如何快速定位合适的机器学习算法

我们可以从**问题本身、标签的类型、数据集的大小、特征的数量、计算时间的要求、以及模型的可解释性**这些方面，来比较和选择算法。

从问题的类型开始

要选择合适的算法，我们首先还是要回到问题本身。把问题定义好，并明确要解决的目标，是选择算法的前提。

如果目标是对有标签的数据进行预测或分类，那么就需要从监督学习算法家族中**进行选择；**如果目标是从没有标签的数据中挖掘出一些信息，或者是对数据做特征工程，比如降低数据的维度，那么就需要在**无监督学习**算法家族中进行选择。

如果目标是生成没有见过的新数据，例如生成完全不存在的人脸，或者让机器续写个哈利波特啥的，那么生成式机器学习算法值得研究一下。近年来非常吸引眼球的生成式对抗网络GAN（Generative Adversarial Networks）、Google推出的DeepDream算法，以及变分自编码器VAE等，都是解决这类问题的优秀算法。

现在，我们回到最为主流的监督学习问题，来继续算法的选择。在监督学习下面，我们又有两大类问题，就是回归问题和分类问题，我们可以根据标签是连续的数值，还是离散的数值，来确定问题的类型。这两大类问题，均拥有超多的应用场景，也各自拥有一大批的算法。我们首先看一下针对回归问题的算法选择指南。

回归问题算法选择指南

当明确了问题本身属于回归问题后，我们就需要在线性回归、决策树、随机森林、XGBoost、朴素贝叶斯以及神经网络这些常见的回归算法中进行选择。这时候，我们就需要考虑数据集大小、特征的维度（也就是特征的多少）、训练所需的时间等因素。

当回归问题并不复杂，并且希望从易于解释的模型开始时，建议选择线性回归模型作为可靠的首选基准算法，**线性回归对大、小数据集都适用，而且也可以处理高维特征数据。**

如果数据集特征和数据样本数量都不是很多，而且特征之间有关联关系，那么SVM**（支持向量机）**可以作为一个选择，这个模型对于特征数量有限、意义接近的中等大小的数据集来说比较强大。在使用SVM之前，需要进行特征缩放。

如果手头上的数据集样本数量和特征数量都十分巨大，那么可以考虑朴素贝叶斯模型，它比线性模型的速度要快得多，适用于非常大的数据集和高维数据，不过，它的性能通常要低于线性模型。

对模型运行速度有要求的时候，比如模型是在线训练、在线预测，那么决策树可以作为一个选择。因为它解决回归问题的速度很快，而且也不需要数据缩放。决策树的另一个优点是可以进行可视化，而且非常容易解释。

不过如果参数设置不当，决策树就会出现过拟合的现象；如果限制了树的深度的话，又比较容易出现精度不够的问题。因此，决策树通常是作为集成学习方法的基模型而存在的，很少独立使用。

如果追求的是模型的性能和表现，那么，随机森林几乎总是比单棵决策树的表现要好，它的性能非常强大，也不需要数据缩放。**但随机森林并不合适处理高维稀疏数据集。**

当然，XGBoost**的性能通常比随机森林还略高一点。**与随机森林相比，XGBoost**的训练速度虽然更慢，但预测速度更快**，需要的内存也更少。不过，在参数方面，XGBoost可调的外部参数比随机森林更多，所以调参时候会更繁琐一些。

另外，如果问题是**时间序列问题，或者特征数量极为巨大，又没有良好的结构，而且特征之间可能存在非线性依赖关系的时候，应该考虑使用深度神经网络。**因为深度神经网络可以构建非常复杂的模型来解决这类问题，特别是对于大型数据集而言，很有优势。

但是，**深度神经网络难于解释**，如果需要优化过程的具体推导细节，那深度学习模型很难给出。此外，它对数据缩放和参数选取也比较敏感，而且，大型神经网络需要很长的训练时间。这些都是需要根据实际情况考量的因素。

总之，面对新数据集，我建议通常最好先从简单模型开始，比如线性模型、朴素贝叶斯或最近邻KNN，看能得到什么样的结果。对数据有了进一步了解之后，可以考虑构建更复杂模型的算法，比如随机森林、梯度提升决策树、SVM或神经网络。

分类问题算法选择指南

分类问题的算法选择，在思路上和回归问题略有不同。对于分类问题，我更多是基于问题的性质来寻找合适的算法，而不是从数据集大小、特征数目的多少以及训练时间的快慢来考量。

当我们要解决用户是否会流失、欺诈检测、促销和裂变效果等问题时，可以采用逻辑回归算法作为这些普通分类问题的基准算法，因为它简单且易于解释。当然，其它分类算法也可以用于解决这类问题。

而对于投资方面的决策、银行贷款的评估决策等，我们经常使用决策树算法。因为决策树算法最直观，最接近人类的思维方式，算法也非常易于用图表展示。当然，决策树在分类问题中也容易出现过拟合问题，因此，决策树很少单独作为一种算法来解决问题，一般都是作为集成学习方法的基模型出现。

对于类别比例不平衡的问题，比如检测生产系统中的坏件，预测高风险的病患等，推荐采用随机森林算法，它解决了决策树算法的过拟合问题。同时，XGBoost算法和随机森林类似，也是解决各种分类问题的好选择，性能比随机森林略高。

如果要解决文本的情感属性判断、文本分类、推荐系统或者是人脸识别问题，那么朴素贝叶斯算法是深度学习出现之前的首选算法。和深度学习相比，朴素贝叶斯算法的优势在于可解释性较好。因此，在要求算法具有良好的解释性时，可以通过朴素贝叶斯算法解决这类问题。

而对于疾病诊断和判断、手写数字识别等多种分类问题，SVM是一种比较好的解决方案。当然深度学习也可以解决这类问题，和深度学习相比，SVM的优势也在于可解释性较好。因此，在要求算法具有良好的解释性时，我们用SVM算法来解决多种分类问题比较好。

最后，我不得不提的是，深度学习神经网络实为解决大数据时代的分类问题的大杀器，我们前面所说的各种分类问题，都可以用深度学习来解决。尤其是非结构化的分类问题，比如语音识别、计算机视觉、自然语言处理、文本分类、自动驾驶等，深度学习都是首选项。

当然，非要“鸡蛋里挑骨头”的话，那就是在数据量小或者特征结构很好的情况下，深度学习的优势体现得不是很明显，我们还是要尽量选择简单的方案；另外，深度学习的可解释性不好，就像一个黑箱，因此，调参的过程也比较费力。

好了，这就是分类问题算法选择的一些原则。下面，再总结一下选择算法时的考量因素。

选择算法时的其它考量因素

我们前面说，选择算法时有两个最主要的出发点，第一是问题的类型，我们要回到问题本身，看它是监督学习还是无监督学习；第二是预测的标签类型，一般来说，监督学习问题最为主流，这时候我们需要根据标签来进一步确定，问题属于回归问题还是分类问题。

除此之外，我们在选择算法时还有一些其他的考虑因素，我们一起看看。

训练数据的大小

通常，我们都希望能收集到大量数据，获得更可靠的预测。但很多时候，数据的量是一个制约因素，我们也没办法。因此，如果训练数据集较小，也就是数据样本较少，但是特征数较多（如文本数据），这时应该选择具有高偏差、低方差的算法，比如如线性回归、朴素贝叶斯或线性SVM。

如果训练数据足够大，并且特征数量也比较少，那就可以使用低偏差、高方差的算法，如 KNN、决策树或核SVM。当然，对于数据很多的情况，神经网络也是很好的选择。

特征的数量

除数据样本量之外，特征的数量也影响模型的选择。数据集可能有大量的特征，有些特征可能并不是和问题十分相关，因此也并不重要。某种数据集，例如遗传学问题或文本数据，与数据样本的数量相比，特征的数量可能非常大。

过多的特征会使多数算法的训练时间过长，不过，SVM模型就不会受特征数量的约束，因此，对于大特征空间和较少数据量的情况，SVM模型是我们的最佳选择。当然，如果要用其他的机器学习模型，可以使用PCA和特征选择技术来降低特征的维度，选择重要特征进行学习。

除此之外，深度学习也为我们处理巨大特征量的问题提供了非常好的解决方案。当特征数量巨大且特征空间结构复杂时，深度学习神经网络是绝佳选择。

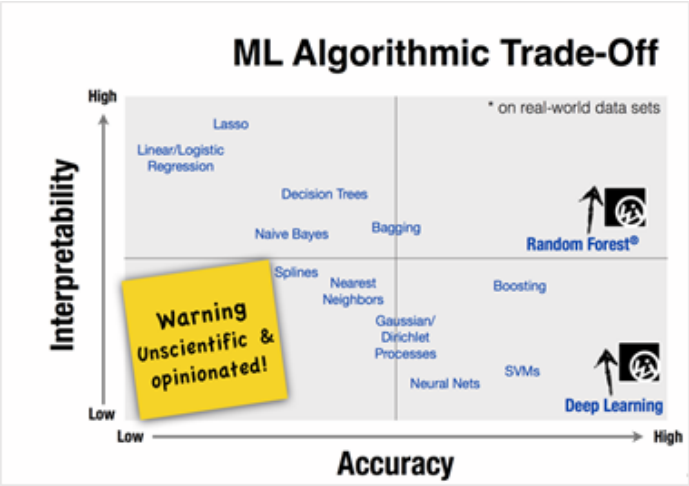
性能和可解释性的权衡

有时我们还必须在模型的性能和可解释性之间进行权衡，比如有些合规性文件会要求公司在应用AI时，在给出判断结论的同时给出算法推导的原理和过程，这时候，我们采用相对简单的模型就能降低此部分合规性文档的解释难度。

高度可解释的算法（如线性回归模型）意味着我们可以轻松理解任何单个预测变量如何与要预测的目标相互关联，因为模型简单，所以很容易解释。但是，线性回归产生的映射函数范围小，其形状只能覆盖很小的特征空间，因此这类算法也被称为限制性算法。

也有些算法非常灵活，因为它们可以生成更广泛的映射函数形状。比如深层神经网络模型，它可以拟合任意形状的函数，但灵活的模型是以低可解释性为代价，来提供更高的准确性的。一般来说，随着算法灵活性和准确性的增加，其可解释性会降低。

下图显示了各种算法在准确性和可解释性之间的权衡。



至于我们要使用哪种算法，就要取决于业务问题的目标了。如果目标是进行严密的推理，比如疾病的确诊过程，那么限制性模型会更好，因为它们更具可解释性。如果以更高的精度为目标，那么灵活的模型会更好。

速度或训练时间

更高的准确度通常也意味着更长的训练时间。此外，更大规模的数据集样本也需要更多时间来训练。在实际应用中，这个因素也会驱动算法的选择。

朴素贝叶斯、线性回归和逻辑回归这样的算法易于实现且运行迅速。像 SVM 这样的算法涉及参数调整，需要更多时间，而神经网络和随机森林，也都需要大量时间来训练数据。因此，如果机器学习模型是在线的应用，比如在网页上，用户选择了一个商品，需要在一两秒钟之内呈现出推荐列表的清单，这时候就不应该选择最精准但会耗时很长的算法了，而应该考虑朴素贝叶斯等更快的算法。

数据的线性程度

最后，我们还可以通过评估数据特征和标签之间的线性程度，来选择具体的算法。

许多算法都假设不同类别可以由直线（或更高维的平面或者空间）来分隔，比如逻辑回归和支持向量机。线性回归算法则假设特征和标签数据的分布趋势遵循着一条直线。如果数据是线性的，那么这些算法的表现相当不错。

然而，数据并不总是线性的，因此对于更为复杂的数据集，我们需要可以处理高维和复杂数据结构的算法，比如核 SVM、随机森林、XGBoost和神经网络。

那么，如何找出监督学习数据集的线性程度呢？对于回归问题，显示特征和标签之间的散点图，看看是否能够拟合出线性回归线；对于分类问题，我们先尝试逻辑回归或线性SVM 并检查误差值和分类性能。如果误差值很高、分类性能差，则意味着数据不是线性的，需要复杂的算法来拟合。

总结一下

好了，到这里我们这一讲就结束了。关于快速定位合适的机器学习算法，这其中的要点是定义好问题，明确是监督学习问题还是无监督学习问题，是分类问题还是回归问题，这能让我们排除掉不相干的算法。

在选择具体的算法时，建议从训练数据的大小、特征的数量、是着重考量模型的性能还是考量模型的可解释性、是否要求模型有很快的训练速度，以及数据的线性程度这几个方面，来选择最适宜的算法。

另外，建议从探索数据开始，熟悉数据。因为“更好的数据往往胜过更好的算法”，对数据集的特征之间的性质了解得越清楚，越有可能得到高性能的模型。当然，对模型背后的原理和假设越了解，以及模型外部参数设定的含义越熟悉，也越有可能得到高性能的模型。

在定位合适的机器学习算法时，还有一个最基本的原则是，从简单的模型开始构建基准模型，然后尝试更复杂的方法。