圖解最常用的10個機器學習算法!

3D視覺工程師 2022-03-02 21:54

作者：james_aka_yale

鏈接：https://medium.com/

編輯：王萌(深度學習衝鴨公眾號)

在機器學習領域，有種說法叫做“世上沒有免費的午餐”，簡而言之，它是指沒有任何一種算法能在每個問題上都能有最好的效果，這個理論在監督學習方面體現得尤為重要。

舉個例子來說，你不能說神經網絡永遠比決策樹好，反之亦然。模型運行被許多因素左右，例如數據集的大小和結構。

因此，你應該根據你的問題嘗試許多不同的算法，同時使用數據測試集來評估性能並選出最優項。

當然，你嘗試的算法必須和你的問題相切合，其中的門道便是機器學習的主要任務。打個比方，如果你想打掃房子，你可能會用到吸塵器、掃帚或者拖把，但你肯定不會拿把鏟子開始挖坑吧。

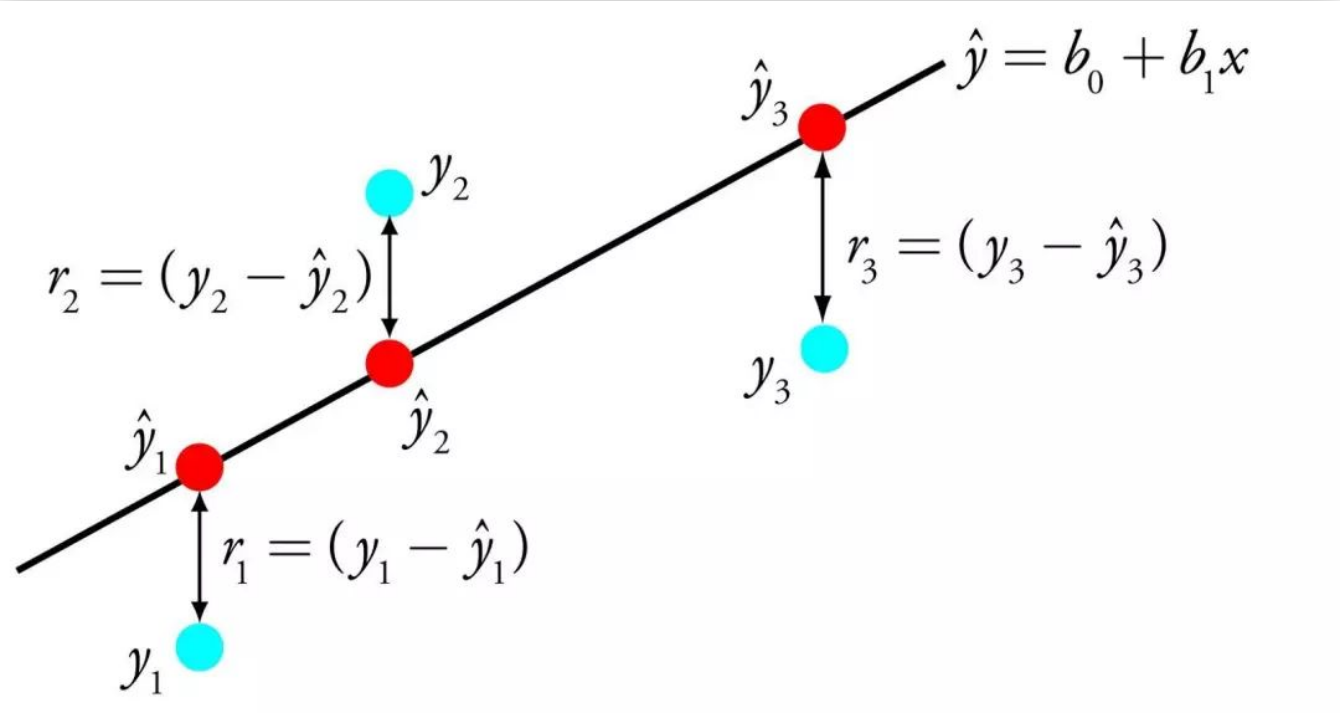
對於渴望了解機器學習基礎知識的機器學習新人來說，這兒有份數據科學家使用的十大機器學習算法，**為你介紹這十大算法的特性，便於大家更好地理解和應用，快來看看吧。**

01 線性回歸

線性回歸可能是統計學和機器學習中最知名和最易理解的算法之一。

由於預測建模主要關注最小化模型的誤差，或者以可解釋性為代價來做出最準確的預測。我們會從許多不同領域借用、重用和盜用算法，其中涉及一些統計學知識。

線性回歸用一個等式表示，通過找到輸入變量的特定權重（B），來描述輸入變量（x）與輸出變量（y）之間的線性關係。



Linear Regression

舉例：y = B0 + B1 * x

给定输入x，我们将预测y，线性回归学习算法的目标是找到系数B0和B1的值。

可以使用不同的技术从数据中学习线性回归模型，例如用于普通最小二乘和梯度下降优化的线性代数解。

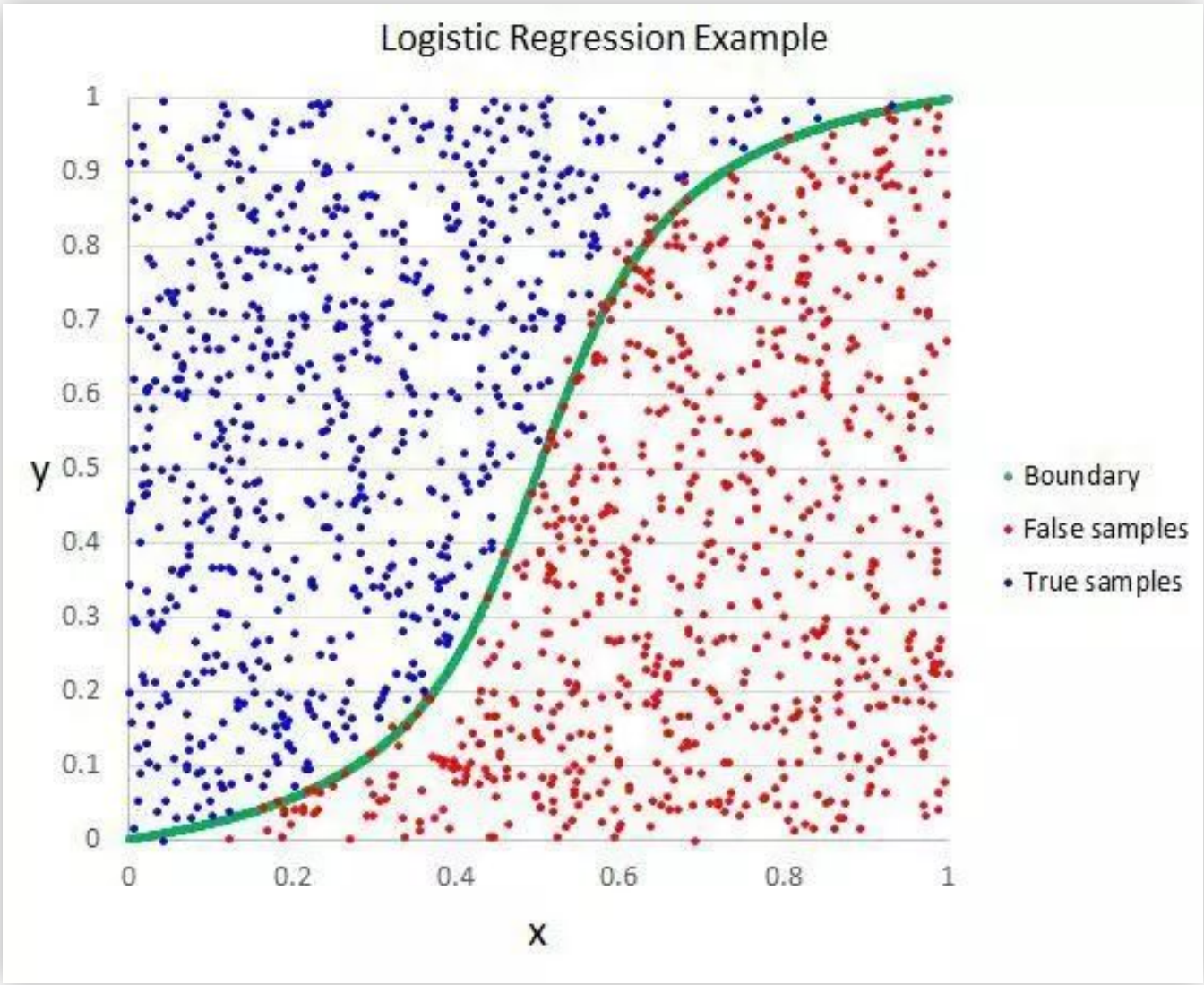
线性回归已经存在了200多年，并且已经进行了广泛的研究。 如果可能的话，使用这种技术时的一些经验法则是去除非常相似（相关）的变量并从数据中移除噪声。 这是一种快速简单的技术和良好的第一种算法。

02 逻辑回归

逻辑回归是机器学习从统计领域借鉴的另一种技术。 这是二分类问题的专用方法（两个类值的问题）。

逻辑回归与线性回归类似，这是因为两者的目标都是找出每个输入变量的权重值。 与线性回归不同的是，输出的预测值得使用称为逻辑函数的非线性函数进行变换。

逻辑函数看起来像一个大S，并能将任何值转换为0到1的范围内。这很有用，因为我们可以将相应规则应用于逻辑函数的输出上，把值分类为0和1（例如，如果IF小于0.5，那么 输出1）并预测类别值。



Logistic Regression

由于模型的特有学习方式，通过逻辑回归所做的预测也可以用于计算属于类0或类1的概率。这对于需要给出许多基本原理的问题十分有用。

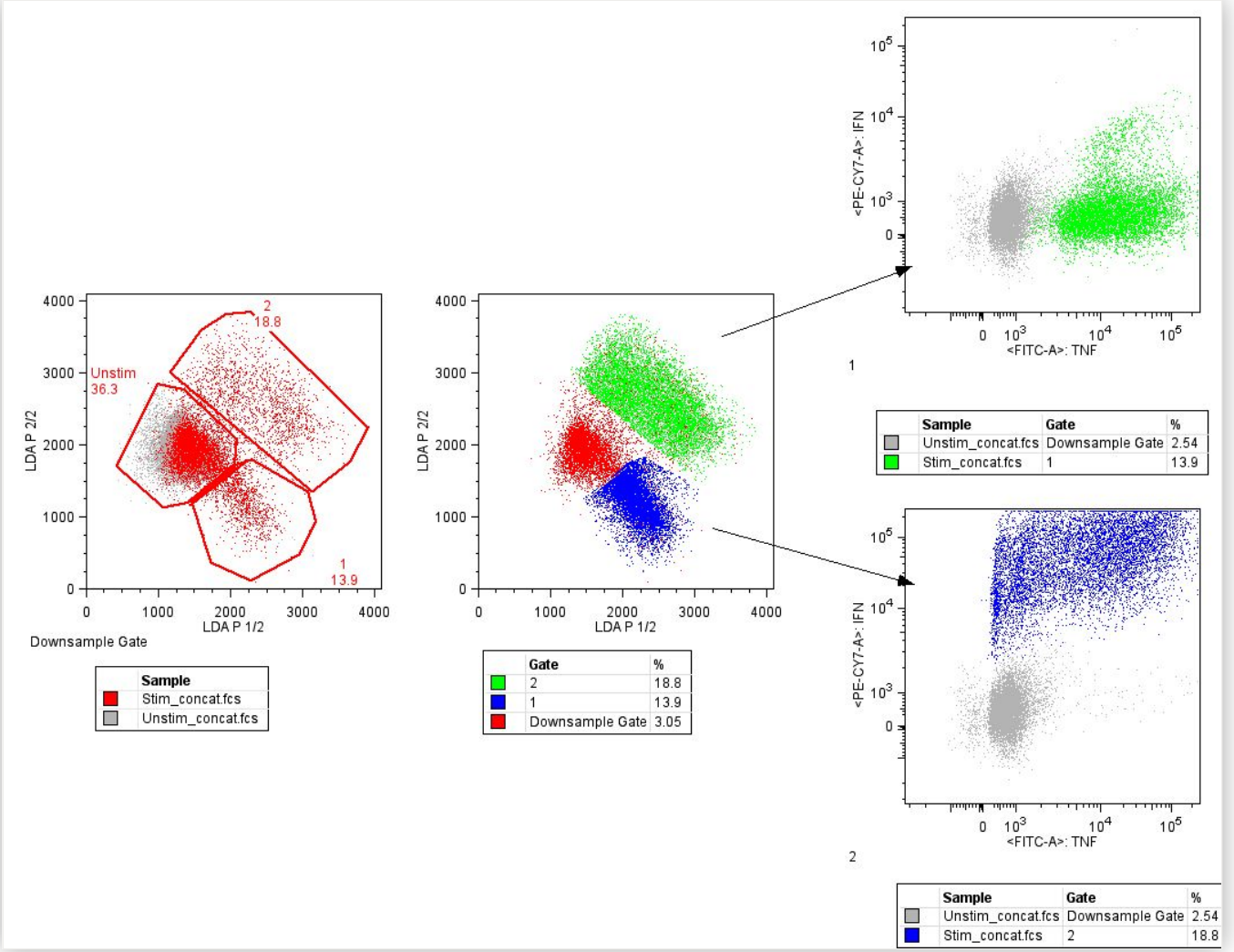
与线性回归一样，当你移除与输出变量无关的属性以及彼此非常相似（相关）的属性时，逻辑回归确实会更好。 这是一个快速学习和有效处理二元分类问题的模型。

03 线性判别分析

传统的逻辑回归仅限于二分类问题。 如果你有两个以上的类，那么线性判别分析算法（Linear Discriminant Analysis，简称LDA）是首选的线性分类技术。

LDA的表示非常简单。 它由你的数据的统计属性组成，根据每个类别进行计算。 对于单个输入变量，这包括：

- **每类的平均值。**
- **跨所有类别计算的方差。**



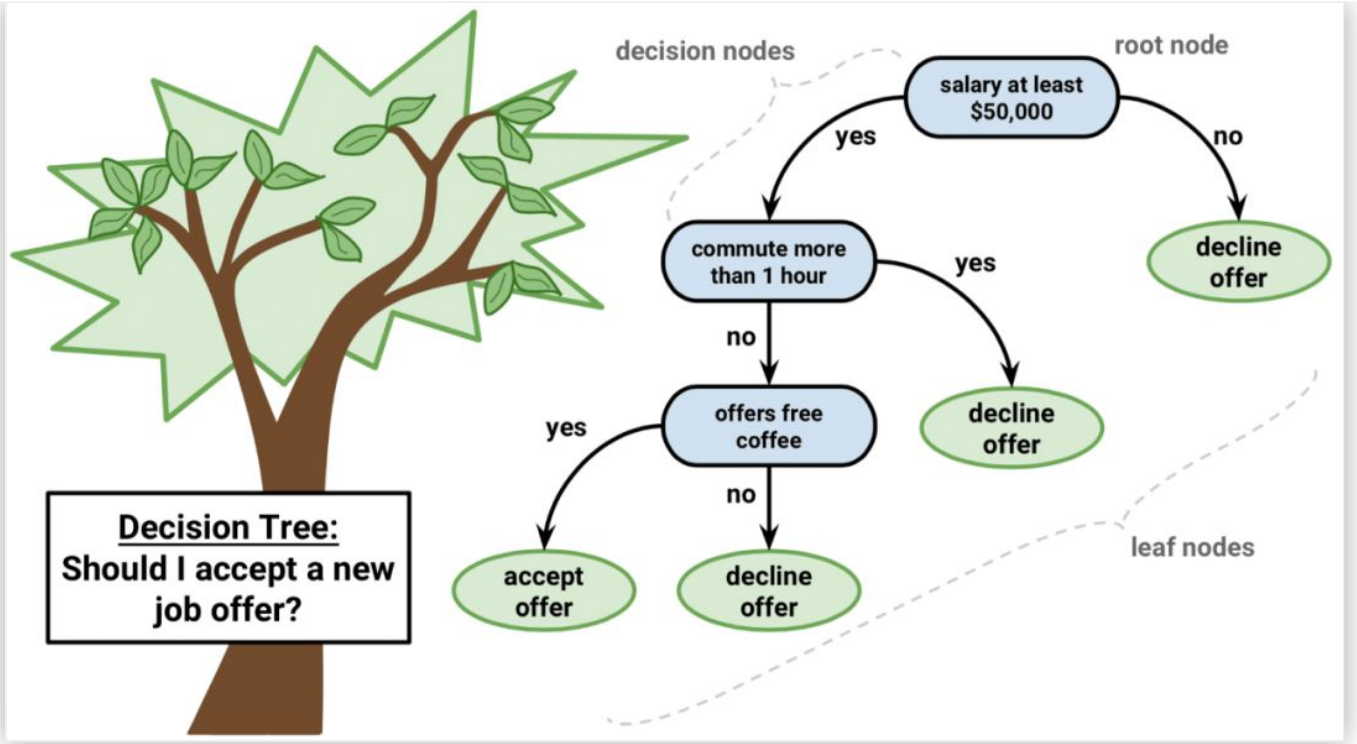
Linear Discriminant Analysis

LDA通过计算每个类的判别值并对具有最大值的类进行预测来进行。该技术假定数据具有高斯分布（钟形曲线），因此最好先手动从数据中移除异常值。这是分类预测建模问题中的一种简单而强大的方法。

04 分类和回归树

决策树是机器学习的一种重要算法。

决策树模型可用二叉树表示。对，就是来自算法和数据结构的二叉树，没什么特别。 每个节点代表单个输入变量（x）和该变量上的左右孩子（假定变量是数字）。



Decision Tree

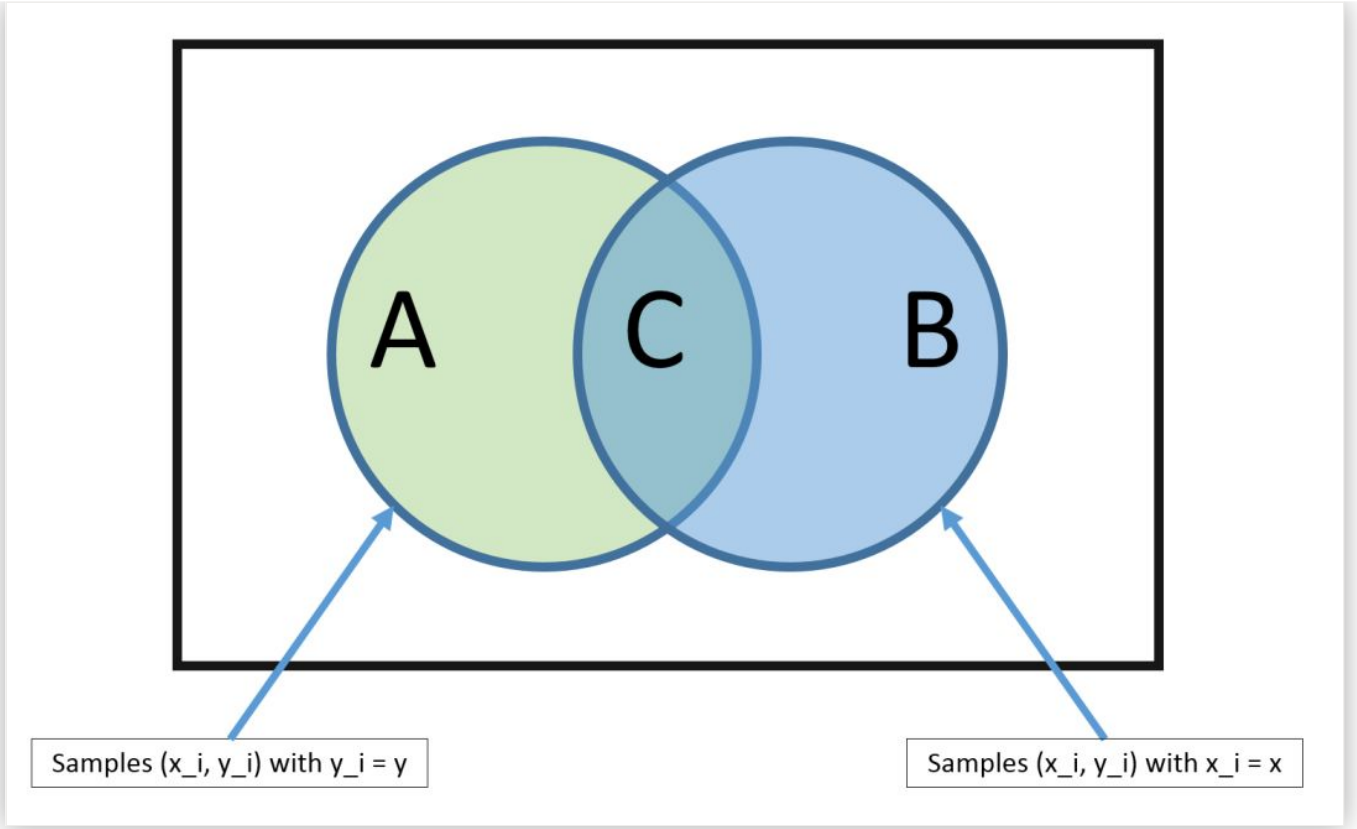
树的叶节点包含用于进行预测的输出变量（y）。 预测是通过遍历树进行的，当达到某一叶节点时停止，并输出该叶节点的类值。

决策树学习速度快，预测速度快。 对于许多问题也经常预测准确，并且你不需要为数据做任何特殊准备。

05 朴素贝叶斯

朴素贝叶斯是一种简单但极为强大的预测建模算法。

该模型由两种类型的概率组成，可以直接从你的训练数据中计算出来：1）每个类别的概率；2）给定的每个x值的类别的条件概率。 一旦计算出来，概率模型就可以用于使用贝叶斯定理对新数据进行预测。 当你的数据是数值时，通常假设高斯分布（钟形曲线），以便可以轻松估计这些概率。



Bayes Theorem

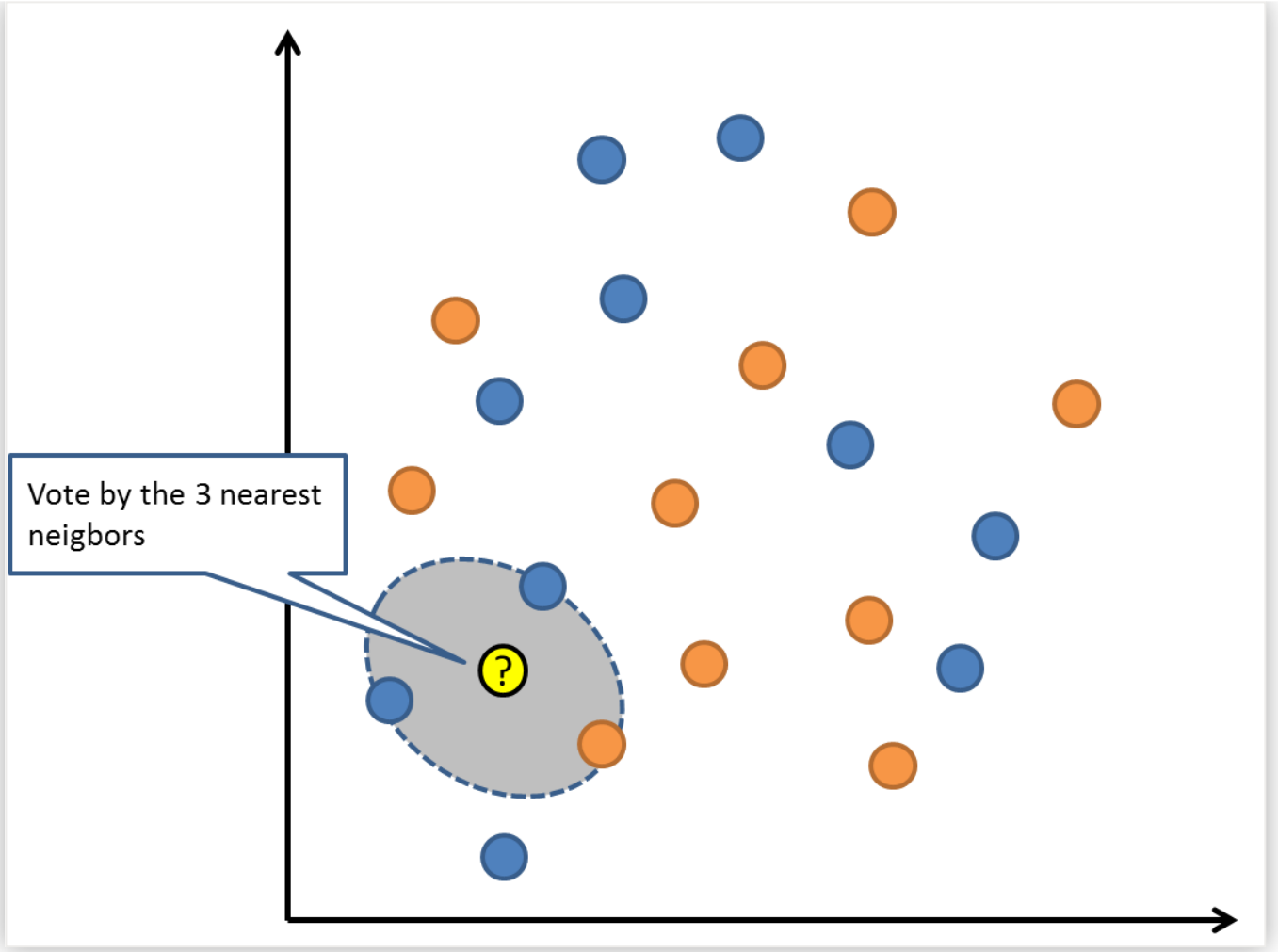
朴素贝叶斯被称为朴素的原因，在于它假设每个输入变量是独立的。 这是一个强硬的假设，对于真实数据来说是不切实际的，但该技术对于大范围内的复杂问题仍非常有效。

06 K近邻

KNN算法非常简单而且非常有效。KNN的模型用整个训练数据集表示。 是不是特简单？

通过搜索整个训练集内K个最相似的实例（邻居），并对这些K个实例的输出变量进行汇总，来预测新的数据点。 对于回归问题，新的点可能是平均输出变量，对于分类问题，新的点可能是众数类别值。

成功的诀窍在于如何确定数据实例之间的相似性。如果你的属性都是相同的比例，最简单的方法就是使用欧几里德距离，它可以根据每个输入变量之间的差直接计算。



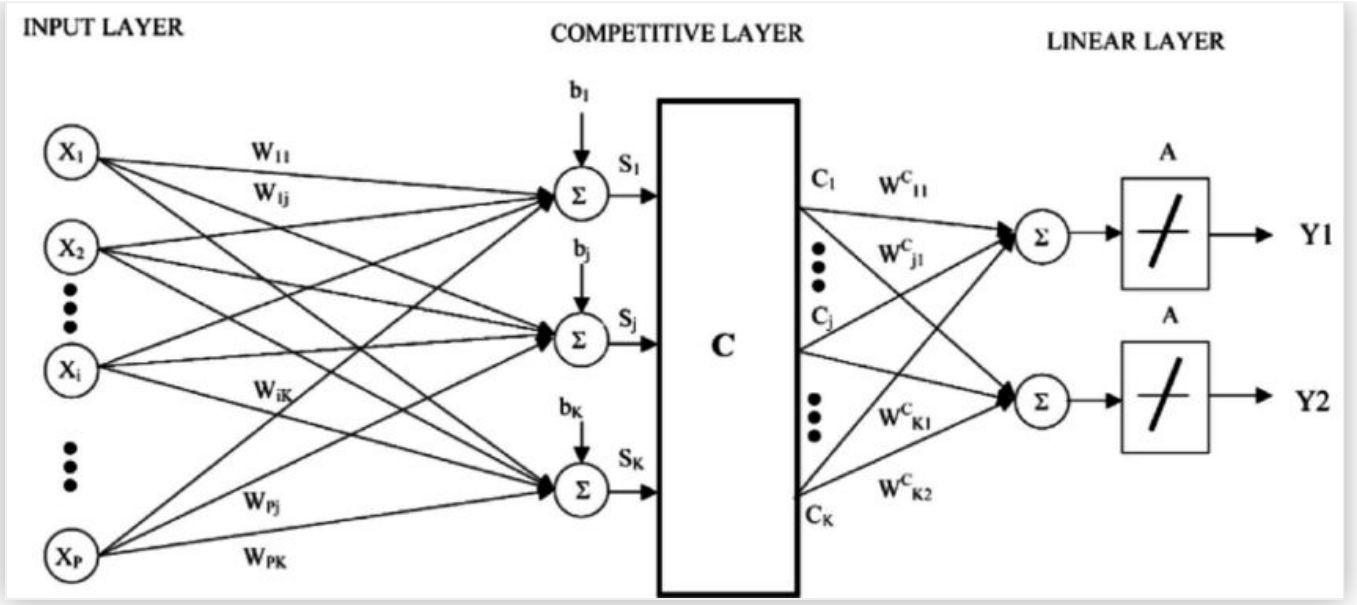
K-Nearest Neighbors

KNN可能需要大量的内存或空间来存储所有的数据，但只有在需要预测时才会执行计算（或学习）。 你还可以随时更新和管理你的训练集，以保持预测的准确性。

距离或紧密度的概念可能会在高维环境（大量输入变量）下崩溃，这会对算法造成负面影响。这类事件被称为维度诅咒。它也暗示了你应该只使用那些与预测输出变量最相关的输入变量。

07 学习矢量量化

K-近邻的缺点是你需要维持整个训练数据集。 学习矢量量化算法（或简称LVQ）是一种人工神经网络算法，允许你挂起任意个训练实例并准确学习他们。



Learning Vector Quantization

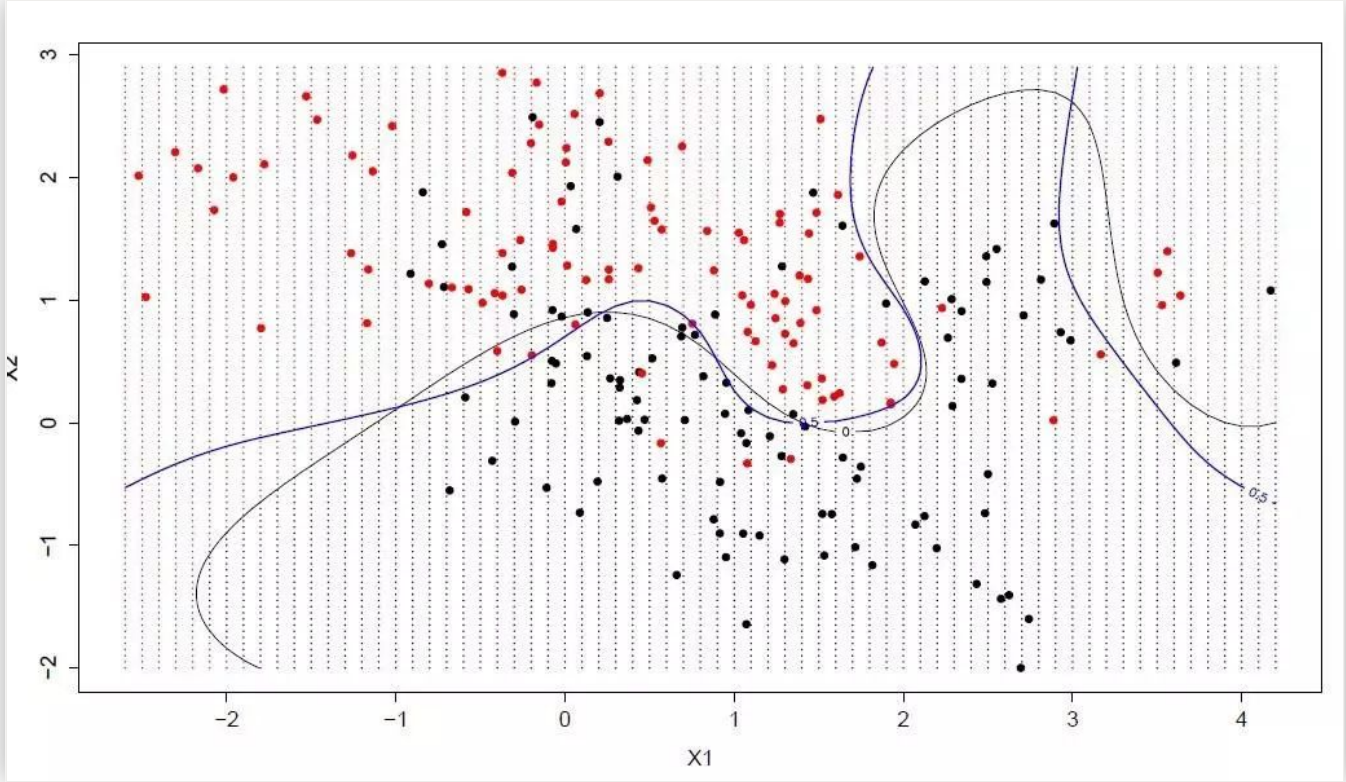
LVQ用codebook向量的集合表示。开始时随机选择向量，然后多次迭代，适应训练数据集。 在学习之后，codebook向量可以像K-近邻那样用来预测。 通过计算每个codebook向量与新数据实例之间的距离来找到最相似的邻居（最佳匹配），然后返回最佳匹配单元的类别值或在回归情况下的实际值作为预测。 如果你把数据限制在相同范围（如0到1之间），则可以获得最佳结果。

如果你发现KNN在您的数据集上给出了很好的结果，请尝试使用LVQ来减少存储整个训练数据集的内存要求。

08 支持向量机

支持向量机也许是最受欢迎和讨论的机器学习算法之一。

超平面是分割输入变量空间的线。 在SVM中，会选出一个超平面以将输入变量空间中的点按其类别（0类或1类）进行分离。在二维空间中可以将其视为一条线，所有的输入点都可以被这条线完全分开。SVM学习算法就是要找到能让超平面对类别有最佳分离的系数。



Support Vector Machine

超平面和最近的数据点之间的距离被称为边界，有最大边界的超平面是最佳之选。同时，只有这些离得近的数据点才和超平面的定义和分类器的构造有关，这些点被称为支持向量，他们支持或定义超平面。在具体实践中，我们会用到优化算法来找到能最大化边界的系数值。

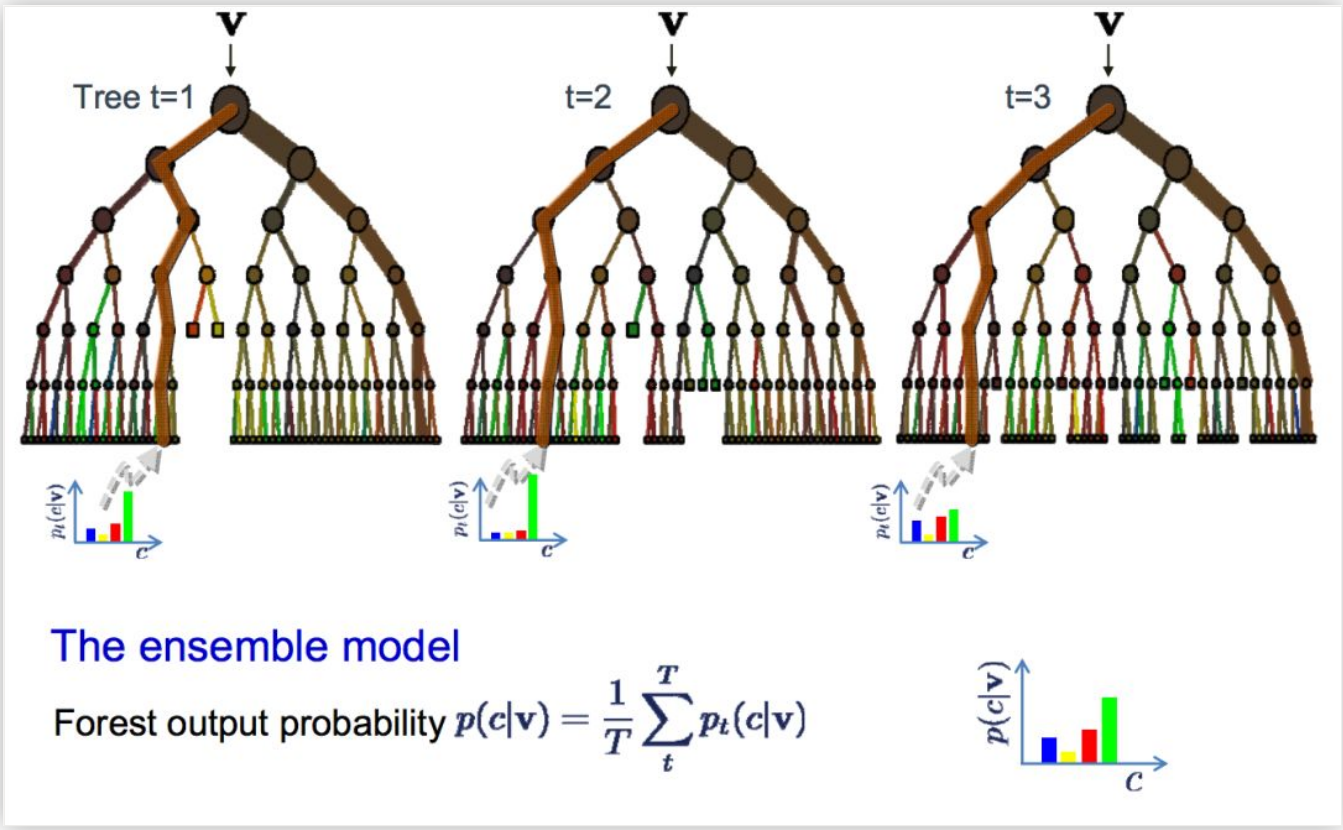
SVM可能是最强大的即用分类器之一，在你的数据集上值得一试。

09 bagging和随机森林

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

bootstrap是一种强大的统计方法，用于从数据样本中估计某一数量，例如平均值。 它会抽取大量样本数据，计算平均值，然后平均所有平均值，以便更准确地估算真实平均值。

在bagging中用到了相同的方法，但最常用到的是决策树，而不是估计整个统计模型。它会训练数据进行多重抽样，然后为每个数据样本构建模型。当你需要对新数据进行预测时，每个模型都会进行预测，并对预测结果进行平均，以更好地估计真实的输出值。



Random Forest

随机森林是对决策树的一种调整，相对于选择最佳分割点，随机森林通过引入随机性来实现次优分割。

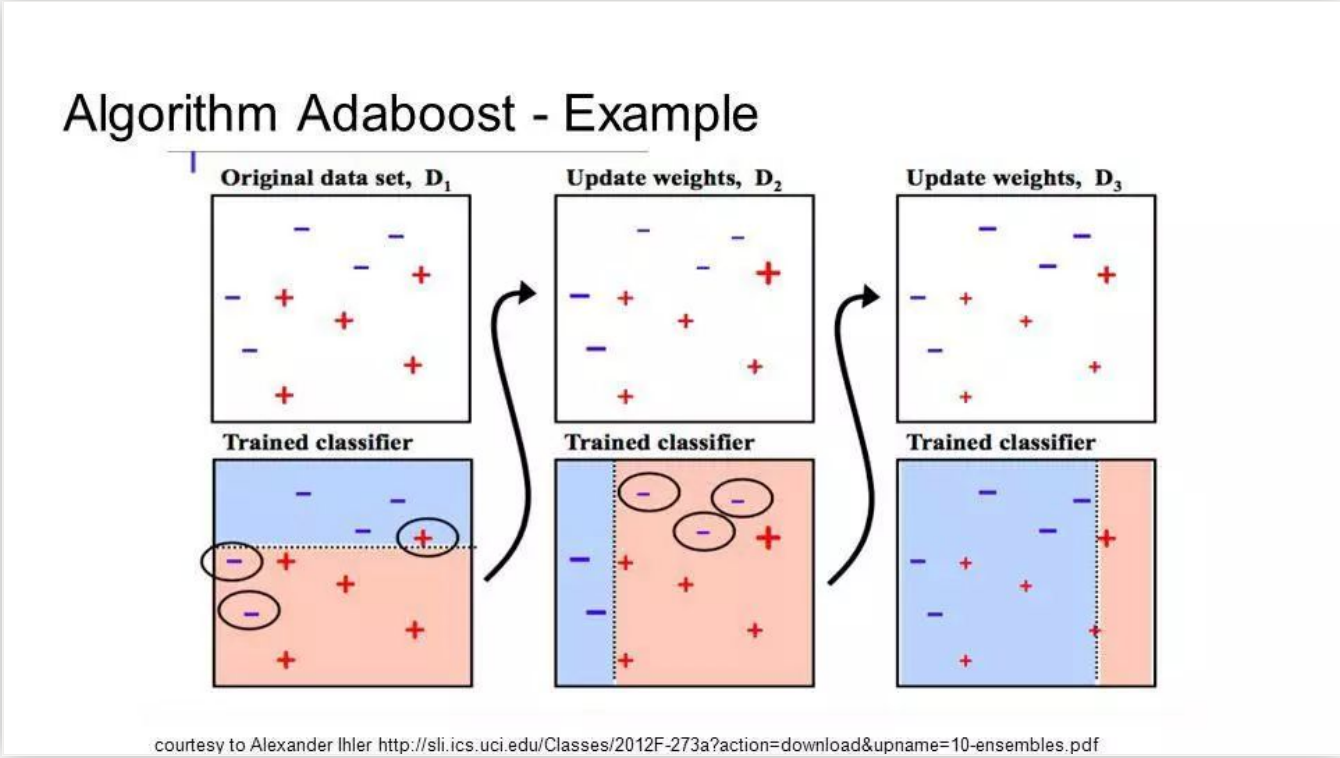
因此，为每个数据样本创建的模型之间的差异性会更大，但就自身意义来说依然准确无误。结合预测结果可以更好地估计正确的潜在输出值。

如果你使用高方差算法（如决策树）获得良好结果，那么加上这个算法后效果会更好。

10 Boosting和AdaBoost

Boosting是一种从一些弱分类器中创建一个强分类器的集成技术。 它先由训练数据构建一个模型，然后创建第二个模型来尝试纠正第一个模型的错误。 不断添加模型，直到训练集完美预测或已经添加到数量上限。

AdaBoost是为二分类开发的第一个真正成功的Boosting算法，同时也是理解Boosting的最佳起点。 目前基于AdaBoost而构建的算法中最著名的就是随机梯度boosting。



AdaBoost

AdaBoost常与短决策树一起使用。 在创建第一棵树之后，每个训练实例在树上的性能都决定了下一棵树需要在这个训练实例上投入多少关注。难以预测的训练数据会被赋予更多的权重，而易于预测的实例被赋予更少的权重。 模型按顺序依次创建，每个模型的更新都会影响序列中下一棵树的学习效果。在建完所有树之后，算法对新数据进行预测，并且通过训练数据的准确程度来加权每棵树的性能。

因为算法极为注重错误纠正，所以一个没有异常值的整洁数据十分重要。

初学者在面对各种各样的机器学习算法时提出的一个典型问题是“我应该使用哪种算法？”问题的答案取决于许多因素，其中包括：

- **数据的大小，质量和性质;**
- **可用的计算时间;**
- **任务的紧迫性;**
- **你想要对数据做什么。**

即使是一位經驗豐富的數據科學家，在嘗試不同的算法之前，也無法知道哪種算法會表現最好。雖然還有很多其他的機器學習算法，但這些算法是最受歡迎的算法。如果你是機器學習的新手，這是一個很好的學習起點。