ML常见问题总结

判别模型/分类模型，建模的是P(Y|X)条件概率的分布。  // Y1=0.9， Y2=0.1

1、LR为什么是线性模型

LR是对数线性模型，因为对数几率 log(p/1-p) = wx。 p= sigmoid(wx)。

2、LR如何解决低维不可分

低维映射到高维，通过核函数可以实现，核函数解决了高维映射的大量内积操作。即在特征空间的内积操作等于他们在原始空间使用核函数计算的结果。

哪些核函数呢？ // 线性核，用于文本分类； 多项式核；高斯核；拉普拉斯核；sigmoid核

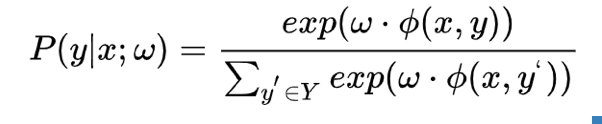
3、从图模型角度看LR

图模型: HMM/CRFs，跟LR相关的是CRF模型，LR是一个二分类模型，其扩展到多分类是softmax模型，softmax模型的序列化模型，可以扩展为CRFs，其实是linear-CRF。

LR和softmax是等价的，而且也可证最大熵和softmax也等价，即可证LR和最大熵的等价性。

LR(最大熵模型)统计的是训练集中的各种数据满足特征函数(其中，LR特征函数就是数据本身，跟贝叶斯一致)的频数(conditional)；贝叶斯模型统计的是训练集中的各种数据的频数；CRF统计的是训练集中相关数据 (比如说相邻的词，不相邻的词不统计，由马尔科夫性决定) 满足特征函数的频数。

个人感觉，最大熵模型是一个大集合，CRF本质上也是最大熵模型，只是特征函数为相邻节点的数据；LR/softmax也是最大熵模型，其特征函数是输入特征本身。



 // CRF，也是对数线性模型。  其特征函数是对相邻的输入/节点/词进行统计相关顺序。而LR/softmax则是对自身输入x进行统计。

所以三者都是对数线性模型。

4、LR的损失函数的推导

使用极大似然函数去估计模型的参数

p(Y=1 | x) = p(x);  p(Y=0 | x) = 1 - p(x)

似然函数-> 对数似然-> 构建损失函数(加上负号)

所以，在LR中损失函数(log损失函数)最小化跟最大化似然函数是等价的，前者可以由后者推导而来。

5、LR公式中为何用e？

跟1中的解释一致，因为log几率是线性。

6、L1-norm 与L2-norm

L1-norm假设参数服从laplace分布，容易导致稀疏解，可以用于特征选择。

L2-norm假设参数服从正态分布。可以防止过拟合，当然一定程度上L1也可以。 L2倾向于产生参数值较小的参数，这样对于特征的扰动容忍度加大，泛化能力强。

7、对比LR与线性模型

LR是在线性模型基础上加了非线性激活函数sigmoid，使得预测至域在0-1之间，适合分类任务。

8、LR与SVM的区别与联系

相同点

-都是分类模型、广义线性模型、判别模型、监督学习算法。

不同点

1. 损失函数：LR=log损失函数；SVM=hinge损失函数；
2. 分类原理不同： 分类原理的不同，LR基于概率理论，通过极大似然估计的方法估计出参数的值，而SVM基于几何间隔最大化原理，认为存在最大几何间隔的分类面为最优分类面
3. 样本对模型参数影响：LR是所有样本都有影响，而SVM只是少量样本有影响的（支持向量），在支持向量外添加样本点是没有影响的。
4. SVM的损失函数就自带正则，LR必须另外在损失函数上添加正则化。
5. 对于线性不可分的情况，SVM的核函数可以帮助将低维不可分的数据转换到高维，变成线性可分的；而LR用到核函数计算比较少，因为所有样本点都要参与计算。

关于LR和SVM的选择： //模型选型

如果Feature的数量很大/比较小(手动构建特征)，样本数量很大，这时候选用LR或者是Linear Kernel的SVM

如果Feature的数量比较小，样本数量一般，不算大也不算小，选用SVM+Gaussian Kernel

考虑了计算量以及分类原理以及样本对模型的影响。

9、随机误差项服从正态分布。随机误差项是一个期望值或平均值为0的随机变量。

10、连续特征离散化的好处

计算简单(**稀疏向量内积很快**)、**增强模型泛化性能**(年龄320上可以归为100以上)。

**11、为何选择LR？**

1）逻辑回归对线性关系的拟合效果好

2）逻辑回归计算快

3）LR返回的概率值可以用于权重表征

4）逻辑回归还有抗噪能力强的优点  //离散变量，即泛化性能强。

——————————————

**1、最大似然估计、最大后验估计、贝叶斯估计三者之间的关系**

首先，每个模型定义了一个假设空间，即SVM和LR的假设空间是不同的，选择一个模型，相当于选择了一个假设空间，模型的参数相当于假设空间中的解，有无数个解，对应有无数个参数。如何选择参数？  以下三种策略。

最大似然估计: 生成训练样本的最大可能性，p(D|alpha)

最大后验估计: 结合先验知识，估计模型参数，P(alpha|D) 后验。

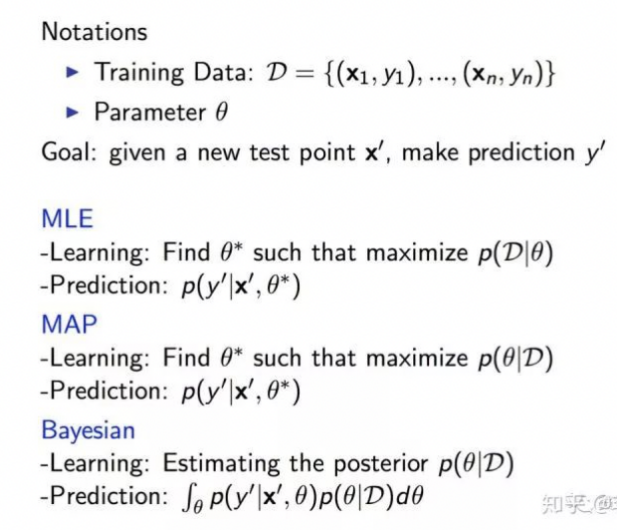
跟最大似然估计的关系是: 根据贝叶斯定理。  p(alpha | D) = p(D|alpha) \* p(alpha) / P(D)。 p(alpha)为先验知识。

当数据量较小的时候，先验知识能起一定作用，当数据量非常非常大时候，最大后验估计接近于**最大似然估计。**可以理解为此时先验知识可以从大数据中获取，此时已没多少参考价值。

//一定程度上解释了为何大数据，深度模型上用最大似然估计的比较多。

以上两者: 均是通过参数估计，选择最好的模型。

贝叶斯估计: 通过观测数据估计一个后验分布，基于后验分布做一个群体性决策，并不是选择最好的模型。学习的是后验分布P(alpha|D)的分布，然后对alpha进行积分(有限个就是加和，无限个就是积分)



1、什么是集成学习算法？

Ensemble算法，即训练多个分类器，将多个分类器的结果进行汇总(加权平均/投票法)去预测。

最简单的如voting, average

其他如badding, boosting, stacking

stacking的思路:

分层结构，第一步是训练m个基分类器(K折交叉，每个基分类器k次训练，预测整个train data的预测结果，这个是为了防止过分拟合，如果直接train整个数据集，然后预测train，则严重过拟合)，与真实label，拼成整个训练集。测试集则直接预测K次，求平均即可。//这个方法是最大程度上利用了原始数据。

第二步，则用第m + 1个基分类器去拟合新训练集和labels，然后再新测试集上预测，得到最终的测试集上的label。

stacking优点: 极大增强了diversity(模型 & 数据)，训练了m \* k个基分类器。第二层的分类器主要学习如何组合这些第一层的基分类器。

stacking支持异构的模型。

比如第一层的基分类器可以是KNN，朴素贝叶斯，随机森林等；元分类器可以是LR。

2、集成学习主要有哪几种框架？

两种: boosting (后一个分类器加大前一个分类器的错分样本的权重)和bagging（随机有放回的抽样N个样本）

3、常用的bagging算法有哪些？

随机森林

4、常用的boosting算法有哪些？

Adaboost, gbdt, xgboot

其中，Adaboost是指，每次放大错误样本的权重，最后迭代式训练m个基分类器进行加权和，本质是一个加法模型。

5、随机森林的算法的原理?

并行训练m个基分类器(决策树)，最后采取投票方式预测。行采样和列采样，行采样是指N个训练数据中有放回地抽取N个训练样本；列采样是指随机抽取m个特征m << M进行特征选择。

叶节点的判断: 选择分裂的属性跟父亲一样，或者没有属性可以选择，或者里面的所有样本的都是指向的同一个分类，则达到叶子。

没有剪枝，因为随机性保证了其不会过拟合。

6、随机森林的随机性体现在哪里？

体现在行采样和列采样上。

7、随机森林的优缺点

优点:

1. 很多数据集上，表现比较好，因随机性不容易过拟合& 抗噪声能力，也是其受欢迎的原因
2. 能处理高维度即特征很多的数据，且不用做特征选择。  //因为列采样特征随机选择
3. 对数据集的适应能力强：既能处理离散型数据，也能处理连续型数据，数据集无需规范化
4. 训练速度快，因其并行性 & 每次特征选择子集。
5. 模型泛化能力强，因其多样性的训练数据ensemble方式。
6. 因有很好的方法填充缺失值，即使有很大一部分数据缺失，也能保持很高的准确度。
7. 对于不平衡的数据集，也能平衡误差。

缺点:

- 随机森林已经被证明在某些噪音较大的分类或回归问题上会过拟合

- 级别划分较多的属性会对随机森林产生更大的影响，所以随机森林在这种数据上产生的属性权值是不可信的。

8、随机森林为何不能用全样本去训练m棵决策树

这样使得在训练的时候，每一棵树的输入样本都不是全部的样本，使得相对不容易出现over-fitting

9、随机森林和GBDT的区别？

10、简述下GBDT原理

梯度提升树， 属于boosting集成学习的一种，后一棵树学习前一棵树的负梯度值。因此使用的是CART的回归树

11、GBDT常用的损失函数有哪些?

分类：指数损失，log损失、交叉熵损失函数

回归: 平方损失，绝对值损失。

12、GBDT如何用于分类？

损失函数选择分类的损失函数。

13、为什么GBDT不适合使用高维稀疏特征？

在高维稀疏特征的时候，线性模型会比非线性模型好的原因：带正则化的线性模型比较不容易对稀疏特征过拟合。

而GBDT的**正则**是对节点数和深度的，只需要少量节点就可以划分高维稀疏特征，所以惩罚就很小, 导致容易过拟合。

14、GBDT算法的优缺点

优点: 效果好；灵活处理各类型数据。

缺点：难以并行学习。

15、简述XGBoost

XGBoost是GBDT的一个改进，性能有很好的提升，比如利用CPU进行多线性的并行计算。 并行性主要体现在特征选择的并行性。本身boosting算法是一个串行迭代的算法。他们之间也有明显的区别，主要体现在xxx。

1. XGBoost跟GBDT有什么不同？ 主要6点不同。
2. 传统GBDT以CART树为基学习器，XGBoost还支持线性分类器，这时候XGBoost相当于带L1和L2的正则化的LR分类器或者线性回归。
3. 传统的GBDT在优化时，只用到了一阶导数信息，而XGBoost对代价函数进行了二阶泰勒展开，同时用到了一阶、二阶导数。
4. XGBoost加入正则化，平衡方差和偏差，使得模型更简单，降低方差，防止过拟合。这也是由于GBDT的一个点。
5. 列采样。 XGBoost借鉴了RF的做法，支持列采样，不仅可以防止过拟合，还可以减少计算。
6. 对缺失值的处理。对特征值有缺失值的处理，XGboost可以自动学习到其分裂的方向。
7. XGBoost支持并行性。主要体现在特征选择上，预先将特征值进行排序，并存储在block结构中，后续迭代反复使用这个结构。在节点分裂时候，需要计算每个特征的信息增益，这个特征之间计算可以开多线程并行。
8. XGboost为何可以并行训练

- 预排序，Block， 特征选择的多线程并行。

1. XGboost防止过拟合的方法

- 加入正则项； 支持列采样。

19、XGboost为何这么快

- 特征选择并行性

- 列采样减少计算量。

20、K-means原理

首先选择K个初始中心，K为超参数，可以由开发集确定，然后求将每个点分配到中心中，重新计算中心的位置。

21、K-means对异常值是否敏感？为何？

敏感，导致计算的中心点偏离比较严重。

22、如何评估聚类效果

类间距，类内距

23、超参数K如何选择？

24、K-means算法的优缺点

25、SVM为何采用间隔最大化

当训练数据线性可分时，存在无穷个分离超平面可以将两类数据正确分开。

感知机利用误分类最小策略，求得分离超平面，不过此时的解有无穷多个。

线性可分支持向量机利用间隔最大化求得最优分离超平面，这时，解是唯一的。另一方面，此时的分隔超平面所产生的分类结果是最鲁棒的，对未知实例的泛化能力最强。

然后应该借此阐述，几何间隔，函数间隔，及从函数间隔—>求解最小化1/2 ||w||^2 时的w和b。即线性可分支持向量机学习算法—最大间隔法的由来。

26、SVM为什么引入核函数

- 解决线性不可分问题，低维映射到高维，内积计算量很大，通过核函数降低计算量，达到低维映射到高维的目的。即在特征空间的内积等于它们在原始样本空间中通过核函数K计算的结果。

27、SVM核函数之间的区别

- 线性核函数、多项式核函数、高斯核、拉普拉斯核、sigmoid核

38、为什么SVM对缺失数据敏感

这里说的缺失数据是指缺失某些特征数据，向量数据不完整。SVM没有处理缺失值的策略（决策树有）。而SVM希望样本在特征空间中线性可分，所以特征空间的好坏对SVM的性能很重要。缺失特征数据将影响训练结果的好坏。

39、SVM优缺点分别说三个

SVM原理

SVM是一种二类分类模型。它的基本模型是在特征空间中寻找间隔最大化的分离超平面的线性分类器。（间隔最大是它有别于感知机）

（1）当训练样本线性可分时，通过硬间隔最大化，学习一个线性分类器，即线性可分支持向量机；

（2）当训练数据近似线性可分时，引入松弛变量，通过软间隔最大化，学习一个线性分类器，即线性支持向量机；

（3）当训练数据线性不可分时，通过使用核技巧及软间隔最大化，学习非线性支持向量机。

注：以上各SVM的数学推导应该熟悉：硬间隔最大化（几何间隔）—学习的对偶问题—软间隔最大化（引入松弛变量）—非线性支持向量机（核技巧）。

缺点:

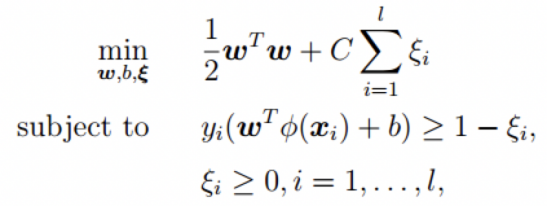
- 对核函数和参数敏感

- 对缺失数据敏感

- 计算量很大，训练时间非常慢。

优缺点

1. SVM的超参数C如何调节



这里的参数C代表的是在线性不可分的情况下，对分类错误的惩罚程度。

C值越大，分类器就越不愿意允许分类错误（“离群点”）。如果C值太大，分类器就会竭尽全力地在训练数据上少犯错误，而实际上这是不可能/没有意义的，于是就造成过拟合。

而C值过小时，分类器就会过于“不在乎”分类错误，于是分类性能就会较差。

默认值是1.0

41、SVM的核函数如何选择: 一般选择线性核和高斯核。前者参数少，比较快，后者参数多，效果好些。

同LR与SVM的选择

1. 简述决策树的构建过程

特征选择的过程，叶子结束条件。

树模型的优缺点

优点:

- 可解释性强

- 具有伸缩不变性(不用归一化特征)

- 有特征组合的作用

- 可自然处理缺失值

- 对异常点有鲁棒性

- 有特征选择作用

- 可扩展性强，容易并行

缺点:

- 缺乏平滑性 （回归预测，只能输出有限的若干种数值）

- 不适合处理高维稀疏数据。

1. ID3决策树与C4.5决策树的区别
2. CART回归树算法推导

45、从数学原理上解释下熵公式为何可以作为信息不确定的度量。

取值范围0-log|X|，需要多少字节编码

46、决策树如何防止过拟合，说说具体方法。

47、为什么要对特征进行归一化？

防止某些特征够大，对损失函数的计算影响过大，归一化到同一个尺度，方便计算。

如果不归一化：对于线性模型，特征值差别很大时，比如说LR，我有两个特征，一个是(0,1)的，一个是(0,10000)的，运用梯度下降的时候，损失等高线是椭圆形，需要进行多次迭代才能到达最优点。

但是如果进行了归一化，那么等高线就是圆形的，促使SGD往原点迭代，从而导致需要的迭代次数较少。

归一化与标准化的区别：后者统一均值为0，方差为1，前者则是统一范围就行。

48、什么是组合特征？如何处理高维度组合特征？

比如，长、宽，面积=长 \* 宽 即为组合特征。**矩阵降维，转为低维度向量。**

49、为什么一些场景使用余弦相似度而非欧式距离？

余弦的优点在于对向量长度不敏感。

50、one-hot的作用是什么？ 为什么不直接使用数字作为表示。

向量化。

51、降低过拟合和欠拟合的方法

52、L1和L2正则先验分别服从什么分布？

正则的作用：

- 从偏差-方差角度，减小方差，减少过拟合。

**- 从贝叶斯角度，相当于对参数引入了先验分布。**

L1服从拉普拉斯分布，对参数加入了分布约束，大部分取值为0；

L2服从高斯分布，对参数加入了分布约束，大部分取值较小。

53、对于树形结构为何不需要归一化？

因为节点属性取值的缩放对节点分裂选择没有影响，因此对树结构没有影响。 树节点分裂选择仅跟**特征值排序有关，排序不变，那么分裂点的选择不变。**

54、什么是数据不平衡，如何解决？

过采样、千采样、权重分配、ensemble、多任务的联合学习

55、LR相比于线性回归，有何异同点。

相同点: 广义上的线性模型；都是**优化最大似然函数；**

不同点，一个分类，一个回归；

56、LR的损失函数 done

57、LR处理多标签分类问题，一般怎么做。

M个分类器。

Cm2个分类器

Softmax

58、全概率公式和贝叶斯公式Done

59、朴素贝叶斯有没有超参数可以调？ 无

60、朴素贝叶斯的工作流程是什么样的？独立性假设，贝叶斯公式的分子，包含先验。

61、朴素贝叶斯对异常值敏感不？ //不敏感。

关于HMM与CRF

1. 两者区别于联系

多项式分布（Multinomial Distribution）是二项式分布的推广。 二项分布的典型例子是扔硬币，硬币正面朝上概率为p, 重复扔n次硬币，k次为正面的概率即为一个二项分布概率。 把二项分布公式推广至多种状态，就得到了多项分布。

贝叶斯模型中的先验p(y)服从多项分布，p(x|y)服从正态分布。

