机器学习新手必学十大算法指南

摘要： 本文为机器学习新手介绍了十种必备算法：线性回归、逻辑回归、线性判别分析、分类和回归树、朴素贝叶斯、K-近邻算法、学习向量量化、支持向量机、Bagging和随机森林、Boosting和AdaBoost。

在机器学习中有一种"无免费午餐（NFL）"的定理。简而言之，它指出没有任何一个算法可以适用于每个问题，尤其是与监督学习相关的。

因此，你应该尝试多种不同的算法来解决问题，同时还要使用"测试集"对不同算法进行评估，并选出最优者。

大原则

然而，这些都有一个共同的原则，那就是所有监督机器学习算法都是预测建模的基础。

机器学习算法包括目标函数(f)，输入映射变量(X)，生成输出变量(y)：Y=f(X)。这是一个通用的学习任务，希望在给出新案例的输入变量（X）能预测出（Y）。

最常见的机器学习方式是Y= f(X)的映射来预测新的X。这被称为预测建模或预测分析。

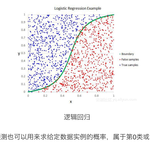
对于渴望了解机器学习基础的机器学习新手来说这非常难，那么下面来为大家介绍数据科学家最常使用的10种机器学习算法。

1 — 线性回归

线性回归是统计学和机器学习中最著名和最容易理解的算法之一。

预测建模主要关注如何最小化模型的错误或如何做出最准确的预测，而相应的代价是解释能力的欠缺。我们将从许多不同的领域借用、重用甚至启蒙算法和统计数据，来实现这个目标。

线性回归是一个方程，通过找到拥有特定权重的被称为字母系数(b)的输入变量，来描绘出最适合输入变量X和输出变量Y关系的一条线。



线性回归

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。

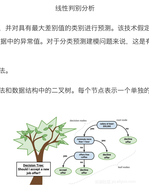
线性回归已经存在了200多年，使用这种技术的一些经验是：尽可能的去除相似变量，并从数据中去除噪声。

2 — 逻辑回归

逻辑回归是机器学习从统计学领域借鉴的另一种技术。它是解决二进制分类问题的首选方法。

逻辑回归与线性回归的相似点在于，两者目标都是找出每个输入变量加权的系数值。不同于线性回归的是，输出的预测需要用非线性函数的逻辑函数进行变换的。

逻辑函数看起来像一个大S，并将任何值转换成0到1的范围内。这是有用的，因为我们可以将逻辑函数的输出视范成0和1（例如，如果小于0.5，则输出1），并预测类别值。



逻辑回归

由于模型的学习方式，逻辑回归的预测也可以用来求给定数据实例的概率，属于第0类或第1类。当需要对预测结果做出合理解释时这非常有用。

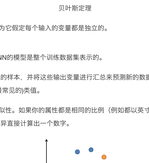
与线性回归一样，当你删除与输出变量无关的属性和非常相似(相关)的属性时，逻辑回归的效果会更好。对于二元分类问题，这是一个快速且有效的模型。

3 — 线性判别分析

逻辑回归算法是传统的分类算法，如果你有两个以上的类，那么线性判别分析算法是首选的线性分类技术。LDA的表达式非常直接，由数据统计值组成，为每个类别分别计算。对单个输入变量来说包括：

1)各个类的平均值。

2)所有类的计算方差。



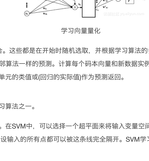
线性判别分析

线性判别分析计算每个类别的差别值，并对具有最大差别值的类别进行预测。该技术假定数据的分布遵循高斯分布（钟形曲线），因此在开始分析之前，需要移除数据中的异常值。对于分类预测建模问题来说，这是有效的方法。

4 — 分类和回归树

决策树是机器学习预测建模的重要算法。

决策树模型为二叉树形式，就是像算法和数据结构中的二叉树。每个节点表示一个单独的输入变量（x）和该变量上分裂点（假设变量为数值）。



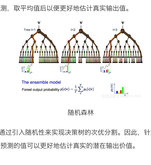
决策树

树的叶节点包含用于预测的输出变量（y）。预测从树干开始遍历，直到到达叶节点并输出叶节点的值。树的学习速度和预测速度都非常快。并且不需要对数据进行任何的准备。

5 — 朴素贝叶斯

朴素贝叶斯是一种简单但功能强大的预测建模算法。

该模型由两种类型的概率组成。这两种概率可以从训练数据中直接计算：1）每个类别的概率，2）给定每个x值的每个类别的条件概率。计算出来后，就可以用贝叶斯定理对新数据进行预测。当你的数据是实值时，通常采用高斯分布（钟形曲线），这样你就可以很容易的估计出这些概率。



贝叶斯定理

朴素贝叶斯之所以被称为朴素，是因为它假定每个输入的变量都是独立的。

6 — K-近邻算法

KNN算法是非常简单有效的，因为KNN的模型是整个训练数据集表示的。

通过在整个数据集中搜索K个最相似的样本，并将这些输出变量进行汇总来预测新的数据点。对于回归问题，这可能是平均输出变量，对于分类问题，这可能是模式(或最常见的)类值。

关键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（例如都以英寸为单位），最简单的就是使用Euclidean距离，你可以根据每个输入变量之间的差异直接计算出一个数字。

K-近邻算法

KNN需要大量的内存或空间来存储所有的数据，但是只有在需要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。

当有大量的输入变量时距离或紧密性可能会弄混，导致算法性能下降。所以建议只是用那些与预测变量最相关的输入变量。

7 — 学习向量量化

K-近邻算法的缺点之一是你需要利用整个数据集进行训练，而学习向量量化算法（LVQ）是一种人工神经网络算法，可以让你选择训练实例的个数，并精确地学习这些实例应该是什么样的。

学习向量量化

LVQ的表示是一个码本向量的集合。这些都是在开始时随机选取，并根据学习算法的多次迭代对数据集进行总结。在学习之后，这些向量表可以用来做类似K-近邻算法一样的预测。计算每个码本向量和新数据实例之间的距离。找到最相似的邻居（最佳匹配的码本向量）。然后将最佳匹配单元的类值或回归的实际值作为预测返回。

8 — 支持向量机

支持向量机可能是最受欢迎的机器学习算法之一。

超平面是一个分割输入变量空间的线。在SVM中，可以选择一个超平面来将输入变量空间中的点与它们的类（0类或1类）分开。在二维中，你可以将其想象成一条线，假设输入的所有点都可以被这条线完全隔开。SVM学习算法可以找到能够被超平面完美分割类别的系数。

支持向量机

超平面和最近数据点之间的距离被称为间隔。能够区分这两个类的最好或最优的超平面是有最大间隔的直线。这些与定义超平面和分类器构造有关的点成为支持向量。在实践中，可以使用优化算法来找到最大化间隔的系数的值。

SVM可能是最强大的分类器之一，值得在你的数据集上尝试。

9 — Bagging和随机森林

随机森林是最受欢迎和最强大的机器学习算法之一。它是一种集成机器学习算法，称为Bootstrap聚合或bagging。

Bootstrap是一种强大的统计方法，用于从数据样本中估计数量。就像一种平均值。你需要从数据中抽取大量样本，计算平均值，然后将计算所有平均值的平均值，以更好地估计真实平均值。

在Bagging中，可以用上述相同的方法估计整个数据模型。最常见的是决策树。选取训练数据中的多个样本，然后构建模型。当你需要预测新数据时，每个模型都会做出预测，取平均值以便更好地估计真实输出值。

随机森林

随机森林是对这种方法的一种调整。通过引入随机性来实现决策树的次优分割。因此，针对每个数据样本创建的模型与其它方式相比会有所不同，但仍然非常精确。结合预测的值可以更好地估计真实的潜在输出价值。

10 — Boosting和AdaBoost

boost是一种集成技术。它试图从许多弱分类器中创建一个强大的分类器。根据训练数据构建模型，然后创建第二个模型来纠正第一个模型中的错误。直到可以完美的预测模型，或者达到了模型最大量。

AdaBoost是第一个真正成功的用于二进制分类的增强算法。现代的增强方法都建立在AdaBoost上，最显著的是随机梯度增强机。

演算法

AdaBoost是用于短决策树的。在创建第一个树后，通过每个训练实例上的树的性能来衡量下一个创建的树对每个训练实例分配的权重。难以预测的训练数据将获得更多的权重。

依次创建模型后，每个模型都要更新训练实例上的权重，以确保序列中下一模树执行的学习。在所有的树都建立之后，对新的数据进行预测，每棵树的性能都取决于它对训练数据的准确性。

因为大量的注意力都放在了纠正算法的错误上，所以要有干净的数据和离群值。