可能是全网特征工程实操最通透的...

原创 砍手豪 包大人 包包算法笔记 2022-02-14 09:00

收录于话题
#机器学习 10 #深度学习 11 #特征工程 2

点击蓝字关注我吧～

之前在我写的特征工程方法论里面提了一嘴，用automl搜索+人工启启发式可以高效稳定地完成特征工程，并取得不错的效果。
原文：可能是全网写特征工程最通透的...
不过这篇文章最大的问题是太过于侧重于**是什么，和为什么了**，至于**怎么办**埋了一个很大的伏笔。

包包算法笔记
包大人的算法，程序，机器学习，职场，理财闲谈。
62篇原创内容

公众号

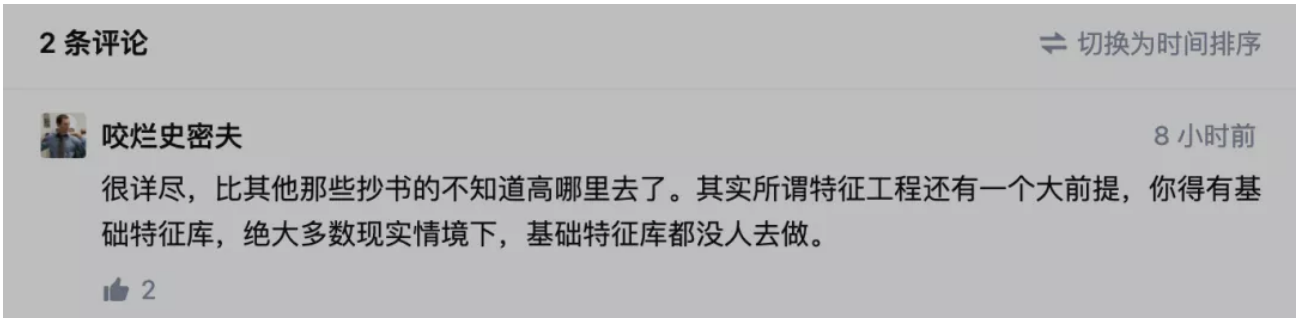
砍老师今天把这个思路给大家详细的写出了，可以实操。可以通过点击访问原文直达砍老师的知乎原文，作为kaggle历史总排名12的**GrandMaster**还有很多干货～

可能是全网特征工程**实操**最通透的...

背景

目前网上能搜到的讲特征工程方法基本都是教材里的那一套：缺失值填充，归一化，category特征one-hot，降维等等。但是指望靠这些提升模型性能是远远不够的，特别是对强大的xgb/lgb上述方法几乎是毫无意义。也有一些文章总结了特定业务的特征工程，但是对其他任务也没有泛化能力。

包大人插一嘴，这个评论很有水平，很多回答扯那些老掉牙的预处理。这篇文章基本就是从基础特征出发，衍生到高阶的实操方法论。这个评论的人可能是量化或者金融从业者，他们在基础因子库+启发式人力搜索上走了挺远了～



画导图抄书唬人，但是真的没什么水平～

砍手豪：本文探讨和介绍一下我的特征工程方法论：1.类automl的暴力特征字典思路 2.基于业务理解的特征工程思路 3.基于特征重要性的特征工程思路；然后是上述三者的反复迭代螺旋上升。

要点一

1.类automl的暴力词典搜索

暴力特征字典指的是当给定数据，能在想象力范围能组合出尽可能多的特征，并形成Pipline，加快特征尝试和迭代的速度，就像automl一样。打个比方，当给你两个类别特征A 和B，你能制造出多少个特征用于迭代？简单写十个：

```
count:A_COUNT、B_COUNT、A_B_COUNT
nunique: A_nunqiue_B  (按B对称的下文省略)
ratio: A_B_COUNT/A_COUNT 在A里各个B类所占的比例
average:A_COUNT/A_nunqiue_B A里各个B类的平均数
most: A_most_B 在A类里出现最高的B是哪个
pivot: A_B1_count、A_B2_count A和B类里特定的B1、B2的联合统计
pivot2: A_B1_count-A_B2_count A的B1行为和B2行为的加减乘除
stat1: A_stat_A_B_COUNT 基于A_B_COUNT对A的描述，
stat2 : A_stat_B_COUNT 基于B_COUNT对A的描述，
序列化：初步LDA，NMF，SVD，进一步Word2Vec，doc2vec 再进一步 图神经网络deepwalk，pPRoNE
```

如果再加上numeric、time、target特征，几乎可以组合成无穷无尽的特征

提升方法：可以看各个数据挖掘的比赛获胜solution，我最初（17年）就是反复看当时几个kaggle GM plantgo&pipiu、Eureka&weiwei、Little Boat&jiwei liu的获胜方案开源，拓宽自己对特征工程的想象力。

缺陷：这类会产生大量特征，比如给五个category特征，就能组成(2**5-1)共31个count特征 ，自然也有大量无用特征，会降低模型质量和速度。
2.基于业务理解的特征工程思路

要点二

2.基于业务理解做特征

通过内在的业务逻辑去分做特征，可以先想业务逻辑，然后数据分析验证，也可以数据分析验证，然后得到业务逻辑，最大的好处是可解释性强，在此基础上泛化能力更强，而且模型规模小。举几个例子：

在Instacart Market Basket Analysis比赛，预测美国用户在线上商店的购物，我想我平时上班，买水果零食这种可买可不买的都放在周末，然后对这个数据里进行分析，发现在Instacart里，酒类商品的销量也集中在周末，因此做了很多item 和 time 交叉的特征，对模型提升较大。

在TalkingData AdTracking Fraud Detection Challenge比赛里，任务是判断虚假点击，通过数据分析发现低频IP容易是Fraud样本，仔细想这些Fraud点击都是自动化程序q每次随机生成的ip，因此容易是低频ip，而正常的ip因为是运营商动态分配共享的，因此普遍频率高。因此做了对channel，ad和ip频次的交叉特征，对模型提升较大。

提升方法：通过努力的数据分析，以及多交流获取业务的内在逻辑形式。在实际中就是多加几个行业群，多看论文多交流，在比赛中就是多逛论坛，看其他人的讨论。

缺陷：凭借业务逻辑做特征，容易遗漏掉强特征。很多时候并不能琢磨出全部的内在业务逻辑，甚至会主动的筛掉一些实际有价值的特征。

要点三

3.基于特征重要性表的特征工程思路

xgb/lgb可以输出特征重要性表，比起相关性分析，通过特征重要性表我们可以迅速在大量特征中获取强特征。在此基础上我们可以对强特征做更深层次的挖掘。

在Two sigma Rental-Listing-Inquireies里，GM little boat提到，既然manager id是强特征，那我们就可以用各种category，numeric特征去描述它。这里涉及到一个问题，很多人说FM，深度学习因为embedding的存在而具有了向新id泛化的能力，而树模型只会记忆。其实在我看来，特征工程就是一个人工embedding的过程，让高维度的类别特征数值向量化，因此也提升了树模型的泛化能力。回到这个比赛，就是特征重要性表为我们指明了特征工程努力的方向。在IJCAI2018 的比赛里，top2 solution 就是采用将特征重要性表靠前的数值特征暴力交叉，期望通过这种方法提高模型获取更多有价值的特征。

类似的，如果看到一个数值特征特征重要性很强，我们也可以用类别特征和其交叉。如果一个统计特征很重要，我们可以增加一个时区维度，比如最近一周，最近一个月的相应统计特征。如果距离上次时间很重要，我们可以增加距离上两次，上次三次的时间特征。等等。

进一步，特征重要性表也可以知道深度学习模型子结构的选择，序列特征对应rnn类，交叉特征o对应fm类，文本特征对应nlp类，如果特征不重要，就不用上相应的结构了，如果重要，就可以对将特定的特征输入对应的子结构了。

提升方法：经验的积累，如何将一个特征发散开来。

缺陷：首先得做出强特征，然后才能在强特征基础上发散，因此依赖一个好的特征重要性表

上文讲了三个我所认为最主要的特征工程思路，但是他们各有各的缺陷，因此
如何将其结合起来互补，螺旋迭代提升就是接下来能做的了。

要点四

4.类automl的暴力特征字典思路对基于业务理解的特征工程思路的协助

前文说到，基于业务理解的特征工程容易遗漏特征，不能挖掘全部可能存在的业务逻辑。那么我们可以先暴力特征字典全部罗列起来，然后在赋予其业务逻辑，看其在当前业务下是否有效。再回到第一项的暴力特征字典。我们把category A和B替换成user，item

```
count:user_COUNT（用户活跃度）、item_COUNT（商品热度）、user_item_COUNT（用户对特定商品的喜爱）
nunique: user_nunqiue_item  (一个用户购买多少种商品) item nunique_user (一个商品被多少个不同用户购买)
ratio: user_item_COUNT/user_COUNT （某个商品在user购买中的比例，喜爱程度）
average:user_COUNT/user_nunqiue_item （平均每类商品的购买量）
most: user_most_item （用户最喜爱的品类）
pivot: user_item1_count、user_item2_count （用户和特定商品的交互）
pivot2: user_item1_count-user_item2_count （用户不同行为的差值，比如生活用品和娱乐用品的比例）
stat1: user_stat_user_item_COUNT （max:买的最多的商品的数量，std：不同商品的分散度，是专宠还是偏爱）
stat2 : user_stat_item_COUNT （mean:用户是喜欢热门商品还是冷门商品）
序列化：初步LDA，NMF，SVD（用商品描述用户画像）
进一步Word2Vec，doc2vec 再进一步 图神经网络deepwalk，pPRoNE（刻画商品和用户的共现性和相似性）
```

要点五

5.类automl的暴力特征字典思路对基于特征重要性表的特征工程思路的协助

首先我们原始 data去跑特征重要性表，知道某个category特征或numeric很重要，要进一步挖掘这个特征的时候，比如前文说的“在Two sigma Rental-Listing-Inquireies里，GM little boat提到，既然manager id是强特征，那我们就可以用各种category，numeric特征去描述它。”我们就可以基于暴力特征字典去强化这个特征，看看如何去拓展这个强特征的维度。

要点六

6.基于业务理解的特征工程思路 和 基于特征重要性表的特征工程思路 对 类automl的暴力特征字典思路的协助

类automl的暴力特征字典思路最大的问题是可以产生无数的特征，比如五个类别特征就能产生31种count特征，这时候我们可以基于特征重要性表，把特征重要性低的类别特征从组合中删去，也可以基于业务理解，把一些明显无相关性的category交叉移除。这样就不会产生过多无用的特征变成噪音降低模型速度和精度。

要点七

7.基于业务理解的特征工程思路 和 基于特征重要性表的特征工程思路 的相互迭代

其实比起数据分析，特征重要性表是一个可以更快的理解业务逻辑的方法
如果一个特征重要性表里存在一个我们原本认为应该无关紧要的特征却有很高的重要性，其实就可以增强我们对业务的理解，我们需要从业务角度思考为什么这个特征有好的效果，然后从业务角度上去做一个更好的特征。
比如特征重要性表里category A 和 numeric B特征都很重要，虽然无论树模型还是深度学习模型都已经有很强的特征交叉能力了，但经过业务分析，其实是 A_mean_B特征影响结果，原始的A和B还是不如我们直接把A_mean_B做出来效果好。

因此，通过观察特征重要性表，思考背后真正的业务逻辑，找出真正和target直接相关的特征，既能提升对业务的理解，也能够提升模型的性能。

在Avito Demand Prediction Challenge（类似闲鱼的一个app转化预测）里，大家发现各种**category_mean_price - price**有很高的特征重要性，因此冠军little boat思考出这不就是合理价格和卖家出价的差影响转化率嘛，于是干脆先建了一个子模型，先预测出pred_price,然后用pred_price-price用于转化率模型，取得了更好的效果，这就是基于特征重要性来理解业务，深挖特征的一个好的案例。

总之，就是三种特征工程思路相互补充，反复迭代，最后通过验证集取得一个好的特征组合。

小结

本文从方法论角度探讨和总结了我的特征工程方法，基本上毫无保留。但是要反思的就是，使用这一套方法论是无法和最好的特征工程大师（比如国内的江离、otto数据挖掘俱乐部）还是有很大差距，我猜测一下可能用以下两种原因：

1. 高手们还有其他角度的特征工程构造逻辑
2. 现有的特征工程逻辑我做的还不够好，比如即便我观察特征重要性表知道某些特征很重要后，也经常无法真正挖掘出反应业务逻辑的深层特征，需要后续看其他人的开源才能焕然大悟。

在这里抛砖引玉，供大家参考。
