_res

推荐系统心得分享

在用的模型总结和场景理解以及改进方向

Zhaojun HAO

MAR 28 2019

目录

1. 推荐系统
   - 简介
   - 特点
   - 现有方法
   - Factorization Machine (FM) 和 Field-aware Factorization Machine (FFM)
   - Wide & Deep (Wide and Deep)
   - 未来工作

信息过滤系统

推荐系统是一种信息过滤系统，用于在商品服务数量过大的背景下推荐用户可能感兴趣的商品或服务，增加商业利益.
从一定程度上，推荐系统是个排序问题. 给定一定的条件，比如用户和时间，来给出排名最靠前的物品. 不过因为实用性原因，推荐系统往往是对特定场景下每个物品打分，然后根据这个输出的分数来排序.

特点 1

推荐系统往往有以下特点:

- 在输入空间的层面, 数据稀疏 (因为模型训练一般用的是梯度法, 稀疏的输入会导致训练时参数更新缓慢) 且影响因子众多. 不论是用户, 物品, 还是上下文情景, 种类都会很多. 很多隐空间或者嵌入的方法也就应运而生. 用户的消费行为背后往往有复杂的经济, 社会, 个体心理因素, 所以要求我们合理选择自变量.
- 在输出空间的层面, 标签不容易打, 预测量很难准确给出. 目前的预测量基本上是基于样例的, 也就是说每次推荐行为都被看做是独立的. 很多模型的假设是用户的特征提取充分, 拿着用户特征和物品比对, 就可以得出兴趣度或评分. 但在很多业务场景中, 往往用户的消费行为是和商家一系列互动的结果.

特点 2

- 在目标函数的层面，因为输出空间的多样化，存在着多种目标函数. 目前的目标函数一般有二元分类情况下的 hinge loss(不可导),logit loss; 多分类情况下的 cross-entropy; 回归情况下的 MAE(不可导),RMSE(Root Mean Square Error). 如何选择目标函数，把业务目标考虑进去是个大问题.
- 在评价方法的层面，目前使用的离线评价方法往往有分类情况下的 precision,recall,F1; 多分类情况下的 accuracy; 连续情况下的 auc. 但是离线评价不是很有说服力，因为业务场景和用户需求很可能会随时间改变. 而在线的效果评估一般使用 AB test，如何结合业务定义评价方法是个问题. 比如有的用点击率，有的用最终转化率，有的用观看时长. 因实际业务情况而调整.

所以推荐系统从特征提取，数据集建设，模型训练到模型评价都绝非易事.

常用方法

推荐系统，按照实现方法，分为基于规则和机器学习两种类型. 我们重点在于机器学习模型.
最基本的模型，按照数据设定和训练过程，分为有监督和无监督. 有监督模型用的是有标签的数据集，通过不断减小预测值和真实值差别来训练模型. 无监督的有:

- Collaborative Filtering (联合过滤)，分为基于用户的和基于物品的，利用了评分矩阵
- Deep Walk (Deep Walk)，利用了行为数据对物品做嵌入

有监督的有:

- SVD (奇异值分解法)，将用户和物品映射到隐空间，利用了评分矩阵
- Logistic Regression (逻辑回归)，属于Generalized Linear Model (广义线性模型)，利用了行为数据的一阶特征
- Fm和FFM，利用了行为数据的一阶和二阶特征
- Wide and Deep，利用了行为数据的高阶特征和特征嵌入

FM

FM可以用于任何实数特征向量，显式得利用行为数据的高阶（一般为了计算效率使用二阶）特征组合，比如情人节，巧克力. 并且考虑到输入特征稀疏的特点，比如为了学习情人节和巧克力的二阶效果，不要求历史上出现过两者的组合. 凡是出现过情人节或巧克力的样例都可以帮助学习特征的隐向量.

1) Model Equation: The model equation for a factorization machine of degree $d = 2$ is defined as:

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j \quad (1)$$

where the model parameters that have to be estimated are:

$$w_0 \in \mathbb{R}, \quad \mathbf{w} \in \mathbb{R}^n, \quad \mathbf{V} \in \mathbb{R}^{n \times k} \quad (2)$$

And $\langle \cdot, \cdot \rangle$ is the dot product of two vectors of size k:

$$\langle \mathbf{v}_i, \mathbf{v}_j \rangle := \sum_{k} v_{i,f} \cdot v_{j,f} \quad (3)$$

FFM

FFM是FM的改进，加入了二阶特征组合时特征所属领域的考虑. 比如情人节，和巧克力，属于物品，组合时的隐向量不同于和下雨，属于天气，组合时的隐向量.

$$\phi_{\mathrm{FFM}}(w, x) = \sum_{j_1=1}^{n} \sum_{j_2=j_1+1}^{n} (w_{j_1, f_2} \cdot w_{j_2, f_1}) x_{j_1} x_{j_2}, \quad (4)$$

where f_1 and f_2 are respectively the fields of j_1 and j_2. If f is the number of fields, then the number of variables of FFMs is nfk, and the complexity to compute (4) is $O(\bar{n}^2 k)$. It is worth noting that in FFMs because each latent vector only needs to learn the effect with a specific field, usually

$$k_{\mathrm{FFM}} \ll k_{\mathrm{FM}}.$$

Figure: FFM 公式

Wide and Deep结构

基于行为数据, 联合训练带有特征转换的线性模型, 也就是 wide 部分, 和用于稀疏特征的神经网络,deep 部分.

- wide 部分用于记忆已经出现的特征组合, 而特征转换主要是Cross-Product Transformation (特征叉乘转换).
- 神经网络用于泛化到没有见过的特征组合.

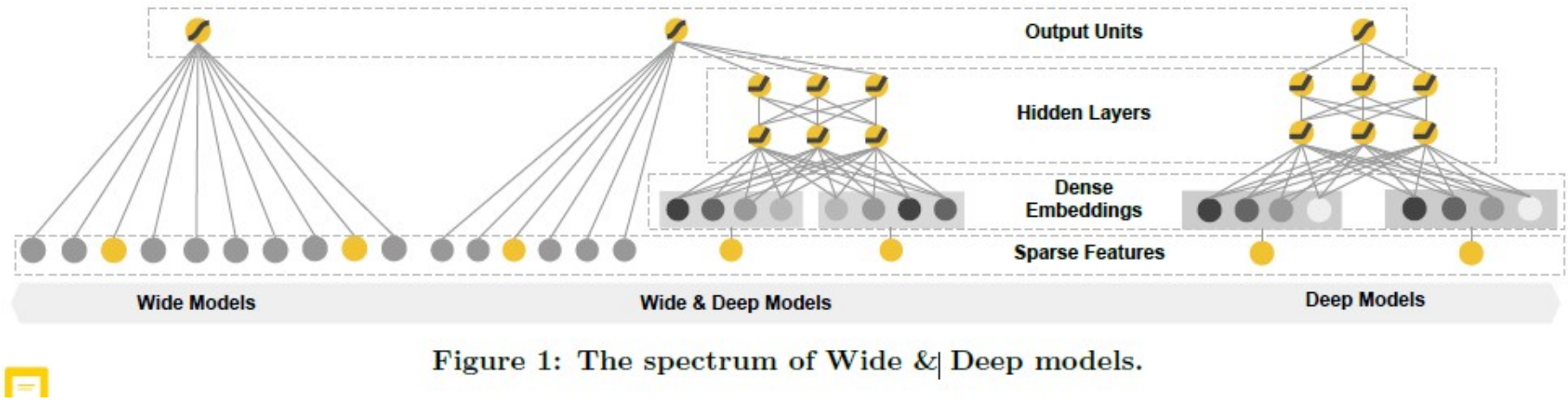

Figure 1: The spectrum of Wide & Deep models.

Figure: wd 模型结构

Wide and Deep输出

Wide and Deep的 wide 部分和 deep 部分的结果作为逻辑回归的输入，得到模型的输出，这样在模型训练过程中两个部分的参数是一起训练的. 不同于在输出阶段融合不同模型的 ensemble 方法，联合训练需要的参数更少，在Wide and Deep中 wide 部分只是对 deep 部分补充，不需要全部训练.

For a logistic regression problem, the model's prediction is:

$$P(Y = 1|\mathbf{x}) = \sigma(\mathbf{w}_{wide}^T[\mathbf{x}, \phi(\mathbf{x})] + \mathbf{w}_{deep}^T a^{(l_f)} + b) \quad (3)$$

where Y is the binary class label, $\sigma(\cdot)$ is the sigmoid function, $\phi(\mathbf{x})$ are the cross product transformations of the original features $\mathbf{x}$, and b is the bias term. $\mathbf{w}_{wide}$ is the vector of all wide model weights, and $\mathbf{w}_{deep}$ are the weights applied on the final activations $a^{(l_f)}$.

Figure: wd 输出

数据处理

```
hive> select uid, cid, cidsinsection, day, hour from ods_app_action_comic_click_h where day='2019-03-21' limit 2;
OK
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
64460573        9680    7119|17745|106574|25934|9680    2019-03-21      00
64460573        106346  106574|106309|106346|106861     2019-03-21      00
```

Figure: 点击原始数据

名称	修改日期	类型	大小
cid_humanType.json	2019/3/18 15:56	JSON 文件	1,186 KB
neg	2019/3/18 17:48	OpenOffice.org ...	73,311 KB
pos	2019/3/18 17:48	OpenOffice.org ...	26,739 KB
pred_test	2019/3/21 12:08	OpenOffice.org ...	0 KB
test	2019/3/19 15:41	OpenOffice.org ...	31,672 KB
train	2019/3/19 15:41	OpenOffice.org ...	95,021 KB
uid_ind.json	2019/3/18 15:57	JSON 文件	7,912 KB
uid_interest.npz	2019/3/18 15:57	NPZ 文件	2,706 KB
valid	2019/3/19 15:41	OpenOffice.org ...	31,675 KB

wide 部分输入

wide 部分特征可以是实数型，也可以是类别型，交叉部分特征需要类别型

```
# wide columns and deep columns.
base_columns = [
    dotw, hour, cid_ht,
    theme_1, theme_2, theme_3, theme_4, theme_5, theme_6, theme_7, theme_8, theme_9, theme_10, theme_11,
    theme_12, theme_13, theme_14, theme_15, theme_16, theme_17, theme_18, theme_19, theme_20, theme_21, theme_22,
    theme_23, theme_24, theme_25, theme_26
]
```

Figure: wide 变量

```
crossed_columns = [
    tf.feature_column.crossed_column(
      [dotw, hour], hash_bucket_size=168),
    tf.feature_column.crossed_column(
```

deep 部分输入

deep 部分特征需要是类别型，因为 deep 部分做的是对离散特征的嵌入.

```
deep_columns = [
    tf.feature_column.embedding_column(dotw, 5),
    tf.feature_column.embedding_column(hour, 5),
    tf.feature_column.embedding_column(cid, 8),
    theme_1, theme_2, theme_3, theme_4, theme_5, theme_6, theme_7, theme_8, theme_9, theme_10, theme_11,
    theme_12, theme_13, theme_14, theme_15, theme_16, theme_17, theme_18, theme_19, theme_20, theme_21, theme_22,
    theme_23, theme_24, theme_25, theme_26
]
```

Figure: deep 变量

结果

用两周 1000 000 条数据训练的模型在 5 000 条样例上的结果

```
precision: 0.7826806398234 97
recall: 0.5880646498135101
f1: 0.6715570279223853
acc: 0.7826806398234 97
auc: 0.7178823442341614
```

Figure: wd_res

FM和Wide and Deep的对比

出于计算效率的考虑,FM一般只是用二阶特征, 而且二阶特征做内积时的隐向量都维度一样. 而Wide and Deep中不同的 deep 部分特征的隐向量的维度可以不同, 并且可以实现多个特征的非线性组合 $[featureEmb_1; featureEmb_2; ...; featureEmb_m] \rightarrow R^d$.
Wide and Deep的特征叉乘转换需要手工选择, 并且 deep 部分是全部 deep 部分特征的全连接, 也就是说 deep 部分体现的众多特征的高阶交互, 没有单独提炼出二阶的特征交互等解释性比较强的部分.

未来工作

1. 用户画像和标签系统，优化输入空间，比如通过 NLP 重新将漫画分类
2. 数据漏斗，优化输出空间，比如根据用户后续的动作来为用户对漫画的兴趣打分
3. 明确目标，优化目标函数和评价指标，这个要求手动修改现有的模型实现.
4. 模型融合，多个模型的结果如何融合.

结尾

谢谢