《逻辑回归》

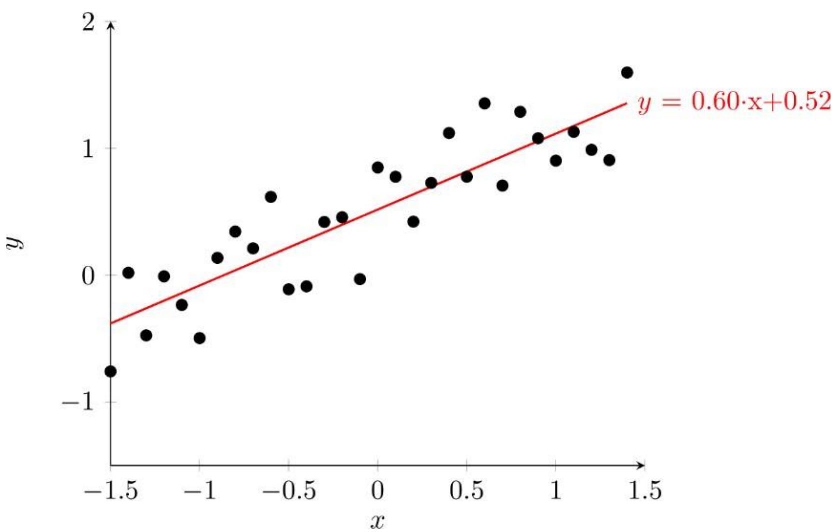
崔志勇
交通科学与工程学院
2024年5月

基础

线性模型（Linear Model）

- 线性模型是通过样本特征的线性组合来进行预测的模型

$$f(\mathbf{x};\mathbf{w},b)=\mathbf{w}^{\mathrm{T}}\mathbf{x}+b$$



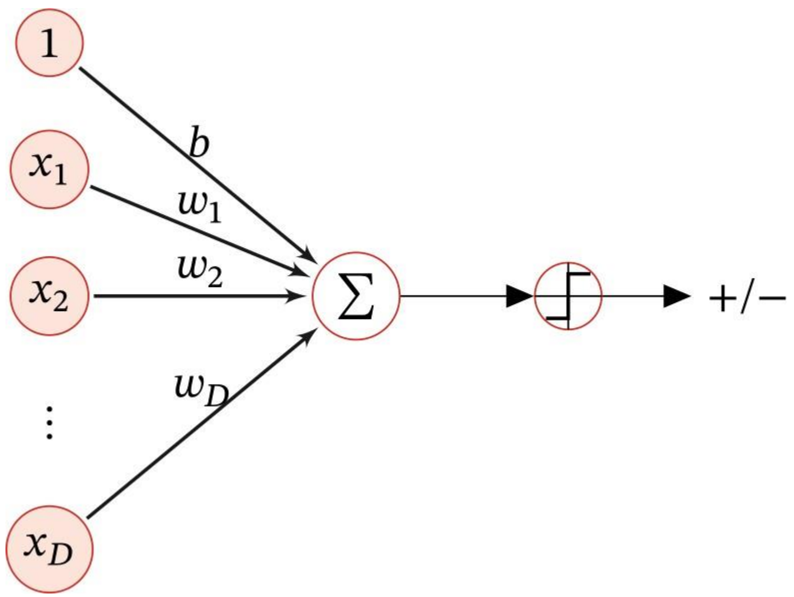
- 分类问题中，由于输出目标 y 是 一些离散的标签，而 f 值域为实数，因此无法直接进行预测，需要引入一个非线性的决策函数（Decision Function）$g(\cdot)$来预测输出目标

$$y=g(f(\boldsymbol{x};\boldsymbol{w})),$$

线性模型（Linear Model）

二分类问题中， $g(\cdot)$ **可以是符号函数（Sign Function）**

$$
\begin{aligned}
g\big(f(\boldsymbol{x};\boldsymbol{w})\big) &= \operatorname{sgn}\big(f(\boldsymbol{x};\boldsymbol{w})\big) \\
&\triangleq \begin{cases} +1 & \text{if} \quad f(\boldsymbol{x};\boldsymbol{w}) > 0, \\ -1 & \text{if} \quad f(\boldsymbol{x};\boldsymbol{w}) < 0. \end{cases}
\end{aligned}
$$



损失函数？

Logistic 回归

二分类问题中， $g(\cdot)$ 可以是符号函数（Sign Function)

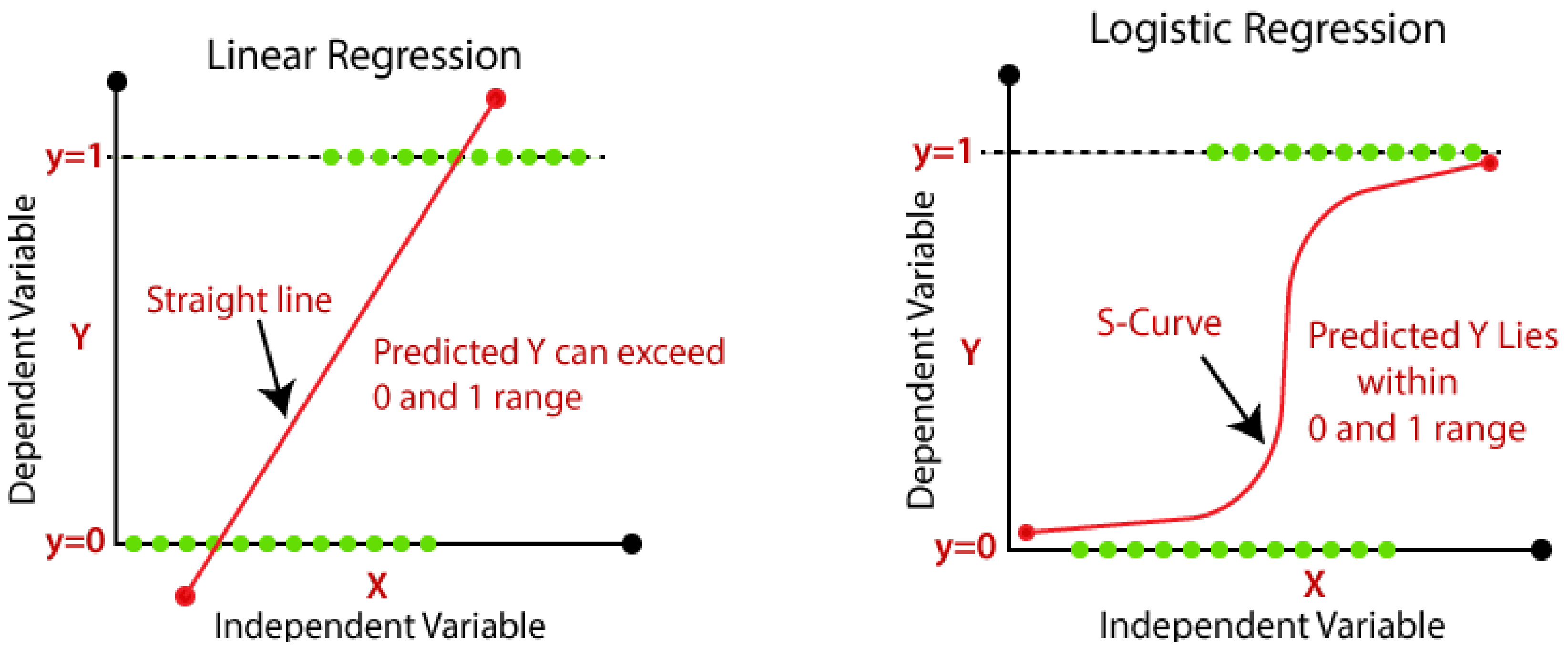
➢ 将分类问题看作条件概率估计问题

- 为了解决连续的线性函数不适合进行分类的问题，引入非线性函数 g 来预测类别标签的条件概率 $p(y = c|x)$。
- 以二分类为例， $p(y=1|\mathbf{x}) = g(f(\mathbf{x};\mathbf{w}))$
- 函数 f： 线性函数
- 函数 g：把线性函数的值域从实数区间“挤压”到了(0,1)之间，可以用来表示概率。

如何构建函数 g ?

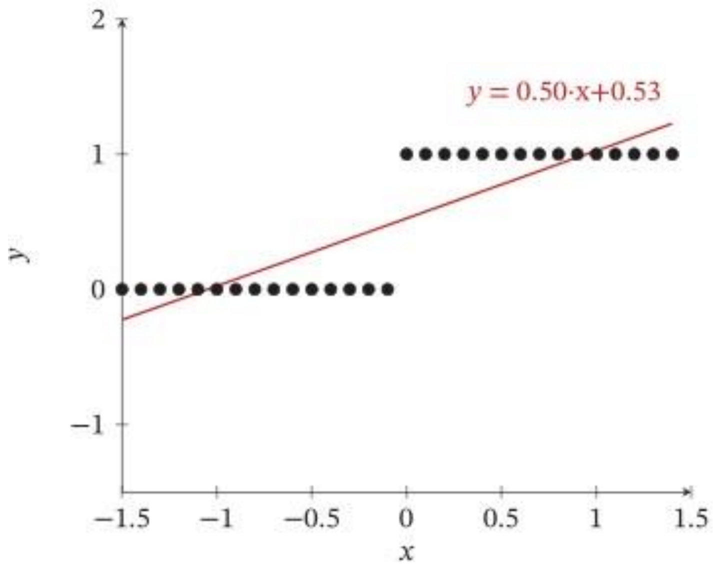
Logistic 回归

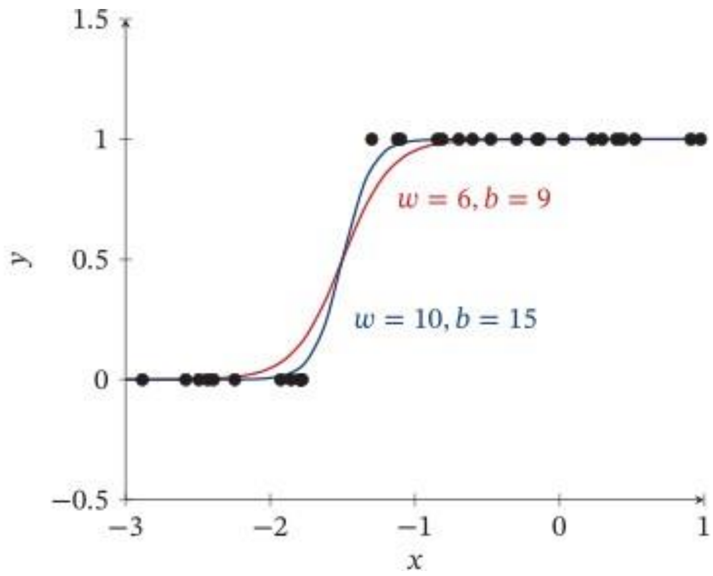
我们可以使用线性回归来对数据进行分类吗?



Logistic 回归

➢ **使用线性回归和Logistic回归来解决一维数据的二分类问题的示例。**



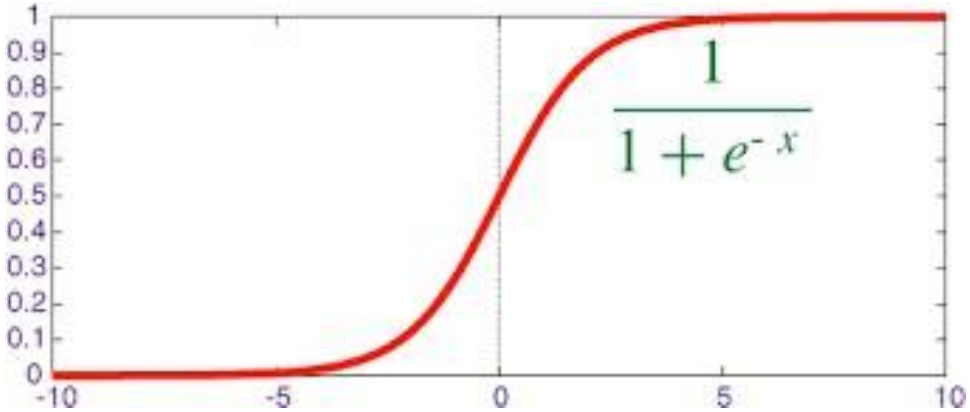


Logistic 回归

➢ Logistic 函数

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

除了logisitc，还有什么函数$f: R \rightarrow (0,1)$?



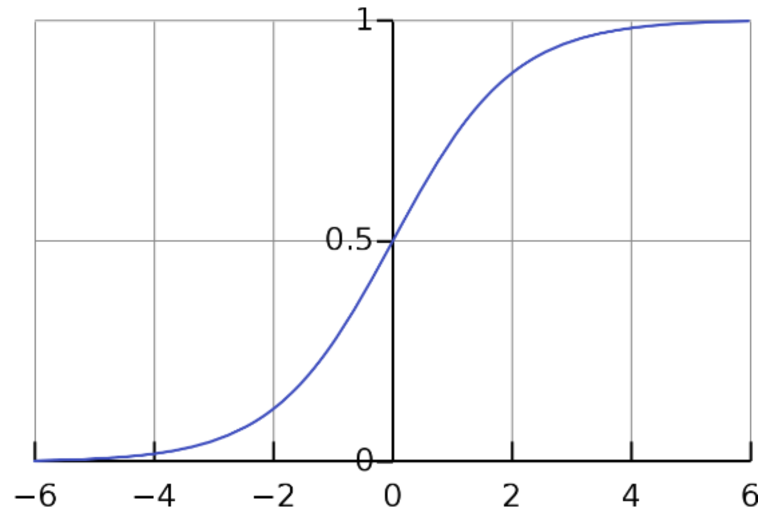
➢ Logistic 回归

$$p(y = 1|\boldsymbol{x}) = \sigma(\boldsymbol{w}^{\top}\boldsymbol{x})$$
$$\triangleq \frac{1}{1 + \exp(-\boldsymbol{w}^{\top}\boldsymbol{x})}$$

Logistic 回归

➢Logistic 函数

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



➢Logistic 回归模型

$$p(y = 1|\mathbf{x}) = \sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x}) \triangleq \frac{1}{1 + \exp(-\mathbf{w}^{\mathrm{T}}\mathbf{x})}$$

➢学习准则：交叉熵

$$\mathcal{R}(\mathbf{w}) = -\frac{1}{N}\sum_{n=1}^{N}\left(y^{(n)}\log \hat{y}^{(n)} + (1 - y^{(n)})\log(1 - \hat{y}^{(n)})\right)$$

➢优化算法： 梯度下降

$$\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t + \alpha\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}^{(n)}\left(y^{(n)} - \hat{y}_{\boldsymbol{w}_t}^{(n)}\right)$$

Logistic 回归

分类模型：目标变量是分类变量（离散值）。

回归模型：目标变量是连续性数值变量。

- 假设x是一个m维二元向量，即$x \in \{0,1\}^m$，可形象化的**将x想象为二值图像中的像素。**

- 目标值y也是二元的, $y \in \{0,1\}$, 但我们的预测$\hat{y} = f(x)$实际上是区间$[0,1]$中的某一特定值。因此我们可以设定一个阈值，预测值$\hat{y}$大于阈值则分类为1，否则分类为0。

- 目标类是“猫”，数据是图像和标签对(x, y)， $y = 1$表示x是猫图像，而$y = 0$表示x不是猫。

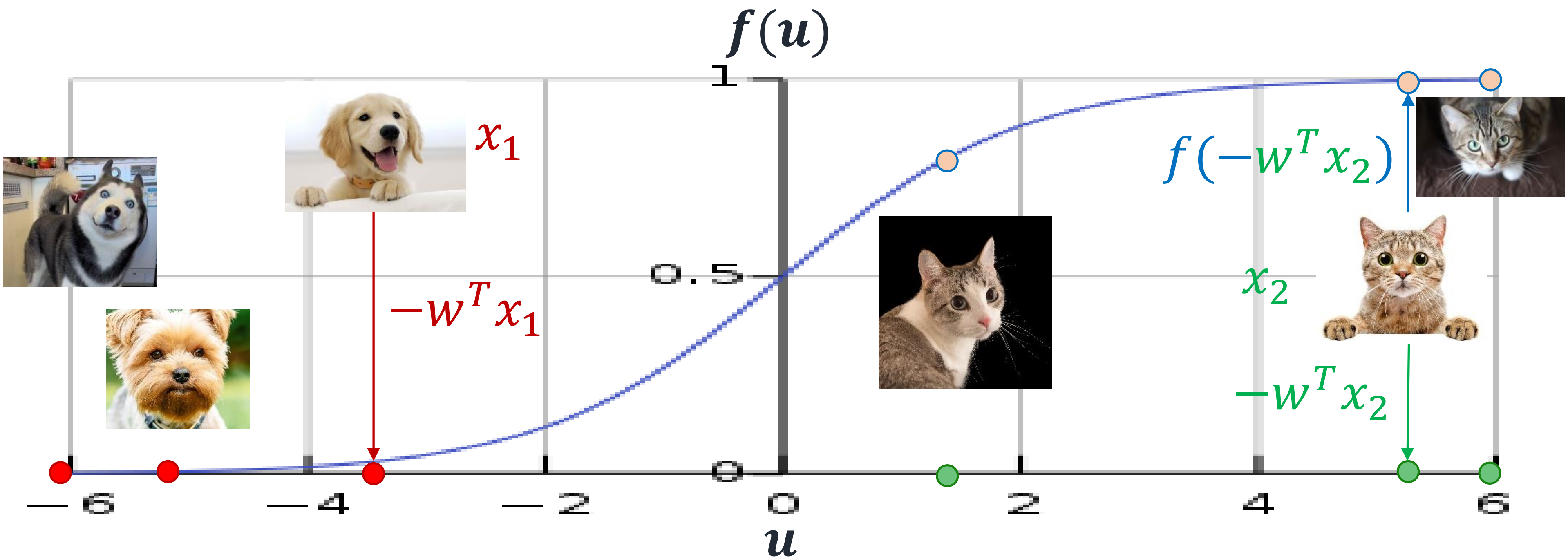
- 可以**将模型的输出$\hat{y} = f(x)$想象为图像x是猫的概率。**

Logistic 回归

设$w \in \mathbb{R}^m$是***Logistic***模型的权重向量，那么输出为“猫”的概率值就由下式确定:

$$f(x) = \frac{1}{1 + \exp(-w^T x)}$$

我们可以将线性方程写成$u = -w^T x$， 其中$f(u) = 1/(1 + \exp(-u))$是***Logistic***函数。



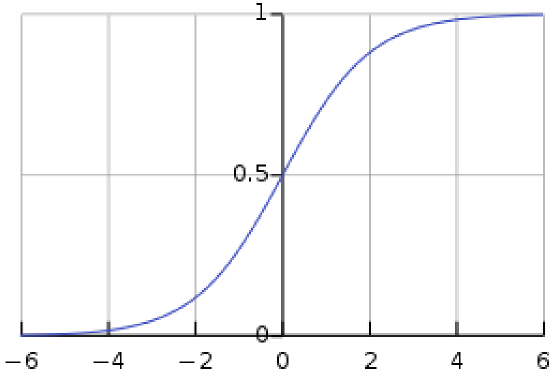
Logistic 回归

$Logistic$ 回归：

$$f(x) = \frac{1}{1 + \exp(-w^T x)}$$

我们可以注意到上式的输出为：

$$f(x) = \begin{cases} < 0.5 & \text{if} \quad w^T x < 0 \\ 0.5 & \text{if} \quad w^T x = 0 \\ > 0.5 & \text{if} \quad w^T x > 0 \end{cases}$$



选择阈值$y_{threshold} = 0.5$时可以分类得到$w^T x > 0$ 的值。

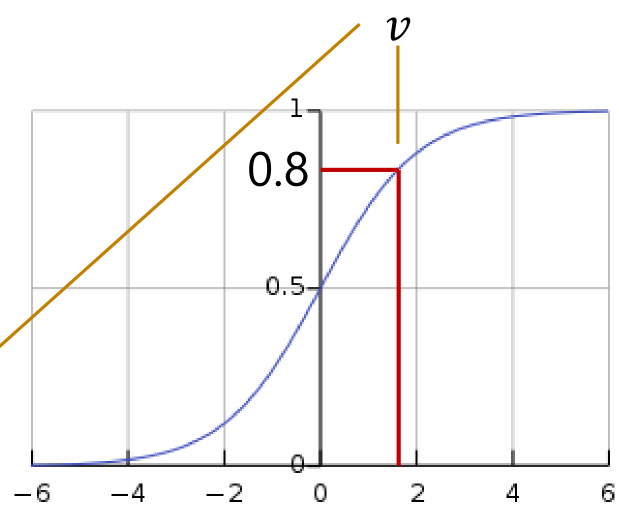
Logistic 回归

Logistic 回归：

$$f(x)=\frac{1}{1+\exp(-w^T x)}$$

我们可以注意到上式的输出为：

$$f(x)=\begin{cases} <0.8 & \text{if} \quad w^T x < v \\ \quad 0.8 & \text{if} \quad w^T x = v \\ >0.8 & \text{if} \quad w^T x > v \end{cases}$$



选择阈值$y_{threshold}=0.8$时可以分类得到$w^T x > v$的值。

Logistic 回归

➢学习准则

- 模型预测条件概率 $p_\theta(y|\boldsymbol{x})$

$$p_\theta(y=1|\boldsymbol{x})=\sigma(\boldsymbol{w}^T\boldsymbol{x})$$

- 真实条件概率 $p_r(y|\boldsymbol{x})$
  - 对于一个样本(x, y^*)，其真实条件概率为

$$p_r(y=1|\boldsymbol{x})=y^*$$
$$p_r(y=0|\boldsymbol{x})=1-y^*$$

如何衡量两个条件分布的差异?

Logistic 回归-损失函数

我们可以使用例如y 和 $\hat{y}=f(x)$ 之间的平方损失，但还有更自然（且有效）的选择。

它基于我们的假设： $\hat{y}=f(x)$是$\boldsymbol{x}$属于目标类别的概率。

基于这个假设，我们可以**计算正确分类的概率**，并**最大化**它。

正确分类的概率（对于一个输入）为

$$p_{correct}=\begin{cases}\hat{y} & \text{if} \quad y=1 \qquad \text{(true positive probability)}\\ 1-\hat{y} & \text{if} \quad y=0 \qquad \text{(true negative probability)}\end{cases}$$

我们可以将其变成一个简单的表达式

$$p_{correct}=y\,\hat{y}+\ (1-y)(1-\hat{y})$$

Logistic 回归-损失函数

我们可以使用 $-p_{correct}$作为每个观测值的损失:

$$-p_{correct} = -y\,\hat{y} - (1 - y)(1 - \hat{y})$$

我们可以将所有观察结果相加，并将其最小化，以最大化算法的整体准确性。有时会这样做，并且可能会产生良好的结果。

但更常见的是，我们使用**正确结果概率的负对数**（ the negative log of the probability of a correct result ）：

$$-\log p_{correct} = -y\log\hat{y} - (1 - y)\log(1 - \hat{y})$$

这称为**交叉熵损失*Cross-Entropy Loss*** (n 是观测值的数量):

$$L = -\sum_{i=1}^{n} y_i \log\hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i)$$

在这种情况下，**交叉熵损失**是**每个标签正确的负对数概率** 。

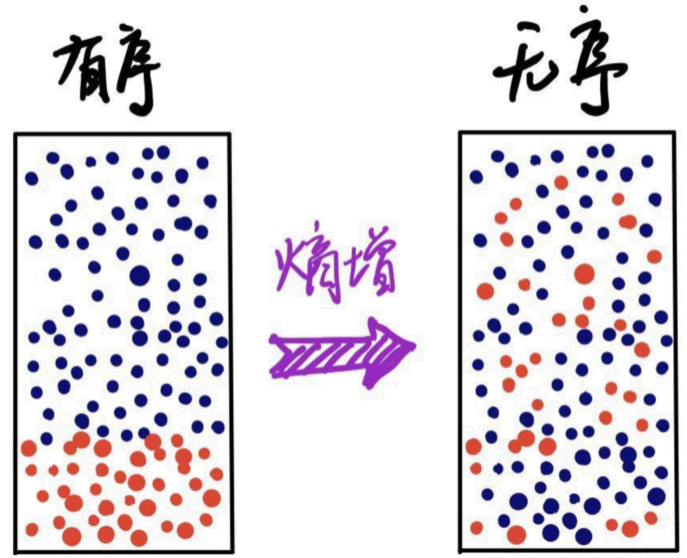
- 由于标签错误是独立的，我们应该将它们相乘以获得一切正确的总体概率。
- 取对数可以将该乘积转化为总和。

Logistic 回归

➢ **学习准则--** 熵（Entropy）

- 在信息论中，熵用来衡量一个随机事件的不确定性。
- 自信息（Self Information）：$I(x) = -\log(p(x))$
- 熵:

$$
\begin{aligned}
H(X) &= \mathbb{E}_X[I(x)] \\
&= \mathbb{E}_X[-\log p(x)] \\
&= -\sum_{x \in \mathcal{X}} p(x) \log p(x)
\end{aligned}
$$



- 熵越高，随机变量的信息越多；熵越低，随机变量的信息越少 。
- 在对分布 $q(y)$ 的符号进行编码时，熵 $I(q)$ 也是理论上最优的平均编码长度，这种编码方式称为熵编码（ Entropy Encoding ）。

Logistic 回归

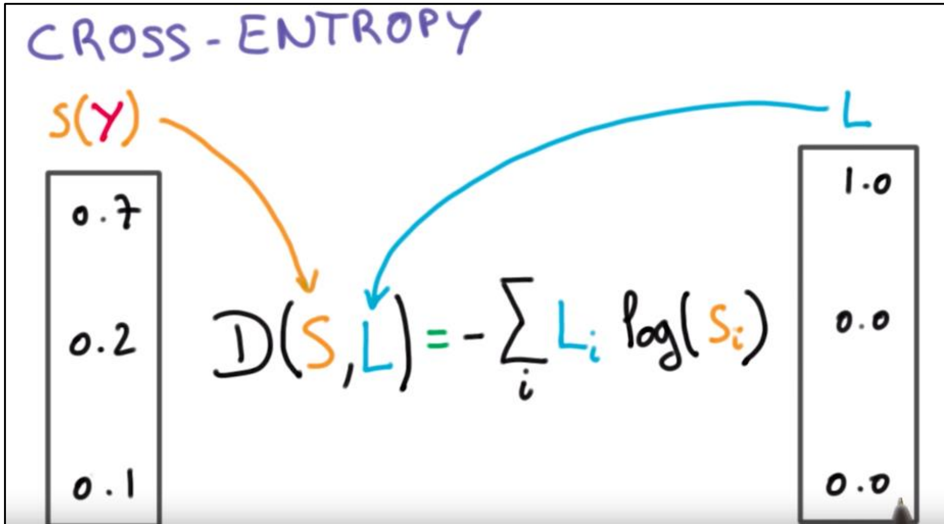
➤**学习准则** -- 交叉熵（Cross Entropy）

设q为估计的概率分布， p为真实的概率分布。

- 交叉熵是按照概率分布 q 的最优编码对真实分布为 p 的信息进行编码的长度。

$$
\begin{aligned}
H(p,q) &= \mathbb{E}_p[-\log q(x)] \\
&= -\sum_x p(x) \log q(x)
\end{aligned}
$$

- 在给定q 的情况下：交叉熵越小，代表两个分布越接近；
- 交叉熵越大，代表两个分布差距越大。



Logistic 回归

➢ **损失函数** -- 交叉熵损失（Cross Entropy Loss）

Logistic回归采用交叉熵作为损失函数

$$L = -\sum_{i=1}^{n} y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)$$

给定N个训练样本$\{(x(n), y(n))\}_{n=1}^{N}$， 用Logistic回归模型对每个样本$x(n)$进行预测，输出其标签为1的后验概率，记为模型预测条件概率$\hat{y}^{(n)}$：

$$\hat{y}^{(n)} = \sigma(\boldsymbol{w}^{\top} \boldsymbol{x}^{(n)}), \qquad 1 \leq n \leq N.$$

由于$y^{(n)} \in \{0, 1\}$， 样本$(\boldsymbol{x}^{(n)}, y^{(n)})$的真实条件概率可以表示为:

$$p_r(y^{(n)} = 1 | \boldsymbol{x}^{(n)}) = y^{(n)},$$

$$p_r(y^{(n)} = 0 | \boldsymbol{x}^{(n)}) = 1 - y^{(n)}.$$

Logistic 回归

➢ **损失函数** -- 交叉熵损失（Cross Entropy Loss）　$\hat{y}^{(n)} = \sigma(\boldsymbol{w}^{\top}\boldsymbol{x}^{(n)}), \qquad 1 \le n \le N.$

- 使用交叉熵损失函数，模型的经验风险函数为：

$$
\begin{aligned}
\mathcal{R}(\boldsymbol{w}) &= -\frac{1}{N}\sum_{n=1}^{N}\left(p_r(y^{(n)}=1|\boldsymbol{x}^{(n)})\log\hat{y}^{(n)} + p_r(y^{(n)}=0|\boldsymbol{x}^{(n)})\log(1-\hat{y}^{(n)})\right) \\
&= -\frac{1}{N}\sum_{n=1}^{N}\left(y^{(n)}\log\hat{y}^{(n)} + (1-y^{(n)})\log(1-\hat{y}^{(n)})\right).
\end{aligned}
$$

$$
\begin{aligned}
\frac{d}{dx}\sigma(x) &= \frac{d}{dx}\left(\frac{1}{1+e^{-x}}\right) \\
&= \frac{-(1+e^{-x})'}{(1+e^{-x})^2} \\
&= \frac{e^{-x}}{(1+e^{-x})^2} \\
&= \left(\frac{1}{1+e^{-x}}\right)\left(\frac{e^{-x}}{1+e^{-x}}\right) \\
&= \left(\frac{1}{1+e^{-x}}\right)\left(\frac{1+e^{-x}}{1+e^{-x}} - \frac{1}{1+e^{-x}}\right) \\
&= \sigma(x)\left(\frac{1+e^{-x}}{1+e^{-x}} - \sigma(x)\right) \\
&= \sigma(x)\,(1-\sigma(x))
\end{aligned}
$$

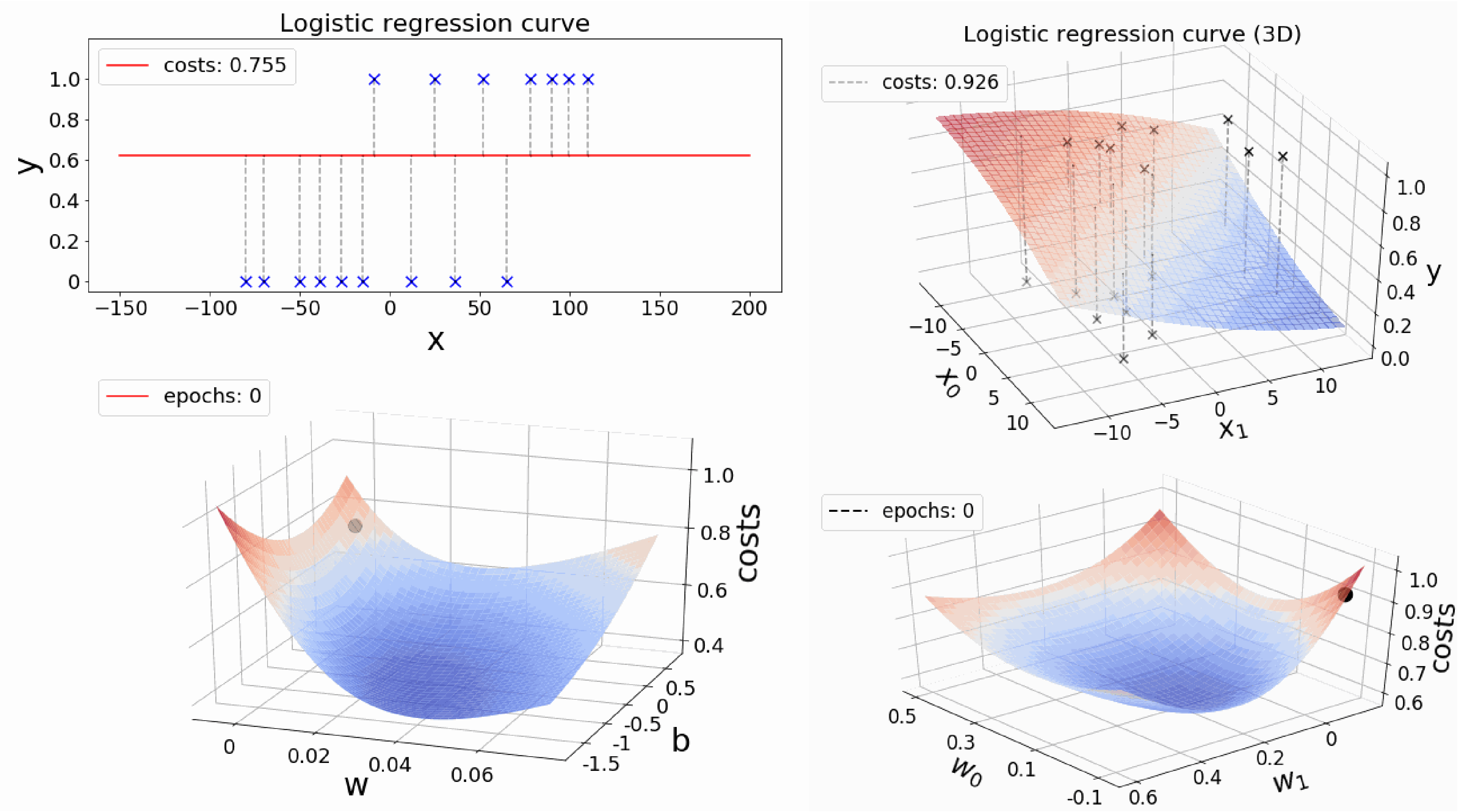
- 梯度为：

$$
\begin{aligned}
\frac{\partial\mathcal{R}(\boldsymbol{w})}{\partial\boldsymbol{w}} &= -\frac{1}{N}\sum_{n=1}^{N}\left(y^{(n)}\frac{\hat{y}^{(n)}(1-\hat{y}^{(n)})}{\hat{y}^{(n)}}\boldsymbol{x}^{(n)} - (1-y^{(n)})\frac{\hat{y}^{(n)}(1-\hat{y}^{(n)})}{1-\hat{y}^{(n)}}\boldsymbol{x}^{(n)}\right) \\
&= -\frac{1}{N}\sum_{n=1}^{N}\left(y^{(n)}(1-\hat{y}^{(n)})\boldsymbol{x}^{(n)} - (1-y^{(n)})\hat{y}^{(n)}\boldsymbol{x}^{(n)}\right) \\
&= -\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}^{(n)}(y^{(n)}-\hat{y}^{(n)}).
\end{aligned}
$$

梯度下降算法更新梯度

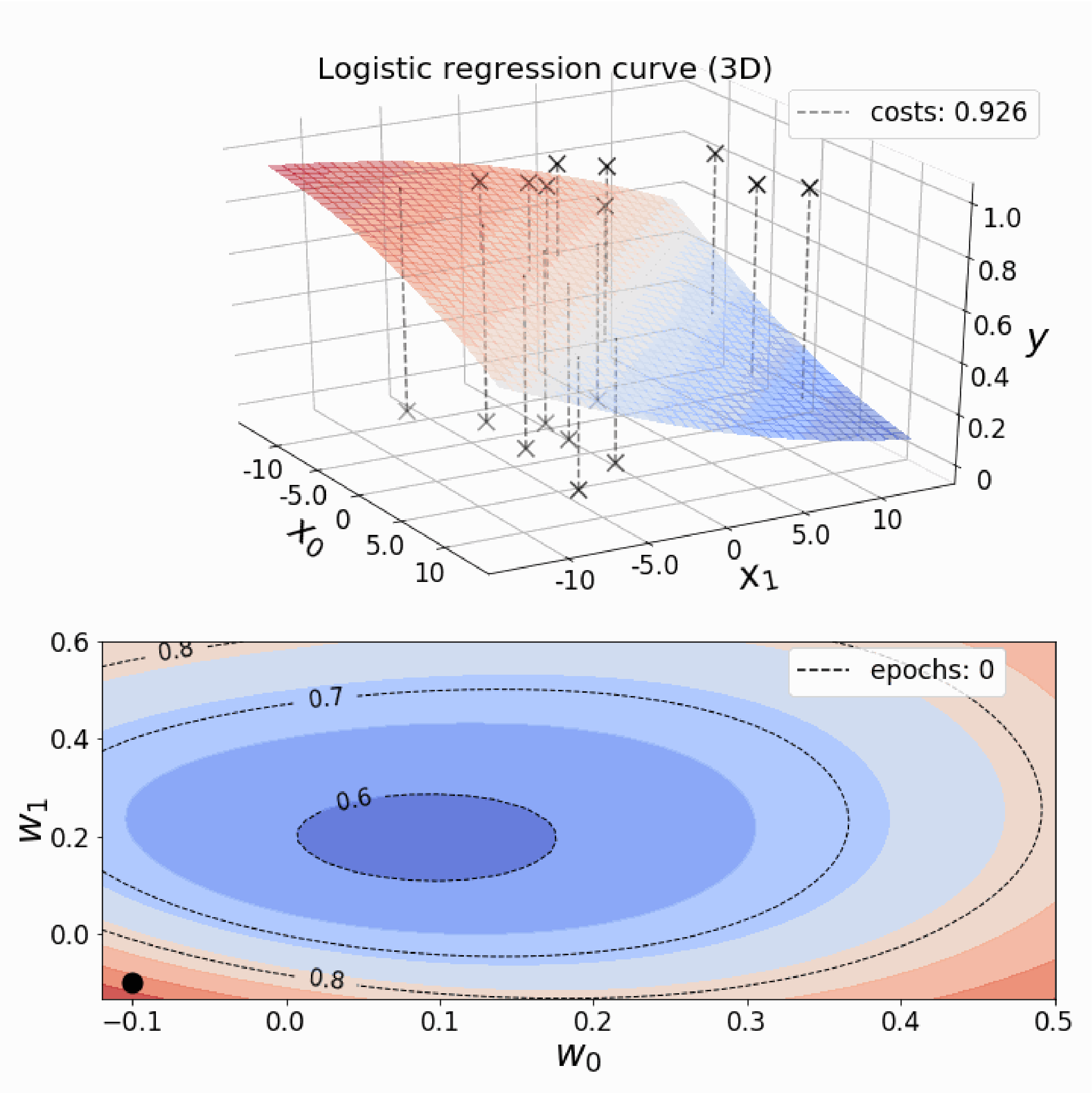
$$
\boxed{\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t + \alpha\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}^{(n)}\left(y^{(n)} - \hat{y}^{(n)}_{\boldsymbol{w}_t}\right),}
$$

Logistic 回归



https://towardsdatascience.com/animations-of-logistic-regression-with-python-31f8c9cb420

Logistic 回归



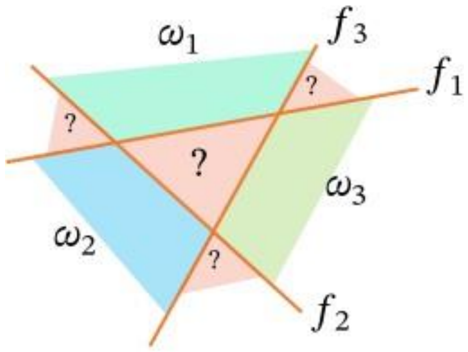
https://towardsdatascience.com/animations-of-logistic-regression-with-python-31f8c9cb420

➢ 多分类问题

多分类问题是指分类的类别数大于2，多分类一般需要多个线性判别函数，但设计这些判别函有很多种方式。

假设一个多分类问题的类别为$\{1, 2, \cdots, C\}$， 常用的方式有以下三种：

- **“一对其余”方式**：把多分类问题转换为 C 个“一对其余”的二分类问题。这种方式共需要 C 个判别函数，其中第 c 个判别函数 f_c 是将类别 c 的样本和不属于类别 c 的样本分开。



(a) “一对其余”方式

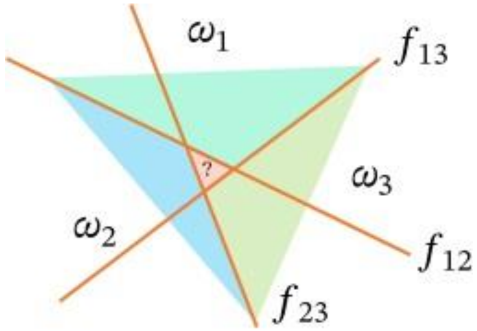
Logistic 回归

➢ 多分类问题

多分类问题是指分类的类别数大于2，多分类一般需要多个线性判别函数，但设计这些判别函有很多种方式。

假设一个多分类问题的类别为$\{1, 2, \cdots, C\}$，常用的方式有以下三种：

- **"一对一" 方式**：把多分类问题转换为$C(C-1)/2$个"一对一"的二分类问题。这种方式共需要 $C(C-1)/2$个判别函数，其中第(i, j) 个判别函数是把类别 i 和类别 j 的样本分开。



(b) "一对一"方式

"一对其余"方式和"一对一"方式都存在一个缺陷：特征空间中会存在一些难以确定类别的区域。

Logistic 回归

➢ 多分类问题

多分类问题是指分类的类别数大于2，多分类一般需要多个线性判别函数，但设计这些判别函数有很多种方式。

假设一个多分类问题的类别为$\{1, 2, \cdots, C\}$， 常用的方式有以下三种：

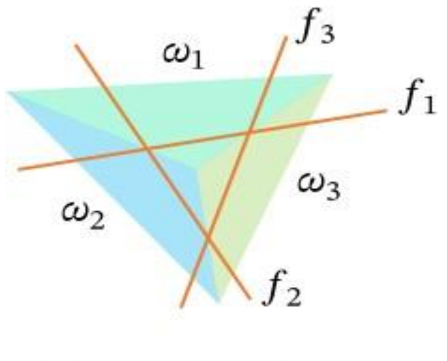
- **“argmax”方式**：这是一种改进的“一对其余”方式，共需要 C 个判别函数

$$f_c(\boldsymbol{x}; \boldsymbol{w}_c) = \boldsymbol{w}_c^{\top}\boldsymbol{x} + b_c, \qquad c \in \{1, \cdots, C\}$$

对于样本$\boldsymbol{x}$，如果存在一个类别c，相对于所有的其他类别$c'(c' \neq c)$有$f_c(\boldsymbol{x}; \boldsymbol{w}_c) > f_{c'}(\boldsymbol{x}, \boldsymbol{w}_{c'})$，那么$\boldsymbol{x}$属于类别$c$。

“argmax”方式的预测函数为:

$$y = \arg\max_{c=1}^{C} f_c(\boldsymbol{x}; \boldsymbol{w}_c)$$



(c) “argmax”方式

Logistic 回归

➢多分类问题

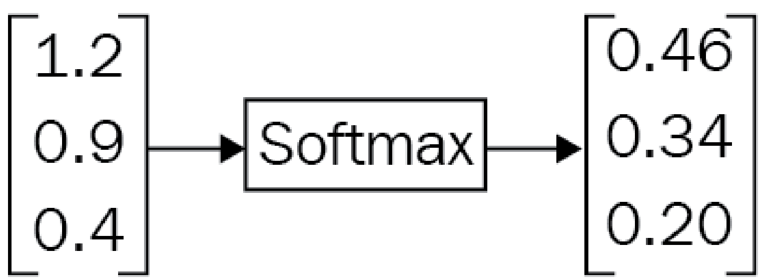
定义 3.2 – 多类线性可分：对于训练集 $\mathcal{D} = \{(\boldsymbol{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$，如果存在 C 个权重向量 $\boldsymbol{w}_1^*, \cdots, \boldsymbol{w}_C^*$，使得第 $c(1 \le c \le C)$ 类的所有样本都满足 $f_c(\boldsymbol{x}; \boldsymbol{w}_c^*) > f_{\tilde{c}}(\boldsymbol{x}, \boldsymbol{w}_{\tilde{c}}^*), \forall \tilde{c} \neq c$，那么训练集 $\mathcal{D}$ 是线性可分的.

从上面定义可知，如果数据集是多类线性可分的，那么一定存在一个“argmax”方式的线性分类器可以将它们正确分开。

Logistic 回归

Softmax回归（ Softmax Regression），也称为多项（Multinomial）或多类（Multi-Class）的Logistic回归

$$\mathrm{softmax}(x_k) = \frac{\exp(x_k)}{\sum_{i=1}^{K} \exp(x_i)}$$



Logistic 回归

- 给定一个样本$\boldsymbol{x}$，Softmax 回归预测的属于类别c的条件概率为：

$$p(y=c|\boldsymbol{x}) = \mathrm{softmax}(\boldsymbol{w}_c^{\top}\boldsymbol{x}) = \frac{\exp(\boldsymbol{w}_c^{\top}\boldsymbol{x})}{\sum_{c'=1}^{C}\exp(\boldsymbol{w}_{c'}^{\top}\boldsymbol{x})},$$

$\boldsymbol{w}_c$是第c类的权重向量

$$\Rightarrow$$

$$\hat{\boldsymbol{y}} = \mathrm{softmax}(\boldsymbol{W}^{\top}\boldsymbol{x}) = \frac{\exp(\boldsymbol{W}^{\top}\boldsymbol{x})}{\mathbf{1}_C^{\top}\exp(\boldsymbol{W}^{\top}\boldsymbol{x})},$$

其中$\boldsymbol{W} = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_C]$是由$C$个类的权重向量组成的矩阵，$\mathbf{1}_C$为$C$维的全1向量，$\hat{\boldsymbol{y}} \in \mathbb{R}^C$为所有类别的预测条件概率组成的向量，第$c$维的值是第$c$类的预测条件概率.

- Softmax 回归的决策函数可以表示为：

$$\hat{y} = \arg\max_{c=1}^{C} p(y=c|\boldsymbol{x}) = \arg\max_{c=1}^{C} \boldsymbol{w}_c^{\top}\boldsymbol{x}.$$

Logistic 回归

- **学习准则 --**交叉熵损失

  - **风险函数**

$$\mathcal{R}(\boldsymbol{W}) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C}\boldsymbol{y}_c^{(n)}\log\hat{\boldsymbol{y}}_c^{(n)}$$

$$= -\frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{y}^{(n)})^{\top}\log\hat{\boldsymbol{y}}^{(n)}, \qquad \hat{\boldsymbol{y}}^{(n)} = \mathrm{softmax}(\boldsymbol{W}^{\top}\boldsymbol{x}^{(n)})$$

  - **风险函数的梯度**

$$\frac{\partial\mathcal{R}(\boldsymbol{W})}{\partial\boldsymbol{W}} = -\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}^{(n)}\left(\boldsymbol{y}^{(n)} - \hat{\boldsymbol{y}}^{(n)}\right)^{\top}$$

推导参见教材P61

Logistic 回归

➢ **模型**

$$P(y=c|\mathbf{x}) = \mathrm{softmax}(\mathbf{w}_c^\top \mathbf{x}) = \frac{\exp(\mathbf{w}_c^\top \mathbf{x})}{\sum_{i=1}^{C} \exp(\mathbf{w}_i^\top \mathbf{x})}.$$

➢ **学习准则：交叉熵**

$$\mathcal{R}(\boldsymbol{W}) = -\frac{1}{N}\sum_{n=1}^{N} (\boldsymbol{y}^{(n)})^\top \log \hat{\boldsymbol{y}}^{(n)},$$

➢ **优化算法：**梯度下降

$$\boldsymbol{W}_{t+1} \leftarrow \boldsymbol{W}_t + \alpha\left(\frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}^{(n)}\left(\boldsymbol{y}^{(n)} - \hat{\boldsymbol{y}}_{\boldsymbol{W}_t}^{(n)}\right)^\top\right)$$

$\hat{\boldsymbol{y}}_{\boldsymbol{W}_t}^{(n)}$ 是当参数为 $\boldsymbol{W}_t$ 时，Softmax 回归模型的输出.