lec10.3 Logistic Regression

Logistic Regression

这属于判别模型

我们考虑一个类标为 $\{-1, +1\}$ 的二分类问题：希望构建一个概率分类器，该分类器输出一个特定训练实例 X 是正例 $(y = +1)$ 或负例 $(y = -1)$ 的概率

Logistic Regression

Main Idea

定义 separating hyperplane H：

分离超平面 H 由特征权重 $W = (w_1, \ldots, w_d)$ 和偏置参数 b 参数化：

$$H = \left\{ b + \sum_{i=1}^{d} w_i x_i = 1 \mid x_1, \ldots, x_d \right\}$$

感知器分类方法：

在感知器中，对于一个输入对象 $X = (x_1, \ldots, x_d)$，我们只使用 $b + W^T X$ 的符号来进行分类：

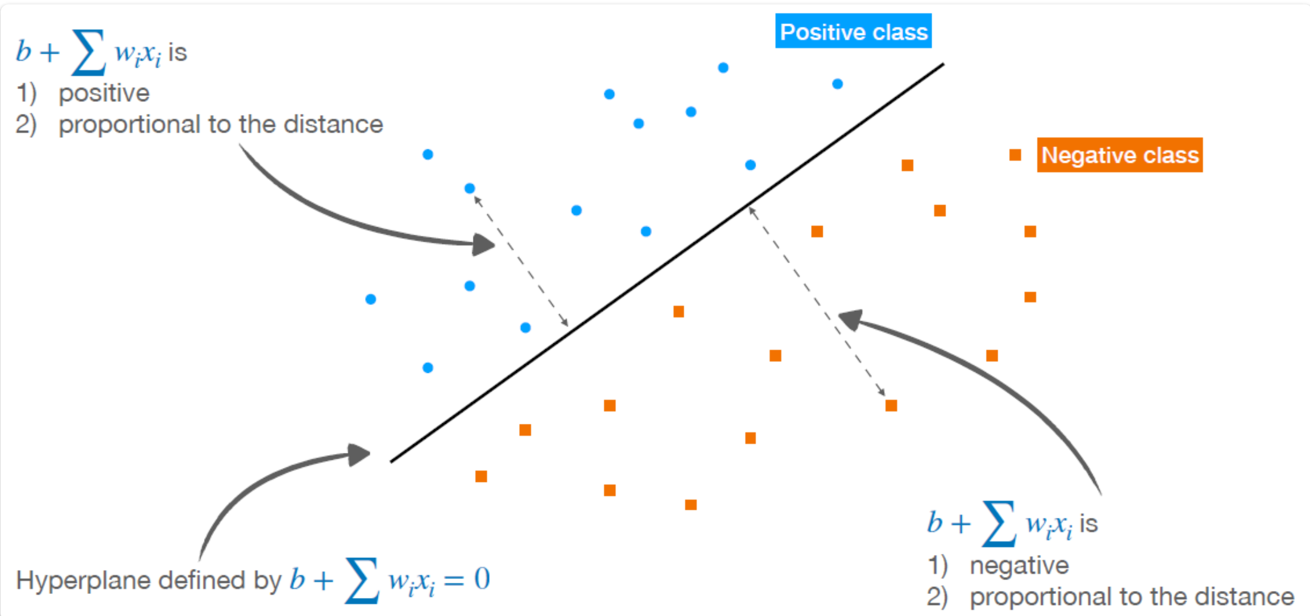
$$b + W^T X = b + \sum_{i=1}^{d} w_i x_i$$

这个值告诉我们点位于超平面创建的两个半空间中的哪一个

逻辑回归：

实际上 $b + W^T X$ 的值传递了额外的有用信息：他与点 X 到超平面 H 的距离成正比。我们使用：

- $b + W^T X$ 分类对象 X
- $|b + W^T X|$ 来量化我们对分类的置信度，值越大，点 X 距分离超平面 H 越远



图中，超平面由 $b + \sum_{i=1}^{d} w_i x_i = 0$ 定义，正负类别在超平面的两侧

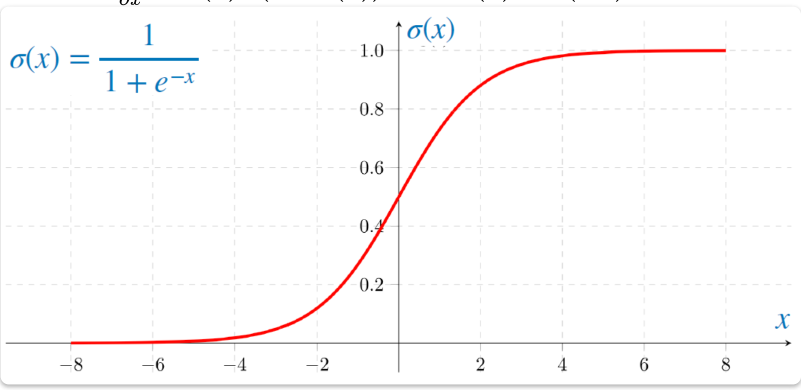
- $b+\sum_{i=1}^{d} w_i x_i$ 为正时，点位于正类区域，且值越大，距离超平面越远
- $b+\sum_{i=1}^{d} w_i x_i$ 为负时，点位于负类区域，且值越小，距离超平面越远

为了将置信分数 $b+W^TX \in (-\infty,+\infty)$ 解释为概率，我们希望将其映射为为区间 $[0,1]$ 内的值，而这需要 sigmoid 函数

Logistic Sigmoid：

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

其中，$x \in (-\infty,+\infty)$，$\sigma(x) \in [0,1]$
特别地，$\frac{\partial \sigma}{\partial x}=\sigma(x)\cdot(1-\sigma(x))$，$1-\sigma(x)=\sigma(-x)$



Discriminative Classifier 判别分类器

假设条件分布 $P(C \mid X)$ 具有特定形式 $P_\theta(C \mid X)$，它依赖于一些参数 $\theta=(\theta_1,\ldots,\theta_k)$

我们可以使用训练集来找到这些参数 $\theta_1,\ldots,\theta_k$，使得所得到的分布在假定形式的所有分布中是最佳的（在上两个笔记中有详细解释）

Model Assumption 模型假设

对于一个对象 $X=(x_1,\ldots,x_d)$，X 属于正类的概率可以被建模为：

$$P(y=+1 \mid X)=\sigma(a)=\frac{1}{1+e^{-a}}$$

其中，$a=b+W^TX$
因此，X 属于负类的概率为：

$$P(y=-1 \mid X)=1-P(y=+1 \mid X)=1-\sigma(a)=\sigma(-a)=\frac{1}{1+e^{a}}$$

整理两式为：

$$P(y=t \mid X)=\sigma(t\cdot a)=\frac{1}{1+e^{-t\cdot a}}$$

其中，$t \in \{-1,+1\}$

Choosing/Fitting Parameters 选择/拟合参数

定义训练集 $\mathscr{D}$ 为

$$\mathscr{D}=\{(X_1,y_1),\ldots,(X_n,y_n)\}$$

其中，(X,y) 为对象-标签对，$y=\{-1,+1\}$

我们希望在给定训练集 $\mathscr{D}$ 的情况下，找到能够使得似然函数 ℓ 最大化的参数 $b, w_1, \ldots, w_d$，因此使用朴素贝叶斯 MLE 进行估计，令 $a_i = b + W^T X_i$，定义似然函数为：

$$\ell(b, w_1, \ldots, w_d \mid \mathscr{D}) = \prod_{i=1}^{n} \sigma(y_i a_i)$$

其中，σ 是 logistic sigmoid 函数；这个式子是所有样本的联合概率，我们希望这个概率最大化
上式等价于最小化 $-\ell$ 或负对数似然函数：

$$-\ell\ell = -\log \ell = -\sum_{i=1}^{n} \log \sigma(y_i a_i)$$

这里取对数是为了方便计算

接下来我们计算梯度，即 $\frac{\partial \ell\ell}{\partial b}$ 和 $\frac{\partial \ell\ell}{\partial w_k}$，其中，$k = 1, \ldots, d$

$$\frac{\partial \ell\ell}{\partial b} = \sum_{i=1}^{n} y_i \cdot \sigma(-y_i a_i)$$
$$\frac{\partial \ell\ell}{\partial w_k} = \sum_{i=1}^{n} y_i \cdot \sigma(-y_i a_i) \cdot x_k^{(i)}$$

其中，$X_i = \left(x_1^{(i)}, \ldots, x_d^{(i)}\right)$

对于 $\sum_{i=1}^{n} y_i \cdot \sigma(-y_i a_i)$：

- 如果 $y_i = +1$，则 $\sigma(-y_i \cdot a_i) = \sigma(-a_i) = 1 - \sigma(a_i) = P(y = -1 \mid X_i)$
- 如果 $y_i = -1$，则 $\sigma(-y_i \cdot a_i) = \sigma(a_i) = P(y = +1 \mid X_i)$

因此，$\sigma(-y_i \cdot a_i)$ 是误分类训练对象 X_i 的概率，且

$$\sum_{i=1}^{n} y_i \cdot \sigma(-y_i a_i) = \sum_{X_i \in \mathscr{D}_+} P(y = -1 \mid X_i) - \sum_{X_i \in \mathscr{D}_-} P(y = +1 \mid X_i)$$

通过这些步骤，我们可以使用梯度下降法来最小化负对数似然函数，从而找到最佳参数 b 和 W 来拟合逻辑回归模型

Update Rule 更新规则

这里使用梯度下降法寻找极值

1. 选择一个初始点 Z_0
2. 根据以下公式迭代更新

$$Z_{i+1} = Z_i - \gamma_i \cdot \nabla_Z f(Z_i)$$

其中，γ_i 是学习率（步长），$\nabla_Z f(Z_i)$ 是 Z_i 处的梯度

对于负对数似然函数

$$-\log \ell = -\sum_{i=1}^{n} \log \sigma(y_i a_i)$$

的梯度 $\frac{\partial \ell\ell}{\partial b}$ 和 $\frac{\partial \ell\ell}{\partial w_k}$，计算如下：

$$\frac{\partial \ell\ell}{\partial b} = \sum_{i=1}^{n} y_i \cdot \sigma(-y_i a_i)$$
$$\frac{\partial \ell\ell}{\partial w_k} = \sum_{i=1}^{n} y_i \cdot \sigma(-y_i a_i) \cdot x_k^{(i)}$$

其中，$k = 1, \ldots, d$，$a_i = b + W^T X_i$

因此我们便得到了以下的更新规则：

$$b \leftarrow b + \mu \sum_{i=1}^{n} y_i \cdot \sigma(-y_i a_i)$$
$$W \leftarrow W + \mu \sum_{i=1}^{n} y_i \cdot \sigma(-y_i a_i) \cdot X_i$$

其中，μ 是学习率（步长）

Batch 与 Online 梯度下降优化方法的介绍

Batch（批量梯度下降）

- 使用整个训练数据集：在每次迭代中，使用整个训练数据集来更新权重向量
- 常用算法：批量学习逻辑回归的常用优化算法是有限内存 BFGS (Limited Memory BFGS，L-BFGS) 算法
- 性能特点：批量梯度下降版本相比在线版本较慢，但在许多情况下显示出略微提高的准确性

Online（在线梯度下降）

- 使用单个训练样本：在每次迭代中，仅使用一个训练样本来更新权重向量
- 常用算法：使用随机梯度下降算法（Stochastic Gradient Descent, SGD）
- 性能特点：SGD 版本可能需要多次迭代整个数据集才能收敛（如果收敛的话）
- 应用场景：SGD 是一种经常用于大规模机器学习任务的技术，即使目标函数是非凸的

总结

- Batch：适用于数据集较小或计算资源充足的情况，因为它在每次更新时使用整个数据集，从而可能获得更高的精度，但计算成本较高
- Online：适用于大规模数据集或计算资源有限的情况，因为它在每次更新时仅使用一个样本，计算成本较低，但可能需要更多的迭代才能达到收敛

算法

逻辑回归在线算法（随机梯度下降）

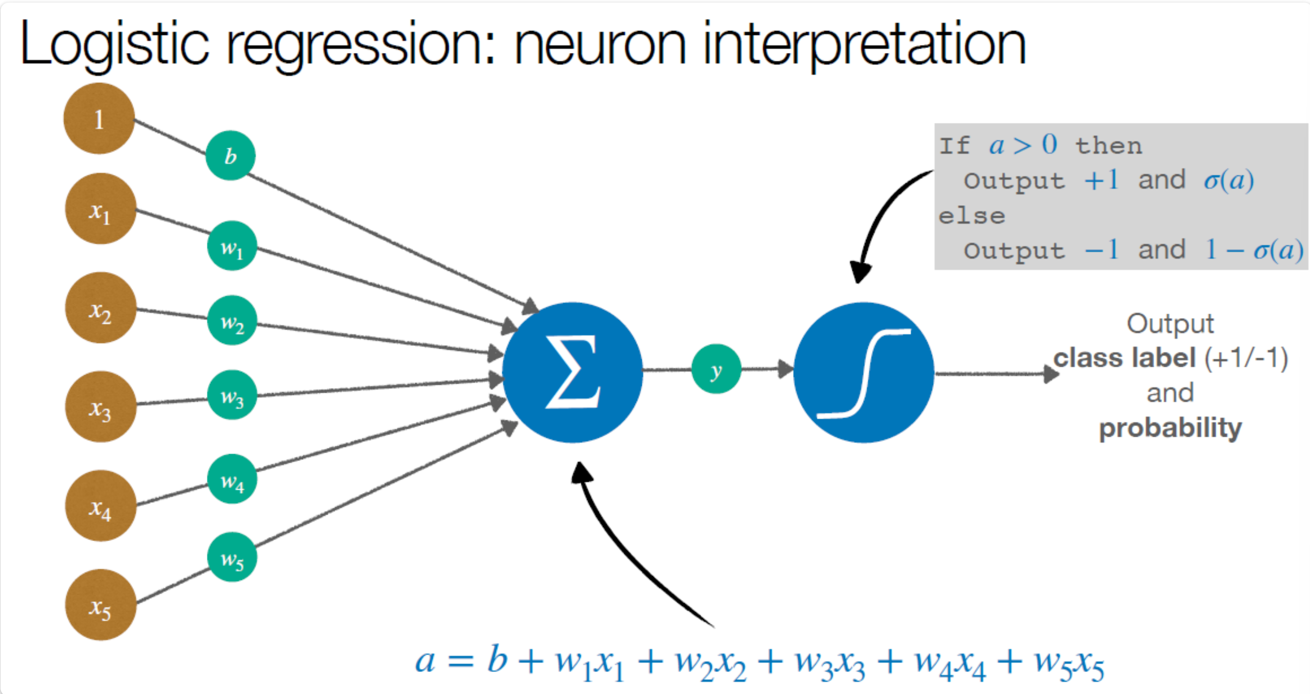
$$
\begin{aligned}
&\text{LogisticRegression}(\text{TrainingData: } \{(X_1, y_1), \ldots, (X_n, y_n)\}, \text{LearningRate: } \mu, \text{MaxIter}) \\
&\quad w_i = 0 \text{ for all } i = 1, \ldots, d \\
&\quad b = 0 \\
&\quad \text{for iter} = 1 \ldots \text{ MaxIter do} \\
&\quad\quad \text{for } i = 1 \ldots n \text{ do} \\
&\quad\quad\quad a_i = b + W^T X_i \\
&\quad\quad\quad w_j = w_j + \mu y_i \cdot \sigma(-y_i a_i) \cdot x_j^{(i)}, \text{ for all } j = 1, \ldots, d \\
&\quad\quad\quad b = b + \mu y_i \cdot \sigma(-y_i a_i) \\
&\quad \text{return } b, w_1, \ldots, w_d
\end{aligned}
$$

逐样本进行权重和偏置的更新，通过多次迭代优化参数

逻辑回归预测

$$
\begin{aligned}
&\text{LogisticRegressionTest}(b, w_1, w_2, \ldots, w_d, X) \\
&\quad a = b + W^T X \\
&\quad \text{if } a > 0 \text{ then} \\
&\qquad \text{predictedLabel} = +1 \quad \#\text{positive class} \\
&\qquad \text{probability that } X \text{ belongs to the positive class} = \sigma(a) \quad \#\text{confidence} \\
&\quad \text{else} \\
&\qquad \text{predictedLabel} = -1 \quad \#\text{negative class} \\
&\qquad \text{probability that } X \text{ belongs to the negative class} = 1 - \sigma(a) \quad \#\text{confidence}
\end{aligned}
$$

根据参数和新的输入样本，计算逻辑回归模型的输出值，并确定样本的类别以及对应的概率



对 Logistic Regression 施加正则化

L2 正则

记 $L(\mathscr{D}, W)$ 为使用权重向量 W 对数据集 $\mathscr{D}$ 进行分类的损失，我们希望在 W 上施加 L2 正则化。这意味着我们不仅要最小化分类损失，还要最小化权重向量的 L2 范数

结合 L2 正则化项，总体目标函数可以写为：

$$J(\mathscr{D}, W) = L(\mathscr{D}, W) + \lambda \|W\|_2^2 = L(\mathscr{D}, W) + \lambda \sum_{i=1}^{d} w_i^2$$

其中，λ 被称为正则化系数，通常通过交叉验证来设置

总体目标函数的梯度变为损失梯度和加权后的权重向量的梯度之和：

$$\nabla_W J(\mathscr{D}, W) = \nabla_W L(\mathscr{D}, W) + 2\lambda W$$

带 L2 正则化的逻辑回归更新规则

不带正则化时

对于一个训练样本 (X, y)，更新权重向量 W 的规则为：

$$W \leftarrow W + \mu y \cdot \sigma(-ya) \cdot X$$

带 L2 正则化

$$\begin{aligned} W &\leftarrow W - \mu(-y \cdot \sigma(-ya) \cdot X + 2\lambda W) \\ &= (1 - 2\mu\lambda)W + \mu y \cdot \sigma(-ya) \cdot X \end{aligned}$$

一对多方法（One-vs.-Rest Approach）

在这个方法中，我们假设二分类算法 A 可以输出一个数值分数，表示其对某个对象属于特定类别的“置信度”

1. 为每个类别 i 训练二分类器：
   - 对于每个类别 i，使用该类别的对象作为正样本，其他所有类别的对象作为负样本来训练二分类器 A
   - 记得到的分类器为 A_i
2. 这样，我们将得到 k 个预测模型，k 是类别的总数
3. 对新对象进行预测：
   - 对于一个新的对象 X，应用所有预测模型 $A_1, \ldots, A_k$
   - 每个模型 A_i 会输出一个置信度分数，表示该对象属于类别 i 的置信度
4. 输出最终类别：
   - 输出对对象 X 置信度最高的类别标签 y：

$$y = \underset{i \in \{1, \ldots, k\}}{\operatorname{argmax}} A_i(X)$$