第七章 人工智能，7.1 基于深度强化学习与自适应在线学习的搜索和推荐算法研究(作者：灵培、霹雳、哲予)

7.1 基于深度强化学习与自适应在线学习的搜索和推荐算法研究

1. 搜索算法研究与实践

1.1 背景

淘宝的搜索引擎涉及对上亿商品的毫秒级处理响应，而淘宝的用户不仅数量巨大，其行为特点以及对商品的偏好也具有丰富性和多样性。因此，要让搜索引擎对不同特点的用户作出针对性的排序，并以此带动搜索引导的成交提升，是一个极具挑战性的问题。传统的Learning to Rank（LTR）方法主要是在商品维度进行学习，根据商品的点击、成交数据构造学习样本，回归出排序权重。LTR学习的是当前线上已经展示出来商品排序的现象，对已出现的结果集合最好的排序效果，受到了本身排序策略的影响，我们有大量的样本是不可见的，所以LTR模型从某种意义上说是解释了过去现象，并不一定真正全局最优的。针对这个问题，有两类的方法，其中一类尝试在离线训练中解决online和offline不一致的问题，衍生出Counterfactural Machine Learning的领域。另外一类就是在线trial-and-error进行学习，如Bandit Learning和Reinforcement Learning。

在之前我们尝试了用多臂老.虎.机.模型（Multi-Armed Bandit，MAB）来根据用户反馈学习排序策略，结合exploration与exploitation，收到了较好的效果。

公告

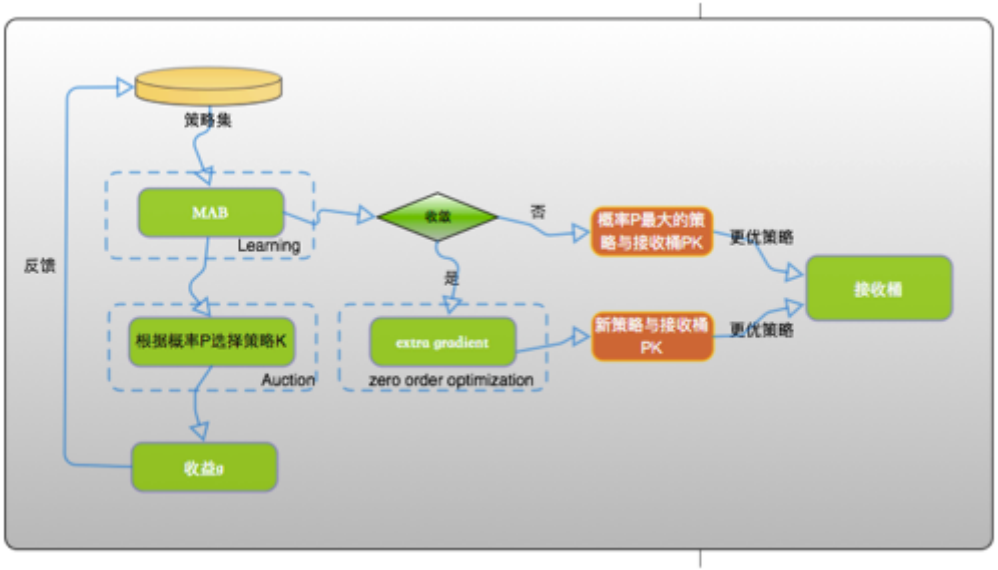
昵称：jiapeng
园龄：2年7个月
粉丝：14
关注：17
+加关注

< 2017年7月 >

日	一	二	三	四	五	六
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

搜索

找找看
谷歌搜索



后来更进一步，在原来的基础上引入状态的概念，用马尔可夫决策过程对商品搜索排序问题进行建模，并尝试用深度强化学习的方法来对搜索引擎的排序策略进行实时调控。

实际上，如果把搜索引擎看作智能体（Agent）、把用户看做环境（Environment），则商品的搜索问题可以被视为典型的顺序决策问题。Agent每一次排序策略的选择可以看成一次试错（Trial-and-Error），把用户的反馈，点击成交等作为从环境获得的奖赏。在这种反复不断地试错过程中，Agent将逐步学习到最优的排序策略，最大化累计奖赏。而这种在与环境交互的过程中进行试错的学习，正是强化学习（Reinforcement Learning，RL）的根本思想。

本文接下来的内容将对具体的方案进行详细介绍。

1.2 问题建模

马尔可夫决策过程（Markov Decision Process，MDP）是强化学习的最基本理论模型。一般地，MDP可以由一个四元组<S, A, R, T>表示：（1）S为状态空间（State Space）；（2）A为动作空间（Action Space）；（3）$R: S\times A\times S \to \mathbb{R}$为奖赏函数；（4）$T: S\times A\times S \to [0,1]$为环境状态转移函数（State Transition Function）。

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

随笔分类

ASP.NET(6)
DHTMLX
Docker(3)
Hadoop学习(3)
Hibernate(5)
Java(62)
Javascript(38)
JQuery(14)
Linux学习(13)
maven(3)
MyBatis(5)
NoSQLDB(3)
Oracle(9)
OtherBeautifulBlogs(35)
Python学习(1)
Spring(16)
Struts(3)
构建相关(1)
算法(2)
杂谈(5)

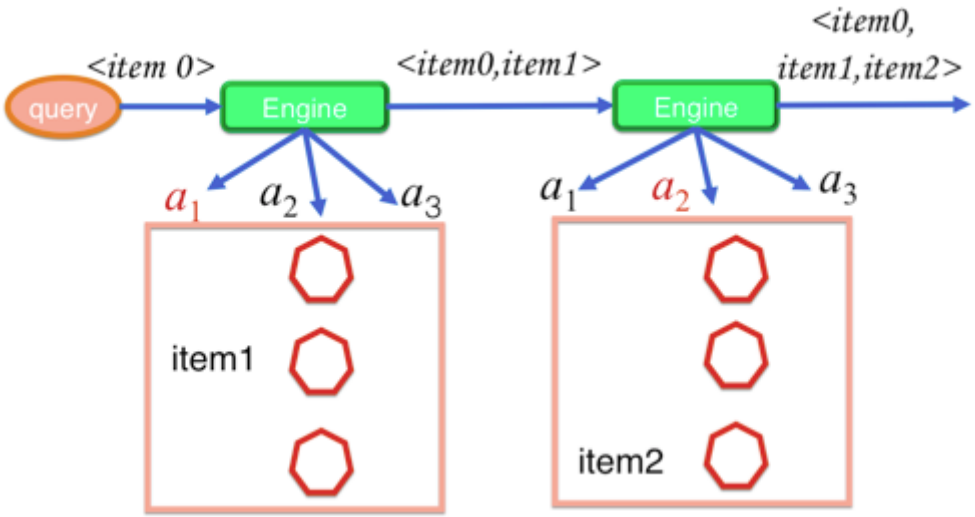
随笔档案

我们的最终目标是用强化学习进行商品搜索排序策略的学习，在实现的过程中，我们一步一步完成了从简单问题到复杂问题的过渡，包括：

1. 基于值表（Tabular）强化学习方法的商品价格档T变换控制（离散状态、离散动作问题）；
2. 基于值表（Tabular）强化学习方法的商品展示比例控制（离散状态、离散动作问题）；
3. 基于强化学习值函数估计（Value Function Approximation）的商品排序策略调控（连续状态、离散动作问题）；
4. 基于强化学习策略估计（Policy Approximation）的商品排序策略调控（连续状态、连续动作问题）。

1.2.1 状态定义

假设用户在搜索的过程中倾向于点击他感兴趣的商品，并且较少点击他不感兴趣的商品。基于这个假设，我们将用户的历史点击行为作为抽取状态特征的数据来源。具体地，在每一个PV请求发生时，我们把用户在最近一段时间内点击的商品的特征作为当前Agent感知到的状态。当然，在不同的问题中，状态的表示方法会有所不同。例如，在值表强化学习方法中，状态为可枚举的离散变量；在值函数估计和策略估计方法中，状态则表示为特征向量。



2017年7月 (9)
2017年6月 (7)
2017年4月 (1)
2017年3月 (3)
2017年2月 (6)
2017年1月 (4)
2016年12月 (32)
2016年11月 (2)
2016年10月 (5)
2016年9月 (4)
2016年8月 (10)
2016年7月 (4)
2016年6月 (5)
2016年5月 (7)
2016年4月 (7)
2016年3月 (4)
2016年2月 (15)
2016年1月 (12)
2015年12月 (3)
2015年11月 (7)
2015年10月 (6)
2015年9月 (7)
2015年8月 (17)
2015年7月 (18)
2015年6月 (10)
2015年5月 (18)
2015年4月 (3)

GIS相关网站

ARCGIS API

开源网站

1.2.2 奖赏函数定义

Agent给出商品排序，用户根据排序的结果进行的浏览、商品点击或购买等行为都可以看成对Agent的排序策略的直接反馈。在第四章中，我们将利用奖赏塑形（Reward Shaping）方法对奖赏函数的表达进行丰富，提高不同排序策略在反馈信号上的区分度。

1.3 算法设计

由于篇幅有限，我们仅对强化学习在搜索中的使用给出2个实例。

（1）Tabular方法

我们在排序中要引入价格的因素来影响最终展示的价格，若以GMV为目标，则简单可以表示为cvr*price，同时我们又想控制价格的作用程度，所以目标稍作修改：$cvr * price^t$，加入一个变量t来控制价格的影响。这个t值的范围很有限，可以用MAB或CMAB来找到最优解。

我们用强化学习的视角来对这个问题进行抽象，把用户前2次点击的商品价格档位（0~7，从低到高）作为状态。这个状态表示的是用户之前点击商品的价格偏好，如果两次都点击1档商品，说明用户偏好低价商品，很有可能接下来用户只对低价商品感兴趣，如果这个状态转移分布是稳定的（stationary），那么一个统计模型可以就可以描述这种规律。而实际上，用户的行为是受我们排序模型的影响的，用户点击1档商品也可能是因为当前的排序策略只给用户展示了1档商品，并不一定是用户的本质需求。在接下来用户的搜索过程中，我们可以有的选择1是只出1档商品让用户的需求快速收敛，选择2是投放一些附近档位的商品供用户选择，如果用户选择了其他档位的商品，进行了状态的转移，就可能找到一个更好的路径，最终的收益和我们所有的过程中的投放策略都相关。从每个时间点上看，策略可能不是最优的，但全局上可能是最优的。

具体地，当用户进行了搜索后，根据用户的状态s，和Q表（下图）进行一个epsilon-greedy的投放，选择一个动作a（上文中的价格指数t），执行这个a的排序结果展示给用户，并记录下这次的状态s与动作a，以及用户对这次搜索结果的反馈r，从用户的点击与否的反馈，再对Q表进行更新。

Bootstrap
DHTML官网
JqueryEasyui
开源中国社区

其他博园

Java孤傲苍狼
黄勇博客
张善友博园

学习网站

MSDN，我告诉你(系统下载)
汇智学习网
廖雪峰的官方网站
牛客网
阮一峰的网络日志

最新评论

1. Re:阿里巴巴2016双11背后的技术(不一样的技术创新)
谢谢分享
--daconglee
2. Re:前端工程师理解
突然觉得做前端可真不容易呀~
--幸福的菜菜

Q(s,a)	a1	a2	…	a10
s1	Q(s1,a1)	Q(s1,a2)	…	Q(s1,a10)
s2	Q(s2,a1)	Q(s2,a2)	…	Q(s2,a10)
…	…	…	…	…
s64	Q(s64,a1)	Q(s64,a2)	…	Q(s64,a10)

根据Q-Learning公式进行权重更新。

$$Q_{t+1}(s_t,a_t) \leftarrow (1-\alpha)Q_t(s_t,a_t) + \alpha\left[r_t + \gamma \max_{a'} Q_t(s_{t+1},a')\right]$$

接下来，由于用户点击了某商品，他的状态发生了转移，就找到对应的状态继续进行epsilon-greedy的投放。再继续进行学习，直到收敛。

（2）DDPG方法

例如一个线性排序模型，f(x|w)=w^T x,x∈R^m，x是m维的特征向量，我们学习每个用户状态s的最优参数w，即π(s)→w^*。这种假设需要使用策略估计的方法。策略估计（Policy Approximation）方法是解决连续状态/动作空间问题的有效方法之一。其主要思想是用参数化的函数对策略进行表达，通过优化参数来完成策略的学习。通常，这种参数化的策略函数被称为Actor。假设我们一共调控m（m≥0）个维度的排序权重，对于任意状态s∈S，Actor对应的输出为：

$$\mu_\theta(s) = \left(\mu_\theta^1(s), \mu_\theta^2(s), \ldots, \mu_\theta^m(s)\right)$$

其中，θ 为Actor的参数，对于任意 i（1≤i≤m），μ_θ^i (s) 是关于状态的一个函数，代表第i维的排序权重分，其形式可根据实际情况而定，我们的方案采用深度神经网络作为Actor函数。这种方式在不同的状态之间可以通过神经网络来共享一些参数权重。

阅读排行榜

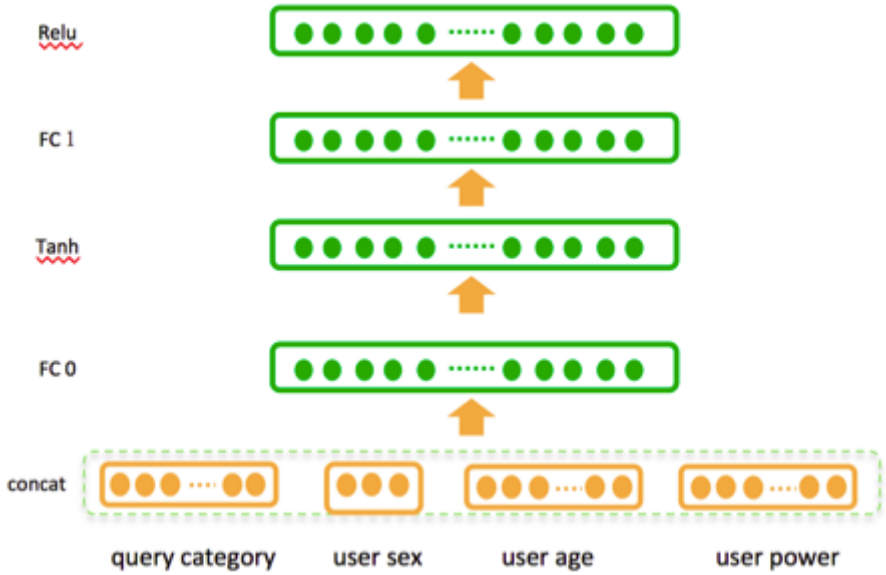
1. 初识Hadoop一，配置及启动服务(4373)
2. CentOS7安装Nginx并部署(3783)
3. 阿里巴巴2016双11背后的技术(不一样的技术创新)(3231)
4. SpringMVC处理请求流程(3196)
5. PL/SQL 中查询CLOB字段内容(1889)

评论排行榜

1. 前端工程师理解(1)
2. 阿里巴巴2016双11背后的技术(不一样的技术创新)(1)

推荐排行榜

1. java WebSocket Demo(2)
2. 阿里巴巴2016双11背后的技术(不一样的技术创新)(1)
3. CentOS7安装Nginx并部署(1)
4. SpringMVC处理请求流程(1)
5. jQuery插件开发教程(1)



强化学习的目标是最大化任意状态s上的长期累积奖赏，根据策略梯度定理， Actor函数的参数θ的更新公式可以写为：

$$\theta_{t+1} \leftarrow \theta_t + \alpha_\theta \nabla_\theta \mu_\theta(s) \nabla_\alpha Q^u(s,a)|_{a=\mu_\theta(s)}$$

其中，∇_θ μ_θ (s)为Actor神经网络在状态s上关于θ的梯度，Q^μ (s,a)为状态动作对（State-Action Pair）(s,a)的长期累积奖赏。因为s和a都是连续的数值，我们采用深度神经网络作为估计器对Q^μ (s,a)进行学习，具体的学习算法可参考深度Q学习算法DQN [1]。

1.4 奖赏塑形

我们最初采用的奖赏函数仅基于用户在每一个PV中的点击、成交行为反馈来构建。然而，在淘宝主搜这种大规模应用的场景中，我们较难在短时间内观察到不同的排序策略在点击和成交这样的宏观指标上的差别。因此，长期累积奖赏关于不同学习参数的梯度并无明显区别，导致学习算法收敛缓慢。因此，我们有必要在奖赏函数中引入更多的信息，增大不同动作的区分度。

在进行强化学习方案的同时，我们用Pointwise LTR进行了一些对比实验，发现Pointwise LTR这种直接在商品特征上进行学习的方式在求取策略梯度的时候，能够将不同排序策略更为显著地区分开。参照这个思路，我们将商

品的一些属性特征加入到奖赏函数的定义中，通过奖赏塑形（Reward Shaping）的方法[2, 3]丰富其包含的信息量。

奖赏塑形的思想是在原有的奖赏函数中引入一些先验的知识，加速强化学习算法的收敛。简单地，我们可以将“在状态s上选择动作a，并转移到状态s'”的奖赏值定义为：

$$R(s,a,s') = R_0(s,a,s') + \Phi(s)$$

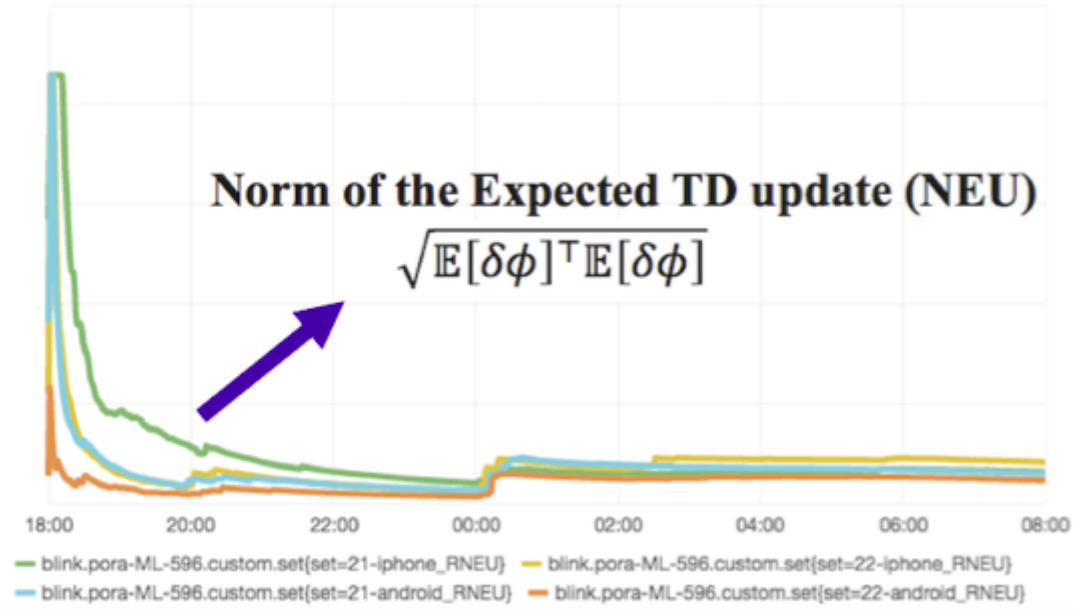
其中，R_0 (s,a,s')为原始定义的奖赏函数，Φ(s)为包含先验知识的函数，也被称为势函数（Potential Function）。我们可以把势函数Φ(s)理解学习过程中的子目标（Local Objective）。根据上面的讨论，我们把每个状态对应PV的商品信息纳入Reward的定义中，将势函数Φ(s)定义为：

$$\Phi(s) = \sum_{i=1}^{K} \mathbb{L}(i|\mu_\theta(s))$$

其中，K为状态s对应PV中商品的个数，i表示的第i个商品，μ_θ (s)为Agent在状态s执行的动作，L(i|μ_θ (s))表示排序策略为μ_θ时商品的点击（或成交）的似然（Likelihood）。因此，Φ(s)也就表示在状态s上执行动作μ_θ (s)时，PV中所有商品能够被点击（或购买）的似然概率之和。

1.5 实验效果

在双11期间，我们在无线搜索排序的21和22号桶对强化学习方案进行了测试。下图展示了我们的算法在学习的过程中的误差（RNEU）变化情况，截取的时间范围为11月10日18:00到11月11日8:00。



可以看到，从11月10日18:00启动开始，每个桶上的RNEU开始逐渐下降。到当天20:00之后，下降趋势变得比较缓和，说明学习算法在逐步往最优策略进行逼近。但过了11月11日0点之后，每个桶对应的RNEU指标都出现了陡然上升的情况，这是因为0点前后用户的行为发生了急剧变化，导致线上数据分布在0点以后与0点之前产生较大差别。相应地，学习算法获取到新的reward信号之后，也会做出适应性地调整。

2. 推荐算法研究与实践

2.1 背景介绍

双11主会场是一个很复杂的推荐场景。从推荐的业务形式上看，双11主会场分为三层：分别是楼层、坑位以及具体素材图的推荐。2016年的双11主会场在整体的组织形式上与去年的双11主会场类似，但具体业务的构成及组织有较大的不同。首先，可推荐的楼层多于十层，我们需从中挑选数层进行展示，并有可能根据时间段和业务的需求进行调整。因此，展现形式的多变对模型的日志特征学习造成了一定的干扰。其次，坑位的构成分为三种会场入口：第一行是行业会场，第二行对应店铺会场，第三行对应是标签会场。最后，在楼层以及坑位都确定之后，我们需要每个的坑位入口上选择具体的素材。2016年双11主会场的素材有两种不同的展现形式，分别是双

素材图以及单素材图。双素材图模式能提升用户的点击欲望，增强视觉感官冲击力，但也会对用户的真实点击行为数据造成一定程度的干扰或噪声，甚至对排序的模型产生比较大的偏置。

由于2016年双11首图宝贝素材总量在百万张且坑位数上百，我们会根据楼层的次序对参与打分的候选集进行配额，根据楼层的实时点击率分配楼层的打分量。在各类业务以及填坑逻辑及调控流量的限制下，推荐结果并不一定能按照原有的打分高低进行展示。因此，我们需要考虑打分宝贝数与工程实现上的平衡关系。由于主会场的QPS高达数万，一味地增大打分量是不可取的。为了解决这一问题，我们在初选的match召回方式上做了大量的努力，如提升用户的多重兴趣覆盖、增大有效的候选宝贝。

根据在2015双11的一些经验并结合2016年双11前期的系统压测情况，在2016年双11主会场我们采用了素材模型驱动的模式。从个性化推荐算法的角度来说，我们在2016年双11主会场尝试了多种新颖的排序模型，并做了严格的效果对比。具体的排序模型涉及LR、FTRL、GBDT+FTRL融合模型以及WIDE&DEEP模型，同时为了克服data drift的波动在日常的首图场景还尝试了Adaptive-Online-Learning的算法，以及尝试了强化学习的思路。在后面的章节，会从算法层面逐一阐释。

2.2 算法模型

2.2.1 GBDT+FTRL模型

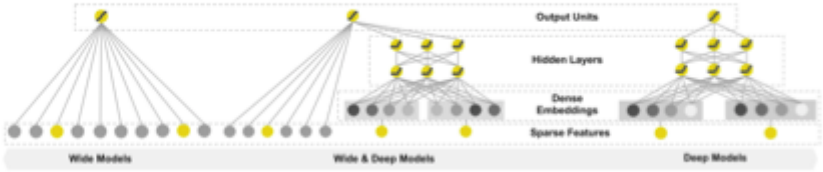
采用非线性模型学习intermediate feature，作为ID feature和cross feature的补充，最终输入到线性model来做CTR预估，最早是由Facebook提出的，思路大致如下：采用raw features（一般是统计类特征）训练出GBDT模型，获得的所有树的所有叶子节点就是它能够generate出来的特征空间，当每个样本点经过GBDT模型的每一个树时，会落到一个叶子节点，即产生了一个中间特征，所有这些中间特征会配合其他ID类特征以及人肉交叉的特征一起输入到LR模型来做CTR预估。显然，GBDT模型很擅长发掘有区分度的特征，而从根到叶子节点的每一条路径体现了特征组合。对比手工的离散化和特征交叉，模型显然更擅长挖掘出复杂模式，获得更好的效果。我们通过GBDT来做特征挖掘，并最终与FTRL模型融合的方案如下图：

输入到GBDT的特征非常关键，这些特征决定了最终产出的中间特征是否有效。我们有一套灵活的特征生成流程，可以方便做各种维度的特征提取以及交叉统计。GBDT+FTRL中主要用到的特征包含两部分：第一部分是用户/宝贝ID与对方泛化维度交叉统计的特征，包含各种基础行为的次数以及CTR等。

第二部分是来自于match阶段的一些连续类特征。推荐的match阶段负责粗选出一部分跟用户相关的content，该过程中会有多个模型分出现，例如做trigger selection的model分， content的最终match score等，这些分数来自于不同离线model，最终作为feature在online rank model中，能获得非常好的ensemble效果。

2.2.2 Wide & Deep Learning模型

借鉴Google今年在深度学习领域的论文《Wide & Deep Learning for Recommender Systems》中所提到的Wide & Deep Learning框架（以下简称为WDL），并将其结合基于搜索事业部自研的机器学习平台的在线学习技术，我们研发了一套适用于推荐业务的WDL模型算法。下文将会对这一技术进行详述。



WDL模型的原理框架如上图所示：它将深度神经网络(DNN)网络和逻辑回归(Logistic Regression)模型并置在同一个网络中，并且将离散型特征(Categorical Feature)和连续型特征(Continuous Feature)有机地结合在一起。WDL模型主要由wide侧和deep侧组成。Wide侧通过特征交叉来学习特征间的共现，而deep侧通过将具有泛化能力的离散型特征进行特征嵌入(embedding)，和连续型特征一起作为深度神经网络的输入（可以认为是一种特殊的深度神经网络，在网络的最后一层加入了大量的0/1节点），从理论上来说，我们可以把deep侧看作传统矩阵分解(matrix factorization)的一种泛化实现，值得注意的是特征嵌入的函数是和网络中其他参数通过梯度反向传播共同学习得到。模型的预测值采用如下公式进行计算：

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

其中，wide侧和deep侧合并在一起计算后验概率P(Y=1|x)；在误差反向传播(Backpropagation)的计算过程中时，我们对两个方向同时进行计算。

2.2.3 Adaptive-Online-Learning（自适应在线学习）

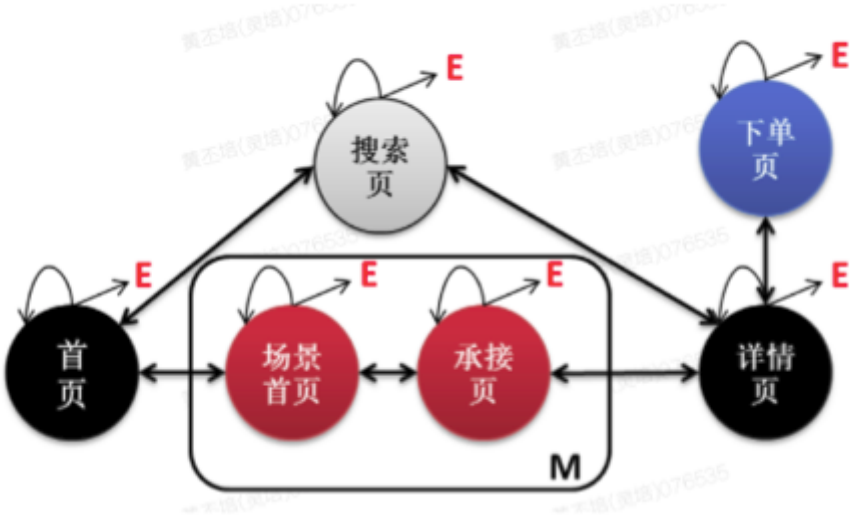
传统的在线学习模型没有一种机制很好的判断模型应该采用的多长时间的日志进行训练，目前业界的在线学习模型也都是通过经验值的方式来进行数据截断，自适应学习(adaptive learning)的最大优势就在于能够通过自我学习的方法适应业务的多变性。其实现原理在于保留下来每一个时刻开始到现在的数据学习到的模型，然后根据有效的评测指标，计算出各个模型的权重信息，并同时捕捉到数据分布快速变化波动的情况下的用户实时兴趣的细微差别，从而融合出一个最优的模型结果。

2.2.3 Adaptive-Online-Learning（自适应在线学习）

传统的在线学习模型没有一种机制很好的判断模型应该采用的多长时间的日志进行训练，目前业界的在线学习模型也都是通过经验值的方式来进行数据截断，自适应学习(adaptive learning)的最大优势就在于能够通过自我学习的方法适应业务的多变性。其实现原理在于保留下来每一个时刻开始到现在的数据学习到的模型，然后根据有效的评测指标，计算出各个模型的权重信息，并同时捕捉到数据分布快速变化波动的情况下的用户实时兴趣的细微差别，从而融合出一个最优的模型结果。

2.2.4 Reinforcement Learning（强化学习）

相比对每个推荐场景单独进行个性化推荐的策略，基于强化学习框架（Reinforcement Learning）的推荐系统根据全链路的数据进行整合，同时响应多个异构场景的推荐请求。下图中我们对手机淘宝（天猫）客户端的数据/流量通路进行抽象：每个圆圈代表一个独立的手淘场景，E代表用户在该场景随时离开，箭头代表流量可能地流动方向。



基于以上的数据通路图，我们可以很自然地将全链路多场景的推荐任务理解为一个连续的决策问题：作为一个智能决策者（agent），推荐系统需要持续不断地决定应该为用户推荐怎样的内容（比如，商品、店铺、品牌以及活动）。强化学习正是一种对智能决策者进行建模的最佳方式：通过对智能决策者短期状态的变化进行递归式建模，最终引导其渐进式地优化长期目标。

手淘上的推荐场景相当丰富，最具代表性的是一个页面以列表的形式同时推荐多个商品的场景。为了便于读者理解，我们首先介绍单个商品的推荐场景，之后再过渡到多商品的推荐场景。在单商品的推荐场景，a对应的是单个商品。我们的目标是学习在状态s下采取动作a所能获得的累积奖励（的期望值）。我们用Q(s,a)来表示这一期

望值。在这种情况下，我们只需要选择一种函数映射关系（如线性函数或神经网络）将s和a所代表的向量映射到标量上对目标函数Q(s,a)进行拟合。

$$Q(s,a) = \mathbb{E}[R|s,a] \qquad (1)$$

我们把这一定义延伸到典型的多商品推荐场景。由于文章长度有限，我们下面介绍一种最简单的思路，即假设用户是否会点击单商品的决策是独立的。也就是说，假设用户如果喜欢商品A，用户不会因为在同一推荐列表中见到了他更喜欢的商品B而放弃点击商品A。在这一假设下，我们对展示每个商品所获得的累积奖励的计算也是独立的。通过一系列的推导，我们可以得到一个对状态s下商品i能得到的分数f(s,i)的递归定义。

$$f(s,i) = I(s_i)[r_i + \gamma \sum_{j \in a_i} f(s_i,j)] \qquad (7)$$

通过等式(7)，我们可以迭代计算对无偏估计值进行求解。实际情况中用户必然会因为推荐商品的组合问题产生更复杂的行为，这样一来必然导致累积奖励独立计算的假设不成立。但以此为本，我们可以推导出基于更复杂假设下的计算累积奖励估计量的递归公式。

参考文献

[1] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M.A., Playing atari with deep reinforcement learning. CoRR abs/1312.5602, 2013.

[2] A. Y. Ng, D. Harada, and S. J. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In Proceedings of the 16th International Conference on Machine Learning, pages 278–287, 1999

[3] E. Wiewiora. Potential-based shaping and Q-value initialization are equivalent. Journal of Artificial Intelligence Research, 19(1):205–208, 2003

分类: OtherBeautifulBlogs

好文要顶 关注我 收藏该文

jiapeng
关注 - 17
粉丝 - 14
+加关注

0 0

« 上一篇：第六章 大数据，6.3 突破传统，4k大屏的沉浸式体验(作者： 彦川、小丛)
» 下一篇：第七章 人工智能，7.2 颠覆传统的电商智能助理-阿里小蜜技术揭秘(作者：海青)

posted @ 2016-12-30 14:29 jiapeng 阅读(149) 评论(0) 编辑 收藏

刷新评论 刷新页面 返回顶部

注册用户登录后才能发表评论，请 登录 或 注册，访问网站首页。

【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库
【免费】从零开始学编程，开发者专属实验平台免费实践！
【推荐】现在注册又拍云，首月可享 200G CDN流量，还可免费申请 SSL 证书
【推荐】阿里云“全民云计算”优惠升级

最新IT新闻:

· Adobe已宣布Flash结束，开源它是最好的方案吗？
· 微信“电子发票”三大升级：实在太给力了
· 学习下蒂姆·库克给予的12条商业建议
· AI个性化学习系统会对传统教育产生的14种影响
· 农行、招行已大规模应用“刷脸取款”，中小银行为何还未上道儿？
» 更多新闻...

最新知识库文章:

· 小printf的故事：什么是真正的程序员？
· 程序员的工作、学习与绩效
· 软件开发为什么很难
· 唱吧DevOps的落地，微服务CI/CD的范本技术解读
· 程序员，如何从平庸走向理想？
» 更多知识库文章...

Copyright ©2017 jiapeng