链滴

2021 年 4 月底，腾讯应用研究岗暑期实习面试题 12 道

作者：julyedu

原文链接：https://ld246.com/article/1623143242879

来源网站：链滴

问题1：决策树有多少种，分别的损失函数是什么?

决策树有多少种，分别的损失函数是什么?决策树有三种:分别为ID3，C4.5，Cart树

ID3损失函数：

$$
\begin{aligned}
H(D \mid A) &= \sum_{i=1}^{n} \frac{|D_i|}{|D|} H(D_i) \\
&= -\sum_{i=1}^{n} \frac{|D_i|}{|D|} \left(\sum_{k=1}^{K} \frac{|D_{ik}|}{|D_i|} \log_2 \frac{|D_{ik}|}{|D_i|} \right)
\end{aligned}
$$

C4.5损失函数 :

$$
\begin{aligned}
\operatorname{Gain}_{\text{ratio}}(D, A) &= \frac{\operatorname{Gain}(D, A)}{H_A(D)} \\
H_A(D) &= -\sum_{i=1}^{n} \frac{|D_i|}{|D|} \log_2 \frac{|D_i|}{|D|}
\end{aligned}
$$

Cart树损失函数 :

$$
\begin{aligned}
\operatorname{Gini}(D) &= \sum_{k=1}^{K} \frac{|C_k|}{|D|} \left(1 - \frac{|C_k|}{|D|} \right) \\
&= 1 - \sum_{k=1}^{K} \left(\frac{|C_k|}{|D|} \right)^2 \operatorname{Gini}(D \mid A) \\
&= \sum_{i=1}^{n} \frac{|D_i|}{|D|} \operatorname{Gini}(D_i)
\end{aligned}
$$

原文链接：2021 年 4 月底，腾讯应用研究岗暑期实习面试题 12 道

问题2：决策树的两种剪枝策略分别是什么？

决策树的剪枝基本策略有预剪枝(Pre-Pruning)和后剪枝(Post-Pruning)。

预剪枝核心思想：

在每一次实际对结点进行进一步划分之前，先采用验证集的数据来验证如果划分是否能提高划分的准性。如果不能，就把结点标记为叶结点并退出进一步划分；如果可以就继续递归生成节点。

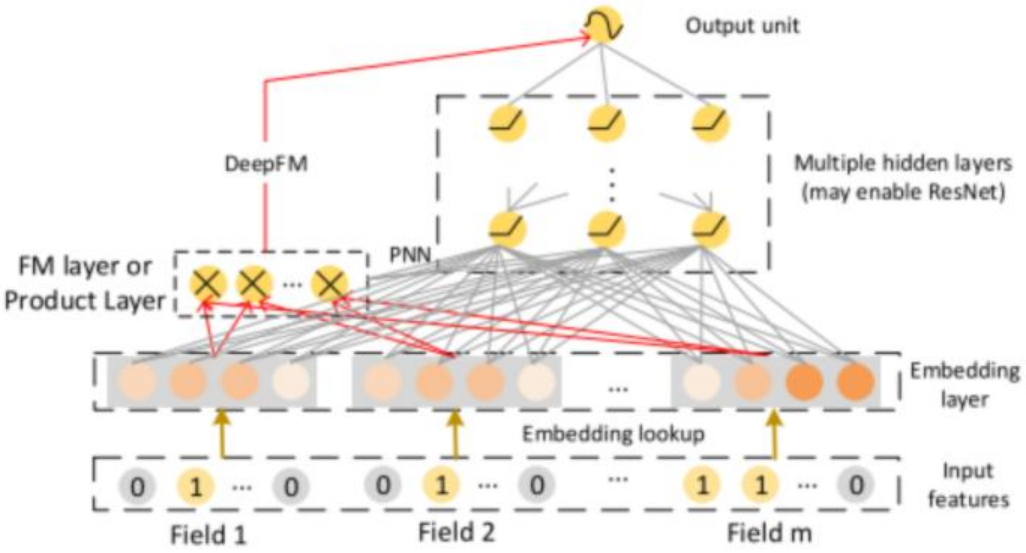
后剪枝核心思想：

后剪枝则是先从训练集生成一颗完整的决策树，然后自底向上地对非叶结点进行考察，若将该结点对的子树替换为叶结点能带来泛化性能提升，则将该子树替换为叶结点。

问题3：信息增益比跟信息增益相比，优势是什么？

以信息增益作为划分训练集的特征选取方案，存在偏向于选取值较多的特征的问题。信息增益比可以决该问题。

问题4：介绍XdeepFM算法，XdeepFM跟DeepFM算相比，优势是什么？



上图为xDeepFM的总体结构，有三个分支:Linear(稀疏的01向量作为输入)、DNN(经过embedding稠密向量作为输入)、CIN（压缩感知层)。

xDeepFM如果去掉CIN分支，就等同于Wide & Deep。

xDeepFM将基于Field的vector-wise思想引入Cross，并且保留了Cross的优势，模型结构也很elegan，实验效果也提升明显。如果说DeepFM只是“Deep & FM”，那么xDeepFm就真正做到了”Dee”Factorization Machine。xDeepFM的时间复杂度会是其工业落地的一个主要性能瓶颈，需要重点化。

问题5：对于长度较长的语料，如何使用Bert进行训练？

对于长文本，有两种处理方式，截断和切分。

-截断：一般来说文本中最重要的信息是开始和结尾，因此文中对于长文本做了截断处理。

head-only:保留前510个字符

tail-only:保留后510个字符

head+tail:保留前128个和后382个字符

-切分:将文本分成k段，每段的输入和Bert常规输入相同，第一个字符是[CLS]表示这段的加权信息。中使用了Max-pooling,Average pooling和self-attention结合这些片段的表示。

问题6：请介绍k-mean算法的原理。

1、选取K个点做为初始聚集的簇心

2、分别计算每个样本点到K个簇核心的距离（这里的距离一般取欧氏距离或余弦距离），找到离该点近的簇核心，将它归属到对应的簇

3、所有点都归属到簇之后，M个点就分为了K个簇。之后重新计算每个簇的重心（平均距离中心），其定为新的“簇核心”；

4、反复迭代2-3步骤，直到达到某个中止条件。

问题7：逻辑回归怎么分类非线性数据?

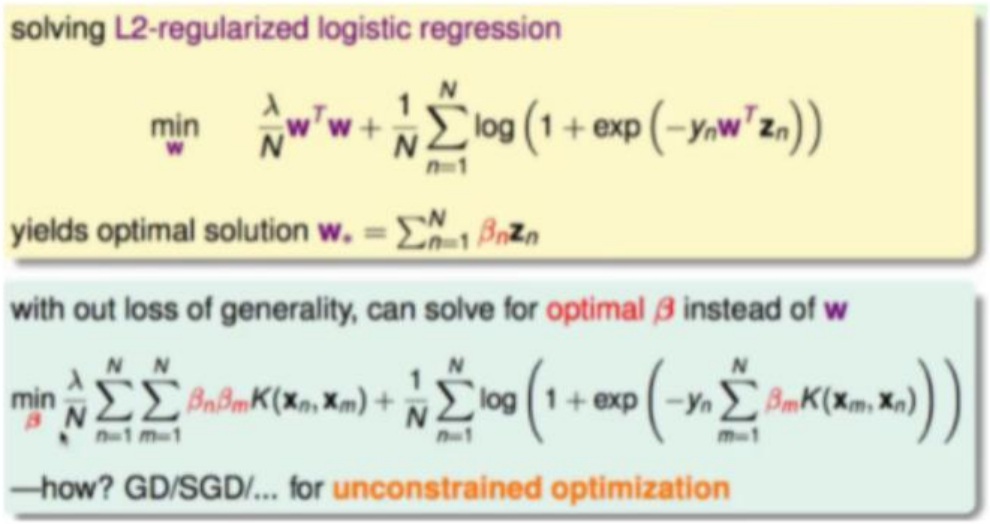
可以，只要使用kernel trick。

不过，通常使用的kernel都是隐式的，也就是找不到显式地把数据从低维映射到高维的函数，而只能计算高维空间中数据点的内积。在这种情况下，logistic regression模型就不能再表示成 $w^T x + b$ 的形式（primal form），而只能表示成 $\sum_i a_i \langle x_i, x \rangle + b$ 的形式（dual form）。忽略那个 b 的话，primal form的模型的参数只有 w，只需要一个数据点那么多的存储量；而dual form的模型不仅要存储各个 a_i，还要存储训练数据 x_i 本身，这个存储量就大了。

SVM也是具有上面两种形式的。不过，与logistic regression相比，它的dual form是稀疏的——只有支持向量的 a_i 才非零，才需要存储相应的 x_i。所以，在非线性可分的情况下，SVM用得更多。

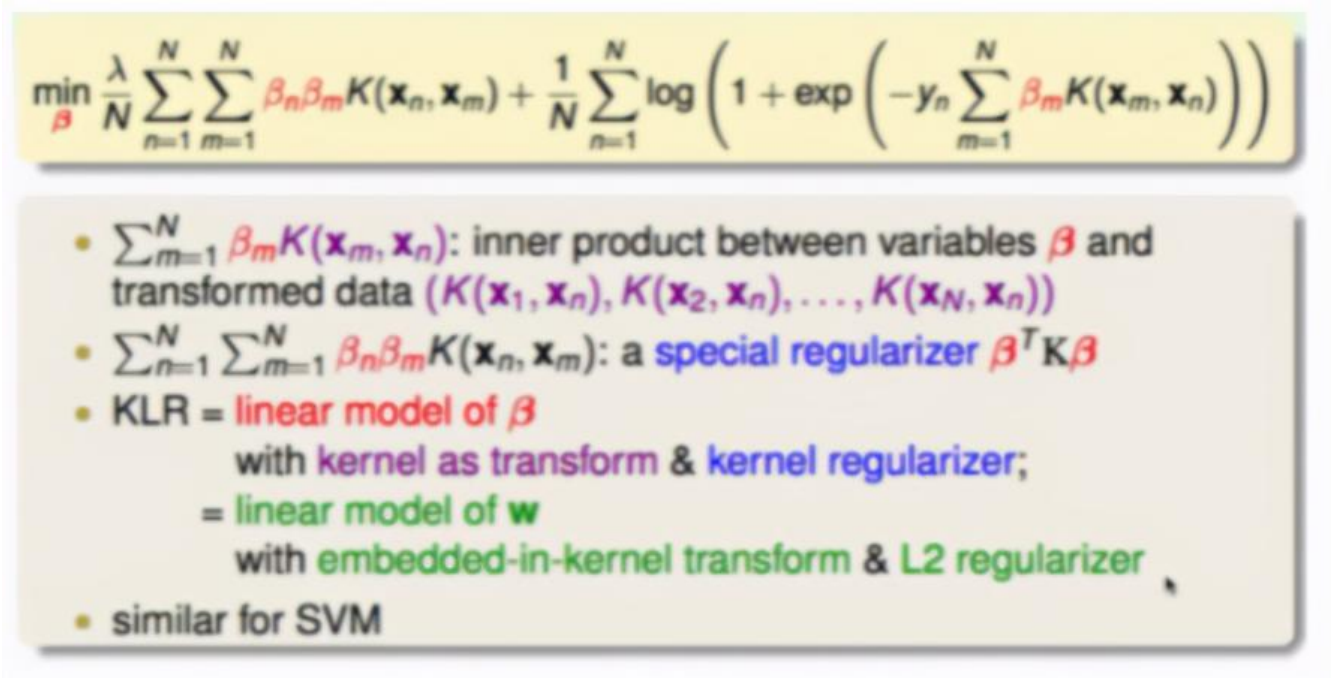
问题8：逻辑回归引入核方法后损失函数如何求导?

用核函数的方法来解决一个L2正则化的逻辑回归如下图:



我们直接将W*表示成β的形式带到我们最佳化的问题中，然后就得到一个关于β的无条件的最佳化问。这时我们可以用梯度下降法或随机梯度下降法来得到问题的最优解。

再仔细观察核函数逻辑回归之后会发现它可以是一个关于β的线性模型:



其中kernel函数既充当了转换的角色有充当了正则化的角色，这种角度同样适用于SVM演算法。需要意的是:SVM的解α大多都是O，核函数逻辑回归的解β大多都不是0这样我们会付出计算上的代价。

问题9：请介绍几种常用的参数更新方法。

梯度下降：在一个方向上更新和调整模型的参数，来最小化损失函数。

随机梯度下降(Stochastic gradient descent，SGD)对每个训练样本进行参数更新，每次执行都进行次更新，且执行速度更快。

为了避免SGD和标准梯度下降中存在的问题，一个改进方法为小批量梯度下降（Mini BatchGradient escent)，因为对每个批次中的n个训练样本，这种方法只执行一次更新。

使用小批量梯度下降的优点是：

1)可以减少参数更新的波动，最终得到效果更好和更稳定的收敛。

2)还可以使用最新的深层学习库中通用的矩阵优化方法，使计算小批量数据的梯度更加高效。

3)通常来说，小批量样本的大小范围是从50到256，可以根据实际问题而有所不同。

4)在训练神经网络时，通常都会选择小批量梯度下降算法。

SGD方法中的高方差振荡使得网络很难稳定收敛，所以有研究者提出了一种称为动量(Momentum)技术，通过优化相关方向的训练和弱化无关方向的振荡，来加速SGD训练。

Nesterov梯度加速法，通过使网络更新与误差函数的斜率相适应，并依次加速SGD，也可根据每个数的重要性来调整和更新对应参数，以执行更大或更小的更新幅度。

AdaDelta方法是AdaGrad的延伸方法，它倾向于解决其学习率衰减的问题。Adadelta不是累积所有前的平方梯度，而是将累积之前梯度的窗口限制到某个固定大小w。

Adam算法即自适应时刻估计方法(Adaptive Moment Estimation)，能计算每个参数的自适应学习。这个方法不仅存储了AdaDelta先前平方梯度的指数衰减平均值，而且保持了先前梯度M(t)的指数减平均值，这一点与动量类似Adagrad方法是通过参数来调整合适的学习率n，对稀疏参数进行大幅新和对频繁参数进行小幅更新。

因此, Adagrad方法非常适合处理稀疏数据。

问题10：请介绍Wide&Deep模型。

1.1 Memorization和Generalization

Wide&Deep Mode 就是希望计算机可以像人脑一样，可以同时发挥 memorization和generalization的作用。--Heng-Tze Cheng(Wide&Deep 作者)

1.2 Wide 和 Deep

同样，这两个词也是通篇出现，究竟什么意思你明白了没?其实，,Wide也是一种特殊的神经网络，他输入直接和输出相连。属于广义线性模型的范畴。Deep 就是指Deep Neural Network，这个很好理。Wide Linear Model用于memorization;Deep Neural Network用于 generalization。左侧是 Wid-only，右侧是Deep-only，中间是 wide & Deep:

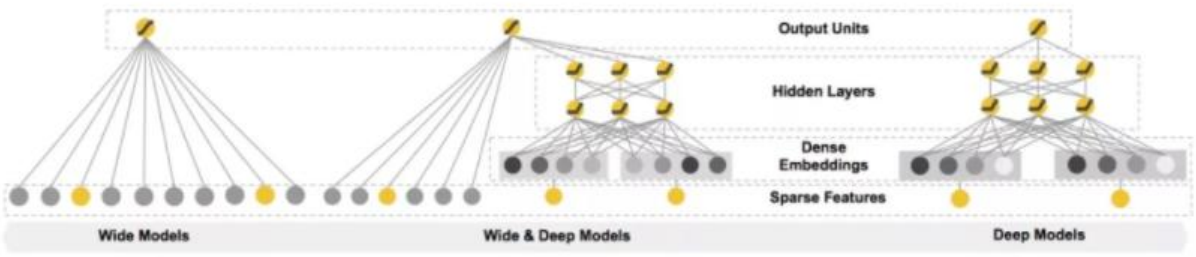
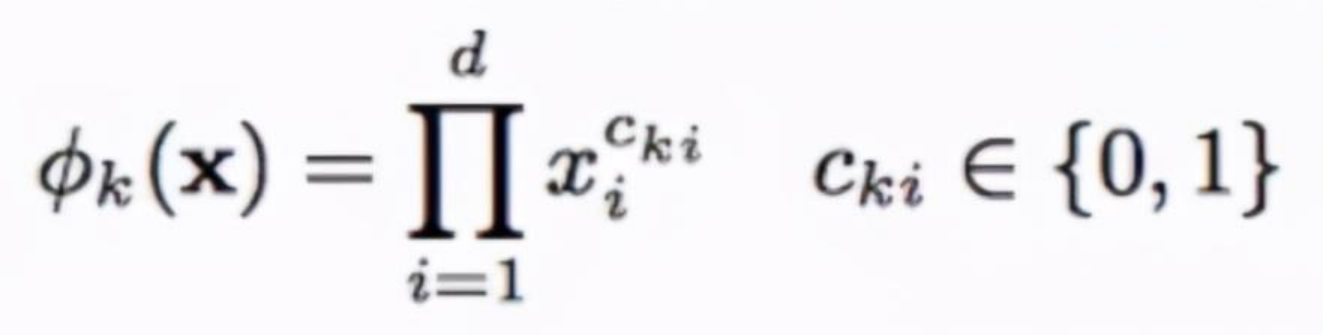


Figure 1: The spectrum of Wide & Deep models.

1.3 Cross-product transformation

Wide 中不断提到这样一种变换用来生成组合特征，也必须搞懂才行哦。它的定义如下:



$$\phi_k(\mathbf{x}) = \prod_{i=1}^{d} x_i^{c_{ki}} \quad c_{ki} \in \{0, 1\}$$

k表示第k个组合特征。i表示输入x的第i维特征。C_ki表示这个第i维度特征是否要参与第k个组合特征构造。d表示输入x的维度。那么到底有哪些维度特征要参与构造组合特征那?这个是你之前自己定好，在公式中没有体现。

饶了一大圈，整这么一个复杂的公式，其实就是我们之前一直在说的one-hot之后的组合特征：仅仅输入样本x中的特征gender=female和特征language=en同时为1，新的组合特征AND(gender=female, language=en)才为1。所以只要把两个特征的值相乘就可以了。

Cross-product transformation可以在二值特征中学习到组合特征，并且为模型增加非线性。

问题11：Xgboost、lightGBM和Catboost之间的异同?

树的特征

三种算法基学习器都是决策树，但是树的特征以及生成的过程仍然有很多不同

CatBoost使用对称树，其节点可以是镜像的。CatBoost基于的树模型其实都是完全二叉树。XGBoos的决策树是Level-wise增长。Level-wise可以同时分裂同一层的叶子，容易进行多线程优化，过拟合险较小，但是这种分裂方式也有缺陷，Level-wise对待同一层的叶子不加以区分，带来了很多没必要开销。实际上很多叶子的分裂增益较低，没有搜索和分裂的必要。

LightGBM的决策树是Leaf-wise增长。每次从当前所有叶子中找到分裂增益最大的一个叶子(通常来是数据最多的一个)，其缺陷是容易生长出比较深的决策树，产生过拟合，为了解决这个问题，LightGM在Leaf-wise之上增加了一个最大深度的限制。

对于类别型变量

调用boost模型时，当遇到类别型变量， xgboost需要先处理好，再输入到模型，而lightgbm可以指类别型变量的名称，训练过程中自动处理。

具体来讲，CatBoost可赋予分类变量指标，进而通过独热最大量得到独热编码形式的结果(独热最大：在所有特征上，对小于等于某个给定参数值的不同的数使用独热编码;同时，在CatBoost语句中设“跳过”，CatBoost就会将所有列当作数值变量处理)。

LighGBM也可以通过使用特征名称的输入来处理属性数据：它没有对数据进行独热编码，因此速度独热编码快得多。LGBM使用了一个特殊的算法来确定属性特征的分割值。(注：需要将分类变量转化整型变量;此算法不允许将字符串数据传给分类变量参数)和CatBoost以及LGBM算法不同，XGBoost身无法处理分类变量，只接受数值数据，这点和RF很相似。实际使用中，在将分类数据传入XGBoost之前，必须通过标记编码、均值编码或独热编码等各种编码方式对数据进行处理。

问题12：情景题：有一个大小为1G的文件，文件中每行一个词，每个词最大为16kb，现有内存为1M的计算机，

出词频前100的词。

1.分而治之/hash映射

顺序读取文件，对于每个词x，取hash(x)%5000，然后把该值存到5000个小文件(记为xO,x1,....4999)。这样每个文件大概是200k左右。当然，如果其中有的小文件超过了1M大小，还可以按照类似的方继续往下分，直到分解得到的小文件的大小都不超过1M。

2.hash_map

统计对每个小文件，采用trie树/hash_map等统计每个文件中出现的词以及相应的频率。

3.堆/归并排序

取出出现频率最大的100个词(可以用含100个结点的最小堆)后，再把100个词及相应的频率存入文件这样又得到了5000个文件。最后就是把这5000个文件进行归并（类似于归并排序)的过程了。