_i, \mathbf{w}) = h(\mathbf{x}_i, \mathbf{w})^{y_i}(1 - h(\mathbf{x}_i, \mathbf{w}))^{(1-y_i)}$$

$$
\begin{cases}
y_i = 1 : & p(y_i = 1|\mathbf{x}_i, \mathbf{w}) = h_i^1 \cancel{(1 - h_i)^{(1-1)}} \quad \text{=> h} \\
y_i = 0 :: & p(y_i = 0|\mathbf{x}_i, \mathbf{w}) = \cancel{h_i^0}(1 - h_i)^{(1-0)} \quad \text{=> 1-h}
\end{cases}
$$

$$
\begin{aligned}
p(y_1 = 0, y_2 = 0, y_3 = 1, y_4 = 1) &= (1 - p(y_1 = 1))(1 - p(y_2 = 1)) \\
&\quad \times p(y_3 = 1)p(y_4 = 1), \\
&= \frac{\exp(-z_1)}{1 + \exp(-z_1)} \times \frac{\exp(-z_2)}{1 + \exp(-z_2)} \\
&\quad \times \frac{1}{1 + \exp(-z_3)} \times \frac{1}{1 + \exp(-z_4)}
\end{aligned}
$$

Loss function: $l(\mathbf{w}) = -\frac{1}{n} \log L(\mathbf{w}|D) = -\frac{1}{n} \sum_{i=1}^{n} [y_i \log \underbrace{h(\mathbf{w}, \mathbf{x}_i)}_{h} + (1 - y_i) \log(1 - \underbrace{h(\mathbf{w}, \mathbf{x}_i)}_{h})]$

decision boundaries

$$
d = \frac{1}{1 + \exp(-[\mathbf{w}^t \mathbf{x} + w_0])},
$$

$$
\Rightarrow \mathbf{w}^t \mathbf{x} + w_0 = \log\left(\frac{d}{1 - d}\right)
$$

$$
x_2 = \frac{1}{w_2}\left[\log\left(\frac{d}{1 - d}\right) - w_0 - w_1 x_1\right]
$$

SGD update

LF求导：
$$
\begin{aligned}
\frac{\partial l(\mathbf{w})}{\partial w_j} &= -\left[y_i \frac{1}{h(\mathbf{w}, \mathbf{x}_i)} - (1 - y_i)\frac{1}{1 - h(\mathbf{w}, \mathbf{x}_i)}\right] h(\mathbf{w}, \mathbf{x}_i)(1 - h(\mathbf{w}, \mathbf{x}_i))x_i^j, \\
&= -[y_i(1 - h(\mathbf{w}, \mathbf{x}_i)) + (1 - y_i)h(\mathbf{w}, \mathbf{x}_i)]x_i^j, \\
&= -[y_i - h(\mathbf{w}, \mathbf{x}_i)]x_i^j
\end{aligned}
$$

The general SGD update is given by:

$$
w_j \leftarrow w_j - \lambda \frac{\partial l(\mathbf{w})}{\partial w_j}
$$

Thus,

$$
w_j \leftarrow w_j + \lambda[y_i - h(\mathbf{w}, \mathbf{x}_i)]x_i^j
$$

Therefore $y_i - h(\mathbf{w}, \mathbf{x}_i)$ is the difference between our prediction and the actual output.

- If $y_i = 1$ and $h(\mathbf{w}, \mathbf{x}_i) \approx 1$ then $y_i - h(\mathbf{w}, \mathbf{x}_i) \approx 0$ so there is no update,
- If $y_i = 0$ and $h(\mathbf{w}, \mathbf{x}_i) \approx 1$ then $y_i - h(\mathbf{w}, \mathbf{x}_i) \approx 1$ so there is an update in the direction of $\mathbf{x}_i$.

感知机： 减少预测的分布与真实分布的误差函数。

逻辑回归： 是预测每个类别的概率，取概率最大。并不需要预测结果是A是B，而只需要预测他们分别为A或者为B的概率，取概率大者

置信区间：a = 样本均值 - z标准误差

b = 样本均值 + z标准误差

$$SE = \frac{s(\text{样本标准差})}{\sqrt{n}}$$

Loss function 理解：

损失函数是用来评价模型的预测值 和真实值 的不一致程度，通常使用 来表示。该函数是一个非负实值函数，值越小，则表示针对训练数据的模型的性能越好。
损失函数用来评价预测值和真实值间的关系

对loss function求导得到方向斜率再乘λ步长得到SGD下降的距离

原版loss function = $$\min_{\theta} \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

求导：

$$\frac{\partial J}{\partial \theta 0} = \frac{1}{m} \sum_{i=1}^{m} (h\theta(x^{(i)}) - y^{(i)})$$

$$\frac{\partial J}{\partial \theta 1} = \frac{1}{m} \sum_{i=1}^{m} (h\theta(x^{(i)}) - y^{(i)}) x_i^{(i)}$$

下面列了一些比较经典的机器学习算法的随机梯度：

Loss	Stochastic gradient algorithm
Adaline (Widrow and Hoff, 1960) $Q_{\text{adaline}} = \frac{1}{2}\left(y - w^\top \Phi(x)\right)^2$ $\Phi(x) \in \mathbb{R}^d,\ y = \pm 1$	$w \leftarrow w + \gamma_t \left(y_t - w^\top \Phi(x_t)\right) \Phi(x_t)$
Perceptron (Rosenblatt, 1957) $Q_{\text{perceptron}} = \max\{0, -y\, w^\top \Phi(x)\}$ $\Phi(x) \in \mathbb{R}^d,\ y = \pm 1$	$w \leftarrow w + \gamma_t \begin{cases} y_t\, \Phi(x_t) & \text{if } y_t\, w^\top \Phi(x_t) \le 0 \\ 0 & \text{otherwise} \end{cases}$
K-Means (MacQueen, 1967) $Q_{\text{kmeans}} = \min_k \frac{1}{2}(z - w_k)^2$ $z \in \mathbb{R}^d,\ w_1 \dots w_k \in \mathbb{R}^d$ $n_1 \dots n_k \in \mathbb{N}$, initially 0	$k^* = \arg\min_k (z_t - w_k)^2$ $n_{k^*} \leftarrow n_{k^*} + 1$ $w_{k^*} \leftarrow w_{k^*} + \frac{1}{n_{k^*}}(z_t - w_{k^*})$
SVM (Cortes and Vapnik, 1995) $Q_{\text{svm}} = \lambda w^2 + \max\{0, 1 - y\, w^\top \Phi(x)\}$ $\Phi(x) \in \mathbb{R}^d,\ y = \pm 1,\ \lambda > 0$	$w \leftarrow w - \gamma_t \begin{cases} \lambda w & \text{if } y_t\, w^\top \Phi(x_t) > 1, \\ \lambda w - y_t\, \Phi(x_t) & \text{otherwise.} \end{cases}$
Lasso (Tibshirani, 1996) $Q_{\text{lasso}} = \lambda \lvert w \rvert_1 + \frac{1}{2}\left(y - w^\top \Phi(x)\right)^2$ $w = (u_1 - v_1, \dots, u_d - v_d)$ $\Phi(x) \in \mathbb{R}^d,\ y \in \mathbb{R},\ \lambda > 0$	$u_i \leftarrow \left[u_i - \gamma_t\left(\lambda - (y_t - w^\top \Phi(x_t))\Phi_i(x_t)\right)\right]_+$ $v_i \leftarrow \left[v_i - \gamma_t\left(\lambda + (y_t - w_t^\top \Phi(x_t))\Phi_i(x_t)\right)\right]_+$ with notation $[x]_+ = \max\{0, x\}$.