Ad Serving Using a Compact Allocation Plan

译者：Koala++/屈伟

1 Introduction

在线广告的核心问题是广告投放，比如如何迅速地向用户展示某个广告，并得到一个全局的目标函数最优解。 本文主要关注合约式广告，合约式广告是一个有数十亿的行业。在合约式广告中广告主会购买未来一段时间（2011年七月～八月）的某些定向用户的流量 （比如，加州访问体育方面的男性用户的100万次访问），广告系统会在广告主购买的展示时间的几个月前确定是否这个合约是可以保证的。 在广告系统确认后可以保证后，在广告主购买的时间内，就会展示他的广告。在用户访问时，广告系统需要在瞬间决定应该出哪个合约广告。 才能保证最终所有的合约都能得到满足，并且，广告系统还要满足另一个隐性的目标，即对每个合约广告， 展示次数应该是尽可能均匀分布的，比如，一个合约广告在合约中规定在两个月内展示，广告就不应该在前几天就达到展示次数。

上述的场景可以建模成一个分配问题，定义如下：用户访问集合为I，合约广告集合为J。一个可行的分配边集合为$E \subseteq I \times J$， 可行的分配边即是用户满足广告的定向条件时，连接用户和广告的边。每个合约广告$j \in J$都有d_j次展示次数的要求，另外， 还可能有其它的目标函数，比如，展示次数应该是均匀分布的。我们目标就是，找出最大化目标函数的分配方案。

即使在离线方式下解决这个问题都很困难，它涉及大数据量处理，首先I集合特别大，对大的广告系统， 每天有百亿级的用户访问，因为合约广告可能解决的是一年后的分配问题，所以可能是十万亿级到百万亿的集合大小 （注意，$E \subseteq I \times J$），其次，我们通常并不准确地知道I集合，只是能近似知道，比如， 我们不能准确地知道一个用户是否会访问某页面，只有基于他历史行为的统计估计。

当然，现实中，我们需要实时地解决这个问题，这就更加困难了，每次请求处理时间应该在一百微秒内完成， 另外，首先服务是分布在几百台，上千台机器上的，并行地进行服务，它们之间并不互相通信， 但它们却需要共同地完成最终最大化目标函数的目标。最后，因为用户行为是长尾行为， 广告系统需要明确地处理历史上从未出现的用户访问。

其中一个简单的处理方法是：一个用户访问后，先得到所有可满足定向条件的广告，然后基于已经展示的广告次数做出选择， 比如，有两个可满足的广告，其一广告严重under-delivering（未达到目标展示次数），另一个已经离目标展示次数比较接近了 （甚至已经超过了(over-delivering)），那么广告系统将选择under-delivering广告。尽管这个方法概念上简单，可行，但它存在一些问题。 首先，这个方法需要定义under-delivering这个概念，最直接的方法是简单假设广告应该是均匀速度展示的， 比

如一个合约要求30天展示3千万次，那么就一天展示一百万次，这种方法的优点是能使展示的分布很均匀， 但是它忽略了流量本身在时间上变化是非常大的。比如，工作日的流量比周末的流量要小，夜间的流量比日间的流量小的多。 更进一步，可能出现满足某些合约的流量忽然减小，这就会使得均匀分布的策略不可行。这个情况出现的场景之一是， 有些广告主会进行排它方式的流量购买（一个体育公司可以将所有Yahoo!超级碗的流量买走）， 这就需要广告系统在这个合约生效前，对其它相关的合约广告多进行展示，其次要使用这个方法， 广告系统准实时地保存每个广告已经展示了多少次，这样才能实时计算under-delivery的量。但是在有一千台服务器遍布在全球， 不同地区的人有着不同的访问模式时，就会造成不同服务器之间流量很不均匀，所以这种情况下，要准实时保存会非常挑战。

2 Model And Problem Statement

我们先看一个合约式广告的具体例子，图中有三个广告主，它们的广告定向条件分别是1.男性，2.加州，3. 年龄5段。 有6种不同类型的用户，6种类型的用户都满足年龄定5段定向合约广告定向条件。有些用户是明确知道是来自加州，华盛顿， 内华达，而有些用户不知道来自于哪里，性别也一样，有些知道是男性，有些不知道。用户结点旁边的数字表示所有这种类型用户预计访问的次数。 比如图中第4个结点，预测加州年龄5段，性别未知的用户访问次数为100K次，Demand端的数字表示合约中要求的展示次数， 图中的边表示某类型的用户i是满足右边广告j的定向条件的。

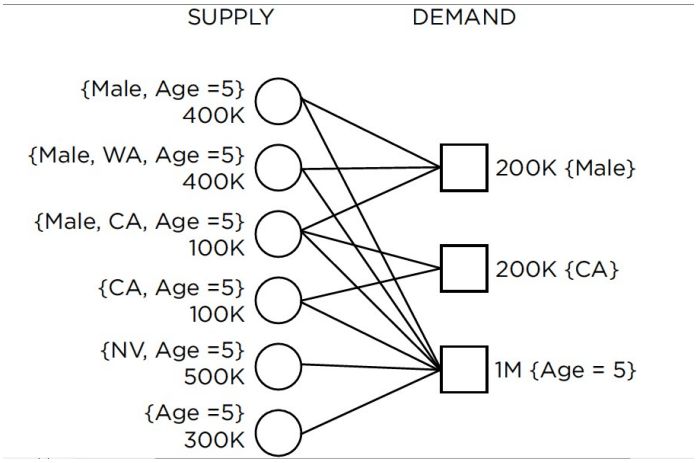

Figure 1: 一个分配问题的例子

广告系统要找到如何将左边的用户访问次数分配给右边的合约广告，比如，满足第2个广告的定向条件的第3个用户结点（Male, CA, Age=5）， 和第4个用户结点（CA, Age=5），都必须把流量全部分配给第2个广告，否则第2个广告的需求量就满足不了。 虽然第3个用户结点（Male, CA, Age=5）三个广告的定向条件都满足。

上面的例子当然只是一个小的示例。实际的问题至少在三个维度上数据量上要大的多。1. 用户的访问是每天十亿级的， 并且要在几个月前保证合约的量是可以完成的。2. 定向条件有数百种，包括人口学信息，年龄，地域，性别，明确兴趣，推测兴趣。 最近浏览行为等等。页面信息有主题，广告大小等等，时间信息有日期，用户时间，标准时间， 3.大的广告平台可以有几百上千的合约。所以，对于supply和demand端来讲都非常大，并且，在实际中，广告主的定向条件都是复杂的属性布尔表达式表示的。

2.1 Problem Statement

令I是[预测]的用户访问集合，J是合约集合，我们个分配问题建模为一个分图，$G = (I \cup J, E)$，边$(i, j) \in E$表示用户i满足合约j的定向条件，二分图可以表达合约的满足度。比如， 一个广告结点j只有少量的外边表示它是一个精细定向的合约广告，即表示只有少数类型的用户可以满足这个合约的定向条件。 一个用户结点$i \in I$有很多的外边，表示它是一个满足度高的用户，这个类型可以满足多种合约广告的定向条件。

除了定向条件的约束外，每个合约$j \in J$还有量的需求d_j，类似的，一个用户可能有多个同类型的用户， 每个结点$i \in I$旁边有这个类型的总量s_i的标签。

我们希望可以找到一个可行的分配方案，分配方案是每条边上有一个值x_{ij}，表示有x_{ij}比例的i结点流量分配给j合约广告。 如果x满足下面的结果，我们认为x是一个可行解。

$$\begin{aligned}
&\forall_j \textstyle\sum_{i \in \Gamma(j)} x_{ij} s_i \geq d_j && \text{demand constraints} \\
&\forall_i \textstyle\sum_{j \in \Gamma(i)} x_{ij} \leq 1 && \text{supply constraints} \\
&\forall_{(i,j) \in E} x_{ij} \geq 0 && \text{non-negativity constraints}
\end{aligned}$$

其中$\Gamma(i)$表示i的邻居，$i : \Gamma = \{i : (i, j) \in E\}$，$\Gamma(j)$也是相似的定义。

尽管我们没有明确的表达，但一个重要的特性是平滑地展示广告。广告主不会希望它们所有的需求流量在一个小时就展示完了（受众也不希望）， 虽然这种方式也达到了他对量的需求。所以任何广告展示系统都应该每天对于一个广告展示次数近似地与当天的流量成比例。

3 Solution Overview and System Architecture

在提出我们的解决方案之前，我们先讨论几种可能的方案。

也许最简单的方案是对所有满足定向的广告进行随机地选取，这种方法是在时间空间复杂度都

很低，并支持大吞吐量， 不需要机器之间的通信，并有很好的泛化能力，可以处理没有见过的用户类型，不幸的是，它即失掉了正确性，因为这种方法并没有对合约进行区分， 所以，它会导致严重的under-delivery，并影响收入，比如，图一中，第3个结点(加州，男性，Age=5)的用户都应该分配给第2个合约， 否则第2个合约就会under-delivery，虽然它也可以满足第3个合约的定向。

另一个提高正确率的方案如下：离线地解决这个问题，计算出所有二分图边上的分配值x_{ij}，然后将这些值保存到广告投放服务器上， 在线时，当一个用户访问时，投放机器会直接查分配方案，将这次访问分配给合适的合约广告，这种方案明显是正确率高的 （因为它有着最优分配方案），但它的空间复杂度很高，因为二分图中有大量的结点和边要保存。另外更重要的是它不可泛化， 特别是，实践中我们无法预测出所有可能的用户访问（因为用户类型包含用户的特征，比如用户兴趣，IP地址，他们访问的页面，等等）， 举一例子说明，假设在上例中，有一个女性，Age=5的用户访问，因为这种用户没有相应的x_{ij}，所以虽然有两个可以满足定向的广告， 但投放机器即没有相应的分配信息来选择其中一个可选的广告。在这个例子中，我们应该展示“CA”合约，因它是满足度更低， 但是这个信息很难从x_{ij}中推断出来。

3.1 System Architecture

下图是系统结构图，有一个模块为Allocation Plan Generator，它接收预测用户访问预测结果和合约广告集合， 并计算分配方案，我们在用户访问的抽样样本上计算这个问题，在抽样样本上计算会更快并更可行， 计算出的分配方案会发送到所有广告投放机器上。

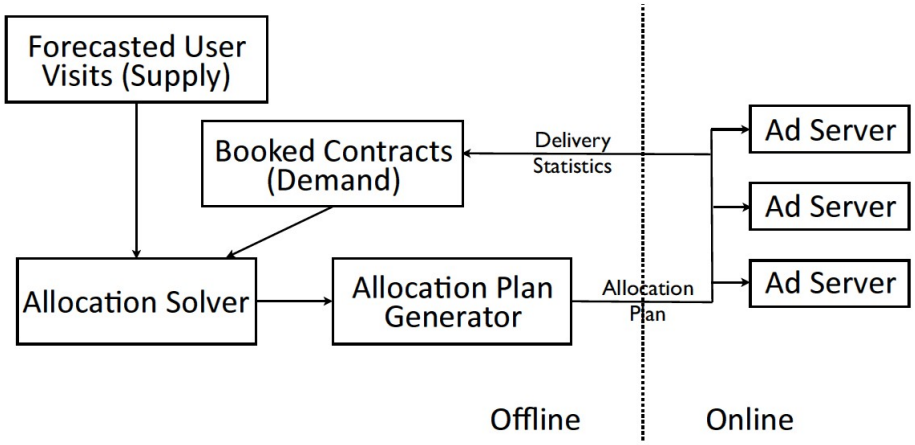

Figure 2: 系统架构图

所有的广告投放服务器在启动时加载分配方案，并将方案读入内存中。在线处理时，当有一个用户访问时， 它会先查找用户分配方案，然后根据分配方案选择一个广告展示，因为在线部分都是局部计

算的，所以是不需要广告投放服务器彼此通信的， 而且也不需要服务器维护全局状态，甚至不需要维护本机的展示次数。

在图中，还有一个从Ad Server到Booked Contract Demand的反馈循环，这样一个合约的需求量会根据已经展示的量进行修改， 因为在预测时可能会有误，所以Alloction Plan Generator会定期重算，根据剩余的需求展示量再次计算一个新的分配方案， 发送给Ad Server。

4 Rate-Based Algorithms

4.1 HWM Algorithm

离线部分

离线的HWM使用了一个简单即非常有效的启发算法来产生分配方案，算法为每个合约j产生一个服务率α_j，和一个分配顺序。 分配顺序是让满足度低的合约可以有较高的优先权得到更多的可行流量，通过对每个合约仅设置这两个值，HWM算法就创建了一个紧凑， 健壮的分配方案。

在线服务时，每个合约得到一次展示的α_j比例，除非流量不足的情况下，如果流量不足，会通过分配顺序来解决， 一个分配顺序靠前的合约j会得到α_j比例，下一个合约j'会得到$\alpha_{j'}$，如果不足$\alpha_{j'}$， 新得到到剩下所有的所有比例。

算法计算出每个合约j的可行流量S_j，并根据S_j来决定分配顺序，有着较小的S_j值的合约会排在分配顺序靠前的位置， 为了决定服务率α_j，HWM进行以下步骤：

1. 对所有的i，初始化剩余流量$r_i = s_i$。

2. 以分配顺序对每个j，进行：

(a). 解决下面的α_j

$$\sum_{i \in \Gamma(j)} min\{r_i, s_i\alpha_j\} = d_j$$

如果无解，设置$\alpha_j = 1$。

(b) 对所有的$i \in \Gamma(j)$，进行更新$r_i = r_i - min\{r_i, s_i\alpha_j\}$。

注意，以分配顺序进行计算是一个简单的启发方法，另外离合约截止日期远的合约也应该有高的优先级，还有特别精细的定向， 它们的定向条件只有少数的用户可以满足，所以也应该有高的优先级。

通过HWM离线产生的紧凑的分配方案如图3所示，注意只有图右边的部分会发给投放服务器，注意，分配方案依赖于流量预测， 所以不正确的流量预测会产生次优的结果。

在线部分

给定一个分配方案，其中有分配顺序和服务率α_j，广告投放服务器的任务是选择其一满足定向的合约，下面是它的算法：

1. 给定一个展示i，令$J = \{c_1, c_2, \cdot, c_{|J|}\}$为满足定向合约广告列表，以分配顺序排序。

2. 如果$\sum_{j=1}^{|J|} \alpha_j > 1$，令$l$为满足$\sum_{j=1}^{l} \alpha_j \leq 1$条件的最大值， 最后令$\alpha'_{l+1} = 1 - \sum_{j=1}^{l} \alpha_j$，注意因为$l$的定义，$\alpha'_{l+1} < \alpha_{l+1}$。

3.以α_j的概率选择一个合约$j \in [1, l]$，并以α'_{l+1}的概率选择$l+1$合约广告， 注意如果在$\sum_{j=1}^{|J|} \alpha_j < 1$的情况下，可能没有合约会被选中。

考虑图3中的例子。如果一个类型为{CA Age=5}的用户来了，$l = 1$，并且第2个合约结点的分配顺序为1，且有服务率1， 所以这种类型的用户总是分配给第二个合约的，再假设{Male, Age=5}类型的用户来了，那么只1/4的概率会分配给合约1， 有5/8的概率会分配给合约3，并且有剩下的1/8概率是不分配的（如果没有合约被选中，那么这个流量就会给到非合约式广告）。

4.2 Robustness to Forecast Errors

因为基于比率的算法产生的是服务率而不是直接的目标x_{ij}，所以它对负载均衡，服务失效等等都是健壮的。 但是它对预测错误是敏感的，比如，如果流量预测的结果是实际值的两倍，那计算出的服务率就比正确值大了一倍， 在本节中，我们会分析在理论的设置下，频繁的再优化会极大的减轻这个问题。

让我们来看一个对于所有合约，流量预测比实际流量多的例子。起初，所有的合约都是under-delivery的， 因为实际的流量比预测的流量小，那么$alpha_j$在分配方案中比正确值要小，随着时间推移，服务率值是会增长的， 这并不是因为流量预测的错误被捕获了，而是因为合约的结束日期近了，特定的合约的展示紧迫性也就提高了。

Table 1: 由错误率引起的服务率增长

天	预测流量	剩余展示量	服务率	实际展示量
1	5M	2.5M	0.50	0.4M
2	4M	2.1M	0.525	0.42M
3	3M	1.68M	0.56	0.45M
4	2M	1.23M	0.62	0.49M
5	1M	0.74M	0.74	0.59M
6		0.15M		

为了说明这个问题，我们举一个例子，假设有一个合约时间为5天合约j，流量预测每天有1M的展示量，合约j的要求展示量为2.5M， 如果它是系统中唯一的合约，那么我们所计算的服务率

应该是$alpha_j$=2.5/5M=0.5，如果流量预测完全正确， 那么展示量会在第5天结束时完成2.5M次展示，但是，如果实际的展示流量是800K每天（20%的流量预测错误率）， 那么在第一天结束时，服务率会再次计算为$alpha_j$=2.1M/4M=0.525，以此类推，我们可以得到表一中的结果， 从表中可以看到在5天后，有.15M/5M=6%的under-delivery比例。注意，如果没有再优化的过程，会有恰为20%的under-delivery比例， 所以频繁的再计算会减轻预测错误的影响。

我们将上述的例子形式化为下面的定理。

定理1. 假设我们有k次再优化迭代，流量预测错误（错误率为1减去实际流量与预测流量比）为r，且合约是可完成的， 在$r > 0$的情况下，undervery是一个正值，且边界为$\frac{r+r^2}{k^{1-r}}$，在$r < 0$的情况下，over-delivery是正值，并边界为$\frac{|r|}{k^{1-r}}$。

证明：在第i轮，优化器会分配$1/(k-r+1)$份流量，其中有(1-r)份会展示合约广告，所以广告主的需求量会减少比例为$1-\frac{1-r}{k-i+1}$， 那么在最后一轮，under-delivery的流量比例为：

$$\prod_{i=1}^{k}(1-\frac{1-r}{k-i+1}) = \prod_{i=1}^{k}(\frac{k-i+r}{k-i+1}) = \frac{r}{k}\prod_{k=1}^{k-1}(1-\frac{r}{i})$$

注意，当$r > 0$时，这个值是正值（即意味着under-delivery），当$r < 0$时，这个值是负值（即意味着over-delivery）， 首先，考虑$r > 0$的情况，我们利用$\sum_{i=2}^{k-1}\frac{1}{r} \leq \ln(k-1) < \ln k$事实。我们有：

$$\frac{r}{k}\prod_{i=1}k-1(1+\frac{r}{i}) \leq \frac{r(1+r)}{k}\exp(\sum_{i=2}k-1\frac{r}{i}) \leq \frac{r+r^2}{k}\exp(r\ln k) = \frac{r+r^2}{k^{1-r}}$$

现在$r < 0$的情况，我们利用公式$\sum_{i=1}^{k-1}\frac{1}{r} > \ln k$，现在，我们利用$\sum_{i=1}^{k-1}\frac{1}{r} \geq \ln k$事实，我们有：

$$\frac{|r|}{k}\prod_{i=1}k-1(1+\frac{r}{i}) \leq \frac{|r|}{k}\exp(\sum_{i=1}k-1\frac{r}{i}) \leq \frac{|r|}{k}\exp(r\ln k) = \frac{|r|}{k^{1-r}}$$

所以，即使在错误率大到2X的情况，它导致计算出的服务率只有正确值的0.5，但是它不会导致50%的under-delivery量，如果在一个一星期长的合约中， 每2小时重新优化（计算）一次服务率，我们有84次修正，最终的under-delivery量只有8.2%，相似的，如果预测流量为0.5X， 服务率为正确值的2倍，那么也不会有100%的over-delivery，而是只有小于1%的over-delivery。

4.3 Feedback-Based Correction

在前一节中我们看到频繁的再优化可以减轻流量预测错误的影响，但是尽管这样不会导致大量的under-delivery， 但线下优化在开始时发现流量预测错误的速度很慢，甚至在流量预测错误率很高的

情况下。 另一个解决方法是引入一个反馈系统，并根据错误率的高低相应地去进行修正错误。

这种反馈系统是控制论的领域，有很多反馈控制器可以实现这个目标。我们给出一个非常简单的解决方案， 这个解决方案在实践中表现很好。这个解决方案用两个参数进行控制：δ，控制展示过程中允许的松驰， $(\beta+, \beta-)$控制对服务率的提升。

如果在合约的合约时间内，合约delivey的速度落后了δ小时，反馈系统会将剩余未展示的量乘上$\beta-$系数。 比如：令δ=12，考虑一个合约时间为7天，展示量为70M的合约，理想情况下，合约会在第3天晚展示3M次。 但如果在第4中午只有2M次展示，反馈系统会将剩余的展示量(5M=7M-2M)乘上因子$\beta+$，即5M$\times\beta+$， 另一种情况，如果在第3天中午已经有3M次展示，反馈系统会将剩余的展示量乘上因子$\beta-$。