清风第四次直播：

利用 Matlab 快速实现机器学习

1. 软件要求

- 安装 MATLAB2017a 以上的版本，需要用到机器学习工具箱
- 在电脑配置允许的前提下，版本越新越好

2. 课程目标

（1）初级目标

- 了解机器学习中的基本概念。
- 了解一些常用的机器学习算法的思想，例如 K 最近邻（KNN）、决策树、SVM 和一些集成算法（如装袋法（Bagging），提升法（Boosting））。
- 借助 MATLAB 工具箱实现机器学习中的算法（非常傻瓜、有手就行）。

（2）高级目标（怎么在千篇一律的论文中脱颖而出？论文的创新点和优势）

- 在 MATLAB 自动生成的代码上进行二次加工，使用网格搜索等策略进行自动调参。
- 学习 MATLAB 工具箱中没有集成的算法，例如近邻成分分析（NCA）、最大相关最小冗余(MRMR)等

- **注意：本直播课的目的主要是帮助大家初步理解各种算法的思想，不会系统讲解每一个算法背后的数学理论。我们主要以应用为主！**

3. 资料下载

微信公众号《数学建模学习交流》后台发送“机器学习”四个字获取。

4. 推荐的视频

我们这个视频以应用为主，主要是帮助大家快速使用机器学习的算法去解决遇到的问题。如果你是以求职就业为目的，就需要学习更系统深入的课程，下面这些资料供大家参考。（对数学和统计的功底要求较高，大家也可以去知乎搜索其他人的推荐）

(1) Python 课程

市面上的机器学习视频几乎都是使用的 Python，因此大家可以在 B 站找一个播放量高的视频入门，入门后至少要学习 Numpy、Pandas 以及 Matplotlib 这三个包。

(2) 机器学习通识课程

吴恩达（B 站有视频）、李宏毅（B 站有视频），吴恩达的稍微简单一点。

(3) 偏向于原理和推导的课程和书籍（较难）

机器学习-白板推导系列（B 站有视频）

周志华著. 机器学习（书籍），B 站有配套讲解视频

李航著. 统计学习方法（书籍），B 站有配套讲解视频

(4) 偏向于应用的课程

菜菜的机器学习 sklearn

唐宇迪的机器学习视频

一、 基本概念

1.1 什么是机器学习？

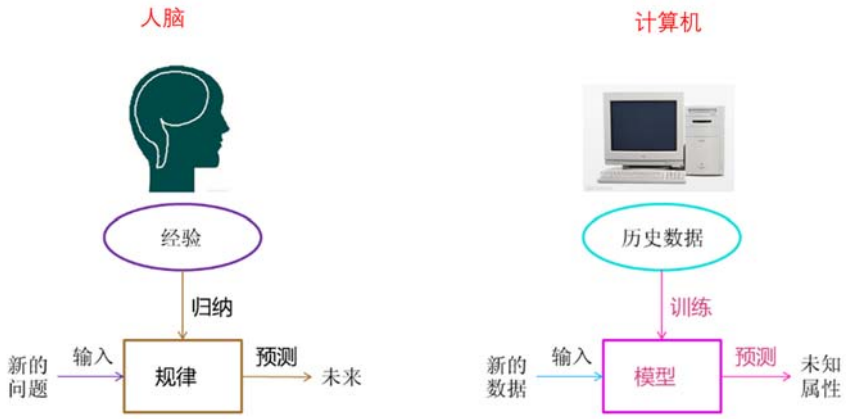
问题来源：夏天买西瓜怎么挑到好瓜？

历史经验：色泽青绿、根蒂蜷缩、敲声浊响的瓜就是好瓜。

思考：上面对经验的利用是靠我们人类自身完成的，计算机能帮忙吗?

参考书籍：周志华著. 机器学习, 北京: 清华大学出版社

书中对于机器学习的一个定义： 机器学习正是这样一门学科,它致力于研究如何通过计算的手段,利用经验来改善系统自身的性能.在计算机系统中,“经验”通常以“数据”形式存在,因此,机器学习所研究的主要内容,是关于在计算机上从数据中产生“模型”(model)的算法, 即“学习算法”(learning algorithm).有了学习算法,我们把经验数据提供给它,它就能基于这些数据产生模型;在面对新的情况时(例如看到一个没剖开的西瓜),模型会给我们提供相应的判断(例如好瓜).



机器学习的思路：

在市面上买 100 个西瓜，通过仪器或者观察来收集这些西瓜的特征数据，然后通过品尝来判断这些西瓜是否为好瓜，得到的部分数据如下表所示：

编号	色泽	根蒂	敲声	纹理	脐部	触感	密度	含糖率	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	0.697	0.460	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	0.774	0.376	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	0.634	0.264	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	0.608	0.318	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	0.556	0.215	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	0.403	0.237	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	0.481	0.149	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	0.437	0.211	是

上面这个表中的数据就是历史数据，我们的样本数（西瓜的个数）为 100。

输入变量（自变量）　　输出变量（因变量）

编号	色泽	根蒂	敲声	纹理	脐部	触感	密度	含糖率	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	0.697	0.460	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	0.774	0.376	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	0.634	0.264	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	0.608	0.318	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	0.556	0.215	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	0.403	0.237	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	0.481	0.149	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	0.437	0.211	是

假设现在我们又买了 10 个新的西瓜，并同样测得了这些西瓜的特征数据（上表中的自变量），我们如何通过计算机判断这 10 个西瓜是不是好瓜呢？

这个问题实际上就是机器学习中要解决的分类问题，下面我们介绍机器学习的划分。

1.2 机器学习的划分

下面是书中对于机器学习的一个划分：

机器学习一般包括监督学习（supervised learning）、无监督学习（unsupervised learning）、强化学习（reinforcement learning）。有时还包括半监督学习(semi-supervised learning)、主动学习(active learning)。

（注意：这本书中对于算法理论的推导要比周志华的书详细很多，但总的来说两本书读起来都很困难，需要很深厚的数学和统计基础，b 站可以找到这两本书的导读视频，有志于从事相关工作的同学可以好好学）

参考书籍：李航著. 统计学习方法(第 2 版), 北京: 清华大学出版社

1.2.1 监督学习

书中的定义：监督学习是指从标注数据中学习预测模型的机器学习问题。标注数据表示输入输出的对应关系，预测模型对给定的输入产生相应的输出。监督学习的本质是学习输入到输出的映射的统计规律。

翻译成白话：我们的数据既有输入变量又有输出变量（既有特征 feature 又有标签 label），我们要找到输入变量和输出变量之间的关系。

输入变量X　　输出变量Y

编号	色泽	根蒂	敲声	纹理	脐部	触感	密度	含糖率	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	0.697	0.460	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	0.774	0.376	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	0.634	0.264	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	0.608	0.318	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	0.556	0.215	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	0.403	0.237	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	0.481	0.149	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	0.437	0.211	是

Y = F(X)，F就表示X和Y对应的关系
（注意：F不一定代表一个具体的函数，这里只表示一种记号，代表从输入到输出的一种映射）

监督学习根据输出变量 Y 的数据类型不同，又可以分成两种类型[1]：

（1） 当输出变量 Y 取有限个离散值时，称为**分类问题**。

举例：

- 判断西瓜的好坏（好瓜/坏瓜）
- 判断肿瘤的性质（良性/恶性）
- 根据鸢尾花的花萼长度、花萼宽度、花瓣长度和花瓣宽度这四个指标来判断它的种类（山鸢尾/杂色鸢尾/维吉尼亚鸢尾）

在分类问题中，当 Y 只取两类时，我们称为二分类问题，当分类的类别为多个时，称为多分类问题。

（2） 输出变量 Y 为连续型变量，称为**回归问题。**（此回归非彼回归）

举例：

- 给定房屋的一些信息（户型、是否靠近地铁等），预测房价
- 给定土地的施肥量，预测农作物的产量

1.2.2 无监督学习

书中的定义：无监督学习是指从无标注数据中学习预测模型的机器学习问题。无标注数据是自然得到的数据，预测模型表示数据的类别、转换或概率。无监督学习的本质是学习数据中的统计规律或潜在结构。

翻译成白话：我们的数据全部都是输入变量，没有输出变量。我们希望得到数据之间隐藏着的结构和规律。

无监督学习最常见的两种用法：聚类和降维。

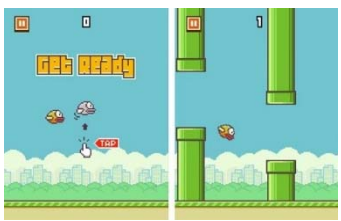
聚类的例子：银行收集了客户的许多个人信息，根据这些个人信息可以将客户划分到不同的用户群体（例如：贵宾客户、重点客户、普通客户、可能流失的客户等），银行可以为不同的用户群体制定出相应的个性化营销方案。

降维的例子：输入变量的维度太大（指标个数太多了），我们需要通过降维的方法来构造出少数几个指标，这几个指标能保留原来这些输入变量的绝大部分信息。

注意：有很多同学区分不开聚类和分类的概念，事实上你只要知道监督学习和无监督学习的核心区别就行了（有无输出变量 Y）。在分类中，类别是已知的；而在聚类中，类别是不知道的，我们是通过数据的特征属性将数据划分到某几类中，这几个类代表的含义需要我们自己根据聚类的结果来定义。

1.2.3 强化学习

书中的定义：强化学习(reinforcement learning)是指智能系统在与环境的连续互动中学习最优行为策略的机器学习问题。假设智能系统与环境的互动基于马尔可夫决策过程(Markov decision process)，智能系统能观测到的是与环境互动得到的数据序列。强化学习的本质是学习最优的序贯决策。



举个例子：如何让电脑玩游戏？以 flappy bird 这款游戏为例，电脑怎么知道下一步小鸟要采取怎样的行动呢？

通过不断与环境的交互和试错的过程，最终完成特定目的或使得整体行动收益最大化。（做对了给奖励，做错了给惩罚）

[1] 李航老师的教材将监督学习分成了三种：分类问题、标注问题和回归问题。可以认为标注问题是分类问题的一个推广。

1.3 模型评估指标

从下面开始，本视频只介绍监督学习的内容，即回归问题和分类问题。

我们先来介绍一下衡量模型结果好坏的一些指标。

1.3.1 回归问题的评估指标

以预测房价为例，假设我们有 n 个样本（即 n 个房屋的信息和价格数据），这 n 个房屋的真实价格用向量$y=[y_1,y_2,\cdots,y_n]$表示，我们建立的机器学习模型得到的这 n 个房屋价格的估计值（预测值）是向量$\hat{y}=[\hat{y}_1,\hat{y}_2,\cdots,\hat{y}_n]$。

- *SSE 误差(或残差)平方和（Sum of Squares due to Error）*

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

范围[0,+∞)，当预测值与真实值完全吻合时等于 0。误差越大，该值越大。它的量纲是原来数据量纲的平方。

- *MSE 均方误差（Mean Square Error），就是 SSE 除了一个样本数 n。*

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

- *RMSE:均方根误差（Root Mean Square Error），其实就是 MSE 加了个根号*

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

它的量纲和原来数据的量纲相同，这个用的较多。

- *MAE: 平均绝对误差（Mean Absolute Error）*

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

- *MAPE: 平均绝对百分比误差（Mean Absolute Percentage Error）*

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

- *SMAPE：对称平均绝对百分比误差（Symmetric Mean Absolute Percentage Error）*

$$SMAPE=\frac{100\%}{n}\sum_{i=1}^{n}\frac{|y_i-\hat{y}_i|}{(|y_i|+|\hat{y}_i|)/2}$$

上面介绍的这些指标都有其优点和缺点，感兴趣的同学可以在网上搜索更多相关的内容学习。

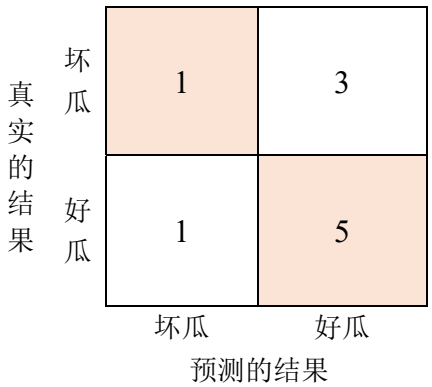
1.3.2 分类问题的评估指标

先举个例子，假设现在我们要预测西瓜的好坏。下表是我们最后得到的预测结果：

真实的结果	预测的结果
坏瓜	好瓜
坏瓜	好瓜
坏瓜	坏瓜
坏瓜	好瓜
好瓜	好瓜
好瓜	好瓜
好瓜	好瓜
好瓜	好瓜
好瓜	坏瓜
好瓜	好瓜

可以看到，真实的结果中有 4 个坏瓜，有 3 个被错误的预测成了好瓜；真实的结果中有 6 个好瓜，有 1 个错误的被预测成了坏瓜。

我们可以用下面这个表格来可视化这个预测的结果，这个表格又被称为混淆矩阵（Confusion Matrix）。



根据上面这个混淆矩阵，我们可以计算出很多衡量分类结果好坏的指标。

在此之前，我们需要定义分类结果中的正类（positive）和负类（negative），这里的正类和负类实际上借用了医学中的阳性（positive）和阴性（negative）的概念，医学中一般阴性代表正常，而阳性则代表患有疾病。**在机器学习中，我们通常将更关注的事件定义为正类事件。（生活中我们通常会更关注那些结果不好的情况的出现）**[2]

例如上面的西瓜分类的例子中，如果我们更关注坏瓜，就定义坏瓜为正类，好瓜为负类。（有些地方也用 0 和 1 表示分类结果，一般正类记为 1，负类记为 0）。

下面请大家思考，以下几个例子怎样定义正负类比较合适。

（1） 判断肿瘤是良性还是恶性。

（恶性为正类）

（2） 银行发放贷款前，会对申请人进行审核，判断是否可能会违约。

（违约为正类）

（3） 买股票前，我们预测股票未来会上涨还是下跌。

（下跌为正类）

[2]这种划分正类和负类的标准也不是绝对的，如果你更关注好瓜，那么你也可以把好瓜定义成正类；另外有时候我们很难去区分结果的好坏，例如我们要对猫和狗的图片进行分类，这时候正类和负类无论怎么定义都行。

接下来我们要将混淆矩阵中的四个数值分别定义成四个指标：

指标	含义
True Positive（TP）	将正类预测为正类的数量
False Negative（FN）	将正类预测为负类的数量
False Positive（FP）	将负类预测为正类的数量
True Negative（TN）	将负类预测为负类的数量

真实的结果	坏瓜 P（正类）	TP 1	FN 3
	好瓜 N（负类）	FP 1	TN 5
		坏瓜 P（正类）	好瓜 N（负类）
	预测的结果		

这几个指标大家记不下来没关系，截个图以后会用就行！

下面我们开始定义一系列的评估指标：

（1）分类准确率(Accuracy)

实际上就是正确分类的样本数与总样本数的比例，Accuracy $= \frac{TP+TN}{TP+FN+FP+TN} = 0.6$

准确率有一定的局限性，假设我们分类的目标是识别好人和坏人，好人有 95 个，坏人只有 5 个（样本不平衡）。如果我们设计的模型是将所有的人全部都识别成好人，那么分类准确率为 95%，但是这个模型并没啥用，因为一个坏人都没别识别出来。

（2）查全率或召回率（Recall，记为 R）

在实际为正类的样本中，预测正确的比例，西瓜例子中**R** $= \frac{TP}{TP+FN} = 0.25$

而上面这个识别坏人的例子中，坏人为正类，此时查全率 R为 0.

（3）查准率或精确率（Precision，记为 P）

在预测为正类的样本中，预测正确的比例，西瓜例子中**P** $= \frac{TP}{TP+FP} = 0.5$

通常来说，查全率和查准率是负相关关系的。

怎么理解查全率和查准率的关系：（知乎：李韶华的回答）

假设我们的目的是要找到人群中隐藏的坏人（把坏人当成正类）。

如果看重查全率 R：宁可错杀一千个好人，不可漏过一个坏人。（全部识别成坏人时查全率为 1）

如果看重查准率 P：宁可漏过坏人，不可错杀无辜的好人。（让 FP 尽量小一点，没有充足的证据不会轻易判断一个人是坏人）

（4）F_1 分数（F1 Score）

F_1 分数是查全率和查准率的调和平均数。$\frac{2}{F_1}=\frac{1}{R}+\frac{1}{P}$，化简可得$F_1=2*\frac{P*R}{P+R}$

如果带入查全率 $R=\frac{TP}{TP+FN}$ 和查准率 $P=\frac{TP}{TP+FP}$可得：$F_1=\frac{2TP}{2TP+FP+FN}$

事实上 F_1 分数是F_β分数在β等于 1时的特例，$F_\beta=(1+\beta^2)*\frac{P*R}{\beta^2*P+R}$ （$\beta\geq 0$）

β大于 1 时，查全率的权重高于精确率； β小于 1 时，精确率的权重高于查全率。

注意：F_1 和F_β的范围都是位于[0,1]之间，越接近 1 表示分类效果越好。

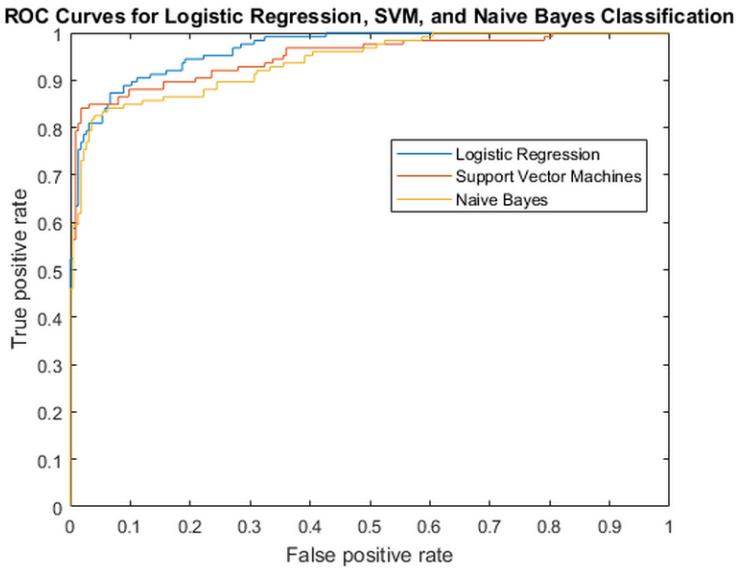
（5）ROC 曲线和 AUC

ROC 曲线全称为受试者工作特征曲线（receiver operating characteristic curve），它是根据一系列不同的分类阈值，以真正类率(True Postive Rate, TPR)为纵坐标，假正类率（False Positive Rate, FPR）为横坐标绘制的曲线。

其中，真正类率 TPR 就是我们前面介绍的查全率 R，它在 ROC 曲线中又可以被称为灵敏度(sensitivity)；假正类率（FPR）的计算公式是 FPR= FP / (FP + TN)，它表示将负类错误的识别为正类的样本占所有负类样本的比例，一般我们记 1-FPR 为特异性(specificity)。

它的原理我这里就不具体介绍了，我只介绍一下它的使用用法。

（1）将不同的模型的 ROC 曲线绘制在同一张图内，最靠近左上角的那条曲线代表的模型的分类效果最好。



（2）实际任务中，情况很复杂，如果两条 ROC 曲线发生了交叉，则很难一般性地断言谁优谁劣。因此我们引入 AUC，AUC（Area Under Curve）被定义为 ROC 曲线与下方的坐标轴围成的面积，AUC 的范围位于[0,1]之间，AUC 越大则模型的分类效果越好，如果 AUC 小于等于 0.5，则该模型是不能用的。通常 AUC 大于 0.85 的模型就表现可以了。

1.4 模型的泛化能力

模型的泛化能力(generalization ability)是指由该模型对未知数据的预测能力。

还是举西瓜分类的例子，我们利用 100 个西瓜的数据建立了一个模型，不妨称其为“西瓜分类器”，它能帮助我们区分好瓜和坏瓜。

那么我们怎样去评价这个“西瓜分类器”好不好用呢？

有同学会想，前面我们学了那么多用来评价分类结果好坏的评估指标，比如分类准确率、F1 分数、AUC 等。如果这些指标都很大的话不就说明我们的模型很好吗？

事实上这是不够的，我们更加关心的是这个“西瓜分类器”在面对一组新的数据的时候，它的分类能力还是不是足够好。

打个比方，准备期末考试的时候，大家都做课后作业题复习，可能你每道课后题都会做，但上了考场还是啥都不会。

回到我们西瓜分类的例子上，怎么去衡量这个“西瓜分类器”的泛化能力是否强呢？我们可以再去买 20 个西瓜，把这些西瓜的特征数据（输入数据 X）输入到我们的“西瓜分类器”中，然后看它的预测结果（输出结果为 Y）；另一方面，我们可以人工去判断这 20 个西瓜是好瓜还是坏瓜，这样就可以得到真实的结果。接下来只需要将预测结果和真实结果进行对比，看看“西瓜分类器”的分类准确率是不是很高。如果“西瓜分类器”表现的很好，那就说它的泛化能力很强；如果“西瓜分类器”表现的很糟糕，那么它的泛化能力就很弱，这时候你就要怀疑我们的“西瓜分类器”的模型是否存在问题。

1.5 留出法

前面我们为了衡量“西瓜分类器”的泛化能力，我们又买了 20 个西瓜用于测试。但大多少时候我们获取新的测试样本的成本较高，例如银行判断申请贷款的客户是否会违约时，需要等到客户还钱的时候才知道结果。因此，我们需要想一个办法，只使用已有的样本数据来对模型的泛化能力进行一个评价。

实际上这个办法很容易想到：还是假设我们现在有 100 个西瓜的数据，这些西瓜的特征数据 X 以及是否为好瓜 Y 我们是知道的。

我们只拿出 80 个西瓜来训练我们的“西瓜分类器”，剩下的 20 个西瓜我们假装不知道它们是好瓜还是坏瓜。接下来，我们把这 20 个西瓜的 X 输入到我们的“西瓜分类器”中，来得到预测结果，并和这 20 个西瓜的真实类别进行对比来计算分类准确率，这个结果就能反映模型的泛化能力的好坏。我们将这里的 80 个西瓜称为**训练集(train set)**，它们用来训练我们的模型，得到我们模型中的待估参数；剩下的 20 个西瓜我们不参与模型的训练过程，只用来最后对模型的好坏进行测试，因此被称为**测试集(test set)**。我们将上面这种对泛化能力进行评估的方法称为**留出法（Hold-Out）**。

有以下几点要说明：

（1） 假设我们总共的样本量为 N，我们要将其划分为训练集和测试集，这两个集合的划分比例通常设置为：6:4、7:3 或 8:2。

（2） 训练集和测试集的划分既要随机，又要尽可能保持数据分布的一致性（在分类问题中就是类别比例的相似），例如原来 100 个瓜中有 60 个好瓜，40 个坏瓜，那么你按照 8:2 的比例生成训练集和测试集时，尽量保证测试集中的 20 个样本内有 12 个好瓜和 8 个坏瓜。在分类任务中，保留类别比例的采样方法称为分层采样（stratified sampling）。

下面请大家思考一下留出法有什么缺陷？

1.6 交叉验证

在留出法中，用于评价模型泛化能力的测试集只是所有样本的一部分，而且这个结果不是很稳定，对模型的泛化能力的评价依赖于哪些样本点落入训练集，哪些样本点在测试集。

下面我们介绍一种用的更多的方法：**k 折交叉验证（K-fold cross-validation）**。

我们先将数据集 D 随机的划分为 k 个大小相似的互斥子集。每一次用 k-1 个子集的并集作为训练集，剩下的一个子集作为测试集；这样就可以获得 k 组训练/测试集，从而可进行 k 次训练和测试，最终返回的是这 k 次测试的平均结果，通常 k 取 10，此时称为 10 折交叉验证。

下面的图片来自周志华老师的机器学习一书，非常形象：

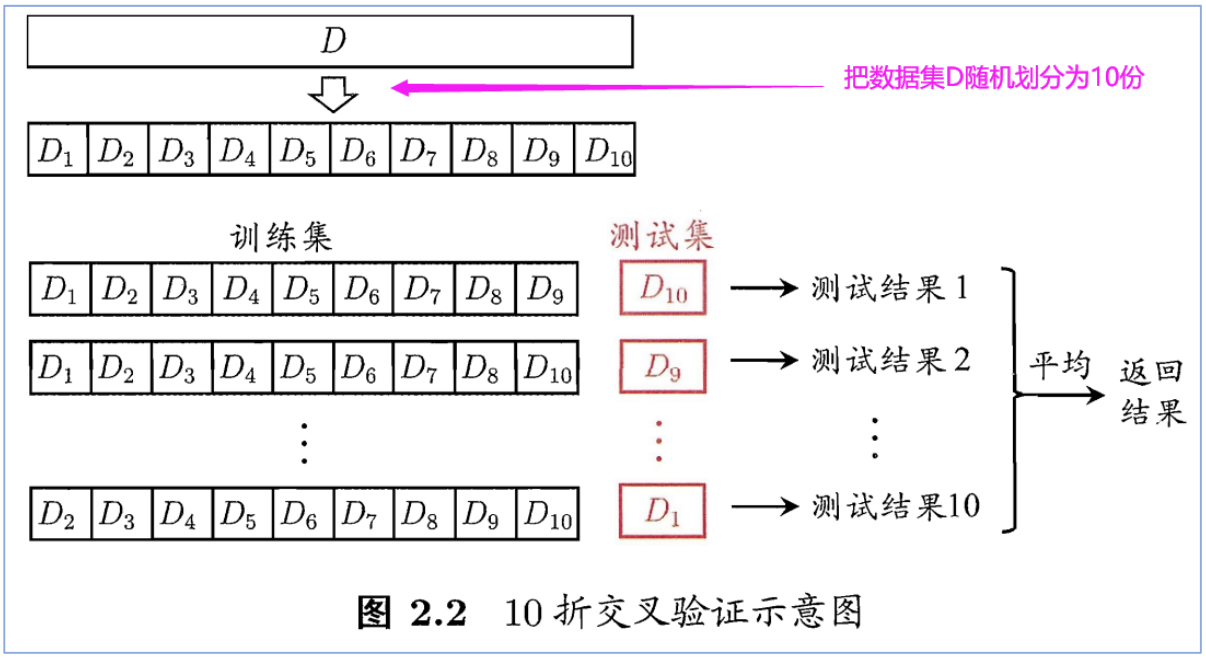


图 2.2　10 折交叉验证示意图

思考：假设总共的样本量为 N，如果我们取 k=N 会发生什么情况？

下面是书中的介绍，大家了解即可：

这种情况是交叉验证法的一个特例：留一法(Leave-One-Out，简称 LOO)，即把数据集分成了 N 份，每次用 N-1 个数据来训练模型，剩下的 1 个数据来进行测试。显然，留一法不受随机样本划分方式的影响，因为 N 个样本只有唯一的方式划分为 N 个子集——每个子集包含一个样本；留一法使用的训练集与初始数据集相比只少了一个样本，这就使得在绝大多数情况下，留一法中被实际评估的模型与期望评估的用整个数据集训练出的模型很相似。因此，留一法的评估结果往往被认为比较准确。然而，留一法也有其缺陷：在数据集比较大时，训练 N 个模型的计算开销可能是难以忍受的(例如数据集包含 1 百万个样本，则需训练 1 百万个模型)，而这还是在未考虑算法调参的情况下。另外，留一法的估计结果也未必永远比其他评估方法准确；“没有免费的午餐”定理对实验评估方法同样适用。

1.7 选择最好的模型

对于同一个问题，我们可以建立不同的模型去解决。例如前面介绍的西瓜分类问题中，我们可以使用决策树、K 最近邻（KNN）、支持向量机（SVM）等常用的机器学习模型。那么，我们应该怎样衡量一个模型的好坏呢？

我们前面介绍了留出法和交叉验证法，这里面都需要将数据分成训练集和测试集。因此，我们可以在同一个训练集下，分别对这些模型进行训练，然后将这些模型分别在测试集上进行预测，并比较不同模型的泛化能力，我们选择泛化能力最好的模型。（该模型在测试集上的表现最好，例如误差最小，具体的评价指标在前面有介绍）

另外，大多数的模型中都需要设定一些参数(parameter)，参数不同得到的结果可能有很明显的差异。因此，除了要对模型进行选择外，还需要对模型中的参数进行设定，这就是机器学习中常说的“参数调节”或简称“调参”(parameter tuning)。通常调参依赖于经验，我们后面会介绍网格搜索的方法，来自动搜索使模型效果最好的参数。

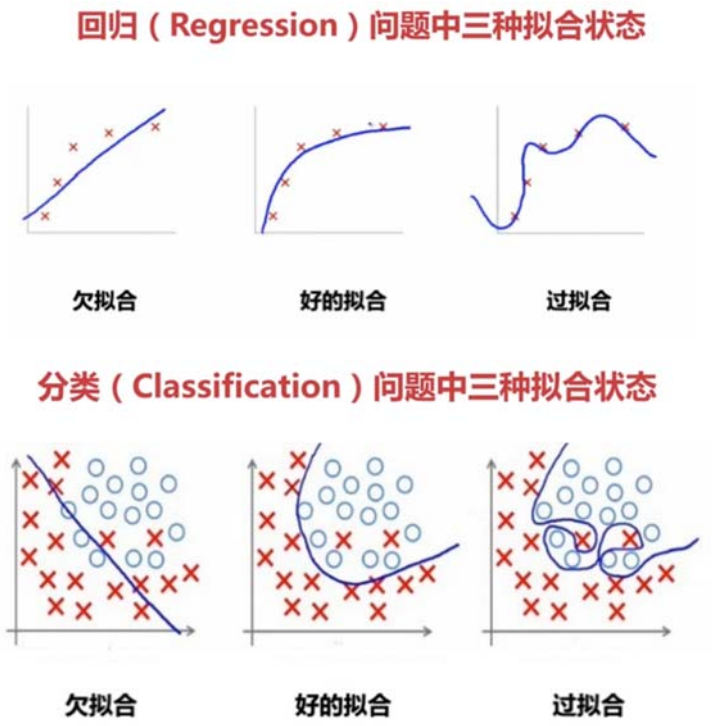
下面这一点很容易被大家忽视，这在周志华老师的书中有介绍。

给定包含 N 个样本的数据集 D，在选择模型的过程中，因为需要留出测试集的数据进行评估测试，所以我们只使用了训练集的数据来训练模型，这会导致测试集的信息在训练模型的过程中没有被利用到。因此，在模型选择和参数都调整完成后，我们应该使用完整的数据集 D 来重新训练模型。这个模型在训练过程中使用了所有 N 个样本，这才是我们最终需要的模型。

1.8 欠拟合和过拟合

过拟合（overfitting）指的是模型在训练集上表现的很好，但是在测试集上表现的并不理想，也就是说模型对未知样本的预测表现一般，泛化能力较差。

如果模型不仅在训练数据集上的预测结果不好，而且在测试数据集上的表现也不理想，也就是说两者的表现都很糟糕，那么我们有理由怀疑模型发生了欠拟合（underfitting）现象。



训练集上的表现	测试集上的表现	可能的原因
好	不好	过拟合
好	好	好的拟合（适度拟合）
不好	不好	欠拟合

下面分享两个来自知乎上的段子：

（1）以期末考试举例子：

过拟合：课后题全能做对，但是理解的不好，好多题的答案都是强行背下来的，上考场后题目稍微变一点就懵逼。

欠拟合：只看书上的知识点，课后题没怎么做，上了考场大部分题目都不会。

（2）一句话概括：

欠拟合："你太天真了"

过拟合："你想太多了"

可能产生过拟合的常见原因：

（1） 模型中参数设置的过多导致模型过于复杂

（2） 训练集的样本量不够

（3） 输入了某些完全错误的的特征

举个极端的例子：样本的编号。现在有 100 个西瓜，编号 1-60 的是好瓜，编号 61-100 的是坏瓜，如果你把编号作为了输入变量放入了我们的模型，那么有可能模型会将编号作为一个最重要的识别变量来对西瓜进行分类，模型会认为只要编号小于等于 60 的都是好瓜，此时在训练集上的误差一定为 0。。。。。如果这时候你拿来编号大于 100 的需要判断好坏的瓜，模型都会认为是坏瓜！

解决过拟合的方法：

（1） 通过前面介绍的交叉验证的方法来选择合适的模型，并对参数进行调节。

（2） 扩大样本数量、训练更多的数据

（3） 对模型中的参数增加正则化（即增加惩罚项，参数越多惩罚越大）

欠拟合则和过拟合刚好相反，我们可以增加模型的参数、或者选择更加复杂的模型；也可以从数据中挖掘更多的特征来增加输入的变量，还可以使用一些集成算法（如装袋法（Bagging），提升法（Boosting））。

（注意：有可能模型的输入和输出一点关系都没有，举个极端的例子，你买的西瓜好坏和你的个人特征没任何关系，例如你的性别身高体重等）

关于欠拟合和过拟合的问题，我这里介绍的只是一个大概的思想。事实上，针对不同的机器学习算法，通常有特定的应对思路，例如在树模型中，我们可以控制树的深度来防止过拟合、神经网络在训练过程中使用 dropout 的策略来缩减参数量避免过拟合。有兴趣深入了解的同学可以学习前面介绍的机器学习通识课程。

以后大家只需要知道这两个概念就可以啦，总而言之我们要保证我们的模型在测试集上表现的足够好。

二、 常用的机器学习算法的思想

注意，下面只是带大家简单了解各种机器学习的思想，不会深入讲解里面的细节。我们的主要目的是教给大家先用起来，至于背后的原理如果你有兴趣可以去看最前面推荐的视频或书籍。

B 站有一个 UP 主叫《五分钟机器学习》，大家可以提前看看他的视频。

2.1 K 最近邻（KNN）

KNN 算法的核心思想是如果一个样本在特征空间中的 K 个最相邻的样本中的大多数属于某一个类别，则该样本也属于这个类别，并具有这个类别上样本的特性。该方法在确定分类决策上只依据最邻近的一个或者几个样本的类别来决定待分样本所属的类别。

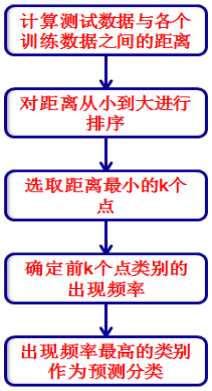


图 1：KNN 算法用于分类的流程图

下面通过一个简单的例子说明一下：如下图，绿色圆要被决定赋予哪个类，是红色三角形还是蓝色四边形？如果 K=3，由于红色三角形所占比例为 2/3，绿色圆将被赋予红色三角形那个类，如果 K=5，由于蓝色四边形比例为 3/5，因此绿色圆被赋予蓝色四边形类。这说明了 KNN 算法的结果很大程度取决于 K 的选择。

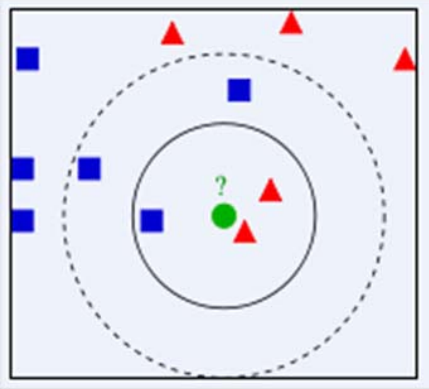


图 2：KNN 简单例子

图 1 来源：https://blog.csdn.net/lyq_12/article/details/81041007

图 2 来源：https://blog.csdn.net/qq_42138454/article/details/105019087

2.2 决策树（Decision Tree）

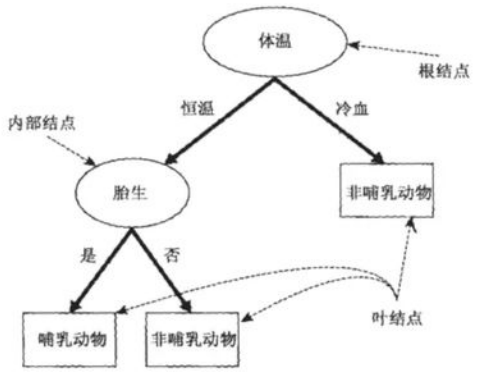
以下内容来自：https://zhuanlan.zhihu.com/p/58592619（菜菜的机器学习 sklearn）

决策树是一种监督学习算法，它能够从一系列有特征（输入数据）和标签（输出数据）的样本中总结出决策规则，并用树状图的结构来呈现这些规则，决策树可以解决分类和回归问题。

我们以分类问题为例，来简单了解一下决策树是如何工作的。决策树算法的本质是一种图结构，我们只需要问一系列问题就可以对数据进行分类了。比如说，来看看下面这组数据集，这是一系列已知物种以及所属类别的数据：

名字	体温	表皮覆盖	胎生	水生动物	飞行动物	有腿	冬眠	类标号
人类	恒温	毛发	是	否	否	是	否	哺乳类
鲑鱼	冷血	鳞片	否	是	否	否	否	鱼类
鲸	恒温	毛发	是	是	否	否	否	哺乳类
青蛙	冷血	无	否	半	否	是	是	两栖类
巨蜥	冷血	鳞片	否	否	否	是	否	爬行类
蝙蝠	恒温	毛发	是	否	是	是	是	哺乳类
鸽子	恒温	羽毛	否	否	是	是	否	鸟类
猫	恒温	软毛	是	否	否	是	否	哺乳类
豹纹鲨	冷血	鳞片	是	是	否	否	否	鱼类
海龟	冷血	鳞片	否	半	否	是	否	爬行类
企鹅	恒温	羽毛	否	半	否	是	否	鸟类
豪猪	恒温	刚毛	是	否	否	是	是	哺乳类
鳗	冷血	鳞片	否	是	否	否	否	鱼类
蠎蠎	冷血	无	否	半	否	是	是	两栖类

我们现在的目标是，将动物们分为哺乳类和非哺乳类。那根据已经收集到的数据，决策树算法为我们算出了下面的这棵决策树：



假如我们现在发现了一种新物种 MATLAB，它是冷血动物，体表带鳞片，并且不是胎生，我们就可以通过这棵决策树来判断它的所属类别。

可以看出，在这个决策过程中，我们一直在对记录的特征进行提问。最初的问题所在的地方叫做根节点，在得到结论前的每一个问题都是中间节点（内部节点），而得到的每一个结论（动物的类别）都叫做叶节点。

决策树算法的核心是要解决两个问题：

1）如何从数据表中找出最佳节点和最佳分枝？

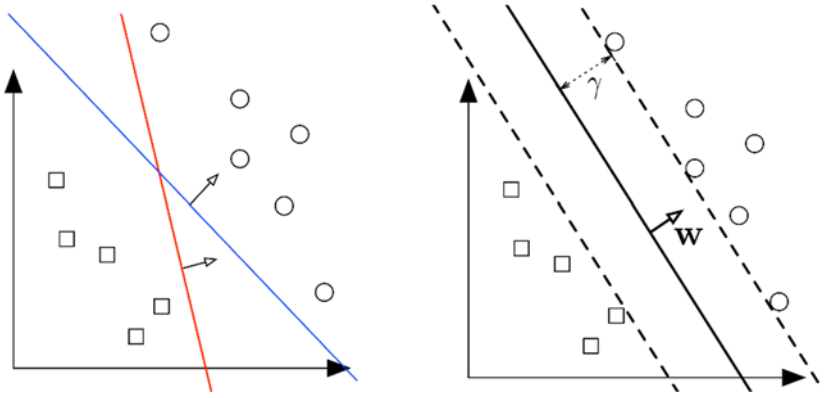
2）如何让决策树停止生长，防止过拟合？

这两个问题背后涉及到的数学原理较为复杂，常见的设计决策树的方法有 ID3、C4.5 和 CART 算法，有兴趣的同学可以看李航老师的教材。

2.3 支持向量机（SVM）

2.3.1 线性支持向量机

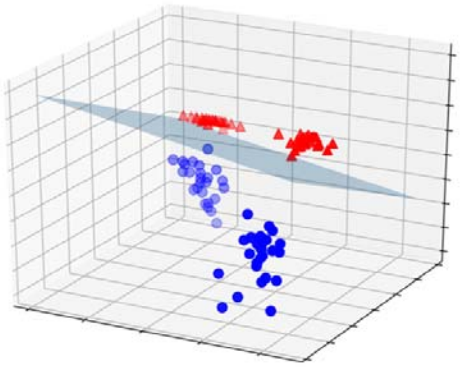
假设我们有两类数据，分别用圆形和方块表示。另外假设我们只有两个输入特征，这样就可以用一个二维的坐标轴上来进行可视化啦，如图所示，横轴上是第一个特征，纵轴上是第二个特征：



我们假设这两类能通过一条直线严格区分开，如图所示：红色和蓝色这两条直线都可以将这两类轻松分离，你也可以找到更多的直线。那么问题来了，哪条直线分类的效果更好？即泛化能力更强。这就是 SVM 要解决的事情。

上面介绍的是只有两个输入特征的例子，我们找到的是一个直线来分开两类。

如果有三个输入特征，则需要画一个三维图形，我们要找到的是一个平面来分开两类。



如果输入的特征更多，那么我们找到的用于分开两类的面称为超平面。

在 SVM 中，定义了一个叫做几何间隔的概念：先计算每个样本点到这个超平面的距离，然后找到这些距离的最小值。

我们假定样本点是线性可分的，即能找到一个超平面将样本分成两类。那么，SVM 要找的超平面就是能正确划分两类数据且让几何间隔达到最大的那个超平面。这种分类方式在 SVM 中称为硬间隔最大化。

（SVM 本质上可以转换成一个优化问题，有目标函数和约束条件）

但是实际上，现实任务中很难确定这样的超平面将我们的样本完全分成两类，所以可以在 SVM 中引入松弛变量，这样允许一些样本出错，但我们希望出错的样本越少越好。（具体可以看周志华或者李航老师的教材）

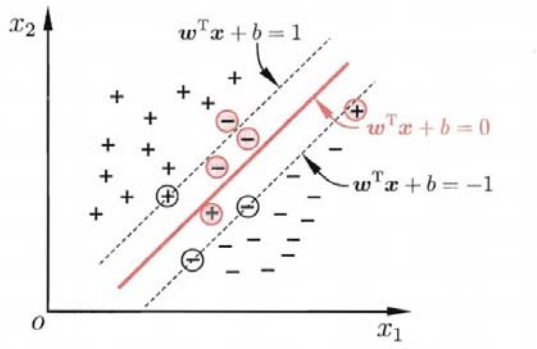
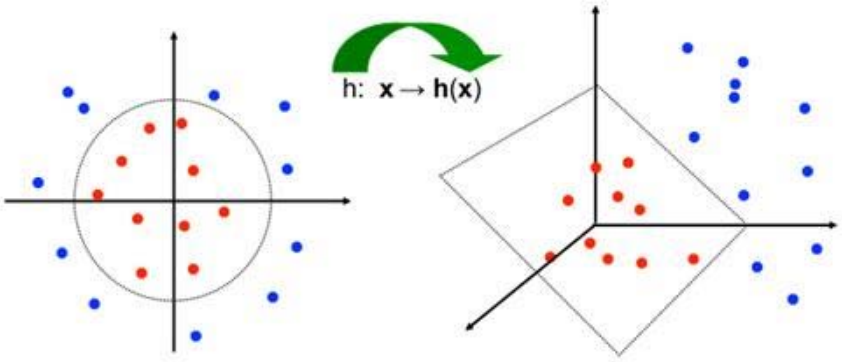


图 6.4 软间隔示意图. 红色圈出了一些不满足约束的样本. （周志华教材）

2.3.2 非线性支持向量机



非线性分类问题是指通过利用非线性模型才能很好地进行分类的问题。上图左侧是一个分类问题，红色和蓝色表示两类点。由图可见，无法用直线(线性模型)将正负实例正确分开，但可以用一条椭圆曲线(非线性模型)将它们正确分开。

假设我们可以通过一个转换函数将低维空间的数据集映射到高维空间的数据集，这时候的数据会变得容易线性可分，如上图右侧所示。

定义 7.6（核函数） 设 $\mathcal{X}$ 是输入空间（欧氏空间 $\mathbf{R}^n$ 的子集或离散集合），又设 $\mathcal{H}$ 为特征空间（希尔伯特空间），如果存在一个从 $\mathcal{X}$ 到 $\mathcal{H}$ 的映射

$$\phi(x):\mathcal{X}\to\mathcal{H} \tag{7.65}$$

使得对所有 $x,z\in\mathcal{X}$，函数 $K(x,z)$ 满足条件

$$K(x,z)=\phi(x)\cdot\phi(z) \tag{7.66}$$

则称 $K(x,z)$ 为核函数，$\phi(x)$ 为映射函数，式中 $\phi(x)\cdot\phi(z)$ 为 $\phi(x)$ 和 $\phi(z)$ 的内积。

核技巧的想法是，在学习与预测中只定义核函数 $K(x,z)$，而不显式地定义映射函数 ϕ。通常，直接计算 $K(x,z)$ 比较容易，而通过 $\phi(x)$ 和 $\phi(z)$ 计算 $K(x,z)$ 并不容易。注意，ϕ 是输入空间 $\mathbf{R}^n$ 到特征空间 $\mathcal{H}$ 的映射，特征空间 $\mathcal{H}$ 一般是高维的，甚至是无穷维的。可以看到，对于给定的核 $K(x,z)$，特征空间 $\mathcal{H}$ 和映射函数 ϕ 的取法并不唯一，可以取不同的特征空间，即便是在同一特征空间里也可以取不同的映射。

（李航教材）

注意，这个映射函数我们一般不用定义出来，我们只需要给定一个核函数就能完成上述这个过程。

3. 核技巧在支持向量机中的应用

我们注意到在线性支持向量机的对偶问题中，无论是目标函数还是决策函数（分离超平面）都只涉及输入实例与实例之间的内积。在对偶问题的目标函数 (7.37) 中的内积 $x_i \cdot x_j$ 可以用核函数 $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$ 来代替。此时对偶问题的目标函数成为

$$W(\alpha) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^{N}\alpha_i \tag{7.67}$$

同样，分类决策函数中的内积也可以用核函数代替，而分类决策函数式成为

$$\begin{aligned} f(x) &= \text{sign}\left(\sum_{i=1}^{N_s} a_i^* y_i \phi(x_i) \cdot \phi(x) + b^*\right) \\ &= \text{sign}\left(\sum_{i=1}^{N_s} a_i^* y_i K(x_i, x) + b^*\right) \end{aligned} \tag{7.68}$$

这等价于经过映射函数 ϕ 将原来的输入空间变换到一个新的特征空间，将输入空间中的内积 $x_i \cdot x_j$ 变换为特征空间中的内积 $\phi(x_i) \cdot \phi(x_j)$，在新的特征空间里从训练样本中学习线性支持向量机。当映射函数是非线性函数时，学习到的含有核函数的支持向量机是非线性分类模型。

也就是说，在核函数 $K(x, z)$ 给定的条件下，可以利用解线性分类问题的方法求解非线性分类问题的支持向量机。学习是隐式地在特征空间进行的，不需要显式地定义特征空间和映射函数。这样的技巧称为核技巧，它是巧妙地利用线性分类学习方法与核函数解决非线性问题的技术。在实际应用中，往往依赖领域知识直接选择核函数，核函数选择的有效性需要通过实验验证。

7.3.3 常用核函数

1. 多项式核函数 (polynomial kernel function)

$$K(x, z) = (x \cdot z + 1)^p \tag{7.88}$$

对应的支持向量机是一个 p 次多项式分类器。在此情形下，分类决策函数成为

$$f(x) = \text{sign}\left(\sum_{i=1}^{N_s} a_i^* y_i (x_i \cdot x + 1)^p + b^*\right) \tag{7.89}$$

2. 高斯核函数 (Gaussian kernel function)

$$K(x, z) = \exp\left(-\frac{\|x - z\|^2}{2\sigma^2}\right) \tag{7.90}$$

对应的支持向量机是高斯径向基函数（radial basis function）分类器。在此情形下，分类决策函数成为

$$f(x) = \text{sign}\left(\sum_{i=1}^{N_s} a_i^* y_i \exp\left(-\frac{\|x - x_i\|^2}{2\sigma^2}\right) + b^*\right) \tag{7.91}$$

至于选择哪种核函数，这个和你的数据有很大关系。你可以把这个问题当成一个模型选择的问题，通过交叉验证的方法尝试不同的模型。

2.4 集成学习算法

集成学习（ensemble learning）是时下非常流行的机器学习算法，它本身不是一个单独的机器学习算法，而是通过在数据上构建多个模型，集成所有模型的建模结果。

我们前面介绍的单一模型可以被称为“个体学习器”，如果是用于分类问题，也可以被称为“弱分类器”，例如决策树。

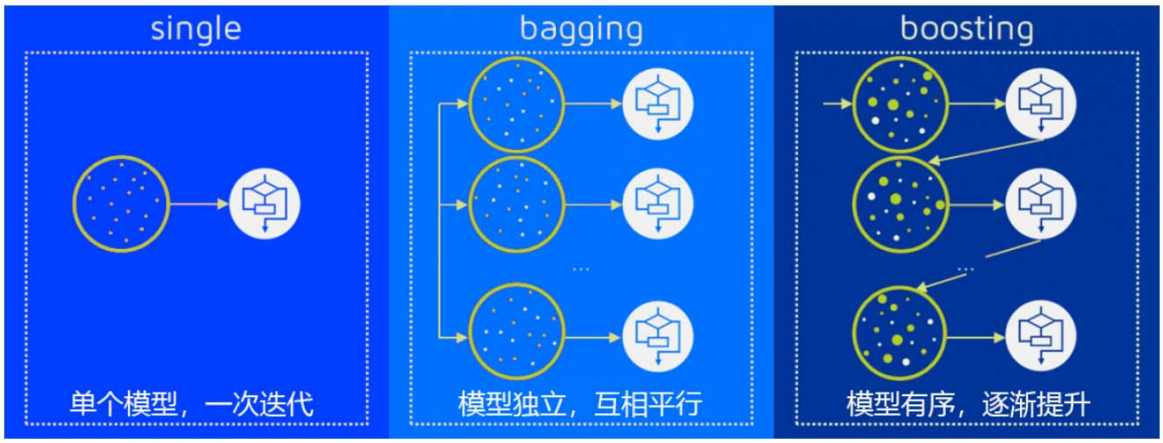
下面的分类参考周志华教材。

根据个体学习器的生成方式，目前的集成学习方法大致可分为两大类：

（1） 个体学习器间存在强依赖关系、必须串行生成的序列化方法。这种方法我们称为提升法（Boosting），其代表模型有 Adboost(自适应提升算法)、GBDT(梯度提升决策树)、Xgboost（极端梯度提升算法）。

（2） 个体学习器间不存在强依赖关系、可同时生成的并行化方法。这种方法我们称为装袋法（Bagging），另外，大家经常听到的随机森林(Random Forest)算法可以视为装袋法（Bagging）的一种变形，它们都是对决策树进行集成。具体的区别可以看教材。

下面这个图很形象，供大家参考：



课后参考的视频：

初步了解可以看这两个短视频：

【五分钟机器学习】Adaboost：前人栽树后人乘凉

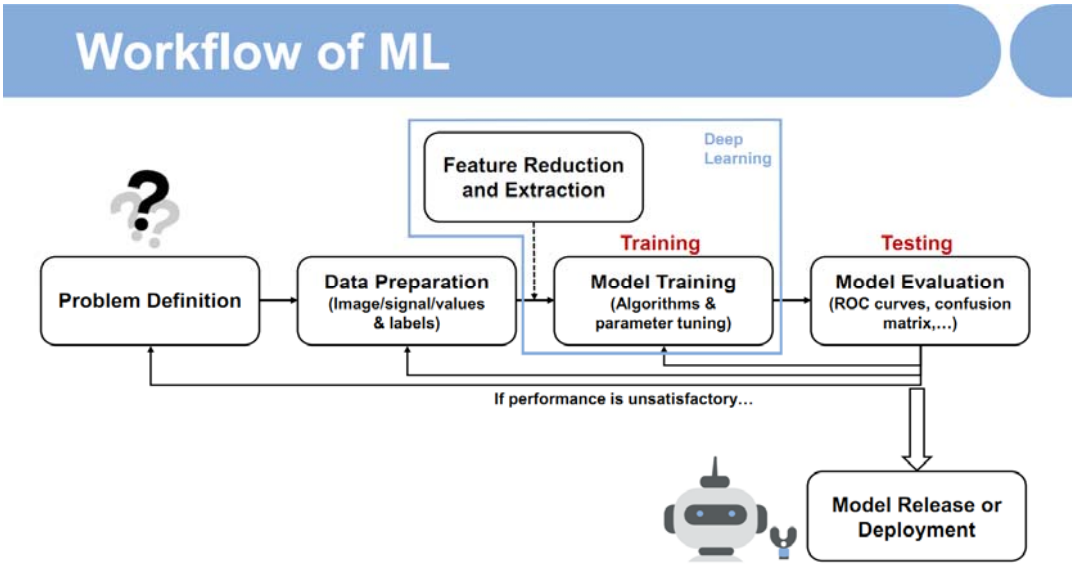
【五分钟机器学习】随机森林（RandomForest）：看我以弱搏强

这里有一个更加详细的介绍视频：

【菊安酱的机器学习】第 6 期 Adaboost 算法

三、 MATLAB 中机器学习的应用

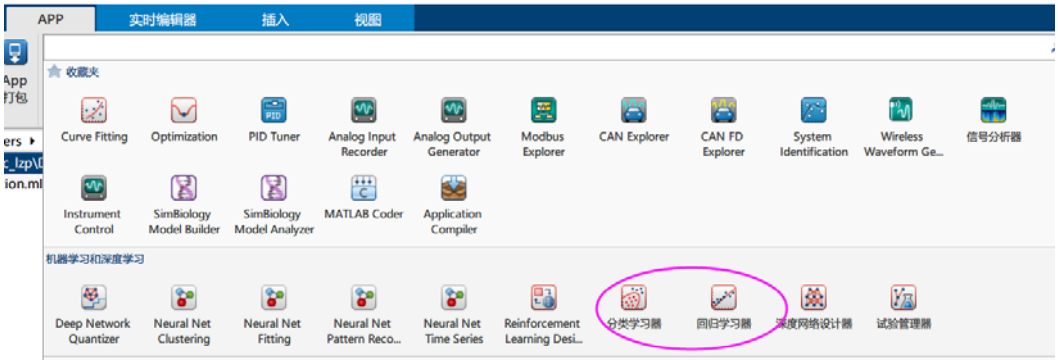
3.1 工具箱介绍



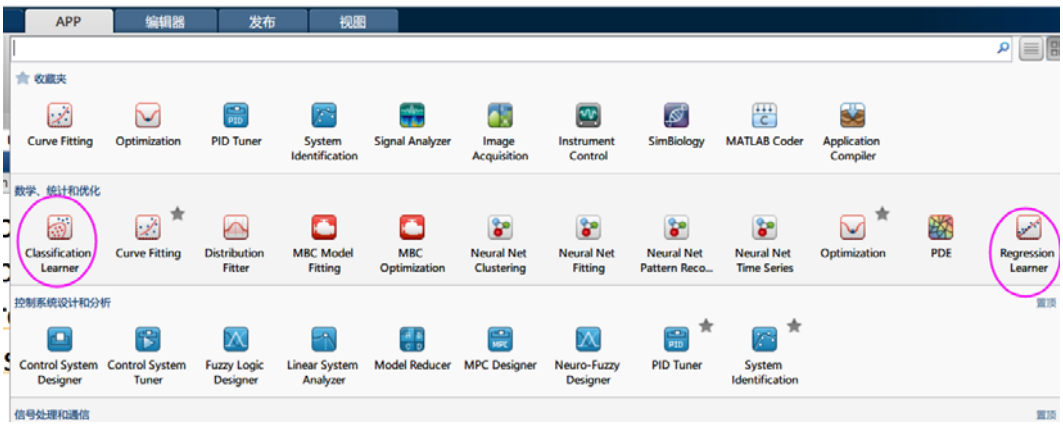
图片作者:Chia-Feng Lu(卢家锋) 台湾阳明大学

需要用到的 MATLAB 工具箱：

2021a 版本：



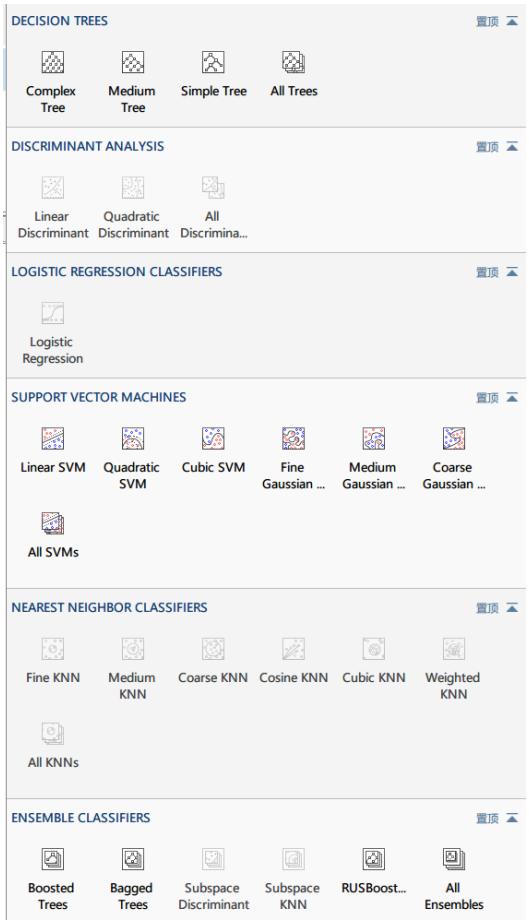
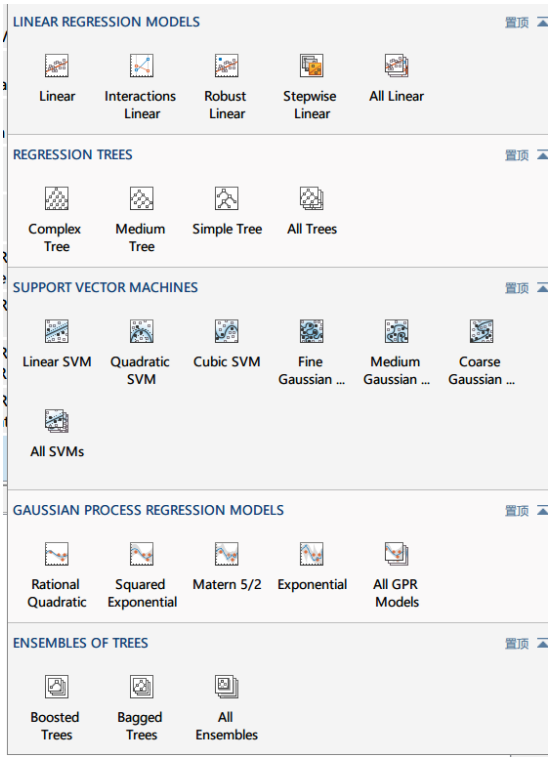
2017a 版本：



2021a 版本中包含的分类模型和回归模型：

分类模型	回归模型
	

2017a 版本中包含的分类模型和回归模型：

分类模型	回归模型
	

3.2 傻瓜操作步骤

请见视频演示，大致包含以下几个步骤：

（1） 调整本地的数据的形式为标准形式
（2） 在 matlab 中导入数据
（3） 选择模型进行训练，比较不同模型的好坏
（4） 使用网格搜索进一步调整参数（进阶：需要写代码）
（5） 使用所有数据进行训练，得到最终模型
（6） 进行预测

3. 3 其他有用的机器学习函数

Neighborhood Component Analysis (NCA)
ReliefF or RReliefF algorithm
minimum redundancy maximum relevance (MRMR) algorithm
参考：Dimensionality Reduction and Feature Extraction

四、 完整的机器学习实战案例

4. 1 泰坦尼克号生存预测

4. 2 影响众筹项目是否成功的因素分析

4. 3 房价预测

没讲完的部分后续通过直播或者录播的方式放在公众号（2021/8/15 前讲完应对国赛）。
大家可以在微信公众号《数学建模学习交流》后台发送“机器学习”四个字获取。