3.机器学习系统的设计

- **流程**

  确定用于训练的经验类型→确定目标函数→确定要学习函数的表示→确定学习算法（梯度下降、线性规划）

- **基本概念**

  1. **实例空间**X
  2. **假设空间**H
  3. **训练样例空间**D，**目标概念**C（标签）

  求解$h \in H \quad s.t.\ \forall x \in X,\ h(x) = c(x)$

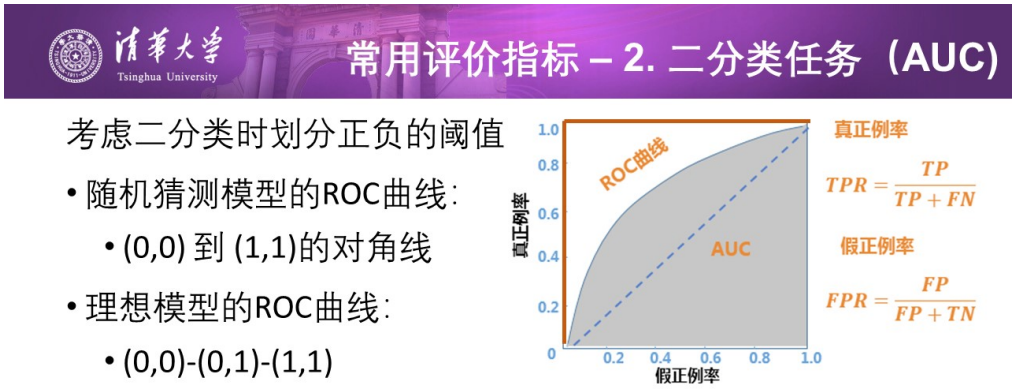
- **评价指标**

  1. **回归任务**：平均绝对误差（MAE）、均方误差（MSE）、均方根误差（RMSE）
  2. **分类任务**：准确率（Accuracy）、精度（Precision）、召回率（Recall）、AUC等

	$\hat{y}=1$	$\hat{y}=0$
$y=1$	TP（真阳性）	FN（假阴性）
$y=0$	FP（假阳性）	TN（真阴性）

$$\begin{cases} \mathtt{Precision} & = \frac{\mathtt{TP}}{\mathtt{TP+FP}} \\ \mathtt{Recall} & = \frac{\mathtt{TP}}{\mathtt{TP+FN}} \\ \mathtt{F}_\beta & = \frac{1}{\frac{1}{1+\beta^2}\cdot(\frac{1}{P}+\frac{\beta^2}{R})} = \frac{(1+\beta^2)\cdot P\cdot R}{(\beta^2\cdot P)+R} \\ \mathtt{F}_1 & = \frac{2PR}{P+R} \end{cases}$$

  - **真正例率**：$\mathtt{TPR} = \frac{\mathtt{TP}}{\mathtt{TP+FN}}$（所有正例中被预测为正例的比例）
  - **假正例率**：$\mathtt{FPR} = \frac{\mathtt{FP}}{\mathtt{FP+TN}}$（所有负例中被预测为正例的比例）
  - **ROC曲线**：根据预测值对样本进行排序，设置阈值，大于阈值的样本预测为正例，小于阈值的样本被预测为负例。根据阈值的不同，**以真正例率为纵坐标，以假正例率为横坐标**，可以得到**ROC曲线。**



对于理想模型，其所有正例的预测值大于所有负例的预测值，因此其ROC曲线为折线。显然曲线越接近理想模型的ROC曲线，说明预测效果越好。因此可以使用**AUC**（即ROC曲线下方与x轴所夹的面积）来衡量预测效果的好坏。

> **AUC**的简便计算方法：将测试样例排序后，设共有n_1个预测正例，n_0个预测负例，设r_i表示第i个真实负例的**秩**（排序位置）。记$S_0 = \sum r_i$。则**AUC**可以简便计算为：
>
> $$\hat{A} = \frac{S_0 - n_0(n_0+1)/2}{n_0 n_1}$$

3. **特定任务**：
   - **搜索**、**推荐**：Precision@K，Recall@K，NDCG@K，Hit@K
   - **对话系统**：BLEU
   - ...

   1. **Hit@K**：给出的前K个推荐中，是否有正例
   2. **DCG@p**（Discounted Cumulative Gain）：对一个特定位次p的累积增益

      $$\mathtt{DCG}_p = \mathtt{rel}_1 + \sum_{i=2}^{p} \frac{\mathtt{rel}_i}{\log_2 i}$$

      或

      $$\mathtt{DCG}_p = \sum_{i=1}^{p} \frac{2^{\mathtt{rel}_i} - 1}{\log(1+i)}$$

   3. **NDCG**（Normalized DCG）

      用实际的DCG值除以理想排序下的DCG值。

   4. **BLEU**（bilingual evaluation understudy）双语替代评价，多用于机器翻译。原理是**检查译文中的每个n-gram是否在参考译文中出现**（并且，每个词在译文中的有效频次**不应**超过参考译文中的频次）。

      $$\mathtt{BLEU} = \mathtt{BP} \times \exp\left(\frac{1}{n}\sum_{i=1}^{n}\ln(p_i)\right)$$

      其中，BP是一个与译文长度相关的系数，$\mathtt{BP} = \exp\left(1 - \frac{\text{参考译文长度}}{\text{模型译文长度}}\right)$。$p_i$表示i-gram对应的**修正精度**。另外，规定若存在某个精度为0，则BLEU值置为0。

      同时，也可以对精度进行拉普拉斯平滑，也即计算精度时将分子分母同时加一，以避免精度为0的情况出现。

4.决策树学习

4.1决策树基础

- **节点混杂度**：
  1. **熵**

     $$\mathtt{Entropy}(N) = -\sum_j P(w_j)\log_2 P(w_j)$$

     熵越大，代表节点混杂度越大。

  2. **基尼混杂度**

$$i(N) = \sum_{i \neq j} P(w_i)P(w_j) = 1 - \sum_j P^2(w_j)$$

3. **错分类混杂度**

$$i(N) = 1 - \max_j P(w_j)$$

使用节点中占比最大的类作为该节点的标签。

ID3：选用**信息增益**最大的特征作为下一步的分类特征。

4.2过拟合问题及剪枝

4.2.1错误降低剪枝

- 当数据的分裂在统计意义上并不显著时，就停止增长：**预剪枝**

  停止分裂有几个比较常用的条件：

  1. 到达一个节点的训练样本数小于训练集合的一个特定比例（例如5%）
  2. 设定一个较小的阈值，如果满足下述条件就停止分裂

$$\Delta i(s) \leq \beta$$

- 构建一棵完全树，然后做**后剪枝**

  将数据集分为训练集和验证集，在训练集上做训练，在验证集上做剪枝。在验证集上测试减去每个可能节点（和以其为根的子树）的影响，**贪心地**去掉某个可以提升**验证集准确率**的节点。减去可能节点后，新的节点可以简单地赋值成最常见的类别。

4.2.2规则后剪枝

1. 把树转换成等价的由**规则**构成的集合

   例如，`if (outlook=sunny) and (humidity=high) then playTennis=no`

   > 之所以要将树转换为规则，是因为如果子树被剪枝，就只有两种可能，要么完全删除，要么完全保留，无法实现对某个特定前件的剪枝。并且，转换为规则后，可读性也得到了提升。

2. 对每条规则进行剪枝，去除那些能够提升该规则准确率的**规则前件**
3. 将规则排序成一个序列（根据规则的准确率从高往低排序）
4. 用该序列中的最终规则对样本进行分类（**依次查看其是否满足规则序列**）

在以上的过程之后，所有的规则可能不再能恢复成一棵树。