机器学习

Machine Learning

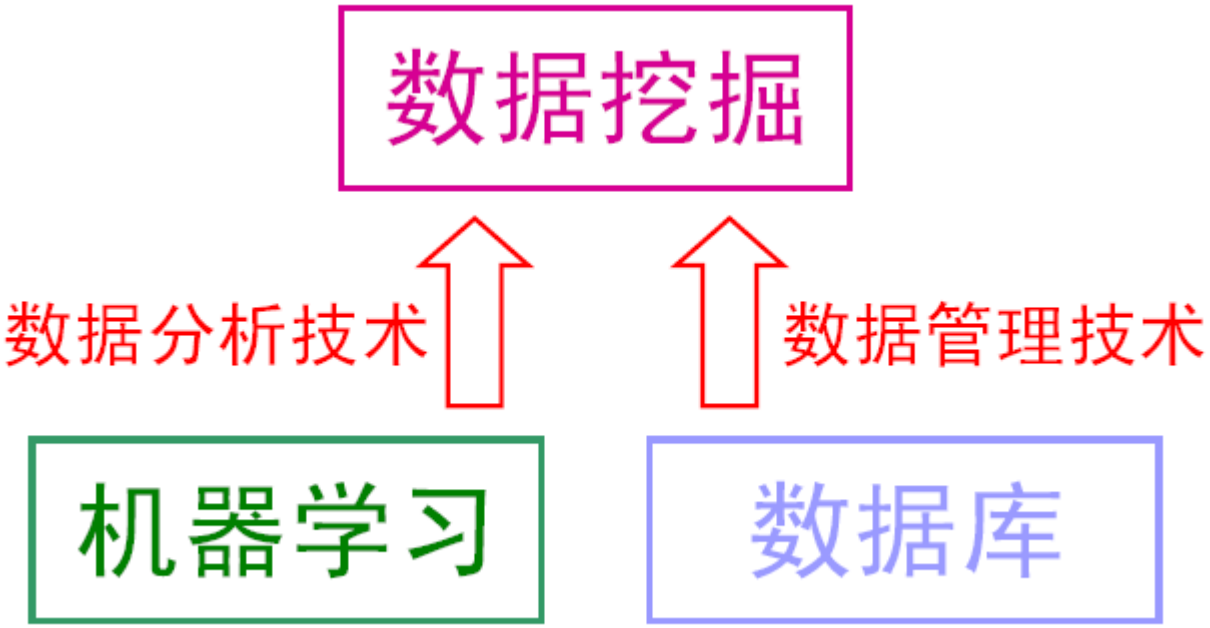
回顾
Review

- 什么是机器学习?
- 机器学习三要素?
- 过拟合/欠拟合及其特点？
- 监督学习与非监督学习的区别?
- 数据挖掘与机器学习的关系?

思考：机器学习与数据挖掘的关系？

- 机器学习是数据挖掘的重要工具。
- 数据挖掘不仅仅要研究、拓展、应用一些机器学习方法，还要通过许多非机器学习技术解决数据仓储、大规模数据、数据噪音等等更为实际的问题。
- 机器学习的涉及面更宽，常用在数据挖掘上的方法通常只是“从数据学习”，然则机器学习不仅仅可以用在数据挖掘上，一些机器学习的子领域甚至与数据挖掘关系不大，例如强化学习与自动控制等等。
- 数据挖掘试图从海量数据中找出有用的知识。
- 大体上看，数据挖掘可以视为机器学习和数据库的交叉，它主要利用**机器学习界提供的技术来分析海量数据，利用数据库界提供的技术来管理海量数据**。

思考：机器学习与数据挖掘的关系？



机器学习
Machine Learning

- 正则化
- 特征工程
- 模型的评估
- 模型的优化
- 参数问题

什么是正则化

- 正则化（Regularization）是机器学习和统计建模中的一种技术，用于防止模型在训练数据上过拟合。过拟合发生在模型过于复杂，以至于它不仅学习到了数据中的实际模式，还学习到了数据中的噪声。正则化通过引入额外的约束、惩罚或干扰，来限制模型的复杂性，从而提高模型在新数据上的泛化能力。所有对抗过拟合，干扰优化的方法都是正则化
  - 约束目标函数：在目标函数中增加模型参数的正则化项
  - 约束模型结构：对模型的结构进行约束，如Dropout
  - 数据增强：通过对样本集中的样本进行额外的操作（通常是加入随机噪声），增加样本集的数据量，提高训练模型的鲁棒性，减少过拟合的风险。
  - 约束优化过程：在优化过程中施加额外步骤干扰，如Early Stop等

约束目标函数方法

- L1 正则化
  - 原理：L1 正则化通过将模型的参数绝对值的和添加到损失函数中来惩罚模型的复杂性。L1正则化具有稀疏性，它在优化过程中倾向于将一些参数收缩为零，从而实现特征选择。

$$w^* = \arg\min_w \frac{1}{N}\sum_{n=1}^{N} L(y_n, f(x_n); w) + \lambda||w||_1$$

其中，$\|w\|_1 = \sum_i |w_i|$ ，w是参数向量

约束目标函数方法

- L2 正则化
  - 原理：L2 正则化通过将模型的参数平方和添加到损失函数中来惩罚模型的复杂性。它会将参数值收缩但不会将它们变为零。

$$w^* = \arg\min_w \frac{1}{N}\sum_{n=1}^{N} L(y_n, f(x_n); w) + \lambda||w||_2^2$$

其中，$\|w\|_2^2 = \sum_i^n w_i^2$ ，w是参数向量

约束目标函数方法

- L1/L2 正则化
  - 原理：L1/L2 正则化是 L1 和 L2 正则化的结合，结合了两者的优点。它可以同时进行特征选择和参数收缩。

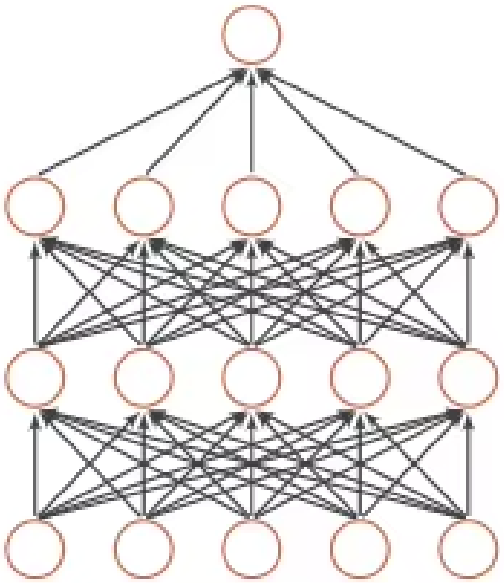
$$w^* = \arg\min_w \frac{1}{N} \sum_{n=1}^{N} L(y_n, f(x_n); w) + \lambda_2 ||w||_2^2 + \lambda_1 ||w||_1$$

- 其他方法

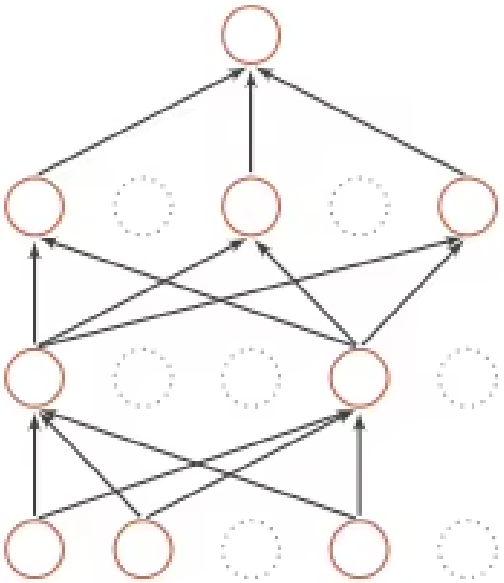
  谱正则化, 自正交性正则化, WEISSI正则化, 梯度惩罚等

约束模型结构的方法

- Dropout
  - 原理：Dropout 是一种用于神经网络的正则化方法，它在每次训练迭代中随机丢弃（即忽略）部分神经元。这样可以防止神经网络过度依赖某些特定的神经元，从而提高泛化能力



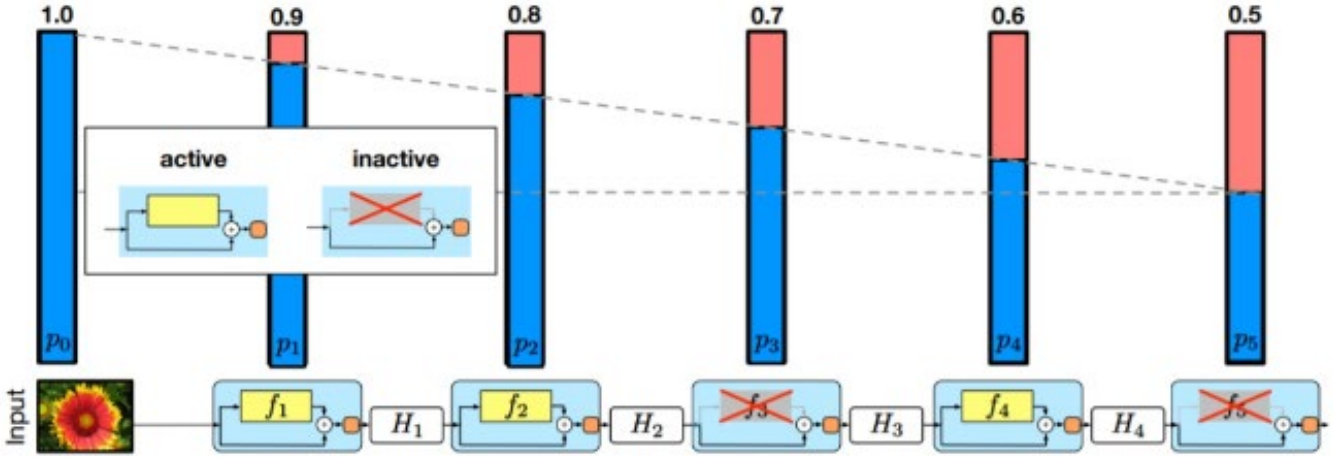
(a) 标准网络



(b) 应用丢弃法后的网络

约束模型结构的方法

- 随机深度 Stochastic Depth
  - 原理：也是神经网络中使用的技术，指在训练时以一定概率丢弃网络中的模块（令其等价于恒等变换）；测试时使用完整的网络，并且按照丢弃概率对各个模块的输出进行加权

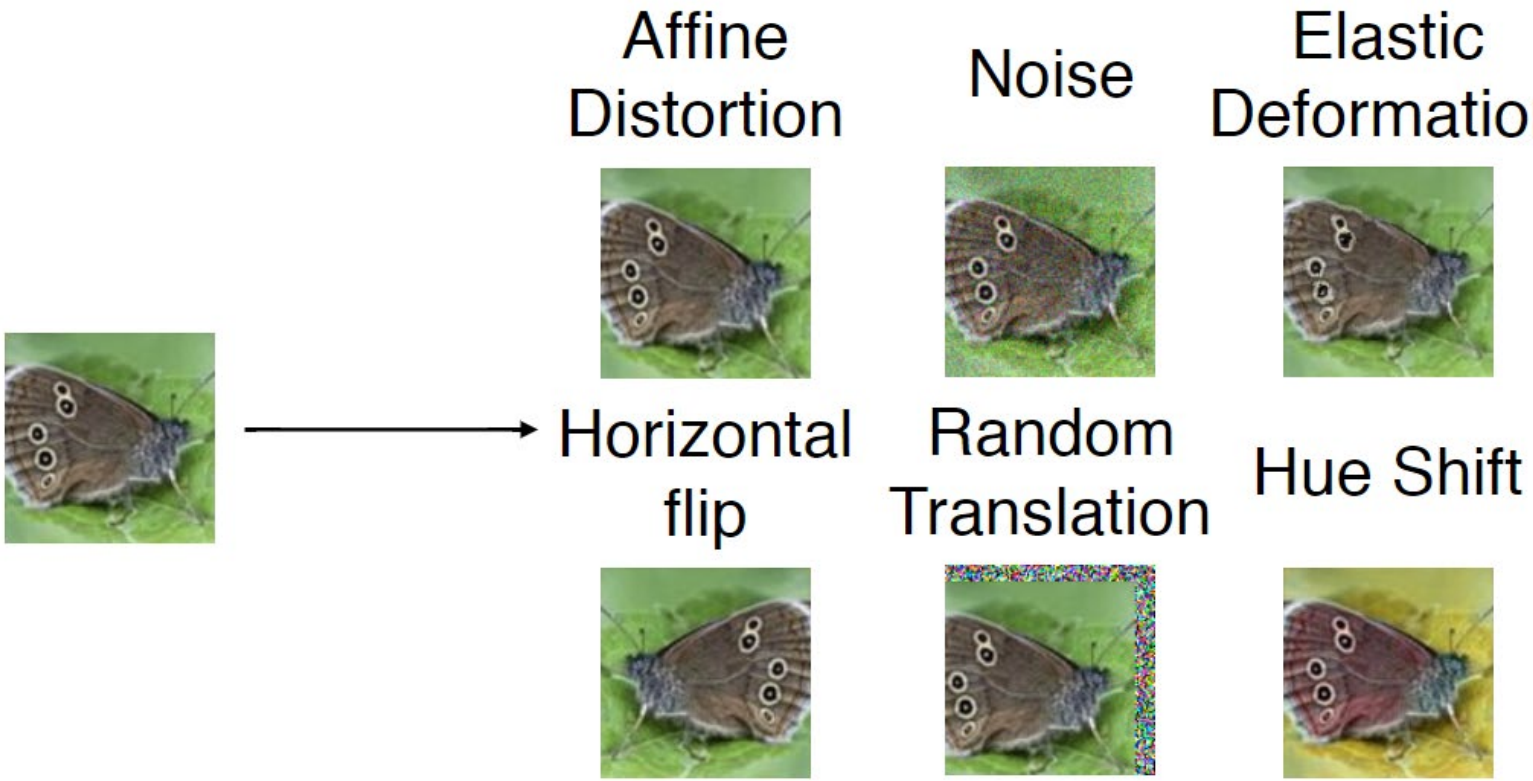


数据增强

- 图像数据的增强主要是通过算法对图像进行转变，引入噪声等方法来增加数据的多样性。
- 旋转（Rotation）：将图像按顺时针或逆时针方向随机旋转一定角度；
- 翻转（Flip）：将图像沿水平或垂直方法随机翻转一定角度；
- 缩放（Zoom In/Out）：将图像放大或缩小一定比例；
- 平移（Shift）：将图像沿水平或垂直方法平移一定步长；
- 加噪声（Noise）：加入随机噪声

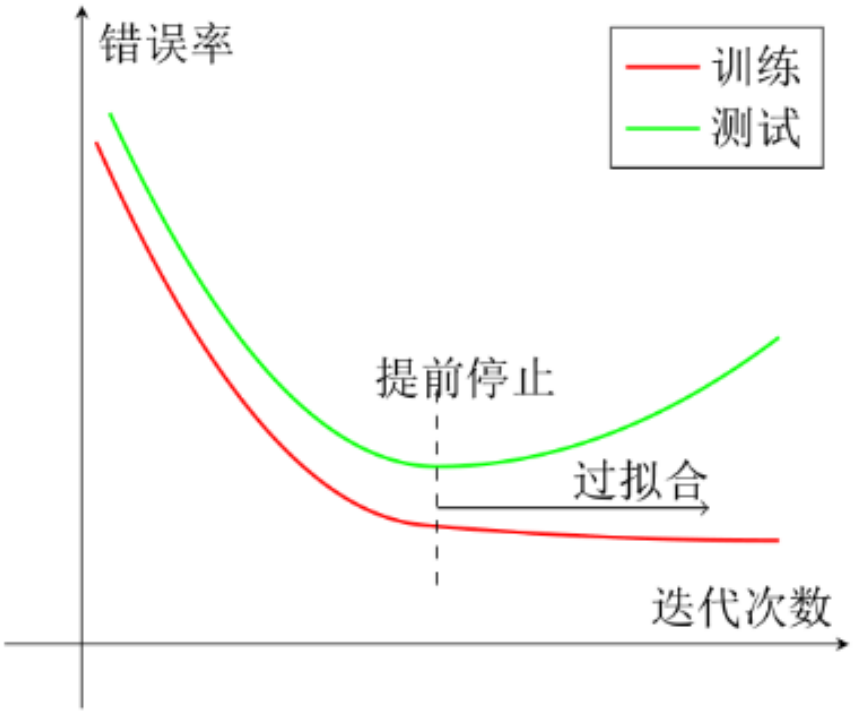
数据增强

- 图像数据的增强主要是通过算法对图像进行转变，引入噪声等方法来增加数据的多样性。



约束优化过程的方法

- Early Stopping
  - 原理：指训练时当观察到验证集上的错误不再下降，就提前停止迭代



正则化方法

- 在应用正则化时，需要考虑以下几个注意事项：
  - 选择适当的正则化方法：不同正则化方法适用于不同类型的模型和数据。选择与模型和任务匹配的方法可以有效提升性能。例如，L1正则化适用于特征选择、Dropout适合深度学习模型
  - 平衡正则化强度：过强的正则化可能会使模型过于简单，导致欠拟合；而正则化过弱则可能无法有效防止过拟合。需要找到一个平衡点。
  - 监控训练过程：通过监控训练和验证损失，确保正则化能够有效地减少过拟合

机器学习
Machine Learning

- 正则化
- 特征工程
- 模型的评估
- 模型的优化
- 超参数

特征工程

- **没有高质量的数据，就没有高质量的挖掘结果**
  - 高质量的数据意味着我们能提取到高质量的特征。
  - 特征工程是指在机器学习过程中，从原始数据中提取、选择和转换特征的过程，其目的是提高模型的性能。
    - 特征理解
    - 特征增强
    - 特征选择与转换

特征工程

- **特征理解**
  - 在拿到数据的时候，第一步需要做的是理解它，这便是特征理解。

等级	属性	案例	描述性统计	可视化
定类	离散 无序	血型（A / B / O / AB型）、性别（男 / 女）、货币（人民币 / 美元 / 日元）	频率 / 占比 / 众数	条形图、饼图
定序	有序 比较	期末绩点（A、B、C、D、E、F）、问卷答案（非常满意、满意、一般、不满意）	频率 / 众数 / 中位数 / 百分位数	条形图、饼图、箱形图
定距	数值差别	温度	频率 / 众数 / 中位数 / 均值 / 标准差	条形图、饼图、箱形图、直方图
定比	连续 存在绝对零点	收入 重量	均值 / 标准差	同上

特征工程

- **特征理解**

特征 ↑　　标记 ↑

编号	色泽	根蒂	敲声	好瓜
1	青绿	蜷缩	浊响	是
2	乌黑	蜷缩	沉闷	是
3	青绿	硬挺	清脆	否
4	乌黑	稍蜷	沉闷	否

训练集←

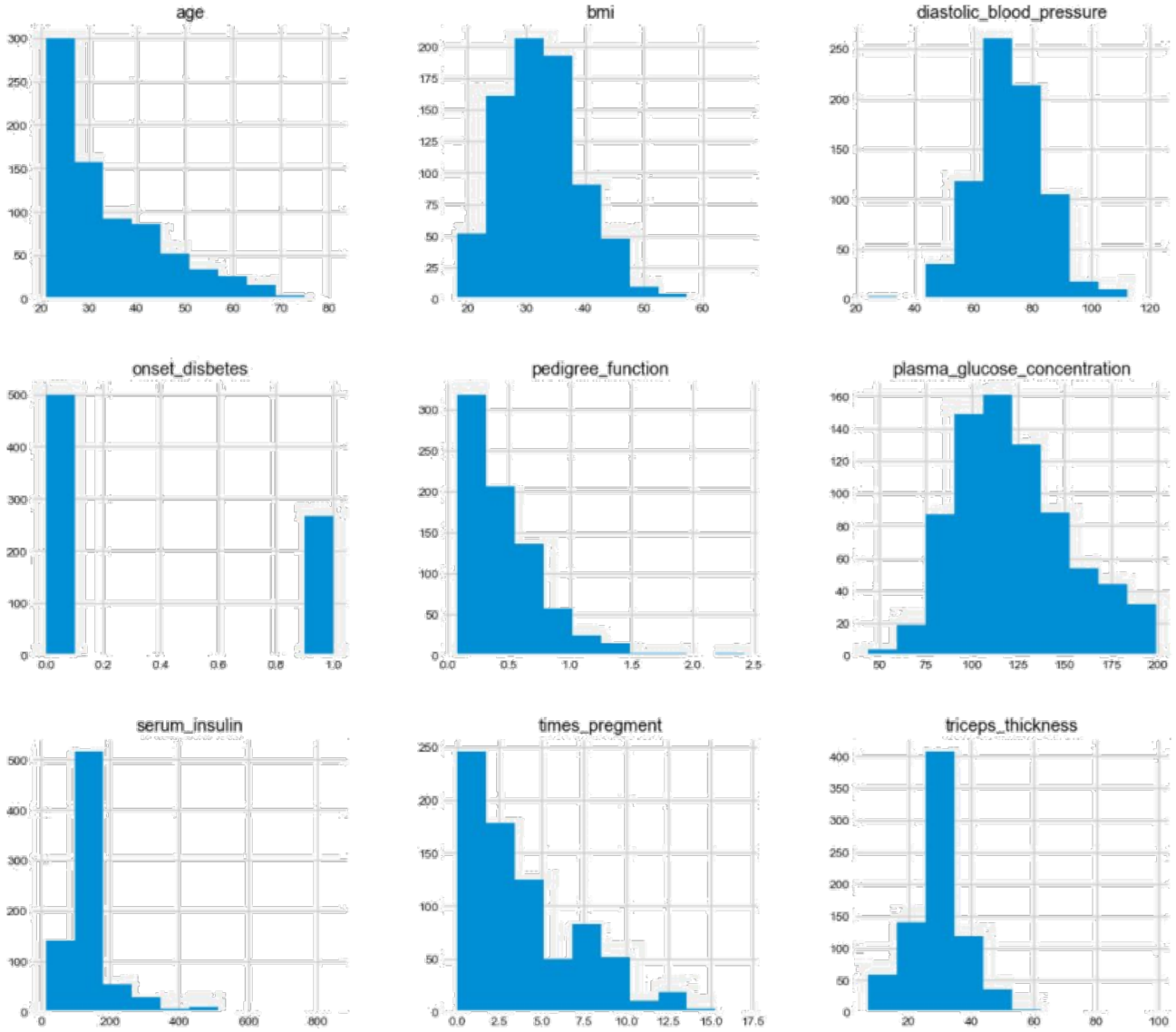
1	青绿	蜷缩	沉闷	?

测试集←

特征工程

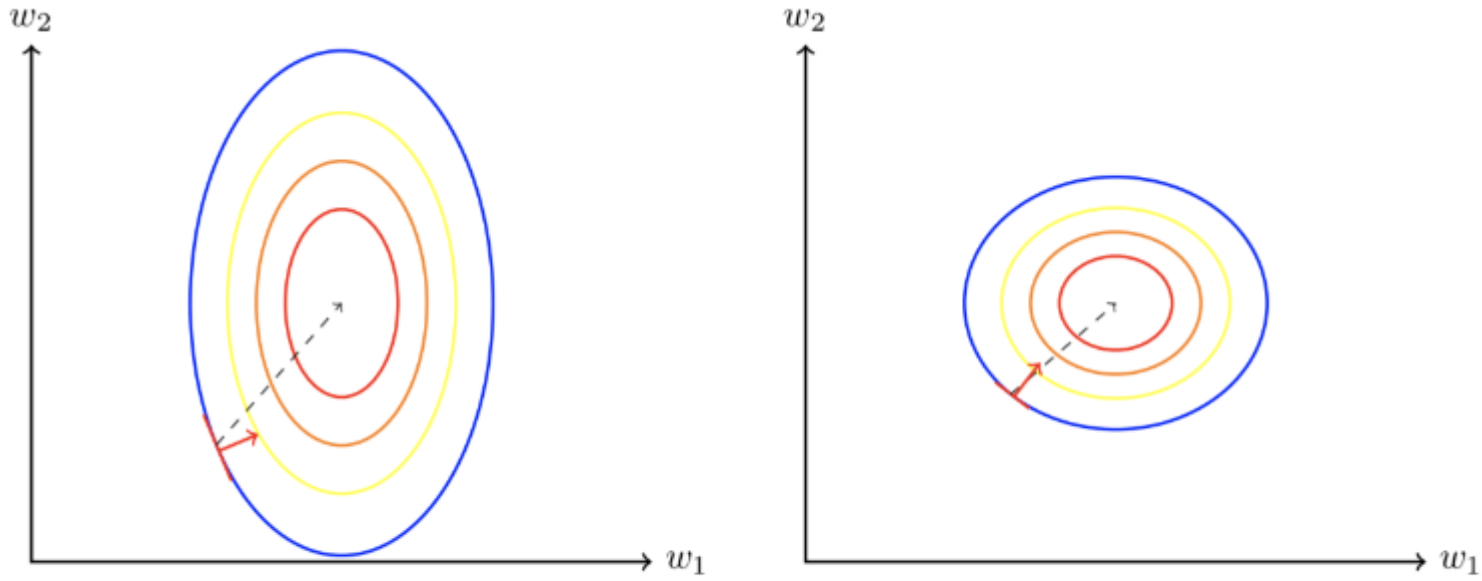
- **特征增强**
  - 这一步其实就是数据清洗和预处理了，比如清除空值、日期转换、标准化和归一化等。
- 为什么要做数据规范化：一些数据挖掘方法，需要对数据进行标准化以获得最佳的效果。
  - 例如，对于分类算法，如涉及神经网络的算法或诸如最临近分类和聚类的距离度量分类算法，都需要将训练样本属性度量输入值规范化，这样有助于加快学习阶段的速度。
  - 对于基于距离的方法，规范化可以帮助防止具有较大初始值域的属性与具有较小初始值域的属性相比，权重过大。

特征工程



查看特征分布，从图中可以看出一个问题，每个特征之间的量纲都是不一样的，这样的问题对于KNN这种基于距离的模型来说是一个致命的漏洞

特征工程



(a) 未归一化数据的梯度　　(b) 归一化数据的梯度

归一化可以显著加速梯度下降算法的收敛速度,提高稳定性和计算效率。

特征工程

- **特征选择和变换**
  - 特征选择和变换的出现是在数据经过多种创新特征的方法后，依然想要获取更多维度的信息。但是，添加更多特征会使所有模型变得更加复杂，从而增大过拟合的可能性。在添加新特征或处理一般的高维数据集时，最好将特征的数量减少到只包含最有用的那些特征，并删除其余特征。这样会得到泛化能力更好、更简单的模型。于是为了判断每个特征的作用有多大，我们需要将数据划分为训练集和测试集，并只在训练集上拟合特征选择。

特征工程

- **特征选择和变换**
  - 过滤方法（Filter Methods）：通过统计特征与目标变量的关系来选择特征，例如卡方检验、信息增益、相关系数等。这些方法通常不依赖于模型训练，计算效率高，但可能忽视特征之间的相互关系。
  - 包裹方法（Wrapper Methods）：将特征选择视为一个搜索问题，通过训练模型并评估模型性能来选择特征。例如，递归特征消除（RFE）就是一种包裹方法。它的优点是考虑了特征间的相互作用，但计算成本较高。
  - 嵌入方法（Embedded Methods）：将特征选择过程融入模型训练中。例如，Lasso回归（L1正则化）可以自动进行特征选择。嵌入方法结合了过滤方法和包裹方法的优点，但可能依赖于特定模型。

特征工程

- **特征选择和变换**
  - 主成分分析（PCA）：一种降维技术，通过线性变换将数据投影到主成分上，选择能够解释数据方差的大部分主成分。PCA并不是直接的特征选择方法，但它能通过提取重要的主成分来减少特征数量。
  - 线性判别分析（LDA）：与PCA类似，LDA用于降维，但它通过最大化类别间的分离度来选择特征，特别适用于分类问题。
  - 基于模型的方法：使用某些模型的特征重要性度量来选择特征，如随机森林、梯度提升树等。这些模型内置的特征重要性评估可以帮助选择关键特征。

机器学习
Machine Learning

- 正则化
- 特征工程
- **模型的评估**
- 模型的优化
- 超参数

模型的评估

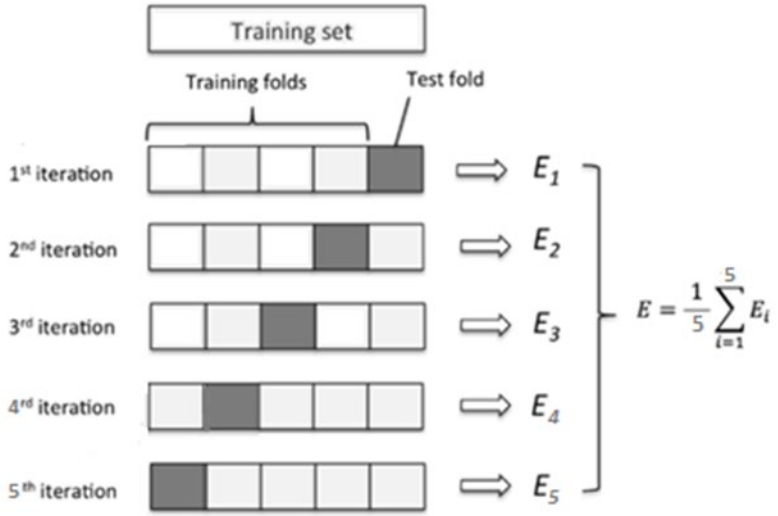
- 模型评估是机器学习和数据挖掘中一个至关重要的过程，用于衡量和验证模型的性能。
  - 一方面，帮助我们从模型的假设空间中选择最佳模型（测试集）
  - 另一方面，帮助我们了解模型在未见数据上的表现，确保模型的泛化能力和有效性（验证集）
    - 交叉验证
    - 模型评价指标
    - 偏差-方差权衡（Bias-Variance Tradeoff）

模型的评估

- **交叉验证**
  - 主要用于防止模型过于复杂而引起的过拟合，是一种评价数据集泛化能力的统计方法。其基本思想是将原始数据进行划分，分成训练集（train_set）和测试集（test_set），训练集用来对模型进行训练，测试集用来测试训练得到的模型，以此作为模型的评价指标。
  - k折交叉验证（k-fold cross-validation）
  - 分层k折交叉验证（Stratified k-fold cross validation）
  - 留一交叉验证（Leave one out Cross-validation）
  - 打乱划分交叉验证（shuffle-split cross-validation）

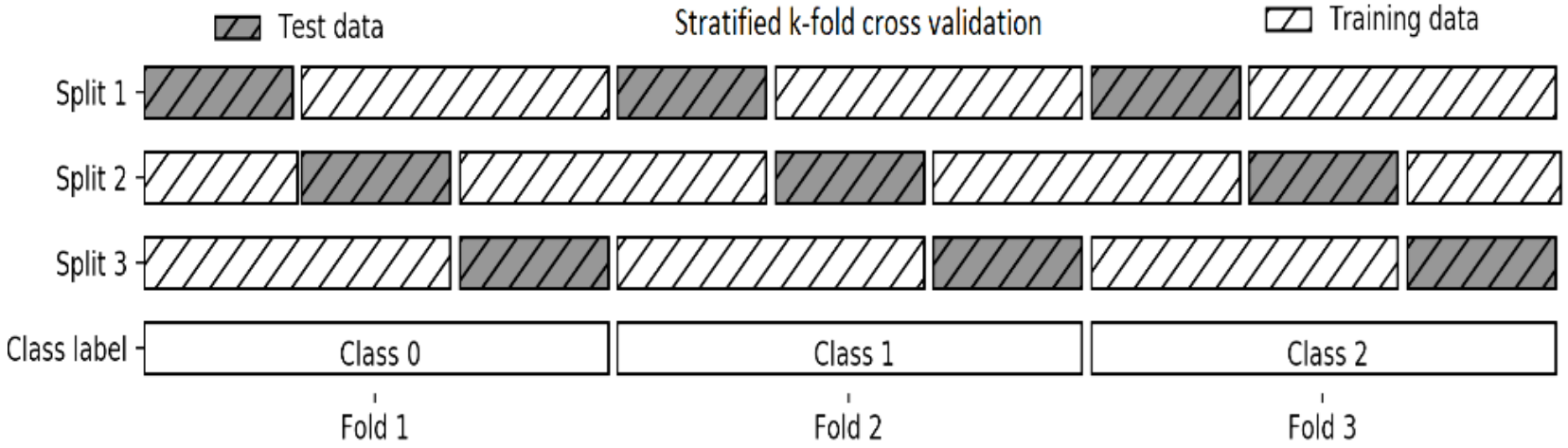
模型的评估

- **k折交叉验证（k-fold cross-validation）**
  - 将数据集等比例划分成K份，每一部分叫作折（fold）。以其中的一份作为测试集，其他的K-1份数据作为训练集，这样就完成了一次验证。因此，K折交叉验证只有实验K次才算完整的完成，也就是说交叉验证实际是把验证重复做了K次，每次验证都是从K个部分选取一份不同的部分作为测试集（从而保证K个部分的数据都分别做过测试集），剩下的K-1个部分当作训练集，最后取K次准确率的平均值作为最终模型的评价指标。



模型的评估

- **分层k折交叉验证(Stratified k-fold cross validation)**
  - 同样属于交叉验证类型，分层的意思是说在每一折中都保持着原始数据中各个类别的比例关系，比如说：原始数据有3类，比例为1:2:3，采用3折分层交叉验证，那么划分的3折中，每一折中的数据类别保持着1:2:3的比例，因为这样的验证结果更加可信，如图所示：



模型的评估

- **留一交叉验证（Leave one out Cross-validation）**
  - 是一种特殊的交叉验证方式。顾名思义，如果样本容量为n，则k=n，进行n折交叉验证，每次留下一个样本进行验证。由于每一折中几乎所有的样本皆用于训练模型，因此最接近原始样本的分布，这样评估所得的结果比较可靠。但其缺点也很显然，就是比较耗时，因此适合于数据集比较小的场合

- **打乱划分交叉验证（shuffle-split cross-validation）**
  - 是一种非常灵活的交叉验证。该方法控制更为灵活，可以控制每次划分时训练集测和试集的比例(通过train_size和test_size来控制），以及划分迭代次数(通过n_splits来控制）。这种灵活的控制，甚至可以存在有的数据既不在训练集也不在测试集的情况。

模型的评估

模型评价指标

- 对于一个模型来说，如何评价该模型的好坏，针对不同的问题需要不同的模型评价标准，这是机器学习中的一个关键性的问题。具体来讲，评价指标有两个作用，其一是了解模型的泛化能力，可以通过同一个指标来对比不同模型，从而知道哪个模型相对较好；其二是可以通过这个指标来逐步优化我们的模型。因此，在选择模型与调参时，选择正确的指标是很重要的。
- 分类任务评价指标：准确率(Accuracy)、精确率(Precision)、召回率(Recall)、F1分数(F1 Score)、ROC曲线和AUC
- 回归任务评价指标：均方误差(MSE)、均方根误差(RMSE)、平均绝对误差（MAE）、决定系数(R-squared)

模型的评估

- **分类任务的评价**
  - **误分类：**误分类是指将被调查对象的特征错误地分到原本不属于它的类别中
  - 假阳性（false positive）属于第一类错误（type I error）
  - 假阴性（false negative）属于第二类错误（type II error）
  - 所有的分类器都存在偏好，因此都存在误分类的现象
  - 一个好的模型应该尽量减少第一类错误和第二类错误。可以通过调整分类器的阈值来平衡这些不同类型的错误，如何平衡和优化这两种错误取决于具体应用的需求和场景，这样我们的模型才有实际的应用价值

模型的评估

分类任务的评价

- **混淆矩阵（confusionmatrix）：**用于评估分类模型性能的工具，用n行n列的矩阵形式来表示。混淆矩阵是总结分类模型预测结果的情形分析表，以矩阵形式将数据集中的记录按照真实的类别与分类模型预测的类别进行汇总。其中矩阵的行表示真实值，矩阵的列表示预测值。下图以二分类问题为例展示混淆矩阵

	预测为正类 (Positive)	预测为负类 (Negative)
实际为正类 (Positive)	真阳性 (True Positive, TP)	假阴性 (False Negative, FN)
实际为负类 (Negative)	假阳性 (False Positive, FP)	真阴性 (True Negative, TN)

真阳性 (TP)：实际为正类的样本被正确预测为正类。
假阴性 (FN)：实际为正类的样本被错误预测为负类。
假阳性 (FP)：实际为负类的样本被错误预测为正类。
真阴性 (TN)：实际为负类的样本被正确预测为负类。

模型的评估

分类任务的评价

	预测为正类 (Positive)	预测为负类 (Negative)
实际为正类 (Positive)	真阳性 (True Positive, TP)	假阴性 (False Negative, FN)
实际为负类 (Negative)	假阳性 (False Positive, FP)	真阴性 (True Negative, TN)

- 准确率 (Accuracy)：所有预测中正确预测的比例

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- 精确率 (Precision)：预测为正类的样本中，实际为正类的比例，也叫查准率

$$\text{Precision} = \frac{TP}{TP + FP}$$

模型的评估

- 分类任务的评价

	预测为正类 (Positive)	预测为负类 (Negative)
实际为正类 (Positive)	真阳性 (True Positive, TP)	假阴性 (False Negative, FN)
实际为负类 (Negative)	假阳性 (False Positive, FP)	真阴性 (True Negative, TN)

  - 召回率 (Recall)：所有实际为正类的样本中，被正确预测为正类的比例。也叫敏感性（sensitivity），又叫查全率。顾名思义，“查全”表明预测为真覆盖到了多少实际为真的样本，换句话说遗漏了多少

$$\text{Recall} = \frac{TP}{TP + FN}$$

模型的评估

分类任务的评价

	预测为正类 (Positive)	预测为负类 (Negative)
实际为正类 (Positive)	真阳性 (True Positive, TP)	假阴性 (False Negative, FN)
实际为负类 (Negative)	假阳性 (False Positive, FP)	真阴性 (True Negative, TN)

- F1分数 (F1 Score)：精确率和召回率的调和平均数，是用来衡量分类模型综合性能的一种指标。它同时兼顾了分类模型的准确率和召回率。它的最大值是1，最小值是0。

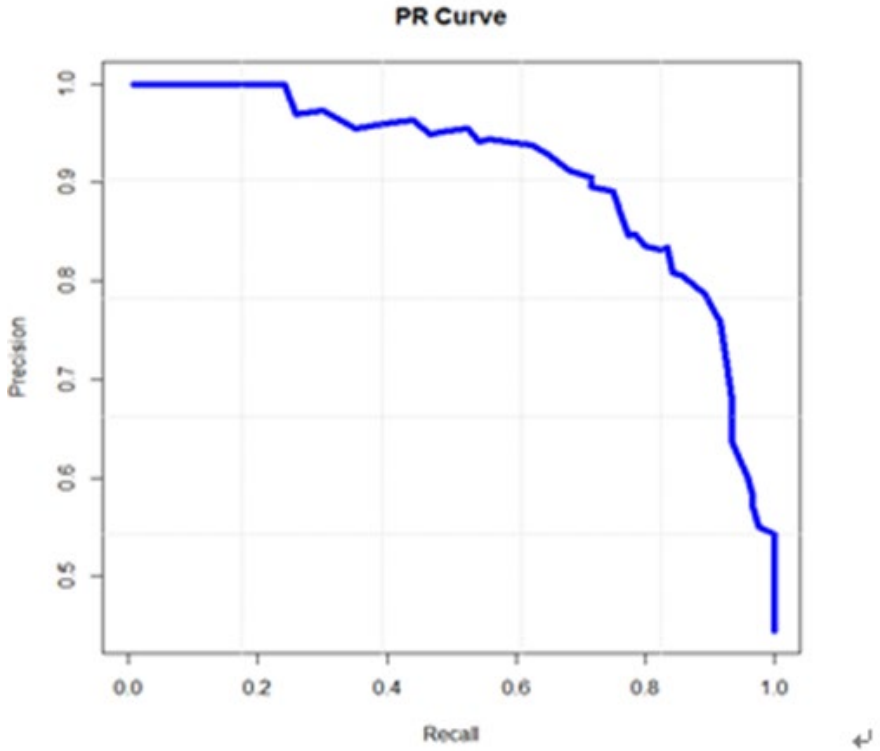
$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- 调和平均数的性质是，当精确率和召回率二者都非常高的时候，它们的调和平均才会高。如果其中之一很低，调和平均就会被拉得接近于很低的那个数

模型的评估

分类任务的评价

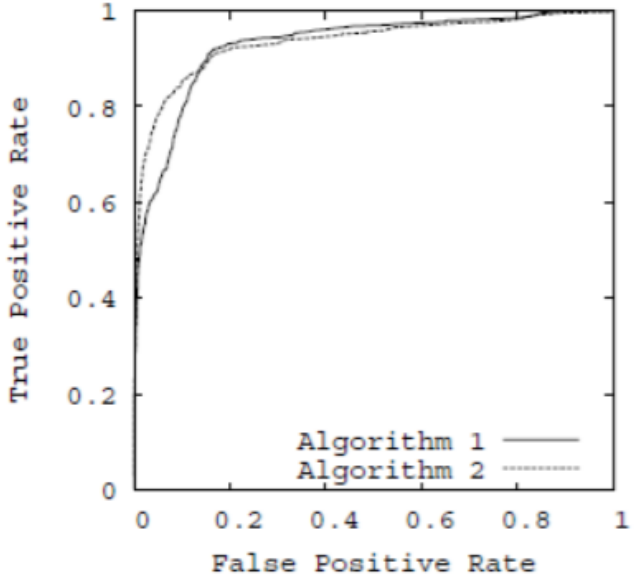
- 精确率-召回率曲线（P-R曲线）：也叫查准率-查全率曲线，是一对相互矛盾的性能指标，而事实上我们所期望的是既能保证查准率，又能提升查全率。对分类问题来讲，通过不断调整分类器的阈值，可以得到不同的Precision-Recall值，遍历所有可能的阈值，从而可以得到一条曲线。通常随着分类阈值从大到小变化，查准率减小，而查全率增加。



模型的评估

分类任务的评价

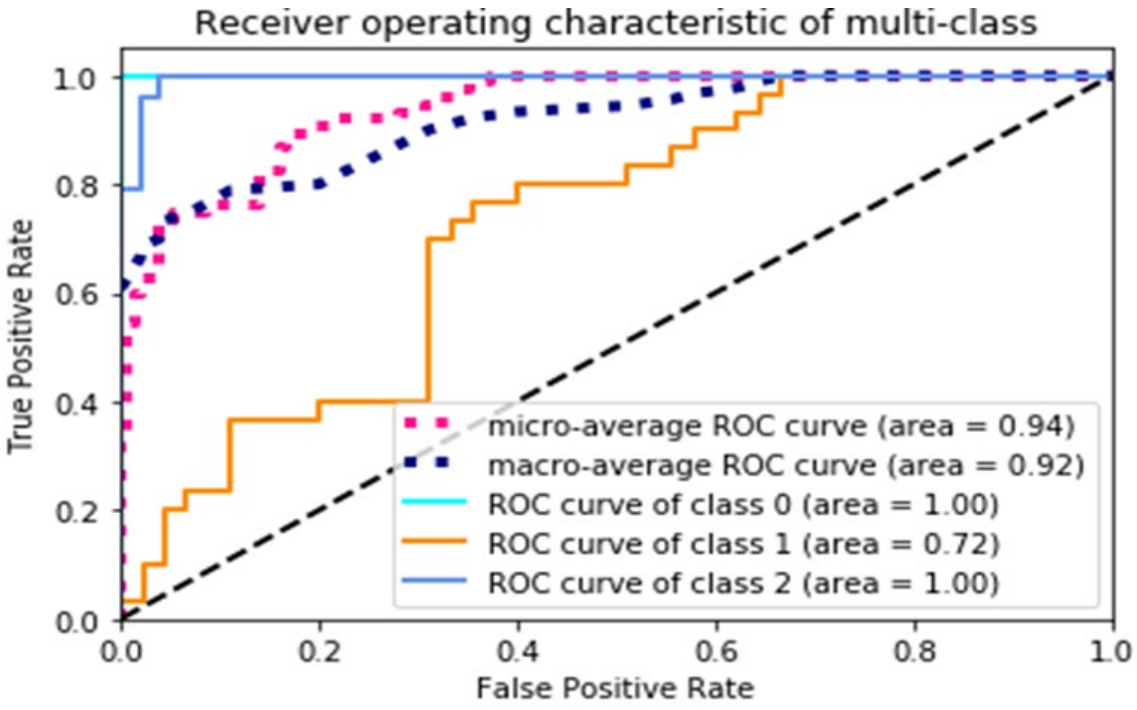
- ROC曲线:受试者工作特征曲线(receiver operator characteristic curve, ROC曲线),通常被用来评价一个二值分类器的优劣。ROC曲线的横坐标是假阳性率(false positive rate，FPR)，纵坐标是真阳性率(true positive rate，TPR)。
- TPR表示在所有实际为阳性的样本中，被正确地判断为阳性的比率，即TPR=TP/(TP+FN)。
- FPR表示在所有实际为阴性的样本中，被错误地判断为阳性之比率，即FPR=FP/(FP+TN)。



模型的评估

分类任务的评价

- TPR越高，FPR越低，则可以证明分类器分类效果越好。但是两者又是相互矛盾的，所以单凭TPR和FPR的两个值是没有办法比较两个分类器的好坏的，因此在机器学习里提出了ROC曲线。
- 也就是画出来的ROC曲线越靠近左上越好



模型的评估

分类任务的评价

- AUC（Area Under roc Curve）:AUC值为ROC曲线所覆盖的区域面积,是一种用来度量分类模型好坏的一个标准，显然，AUC越大，分类器分类效果越好。
- AUC = 1，是完美分类器，采用这个预测模型时，不管设定什么阈值都能得出完美预测。但绝大多数预测的场合，不存在完美分类器。
- 0.5 < AUC < 1，优于随机猜测，若妥善设定阈值，分类器将具有预测价值。
- AUC = 0.5，跟随机猜测一样，模型没有预测价值。
- AUC < 0.5，比随机猜测还差，但只要总是反预测而行，就优于随机猜测。

模型的评估

- **回归任务的评价**
  - **均方误差**（Mean squared error，MSE），是反映估计量与被估计量之间差异程度的一种度量，其值越小说明拟合效果越好，所以常被用作线性回归的损失函数。

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

  - **均方根误差**（Root Mean Squared Error, RMSE）：MSE的平方根。
  - **平均绝对误差**（Mean absolute Error，MAE），预测目标值和实际目标值之间误差的绝对值的平均数，可以更好地反映预测值误差的实际情况，其值越小越好。

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

模型的评估

- 回归任务的评价
  - **中位绝对误差**（Median absolute error，MedAE）通过取目标值和预测值之间的所有绝对差值的中值来计算损失，其值越小越好

$$MedAE(y,\hat{y}) = median(|y_1 - \hat{y}_1|,\ldots,|y_n - \hat{y}_n|)$$

  - R2决定系数（R Squared）表征回归方程在多大程度上解释了因变量的变化，或者说方程对观测值的拟合程度如何。R2决定系数的最优值为1（完全拟合），为0时，说明模型和样本基本没有关系，也可为负，为负时说明模型非常差

$$R^2(y,\hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

回顾
Review

- 什么是正则化?
- 正则化有哪几种方式?
- 特征工程主要包括哪些内容？
- 模型的评估包括哪些方面?
- 分类任务和回归任务有哪些常用的评价指标?

模型的评估

偏差和方差的权衡

- 模型的评估过程中还要考虑偏差（模型对训练数据的拟合程度）和方差（模型对训练数据的小波动的敏感程度）。一个理想的模型应该在这两者之间找到平衡，既能在训练数据上表现良好，也能在测试数据上保持较好的性能。
- 偏差（Bias）：模型预测的期望值与真实值之间的差距。高偏差通常表示模型过于简单，无法捕捉数据的复杂模式，导致系统性误差。偏差反映了模型的系统性错误。。
- 方差（Variance）：模型预测对训练数据的波动敏感性。高方差通常表示模型过于复杂，对训练数据的噪声和细节过度拟合，导致模型在不同数据集上的表现不稳定。方差反映了模型的随机误差。
- 噪声（Irreducible Error）：无法通过任何模型减少的误差，通常由数据本身的随机性或测量误差造成。

模型的评估

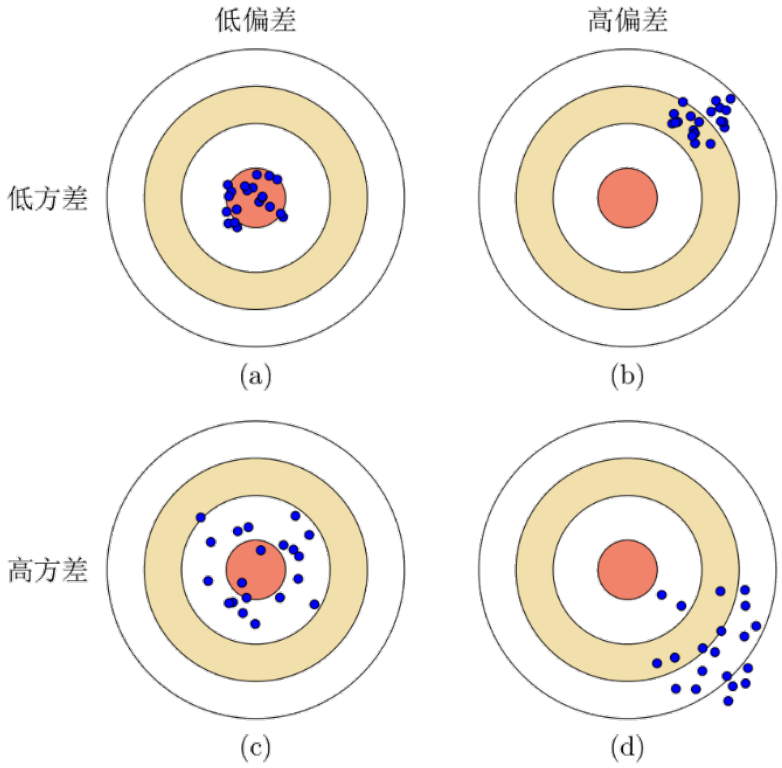
- **偏差和方差的权衡**
  - 在回归问题中，总误差（Total Error）可以表示为：

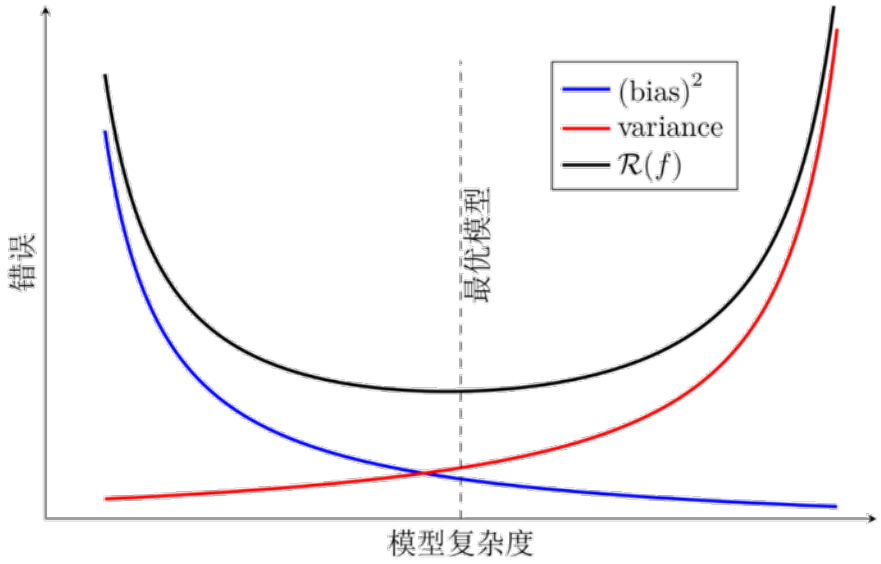
$$\text{Total Error} = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$

  - 在模型训练过程中，偏差和方差之间存在权衡关系
  - 欠拟合：模型过于简单，偏差高，方差低。
  - 过拟合：模型过于复杂，偏差低，方差高。
  - 最佳模型：在两者之间找到平衡，使得偏差和方差的总和最小化，从而达到最佳的泛化性能

模型的评估

- 偏差和方差的权衡





机器学习
Machine Learning

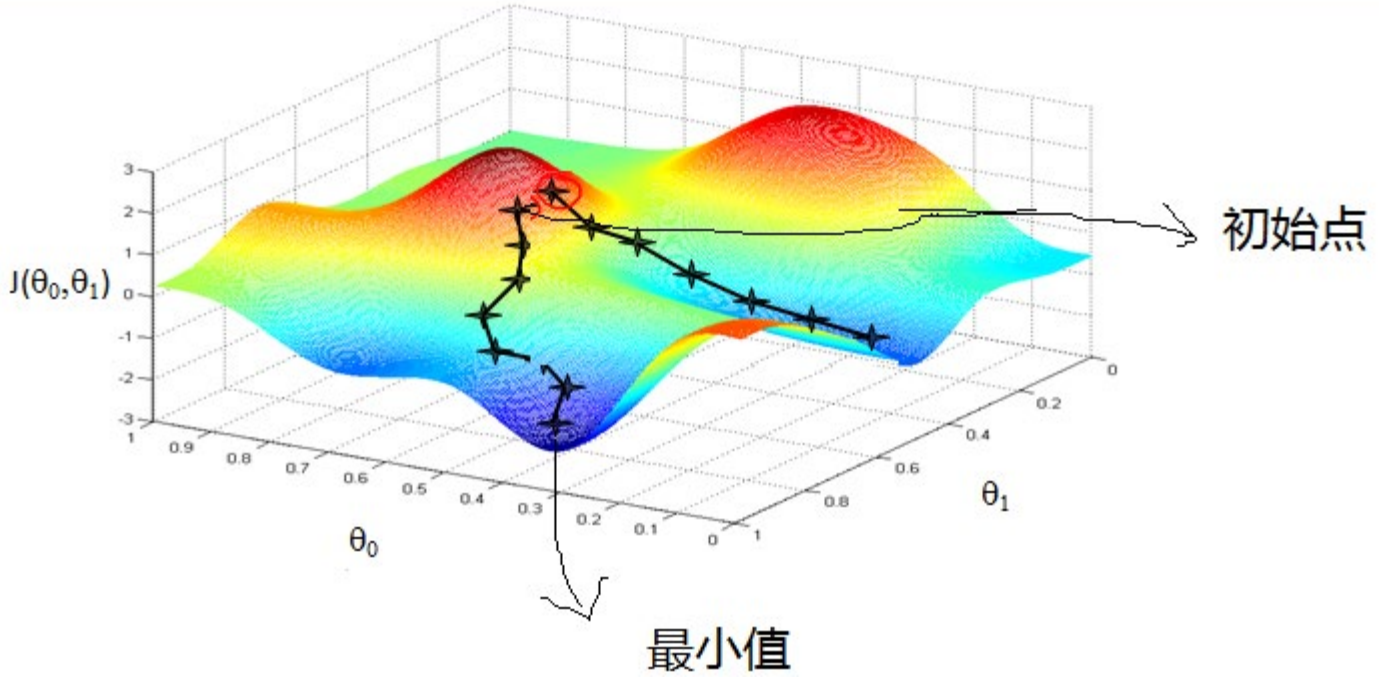
- 正则化
- 特征工程
- 模型的评估
- **模型的优化**
- 超参数

模型的优化

- **梯度下降法**
  - 想象一个场景，我们站在环形山内侧的山坡上，想要前往环形山内的最低点。但是暮霭沉沉，只能看清周围很小的一块区域，于是我们不得不通过周围这一小块区域的信息来寻找最低点的方向。那么怎么走才可以最快地到达最低点呢？

模型的优化

- **梯度下降法**
  - 根据常识，我们会很自然地想到：在当前位置查看四周的坡度，找出下坡坡度最大的方向，向着那个方向走一段；到达新的位置后，再根据新位置四周的地形，找出下坡坡度最大的方向走一段；直至我们发现四周都是上坡，已经无法继续下坡为止，那个位置大概率就是环形山内的最低点。



模型的优化

- **梯度下降法**
  - 如何确定坡度最大的方向？
  - 在微积分里面，对多元函数的参数求偏导数，把求得的各个参数的偏导数以向量的形式写出来，就是梯度。比如函数f(x,y), 分别对x,y求偏导数，求得的梯度向量就是(∂f/∂x, ∂f/∂y)T,简称grad f(x,y)或者▽f(x,y)。

$$\text{gradf}\ (x,\ y) = \nabla f(x,y) = \left\{ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right\}$$

  - 从几何意义上讲，梯度就是函数变化最快的地方。具体来说，对于函数f(x,y),在点(x0,y0)，沿着梯度向量的方向就是(∂f/∂x0, ∂f/∂y0)T的方向是f(x,y)增加最快的地方。或者说，沿着梯度向量的方向，更加容易找到函数的最大值。反过来说，沿着梯度向量相反的方向，也就是 -(∂f/∂x0, ∂f/∂y0)T的方向，梯度减少最快，也就是更加容易找到函数的最小值

模型的优化

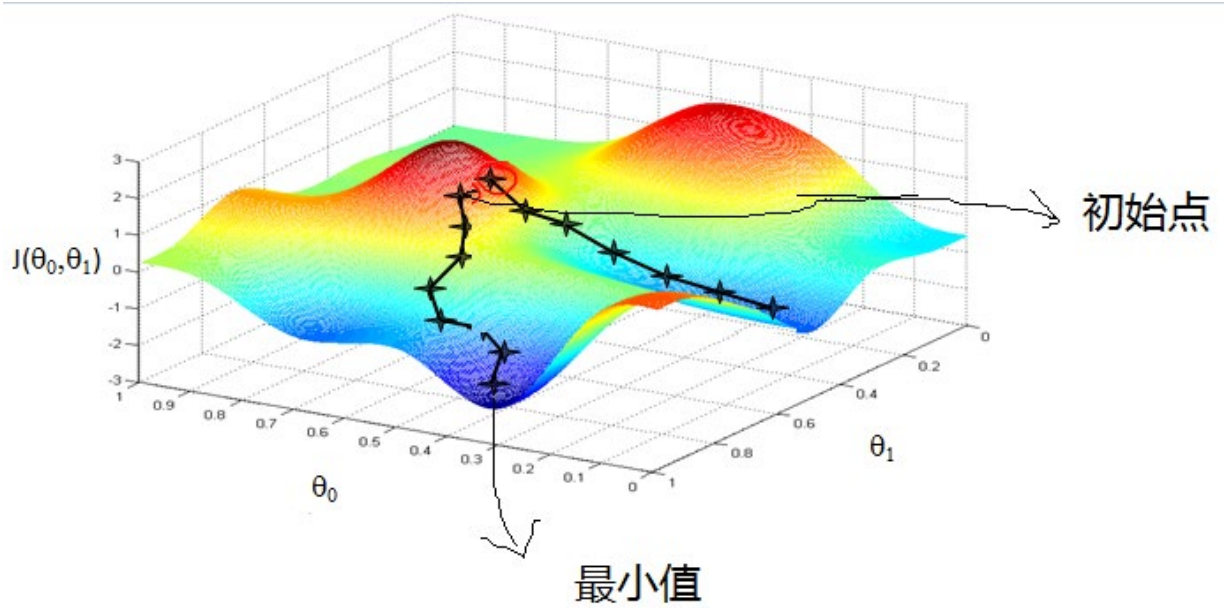
梯度下降法

- 由于我们不知道怎么下山，于是决定走一步算一步，也就是在每走到一个位置的时候，求解当前位置的梯度，沿着梯度的负方向，也就是当前最陡峭的位置向下走一步，然后继续求解当前位置梯度，向这一步所在位置沿着最陡峭最易下山的位置走一步。这样一步步的走下去，一直走到觉得我们已经到了山脚。
- 这就是梯度下降的基本思路。梯度下降法是一种迭代算法。先选取适当的自变量初值，通过不断迭代更新自变量，寻找目标函数（损失函数）的极小化，直至收敛（举例中四周都是上坡的情况）。在迭代的每一步，均选择使目标函数值下降最快的方向，以这个方向更新自变量，从而达到减小函数值的目的。

模型的优化

梯度下降法

- 梯度下降不一定能够找到全局的最优解，有可能是一个局部最优解。当然，如果损失函数是凸函数，梯度下降法得到的解就一定是全局最优解



- 步长（学习率Learning rate）：步长决定了在梯度下降迭代的过程中，每一步沿梯度负方向前进的长度。用上面下山的例子，步长就是在当前这一步所在位置沿着最陡峭最易下山的位置走的那一步的长度。

模型的优化

- **梯度下降法**

3. 算法过程：

1）确定当前位置的损失函数的梯度，对于θ向量,其梯度表达式如下：

$$\frac{\partial}{\partial\theta}J(\theta)$$

2）用步长乘以损失函数的梯度，得到当前位置下降的距离，即$\alpha\frac{\partial}{\partial\theta}J(\theta)$对应于前面登山例子中的某一步。

3）确定θ向量里面的每个值,梯度下降的距离都小于ε，如果小于ε则算法终止，当前θ向量即为最终结果。否则进入步骤4.

4）更新θ向量，其更新表达式如下。更新完毕后继续转入步骤1.

$$\theta=\theta-\alpha\frac{\partial}{\partial\theta}J(\theta)$$

新参数 = 原参数 - 学习率 x 梯度

较高的学习率可以使模型快速收敛，但也可能导致过度调整甚至发散(不收敛)。
较低的学习率虽然稳定，但收敛速度慢，可能需要更多的训练时间和迭代次数。
因此实际应用中需要二者的折中，设置最佳的学习率。

超参数（Hyperparameter）

- 什么是超参数
  - 超参数是用于控制机器学习算法学习过程的参数，与模型参数不同，模型参数是在训练过程中通过数据学习得到的，而超参数是在训练之前设置的，并且在训练过程中保持不变。
- 常见的超参数举例
  - 学习率（Learning Rate）：控制模型参数更新的步长
  - 正则化参数（Regularization Parameter）：控制模型复杂度，防止过拟合。
  - 神经网络结构（Neural Network Architecture）：例如层数、每层的神经元数量等。
  - 数据集的划分比例等