偏差和方差

偏差：算法期望预测和真实预测之间的偏差程度。反应的是模型本身的拟合能力。

方差：度量了同等大小的训练集的变动导致学习性能的变化，刻画了数据扰动导致的影响。

机器学习为什么对数据进行归一化？

归一化的目的是处理不同规模和量纲的数据，使其缩放到相同的数据区间和范围，以减少规模、特征、分布差异对模型的影响

（1）归一化加速GD求解最优解的速度。比如收敛路径呈Z字型，导致收敛太慢；

（2）归一化可能提高精度。

机器学习什么情况下对数据进行归一化？

（1）使用了梯度下降算法，如LR、SVM等；

（2）计算样本点距离时，如KNN、K-Means等。

机器学习什么情况下不需要归一化？

概率模型（决策树）不需要归一化。

归一化方法：

（1）max-min法：容易受极端值的影响，一定程度上会破坏原有的数据结构；

（2）z-score法：会改变原有数据的分布，不适合对稀疏数据做处理，不适合根据变量差异程度的聚类分析；

（3）RobustScaler：适用于存在离群点的数据。

（4）上述方法分析：在分类中，聚类算法，数据符合正态分布中，需要使用距离来度量相似性或者使用PCA降维时，z-score表现得较好。在不涉及距离测量，协方差计算，数据不太符合正态分布时，可以使用第一种方法或其他方法。

LR作为线性模型，如何拟合非线性情况？

特征侧：离散化、交叉组合

模型：引入kernel，或推广到FM等model。

海量离散特征用简单模型、少量连续特征用复杂模型。

SVM优缺点：

1、解决小样本情况下的机器学习问题；

2、提高泛化能力；

3、处理高维空间数据；

4、解决非线性问题；

5、对于线性问题没有通用的解决方案，谨慎选择kernel函数；

6、处理分类问题时，要求解函数的二次规划问题，需要大量的存储空间。

Random Forest：

1、为什么随机选取数据集？

如果不随机的话，训练出来的多棵树的分类结果是一样的，违背了bagging思想

2、为什么有放回抽样？

RF在分类时是求同，有放回的抽样会产生相同的训练样本；如果不是有放回抽样，训练出来的每棵树的结果存在很大偏差，这样对分类结果没有任何帮助。所以有放回抽样能够减小bias，RF的目的是减小variance。

3、影响RF分类结果的因素：

（1）任意两棵树的相关性：相关性越大，错误率越高；

（2）每棵树的分类能力：分类器能力越强，错误率越低；

（3）唯一可调参数：RF中特征子集的数量，数量越大，性能越好。

4、优缺点

（1）RF引入两个随机性，抗噪能力增强；

（2）RF在分类中对各个变量的重要性进行估计，对泛化误差进行无偏估计；

（3）可处理高维数据，对数据集的适应能力强；

（4）性能由于单预测器，分类精度与boosting算法差不多，运行速度更快；

（5）训练数据较少或噪声数据较大时，会发生overfitting。

GBDT

优缺点：

（1）灵活处理各种离散值和缺失值；

（2）对异常值具有鲁棒性；

（3）相对SVM，预测的准确率较高；

（4）由于弱学习器之间存在依赖关系，难以并行训练；

（5）不适合处理高维稀疏数据，缺乏平滑处理。

1、算法流程：多轮迭代，每轮迭代产生一个弱分类器，每个分类器在上一轮分类器的残差基础上进行训练。对弱分类器的要求一般是低variance和高bias的简单分类器，如CART树。最终分类器是将每轮训练得到的弱分类器加权求和得到的。（采用不放回采样）

2、损失函数：一般损失函数的负梯度（一般损失函数存在优化困难问题）

3、选择特征：以CART树为例，原始GBDT遍历每个特征，然后对每个特征遍历它所有可能的切分点，找到最优特征J和最优切分点。即

min(J){min(c1)sum(y-c1)^2+min(c2)sum(y-c2)^2}

4、分裂节点的评价指标：分裂时选择使得误差下降最多的分裂。

5、构建特征：利用GBDT产生特征的组合(CTR预估中，工业届一般用LR处理非线性数据，其中就可以用这种特征组合的方式增强LR对非线性分布的拟合能力)。比如，利用GBDT生成两棵树，根据叶子结点的个数生成一个向量，然后将样本分别输入进去，样本分到叶子结点处该向量位置值为1，其余为0，根据得到的这个向量作为该样本的组合特征。

6、用于分类：GBDT无论用于分类还是回归都是用CART树。对于K分类问题，每次迭代构建K棵树，然后利用softmax求出属于每棵树的概率，残差是真实label与第k棵树的概率的差值。

7、GBDT减少误差的方式：每棵树拟合当前模型预测值和真实值之间的误差。

8、正则化：

学习率：越小越易overfitting

子采样比：越小越易underfitting

剪枝操作

9、参数

步长、学习率：减小步长，提高泛化能力；学习率过大容易overfitting；

树最大深度：太大overfitting

内部节点再划分所需最小样本数和叶子节点最少样本数：内部节点再划分所需最小样本数越大越防overfitting

子采样比：太小容易underfitting

相比于SVM、LR，GBDT的优势：

基于树模型，继承了树模型的优点（对异常点鲁棒，不相关特征干扰性低，能处理缺失值，受噪声干扰小）

GBDT和XGBoost的区别

1、XGBoost支持线性分类器，传统GBDT以CART作为基分类器，XGBoost支持线性分类器，相当于带L1和L2正则化的LR或线性回归。

2、XGBoost对损失函数进行二阶泰勒展开。XGBoost支持自定义代价函数，代价函数需一阶和二阶可导。

3、XGBoost在代价函数里加入正则项，用于控制模型的复杂度。

4、XGBoost支持列抽样，借鉴了RF的做法，能降低overfitting，减少计算。

5、XGBoost工具支持并行，并行是在特征上做的，GBDT每次节点分裂时，需要对所有特征计算并排序，而XGBoost预先对数据排序并保存为block结构，迭代过程中重复使用这个结构，大大减少计算量。在节点分裂时，特征增益的计算可以开多线程实现。在内存有限或分布式的情况下，数据无法一次性载入时，XGBoost用可并行的近似直方图算法能高效生成候选的分割点。

高维稀疏特征的时候，LR的效果为什么会比GBDT好？

因为LR等线性模型的正则项是对权重进行惩罚，而树模型的惩罚项通常为叶子结点数量和深度。因此带正则化的线性模型不容易overfitting。

为什么GBDT每棵树的深度很浅，而RF每棵树的深度很深？

SVM中假设分离超平面系数非常大怎么办？

LR归一化问题，什么情况下可以不归一化，什么情况下必须归一化，为什么？

提到LR损失函数要能知道交叉熵，为什么使用交叉熵？使用交叉熵为损失函数的优化问题是在优化什么量？交叉熵和KL散度、相对熵的关系？

SGD和BGD的区别？不同的GD方法有什么区别和联系？二阶优化算法有什么？对比off-line和on-line learning的区别？

调参，要明白不同超参数的含义，以及给定一个特定的情况，大概要调整哪些参数以及如何调整？

关于L1和L2正则，其几何解释和概率解释

LR的分布式实现逻辑是怎样的？数据并行和模型并行的区别？P-S架构是什么东西？

TF-IDF中IDF为什么用对数，涉及到信息论的知识，它是指IDF相当于一个权重，当该词越容易分辨，则相应权重也会特别大。

word2vec模型：输入层~映射层~输出层。映射层到输出层采用负采样和层序softmax。

训练方法：负采样、层序softmax

层序softmax：原来的softmax中是从字典中选一个单词，而层序softmax的思路是将一次分类分解为多次分类，因此采用树的结构，而为什么采用Huffman树呢？是因为这样构建的话（根据词频构造），出现频率越高的词所经过的路径越短，从而使得所有单词的平均路径长度最短。

Huffman树的节点含义：非叶子节点可以认为是神经网络中隐藏层参数（初始化长度为m的零向量），叶子节点为字典中的词向量（随机初始化），每个词具有唯一的编码。

因为层序softmax的映射层到输出层是Huffman树，在训练过程中，对于CBOW来说，先将n-1个背景词向量相加然后作为映射层的输入，然后根据Huffman树计算每个非叶节点向量和输入向量的乘积再用sigmoid计算概率，依次递归下去直至叶节点，因为每个词都对应有唯一的一个编码，所以训练过程中的损失函数就是根据预测词的编码（分类的思想）累乘每次分叉时的概率，使得该概率值最大。

skip-gram类似，不过因为由中心词预测n-1个背景词，所以需要循环n-1遍。

负采样法：背景词是出现在中心词时间窗口内的词。噪声词是和中心词不同时出现在该时间窗口的词，噪声词由噪声词分布采样得到。噪声词分布可以是字典中的所有词的词频概率组成（一般是单字概率的3/4次方）。

为什么有了层序softmax后还要负采样的方法？

虽然层序softmax可以提高模型的训练效率，但是如果训练样本中的中心词是一个生僻词，那么模型得到的Huffman编码就是一个很长的序列。

Glove模型不仅可以很好地表示词与词之间的类比关系，还能够知道每个词的全局统计信息。

Fasttest在word2vec基础上加入了n-gram的信息。

为什么有了word2vec，还要Glove、ELMO等？

word2vec学习到词之间的类比信息，适合局部间存在很强关联性的文本，但是缺乏词与词之间的共现信息。词义相近的词对贡献次数多，词义差得比较远的词对共现次数比较少，但是其实它们的区分度并不明显。

Glove：对word-pair共现进行建模，拟合不同word-pair的共现之间的差异。相比于word2vec，Glove更容易并行化，速度更快。

ELMO：前两者学到某个词的词向量是固定的，不能很好处理一词多义。ELMO是基于整个语料训练的，而不是窗口，因而能更好地表达一个词。

Bert：Bert中利用mask的方法有点像负采样，只不过Bert是基于整个语料做的，Bert主要是推翻了现有的对于某个任务必须特定的模型结构这一做法。

|  |  |  |  |
| --- | --- | --- | --- |
| 预测  实际 | 1 | 0 |  |
| 1 | TP | FN | TP+FN |
| 0 | FP | TN | FP+TN |
|  | TP+FP | FN+TN | TP+FN+FP+TN |

准确率：（TP+TN）/（TP+TN+FP+FN）

错误率：（FP+FN）/（TP+TN+FP+FN）

召回率(Recall)、查全率：实际的正样本中，分类成正样本的比例，（TP）/（TP+FN）

精确率(Precision)、查准率：分类成正样本中，实际的正样本的比例，（TP）/（TP+FP）

真正例率（TPR）：（TP）/（TP+FN），相当于Recall

假正例率（FAR、FPR）：（FP）/（FP+TN），也称误报率

拒真率（FRR）：实际的正样本中，分类成负样本的比例，（FN）/（TP+FN），相当于1-Recall

ROC曲线：不平衡数据集中最常用的指标之一



（1）阈值变大，样本分类成负样本的可能性增大（FP↓，FN↑）；

阈值变小，样本分类成正样本的可能性增大（TP、FP↑）；

（2）好的分类模型应该尽可能位于坐标轴的左上方；

随机猜测模型应该靠近于（0,0）和（1,1）这条对角线上；

（3）AUC面积值位于[0,1]，值越大说明模型分类越好；

（4）针对不平衡数据，应该如何选择阈值？

答：相等错误率（ERR），FAR随阈值的增大而减少，FRR随阈值的增大而增大。当FAR=FRR时，此时阈值最合适。