翻译 | AI 科技大本营（rgznai100）
参与 | 林椿眄
编辑 | 波波，Donna

在机器学习领域，“没有免费的午餐” 是一个不变的定理。简而言之，没有一种算法是完美的，可以作为任何问题的最佳解决方案。认清这一点，对于解决监督学习问题（如预测建模问题）尤其重要。

我们不能总说神经网络就是比决策树好，反之亦然。影响算法性能的因素有很多，比如数据集的大小和结构。

因此，对于自己的问题，要尝试多种不同的算法，并使用测试数据集来评估各个算法的性能，以选出效果最优的那一个。

当然，前面所尝试的算法必须要适合自己的问题，这也正是你要选对正确的机器学习任务的地方。比如，需要打扫房子的时候，你会使用真空吸尘器、扫帚或拖把，但绝不应该用铲子在屋内挖坑。

## ▌重要的原则

话虽如此，但所有用于预测建模的有监督机器学习算法却有一个共同的原则：

**机器学习算法的本质是找到一个目标函数（f），使其成为输入变量（X）到输出变量（Y）之间的最佳映射：Y = f（X）**

这是最常见的学习任务，给定任意新的输入变量（X），我们就能预测出输出变量（Y）的值。因为我们不知道目标函数（f）的形式或样子，所以才要机器去把它找出来。不然的话，我们就可以直接用目标函数来进行预测了，而非还要用机器学习算法来学习数据了。

最常见的机器学习类型就是找到最佳映射 Y = f（X），并以此来预测新 X 所对应的 Y 值。这一过程被称为预测建模或预测分析，目标是尽可能到出最为准确的预测。

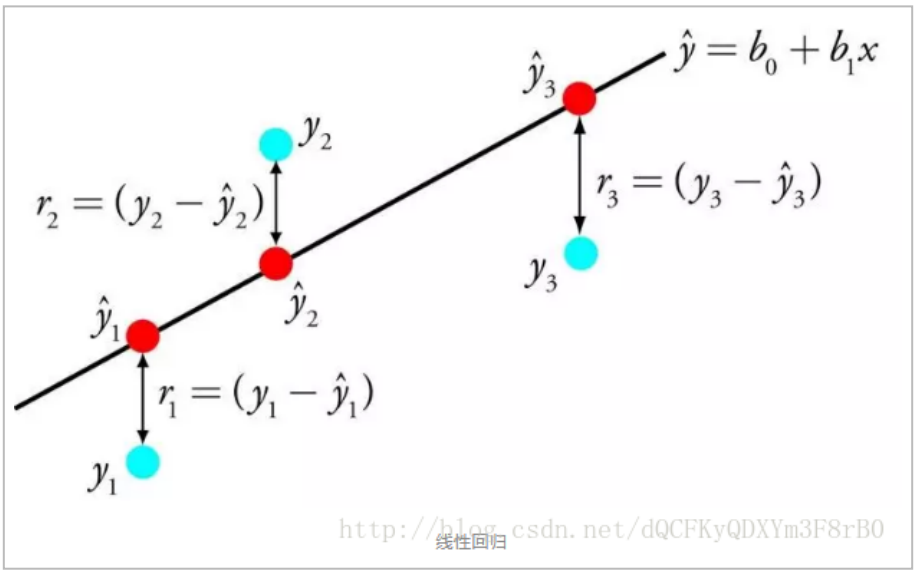
对于渴望理解机器学习基本概念的各位新手，我们特地整理出数据科学家最常用的十大机器学习算法，便于大家快速上手。

## ▌1- 线性回归

线性回归可能是统计学和机器学习中最为知名、最易于理解的一个算法。

预测建模主要关注的是如何最小化模型的误差，或是如何在一个可解释性代价的基础上做出最为准确的预测。我们将借用、重用和窃取包括统计学在内的多个不同领域的算法，并将其用于这些目的。

线性回归所表示的是描述一条直线的方程，通过输入变量的特定权重系数（B）来找出输入变量（x）和输出变量（y）之间最适合的映射关系。



线性回归

线性回归

例如：y = B0 + B1 * x

给定输入 x，我们可以预测出 y 的值。线性回归学习算法的目标是找到系数 B0 和 B1 的值。

找出数据的线性回归模型有多种不同的技巧，例如将线性代数解用于普通最小二乘法和梯度下降优化问题。

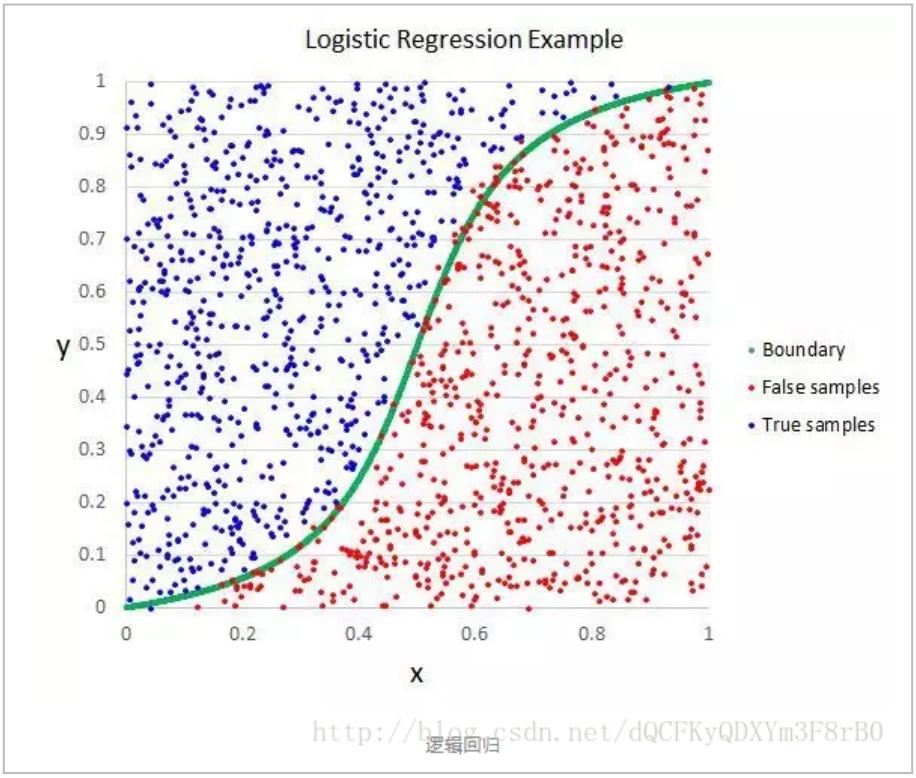
线性回归业已存在 200 多年，并已被广泛研究过。使用该算法的一些窍门，是尽可能地去除非常相似的相关变量以及数据中的噪声。这是一个快速、简单而又好用的算法。

## 2 - 逻辑回归

逻辑回归是机器学习借自统计领域的另一项技术，用于解决二元分类问题（有两个类值的问题）。

逻辑回归就像线性回归，因为它的目标是找出每个输入变量的加权系数值。与线性回归不同的是，逻辑回归预测输出值的函数是非线性的，也被称为逻辑函数。

逻辑回归的函数图像看起来是一个大的 S 形，并将任何值转换至 0 到 1 的区间。这种形式非常有用，因为我们可以用一个规则把逻辑函数的值转化成 0 和 1（例如，如果函数值小于 0.5，则输出 1），从而预测类别。



逻辑回归

基于模型学习的方式，逻辑回归的输出值也可以用来预测给定数据实例属于类别 0 和类别 1 的概率。当你的预测需要更多依据时，这一点会非常有用。

跟线性回归一样，当你剔除与输出变量无关或与之除非常相似（相关）的属性后，逻辑回归的效果会更好。对于二元分类问题，它是一个易于上手、快速而又有效的模型。

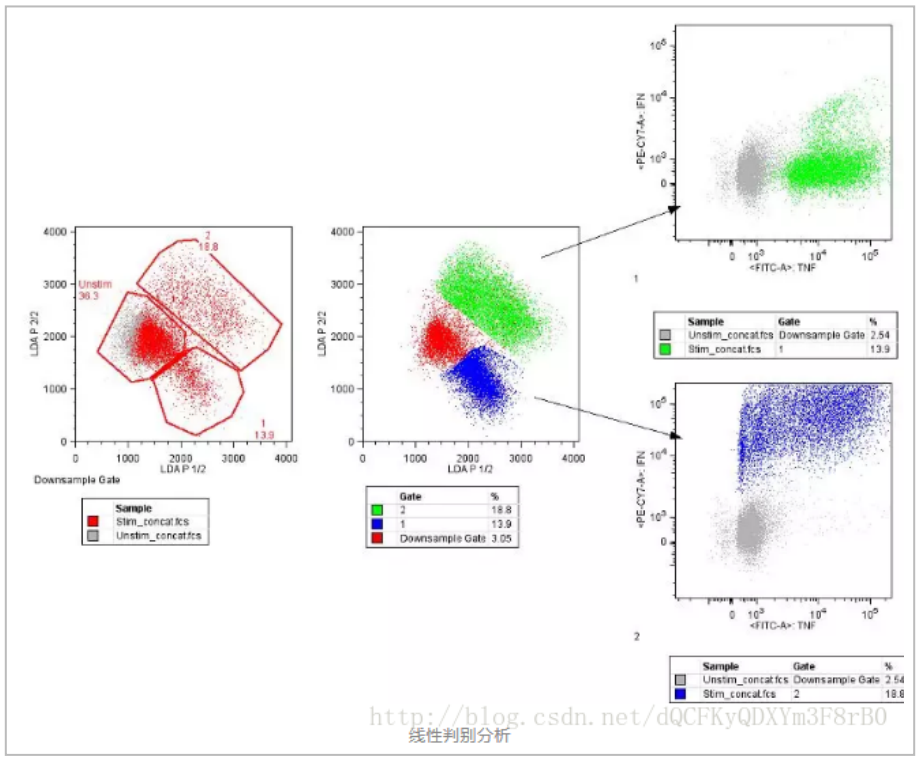
### 3 - 线性判别分析

一般来说，逻辑回归仅限于二元分类问题。 但如果分类类别超过两个，线性判别分析就成为你首选的线性分类算法。

线性判别分析的表达式非常简单。 它由数据的统计属性组成，并计算每个类别的属性值。对于单个输入变量，它包括：

- 每个类别的平均值。

- 所有类别的方差。



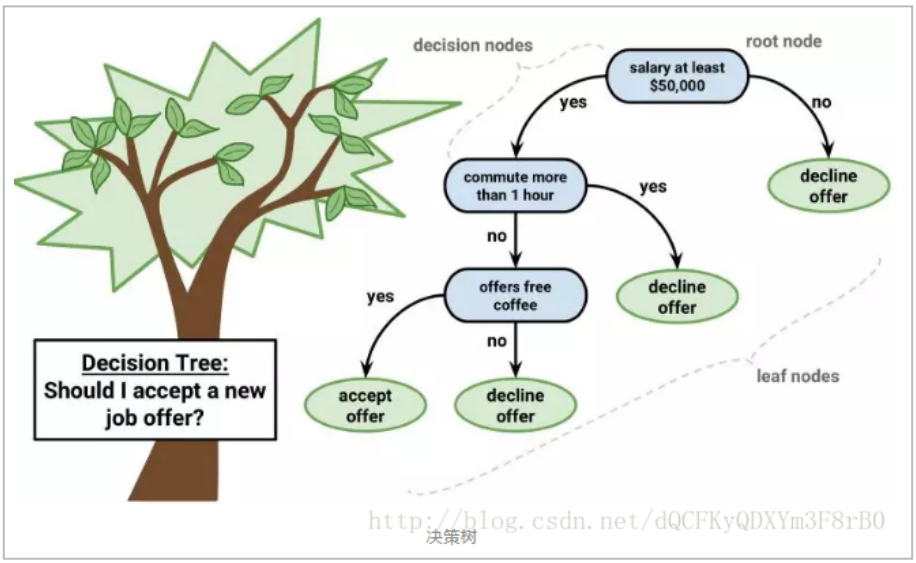

线性判别分析

线性判别分析通过计算每个类别的差别值，并对拥有最大值的类别进行预测。 该方法假定数据服从高斯分布（钟形曲线），因此预测前从数据中移除异常值会是一个很好的习惯。对于分类预测问题来说，它是一个简单而又强大的方法。

## 4 - 分类和回归树

决策树是用于预测建模的一种重要机器学习算法。

决策树模型的表现形式为二叉树，也就是来自算法和数据结构方面的二叉树，没有什么特别。树上每个节点代表一个输入变量（x）与一个基于该变量的分离点（假定这个变量是数字）。



决策树

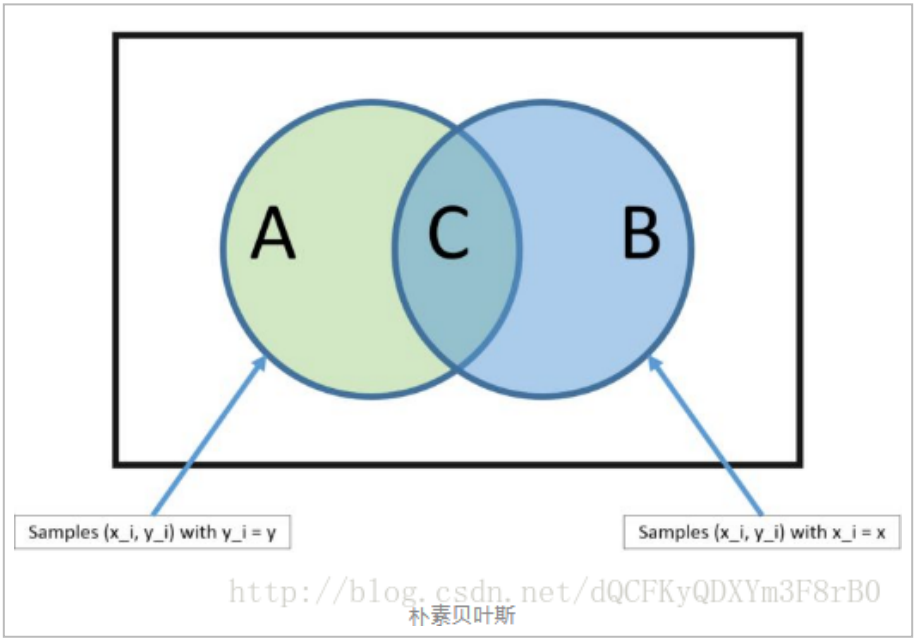
叶节点包含了用于预测的输出变量（y）。预测是通过遍历树的分离点开始，直到抵达每一个叶节点，并输出该叶节点的分类值。

决策树算法学习起来很快，预测速度也很快。决策树对于各种各样的问题都能做出准确的预测，并且无需对数据做任何特殊的预处理。

## 5 - 朴素贝叶斯

朴素贝叶斯是一种简单而又强大的预测建模算法。

该模型由两种概率组成，它们都能从训练数据中直接计算出来：1）每个类别的概率; 2）对于给定的 x 值，每个类别的条件概率。 一旦计算出来，概率模型就可以用于使用贝叶斯定理对新的数据进行预测。 当你的数据是实值时，通常会假定一个高斯分布（钟形曲线），这样你就很容易计算出这些数据的概率。



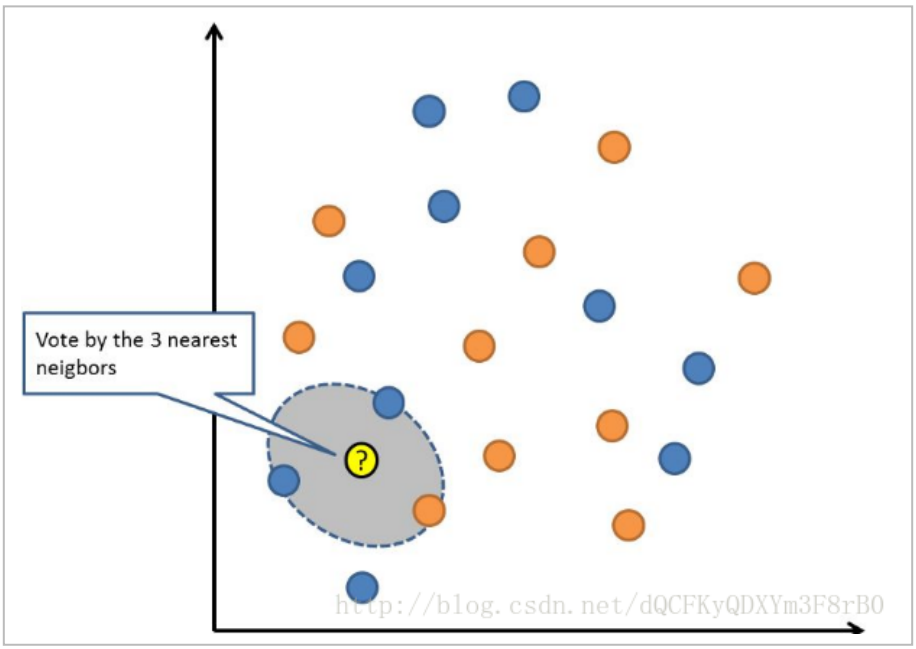

朴素贝叶斯假定每个输入变量都是独立，所以被称为 “朴素的”。这是一个强假设，对真实数据而言有点不切实际，但该方法在大范围的复杂问题上非常有效。

### 6 - K - 最近邻算法

K - 最近邻算法是一种非常简单和有效。它的模型所表示是整个训练数据集，看上去很简单，对吧？

对于给定的训练数据，通过搜索整个数据集中 K 个最相似的实例（邻居），汇总这 K 个实例的输出变量可以预测新的数据点。对于回归问题，它可能是输出变量的平均值；对于分类问题，它可能是模式（或最常见的）类别值。

使用 K - 最近邻算法的诀窍，是在于如何确定数据实例之间的相似性。最简单的方法，如果你的属性在欧几里德距离上尺度相同（例如均以英寸为单位），那么基于每个输入变量之间的差异，你就可以直接计算其数值来确定相似性。

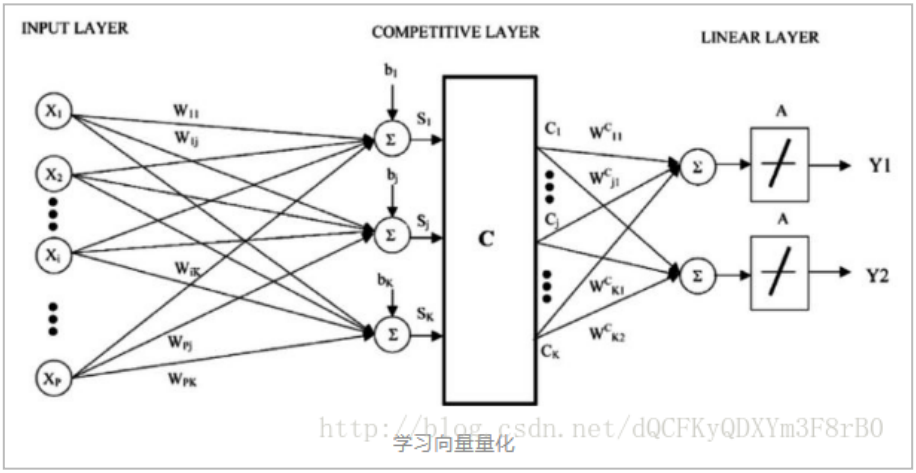



K - 最近邻算法可能需要大量的内存或存储空间来储存所有数据，但只有在预测时才会执行计算（或学习）。你也可以随时更新和管理你的训练实例，以保持预测的准确性。

距离或紧密度的概念在非常高的维度（大量的输入变量）中可能会失效，因为输入变量的数量对于算法性能有着很大的负面影响。这就是维度灾难。这就要求你只使用那些与预测输出变量最相关的输入变量。

### 7 - 学习向量量化

K - 最近邻算法的一个缺点是你需要使用整个训练数据集。而作为人工神经网络，学习向量量化算法（简称 LVQ）允许你选择训练实例的数量，并能准确地学习这些实例所应有的特征。




学习向量量化

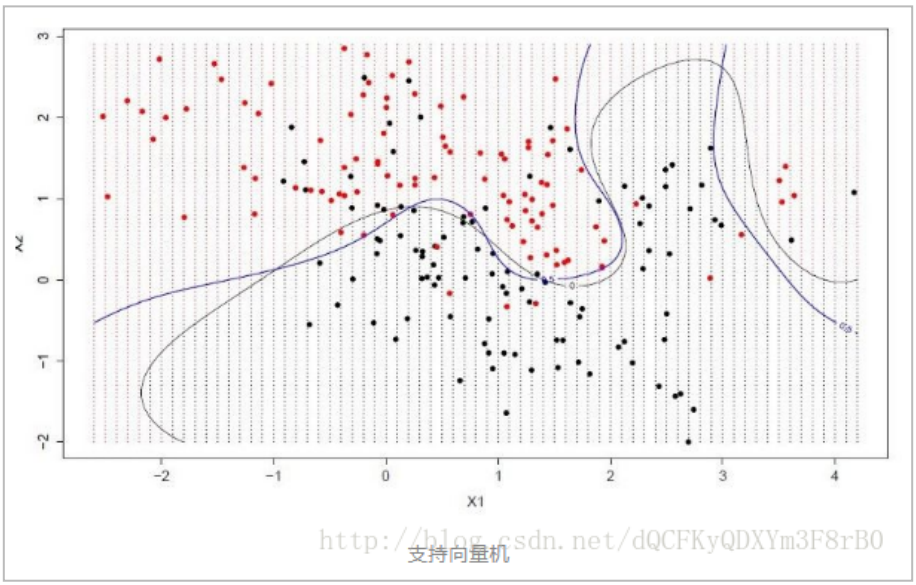
学习向量量化算法所表示的是码本向量的集合。这些向量在初始化的时候随机选择出来，并在学习算法的多次迭代中优化成最能概括训练数据集的集合。在学习完成后，码本向量可以像 K - 最近邻算法一样进行预测。通过计算每个码本向量和新数据实例之间的距离来找到最相似的邻居（最佳匹配码本向量），然后返回最佳匹配单元的类别值或（在回归情况下的实际值）作为预测。如果能重新调整数据使其处于相同的区间（如 0 到 1 之间），则可以获得最佳的预测结果。

如果 K - 最近邻算法在你的数据集上已经给出了很好的预测结果，那么可以尝试用学习向量量化算法来减少整个训练数据集的内存存储需求。

## 8 - 支持向量机

支持向量机可能是最受欢迎、讨论最为广泛的机器学习算法之一。

超平面是输入变量空间内的一条分割线。在支持向量机中，超平面可以通过类别（0 类或 1 类）最佳分割输入变量空间。在二维空间内，超平面可被视为一条线，我们假设所有的输入点都可以被该线完全分开。支持向量机的目标是找到一个分离系数，让一个超平面能够对不同类别的数据进行最佳分割。



支持向量机

超平面与最近的数据点之间的距离被称为边距。在分离两个类上具有最大边距的超平面被称为最佳超平面。超平面的确定只跟这些点及分类器的构造有关。这些点被称为支持向量，它们支持并定义超平面。在实践中，可以使用优化算法来找到能够最大化边距的系数。

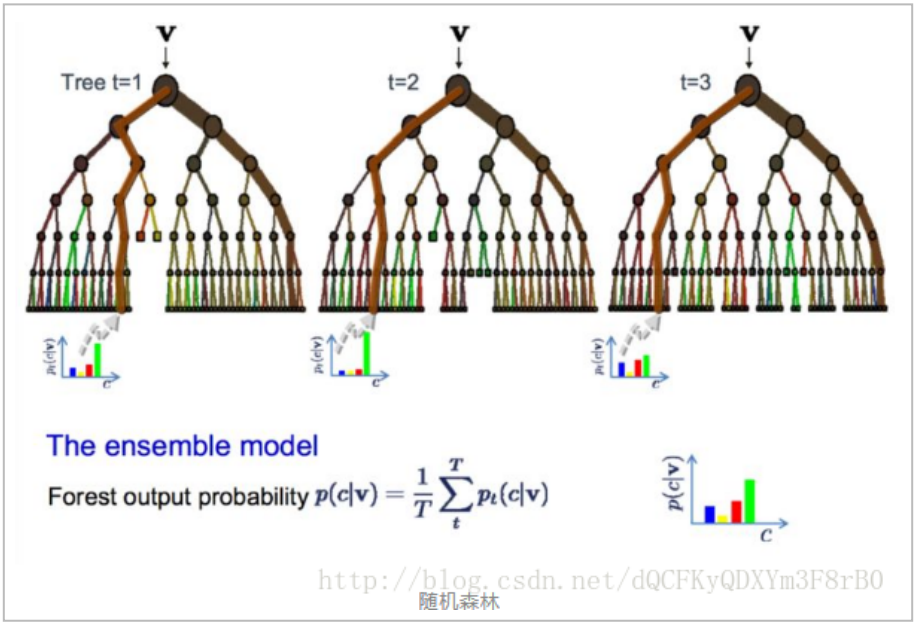
支持向量机可能是最为强大的 “开箱即用” 分类器之一，值得你尝试。

## ▌9 - bagging 算法和随机森林

随机森林是最流行、最强大的机器学习算法之一。它是一种被称为 Bootstrap Aggregation 或 Bagging 的机器学习集成算法。

Bootstrap 是一种从数据样本中估算数量的强大统计方法。换句话说，你需要抽取大量的数据样本、计算平均值，然后再计算所有均值的平均，以便更好地估计整体样本的真实平均值。

bagging 算法也使用相同的方式，但用于估计整个统计模型的最常见方法是决策树。训练数据中的多个样本将被取样，然后对每个数据样本建模。对新数据进行预测时，每个模型都会进行预测，并对每个预测结果进行平均，以更好地估计真实的输出值。



随机森林

随机森林是对 bagging 算法的一种调整，它不是选择最佳分割点来创建决策树，而是通过引入随机性来得到次优分割点。

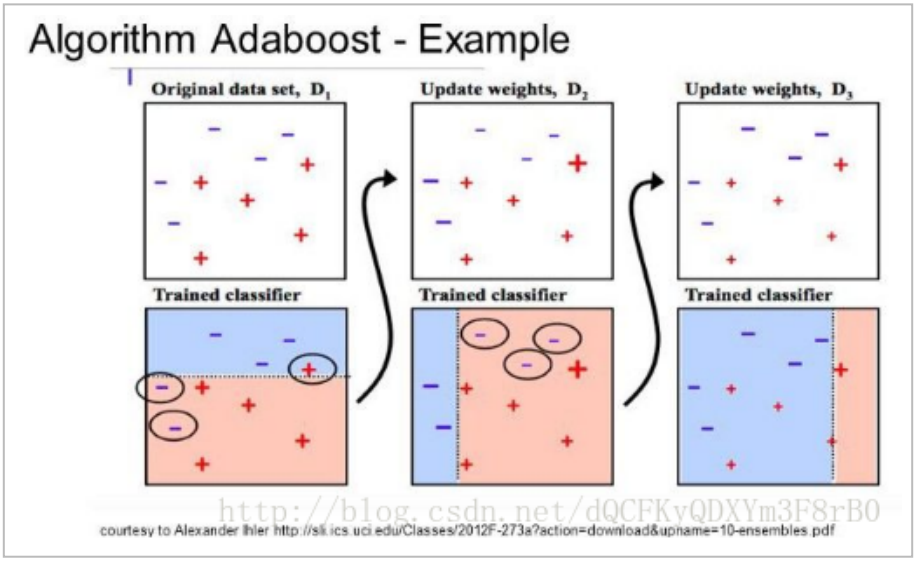
因此，针对每个数据样本所创建的模型，会与其他方式有所不同，但仍能以其独特和不同的方式准确预测。结合所有模型的预测，可以更好地估计潜在的真实输出。

如果用方差较高的算法（如决策树）能够获得较好的结果，那么通过 bagging 算法通常可以获得更好的结果。

## 10 - Boosting 和 AdaBoost 算法

Boosting 是一项从多个弱分类器中构建强分类器的集成预测技术。它从训练数据中构建模型，然后通过修正前一个模型的错误创造出第二个模型。以此类推，模型不断叠加，直至能够完美预测训练数据集，或达到可添加的模型的数量上限。

在针对二元分类所开发的 boosting 算法中，AdaBoost 是第一个成功的。它是理解 boosting 算法的最佳起点。现代 boosting 方法基于 AdaBoost 而构建，最典型的例子是随机梯度加速器。




通常，AdaBoost 算法与决策树一起工作。第一个决策树创建后，决策树在每个训练实例上的性能，都被用来衡量下一个决策树针对该实例所应分配的关注程度。难以预测的训练数据被赋予更大的权重，而容易预测的数据则被赋予更小的权重。模型依次被创建，每次更新训练实例的权重，都会影响到序列中下一个决策树学习性能。所有决策树完成后，即可对新输入的数据进行预测，而每个决策树的性能将由它在训练数据上的准确度所决定。

由于模型注意力都集中于纠正上一个算法的错误，所以必须确保数据是干净无异常的。

## 最后的建议

初学者常常会被眼花缭乱的机器学习算法所困扰，提出 “我该使用哪种算法？” 这样的问题。

此问题的答案取决于许多因素，包括：

（1）数据的大小、质量和性质;
（2）可用的计算时间;
（3）任务的紧迫性;
（4）你想要用数据来做什么。

即使是一位经验丰富的数据科学家，在尝试不同的算法之前，也无法回答哪种算法的性能会是最好的。机器学习的算法还有很多，但以上这些是最受欢迎的算法。如果你刚入门机器学习，这

将是一个很好的学习起点。

作者 | James Le
原文链接 | https://towardsdatascience.com/a-tour-of-the-top-10-algorithms-for-machine-learning-newbies-dde4edffae11 (https://towardsdatascience.com/a-tour-of-the-top-10-algorithms-for-machine-learning-newbies-dde4edffae11)

---

全文完