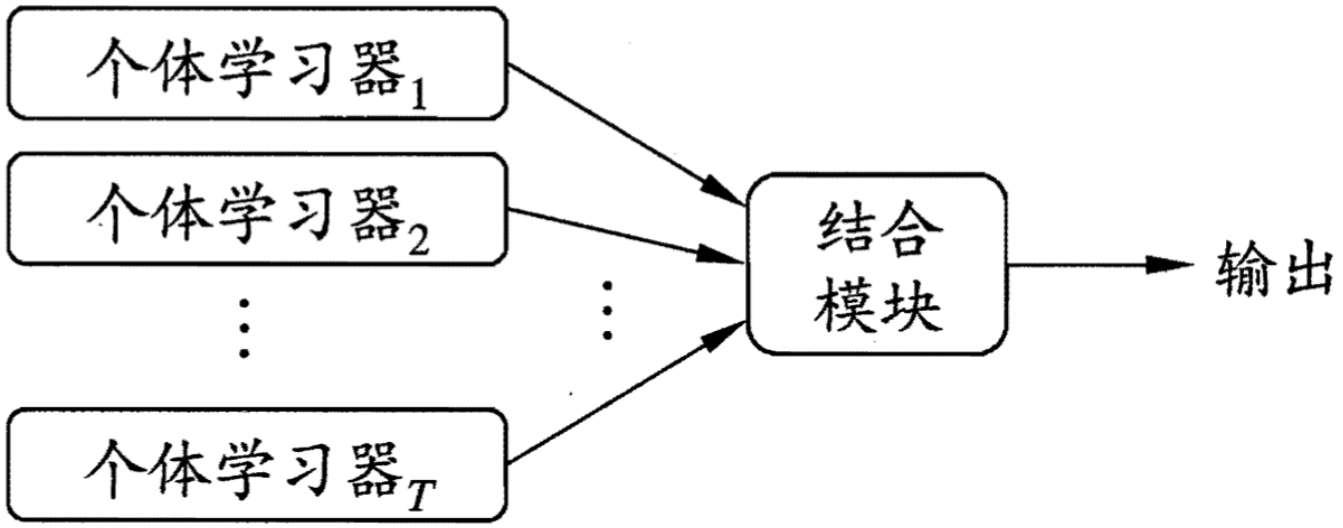
模型组合器
(集成学习)

集成学习

- 集成学习（Ensemble Learning）通过结合多个学习器（通常称为“个体学习器”或“基学习器”）来提高整体的学习性能。



集成学习

同质与异质学习器：

- 同质集成：个体学习器是同种类型的算法。例如，随机森林
- 异质集成：个体学习器是不同类型的算法。

基学习器（个体学习器）的选择：

- 基学习器之间应该“好而不同”，不同指的是误差之间应当相互独立。

常见的集成学习策略：

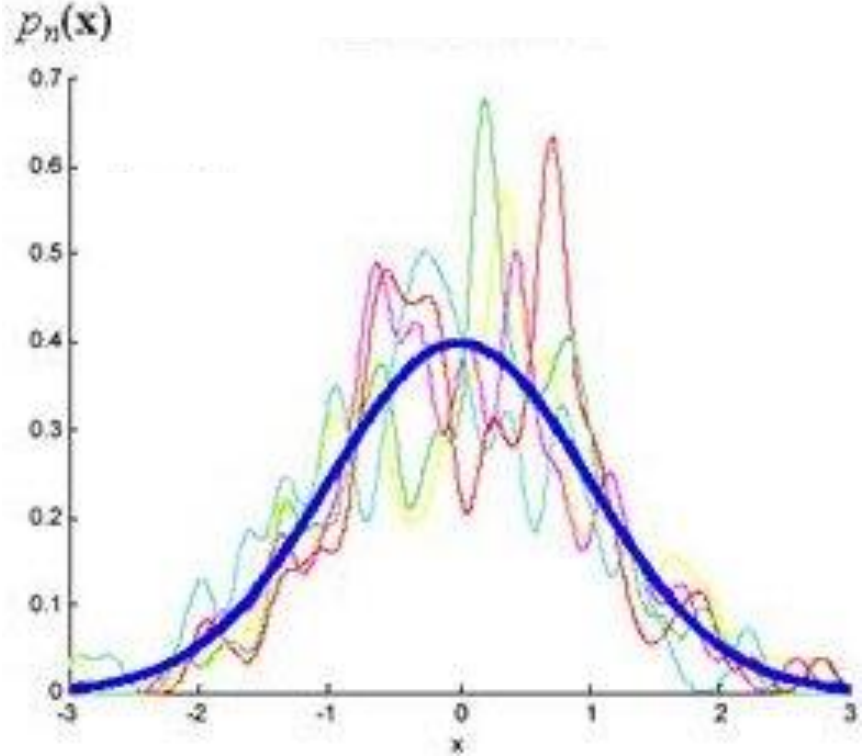
- Bagging：个体学习器是独立生成的，如随机森林。学习器之间不存在强依赖关系。
- Boosting：后一个学习器的构建依赖于前一个学习器的结果。学习之间存在强依赖关系。

缺点？（计算成本高，模型复杂，解释性差。。。）

集成学习

为什么集成学习有效?

- **对于Bagging这一类算法，**因为基学习器的误差相互独立时，误差可以相互抵消。从而降低模型方差。



集成之后的输出结果（深蓝色线），以及每棵树输出的结果（其它颜色线）

集成学习

为什么集成学习有效?

- **对于Boosting一类算法**，因为每个新加入的分类器都专注于前一个分类器错误分类的样本，那么如果每个分类器都能有效地减少错误，整体错误率将随着分类器数量的增加而指数级下降。

（例）在AdaBoost中（Boosting的一个代表方法）

整体模型的错误率由下式给出:

$$\exp\left(-2\sum_{t=1}^{T}\left(\frac{1}{2}-\epsilon_t\right)^2\right)$$

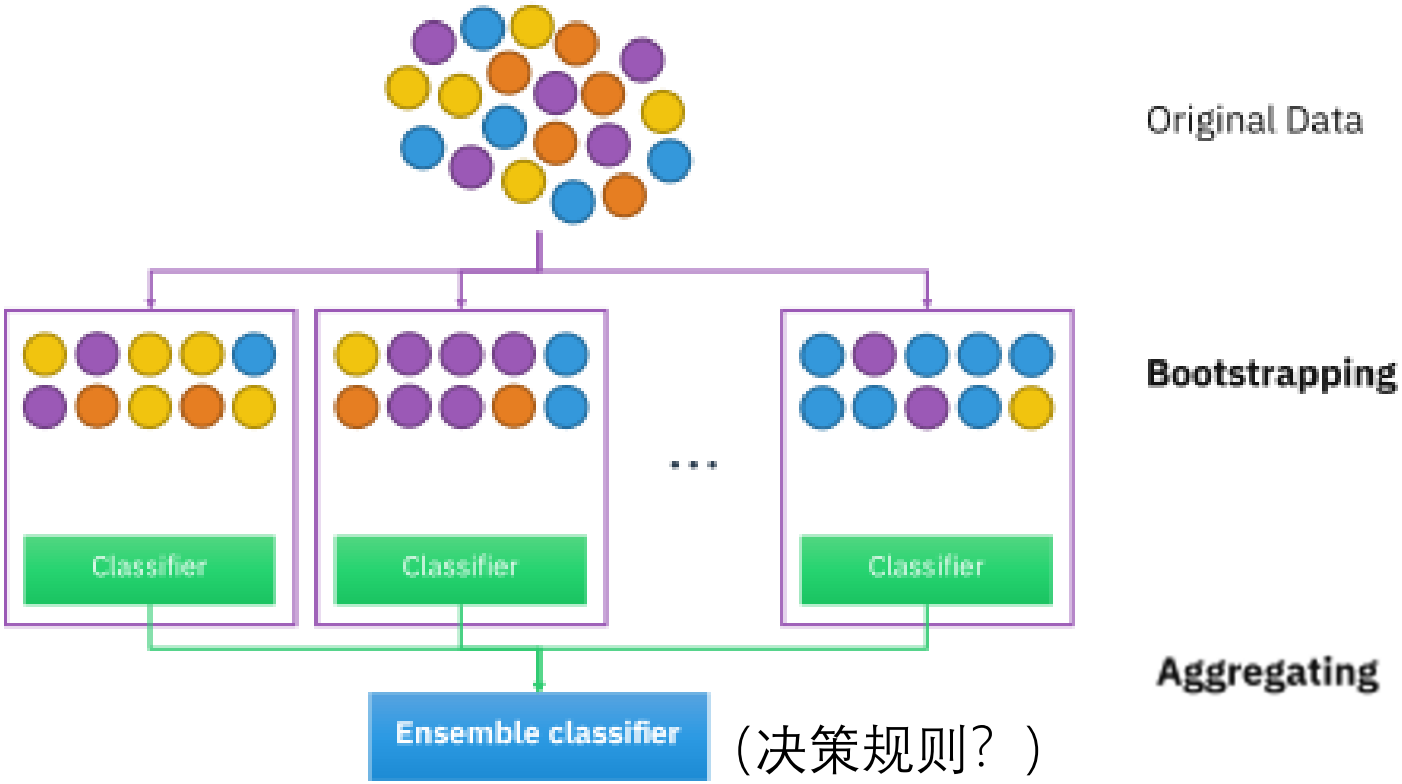
t代表的是分类器的编号， ϵ_t是分类器t的错误率

如果每一个弱分类器的表现都至少比随机猜测好（即每个 ϵ_t都小于0.5），那么这个错误率上界会随着分类器数量T的增加而指数级下降。

集成学习： Bagging

"Bagging" 是 "Bootstrap Aggregating" 的缩写

Bagging的基本流程：



（决策规则？）

随机森林 = Bagging （决策树基学习器） + 每个决策点的特征子集随机选取

集成学习：Bagging （bootstrap aggregating）

为什么集成学习使用Bootstrap抽样？

- 欲得到泛化性能强的集成，集成中的个体学习器应尽可能相互独立；

  第一种做法：对训练样本进行采样，产生出若干个完全不同的子集，再从每个数据子集中训练出一个基学习器。

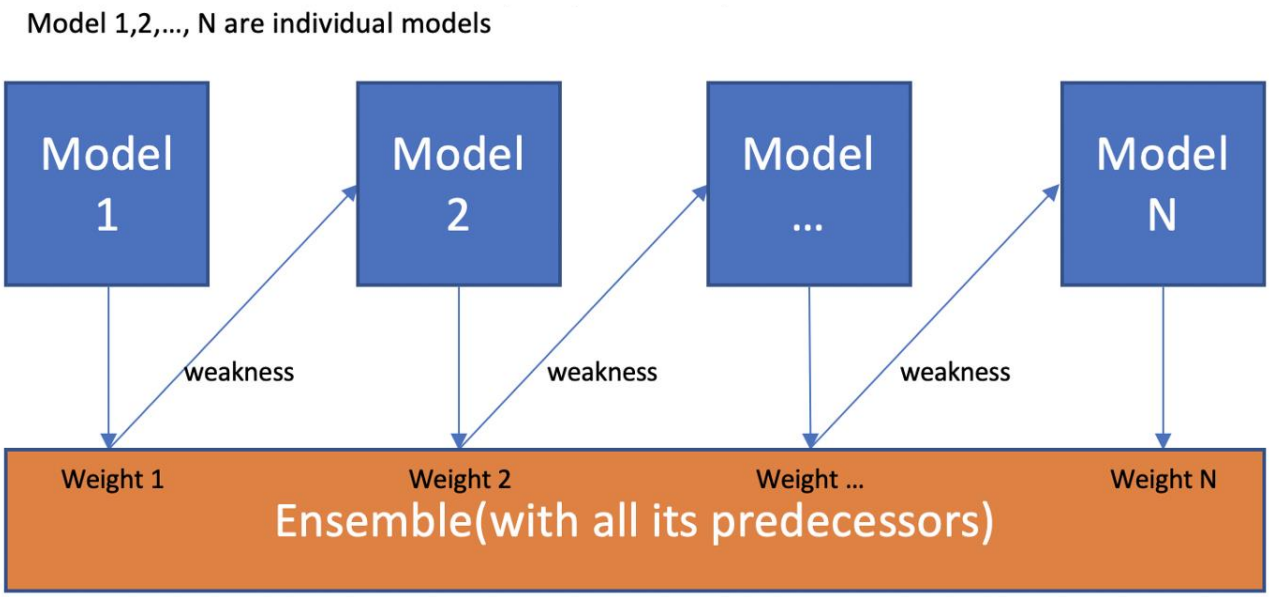
  第二种做法：**考虑使用相互有交叠的采样子集（自助采样法）。**既能保证数据量的充足，也能保证数据集之间的不同。

回顾

自助采样法 (bootstrap sampling)：给定包含 m 个样本的数据集，先随机取出一个样本放入采样集中，再把该样本放回到初始数据集，使得下次采样时该样本仍有可能被选中，这样，经过 m次随机采样操作，得到含 m 个样本的采样集，初始训练集中有的样本在采样集里多次出现，有的则从未出现.

集成学习：Boosting

- Boosting是将一系列**弱学习器**提升为一个**强学习器**的集成学习算法。首先训练一个基学习器，然后**增加被错误分类的样本的权重**，以使后续学习器更加关注这些分类错误的样本。同时**更新弱学习器的权重**。这一过程重复进行，直至达到预定数量的基学习器。



*弱学习器指的是泛化性能略优于随机猜测的学习器；例如在二分类问题上精度略高于50%的分类器；强学习器的预测结果接近真值。

集成学习： Boosting

- 训练集中的样本权重如何调整?
- 弱学习器的权重系数如何确定?

- **调整训练集样本权重**：提升那些在前一轮被错误分类的样本的权重，或同时降低被正确分类样本的权重。这样，后续的弱学习器会更加关注那些之前被错误分类的样本。随着每一轮的迭代，整个模型逐渐专注于那些更难分类的样本，实现了“分而治之”的策略。

  （例）当使用决策树作为基学习器的时候，可以使得具有更大权重的样本在计算信息增益时有更大的影响。因此，当样本的权重增加时，决策树更有可能选择能够正确分类这些样本的节点特征。

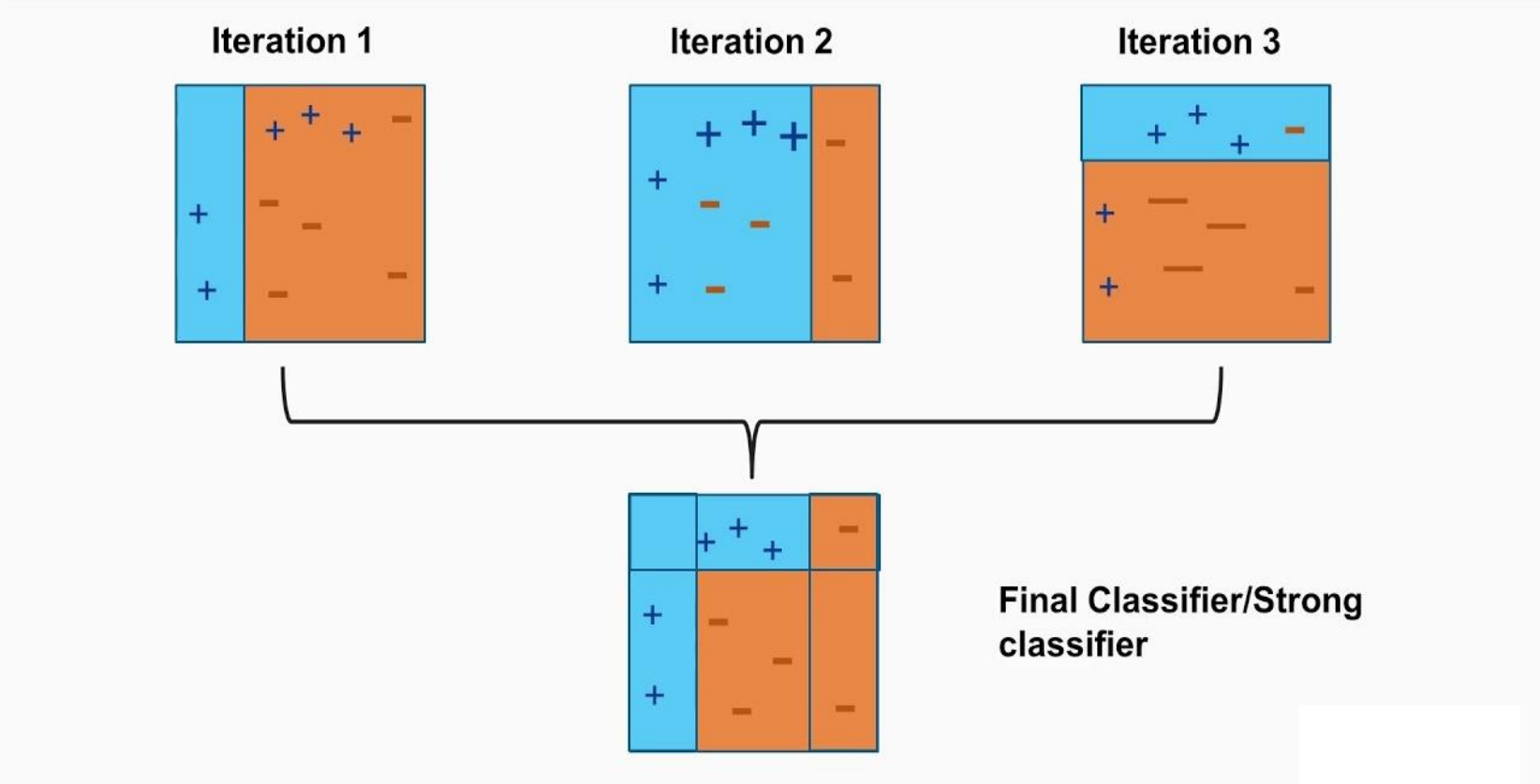
- **确定弱学习器的权重系数**：为每个弱学习器分配一个权重系数，这个系数基于学习器的分类误差率。误差率较低的弱学习器（即更准确的学习器）会获得更高的权重，使得它们在最终的决策中有更大的影响力。

  （例）在AdaBoost中，某弱学习器的权重系数 $\alpha = \frac{1}{2}\ln\left(\frac{1-\epsilon}{\epsilon}\right)$

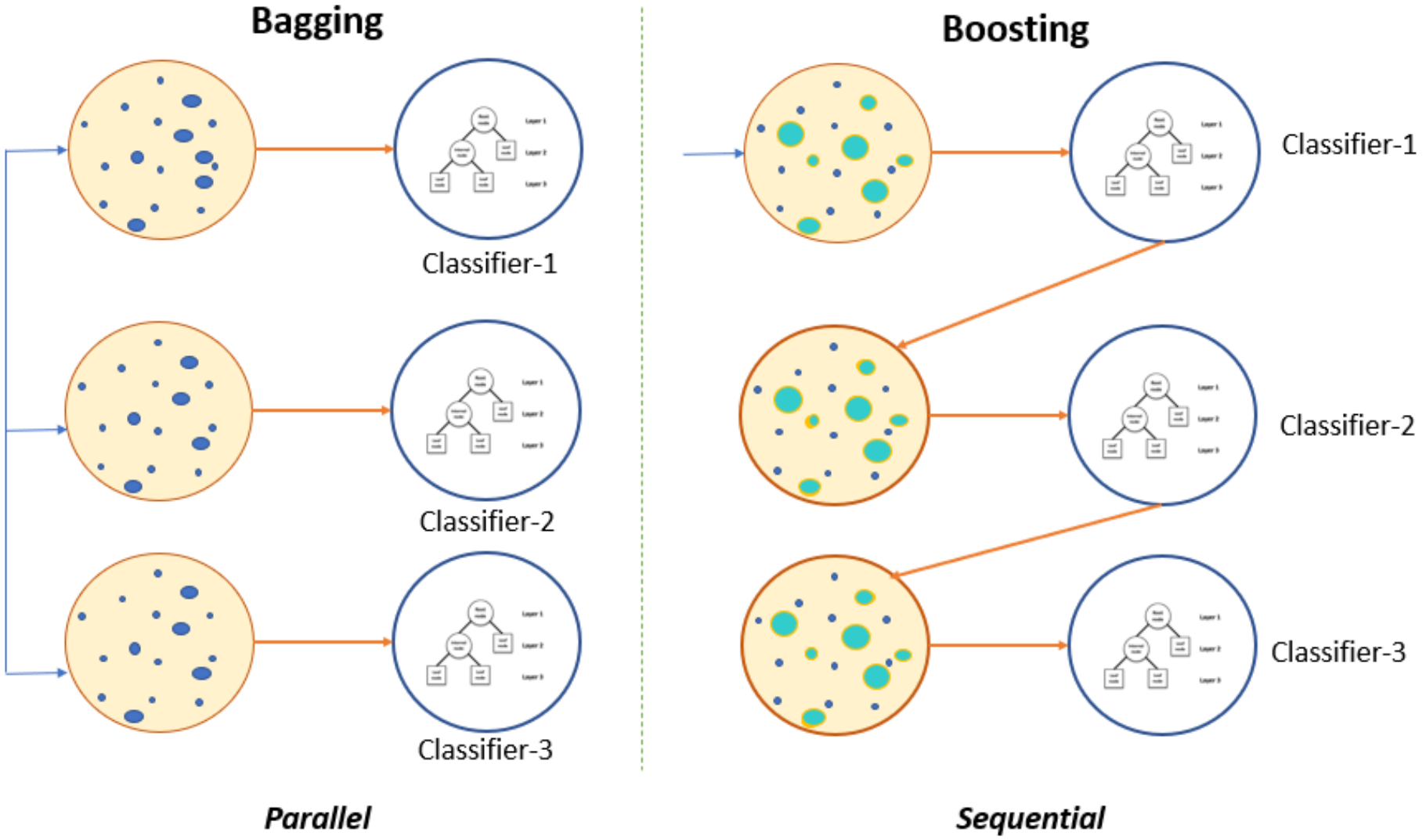
  这里，ϵ是该弱学习器的错误率

集成学习：Boosting（例）

AdaBoost方法例：



集成学习： Bagging *vs.* Boosting



模型训练时候的比较。推断?

Bagging和Boosting的对比

Bagging 的特点：

- **减少方差：**多个模型的结果将会被平均（回归问题）或者是多数投票（分类问题），这样可以减少模型因数据随机波动造成的方差。
- **防止过拟合：**单个模型可能会在训练集上学得太好，以至于学到了数据中的噪声。当多个这样的模型平均或投票时，噪声的影响会减弱，因为不太可能所有模型都在相同的噪声上过拟合。
- **并行处理：**由于每个分类器是独立的，因此可以并行训练，提高效率。

Boosting 的特点：

- **减少偏差：**通过专注于难以分类的样本，Boosting 能够创建一个强分类器，从而减少模型的偏差。
- **过拟合风险**：如果数据噪声较大，Boosting 方法可能会过分关注错误样本，从而导致过拟合。
- **计算复杂**：由于模型是顺序构建的，不能进行并行处理，这会增加训练时间。

判断题

1. Bagging 是一种只能用于分类任务的集成学习技术。
2. Boosting 方法通过训练多个弱分类器并最终结合它们的预测结果来提高模型的准确性。
3. AdaBoost算法通过增加之前分类错误的样本权重来迫使后续分类器更加关注这些样本。
4. 集成学习方法不适用于处理高维数据。
5. 在集成学习中，如果单个模型过于复杂，则整体模型一定会过拟合。
6. Boosting算法在训练过程中可以并行训练各个分类器。
7. 集成学习中使用的所有模型都必须是同一类型的模型。

判断题

1. Bagging 是一种只能用于分类任务的集成学习技术。（错误）
2. Boosting 方法通过训练多个弱分类器并最终结合它们的预测结果来提高模型的准确性。（正确）
3. AdaBoost算法通过增加之前分类错误的样本权重来迫使后续分类器更加关注这些样本。（正确）
4. 集成学习方法不适用于处理高维数据。（错误）
5. 在集成学习中，如果单个模型过于复杂，则整体模型一定会过拟合。（错误）
6. Boosting算法在训练过程中可以并行训练各个分类器。（错误）
7. 集成学习中使用的所有模型都必须是同一类型的模型。（错误）

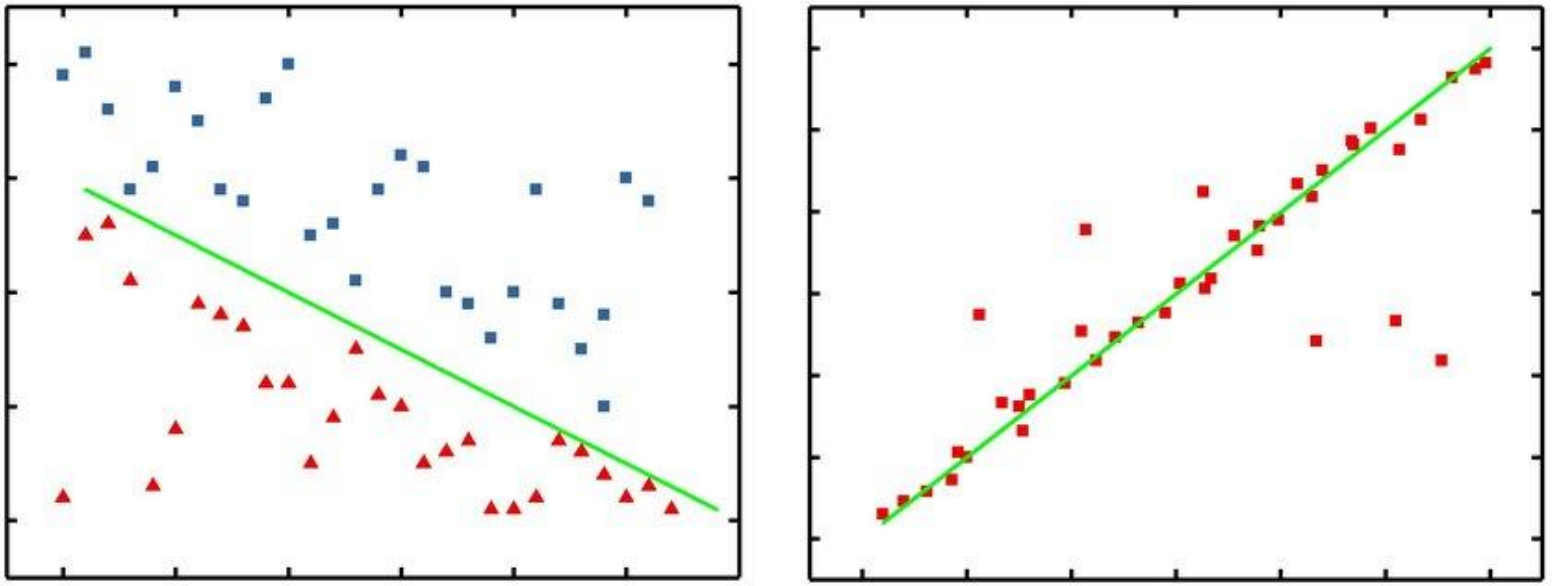
附：为什么随着集成中个体分类器数量的增加，整体的错误率可以指数级下降，理论上趋向于零。

1. **分类器的权重计算 (α_t)**：
   - 在AdaBoost算法中，每个弱分类器根据其错误率 ϵ_t 计算权重 α_t。这里，ϵ_t 是分类器在加权训练数据上的错误率，即分类错误的样本权重之和。
   - 分类器的权重由公式 $\alpha_t = \frac{1}{2} \ln \left(\frac{1-\epsilon_t}{\epsilon_t} \right)$ 计算。这意味着，如果一个分类器的错误率低（ϵ_t 小），那么它的权重 α_t 将会较大，这使得该分类器在最终的模型中有更大的影响力。
2. **数据权重更新 (D_t(i))**：
   - 在每一轮训练后，数据的权重会根据分类器的表现进行更新，以便下一个分类器能够关注那些被之前分类器错误分类的样本。
   - 权重更新公式为：$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$，其中 Z_t 是一个归一化因子，确保更新后的 D_{t+1} 仍然是一个有效的概率分布。
   - 如果样本 i 被正确分类，则 $y_i h_t(x_i)$ 为正，导致 $\exp(-\alpha_t y_i h_t(x_i))$ 变小，相应的权重 $D_{t+1}(i)$ 减小；如果样本被错误分类，则权重增加。
3. **整体模型的错误率**：
   - AdaBoost算法的目标是最小化指数损失函数，这可以推导出整体模型的错误率上界。该上界由 $\exp(-2 \sum_{t=1}^{T} (\frac{1}{2} - \epsilon_t)^2)$ 给出，表明了如果每轮弱分类器的表现都至少比随机猜测好（即 $\epsilon_t < 0.5$），那么随着分类器数量 T 的增加，整体错误率会指数级减少。

逻辑回归

逻辑回归（Logistic regression）

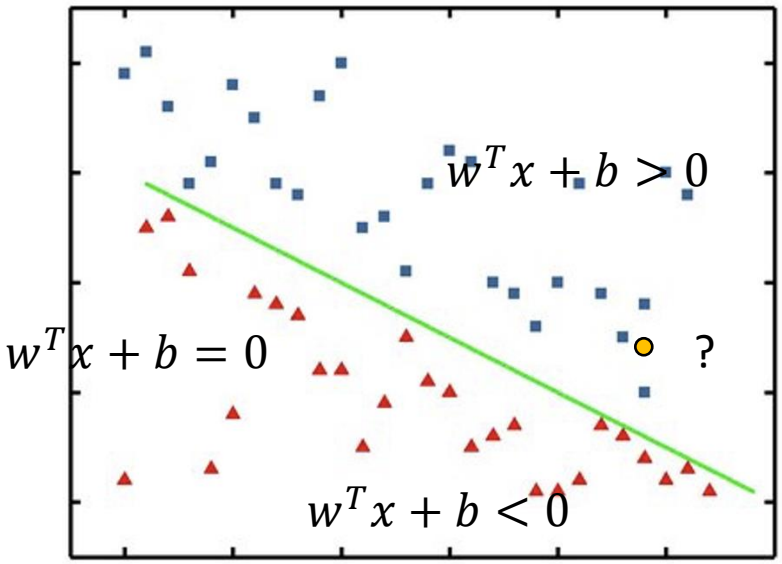
- 逻辑回归虽名为“回归”，但实际上是用于“分类”问题。“逻辑”源于“Logistic”的翻译，但实际上的logistic指的是Logistic函数（对数几率函数）。
- 逻辑回归的核心是使用Logistic函数（或称对数几率函数）来建模分类概率。因此，逻辑回归有时也被称为**对数几率回归**。
- 逻辑回归与线性回归：线性回归用线性模型预测连续的输出变量，而逻辑回归用线性模型预测二元结果（是/否，1/0）

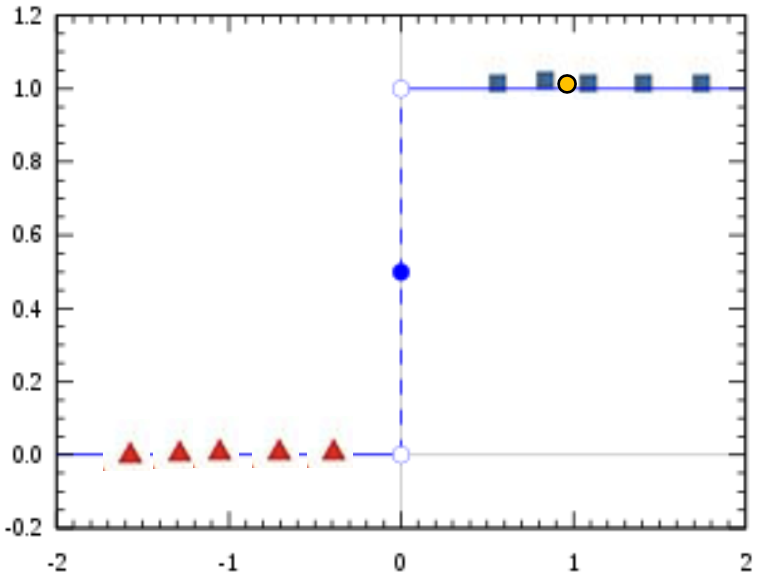


(a) Logistic Regression　　(b) Linear Regression

线性模型的二元分类

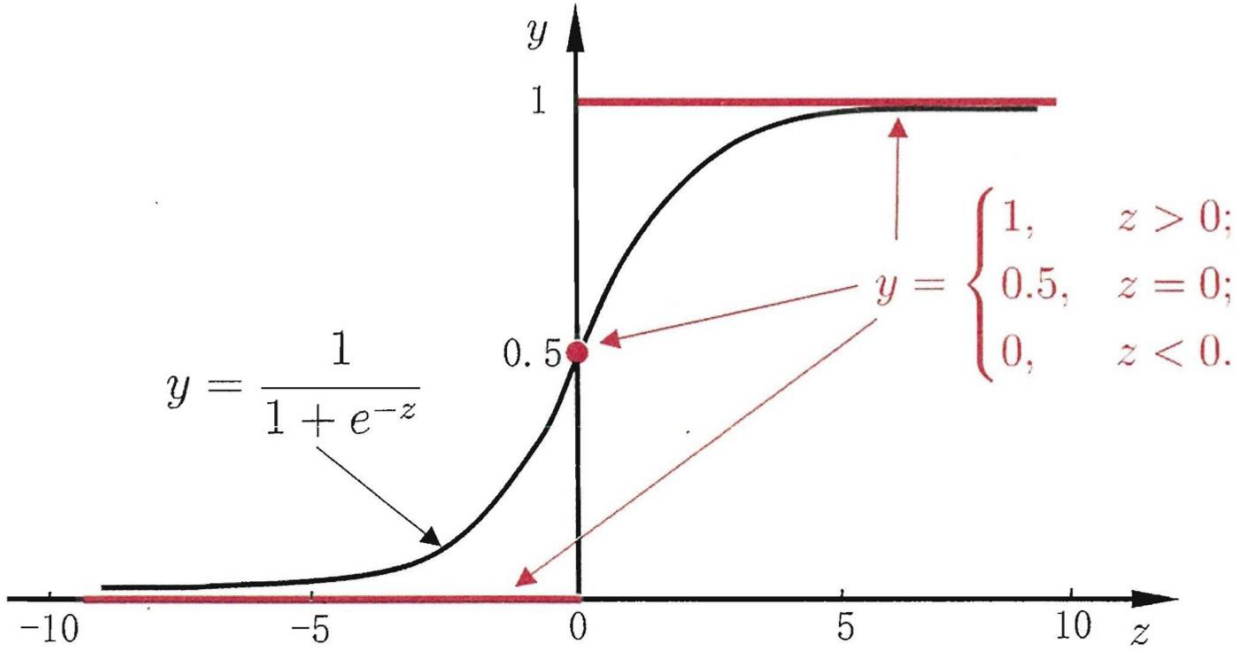
- 逻辑回归通过线性模型进行二分类呢?
  - ✓ 最简单的方式是将线性模型的输出通过单位阶跃函数映射到{0, 1}就可以解决二分类问题。
  - ✓ 这种方法中存在局限性。如果直接将线性回归模型的输出通过单位阶跃函数来映射到{0, 1}，则**缺乏概率解释**。





单位阶跃 vs.对数几率

- 将单位阶跃函数替换为对数几率函数（logistic函数）？

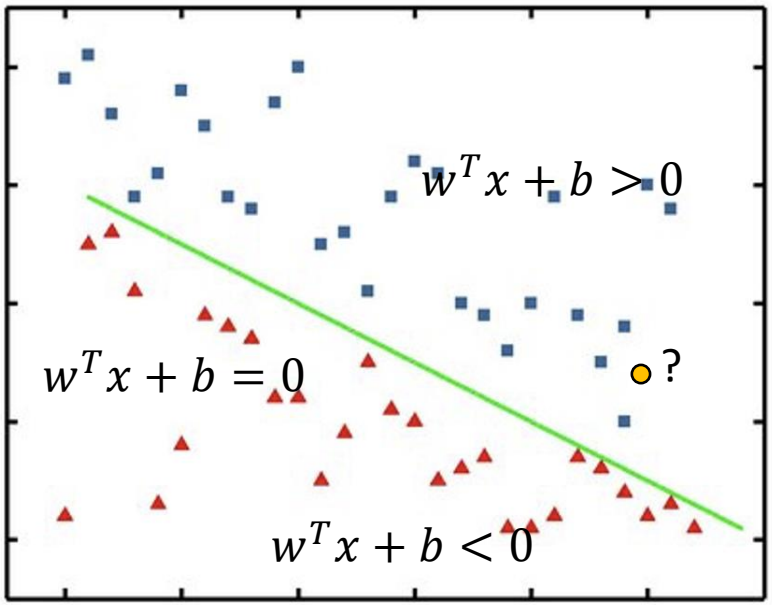


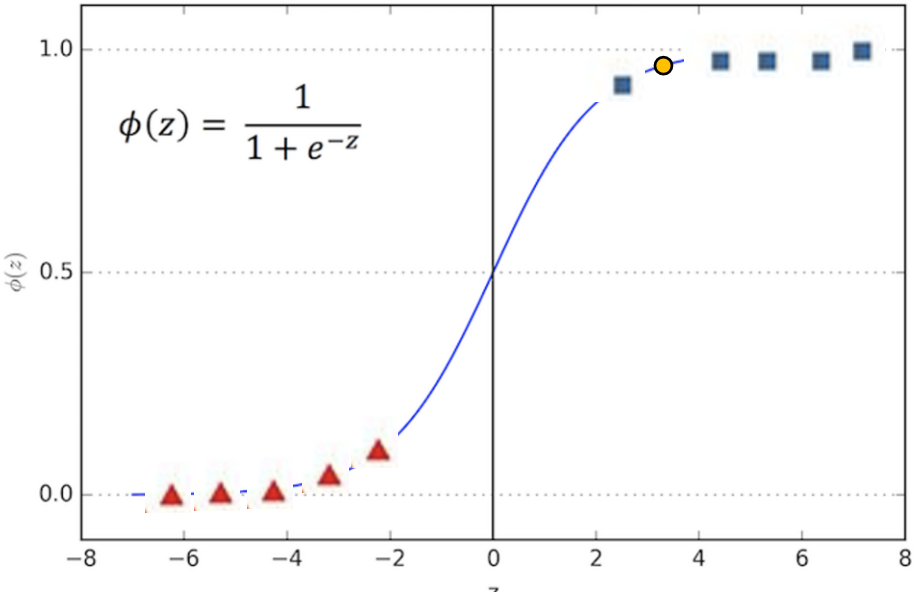
对数几率函数的特性

- logistic函数的输出始终在0和1之间，这使得它非常适合表示概率。
- 当z的值增加时，Sigmoid函数逼近1；当z的值减少时，函数值逼近0。函数在$z = 0$附近最为敏感。
- logistic函数是一个平滑的连续函数，其导数在每一点都存在。

逻辑回归（Logistic regression）

- 更为合适的解决方案：用对数几率函数代替单位阶跃函数
- 通过Logistic函数（或对数几率函数）可以将线性模型的输出映射到(0，1)区间内，这样的输出就可以解释为属于某类的概率。Logistic函数不仅将输出限制在0和1之间，还提供了平滑的概率过渡。





如何将线性回归输出转换为概率

在逻辑回归中，首先构建一个线性模型来预测一个事件：

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n$$

其中，β0, β1, …, βn是模型参数，x1, x2, …, xn是特征值。然后，将这个线性模型的输出z作为logistic函数的输入，即：

$$P(Y = 1) = \sigma(z) = \frac{1}{1+e^{-z}}$$

这里，***P*表示分类为某事件的概率**。这个转换使得模型的输出可以解释为事件发生的概率。当z值增大，概率接近1，表示这个事件更可能发生；当z值减小，概率接近0，表示这个事件不太可能发生。

参数估计

- **参数估计**的目的是为了找出逻辑回归模型中的参数（因为logistic函数没有未知的参数，所以就是找线性模型的参数）

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n$$

- **目标：**希望每个样本属于其真实标记的概率越大越好，
  如果是正类样本，那就希望预测它为正类别的概率接近1。
  如果是负类样本，那就希望预测它为负类别的概率接近1。

- **逻辑回归模型的策略**：可以计算所有样本的预测其类别的概率的乘积（**似然函数**），得到的乘积越大，则越接近于我们的目标。

通过最大化似然函数，得到线性模型的参数 ➡ 最大似然估计

参数估计

- 最大似然估计:

样本x属于事件A的概率:$p(y=1 \mid \boldsymbol{x}) = \dfrac{e^{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b}}{1+e^{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b}}$

样本x属于事件B的概率:$p(y=0 \mid \boldsymbol{x}) = \dfrac{1}{1+e^{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b}}$.

似然函数

$$L(\omega, b) = \prod_{i=1}^{m} p(y_i | x_i; \omega, b)$$

取对数，将乘积转换为求和，这在数学上更易于优化。（得到对数似然）

优化目标函数:

$$\ell(\boldsymbol{w}, b) = \sum_{i=1}^{m} \ln p(y_i \mid \boldsymbol{x}_i; \boldsymbol{w}, b)$$

参数估计

假设我们有以下数据集:

特征 X	输出 Y
1	0
2	0
3	1

我们想要使用逻辑回归模型来预测Y。

逻辑回归模型
逻辑回归模型的形式是:

$$P(Y=1|X=x)=\frac{1}{1+e^{-(\beta_0+\beta_1 x)}}$$

其中，β_0和β_1是我们要估计的模型参数。

参数估计

似然函数

对于这个数据集，似然函数L是所有观测数据在给定参数下的联合概率：

$$L(\beta_0, \beta_1) = P(Y = 0|X = 1) \times P(Y = 0|X = 2) \times P(Y = 1|X = 3)$$

代入逻辑回归模型，我们得到：

$$L(\beta_0, \beta_1) = (1 - \frac{1}{1+e^{-(\beta_0+\beta_1)}}) \times (1 - \frac{1}{1+e^{-(\beta_0+2\beta_1)}}) \times \frac{1}{1+e^{-(\beta_0+3\beta_1)}}$$

对数似然函数是：

$$\log L(\beta_0, \beta_1) = \log(1 - \frac{1}{1+e^{-(\beta_0+\beta_1)}}) + \log(1 - \frac{1}{1+e^{-(\beta_0+2\beta_1)}}) + \log(\frac{1}{1+e^{-(\beta_0+3\beta_1)}})$$

这个函数相对于β_0和β_1的最大值可以通过数值方法（比如梯度下降）求得。

课题习题

判断对错：

1.逻辑回归解决的是回归问题。

2.逻辑回归模型本质上是一个线性模型。

3.逻辑回归模型的输出值是介于0和1之间的连续值。

4.逻辑回归模型可以直接处理缺失数据。

5.标准的逻辑回归只能用于处理二分类问题。

6.逻辑回归要求自变量和因变量（事件发生概率）之间具有线性关系。

课题习题答案

1.逻辑回归解决的是回归问题。（错误）

2.逻辑回归模型本质上是一个线性模型。（正确，逻辑回归尽管预测的是概率，但其核心是线性的，因为它使用的是自变量的线性组合来预测。

3.逻辑回归模型的输出值是介于0和1之间的连续值。（正确，逻辑回归通过sigmoid函数）

4.逻辑回归模型可以直接处理缺失数据。（错误，没有像随机森林这种数据随机选取，节点特征随机选取的机制，数据缺失，就无法进行计算）

5.标准的逻辑回归只能用于处理二分类问题。（正确）

6.逻辑回归要求自变量和因变量（事件发生概率）之间具有线性关系。（错误）