一初 (/users/72142) 2017-09-14 19:15:37 (最初创作于: 2014-12-30 14:13:44) 发表于: 新零售平台事业部数据技术团队 (/teams/388)

知识体系: 机器学习 (/articles/?kid=201) 其它 (/articles/?kid=375) 其他 (/articles/?kid=515)
修改知识体系

文章标签: ftrl算法优化 (/search?q=ftrl算法优化&type=INSIDE_ARTICLE_TAG) 共享业务事业部-数据应用-基础推荐算法 (/search?q=共享业务事业部-数据应用-基础推荐算法&type=INSIDE_ARTICLE_TAG) 分布式在线学习算法 (/search?q=分布式在线学习算法&type=INSIDE_ARTICLE_TAG) onlinelearning算法简介 (/search?q=onlinelearning算法简介&type=INSIDE_ARTICLE_TAG) 技术节 (/search?q=技术节&type=INSIDE_ARTICLE_TAG) 修改标签 标签历史 (/articles/27545/tags/history)

OnlineLearning 在个性化推荐中的应用实践

算法整体介绍

背景

用户在淘宝的行为丰富且多样，通过离线训练的model很难捕获用户时刻变化的行为。因此我们需要实时的样本采集、特征计算、模型训练和打分排序，来满足用户实时个性化的需求，为用户提供精准推荐。

Onlinelearning算法简介

Online算法VS离线算法

批处理的离线机器学习方法在每次迭代计算的过程中，需要把全部的训练数据加载到内存中计算（例如计算全局梯度）， 虽然有分布式大规模的机器学习平台，在某种程度上批处理方法对训练样本的数量还是有限制的，onlinelearning不需要cache所有数据，以流式的处理方式可以处理任意数量的样本。研究onlinelearning有两个角度，在线凸优化和在线Bayesian。

在线凸优化的方法

Shai Shalev-Shwartz 在Foundations and Trends in Machine Learning中有篇关于onlinelearning的综述 Online Learning and Online Convex Optimization，这个既有理论又有实践，是onlinelearning很好入门资料。

截断梯度法

在线梯度下降的方法不容易产生稀疏解，为了产生稀疏解，简单的根据设定阀值进行截断的方法显得粗暴，而且效果有限。张潼等提出截断梯度法对此做了改进：

$$W^{(t+1)} = T_1\left(W^{(t)} - \eta^{(t)}G^{(t)}, \eta^{(t)}\lambda^{(t)}, \theta\right)$$

$$T_1(v_i, \alpha, \theta) = \begin{cases} max(0, v_i - \alpha) & if\ v_i \in [0, \theta] \\ min(0, v_i + \alpha) & if\ v_i \in [-\theta, 0] \\ v_i & otherwise \end{cases}$$

(http://img4.tbcdn.cn/L1/461/1/47c018672a373aef5181b63c7d3543033f1f3193)

FOBOS算法

先前向后切分（FOBOS，Forward-Backward Splitting）是由John Duchi 和Yoram Singer提出，在FOBOS中，将权重的更新分为两个步骤：

$$W^{(t+\frac{1}{2})} = W^{(t)} - \eta^{(t)}G^{(t)}$$

$$W^{(t+1)} = argmin_W\left\{\frac{1}{2}\left\|W - W^{(t+\frac{1}{2})}\right\|^2 + \eta^{(t+\frac{1}{2})}\Psi(W)\right\}$$

(http://img3.tbcdn.cn/L1/461/1/3484e9bd910497959038daa3862bc3c5ff1b4e27)
前一个步骤就是一个标准的梯度下降过程，后一步是对前一步下降的结果进行微调。观察第二个步骤发现对W的微调也分两部分，第一部分保证微调不要偏离第一步找到的梯度下降方向太远，第二部分则用于处理正则化，产生稀疏解。

RDA算法

TG、FOBOS都还是建立在SGD的基础上，属于梯度下降类型的方法，这类型方法的优点就是精度比较高，并且TG和FOBOS也都能产生稀疏解。正则对偶平均（RDA，Regularized Dual Average）是从另一个方面来求解Online Optimization。RDA算法中优化的是前T轮所有loss的平均，可以达到平均效果较好。

FTRL-P算法

FTRL-P是google McMahan提出的算法，它综合考虑了FOBOS和RDA的优点，兼具FOBOS的精确性和RDA优良的稀疏性，其特征权重更新公式为：

$$\boldsymbol{w}_{t+1} = argmin_w\left(\sum_i^t \boldsymbol{g}_i * \boldsymbol{w} + \frac{1}{2}\sum_i^t \sigma_i * \left\|\boldsymbol{w} - \boldsymbol{w}_i\right\|_2^2 + \lambda_1\|\boldsymbol{w}\|_1 + \frac{\lambda_2}{2}\|\boldsymbol{w}\|_2^2\right)$$

(http://img2.tbcdn.cn/L1/461/1/4ae6f2d825644a0132251a26dd8d749c8c036ae6)
该优化公式分为3部分，第一部分表示累计梯度和，就是让损失函数下降的方向；第二部分是表示新的迭代结果不要偏离已经产生的迭代结果太远；第三部分是正则项，有用于产生低遗憾的强凸二范数和产生稀疏解的一范数。另外值得一提的是FTRL采用针对每个特征不同的自适配的学习速率，直观的解释是如果特征出现的次数多，对这个这个特征的权重比较置信，学习速率就比较小，对于很少出现的特征学习速率就比较大。

在线Bayesian 方法

AdPredictor 算法

AdPredictor是微软剑桥研究院在TrueSkill算法的基础上提出的用于Bing CTR预估的在线算法。AdPredictor是Bayesian 的方法，模型的更新可认为是已知先验，又有新的数据到达怎样来更新后验，这也是一个典型的online方法。该算法用概率图模型的消息传递求解，如果因子图中传递的消息全部是高斯消息，求解过程会比较简单，但是ctr预估的似然函数不是高斯分布，导致很难计算，算法就采用了期望扩散算法做高斯近似。AdPredictor算法本身结论很优雅，微软的paper直接给出了更新公式，但是整个推导过程比较复杂。

基于内容的在线矩阵分解算法

源于Netflix推荐大赛，矩阵分解的方法在推荐应用中十分流行。传统矩阵分解算法的输入是用户对物品的打分矩阵，没有用户和物品的特征信息，这可能是由于Netflix 为了保护用户的隐私和内部数据才没有公开这些信息，但实际上，这些信息对推荐效果很重要。基于内容矩阵分解的算法就是兼顾用户和商品的meta 信息和打分矩阵，微软的 《Matchbox: Large Scale Bayesian Recommendations (http://research.microsoft.com/apps/pubs/default.aspx?id=79460)》就是这种方法。

业务贡献

Onlinelearning作为模型精排序的组件在多个猜你喜欢场景上线《PC主路径猜你喜欢算法架构 (http://www.atatech.org/articles/38507?rnd=874474159)》也在detail页面店内个性化的搭配推荐上线，具有突出的表现。

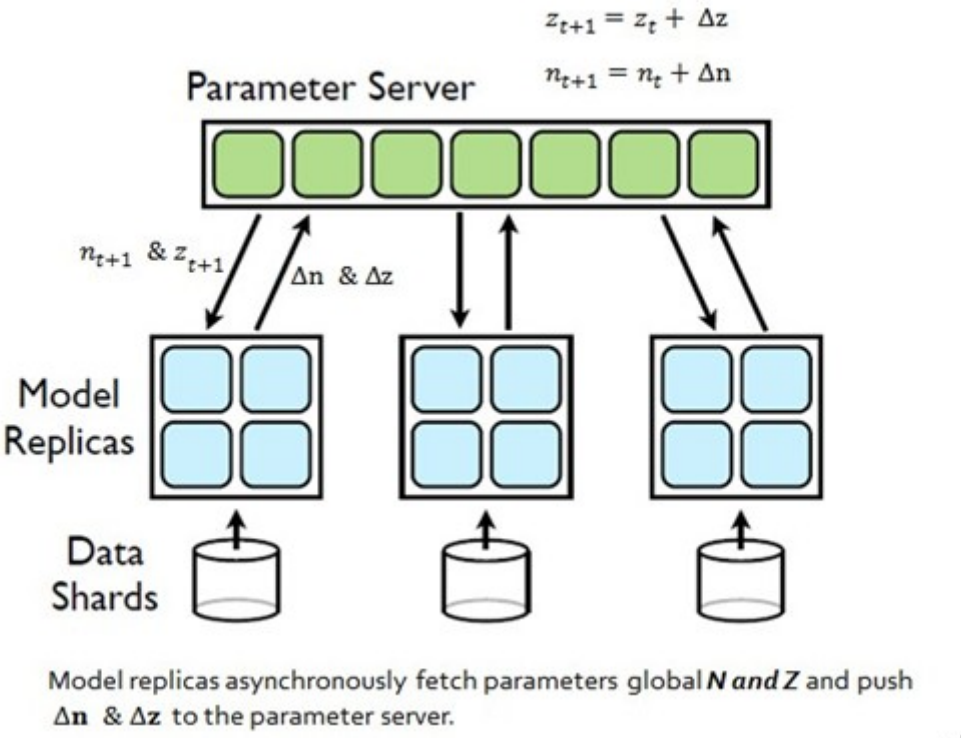
场景名称	日PV	核心指标（ctr）增幅
已买到猜你喜欢	5000w	**17.43%**
购物车猜你喜欢	400w	**14.41%**
评价完成页猜你喜欢	400w	**20.18%**
付款成功页猜你喜欢	500w	**17.07%**
收藏夹-宝贝list猜你喜欢	500w	**5.86%**
收藏夹-店铺list猜你喜欢	400w	**17.93%**
订单详情页猜你喜欢	300w	**16.17%**
物流查询页猜你喜欢-集市	400w	**14.35%**

创新性

分布式并行算法和系统架构

设计分布式并行在线学习算法

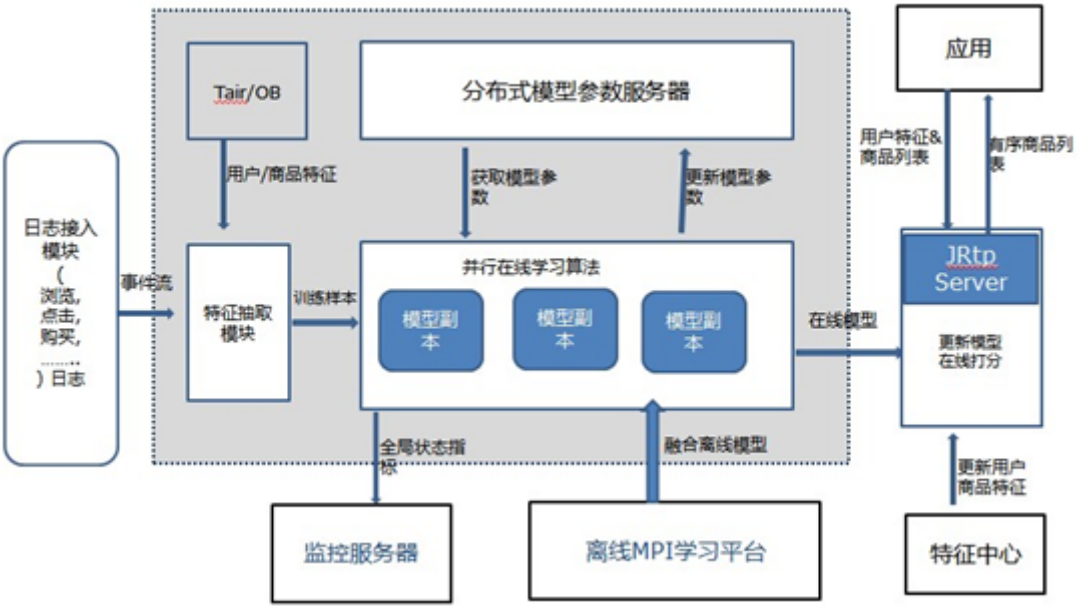
Google提出的FTRL单机版算法数据处理能力有限，远不能够满足集团海量实时数据在线训练的需求。利用Storm实时流式计算框架和parameter Server算法架构设计实现了FTRL的分布式并行版算法。



(http://img4.tbcdn.cn/L1/461/1/1c08a8a212f4edaf504ec12558ee06de48935278)

深入分析FTRL算法发现模型的更新只与参数 n 和 z 有关， 而 n 和 z 的更新是线性增量更新的。 因此 可以利用PS架构做分布式并行算法， 多个worker 并行计算 n 和 z ， PS做全局 n z 增量的累加。如上图所示，整个模型分为多个shards 保存在ParametrServer中并被多个worker更新，训练数据由多个worker并行处理，每个worker 把本地最新模型参数发送给ParameterServer， 并从ParameterServer中获取全局的最新信息，共享的ParameterServer起到全局信息同步作用，worker与worker之间是完全异步操作。

在线学习平台系统架构



(http://img1.tbcdn.cn/L1/461/1/3d8bb92d8418308e9d7275923e725820791e912e) 整个在线学习平台是基于JStorm开发，在 Storm 的jobs 中完成接入日志、在线特征提取、模型训练以及更新模型到在线打分服务。

对FTRL算法的改进优化

1. 将模型参数投影到约束子空间

$$w_{t+1} = \Pi_{\Omega}\left(w_t - \eta_t g_t^f\right) = \underset{w \in \Omega}{\operatorname{argmin}} \left\{ \left\| w - \left(w_t - \eta_t g_t^f\right) \right\|_2^2 \right\}$$

(http://img1.tbcdn.cn/L1/461/1/0b7903b11d66c21e33d6136593485438aeb7d7db)
在线训练不能保证输入样本具有稳定的分布，数据噪音比较大，对模型稳定性和准确性有较大的干扰。把模型更新分为2步，第一步按照原算法更新得到wt，第二步按照上述公式把wt投影到约束空间Ω中，下次迭代以投影后的值为基础更新，实验表明投影后算法有较快速的收敛性和较好的AUC指标。

2. Scale batch更新的delta N 和 delta Z

每个worker并不是直接把batch更新的 delta_n 和 delta_z 发送给PS， 而是根据worker的数量和收到的样本数量做一下 Scale(delta_n, delta_z, worker_num, ins_num). 每个worker 的更新过程不是信息无损的连续过程，每次拉到的全局模型参数时，这是存在增量更新跳跃的，这跳跃量就是别的worker更新的增量， 做下scale 可平滑这个增量。

3. 给正负样本设定不同训练权重

原始的FTRL-P算法中，每个样本的权重是一样的，淘宝的数据即使经过预处理，正负样本的比例差距还是比较大，尝试给正负样本赋不同的权重。由于正样本数据较少，尝试增加正样本的权重，减少负样本的权重。FTRL-P算法对损失函数进行一阶泰勒展开，用负梯度表示损失，我们尝试把权重作用在梯度上。实验发现增加正样本的权重或减少负样本的权重会增加

AUC的值，这与预期一致。 但是减少负样本的权重带来了一个负面的影响，就是增加测试集的总体Loss， 这是可以解释的，减少负样本的权重就相当于在算法中降低了负样本损失带来的影响，这样负样本的损失就会膨胀。

4. 正样本在多个并行模型训练之间复制

这是解决正负样本分布不均的另一种方案，实践表明这种方法也很有效。负样本可以在多个并行训练的模型之间分发路由， 所有的正样本都发送给每一个并行模型训练器，这样每个并行模型训练器就能收到相对较多的正样本，缓解样本分布不均的问题。这个做法也有问题，他其实是改变了样本分布的，学习出来的模型不能很好反应真是的样本分布，最后通过对型做校验可解决这个问题。

5. 离线模型融合

在线训练有冷启动的问题，可以考虑利用离线的模型缓解冷启动，并在某种程度上对在线模型进行修正，设计两种方式融合离线模型的方法。
1. 把离线模型放在优化目标函数中做初始化约束

$$w_{t+1} = argmin_w(\sum_i^t g_i * w + \frac{1}{2}\sum_i^t \sigma_i * ||w - w_i||_2^2 + \lambda_1||w||_1 + \frac{\lambda_2}{2}||w||_2^2 + \frac{\lambda_3}{2}||w - w_0||_2^2)$$

(http://img3.tbcdn.cn/L1/461/1/a11afb21f78238771dc337ee94e24427f415a2fa)

其中w_0 是离线模型， $\frac{\lambda_3}{2}||w - w_0||_2^2$

(http://img1.tbcdn.cn/L1/461/1/19e2758db9bc61cc06b42c7c71c281a4d6b224de) 是我们添加的约束，表示模型不要偏离离线模型太远。
2. 把离线模型放在优化目标函数中做Adaptive约束

$$w_{t+1} = argmin_w(\sum_i^t g_i * w + \frac{1}{2}\sum_i^t \sigma_i * ||w - w_i||_2^2 + \lambda_1||w||_1 + \frac{\lambda_2}{2}||w||_2^2 + \frac{1}{2}\sum_i^t \varphi_i * ||w - w_0||_2^2)$$

(http://img4.tbcdn.cn/L1/461/1/54db4e22c6f7059001ec13b5c0e2ba0da087ca8f)
是我们设计的自适应的离线约束，这个方法在训练过程中不断增强化离线模型的约束，能够在新的数据变化趋势和离线模型之间平衡。

6.利用模型均值进行预测

为了减少在线预测时模型不稳定性，模型更新可按照原始算法进行，预测打分使用当前模型的平均值预测，采用流式数据求均值的方法计算当前模型的均值。

$W_{t+1} = \frac{1}{t+1}\omega_{t+1} + \frac{t}{t+1}W_t$，其中 W_t 是流式的模型均值，ω_t 是当前最新值。

(http://img1.tbcdn.cn/L1/461/1/434377167cf724b723a5d92cf5825e95e0c3b065)
实验表明这个方法很有效。

7 克服在线模型更新come to a halt

问题：Adapive learning Rating 随着样本的增加，每个特征的学习速率会越来越小，这样导致在线更新对数据变化不敏感，有如下2种方法可以解决这个问题。
方法1：离线模型周期训练， 每次都用离线模型初始化在线训练，利用在线学习捕获2次离线学习之间的Gap。
方法2：ParameterServer 端周期的decay 模型参数

$$Z_{t+1} = Z_t * decay\_facor + \Delta Z$$
$$N_{t+1} = N_t * decay\_factor + \Delta N$$

(http://img3.tbcdn.cn/L1/461/1/5eec21ade6c237f7aff7a05ab319e3b1fb652fcf.png)

业界认知度

Google在2013年KDD上发表了FTRL算法后，在业界引起了巨大的反响，国内外各大IT公司纷纷上线该算法。
Amazon在他们的搜索广告中上线该算法取得了不错的效果；
Yahoo在新闻推荐中也有尝试该算法发；
国内网易、搜狐、新浪、百度都有上线该算法，也都取得了不错效果；
集团多个团队也都上线在线学习系统，并取得不错的业务效果。

佐炫 (/users/160957) 旭廷 (/users/160966) 恒扬 (/users/247075) 蓝壹 (/users/254519)

誓空 (/users/338005)

相似文章

- 机器学习中的一些实现选择 (/articles/24958)
- 搜索双链路实时计算体系@双11实战 (/articles/44909)
- 基于 XDL 的深度点击率预估模型探索 (/articles/67232)
- 基于实时Stream的推荐算法及并行架构 (/articles/43759)
- 智能化搜索排序在线学习算法----DRL... (/articles/66101)
- 淘宝展示广告中的OCPC智能调价技术 (/articles/74316)

下一篇：ICML2016 开会总结 (/articles/577…

1F 古飞 (/users/19930) 2014-12-30 18:53:22

赞！专业！

布青 赞同

1 (/comments/41398/voteup) | 0

2F 弗忧 (/users/101549) 2014-12-30 19:13:18

赞

0 (/comments/41401/voteup) | 0

3F 口肃 (/users/70089) 2014-12-30 19:21:56

赞，期待online learning在纷杂的淘宝推荐等大场景的预测模型中得到实时的突破！

0 (/comments/41402/voteup) | 0

4F 陈霖 (/users/15675) 2014-12-31 05:20:56

赞

0 (/comments/41417/voteup) | 0

5F 希宏 (/users/12059) 2014-12-31 09:44:43

点个赞

0 (/comments/41429/voteup) | 0

6F 怀一 (/users/10579) 2014-12-31 09:50:46

赞，online learning 高大上！

0 (/comments/41434/voteup) | 0

7F 澄真 (/users/4577) 2014-12-31 09:51:46

看起来很牛逼的样子啊~

0 (/comments/41435/voteup) | 2

怀一 (/users/10579) 2014-12-31 09:52:07

不只是看起来牛B，是真牛B

0 (/comments/41435/subcomments/11972/voteup)

澄真 (/users/4577) 2014-12-31 10:05:00

@怀一 (/users/10579) 关键哥看不懂，哈哈哈

0 (/comments/41435/subcomments/11976/voteup)

写下你的评论...

8F 濂溪 (/users/69828) 2014-12-31 09:53:33

先收藏再学习。

0 (/comments/41436/voteup) | 0

9F 无待 (/users/78951) 2014-12-31 10:03:11

看起来好高大上

0 (/comments/41444/voteup) | 0

10F 沐剑 (/users/20091) 2014-12-31 10:03:41

赞！干货噢

0 (/comments/41445/voteup) | 0

11F 芳瑞 (/users/15582) 2014-12-31 10:16:05

牛气，tpp又上升了一个台阶！

0 (/comments/41447/voteup) | 0

12F 姬望 (/users/76749) 2014-12-31 10:27:56

不明觉厉！

0 (/comments/41451/voteup) | 0

13F 三桐 (/users/5262) 2014-12-31 15:01:46

很清楚！为什么模型学习过程中需要融合离线模型?

0 (/comments/41491/voteup) | 2

一初 (/users/72142) 2014-12-31 15:32:37

找到一个好的起点， 快速收敛。 另外也能用离线模型平衡在线模型。

0 (/comments/41491/subcomments/11989/voteup)

绍成 (/users/66554) 2014-12-31 23:55:33

@三桐 (/users/5262) online learning模型一般稳定性比较差，在线模型加上离线模型约束可以有效的缓解这种情况

0 (/comments/41491/subcomments/12002/voteup)

写下你的评论...

14F 比谦 (/users/68093) 2014-12-31 15:22:54

非常好的干货分享，赞~

0 (/comments/41498/voteup) 0

15F 竹堡 (/users/94087) 2014-12-31 17:06:50

写的挺好，表示学习。

0 (/comments/41515/voteup) 0

16F 然正 (/users/13554) 2014-12-31 20:25:32

期待发威

0 (/comments/41886/voteup) 0

17F 铁杉 (/users/68195) 2015-01-05 10:50:34

赞！先收藏了!

0 (/comments/42013/voteup) 0

18F 龙缘 (/users/100023) 2015-04-15 11:14:44

赞！！

0 (/comments/52940/voteup) 0

19F 叶歆 (/users/19766) 2015-04-15 12:01:07

收藏，赞!

0 (/comments/52962/voteup) 0

20F 文治 (/users/82238) 2015-08-06 11:36:52

非常专业的好文，干货好多。。。很可喜地看到Online Learning在阿里发挥作用

0 (/comments/63586/voteup) 0

◀ 1 2 (/articles/27545/comments?p=2) ▶ (/articles/27545/comments?p=2)

写下你的评论...

评论

ATA社区常见问题汇总 (/articles/67826)