

美团推荐调研报告

1 小组成员

金宇航 181250062
张许妍 181250187
吴昊宇 181250154
冯泊涓 181250030
周瑾瑜 181250205

2 美团推荐场景概述

美团是国内领先的生活服务电子商务平台，以“帮助大家吃得更好，生活更好”作为企业文化使命，业务涵盖餐饮、外卖、电影、休闲娱乐、旅游出行等多个方面，聚集了大量用户，也获取了海量的用户行为信息——这为美团推荐系统的应用和优化提供了不可或缺的条件。

本小组着眼于美团的智能数据应用系统——美团推荐，针对其如何利用用户数据理解用户意图、如何“投其所好”进行个性化推荐，进行了资料调查和研究。

本调研报告包含以下内容：

- 美团推荐应用概述
- 美团推荐框架概述
- 数据准备
- 推荐候选集触发
- 候选集重排序
- 评估以及扩展

2.1 美团推荐应用概述

2.1.1 猜你喜欢



- 美团最重要的推荐产品
- 包罗万象：美食、电影、娱乐、团购.....
- 2年来交易额占比从0.7%提高到7~8%

2.1.2 首页频道推荐



- 固定频道+个性化推荐频道
- 为美团带来40%以上交易额

2.1.3 今日推荐个性化推送

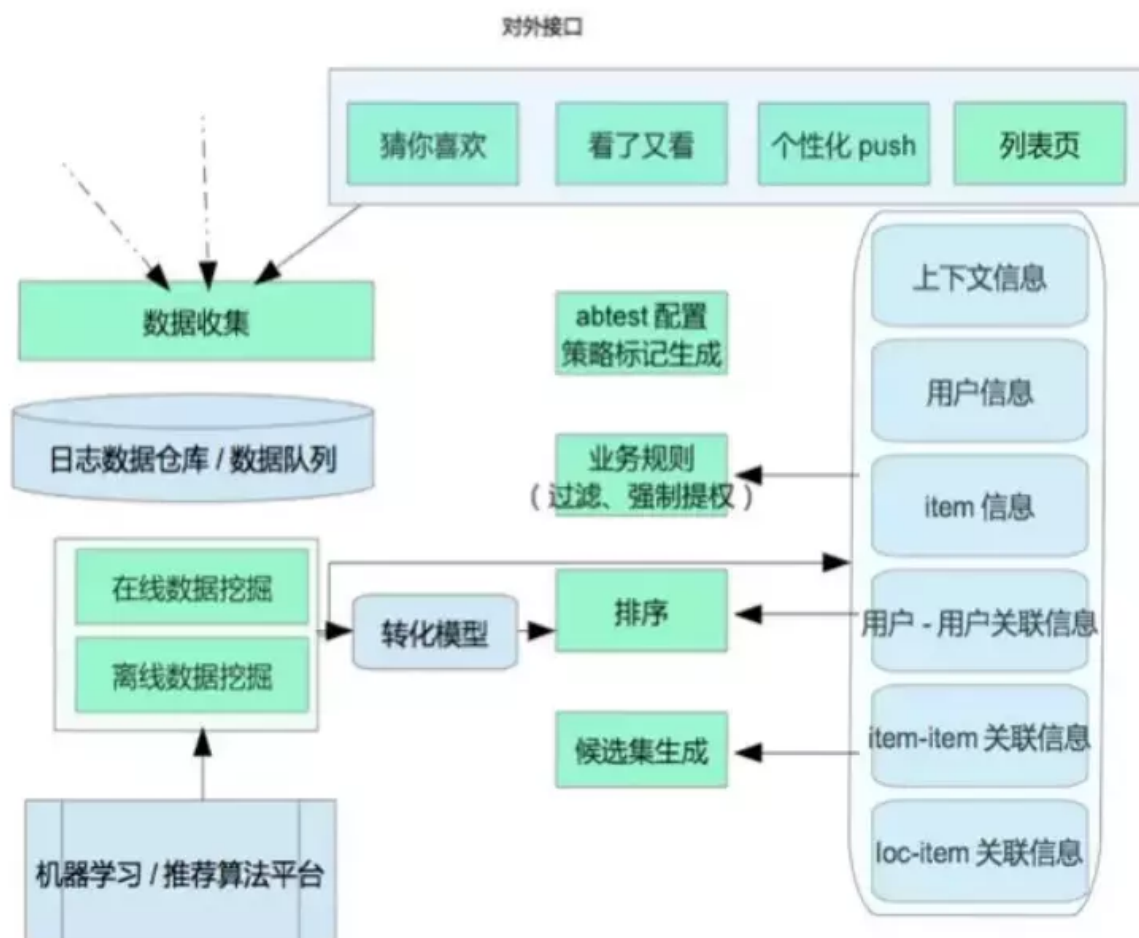
- 通知栏推送
- 在用户打开APP前狙击用户意图，促使其进入APP，提高DAU

2.1.4 品类列表的个性化排序

- 智能排序
- 个性化优化的重要位置

2.2 美团推荐框架概述

推荐系统整体示意图：



从框架角度看，推荐系统基本分为数据层、候选集触发层、融合过滤层、排序层。

2.2.1 数据层

包括数据生成、数据存储。

数据生成：利用各种数据处理工具对原始日志进行数据清洗，生成格式化数据。

数据存储：将格式化数据存入不同类型的存储系统中，供下游的算法和模型使用。

2.2.2 候选集触发层

从用户的历史行为、实时行为、地理位置等角度，利用各种触发策略产生推荐的候选集。

2.2.3 融合过滤层

融合：对触发层产生的不同候选集进行融合，提高推荐策略的覆盖度和精度。

过滤：承担一定的过滤职责，从产品、运营的角度确定一些人工规则，过滤掉部分非法item。

2.2.4 排序层

利用机器学习模型，对触发层筛选出来的候选集进行重排序。

3 数据准备

3.1 数据获取

美团的推荐系统对外提供了一些服务接口如猜你喜欢、看了又看等等，在响应请求的同时，会打印一些必要的日志，记录这次请求的一些必要的上下文信息以及用户及item相关的特征信息，以便生成训练数据。这些日志通过flume传输到HDFS上面。除了推荐系统以外的美团App其他后台服务，也会把各自的日志传递给HDFS，以方便后续进行数据挖掘。借助Hadoop、Hive、Spark等平台以及美团自己实现的一些机器学习/推荐通用算法，对原始日志进行处理，从而得到需要的各种数据及模型：包括用户的profile信息，用户之间的相似度，item之间的相似度。

3.2 数据类别和应用

数据乃算法、模型之本。美团作为一个交易平台，同时具有快速增长的用户量，因此产生了海量丰富的用户行为数据。不同类型的数据的价值和反映的用户意图的强弱有所不同。美团使用的数据主要可以分为以下4类。

行为类别	行为详情
主动行为数据	搜索、筛选、点击、收藏、下单、支付、评分
UGC	文本评价、上传图片
负反馈数据	左滑删除、取消收藏、取消订单、退款、差评、低评
用户画像	用户人口属性、美团DNA、品类偏好、消费水平、工作地与居住地

1. 用户主动行为数据记录了用户在美团平台上不同的环节的各种行为，这些行为一方面用于候选集触发算法（在下一部分介绍）中的离线计算（主要是浏览、下单），另外一方面，这些行为代表的意图的强弱不同，因此在训练重排序模型时可以针对不同的行为设定不同的回归目标值，以更细地刻画用户的行为强弱程度。此外，用户对deal的这些行为还可以作为重排序模型的交叉特征，用于模型的离线训练和在线预测。
2. 负反馈数据反映了当前的结果可能在某些方面不能满足用户的需求，因此在后续的候选集触发过程中需要考虑对特定的因素进行过滤或者降权，降低负面因素再次出现的几率，提高用户体验；同时在重排序的模型训练中，负反馈数据可以作为不可多得的负例参与模型训练，这些负例要比那些展示后未点击、未下单的样本显著的多。
3. 用户画像是刻画用户属性的基础数据，其中有些是直接获取的原始数据，有些是经过挖掘的二次加工数据，这些属性一方面可以用于候选集触发过程中对deal进行加权或降权，另外一方面可以作为重排序模型中的用户维度特征。
4. 通过对UGC数据的挖掘可以提取出一些关键词，然后使用这些关键词给deal打标签，用于deal的个性化展示。

4 推荐候选集触发

4.1 策略触发

在获取到足够的数据之后，之后的关键就是如何使这些数据发挥作用。本部分将从美团系统中的不同应用场景出发，探究在这些场景下使用到的相关算法和模型。

4.1.1 协同过滤

4.1.1.1 应用场景

协同过滤是推荐算法中的经典算法之一，适用于用户行为较为充足的场景。美团上有大量入驻商铺，用户可以对这些商铺和商铺中的商品进行评价和打分，与此同时这些评价记录和用户的收藏等用户数据也会被作为生成用户画像的重要数据。这些数据支持着美团运作user-based和item-based相结合的协同过滤推荐系统。

	群体/个体	计算代价	适用场景	冷启动	可解释性	实时性
user-based	更依赖与当前用户相近的用户群体的社会化行为	适用于用户数较少的场合	时效性强，用户个性化兴趣不太显著的场合	新加入的物品能很快进入推荐列表	弱	用户新的行为不一定导致推荐结果的变化
item-based	更侧重用户自身的个体行为	适用于物品数较少的场合	长尾物品丰富，用户个性化需求强烈的场合	新加入的用户能很快得到推荐	强	用户新的行为一定导致推荐结果的变化

4.1.1.2 逻辑探究

- 数据清洗：清除作弊、刷单、代购等噪声数据。这些数据的存在会严重影响算法的效果，因此要在第一步的数据清洗中就将这些数据剔除。
- 合理选取训练数据：选取的训练数据的时间窗口不宜过长，当然也不能过短。具体的窗口期数值需要经过多次的实验来确定。同时可以考虑引入时间衰减，因为近期的用户行为更能反映用户接下来的行为动作。
- 尝试不同的相似度计算方法：在实践中，美团采用了一种称作loglikelihood ratio的相似度计算方法。在mahout中，loglikelihood ratio也作为一种相似度计算方法被采用。
 - loglikelihood ratio相似度计算公式：

	与item j发生交互	没有与item j发生交互
与item i发生交互	K11	K12
没有与item i发生交互	K21	K22

那么item i与item j的相似度如下

$$\begin{aligned}
S &= 2 \times (H(\text{colSum}(K)) + H(\text{rowSum}(K)) - H(K)) \\
&= 2 \times \left(\frac{K_{11}}{N} \log\left(\frac{K_{11}}{N}\right) + \frac{K_{12}}{N} \log\left(\frac{K_{12}}{N}\right) + \frac{K_{21}}{N} \log\left(\frac{K_{21}}{N}\right) + \frac{K_{22}}{N} \log\left(\frac{K_{22}}{N}\right) \right. \\
&\quad \left. - \left(\frac{K_{11} + K_{12}}{N} \log\left(\frac{K_{11} + K_{12}}{N}\right) + \frac{K_{21} + K_{22}}{N} \log\left(\frac{K_{21} + K_{22}}{N}\right) \right) \right. \\
&\quad \left. - \left(\frac{K_{11} + K_{21}}{N} \log\left(\frac{K_{11} + K_{21}}{N}\right) + \frac{K_{12} + K_{22}}{N} \log\left(\frac{K_{12} + K_{22}}{N}\right) \right) \right)
\end{aligned}$$

- 推荐场景的loglikelihood ratio的计算逻辑

假设在给出的数据集中一共有M个用户，这M个用户与所有的物品发生了N次交互行为，也就是N次的点击记录。对于给定的物品中的每一个物品i，其出现在用户的交互行为中的次数为Fi。物品j和物品i共同出现在某一个用户行为中的交互次数为Fij。基于上述给出的数据符号描述，对于物品i和物品j构建物品的共现次数关系矩阵如下：

	与物品j发生交互	没有与物品j发生交互
与物品i发生交互	Fij 或 Fji	Fj - Fji
没有与物品i发生交互	Fi - Fij	N - Fi - Fj + Fij

4.1.2 location-based

4.1.2.1 应用场景

数据显示，美团的冷启动用户（单月交易数目少）占比高，并且移动终端用户占比高。这意味着大部分美团用户的使用场景，很可能是和朋友一起逛街，逛完了就顺手打开美团看看周边有什么饭店，再决定去哪里吃饭。又或者是看完一场电影，打开美团看看周边有什么酒店。因此地理因素是影响美团用户消费的一个重要因素。

4.1.2.2 逻辑探究

- 区域粒度划分：挖掘地理信息和用户行为，从而生成覆盖范围大约十几平方公里的实时商圈。当用户的实时位置并不总是能够获取到，或者用户的实时商圈，可推荐的item数量太少时，美团将会扩大地理信息的挖掘量，从用户周末/平时常去商圈，用户工作地/居住地附近商圈等商圈信息获取信息进行推荐。
- 时间粒度划分：不同时间的用户需求是不同的，粒度过大的时间划分会导致推荐效果不准确，无法很好吻合用户需求，而粒度过小的时间划分会导致数据量太过于稀疏。
- 推荐列表生成：当新的线上用户请求到达时，根据用户的几个地理位置对相应地理位置的区域消费热单和区域购买热单进行加权，最终得到一个推荐列表。

4.1.3 query-based

4.1.3.1 应用场景

搜索是一种强用户意图，在大多数情况下比较明确地反映了用户的意愿。虽然大部分时候这些搜索没有促成最终的转换，但这并不代表它们毫无用处。美团将这一部分信息收集起来，以生成成面向单个用户的推荐。

4.1.3.2 逻辑探究

- 对用户过去一段时间的搜索无转换行为进行挖掘，计算每一个用户对不同query的权重。
- 计算每个query下不同deal的权重。
- 当用户再次请求query时，根据用户对不同query的权重及query下不同deal的权重进行加权，取出权重最大的TopN进行推荐。

4.1.4 graph-based

4.1.4.1 应用场景

在上文提到的协同过滤算法中，user之间或者deal之间的图距离是两跳，但对于更远距离的关系则不能考虑在内，这也就意味着推荐结果很可能缺乏多样性。倘若将user与deal的关系视作一个二部图，那么相似性就可以通过数个user-deal传递到距离更远的user/deal。这样就突破了协同算法的限制。

4.1.4.2 逻辑探究

美团采用了simrank算法，其基本思想是如果两个实体与另外的相似实体有相关关系，那它们也是相似的，即相似性是可以传播的。

- 基本公式：

$$s(a, b) = \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s(I_i(a), I_j(b)) \quad (1)$$

其中 $s(a, b)$ 是节点 a 和 b 的相似度，当 $a=b$ 时， $s(a, b)=1$ 。 $I_i(a)$ 表示 a 的第 i 个in-neighbor。当 $I(a)$ 或者 $I(b)$ 为空集时， $s(a, b)=0$ 。参数 C 是一个阻尼系数，它的含义可以这么理解：假如 $I(a)=I(b)=\{A\}$ ，按照(1)式计算出 $\text{sim}(a, b)=C \cdot \text{sim}(A, A)=C$ ，所以 $C \in (0, 1)$ 。在实践中 C 通常取0.6-0.8。

4.1.5 其他替补策略

4.1.5.1 应用场景

虽然美团拥有一系列基于用户历史行为的候选集触发算法，但是对于部分新用户和历史行为不太丰富的用户，这些触发算法的候选集太小，因此需要使用一些替补策略进行填充。这些策略很有可能不是基于某种大数据算法，而是简单进行筛选分类的结果。

4.1.5.2 逻辑探究

- 热销单：在一定时间内销量最多的item，可以考虑时间衰减的影响等。
- 好评单：用户产生的评价中，评分较高的item。
- 城市单：满足基本的限定条件，在用户的请求城市内的。

5 候选集重排序

对于不同算法触发出来的候选集，只是根据算法的历史效果决定算法产生的item的位置有些不合理，同时，在每个算法的内部，不同item的排序也只是简单的由一个或者几个因素来决定，只适用于第一步的初选流程。最终的候选集排序结果需要借助机器学习，使用相关的排序模型，综合各个方面的因素来确定。

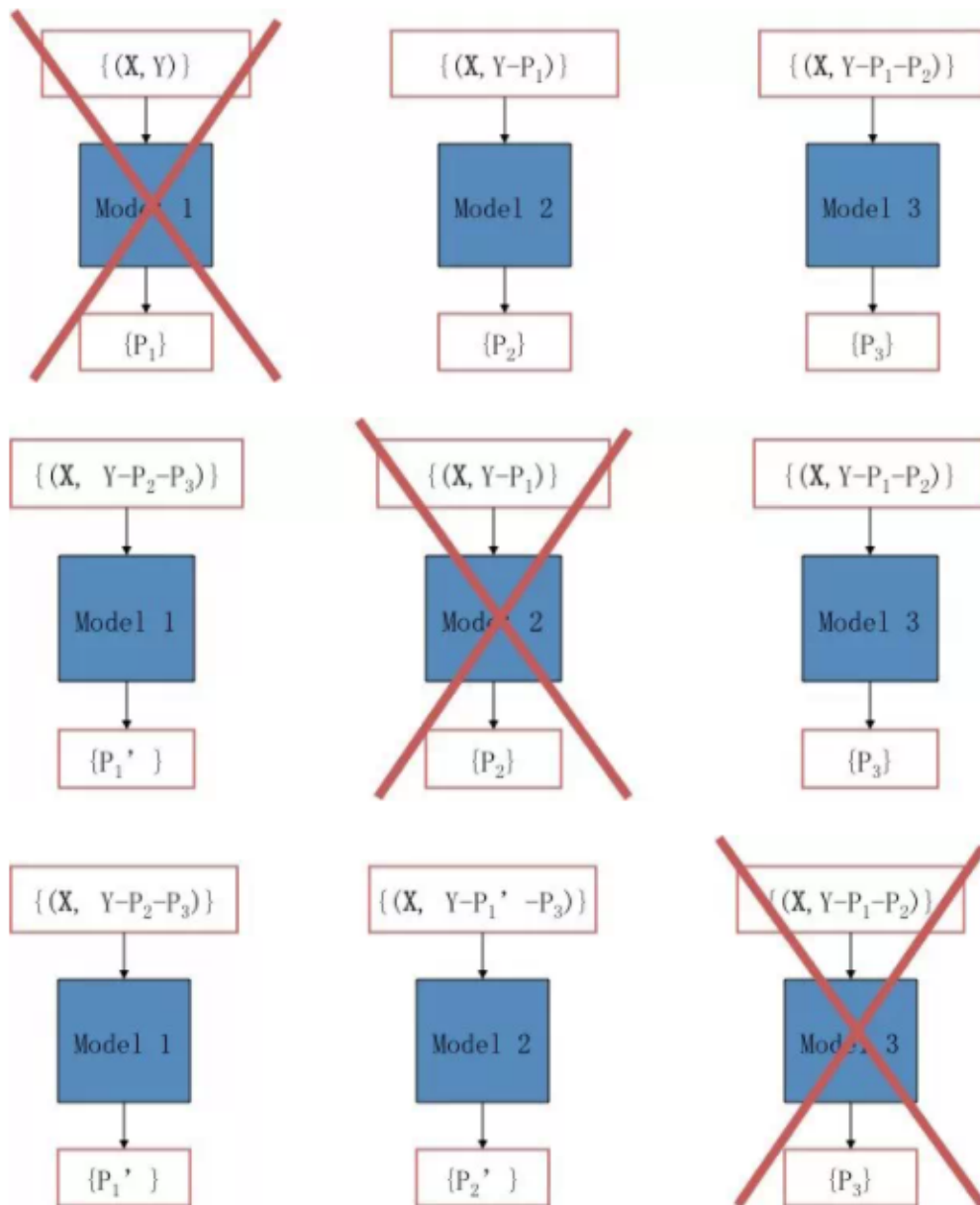
5.1 模型

在实践中，线性模型模型和非线性模型都有应用。非线性模型能够更好地捕捉特征中的非线性关系，但是训练和预测的代价较高，且周期较长；而线性模型较为简单，效率较高，周期短，但需要凭借领域知识和经验人工对特征事先做一些处理。

5.1.1 非线性模型

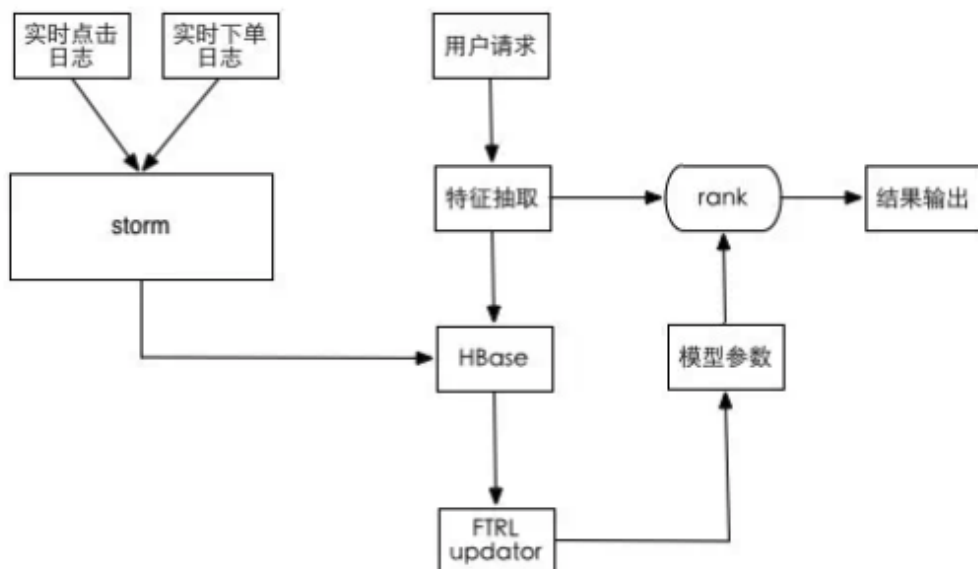
主要采用非线性树模型Additive Groves (简称AG), AG是一个加性模型, 由多个Grove组成, 由bagging方法得出最后的预测结果, 减少过拟合的影响。

每一个Grove有多棵树组成, 在训练时每棵树的拟合目标为真实值与其他树预测结果之和之间的残差。当达到给定数目的树时, 重新训练的树会逐棵替代以前的树。经过多次迭代后, 达到收敛。



5.1.2 线性模型

线性模型中逻辑回归应用较多, 同过引入online learning, 接入实时数据流, 使用FTRL对模型进行在线更新。



5.2 样本采样

根据提高item的下单转化效果的最终目标，需要同时使用点击行为和下单行为，并做不同的处理。美团主要采用两种方式，一是提高下单样本的采样比例，二是提高下单行为的标记值。另外，美团对负例也较为关注，如排名较高却被用户自动忽略或者删除的item。

5.3 去除position bias

由于item在展示列表中的位置会在比较大的程度上影响用户对其的点击和下单概率，因此需要在计算item的点击和下单时，综合考量item被展示的位置。美团在两个地方进行了对position bias的处理。

- 计算item的历史点击（ctr_o）和下单（cvr_o）时，首先计算出每个位置的历史平均点击（ctr_p）和下单（cvr_p），根据item展示的位置，计算为 $\text{ctr}_o / \text{ctr}_p$ 和 $\text{cvr}_o / \text{cvr}_p$
- 产生训练样本时，把展示位置作为特征放在样本里，在使用模型的时候，把展示位置统一设为0。

5.4 特征工程

特征工程是排序模型最重要的工作，排序带来的效果提升大部分是由特征工程带来的。特征工程的主要工作是不停地去解除和理解业务数据，试图从中挖掘出和用户转化相关的特征。

美团使用的主要特征包括：

- 上下文特征：时间，商圈，天气，温度等
- item特征：价格，销量，用户评分等。
- 用户特征：用户的属性特征，如年龄，性别，品类偏好，价格偏好等

5.5 策略效果评估

美团进行策略效果对比使用的方法是abtest，优点在于能够对多个策略的效果差异给出定量的评估，但周期较长。因此，美团引入interleaving效果评估方式作为补充，所需流量较小，灵敏度较高，但只能给出定性结论。

Interleaving的基本思想是把两个策略的结果混合在一起，通过停机分析用户选择哪个策略的概率更大。具体列表混合的实现方式有多种，以较为简单的Balanced方式为例。

Rank	Input Ranking		Interleaved Rankings					
	A	B	Balanced		Team-Draft			
			A first	B first	AAA	BAA	ABA	...
1	a	b	a	b	a^A	b^B	a^A	
2	b	e	b	a	b^B	a^A	b^B	
3	c	a	e	e	c^A	c^A	e^B	
4	d	f	c	c	e^B	e^B	c^A	
5	g	g	d	f	d^A	d^A	d^A	
6	h	h	f	d	f^B	f^B	f^B	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

如上图所示，Input Ranking为A，B策略的原始列表，Balanced为合并后的列表，合并方式为，如果是A first，则以此取A列表的a，B列表的b，A列表的b（重复则继续顺延），B列表的e.....
采用这种列表混合方式的效果评估方式如下：

$$\Delta_{AB} = \frac{wins(A) + \frac{1}{2}ties(A, B)}{wins(A) + wins(B) + ties(A, B)} - 0.5$$

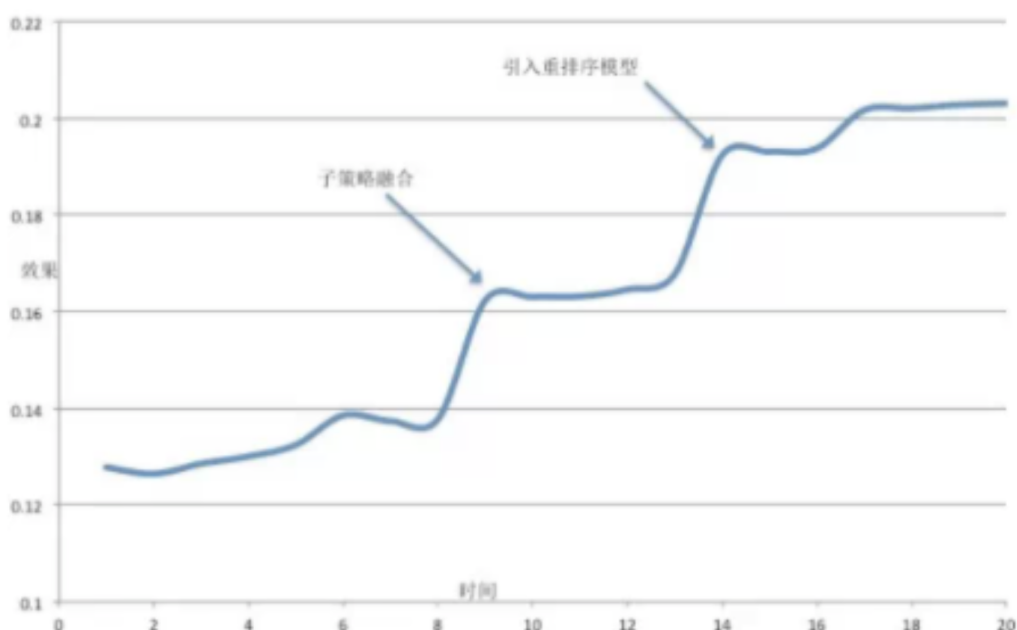
其中用户点击的item在A列表排序靠前，则wins（A）加一，B靠前则wins（B）加一，否则ties（A，B）加一， Δ_{AB} 为正则表示A策略较优。

6 评估以及扩展

6.1 评估

以数据为基础，用算法去雕琢，只有将二者有机结合，才会带来效果的提升。对美团而言，以下两个节点是他们认为的优化过程中的里程碑：

- 将候选集进行融合：提高了推荐的覆盖度、多样性和精度
- 引入重排序模型：解决了候选集增加以后deal之间排列顺序的问题



6.2 扩展：深度学习在美团推荐平台排序中的应用

6.2.1 背景

到目前为止，点评推荐排序系统尝试了多种线性、非线性、混合模型等机器学习方法，如逻辑回归、GBDT、GBDT+LR等。通过线上实验发现，相较于线性模型，传统的非线性模型如GBDT，并不一定能在线上AB测试环节对CTR预估有比较明显的提高。而线性模型如逻辑回归，因为自身非线性表现能力比较弱，无法对真实生活中的非线性场景进行区分，会经常对历史数据中出现过的数据过度记忆。

但这种推荐并没有结合当前场景给用户推荐出一些有新颖性的Item。为了解决这个问题，就需要考虑更多、更复杂的特征，比如组合特征来替代简单的“距离”特征。怎么去定义、组合特征，这个过程成本很高，并且更多地依赖于人工经验。

而深度神经网络，可以通过低维密集的特征，学习到以前没出现过的一些Item和特征之间的关系，并且相比于线性模型大幅降低了对于特征工程的需求，从而吸引美团进行探索研究。

在实际的运用当中，美团根据Google在2016年提出的Wide & Deep Learning模型，并结合自身业务的需求与特点，将线性模型组件和深度神经网络进行融合，形成了在一个模型中实现记忆和泛化的宽深度学习框架。在接下来的部分中，将会讨论如何进行样本筛选、特征处理、深度学习算法实现等。

6.2.2 样本的筛选

数据及特征，是整个机器学习中最重要两个环节，因为其本身就决定了整个模型的上限。点评推荐由于其自身多业务（包含外卖、商户、团购、酒旅等）、多场景（用户到店、用户在家、异地请求等）的特色，导致美团的样本集相比于其他产品更多元化。美团的目标是预测用户的点击行为。有点击的为正样本，无点击的为负样本，同时，在训练时对于购买过的样本进行一定程度的加权。而且，为了防止过拟合/欠拟合，将正负样本的比例控制在10%。最后，还要对训练样本进行清洗，去除掉Noise样本(特征值近似或相同的情况下，分别对应正负两种样本)。

同时，推荐业务作为整个App首页核心模块，对于新颖性以及多样性的需求是很高的。在点评推荐系统的实现中，首先要确定应用场景的数据，美团的数据可以分为以下几类：

- **用户画像**：性别、常驻地、价格偏好、Item偏好等。
- **Item画像**：包含了商户、外卖、团单等多种Item。其中商户特征包括：商户价格、商户好评数、商户地理位置等。外卖特征包括：外卖平均价格、外卖配送时间、外卖销量等。团单特征包括：团单适用人数、团单访购率等。
- **场景画像**：用户当前所在地、时间、定位附近商圈、基于用户的上下文场景信息等。

6.2.3 深度学习中的特征处理

深度学习能自动对输入的低阶特征进行组合、变换，得到高阶特征的特性，因此美团希望借助于深度学习来节约特征工程中的巨大投入，更多地让点击率预估模型和各辅助模型自动完成特征构造和特征选择的工作，并始终和业务目标保持一致。下面是一些美团在深度学习中用到的特征处理方式：

6.2.3.1 组合特征

对于特征的处理，美团将很多组合特征引入到模型训练中。因为不同特征之间的组合是非常有效的，并有很好的可解释性，比如将“商户是否在用户常驻地”、“用户是否在常驻地”以及“商户与用户当前距离”进行组合，再将数据进行离散化，通过组合特征，美团可以很好的抓住离散特征中的内在联系，为线性模型增加更多的非线性表述。组合特征的定义为：

$$\phi_k(x) = \prod x_i^{c_{ki}}, c_{ki} \in (0, 1)$$

6.2.3.2 归一化

归一化是依照特征矩阵的行处理数据，其目的在于样本向量在点乘运算或其他核函数计算相似性时，拥有统一的标准，也就是说都转化为“单位向量”。在实际工程中，美团运用了两种归一化方法：

Min-Max:

$$x' = \frac{x - \min}{\max - \min}$$

\min 是这个特征的最小值， \max 是这个特征的最大值。

Cumulative Distribution Function (CDF)

CDF也称为累积分布函数，数学意义是表示随机变量小于或等于其某一个取值 x 的概率。其公式为：

$$x' = \int_{-\infty}^x f(x) dx$$

在美团的线下实验中，连续特征在经过CDF的处理后，相比于Min-Max，CDF的线下AUC（AreaUnderRoc 即 Roc 曲线与坐标轴形成的面积）提高不足0.1%。他们猜想是因为有些连续特征并不满足在（0，1）上均匀分布的随机函数，CDF在这种情况下，不如Min-Max来的直观有效，所以美团在线上采用了Min-Max方法。

6.2.3.3 快速聚合

为了让模型更快的聚合，并且赋予网络更好的表现形式，美团对原始的每一个连续特征设置了它的super-liner和sub-liner，即对于每个特征 x ，衍生出2个子特征：

$$x^2$$
$$\sqrt{x}$$

6.2.4 优化器 (Optimizer) 的选择

在深度学习中，选择合适的优化器不仅会加速整个神经网络训练过程，并且会避免在训练的过程中困到鞍点。

6.2.4.1 Stochastic Gradient Descent (SGD)

SGD 是一种常见的优化方法，即每次迭代计算Mini-Batch的梯度，然后对参数进行更新。其公式为：

$$\nu_t = \mu \nabla_{\theta} J(\theta)$$

缺点是对于损失方程有比较严重的振荡，并且容易收敛到局部最小值。

6.2.4.2 Momentum

为了克服SGD振荡比较严重的问题，Momentum将物理中的动量概念引入到SGD当中，通过积累之前的动量来替代梯度。即：

$$\nu_t = \gamma \nu_{t-1} + \mu \nabla_{\theta} J(\theta)$$

$$\theta = \theta - \nu_t$$

相较于SGD，Momentum就相当于在从山坡上不停的向下走，当没有阻力的话，它的动量会越来越大，但是如果遇到了阻力，速度就会变小。也就是说，在训练的时候，在梯度方向不变的维度上，训练速度变快，梯度方向有所改变的维度上，更新速度变慢，这样就可以加快收敛并减小振荡。

6.2.4.3 Adagrad

相较于SGD，Adagrad相当于对学习率多加了一个约束，即：

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\mu}{\sqrt{\sum g_{t,i} + \epsilon}}$$

Adagrad的优点是，在训练初期，由于 g_t 较小，所以约束项能够加速训练。而在后期，随着 g_t 的变大，会导致分母不断变大，最终训练提前结束。

6.2.4.4 Adam

Adam是一个结合了Momentum与Adagrad的产物，它既考虑到了利用动量项来加速训练过程，又考虑到对于学习率的约束。利用梯度的一阶矩估计和二阶矩估计动态调整每个参数的学习率。Adam的优点主要在于经过偏置校正后，每一次迭代学习率都有个确定范围，使得参数比较平稳。其公式为：

$$\theta_{t+1} = \theta_t - \frac{\mu}{\sqrt{v_t^1} + \epsilon} m_t^1$$

其中：

$$m_t^1 = \frac{m_t}{1 - \beta_1^t}$$

$$v_t^1 = \frac{v_t}{1 - \beta_2^t}$$

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

通过实践证明，Adam结合了Adagrad善于处理稀疏梯度和Momentum善于处理非平稳目标的优点，相较于其他几种优化器效果更好。同时，美团也注意到很多论文中都会引用SGD，Adagrad作为优化函数。但相较于其他方法，在实践中，SGD需要更多的训练时间以及可能会被困到鞍点的缺点，都制约了它在很多真实数据上的表现。

6.2.5 损失函数的选择

深度学习同样有许多损失函数可供选择，如平方差函数（Mean Squared Error），绝对平方差函数（Mean Absolute Error），交叉熵函数（Cross Entropy）等。而在理论与实践，美团发现Cross Entropy相比于在线性模型中表现比较好的平方差函数有着比较明显的优势。其主要原因是在深度学习通过反向传递更新 W 和 b 的同时，激活函数Sigmoid的导数在取大部分值时会落入左、右两个饱和区间，造成参数的更新非常缓慢。具体的推导公式如下：

一般的MSE被定义为：

$$C = \frac{1}{2}(a - y)^2$$

其中 y 是我们期望的输出， a 为神经元的实际输出 $a = \sigma(Wx + b)$ 。由于深度学习反向传递的机制，权值 W 与偏移量 b 的修正公式被定义为：

$$\frac{\partial C}{\partial W} = (a - y)\sigma'(a)x^T$$

$$\frac{\partial C}{\partial b} = (a - y)\sigma'(a)$$

因为Sigmoid函数的性质，导致 $\sigma'(z)$ 在 z 取大部分值时会造成饱和现象。

Cross Entropy的公式为：

$$H(y, a) = - \sum y_i \log(a_i)$$

如果有多个样本，则整个样本集的平均交叉熵为：

$$H(y, a) = -\frac{1}{n} \sum \sum y_{i,n} \log(a_{i,n})$$

其中 n 表示样本编号， i 表示类别编号。如果用于Logistic分类，则上式可以简化成：

$$H(y, a) = -\frac{1}{n} \sum y \log(a) + (1 - y) \log(1 - a)$$

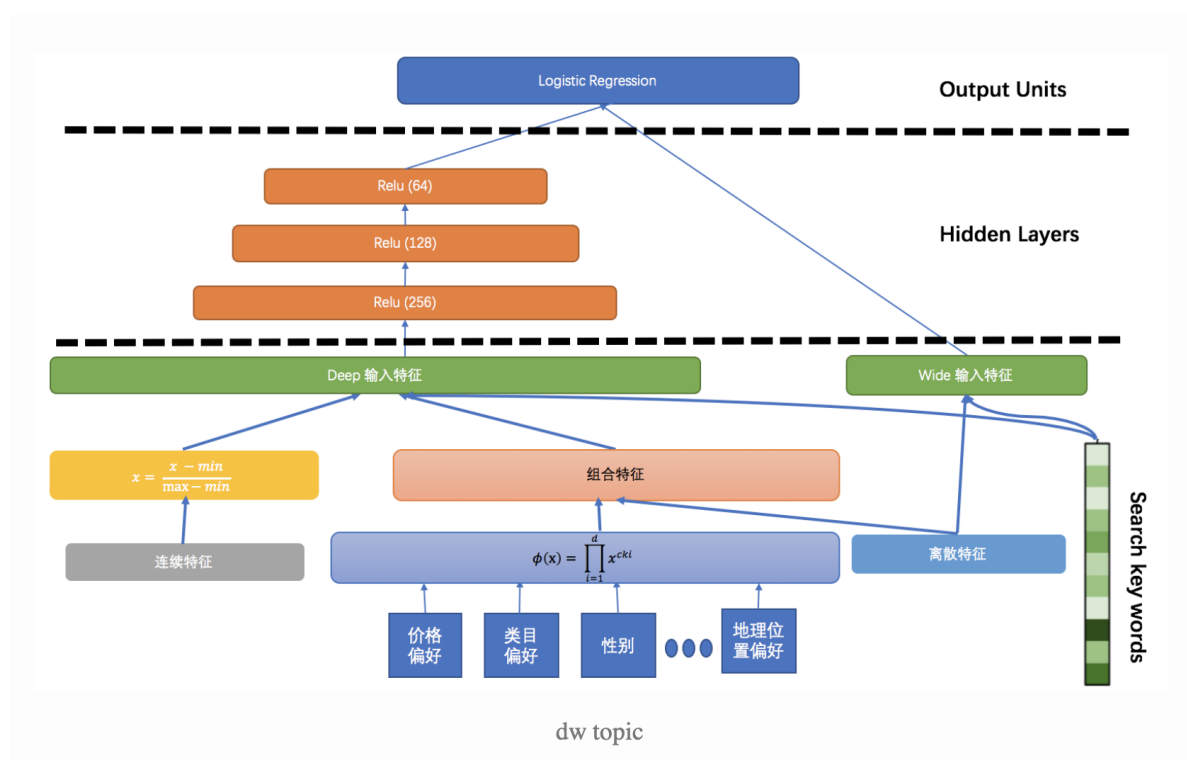
与平方损失函数相比，交叉熵函数有个非常好的特质：

$$H(y, a) = \frac{1}{n} \sum (a_n - y_n) = \frac{1}{n} \sum (\sigma(z_n) - y_n)$$

可以看到，由于没有了 σ' 这一项，这样一来在更新 w 和 b 就不会受到饱和性的影响。当误差大的时候，权重更新就快，当误差小的时候，权重的更新就慢。

6.2.6 宽深度模型框架

在实验初期，美团只将单独的5层DNN模型与线性模型进行了比对。通过线下/线上AUC对比，发现单纯的DNN模型对于CTR的提升并不明显。而且单独的DNN模型本身也有一些瓶颈，例如，当用户本身是非活跃用户时，由于其自身与Item之间的交互比较少，导致得到的特征向量会非常稀疏，而深度学习模型在处理这种情况时有可能会过度的泛化，导致推荐与该用户本身相关较少的Item。因此，美团将广泛线性模型与深度学习模型相结合，同时又包含了一些组合特征，以便更好的抓住Item-Feature-Label三者之间的共性关系。在宽深度模型中的宽线性部分可以利用交叉特征去有效地记忆稀疏特征之间的相互作用，而在深层神经网络部分通过挖掘特征之间的相互作用，提升模型之间的泛化能力。下图就是美团的宽深度学习模型框架：



在离线阶段，美团采用基于Theano、Tensorflow的Keras作为模型引擎。在训练时，分别对样本数据进行清洗和提权。在特征方面，对于连续特征，用Min-Max方法做归一化。在交叉特征方面，美团结合作业务需求，提炼出多个在业务场景意义比较重大的交叉特征。在模型方面使用Adam做为优化器，用Cross Entropy做为损失函数。在训练期间，与Wide & Deep Learning论文中不同之处在于，美团将组合特征作为输入层分别输入到对应的Deep组件和Wide组件中。然后在Deep部分将全部输入数据送到3个ReLU层，在最后通过Sigmoid层进行打分。

宽深度模型中的宽线性部分可以利用交叉特征去有效地记忆稀疏特征之间的相互作用，而深层神经网络可以通过挖掘特征之间的相互作用，提升模型之间的泛化能力。在线实验结果表明，宽深度模型对CTR有比较明显的提高。同时，美团也在尝试将模型进行一系列的演化：

1. 将RNN融入到现有框架。现有的Deep & Wide模型只是将DNN与线性模型做融合，并没有对时间序列上的变化进行建模。样本出现的时间顺序对于推荐排序同样重要，比如当一个用户按照时间分别浏览了一些异地酒店、景点时，用户再次再请求该异地城市，就应该推出该景点周围的美食。
2. 引入强化学习，让模型可以根据用户所处的场景，动态地推荐内容。

深度学习和逻辑回归的融合使得他们可以兼得二者的优点，也为进一步的点击率预估模型设计和优化打下了坚实的基础。

7 参考文献

1. H. Cheng, L. Koc, J. Harmsen et al, [Wide & Deep Learning for Recommender Systems](#), DLRS 2016 Proceedings of the 1st Workshop on Deep Learning for Recommender Systems.

2. P. Covington, J. Adams, E. Sargin, [Deep Neural Networks for YouTube Recommendations](#), RecSys '16 Proceedings of the 10th ACM Conference on Recommender Systems.
3. H. Wang, N. Wang, D. Yeung, Collaborative Deep Learning for Recommender Systems, KDD '15 Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.
4. 潘晖, [深度学习在美团推荐平台排序中的运用](#), 美团技术团队