<https://arxiv.org/> 论文网站

做算法：算法工程师，要看最新的论文，看看可行性

做工程：”调参工程师”，读旧论文，用已知有效的方法解决问题

（链，树，图的构建、遍历、查找、排序）

【1】特征选取：从业务角度区分输入数据包含的特征，并认识到这些特征对结果的贡献。

a) 对数据本身和其对应的业务领域有所了解。

b) 能够根据需要标注数据。

c) 知道如何从全集中通过划分特征子集、加减特征等方法选取有效特征集。

【2】向量空间模型（VSM）构建：了解如何将自然语言、图片等人类日常使用的信息转化成算法可以运算的数据。

a) 能够把文字、语音、图像等输入转化成算法所需输入格式（一般为实数空间的矩阵或向量）。

b) 能够根据信息熵等指标选取有效特征。

【3】数据清洗和处理：对直接的业务数据进行筛选并转换为模型可处理形式。

a) 能够运用统计学方法等ETL手段清洗输入数据。

b) 能够对数据进行归一化（normalization）， 正则化（regularization）等标准化操作。

c) 能够采用bootstrap等采样方法处理有限的训练/测试数据，以达到更好的运算效果。

2.4.3 模型验证

算法+数据就能够得到模型。但是，

这个模型的质量如何？

这个模型和那个模型比较，哪个更适合解决当前问题？

在做了如此这般的优化之后得出了一个新的模型，怎么能够确认它比旧的模型好？

为了解答这些问题，就需要掌握度量模型质量的方法。为此，需要做到：

i） 了解 bias，overfitting 等基本概念，及针对这些情况的基本改进方法。

ii） 了解各种模型度量指标（e.g. Accuracy, Precision,Recall, F1Score……）的计算方法和含义，及其对模型质量的影响。

iii) 能够构建训练集、测试集，并进行交叉验证。

iv) 能够运用多种不同的验证方法（e.g. 2-Fold cross-validation,K-Fold cross-validation, Leave-One-Out cross-validation……）来适应不同的数据集。

现阶段而言数据远比算法重要，对模型的影响远胜算法。有直接作用的都是有监督学习，深度学习也是依赖海量标注好的数据。

目前还没有完全自动化标注技术出现，人工标注仍然是必要和主流。

3.3.1 什么叫做标注

举个很简单的例子说明一下什么是数据标注：

在开发聊天机器人的时候，我们需要训练意图判定和实体识别模型，因此也就需要标注用户问题的意图和出现的实体。

这是用户问题原始数据：“00183号商品快递到伊犁邮费多少？”

这样一句话，很显然问它的用户是想知道某一种商品发往某地的邮费。邮费是商品的一个属性，我们把所有查询商品属性的意图都定义为“商品查询”。

因此，这样一句话的意图是“商品查询”。其中有包含了几个实体，分别是商品Id，目的地和商品属性。

这句话被标注出来以后，就是下面这个样子：

[00183]<-{商品Id}号商品快递到[伊犁]<-{目的地}[邮费]<-{商品属性}多少？||商品查询

具体格式不必纠结。核心一点：标注就是将原始数据内全部或者部分内容，按照业务需求打上定义好的标签。------没有专业要求，属于脏活累活，谁都能干。

在工程学中，有多种构建键-值存储系统的方式，每个设计都会构造一组不同的关于使用模式的假设集合。在统计建模中，有很多分类器构建算法，每个算法构造一组不同的关于数据的假设集合。

当处理少量数据时，尝试尽可能多的算法，然后挑选最好的一个的做法是比较合理的，因为此时实验成本很低。**但当遇到“大数据”时，提前分析数据，然后设计相应“管道”模型（预处理，建模，优化算法，评价，产品化）是值得的。**

正如我之前文章中所指出的，有很多种方法来解决一个给定建模问题。每个模型做出不同假设，如何导引和确定哪些假设合理的方法并不明确。在业界，大多数实践者是挑选他们更熟悉而不是最合适的建模算法。在本文中，我想分享一些常见错误（不能做的），并留一些最佳实践方法（应该做的）在未来一篇文章中介绍。

1. 想当然地使用缺省损失函数

许多实践者使用缺省损失函数(如，均方误差)训练和挑选最好的模型。实际上，现有损失函数很少符合业务目标。以欺诈检测为例，当试图检测欺诈性交易时，业务目标是最小化欺诈损失。现有二元分类器损失函数为误报率和漏报率分配相等权重，为了符合业务目标，损失函数惩罚漏报不仅要多于惩罚误报，而且要与金额数量成比例地惩罚每个漏报数据。此外，欺诈检测数据集通常含有高度不平衡的标签。

在这些情况下，偏置损失函数能够支持罕见情况（如，通过上、下采样）。

2．非线性情况下使用简单线性模型

当构建一个二元分类器时，很多实践者会立即跳转到逻辑回归，因为它很简单。但是，很多人也忘记了逻辑回归是一种线性模型，预测变量间的非线性交互需要手动编码。回到欺诈检测问题，要获得好的模型性能，像“billing address = shipping address and transaction amount < $50”这种高阶交互特征是必须的。因此，每个人都应该选择适合高阶交互特征的带核SVM或基于树的分类器。

3．忘记异常值

异常值非常有趣，根据上下文环境，你可以特殊关注或者完全忽略它们。以收入预测为例，如果观察到不同寻常的峰值收入，给予它们额外关注并找出其原因可能是个好主意。但是如果异常是由于机械误差，测量误差或任何其它不可归纳的原因造成的，那么在将数据输入到建模算法之前忽略掉这些异常值是个不错的选择。

相比于其它模型，有些模型对异常值更为敏感。比如，当决策树算法简单地将每个异常值计为一次误分类时，AdaBoost算法会将那些异常值视为“硬”实例，并为异常值分配极大权值。如果一个数据集含有相当数量的异常值，那么，使用一种具有异常值鲁棒性的建模算法或直接过滤掉异常值是非常重要的。

4．样本数少于特征数（n<<p）时使用高方差模型

SVM是现有建模算法中最受欢迎算法之一，它最强大的特性之一是，用不同核函数去拟合模型的能力。SVM核函数可被看作是一种自动结合现有特征，从而形成一个高维特征空间的方式。由于获得这一强大特性不需任何代价，所以大多数实践者会在训练SVM模型时默认使用核函数。然而，当数据样本数远远少于特征数（n<<p）—业界常见情况如医学数据—时,高维特征空间意味着更高的数据过拟合风险。事实上，当样本数远小于特征数时，应该彻底避免使用高方差模型。

5．尚未标准化就进行L1/L2/等正则化

使用L1或L2去惩罚大系数是一种正则化线性或逻辑回归模型的常见方式。然而，很多实践者并没有意识到进行正则化之前标准化特征的重要性。

回到欺诈检测问题，设想一个具有交易金额特征的线性回归模型。不进行正则化，如果交易金额的单位为美元，拟合系数将是以美分为单位时的100倍左右。进行正则化，由于L1/L2更大程度上惩罚较大系数，如果单位为美元，那么交易金额将受到更多惩罚。因此，正则化是有偏的，并且趋向于在更小尺度上惩罚特征。

为了缓解这个问题，标准化所有特征并将它们置于平等地位，作为一个预处理步骤。

6． 不考虑线性相关直接使用线性模型

设想建立一个具有两变量X1和X2的线性模型，假设真实模型是Y=X1+X2。理想地，如果观测数据含有少量噪声，线性回归解决方案将会恢复真实模型。然而，如果X1和X2线性相关（大多数优化算法所关心的），Y=2\*X1, Y=3\*X1-X2或Y=100\*X1-99\*X2都一样好，这一问题可能并无不妥，因为它是无偏估计。然而，它却会使问题变得病态，使系数权重变得无法解释。

7. 将线性或逻辑回归模型的系数绝对值解释为特征重要性

因为很多现有线性回归量为每个系数返回P值，对于线性模型，许多实践者认为，系数绝对值越大，其对应特征越重要。事实很少如此，因为：

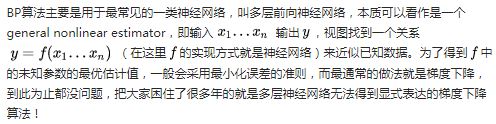
(a)改变变量尺度就会改变系数绝对值；

(b)如果特征是线性相关的，则系数可以从一个特征转移到另一个特征。此外，数据集特征越多，特征间越可能线性相关，用系数解释特征重要性就越不可靠。

这下你就知道了机器学习实践中的七种常见错误。这份清单并不详尽，它只不过是引发读者去考虑，建模假设可能并不适用于手头数据。为了获得最好的模型性能，

挑选做出最合适假设的建模算法—而不只是选择你最熟悉那个算法，是很重要的。

**BP传播算法：**



BP算法实际上是一种近似的最优解决方案，背后的原理仍然是梯度下降，但为了解决上述困难，其方案是将多层转变为一层接一层的优化：只优化一层的参数是可以得到显式梯度下降表达式的；而顺序呢必须反过来才能保证可工作——由输出层开始优化前一层的参数，然后优化再前一层……跑一遍下来，那所有的参数都优化过一次了。但是为什么说是近似最优呢，因为数学上除了很特殊的结构，step-by-step的优化结果并不等于整体优化的结果！不过，好歹现在能工作了，不是吗？至于怎么再改进（已经很多改进成果了），或者采用其他算法（例如智能优化算法等所谓的全局优化算法，就算是没有BP这个近似梯度下降也只是局部最优的优化算法）那就是新的研究课题了。

adaboost算法 http://blog.csdn.net/zxc024000/article/details/51577324

优点：

（1）精度很高的分类器。

（2）提供的是框架，可以使用各种方法构建弱分类器。

（3）简单，不需要做特征筛选。

（4）不用担心过度拟合。

实际应用：

（1）用于二分类或多分类。

（2）特征选择

（3）分类人物的baseline。

它的自适应在于：前一个弱分类器分错的样本的权值（样本对应的权值）会得到加强，

权值更新后的样本再次被用来训练下一个新的弱分类器。在每轮训练中，

用总体（样本总体）训练新的弱分类器，产生新的样本权值、该弱分类器的话语权，

一直迭代直到达到预定的错误率或达到指定的最大迭代次数。

**各模型／方法比较**

**无监督聚类、异常检测:** 结果不可控，缺乏解释性，但是可以发现新的异常。比较流行的是:GMM、isolationforest、统计方法等。不过可以设计一个可解释强的图模型，使用图分割算法会有一定的收获。

**有监督分类:** 在训练样本充足的情况下，有监督绝对为王。如果模型需要可解释性，LR是第一选择，不过需要做较多的特征工程才能达到一定的效果。黑箱模型的话，XGBoost、GBM、RF是比较流行的，还可以Stacking。

**强规则/策略:**可以快速上线，准确率高，但是对问题的召回率取决于规则制定者和问题难度。

**半监督:**

PU learning:虽然该类方法解决场景很match，但近两年好像没有新的研究，测的效果也一般。

图论的相关传播算法：比如标签传播、基于贝叶斯网络等算法。

人与机器共同学习:个人比较推崇的一种方法，机器学到的是人的经验。而人借助机器可以更有效率地提升对问题的认知，再把新的经验作用到机器上，如此循环，直至收敛。

XGBoost、LightGBM、GBDT

当正负样例大致相当时，可以用有监督的方法来做。

当正负样例数量悬殊时（如黑样例特别少），先用无监督学习过滤大量的白样例，常用方法有聚类算法、one class SVM，isolation forest等。当把明显的白样例过滤掉后，这个时候候选的黑样例和白样例就没那么夸张了，可以