

刘鹏

# 广告的秘密

互联网计算广告技术

July 28, 2014

Higher Education Press



**Dedicated to \*\*\*\*\***



---

# Preface

广告支撑着互联网行业的大半壁江山。



---

# Contents

---

## Part I 写给想从事互联网广告的朋友们

---

<b>1</b>	<b>在线广告综述</b>	<b>3</b>
1.1	广告的定义与目的	4
1.2	在线广告类型	5
1.3	互联网广告简史	8
1.4	广告有效性原理	11
<b>2</b>	<b>如何入门</b>	<b>15</b>
2.1	本书在线课程	15
2.2	相关阅读书籍点评	15
2.3	相关从业技术概览	15
<b>3</b>	<b>计算广告基础</b>	<b>17</b>
3.1	互联网广告的技术特点	18
3.2	计算广告的核心问题	19
3.2.1	投资回报率的数学表达	19
3.2.2	付费模式产生的原因——谁来估计展示价值	20
3.2.3	广告算法核心任务——如何量化展示价值	21
3.3	计算广告系统架构	22

---

## Part II 计算广告的今生前世

---

<b>1</b>	<b>合约时代</b>	<b>27</b>
1.1	前合约时代	28
1.2	受众定向技术的发展	29

1.3	担保式投放的成熟 .....	29
1.4	代表公司 .....	30
1.4.1	担保式投放:Yahoo .....	30
1.4.2	受众定向:Criteo .....	30
1.5	合约广告的优缺点 .....	30
<b>2</b>	<b>竞价时代 .....</b>	<b>31</b>
2.1	竞价的位置拍卖理论简述 .....	32
2.1.1	GSP .....	33
2.1.2	VCG .....	34
2.2	按点击付费的竞价系统之崛起 .....	34
2.2.1	广告联盟模式 .....	34
	代表案例介绍:Google Display Network, 阿里妈妈 .....	35
	阿里妈妈联盟 .....	36
2.2.2	搜索广告模式 .....	37
	代表案例介绍:Google, Baidu .....	39
2.2.3	广告购买平台 .....	39
	非RTB的流量的ROI优化简述:SEM,SEO .....	39
	代表案例介绍: SEM公司,SEO公司,Efficient frontier .....	40
2.3	应运而生的计算广告核心技术 .....	40
2.3.1	广告系统的检索技术 .....	40
2.3.2	点击率预测模型 .....	41
<b>3</b>	<b>实时交易时代 .....</b>	<b>43</b>
3.1	实时竞价的广告交易平台 .....	44
3.1.1	广告交易平台架构 .....	44
3.1.2	代表案例介绍:rightmedia, Google adx .....	44
	Rightmedia .....	44
	DoubleClick Adx .....	44
3.2	需求方平台(DSP) .....	45
3.2.1	重定向 .....	45
	个性化重定向 .....	46
	搜索重定向 .....	47
3.2.2	新客推荐(Look-alike) .....	47
3.2.3	代表案例介绍 .....	48
	Criteo .....	48
	InviteMedia .....	49



3.3	供给方平台(SSP)	49
3.3.1	SSP系统架构	51
	AdMeld	51
3.3.2	代表案例介绍:Admeld	51
3.4	数据加工与交易	51
3.5	数据管理平台(DMP)	52
3.5.1	DMP案例	53
	Bluekai	53
	AudienceScience	54
3.6	广告程序交易谱系	54

---

## Part III 计算广告核心技术详解

---

<b>1</b>	<b>基础知识准备</b>	59
1.1	信息检索	59
1.1.1	信息检索	59
	倒排索引	59
	向量空间模型	61
1.2	最优化方法	63
	下降单纯型法	63
	最速下降法	64
	拟牛顿法	65
	节省空间的拟牛顿法-LBFGS	65
1.3	统计机器学习	65
	最大熵原理	65
	指数族分布	65
	混合模型和EM算法	66
	贝叶斯学习	68
	共轭先验	69
	经验贝叶斯	70
	变分法	71
<b>2</b>	<b>合约广告中的核心技术</b>	73
2.1	广告投放机系统架构	73
2.2	在线分配问题描述	74
2.2.1	在线分配问题表达	74
	供给与需求二部图	74

需求约束.....	75
供给约束.....	75
问题框架.....	75
2.2.2 在线分配问题举例 .....	76
GD问题 .....	76
MRA问题 .....	77
Ad words问题.....	77
2.3 分配问题优化算法.....	77
2.3.1 直接求解的原始分配方案.....	78
2.3.2 无状态的紧凑分配方案 .....	78
2.3.3 启发式的分配方案HWM.....	82
2.3.4 综合分配方案Shale.....	84
2.3.5 最差分配性能研究 .....	84
2.4 预定问题的在线优化算法 .....	86
2.4.1 基于HWM的增量算法.....	86
2.4.2 基于Shale的增量算法.....	86
2.5 流量预测 .....	86
2.5.1 基于时间序列的流量预测.....	86
2.5.2 基于bitmap的流量预测 .....	86
2.5.3 基于检索的流量预测 .....	86
2.5.4 频次控制下流量预测的修正.....	86
2.6 频次控制 .....	87
<b>3 受众定向核心技术 .....</b>	<b>89</b>
3.1 定向方法综述.....	90
3.2 上下文定向 .....	93
3.2.1 半在线抓取系统 .....	93
3.3 行为定向 .....	94
3.3.1 受众定向标签体系 .....	95
3.4 行为定向数据来源.....	95
3.4.1 行为定向建模问题 .....	96
3.4.2 行为定向特征生成 .....	97
3.4.3 行为定向计算过程 .....	99
3.4.4 行为定向的评测 .....	100
3.5 文本主题挖掘.....	101
3.5.1 无监督主题模型 .....	101
LSA模型 .....	102

PLSI模型和GaP模型 .....	102
LDA模型 .....	103
3.5.2 有监督主题模型 .....	104
<b>4 竞价广告核心技术 .....</b>	<b>105</b>
4.1 位置拍卖市场 .....	105
4.1.1 定价策略 .....	106
广义第二高价 .....	106
VCG定价 .....	107
4.1.2 市场保留价 .....	107
4.1.3 价格挤压 .....	108
4.1.4 整体竞价和计价过程 .....	108
4.2 广告网络系统架构 .....	109
4.3 广告检索 .....	110
4.3.1 布尔表达式的检索 .....	111
4.3.2 WAND算法 .....	115
4.3.3 线性评价函数的意义 .....	117
4.3.4 搜索广告的检索 .....	117
4.4 广告排序概述 .....	117
4.5 点击率预测模型:逻辑回归 .....	118
4.5.1 广义线性模型视角 .....	119
4.5.2 最大熵视角 .....	119
4.5.3 基本优化算法 .....	119
4.5.4 信任域法 .....	123
4.5.5 ADMM计算框架 .....	127
4.5.6 点击率模型的校正 .....	131
4.6 点击率模型举例:GBDT .....	132
4.6.1 点击率预测特征 .....	132
点击率特征的构成 .....	132
静态与动态特征 .....	132
位置偏差与CoEC .....	133
常见的偏差特征 .....	134
点击反馈的平滑 .....	135
4.6.2 点击率预测评测 .....	136
4.6.3 智能频次控制 .....	139
4.6.4 探索与利用 .....	140
UCB方法 .....	141

Contextual Bandit .....	141
4.7 点击价值估计 .....	142
4.8 搜索广告特定领域核心技术 .....	142
4.8.1 搜索广告系统架构 .....	143
4.8.2 搜索广告概览 .....	143
4.8.3 短时行为反馈 .....	144
4.8.4 流式计算平台 .....	145
4.8.5 关键词推荐 .....	145
4.8.6 竞价优化 .....	145
4.8.7 查询扩展 .....	145
基于推荐的方法 .....	146
SVD++ .....	146
基于主题模型的方法 .....	146
基于历史效果的方法 .....	147
4.8.8 广告条数个性化 .....	147
<b>5 广告交易市场核心技术 .....</b>	<b>149</b>
5.1 DSP系统架构 .....	149
5.2 数据管理平台DMP架构 .....	150
5.3 实时竞价系统架构 .....	150
5.4 实时竞价流程 .....	151
5.5 Cookie Mapping .....	152
5.6 询价优化 .....	154
<b>6 广告其他相关技术 .....</b>	<b>155</b>
6.1 创意优化 .....	156
6.2 流量保护和效果监测 .....	156
6.2.1 爬虫流量去除 .....	157
6.2.2 反作弊 .....	157
6.2.3 效果监测 .....	158
6.2.4 广告安全 .....	158
6.3 隐私保护和数据安全 .....	158
6.3.1 用户身份标识 .....	159
6.3.2 去匿名化技术及应用 .....	160
6.3.3 RTB中的数据安全问题的 .....	160
6.4 SoLoMo时代的广告 .....	161
6.4.1 移动广告 .....	161

6.4.2 社交广告.....	161
-----------------	-----



# **Part I 写给想从事互联网广告的朋友们**





## 在线广告综述

计算广告与网页搜索、推荐系统，都是互联网时代具有挑战性的大规模计算问题。由于数据规模的要求，这些系统都较广泛地采用检索(Retrieval)加排序(Ranking)这样类搜索的系统架构，因而有非常多的相似之处。于是，大家也往往会产生这样的观念：只要基本掌握基于搜索的技术框架，可以比较容易地搭建和运营好好一个广告系统。因此本书伊始，再展开产品和技术讨论之前，我们先来探讨一下与广告这项业务本身的内在规律，这些讨论对我们后面具体问题的建模和求解，有着非常有意义的指导作用。

熟悉在线广告市场的朋友都知道，这一领域的产品形态和业务逻辑相当复杂。为了对这些复杂的内容先有个宏观上的把握，我们从两个方面来探讨广告的内在规律，一是它的内涵，即这种商业活动的定义与目的；一是它的外延，即在线广告发展的简要历史和其中产生的关键产品形态。对其中的许多概念和观点，读者未必能够马上准确深入地形成印象，然而随着后文内容的不断展开和发展，能够剥茧抽丝一般层层递进地加深理解，正是本书希望能够做到的。这部分还有另外一个目的，就是尽可能集中地对互联网广告的产品和技术术语做一介绍，以方便后面的讨论。

相比传统的线下广告，在线广告的具体形式由于互联网产品媒体形态、交互方式等方面非常大的差异性，也呈现出各种各样的表现形式。我们也将对其中比较常见的形式作简要介绍，也是希望读者能对在线广告的具体形式有直观的了解。同时，也希望大家能够与广告的根本目的相对照，把握这些多样的形式下在线广告产品的本质特点。

在了解了在线广告的概貌后，我们会对传统广告中发展起来的广告有效性理论做一些简要的回顾。通过把广告产生效果的过程分解为若干阶段，并讨论其中各阶段关键的影响因素，可以对在线广告情形下受众定向、创意优化及其他计算问题中有价值的技术点有概要性的把握。虽然这部分与具体的计算技术无关，却对计算广告中的一些根本原理有深入的揭示，希望读者有所重视。

本章的最后，还介绍了若干广告和在线广告领域中重要的行业协会。了解这些协会在广告业务中代表的利益方，以及他们对整个在线广告市场产品和技术形态的推动作用，对于有志从事广告技术和业务的读者来说也是必要的。

本章的内容与计算基本无关，其目的在于让读者在进入计算广告领域的一开始，就建立起以下重要的观念：首先，广告不同于搜索或推荐，它首先是一项商业活动，其次才是一项在互联网环境下需要技术优化的商业活动；其次，这一商业活动中广告主、媒体和用户的利益都需要被认真考虑和满足，才能达到整个市场的平衡和不断发展，在线广告市场所有产品和商业形式的演进，都是在这一主题下发生的。如果大家能在商业逻辑的框架下思考和探索计算广告技术，对理解和学习后文中的产品、架构和算法大有帮助。

## 1.1 广告的定义与目的

让我们先从了解广告的一般目的开始。什么是广告？对此我们可以参照William F. Arens在《当代广告学》[1]中给出的定义：

广告是由已确定的出资人通过各种媒介进行的有关产品(商品、服务和观点)的，通常是有偿的、有组织的、综合的、劝服性的非人员的信息传播活动。

这一定义虽然看似冗长，却并无赘语。此处我们只解释两个关键点：首先，它指出了广告活动的两个主动参与方：出资人(sponsor)和媒体(medium)。在数字广告复杂的市场结构中，我们可以用更一般性的术语来描述它们：需求方(Demand)和供给方(Supply)。这里的需求方，可以是广告主(advertiser)，代表广告主利益的代理商(agency)或者其它技术形态的采买方；这里的供给方，可以是媒体，也可以是其他技术形态的变现平台。当然，广告还有一个被动的参与方，即受众(audience)。请大家从现在开始，就牢牢建立起这样的概念：出资人、媒体和受众这三者的利益博弈关系是广告活动永远的主线，这一主线将贯穿于商业和产品形态的整个演化过程中。其次，该定义阐明了广告必须是有偿的、非人员的信息传播活动，这两点限制，前者使得广告的目标变得明确，后者使得这一目标可以采用计算的方式来优化，而这些都是计算广告产生的根本基础。

广告这一商业行为，其本质目的是什么呢？在不同的时代，广告主与媒体对这一问题存在着不同的认知。在传统媒体时代，供给方与需求方的在市场地位上有着相当的距离，不论你运营的是电视台、机场、或八卦杂志，都与大多数广告主需要的转化行为之间，有相当大的差距，这一阶段广告的目的，是希望借助媒体的力量来快速接触大量用户，以达到宣传品牌形象，提升中长期购买率与利润空间的目的。这种目的的广告，我们称为品牌广告(Brand Awareness)。当然，也有许多广告商希望能利用广告手段马上带来大量的购买行为，这种目的的广告，我们称为效果广告(Direct Response)。传统广告市场缺乏大量投送和优化效果广告的能力。这是因为，对短期效果的追求要求广告精准地送达目标人群，而这在传统媒体上缺乏有效地技术手段。我能够想起的以效果为目的的传统广告，恐怕只有在写字楼下散发的快餐传单。而数字媒体的出现，使得效果广告空前蓬勃地发展起来，这主要有两方面的原因：一是数字媒体的特点可以让我们低成本地投送个性化广告；二是一些在线服务，比如搜索、电子商务，由于可以更清楚地了解用户的意图，也就使广告效果的优化更加容易。

既然有品牌和效果这两种目标，究竟如何描述广告这种商业活动的根本目的呢，我们仍然借用《当代广告学》中的见解：

广告的根本目的，是广告主通过媒体达到低成本的用户接触。

也就是说，按某种市场意图接触相应的人群，进而影响其中的潜在用户，使他们选择广告主产品的机率增加，或者对产品性价比的苛求程度降低，才是广告的根本目的。至于短期内的转化效果，由于市场意图或媒体性质的不同，并不是直接可比。换句话说，我们如果仅仅以效果为目的来思考问题，可能会在某种程度上远离广告的本质和正确方法。举个例子，某个感冒药广告商，如果以短期效果为导向，那么最佳的策略是把广告投放给那些现在感冒的人，不过这显然是一个荒谬的决策；再比如，某汽车广告商为了提升自己的品牌形象，希望对自己竞品品牌的用户加强宣传，而在这部分人群上，广告的直接效果甚至不如随机投放。希望大家认清这一概念，这样在遇到多种广告策略的效果比较时，避免一些偏颇的结论。广告的“低成本”，是与那些“人员”的信息传播活动相对比的，即相比于那些由市场或销售人员完成的劝服活动，广告搭媒体流量的便车理应更加经济有效。要确定是否真的成本较低，需要用到一个“投入产出比(Return over Investment, ROI)”的评价指标，即某次广告活动的总产出与总投入的比例。在实际中，广告活动的总投入容易确定，但总产出的确定却不那么容易，特别是在投放以中长期收益为目标的品牌广告时。因此，绝对的ROI有时难以计算，不过通过比较广告投入与人员性劝服活动的投入，我们仍然可以评估广告的成本是否令人满意。

## 1.2 在线广告类型

在线广告除了商业和产品有诸多形态，其与用户接触的形式也有多种多样的选择，并且随着互联网产品的发展变得越来越丰富。诸多在线广告的类型，在投送方式和用户交互方式两方面有不同的选择，下面我们对其中一些常见的和重要的类型做一简要介绍。

一、条幅广告(Banner Ad): 这是显示广告中最传统，也是最典型的形式。这种广告一般是嵌入在页面中相对固定位置的图片，与内容一样需要占据固定的版面，因此，这种广告一般要有底层的备选广告素材，以防没有合适的广告匹配时页面上开天窗。目前，条幅广告大多数也都不再是静止的图片，二是由flash或其他技术方式实现的动态素材。下图给出了条幅广告的一个示例。



Fig. 1.1. 条幅广告示例

二、文字链广告(Textual Ad): 这种广告的素材形式是一段链接到广告主落地页的文字，在搜索广告中为主流形式，同时在媒体广告中也被广泛采用。文字链广告有时像条幅广告那样占据固定的版面，有时也可以穿插在大量内容链接条目中。在后一种情形下，广告投放引擎可以灵活决

定是否投出该文字链广告, 以及投出的条目数, 典型的例子仍然可以比如搜索广告, 例如下图给出的示例。

我司代理各报类文字广告, 价格优惠, 电话: 400-831-3770 [www.bj-hhtc.cn](http://www.bj-hhtc.cn)  
 华恒天成专业文字广告, 资深专业, 专注于服务, 创造品牌, 电话: 87765716.

Fig. 1.2. 文字链广告示例

三、富媒体广告(Rich Media Ad): 这类广告往往是利用视觉冲击力较强的表现形式, 在不占用固定版面位置的情况下, 向用户侵入式地投送广告素材。富媒体广告常见的形式有弹窗、对联、全屏等。它比较适合在高质量的媒体做一些品牌性质比较强的广告投放, 但是对用户的使用体验往往影响也较大。富媒体广告与条幅广告不同, 没有防天窗的问题, 在未售出的情况下可以静默。一些门户网站的首页有时会为某个品牌广告主提供专门定制的、交互形式很复杂的富媒体广告, 这样的广告一般不采用受众定向的投送方式, 也主要强调创意的冲击力和交互形式的特色。下图给出了一个富媒体广告中的弹窗广告示例。

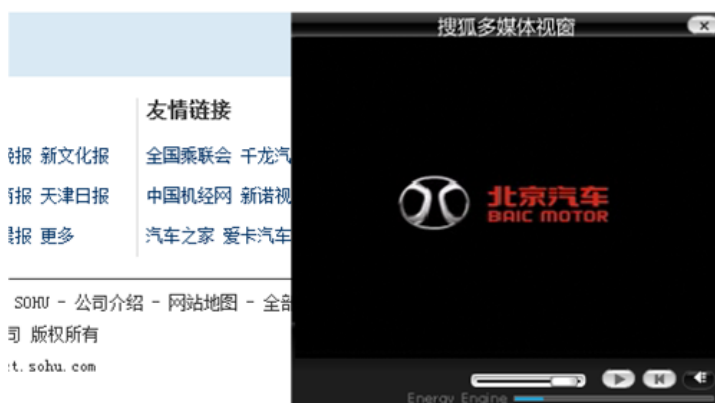


Fig. 1.3. 富媒体广告示例

四、视频广告(Video Ad): 随着在线视频的快速发展, 在视频流播放的间隙插入的广告也成为互联网广告的一种重要形式。根据插入位置的不同, 视频广告又可以分为前插片、后插片、暂停等品类。视频广告由于载体的独特性质, 其效果和广告创意会比较比较与类似线下的电视广告。相应地, 有关视频广告的效果评价, 除了计算与条幅广告一样的点击率, 还可以采用用户观看时长等更接近于用户印象的指标。

视频广告有两种最主要的形式: 在视频内容播放之前的前插片广告, 以及视频播放暂停时的广告。我们在上图中给出了这两种视频广告形式的示例。前插片广告一般采用短视频的形式, 创



Fig. 1.4. 视频广告示例：1. 前插片广告(左)；2. 暂停广告(右)

意的冲击力和表现力要远远强于普通的显示广告，因此CPM价格往往也比较高；暂停广告则与普通的条幅广告区别不大。

五、社交广告(Social Ad)：社交网络的兴起给广告的传播渠道和能力都赋予了新的空间。在社交网络环境下嵌入的广告，可以通称为社交广告。社交广告中最典型的形式，是插入在社交网络信息流中的广告，这种方式最早见于Twitter，产品称为“Promoted Tweets”。这种方式力求在用户自然关注的交互过程中尽可能自然地插入广告，也被归于所谓“Native Ad”的范畴中。结合前一章介绍的广告有效性原理，其效果可以想见相当不错。个人认为，“社交广告”与“社交网络中的广告”是两个不同的概念，例如在社交网络页面上竞价售卖的文字链或条幅广告，其本质并不因为处于社交网络中有太大的变化。社交广告希望达到的效果，是通过用户的扩散式传播获得更大的影响力，以及更可信的口碑，从这个意义上讲，在信息流的交互中挖掘价值是一条前途光明的道路。关于社交广告的独特作用，以及产品和技术上的特点，我们会在第九章中专门讨论。

六、移动设备广告(Mobile Ad)：移动互联网在近几年得到了爆发式的增长，并且大有快速取代桌面互联网的趋势。严格来说，移动互联网上的广告形与桌面电脑上的广告没有本质的区别，不过由于一定设备上应用的大量普及，广告也由Web访问的页面上被搬进了应用里。于是，也产生了在应用中插入广告的SDK和相应的广告网络。关于移动广告的一些特点和发展趋势，我们会在第九章中再讨论。

七、邮件营销广告(Email Direct Marketing, EDM)：这是通过电子邮件的方式向目标用户传递价值信息的一种网络营销手段。与上面各种广告形式都不同，EDM是一种主动的广告形式，它不需要等到用户接触的机会产生时才被动地提供广告，而是可以随时向认为合适的用户发送推广信息。不过也正因为如此，EDM非常容易变成垃圾邮件的主要来源。因此，对EDM的运营者而言，精准地把握用户兴趣，非常有节制地提供对用户用价值的相关信息是非常关键的。比起显示广告，EDM中受众定向的利用更加直接，从某种意义上说也更为重要。下图给出了邮件营销广告的一些示例。

		精品订阅	【139邮箱精品订阅】《意林》系列杂志5月里限时免费咯！读经典文摘，享...	5-23(四) 02:46	
		海信	海信电视“节能补贴收官，以旧换新重现”，领取340元折旧券！	5-22(三) 03:19	

Fig. 1.5. 邮件营销广告(EDM)示例

### 1.3 互联网广告简史

在开始讨论计算广告技术之前，我们有必要先从市场需求和产品形态的角度快速浏览一下在线广告发展的历程。这么做的动机，是因为广告市场的概念、技术和术语繁多，如果不是对这些有基本的了解，很难深入探讨并理解具体的产品或技术。

让我们回到上个世纪末，那时在线媒体，例如美国在线(American Online, AOL)、雅虎(Yahoo!)等网站，刚刚产生不久。他们已经取得了不错的流量规模，可是投资人当然希望这些媒体也能够给他们带来真金白银。要对这些线上流量进行变现，最直觉的方法，就是把网站的HTML页面当成是杂志的版面，在里面插入广告位，按照杂志的广告那样去售卖。供给方有了，那么需求方呢？线下广告的4A代理公司也就把这些网站当成一本本新的杂志，按原来的思路和逻辑进行采买。这种在互联网上展示条幅(banner)广告创意的产品形式，我们称为显示广告(Display Advertising)。这一阶段显示广告的售卖模式，我们称为合约式广告(Agreement-based Advertising)：即采用合同约定的方式确定某一广告位在某一时间段为某特定广告主所独占，并且根据双方的要求，确定广告创意的允许范围以及同一页面上的某种广告主排他策略。当然，在这样的采买模式下，还没有上任何对计算的需求，因为此时唯一需要技术支持人员帮忙的，就是把广告主的创意作为一个HTML的片段插入到媒体的页面中。然而，情况很快发生了变化。互联网媒体大多数是上市公司，为了使得营收的增长达到华尔街的预期，当然希望逐年甚至逐季提高自己的广告位报价。在流量快速增长，或者数字广告逐渐为广告主认知的阶段，这样的提价是比较容易被市场接受的。可是当尘埃落定，媒体的流量和品牌认知度都相对稳定以后，再想向广告主收取更多的费用就不那么容易了。这是否意味着这一市场将降低发展速度甚至趋近饱和呢？恰恰相反，精彩的故事才刚刚开始！

聪明的互联网广告运营者们经过探索，很快就发现了在线广告不同于传统媒体广告的本质特点：我们是可以对不同的受众呈现不同的广告创意的！在今天看来再平常不过的这个观念，在计算广告整个历史上闪烁着最耀眼的光辉，也是在线广告的效果和市场规模不断发展的核心驱动力。认识到了这一点，媒体马上找到了一条能使得广告位报价继续提高的思路：可以把同一个广告位的男性受众和女性受众卖给不同的广告主，比如对男性受众展示某剃须刀品牌的广告，而对女性受众展示某化妆品品牌的广告。这样的广告投放方式，称为定向广告(Targeted Advertising)。很显然，这样的广告系统已经对计算技术产生了两个具体需求：一是受众定向(Audience Targeting)，即通过技术手段标定某个用户的性别、年龄或其他标签。二是广告投放机(Ad Server)，它将广告投送由直接嵌入页面变为实时响应前端请求，并根据用户标签自动决策和返回合适的广告创意。由于从传统的品牌广告市场直接延伸而来，此时的定向广告仍然以合约的方式进行：媒体向广告主保证某个投放量，并在此基础上确定合同的总金额以及量未完成情况下的赔偿方案。这种“担保式投送(Guaranteed Delivery)”的交易方式，逐渐成为互联网合约



式广告的主要商业模式。一般来说，这样的广告合约仍然主要面向品牌广告主，并且按照按千次展示付费(Cost Per Mille, CPM)的计费方式。

合约广告系统中有一个重要的计算问题，即在满足各合约目标受众量要求的同时，尽可能为所有广告商分配到质更好的流量。这一问题有两个难点：一是如何有效地将流量分配到各个合约互相交叉的人群覆盖上；二是要在在线的环境下实时且经济地完成每一次展示决策。这个问题，我们称之为在线分配(Online Allocation)。如果将各合约的量看做约束条件，将某种意义上质的看做目标函数，我们可以利用带约束优化(Constrained Optimization)的数学框架来探索这一问题。为了得到在线环境下切实可行的解决方案，学术界和工业界的同仁进行了理论和工程方面大量的研究，有一些高效且简便的实用方案已经为各媒体广泛采用。

大多数广告系统的架构，就其在线部分而言，都非常像一个搜索引擎。具体地说，就是倒排检索加上排序这样的两阶段解决方案。广告的检索技术与搜索虽然有一定的差别，但基本框架非常类似。至于排序的准则，则与搜索有较大的区别：搜索排序以优化相关性为目标，但广告排序是以优化投入产出比为目标。

需要注意，显示广告领域定向投放的最初动机，是供给方为了拆分流量以获得更高的营收。如果一开始就提供非常精细的定向，反而会造成售卖率的下降，因此，最初的定向标签往往都设置在较粗的粒度上，最典型的是一些人口属性标签。不过，开弓没有回头箭！受众定向显然更符合需求方的口味和利益。不要忘了，广告市场的钱是全部来自需求方的，他们的利益被满足得越好，市场的规模就会越大。因此，受众定向产生以后，市场向着精细化运作的方向快速发展，主要有两方面的发展趋势：一是定向标签变得越来越精准；二是广告主的数量不断膨胀。在这些趋势下，仍然按照合约的方式售卖广告会碰到越来越多的麻烦：首先，很难对这些细粒度标签组合的流量做准确预估；其次，当一次展示同时满足多个合约的时候，仅仅按照量约束下的在线分配策略进行决策，有可能浪费掉了很多本来可以卖得更贵的流量。既然量的约束带来了这么大的麻烦，有没有可能大胆一点，抛弃量的保证，而采用最唯利是图的策略来进行广告决策？这样的思路催生了计算广告历史上革命性的产品模式——竞价广告(Auction-based Advertising)。在这种模式下，供给方只向广告主保证单位流量的成本，但不再以合约的方式给出量的保证，换言之，对每一次展示，都基本按照收益最高的原则来决策。

上面是从显示广告规律看竞价产生的原因，实际历史的足迹却并非如此。竞价广告产生的最初场景，是在互联网广告最主要的金矿——搜索广告(Search Ad)中。在上个世纪末，以谷歌(Google)为代表的搜索引擎技术成熟以后，迅速成为互联网新的入口点。与门户网站不同，搜索引擎从一开始就没有被当做媒体来看待，因此搜索流量的变现，也采用了与服务自然结合的付费搜索(Paid Search, Sponsored Search)模式。从广告的视角来看，我们也可以把付费搜索看做一种定向广告，即根据用户的即时兴趣定向投送的广告，而即时兴趣的标签就是关键词。很显然，这种定向广告从一开始就直接达到了非常精准的程度，也就很自然地采用了竞价的方式售卖。

当搜索广告产生了巨大的收益以后，搜索引擎开始考虑将这样的变现方式推广到其他互联网媒体上：将用户的即时兴趣标签有搜索词换成正在浏览页面中提取的关键词，可以将这套竞价广告系统从搜索结果页搬到媒体页面上，这就产生了上下文广告(Contextual Advertising)。上下文广告的初期，创意的形式也是与搜索广告一样的文字链接，这让许多产品讨论将它与显示广告分

开对待。不过当我们把它与上一段中精细化定向的显示广告对比来看时，就知道这样的区分实际上毫无必要，或许从创意形式上把显示广告和文字链广告(Textual Advertising)区别开更符合分类的逻辑。

从宏观市场上看，竞价广告与合约广告有很大的不同。没有了合约的保证，大量的广告主处在一个多方博弈的环境中。与直觉不同的是，在如何收取广告主费用这一点上，我们并非按照微观上最优的方案实施，就可以达到整个市场最大的收益。关于定价机制的深入研究，产生了广义第二高价(Generalized Second Pricing)这一竞价重要的理论。

基于竞价机制和精准人群定向这两个核心功能，在线广告分化出了广告网络(Ad Network)这种新的市场形态。它批量地运营媒体的广告位资源，按照人群或上下文标签售卖给需求方，在需求发生冲突时用竞价的方式决定流量分配。广告网络的结算，以按点击付费(Cost per Click, CPC)的方式为主，这一点有数据和业务方面多层次的原因，我们在后文会详细探讨。虽然我们不太能指这种方式的千次展示收益(Revenue per Mille, RPM)可以达到合约式品牌广告的水平，但它使得大量中小互联网媒体有了切实可行的变现手段：这些媒体有一定的流量，但还不值得建立自己的销售团队面向品牌广告商售卖，直接把自己的广告库存(Inventory)托管给广告网络，借助广告网络的销售和代理团队为自己的流量变现。当然，作为广告网络的一个特例，搜索广告的变现能力可是一枝独秀的。

我们说过，采买的量和质都是广告商永远存在的需求。广告网络既然只通过出价接口提供质的标准，那么由谁来保证量呢？自然会有需求方的某种产品来完成。当广告网络产生以后，4A代理公司当然也要对自己的广告采买方式做出调整，因为此时的流量采买发生了几点显著变化：一是更多地面向受众而非媒体或广告位进行采买，这当然是与受众定向的流行有直接关系；二是需求方的代理需要采用技术的手段保证广告主量的要求，并在此基础上帮助广告主优化效果。这又是一个受限优化，听起来很像前文提到的在线分配问题。但是实际上，这个问题有着本质的不同：由于只能在广告网络定义好的定向标签组合上预先指定出价，而不能控制每一次展示的出价，因此，市场看起来象一个黑盒子，需求方只能靠选择合适的标签组合，以及阶段性调整出价来间接控制效果。这种面向多个广告网络或媒体按人群一站式采买广告，并优化投入产出比的需求方产品，我们称为自动交易终端(Automatic Trading Desk, ATD)。

可以看出，在广告网络中，核心的竞价逻辑是封闭的。这种封闭的竞价机制，仍然不能完全满足需求方越来越明确的利益要求。试想下面的两个例子：一、某电子商务网站需要通过一次广告投放来向它的忠实用户推广某产品；二、某银行希望通过自己的信用卡用户在网络上找到类似的潜在用户群，并通过广告争取这批潜在用户。很显然，任何广告网络都不可能直接为这两个需求提供人群标签。与其想办法满足这些不断产生的定制化需求，不如干脆设计一种开放的竞价逻辑，让需求方按自己的人群定义来挑选流量！这样的目标催生了实时竞价(Real Time Bidding)技术。所谓实时竞价，就是把拍卖的过程由广告主预先出价，变成每次展示时实时出价。只要把广告展示的上下文页面url，以及访客的cookie等信息传给需求方，它就有充分的信息来完成定制化的人群选择和出价。需要注意的是，越是定制化的人群选择，往往就越精细，也就意味着单一媒体能达到的reach会相当有限。于是，市场上产生了大量聚合各媒体的剩余流量，采用实时竞价方式为他们变现的产品形态——广告交易平台(Ad Exchange)。这个名称让我们很容易联想



起股票交易所。事实上，如果我们把广告网络的交易方式想象成场外交易市场(Over-the-counter Market)，那么广告交易平台与股票交易所确实有着类似的作用。

通过实时竞价的方式，按照定制化的人群标签购买广告，这样与广告交易平台接口的产品，我们称为需求方平台(Demand Side Platform, DSP)。由于实时竞价主要采用按展示次数计费的方式(原因后文会详细讨论)，DSP需要尽可能准确地估计每一次展示带来的期望价值。在这一点上，DSP比ATD要幸运得多了，因为充分的环境信息使得深入的计算和估计成为可能。基于DSP的广告采买，非常类似于股票市场上的程序交易，我们把这样的广告采买方式也叫做程序购买(Programmatic Buy)。除了RTB以外，还有其他几种程序购买的交易方式，如优选(Preferred Match)以及前文提到的ATD中的自动出价购买。可以预见，与金融市场一样，在线广告中程序购买的作用和地位将会不断加强，这是由数字广告的本质特点以及广告主利益最大化的趋势所决定的。

可能有的初次接触在线广告的读者，会对这部分中提到的大量概念和商业逻辑感到无所适从。不过没关系，上面所有用黑体字标出的关键概念，我们在后文某个地方都会有详细的讨论。而我们写这一段的目的，也仅仅是让大家对本书讨论的范畴有个全局性的认识，从而在后文中接触到某个具体问题不会只见树木，不见森林。

由于在线广告存在着较为复杂的市场结构，LUMA将北美市场的主要代表公司作了一次非常著名的总结，并绘制成了图1.3中的“Display Landscape”图谱。这一图谱的骨架与上面我们介绍的在线广告简史，有着非常紧密的联系，因此也是本书在广告产品方面重要的提纲。基本上可以说，这一图谱是从两端向中间逐渐发展和形成的：首先是合约阶段，广告主通过代理公司(Agency)从媒体方采买广告，而媒体方的Ad Server则负责完成和优化各个广告主的合约；然后，市场进化出了竞价售卖方式，从而在靠近供给方产生了广告网络(Ad Network)这样的产品形态，而需求方的代理公司为了适应这一市场变化，孵化除了对应的广告采买平台(Media Buying Platform)；最后，当市场产生了实时竞价方式交易时，供给方进化出了广告交易平台(Ad Exchange)，而需求方则需要用需求方平台(DSP)与其对接来出价和投送广告。图中的下半部分，多是一些对这一骨干市场结构起支持作用的产品，或者在细分领域的特异化产品。我们在后面介绍到相关部分时，将会给出相应的介绍。

图1.3中的个别术语，与本书中的惯例略有区别。比如“广告采买平台(Media Buying Platform)”，我们会在书中用相近似的概念“自助交易终端(ATD)”来代替；而途中的“Yield Optimizer”这一产品，我们在书中将采用后来被更广泛采用的名称“供应方平台(SSP)”，这一点请大家留意。

## 1.4 广告有效性原理

为了探讨用计算技术优化广告投放的效果，我们还要先看看广告从产生用户接触开始，是如何产生最终效果的。这一问题传统广告领域一个重要的研究课题，我们直接借鉴了前人的研究成果，按照方便计算广告讨论的原则，用一个三段式信息传播模型来解剖广告由物理上产生到最终产生转化行为的全过程，这一模型如图1.4所示。

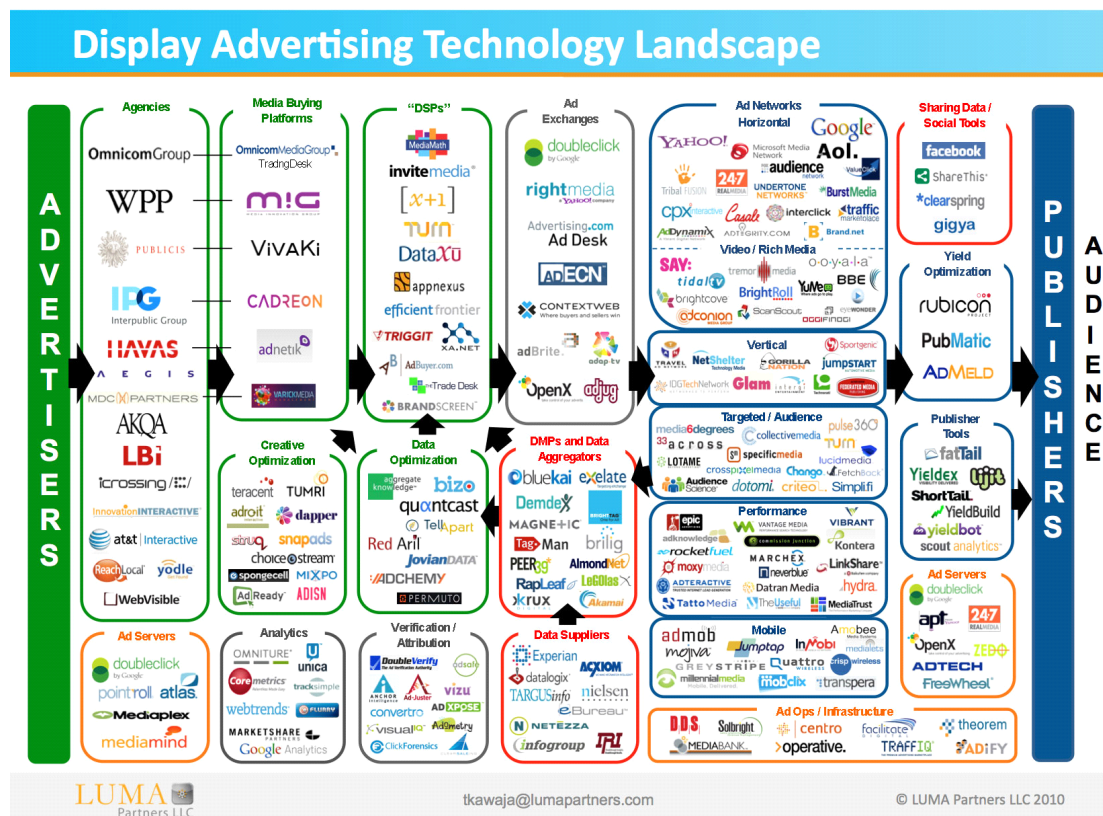


Fig. 1.6. LUMA总结的显示广告市场结构和代表公司图谱

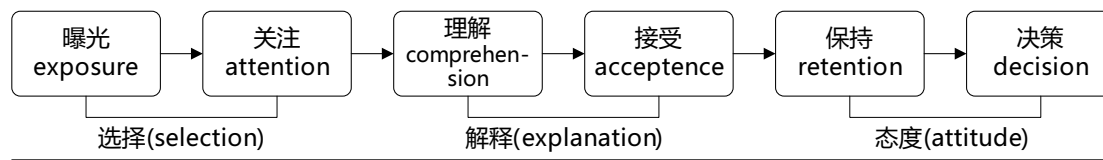


Fig. 1.7. 广告效果产生过程示意

根据上图，这以有效性模型把广告活动的整个信息接收过程分为三个大阶段：选择(Selection)、解释(Interpretation)与态度(Attitude)；或者进一步分解为六个小阶段：曝光、关注、理解、接受、保持与决策，其中每两个小阶段对应一个大阶段。下面我们分别来讨论每个阶段的意义和关键点。

一、曝光(Exposure)阶段：这一阶段指的是广告物理上展现出来的过程，此阶段的有效性往往与广告位的物理属性有关，并没有太多可以通过技术优化的空间。实际的广告实践中，曝光的有效性对最终结果的影响往往远远高于其他技术性因素，所以才会有传统广告中“位置为王”的说

法。北京的读者们一定有直观感受，东三环北端\*\*大厦外立面正对着整条东三环北路的广告牌有多么好的曝光效果。在互联网广告中，位置的影响有时会更加显著，因此如何从算法上消除由于由此带来的点击率预估偏差，是一个非常重要的实际问题。

二、关注(Attention)阶段：这一阶段指的是受众从物理上接触到广告到意识上注意到它的过程。对广告而言，这一过程并非那么简单。笔者曾经看到一位好友在浏览某社交网站，他的小儿子在旁边喊：“爸爸，快看网页上的恐龙！”这位朋友找了一分多钟，都没有找到恐龙，而实际上恐龙就在网页上端最醒目的广告位上。那么如何使得关注阶段的效率提高呢？我们介绍几个重要的原则：1. 尽量不要打断用户的任务。这一点是上下文相关的广告投送的原理基础，同时也能说明为什么文字链广告点击率普遍高于条幅广告。上面我举的例子，也可以用这个原则解释来解释，当用户明确辨识出某个固定不变的广告位，并且不再认为它与自己当浏览网页的任务有关联时，他会下意识地屏蔽其中的内容。2. 明确传达向用户推送此广告的原因，这一点是受众定向广告创意优化的重要方向。3. 内容符合用户的兴趣或需求，这是受众定向的原理基础。

三、理解(Comprehension)阶段：受众意识到了广告的存在，并不意味着他一定能够理解广告传达的信息。再举一个例子，笔者有一次试玩了一款网页游戏，然后被某游戏广告网络定向到，并多次看到某宣传“四维城战新模式”的游戏广告。应该说这样的定向是精准的，我也非常认真地关注了广告的内容，不过这些内容我确实不能直观理解，也就谈不上后续的转化。理解阶段有哪些原则呢？1. 广告内容要在用户能理解的具体兴趣范围内，这就说明了真正精准的受众定向有多么必要。2. 要注意设定与关注程度相匹配的理解门槛。比如在电视广告中，可以用有一定情节的短故事来宣传品牌；在路牌广告中，创意制作原则是将若干主要市场诉求都表达出来；而对于互联网广告，由于用户的关注程度非常低，我们应该集中强调一个主要诉求以吸引用户的注意力。

四、接受(Acceptance)阶段：受众理解了广告传达的信息，并不一定表示他认可这些信息。广告领域有一句名言：“我知道有一半的广告预算浪费了。”实际的情况还有可能更糟，如果表达的信息不适当，甚至有可能有三分之一的广告展示起到负面效果！在使用幽默、性感这样非常规的广告手段时，要特别注意这一点。广告的上下文环境对于广告接受程度也有着很大的影响，同一个品牌广告出现在某游戏社区上和门户网站首页上，用户会倾向于认为后者更具说服力，这也就是优质媒体的品牌价值。在定向广告越来越普遍的今天，如何让合适的广告出现在合适的媒体上，即广告安全(Ad Safety)的问题，正在引起大家越来越多的关注。

五、保持(Retention)阶段：对于不仅仅追求短期转化的广告商，当然希望广告传达的信息给用户留下长久的记忆，以影响他长时间的选择。品牌广告商在创意设计上了大量的精力提高此阶段效果，我们想想那些充满艺术性或浪漫气质的电视汽车广告，可以对此有直观的认识。

六、决策(Decision)阶段：成功广告的最终作用是带来用户的转化行为，虽然这一阶段已经离开了广告的业务范围，但好的广告还是能够为转化率的提高做好铺垫。特别是对于电商或团购业务，在创意上强调哪些信息以打动那些价格敏感的消费者，是有相当的学问的。

定性地讲，越靠前的阶段，其效果的改善对点击率的贡献越大；而越靠后的阶段，其效果的改善对转化率的贡献越大。但是以上各个阶段的划分，绝非孤立的和绝对的，而某一项具体的广

告策略或技术，也往往会对几个阶段的效果同时发生影响。虽然这样的有关广告有效性模型的讨论多见于传统广告的研究中，

## 2

---

## 如何入门

2.1 本书在线课程

2.2 相关阅读书籍点评

2.3 相关从业技术概览



## 计算广告基础

通过上一章的介绍，大家应该对在线广告业务上的本质和市场发展脉络有了初步的认识。本章的重点，是介绍在线广告中计算到底是为了解决哪些问题，以及解决这些问题所需要的系统框架与算法基础知识。

在互联网广告中，计算之所以可以发挥巨大的作用，与它的一些根本技术特点有很大关系，这也正是本章的出发点。总的来说，可衡量的效果以及相应的计算优化，是在线广告区别于线下广告的主要特点。在这些特色的基础上，我们对Andrei Broder提出的计算广告核心挑战稍作推广，得到贯穿本书的计算广告核心问题，即ROI优化问题的概念性框架表达。这一框架除了可以生发后面许多的计算和系统问题，还对在线广告市场结构和计费方式的一致理解很有帮助，我们在本章中对后一点着重做了展开讨论。

我们将会看到，在线广告多种多样的计费方式，实际上反映着市场结构的分工不同：具体来说，与供给方和需求方如何分工估计点击率和点击价值，从而完成整个市场的资源优化配置有关。对若干常见计费方式的深入理解，对于把握计算广告领域的核心问题，和评估每个问题在特定情形下的困难程度，有着非常重要的指导意义，也对于各种在线广告产品形态的把握有很大帮助。

为了方便后免各章的内容展开，特别是各种广告产品中的系统架构和算法讨论，我们在本章中将进行两方面的准备工作：首先，我们给出一个示意性的广告系统统一架构，虽然这一架构的各个组成模块在不同的广告产品中各有取舍和变形，却十分有利于我们从宏观上把握广告系统的全貌和各种产品在技术方面的内在联系。后文各种广告产品的架构讨论，都会在这个统一框架的基础上进行。其次，我们对计算广告中广泛使用的一些基础知识，如信息检索、最优化方法、统计及其学习等，作提纲挈领式的简要介绍。这样的是为了在后文讨论到相关算法时，可以有相对统一的展开基础，并避免在广告专门算法的讨论上穿插过大篇幅的基础内容。如果读者对某部分得知市已有相当的基础，则可以不作详细阅读，只简单了解本书的一些惯例数学表达；如果觉得某部分需要深入了解，那么还需要根据本章的简要提纲，参照其他专门文献，以获得更全面和深入的内容。

### 3.1 互联网广告的技术特点

从上一章的在线广告简史中，大家肯定已经发现了不少在线广告不同于传统广告的特点。在这些不同点当中，有一些对我们正确理解在线广告市场，对该市场的效果优化探究合适的解决方案，有着非常重要的指导意义。我们把这些重要特点总结出来，以便引出计算广告的核心问题与框架。

一、技术和计算导向。数字媒体的特点使在线广告可以进行精细的受众定向，技术又使得广告决策和交易朝着计算驱动的方向发展。之所以有这一特点，我们才会在这里讨论煞有介事地计算广告。实际上，受众定向这一思想在线下广告中也曾经被尝试过，比如试图把信用卡纸质帐单背面的广告按照信用卡用户的年龄和性别做一些定制化，不过由于非数字的媒体上这么做的成本太高，因而无法规模化。在数字媒体上进行受众定向，其成本可以控制得非常低，这也直接催生了在线广告的计算革命。除了受众定向，由于在线广告中独特拍卖性质的市场的存在，对于广告效果精确的预估和优化能力也是非常重要的。可以说，从来没有任何传统广告形式象在线广告那样，需要大规模地收集并利用数据，而这正是在线广告最吸引人之处。

二、效果的可衡量性。在线广告刚刚产生的时候，大家对这种广告最多的称道之处，是它可以以展示和点击日志的形式直接记录广告效果。当然，我们也可以利用这些日志优化广告效果，这同样是计算广告非常重要的方法论。不过，点击率这一指标从是否在绝对意义上能够反映广告效果，个人认为值得探讨。从98年到今天，条幅显示广告的点击率从10%逐渐降低到降至0.1%，难道这说明广告的效果下降了两个数量级么？快速增长的显示广告市场规模显然给出了否定的回答。我们认为，在同一个时期，点击率的绝对值并没有那么重要，而在一个特定时期不同广告和算法表现出来的差异，才是更有意义的。从这一点来看，可衡量性仍然可以认为是在线广告的一个根本特点。

三、创意和投放方式的标准化。标准化的驱动力来自于受众定向与程序购买。既然需求方关心的是人群而非广告位，创意尺寸的统一化与一些关键接口的标准化非常关键。这些接口标准中，比较典型的有上一章中介绍的视频广告的VAST标准和实时竞价的Open RTB标准等。实践表明，有越来越多的广告产品和平台愿意根据这些市场标准来设计自己的规范和接口，因为这样大家可以充分利用整个市场的流动性，更快地创造更多的价值。

四、媒体概念的多样化。随着Web 2.0的普及，赋予了更多交互功能的互联网媒体与线下媒体有大有不同。随着交互功能的不同，这些媒体与转化行为的距离也就不同。举个例子，对在线购物行业而言，门户网站、垂直网站、搜索引擎、电商网站、返利网，在转化链条上一个比一个更靠近购买行为。我们从直观就可以知道，越接近需求方的媒体上的广告，其带来的流量一定可以达到越高的ROI，不过离“引导潜在用户”这样的广告目的也就越远。因此我们在从需求方看在线广告时，应该注重各种性质媒体的配合关系，并从整合营销的角度去审视和优化整体的效果。试想，如果一家电商只用返利网作为线上广告渠道，ROI一定可以做到很高，可是返利网能给他带来潜在用户么？<sup>1</sup>

<sup>1</sup> 2012年底，淘宝发布声明，宣布中止与以现金形式返利的淘宝客网站的合作，正是反映了市场对此问题的认知正在趋向一致和理性。



五、数据驱动的投放决策。与工业革命时期机器化的根本驱动力——电力相类比，互联网化的根本驱动力可以认为是数据的深加工和利用。这一点在大数据概念被广泛认知的今天已经成为老生常谈。前面提到的在线广告的计算技术，在很大程度上也要依赖于对于数据的大规模利用。广泛收集用户的行为数据和广告反馈数据，利用云计算的基础设施对用户打上合适的标签，同样根据数据在多个广告竞争同一次展示时作出决策，再将投放的结果统计数据反馈给广告操作人员以调整投放策略。已经成为在线广告的基本投放逻辑。因此，现代的在线广告系统，可以认为就是一个大数据处理平台，而且其对数据处理的规模和响应速度的要求都相当高。从这个意义上讲，在线广告是互联网业务中技术含量较高的产品之一。

## 3.2 计算广告的核心问题

Andrei Broder在提出计算广告这一概念的时候，同时也给出了该课题的核心研究挑战(注意是“挑战”而不是“定义”)。这一核心挑战，他的表述是“Find the best match between a given user, a given context, and a suitable ad”。我们结合近年来市场的发展以及实际业务中的一些体会，对此表述稍做加工如下，给出计算广告的核心问题如下：

计算广告的核心问题，是为一系列用户与环境的组合，找到最合适的广告投放策略以优化整体的投入产出比(ROI)。

与Andrei Broder的表述相比较，我们主要进行的两方面的微调：一、我们强调广告问题优化的是一组展示上的效果，而非孤立的某一次展示上的效果。这是由于广告活动中普遍存在着量的约束，在这一约束下进行ROI优化，其最优解往往与每次展示独立决策时有很大的不同；二、描述中去掉了“given”的字眼，这是由于在某些广告产品中，系统并不一定能拿到确定的用户或上下文唯一标识，但这并不意味着完全无法进行计算优化。同样地，我们也强调优化的结果是“广告投放策略”而不一定是具体的广告，这也是因为有些产品的策略并不是直接决定最后的展示。这些看来古怪的咬文嚼字，其实只是做一个铺垫，相信读完本书后面的部分，大家就能更深入地体会做这些调整的原因。

### 3.2.1 投资回报率的数学表达

对这一核心挑战，可以概念上表达为下面的最优化问题：

$$\max_{a_1, \dots, a_T} \sum_{i=1}^T r(a_i, u_i, c_i) \bigg/ \sum_{i=1}^T p_i \quad (3.1)$$

表达式中的 $a, u, c$ 三个变量，分别代表广告、用户与上下文，即广告活动的三个参与主体。 $i$ 代表从第1次到第 $T$ 次之间的某一次广告展示。我们优化的目标，就是在这 $T$ 次展示上的总产出( $r$ )与总投入( $p$ )的比，即ROI。注意这里有一个隐含的假设，即整体的ROI可以被分解到每次展示上。这一假设显然是不太合理的，但是考虑到实际线上决策时，必须对每个展示马上完成计算，所以

从实用出发我们仍然采用这一假设。在实际的系统中，我们会采用频次特征之类的方法解决展示之间相关性的问题。

下面我们再来看看ROI表达式进一步的分解方式，以便引出关于在线广告市场上若干主流计费方式的重要讨论。对一个广告市场中具体的产品形态而言，我们往往能够主动优化的是产出(return)而非投入(investment)的部分，因此，我们主要关注回报的部分。在一次广告展示产生后，有可能发生哪些后续行为呢？首先是点击，这是在媒体网站上发生的；然后是转化，这是在广告主网站上或线下发生的。按照点击和转化两个阶段对回报 $r$ 进行分解，是实践中比较合理且容易操作的方式：

$$r(a, u, c) = \mu(a, u, c) \cdot v(a, u) \quad (3.2)$$

在后文中，我们都沿用这样的符号表示：用 $\mu$ 表示点击率(Click through Rate, CTR)，用 $v$ 表示点击价值(Click Value)，而这两部分的乘积，即定量地表示了某次或若干次展示的期望CPM值，我们称之为expected CPM(eCPM)<sup>2</sup>。请大家特别关注eCPM这个指标，因为它是计算广告中最常被提及，也最有代表性的定量评估收益的指标，本书中有大量的计算问题都是围绕它展开的。实际上，点击价值是还可以进一步分解的，比如对于在线购物的网站，我们可以将点击价值进一步分解为到达率、转化率(Conversion Rate)和客单价的乘积。不过这部分的深入解剖与行业密切相关，而且更多地属于站内运营而非广告的范畴，因此在本书中将制作简要的讨论。与这里的分析相对应，我们认为点击率是 $(a, u, c)$ 三者的函数，而转化率可以近似为 $(a, c)$ 的函数。

### 3.2.2 付费模式产生的原因——谁来估计展示价值

对于大多数广告产品而言，对需要得到给定 $(a, u, c)$ 三元组的eCPM以便进行决策。可是由于广告市场的协作关系复杂，并非每个广告产品都可以对eCPM中的两部分做出较准确的估计。根据eCPM的分解决定哪部分由谁来估计，是广告市场各种计费模式产生的根本原因。我们来看一下三种主要的计费模式：

一、CPM(Cost per Mille)计费，即按照千次展示计费，这里的“mille”是拉丁文“千次”的意思。这种方式，是供给方与需求方约定好千次展示的计费标准，至于这些展示是否能够带来相应的收益，由需求方来估计和控制其中的风险。对于品牌广告，由于目标是较长时期内的利益，很难通过对短期数据进行分析的方式直接计算点击价值，而点击率也因为对于用户接触的核心要求变得不是唯一重要的因素。在这种情况下，由需求方自行根据其市场策略与预算控制单位流量的价格并按CPM方式计费，是比较合理的交易模式。

二、CPC(Cost per Click)计费，即按点击计费。这种方式最早产生于搜索广告，并很多为大多数效果广告网络所普遍采用。这种方式是把点击率的估计交给供给方(或者中间市场)，而把点击价值的估计交给需求方，而需求方通过出价的方式向市场通知自己的估价<sup>3</sup>。这一分解方式的原理

<sup>2</sup> 由于CPM是千次展示的收益，因此eCPM实际上还要乘以1000才能与其相比较，为了表达简单起见，我们在本书中略去1000这一固定系数。

<sup>3</sup> 当然需求方不会完全按照其点击价值来出价，而是会寻求更低的价格以获得套利空间。因此，如何在市场机制上避免广告主积极地调整出价，以促进市场竞争的激烈程度，是竞价体系设计的关键。我们将在第五章中讨论这一问题。

理在于：我们相信供给方的通过其收集的大量用户数据，可以准确地估计点击率；而转化效果是广告商站内的行为，当然他们自己的数据分析体系更能够准确地对其作出评估。

三、CPS(Cost per Sale)/CPA(Cost per Action)/ROI计费，即按照销售订单数、转化行为数或投入产出比来计费，而这些都是按照转化付费的一些变种。应该说这是一种极端的情况，即需求方只按照最后的转化收益来结算，从而极大程度上规避了风险。在这种计费方式下，供给方或中间市场除了估计点击率，还要对点击价值作出估计，才能合理地决定流量分配。这一方式存在两个很明显的问题：一是转化行为并非供给方能够控制，因此也无法进行准确的估计和优化。只有那些转化流程和用户体验相似的广告商组成的广告网络，按转化付费才比较合理，典型的例子比如淘宝直通车；二是存在广告主故意降低转化率，以低成本赚取大量品牌曝光的可能。因此，我们认为这种方式只适合于一些垂直广告网络(Vertical Network)。另外在DSP中，由于需要完全代表广告主利益出价和优化，因此也经常会跟广告主之间按照CPS计费的情形。

四、CPT(Cost per Time)计费，这是针对大品牌广告主特定的广告活动，将某个广告位以独占式方式交给某广告主，并按独占的时间段收取费用的方式。严格来说这是一种销售方式，而非一种计价模式，因为价格是双方事先约定，无需计量。这种方式在欧美市场并非经常采用，不过在中国的门户网站广告中是一种主流模式。CPT还有一种变形，即轮播式CPT，它是将某一广告位的流量按照某一cookie接触到的次数划分成多轮，在其中的若干轮独占式售卖给某广告主，这同样是中国市场很常见的一种售卖方式。CPT这样独占式的售卖虽然有一些额外的品牌效果和橱窗效应产生，但是非常不利于受众定向和程序交易的发展，因而长期看来比例会有下降的趋势。

综合起来看，可以认为对于效果广告，CPC计费方式最有利于发挥供给方和需求方的长处，因而在市场上被广泛接受。而对于品牌广告，由于效果和目的有时不便于直接衡量，可以考虑按照CPM的方式计费。而CPS的计费方式，只在一些特定的环境下才比较合理。

### 3.2.3 广告算法核心任务——如何量化展示价值

公式3.1描述的计算广告核心问题，从算法优化与系统架构两个角度看，分别面临着不同的挑战，也可以进一步分解成一些相对具体的课题，或者从其他相关的研究和工业实践中寻找解决方案，或者在没有合适的方案时做具体探索。

从算法优化的角度看，该问题的特征提取，即对 $a, u, c$ 打标签以方便挖掘的技术，对应产生了受众定向问题；如果不考虑全局最优，则主要依靠eCPM估计，特别是CTR预测来完成每一次展示时的局部优化；如果考虑到量的限制和投放时即时决策的要求，就产生了在线分配的问题；为了在多方博弈的市场中达到动态平衡时的收益最大化，则需要对定价策略做深入研究；为了更全面地采样整个 $(a, u, c)$ 的空间以便更准确地估计点击率，需要用到强化学习(Reinforcement Learning)中的探索与利用(Explore and Exploit, E&E)算法；而在DSP快速发展的今天，推荐算法也被广泛使用在个性化重定向当中。

从系统架构的角度看，我们需要用到实时索引技术服务于广告候选的检索；用到No-SQL的在线存储技术为投放时提供用户、上下文标签和其他特征；大量使用Hadoop这样的分布式计算平台进行大规模数据挖掘；用到最新的流计算平台实现短时用户行为反馈；以及在广告交易环境下



是与其他各个功能性模块打交道，并将他们串联起来完成在线广告投放决策。一般来说，为了扩展性的考虑，我们都采用类搜索的投放机架构，即先通过倒排索引从大量的广告候选中等到少量符合条件的或相关的候选，再在这个小的候选集上应用复杂而精确的排序方法找到综合收益最高的若干个广告。对广告投放机来说，最重要的指标是能同时处理的并发数，以及广告决策的延迟。

二、广告检索，包括图中的Ad index和Ad retrieval两部分。这部分功能，可以类比于人体的心脏，它主要的功能，是实时接受广告投放信息，建立倒排索引，以及在线时根据用户与上下文标签从索引中查找广告候选。实际上，检索技术的重要性体现在所有Web-scale的技术挑战上，也同样是大规模计算广告系统的基础。

三、广告排序，包括图中的Ad ranking和Click modeling两部分。这部分可以类比于人体的大脑，因为它是广告效果优化的关键。其关键技术，在于离线分布式计算平台上的海量数据支持的点击率预测模型的训练。当然线上如何高查询模型需要的特征并进行高效计算，也是非常关键的。另外，在需要估计点击价值的广告产品中，我们还需要一个点击价值估计的模型，或者一些简单的规则，但是不像点击率预测那样有较为稳定统一的建模方法，因此我们在这里主要强调Click modeling。

四、数据高速公路，即图中的Data highway。这部分完成的功能，是将在线投放的数据准实时传输到离线分布式计算平台与流式计算平台上，供后续处理和建模使用，它可以类比于人体的循环系统。由于在进行受众定向建模时，需要用到广告系统以外的其他用户日志数据或第三方合作数据，因此数据高速公路也担负着收集这些数据源的任务。

五、用户日志生成，即图中的Session log generation。从各个渠道收集来日志，需要先整理成以用户ID为key的统一存储格式，我们把这样的日志称为用户日志(Session log)。这样整理的目的，是为了让后续的受众定向过程更加简单高效<sup>4</sup>。

六、商业智能(Business Intelligence, BI)系统，包括ETL(Extract-Transform-Load)过程, Dashboard和Cube。这些是所有以人为最终接口的数据处理和分析流程的总括。因为它担负着对外信息交流的任务，可以类比于人的嘴。由于实际的广告运营不可能完全通过机器的决策来进行，其间必然需要有经验的操作者根据数据反馈对一些系统设置做及时调整。因此，实现一个功能强大，交互便利的BI系统是非常重要的。

七、行为定向，包括结构化标签库(Structural label base), Audience targeting, 以及User attributes的cache：这部分完成的是挖掘用户日志，根据日志中的行为给用户打上结构化标签库中某些标签的过程。这部分是计算广告的原材料加工厂，相当于人体的胃，也因此在整个系统中具有非常关键的地位。

八、上下文定向，包括半在线页面抓取(Near-line page fetcher)和Page attributes的cache：这部分与行为定向互相配合，负责给上下文页面打上标签，用于在线的广告投放中。这里的抓取系统比搜索系统要简单，但也有不太一样的需求，我们后面会介绍到。

九、定制化用户划分，即图中的Customized audience segmentation：由于广告是媒体替广告主完成用户接触，那么有时需要根据广告主的逻辑来划分用户群，这部分也是具有鲜明广告特色的

<sup>4</sup> 关于为什么要这样做，可以参照第四章中行为定向部分中的讨论。

模块。这个部分指的是从广告主处收集用户信息的产品接口，而收集到的数据如果需要较复杂的加工，也将经过数据高速公路导入受众定向模块来完成。这这是广告独特的功能模块，推荐系统和搜索系统是不需要这一功能的。

十、在线行为反馈：这部分指的是一些需要准实时完成的一些任务，包括短时的用户行为标签和短时用户点击反馈等。当然，在利用日志完成这些逻辑之前，必须要进行的步骤是反作弊(Anti-spam)与计价(Billing)。为了组织这样一些前后有依赖关系的数据流加工过程，我们经常选用流式管理平台作为基础设施。需要特别指出，这一部分对于在线广告系统的效果提升意义重大：在很多情形下，把系统信息反馈调整做得更快，比把模型预测做得更准确效果更加显著。

十一、广告管理系统：这部分是广告操作者，即客户执行(Account execute, AE)与广告系统的接口，AE通过广告管理系统定制和调整广告投放，并且与数据仓库交互，获得投放统计数据以支持决策。一般来说，广告系统中只有这部分是面向用户的产品。根据对操作对象开放程度的不同，这一系统有时又有开放自助的需求，在这种情况下，还需要包含相应的财务结算功能。对这部分，读者可以从很多自助式广告产品，比如AdSense，的投放界面中了解这部分的功能，我们在本书中将不做重点讨论。

十二、实时竞价接口：这是广告交易市场实时向DSP发起广告询价请求，并根据竞价结果胜出DSP的程序交易接口。它包括作为需求方时使用的RTBS(RTB for Supply)，以及作为供给方时使用的RTBD(RTB for Demand)。这也是一个广告系统特有的功能模块，在程序化交易原来越流行的今天是广告系统很重要的功能之一。

实际上，并不是每一个广告系统都需要以上所有的功能模块。这样的架构图和模块划分，是为了后文比较容易在各种类型的广告系统之间做架构上的对比。另外需要说明，这样一架构描述，主要是根据竞价广告系统的骨架来进行的，对于其他类型的广告系统，虽然概念上也可以套用，可以术语和习惯上并不直接。为了行文的方便和一致性，本书还是会在这一总架构的基础上讨论各种广告系统。

## **Part II 计算广告的今生前世**





## 合约时代

从这一章开始，我们将会对在线广告具体的产品形态和相应的计算问题展开讨论。我们先从按照合约的方式售卖的广告开始，目的是希望读者能对这类在线广告的一般共性、投放系统、关键算法和其他相关问题能有全貌的了解。

互联网广告业务开始阶段，拥有流量的媒体与需要广告资源的代理商是市场的主要参与者。线下广告的商业逻辑也被照搬到了线上，由广告代理公司和媒体签订协议，确保某些广告位在某时间段为制定的广告商所占有，同时广告商一次性支付广告费用。这种方式与技术的关系并不大，唯一需要用到的系统就是广告排期系统，因此本书将只做简单介绍。

我们讨论的合约式广告的重点，是按CPM计费、担保式投送的受众定向广告。这种售卖方式切合了在线广告的关键优势，又兼顾了线下品牌广告商的传统习惯，因此比较早地产生且被市场接受。面向合约式CPM广告的投放系统，需要解决受众定向、流量预测、点击率预测这三个基本问题，并采用在线分配的方式完成实时决策。关于受众定向和点击率预测这两项技术，其重要程度已经超出了合约广告的范畴，我们会在后续的章节中展开介绍。而在线分配问题，即在一组合约的量的约束条件下，对每个在线到达的展示做投放决策以优化某种目标这一问题，是本章在算法方面讨论的重点。

担保式投送的决策逻辑比较复杂，而且在目前竞价广告越来越重要的市场环境中显得有些古怪，因此有时会被技术人员和产品忽视。然而，此问题的研究却对广告中广泛存在的“量的约束下优化质”这一根本诉求给出了重要的框架，并在各种市场形态中都有变形后的具体表现形式，比如后文中Ad Exchange中的询价优化问题。因此，我们希望能从两个方面对此问题的一般性思路做清晰的介绍：一是在未来流量的情况未知的情形下，如果估计在线分配算法的最差性能；二是在根据历史数据能进行相对合理的流量预测的情形下，如果利用这一信息搭建实用的在线分配系统。

按CPM售卖的合约广告，除了上述的核心算法，还有两项广泛应用的支持技术，即流量预测和频次控制。其中流量预测是在线分配的基础，也是后面竞价广告系统中广泛使用到的功能。流量预测的方法有多种类型，我们将介绍一种基于信息检索技术的基本方案。而频次控制则是广告

主为了展示的有效性提出的控制性要求。关于这两项技术的业务背景，和常用的技术方案，我们也将在本章中做一探讨。

合约广告阶段，需求方的技术并没有得以充分发展。这是由于所有广告投放的执行要求都以合约的形式交由供给方来完成了，需求方并没有技术上优化的空间。随着需求方对深入优化效果的需求进一步发展，产生了按照竞价方式来售卖的广告系统，希望大家通过本章中合约广告的讨论，能够理清这方面的认识。

## 1.1 前合约时代

最早产生的广告售卖方式，是媒体和广告主约定在某一时间段内，用某些广告位的流量为该广告主投送广告，相应的结算方式为CPT方式。这是一种典型的线下媒体广告投放模式，因而在互联网广告产生的早期也自然地被采用。这种方式的缺点，是基本无法做到按受众类型投放广告，因而也无法进行深入的效果优化。不过这种方式也存在一些现实的好处：首先，广告素材可以直接插入媒体页面，并通过CDN加速访问，因而使得广告投放延迟做得到很小；另外，在一些有价值媒体上广告位中长期独占式的购买，有利用形成“橱窗效应”，塑造不断攀升的品牌价值和转化效果；还有一点，这种销售由于可以向广告主提供一些额外的附加服务，比如同一个页面上的竞品互斥，使得高溢价的流量变现成为可能。

CPT售卖还有一种变形的形式，即按照轮播售卖。在这种方式中，同一个用户对同一个广告位的访问，被依次标上一组循环的轮播顺序号，比如{1,2,3}。将其中具有同样顺序号的展示作为一个虚拟的广告位，售卖给广告主。需要注意的是，对某一个用户而言，第一次展示的顺序号不应该设为1，而是应该按相等概率从所有轮播顺序号中随机选取一个，并从此开始累加和循环。这样做是为了保证各个轮播分配到的流量一致。这种轮播的售卖方式，在广告位独占式售卖库存不够，而广告主又需要确定的展现规则保证时，被较广泛地采用，特别是在中国门户网站的品牌广告中。

在CPT售卖的情形下，供给方和需求方的计算需求和技术成分都不太高。需求方的参与者，往往是4A或其他代理公司，对于广告主质和量两方面的需求，都是代理公司的人员通过对媒体广告位的历史经验，以及对广告主业务的了解，通过人工优化的方式来满足。对于供给方即媒体而言，则需要一个与代理公司沟通需求，并在合同确定以后自动地执行合同的广告管理工具，或者我们称为广告排期系统。

广告排期系统的代表性产品，有DoubleClick的DFP，以及中国市场上好耶(Allyes)的类似产品。当然，这些都是这些产品早期的形态，随着按受众售卖广告的方式越来越普及，这些产品的功能也都逐渐演进，从广告排期管理逐渐拓展出其他售卖方式下媒体需要的功能，如果结合了Dynamic Allocation和RTB等功能，也就接近于供应方平台产品了。

## 1.2 受众定向技术的发展

当然，随着受众定向技术的发展，广告位独占式售卖的执行方式也发生了很大的变化。即使某个广告位在某个时间段内全部投放一个广告主的创意，也并不意味着一定要投放同样的一款创意，而受众定向在其中也可以起到很重要的作用。比如某汽车生产商广告主，旗下可能有多个系列的产品，例如小型车、紧凑型车、豪华车、SUV等，而这些车型的潜在购买人群其实也有很大的区别，如果能够对这些系列的受众分别投送相应的创意，可以取得更好的效果。另外，即使在受众上无法区分的情形下，也可以利用频次控制的方式向同一用户递进式地展示一系列创意以达到更好的效果，这方面的技术我们将在本章稍晚加以介绍。这些与受众定向结合的广告位独占式售卖，实际上与其他非独占式的售卖在系统实现上没有本质区别了。

## 1.3 担保式投放的成熟

除了将广告位以独占的CPT方式售卖意外，担保式投送(guaranteed delivery, GD)是另一种常见的合约广告形式。一般来说，GD采用的是CPM结算方式，在合约中明确保证分配给广告主的流量下限。我们从供给方和需求方两方面来看这种售卖方式出现的合理性。媒体从按固定广告位售卖变为按CPM售卖，初衷是为了在受众定向的基础上提高单位流量的变现能力，可是面向的仍然是原来的品牌广告主。广告主按广告位采买时，比较容易预估自己拿到的流量，可是按照人群定向的方式采买，流量有诸多不确定的因素。因此，需求方希望在合约中加入对量的保证，才能放心地采买。

从计算的角度来看，GD中量的要求，实际上对应了我们优化的一组约束条件。因此，合约式广告中最重要的计算问题，是讨论这一在量约束情况下的在线分配问题，我们也将在本章中重点介绍这方面的技术。这一量约束下只得优化问题，从侧面反映了广告三方博弈的本质：广告主是通过媒体来完成自己的市场目标，因此一般会有某种协议来约束媒体方。在上面的GD问题中，这一约束是固定合约的总金额时对量的下限的要求，而在竞价的环境中，则往往会有为了控制花费的预算上限要求。这些问题虽然市场逻辑有很大不同，都可以表示成为类似的数学优化形式。

这里仍然要再次强调我们的观点，那就是广告主对量的要求是切实存在的，而且在某种意义上并不比对质的要求低。GD的这种合约方式，是在供给方完全控制投放过程的情况下，相当合理的一种市场规范。随着在线广告产品的发展，和市场分工的不断细化，媒体直接向广告主以合约的方式保证流量成为诸多选择中的一种。不过可以肯定的是，对大多数广告主，特别是有一定品牌诉求的广告主，一定有某一个市场环节会向其提供保证量的服务。这一点贯穿于在线广告的发展逻辑，也是广告业务的重要特点。

在GD这种合约广告系统中，除了在线分配，还有几个基础性的支持技术。一是受众定向，其作用不言而喻；二是流量预测(Traffic Forecasting)，对历史上各种人群的流量部分的了解对于做好在线分配十分关键；三是点击率预测，这在优化目标的计算时经常用的上。受众定向和点击率预测的问题由于贯穿于计算广告整个历史进程，我们会在后面合适的章节中做完整的介绍。流量预测的框架思路本章中会讨论，但具体实现也会用到后文中的广告索引等技术。本章的重点，

在于深入探讨量的约束下对质的优化这一广告中独特而重要的问题，这对于我们深刻理解广告市场的本质问题相当有帮助。我们将总结一些理论分析成果与在线实用方案，以及框架性的系统实现。

需要说明的是，GD这个说法，是Yahoo!在其显示广告优质客户市场上的一个具体的产品名称，并非市场上通用的术语。我们认为这个词比较准确地描述了合约广告中量的约束与质的优化的关系，因而用它来代表这一类合约广告。当然，这里的担保并不一定是展示量上的担保，也可以是点击数或者到达数等指标上的担保。只要是在某个量上有下界式的约束，就可以认为是GD一类的问题。

## 1.4 代表公司

### 1.4.1 担保式投放:Yahoo

### 1.4.2 受众定向:Criteo

## 1.5 合约广告的优缺点

在了解了合约的广告交易方式，特别是保量销售的GD方式的原理和技术以后，我们可以回顾一下这种交易方式的优缺点，并探讨一下市场上越来越多的按竞价方式售卖的广告产生的原因。竞价广告的概念要在后文才介绍，但读者可以大致理解成不保量，只根据广告主出价来优化市场收益的广告运营方式。如果与GD广告销售模式对比，也可以归类为NGD( Non-Guaranteed Delivery)的广告销售模式。

从供给方或广告市场方来看，合约广告和竞价广告的对比，可以类比于计划经济和市场经济的区别。在合约广告的情况下，所有的量的保证和质的优化，都是由媒体方的Ad Server 来统一完成，这也直观地反应在GD合约广告比较复杂的planning和allocation过程。而在竞价广告的情况下，市场只负责制定竞价和收费的规则，而各广告主量的保证完全采用市场竞争的方式来完成。在这种情况下，市场方需要仔细设计宏观竞争机制，但是不一定需要实现合约广告那样的allocation功能。

从需求端来看，合约广告的采买方式对广告主来说缺乏透明性，唯一能做的就是在合约的层面预先约定好一些最关心的利益条款，但是很难做深入的优化。不过，合约的采买方式也有一定的好处，特别是对量的保证可以有预先的约定，这对于品牌性质较强的广告活动来说比较有意义的。

## 竞价时代

随着受众定向技术的发展，以及搜索业务变现的要求，在显示广告和搜索广告中都产生了竞价这种新的交易模式。如果我们对比前面的合约广告，竞价交易模式的本质，是将量的约束从交易过程中去除，仅仅采用“价值高者得”的决策方案来投放广告。竞价顺应了定向广告向精细化发展的趋势要求，也为大量无法用合约形式售卖的剩余流量找到了可能的变现渠道，使得大量中小广告主的参与在线广告的可能性和积极性大大增强，也是的在线广告的商业环境与线下广告产生了本质的区别。

在竞价的 market 环境中，产生了广告网络这种批量采购各种媒体剩余流量，然后主要按照点击付费的方式售卖给需求方的产品形式。这种产品形态对于提高整个在线广告市场的流动性发挥了很大作用。如何设计合理的 market 规则和定价策略，使得整个市场的竞争更加充分，对于广告网络运营者的收益有相当大的影响。这方面有关宏观 market 上的讨论，在竞价时代显得非常重要，也成为在线广告领域得到充分重视和深入研究的问题，我们在本章中会对这方面的理论和实用方法做适当的回顾。这也是合约广告系统中不存在的问题，充分认识定价策略背后的意义，对于把握竞价广告系统的本质非常关键。

竞价广告中，计算的作用更加突出，这一方面是因为竞价本身需要对  $eCPM$  做尽可能准确的估计，也是因为中小广告主的规模使得计算的效率要求很高。具体而言，有两项技术非常关键：一是当大量中小广告主参与市场后，如何根据广告的一些业务要求设计更高效合理的索引和检索技术，这部分将是本章展开讨论的重点；二是为了完成  $eCPM$  估计，对给定  $(a, u, c)$  组合上的点击率预测技术。另外，由于大量中小广告主的参与，竞价广告系统对  $eCPM$  的估计往往面临统计不足的情形。如何设计有效的策略探索未知领域，并利用探索的结果更好的估计  $eCPM$ ，也是一个非常有趣的问题。由于  $eCPM$  估计本身是整个广告生态系统中非常核心的计算问题之一，我们将在下一章中对这方面的算法做专门的探讨。

对应于广告网络的产生，需求方的产品和技术也在发生变化。关键的变化有两点：一是由面向广告位采买变成面向人群的跨网络采买；二是帮助广告主在竞价环境中完成量的要求，这一点是竞价市场不再直接保证的。能达到这些需求方目的的产品，我们叫做交易终端。在交易终端

里，在量的约束下完成ROI的优化的问题依然存在，并且变得比在线分配问题更加困难，这也成为高端的采买平台提供的服务之一。

搜索广告的商业形态，与显示广告网络本质上是一样的，都是按照点击结算的竞价广告市场，因此本章中有关宏观市场和检索技术方面的讨论同样适用于搜索广告。不过由于搜索问题的特殊性，以及搜索广告重要的商业地位，对它的技术探索有很多深入定制化的部分，这些特殊的地方我们将在第七章中做专门介绍。

## 2.1 竞价的位置拍卖理论简述

竞价广告是象拍卖那样销售广告。也就是根据广告主的出价，以及由此计算出的期望收益，决定谁可以得到某次展示的广告位。在竞价广告产生的开始阶段，出价是广告主阶段性调整的，而到了广告交易市场阶段，广告主可以对每次展示实时调整出价<sup>1</sup>，这两种方式从拍卖市场的宏观角度看，没有本质差别。

让我们先来看看怎么具体描述竞价广告问题，并从宏观市场的角度了解一些非常重要的结论。假设有一组广告位可以被占用，将这些广告位按照其经验价值排名，分别记为 $s = 1, 2, \dots, S$ ，对Banner广告而言，这里的 $S$ 一般为1。在某次广告请求中，有一组广告 $a = 1, 2, \dots, A$ 出价参与拍卖，每个广告的出价记为 $b_a$ ，系统将前 $S$ 个高出家的广告依次放到前面排序好的 $S$ 个广告位上，这样的问题称为位置拍卖(Position Auction)。根据前文的讨论，当某个广告 $a$ 被放在 $s$ 位置上时，其期望收益即eCPM为 $r_{as} = \mu_s v_a$ 。这里我们做了一些假设，比如点击率 $\mu$ 仅与位置 $s$ 有关，而点击价值 $v$ 仅与广告 $a$ 有关，这些假设在搜索广告给定某具体关键词的情形下，可以说基本合理，对于显示广告的情形虽然非常粗略，但并不太影响对竞价问题宏观市场的讨论。

围绕位置拍卖最重要的研究，是广告的定价(Pricing)问题。定价问题探讨的是在一次具体的拍卖交易中，给定各参与者的出价，以及他们的期望收益，如何对最后获得某个位置的广告商收取合适的费用。乍听起来，讨论这个问题有点多余，按照广告主自己的出价收取不就可以了么？为了阐释清楚此项研究的动机，我们先来看看下面的例子：假设有某个单位置( $S = 1$ )的广告机会在竞拍，开始有两个广告主参与，甲出价1元，乙出价2元，当然甲赢得了此次竞价，如果按照其出价来收费，市场就向他收取2元的费用。在广告市场里，这一拍卖机会还会重复出现(对应于不断产生的展示)，因此广告主可以也存在调整出价的机会，假设乙在发现自己2元钱能拿到流量以后，自然会想到，是不是可以调低出价，用更低的成本拿到流量？乙将一直不断尝试，知道把出价调低到1.01元，发现继续调低就拿不到位置了。于是系统稳定在甲出价1元，乙出价1.01元。此时假设又有一个广告主丙加入竞争，并希望赢下此广告位，那么以此类推，他在不断调整后将会出价1.02元，市场的收入也就是1.02元。我们有可能通过调整定价策略来影像系统的总收益么？这样的机会显然存在。比如我们在甲出2元，乙出1元参与竞价时，并不对获胜的甲收2元，而是收取其下一名即乙的出价1元，那么甲就没有动力调低其出价了。那么当丙加入时，就需要

<sup>1</sup> 实时竞价的具体产品和技术将在第八章中介绍。

出价2元以上才可以赢得竞价，市场的收入也就变成了2元(不论丙出价多少，我们都按其下一位即甲的出价来收费)。这个简单的例子告诉我们，在广告这样的参与者可以针对同一个标的物不断调整出价的拍卖环境中，通过聪明的定价策略，完全可能为整个市场创造更高的收益以及其他好处。

在定价问题上，我们在微观上的直觉未必可以推广到宏观市场。让我们来看看如何对上面的位置拍卖问题作进一步探讨。从整个市场的角度来看，我们重点需要研究的是市场处于稳定状态下的收益和其他情况。而所谓稳定，指的是整个竞价系统处于纳什均衡(Nash Equilibrium)状态，也即每个广告商都通过出价得到了最符合自己利益的位置。对某一次位置竞价来说，其对称纳什均衡(Symmetric Nash Equilibrium)状态可以表示为下式：

$$\begin{aligned}\mu_s(v_s - p_s) &\geq \mu_t(v_s - p_t), \forall t > s \\ \mu_s(v_s - p_s) &\geq \mu_t(v_s - p_{t-1}), \forall t < s\end{aligned}\quad (2.1)$$

注意这里的下标意义有所调整，这里的 $v_s$ 指的是排在 $s$ 位置上的广告上的点击价值，并非 $s$ 位置带来的点击价值；而 $p_s$ 指的是市场向排在 $s$ 位置上的广告收取的费用，即定价。这一均衡状态的意义很容易理解：对于最终位置排名竞价结果中的每一条广告，其收益都比排在其他位置上要高。显然，在这样的状态下，每个广告商都达到了自己最优的状态，整个系统也就随之稳定下来。

在公式4.1中，市场方能够调整的策略，只有 $p_s$ 的确定方式，也就是定价策略。随着定价策略的不同，市场达到稳定状态时的宏观收益情况和稳定的程度都有所不同。因此，有关竞价市场宏观性质的研究，主要目的是寻找更好的定价策略以优化整体收益。关于位置竞价问题纳什均衡状态的数学分析，由于与本书的产品和技术重点有一定差距，因此不再介绍这方面的内容，有兴趣的读者可以阅读文献[43]。我们只介绍两种竞价市场用到的定价策略，以及与宏观市场相关的几个策略问题。

### 2.1.1 GSP

在线广告竞价市场最常见的定价策略，是广义第二高价(Generalized Second Pricing, GSP)策略；另外有一种VCG(Vickrey - Clarke - Groves)定价策略，虽然理论上比GSP更好，但是由于原理较复杂，向广告主解释起来有难度，因此在实用系统中采用的并不多。下面我们分别介绍一下这两种定价策略。

先来看看是么叫第二高价(Second Pricing)。所谓第二高价，指的是在位置拍卖中，向赢得某个位置的广告商收取其下一位广告主的出价。如果是按照CPM结算，那么第二高价就可以直接应用了。需要考虑的问题是，在CPC计算的竞价广告系统中，广告主的出价是针对点击的，而竞价是针对eCPM的，因此要对两者做一下换算以实现公平的第二高价，这就是所谓的广义第二高价，其定价公式为：

$$p_s = \mu_{s+1}b_{s+1} / \mu_s + \Delta \quad (2.2)$$

如果将等式两边同时乘以 $\mu_s$ ，可以看出广义第二高价实际上仍然是eCPM上的第二高价。读者也可以自行验证，在广义第二高价情形下，对某广告主的定价是一定不会大于其出价的。实际上，

这种定价策略也同样适用于CPS结算的竞价市场，并且只需要将公式中的 $\mu$ 换成 $\mu v$ 即可。公式最后的 $\Delta$ ，一般为广告系统结算货币的最小单位，比如1美分，这是一种历史惯例，也在某种程度上让广告主心理上感觉更加公平。

对比上面提到的竞价广告的例子，可以直觉地知道，广义第二高价可以基本上满足市场稳定情形下收益优化的需求。不过需要说明的是，这种满足只是直觉上的而非理论意义下最优的，这一点读者可以对比下面介绍的VCG定价策略来理解。虽然并非理论上最优，广义第二高价却有着实现简单、容易向广告主解释等诸多操作中的优点，因此在实际的竞价广告系统中是最主流的定价策略。

### 2.1.2 VCG

VCG定价是Vickrey, Clarke和Groves在研究竞价系统均衡状态时得到的一种理论上较为优越的定价策略。其基本思想是：对于赢得了某个位置的广告主，其所付出的成本应该等于他占据这个位置给其他市场参与者带来的价值损害。在这一原则下，VCG的定价策略可以表示为：

$$p_s = \sum_{t>s} (\mu_{t-1} - \mu_t) v_t \quad (2.3)$$

这种定价策略直觉上的合理性很容易理解。实际上，理论分析表明，VCG定价策略的优越性体现在如下几个方面：首先，在这种定价策略的稳定状态下，整个市场是truth-telling的。所谓truth-telling，可以理解为每个广告主都找到了自己的最优状态。其次，相对于其他的定价策略，这种定价向广告主收取的费用是最少的。

虽然有以上诸多的优点，VCG定价在在线竞价广告中却并不是一种主流的方式。究其原因，主要是由于这种定价方式的逻辑过于复杂，比较难以向广告主解释清楚；另外在广告主和媒体存在博弈关系的情形下，媒体是否正确地计算了“给其他市场参与者带来的价值损害”也很难验证。不过这种定价方法也有其市场空间，有的广告厂家，比如Facebook，在实际的系统中采用了这一定价方法。

## 2.2 按点击付费的竞价系统之崛起

### 2.2.1 广告联盟模式

对显示广告而言，合约式的售卖方式必然无法消耗所有的库存，实际中由于为了控制售卖比例已获得更高的品牌溢价空间，这样卖不出去的广告流量可能比我们想象得更多，这部分流量，我们称为剩余流量(Remnant Inventory)。竞价交易模式的产生，为这部分流量提供了变现的机会，于是产生了广告网络这一产品形态。

广告网络的基本业务，是批量聚合各媒体的剩余流量，按照人群或上下文标签的流量切割方式售卖给广告主。由于是按人群售卖，广告网络会极力淡化广告位的概念。另外，广告网络一般不会向广告主约定或保证能够买到的量，而只是根据变现能力来决定每次展示分配给哪个广告



主。与合约式的广告系统不同，广告网络使得为了保证合约而设计的复杂的在线分配算法必要性大大降低(如前文所述，在优化的意义上，在线分配技术对于广告网络依然很有价值，但已不再是必需的技术)，使得其中的计算技术可以把精力集中在对eCPM的估计上。从商业角度来看，广告网络的销售模式与合约的方式相比，也有两点优势：首先是无需再满足广告主品牌独占的要求，这使得让国美和苏宁同时参与同一个人群的竞价，提高市场流动性成为可能，而在合约广告中，这一点是很难做到的。其次，由于广告网络不再保证量，二是根据实际消耗来结算，一般来说财务上采用广告主先充值的方式，这区别于合约广告投放结束后计算的方式，结果使得广告网络运营方的现金流状况大为改善。

广告网络的存在CPM、CPC和CPS等不同的结算方式，不过最主流的方式是CPC。从计算的角度，我们非常有必要分析一下为什么CPC是最合适的。首先从Demand来看，既然是各种媒体的不同广告位聚合在一起售卖，广告主无法知道每个媒体上广告的具体位置。而根据我们前面的讨论，位置对于广告的曝光效果影响巨大，因此实际上广告主根本无法评估每次展示的出价，而在点击上出价，这个问题就没那么严重了。另外从Supply来看，由于淡化广告位的概念，并且聚合了多个媒体的流量，广告网络可以接触到同一个用户比较丰富的网络行为，并且知道每次展示所在的媒体与广告位位置，所以比广告主更容易估计点击率。根据第一章中的讨论，由广告网络负责估计点击率，需求方根据对点击价值的估计来出价，是最合理的市场分工。

一般来说，广告网络中的投放决策过程对媒体而言都类似于一个黑盒子：媒体只需要在广告位上加上广告网络的JS代码，而无需也无法关心每次展示的投放结果。但是对于一些有特殊业务需求，或者特殊数据来源的媒体或媒体组合来说，有时候希望能够直接从广告网络的广告库中挑选广告，并能够创造比广告网络自动挑选更多的价值。因此，某些广告网络也会对一部分合作供给方开放广告库供其自行挑选，广告网络的这种运营模式，我们可以称为联盟(Affiliate)模式。

### 代表案例介绍:Google Display Network, 阿里妈妈

实际的广告网络有两种不同的业务方向，一中是广泛承接各种广告库存，并面向所有品类的广告主进行服务，我们称之为水平广告网络；另一种则专门服务于某一种类型的广告主，例如电商、游戏等，并寻找相关的媒体资源来搭建网络，我们称之为垂直广告网络。下面我们分别举一个例子来介绍这两种类型的广告网络。

Google自从AdSense进入上下文广告领域后，又于2008年收购了显示广告领域的巨头DoubleClick。在整合旗下Utunbe、Google Finance、以及AdSense等一些广告资源基础上，退出了Google Display Network这一显示广告领域最重要的广告网络之一。

由原来AdSense这种为搜索广告的衍生产品存在的地位，进化到独立的GDN，表现出Google对显示广告领域的重视和独立发展的决心。

除了GDN，Google还通过收购等方式，在显示广告领域从供给端到需求端进行了全系列的布局。从LUMA的显示广告市场图中，我们把Google的部分产品和子公司标示出来，可以更清晰地了解这一布局：

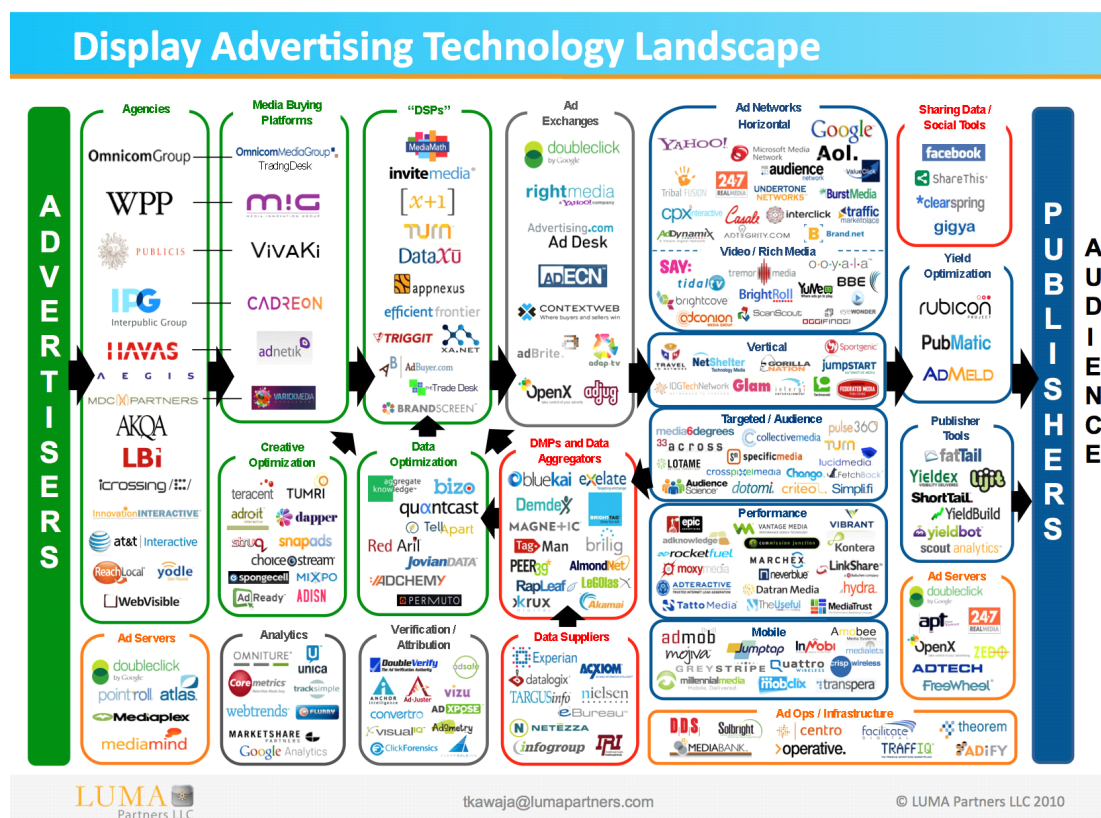


Fig. 2.1. Google在显示广告领域的布局

## 阿里妈妈联盟

阿里妈妈是阿里巴巴集团旗下的广告产品之一，目的是为了利用淘宝和天猫网以外的网站为阿里巴巴平台上的卖家带来流量和转化，并且根据转化的比例给网站一定的佣金。由于阿里妈妈平台上的广告主基本上都是淘宝和天猫的网店，因此可以把它看成是一个面向电商类型广告主的垂直广告网络。

加入阿里妈妈网络的媒体，主要是为淘宝等商家带来流量，因此阿里妈妈把它们称为“淘宝客”。淘宝客网站有两种方式利用阿里妈妈联盟中的广告需求变现：一种方式是像普通的广告网络那样，直接在广告位上加一段阿里妈妈的广告投放代码，由阿里妈妈的广告服务器来进行广告决策；另一种方式，是淘宝客网站通过API拿到相关的广告，并自行决定在某个页面、或对某种用户展示的什么样的商品，这种方式是其他的广告网络没有的。乍看起来，由于阿里妈妈掌握的用户数据和运算能力远远超过一个个孤立的淘宝客网站，让阿里妈妈负责广告决策，应该可以达到更好的eCPM，不过由于电商广告网络特殊的性质，一些购物导航或购物分享类型的个人博客或网站，可以通过后一种手段，利用软文的形式非常有针对性和全面地介绍和宣传某种产品，

而这种软文广告能达到的效果远远高于普通的在线广告。因此，这种将广告库开放，由淘宝客网站自行选择广告的方式，其变现能力在充分优化后也相当可观。不过由于后一种方式无法规模化，并且通常只在转化流程相当完善的情形下才有意义，因此并不见得适用于一般的广告网络，但在电商垂直广告网络中是不错的方式。

由于在阿里巴巴的生态系统中大量天然广告需求的存在，并且可以收集到用户非常接近于转化的高价值行为数据，阿里妈妈的精准广告投放能力和市场竞争程度，都是非常高的。

### 2.2.2 搜索广告模式

从商业逻辑和产品形态上看，搜索广告可以认为是广告网络的一个特例。它是以上下文查询词为粒度进行受众定向，并按照竞价方式售卖和CPC结算的广告网络。从检索、点击率预测等关键环节的技术上看，搜索广告与显示广告网络并没有本质的区别。不过，由于搜索广告本身的变现能力，即eCPM远远高于显示广告，因此其市场重要程度也就得以彰显。对与搜索广告的一些独特问题和算法的研究，也就受到了非常高度的重视。在本章中，我们将在前一章讨论的竞价广告系统的基础上，着重探讨搜索广告相关的一些特殊问题。

搜索广告与一般广告网络最主要的区别，是上下文信息非常强，因此用户标签的作用受到很大的限制。因此，关于搜索广告的研究，有两个技术上的重点：一是查询词的扩展，即如何对简短的上下文信息做有效的拓展，由于搜索广告的变现水平高，这样的精细加工是值得而且有效的；二是根据用户同一个搜索session内的行为对广告结果的调整，因为围绕同一个目的一组搜索，往往对于更准确地理解用户意图有很大帮助。本章中，我们将围绕这两个重点问题，概要性的介绍搜索广告中的思路与具体技术。

尽可能快速地将用户搜索行为反馈到广告决策中的需求，催生了流式计算平台。这一技术与Hadoop这样的离线分布式计算平台相呼应，可以更有效地完成计算广告中的数据处理任务：对于那些长时期积累的海量用户行为数据，我们可以用Hadoop进行离线的、实时性不强的定期挖掘；对于那些短时用户行为，可以用流计算平台进行半实时加工和快速反馈，以便迅速指导线上决策。这样的计算分工和系统架构，其实不仅适用于搜索广告，也适用于显示广告网络里的短时受众定向和短时用户行为反馈。

对搜索广告这个产品，不同搜索引擎提供商有不同的称呼，比如Paid Search, Search Ad, Sponsored Search等。这些词汇概念上非常相似，但也略有差别，个人比较倾向于采用“Sponsored Search”这样的说法，而“Paid Search”有时会让读者对是谁付费产生误解。至于“Search Ad”，实际上还应包括搜索引擎中的其他广告形式，比如百度品牌专区，因而并不是本章讨论的狭义的完全竞价的搜索广告网络。

本章中讨论的搜索广告，主要以通用搜索引擎为蓝本。实际上，很多垂直类搜索，特别是电子商务类搜索也有很强的广告变现能力，但产品形态可能会有所不同，大家可以有选择性地与下面讨论的问题来对照。另外，插入在自然搜索结果中的竞价排名模式，也不在我们讨论的范围内。

如下图所示，搜索广告的展示区域，一般来说分为北(North)、东(East)、南(South)三个部分。北侧和东侧的所有位置，构成位置拍卖的标的物集合，但是并不是一定要在这些位置上出广

**Fig. 2.2. 搜索广告竞价位置示例**

搜索广告的受众定向标签，即是上下文的搜索词。由于搜索词非常强地表征着用户的兴趣，搜索广告可以进行非常精准的定向。相对这样的上下文信息，根据用户历史行为得到的兴趣标签重要性大打折扣，这一方面是因为其信号远不如上下文搜索词强烈，另一方面是因为用户这样强

烈兴趣的任务是绝不能被打断的(参见第二章中广告有效性原理部分)。这是搜索广告区别于显示广告网络的最大特点。

既然搜索词的重要性极高,粒度又非常细,如何根据广告主的需求对其关键词进行合理的拓展,找到那些相关而又效果不错的关键词,这对于需求方和供给方来说都有很大意义:需求方需要通过扩展关键词以拿到更多流量;供给方则需要借助此来变现更多流量和提高竞价的激烈程度。因此,查询扩展(Query Expansion)是搜索广告的重要技术。搜索广告的查询扩展,与搜索中的这个问题,有相通之处,又有一些显著的区别。

虽然按照用户历史行为做定向不适用于搜索广告,用户在一个session内的一系列查询,还是会对准确理解用户意图大有帮助。另外,前一章介绍的点击反馈特征,也存在着快速更新的需求。要达到这两个目的,系统上的挑战要大一些,原先那样基于Hadoop的离线挖掘模式就不适用了。正是这一需求的存在,催生了流式计算(Stream Computing)的技术平台,目的是利用用户短时的行为快速、连续地得到一些统计信息,并反馈给线上决策系统。流式计算对于广告中许多问题都非常有用,我们以搜索广告为出发点,拓展讨论到这一技术在显示广告短时用户行为反馈中的用处。

## 代表案例介绍:Google, Baidu

### 2.2.3 广告购买平台

竞价广告市场的产生,对需求方的技术也提出了新的要求:原来通过直接与媒体签订保量合约的采买方式,需要让位于通过参与竞价帮助广告主完成量与质需求的采买产品。具体地说,这样的产品需要具备一项基本功能:即按照广告主预算跨媒体和广告网络一站式采买某种人群的广告投放机会。另外,还可以具备一项高级功能,即机器驱动的自动ROI优化功能。这样的产品形态,我们称为媒体采买平台(Media Buying Platform, MBP),与之类似的概念,还有所谓交易终端(Trading Desk),可以认为是同一类的产品。

从市场发展来看,许多媒体采买平台都是4A或其他代理公司为了适应市场技术形态的变化,收购或者孵化出来的子公司,这从图\*中这两部分产品之间的连接线就可以看出来。值得注意,在竞价广告市场上,广告主量的需求仍然是存在的,因此保量的合约并不是消失了,而是由代理与媒体之间下沉到了广告主与媒体采买平台之间。

媒体采买平台的技术挑战主要在于ROI的优化部分,要合理地选择需要的受众定向条件,并在每个人群上优化出价,以保证量的要求和优化收益,是一个复杂的组合优化问题。由于无法具体控制每一次展示,其困难程度要高于后文实时竞价的情况下的DSP优化。这方面的市场需求,以搜索广告网络中的优化问题最为突出,因此我们将在下一章中具体介绍相关技术。

## 非RTB的流量的ROI优化简述:SEM,SEO

搜索广告对应的需求方问题,即通过竞价采买搜索引擎关键词来做广告的问题,我们称之为搜索引擎营销(Search Engine Marketing, SEM)。从本质上看,SEM与前文提到的媒体采买平台面临的

挑战类似，即流量选择(此处为关键词选择)和出价。不过对SEM来说，各个标签即关键词的流量是互不重叠的，这与显示广告受众定向标签之间可以做复杂的布尔运算不同。

关键词选择和出价两个问题都有相当的难度。首先来看关键词选择，如果广告上提供一些种子词(这一点往往是比较容易做到的)，那么这个问题看起来非常类似于4.8.7中的查询扩展问题。但是从市场地位上来看，又有着相当大的不同：搜索广告运营方进行查询扩展的目的，是为了提高自己的营收；而需求方进行关键词选择的目的，是为了提高广告主采买流量的ROI。这两个目标显然有着很大的区别，因此广告主更信赖代表需求方利益的SEM公司。不过SEM公司由于数据来源的问题，通过4.8.7中讨论的各种方法都需要长期的数据积累才能实现。而出价的问题，传统的认识是广告主根据到达率、自己网站的转化率和客单价来估计点击价值，并以此点击价值为参考来出价。不过这种认识忽略了一个问题，那就是各个关键词的转化率、客单价和市场竞争水平都有很大的区别，因此只有在不同关键词上做不同的出价，才能更深入地优化整体采买的ROI。由于搜索广告的定向粒度很细，可以看出关键词选择和出价都是规模很大的优化问题，而且由于广告网络半封闭的竞价机制，数据的反馈和调整都不够及时，因此这一优化问题技术挑战是比较大的。这种面向广告网络中非RTB流量的ROI优化，也成为广告技术产品的一个重要方向。

### 代表案例介绍: SEM公司,SEO公司,Efficient frontier

Efficient Frontier开始是一家专门从事SEM优化的广告技术公司，后来也涉足显示广告网络的优化领域。现在已经被Adobe旗下的Omniure所收购，其产品成为Adobe Digital Market Suite中的一部分，称为Adlens。

从Efficient Frontier这家公司的名称，就可以大致了解到其技术方向。Efficient Frontier，是量化金融领域的术语，这一属于产生于著名的投资组合选取(Portfolio Selection)理论[34]，这是通过计算的方法确定一个投资组合中各个品种的投资比例，以达到期望收益最优情况下风险最小的理论。在各个投资期望收益水平下，相应的最小组合风险对应的曲线，就称为Efficient Frontier。图\*中给出了Efficient Frontier的一个示例。

如果将搜索广告里的标的物，即关键词，类比于股票，那么在一定预算的限制下分配各个关键词上的投入，以求达到整体ROI的最优，从框架上看非常类似于上述的Portfolio Selection问题。因此，Efficient Frontier公司按照这一思路，将量化金融里的Portfolio Selection理论与计算广告技术结合起来，创造性地发展出了大规模的关键词出价优化技术。

## 2.3 应运而生的计算广告核心技术

### 2.3.1 广告系统的检索技术

竞价广告市场中，大量中小广告主的参与，以及复杂的定向条件，对检索技术提出了更高的要求。倒排索引是搜索引擎的关键技术，而广告的检索上也采用这样的框架。但是广告的检索问题

也有一些自身的特点和需求，我们需要讨论一下这些需求下一些专用的检索技术。在合约式广告系统中，广告主数目一般来说不算太大，因而采用一般的检索方案完全可以满足要求；然而在服务大量中小广告主的广告网络中，探讨高效且精准的检索技术，有非常切实的产品意义。

第二章中介绍的基本倒排索引技术在广告问题上遇到了两个挑战：一是广告指定的定向条件组合，可以看成是一个由与或关系连接的布尔表达式，这样的布尔表达式的检索，很显然与上面搜索引擎对文本的“bag of words”假设不太一样；二是在上下文和用户兴趣信号比较丰富时，广告检索中的查询可能相当长，甚至会由上百个关键词组成，这种情况下的检索，也与搜索引擎之中主要由1-4个term组成的query情形有区别：试想你把100个关键词同时输入到搜索框中，返回的结果会是你想要的么？根据这些切实的需求，我们来讨论一下在线广告系统中的检索技术。

### 2.3.2 点击率预测模型

计算广告中一项非常重要的技术是点击率的预测。在竞价广告的环境下，估计点击率的目的是与点击价值(对广告网络而言即是CPC计费时的bid)相乘得到eCPM，再根据eCPM对广告进行排序。从这里可以看出，广告的点击率估计与搜索里的排序问题有关联，但又有明显的不同：点击率估计不能像搜索排序那样只要求结果排序的正确性。这—是因为点击率需要乘以bid才得到最后的排序，二是因为对一些需求方技术平台而言，需要尽可能准确地eCPM用于出价。因此，作为各种广告系统中通用的一项技术，点击率预测更适合被建模成回归(Regression)问题而不是排序问题。





## 实时交易时代

在线广告发展到竞价阶段，可以说基本的计算格局已经建立，大多数重要的计算问题，也都浮出水面并得到研究。然而，随着需求方进一步优化流量的目标进一步加强，广告网络在产品形态上已经无法完全满足需要。市场的发展方向，是向需求方进一步开放。具体地说，除了允许广告主按照已经定义好的用户划分来购买，进一步提供广告主自行选择流量和在每次展示上独立定价的功能。这样的功能，必然要求竞价这一过程在每次展示时实时进行，也就是实时竞价(Real Time Bidding, RTB)。

RTB的产生，使得广告市场向着透明的比价平台的方向发展，这样的平台就是广告交易平台(Ad Exchange)，其主要特征即是用RTB的方式实时得到广告候选，并按照其出价简单完成投放决策。与广告交易平台对应的采买方，我们称为需求方平台(Demand Side Platform, DSP)。在这样的交易市场中，需求方对于流量的选择和控制能力达到了极致，因此其技术和算法的挑战也相当大；而供给方的变成了简单的比较平台，因此结构上相对清晰简单。由于实时竞价的功能需求，广告交易市场解决供给方和需求方用户身份对应的问题，这需要用到cookie-mapping的技术。

虽然供给方在广告交易中遇到的算法问题不多，但还是要解决好带宽资源的限制下，用更少的询价请求完成尽可能高效的变现这一问题。从优化角度来看，这很像前面的在线分配问题，同样的带约束优化思路也仍然可以适用。

从需求方来看，定制化的用户划分能力使得广告主可以象优化自己的推荐系统那样优化广告购买，唯一的区别是这个推荐系统是放在站外的。出价需求的存在和广告主预算范围内的套利，要求DSP具备点击率预测、点击价值估计、流量预测、站外推荐等多方面的运算能力。除了站外推荐，广告市场新的发展趋势是根据广告主提供的种子用户，利用海量的媒体数据为广告主找到行为相似的潜在用户。这样的一些定向方式，都是针对 $(a, u)$ 组合的定制化用户划分，它们使得在线广告的受众定向精准程度达到了前所未有的水平。

在需求方的利益得到了充分的保证以后，媒体的变现手段也发生了相应的变化。到现在为止，媒体至少有四中常用的广告变现选择：担保式投送的合约售卖方式、自营广告网络、托管给其他广告网络、通过RTB变现。如果动态地选择这四种模式中变现价值最高的那种，以最大化媒

体收益，这是供给方面临的市场需求。在这样的需求驱动下，产生了供给方平台这样的完全优化媒体利益的产品。

当在线广告市场进入到这个阶段以后，程序交易(Programmatic Trade)的方式已经成为举足轻重的力量，它使得整个在线广告市场越来越向着数据驱动、计算导向的方式前进。我们在本章中也对程序购买的若干种主要方式，以及对应的供应方和需求方产品做了一下小结，并与合约导向的优先销售方式做了对比。准确理解每一种交易方式的特点、长处和适用场景，无论是对于供给方更好地变现流量，还是需求方更准确有效地投送广告，都有相当重要的意义。

### 3.1 实时竞价的广告交易平台

关于广告交易平台，即Ad Exchange产生的原因，市场上存在着不同的解释。最早的Ad Exchange产品RightMedia的初衷，是在纽约的一些广告代理公司之间交换剩余流量。因此，最初的Ad Exchange也并不是采用RTB的交易方式。但是非实时竞价的方式并不能从根本上满足流量交换的需求，因为剩余流量只有按照需求方最精确的定向方式来交换，才能获得最高的价值。所以当RTB产生以后，迅速成为Ad Exchange的标配甚至最主要的功能。

#### 3.1.1 广告交易平台架构

#### 3.1.2 代表案例介绍:rightmedia, Google adx

Ad Exchange从历史发展来看，有两种侧重点不同的产品方向。一种是早期的Rightmedia模式，主要产品目标是在一些代理商之间交换流量，因而是一个只有在交易双方有协议约定的情形下才可以竞价采买市场；另外一种是DoubleClick的模式，它提供的是一个完全透明，理论上任何DSP都可以对任何流量竞价的公开交易市场。对于这两家具有代表性的Ad Exchange，我们在下面做对比性的介绍。

##### Rightmedia

Rightmedia成立于×年，后被Yahoo!所收购。

##### DoubleClick Adx

Google在收购了DoubleClick以后，于\*年推出了基于实时竞价的Ad Exchange，称为DoubleClick Adx。

## 3.2 需求方平台(DSP)

对Ad Exchange相对应，以RTB方式购买广告的产品形态就是需求方平台(DSP)。这一产品的核心特征有两个：一个当然是支持RTB方式的流量购买，另一个是需要支持定制化的用户划分。大家需要仔细体会，这两个核心特征其实是同一问题的两个方面：为了能够按定制化的用户划分采买广告，需要市场开放竞价接口；而如果按照供给方定义的用户划分来采买，那么像广告网络那样的非实时竞价就基本可以满足。

那么什么是定制化的用户划分呢？从受众定向的角度看，那些形如 $f(a, u)$ 的定向方式，即广告主相关的定向方式，就是定制化用户划分。举两个例子来直观地理解一下：第一个例子，假设某电商网站准备进行一次面向其老客户的广告投放，这里的“老客户”就是一种定制化用户划分，显然只有该电商自己才能找到这个用户群，而任何媒体或广告网络，无论数据能力有多强，也无法加工出这样的标签。第二个例子，某银行希望通过广告接触到自己信用卡的潜在客户。要找到这些潜在客户，我们需要以银行现有的客户作基础，分析其行为和人口属性有哪些特征，然后再根据这些特征去拓展可能的潜在客户。这一过程既需要需求方的数据，又需要媒体数据，而其加工出的人群仍然是与广告主相关的，因此也属于定制化用户划分。

DSP与其它广告产品的相比，多了定制化用户划分功能，即图中的Customized Audience Segmentation部分。这部分通常的方式是在广告主网站上布DSP域名的代码，收集到用户行为并进行离线加工分析，再将加工出的标签用于广告投放。由于可以细分到每个展示的粒度来决策和出价，这使得需求方可以向一个推荐系统那样精细化执行广告活动，也使得推荐和广告这两向互联网重要的技术找到了完美的切合点。下面我们从两种常用的定制化用户标签入手，深入看一下广告和推荐的关系。

### 3.2.1 重定向

重定向(Retargeting)是广告中最早产生，也最广泛使用的一种定制化标签。重定向的概念很简单，即把那些曾经对广告主服务发生明确兴趣的用户找出来，再次向他们投放该广告主的广告。在不同的广告类型上，重定向主要有两种的目的：

一、用于品牌广告。当用户已经选择过某品牌的服务或产品后，如果在比较高质量的媒体上看到该品牌的广告，他会进一步肯定自己决策的正确性，从而对该品牌的认知度也大大加强。这种用途下，应当以宣传品牌而不是具体产品为主要诉求。

二、用于效果广告。当用户曾经考虑过某种产品，但没有完成最终转化，通过在线广告将他找回，点击率和转化率都会明显高于平均水平。如果用户已经选择了该产品，那么可以利用推荐技术为他推送相关的产品广告。

我们可以看出，在这两种用途中，重定向都绝不仅仅是你浏览过什么产品，就再给你推送该产品的广告创意这么简单。重定向可以从信息来源和使用信息的精细程度上区分为下面三种类型：

一、网站重定向(Site Retargeting)，即将在一段时间内到达过广告主网站的用户群作为重定向集合。这样的重定向流量其eCPM一般来说要比无定向流量高出一个数量级，因此需要尽可能扩

大投放量。在媒体上采买这种重定向流量时，能够得到的量有两个主要影响因素，一是广告主网站本身的独立访客量水平，二是这些访客与媒体的重合程度。前一个因素没有办法通过广告手段扩大，而后一个因素则要求尽可能多地通过各种渠道采买重定向流量。

要进行网站重定向，需要一个接口将访问广告主网站的用户集合传送给DSP，即架构图中的Customized Audience Segmentation部分。这个接口也有两种主流的实现方式：第一种是直接广告主的网站上布设DSP域名的JS代码或者外链图片(也可以是不可见的beacon)，这样DSP就可以直接收集到访客的记录，再自行加工分析即可；第二种方式，是用线下数据接口的方式，定期将广告主或者其委托的DMP收集到的访客集合批处理式地传送给DSP。当然，前提是DSP与广告主或其DMP之间建立起了cookie-mapping的机制。这两种方式各有各的优缺点：前者能够实时地获取访客信息，但是需要一段时间的数据积累才能覆盖广告主用户集合的大部分，而且在多个DSP同时服务于一家广告主时需要加多段跟踪代码，有降低页面响应速度的风险；后者虽然可以迅速得到访客集合，并且避免了页面因多组跟踪代码而变得太重，却在数据更新时有一定的延迟，并且对广告主方的技术要求较高(跟DMP对接时例外)。

二、个性化重定向(Personalized Retargeting)。对重定向流量进行深入加工，按照品类和购买阶段等因素进行创意上的深度个性化，就是个性化重定向。具体来说，个性化重定向可以在两个方面做深入挖掘：一是对于处于不同购买阶段的用户，采用合适的创意推动他尽快完成转化行为，这里的购买阶段，包括浏览、搜索、加入购物车等；二是对于已经有过一些购买记录的用户，使用推荐技术向其展示相关的商品以提升二次购买率。从这两层意义上看，这一方法与推荐有非常多的共通之处：以广告产品的视角看，我们称之为个性化重定向；而从推荐产品的视角看，则可以认为是一种站外推荐(Off-site Recommendation)，换句话说，是将原来广告主网站上的推荐模块搬到了站外。

三、搜索重定向(Search Retargeting)，即将搜索过跟广告主直接相关的关键词的用户群作为重定向集合。这样的方式也可以获得很精准的用户群，其绝对量也要高于网站重定向，不过对于非主要搜索引擎提供上来说，依靠搜索重定向能覆盖的人群比例未必会高于网站重定向。

在以上重定向方式中，简单的网站重定向和搜索重定向实现起来都比较直接，我们重点来讨论一下个性化重定向需要的技术方案。

## 个性化重定向

个性化重定向与传统的受众定向方式有比较大的技术差别，虽然这些差异化的技术都不算困难。其核心支持技术，主要有以下几个关键：

一、动态创意。个性化重定向的核心，是在某个用户到来时，实时决定给他展示什么商品，以及用什么方式展示。由于广告主的商品数量往往相当大，因此显然不可能为所有的商品预先准备好创意，而如果考虑到同时推荐若干商品的组合关系，就更加不能采用静态的创意方式。因此，动态创意技术，是个性化重定向最重要的支持技术。这是在线广告朝着彻底个性化、动态化方向发展迈出的重要一步。

二、推荐技术。由于可以把个性化重定向因而看作是站外推荐，显然推荐技术也是必要的。不过与站内推荐相比，这样的推荐还是有一些不同之处：首先站内商品页上的推荐主要根据上下

文信息来进行,而站外推荐则是根据用户信息来进行;另外,站内推荐由于是发生在某个特定的购买环节上的,因而往往不需要根据用户的购买阶段来调整创意,而站外推荐这么做则有非常大的必要。

三、广告主商品库存实时接口。对于站外的商品推荐,如果用户在点击某单品到达广告主网站时,发现该商品已经售完或下架,或者是价格与创意上的宣传不符,会对该广告主的品牌形象有较严重的伤害。为了尽可能避免这种情况的发生,个性化重定向的技术提供者往往需要统一提供一个准实时的商品库接口,使得广告主可以比较简单地将自己的库存和价格信息及时地同步过来。

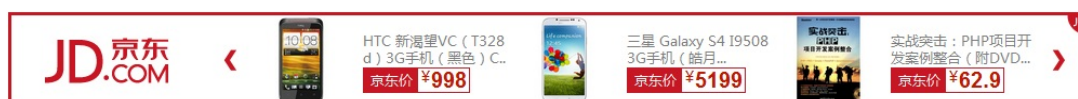


Fig. 3.1. 个性化重定向广告示例

在图3.2.1中,我们给出了个性化重定向广告的一个示例。该广告投放的对象,是某个京东商城的用户。该用户很可能在京东商城关注过某种手机商品,以及有关PHP编成的技术书籍。因此,广告系统根据他的个性化购物兴趣,结合京东商城的商品库存,为其推荐了相关的手机和技术书籍。可以想见,这样的广告其点击率和转化率都会比较高。

## 搜索重定向

### 3.2.2 新客推荐(Look-alike)

重定向的方式虽然精准,但是量受到极大的限制,而且与广告希望接触某种类型潜在人群的目的也有距离。对于很多广告主来说,除了已经对自己的产品发生明确兴趣的用户,他们更希望能有系统性的手段接触到那些“很有可能”对自己产品发生兴趣的用户。对于那些快速成长期的电商、或者是网站流量并不大的线下业务广告商(比如银行、汽车),这方面的需求尤其强烈。“很有可能对自己产品发生兴趣的用户”这样的用户划分,从目的上来说很明确,但从做法上来说很模糊。基本的思路是这样:由广告主提供一部分种子用户,再由拥有更丰富数据的广告平台分析这些用户网络行为的特征,并根据这些特征找到具有相似特征的拓展人群。很显然,这也是一种定制化用户标签,即 $f(a,u)$ 的定向方式。

从推荐的角度来看Look-alike,可以认为这是一种“新客推荐”的方式,即向原来没有关注广告主产品的潜在用户推广该广告主产品。当然既然是推荐,就不是漫无目的的撒网式推广,而这样的推荐技术,考虑的是没有站内行为情况下的推荐,是原有狭义推荐问题的扩展,也是广告本质目的与推荐需求的重要切合点。

关于Look-alike的具体建模方法,目前市场上还没有大家公认的通用方案。不过,如果我们从这种定向方式的本质,即 $t(a,u)$ 的特点出发,并且以优化效果为主要目的的话,也可以对前文讨论的点击率预测模型做一些变化,得到一种Look-alike的可行方法。

前文讨论的点击率模型，目的是任意给定一组 $(a, u, c)$ 的组合，按照一个预先训练好的模型对其点击率进行预估。如果我们变换一下思路，只用某一个广告主的投放数据，训练一个 $(a, u)$ 组合相关特征的点击率模型 $p(h = 1|a, u)$ ，那么这个模型描述就是一个专门对与该广告主，不同用户的点击率水平的评估函数。由于此模型与 $c$ 无关，因此同一个用户的不同展示可能出现有时为正例、有时为负例。如果对这一模型再做一点加工，将所有在训练集中出现过的曾经访问过该广告主网站的用户，其对应样本都置为正样本，那我们得到的就是一个评价该用户是否有可能成为该广告主访客的函数了。

### 3.2.3 代表案例介绍

定制化用户划分和RTB技术的产生，催生了大量以技术方式优化广告采买的DSP公司。这些公司当中，有些以CPM采买，并优化ROI的套利模式为主；有的则以透明的采买和效果优化功能性服务为主，我们将对这两种类型的DSP分别举例说明。

#### Criteo

Criteo是专注于个性化重定向产品的一家总部位于法国的广告技术公司。需要首先说明的是，Criteo从其历史业务来看，并非我们认为是严格意义上的DSP，这主要是由于它主要的流量才买方式，并不是实际竞价，而是一种被称为优选(Preferred Math)的交易方式，我们先来看一下这种交易的过程。

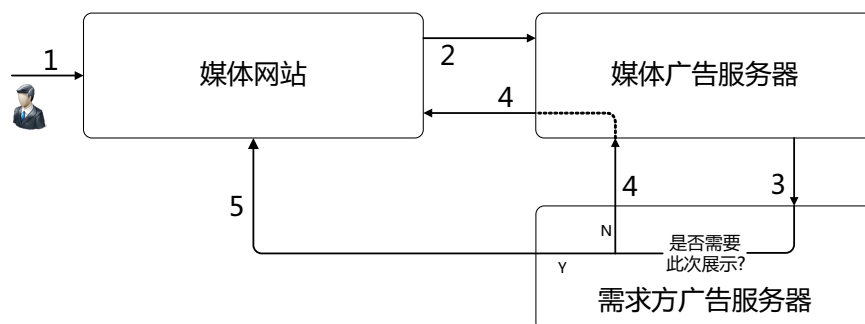


Fig. 3.2. 优选购买方式过程示意

优选过程的示意如图3.2.3所示，其关键是允许某个需求方按照自己的意愿来挑选流量，但是又可以避免复杂的cookie-mapping和竞价过程。这个过程主要有以下几个步骤：1. 用户浏览媒体网站；2. 媒体网站JS代码向媒体的广告服务器发起请求；3. 媒体广告服务器向需求方的广告服务器发起请求，询问是否需要此次广告展示机会；4. 需求方广告服务器根据自己的逻辑判断，如果需要此次展示机会，则返回相应的广告创意；5. 需求方如果不需要此次展示机会，则通知媒体广告服务器，由媒体广告服务器在从自有广告库中选择合适的创意返回。

优选过程使得需求方可以比较自由地挑选流量，同时也要付出较高的CPM成本。与前面的RTB过程相比较，优选过程也可以从概念上理解为一个只有唯一参与者的RTB过程。优选过程的一个缺点，是决策过程可能存在比较多的服务器往返：在极端情况下，需要三次服务器的往返才能够得到最后的广告。这有可能使得用户看到的广告延迟增大，从而影响广告效果。这一过程中的第3个步骤有可能存在更复杂的逻辑，即媒体方可以根据实现与需求方的一些信息对接，初步判断某些展示机会是需求方肯定不需要的，这样的情况下，就不再向需求方广告服务器发起请求，从而减少这部分流量的服务器往返次数。

Criteo不同于其他的需求方技术公司，基本上只按照个性化重定向方式采买广告。其核心技术也就是前面介绍的三项。下图中给出了Criteo个性化重定向创意的几个例子。Criteo按照前面介绍的优选方式，以CPM方式与媒体结算，但是与广告主之间的结算完全采用CPC方式，从而实现套利，并且也比较容易被效果类广告主接受。在个性化重定向的方案框架内，Criteo还提供了User、Category、Data和Banner四个维度上比较灵活的优化功能，即广告主可以根据不同的用户细分、商品种类、具体数据和创意类型设置不同的点击出价，从而达到非常精细的ROI管理和优化的目的。这样的显示广告运营和投送方式，其精细程度与搜索广告有相似之处：都是在非常精准的流量细分上以效果为导向投放广告，并且能够在非常精细的粒度上控制出价。不过这种方式对与显示广告来说有一些先天的缺点：个性化重定向不同于搜索广告，需要预先在广告主网站布设跟踪代码，这使得新广告主加入的进程变得大为复杂。因此，专门从事个性化重定向的公司，在初期会面临需求不足的问题，并且需要运营商大量的努力和效果上良好的记录才能逐步改善，而Criteo也同样经历了这一过程。

## InviteMedia

InviteMedia原是一家独立的DSP公司，后背Google收购，现已整合进DoubleClick的广告系列产品，并改名为DoubleClick Bid Manager。

InviteMedia是广告交易市场上的先行者之一。

## 3.3 供给方平台(SSP)

现在我们可以来看看在程序交易的环境下，供给方的综合性技术方案了。对于媒体而言，无需把全部流量的变现都押在一种交易方式上：既可以通过直接销售来高溢价地售卖品牌广告，也可以灵活地综合使用各种程序交易方式以追求更高的eCPM。参照上一节中的广告交易方式谱系，媒体的统一变现平台需要这样的逻辑：当广告请求到达时，首先检查优质销售方式有无需求，包括CPT和GD的合约，如有需求，按照优先级即在线分配的方案完成投放；如果没有优质销售合约，则从自运营广告主库中找出eCPM较高的，并估算可供调用的若干Network的期望eCPM，在这两者之间找到较高的，以此为低价通过RTBD接口再向接入的各DSP实时询价。可以看出，在这样的逻辑中，广告请求导师是被分配到自运营广告库，还是其他Network，或者是DSP，是根据他们的收益在线动态决定的，这样的方案称为动态分配(Dynamic Allocation)。这样的统一接

入各种广告需求以优化媒体广告变现能力的技术，我们称为收益管理(Yield Management)，对应的产品形态，就叫做供给方平台(Supply Side Platform, SSP)。

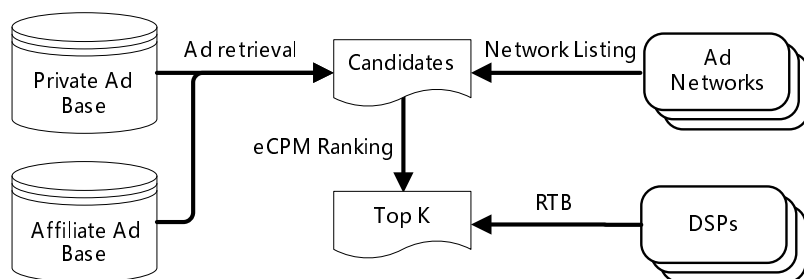


Fig. 3.3. 动态分配决策过程示意

SSP中多种广告源动态分配的概要性决策过程如图3.3所示。首先，从自有广告库(Private Ad Base)、联盟广告库(Affiliate Ad Base)中根据当前 $(u, c)$ 上的标签信息检索合适的广告候选，并估计每个广告候选的eCPM；同时，也估计出各个合作的广告网络在当前 $(u, c)$ 情况下大致的eCPM。然后，对所有候选广告和合作广告网络按照eCPM进行统一排序，再根据排序结果中最高的eCPM做低价调用RTB接口，向各个合作的查询更高变现价值的广告。先进行本地广告检索和eCPM估计，再进行RTB的好处是可以根据当前展示的情况动态调整RTB的低价，而当低价高到一定程度时，即可以略去RTB的过程，从而减轻整个Ad Server的负担，这样一来，此过程即可以充分利用RTB带来的大量广告主和市场流动性，又充分利用了本地广告库抬高竞价的水平线，从而在提高了收益的同时不至于耗费大量的服务成本。当然，这里介绍的动态分配过程是非常概要和粗略的。在实际的媒体变现中，除了按照eCPM竞价的需求，还需要处理合约广告的保量需求，因此在按eCPM排序之前，还需要有一个对合约广告的在线分配过程，并对那些合约广告不需要的量再进行竞价。

上面的过程里SSP有一个特有的计算问题，这就是其中的网络优化(Network Optimization)问题。这个问题，是对某次展示机会的 $(u, c)$ ，对 $N$ 个候选的Ad Network估计其eCPM以优化收益。相对于CPC计费的Network里的eCPM估计，即点击率预测问题，此问题的挑战主要在于我们 $a$ 维度上只能对Network做粗略的选择，而无法精细地比较各个具体的广告。Network Optimization需要特殊考虑的因素，主要由两个，一个是各候选Network的分成模式和比例；二是各Network的返回率。其中第二个问题是考虑到Network在接收到广告请求后，由于网络延迟、广告资源等各方面的原因，并不一定能够成功返回广告，而这些也是影响收益的重要因素。关于点击率预测，其框架技术与前面的讨论并无质的区别，我们可是可以用 $(u, c)$ 上的受众定向特征，与现在代表广告网络的 $a$ 上的特征做组合，采用静态和动态特征结合的思路来预测。不过在实际系统中，由于对Network自己的定向方式和广告库一无所知，使用人群标签可能帮助并不大，反而是使用广告位和时间段这样的基本划分比较稳健。当然，如果接入的Network中有一些是垂直性的，相应的人群标签就会起作用了。



我们也可以从另外一个有趣的视角来看网络优化。前面我们提到Ad Exchange中的询价优化问题，即估计个DSP可能的出价(由于是CPM竞价，出价即是eCPM)，并选择少量的DSP询价这一过程。考虑极端情况，即只选择一个估计eCPM最高的DSP，那么其计算就变成网络优化问题了。

### 3.3.1 SSP系统架构

提供较全面收益管理功能的SSP公司，数目并不多，主要有已经被Google收购的Admeld，以及Rubicon和Pubmatic等。我们以Admeld为例说明一下SSP的业务和技术。

#### AdMeld

### 3.3.2 代表案例介绍:Admeld

## 3.4 数据加工与交易

要提高定向的精准程度与人群覆盖率，技术并不是最重要的因素。那么什么才是决定性的呢？其实是数据的来源与质量。这是正确认识精准广告业务非常重要的观念。为了进一步强调这一观念，我们以大家更容易理解的石油加工工业为例与作类比：在石油工业中，从油田挖掘出的原油是整个行业的原材料，炼油厂的作用是把这一原材料加工成能直接做燃料用的汽油等商品，再输送给加油站这样的销售终端。在精准广告中，可以把用户的行为类比为石油工业中的原材料，而日志收集和清洗系统的作用就相当于油田的挖掘设备。而受众定向的平台，就可以类比为炼油厂，它把原油，即清洗过的日志，加工成用户标签，而这些用户标签就像汽油一样，是可以被销售和使用的了。而传统广告中起关键作用的广告位，在这里仅仅变成了加油站，负责完成产品消费的过程而已。

那么我们如何正确认识技术在精准广告业务中的作用呢？从上面的类比可以看出，技术的地位相当于挖掘设备和炼油设备，当然有着无可置疑的重要性。技术能力的高低，直接影响着数据采集和变现的有效性。不过从另一个角度说，技术的作用也不能被过分夸大。巧妇难为无米之炊，没有了高质量的原材料，即用户数据，再高明的技术也是没有用武之地的。

既然数据本身有这么根本性的作用，从某种意义上实际上是精准广告市场的核心，那么围绕数据本身的加工与交易，就与广告的投放技术一样令人瞩目。有哪些数据是对精准广告业务有直接贡献的呢，我们可以按照下面的分类来总结：

一、用户标识。对广告而言，如果标定哪些行为来自于同一个用户，是非常关键的问题。不管你能拿到多少访问日志，如果没有办法把它们跟你的投放系统联系起来，这些数据都没有办法发挥作用。对于浏览器行为，我们最常使用的用户标识是cookie，但是由于存在同时使用多个浏览器，cookie过期，或用户主动清除cookie的情况，这种用户标识的长期一致性并不算太好。不过好在对广告来说，起关键作用的还是用户近期内的行为，所以用cookie作用户标识还是有效且为业界广泛采用的基础方案。如果运营广告业务的域名同时提供其他有永久身份的服务，比如email、SNS等，那么可以用这一永久身份找回过期或被清除的cookie，这样用户身份的上一致

性就会有所改善。当然，如果广告业务域名和用永久身份服务的域名不同，也不是完全没有办法，在后者同意的前提下，可以采用一种cookie映射的方法来对应彼此的用户身份，这一点的细节我们在后文谈到广告交易平台时再讨论。因此，高质量的用户标识，本身就是一种非常有价值的数 据，也是可以在市场交换和售卖的。

二、用户行为。业内通常认为，主要有以下几种类型的在线行为是可以被广泛采集，且对于受众定向或广告决策有明确作用的：转化(transaction)、预转化(pre-transaction)、搜索广告点击(sponsored search click)、显示广告点击(ad click)、搜索点击(search click)、搜索(search)、分享(share)、页面浏览(page view)、广告浏览(ad view)。按照对效果广告的有效性，这些行为可以分为三档：第一档是转化与预转化行为，转化指的是在需求方网站上的具体决策步骤(例如电商的购物行为，游戏的充值等)，预转化指的是以及转化前的一些准备动作，比如浏览、搜索等。这两种行为由于直接明确反映了用户兴趣，因此信息有效性远远高于其余行为。第二档是搜索广告点击、显示广告点击、搜索点击和搜索，这些行为虽然离转化稍远一些，但是也基本属于用户主动行为，有效性也较高。最后一档是分享、页面浏览和广告浏览，这些行为往往都产生于用户在网上无目的浏览时，因此有时不能代表其真正意图，其中分享的有效性好一些，而广告浏览则是与效果负相关的——看得次数越多，效果越差。一般而言，越靠近转化的行为，越主动的行为，对受众定向的指导性越强。

三、人口属性。人口属性本身是常用的一种定向标签，因此其数据来源很重要。一般来说，只有一些能够与用户实名身份绑定的服务可以得到此信息。我们也可以利用网络行为数据来进行人口属性标签的预测，但是这样做的准确程度一般来说很有限，而且这仍然需要一些标定的数据用于训练。对于某些人口属性，可能一些特别的信息比较容易给出准确的判定，比如用语音服务记录的声音信号，可以将男女区分得相当准确。

四、地理位置。地理位置信息随着能获得的精度不同，其用途也会有相当大的差异。如果只能根据IP进行映射，我们往往只能拿到精确到城市级别的地理位置，当然这对于很多广告投放来说，已经有相当的价值。而在移动互联环境下，GPS或蜂窝可以提供的定位往往可以准确到几百米的范围，这就使得Hyper-local的区域广告商投放定向广告成为可能。这样的广告定向，对于餐饮等受地理位置限制极大的线下业务广告商是非常有价值的。

五、社交关系。社交关系数据的主要意义，在于为用户属性和行为标签提供了平滑的可能性。这一方法的假设是同一个圈子的用户可能在某些方面有相同的兴趣，因此受众定向可以转化为对这个小圈子共性的挖掘。从这个意义上来看，强关系类型的SNS比弱关系的有优势，关注人群比被关注人群的信息意义更大。

### 3.5 数据管理平台(DMP)

既然数据的收集和加工是广告市场上非常重要的环节，也就产生了专从事这方面业务的产品，称为数据管理平台(Data Management Platform, DMP)。它的主要功能，是逻辑聚合各种来源的在线的用户行为数据，按照需要方需要的加工成有价值的用户标签，然后在广告市场上使用这些标签来变现。

DMP有几个核心功能。一、它可以为网站(可以是媒体、也可以是广告主网站)提供受众定向功能,并将得到的用户标签应用于网站业务。在这一过程中,除了加工一些通用标签,DMP还需要能够比较灵活地按照网站定义的用户标签来加工受众人群;二、如果媒体网站授权,DMP可以提供接口对加工出来的用户标签进行变现,并与网站进行分成;三、广告主网站可以通过DMP与广告采买渠道进行更方便地数据对接。这一点可以从一个典型的应用来理解:假如某广告主需要通过外部广告平台做重定向(见第7章),那么需要将自己的用户集合通过某种技术方式通知广告平台。如果每个广告平台都采用在广告主网站上加跟踪代码的方式来收集用户,有两个的弊端,一是多个广告平台同时加代码,有可能使页面变得太重;二是访客的积累需要可能长达数周的时间,这使得广告平台重定向的效率降低。如果由DMP唯一负责广告主网站的用户积累和划分,并通过数据接口的方式传送给广告平台,那么可以极大程度上解决上述问题。

### 3.5.1 DMP案例

在北美的在线广告市场上,有不少以DMP类似业务为主要方向的公司。在这里我们介绍两家比较有代表性的:一家是Bluekai,一家是AudienceScience。在\*图中,后者实际上被划分在“Targeted/Audience”这个板块中,不过由于其主要业务也是面向其它网站的数据加工,因此可以放在DMP板块中加以比较。

#### Bluekai

Bluekai的主要业务模式,是聚合大量中小媒体的有价值行为数据,使用受众定向技术为用户打上标签,并对外售卖标签以获取收入。Bluekai同时提供面向媒体、数据提供商和广告主的一系列产品:包括一个DMP、一个数据交换平台(Bluekai Exchange)、以及一个数据分析系统。这几项产品都围绕一个商业目标展开:那就是帮助有数据变现需求的参与者能够自由、灵活地通过技术对接的方式与广告主进行交易。对于媒体或者其他拥有数据者,可以通过Bluekai Exchange将自己的数据公开式地出售给市场上的需求方,同时可以比较自主地控制定价;对于广告主,可以通过Bluekai提供的DMP产品和第三方数据标签,与自己的第一方数据结合起来,对自己的人群进行更灵活的划分,并按此购买广告。Bluekai通过数据交易获得的收入,其中很大比例还将返还给数据提供方。

通过这种数据交换方式,广告市场上最有价值的数据资源被盘活利用了:数据拥有者不需要直接涉足复杂的广告业务,也可以对数据进行变现;而数据需求者也可以方便地找到数据购买来源,以快速提高自己广告投放的效果。

Bluekai有着比较开放的标签体系,根据广告主使用目的和数据来源的不同,这些标签又分成如下的几种:Intent、B2B、Past Purchases、Geo/Demo、Interest/Lifestyle、Branded、Estimated Financial/Economic。这是一个比较典型的半开放的标签体系,其中的Intent部分,非常类似于前文中的兴趣定向体系,这部分是由Bluekai根据通过Data Exchange收集的用户行为加工的。而其他的部分,有的是由其它数据提供商直接提供,比如由Bizo提供的B2B标签;也有的是根据多家

的数据融合决策而成，比如Demo/Geo标签，实际上融合了Bizo、Datalogix、Expedia等多家数据提供商的数据源。

之所以说Bluekai的标签体系是开放式的，是因为它会根据数据的来源和市场需求不断拓展和调整标签的类别和内容，力求能够满足尽可能多的广告主的特质化需求。Bluekai主要的几种标签、覆盖用户量和主要数据来源，我们列在下表中。

**Table 3.1. Bluekai标签体系主要类别**

类别	描述	数据来源	规模(用户数)
Intent	最近输入词表现出某种产品或服务需求的用户	Bluekai Intent	160+MM
B2B	职业上接近某种需求的用户	Bizo	12+MM
Past Purchase	根据以往消费习惯判断可能购买某产品的用户	Addthis, Alliant	65+MM
Geo/Demo	地理上或人口属性上接近某标签的用户	Bizo, Datalogix, Expedia	
Interest/LifeStyle	可能喜欢某种商品，或某种生活风格的用户	Forbes, i360, IXI, ...	103+MM
Estimated Financial	根据对用户财务状况的估计做的分类	V12	

## AudienceScience

AudienceScience是广告市场上首先明确提出“Audience Targeting”这一概念的公司，并且长期专注于这方面的数据加工和算法建设。

就其数据聚合和收入分成的模式来说，AudienceScience与Bluekai有很多相似之处。两者主要的一个区别，是AudienceScience并不是通过售卖标签来获得收入，而是通过运营一个自有的广告网络来变现。这样做的原因，是AudienceScience认为数据加工业务在扣除媒体分成以后利润空间太小，而自营广告网络有可能获得更大的套利空间。

## 3.6 广告程序交易谱系

到目前为止，我们已经接触到了在线广告市场上大多数常见的交易方式，有必要对它们做一个系统的总结。按时间段和广告位独占式的售卖，以及GD这种合约式的售卖，是销售与客户之间将广告投放的关键方面约定下来的交易方式，在其中人的因素起了相当大的作用，不过这种售卖一般来说由于能满足一些品牌性需求，溢价能力也比较好，在高质量的媒体上属于优先考虑的售卖方式，我们把它们称为优质售卖(Premium Sales)。从技术层面来说，优先售卖整体对计算的要求不算特别深入，而且主要的技术集中在供给方(比如在线分配、受众定向)。在Ad Network和Ad Exchange产生以后，广告交易的一些细节越来越多地依赖在线计算而非事先约定，对需求方和供给方而言，也都产生了一些新的程序交易(Programmatic Trade)方式。

	程序交易 (Programmatic Trade)			优质售卖 (Premium Sales)	
供给方	RTBD	Preferred Match	Network Optimization	Guaranteed Delivery	CPT
中间平台	Ad Exchange		Ad Network	Ad Server	
需求方	RTBS	Preferred Match	Portfolio Selection		

Fig. 3.4. 在线广告主要交易方式一览

当竞价广告产生以后，对供给方而言，可以通过将广告位直接托管给Ad Network的方式变现，也可以同时使用给多个广告联盟，按照一些准则或计算接过来灵活地对不同的流量分割选择不同的Ad Network，这是供给方利用Ad Network变现的一种程序交易方式，称为网络优化(Network Optimization)，我们在下文中的供给方平台中还会讨论。对需求方而言，则存在选择合适的人群标签并合理出价，以优化整体ROI的程序交易要求。在实时竞价的环境中，程序交易的过程变得更加简单直接，供应方的RTBD接口和需求方的RTBS接口通过Ad Exchange直接对接，并完成精细的流量采买。除了这两种市场上常见的程序交易结构，还有一种称为优选匹配(Preferred Match)的交易机制，也属于程序交易的范畴，这种交易方式是需求方与供给方约定一个CPM采购价格(一般来说该价格高于市场平均水平)，但是并不一定完全接受供给方发送过来的流量，对于其不需要的部分，再发回给供给方投放其他优先级更低的广告。这种方式是供给方与需求方之间直接发生的程序交易，但是扩展性不强，不适合于大量广告主同时接入的方式。正如其名字表示的那样，优选方式主要适合于媒体将剩余流量以较高的溢价售卖给特别精准的高价值需求方，比如Criteo那样的个性化重定向平台。我们在图3.4中列出了包括优质售卖和程序交易等各种在线广告主要的交易方式，希望对大家从技术角度全面理解市场结构有所帮助。

上面谈的是各种在线广告的交易方式，在实际的广告市场中，这些交易方式又是怎样通过具体的产品联系起来的呢？我们在图3.6中示意性地表示了在线广告市场各种产品的配合关系。从媒体(Publisher)角度看，有三种产品可以帮助其变现流量：一、可以将广告位托管给广告网络，由广告网络决策广告投放，从赚取分成，如图中的“Publisher 1”；二、可以将广告位对接到广告交易平台，以实时竞价的方式变现，如图中的“Publisher 2”；三、也可以将广告位托管给SSP，这样可以同时对接多个广告网络和实时竞价接口，并按照动态分配的逻辑选择变现最高的需求方(这一产品的具体情况我们在下一节中介绍)，如图中的“Publisher 3”。

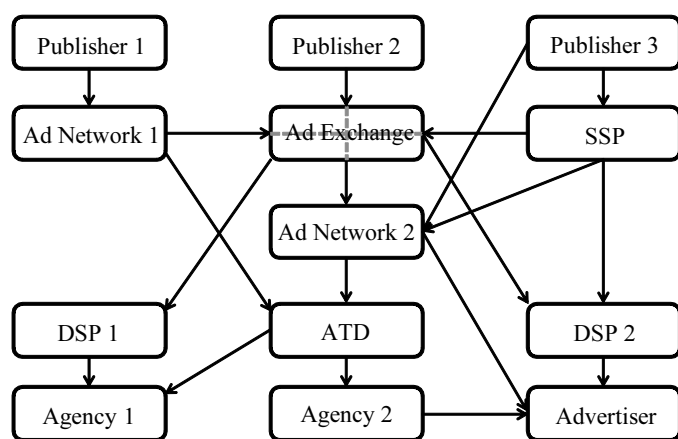


Fig. 3.5. 在线广告市场产品交互关系示意

## **Part III 计算广告核心技术详解**





# 基础知识准备

## 1.1 信息检索

### 1.1.1 信息检索

从上面的计算广告系统架构可以看出，广告也采用的是类似搜索的技术框架以获得良好的扩展性。所以我们有必要对信息检索和搜索引擎的基本技术有所了解。不过广告相关的搜索技术范畴远远小于完整的网页搜索引擎，特别是不需要全网爬虫和Page Rank算法的部分。因此，我们重点介绍与广告相关的一些搜索基础技术，包括倒排索引和向量空间模型。

#### 倒排索引

倒排索引是现代搜索引擎的核心技术之一，其核心目的，是将从大量文档中查找包含某些词的文档集合这一任务，用 $O(1)$ 的时间复杂度完成。此处的 $O(1)$ 是相对文档的数目而言的。也就是说，利用倒排索引技术，可以实现与文档集大小无关的检索复杂度，这一点对于海量内容的搜索来说，是至关重要的。为了说明倒排索引的基本概念，我们看一下下面的例子。假设我们有如下的几篇文档：

$D_0$  = “谷歌地图之父跳槽Facebook”

$D_1$  = “谷歌地图之父加盟Facebook”

$D_2$  = “谷歌地图创始人拉斯离开谷歌加盟Facebook”

$D_3$  = “谷歌地图创始人跳槽Facebook与Wave项目取消有关”

$D_4$  = “谷歌地图创始人拉斯加盟社交网站Facebook”

对每篇文档都进行分词以后，可知这些文档中包含的关键词(term)有：{谷歌，地图，之父，跳槽，Facebook，加盟，创始人，拉斯，离开，与，Wave，项目，取消，有关，社交，网站}。首先，去掉“与”这样的没有实际表意作用的停止词(stop word)，我们对每一个建立一个链表，表中的每个元素都是包含该词的某篇文档的标识。于是，与上面的文档集对应的倒排索引，也就是所有关键词的倒排链集合，可以表示如下：

谷歌 $\rightarrow \{D_1, D_2, D_3, D_4, D_5\}$ , 地图 $\rightarrow \{D_1, D_2, D_3, D_4, D_5\}$ , 之父 $\rightarrow \{D_1, D_2, D_4, D_5\}$ ,  
 跳槽 $\rightarrow \{D_1, D_4\}$ , Facebook $\rightarrow \{D_1, D_2, D_3, D_4, D_5\}$ , 创始人 $\rightarrow \{D_3\}$ ,  
 加盟 $\rightarrow \{D_2, D_3, D_5\}$ , 拉斯 $\rightarrow \{D_3, D_5\}$ , 离开 $\rightarrow \{D_3\}$ , Wave $\rightarrow \{D_4\}$ ,  
 取消 $\rightarrow \{D_4\}$ , 项目 $\rightarrow \{D_4\}$ , 有关 $\rightarrow \{D_4\}$ , 社交 $\rightarrow \{D_4\}$ , 网站 $\rightarrow \{D_4\}$ 。

为了后文一些实例的方便, 我们用下面一段代码中的类结构来表示一个倒排索引。这个类结构派生于Hash Map, 其中的key为关键词即term, 典型情况下, 该key是string类型, 但是在后文介绍的布尔表达式检索等场景中, 其key的类型可能会发生变化。因此, 为了逻辑统一起见, 我们引入了模板参数来泛化此处的数据类型。而Hash Map的value就是倒排链, 是一个由索引条目(Entry)组成的链表。每个索引条目有两个域, 第一个是该条目对应的文档的ID, 第二个是一个辅助变量, 比如可以用于表示目前关键词在对此文档的TDIDF, 在后文的其他索引类型中也会有独特的应用。当然, 这种结构只是一个概念上的表达, 实际的倒排索引还要存储很多其他信息, 我们为了便于突出主要概念, 在本书中采用这样简单的概念性描述。

倒排索引最基本的操作有两项: 一是向索引中加入一个新文档, 二是给定一个由多个关键词组成的查询(query)时, 返回对应的文档集合。我们也在下面的代码中对这两项基本功能的实现做了描述。需要注意的是: 在倒排索引中, 由于文档ID是在加入倒排索引时被在线分配的, 因此每个倒排链都可以确保是有序的, 这会在后文的应用中得到具体利用。

```

1  typedef pair<int, float> Entry;
2
3  template <class TKey>
4  class InvIndex : public map<TKey, list<Entry> > {
5  public:
6      vector<vector<TKey> > docs; // 文档正排表
7
8  public:
9      void add(vector<TKey> & doc) {
10         // 在正排表里记录该文档
11         docs.push_back(doc);
12         int curDocID = docs.size() - 1;
13
14         // 遍历document里所有的term
15         for (int w = 0; w < doc.size(); w++) {
16             map<TKey, list<Entry> >::iterator it;
17             it = find(doc[w]);
18
19             // 如果该term的倒排链不存在
20             if (it == index.end()) {
21                 list<int> newList;
22                 index[doc[w]] = newList;

```

```

23         it = find(doc[w]);
24     }
25
26     // 在倒排链末尾加入新的文档ID
27     it -> second.push_back(curDocID);
28 }
29 }
30
31 void retrieve(vector<TKey> & query, vector<int> & docIDs) {
32     int termNum = query.size();
33
34     // 合并所有term的倒排链
35     list<Entry> result;
36     for (int t = 0; t < termNum; t++) {
37         map<TKey, list<Entry> >::iterator it;
38         it = index.find(query[t]);
39         if (it == end()) // 该term倒排链不存在则跳过
40             continue;
41
42         result.merge(it -> second);
43     }
44
45     // 得到返回的文档ID集合
46     docIDs.clear();
47     list<Entry>::iterator lit = result.begin();
48     for (; lit != result.end(); lit++)
49         docIDs.push_back(lit -> first);
50 }
51 };

```

实际的倒排索引远远比这里示例性的代码复杂。其主要工程难点有很多，比如如何设计精简的数据结构以节省对内存的使用，以及如果准实时地将新的文档加入倒排索引。这些问题由于是信息检索领域专门的研究课题，又非广告的特殊需求，我们不再深入介绍。需要自行实现广告检索部分的读者，可以深入学习开源的倒排索引工具Lucene，或者参考这方面专门的文献。

## 向量空间模型

如果说倒排索引技术是大规模信息检索的基石，那么向量空间模型(Vector Space Model, VSM)则是信息检索中最基础且最重要的文档相似度量方法之一。VSM的核心有两点，文档的表示方法和相似度计算方法，我们在下面分别介绍。

首先，我们对每个文档采用“bag of words”假设，即用各个关键词(term)在文档中强度组成的矢量来表示该文档：

$$D = (x_1, x_2, \dots, x_M)^T \quad (1.1)$$

其中 $x_m$ 一般采用词表中第 $m$ 个词在 $D$ 中对应的TFIDF(Term frequency - Inverse Document Frequency)值，这是一种信息检索中最常见的词强度度量，它可以分解为两个量的乘积：这两个量是词频(Term Frerquency, TF)，即某文档中该词出现的次数；和倒数文档频率(Inverse Document Frequency, IDF)，即该词在所有文档中出现的频繁程度的倒数。IDF的引入，是考虑对那些广泛出现在各个文档中的常用词对主题的鉴别力并不强，因而需要降低其权重。IDF的计算方法有若干种，最常用的形式为：

$$\text{IDF}(m) = \log(N/\text{DF}(m)) \quad (1.2)$$

其中 $\text{DF}(m)$ 为词 $m$ 在其中出现的文档的总数目， $N$ 为总文档数目。在广告应用中如何计算IDF值，在某些情形下需要不同的处理：例如在处理对广告主有价值的竞价标的词时，可以采用所有广告描述，而不是互联网上的网页作为文档集合。相应地，在根据关键词进行广告检索时，也应该使用这种方法得到的TFIDF。

这样的bag of words文档表示方法，是对自然语言最简单粗略的一种近似表示。它完全忽略了词的前后接续关系，以及更高阶的语法因素的影响，因而并不太可能具有精细的文档描述能力。不过，这种方法在信息检索中的作用无疑是巨大的，因为它通过极为简单经济的操作对文档进行了简化，同时又比较好地保留了文档的概貌，这对于海量文档数据的处理和索引非常有利。时至今日，虽然学术家在自然语言处理方面取得了许多进展，这种简单的方法，仍然是工程实践中信息检索和文档主题挖掘的最常用文档表示。如果我们考虑更精细的文档描述，可以进一步加入文档的n-gram信息，但是也会带来数据的爆炸式增长和模型估计稳健性上极大的挑战。

采用baog of words的文档表示方法，在计算两个文档的相似度时，我们一般是用其对应矢量的余弦距离：

$$\cos(D_1, D_2) = \frac{D_1^T D_2}{\|D_1\| \cdot \|D_2\|} \quad (1.3)$$

余弦距离的最显著好处，是当两个矢量归一化不好时，仍然可以得到比较稳健的比较结果。比如有两篇一样的文档，将其中的一篇内容重复一遍，再去计算余弦距离仍然是0，而如果采用其他方式比如欧氏距离，结果就不再是0了。再比如两个人对各种电影打分，甲倾向于给较高的分数，乙倾向于给较低的分数，那么在一组三部电影上，甲给出的分数{3.6, 3.6, 4.8}，和乙给出的分数{3.0, 3.0, 4.0}，实际上一致程度相当高，这也可以被余弦距离比较公允地度量出来。

了解了上面的这些内容，读者应该可以建立起对海量文档进行检索的基线方案了：在检索索引阶段，需要对文档集合分词，并按照bag of words表示得到每个文档的TDIDF矢量，对分词后的文档集合建立倒排索引。当在线的查询到来时，也进行分词，并从倒排索引中查出所有符合要求的文档候选，并对其中的每个候选评价其与查询的余弦距离，按距离由小到大进行排序。这样的一个基本框架，也同样适用于广告这一大估摸数据挖掘问题，也是图3.1的骨干性原理。不过，实际的搜索系统也好，广告系统也好，都不是简单地采用这样的基线系统，而是在检索、排序甚至分词等各个部分都有着更具体深入的算法，这些构成了该框架的血肉。总之，请大家在充分理

解这样的类搜索式的骨干架构的基础上，再去深入挖掘各部分的算法和技术，方能举一反三，万变不离其宗。

## 1.2 最优化方法

为了探索比上面的极限系统更加有效的计算广告方案，我们必然会碰到大量的与数据挖掘和机器学习相关的算法问题。在这些与数据相关的问题中，最重要的基础技能，是最优化(Optimization)的理论和方法。这个主题讨论的是在给定一个数学上可以明确表达的优化目标以后，如何用系统性的方法和思路找到该目标的最优解。这部分的书籍和文章卷帙浩繁，笔者认为有必要从工程的角度出发，整理一下在面临各类目标函数时的一般性思路，并希望大家能够认清“模型”和“优化”这两个概念的联系与区别。

最优化问题讨论的是，给定某个确定的目标函数(Objective Function)，以及该函数自变量的一些约束条件，求解该函数的最大或最小值的问题。这样的问题可以表示为下面的一般形式：

$$\begin{aligned} \min f(x) \\ \text{s.t. } g(x) \leq 0 \\ h(x) = 0 \end{aligned} \quad (1.4)$$

这里 $f(x)$ 是一个关于自变量 $x$ 的目标函数，而 $g(x)$ 和 $h(x)$ 为 $x$ 的矢量函数，对应着一组不等式和等式约束条件，其中 $g(x) \leq 0$ 表示矢量 $g(x)$ 的每一个元素都小于或等于0。根据约束条件以及目标函数的性质不同，最优化问题求解的思路也有很大的不同。我们先从没有约束条件的情形开始，这方面的方法是大多数最优化算法的基础。我们首先根据目标函数 $f$ 是否可导，可以把相应的优化问题分成两类，这两类问题的工程解决方案有比较大的差别。

### 下降单纯型法

有些问题中， $f$ 不可导或者工程上求导代价极大<sup>1</sup>。这种情形下，假设函数值是连续的，我们有一种自然的思路，那就是采用不断试探的方法：在自变量为一维的情况下，给定一个初始区间，假设区间内有唯一的最小值，可以按照黄金分割的方法不断缩小区间以得到最小值。

上面的方法也可以被推广到自变量是高维的情形，对应的算法我们称为下降单纯型法(Downhill Simplex Method)。这一方法有一个更直观的称呼，即阿米巴(Ameoba)变形虫法。简单地讲，将一维空间上用两个点限制的区间不断变形的思路加以推广，在 $N$ 维空间中，我们也可以选择一个 $N+1$ 个点张成的超多面体，或称为单纯型(Simplex)，然后对这一单纯型不断变形以收敛到函数值的最小点。下降单纯型法的示例性代码如下所示。

<sup>1</sup> 有时我们需要遍历所有数据或者使用很大内存才能得到目标函数的导数，这种情况实际上在工程中比问题不可导要更常遇到。

```

1 void Ameoba() {
2 }

```

## 最速下降法

当 $f$ 可以比较容易地求导时，基于梯度的方法是首要选择，这也是我们将来讨论的重点类型。我们先来看一下梯度的定义。假设自变量

$$x = (x_1, x_2, \dots, x_N)^T \in \mathbb{R}^N$$

那么函数 $f(x)$ 在 $x$ 点的梯度可以写成

$$\nabla f(x) = \left( \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right)^T$$

梯度的几何意义，是 $f$ 在 $x$ 点函数值上升最快的方向，因此它是一个与 $x$ 维数相等的矢量。而利用梯度的优化方法，概念上就是每次都沿着负梯度方向按某步长前进一小步，这样的方法称为梯度下降法(Gradient Descent)。其更新公式为：

$$x \leftarrow x - \varepsilon \nabla f(x)$$

其中 $\varepsilon$ 控制着沿负梯度方向下降的速度，称为学习率(Learning Rate)。

很多工程上的目标函数，都具有可分解的特性，即整个训练集上的梯度可以表示为各个训练样本梯度的和。在这种情况下，一个可行但效率并不太高的并行实现就是将计算梯度的过程分解到各个数据划分上分别完成，然后将各部分的梯度相加并更新参数。显然，这样的计算过程非常容易在map/reduce框架下实现。然而每迭代一步，都要用到训练集所有的数据，大数据下可想而知这种方法的迭代速度。

同时，梯度下降的收敛速率可证明为线性收敛，线性收敛速率不大于令hessian矩阵特征值最小为 $a$ ，最大为 $A$

且存在锯齿搜索问题

<http://zh.wikipedia.org/wiki/>综上，梯度下降法的缺点是：靠近极小值时速度减慢。直线搜索可能会产生一些问题。可能会‘之字型’地下降。

在线学习(Online Learning)中，梯度下降的方法还有另外一种变形，也就是随机梯度下降(Stochastic Gradient Descent, SGD)的方法。SGD是一个重要的简化，在实际大数据的情况下，他比经典优化方法效果要好。因为经典优化方法算一次下降方向就已经需要很大计算量了。而SGD每一次迭代中，梯度的估计并不是精确的计算梯度,而是基于随即选取的一个样例。相对标准的梯度下降很快很多，有如下关键区别：标准梯度下降是在权值更新的每一步对多个样例求和，而随机梯度下降的权值是通过考查某个训练样例来更新的SGD需要的一次迭代计算量非常小，但是代价是他非常可能不是下降方向. 在理论上，SGD的性质少。需要一个有经验的人选择

很好的步长（学习速率）。如果标准误差曲面有多个局部极小值，随机梯度下降有时可能避免陷入这些局部极小值中。在计算上，SGD有个很大的缺点是不容易并行。

值得注意的是，Jeff Dean等人实现了一个随机梯度下降算法的分布式并行化，加快了模型训练速度。（Asynchronous Distributed Stochastic Gradient Descent, ），用以训练非凸的深度学习模型，效果比传统sgd更好。训练数据被分成多份（Data Shard），每份训练数据由一个模型训练集群复本（Model Workers）来处理。每个模型训练集群复本又可能包含很多台服务器，每台服务器将负责某些模型分区的训练。模型的参数统一存储在一系列参数服务器（Parameter Server）上，也做了水平分区。

## 拟牛顿法

### 节省空间的拟牛顿法-LBFGS

## 1.3 统计机器学习

机器学习(Machine Learning)是近年来得到快速发展和广泛应用的研究领域，它研究的是用数据或以往的经验优化计算机程序或算法的性能。从机器学习的方法论来看，可以大致分为统计的方法和非统计的方法。非统计的方法种类非常多，并且往往最后都归结于一个具体的优化问题，因此笔者认为可以通过深入掌握优化理论和算法，更加有效地把握各种非统计类方法。而统计类机器学习方法，虽然一般性的优化方法也广泛用到，三是也产生了一些在概率模型框架下系统性的方法和思路，下面我们把这一类的方法脉络稍加整理，供大家深入学习时做个简要纲领。

## 最大熵原理

统计机器学习中，指数族形式的分布由于求解的方便性，有非常重要的工程地位，我们先来看一下这一族分布形式产生的原因。要了解指数族形式产生的原因，需要先了解最大熵(Maximum Entropy)的原理。最大熵原理告诉我们，当我们在某些约束条件下选择统计模型时，需要尽可能选择满足这些条件的模型中不确定性最大的那个。如果采用熵作为统计不确定性的度量，这个问题就变成一个在这些约束下优化熵的问题。

## 指数族分布

出于上面对最大熵原理和求解过程的讨论，我们可以了解到指数形式的分布在统计机器学习中的重要地位。实际上，这一族分布是大多数模型在设计时的基本构件，也是我们学习统计机器学习必须掌握的基础。指数族分布的归一化形式(Canonical Form)可以表示为：

$$p(x|\theta) = h(x)g(\theta)\exp\{\theta^T u(x)\} \quad (1.5)$$

在这一形式中， $u(x)$ 是对变量 $x$ 的某个固定的矢量变换函数， $\theta$ 即为指数族分布的参数，而 $g(\theta)$ 为使得概率密度曲线下面积为一的归一化项。

指数族分布在建模上被广泛采用的,是因为一个重要的特性。这一特性是,指数族分布参数的最大似然估计,可以完全由其充分统计量(Sufficient Statistics)得到。这里的充分统计量,指的是训练集上变换函数 $u(x)$ 的统计量,即 $\sum_{i=1}^N u(x_i)$ 。在给定了充分统计量以后,  $\theta$  的最大似然解可以通过解下式求得:

$$-\nabla \ln g(\theta_{ML}) = \frac{1}{N} \sum_{i=1}^N u(x_i) \quad (1.6)$$

这一概念强调的是,在给定充分统计量以后,最大似然估计过程与数据无关。根据充分统计量的形式,我们很容易得出,无论什么样的指数族分布,我们都只需要遍历一遍数据,就可以得到最大似然解,这一点实际上对应了一个非常简便的map/reduce实现。这也是指数族分布在大数据运算上带给我们的最大便利性,请大家细心体会。由于指数族的分布形式与最大熵原理的本质联系,这一族的许多重要分布都可以从最大熵的角度加以解释。我们在下表中列举了几种重要的指数族分布形式,以及其主要用于描述的变量类型,希望对大家有所帮助。

**Table 1.1.** 若干重要指数族分布形式

分布	$u(x)$	解释	使用场景	示例
Gaussian	$\begin{bmatrix} x \\ x^2 \end{bmatrix}$	给定均值方差情形下熵最大的分布	一般实变量	
Gamma	$\begin{bmatrix} x \\ \ln x \end{bmatrix}$	给定均值方差, 且 $x > 0$ 情形下熵最大的分布	非负实变量	
Beta	$\begin{bmatrix} \ln x \\ \ln(1-x) \end{bmatrix}$	给定均值方差, 且 $x \in (0, 1)$ 情形下熵最大的分布	某区间内的实变量	
Multinomial	$\ln x$	给定均值方差, 且 $x \in (0, 1)$ 情形下熵最大的分布	离散变量	

从上面的示例中,还可以发现指数族分布的另一个重要特点,那就是这些分布都是单模态(Uni-modal)的。所谓单模态,可以理解为分布从几何形态上来看只有一个峰或者一个谷,这实际上说明了指数族分布虽然数学上使用方便,但其实际上的描述能力是有限的,并不适合于表达多种因素并存的随机变量。

## 混合模型和EM算法

由于指数族分布是单模态的,因而不适用于存在比较复杂的数据分布。为了解决这个问题,同时又能充分利用到指数族分布的一些方便的性质,工程领域产生了采用多个指数族分布叠加的部分来建模的实用方法,即混合模型(Mixture Model)。如果我们用 $q(x)$ 来表示某种指数族分布形式,那么其相应的混合模型为:

$$p(x|w, \Theta) = \sum_{k=1}^K w_k h(x) g(\theta_k) \exp\{\theta_k^\top u(x)\} \quad (1.7)$$



其中 $w = (w_1, \dots, w_K)$ 为各个组成分布先验概率，而 $\Theta = (\theta_1, \dots, \theta_K)$ 表示各个组成分布的参数。这一分布的图模型表达如下图所示：

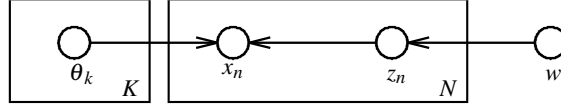


Fig. 1.1. 混合分布的概率图模型表示

在许多常见的机器学习模型当中，根据多个变量的条件依赖关系，可以用上面这样的有向图模型来比较清晰地表达整体的联合分布。有向图模型表达的信息是：图中的每一个节点代表一个随机变量，而给定了该变量所有入边对应的起始节点后，该变量的分布与其他所有变量都条件无关。需要指出，有向图模型本身只给出了条件依赖关系，并没有明确各条件分布的形式。一般来说，我们在工程中的思路中用图模型来表达我们先验的变量结构关系，然后对每个条件分布选取合适的指数族分布来建模，而混合分布模型就是了解这种工程思路的最典型例子。按照上面的有向图模型表示，我们引入了Multinomial变量 $z = (z_1, \dots, z_K)^T$ 来明确表示状态，可以把混合分布改写成结构更清晰的表达式：

$$p(x|w, \Theta) = \sum_z \prod_k w_k^{z_k} \left\{ h(x) g(\theta_k) \exp\{\theta_k^\top u(x)\} \right\}^{z_k} \quad (1.8)$$

在混合模型的最大似然求解过程中，EM(Expectation-Maximization)算法起着非常重要的作用。从上面的概率图模型例子可以看出，除了要求解的参数 $w, \Theta$ 和观测到的变量 $x$ ，还存在一个变量 $z$ ，我们把这样的变量称为隐变量(Hidden Variable)。EM算法就是为了解决有隐变量存在时的最大似然估计问题的。这是一种迭代的算法，每个迭代又可以分为E-step和M-step，在E-step阶段，我们将参数变量和观测变量都固定，得到隐变量的后验分布；而在M-step，我们将用得到的隐变量的后验分布和观测变量再去更新参数变量。以上面的混合分布问题为例，我们在EM算法的每一步迭代当中，都转而求解下面的辅助函数优化问题：

$$\max_{w, \theta} Q(w, \theta; w^{\text{old}}, \theta^{\text{old}}) = \max_{w, \theta} \sum_z p(z|X, w^{\text{old}}, \theta^{\text{old}}) \ln p(X, w, \theta|z) \quad (1.9)$$

由于此时的隐变量 $z$ 是离散的，因此等式右边为求和的形式，如果在其他问题中遇到的隐变量是连续的，那么只需要将求和号换成积分号即可。

对应于公式1.16，指数族混合分布EM算法的E-step和M-step可以很容易求出，其结果如下式：

$$\text{E-step: } \gamma_i(k) \triangleq p(z_k = 1 | \Theta^{\text{old}}, w^{\text{old}}, x_i) = \frac{w_k^{\text{old}} g(\theta_k^{\text{old}}) \exp\{u^\top(x_i) \theta_k^{\text{old}}\}}{\sum_l w_l^{\text{old}} g(\theta_l^{\text{old}}) \exp\{u^\top(x_i) \theta_l^{\text{old}}\}} \quad (1.10)$$

$$\begin{aligned} \text{M-step: } -\nabla \ln g(\theta_k^{\text{new}}) &= \frac{1}{N} \sum_{i=1}^N \gamma_i(k) u(x_i) \\ w_k^{\text{new}} &= \frac{1}{N} \sum_{i=1}^N \gamma_i(k) \end{aligned} \quad (1.11)$$

在混合分布的情形下，这样的一种分解使得许多非指数族分布的模型再进行最大似然估计时，其M-step形式上与简单的指数族分布是一致的，这也使得指数族分布工程上的便利性得以继续发挥。虽然M-step的形势与指数族最大似然估计的形式1.6非常相近，我们却不宜将等式右边的部分也称为充分统计量，因为这一过程是迭代进行的，需要多次访问数据才能完成最大似然估计，因此，简单地称其为统计量更为准确。

指数族分布的混合模型在工程中的应用同样很广泛，只要是单模态分布不易刻划的数据分布，都可以考虑用某种指数族分布叠加的方式来更精确地建模。常见的混合模型，例如高斯混合模型(Mixture of Gaussians, MoG)，以及PLSI(Probabilistic Latent Semantic Index)，后这可以认为是建立在Multinomial分布基础上的混合模型，在文本主题分析中有着广泛的应用。

需要注意的是，指数族混合分布的EM算法只是EM算法的一种较简单的特殊情况，这一算法广泛应用于各种隐变量存在的统计模型训练中，有关这方面更详细的理论和应用介绍，请大家参看[16]和[5]中更多的介绍。

## 贝叶斯学习

以上讨论的模型参数估计方法，都是在最大似然准则下进行的。最大似然准则，是把模型的参数看成固定的，然后找到使得训练数据上似然值最大的参数，这是一种参数点估计(Point Estimation)的方法。这样的点估计方法，在实际中如果遇到数据样本不足的情形，往往会产生比较大的估计偏差。对此，工程上常常用到贝叶斯学习(Bayes Learning)的方法论。为了介绍贝叶斯学习的基本概念，我们先从下面的贝叶斯公式入手了解一下其中的关键概念：

$$\underbrace{P(\theta|D)}_{\text{posterior}} = \frac{\overbrace{p(D|\theta)p(\theta)}^{\text{likelihood prior}}}{\underbrace{p(D)}_{\text{evidence}}} \quad (1.12)$$

在贝叶斯体系下，模型参数 $\theta$ 不再被认为是固定不变的量，而是服从一定分布的随机变量。在没有数据支持的情况下，我们对其有一个假设性的分布 $p(\theta)$ ，这称为先验分布(Prior)，而在观测到数据集 $D = \{x_1, \dots, x_N\}$ 以后，根据数据集上表现出来的似然值(Likelihood) $p(D|\theta)$ ，我们可以得到调整后的后验分布 $p(\theta|D)$ 。先验分布、后验分布和似然值之间的变换关系，就通过上面的贝叶斯公式表达出来。等式右侧的分母项，也是贝叶斯学习中的一个重要概念，称为Evidence，它可以展开表示为 $p(D) = \int p(D|\theta)p(\theta)d\theta$ 。由贝叶斯公式和这些重要概念出发，我们将三种常见的模型估计方法对比在下面的表中：

概率统计模型有两个主要任务：一是参数估计(Parameter Estimation)，二是预测(Prediction)。其中第二项任务指的是给定一组训练数据 $D$ ，评估某新的观测数据 $o$ 的概率。在最大似然体系中，参数估计是根据似然值最大化得到的点估计，而预测过程就直接利用估计出来的参数计算新的似然值即可。而在贝叶斯体系中，参数的点估计为其后验分布所代替，也就意味着参数在估计结果中具有不确定性(Uncertainty)，于是，在预测过程中，需要用积分的方式将不同参数的可能性都加以考虑，这是两者非常本质的差别。还有一种常见的参数估计方法，称为最大后验概

Table 1.2. 若干常见模型估计方法

	参数估计	预测
ML	$\hat{\theta}_D^{\text{ML}} = \arg \max_{\theta} p(D \theta)$	$p(o D) = p(o \hat{\theta}_D^{\text{ML}})$
Bayes	$p(\theta D) = p(D \theta)p(\theta)$	$p(o D) = \int p(o \theta)p(\theta D)d\theta$
MAP	$\hat{\theta}_D^{\text{MAP}} = \arg \max_{\theta} p(\theta D)$	$p(o D) = p(o \hat{\theta}_D^{\text{MAP}})$

率(Maximum a Posterior, MAP)方法, 本质上也是点估计方法, 只不过同样引入了先验部分来对参数作规范化, 因此, 其参数估计形式上是对贝叶斯后验概率求极值, 而预测过程则与最大似然情形一样。

### 共轭先验

贝叶斯方法的关键问题, 是如何选择公式1.12中的先验分布 $p(\theta)$ , 这一点两层含义: 一是如何选择先验分布的形式, 而是如何确定先验分布中的参数。之所以要讨论这个问题, 是因为虽然先验分布的形式是我们选择的, 但后验分布 $p(D|\theta)$ 的形式却无法选择, 而后验分布才是在使用中关键的分布, 其形式如果过于复杂, 会给实际应用带来很大困难。如果我们能够找到一种先验分布, 使得相应的后验分布也具有同样的形式, 无疑是最方便的。满足这种条件的先验分布, 我们就称为共轭先验(Conjugate Prior)。

对于指数族分布的似然函数而言, 可以很容易地发现共轭先验总是存在的, 这又一次证明了指数族分布在工程上带来的便捷性。对于公式1.5给出的指数族分布形式, 其共轭先验分布可以一般性地写成:

$$p(\theta|\eta) = \exp \left\{ \chi^T \theta - v g(\theta) - b(\chi, v) \right\} \quad (1.13)$$

值得注意, 这一指数族分布的共轭先验分布, 仍然是指数族的形式, 其用到的数学工具也就与前面的讨论一致。这一先验分布的参数 $\eta = \{\chi, v\}$ , 我们称之为超参数(Hyper-parameter), 这一 $\eta$ 的具体值实际控制着先验分布的具体形状。

将前文介绍的几种典型指数族分布与这一共轭先验通式相对照, 我们可以得到其对应的共轭先验: 1. 对于Gauss分布, 如果仅仅考虑其均值的不确定性, 则对应的共轭先验仍然是Gauss分布; 2. 对于Gamma分布, 其对应的共轭先验称为Wishart分布; 3. 对于Multinomial分布, 其对应的共轭先验是Dirichlet分布。Multinomial-Dirichlet这一共轭对是后文介绍的文本主题分析中非常重要的分布形式。

当模型为指数族分布, 并选择共轭先验的情形下, 对应的后验分布 $p(\theta|D)$ 可以很简单地写成下面的形式:

$$\tilde{\chi} = \chi + \sum_{i=1}^N u(x_i) \quad (1.14)$$

$$\tilde{v} = v + N \quad (1.15)$$

这里我们用变量上的波浪线代表后验。我们又一次看到，指数族分布的充分统计量在这里仍然发挥了核心作用，其结果是使得贝叶斯学习中得到后验概率的计算非常简便。需要特别指出的是，选择共轭的先验形式，从贝叶斯体系来看并没有太多理论上的必然性，而主要是为了满足工程上的方便性。因此，可以认为这完全是一种工程性的方案。

同样是从工程上来说，我们采用贝叶斯方案的目的是，为了具体的对模型参数进行约束，以提高估计的稳健型。因此，对超参数分参数的选择同样十分关键，因为超参数的具体取值决定了模型参数的自由程度。在实际应用中，我们可以根据一些领域知识和经验来设定超参数值，但是这样的方法有两个问题：1. 当模型过于复杂，超参数数目太多时，不太可能都根据经验相对合理地设定超参数。2. 采用这种主观的方式设定超参数，必然导致在一个固定的数据集上，参数估计的结果会随着主观超参数的不同而变化，这有些背离数据建模的客观性。因此，我们有必要探索一种数据驱动的超参数设定方法。

### 经验贝叶斯

数据读懂的超参数决定方法中，经验贝叶斯的方法值得大家注意。在公式1.12中，右边的分母，即evidence，是将模型参数积分后的似然值的期望。我们注意到，在似然值和先验部分的形式确定的前提下，evidence仅仅是先验部分的函数。从概念上来看，如果把evidence认为是超参数对应的似然值，我们也可以用优化evidence的方式来找到最优的超参数。这种根据数据来确定超参数的方法，就称为经验贝叶斯，其优化问题可以表示为：

$$\hat{\eta} = \arg \max_{\eta} \int \prod_{i=1}^K p(X_i | \theta_i) p(\theta_i | \eta) d\theta_i$$

由于是根据evidence来确定超参数，这一方法框架又称为Evidence Framework。需要说明，Evidence Framework除了能够用于确定超参数，同样可以用于在若干种先验部分形式中做选择，选择标准仍然是判断各种分布的evidence的大小。上式中还有一点需要特别注意，那就是我们是假设 $i = 1, \dots, K$ 个模型共享同一个先验分布。从后文的讨论可以得知，只有当 $K > 1$ 的时候，上面的经验贝叶斯问题才会有非退化的解。

一般来说，公式1.16是一个比较复杂的优化问题。不过在指数族分布的情形下，个优化问题仍然有着比较清晰地解法思路：我们注意到在公式1.16中， $X$ 为观测量， $\eta$ 为参数，而 $\theta$ 实际上是隐变量。因此，最直接的思路仍然是使用EM算法来求解。当 $p(x|\theta)$ 为指数族分布，而 $p(\theta|\eta)$ 为其共轭先验分布时，对应的EM辅助函数可以写成下面的表达式：

$$\begin{aligned} Q(\eta, \eta^{\text{old}}) &= \sum_{i=1}^K \int_{\theta_i} p(\theta_i | X_i, \eta^{\text{old}}) \ln p(X_i, \theta_i | \eta) d\theta_i \\ &= \sum_{i=1}^K \int_{\theta_i} p(\theta_i | X_i, \eta^{\text{old}}) \ln p(\theta_i | \eta) d\theta_i + \text{const} \\ &= \sum_{i=1}^K \int_{\theta_i} p(\theta_i | \tilde{\eta}_i^{\text{old}}) \ln p(\theta_i | \eta) d\theta_i + \text{const} \end{aligned} \quad (1.16)$$

请注意我们在这里用到了共轭先验的性质，即后验分布有着与先验分布一样的行为，并且将地 $i$ 个模型的后验超参数记为 $\tilde{\eta}_i^{\text{old}}$ 。仔细观察这一结果，如果我们把 $\theta$ 当成数据， $\eta$ 当成参数，那么已知的后验分布 $\frac{1}{K} \sum_{i=1}^K p(\theta_i | \tilde{\eta}_i^{\text{old}})$ 可以看成是数据的分布，而 $\ln p(\theta_i | \eta)$ 则相当于参数 $\eta$ 在此数据集上对应的似然值。于是，对此辅助函数的优化，相当于是在此数据分布上对 $\eta$ 进行最大似然估计。又由于 $p(\theta | \eta)$ 也是指数族分布，其最大似然估计可以通过充分统计量得到。因此，易于验证，该经验贝叶斯问题的E-step和M-step可以表示成下面的形式：

$$\begin{aligned} \text{E-step: } \tilde{\chi}^{\text{old}} &= \chi^{\text{old}} + \sum_{i=1}^N u(x_i), \quad \tilde{\mathbf{v}}^{\text{old}} = \mathbf{v}^{\text{old}} + N \\ \text{M-step: } \langle \theta, g(\theta) \rangle_{p(\theta | \eta^{\text{new}})} &= \frac{1}{K} \sum_{k=1}^K \langle \theta, g(\theta) \rangle_{p(\theta | \tilde{\eta}_k^{\text{old}})} \end{aligned} \quad (1.17)$$

其中的E-step就是采用共轭先验的情形下后验的计算公式，而M-step是一个关于 $\eta^{\text{new}}$ 的方程，此方程是否有闭式解，要根据具体的指数族形式而定。

在上面介绍的一些典型机器学习方法当中，可以发现指数族分布及其充分统计量在计算流程中起着非常关键的枢纽作用。我们可以注意到，不论是指数族分布的最大似然解，指数族混合分布的最大似然解，还是指数族分布的贝叶斯学习，如果采用map/reduce的计算框架，都可以用图1.3来描述。

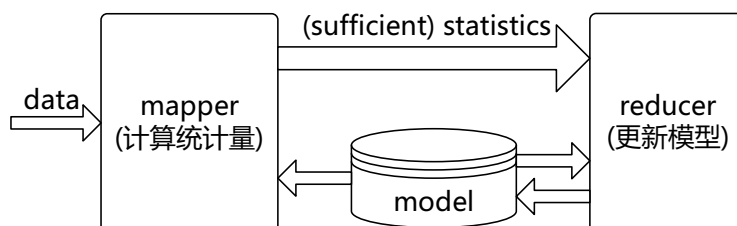


Fig. 1.2. 指数族分布map/reduce学习框架

从这一计算流程可以看出，对于大规模数据上的许多机器学习计算问题，map/reduce是一个可行的选择：因为在机器之间交换的数据只是统计量或者充分统计量，其空间复杂度仅仅与模型的参数数目有关，与数据的多少并无直接关系。因此，不论多大规模的数据集合，总是可以利用map/reduce得到一个可行的解决方案。不过，当算法需要多次迭代才能完成的时候，由于需要在每次map过程中重新加载数据，使得整个过程的I/O负担变得较重，从而降低整个计算过程的效率。可以说，对大规模数据上的指数族分布相关参数估计问题，map/reduce是一个通用的可行方案，但是并不是最高效的方案。当数据的规模可以承受时，可以考虑类似于spark之类的更加高效的计算框架。对然这里的分析主要是针对上面指数族分布相关的估计问题，但是对于一般的基于梯度的优化问题也同样成立。

## 变分法



## 合约广告中的核心技术

### 2.1 广告投放机系统架构

一个概念性的合约广告投放系统是什么样的呢？我们在下图中列出了其主要构成。广告系统接受用户访问页面是产生的广告请求，根据用户和上下文标签从索引中找到符合投放条件的广告合约，然后根据预先计算好的在线分配方案决定返回哪个广告。

在线投放的展示和点击日志经过Data highway，一方面进入离线分布式计算平台以后，通过日志的整理，完成Planning计算，即制定在线分配策略的算法，再将分配方案送给线上投放机使用。另一方面，日志也送到流式计算平台，在反作弊和计价的基础上，再对索引进行尽可能快速调整。

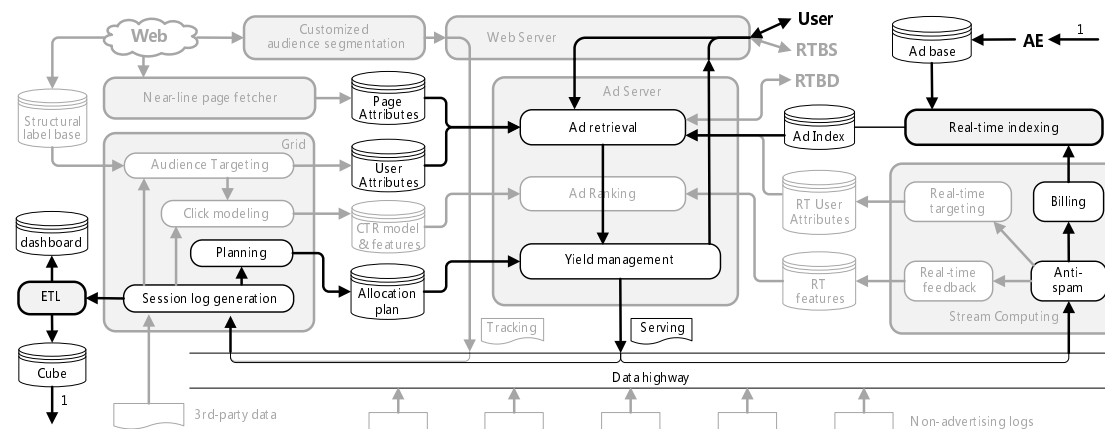


Fig. 2.1. GD广告系统架构示意

Ad server并非专门应用于合约广告系统，而是在各种广告系统中都非常常见的核心模块之一。Ad server本身并没有太复杂的计算逻辑，他所要做的，是将若干关键计算过程平滑地连接起来以满足广告请求，而其优化的目标也是整个系统的平均延时和吞吐率。

## 2.2 在线分配问题描述

在线分配问题，指的是在通过对每一次广告展示进行实时在线决策，从而达到在一定的时间范围内，满足某些量的约束条件的前提下，尽可能地优化广告市场整体收益的过程。大家很容易理解，这个问题计算上最困难的地方在于“在线”，也就是信息尚不全面的时候作出决策；而系统上最困难的地方在于分配策略需要是弱状态的，同时各广告服务器之间耦合程度也要尽量低。我们在下面的讨论中也将重点关注这两个方面。

### 2.2.1 在线分配问题表达

为了讨论在线分配问题，我们需要对计算广告框架问题，即公式3.1做一些变化，并加入一些约束条件，以得到更适合于在线分配问题研究的形式。

#### 供给与需求二部图

从上述的GD问题可以看出，在线分配问题有两个主要的挑战：一是要在量的约束下优化质；二是要实时对每一次展示作出决策。直接在这两个要求下探讨问题，会使得求解过程相当困难。因此，在在线分配问题中，我们一般将此问题简化为一个二部图(Bipartite Graph)匹配的问题，这里的“二部”，指的是代表广告库存的供给节点(其集合记为 $I$ )，是由所有处标签都相同的广告库存合并成的，和代表广告合约的需求节点(其集合记为 $A$ )。

供给节点、需求节点和在线分配二部图的示例见图2.2.1。在这个示例中，下方的六个节点为供给节点，而上面的三个节点为需求节点。如果某个供给节点的受众标签能够满足某个需求节点的要求，我们就在相应的两个节点间建立一条连接边。我们把这个二部图记为 $G = (I \cup A, E)$ ，其中 $E$ 为 $I$ 与 $A$ 之间边的集合，并用 $\Gamma(a)$ 表示所有与需求节点 $a$ 相邻的供给节点的集合，而用 $\Gamma(i)$ 表示所有与供给节点 $i$ 相邻的需求节点的集合。我们的任务，就是要求解由 $i \in I$ 到 $a \in A$ 的 $y_{ia}$ ，使得满足供给方和需求方的约束的同时，某个目标函数达到最大。

请大家注意，与3.1的计算广告一般问题相比，这样的二部图结构实际上假设了在同样一组供给和需求节点之间发生的广告展示，其目标函数或回报 $r$ 是没有差别的，这显然不够准确，但却是更直接地研究在线分配算法的合理近似。在这一近似下， $r$ 由 $(a, u, c)$ 组合的函数，变成了供给节点 $i$ 和需求节点 $a$ 的函数，我们将其记为 $r_{ia}$ 。为了方便起见，从分配问题的物理意义出发，我们往往还假设整体的收益或目标函数是可分的[]，这一目标函数表示为如下的形式：

$$F(s, x) = \sum_i s_i r_{ia}(x) \quad (2.1)$$



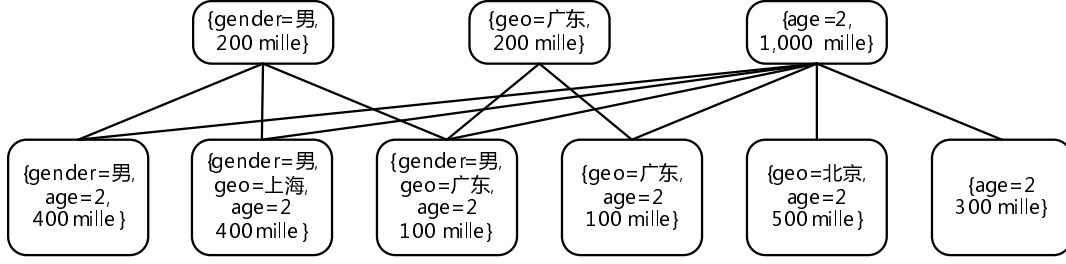


Fig. 2.2. Online Allocation中的二部图匹配问题示意

其中 $s_i$ 为供给节点 $i$ 的总供给量，而 $x = \{x_{ia}\}_{|I| \times |A|}$ 表示中的每个元素表示 $s_i$ 分配给合约 $a$ 的比例。很自然，这里的 $x$ ，就是我们在在线分配问题求解的变量。

在线分配问题的这一目标函数，直观上看与3.1的一般广告问题目标大有不同，不过这实际上是通过二部图假设简化后得到的表示，请大家注意体会这样的根据实际工程问题进行合理假设的方法。另外，在这种表达中，供给节点的数目会随着定向条件的增加而呈几何级数上升，也会使得对应的分配问题变得过于复杂而无法有效求解。

### 需求约束

在线分配问题的第一个约束条件，是分配给广告合约 $a$ 的展示数目要至少等于其约定的量 $C_a$ ，这个约束称为需求约束(demand constraint):

$$\sum_{i \in \Gamma(a)} s_i p_{ia} x_{ia} \geq d_a, \forall a \in A \quad (2.2)$$

注意其中的 $p_{ia}$ 为将供给节点 $i$ 连接到需求节点 $a$ 的代价的一般性表示，其具体意义我们将在后面举例说明。简单起见，我们一般都假设这一需求约束是线性的，实际上这也已经能满足所有常见场景中的需求。

### 供给约束

在线分配问题的另一个约束条件，是每个供给节点被分配出去的量不能多于其总量，这个约束称为供给约束(supply constraint)，其意义不言自明。我们将它表示成下面的形式：

$$\sum_{a \in \Gamma(i)} x_{ia} \leq 1, \forall i \in I$$

### 问题框架

根据上面的讨论，我们从公式3.1定义的计算广告目标出发，引入量的约束，得到下面的在线分配优化问题框架表示：

$$\begin{aligned}
& \max \sum_{(i,a) \in E} F(s, x) \\
\text{s.t.} \quad & \sum_{a \in \Gamma(i)} x_{ia} \leq 1, \forall i \in I \\
& \sum_{i \in \Gamma(a)} s_i p_{ia} x_{ia} \geq d_a, \forall a \in A \\
& x_{ia} \geq 0, \forall (i, a) \in E
\end{aligned} \tag{2.3}$$

除了供给约束和需求约束，上式中还有第三个比较容易理解的约束，它用以保证分配变量非负。公式2.3是一个比较一般性的数学表达，不仅仅适用于GD问题，也适用于其他量约束下的在线分配问题，对这个问题的理论探讨和系统实现，是我们在本章要讨论的重点。而得到的一些算法和结论，也不仅仅可以用于合约式广告系统，在后文介绍的竞价广告系统或广告交易市场中也有着广泛的应用。

如果可以离线对公式2.3进行决策，那么这是一个一般的带线性约束的优化问题。然而在广告投放实际环境中，不可能达到全局最优，而是必须对每次广告展示马上做出决策，这就要求我们设计一种比较聪明的策略，使得整体流量情况尚不明朗时，仍然可以相对合理地作出决策，而最终目的是全部流量上的分配结果与离线最优化的结果尽量接近。

在线分配问题研究的主要方向有两个，一是尽可能充分地利用历史上的投放数据来指导分配过程，以得到实用高效的分配方案；二是从理论角度研究当未来的流量有很大不确定性的情况下，如何设计分配策略才能获得最差情况下尽可能好的效果。我们将在下文中分别介绍这两方面的一些基本思路和方法。

### 2.2.2 在线分配问题举例

由于我们讨论的在线分配技术并不是仅仅适用于GD问题，有必要先对各种典型问题中的目标函数和约束条件做一下分析。除了本章中提到的这些问题，在本书其他章节中还会碰到一些与在线分配技术紧密联系的问题，我们将在后文遇到时再做讨论。

#### GD问题

在线分配的最重要应用，就是上面提到的GD分配问题。我们主要考虑按CPM结算的市场。在GD合约的情形下，在线分配的目标函数实际上是由合约的具体约定来定义的。这里的具体约定，主要指的是指的是当合约未完成(under delivery)时的惩罚。如果假设惩罚为线性，并把合约 $a$ 的单位未完成量的惩罚记为 $p_a$ ，并引入代表合约 $a$ 未完成量的变量 $u_a$ ，那么GD的优化问题可以写成：

$$\begin{aligned}
& \max \sum_{a \in A} p_a u_a \\
\text{s.t.} \quad & \sum_{a \in \Gamma(i)} x_{ia} \leq 1, \forall i \in I \\
& \sum_{i \in \Gamma(a)} s_i p_{ia} x_{ia} + u_a \geq d_a, \forall a \in A \\
& x_{ia} \geq 0, \forall (i, a) \in E, \\
& u_a \geq 0, \forall a \in A
\end{aligned} \tag{2.4}$$

引入变量 $u_a$ 后，这个问题的形式与在线分配的一般问题2.3就一致了，因此我们可以用同样的框架来解决。

### MRA问题

GD的分配方式在实际的品牌广告市场上存在的问题，其原因可以这样来说明：假设广告主甲在合约中要求地域为北京的人群，其实有一层潜在的含义，那就是得到的流量尽可能符合北京人群的自然分布。假设在另一个合约中，广告主乙要的是北京的男性人群，那么为了同时满足两个合约，那么Ad Server会倾向于将北京的男性人群分配给乙。这样一来，甲得到的流量中，女性的比例就会显著高于自然分布，这并不是广告主想要的。

为了尽可能缓解上述问题，我们可以对在线分配的目标函数做一些加工，使得系统在完成合约的同时，尽可能给每个广告主以代表性的流量分布。这样的分配目标，我们称为最大代表性分配(Maximally Representative Allocation, MRA)问题。Preston McAfee等人最初提出这一问题，并对其分配算法进行了研究。如何对“代表性”进行定量的描述，是此问题的关键，如果采用实际分配流量分布与目标流量分布的举例来表示这一差异，那么MRA问题的目标函数可以表示为：

$$\max \sum_{(i,a) \in E} s_i(x_{ia} - \theta_a)^2 / \theta_a + \sum_{a \in A} p_a u_a$$

这里的 $\theta_a$ 代表的是合约 $a$ 的需求量占整体供给的比例，这样做的目的是为了让分配尽量均衡。该优化问题的若干约束与上面的GD问题一样，这里不再重复。

### Ad words问题

Ad words问题，也被称为Budgeted bidder问题，讨论的是在竞价广告环境下，给定各个广告主预算的预算，整体优化市场营收的问题。我们主要考虑按CPC结算的竞价市场，在这种情形下，公式2.4中的目标函数和需求约束都有所变化，其中需求约束体现为如下的形式：

$$\sum_{i \in \Gamma(a)} \mu_{ia} y_{ia} \leq d_a, \forall a \in A \quad (2.5)$$

而目标函数具体化为：

$$\max \sum_{(i,a) \in E} r(a, u_i, c_i) y_{ia}$$

## 2.3 分配问题优化算法

我们需要对未来一段时间内每一次广告展示进行实时在线决策，从而达到满足某些量的约束条件的前提下，尽可能地优化广告市场整体收益的过程。假定未来一段时间内需要投放的合同是已知

的, 如果广告流量的分布在各个循环周期内是近似一致的, 那么在线分配的问题就可以转变为离线计算的问题去近似求解: 只需要上一个流量周期的流量近似为未来的流量, 根据现有合约生成Demand constraints, 再将优化的结果作为下一个周期内的分配策略在线执行即可。

虽然即时决策的限制使得历史数据的预估只是近似, 但流量预估对于投放决策仍可以起到非常强的指导作用。因此, 实用的在线分配算法, 都需要在根据历史数据的流量预测基础上展开。而在流量分布无法预估, 即对未来流量完全没有假设的前提下, 分配算法的效率学者们也进行了深入的研究, 我们在这一章节最后也做一下介绍。

### 2.3.1 直接求解的原始分配方案

在实际的工程系统中, 假定流量的分布是平稳的, 我们会利用历史流量数据来拟合未来流量来构建供给(流量)和需求(合同)节点之间的二部图, 把在线分配转化成离线问题, 离线的对公式2.3进行决策。由于这是一个一般的带线性约束的优化问题, 当优化目标为线性函数或二次函数时, 是一个标准的LP或QP问题, 所以可以采用一些优化工具包来直接求解该问题。当所求解的问题规模较小时, 比如不按人群来售卖, 售卖的保量广告位较少, 广告主较少时, 二部图的规模可以直接采用相关的优化工具包单机直接求解。

在大型的商业系统中, 流量往往在billion(十亿)以上的级别, 由于定向条件的复杂性, 流量端节点的数目会随着定向条件的增加而呈几何级数上升, 合同端节点数也至少在千级别之上, 业务复杂性带来的边 $|E|$ 的数目在million级别以上, 这就会使得对应的分配问题变得过于复杂而无法直接采用优化工具包有效求解。举例来说,  $n$ 为变量的个数(正比于 $|E|$ ), 求解线性规划问题的经典算法如内点法(复杂度 $n$ 的多项式时间), 单纯形法(复杂度在 $O(n^{1.5} - n^2)$ 量级), 在大型商业系统中按小时级别的定期更新求解是近乎不可能的。

### 2.3.2 基于对偶算法的紧凑分配方案

商业系统中不仅考虑离线训练模型时的复杂度, 更需要考虑的问题是线上的快速响应。模型的分配策略不能过于庞大以至于给server带来内存和计算上的很大负担, 而前述原始分配方案中求解出来的原问题的方案过于庞大(变量数正比于 $|E|$ ), 因此我们往往需要一个紧凑分配方案(Compact allocation plan)。

除了紧凑性的要求, 如果分配策略能做到一定程度上无状态, 也对于Ad Server的实现非常有利: 因为这样的话, 多台Ad Server的物理机器之间就不需要频繁进行同步以完成状态更新, 而是根据预先计算好的策略进行投放即可。这对与系统的稳健性和扩展性, 有着非常大的好处。

是否存在某种紧凑的分配策略, 其模型的参数的数目仅仅与合约总数有关, 当线上serving时只利用该模型和一些合同方的信息就能得到一个近似最优解呢。Yahoo的公开paper cite Vee中就提到了这样的一个方案, 证明了对公式2.3这样的在线分配问题, 通过求解对偶问题并只保留局部的对偶解, 就能恢复出原问题的最优解, 构造出同时满足紧凑和无状态的分配策略。

我们对该求解方案做一个简单的介绍, 并给出一个形象的例子以使得读者深刻理解其中的变换过程。我们知道, 每个线性规划问题(称为原问题)都可以变换为一个对偶问题, 原问题的每个

约束条件对应着一个对偶变量，当约束条件越难满足，其对应的对偶变量的值则越大，对偶变量对分配方案有着明确的物理意义，在经济学中又被称作影子价格。但对偶问题的解也并不是紧凑解，其变量数目正比于约束的数目，包括供给方(流量)约束和需求方(合同)约束，前者变量的量级数十万甚至百万千万，但后者的量级在而合同级别。为了分配方案的紧凑性，可否只保留需求方(合同)约束对应的对偶变量，通过数学变换恢复出需求方(合同)约束的对偶变量和分配率 $x$ 呢。在citeVee中，作者就给出了这样的一个方案。要给出这一恢复方法，我们需要先写出2.3的拉格朗日：

$$x_{ia}^* = \max \left\{ 0, g_{ij} \left( -\sum_k a_{ij}^k \alpha_k - \beta_j \right) \right\} \quad (2.6)$$

在[44]文中，通过对相应对偶问题的K.K.T条件的分析，推导得到了一个由 $\alpha$ 恢复 $\beta$ 和 $x$ 最优解的计算方法：

$$\sum_{j \in \tau(i)} g_{ij}(\alpha_j - \beta_i) = 1 \quad (2.7)$$

$$x_{ia}^* = \max \left\{ 0, g_{ij} \left( -\sum_k a_{ij}^k \alpha_k - \beta_j \right) \right\} \quad (2.8)$$

由于 $\alpha$ 的维数正比于合约数目 $|A|$ ，远远小于 $x$ 的维数(正比于 $|E|$ )，因此可以认为这是一个紧凑分配方案。因此，我们只需要在一部分历史数据上通过解对偶问题得到 $\alpha$ ，就可以得到在线时使用的分配比率。

下面我们给出一个简单的例子，和对应的matlab的模拟实验代码来加深读者的印象。假设四个supply,  $s=[40, 70, 60, 80]$ ; 三个demand,  $d=[60, 80, 80]$ ; 目标函数为最大代表性分配(MRA问题)，约束包括供给约束，需求约束，非负约束。二部图如下：

利用xpress等优化工具包求得合同节点的对偶变量 $\alpha_j$ ：

$\alpha=[0.0122, 0, 0.0064]$

已知合同节点的对偶变量，还原流量节点的对偶变量 $\beta_i$ 的matlab仿真代码和结果：

```

1 function beta = get_supply_dual(alpha, G, s, d)
2 [m, n] = size(G);
3 S = s * G;
4 theta = d ./ S;
5 beta = zeros(1, m);
6
7 for i = 1:m
8     tmp1 = 0;
9     tmp2 = 0;
10    for j = 1:n
11        if G(i, j) > 0
12            tmp1 = tmp1 + theta(j) * (alpha(j) + 1);
13            tmp2 = tmp2 + theta(j);

```

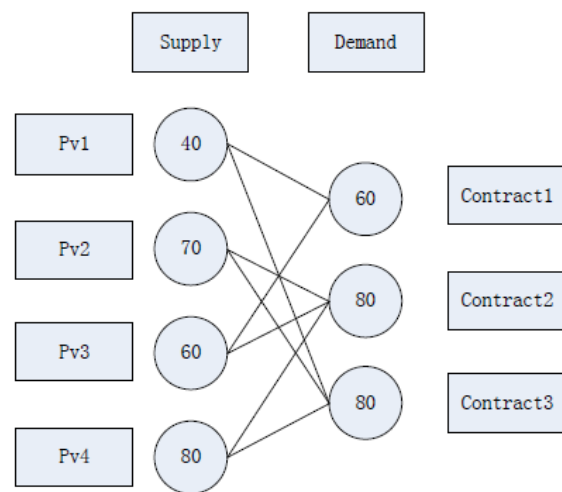


Fig. 2.3. 仿真实验

```

14         end
15     end
16     tmp1 = tmp1 - 1;
17     beta(i) = tmp1 / tmp2;
18     if beta(i) < 0
19         beta(i) = 0;
20     end
21 end

```

还原原问题的原始解 $x_{ij}$ :

```

1 function x = alocation_plan(alpha, beta, G, s, d)
2 [m, n] = size(G);
3 S = s * G;
4 theta = d ./ S;
5 x = zeros(m, n);
6
7 for i = 1 : m
8     for j = 1 : n
9         x(i, j) = max(0, theta(j) * (1 + alpha(j) - beta(i)));
10    end

```

```
11 end
```

对比直接求解原问题的解:

```
1
2 function f = f_origin(x)
3     s = [40, 70, 60, 80];
4     d = [60, 80, 80];
5
6     f = 0;
7     G = [1 0 1;
8          0 1 1;
9          1 1 0;
10         0 1 1];
11     S = s * G;
12     theta = d ./ S;
13     for j = 1:3
14         for i = 1:4
15             if G(i, j) > 0
16                 tmp = x((i - 1) * 3 + j) - theta(j);
17                 f = f + s(i) / theta(j) * tmp * tmp;
18             end
19         end
20     end
21
22     f = f * 0.5;
23 end
```

```
1 clear;
2 clc
3 s = [40, 70, 60, 80];
4 d = [60, 80, 80];
5 G = [1 0 1;
6      0 1 1;
7      1 1 0;
8      0 1 1];
9 S = s * G;
10 theta = d ./ S;
11 A = zeros(3 + 4 + 8, 12);
12 for j = 1:3
```

```

13     for i = 1:4
14         if G(i, j) > 0
15             id = (i - 1) * 3 + j;
16             A(j, id) = -s(i);
17         end
18     end
19 end
20 idx = 3;
21 for i = 1 : 4
22     for j = 1:3
23         if G(i, j) > 0
24             id = (i - 1) * 3 + j;
25             A(idx + i, id) = 1;
26         end
27     end
28 end
29 A(8:19, :) = -eye(12);
30 b = zeros(19, 1);
31 b(1:3) = -d';
32 b(4:7) = ones(4, 1);
33 b(8:19) = zeros(12, 1);
34 x0 = zeros(12, 1);
35 [x, fval] = fmincon(@f_origin, x0, A, b)

```

在实际应用中, 由于使用所有历史数据求解上述问题规模太大, 需要对数据作一些采样以便更高效地得到分配方案。关于采样的方法, 以及采样以后该问题求解的稳定性分析, 文[44]中也都做了详细讨论, 有兴趣的读者可以进一步探索。

### 2.3.3 启发式的分配方案HWM

上述的根据历史流量数据来构造紧凑分配方案的方法原理上可行, 但在实际的工程应用中仍然显得有些复杂, 比如离线仍要耗费大量时间求解对偶解。我们希望实现一种快速算法, 保持前述方法紧凑分配, 无状态的特性, 效果上也能近似最优。前述方案中通过合同节点的对偶变量(是否容易满足约束)即可恢复最优解, 受上面的讨论启发, 我们可以发现, 在最优化目标为MRA问题时, 只要大体确定好每个合同在分配中的相对优先级, 以及分配时得到某次展示的概率, 就可以构造出一种直觉上可行的在线分配方案。HWM(High Water Mark) 算法[13]就是这样一种方案, 虽然其数学上不是完全严谨, 但是由于根据历史数据来制定分配方案本身就具有相当程度的近似, 因此其实际效果也相当不错, 又加上工程上的便利性, 在使用的在线分配方案中可以考虑采用。



HWM算法的关键有两点，一是根据历史流量确定每个广告合约资源的紧缺程度，并进而得到分配优先级；二是根据优先级

```

1 void hwm_plan(vector<float> & supplies, vector<float> & demands,
2               vector<vector<int> > demandLinks,
3               // 上面二维数组为满足各需求节点的所有供给节点号列表
4               vector<int> & orders, vector<float> & rates) {
5     int supplyNum = supplies.size();
6     int demandNum = demands.size();
7
8     // 生成个需求节点对应总供给
9     vector<float> eligibles;
10    eligibles.assign(demands.size(), 0.0f);
11    for (int d = 0; d < demandNum; d++)
12        for (int i = 0; i < demandLinks[d].size(); i++)
13            eligibles[d] += supplies[demandLinks[d][i]];
14
15    // 将需求节点按照总供给生成分配顺序
16
17    // 按分配顺序依次处理各需求节点
18    vector<float> remains = supplies;
19    for (int i = 0; i < demandNum; i++) {
20        int d = orders[i];
21
22        for (int j = 0; j < demandLinks[d].size(); j++) {
23            int s = demandLinks[d][j];
24            remains[s] -= min(remains[s], rates[d] * supplies[s]);
25        }
26    }
27 }

```

针对之前仿真实验的例子，我们也可以根据HWM算法算出分配率。

根据上面离线生成的分配方案，也即对每个需求节点计算出来的分配优先级(order)和分配率(rate)，我们可以很方便地在线上服务中对每次展示做出简单的决策。决策的基本逻辑是：根据分配顺序依次检查各个符合条件的候选，然后按照他们分配率对应的概率来随机选择一个合约投放。

```

1 int hwm_serve(vector<int> & candidates,
2               vector<int> & orders, vector<float> & rates) {
3     int candNum = candidates.size();

```

```

4
5 // 将所有候选按照分配优先级排序
6 for (int c1 = 0; c1 < candNum; c1 ++)
7     for (int c2 = c1 + 1; c2 < candNum; c2 ++)
8         if (orders[c1] > orders[c2]) {
9             SWAP(candidates[c1], candidates[c2]);
10            SWAP(orders[c1], orders[c2]);
11        }
12
13 // 对此次展示产生分配随机数
14 float randValue = (float)rand() / RAND_MAX;
15
16 //计算累积serving rate的截断位置
17 float accRate = 0.0f;
18 for (int c = 0; c < candNum; c ++) {
19     accRate += rates[c];
20
21     // 如果返回-1, 则将此次展示交还server, 再从其他渠道查询广告
22     if (accRate >= 1.0f)
23         return -1;
24
25     if (accRate >= randValue)
26         return c;
27 }
28
29 return -1;
30 }

```

### 2.3.4 综合分配方案Shale

对于前述的基于对偶算法的紧凑分配方案, 虽然在线分配时确实达到了紧凑和无状态的特性, 但是线下基于Xpress等优化工具进行求解的代价仍然是昂贵的, shale中对求解对偶变量的步骤进行了优化, 采用prime-dual的算法迭代进行求解, 每次迭代的过程中改善对偶解。

```

1 clear
2 clc
3 s = [40, 70, 60, 80];
4 d = [60, 80, 80];
5 G = [1 0 1;
6      0 1 1;

```

```

7      1 1 0;
8      0 1 1];
9  S = s * G;
10 alpha = zeros(1,3);
11 beta  = zeros(1, 4);
12 N = 10;
13 for n = 1:N
14     beta = get_supply_dual(alpha, G, s, d);
15     alpha = get_contract_dual(beta, G, s, d);
16 end
17 alpha
18 beta
19 x = allocation_plan(alpha, beta, G, s, d)

```

其中调用的子函数源码如下

已知合同节点的对偶变量，求流量节点的对偶变量

```

1 function beta = get_supply_dual(alpha, G, s, d)
2 [m, n] = size(G);
3 S = s * G;
4 theta = d ./ S;
5 beta = zeros(1, m);
6 for i = 1:m
7     tmp1 = 0;
8     tmp2 = 0;
9     for j = 1:n
10        if G(i, j) > 0
11            tmp1 = tmp1 + theta(j) * (alpha(j) + 1);
12            tmp2 = tmp2 + theta(j);
13        end
14    end
15    tmp1 = tmp1 - 1;
16    beta(i) = tmp1 / tmp2;
17    if beta(i) < 0
18        beta(i) = 0;
19    end
20 end

```

已知流量节点的对偶变量，求合同节点的对偶变量

```

1 function alpha = get_contract_dual(beta, G, s, d)

```

```

2  [m, n] = size(G);
3  S = s * G;
4  theta = d ./ S;
5  alpha = zeros(1, n);
6  for j = 1:n
7      tmp1 = 0;
8      tmp2 = 0;
9      for i = 1:m
10         if G(i, j) > 0
11             tmp1 = tmp1 + s(i) * theta(j) * (1 - beta(i));
12             tmp2 = tmp2 + s(i) * theta(j);
13         end
14     end
15     tmp1 = d(j) - tmp1;
16     alpha(j) = tmp1 / tmp2;
17     if alpha(j) < 0
18         disp('alpha_□<_0')
19         alpha(j) = 0;
20     end
21 end

```

通过prime-dual方法得到的alpha，比较后发现和前述直接通过xpress等优化工具求解的alpha一致。在得到了合同的对偶解后，之后的算法和前述citeVee就一样了。但基于迭代的对偶问题求解无疑更节省了线下的计算时间，同时也能更好的支持插入新合同时的增量incremental求解。

### 2.3.5 最差分配性能研究

对在线分配问题的最差性能，即对未来流量完全没有假设的前提下分配的效率，学者们进行了深入的研究。虽然这方面的研究并不一定对实用的系统有非常直接的帮助，却对我们理解在线分配问题的本质特点和算法方向有重要指导意义，因为我们也要做一下介绍。

最差性能研究的指标，主要是某种在线分配策略的有效性。有效性可以如下定义：如果我们能够预知所有的流量情况，那么可以根据全局的信息求得一个分配的最优解；但是由于分配是在线执行，最优解并不一定达到，如果某种在线分配策略在最差情形下能够达到上述最优解目标函数的 $\epsilon$ 倍，那么我们就说这一分配方案是 $\epsilon$ -competitive的。显然，这里的 $\epsilon$ 是一个 $[0, 1]$ 之间的数，也就是该分配方案有效性的度量。既然是讨论最差情形下的有效性，那么流量预测是可以被忽略的部分，因为我们需要同时考虑哪些预测完全不准确的极端情形。我们来看看对在线分配最差性能的一些基本讨论思路。

公式2.4是一个典型的带约束优化问题，根据第二章中介绍的最优化基础知识，我们知道可以应用拉格朗日乘子法来分析这一问题。问题2.4的Lagrangian可以表达为：

$$F(s, x) + \sum_i \beta_i \left[ \sum_{a \in \Gamma(i)} s_i x_{ia} - s_i \right] + \sum_a \beta_a \left[ \sum_{i \in \Gamma(a)} s_i p_{ia} x_{ia} - d_a \right] - \sum_{(i,a)} \gamma_{ia} s_i x_{ia} \quad (2.9)$$

在[18]中, 作者给出了在Free Disposal<sup>1</sup>前提下, 对该问题进行在线分配的一种优化方案框架。由于不考虑流量预测, 这里把每次展示都当作一个独立的供给节点进行决策。该方案有如下的几个步骤:

1. 初始化每个需求约束的对偶变量  $\beta_a \leftarrow 0$ ;
2. 当一次展示  $i$  到达时, 令  $a' \leftarrow \arg \max_a r_{ia} - \beta_a$  取得最大值的广告合约  $a$ 。(如果该值对所有的广告都为负, 则所有合约都不需要分配。)
3. 令  $x_{ia'} = 1$ , 如果  $a'$  已经被分配了  $d_{a'}$  次展示, 令  $i' \leftarrow \arg \min x_{ia'}$ , 并将  $x_{i'a'}$  设置为 0。
4. 在对偶问题中, 令  $\alpha_i = r_{ia'} - \beta_{a'}$ , 并通过一定的更新规则来更新  $\beta_{a'}$ 。不同的更新规则对应了不同的分配算法, 也相应地会导致不同的分配性能。

[18]中对几种典型的  $\beta_{a'}$  的更新策略进行了讨论, 并且给出了一种有效性为  $(1 - 1/e)$ -competitive 的分配方案, 实际上, 可以证明这是在线分配问题可以达到的有效性的上界。我们将文中讨论的几种在线分配策略在下表中做一个简单的对比说明:

**Table 2.1.** 若干在线分配策略的对比

策略	描述	有效性
Greedy	$\beta_a$ 是分配给 $a$ 的前 $d_a$ 个高权重展示中最低的权重, 也即 $a$ 接受一个新的展示需要抛弃的权重。	$\frac{1}{2}$ -competitive
Uniform Weighting	$\beta_a$ 是分配给 $a$ 的前 $d_a$ 个高权重展示的权重的算术平均。如果分配给 $a$ 的展示少于 $d_a$ 个, $\beta_a$ 是这些展示总权重与 $d_a$ 的比。	$\frac{1}{2}$ -competitive
Exponential Weighting	$\beta_a$ 是分配给 $a$ 的前 $d_a$ 个高权重展示的权重的指数加权。即: 设 $\mu_1 \leq \mu_2 \leq \dots \leq \mu_{d_a}$ , 则: $\beta_a = d_a \left\{ \left[ (1 + d_a^{-1})^{d_a} \right] \sum_{j=1}^{d_a} \mu_j (1 + d_a^{-1})^{j-1} \right\}$	$(1 - \frac{1}{e})$ -competitive

虽然在实际的工程系统中, 不可能不利用历史流量数据, 按照全盲的方式来进行在线分配, 然而, 上面的研究, 仍然对我们把握在线分配问题的合理策略有很大的帮助。这一点, 大家可以通过细心体味 Uniform Weighting 和 Exponential Weighting 这两种策略的区别来把握。

<sup>1</sup> Free Disposal 指的是给某个合约投送的量超过合约要求是无收益也无损失的, 这一点符合大多数广告合约的实际情况。

## 2.4 预定问题的在线优化算法

### 2.4.1 基于HWM的增量算法

### 2.4.2 基于Shale的增量算法

## 2.5 流量预测

### 2.5.1 基于时间序列的流量预测

### 2.5.2 基于bitmap的流量预测

### 2.5.3 基于检索的流量预测

### 2.5.4 频次控制下流量预测的修正

本章讨论的在线分配实用算法对流量预测功能有明确的需求。在后文讨论的广告网络中，一般来说也需要对广告主的定向条件和出价估计能够得到的流量，以辅助广告主进行决策。因此，这是一项在在线广告系统中非常常用的技术。流量预测是一种对广告市场宏观上量的把握，与本章中讨论的技术主题比较一致，因此我们放在这里介绍。

流量预测的问题可以这样描述，给定某广告的一组受众标签和其他条件，以及一个eCPM的阈值，估算在将来某个时间段内能够得到的在该eCPM阈值以下的流量。其中eCPM阈值主要是用于竞价广告系统中，目的是了解在一个某一个出价水平下能够得到的流量。对于合约式广告来说，这个阈值是不需要的，或者为了工程上一致起见，将该阈值设置为一个很大的数。

对于流量预测问题，基本的思路，是尽可能找到相对可比较的历史流量，根据历史数据来预测未来的流量。在具体预测中，可以按照图2.2.1中的方法，将历史数据切分成独立的一些供给节点，也可以进一步挖掘这些供给节点间的联系来得到更加平滑稳健的预测，例如引入时间序列分析的方法，在时间轴上建立联系。

我们仅仅讨论将个供给节点之间视为独立的方法，而关于时间序列分析的方法，请读者参考有关文献。对于按供给节点进行流量预测这一问题，比较有趣的思路，是将其也视为一个检索问题，对比与广告检索问题，这个问题里索引的对象由 $a$ 变成了 $(u, c)$ 聚合成的供给节点，而查询由 $(u, c)$ 上的标签变成了广告要求的标签。对比于第五章中将要详细讨论的广告检索问题，这个问题要简单一些，因为 $(u, c)$ 供给节点不存在DNF的布尔表达式描述，而是简单的特征集合而已。有一点需要说明，在实际的定向广告中，广告主可能对 $u$ 和 $c$ 上的标签设置联合条件，这实际上要求在流量预测中对 $(u, c)$ 对上来划分供给节点建索引，而不是对 $u$ 和 $c$ 分别建立索引，不过这样有可能使得索引的量大大增加，当这样做不实际时，可能还是需要对 $u$ 和 $c$ 分别建立索引。我们需要根据实际情况，来决定采用哪种方案，不过这两种方案的原理是没有区别的。为了简洁起见，在下面的部分中我们只对 $c$ 建立索引这个问题做探讨。

对 $c$ 来预测给定出价下上下文定向的流量，大致分为四个步骤：一、生成供给节点；二、建立索引：将 $c$ 上的标签作为文档的terms，建立倒排索引。这里的倒排索引与一般搜索引擎

相同，但在每条 $c$ 的正排索引部分，需要保存一项信息，即该页面 $c$ 对应的赢得竞价的eCPM分布直方图。三、查询结果：对一条输入的广告，按照其DNF描述的标签之间的与或关系构造查询，得到所有符合条件的 $c$ 的集合。四、估算流量：对每个候选 $c$ ，首先计算其与该广告 $a$ 的eCPM即 $r(a, c) = \mu(a, c)b_a$ (具体计算方法见下一节)，然后根据 $c$ 的eCPM分布直方图计算 $a$ 能赢得的流量，这样遍历所有候选，就可以估算出 $a$ 在出价 $b_a$ 情形下的流量。我们用下面的两段伪代码分别描述流量预测中建索引和检索的过程。

```

1  int forecast_build(InvIndex & supplyIndex) {
2      printf("我爱中文!\n");
3  }

1  int forecast(vector<string> & query, float eCPM,
2      InvIndex & supplyIndex) {
3      // 通过索引找到所有符合条件的供给节点
4      vector<int> supplies;
5      retrieve(query, index, supplies);
6
7      for (int s = 0; s < supplies; s++) {
8          curECPM = calcu_eCPM();
9      }
10 }
```

这样的方案预期是说流量预测，不如说是流量回顾，不过如果索引中的 $(u, c)$ 能够覆盖若干周的周期，其结果还是基本可靠的。

## 2.6 频次控制

关于频次的影响，Herbert E. Krugman博士在1972年提出了著名的三打理论：第一次刺激消费者试着了解信息，去问“这个广告是什么”；第二次刺激消费者去评量，去问“广告内容是什么”、“我曾经看过这个广告吗”；第三次接触时，回忆并开始逃离广告。三次，足以对消费者产生作用。这个理论原理上合理，但是主要适用于传统广告，并且是假设用户已经顺利通过了关注阶段。对于互联网广告，技术手段能够记录到的展示离这种意义下的有效展示有相当大的距离，因此无法直接套用三打理论。不过，一般来说，随着某个用户看到同一个创意频次的逐渐上升，点击率呈逐渐下降的趋势这一点是可以被验证的。因此，在按照CPM采买流量时，广告主有时会要求根据频次控制某个用户接触到某创意的次数，以达到提高性价比的目的。

从计算的角度来看，频次的作用是使得公式3.1中的可分性假设不成立的最主要影响因素。而将频次作为一个可控制的定向条件引入广告系统后，这个问题虽不能被彻底解决，却是大大地缓解了。

频次控制有客户端和服务端两种解决方案。客户端的方案，就是把某个用户对某个广告创意的频次值记录在浏览器cookie里，投放决策是再把这个值传给server来决策创意。这一方案好处是简单易行，可且服务成本低。缺点是扩展性不够好，当同时跟踪多个广告的频次时，cookie可能会变得很重，从而影响响应时间。服务器端的方案，是在后台设置一个专门用于频次记录和更新的cache，当广告请求到来时，在cache中查询候选广告的频次，并最最后实际投放的广告更新频次。

频次的cache，同时存在高并发读和高并发写的要求。而且随着频次控制粒度要求的不同，需要记录的频次数目也变化很大。比如在创意级别控制频次，就比在广告主级别控制频次需要更多的cache。不过好在频次cache的规模有一个上界，即一段时间内的展示总数，因此是远远小于所有可能的(cookie, 创意)组合数目的，这使得cache的实际规模没有我们想象的那么大。另一点值得注意的是，当我们用(cookie, 创意)的组合生成对应的哈希值，用作cache中的key时，实际上不太需要处理哈希冲突问题，因为对极少的比例的冲突组合频次控制不准是可以忽略的。频次控制上述这些特点，以及高并发读写的要求，大多数通用型的No-SQL存储方案并不能很好地用于频次控制的cache 服务，因此我们很可能需要自行实现一个非常轻量级的内存(key, value)方案来满足需求。



## 受众定向核心技术

对于在线媒体上的显示广告和文字链广告，要提高其回报，受众定向是其最重要的核心技术。从计算广告的核心，即优化一组流量上的ROI这一问题的角度来看，受众定向技术即是对广告( $a$ )、用户( $u$ )、上下文( $c$ )这三个维度提取有意义的特征(这些特征也称为标签)的过程。我们说过，上下文标签可以认为是即时的用户兴趣，因此我们把他们统称为受众定向。受众定向虽然不见得是计算广告中最困难的技术，但是确实是在线广告、特别是显示广告最核心的驱动力。因此，我们将这一问题单独拿出来，用一章的篇幅做深入的探讨。

从第一章里介绍的广告有效性模型出发，可以对各种定向技术有原理性的把握。一般来说，对于某一种定向技术，我们需要同时关注其效果和量两方面的指标，同时提供覆盖率较高但精准程度有限的标签，和那些非常精准但量相对较小的标签，有利于市场形成竞争的环境，这也为后一章中讨论的竞价广告提供了基础。从技术框架的角度看，受众定向标签可以分成用户标签、上下文标签和广告主定制标签三种类型，其实现方案也有较大的不同。我们在本章中重点介绍前两种定向技术的做法，而广告主定制化标签属于需求方定义的标签，我们会在第七章中讲到DSP时再介绍。

上下文定向需要对广告所在的页面进行分析，然而这一分析过程与搜索引擎的爬虫有很大的不同。结合广告对上下文信息的需求特点，一般可以采用一种半在线的方式抓取和分析页面，我们也将会在本章中讨论这种方法的特点和优势。

行为定向是根据用户历史上的网络访问行为对用户打标签的过程。哪些网络行为会有价值，是我们在挖掘行为数据来源时需要考虑的问题。我们将列举一些业界公认的有价值的行为数据类型，并给出使用多种数据类型进行行为定向的基本框架。由于海量用户的原始网络行为的数据量一般来说特别巨大，如何设计非常高效的数据组织方式，以及其访问流程，对于行为定向的实用化是非常关键的，这方面工业界的实践和经验总结，也是我们在本章涉及的一项重点内容。行为定向一般采用reach-ctr曲线进行半定量的评价，我们也会介绍这一曲线的用途和解读。

可以注意到，无论是上下文定向，以及在此基础上的行为定向，都广泛使用到文本分类和主题挖掘的技术。而在广告业务中，我们往往要选择那些有监督的主题挖掘方法，将页面内容映射到预先定义好的标签体系上，而不是无监督地自动聚类产生标签。这是由于广告中的标签体系要

向广告主售卖，因此必须是可解释的。对于无监督和有监督这两种类型的主题模型挖掘算法，本章中也做了概要的介绍，供大家在实际应用中参考。

本章还有一部分内容，是讨论广告业务中数据的核心作用，以及相应的产品和商业模式。熟悉了前面的受众定向技术，我们会发现，受众定向的本质，是将用户在网络上的一些行为可以售卖的人群属性。这同时也揭示了精准广告业务的本质：将原材料，即用户行为数据，加工成标签，再将标签售卖给需要的广告主。而广告投放过程已经变成了交付这些标签的载体而已。既然数据加工本身如此重要，这足以成为互联网广告中相对独立的一项业务。于是，数据加工与交易的产品化和规模化，成为在线广告区别于传统广告的一项重要市场特点。在这样的环境下，数据管理平台(DMP)这样面向数据收集、加工和交易的产品也应运而生。有关DMP的功能和典型案例，我们在本章后半段也会讨论。

### 3.1 定向方法综述

随着在线广告技术和业务的发展，产生了各种各样的受众定向方法，这些方法的综合应用，使得广告的精准程度越来越高。在考察某种定向方法时，主要有两个方面的性能需要关注：一是定向的效果，即符合该定向方式的流量上高出平均eCPM的水平；二是定向的规模，即这部分流量占整体广告库存流量的比例。当然，效果好、覆盖率又高的定向方法使我们追求的目标，不过往往难以两全。因此，广告系统有必要同时提供多种定向方法的支持，以达到整体流量上质的最优化。

我们先来看一些市场上比较流行的定向方式。按照其有效性和在广告信息接受过程中起作用的阶段，对照第一章中的广告有效性模型，我们把这些定向方式按照非常粗略的定性评估表示在图3.1中：

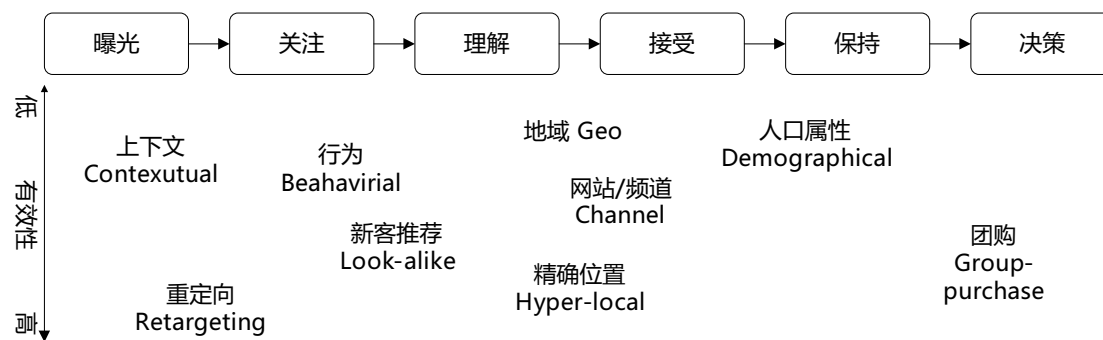


Fig. 3.1. 常见受众定向方法一览

在该图中，水平方向表示的是定向技术在广告信息接收过程中大致起作用的阶段，而垂直方向为大致的评价效果(越往下效果越好)。

总体上看,按照计算框架的不同,这些受众定向的技术可以分为三种类型:1. 用户标签,即可以表示成 $t(u)$ 形式的标签,或者说是以用户历史行为数据为依据,为用户打上的标签;2. 上下文标签,即可以表示成 $t(c)$ 形式的标签,或者说是根据用户当前的访问行为得到的即时标签;3. 定制化标签,即可以表示成 $t(a, u)$ 形式的标签,这这这是一种用户标签,不同之处在于是针对某一特定广告主而言的,因而必须根据广告主的某些属性或数据来加工。对受众定向的一些典型方法,我们举例说明如下:

一、地域定向(Geo-targeting)。这是一种很直觉、也很早就被广泛使用的定向方式。由于很多广告主的业务有区域特性,这种定向方式的作用相当重要,也是所有在线广告系统都必须支持的定向方式。地域定向也可以认为是一种上下文定向,不过由于其计算简单,仅仅需要简单的查表就可以完成,因而我们往往对其单列为一种定向方式。地域定向虽然一般来说效果有限,却是一种不可或缺的流量选择手段。举个例子,假设某电商网站只在北京运营和送货,那么其效果广告一般来说应该定向在北京的区域内,否则一个其他省的顾客点击广告进入购物环节后,如果发现无法结算,将会是非常差的用户体验。

二、人口属性定向(Demographical targeting)。人口属性定向虽然在普遍意义上效果并不算特别突出,但是由于这是线下传统广告比较常用的分析维度,因而在互联网品牌广告中比较早地被采用。人口属性的主要标签,包括年龄、性别、收入水平等。需要说明的是,除非有特别的专门数据来源,例如实名制SNS的注册信息或在线购物的消费记录等,一般情况下要进行准确的人口属性定向并不容易。在人口属性数据覆盖率不足的情况下,如果要按照这种定向进行CPM售卖,我们可以用已知人口属性的用户作为训练集,构造分类器对人口属性进行自动标注。一般来说,采用分类器的方法确定人口属性准确程度有限。在效果广告应用中,一般不需要这样做,因为预测出来的人口属性也是根据用户其他行为得到,因此对eCPM预测并不能提供额外的信息量。

三、频道定向(Channel targeting)。频道定向是完全按照供应方的内容分类体系,将库存按照频道作为划分依据,对各频道的流量投送不同的广告。这种定向方式比较适用于那些离转化需求比较近的垂直类媒体,比如汽车、母婴、购物导航等。对于内容覆盖面比较宽的媒体,这种方式很难取得的效果是有限的。举一个极端的例子,如果我们吧某网站的军事频道作为一个定向标签,那么很难找到直接匹配的广告需求。

四、上下文定向(Contextual targeting)。将频道定向这种方法加以推广,可以根据网页的具体内容来匹配相关的广告,这就是上下文定向。上下文定向的粒度,可以是关键词、主题,也可以是根据广告主需求确定的分类。上下文定向的效果在不同类别的内容上有很大的区别,但是这种方式有一个非常大的好处,那就是覆盖率比较高:对大多数广告展示,不论对当前访问用户的信息了解有多少,往往都可以根据当前浏览的页面推测用户的即时兴趣,从而推送相关广告。由于覆盖率高,上下文定向也是广告网络中首选的定向方法。

五、行为定向(Behavioral targeting)。行为定向是显示广告中非常重要的一种定向方式,其框架是根据用户的历史访问行为,了解用户兴趣,从而投送相关广告。行为定向之所以重要,是因为它提供了一种一般性的思路,使得我们在互联网上收集到的用户日志可以产生变现的价值。因此,行为定向的框架、算法和评价指标,也就奠定了在线广告数据驱动的本质特征,并催生了相关的数据加工和交易的衍生业务。如果我们把上下文定向看成是根据用户单次访问行为的定

向,那么行为定向可以认为是一系列上下文定向的融合结果。因此,上下文定向是行为定向的基础,而且对各种类型的上下文定向,都可以有相对应的行为定向方式。例如,地域定向是根据用户当前访问的IP来确定地理区域,相应地,我们也可以根据用户过去一段时间内的访问中最频繁的地理位置来定向,这种方式实际上得到的跟接近于用户的经常居住地,业界有人称其为“Where-on-earth”定向。

六、精确位置定向(Hyper-local Targeting)。在移动设备上投放广告时,我们有可能获得非常精准的地理位置。比如利用蜂窝信息或者GPS,地理定位的精度完全可以达到街区的粒度。这就使得基于精确地理位置的广告成为可能,也使得大量区域性非常强的小广告主,比如餐饮、美容等有机会投放精准定位的广告,这已经与传统意义上的地域定向有了质的变化,也成为移动广告最重要的机会之一。在桌面环境中,也有的数据提供商,例如Experian,可以提供根据IP信息做的电脑精确定位,在这样数据的支持下,桌面在线广告也可以进行精确位置定向。

七、重定向(Retargeting)。这是一种最简单的定制化标签,其原理是对某个广告主过去一段时间的访客投放广告以提升效果。显然,某个广告主的访客是其独有的信息,因此这属于定制化标签。重定向各种定向方式中被公认精准程度最高、效果最突出的,不过其人群覆盖量往往也较小。这是因为,重定向的覆盖投放量是由广告主固有用户的量和与媒体的重合比例共同决定的。关于重定向的原理和技术,我们将在第八章中具体介绍。

八、新客推荐(Look-alike)。由于重定向的量太小,而且无法满足广告主接触潜在用户的需求,因此不能仅仅依靠它来投送广告。Look-alike定向的思路,是根据广告主提供的种子访客信息,结合广告平台更丰富的数据,为广告主找到行为上相似的潜在客户。这一方法的目的,是希望在同等用户覆盖比率的情况下,达到比一些通用的兴趣标签更好的效果,是Look-alike所希望达到的,这也从实质上体现了广告主数据的核心价值。新客推荐只能说是一种大致的思路,而非具体的方法,其基本原理和做法,我们也将第八章中介绍。

九、团购(Group Purchase)。这并不是定向广告技术,却与其有一定的关联,因此我们在这里一并说明。根据我们的观点,团购也是一种变相的广告形式,这种广告有两个显著的特点:首先是一般都针对区域性的广告主,因此地域定向,或者直接按照地域分类组织,是必要的功能;另外,团购主要是利用价格工具,直接降低用户在决策阶段的门槛,使得价格敏感的用户转化效果有明显的提升,当然,这一手段也是要付出成本的。用类似的手段在创意上直接显示打折或降价的促销信息,也被其他的电商类广告广泛使用。

以上各种定向中,地域定向、频道定向和上下文定向属于 $t(c)$ 的定向方式;人口属性定向、行为定向属于 $t(u)$ 的定向方式;而重定向和Look-alike则是 $t(a,u)$ 的定向方式。各种定向的标签被应用于根据用户和环境信息选取广告候选的过程,因而对广告投送的结果会有比较显著的影响。 $t(c)$ 和 $t(u)$ 两种定向方式,一个根据的是当前页面信息,一个根据的是历史日志数据,因而在系统框架上有比较大的区别。下面我们将对这两种方式的典型代表,即上下文定向和行为定向的实现进行讨论。值得注意,定向技术并非按照上述的分类严格区分,或者一成不变的。各个广告网络或定向技术提供商,能够接触到的数据类型和规模都各不相同,基于这些数据本身进行深入挖掘,并找到对广告投放有意义的信号,才是受众定向在实用中的重点。因此,我们也力图避

免将注意力放在具体的定向方法，而是尽可能就用户日志信息挖掘、上下文信息挖掘这两个问题的通用方法做一些讨论。

## 3.2 上下文定向

我们先来看那些归类为 $t(c)$ 的受众定向方式。这样的定向中有一些根据广告请求中的参数信息经过简单运算就可以得到，比如频道/URL定向，操作系统定向等；另外一类则是根据上下文页面的一些特征标签，比如关键词、主题、分类等进行定向，我们重点讨论这样的上下文定向技术。

我们先抛开标签体系不谈，仅仅从打标签的方法上来看，上下文定向主要可以有如下的几种思路：1. 用规则将页面归类到一些频道或主题分类，这种方法相对简单；2. 提取页面中的关键词，这种方法是将搜索引擎的关键词匹配技术推广到媒体广告上时自然产生的，也是上下文定向的基本方法；3. 提取页面入链锚文本中的关键词，这种方法需要一个全网的爬虫作支持，因此已经超出了一般意义下广告系统的范畴，有兴趣的读者可以参考搜索引擎方面的有关文献；4. 提取页面流量来源中的搜索关键词，这种方法除了页面内容，也需要页面访问的历史记录作支持，从技术方案上来看更接近后面介绍的行为定向；5. 用主题模型将页面内容映射到语义空间的一组主题上，这样做的目的是为了泛化广告主的需求，提高市场的流动性，这一类的方法将在下文专门介绍。

在以上各种思路中，关键词提取是一项基础技术。上下文定向中的关键词提取，可以按照信息检索中的一般方法，即选取页面内容中TF-IDF较高的词作为关键词(见第二章中的具体介绍)；也可以采用需求方驱动的思路，从广告商相关描述中得到商业价值高的关键词表和IDF，再与页面内容中关键词的TF一起计算TF-IDF，来选取关键词。当能够得到比较丰富的广告信息时，比如运营AdSense那样的文本广告，或者可以拿到广告主SEM(Search Engine Marketing)词表时，后一种方法往往更加有效。

确定了对上下文页面打标签的方法以后，在在线广告投放时，页面标签系统需要对Ad Server查询的某一个URL快速返回其对应的标签。复杂的打标签计算是不可能马上完成的，不过在广告的问题中，某一次展示时标签的缺失并不是致命性的。根据广告的这一特点，我们可以用一种半在线的方式来实现页面抓取和打标签的逻辑。

### 3.2.1 半在线抓取系统

上下文页面的有关信息显然不可能在广告请求发生时实时分析得到，那么我们是否需要一个类似于搜索引擎蜘蛛的系统来做预先抓取呢？对于广告系统来说，其实是没有这个必要的。因为页面信息对搜索引擎而言，是服务的主体内容，而对广告系统而言，只是锦上添花的补充信息，我们完全可以设计一个更轻量级，效率更高的页面抓取服务。这种页面抓取服务的关键，是不作任何离线抓取，而在在线服务时产生了实际需求后才尽快抓取，我们把它叫做半在线(near-line)的抓取系统。

下图概念性地表示了半在线抓取系统的工作原理。如图所示，系统用一个cache服务来保存每个URL对应的标签，当在线的广告请求到来时：一、如果该请求的上下文URL在cache中存在，那么直接返回其对应的标签；二、如果该URL在服务中不存在，为了广告请求能及时得到处理，我们就马上返回空的标签集合，同时马上往后台的抓取队列中加入此URL，这样在较短的一段时间(通常为秒至分钟量级)之后，该URL就被抓取下来，并打上标签存入cache中。考虑到页面内容可能会不定期更新，我们可以设置cache合适的TTL(Time To Live)以做到自动更新标签。半在线抓取系统的示意图3.2.1。

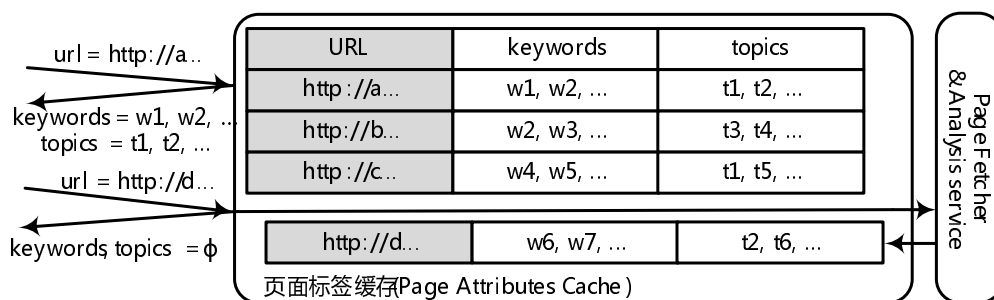


Fig. 3.2. 上下文定向半在线抓取系统示意

这样的方案，有以下两点好处：首先是在线cache的使用效率非常高，仅仅那些最近有广告请求的发生的URL才会被抓取，这样我们不需要耗费大量的爬虫资源去抓取可能根本用不到的页面。其次，因为我们只抓取需要的页面，并且可以在该页面第一次广告请求后很快得到页面标签，页面的信息覆盖率也很高。

### 3.3 行为定向

再来看那些归类为 $t(u)$ 的受众定向方式。这样的定向可以分为两类：一类也是根据广告请求中的某些信息经过简单运算就可以得到，比如地域定向；另外一类则需要大量的历史数据挖掘才能得到，我们重点讨论这一类定向技术的实现方案。这类定向问题包括从用户网上浏览记录加工得到的兴趣定向，以及根据用户历史所在的地域得到的用户主要居住地的“where on earth”定向等。由于这种定向都是根据用户的历史行为进行挖掘的问题，我们把它们统一称为行为定向(behaviorial targeting)。

行为定向是精准广告业务中对数据利用和变现最重要的问题，这一问题可以描述为，根据某用户一段时期内的各种网络行为，将该用户映射到某个定向标签上。有关对行为定向的用到的标签体系、数据来源、建模方法、特征生成和评测指标等问题，我们将在下面分别讨论。

### 3.3.1 受众定向标签体系

行为定向的目的，是把用户的线上或线下行为映射到对广告主有意义的标签体系上。行为定向的标签体系有两种组织方式：一种是按照某个分类法(Taxonomy)制定一个层次标签体系，其中上层的标签是下一层的父节点，在人群覆盖上是包含关系。一些面向品牌广告的受众定向，往往采用这种结构化较强的标签体系，比如Yahoo!的GD广告体系中的行为定向体系，其标签分为多个层次，其中前两层的一部分标签如下所示：

**Finance** - Bank Accounts, Credit Cards, Investment, Insurance, Loans, Real Estate, ...

**Service** - Local, Wireless, Gas & Electric, ...

**Travel** - Europe, Americas, Air, Lodging, Rail, ...

**Tech** - Hardware, Software, Consumer, Mobile, ...

**Entertainment** - Games, Movies, Television, Gambling, ...

**Autos** - Econ/Mid/Luxury, Salon/Coupe/SUV, ...

**FMCG** - Personal care, ...

**Retail** - Apparel, Gifts, Home, ...

...

**Other** - Health, Parenting, Moving, ...

需要指出，这一体系中的标签，是根据需求方的逻辑而制定，某些在媒体方意义很大的分类标签，比如军事等，由于没有明确的需求对应，不宜直接出现在标签体系中。对这样的媒体上的用户，应该用受众定向的方法根据其用户的细分特征映射到上面需求方的标签体系中。

另外一种标签的组织方式，是根据广告主某类特点的定向需求设置相应的标签，所有的标签并不能为同一个分类体系中所描述，也不存在明确的父子关系。这种半结构化的标签体系，往往包含一些比较精准的标签的集合，因而主要适用于多种目标、特别是效果目标并存的广告主的精准流量选择要求。关于这种标签体系，比较典型的是Bluekai采用的标签体系，我们会在后文中再做介绍。

## 3.4 行为定向数据来源

哪些行为会对定向有所贡献呢？互联网广告业界经过探索，对此问题有一些共同的认识。一般来说，有九种行为是确定对行为定向的建模有意义的。在评价某种行为的作用时，主要关注两个因素，一是质，就是上面所说的信息强度，二是量，就是该行为的频繁程度。我们按照这些行为的信息强度和性质，将这些行为分为四组排列如下：

一、决策行为：转化(Conversion)、预转化(Pre-conversion)。这些指的是在广告主的网站中发生的行为，往往对应着非常明确的用户兴趣。比如在电商网站上，转化就对应着最后的下单，而预转化对应下单前的搜索、浏览、比价、加入购物车等多种准备工作。这类行为的价值是最高的，但是也是供给方最难得到的。根据广告主端的数据来进行重定向或者个性化重定向是对此类行为最直接的利用，当然，在行为定向中，这类数据虽然量不大，但却不能忽视。

二、主动行为：搜索广告点击(Sponsored search click)、广告点击(Ad click)、搜索点击(Search click)、搜索(Search)。这一组行为都是用户在网络上在明确意图支配下主动产生的行为，因而也有比较丰富的信息量。其中的几种广告点击行为一般来说量不大，并不能作为定向的主要数据来源。而搜索行为是能够大量获得的最主要的主动行为，需要特别注意挖掘利用。

三、半主动行为：分享(Share)、网页浏览(Page View)。这两类行为都是用户在目的比较弱的网上冲浪过程中产生的。因此，其所设计的兴趣领域对把握用户信息有价值，但是非常细节的内容则精准程度有限。半主动行为的指导意义虽然有限，但是其数据量却是各种行为中最大的。\*\*\*\*\*

四、被动行为：广告浏览(Ad view)。广告浏览严格来说不能算作定向的行为依据，但是由于其频次与相应类别的广告点击负相关，因而在行为定向的建模中也可以使用。

关于以上各类行为数据对广告效果的意义，有两条基本的规律：1. 随着用户主动意图的提升，相应的行为数据信息价值也随之增大；2. 越接近转化的行为，对效果广告的精准指导作用越强。把握这两条规律，有助于大家判断各种各样的行为数据的价值。从后面关于DSP定向技术的介绍中，大家也可以对这些规律有更清晰的认识。不过需要提醒读者的是，不要忘记广告的根本目的是“低成本地接触潜在用户”。从这一点上判断行为数据的作用，会发现更靠近转化的行为更精准，实际上是因为这部分人群已经更加接近于决策的最终阶段，也就是说越发不是“潜在用户”。因此，在行为定向这个问题上，不能单纯追求ROI或者转化效果，而是要根据广告主的具体的人群接触目标来平衡效果和覆盖率。

### 3.4.1 行为定向建模问题

从目标的角度来看，行为定向问题的目标是找出在某个类型的广告上eCPM相对较高的人群。如果进一步假设在该类型的广告上点击价值近似一致，那么问题就转化为找出在该类型广告上点击率较高的人群。虽然我们说过，对品牌广告而言，点击率未必总是合理的评价指标，然而当我们认为在该类型上的各种广告目的均衡存在时，点击率仍然具有相对的衡量意义。因此，我们可以把某个用户在某类广告上的点击数目作为建模的对象。

由于点击行为是离散到达的随机变量，对其数目最自然的概率描述是泊松(Poisson)分布。泊松分布的形式如下：

$$p(h) = \frac{\lambda^h \exp(-\lambda)}{h!} \quad (3.1)$$

其中 $h$ 为某个用户在某个定向类别广告上的点击数目<sup>1</sup>，而 $\lambda$ 即位控制点击行为到达频繁性的参数，我们的模型应该做的就是将用户的行为与此参数联系起来。如果我们利用线性模型联系用户行为和 $\lambda$ ，则有：

$$\lambda = \sum_i w_i x_i(b) \quad (3.2)$$

<sup>1</sup> 在文[12]中，展示数目也被用一个用历史行为决定参数的泊松分布来建模，我们认为，广告并非用户主动行为，因此不宜用历史行为来预测。



这里的 $w$ 即为受众定向模型的需要优化的参数矢量。这里我们将原始行为 $b$ 先经过一个与标签 $l$ 相关的特征选择函数，以得到与此类别相关的evidence，再将此evidence作为特征用在上述模型中。将式3.2代入式3.1，就得到行为定向的整体模型。

上面是一种非常典型的工程建模思路：当我们面对一个多自变量的回归问题时，可以现根据目标值的特性选择合适的指数族分布来描述，并用线性模型将多个自变量和指数族分布的参数联系起来。这样做，可以利用线性模型更新简单和可解释性强的特点，同时又对目标变量的类型有较强的适应性。这种建模方法，称为广义线性模型(Generalized Linear Model, GLM)。当目标是Binomial变量时，将线性函数输出用Logistic sigmoid 函数扭曲到(0,1)即可。当目标是 $R^+$ 上的变量，或者Multinomial变量时，又该如何处理呢？大家可以从GLM的理论中得到答案，不过这离本书太远，就不做赘述了。有关广义线性模型的一般性讨论，请大家参看[]。

公式3.1的行为定向模型，有几点需要注意：首先， $w$ 可以是与标签 $l$ 相关的，即对不同的定向标签训练不同的线性函数。这样做的优点是可以更准确地对每个类别进行建模，但缺点是当有些类别数据不足时估计偏差较大。当 $w$ 与标签相关时，原始行为也可以不经过特征选择函数，或者经过一个与标签无关的选择函数，这是因为类的本质特征已经反映在了对行为进行映射的参数矢量上。其次，就这种建模方法来说，主要适用于有明确需求方意义的标签体系。因此只有广告上有这些标签，才能根据其点击行为来建模。就模型的线性形式来说，在一些按供给方逻辑定义的标签体系上，也可以用类似的行为加权的形式来进行定向。然而，此时其权重参数的确定无法利用相应的广告点击作指导，不过仍然可以借用其他需求方标签体系的训练结果。

上面的行为定向模型特征生成的过程，由于样本量比较大，需要有一个比较高效的方法来简省计算量，我们在下一节中具体讨论。在得到了训练样本以后，需要更新3.1中的行为定向参数。如果行为定向的模型属于指数族分布的范畴，比如线性回归(Linear Regression)模型，那么只需要对数据遍历一次，就可以得到最大似然解。不过对于公式3.1中的模型形式，则需要多次访问数据迭代求解。由于上面介绍的特征生成过程是on-the-fly生成特征，我们需要给出在生成了特征后需要累积的统计量，以及在统计量收集完成后模型参数更新的公式。

根据[12]中的讨论，\*\*\*\*\*

### 3.4.2 行为定向特征生成

行为定向特征的生成过程有两点需要讨论，一是特征选择函数 $x_l$ 的确定；二是模型3.1训练集的组织 and 生成方式，这两方面都对计算性能有着巨大的影响。由于行为定向是在线广告计算问题中数据规模最大的，处理的高效性是我们工程中主要考虑的问题。

我们常用的特征选择函数，是将每一类用户行为先分别映射到标签体系上，再计算出各个行为在某个标签上累积的强度作为模型的特征输入。例如，对于页面浏览行为，我们向上下文定向中介绍的那样，将页面内容转换为标签，作为此次行为的标签；而对于搜索行为，我们可以根据查询词将其映射到某些标签上，在将某个用户在一段时间内在标签 $l$ 的查询词总数，作为 $x_l$ 里的一维。这里要注意的关键，是“一段时间内的行为”，因为过于久远的行为对于用户兴趣的贡献是很小的。如何将行为累计控制在一段时间以内，工程上有两种常用的方法，分别是滑动窗法和时间衰减法，我们用图3.4.2来示例：

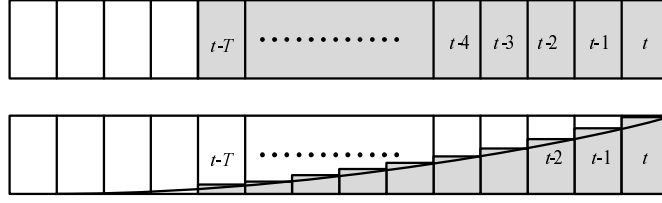


Fig. 3.3. 用户行为累计方法示意：上、滑动窗法；下、时间衰减法

在滑动窗法中，我们设定一个窗长 $T$ ，然后将从当前时间倒推再此窗长内所有属于 $l$ 的行为强度累加起来；而在时间衰减法中，我们并不明确设定窗长，而是转而设定一个衰减因子 $\alpha_l$ ，然后用累积到上一个时间片的特征 $x_l^t$ 与本时间片的行为强度 $b_l^t$ 递归地得到今天的特征 $x_l^{t+1}$ ，其递归更新公式为：

$$x_l^{t+1} = \alpha_l x_l^t + (1 - \alpha_l) b_l^t \quad (3.3)$$

这两种方式并无本质上的区别，其实际对原始行为过滤的窗型，前者为矩形，后者为指数形，并且形状都由唯一的参数来控制。但是从工程角度看，我们更推荐使用第二种方案，因此在这种方法中，只需要保存累积到前一个时间片的特征和当前时间片的行为强度，空间和时间复杂度都较低。

行为定向的训练过程，实际上就是调整各个标签类别上各种特征权重的过程。影响训练结果的有两个主要的因素：一是训练集的长度。由于上述的时间因素的影响，我们准备的训练集需要横跨若干天的范围，一般来说，为了消除星期这一显而易见的周期性影响，这一天数可以选择为7的整数倍。对于一个用户来说，他前一天的行为特征 $x_l(t)$ ，和当天的该标签广告点击次数 $h_l(t)$ ，对应于3.1的一个训练样本。因此，每个用户实际上会生成多个训练样本。二是时间片的大小。这实际上反映了我们对定向的时效性的要求：如果我们希望更快地利用行为信息对定向标签做出调整，必然要缩小这一时间片大小<sup>2</sup>。可以想见，训练集的样本数目，正比于训练集长度，且反比于滑动时间片长度。当用户数目较多、训练集长度较长、而滑动时间片又较短时，总的训练样本数目是很大的。

为了避免计算冗余，使得训练时的空间代价尽可能地小，在[12]中，作者给出了一个复杂度为 $O(1n)$ 的训练样本生成算法，该算法的关键点，是将同一个 $u$ 的所有行为都一次性加载到内存中，并按时间顺序排列成一个事件流。通过在此事件流上向前滑动，依次得到各个 $(u, t)$ 对应的训练样本。另外，在一个滑动时间片内的事件，已经预先按照我们需要的粒度聚合成一些计数，这些计数是生成特征的基础。如果我们采用上文中滑动窗法的特征选择函数，那么需要保留前 $T$ 个时间片的行为事件，并生成相应的训练样本。这一方法看起来非常普通，确实在大规模用户行为分析时必须要注意的，也是我们在计算广告架构(图3.1)中Session Log部分设计的原则，即一定要将所有用户日志预先组织成以用户ID为key的统一格式。我们将行为定向特征生成算法的示例性代码列在下面：

<sup>2</sup> 更快地利用用户行为反馈还有其他系统方面的需求，请参见第7章中“短时行为反馈”一节。

```

1  int bt_sample_gen(vector<vector<float> > & events, int T,
2                    vector<vector<float> > & features,
3                    vector<vector<float> > & targets) {
4      int numSlice = events.size();
5      int dim = events[0].size();
6
7      features.resize(numSlice);
8      features[0] = events[0];
9      for (int s = 1; s < numSlice; s++) {
10         features[s] = features[s - 1];
11         for (int d = 0; d < dim; d++) {
12             features[s][d] += events[s][d];
13             if (s - T >= 0)
14                 features[s][d] -= events[s - T][d];
15         }
16     }
17 }

```

如果采用衰减因子的特征选择函数，则程序将更加简单，并且可以改成串行处理方式，即每次输入一个时间片的行为事件，同时输出当前的特征。

采用这种方案，我们只需要将各个时间片的原始行为和target值保存在持久化存储中，而在训练过程中on-the-fly生成实际的训练样本。由于考虑到用户的上网行为从时间段上来说一般具有很强的局部性，这样的做法可以避免为了生成训练样本而带来的冗余存储，同时引入的额外计算量也很少。

### 3.4.3 行为定向计算过程

虽然公式3.1的行为定向模型看起来稍有点复杂，其核心目的无非是为了得到通过数据建模得到3.3中的行为系数 $w_l$ 。在实际行为定向的计算过程中，我们不需要考虑公式3.1的形式，只需要计算线性函数3.3的值，然后根据某个预先确定的阈值来决策某个用户是否应该被打上某个定向标签。

定向的计算过程比训练过程的数据准备要简单，因为不再需要准备目标值，只需要按照滑动窗法或者时间衰减法得到原始的行为特征 $x_i^t$ ，再根据 $w_i$ 加权求和的到得分 $\lambda$ 。由于这一评价函数也是线性，当特征累积采用时间衰减法时，得分也可以通过昨天的得分衰减后累积上今天得分的方式得到，即：

$$\begin{aligned}
 \lambda^{t+1} &= \sum_i w_i \alpha x_i^t + \sum_i w_i (1 - \alpha) b_i^t \\
 &= \alpha \lambda^t + (1 - \alpha) \sum_i w_i b_i^t
 \end{aligned} \tag{3.4}$$

这一公式揭示了工程中的受众定向系统实现的一个关键点，那就是在线上存储各用户的定向标签得分的cache中，在每个新的时间周期对得分进行乘以 $\alpha$ 进行衰减，再将上一个时间周期收集到的原始行为 $b_i^t$ 加权求和后累加上即可。这样的线上计算过程，更加体现了时间衰减法的优势，特别是当需要对用户的短时行为进行快速反馈时，这样简便的递归式计算方法非常有效。

### 3.4.4 行为定向的评测

对于上面讨论的行为定向模型，其效果并非是一个确定不变的值，这是因为我们可以通过调整线性函数输出 $\lambda$ 的阈值来控制某个标签人群的量，当然在量扩大的情况下一般来说精准性也会降低。对其他形式的行为定向模型，也都具有类似的特点。因此，行为定向模型的评测，需要考虑到量的影响。一般来说，我们可以通过Reach-CTR曲线来进行半定量的评测。在正常情况下，较小的人群规模应该较为精准，也即对该类型广告的CTR较高，而随着人群规模的扩大，该CTR也会逐渐走低。我们把标签接触到的人群规模称为reach，而这一reach和CTR构成的曲线，是评价该标签上的定向是否合理，以及效果如何的重要依据。

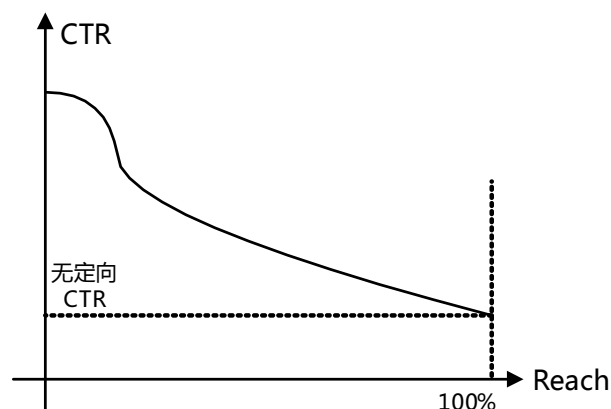


Fig. 3.4. Reach-CTR曲线示意

我们在图3.4.4中给出了reach-CTR曲线的一个示例，我们来了解一下读这个曲线的几个关键点。首先，该曲线应该大体上呈下降的趋势，如果数据质量或定向建模有一些问题，有时会出现非下降的趋势，或者头部较低的情况，这意味着调低用户规模反而使得点击率下降，这显然是不正常的。如果出现这种情形，则需要特别引起注意，认真检查定向流程，或者判断是否已有的数据无法支持该定向标签。其次，reach-CTR曲线最右端一个点的CTR水平是固定的，既无法通过改善数据和模型来提高的，因此这是reach达到100%，也即全部用户的情形下的CTR水平。而该曲线的斜率越大，往往表示定向模型的鉴别力越强。由于实际中一般会将阈值设定得较高，从而达到较好的定向效果，因此我们往往只需要关注该曲线头部的部分即可。

工程中需要注意的是，生成reach-ctr曲线的过程，需要仅仅访问一遍数据就能够完成。因此，我们在前面受众定向的过程中，需要保留的是每个用户在各个标签上的得分值，而不是最后二员的判断结果。给定一批测试用户在所有标签上的定向得分值，生成reach-ctr曲线的过程如下面的代码所示：

```

1 struct Imp {
2     int click;
3     int userId;
4 };
5
6 int reach_ctr(vector<Imp> & imps) {
7     vector<float> threshs;
8     vector<float>
9
10    for (int i = 0; i < imps.size(); i++) {
11
12    }
13 }
```

### 3.5 文本主题挖掘

从上面两种定向的讨论可以知道，对页面的分析和标签化是一项基础工作。页面打标签的粒度，可以精细到关键词，也可以粗略到页面的类型。除了这两种极端情况，我们也可以考虑将页面内容直接映射到一组有概括性意义的主题上，比如将一个讲编程语言的博客页面映射到“IT技术”这样的主题上。如果把页面视为一个文档，这就对应于主题模型(Topic model)的研究问题。总体上看，主题模型有两大类别：一种是预先定义好主题的集合，用监督学习的方法将文档映射到这一集合的元素上；一种是不预先定义主题集合，而是仅仅控制主题的总个数或聚类程度，用非监督学习的方法自动学习出主题集合，以及文档到这些主题的映射函数。广告中的主题挖掘有两种用途：如果仅仅用于广告效果优化的特征提取，那么监督或非监督的方法都可以；如果是用于建设对广告主售卖的标签体系，那么应该优先考虑采用监督学习的方法，因为这样可以预先定义好对广告主有意义且可解释的标签体系，对后续售卖会有很大帮助。

#### 3.5.1 无监督主题模型

为了理清主题模型发展的技术脉络，我们先从非监督方法介绍起。先来看一下问题的描述：假设我们有一个由 $M$ 个词组成的词表，以及一组文档 $\{d_1, d_2, \dots, d_N\}$ ，采用“bag of words”表示，文档 $d_n = \{x_{n,1}, x_{n,2}, \dots, x_{n,M}\}$  ( $1 \leq n \leq N$ )的形式，其中 $x_{n,m}$ 为词表中第 $m$ 个词在 $D_n$ 中对应的词频或TFIDF值。显然，一般情况下，矩阵 $X = \{x_{nm}\}_{N \times M}$ 是非常稀疏的。假设这一文档集合主

题模型对应着 $\{1, 2, \dots, T\}$ 这一组主题，我们的目的就是每个文档得到其在这些主题上的强度 $T_n = \{z_{n,1}, z_{n,2}, \dots, z_{n,T}\} (1 \leq n \leq N)$ 。

### LSA模型

这一问题最初的解决思路，是对上面文档和词组成的矩阵进行奇异值分解(Singular Value Decomposition, SVD)，找到这一矩阵的主要模式，这一方法称为潜在语义分析(Latent Semantic Analysis, LSA)。LSA的方法可以表示如下：

$$\{X\}_{N \times M} = (\alpha_1 \cdots \alpha_K) \cdot \text{diag}(s_1, \dots, s_K) \cdot (\beta_1 \cdots \beta_K)^\top \quad (3.5)$$

其中 $K$ 为矩阵 $X$ 的秩， $s_1 \geq s_2 \geq \dots \geq s_K$ 为 $X$ 的 $K$ 个奇异值。左侧的矩阵就是将潜在语义空间中的主题映射到某个文档的变换矩阵，而右侧的矩阵则是主题映射到某个文档词表中某个词的变换矩阵。我们最多可以得到的潜在语义主题数目，等于矩阵 $X$ 的秩 $K$ ，不过一般情况下，我们都会选择一个远小于 $K$ 的主题数目用来描述整个语义空间。当我们选择的主题数目为 $K$ 时，实际上是用下式对 $X$ 进行了近似：

$$\{X\}_{N \times M} \approx (\alpha_1 \cdots \alpha_T) \cdot \text{diag}(s_1, \dots, s_T) \cdot (\beta_1 \cdots \beta_T)^\top \quad (3.6)$$

这等价于我们令所有的 $s_t (T < t \leq K)$ 都等于0，换句话说，通过这种方式去掉了大多数非主要因素的影响，从而得到了整个语义空间比较平滑的描述，相应得到的文档主题，比起关键词特征来，也就具有更好的泛化能力。

根据奇异值的特性，我们知道所有的奇异值都是非负的，但是LSA得到的两个变换矩阵，不能保证每个元素都为非负值。这一点对应的直觉意义是：如果一篇有某个主题的话，可能该文档中出现某些词的频次的期望值为负。这一点，直观上并不十分容易理解，是LSA模型与后面几种概率文档主题模型不太一样的地方。

### PLSI模型和GaP模型

LSA方法的物理意义清楚，也有成熟的数学工具可以利用，因而在信息检索中得到了比较早的应用。类似的思想也可以用概率建模的方式来表达。概率的方法，是通过文档生成的过程进行建模，来进行主题分析。这一生成过程可以表达为：

1. 每个文档 $d$ 对应一个topic的Multinomial分布 $p(z|d)$ ，据此生成一个主题 $z_i$ ；
2. 给定主题，对应一个词的Multinomial分布 $p(w_n|z_n, \beta)$ ，据此生成一个词 $w_i$ ；

其中的参数 $\beta = (\theta_1, \dots, \theta_K)^\top$ ，而 $\beta_k$ 即为 $P(w|z_k)$ 对应的Multinomial分布的参数。这一模型称为PLSI(Probabilistic Latent Semantic Indexing)[24]。对应于上面的生成过程，矩阵 $Z$ 的生成似然值可以表达为：

$$\begin{aligned} \log P(Z) &= \prod_n P(d_n) \prod_m \sum_z P(z|d_n) P(w_m|z) x_{nm} \\ &= \prod_n \prod_m \sum_z P(z) P(d_n|z) P(w_m|z) x_{nm} \end{aligned} \quad (3.7)$$

PLSI从字面上看,是概率化了的LSA模型,我们可以将 $P(d_n|z)$ 和 $P(w_m|z)$ 分别对应于LSA中的两个变换矩阵。不过这两个模型的物理意义也有所不同:在PLSI中,由Multinomial分布的定义我们知道,变换矩阵即两个条件分布的元素都大于0,也就是说给定一个主体的情况下,某个词的词频的期望值不可能为负,这一点与直觉更为吻合,也是此概率框架合理性的表现。当然,实用中的效果根据数据集特性的不同,PLSI和LSA各有长处。

概率模型的另一个好处,是我们可以直接套用概率模型典型更新方法,并很容易地实现分布式求解。很容易看出,PLSI模型实际上是前文介绍的指数族混合分布的特例,其中的component为Multinomial分布。因此,可以直接套用EM算法,以及其对应的map/reduce或MPI迭代解法来解此问题。而前面的LSA模型用到的SVD分解,则需要一定的技巧才能变成分布式实现的版本,并且远没有EM算法的概念清晰和实现简单。因此,在实际的海量数据上的文档主题模型,PLSI比LSA有一定的实用优势。

PLSI模型用Multinomial分布来描述主题的分布和每个主题中的词分布,而Multinomial分布的特点是只考虑各个变量的相对比例,而不考虑其绝对数值。如果我们采用Gamma-Poisson过程来建模,即假设每个主题生成的概率用独立的Gamma分布来描述,而每个主题中某个词的产生服从Poisson分布,这就对应了Gamma-Poisson(GaP)模型[10]。GaP模型与PLSI模型,如果从变量的依赖关系上看,是非常相似的,只不过两者具体条件分布的形式是不同的,而这种不同也是在通过在指数族中选取不同的分布而得到的。认真比较这两种文本主题模型的联系和区别,对于理解图模型表达的变量依赖关系和分布的具体形式在建模中的作用,有很大的帮助。GaP与PLSI相比,由于没有将每个文档中各个主题变量的强度进行归一化<sup>3</sup>,因而对内容相似的长文本和短文本的概率描述是不同的,而后续的Poisson词产生概率也更加适合离散到达时事件的描述,因而在文本主题建模上有一定的合理性。不过,GaP模型的EM最大似然解法不像PLSI那样有简单的闭式更新公式。在[10]中,作者也是采用一种近似的方法来优化,因此,这一模型在工程中的实用性受到了一定的限制。

## LDA模型

第二章讨论的贝叶斯方法,也可以应用于PLSI模型。这样做的动机,是在当文档信息不足或者噪声较大时,能够利用贝叶斯的框架对结果做有效的平滑。在贝叶斯框架下,我们视为上述PLSI模型的参数 $\theta$ 为随机变量。对于某一篇文章,其生成过程可以描述为先验分布:

1. 根据分布选择文档长度 $N$ ;
2. 根据 $w$ 的先验分布 $\text{Dir}(\alpha)$ 生成 $w$ ;
3. 对每个文档中的词 $n \in \{1, \dots, N\}$ : (a) 根据 $\text{Multi}(w)$ 分布选择一个主题 $z_n$ ; (b) 2. 给定主题,对应一个词的Multinomial分布 $p(w_n|z_n, \beta)$ ,据此生成一个词 $w_n$ ;

把这一生成过程与PLSI相对比可以知道,这相当于PLSI的贝叶斯版本,即给topic的分布 $w$ 加上了先验分布,而先验分布采用的是共轭形式即Dirichlet分布。其对应的图模型见3.5。从第二章中的介绍可知,我们可以采用经验贝叶斯的方案来确定这两个先验分布,这样的模型就是LDA。

<sup>3</sup> 多维Gamma分布随机变量在归一化以后服从Dirichlet分布,也就是Multinomial分布的共轭先验形式。

由PLSI模型到LDA模型,对文档生成过程的描述更为清晰,而根据贝叶斯学习的一般目的我们可知,LDA模型在数据噪声较大,或者每个文档内容较少时可以达到比较稳健估计的效果。

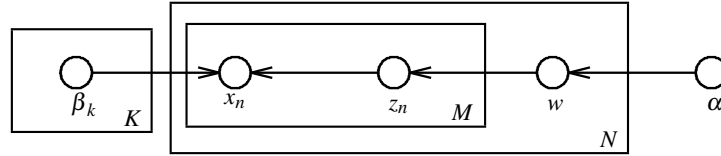


Fig. 3.5. LDA概率图模型表示

如果采用经验贝叶斯的方法来确定超参数 $\alpha$ ,那么此时原来的参数 $w$ 就变成了隐变量,优化的参数除了 $\alpha$ ,还包括 $\beta$ ,优化的目标函数可以写成:

$$p(w|\alpha, \beta) = \int p(w|\alpha) \left( \prod_{n=1}^N \sum_{z_n} p(z_n|w) p(w_n|z_n, \beta) \right) \quad (3.8)$$

由于PLSI模型不是指数族分布,因而其对应经验贝叶斯模型的解不能通过\*的EM方法得到,而是需要采用变分法近似求解。在[6]中,对这一模型的变分解法进行了详细的介绍,大家可以参照。不过在实际的工程实践中,LDA模型更为常用的更新方法是吉布斯采样(Gibbs Sampling)法。

\*\*\*\*\*

### 3.5.2 有监督主题模型

由于售卖的需求,广告中的受众定向标签体系有时是预先定义好的,有监督的主题模型对于广告问题来说更加适用。根据前文的讨论,它可以是一组非结构化的标签集合,也可以是一个结构化的层次标签体系。因此,我们有两种思路来解决此问题:一是采用Multi-label Classification的方法,二是沿用上面的主题模型方法,将其变成有监督主题模型。关于有监督主体模型,以上述的LDA为出发点,研究者也提出了若干种相关的方法。

结合广告定向的情景,有两种有监督主题模型可以关注:一、sLDA(Supervised LDA),这是在某种标签监督下进行主体挖掘的通用模型,适用于标签为各种分布的情形。当标签为离散值时,就对应于根据某种分类进行主题挖掘。二、HSLDA(Hierarchically Supervised LDA)[40]。在此模型中,标注的类型是一个Hierarchy上的层次标签,这非常切合于广告中的需求。

以HSLDA为例, \*\*\*\*\*

值得注意的是,在文档主题挖掘领域,还有一类方法也与Hierarchy有关,比如HLDA或HDP,但是其问题定义是在某个数据集上学习得到一个Hierarchy结构,而不是根据一个给定的Hierarchy上的标准挖掘潜在主题。因此,这类工作不属于有监督主题模型的范畴。



## 竞价广告核心技术

### 4.1 位置拍卖市场

竞价广告是象拍卖那样销售广告。也就是根据广告主的出价，以及由此计算出的期望收益，决定谁可以得到某次展示的广告位。在竞价广告产生的开始阶段，出价是广告主阶段性调整的，而到了广告交易市场阶段，广告主可以对每次展示实时调整出价<sup>1</sup>，这两种方式从拍卖市场的宏观角度看，没有本质差别。

让我们先来看看怎么具体描述竞价广告问题，并从宏观市场的角度了解一些非常重要的结论。假设有一组广告位可以被占用，将这些广告位按照其经验价值排名，分别记为 $s = 1, 2, \dots, S$ ，对Banner广告而言，这里的 $S$ 一般为1。在某次广告请求中，有一组广告 $a = 1, 2, \dots, A$ 出价参与拍卖，每个广告的出价记为 $b_a$ ，系统将前 $S$ 个高出家的广告依次放到前面排序好的 $S$ 个广告位上，这样的问题称为位置拍卖(Position Auction)。根据前文的讨论，当某个广告 $a$ 被放在 $s$ 位置上时，其期望收益即eCPM为 $r_{as} = \mu_s v_a$ 。这里我们做了一些假设，比如点击率 $\mu$ 仅与位置 $s$ 有关，而点击价值 $v$ 仅与广告 $a$ 有关，这些假设在搜索广告给定某具体关键词的情形下，可以说基本合理，对于显示广告的情形虽然非常粗略，但并不太影响对竞价问题宏观市场的讨论。

围绕位置拍卖最重要的研究，是广告的定价(Pricing)问题。定价问题探讨的是在一次具体的拍卖交易中，给定各参与者的出价，以及他们的期望收益，如何对最后获得某个位置的广告商收取合适的费用。乍听起来，讨论这个问题有点多余，按照广告主自己的出价收取不就可以了么？为了阐释清楚此项研究的动机，我们先来看看下面的例子：假设有某个单位位置( $S = 1$ )的广告机会在竞拍，开始有两个广告主参与，甲出价1元，乙出价2元，当然甲赢得了此次竞价，如果按照其出价来收费，市场就向他收取2元的费用。在广告市场里，这一拍卖机会还会重复出现(对应于不断产生的展示)，因此广告主可以也存在调整出价的机会，假设乙在发现自己2元钱能拿到流量以后，自然就会想到，是不是可以调低出价，用更低的成本拿到流量？乙将一直不断尝试，知道把出价调低到1.01元，发现继续调低就拿不到位置了。于是系统稳定在甲出价1元，乙

<sup>1</sup> 实时竞价的具体产品和技术将在第八章中介绍。

出价1.01元。此时假设又有一个广告主丙加入竞争,并希望赢下此广告位,那么以此类推,他在不断调整后将会出价1.02元,市场的收入也就是1.02元。我们有可能通过调整定价策略来影像系统的总收益么?这样的机会显然存在。比如我们在甲出2元,乙出1元参与竞价时,并不对获胜的甲收2元,而是收取其下一名即乙的出价1元,那么甲就没有动力调低其出价了。那么当丙加入时,就需要出价2元以上才可以赢得竞价,市场的收入也就变成了2元(不论丙出价多少,我们都按其下一位即甲的出价来收费)。这个简单的例子告诉我们,在广告这样的参与者可以针对同一个标的物不断调整出价的拍卖环境中,通过聪明的定价策略,完全可能为整个市场创造更高的收益以及其他好处。

在定价问题上,我们在微观上的直觉未必可以推广到宏观市场。让我们来看看如何对上面的位置拍卖问题作进一步探讨。从整个市场的角度来看,我们重点需要研究的是市场处于稳定状态下的收益和其他情况。而所谓稳定,指的是整个竞价系统处于纳什均衡(Nash Equilibrium)状态,也即每个广告商都通过出价得到了最符合自己利益的位置。对某一次位置竞价来说,其对称纳什均衡(Symmetric Nash Equilibrium)状态可以表示为下式:

$$\begin{aligned}\mu_s(v_s - p_s) &\geq \mu_t(v_s - p_t), \forall t > s \\ \mu_s(v_s - p_s) &\geq \mu_t(v_s - p_{t-1}), \forall t < s\end{aligned}\quad (4.1)$$

注意这里的下标意义有所调整,这里的 $v_s$ 指的是排在 $s$ 位置上的广告上的点击价值,并非 $s$ 位置带来的点击价值;而 $p_s$ 指的是市场向排在 $s$ 位置上的广告收取的费用,即定价。这一均衡状态的意义很容易理解:对于最终位置排名竞价结果中的每一条广告,其收益都比排在其他位置上要高。显然,在这样的状态下,每个广告商都达到了自己最优的状态,整个系统也就随之稳定下来。

在公式4.1中,市场方能够调整的策略,只有 $p_s$ 的确定方式,也就是定价策略。随着定价策略的不同,市场达到稳定状态时的宏观收益情况和稳定的程度都有所不同。因此,有关竞价市场宏观性质的研究,主要目的是寻找更好的定价策略以优化整体收益。关于位置竞价问题纳什均衡状态的数学分析,由于与本书的产品和技术重点有一定差距,因此不再介绍这方面的内容,有兴趣的读者可以阅读文献[43]。我们只介绍两种竞价市场用到的定价策略,以及与宏观市场相关的几个策略问题。

#### 4.1.1 定价策略

在线广告竞价市场最常见的定价策略,是广义第二高价(Generalized Second Pricing, GSP)策略;另外有一种VCG(Vickrey - Clarke - Groves)定价策略,虽然理论上比GSP更好,但是由于原理较复杂,向广告主解释起来有难度,因此在实用系统中采用的并不多。下面我们分别介绍一下这两种定价策略。

##### 广义第二高价

先来看看什么叫第二高价(Second Pricing)。所谓第二高价,指的是在位置拍卖中,向赢得某个位置的广告商收取其下一位广告主的出价。如果是按照CPM结算,那么第二高价就可以直接应用

了。需要考虑的问题是，在CPC计算的竞价广告系统中，广告主的出价是针对点击的，而竞价是针对eCPM的，因此要对两者做一下换算以实现公平的第二高价，这就是所谓的广义第二高价，其定价公式为：

$$p_s = \mu_{s+1} b_{s+1} / \mu_s + \Delta \quad (4.2)$$

如果将等式两边同时乘以 $\mu_s$ ，可以看出广义第二高价实际上仍然是eCPM上的第二高价。读者也可以自行验证，在广义第二高价情形下，对某广告主的定价是一定不会大于其出价的。实际上，这种定价策略也同样适用于CPS结算的竞价市场，并且只需要将公式中的 $\mu$ 换成 $\mu v$ 即可。公式最后的 $\Delta$ ，一般为广告系统结算货币的最小单位，比如1美分，这是一种历史惯例，也在某种程度上让广告主心理上感觉更加公平。

对比上面提到的竞价广告的例子，可以直觉地知道，广义第二高价可以基本上满足市场稳定情形下收益优化的需求。不过需要说明的是，这种满足只是直觉上的而非理论意义下最优的，这一点读者可以对比下面介绍的VCG定价策略来理解。虽然并非理论上最优，广义第二高价却有着实现简单、容易向广告主解释等诸多操作中的优点，因此在实际的竞价广告系统中是最主流的定价策略。

## VCG定价

VCG定价是Vickrey, Clarke和Groves在研究竞价系统均衡状态时得到的一种理论上较为优越的定价策略。其基本思想是：对于赢得了某个位置的广告主，其所付出的成本应该等于他占据这个位置给其他市场参与者带来的价值损害。在这一原则下，VCG的定价策略可以表示为：

$$p_s = \sum_{t>s} (\mu_{t-1} - \mu_t) v_t \quad (4.3)$$

这种定价策略直觉上的合理性很容易理解。实际上，理论分析表明，VCG定价策略的优越性体现在如下几个方面：首先，在这种定价策略的稳定状态下，整个市场是truth-telling的。所谓truth-telling，可以理解为每个广告主都找到了自己的最优状态。其次，相对于其他的定价策略，这种定价向广告主收取的费用是最少的。

虽然有以上诸多的优点，VCG定价在在线竞价广告中却并不是一种主流的方式。究其原因，主要是由于这种定价方式的逻辑过于复杂，比较难以向广告主解释清楚；另外在广告主和媒体存在博弈关系的情形下，媒体是否正确地计算了“给其他市场参与者带来的价值损害”也很难验证。不过这种定价方法也有其市场空间，有的广告厂家，比如Facebook，在实际的系统中采用了这一定价方法。

### 4.1.2 市场保留价

为了控制广告的质量和保持一定的出售单价，竞价广告市场往往要设置一个能够赢得某个拍卖位置的最低价格，这一价格我们称之为市场保留价(Market Reserve Price, MRP)。广告主的出价只有

在高于市场保留价时，才能获得竞价机会，同时在赢得某个拍卖位置后，如果根据定价策略算出的付费低于市场保留价，也需要调整到市场保留价的水平上。(以广义第二高价4.2为例，很容易验证这种情况是可能发生的。)

市场保留价有两种设置方法，一是对整个竞价市场采用同样的保留价格；二是根据不同标的物(例如搜索广告里的关键词)的特性设置不同的保留价格。

### 4.1.3 价格挤压

在CPC广告网络中，eCPM可以表示成点击率和出价的乘积。即 $r = \mu \cdot v$ 。但是在有的情况下，我们有动机对此公式做一些微调，把它变成下面的形式：

$$r = \mu^{\kappa} \cdot v \quad (4.4)$$

其中的 $\kappa$ 为一个大于0的实数。我们可以考虑两种极端情况来理解 $\kappa$ 的作用：当 $\kappa \rightarrow \infty$ 时，相当于只根据点击率来排序，而不考虑出价的作用；反之，当 $\kappa \rightarrow 0$ 时，则相当于只根据出价来排序。因此，随着 $\kappa$ 的增大，相当于我们在挤压出价在整个竞价体系中的作用，因此我们把这个因子叫做价格积压(Squashing)因子。

价格积压因子的作用，主要是为了能够根据市场情况，更主动地影响竞价体系向着需要的方向发展。比如说，如果发现市场上存在大量的出价较高但品质不高的广告主，则可以通过调高 $\kappa$ 来强调质量和用户反馈的影响；如果发现市场的竞价激烈程度不够，则可以通过降低 $\kappa$ 来鼓励竞争；如果存在短期的财务压力，则需要将 $\kappa$ 调整到接近于1的范围，往往就可以使得整体营收有所上升。

### 4.1.4 整体竞价和计价过程

综合考虑上述几种竞价广告中常见的决策逻辑，在一次展示到来以后，实际的竞价决策过程可以用下面的代码来概要描述：

```

1  int auction(vector<int> & candidates, vector<float> & bids,
2             float MRP, float squash, int slotNum,
3             vector<int> & results, vector<float> & prices) {
4      int candNum = candidates.size();
5
6      // 遍历各个候选得到eCPM
7      vector<float> eCPMs;
8      eCPMs.resize(candNum, 1e-10f);
9      for (int c = 0; c < candNum; c++)
10         if (bids[c] >= MRP) // 跳过那些出价小于市场保留价的候选
11             eCPMs[c] = calcu_eCPM();
12

```

```

13 // 将所有候选按照eCPM排序
14 for (int c1 = 0; c1 < candNum; c1++)
15     for (int c2 = c1 + 1; c2 < candNum; c2++)
16         if (eCPMs[c1] < eCPMs[c2]) {
17             SWAP(candidates[c1], candidates[c2]);
18             SWAP(eCPMs[c1], eCPMs[c2]);
19         }
20
21 // 得到各竞价结果并计算定价
22 results.clear() prices.clear();
23 for (int r = 0; r < slotNum; r++) {
24     if (eCPMs[r] <= 1e-10f)
25         break;
26
27     // 按照GSP计算定价
28 }
29 }

```

我们用一个例子来直观地说明上述的综合竞价和定价过程：假设有一组广告竞争一个有多个slot的搜索广告展示机会，其出价和系统对其对其点击率的预估如下表中的第二行和第三行所示，那么计算出的eCPM，以及在点击发生时按照GSP和VCG定价策略向每个广告主收取的费用，如下表的后两行所示：

**Table 4.1.** 广告竞价过程示例

排序	出价	点击率	eCPM	GSP定价	VCG定价
1	1	2	3		
2	1	2	3		
3	1	2	3		
4	1	2	3		
5	1	2	3		
6	1	2	3		

## 4.2 广告网络系统架构

广告网络的典型系统架构如图4.2所示，其中广告投放的决策流程为：服务器接收前端用户访问触发的广告请求，首先根据上下文url和用户cookie从Page Attributes 和User Attributes系统中查

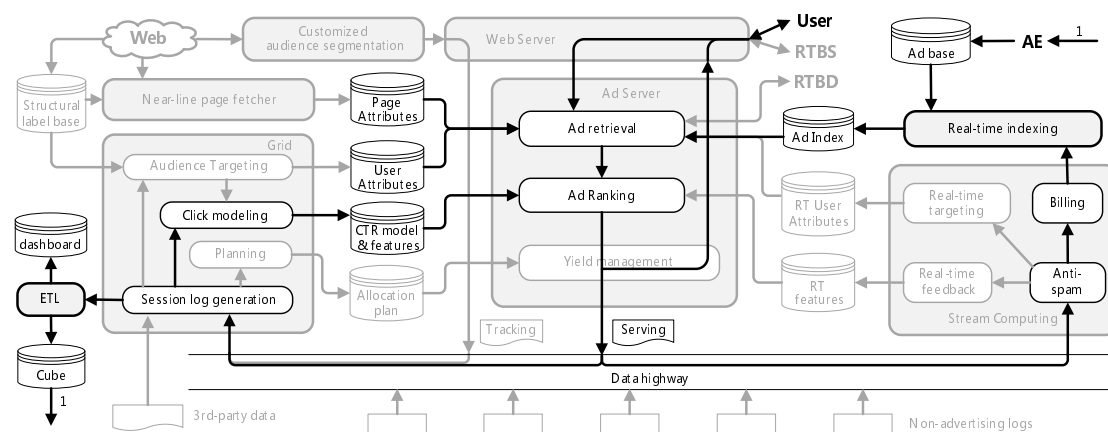


Fig. 4.1. 广告网络(Ad Network)系统架构示意

出相应的上下文标签和用户标签；然后用这些标签，以及其他一些广告请求条件从广告索引中找到符合要求的广告候选集合；最后，利用CTR预估模型计算所有的广告候选的eCPM，再根据eCPM排序选出赢得竞价的广告，并返回给前端完成投放。

从离线计算的流程来看，广告网络需要根据广告投放的历史展示和点击数据，对点击率预测进行建模。当然，实际的广告网络也往往需要同时提供受众定向的功能，因此这部分离线计算也需要进行。不过由于我们只给出最核心的功能块，因此没有特意强调这一部分。

由于广告网络广泛采用点击计费，准实时的计费和点击反作弊功能是必不可少的。我们可以采用下一章中介绍的流式计算平台来完成这些准实时功能，并且将这些结果及时反馈到事实建索引的模块来保证广告的及时上下线。由于计费问题计算相对简单，我们不作展开介绍，而反作弊的思路将在本书最后一章有专门讨论。

### 4.3 广告检索

竞价广告市场中，大量中小广告主的参与，以及复杂的定向条件，对检索技术提出了更高的要求。倒排索引是搜索引擎的关键技术，而广告的检索上也采用这样的框架。但是广告的检索问题也有一些自身的特点和需求，我们需要讨论一下这些需求下一些专用的检索技术。在合约式广告系统中，广告主数目一般来说不算太大，因而采用一般的检索方案完全可以满足要求；然而在服务大量中小广告主的广告网络中，探讨高效且精准的检索技术，有非常切实的产品意义。

第二章中介绍的基本倒排索引技术在广告问题上遇到了两个挑战：一是广告指定的定向条件组合，可以看成是一个由与或关系连接的布尔表达式，这样的布尔表达式的检索，很显然与上面搜索引擎对文本的“bag of words”假设不太一样；二是在上下文和用户兴趣信号比较丰富时，广告检索中的查询可能相当长，甚至会由上百个关键词组成，这种情况下的检索，也与搜索引擎

之中主要由1-4个term组成的query情形有区别：试想你把100个关键词同时输入到搜索框中，返回的结果会是你想要的么？根据这些切实的需求，我们来讨论一下在线广告系统中的检索技术。

#### 4.3.1 布尔表达式的检索

广告检索与普通搜索引擎检索第一个不同，是布尔表达式的检索问题。在受众定向的售卖方式下，一条广告对象不能再被看成是a bag of words，他们应该是一些定向条件组合成的布尔表达式，如下面给出的一些例子：

$$\begin{aligned} a_1 &= (\text{age} \in \{3\} \cap \text{geo} \in \{\text{北京}\}) \cup (\text{geo} \in \{\text{广东}\} \cap \text{gender} \in \{\text{男}\}) \\ a_2 &= (\text{age} \in \{3\} \cap \text{gender} \in \{\text{女}\}) \cup (\text{geo} \notin \{\text{北京}, \text{广东}\}) \\ a_3 &= (\text{age} \in \{3\} \cap \text{gender} \in \{\text{男}\} \cap \text{geo} \notin \{\text{广东}\}) \cup (\text{state} \in \{\text{广东}\} \cap \text{gender} \in \{\text{女}\}) \\ a_4 &= (\text{age} \in \{3, 4\}) \cup (\text{geo} \in \{\text{广东}\} \cap \text{gender} \in \{\text{男}\}) \\ a_5 &= (\text{state} \notin \{\text{北京}, \text{广东}\}) \cup (\text{age} \in \{3, 4\}) \\ a_6 &= (\text{state} \notin \{\text{北京}, \text{广东}\}) \cup (\text{age} \in \{3\} \cap \text{state} \in \{\text{北京}\}) \cup (\text{state} \in \{\text{广东}\} \cap \text{gender} \in \{\text{男}\}) \\ a_7 &= (\text{age} \in \{3\} \cap \text{state} \in \{\text{北京}\}) \cup (\text{state} \in \{\text{广东}\} \cap \text{gender} \in \{\text{女}\}) \end{aligned}$$

在这些例子中，我们用布尔表达式来表示广告的定向人群，并且写成析取范式(Disjunctive Normal Form, DNF)的形式。在这样的表达形式中，有两个概念要先做一下解释：一是每个DNF都可以分解成一个或多个conjunction的并，如 $a_1$ 可以分解成 $j_1 = (\text{age} \in \{3\} \cap \text{geo} \in \{\text{北京}\})$ 和 $k_2 = (\text{geo} \in \{\text{广东}\} \cap \text{gender} \in \{\text{男}\})$ 这两个conjunction；二是每个conjunction又可以进一步分解为一个或多个assignment的交，以 $c_1$ 为例，可以分解为 $\text{age} \in \{3\}$ 和 $\text{geo} \in \{\text{北京}\}$ 这样两个assignment。为了后文算法描述做准备，我们定义assignment、conjunction和DNF的数据结构如下：

```

1 struct Assignment
2 {
3     bool in;      // 表示该assignment是属于还是不属于
4     int label;    // 该assignment指定的标签类型
5     vector<int> values; // 该assignment指定的标签取值集合
6 }
7
8 typedef vector<Assignment> Conjunction;
9
10 typedef vector<Conjunction> DNF;
```

布尔表达式检索的问题有如下的两个特点，是我们设计算法的重要依据：首先，当某次广告请求的定向标签满足某个conjunction时，一定满足包含该conjunction的所有广告，这说明我们只要对conjunction建立倒排索引，并加上一层conjunction  $\rightarrow$  ad的辅助索引即可。其次，在conjunction的倒排索引中，有一项显而易见的直觉可以帮助我们减少计算。这一直觉即是：因此，令sizeof(query)表示广告请求中的定向标签个数，而sizeof(conjunction)表示某conjunction中的

含有“ $\in$ ”的assignment数目，当 $\text{sizeof}(\text{query}) < \text{sizeof}(\text{conjunction})$ 时，该conjunction一定不满足该次请求。

根据上述两个重要特点，我们可以设计出一个为布尔表达式检索定制的算法。该算法维护一个两层的倒排索引，即上面所说的conjunction和ad两层索引，后一个索引按照或的关系进行检索，而前一个索引有不太一样的结构，我们重点来看一下。在conjunction的索引中，我们把每个conjunction分解成一组(key, value)对，例如将 $\text{age} \in \{3, 4\}$ 分解成 $\text{age} \in \{3\}$ 和 $\text{age} \in \{4\}$ 两个term，这些term即是倒排索引的key，而“ $\in$ ”和“ $\notin$ ”操作符则放在倒排链表的具体元素上。利用到上文所说的assignment个数的约束，我们可以做的优化是将这一倒排索引按照 $\text{sizeof}(\text{conjunction})$ 分成若干部分，以提高检索效率。我们仍然以上文的一组广告为例来解释一下。这组广告的DNF可以按如下的方式分解成一些conjunctions：

$$a_1 = j_1 \cup j_4, a_2 = j_2 \cup j_6, a_3 = j_3 \cup j_7, a_4 = j_5 \cup j_4,$$

$$a_5 = j_6 \cup j_5, a_6 = j_6 \cup j_1 \cup j_7, a_7 = j_1 \cup j_7$$

其对应的倒排索引，也可以很容易地写成下面的形式：

$$j_1 \rightarrow \{a_1, a_6, a_7\}, j_2 \rightarrow \{a_2\}, j_3 \rightarrow \{a_3\}, j_4 \rightarrow \{a_1, a_4, a_7\},$$

$$j_5 \rightarrow \{a_4, a_5\}, j_6 \rightarrow \{a_2, a_5, a_7\}, j_7 \rightarrow \{a_3, a_6\}$$

而conjunction的倒排索引，注意到所有conjunction中最大的size为2，我们可以将倒排索引分成三部分，每部分中所有的conjunction其size都一样，按照这样的准则，最终形成的conjunction倒排索引应为下面的形式：

$$\text{size} = 0: (\text{geo}, \text{北京}) \rightarrow \{(j_6, \notin)\}, (\text{geo}, \text{广东}) \rightarrow \{(j_6, \notin)\}, Z \rightarrow \{(j_6, \in)\}$$

$$\text{size} = 1: (\text{age}, 3) \rightarrow \{(j_5, \in)\}, (\text{age}, 4) \rightarrow \{(j_5, \in)\}$$

$$\text{size} = 2: (\text{age}, 3) \rightarrow \{(j_1, \in), (j_2, \in), (j_3, \in)\}, (\text{geo}, \text{北京}) \rightarrow \{(j_1, \in)\}, (\text{gender}, \text{女}) \rightarrow \{(j_2, \in), (j_7, \in)\}, (\text{gender}, \text{男}) \rightarrow \{(j_3, \in), (j_4, \in)\}, (\text{geo}, \text{广东}) \rightarrow \{(j_3, \notin), (j_4, \in), (j_7, \in)\}$$

其中size为零的部分，包含哪些所有只有“ $\notin$ ”操作符的conjunctions。为了保证给定一个assignment，size为零的conjunctions至少出现在一个倒排表中，算法引入Z为一个特殊的term，并且将所有size为零的conjunctions都放在其倒排表中，并赋以一个“ $\in$ ”操作符。

我们在第二章的标准倒排索引类基础上加以改进，将这样的DNF索引类的代码列在下面，方便大家参考。在这段代码中，indexDNF对应上面提到的DNF的倒排索引，而indexConj对应于conjunction的一组倒排索引，其中每一个倒排索引中所有的conjunction都具有相同的size。需要注意的是，在查询函数当中，实际上的输入概念上应该是一个标签列表，而不是conjunction，不过由于conjunction也可以用来表示一个标签列表，为了接口的一致性，我们还是用conjunction来作为输入参数。

```

1 class DNFIndex : public InvIndex<Conjunction> {
2     private:
3         vector<InvIndex<Assignment> > & indexConjs;
4
5     public:
6         void add(DNF & dnf) {

```



```

7      // 更新DNF索引
8      InvIndex<Conjunction>::add(dnf);
9
10     // 遍历每个conjunction
11     int numConj = dnf.size();
12     for (int c = 0; c < numConj; c++) {
13         // 检查本conjunction是否已经被索引
14         if (indexDNFsize == indexDNF.size())
15             continue;
16
17         // 计算conjunction的size, 即其中属于操作符的个数
18         int conjSize = 0, numAsgn = dnf[c].size();
19         for (int a = 0; a < numAsgn; a++)
20             if (dnf[c][a].in) conjSize++;
21
22         // 在size = conjSize的index中加入本conjunction
23         if (indexConjs.size() <= conjSize())
24             indexConjs.resize(conjSize + 1);
25         add(indexConjs[conjSize], dnf[c], conjID);
26     }
27 }
28
29 void retrieve(Conjunction & exps, vector<int> & dnfIDs) {
30     // 只检查哪些size不大于表达式个数的conjunction索引
31     int sizeCheck = min(exps.size(), indexConjs.size());
32     int numExp = exps.size();
33
34     // 依次检查各个conjunction索引
35     for (int s = 0; s < sizeCheck; s++) {
36         // 得到各个相关倒排链, 并指向其第一个Entry
37         vector<list<Entry>::iterator> pointers;
38         for (int e = 0; e < numExp; e++) {
39             map<Assignment, list<Entry> >::iterator it;
40             it = indexConjs[s].find(exps[e]);
41             if (it == indexConjs[s].end()) break;
42             pointers.push_back(it -> second.begin());
43         }
44
45         if (pointings.size() < s) continue;
46

```

```

47         while (true) {
48             // 排序
49             for (int l1 = 0; l1 < s - 1; l1++)
50                 for (int l2 = l1 + 1; l2 < s; l2++)
51                     if (pointers[l1].first > pointers[l2].first
52                         )
53                         SWAP(pointers[l1], pointers[l2]);
54
55             // 检查是否有符合要求的conjunction
56             if (pointers[0] == pointers[s - 1]) {
57                 retrieve();
58
59                 for (int l = 0; l < s; l++)
60                     pointers[l]++;
61             }
62
63             // 每个倒排链向后滑动
64             for (int l = 0; l < s - 1; l++)
65                 while (pointers[l].first < pointers[s - 1].
66                     first)
67                     pointers[l]++;
68         }
69     };

```

媒体广告与搜索的检索技术，还有一点不太一样的地方，即在处理很多个term组成的query时的处理办法。我们考虑上下文定向的情形，当通过网页内容的关键词来匹配广告候选时，往往需要用十多个甚至几十个关键词去查询广告，再进行后续精细的排序。在这一情形下，如果仍然采用一般搜索引擎对query的处理办法，则会陷入两难的境地：如果假设各个term之间是与的关系，基本上不可能得到任何匹配的结果；如果假设各个term之间是或的关系，那么在检索阶段就会返回大量相关性很差的候选，为后续排序制造极大的麻烦。

解决这一问题的基本思路，是在检索阶段就引入某种评价函数，并按这一函数的评价结果来决定返回哪些候选。这一评价函数的设计有两个要求：一是合理性，即对最终排序的评价函数有直觉上合理的近似；二是高效性，即需要存在与倒排索引数据结构相契合的快速评价算法，否则就与在排序阶段展开计算没有差别了。简单来说，前人的研究发现，当我们选用一个线性函数(变量为各term)，且各变量的权重为正式，是可以构造出这样的快速评价检索算法的。我们将这样的线性函数定义为：

$$\text{score}(a, c) = \sum_{f \in \Omega(a) \cap \Omega(c)} \alpha_f v_f(a) \quad (4.5)$$

其中 $\Omega(a)$ 和 $\Omega(c)$ 分别表示 $a$ 和 $c$ 上不为零的特征集合，而 $v_f(a)$ 表示 $f$ 这一特征在 $a$ 广告上的贡献值。虽然常用的向量空间模型(Vector Space Model, VSM)不符合这一要求，但是如果不考虑余弦距离中的归一化分母，那么可以用这一线性函数在检索阶段做近似的预评估。这种情况下， $\alpha_f$ 即为关键词 $f$ 在上下文中的TFTDF，而 $v_f(a)$ 即为 $f$ 在某广告 $a$ 中的TFIDF。虽然 $\alpha_f$ 在不同的查询中取值不同，但在一次查询中是一组常数。

### 4.3.2 WAND算法

将线性评价函数计算过程加快的关键，在于使用两个上界：一是某个关键词 $f$ 在所有文档上贡献值的上界，我们记为 $ub_f$ ；二是某个文档中所有关键词的上界的和，也就是某篇文档对所有不同query的评价函数的上界，我们记为 $UB_a$ 。巧妙地利用这两个上界，可以在检索过程中排除掉大量不可能胜出的候选，从而达到快速评价的目的。这一方法，即为Andrei Broder等人提出的WAND(Weight AND)算法，也是上下文定向广告和内容推荐产品中非常实用的快速检索算法，我们以此算法为例看看长查询的检索过程。

WAND检索算法用到的数据结构，除了一个标准的倒排索引，还有一个用于保存检索过程中Top-K结果的堆，基于这两个数据结构，以及在建索引过程中生成的 $ub_f$ 和 $UB_a$ (这两个值的生成过程比较简单，因此我们略去相应的算法描述)，我们把WAND的检索算法用伪代码描述在下面。

```

1  class WANDIndex : public InvIndex<string> {
2      public:
3          void add(vector<string> & doc) {
4          }
5
6          void retrieve(vector<string> & query, vector<int> & docIDs) {
7              vector<string> terms = query;
8              int curDoc = 0;
9
10             vector<list<Entry> > posting;
11             for (int t = 0; t < terms.size(); t ++){
12                 posting[t] = next(curDoc, 0.0f);
13             }
14
15         private:
16             void next(int & curDoc, float minScore) {
17                 int numTerm = term.size();
18             }

```

```

19 // 按照倒排链中首个docID的升序对terms进行排序
20 for (int t1 = 0; t1 < numTerm - 1; t1 ++){
21     for (int t2 = t1 + 1; t1 < numTerm; t2 ++){
22         if (posting[t1] > posting[t2]) {
23             SWAP(terms[t1], terms[t2]);
24             SWAP(posting[t1], posting[t2]);
25         }
26     }
27
28 // 找到pivot term, 即使得累计的UB大于某个阈值的首个term
29 float accUB = 0.0f;
30 for (int t = 0; t < numTerm; t ++){
31     if ((accUB += terms[t].UB) > theta){
32         break;
33     }
34
35 // 已经没有可以胜出的候选
36 if (accUB < theta)break;
37 if (posting[t] == end())break;
38
39 if (pivot <= curDoc) {
40     // 该pivot已经被处理过, 任选一个倒排链向后滑动
41     int pos = rand() / RAND_MAX * t;
42     posting[pos] = pos.iterator.next(curDoc + 1);
43 }
44 else { // pivot > curDoc
45     if (posting[0] -> second == posting[t] -> second) {
46         // [0..t]之间的当前项一致, 成功找到一个候选
47         curDoc = pivot;
48         return(curDoc, posting);
49     }
50     else {
51         // 不满足, 从前面选一个倒排链向后滑动
52         int pos = rand() / RAND_MAX * t;
53         posting[pos] = pos.iterator.next(pivot);
54     }
55 }
56 }
57 }

```

### 4.3.3 线性评价函数的意义

这里讨论的相关性检索技术，仅仅考虑了相关性评价函数为线性的情况。乍看起来，这一条件严格限制了评价函数的适用范围。然而，如果考虑到广告的排序模型经常采用广义线性模型的建模方法的话，线性评价函数的适用范围就会大大扩展。我们采用后文中提到的基于广义线性模型的CTR预测模型，也同样可以套用此框架。

### 4.3.4 搜索广告的检索

## 4.4 广告排序概述

在第二卷里，我们已经简要介绍了eCPM指标及其分解方法。为了对本章内容提纲挈领，我们有必要对这一重要内容再做一回顾。

在竞价广告的环境下，需要根据eCPM对广告进行排序。按照点击和转化两个发生在不同阶段的行为，eCPM可以分解成点击率和点击价值的乘积，eCPM的估计，主要就是点击率预测和点击价值估计两个任务：

$$r(a, u, c) = \mu(a, u, c) \cdot v(a, u) \quad (4.6)$$

我们认为点击率 $\mu$ 是广告三个行为主体的函数，而点击价值则是用户 $u$ 和广告商 $a$ 的函数。后一点的假设有近似之处，因为实际上媒体的来源会影响用户对广告信息的信任程度，但我们为了概念清楚起见忽略这一影响。

在不同的市场环境下，具体的广告产品可能不需要对这两个量都进行估计，而且估计要求的准确程度也有所区别：对于按CPC结算的广告网络，需要尽可能准确地估计 $\mu$ ，和粗略地估计 $v$ ；对于在广告网络中采买的交易终端，主要需要估计 $v$ ；而对于DSP，则需要对两个都有较强的估计能力。我们将分别对这两部分估计的建模思路和方法做探讨，但是请大家注意，务必要根据实际的产品需求选择合适的建模方法。

另外，在实际的广告产品中，有可能同时存在若干种计费方式，因此其eCPM估算过程也不尽相同，我们用下面的一小段代码来说明在各种计费方式并存的情况下完整的eCPM计算逻辑。对然这一逻辑较为简单，还是很有助于大家深入理解eCPM的分解对应着不同的市场分工这一原理。

```

1  enum BidMode{CPM, CPC, CPS};
2
3  float calcu_eCPM(float bid, BidMode mode) {
4      switch (mode) {
5          case CPM:
6              return bid;
7          case CPC:
8              return predict_ctr() * bid;

```

```

9      case CPC:
10         return predict_ctr() * predict_click_value();
11     }
12 }
```

广告的点击率估计与搜索里的排序问题有关联，但又有明显的不同：点击率估计不能像搜索排序那样只要求结果排序的正确性。这—是因为点击率需要乘以bid才得到最后的排序，二是因为对一些需求方技术平台而言，需要尽可能准确地eCPM用于出价。因此，作为各种广告系统中通用的一项技术，点击率预测更适合被建模成回归(Regression)问题而不是排序问题。

所谓点击率预测，很自然的可以想到基于统计的估计：

$$pctr \approx click/impression \quad (4.7)$$

但如果一个广告从来没有被展示过，我们无法通过统计历史数据来预估未来的点击率，可以想到的解决方法是如果要展示的广告 $x$ 和一个展示过的广告 $y$ 类似，则可以预估 $x$ 的点击率与 $y$ 接近。如果将广告投影到特征空间做比较，则演化为即将介绍的点击率模型。

ctr预估模型从实证主义的统计估计演进为基于模型的点击率数学模型，按模型的复杂程度大体可以分为非线性模型和线性模型，我们对这俩者中的经典代表模型逻辑回归和GBDT做一下简要的介绍，感兴趣的朋友们可以从中体会最优化算法和统计机器学习在点击率预测模型中的重要作用，进而为后续了解业界前沿的点击率模型打下—定的基础。

## 4.5 点击率预测模型:逻辑回归

点击率预测的目的，是在 $(a, u, c)$ 组合与点击的概率之间建立函数关系，因此可以概念性地表示成下面的概率估计问题：

$$\mu(a, u, c) = p(h = 1 | a, u, c) \quad (4.8)$$

由于 $h$ 是个取值于 $\{0, 1\}$ 的二元变量，我们很自然想到的基础方法是逻辑回归(Logistic Regression, LR)，即：

$$p(h = 1 | a, u, c) = \sigma(w^\top x(a, u, c)) \quad (4.9)$$

其中 $x$ 表示 $(a, u, c)$ 组合上的特征矢量，即前文所谈的受众定向的输出及其派生的其他特征； $w$ 为各特征的的加权系数，也就是此模型需要优化的参数； $w^\top x$ 这一线性模型的输出经过Logistic sigmoid函数映射到 $(0, 1)$ 之间，此映射函数形式为：

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (4.10)$$

从方法上看，LR也是利用线性函数来解决非线性目标，属于线性分类器，认为数据在高维空间下线性可分。

$$\begin{aligned} p(h = 1 | a, u, c) &> p(h = 0 | a, u, c) \\ w^\top x(a, u, c) &> 0 \end{aligned} \quad (4.11)$$

### 4.5.1 广义线性模型视角

当我们面对一个多自变量的回归问题，根据目标值的特性，将其概率分布表示成指数族家族形式，基本的线性单元通过各种映射函数(link function) 建立该线性单元和各种分布的响应变量的关系，这就是广义线性模型。

对点击率预估，我们把点击事件( $h$ )看成一个二元取值的随机变量，那么其取值为真(click = 1)的概率就是点击率。因此，点击事件的分布可以写成以点击率 $\mu$ 为参数的二项(Binomial) 分布：

$$p(h) = \mu^h (1 - \mu)^{1-h} \quad (4.12)$$

可以推导，逻辑回归正是当目标值的分布服从Bernoulli分布时广义线性模型的一个特例，映射函数为 $\text{logit}(p) = \log(\frac{p}{1-p})$  函数。因此，有关广义线性模型的性质和结论，也可以直接适用于LR 模型。

通常采用极大似然估计来求解加权系数 $w$ ，如何求解的优化方法在后续章节深入剖析：

$$\begin{aligned} l(w) &= \log p(D|w) \\ &= \sum_{i=1}^n \log p(y_i = 1|x_i, w) + (1 - y_i) \log p(y_i = 0|x_i, w) \end{aligned} \quad (4.13)$$

### 4.5.2 最大熵视角

对LR模型的另一种解读，是把它看做最大熵模型在二分类问题下的特例。因此，我们在文献中看到用最大熵模型来预测点击率，与LR模型指的是同一类方法。最大熵模型的原理并不复杂，不要把鸡蛋放在一个篮子里就是最大熵原理的朴素解释。具体原则如下：

- 1.满足已知训练数据上的约束条件(特征的期望分布与经验分布一致)；

$$\sum_{x,y} \tilde{p}(x) p(y|x) f(x,y) = \sum_{x,y} \tilde{p}(x,y) f(x,y) \quad (4.14)$$

- 2.关于未知分布最合理的推断就是在满足已知知识的前提下，不对未知做任何假设从而做出的最不确定的推断；

$$\begin{aligned} H(p) &= - \sum_{x,y} \tilde{p}(x) p(y|x) \log(p(y|x)) \\ p_* &= \arg \max_p H(p) \end{aligned} \quad (4.15)$$

引入拉详细格朗日乘子，将最大熵转化为无约束优化问题，我们会发现最大熵和最大似然在优化目标上的等价性。这种等价关系数学上的变换并不复杂，我们不在这里讨论。

### 4.5.3 基本优化算法

在基础知识准备一章中，我们已经简要介绍了最优化问题的定义和基础的优化方法，本章起我们将对逻辑回归这一特定模型在海量数据下如何完成工程实现给出较为细致的解答，以期读者能够

领会最优化方法的重要性。在广告实践中，对海量的日志的分析决定了我们通常需要采用分布式计算框架来完成模型的训练求解，我们需要一个能够方便地在分布式计算框架下实现，在海量用户日志上完成训练的方法。

由于LR模型不存在闭式解，其优化方法必然需要迭代进行。典型的map/reduce分布式计算框架下，由于磁盘被用作迭代之间的数据交换手段，迭代的次数直接决定着训练算法的效率。因此，在每个迭代中尽可能完成更复杂深入的运算，减少迭代次数，是此处的关键。这样的思路适用于LR模型训练，也适用于许多map/reduce下的需要迭代求解的机器学习算法。

前文中说到LR模型可以视为最大熵模型的特例，最大熵模型的典型优化方法，比如IIS(Improved Iterative Scaling)也曾经被用于LR的更新。这种方法虽然物理意义明确，计算简单，却有着收敛速度慢的致命弱点。在目标函数可导的一般优化问题中，拟牛顿法是一族最常用的方法，因此也可以直接应用于LR问题的求解，下面我们简要回顾一下拟牛顿法。

牛顿法具有收敛快的特性，严格凸二次函数只一次迭代，但求解时却有着更新不稳定的问题，如Hessian矩阵可能奇异导致无法求逆确定后继点，非正定无法保证下降方向。拟牛顿法迭代的构造Hessian矩阵的逆矩阵的近似，来解决牛顿法的更新稳定性问题。当构造的校正矩阵形如 $\Delta H_k = \alpha_k u_k u_k^T + \beta_k (v_k u_k^T + u_k v_k^T)$ 时，即经典的BFGS方法。BFGS方法在收敛性质和数值计算方面稳定，且仅需一阶导数，不必计算hessian矩阵的逆矩阵。如下附上BFGS迭代求解的代码片段。

```

1
2 SparseVector s = new SparseVector();
3 SparseVector y = new SparseVector();
4 double fx0 = f.eval(x0);
5 SparseVector dfx0 = df.eval(x0);
6 int iter_num = 0;
7 while(iter_num < MAX_ITER_NUM && dfx0.norm_2() > TOL) {
8     iter_num++;
9     //计算搜索方向和步长
10    d = H.multiply(dfx0).scale(-1.0);
11    fxt = lineSearch.search(f, df, x0, dfx0, fx0, d, xt, dfxt);
12    s = xt.minus(x0);
13    y = dfxt.minus(dfx0);
14
15    //迭代更新Hessian矩阵，只存储下三角部分
16    SparseVector tmp = H.multiply(y);
17    double rho = 1.0 / y.dot(s);
18    double tmp2 = tmp.dot(y) * rho * rho + rho;
19    for(int i = 0; i < H.size1(); ++i) {
20        for(int j = 0; j <= i; ++j) {

```



```

21         double t = -rho * (s.getValue(i) * tmp.getValue(j) + s.
                getValue(j) * tmp.getValue(i)) + tmp2 * s.getValue(i)
                * s.getValue(j);
22         H.setValue(i, j, H.getValue(i, j) + t);
23     }
24 }
25
26 //进入下一轮迭代
27 x0.swap(xt);
28 dfx0.swap(dfx);
29 fx0 = fx;
30 }

```

从上述代码片段中可以看出，BFGS需要存储Hession矩阵的逆矩阵的近似 $H_k$ ，这个矩阵一般来说是稠密的，因此空间复杂度为 $O(N^2)$ 。虽然BFGS方法可以用近似Hession的方案来解决更新稳定性问题，但在动辄过亿维度的高维空间建模中，Hession矩阵的尺寸过大，以至于根本无法在内存中存放。

解决这一问题的思路，是仅仅保留最近几次更新的一些状态矢量，然后利用这些状态矢量和当前的梯度，直接计算出更新方向，这种方法，我们称之为L-BFGS(Limited-memory BFGS)。L-BFGS的核心思想，是根据前几次的函数值变化和梯度变化来近似地拟合Hession矩阵的逆。我们先来回顾一下，在BFGS的迭代过程中，如果用 $s_k = x_{k+1} - x_k$ 为前后两次自变量的变化值， $y_k = \nabla_{k+1} - \nabla_k$ 表示前后两次梯度的改变值，则Hession矩阵的更新公式可以表示为：

$$H_{k+1} = \left( I - \frac{s_k y_k^\top}{y_k^\top s_k} \right) H_k \left( I - \frac{y_k s_k^\top}{y_k^\top s_k} \right) + \frac{s_k s_k^\top}{y_k^\top s_k} = V_k^\top H_k V_k + \rho_k s_k s_k^\top \quad (4.16)$$

其中 $\rho = 1/y_k^\top s_k$ ， $V_k = (I - \rho s_k y_k^\top)$ 。如果对此迭代公式展开并做截断，只保留前 $m$ 次的状态量，则其近似方式可以表示成下式：

$$H_k = (V_{k-1}^\top \cdots V_{k-m}^\top) H_k^0 (V_{k-m} \cdots V_{k-1}) + \rho_{k-m} (V_{k-1}^\top \cdots V_{k-m+1}^\top) s_{k-m} s_{k-m}^\top (V_{k-m+1} \cdots V_{k-1}) \\ + \rho_{k-m+1} (V_{k-1}^\top \cdots V_{k-m+2}^\top) s_{k-m} s_{k-m}^\top (V_{k-m+2} \cdots V_{k-1}) + \cdots + \rho_{k-1} s_{k-1} s_{k-1}^\top \quad (4.17)$$

其中 $V_k = I - \rho_k s_k y_k^\top$ ，而 $H_0$ 是在首次迭代时设定的Hession逆的初值，为降低计算复杂度，实际中比较有效的选择是令其为一个对角阵： $H_k^0 = \gamma_k I$ ，其中

$$\gamma_k = \frac{s_{k-1}^\top y_{k-1}}{y_{k-1}^\top y_{k-1}}$$

在这样的表示下， $H_k$ 是可以在每次迭代中高效率地计算出来的。如下附上LBFGS迭代求解的代码片段。

```

1 SparseVector LBFGSLoop(SparseVector q,LinkedList<SparseVector> s,
  LinkedList<SparseVector> y,LinkedList<Double> rho) {
2     if(s.isEmpty()) {
3         return q;
4     }
5     else {
6         Iterator<SparseVector> iter1 = s.descendingIterator();
7         Iterator<SparseVector> iter2 = y.descendingIterator();
8         Iterator<Double> iter3 = rho.descendingIterator();
9         ListIterator<SparseVector> it1 = s.listIterator(0);
10        ListIterator<SparseVector> it2 = y.listIterator(0);
11        ListIterator<Double> it3 = rho.listIterator(0);
12        LinkedList<Double> alpha = new LinkedList<Double>();
13        while(iter1.hasNext()) {
14            double tmp = iter3.next() * q.dot(iter1.next());
15            q.plusAssign(-tmp, iter2.next());
16            alpha.addFirst(tmp);
17        }
18        double tmp = s.getLast().dot(y.getLast()) / y.getLast().dot
          (y.getLast());
19        SparseVector r = q.scale(tmp);
20        while(it1.hasNext()) {
21            double beta = it3.next() * r.dot(it2.next());
22            r.plusAssign(alpha.pollFirst() - beta, it1.next());
23        }
24        return r;
25    }
26 }
27
28 LinkedList<SparseVector> s = new LinkedList<SparseVector>();
29 LinkedList<SparseVector> y = new LinkedList<SparseVector>();
30 LinkedList<Double> rho = new LinkedList<Double>();
31 double fx0 = f.eval(x0);
32 SparseVector dfx0 = df.eval(x0);
33 int iter_num = 0;
34 while(iter_num < MAX_ITER_NUM && dfx0.norm_2() > TOL) {
35     iter_num++;
36     //计算搜索方向和步长
37     q.assign(dfx0);

```

```

38     d.assignTmp(LBFGSLoop(q,s,y,rho).scale(-1.0));
39     fxt = lineSearch.search(f, df, x0, dfx0, fx0, d, xt, dfxt);
40
41     //更新最近m轮状态矢量列表s,y
42     if(m >= M) {
43         s.pop();
44         y.pop();
45         rho.pop();
46     }
47     ++m;
48     s.add(xt.minus(x0));
49     y.add(dfxt.minus(dfx0));
50     rho.add(1.0 / y.getLast().dot(s.getLast()));
51
52     //进入下一轮迭代
53     x0.swap(xt);
54     dfx0.swap(dfxt);
55     fx0 = fxt;
56 }

```

可以很容易验证,上面每一步迭代的空间和时间复杂度都降低到了 $m \times N$ ,如果我们选择一个比较小的 $m$ ,就可以认为其复杂度接近线性,这在大多数较高维度特征空间上建模的应用中就可以达到实用水平了。在迭代的前 $m-1$ 步,L-BFGS和BFGS是没有区别的。

除了L-BFGS,信任域方法(Trust-region)也被证明对解LR问题很有效,而且往往可以更快地收敛。关于Trust-region方法在LR模型上的应用以及实际应用中的收敛特性,我们在下一节中介绍。

#### 4.5.4 信任域法

信任域法是个相对于线搜索方法对偶的优化算法框架,在广泛得到应用的开源实现liblinear中被采用,每一轮内层迭代采用共轭梯度法逼近求解一个约束优化子问题,与上一节中说到的lbfgs方法相比,每一轮迭代的代价增加了,但总体的迭代数目更少,所以往往可以更快的收敛。

梯度下降法,牛顿法,拟牛顿法,共轭梯度法都属于线搜索方法。它们的共同特点是在当前迭代点 $x^k$ 处寻找下一个迭代点 $x^{k+1}$ 时,首先确定一个下降方向:梯度下降法使用的是负梯度方向 $-\nabla f(x^k)$ ;牛顿法使用的是牛顿方向 $-\nabla^2 f(x^k)^{-1} \nabla f(x^k)$ ;拟牛顿法使用的是拟牛顿方向 $-H_k \nabla f(x_k)$ ;共轭梯度法使用的是共轭方向。然后沿着这个下降方向进行一维线搜索。这种搜索策略可以概括为先方向,后步长。

信赖域(Trust-Region)法使用一种不同的搜索策略,其基本思想是:每次迭代时,限制在 $x_k$ 的一个可信球形邻域里直接搜索 $x_{k+1}$ ,同时决定下次迭代的方向和步长,如果当前可信域内

找不到后继点，则缩小信任域范围。记位移 $d = x^{k+1} - x^k$ ，这即是要求搜索空间为 $\|d\|_2^2 \leq \Delta_k$ ，称 $\{x: \|x - x^k\|_2^2 \leq \Delta_k\}$ 为置信域， $\Delta_k$ 为置信半径。另一方面，我们用一个模型函数 $m_k(d)$ 来近似原来的目标函数 $f(x_k + d)$ 。模型函数通常取为二次函数：

$$m_k(d) = f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T H_k d$$

其中 $H_k$ 为 $f$ 在 $x^k$ 处的Hessian矩阵或其一个近似。当 $H_k = \nabla^2 f(x^k)$ 时， $m_k(d)$ 即是 $f$ 在 $x^k$ 处的二阶泰勒展开。当 $f$ 满足一定的光滑条件时，模型函数和原来的目标函数在 $x^k$ 的一个小邻域内的误差比较小，因此我们可以通过寻找模型函数在置信域中的最小点来寻找 $f$ 的改进点。具体的说，每一次迭代需要解决下面形式的子问题：

$$\begin{aligned} \min \quad & f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T H_k d \\ \text{s.t.} \quad & \|d\|_2^2 \leq \delta_k \end{aligned} \quad (4.18)$$

为了实现信赖域搜索策略，还需要明确：一、置信半径的选取；二、子问题4.18的求解。子问题4.18是一个目标函数为二次函数的约束优化问题，在liblinear中采用共轭梯度的近似解来逼近求解，具体细节将在后续介绍，这里先介绍怎样选择合适的置信半径。一般的策略是比较模型函数和目标函数的下降量：

$$\rho_k = \frac{f(x^k) - f(x^k + d)}{m_k(0) - m_k(d)}$$

如果 $\rho_k \leq 0$ ，说明目标函数值没有改进；若模型函数较真实的逼近了目标函数，我们期望 $\rho_k$ 的值接近与1；如果 $\rho_k$ 的值较小，说明在当前置信域内，模型函数和目标函数差别较大，需要缩小当前的置信区域；如果 $\rho_k$ 的值较大，可以在下次迭代时适当伸长收敛半径。我们以liblinear里的开源代码为例，附上Trust Region算法主流程的代码片段：

```

1 //控制是否更新迭代的参数
2 double eta1 = 0.25, eta2 = 0.75;
3 //控制置信半径条件的参数
4 double sigma1 = 0.25, sigma2 = 0.5, sigma3 = 4;
5 f = fun_obj.fun(w);
6 fun_obj.grad(w, g);
7 delta = euclideanNorm(g);
8 double gnorm1 = delta;
9 double gnorm = gnorm1;
10 for(int iter=1; iter <= max_iter; iter++) {
11     //直接求解子问题得到位移s
12     cg_iter = trcg(delta, g, s, r);
13     System.arraycopy(w, 0, w_new, 0, n);
14     daxpy(one, s, w_new);
15     //模型函数的预估下降量和目标函数的真实下降量

```

```

16     gs = dot(g, s);
17     prered = -0.5 * (gs - dot(s, r));
18     fnew = fun_obj.fun(w_new);
19     actred = f - fnew;
20     //初始化第一轮迭代的置信半径
21     snorm = euclideanNorm(s);
22     if (iter == 1) delta = Math.min(delta, snorm);
23     //根据目标函数是否逼近模型函数, 缩放下一步迭代的置信域半径
24     if (fnew - f - gs <= 0)
25         alpha = sigma3;
26     else
27         alpha = Math.max(sigma1, -0.5 * (gs / (fnew - f - gs)));
28     if (actred < eta0 * prered)
29         delta = Math.min(Math.max(alpha, sigma1) * snorm, sigma2 *
30             delta);
31     else if (actred < eta1 * prered)
32         delta = Math.max(sigma1 * delta, Math.min(alpha * snorm,
33             sigma2 * delta));
34     else if (actred < eta2 * prered)
35         delta = Math.max(sigma1 * delta, Math.min(alpha * snorm,
36             sigma3 * delta));
37     else
38         delta = Math.max(delta, Math.min(alpha * snorm, sigma3 *
39             delta));
40     //进入下一轮迭代
41     if (actred > eta0 * prered) {
42         System.arraycopy(w_new, 0, w, 0, n);
43         f = fnew;
44         fun_obj.grad(w, g);
45         gnorm = euclideanNorm(g);
46         if (gnorm <= eps * gnorm1) break;
47     }
48 }

```

如上所述, 在 $x^k$ 的一个小的置信邻域内, 我们用一个二次正定函数近似原来的目标函数, 认为两者误差比较小, 因此我们可以通过寻找模型函数在置信域中的最小点来寻找 $f$ 的改进点。

在trust region的算法框架下, 置信域中子问题的求解是相对lbfgs中的单步迭代来说更为复杂的任务, 我们希望该子问题的求解能够以尽量小的代价完成该任务, 使得整体的迭代性能良好。在开源实现liblinear中, 即采用了共轭梯度法来近似求解的高效方案。和无约束优化中的共轭

梯度法略有不同的是，这里需要满足 $\|d\|_2^2 \leq \delta_k$ 的约束条件，考虑到子算法中位移量是递增的，即 $\|s^i\| \leq \|s^{i+1}\|$ ，我们在位移量超出置信域的该轮内迭代中，将位移的步长受限在置信域即可。

对于目标函数为二次正定函数 $f(x) + \nabla f(x)^T d + \frac{1}{2} d^T H d$ 时，共轭梯度法可以在一组关于 $H$ 共轭的向量组上做简单的 $n$ 次迭代后达到收敛，避免了需要存储和计算 $H$ 矩阵的缺点，这意味着不用直接计算和存储Hessian矩阵，这无疑是非常吸引人的。具体来说，每次迭代中的主要操作是Hessian矩阵与向量 $s$ 的乘积，由于 $X$ 是稀疏的，数据量不是特别大时不需要直接求Hessian矩阵也可以得到该乘积：

$$Hessian.d = (I + CX^T DX)d = d + C.X^T (D(Xd))$$

仍以liblinear里的开源代码为例，附上子问题求解的代码片段：

```

1 double[] d = new double[n];
2 double[] Hd = new double[n];
3 double rTr, rnewTrnew, cgtol;
4 //初始化位移0, 方向d=-g
5 for (int i = 0; i < n; i++) {
6     s[i] = 0;
7     r[i] = -g[i];
8     d[i] = r[i];
9 }
10 cgtol = 0.1 * euclideanNorm(g);
11 rTr = dot(r, r);
12 while (true) {
13     if (euclideanNorm(r) <= cgtol) break;
14     //矩阵乘法计算Hd
15     fun_obj.CalHd(d, Hd);
16     //计算步长
17     double alpha = rTr / dot(d, Hd);
18     daxpy(alpha, d, s);
19     if (euclideanNorm(s) > delta) {
20         //到达置信域外, 重新计算步长约束位移
21         reCalS(alpha, d, s);
22         break;
23     }
24     //计算方向
25     alpha = -alpha;
26     daxpy(alpha, Hd, r);
27     rnewTrnew = dot(r, r);
28     double beta = rnewTrnew / rTr;
29     scale(beta, d);

```

```

30     daxpy(one, r, d);
31     rTr = rnewTrnew;
32 }

```

#### 4.5.5 ADMM计算框架

从上节中置信域方法与拟牛顿法的比较中，我们可以看到，随着每轮迭代的代价增加，迭代次数也随之降低了，因此有可能会带来收敛速度的提升。是否存在一种普适性的思路，使得我们可以对一般的迭代求解问题减少其迭代次数呢？在互联网大规模数据处理的挑战诞生以后，学术界对这个问题也进行了深入的研究，产生了一些颇具启发意义的方法。这里我们介绍一种称为Alternative Directional Method of Multipliers (ADMM)的计算框架。

从方法论上说，要降低迭代数目，必然要求在一个迭代内部完成更复杂的计算。要了解ADMM，需要先介绍一下扩展拉格朗日(Augmented Lagrangian)的方法。在第二章中，我们介绍了带约束优化的拉格朗日，如果只考虑等式约束为一个线形约束( $Ax = b$ )的形式，我们可以构造如下的扩展拉格朗日：

$$L_{\rho}(x, y) = f(x) + y^{\top}(Ax - b) + \frac{\rho}{2} \|Ax - b\|_2^2 \quad (4.19)$$

可以很容易地验证，这一形式可以得到与标准拉格朗日一样的解。引入一个二阶惩罚项，往往会使得问题求解的过程更好地收敛。根据[8]中的介绍，这一问题可以用dual ascent方法求解。而问题得以分布式求解的关键，是当目标函数可以分解成下面的形式时，可以发现存在有效的分解迭代求解方案：

$$\begin{aligned} \min \quad & f(x) + g(z) \\ \text{s.t.} \quad & Ax + Bz = c \end{aligned} \quad (4.20)$$

对应的迭代求解方案，是一个 $x, z, y$ 依次迭代更新的过程：

$$\begin{aligned} x_{k+1} &\leftarrow \arg \min_x (f(x) + (\rho/2) \|Ax + Bz_k - c + s_k\|_2^2) \\ z_{k+1} &\leftarrow \arg \min_z (g(z) + (\rho/2) \|Ax_{k+1} + Bz - c + s_k\|_2^2) \\ s_{k+1} &\leftarrow s_k + Ax_{k+1} + Bz_{k+1} - c \end{aligned} \quad (4.21)$$

其中我们为了表达上的整洁，我们将 $y$ 换成了其归一化的形式 $s = (1/\rho)y$ 。在许多典型的利用ADMM分布式求解的问题当中，上面的第一个公式用于各部分数据的局部参数更新，第二个公式用于将各部分得到的局部优化参数综合成全局的参数；而第三个公式中对对偶变量的更新则是使得整个过程稳定和高效的关键。我们可以结合LR问题的具体情况，来解释一下利用ADMM方法分布式求解的过程。

按照公式4.22的结构，我们可以将LR的优化问题4.9做一些改写，使之变成能更方便地利用ADMM进行分布式更新的形式：

$$\begin{aligned} \min \quad & \sum_{i=1}^N \sigma(F^{(i)}x^{(i)}) + r(z) \\ \text{s.t.} \quad & x^{(i)} - z = 0, \quad i = 1, \dots, N \end{aligned} \quad (4.22)$$

这里的 $i = \{1, \dots, N\}$ 表示数据集分裂后的各个部分， $x^{(i)}$ 对应于某一部分数据上得到的LR参数(对应于上面ADMM问题里的 $x$ )，而 $z$ 为整体决策后的参数。 $F^{(i)}$ 表示由第 $i$ 块数据个样本的特征拼成的矩阵。问题的约束条件是表明求解收敛时各部分的参数应该等于整体参数，这是非常自然需要满足的。目标函数里的 $r(z)$ 代表的是求解过程的对参数的某种regularization项，比如参数矢量的 $L_1$ -norm或 $L_2$ -norm。这一问题与上述的ADMM问题显然是一致的，因此，我们可以得到用ADMM方法迭代求解此问题的方法：

$$\begin{aligned} x_{k+1}^{(i)} &\leftarrow \arg \min_{x^{(i)}} \left( \sigma(F^{(i)}x^{(i)}) + (\rho/2) \|x^{(i)} - z_k + s_k^{(i)}\|_2^2 \right) \\ z_{k+1} &\leftarrow \arg \min_z \left( r(z) + (N\rho/2) \|z - \bar{x}_{k+1} - \bar{s}_k\|_2^2 \right) \\ s_{k+1}^{(i)} &\leftarrow s_k^{(i)} + x_{k+1}^{(i)} - z_{k+1} \end{aligned} \quad (4.23)$$

我们来仔细分析一下这一更新的过程：1. 首先，在每个数据分块上，分别执行第一个公式中的对应更新，得到该数据分块上更新后的参数，这一步是可以分布式进行的，而且各个数据块之间是不需要进行通信的；2. 然后，根据各部分更新得到的参数，执行第二个公式得到综合以后的整体参数 $z$ ；3. 根据第三个公式更新对偶变量 $s$ ，并将更新后的 $z$ 和 $s$ 分发至各个数据块的处理单元。这一过程可以非常自然地用map/reduce方式来实现，其中1步骤对应着各个mapper，而2和3步骤对应着一个唯一的reducer。

在map/reduce的分布式计算框架下，ADMM方法对于效率的提高往往有较好的效果。我们可以将此过程与L-BFGS的迭代的更新过程比较一下，从而更深入地理解这一加速的概念性原理。在L-BFGS当中，每个mapper，即分布式部分计算过程非常简单，只需要在每个样本上都参数求导数，再将导数累加即可。而在ADMM方法中，mapper计算的过程变得复杂了很多，由简单的导数计算变成了一个LR的求解问题，也就是说mapper的计算本身就需要迭代才可以完成。可以直观地这样认为，由于在每个mapper中作了更多的计算工作，使得整体求解过程的收敛更快。同时需要注意的是，实际上在每个mapper中复杂的更新过程并不会带来mapper计算代价的显著增加，这是由于每个mapper所需要处理的数据量有限，因此可以放在内存中，于是在分布式计算中最主要的开销即I/O开销并没有增加。因此，可以认为ADMM的方法是用局部内存的更多的访问换得了全局map/reduce过程的迭代次数减少，从而提高了效率。

我们还是以LR模型为例来介绍ADMM方法在Hadoop中实现的代码片段。先介绍上述步骤一：在每个数据分块上，分别执行第一个公式中的对应更新，得到该数据分块上更新后的参数。即在Hadoop的map阶段，每个mapper中采用lbfgs求解该数据分块上更新的参数。

```

1 //每个mapper中处理的split数据按InstancesWritable格式放在内存中
2 public void map(LongWritable key, InstancesWritable value, Context
   context) throws IOException, InterruptedException {
3     FileSplit split = (FileSplit) context.getInputSplit();

```



```

4      String splitId = key.get() + "@" + split.getPath();
5      splitId = AdmmIterationHelper.removeIpFromHdfsFileName(splitId)
        ;
6      AdmmMapperContext mc;
7      //读取上轮迭代reduce中更新后的参数分发至各个数据块的处理单元。
8      if (iteration == 0) {
9          mc = new AdmmMapperContext(rho);
10     }
11     else {
12         mc = assembleMapperContextFromCache(splitId);
13     }
14     //采用lbfgs,在上轮迭代的XInitial基础上增量更新XUpdate
15     Admm admm = new Admm(value,mc.getSInitial(),mc.getZInitial(),mc
        .getRho(),1.0);
16     LBFGS lbfgs= new LBFGS();
17     SparseVector XUpdate = new SparseVector(mc.getXInitial());
18     double local_loss = lbfgs.minimize(admm, XUpdate);
19
20     AdmmReducerContext reducerContext = new AdmmReducerContext(
21         mc.getSInitial(),
22         mc.getXInitial(),
23         XUpdate,
24         mc.getZInitial(),
25         local_loss,
26         mc.getRho(),
27         regularizationFactor);
28     //将各部分更新得到的参数汇总至reduce, 更新z和s
29     context.write(ZERO, new Text(splitId + ":@" +
        AdmmIterationHelper.admmReducerContextToJson(reducerContext)
        ));
30 }

```

reduce阶段对应着整体参数 $z$ 和对偶变量 $s$ 的更新。

```

1 public void reduce(IntWritable key, Iterator<Text> values, Context
    context) throws IOException,InterruptedException {
2     //将各部分更新得到的参数汇总
3     AdmmReducerContextGroup rcontext = new AdmmReducerContextGroup(
        values, numberOfMappers, LOG, iteration);
4     //根据s和更新后的x更新z
5     SparseVector zUpdated = getZUpdated(rcontext);

```

```

6     SparseVector[] xUpdated = rcontext.getXUpdated();
7     String[] splitIds = rcontext.getSplitIds();
8
9     for (int mapperNumber = 0; mapperNumber < context.
        getNumberOfMappers(); mapperNumber++) {
10         //根据更新后的x和z更新s
11         SparseVector sUpdated = getSUpdated(rcontext, mapperNumber,
            zUpdated);
12         //将更新后的各参数写入字典
13         AdmmMapperContext admmMapperContext =
14         new AdmmMapperContext(sUpdated, xUpdated[mapperNumber],
            zUpdated,
15             rcontext.getRho() * rcontext.getRhoMultiplier(),
16             rcontext.getLambda(), rcontext.getPrimalObjectiveValue
                (),
17             rcontext.getRNorm(), rcontext.getSNorm());
18         String currentSplitId = splitIds[mapperNumber];
19         outputMap.put(currentSplitId, AdmmIterationHelper.
            admmMapperContextToJson(admmMapperContext));
20     }
21     //将参数字典写入hdfs供下轮迭代读取
22     context.write(ZERO, new Text(AdmmIterationHelper.mapToJson(
        outputMap)));
23     if (rcontext.getRNorm() > THRESHOLD || rcontext.getSNorm() >
        THRESHOLD) {
24         context.getCounter(IterationCounter.ITERATION).increment(1)
            ;
25     }
26 }
27 //更新z
28 private SparseVector getZUpdated(AdmmReducerContextGroup context) {
29     int numMappers = context.getNumberOfMappers();
30     SparseVector xAverage = context.getXUpdatedAverage();
31     SparseVector sAverage = context.getSInitialAverage();
32     SparseVector zUpdated = new SparseVector();
33     double zMultiplier = (numMappers * context.getRho()) / (2 *
        context.getLambda() + numMappers * context.getRho());
34     //对Z的二次函数求极小值
35     zUpdated = xAverage.add(sAverage).scale(zMultiplier);
36     return zUpdated;

```

```

37 }
38 //更新s
39 private SparseVector getSUpdated(AdmmReducerContextGroup context,
    int mapperNumber, SparseVector zUpdated) {
40     SparseVector sInitial = context.getSInitial()[mapperNumber];
41     SparseVector xUpdated = context.getXUpdated()[mapperNumber];
42     SparseVector sUpdated = new SparseVector();
43     sUpdated = sInitial.add(xUpdated).minus(zUpdated);
44     return sUpdated;
45 }

```

虽然我们在这里是以LR模型为例来介绍ADMM方法的应用，实际上这种方法可以应用于许多常见的机器学习模型，而且大都在map/reduce的计算框架下可以达到减少总迭代次数，从而提高效率的目的。对于优化方法的选择，实际上没有免费的午餐，ADMM对效率的提高也主要适用于map/reduce这种迭代的开销非常高的计算框架中。

#### 4.5.6 点击率模型的校正

点击率预测问题有一个数据上的挑战，就是正例和负例样本严重不均衡，特别是在显示广告点击率只有千分之几的情况下。在很多建模方法中，这样严重的不均衡会带来模型估计上的问题，我们仍然以LR模型为例，来讨论一下模型存在偏差的原因，以及相应的校正方法。

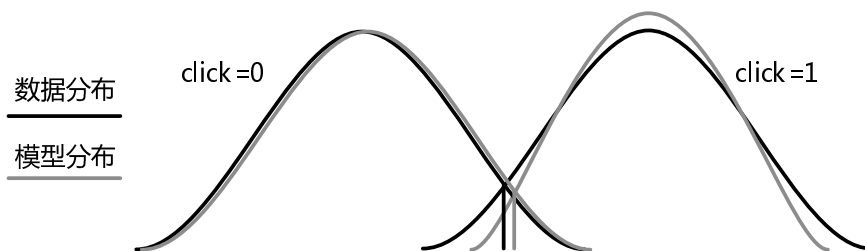


Fig. 4.2. 正负例样本不均衡时点击率模型存在偏差的原因示意

我们可以通过图4.5.6来示例性地描述点击率模型可能存在偏差的原因。假设我们分别用两个高斯分布来描述click=0和click=1情形下的特征分布。熟悉统计的读者都知道，高斯分布方差的最大似然估计是有偏的(为了得到方差的无偏估计，我们需要将样本数目减去1来计算方差)，而这一偏差的方向是对方差有所低估，并且样本数目越少，低估越严重。由于click=1时的数据数量远远小于click=0时的数据数量，对前者的方差低估就会更严重，对应图4.5.6所示，前者的分布(右侧的高斯)会变得更窄一些。加入我们用这两个最大似然估计的高斯分布来决定click=0和click=1两个类的边界点，就会出现比实际边界点向右偏移的情况。这也就意味着更

多的样本被分到了click=0这个类中，或者说意味着点击率将会被系统性地低估一些。这里的解释虽然只是示意性的，却与LR模型里点击率估计有偏的原理基本一致。

幸运的是，消除这一点击率估计的偏差并不困难，这是因为这一偏差是可以计算出来，并且存在闭式解的！实际上对这一偏差的系统性分析可以上升到GLM的层次来研究，而文\*中也对LR情形下的具体情况进行了深入讨论。简单来说，LR模型的最大似然估计存在的偏差可以表示成下面的形式：

$$\text{bias}(w) = (X^T W X)^{-1} X^T W \xi \quad (4.24)$$

上式中， $W = \text{diag}\{\pi_i(1 - \pi_i)w_i\}$ ，其中 $\pi_i = (1 + e^{-w^T f_i})^{-1}$ ； $X = (x_1 \cdots x_N)$ 表示所有样本特征组成的矩阵；

## 4.6 点击率模型举例:GBDT

### 4.6.1 点击率预测特征

点击率预测问题的挑战主要在于如何使模型能捕捉高度动态的市场信号，以达到更准确的预测。这一挑战可以用在线的模型学习算法，或者用快速更新的动态特征来解决，从方法论上说，这两种思路是对偶的，我们会作对比介绍，但重点放在第二种思路，因为其工程扩展性更强。从计算角度来说，另一个挑战是如何在海量的日志数据上快速有效地完成训练迭代，这也将是我们讨论的重点。

上一节主要讨论的是点击率预测模型侧的问题，这一节我们来看看特征侧的技术。从受众定向得到的所有 $f(a, u, c)$ ，以及这些特征的运算，可以组合出大量的特征供模型选择，这是大多数机器学习问题共同的方法。这样的特征生成方法，是点击率预测特征的基础方法，不过在广告这样的问题里也遇到一些挑战：一是组合特征数量可能巨大，使得模型的参数数目也非常大，工程上参数更新和在线计算都需要比较高效的设计；二是模型动态性的本质要求参数快速更新，而在多台ad server之间协同进行online learning并非一件易事。

### 点击率特征的构成

为什么广告展示的决策可以提取出大量的特征呢？这是因为在 $(a, u, c)$ 三个维度上，都存在着人为指定或机器生成的多种标签，这些标签有的相互独立，也有的存在一定的层级关系。比如以 $a$ 上的标签为例，我们在一般的广告运营当中，将广告维度上的标签分成广告主(Advertiser)、广告活动(Campaign)、广告组(Ad Group)、广告创意(Creative)这几个层次。在预测的过程中，这样的层级结构对于我们更稳健地估计某个广告，特别是新广告的点击率有非常大的帮助。

### 静态与动态特征

在机器学习问题中，有一项很重要的方法论，即某项模型侧的技术，一般都可以找到特征侧的对偶方案。那么如何设计特征方案使得模型快速演进的需求变得不那么迫切呢？当然就是让特征变

成快速演进的！我们考虑上面所介绍的特征，假设某个组合用户性别标签为男性，上下文为体育页面，广告候选为运动服装，那么运动服装，男性，体育这样的一个标签组合会被激活，在模型训练中，这个特征的值就被设置为1，这个1永远都是确定的，我们把这样的特征叫做静态特征。如何才能让这个特征“动”起来呢？办法也很简单：当这样一个组合被激活时，我们不再用1作为其参与训练的值，而是采用这个组合历史上一段时期的点击率。这样一来，即使是同一个 $f(a,u,c)$ ，在不同的时间点，其所对应的特征取值也是不同的，这样的特征就是动态特征。

采用历史点击率作为动态特征，可以这样来理解：我们最终预测的是某个特定 $(a,u,c)$ 上的点击率，而其某个特征组合 $f(a,u,c)$ 上的点击率，可以认为是关于最终目标的一个弱决策器。通过对这些对应特征组合的弱决策器的融合，我们可以更容易地预测该 $(a,u,c)$ 上的点击率。这样的弱决策器方案有个最大的好处，那就是这些弱决策其本身只需要简单的数据统计就可以得到，而不需要复杂的训练过程。因此，通过这些简单的弱决策器来捕捉模型的动态部分，这样整体的融合模型就可以不必那么快速地更新了。使用动态特征的另一个好处，是模型的参数数目可以大大减少：对于(地域=北京, 广告类型=电商)和(地域=上海, 广告类型=日化)这两个维度组合的具体实例而言，如果采用静态特征方案，需要对这两个实例分配不同的特征编号；而采用动态特征方案时，可以在模型中共享同一个特征参数，而通过不同实例的不同特征取值来分辨它们。这样一来，整体模型的参数个数，就由各种维度组合总的实例数目降低到了维度组合的种类数目，其离线估计和在线计算都会大为简化。

## 位置偏差与CoEC

使用点击率作为动态特征概念上没问题，但在实际操作中会遇到很大困难，特别是当广告主数量不充分的时候。我们假设某广告网络有两个广告位，一个是某网站首页首屏，一个是某小说网站内容页最下端，如果用点击率作为直接的反馈，很显然前几天投在更多地第一个广告位的广告会表现出更好的效果，而这主要是由于位置带来的偏差。请大家再次回顾一下我们在第一章中讲到的广告有效性模型—位置是曝光有效程度的关键因素。

除了广告位位置，还会有其他一些非定向因素对点击率有比较大的影响，主要的有：广告位尺寸、广告位类型(比如门户首页、频道首页、内容页、客户端)、创意类型(比如图片、flash、富媒体)、操作系统、浏览器、日期和时间等。所有这些因素，都与广告决策没有关系，但是对点击率的影响要远远超过其他定向技术。因此，在这些因素上占据优势的广告，其点击率会被严重高估，如果直接用点击率作反馈，也会造成强者愈强的马太效应。

如何去除位置等因素的影响呢？如果我们有财力和人力，可以采用眼球跟踪(Eye Tracking)的设备来评估用户对页面上广告位的关注程度，在后续的统计中据此做归一化。对于一些极关键的页面，比如搜索广告结果页，这样做是值得和可行的。但对于大量显示广告的广告位来说，这样做显然不切实际。工程上一种合理的办法，是将某广告位相当长一段时期内的平均点击数作为其关注程度的近似评估，我们把这一指标叫做期望点击(expected click)。请大家一定注意，与字面直觉不同，期望点击是对展示有效性的一个评估指标。

期望点击要求评估的是在广告质量完全随机的情况下，广告位或其他属性对应的平均点击率。要严格达到这一目的，需要采用随机出广告的策略做小部分流量的测试。这样的方法同样只

能用于搜索广告等因素简单且非常重要的页面。在考虑多个因素共同作用，或广告环境比较复杂的情况下，我们可以采用从数据中近似地学习出期望点击的方法。该方法概念上很简单，我们只用哪些偏差因素作为特征，训练一个点击率模型，这个模型我们称为偏差模型(bias model)。所谓偏差因素，即那些与广告决策无关的特征，这些特征一般来说与 $a$ 无关，我们把偏差模型概念性地表示为 $\mu_{\text{bias}}(u, c) = p_{\text{bias}}(h = 1|u, c)$ 。偏差模型的形式和训练方法都可以与前面所谈的整体点击率模型一致，因此不再赘言。需要注意的是，偏差模型需要用比整体点击率模型更长时间的数据来训练，这样做的目的是希望消除某段时期广告质量带来的影响。

得到了偏差模型以后，我们可以定义下面的归一化的点击率指标：

$$\text{CoEC} = \frac{\sum_i h_i}{\sum_i \mu_{\text{bias}}(u, c)} \quad (4.25)$$

直观上看，这一指标是点击与期望点击的比值，因此我们称之为Click over Expected Click(CoEC)。由于在分母上考虑了位置以及其他因素的偏差对点击率的影响，这一指标可以更准确地表征某部分流量上广告投放的实际点击率水平，也比较适用于点击反馈这样的动态特征。

采用动态特征和偏差模型的工程方案，点击率预测模型训练的流程分三步完成：首先，用较长一段时间的训练数据，只提取偏差特征并训练偏差模型；然后，利用得到的偏差模型，计算所需要维度组合上的CoEC作为动态特征；最后，用所有动态特征和非偏差特征的动态特征训练整体点击率模型，其中用偏差模型的输出作为点击率的先验。我们可以用下图示例性地描述这一过程。

### 常见的偏差特征

前面说到，除了位置，在线广告中还有一些重要的偏差特征，是我们在建模时必须考虑的。对于一些常用且重要的偏差特征，我们简单地介绍如下：

一、广告位位置。位置的影响在搜索广告和显示广告环境下有一定的区别。对于搜索而言，页面布局简单，位置相对稳定，相应地统计也比较充分，因此可以将位置视为离散的变量，分别计算各个位置的EC。而对于显示广告，特别是在广告网络环境下的显示广告而言，位置的可能性非常多，因此不可能对每种不同的位置都作为独立的变量来考虑。比较合理的方法，是找出其重要影响因素，比如广告位中心相对于页面左上角的坐标，用这样的连续变量作为特征来训练偏差模型。

二、广告位尺寸。尺寸的情形与上面说的位置因素很类似：在创意尺寸选择比较少的情况下，可以作为离散变量来处理；而在尺寸选择很多的情况下，也可以用长宽等连续变量来代替。对于搜索广告，由于各创意尺寸一致，这一因素的影响不存在。

三、广告投放延迟。广告完成决策逻辑，并将最终结果返回给用户的整体时间长短，对点击率有着非常大的影响。如果在前端将广告请求发生的时间和最终展示时间都记录下来，可以为点击率预测模型提供一个重要的偏差特征。

三、日期和时间。实际的观测表明，工作日还是周末，对不同类型的广告点击率有着明确的影响，这主要是由于在不同时间用户任务的集中程度不同，对广告的关注也有所区别。时间的因素，即是工作时间还是休闲时间，也有着类似的特性。因此，日期和时间一般来说也是必须要考虑的偏差特征。除了在模型中显式利用，我们往往还要求所有的训练过程都覆盖7天的整数倍的数据，其目的也是为了避免日期带来的偏差。

四、浏览器。浏览器本身并不对广告效果有明确的影响，不过由于各个浏览器上Ad blocker的覆盖程度有较大区别，因此在实际建模中其影响也相当大。

上面列举的几项都是在通用的广告系统中最常见的偏差特征，也是建模时需要首先考虑的，读者需要结合具体的广告产品，按照“去除与广告决策无关的影响因素”这一原则来确定和使用偏差特征。

### 点击反馈的平滑

用点击率或CoEC这样的点击反馈作为动态特征，大量的长尾维度组合对于更准确地预测点击率是有帮助的。但是要利用好这些长尾的维度组合，还需要解决一个问题，就是在统计不足的维度组合上如何稳健地统计这些动态特征：由于这些组合出现的显示数可能比较少，直接计算CTR或者CoEC都可能是相当不准确的。

以CTR为例，公式4.12给出了点击的生成模型，点击率就是这一模型的参数。在知道每次展示点击与否的情况下，可以很容易得出，参数 $\mu$ 的最大似然估计为：

$$\hat{\mu} = \frac{\sum_i h_i}{N} \quad (4.26)$$

其中N为总的展示数。当估计某些数据不足的维度组合上的点击率时，一般的思路是在分子分母上各加一个常量，以起到平滑的作用：

$$\hat{\mu}' = \frac{\alpha + \sum_i h_i}{\gamma + N} \quad (4.27)$$

很显然， $\alpha/\gamma$ 应该等于某更大流量范围内的平均点击率。可是 $\alpha$ 和 $\gamma$ 的绝对数值就没有太直观的方法可以设置。根据第二章中的介绍，我们也可以采用经验贝叶斯的方法来解决这个问题。

在贝叶斯的框架下，我们可以把 $\mu$ 看成随机变量，由于公式4.12是一个二项分布，其参数 $\mu$ 对应的共轭先验是Beta分布，即：

$$p(\mu|\alpha, \beta) = \text{Beta}(\mu|\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \mu^{\alpha-1} (1 - \mu)^{\beta-1} \quad (4.28)$$

超参数 $\alpha$ 和 $\beta$ 其实就对应于公式4.27中的 $\alpha$ 和 $\gamma - \alpha$ 。我们可以采用经验贝叶斯的方法来估计 $\alpha$ 和 $\beta$ 。将公式4.12, 4.28代入\*\*\*给出的一般指数族分布经验贝叶斯解，可以得到解 $\alpha$ 和 $\beta$ 的具体EM算法：

1. E-step:

$$\tilde{\alpha}_k^{\text{old}} = \alpha^{\text{old}} + \sum_{i=1}^{N_k} h_{k,i}, \quad \tilde{\beta}_k^{\text{old}} = \beta^{\text{old}} + (N_k - \sum_{i=1}^{N_k} h_{k,i}) \quad (4.29)$$

2. M-step:

$$\psi(\alpha_k^{\text{new}}) - \psi(\alpha_k^{\text{new}} + \beta_k^{\text{new}}) = \frac{1}{K} \sum_{k=1}^K \psi(\tilde{\alpha}_k^{\text{old}}) - \psi(\tilde{\alpha}_k^{\text{old}} + \tilde{\beta}_k^{\text{old}}) \quad (4.30)$$

$$\psi(\beta_k^{\text{new}}) - \psi(\alpha_k^{\text{new}} + \beta_k^{\text{new}}) = \frac{1}{K} \sum_{k=1}^K \psi(\tilde{\beta}_k^{\text{old}}) - \psi(\tilde{\alpha}_k^{\text{old}} + \tilde{\beta}_k^{\text{old}}) \quad (4.31)$$

其中M-step需要解关于 $\alpha_k^{\text{new}}$ 和 $\beta_k^{\text{new}}$ 的方程组, 因而并不是闭式解, 不过这一方程组并不难用数值方法求解。

#### 4.6.2 点击率预测评测

点击率模型预测的事点击事件出现的概率, 因此可以采用Precision-Recall(PR)曲线或Receive Operating Characteristic(ROC)曲线来评测。这两个区县实际上是对同样一组统计数据不同侧面的表现: 点击率模型是一个对点击事件进行预测的模型, 因此, 对任何一个具体的样本, 存在下面四种情况: 1. 点击行为被预测为点击行为, 其数目极为 $n_1$ ; 2. 点击行为被预测为非点击行为, 其数目计为 $n_2$ ; 3. 非点击行为被预测点击行为, 其数目计为 $n_3$ ; 4. 非点击行为被预测为非点击行为, 其数目计为 $n_4$ 。对于这四个数值, 有两种常见的视角: 一是观察 $\text{recall} = n_1 / (n_1 + n_2)$ 和 $\text{precision} = n_1 / (n_1 + n_3)$ 的关系, 二是观察 $\text{true positive rate} = n_1 / (n_1 + n_2)$ 和 $\text{false positive rate} = n_3 / (n_3 + n_4)$ 的关系。(实际上 $\text{recall}$ 和 $\text{true positive rate}$ 是一样的。)当然, 是否被预测为点击, 是针对某个点击概率的阈值而言的, 因此, 通过取不同的阈值, 我们就可以得到一条 $\text{precision/recall}$ 的曲线, 或者是 $\text{true positive rate/false positive rate}$ 的曲线, 前者即为PR曲线, 而后者就是ROC曲线。为了方便理解, 我们上述的几个基本量直观地表示在下面的图中:

		标注(Ground truth)	
		P	N
决策 (Hypothesis)	P	True Positives( $n_1$ )	False Positives( $n_3$ )
	N	False Negatives( $n_2$ )	True Negatives( $n_4$ )

Fig. 4.3. 点击率模型评测若干统计量

实际的PR曲线可以参见图4.6.2中的示意。一般来说, PR曲线呈下降的趋势, 不过这一点并没有理论上的保证, 实际数据上局部呈上升趋势的PR曲线也很常见。对广告而言, 我们应该更加关注PR曲线的头部, 因为尾部是 $\text{recall}$ 比较高, 也就是很多广告候选都被考虑时的情形, 而实际的投放环境中, 我们只选择排名最好的一个或几个候选。另外一点需要注意的是, PR曲线下面的面积是没有明确的物理意义的, 因此不能作为有价值的指标来衡量。



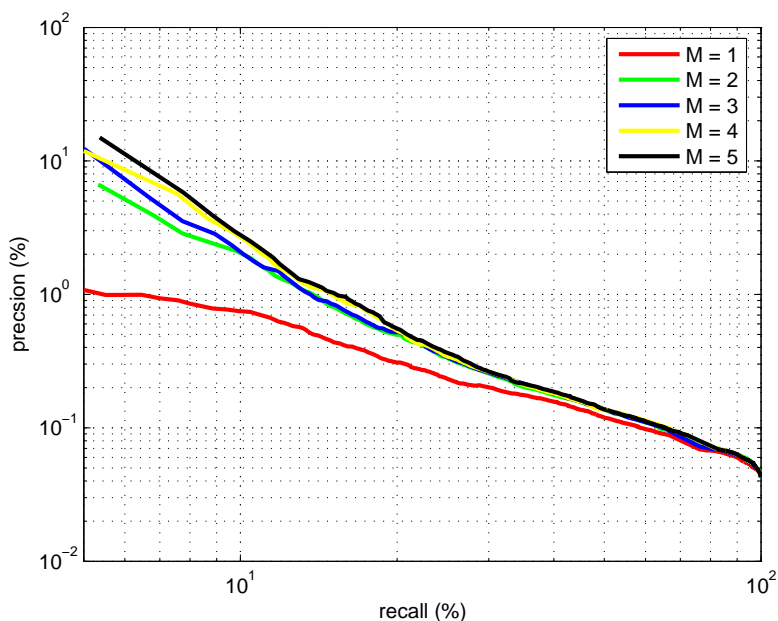


Fig. 4.4. PR曲线示例

实际的ROC曲线可以参见图4.6.2中的示意。一般来说，ROC曲线呈上升的趋势，不过这一点同样没有理论上的保证。与PR曲线不同，ROC曲线下的面积是有着明确的物理意义的，它在一定程度上表征了对click=0和click=1事件估计值排序的正确性。我们把ROC曲线下的面积称为AUC(Area Under Curve)，是评价点击率模型是常用的量化指标。AUC虽然经常被用作点击率模型的质量代表，却有一个非常重要的问题需要引起注意：那就是即使我们只用偏差模型，即对广告排序无直接贡献的模型来预测点击率，AUC往往也处于比随机猜测高得多的水平上，如图4.6.2中所示。因此，模型对广告排序的作用，需要对这两个AUC的差值做评估才能比较公允地加以衡量。

无论是计算ROC曲线还是PR曲线，核心都是要统计上述的 $n_1, n_2, n_3, n_4$ 四组值。严格的统计方法，需要对整个测试集按照模型估算的点击率做排序，不过这样的计算复杂度为 $O(N \log N)$  ( $N$ 为测试集的样本数目)，显然在测试样本量较大时无法实用。因此，我们可以采用近似但对实用来说足够精确的方法，即将整个点击率的取值范围划分成一组区间，并在每个区间上得到一个曲线点。这样的方法示例性代码如下：

```

1  const float INF = 1e10f;
2
3  void curve_acc(vector<float>& scores, int binNum,
4                vector<bool>& labels,
```

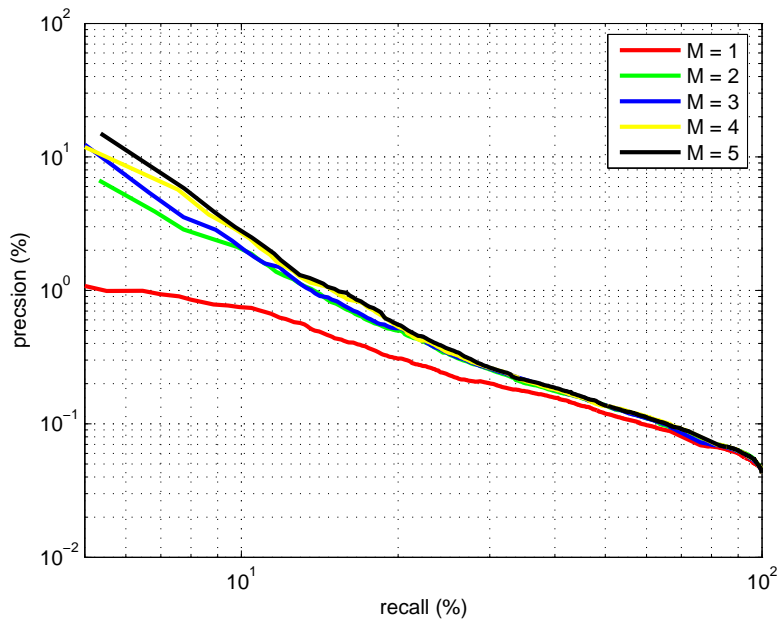


Fig. 4.5. ROC曲线示例

```

5         vector<int>& TPs, vector<int>& FPs
6         vector<int>& TNs, vector<int>& FNs) {
7     assert(scores.size() == labels.size());
8     int sampleNum = (int)scores.size();
9
10    //遍历所有的样本以得到score的最小最大值
11    float minScore = INF, maxScore = -INF;
12    for (int s = 0; s < sampleNum; s++) {
13        if (scores[s] < minScore) minScore = scores[s];
14        if (scores[s] > maxScore) maxScore = scores[s];
15    }
16
17    // 得到各个bin的阈值
18    vector<float> bins;
19    float step = (maxScore - minScore) / binNum;
20    bins.assign(binNum + 1, minScore);
21    for (int b = 0; b < binNum; b++)
22        bins[b + 1] = bins[b] + step;
23

```

```

24 // 再次遍历样本以得到各个bin上的统计值
25 tpRates.resize(binNum, 0);
26 fpRates.resize(binNum, 0);
27 for (int s = 0; s < sampleNum; s++){
28     for (int b = 0; b < binNum; b++){
29         bool positive = (scores[s] > bins[b]);
30         if ( labels[s] && !positive)TPs[b] ++;
31         if ( labels[s] && positive)TPs[b] ++;
32         if (!labels[s] && !positive)FPs[b] ++;
33         if (!labels[s] && positive)FPs[b] ++;
34     }
35 }
36 }

```

此算法需要对数据进行两次遍历，第一次得到点击率取值的范围，第二次得到各个曲线点上的 $n_1, n_2, n_3, n_4$ ，很显然，其计算复杂度为 $O(2N)$ 。另外，这样的算法结构可以非常容易地用两次map/reduce过程来实现，因而特别适合于大规模数据集上的评价。

### 4.6.3 智能频次控制

在第三章中，我们介绍了频次控制的问题。在广告网络环境下，这一问题有相当大的变化。我们考虑在合约式广告的情形下，由于广告主对于位置可以由合约控制，因而可以在某个特定的位置上设定展示频次，这一点在按GD方式售卖的视频前贴片广告中应用最为广泛。但是在广告网络情形下，由于广告主的创意可能出现在各种媒体的各种位置上，而根据4.6.2中的讨论，不同位置的有效展示有相当大的差别。因此，简单设定一个展示数目上的频次来控制用户的接触次数虽然大体也会起到一些作用，但是从本质上说是不太合理的。

在这种情况下，我们需要一个更智能的频次控制方案。最直接的思路，是利用4.6.2中介绍的CoEC概念。由于EC从某种程度上更接近于有效展示数目，我们可以采用EC上的累积计数代替频次来控制用户接触次数。在品牌广告和效果广告两种情况下，智能频次控制的做法也有所不同：在品牌广告中，可以通过EC计数上的直接控制来达到一定用户接触程度的目的，由广告主来直接设定；在效果广告中，则可以将EC的计数，或者频次的计数，作为点击率预测模型的特征直接加入训练，靠点击率模型的作用降低出现频次过高的创意的竞争力。

换句话说，在精细的效果要求下，我们实际上更加认清了频次的本质：它与其他影响点击率的特征是平等的，并且应该放在统一的、数据驱动的计算框架下加以利用。而究竟对某个创意应该将频次控制在几，也不应该是根据经验设定，而是应该放在竞价的环境中自行决定。对于传统广告或者在线合约广告中表现出来的其他类似的经验总结，在竞价广告的环境里，也应该以这种态度去对待。

#### 4.6.4 探索与利用

上面关于点击率预测的讨论中,我们知道需要采取或者模型、或者特征上的手段来更多地捕获动态信息。这也就意味着,对某种类型的 $(a, u, c)$ 组合,如果没有相关历史数据的支持,很难对其合理地估计点击率。由于线上我们总是使用认为最优的策略来投放广告,那些非最优的组合出线机会很少,因而对这部分的估计也就不准确。实际上,无法对特征空间采样构造训练集,是互联网问题区别于其他机器学习问题的重要特点。举例来说,在搜索广告市场中,往往有相当比例的广告商从来得不到展示机会因而退出市场。虽然每一次决策我们都在尽力优化eCPM,很显然这不是系统的最佳宏观状态。

解决这个问题的方法,属于强化学习(Reinforcement Learning)的范畴。直觉的想法,是牺牲一部分流量上最优eCPM的诉求,采用相对随机的策略来采样效果未知的特征空间,这称为探索过程(Exploration);同时根据探索和正常决策的总体流量更有效地预测用户的反馈行为,这称为利用过程(Exploitation)。这样的整体策略,一般称为探索与利用(Explore and Exploit, E&E)。E&E可以形象地类比成玩老虎机是的决策问题:玩家面对老虎机上 $A$ 个有不同期望收益的手柄,需要用尽可能少的筹码探索出收益最高的那个手柄,然后利用这个结果去获取回报。这种简单的 $A$ 中选1的研究问题,也称为Multi-Arm Bandit(MAB)问题。我们先来看看MAB问题的数学描述。

假设有 $A$ 个手柄 $a \in \{1, 2, \dots, A\}$ (这里的手柄可以认为是广告),在每个决策时刻 $t$ (对应于广告展示),我们必须从 $A$ 个手柄中选择一个,而目标是优化许多次决策后的整体收益。每个广告 $a$ 在第 $t$ 次展示的收益计为 $r_t(a)$ ,对于不同的 $t$ ,这些收益是独立同分布的。在 $t$ 时刻,我们用下面的两个量来分别表示该分布的均值 $\langle r(a) \rangle$ 与方差的经验估计(此处先不考虑 $u, c$ 的影响):

$$\begin{aligned}\overline{r_t(a)} &\triangleq \frac{1}{t} \sum_{i=1}^t r_i(a) \\ V_t(a) &\triangleq \frac{1}{t} \sum_{i=1}^t [r_i(a) - \overline{r_t(a)}]^2\end{aligned}\quad (4.32)$$

最优的手柄或广告,定义为期望收益最高的那个:

$$a^* = \arg \max_a \langle r(a) \rangle \quad (4.33)$$

MAB问题有一个简单的基础方法,即总是用比例为 $\epsilon$ 的一小部分流量来做探索,在探索流量上随机选择 $A$ 个广告中的一个;在剩余的 $1 - \epsilon$ 比例的流量上,总是选择经验收益最高的那个广告。这样的基础方法称为 $\epsilon$ -贪婪法。很显然,只要经过足够多次的尝试, $\epsilon$ -贪婪法是一定可以找到最优的那个手柄的。既然如此,还有什么深入研究的必要呢?我们当然是希望能够以更小的代价找到最优手柄。这里的代价,定义为整个过程的回报与一开始就总是选择最优手柄这一策略的回报差值,即我们探索所付出的代价。对于一次选择广告 $a$ 的展示,代价数学上的表达为:

$$\Delta_a \triangleq \langle r(a^*) \rangle - \langle r(a) \rangle \quad (4.34)$$

E&E过程的目标是使得整体的代价(Regret)最低。以 $T_t(a)$ 表示到 $t$ 时刻为止分配给 $a$ 的展示数,则这一代价可以写成:

$$R_t = \sum_{a=1}^A T_t(a) \Delta_a \quad (4.35)$$

假设总共需要进行 $T$ 次展示决策，探索一些系统性的方法，使得我们在对最优广告 $a^*$ 没有先验了解的情形下，以比较低的代价完成这一过程，是这个问题研究的关键。这一问题的思路，需要借鉴到类似于贝叶斯学习的思想，即将估计的不确定性引入解决方案中，我们来看一下一些常见的方法。

## UCB方法

谈到MAB问题经典的思路，不能不介绍Upper Confidence Bound(UCB)这一族的方法。从物理意义上看，此方法的关键，是在每次投放时，不是简单地选择经验上最优的广告，而且考虑到经验估计的不确定性，进而选择估计值有可能达到的上界最大的那个广告。

因此，在每个决策点，UCB的过程主要分成两个步骤：首先根据过去的观测值，利用某种概率模型，计算出每个 $a$ 的期望回报的UCB；然后，简单选择UCB最大的 $a$ 。可以看出，这一算法的关键在于如何计算UCB。文\*中给出的一种UCB计算策略，称为 $\beta$ -UCB策略，是按照下式计算上界：

$$B_{k,s} = \left( \bar{X}_{k,s} + \sqrt{\frac{2V_{k,s} \log(\beta_s^{-1})}{s}} + \frac{16 \log(\beta_s^{-1})}{3s} \right) \wedge 1 \quad (4.36)$$

其中 $\beta_s \triangleq \frac{\beta}{4Ks(s+1)}$ 。相应地，在任意一个时刻 $t$ ，只需要选择令 $B_{k,T_k(t-1)}$ 最大的 $a$ 即可。

$\beta$ -UCB的策略并不对回报的具体参数化模型表达有所假设，而是仅仅通过一阶和二阶的一些统计量来完成策略，因而具有比较好的普适性。这一策略直觉的好处，是我们不可能长时间地选择错误的 $a$ 。

## Contextual Bandit

虽然花了这么大的力气来介绍MAB问题和UCB方法，跟实际的广告问题一对应，我们会发现这离实用还相去甚远。实际广告系统中的主要挑战有两点：首先，我们需要探索的空间是 $(a, u, c)$ 这一组合空间，而不是简单一组广告，这使得探索的复杂程度大大上升。以显示广告系统为例，我们要面临的实际情况是数十万的广告主、数百万的上下文页面、以及数以亿计的用户，即使我们将这些信息按某种层级结构聚合起来，这三者的组合可能性仍然相当庞大，对explore是个挑战。其次，对 $(a, u, c)$ 的某一具体组合，并不像前文假设的那样有一个确定的期望收益，这是由广告问题的高度动态性决定的，对此我们在点击率预测那一部分已经做了深入探讨。

对于需要探索的空间过大的问题，工程上比较常用的思路，是将此空间参数化，在一个维数较低的连续空间中进行探索。这样的E&E问题，可以称为Contextual Bandit 问题。注意这里说的context，不同于上下文定向中提到的context，是指根据 $(a, u, c)$ 组合参数化后的上下文空间位置，虽然术语上有所混淆，我们还是遵循原作者这样表述。

Contextual Bandit的问题，代表性的思路有LinUCB方法[30]。从名字就可以了解到，这一方法是将公式4.32中表达的回报分布由 $a$ 决定，变成由一些环境特征的线性组合决定，也就是说，在某个时刻 $t$ ，我们将某个 $a$ 的期望回报表达成：

$$E(r_{t,a}|x_{t,a}) = f(a, u_t, c_t)^\top \theta_a^* \quad (4.37)$$

可以看出，这样的表达达到了两个目的：首先，将 $(a, u, c)$ 的组合空间，而不仅仅是 $a$ 都纳入了探索的范围以内；其次，用线性组合的连续输出代替了离散的ID值，使得E&E过程可以在如此巨大的空间上实施。在[30]中，这一变换模型被称为disjoint linear model，这里“disjoint”的含义，指的是对于每一个广告 $a$ 适用独立的线性变换参数 $\theta_a^*$ 。细心的读者一定会发现，这样的假设在 $a$ 数量巨大时也会成为阻碍，因此，在实用当中，我们也可以在广告主类型或其他聚合粒度上适用不同的变换参数。

## 4.7 点击价值估计

我们再来看看点击价值估计的问题。一次广告带来的点击在广告主的网站上能够产生多少直接或间接的价值，实际上与广告主的具体业务类型紧密相关。

## 4.8 搜索广告特定领域核心技术

从商业逻辑和产品形态上看，搜索广告可以认为是广告网络的一个特例。它是以上下文查询词为粒度进行受众定向，并按照竞价方式售卖和CPC结算的广告网络。从检索、点击率预测等关键环节的技术上看，搜索广告与显示广告网络并没有本质的区别。不过，由于搜索广告本身的变现能力，即eCPM远远高于显示广告，因此其市场重要程度也就得以彰显。对与搜索广告的一些独特问题和算法的研究，也就受到了非常高度的重视。在本章中，我们将在前一章讨论的竞价广告系统的基础上，着重探讨搜索广告相关的一些特殊问题。

搜索广告与一般广告网络最主要的区别，是上下文信息非常强，因此用户标签的作用受到很大的限制。因此，关于搜索广告的研究，有两个技术上的重点：一是查询词的扩展，即如何对简短的上下文信息做有效的拓展，由于搜索广告的变现水平高，这样的精细加工是值得而且有效的；二是根据用户同一个搜索session内的行为对广告结果的调整，因为围绕同一个目的一组搜索，往往对于更准确地理解用户意图有很大帮助。本章中，我们将围绕这两个重点问题，概要性的介绍搜索广告中的思路与具体技术。

尽可能快速地将用户搜索行为反馈到广告决策中的需求，催生了流式计算平台。这一技术与Hadoop这样的离线分布式计算平台相呼应，可以更有效地完成计算广告中的数据处理任务：对于那些长时期积累的海量用户行为数据，我们可以用Hadoop进行离线的、实时性不强的定期挖掘；对于那些短时用户行为，可以用流计算平台进行半实时加工和快速反馈，以便迅速指导线

上决策。这样的计算分工和系统架构，其实不仅适用于搜索广告，也适用于显示广告网络里的短时受众定向和短时用户行为反馈。

对搜索广告这个产品，不同搜索引擎提供商有不同的称呼，比如Paid Search, Search Ad, Sponsored Search等。这些词汇概念上非常相似，但也略有差别，个人比较倾向于采用“Sponsored Search”这样的说法，而“Paid Search”有时会让读者对是谁付费产生误解。至于“Search Ad”，实际上还应包括搜索引擎中的其他广告形式，比如百度品牌专区，因而并不是本章讨论的狭义的完全竞价的搜索广告网络。

#### 4.8.1 搜索广告系统架构

#### 4.8.2 搜索广告概览

本章中讨论的搜索广告，主要以通用搜索引擎为蓝本。实际上，很多垂直类搜索，特别是电子商务类搜索也有很强的广告变现能力，但产品形态可能会有所不同，大家可以有选择性地与下面讨论的问题来对照。另外，插入在自然搜索结果中的竞价排名模式，也不在我们讨论的范围内。

如下图所示，搜索广告的展示区域，一般来说分为北(North)、东(East)、南(South)三个部分。北侧和东侧的所有位置，构成位置拍卖的标的物集合，但是并不是一定要在这些位置上出广告，这一点与显示广告有的区别。南区的广告，不同的搜索引擎有不同的产品处理方法，有的直接照搬北区广告，有的则直接照搬东区的前几条，在本章中不做讨论。一个搜索广告位置的示例如图4.8.2所示。从这个示例可以看出，搜索广告是一个非常典型的位置竞价问题，就期望点击率而言，北显著高于东区，而同区当中位置越靠上也越高，因此，竞价时位置的排序为 $\{\text{North}_1, \text{North}_2, \dots, \text{East}_1, \text{East}_2, \dots\}$ ，有一点特殊之处是，每次拍卖时的位置数目，是可以跟据广告相关性来动态调整的。

搜索广告的受众定向标签，即是上下文的搜索词。由于搜索词非常强地表征着用户的兴趣，搜索广告可以进行非常精准的定向。相对这样的上下文信息，根据用户历史行为得到的兴趣标签重要性大打折扣，这一方面是因为其信号远不如上下文搜索词强烈，另一方面是因为用户这样强烈兴趣的任务是绝不能被打断的(参见第二章中广告有效性原理部分)。这是搜索广告区别于显示广告网络的最大特点。

既然搜索词的重要性极高，粒度又非常细，如何根据广告主的需求对其关键词进行合理的拓展，找到那些相关而又效果不错的关键词，这对于需求方和供给方来说都有很大意义：需求方需要通过扩展关键词以拿到更多流量；供给方则需要借助此来变现更多流量和提高竞价的激烈程度。因此，查询扩展(Query Expansion)是搜索广告的重要技术。搜索广告的查询扩展，与搜索中的这个问题，有相通之处，又有一些显著的区别。

虽然按照用户历史行为做定向不适用于搜索广告，用户在一个session内的一系列查询，还是会对准确理解用户意图大有帮助。另外，前一章介绍的点击反馈特征，也存在着快速更新的需求。要达到这两个目的，系统上的挑战要大一些，原先那样基于Hadoop的离线挖掘模式就不适用了。正是这一需求的存在，催生了流式计算(Stream Computing)的技术平台，目的是利用用户短时的行为快速、连续地得到一些统计信息，并反馈给线上决策系统。流式计算对于广告中许多





对搜索广告而言，上下文信息即Query的核心作用是不能被弱化的，否则会对相关性和用户反馈有较大负面影响。因此，我们提倡在广告检索阶段，不采用短时行为反馈的到的标签信息来影响Query的检索结果，不过在排序阶段，可以利用短时动态特征来提高那些用户更倾向于选择的结果。

我们以短时动态特征为例，来看看这里需要的实时计算问题。假设我们需要根据广告的展示和点击日志，准实时地统计(geo, advertiser)和(gender, ad category)这两个纬度组合上的CoEC特征，并将这两组特征反馈给线上广告排序系统使用。那么这一计算的过程可以用下面的流程来示意：

#### 4.8.4 流式计算平台

上面短时行为反馈的问题，直接催生了流式计算的技术和平台。流式计算平台的功能，是对在线流入的日志或其他事件信息准实时地进行统计和处理，并将处理的结果输出，以用于线上决策或其他监测。

流式计算平台有若干开源工具可供选择，例如S4和Storm，这两个常用的流式计算工具也有一些不同之处。比如S4不保证每一条送入的事件都保证被处理，而Storm则保证每条事件都被处理，但并不保证仅仅被处理一次。这样的区别也反映了他们在最初设计时面向场景的不同：S4最早用于Yahoo!搜索广告的实时点击反馈，因此吞吐率的要求远远高于对数据一致性的要求；而Storm则主要面向一般的BI或其他数据服务，因此对数据的一致性要求较高，但同时也意味着在极端条件下性能上的损失。我们以Storm为例，了解一下流式计算的基本框架和计算过程。

从上面流式计算支持的逻辑来看，也非常类似于一组Map/reduce任务，不过它与HDFS上的Map/reduce仍然有着本质的不同：HDFS上的Map/reduce是通过分布式的文件系统实现尽可能对计算进行调度，而流式计算仍然是在各台服务器之间调度数据来完成计算。这样的区别，使得它们的适用场景也有着很大的区别：流式计算适用于准实时、快速的数据统计和反馈，但是由于是在调度数据，所以并不适合于海量数据的批量计算；而HDFS上的Map/reduce更适用于数据量非常大，但是计算实时性要求并不太高的情形。实际的数据处理系统中，往往需要两者结合来达到数据量和实时性两方面的要求。

我们以Storm为例来看看流式计算平台上计算的基本流程。

#### 4.8.5 关键词推荐

#### 4.8.6 竞价优化

#### 4.8.7 查询扩展

搜索广告独特的技术中，查询扩展是尤其重要的一项，也是搜索广告研究最多的内容之一。由于这方面具体的技术繁多，我们没办法一一列举，因此先从框架上介绍三种主要思路。

## 基于推荐的方法

如果把用户一个session内的搜索词视为一个有内在联系的整体，我们可以在 $\{\text{session}, \text{query}\}$ 这样的矩阵上通过推荐技术来完成产生相关关键词。这种方法利用的是搜索的日志数据。理论上讲，推荐领域的各种方法思路和方法，都可以用在查询扩展问题上。我们在此以查询扩展的问题为例，介绍一下推荐技术的基本算法和关键点。

给定一组用户 $u = \{1, \dots, M\}$ 和一组关键词 $w = \{1, \dots, N\}$ ，我们可以有一个对应的交互强度矩阵 $\{z_{uw}\} M \times N$ 。如果其中某个用户搜索过某个关键词，则矩阵的相应元素就置为一个相应的交互强度，比如该用户在一段时间内搜索过该词的次数。显然，这个矩阵中会有大多数的单元上是没有历史的搜索行为发生的，但是这种情况下，没有搜过并不意味着用户搜索的可能性为零，而推荐的基本任务，就是根据这个矩阵中已知的元素值，去尽可能预测性地填充那些历史上没有观测到的单元。类似的场景除了搜索，还广泛出现在各种互联网应用当中，比如商品的浏览或购买记录，在线电影的打分记录，都可以抽象出类似的交互强度矩阵以及相应的推荐问题。因此，这个问题在互联网数据挖掘领域得到了充分的重视和广泛的研究。

我们来解释一下为什么可以利用推荐的方法进行关键词扩展。对于某一个关键词 $w = n$ 来说，其原始的交互强度矢量 $(z_{1w}, \dots, z_{Mw})^\top$ ，虽然直觉上讲，我们也可以根据两个关键词对应矢量的相似度来找到近似的关键词，不过由于其元素过于稀疏，这一方法实用中往往并不可行。在经过了推荐算法的平滑以后，这一矢量变成 $(z'_{1w}, \dots, z'_{Mw})^\top$ ，由于推荐算法的平滑作用，矢量中的未知元素也都被填充上相对合理的值，于是就可以稳健地比较关键词的相似度和进行扩展了。

关于推荐算法的综述，大家可以参考×中的介绍。大体上看，推荐算法可以分为基于内存即非参数化的方法，和基于模型的参数化方法。前者大多是用某种维数较低的空间来概要性地刻画交互矩阵，然后根据该空间的生成参数来回复矩阵里未知的值。这种将空间降维的思路，与第四章中介绍的文档主题模型乍看起来很相似，因为后者也是在一个文档与关键词组成的强度矩阵上进行挖掘。不过，这两个问题还是有着比较明确的区别：即在推荐问题中，我们应该把那些未观测到的交互单元视为未知，而在文档主题模型中，合理的方法是认为未在某文档中出现的词交互强度为0。我们以推荐的一种典型算法，即SVD++算法为例，来看一下基于模型的推荐算法的基本思路。

## SVD++

在Netflix举办的推荐算法大赛中，以Yehuda Koren为首的小组获得了头名，并得到了100万美元的大奖。他们采用了一种称为SVD++的算法技术，来预测某个用户对某个电影的评分。

## 基于主题模型的方法

除了利用搜索的日志数据本身，也可以体用一般的文档数据来进行查询词扩展。这类方法实质上就是利用文档主题模型，对某个查询拓展出主题相似的其他查询。这部分的原理和常用模型在3.5中已经作了介绍。

### 基于历史效果的方法

对搜索广告而言，还有一类方法非常重要，那就是利用广告本身的历史eCPM数据来挖掘变现效果较好的相关查询。由于在广告主选择竞价的查询词时，一般来说都会选择多个查询，如果从历史数据中发现，某些查询对某些特定广告主的eCPM较高，按么我们应该将这些效果较好的查询组记录下来，以后当另一个广告主选择了某组查询中的一个时，可以根据这些历史记录，自动地扩展出其他效果较好的查询。

虽然这种方法得到的扩展结果，经常会与前两种方法得到的结果相重合，不过由于这种方法直接使用广告的优化目标，即eCPM来指导查询词扩展，它往往能够成为前两种方法非常重要的补充手段，对于以营收为导向的广告产品来说，有着非常关键的作用。

#### 4.8.8 广告条数个性化

搜索广告的内容本身虽然不宜进行特别深入的个性化，但是在搜索页面中插入的广告条数则存在很大的个性化空间。这样做的基础，是不同用户对于广告，或者相关程度差一些的内容，接受和容忍的程度有着很大的不同。实际上，即使在北美市场这样的用户受教育程度较高的市场上，也至少有三四成的用户不能完全分辨搜索结果和广告。因此，对不同的用户动态调整广告的条目数，可以使得在平均广告数目相同的约束下，整体系统的营收有显著的提高。

广告条数的个性化又是一个带约束优化的问题。这里的约束，是系统在一段时间内整体的广告条数，而优化的目标，则是搜索广告系统的整体营收。



## 广告交易市场核心技术

### 5.1 DSP系统架构

DSP的系统架构如下图所示，其中广告投放的决策流程为：DSP服务器通过RTBS接口拿到广告请求，然后经过与广告网络类似的决策步骤，包括检索和eCPM排序，找到价值最高的广告，并将报价返回给Ad Exchange。与广告网络相比，DSP的广告决策过程有一些难点：一是在eCPM估计时，除了估计CTR，还要估计点击价值，并且由于出价的要求，这一估计要尽可能准确；二是由于DSP是完全面向广告主的产品，广告主量的需求也需要满足，因此还要考虑在线分配的策略。上面讨论的这个决策流程，适用于以套利为目标的DSP，即通过优化算法提升广告主的ROI来赚取更多的利润。也有的DSP产品更接近于透明采买的方式，即广告主按照自己的用户划分和策略完成RTB购买，而DSP收取固定的手续费，这种情况下，对优化的需求就没有那么高。

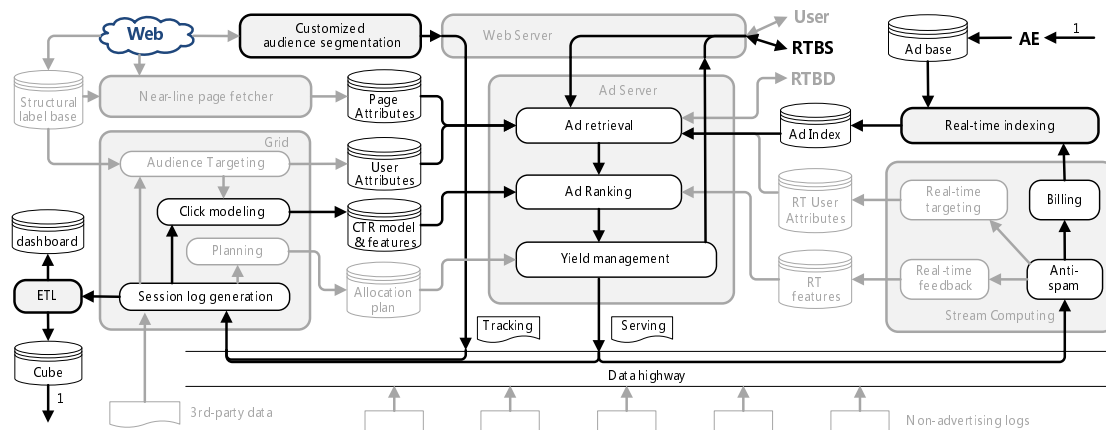


Fig. 5.1. 需求方平台(DSP)系统架构示意

## 5.2 数据管理平台DMP架构

DMP系统的架构可以示意性地表示如下图：通过Data Highway收集各合作媒体或广告主的访问日志，然后把这些日志中的用户信息映射到按照结构化或非结构化的标签体系上。这一架构的核心，是同时对接第三方数据和第一方数据，并根据这些数据对受众群体做灵活的、自定义的划分。虽然这些功能并不直接包括在广告交易环节中，却是数据驱动的在线广告市场中越来越重要的一环。

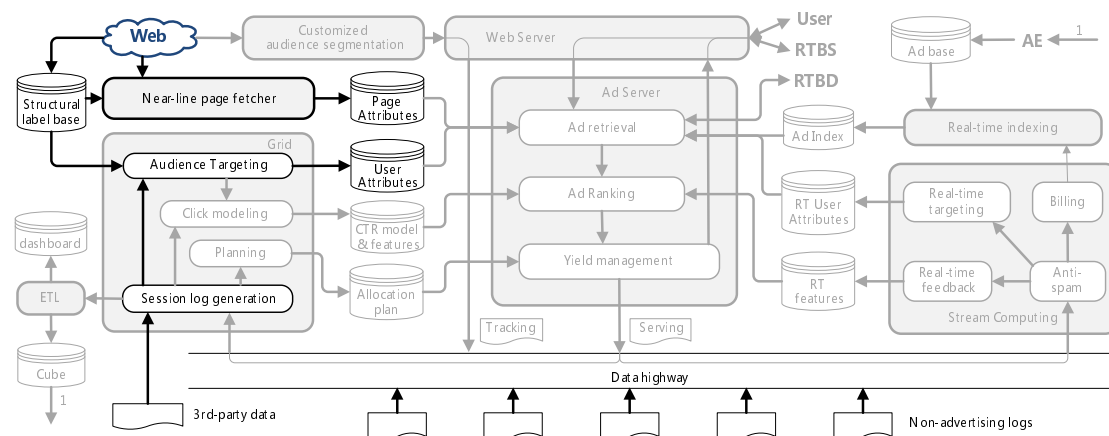


Fig. 5.2. 数据管理平台(DMP)系统架构示意

除了需要用到上面讨论的受众定向技术，DMP还有一个技术问题，就是如何将加工好的用户标签传送给其他标签的购买方，比如某DSP(见第7章)。虽然在图中我们的示意是直接通过在线cache的形式访问，实际上由于DMP与DSP之间跨域且物理上分开的原因，这样的方案并不实际。因此，在DMP中，往往需要提供数据交换(Data Exchange)的产品功能，来进行在线或离线的数据对接。

## 5.3 实时竞价系统架构

我们先来看一下Ad Exchange的系统架构。

上图中给出的是Ad Exchange的骨干架构：当用户访问媒体页面，将广告请求发到Ad Exchange后，Ad Exchange向各个DSP发起询价并完成决策，然后将胜出的DSP返回给媒体页面进行广告投放。从核心概念上看，Ad Exchange既不需要自己的广告索引，也不需要eCPM估计，因而可以用非常简单清晰的架构实现。但是这仅仅是概念，实际产品中，为了支持广告主在不需要定制化用户划分时更方便的图形界面采买，也需要广告索引；而为了实现询价优化(后文有具体介绍)，一定的eCPM估计也是不可少的。下面我们具体介绍一下Ad Exchange用到的相关技术和算法，包括实时竞价、cookie mapping以及带宽限制下的优化等。

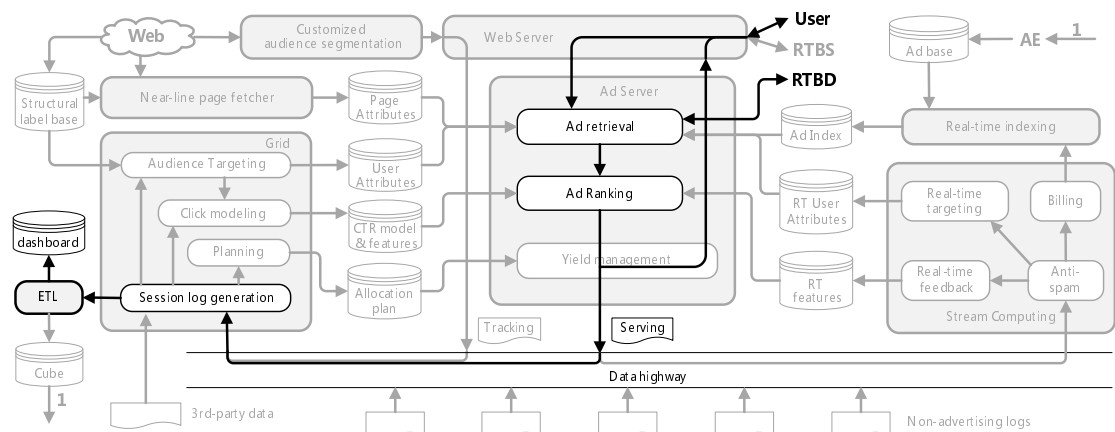


Fig. 5.3. 广告交易平台(Ad Exchange)系统架构示意

5.4 实时竞价流程

实时竞价的整个接口可以分成两部分：线下进行的将供给方与需求方用户ID对应起来的cookie mapping过程，和线上广告请求到来时的竞价和投放过程，如下图所示。下面我们分别介绍一下这两个过程。

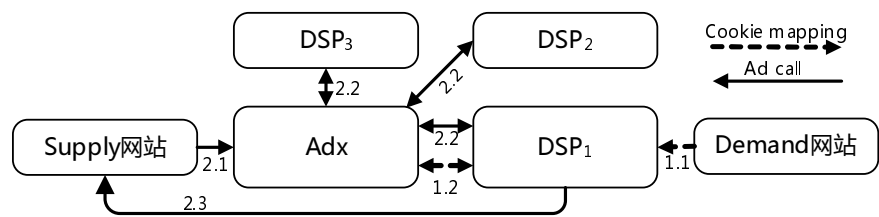


Fig. 5.4. 实时竞价(RTB)接口解析

1. 身份对应(Cookie Mapping): 在RTB应用中，Cookie Mapping一般是由DSP在广告主网站上发起，这样做的原因是一般情况下DSP负责的是加工广告主定制受众标签，因而不需要对所有用户都建立对应关系。这一过程又可以细分为：1.1. 从广告主网站向DSP服务器发起cookie mapping请求，1.2. DSP与Ad Exchange服务器之间通信完成cookie mapping。由于Cookie mapping 这项专门技术的应用范围不仅仅限于RTB，我们将在下文对其做更具体和全面的讨论。

2. 广告请求(Ad Call): RTB的广告请求部分可以分为三个步骤。2.1. 当用户接触到媒体网站的广告位时，前端向Ad Exchange发起广告请求；2.2. Ad Exchange向各DSP传送URL和本域名cookie，发起询价请求。DSP根据预先做好的cookie mapping查出对应的己方cookie，决策是否参与竞价，如果参与，则返回自己的出价。在等待一个固定的时间片后，Ad Exchange选出出价最高的DSP返回给媒体网站；2.3. 媒体网站从胜出的DSP 拿到广告创意并展示。其中2.3, 2.3

两步可以合并为一步，即DSP同时返回出价和广告创意地址，由Ad Exchange 返回给媒体。这样做的好处是减少了一次服务器往返，用户看到的广告延迟也会减少，缺点是Ad Exchange 可以获得DSP 某个广告商的相关受众，因而存在信息泄露风险，不太符合Ad Exchange 中立市场的地位。实际产品中，这两种方式都有采用。

凡事有一利必有一弊。实际竞价的交易方式，虽然给予了广告主最大的流量选择空间，也对系统提出了更高的要求，并且带来了下面的一些实际的问题：一、每次展示都有Ad Exchange服务器与多个DSP服务器的参与，这使得服务器与带宽成本大大增加；二、在询价过程中，Ad Exchange要等待一个约定好的时间片(一般情况下为100ms)，这使得用户看到的广告延迟增加，对CTR有负面影响；三、原理上DSP可以以极低的出价参与竞价，这样虽不能获得流量，却可以低成本得到在媒体网站上的用户行为数据，这里存在着潜在的信息泄露风险。

RTB的接口有两个对接方：在Ad Exchange方实现的部分，我们称为RTBD(RTB for Demand)；在DSP方实现的部分，我们称为RTBS(RTB for Supply)。在各个Ad Exchange 中，RTB 接口的细节和具体参数有很大的不同。显然这对于广告主从不同的Ad Exchange 中统一采买流量是不利的。为了解决这一问题，IAB经过充分市场调研与企业合作，制定了OpenRTB 的接口标准，这一标准涵盖了视频、无线、文字、条幅等多种广告形式下的RTB 问题，并已经为一些Ad Exchange所采用。

## 5.5 Cookie Mapping

在http协议下，cookie是在客户端记录用户身份和信息的主要手段。由于每个服务只能访问在本域名下的cookie，因此不同域名的用户身份不可能直接对应，必须要通过某种技术手段来完成，这样的技术手段称为cookie mapping。Cookie mapping应用的范围很广，除了上面提到的Ad Exchange与DSP之间的身份对应，典型的应用还例如媒体与DMP 之间的身份对应，以及某具有永久用户ID的服务向其他域名提供cookie找回的服务等。

Cookie mapping的情况虽然比较复杂，我们可以重点关注三个问题：由谁发起？在哪里发起？谁保存映射表？最典型的情景有两种：一是涉及到两个域名，即在一个域名的服务上向另一个域名发动的cookie mapping；二是涉及到三个域名，即在一个第三方域名页面上发动的其他两个域名间的cookie mapping。我们分别来讨论这两种情况。

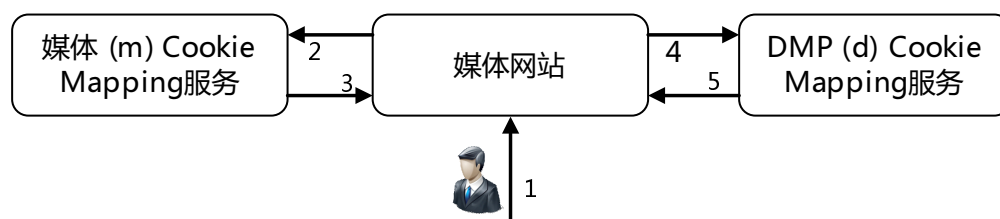


Fig. 5.5. 媒体与DMP间cookie mapping示例



涉及到两个域名的cookie mapping，典型的例子是媒体与DMP之间的身份对应问题。如上图所示，这一过程有5个步骤：1. 用户到达媒体页面；2. 向媒体的cookie mapping 服务请求一段负责此功能的JS代码；3. 媒体的cookie mapping服务返回该段JS 代码；4. 该JS代码判断需要mapping的话，向DMP发起cookie mapping请求，并传送两个参数：DMP的标识，以及媒体方的cookie；5. DMP返回一个 $1 \times 1$ 的beacon，并记录下媒体方cookie与己方cookie的对应关系。考察一下我们关注的三个问题可以发现，这样的cookie mapping是由媒体方在媒体的页面上发动，并由DMP方保存映射关系。这样做有其合理性：媒体需要从DMP获得标签的人群是自己的地访问人群，因此从媒体页面发动；而DMP保存映射，可以比较方便地将自己的用户标签与媒体cookie对应，并传回给媒体。

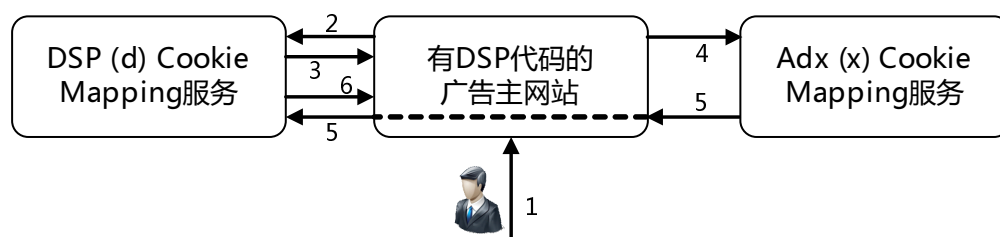


Fig. 5.6. DSP与Ad Exchange间cookie mapping示例

涉及到三个域名的cookie mapping，典型的例子是DSP与Ad Exchange之间的身份对应问题。如上图所示，这一过程有6个步骤：1. 用户访问广告主页面时；2. 选择性加载一个DSP 域名下的iframe；3. DSP判断需要mapping的话，返回包括多个beacon的动态html，此处多个beacon的目的是为了同时与多个Ad Exchange交换cookie；4. 通过其中的某个beacon，向对应的Ad Exchange发送cookie-mapping请求，并带有Ad Exchange标识、DSP标识和DSP cookie三个参数；5. Ad Exchange通过302重定向向DSP返回Ad Exchange 标识以及其域名下的cookie；6. DSP返回一个 $1 \times 1$ 的beacon，并记录下Ad Exchange方cookie与己方cookie的对应关系。这个过程遇上一个过程相比，由于是在第三方的网站上发动mapping，因而较为复杂，需要用到302重定向，不过熟悉前端技术的朋友应该不难理解。仍然考察一下我们关注的三个问题，这次是由DSP在广告主页面上发动mapping，并由DSP保存映射关系。这样做也是符合业务逻辑的：DSP主要需要广告主的人群做深入加工并对这部分人群在Ad Exchange中出价，因此从广告主页面发动；而RTB是cookie的对应由各DSP分别自行完成，比在Ad Exchange 中集中完成显然更加合理，因此这一映射表保存在DSP方。

除了上面两个典型的例子，cookie mapping可能碰到的需求还很多，但大家只要分析清楚关键的那三个问题，并透彻理解上面两种方式，相信很容易举一反三，根据实际需求设计出合理的方案。

## 5.6 询价优化

Ad Exchange中计算虽然不太多，也有一个很重要的问题需要解决，那就是如何在带宽和服务成本的约束下获得更高的eCPM。如果不考虑成本，Ad Exchange的策略非常简单，每次展示都向所有接入的DSP询价即可。可是当DSP数以十计时，服务成本就为上升一个数量级，这显然是无法接受的。因此，Ad Exchange需要解决的一个关键问题，就是如何在带宽或服务成本的约束下，优化整体市场的eCPM水平。要考虑带宽或服务成本的约束，显然就需要对每次展示中询价的DSP数目做精简，因此，这个问题也称为询价优化(Call Out Optimization)。

询价优化有两种典型的思路，一种是工程规则的思路，另一种是将其描述成一个带约束优化优化问题并求解的思路。我们先介绍一下工程规则的思路：考虑到DSP方有相当一部分是按照广告主定制标签来采买流量，因此，这种DSP一般来说只会在自己感兴趣的人群，也就是cookie-mapping映射过的用户群上出价。显然，Ad Exchange是可以先验地知道这一用户群的，因此，对这类DSP中的某个，如果当前Ad Call到达的用户cookie如果没有与其映射过，那么就不需要向该DSP询价。一般来说，这样的规则可以显著降低带宽需求。不过，也有很多的DSP并不是仅仅在广告主用户集合上出价，或者当这样做仍然不能满足带宽成本的要求时，就需要进一步的优化了。

询价优化的问题，由于也是需要在每个Ad Call到来时做决策，因此从框架上非常类似于第三章中的在线分配问题。只不过这里的约束，变成了带宽或服务的成本。由于从商业规则上说，我们不能完全依赖eCPM水平来决定向哪个DSP发起询价，因为这样有可能造成某些DSP完全得不到流量，从而退出市场。因此，实际的询价优化问题的约束，往往设置成各个DSP获得流量比例的一个上限，而这一上限是根据该DSP前面一段时间的花费来决定的。在这样的约束情形下，询价优化的问题可以描述为：

$$\begin{aligned} & \max \sum_{(i,a) \in E} r(a, u_i, c_i) y_{ia} \\ \text{s.t.} \quad & \sum_{i \in \Gamma(a)} \alpha_{ia} y_{ia} \leq d_a, \forall a \in A \end{aligned} \quad (5.1)$$

将此询价优化问题与第三章中的GD问题做对比，可以发现他们在数学上的本质是一样的。

## 广告其他相关技术

在前面几章中，我们以在线广告市场上产品形态发展的过程为主线，对按合约售卖的广告系统、非实时竞价的广告网络、和实时竞价的广告交易市场作了介绍，并对其中与各种产品形态密切相关的计算技术进行了深入讨论。除了这些骨干性的技术以外，在线广告中还有一些非常重要的外围问题，我们在这一章里，将对这些问题展开讨论，目的是为了让大家对实际广告系统的各个环节都有全面而实际的认识。

我们在前面的章节中讨论的所有技术和算法，其核心都是为了优化广告效果。不过前面谈到的广告效果优化思路，基本上集中在受众定向的角度，而在实际的在线广告中，还有一项对广告效果影响巨大的技术，那就是创意优化。创意优化与受众优化性质有所不同，因为创意的改变实际上也改变了广告要表达的诉求。如何在基本的宣传诉求相对稳定的前提下，结合受众定向对创意做调整，这是广告系统不能不考虑的重要问题。对于这方面的方法和思路，我们放在这一章里单独讨论。

与广告效果的度量相关，有两个问题需要介绍，一是如何在流量中去除那些恶意的和非主动的流量，这部分我们称为流量保护。其中去除恶意流量的反作弊问题，由于是一个“道高一尺、魔高一丈”的动态博弈过程，因此并无确定不变的技术和算法，不过也有一些原则和基础方法可以遵循，我们将重点介绍这方面的内容。二是需求方站在自己的利益角度对广告效果的核实性度量，这称为广告监测。这两个问题其实有着相当深入的联系，在今天程序交易和受众定向大量被使用的在线广告市场中，这两个问题在一定程度上可以结合起来考虑，并催生了所谓“广告安全(Ad Safety)”的问题和相应技术。

受众定向和程序交易广告的另一个重要影响，是用户的行为数据存在在不同的广告产品之家被泄露的可能。因此，隐私保护技术与其对立面，即去匿名化技术，从正面或者负面的角度，都与在线广告有着密切的联系。关于隐私保护相关问题，以及其在广告中的可能应用，也是本章涉及到的内容。

本书虽然力求根据在线广告的核心逻辑来展开技术和算法方面的讨论，在实际行文中则是以桌面操作系统的显示广告为主要参照对象。而我们都知，随着社交(Social)、本地(Local)、移动(Mobile)等因素在互联网业务中快速的发展，所谓“SoLoMo”的趋势也越来越显著。在这些

新趋势下的广告系统，虽然业务逻辑的本质没有根本变化，但也带来了一些崭新的产品形态和技术挑战。

## 6.1 创意优化



Fig. 6.1. 品牌性创意和效果性创意

创意对于广告效果的影响无疑是巨大的，然而我们不能把调整创意带来的效果等同于受众定向产生的效果。因为随着创意的改变，实际上广告主通过广告表达的诉求已经发生了变化，其点击行为也就不再与其他创意具有完全的可比性。我们可以通过一个例子来理解这个问题：假如有一个汽车广告主，将一个宣传某款新车概念的品牌性广告创意，变成一个用户填写试驾申请的表单式创意，如图6.2.2所示。毫无疑问的是，后者的点击率会大幅度上升。但是实际上，这两个创意向用户传达的该汽车品牌的形象，却有着相当大的区别：前者的目的是向潜在用户渗透性地宣传品牌的定位，以利于将来长期的用户转化和利润空间；而后者的目的则是短期内的转化效果，但对品牌特质的宣传则有所不足。

因此，我们重点讨论在广告的基本诉求保持相对稳定的前提下，如果对创意做微调以显著提高效果。这样的创意优化，依据的基本原理，是我们在第一章的广告有效性模型中就介绍过的原则：为了提高用户的关注程度，我们需要将向用户推送此广告的主要原因在创意中明确表达出来。

## 6.2 流量保护和效果监测

流量保护(Traffic Protection)这个概念，包括两方面的内容：一是对非主动用户访问行为，主要是爬虫行为的去除；二是反作弊，其对于恶意展示或流量来源的去除。从技术难度上来看，我们主要关注的是后一个问题。反作弊的检测方案，在在线环境和离线环境中都有需求，并且是广告计

价和其他数据统计的基础步骤。另一个在线完成的辅助任务，是为广告主服务的广告监测服务，它为了完成广告的展示和点击计数，当然也需要一定的反作弊处理，因此这两个问题有很多密切的联系，我们在此一并讨论。

### 6.2.1 爬虫流量去除

在广告系统中，并不是所有的广告展示和点击都来自于实际的用户。特别是随着搜索引擎技术的发展，网络爬虫带来的流量往往占相当大的比例。爬虫再打开页面时会触发广告展示，也会在探索新的链接时触发广告点击。爬虫流量的去除比较简单，应为比较规范的搜索引擎会在抓取页面时做声明，而其他一些不做声明的爬虫，也比较容易线下收集其IP，于是用一个黑名单来去除这部分流量即可。

对爬虫流量的过滤，最好在广告服务前端的http服务中直接配置实现，即对于已知的爬虫带来的页面请求，服务器直接返回空，而不做任何广告检索和排序的逻辑，这样对服务器压力的减轻有一定帮助。

### 6.2.2 反作弊

由于广告有若干相关利益方，因此除了那些无恶意的、机器产生的流量，经常也会遇到大量的恶意的展示或点击行为，即作弊行为。作弊的具体手段五花八门，无法一一列举，并且随着反作弊手段的进步而不断进步。要在面对广告作弊时快速找到思路，需要先搞清作弊者的目的。我们知道，广告活动是广告主、媒体与用户之间三方交互的行为，因此广告中的作弊行为主要有两种情况：

一、媒体作弊。媒体是广告活动的主要受益者，因而作弊的动力也最强。展示的作弊相对来说比较困难，因为必然涉及到非正常地触发一些虚假的用户访问。由于大多数广告网络与媒体之间是按照点击的价格来结算，因此点击作弊是最为常见的，当然也存在为了满足CPM订单量的需求而对展示进行作弊的情形。这种作弊的花样繁多，既有将广告展示和点击代码放在非约定位置上或非用户自然行为产生的流量上的方式，也有通过将广告位与内容靠得很近甚至相互重叠来骗取点击的方式。

二、广告主竞争对手作弊。某些广告主的竞争对手，会通过技术手段大量消耗该广告主的预算，达到降低其广告效果的非正常竞争目的。与媒体作弊不同，广告主的竞争对手很难通过控制广告位置或复制代码的方式来作弊，而是通过多次重复点击广告的形式来作弊。由于通过单一IP或cookie来大量点击广告很容易被发现，作弊方往往会通过频繁清除cookie，改变IP，甚至通过木马控制多台用户电脑来达到作弊的目的，当然，这样的作弊手段也被媒体作弊，或者帮助媒体作弊的技术提供商所采用。

针对不同的作弊方式，也有不同的反作弊思路。单一IP或cookie在大量展示或点击的作弊方式是最容易去除的，只需要一定时间段内的展示或点击设定合理的上限，并进而发现那些显著超过上限的IP或cookie加入黑名单即可。对于更复杂的，通过控制多台电脑来产生假点击的作弊方式，点击热点图是一项很有用的反作弊技术：正常的用户点击，在创意上的位置分布往往呈现与

创意关键区域相关的比较自然的分布；而及其产生的用户点击，其分布要么过于均匀，要么过于集中，很容易与自然点击分布相区别。下图给出了一个广告创意正常的点击热点分布于作弊的点击热点分布的示例。通过观察关键位置，关键广告主的点击热点图，往往可以发现作弊行为的一些重要线索。当然，这种方法虽然行之有效，却需要一定程度的人工干预，因此并不适用于在所有流量上普遍采用。



Fig. 6.2. 自然点击热点分布与机器作弊点击热点分布对比

### 6.2.3 效果监测

在线广告区别于线下广告的重要特征，就是其可监测性。不论是CPM还是CPC计费方式，从需求方来看，都存在切实的需求，委托某家第三方监测公司对实际发生的展示或点击数目进行核对，以确保自己的利益。

### 6.2.4 广告安全

为了服务于广告主，将流量监测、作弊监测和品牌保护等一系列任务都一站式地完成，从而保证广告投放过程对于广告主的安全性，市场上产生了广告安全相关的技术公司。

## 6.3 隐私保护和数据安全

熟悉互联网上推荐技术发展的朋友们应该都不会对Netflix百万美元推荐大赛感到陌生。然而，这一推荐大赛只进行了一届就嘎然而止，其背后的原因就不是所有人都关注到的。实际上，这是由于在项相关的用户隐私研究中，Arvind Narayanan等人发现通过一些数据挖掘的技术手段，可以将这些公开的行为数据与Internet Movie Database (IMDb)的数据对应起来，发现他们背后共同的用户。这样的结果让业界感到不安，因为简单地将用户身份等关键信息从数据中去除已经不能满足意思保护的要求。因此，这一比赛也就不再进行了。

上面的例子提醒我们，在充分利用用户信息提高广告效果的同时，绝对不能忽视对用户隐私的尊重和保护。广告中会遇到哪些与用户隐私有关的问题的？我们在这里对其中一些关键点也做一些最基础的讨论，目的是希望大家在应用定向广告技术时对这方面的风险有充分的重视，并可以预先采取一些防控措施。

当然，与隐私保护相对应的技术，就是所谓去匿名化(De-anonymization)，也即通过数据上的一些特征将已经去掉身份标识的两条用户行为记录或其他信息对应起来。这方面的技术如果能够合理利用，对于广告产品主动拓展数据来源，提高定向的精准性会有非常大的帮助，关于这方面的现有技术和广告主可能的正面利用，我们也做简要讨论。

### 6.3.1 用户身份标识

要讨论用户隐私相关问题，我们先来看看在线广告中是如何对用户身份进行跟踪的。在不同的广告形式中，采用的用户身份标识也不尽相同。

对于在Web的环境下投送的在线广告，用户身份的标识可以采用http协议提供的cookie机制来完成。cookie机制在安全性方面有很多好处，比如每个域名下的服务只能访问本域名下的cookie，这实际上是由浏览器保证了不同Web应用之间用户数据的隔离。不过cookie在用户跟踪的有效性方面受到一些限制：首先，用户可以主动清除cookie，于是广告系统对该用户的跟踪就中断了；另外，由于广告网络往往是在其他域名的网站上跟踪用户和投放广告，其种植的cookie是第三方cookie。而对于第三方cookie，浏览器一般有更为严格的限制，有的浏览器甚至会在默认情况下禁止第三方cookie，这也成为行为定向的障碍。随着市场对用户隐私问题越来越关注，W3C还进一步制定了“Do Not Track (DNT)”的标准，用于用户主动向网站要求不要被跟踪，或者不要被网站上的第三方应用所跟踪。cookie的跟踪方式还有一个问题，那就是当某台电脑的用户使用多个浏览器时，其cookie无法直接被统一起来。

在有用户登录信息的广告产品中，用户登陆的身份往往是比浏览器cookie更强的身份标识，而最典型的情形出现在社交广告中。用户登录信息不仅在接续性上远远好于普通的浏览器cookie，还具有能够打通不同的桌面和移动设备的功能。因此，创造网站功能以鼓励用户登陆，是很多从事广告的互联网公司都在努力的方向。当然，也不能认为用户登陆身份的准确性一定好于浏览器cookie。例如，在某些游戏性质较强的产品中，由于一个用户可能创建多个“马甲”参与，会导致其用户标识相当不准确。

在移动互联设备中，特别是应用内，无法使用cookie来跟踪用户。于是，移动广告产品往往利用设备的国际移动设备身份码(International Mobile Equipment Identity, IMEI)号码来唯一地表示设备或用户。不过，如果没有用户等于信息，我们很难将此设备号与某个跟踪到的桌面用户对应起来。对于一些需要用户登陆的移动服务，例如即时通讯工具，则有天然的优势将桌面用户与移动设备关联起来，创造跨屏营销的广告体验。

### 6.3.2 去匿名化技术及应用

实际上, 仅仅将用户的唯一标识从数据中去掉, 是不能达到用户隐私保护的目的的。研究表明, 由于用户的行为信息空间是非常稀疏的, 因此如果我们了解某个用户的一小部分信息, 就有可能在一个匿名化的用户行为数据集上将其找出来, 并从而得知该用户的更多信息。这也就是去匿名化技术。

去匿名化技术在广告系统中有可能有着正面的应用: 如果我们通过计算, 在若干身份标识不一致或者缺失的行为数据集上进行去匿名化, 那么有可能将这几部分数据对应起来, 从而扩展了用户行为的丰富程度, 提高了定向的精准性。如果这一过程能被严格控制在机器执行, 而不允许人直接访问数据, 那么这一过程就是有隐私保证并且有实际价值的。

我们来看看去匿名化技术的现状, 以及广告系统的数据拓展问题对此技术提出的需求。

### 6.3.3 RTB中的数据安全问题

RTB的产生, 使得在线广告市场可以综合利用需求方和供给方的数据来完成更加精准的广告决策。当然, 这样的便利性也是一把双刃剑, 在数据得到更加充分利用的同时, RTB中供给方和需求方对于数据安全性的顾虑和诉求也不能不加以考虑。

我们先来看看供给方的数据安全性问题。由于在RTB过程中, Adx需要向参与竞价的DSP广播每次展示的URL和cookie, 使得DSP理论上存在规模化监听媒体用户行为的可能。如下图所示, 假设有某个恶意的DSP, 对于能够参与竞价的所有广告请求, 都以很低的价格参与竞价, 目的不在于赢得流量, 而在于收集媒体上的用户行为, 这就产生了媒体数据的安全问题, 我们将其称为供给方数据安全(Supply Data Security), 我们在第七章中介绍RTB原理时也曾经提到过这一问题。

供给方的数据安全问题尽管在RTB中确实存在, 但是并不是想象中那样严重。我们可以回顾一下第七章里介绍过的询价优化技术: 由于带宽的限制, 实际上在每次询价时, Adx应该尽可能只向那些最可能赢得竞价的DSP发送询价请求, 而那些以恶意收集数据为目的的DSP, 在理想情况下应该被挡在大部分的询价以外。

再来看看需求方的数据安全性问题。为了表达更加清楚, 我们用下图的例子来说明一下。假设有两个化妆品电商广告主A和B, 两者都通过DSP进行个性化重定向广告采购, 那么他们分别利用RTB的方式接触到了自己的顾客集合。需要注意的是, 这里的顾客集合实际上是广告主的私有数据, 也是特别具有商业价值的数据, 然而, DSP、Adx 和媒体都可能在RTB过程中得到这些顾客集合。如果Adx希望制造更加激烈的竞价环境, 获得更高的利润, 那么它实际上可以将这两个广告主的顾客集合合并在一起, 并生产一个相应的用户标签吸引双方来对此标签竞价。这种做法的实质, 是在竞争对手之间倒卖顾客集合, 并且可以通过比较模糊的标签名字(比如为上面两个化妆品广告主的顾客集合打上“年轻女性”的标签)非常隐蔽地操作。随着竞价激烈程度的增加, 原本属于广告主的利润就向市场其他环节发生了转移, 我们把这个问题称为需求方数据安全性问题。

需求方数据安全性某种意义上比供给方数据安全性更加重要, 因为这决定了广告主是否可以放心地通过RTB进行广告采购。\*\*\*\*\*



## **6.4 SoLoMo时代的广告**

### **6.4.1 移动广告**

移动广告，指的是在各种移动终端上向用户投送的在线广告。

### **6.4.2 社交广告**



---

## References

- [1] William F. Arens, *Contemporary Advertising*, The McGraw-Hill, 2002
- [2] Jean Yves Audibert, Remi Munos, and Csaba Szepesvari, Use of variance estimation in the multi-armed bandit problem, *NIPS'06*, Dec 2006
- [3] J. Bennet and S. Lanning, The Netflix Prize, *KDD Cup and Workshop*, [www.netflixprize.com](http://www.netflixprize.com), 2007.
- [4] Adam Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra, A Maximum Entropy Approach to Natural Language Processing, *Computational Linguistics*, 22(1): 39-71, 1996
- [5] Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006
- [6] David M. Blei, Andrew Y. Ng, and Micheal I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993 – 1022, Jan. 2003
- [7] Stephen Boyd, Lieven Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004
- [8] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*, *Foundations and Trends in Machine Learning*, 3(1): 1 – 122, 2010
- [9] Andrei Z. Broder, David Carmel, Michael Herscovici, Aya Soffer, and Jason Zien, Efficient query evaluation using a two-level retrieval process, *CIKM'03* : 426 – 434, Nov. 2003
- [10] John Canny, GaP: A factor model for discrete data, *SIGIR'04*, July 2004
- [11] Tanmoy Chakraborty, Eyal Even-Dar, Sudipto Guha, Yishay Mansour, and S. Muthukrishnan, Selective call out and Real-time Bidding, *WINE'10*: 145 – 157, Dec, 2010
- [12] Ye Chen, Dmitry Pavlov, and John F. Canny, Large-scale behavioral targeting, *KDD'09*: 426—434, June, 2009
- [13] Peiji Chen, etc., Ad Serving Using a Compact Allocation Plan, \*, 2012.
- [14] Ying Cui, Ruofei Zhang, Wei Li, and Jianchang Mao, Bid landscape forecasting in online ad exchange marketplace, *KDD'11*, Aug, 2011
- [15] Jeffrey Dean and Sanjay Ghemawat, MapReduce: Simplified Data Processing on Large Clusters, *OSDI'04*, Dec, 2004
- [16] A. P. Dempster, N. M. Laird and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *J. Royal Statist. Society (B)*, 39, pp. 1-38, 1977.

- [17] Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz, Internet advertising and the generalized second-price auction: selling billions of dollars worth of keywords, *American Economic Review*, 97:242-259, 2005
- [18] Jon Feldman, Nitish Korula, Vahab Mirrokni, S. Muthukrishnan, and Martin Pal, Online ad assignment with free disposal, *WINE'09*: 374 – 385, Dec, 2009
- [19] Alan F. Gates, etc., Pig: A Structured, High-Level Dataflow System for Hadoop, ...
- [20] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung, The Google File System, *SOSP'03*, Oct, 2003
- [21] Arpita Ghosh, Preston McAfee, Kishore Papineni, and Sergei Vassilvitskii. Representative allocations for guaranteed display advertising. *Manuscript*, 2009.
- [22] Jeff Gill, *Generalized Linear Model: A Unified Approach*, SAGE publications, 2000
- [23] J. Gittins. Bandit processes and dynamic allocation indices, *J. of the Royal Stat. Society*, B(41), 1979
- [24] Thomas Hofmann, Probabilistic latent semantic indexing, *SIGIR'09*: 50-57, 1999
- [25] Bernard J. Jansen and Tracy Mullen, Sponsored search: an overview of the concept, history, and technology, *Int. J. Electronic Business*, 6(2): 114-131, 2008
- [26] Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul, An introduction to variational methods for graphical models, *Machine Learning*, 37: 183 – 233, 1999
- [27] Yehuda Koren, Factorization meets the neighborhood: a multifaceted collaborative filtering model, *KDD'08*: 426 – 434, Aug. 2008
- [28] Gary King, Lanche Zeng, Logistic regression in rare events data, *Political Analysis*, 39(2):137-163, 2001
- [29] Leslie Lamport, Paxos made simple. *SIGACT News*, 32(4):51 – 58, Dec. 2001
- [30] Lihong Li, Wei Chu, John Langford, and Robert E. Schapire, A contextual-bandit approach to personalized news article recommendation, *WWW'10*: 661-670, Apr. 2010
- [31] Chih-Jen Lin, Ruby C.Weng, and S. Sathya Keerthi, Trust Region Newton Method for Large-Scale Logistic Regression, *Journal of Machine Learning Research*, 9:627 – 650, 2008
- [32] David J.C. MacKay, *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, 2003
- [33] Robert Malouf, A comparison of algorithms for maximum entropy parameter estimation, *Proc. 6th Conference on Natural Language Learning* : 1 – 7, 2002
- [34] Harry M. Markowitz, *Portfolio Selection Efficient Diversification of Investments*, John Wiley & Sons, Inc., New York Chapman & Hall, Limited, London, 1959
- [35] Abhilasha Mehta and Scott C. Purvis, Evaluating Advertising Effectiveness Through Advertising Response Modeling (ARM), *1994 Advertising and Consumer Psychology Conference*, May, 1994
- [36] Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. Adwords and generalized online matching. *J. ACM*, 54(5):22, 2007.
- [37] Arvind Narayanan, Vitaly Shmatikov, Robust De-anonymization of Large Sparse Datasets, *Proc. 2008 IEEE Symposium on Security and Privacy*, 111 – 125, 2008.
- [38] Jorge Nocedal, Stephen J. Wright, *Numerical Optimization*, Springer, 1999
- [39] Christopher Olston, Benjamin Reed, Utkarsh Srivastava, Ravi Kumar, and Andrew Tomkins, Pig Latin: A Not-So-Foreign Language for Data Processing, *SIGMOD'08*: 1099-1110, June, 2008

- [40] Adler Perotte, Nicholas Bartlett, Noemie Elhadad, and Frank Wood, Hierarchically Supervised Latent Dirichlet Allocation, *NIPS'11*, December, 2011
- [41] Mark Slee, Aditya Agarwal, and Marc Kwiatkowski, Thrift: Scalable Cross-Language Services Implementation, Facebook, Apr, 2007
- [42] Xiaoyuan Su and Taghi M. Khoshgoftaar, A Survey of Collaborative Filtering Techniques, *Advances in Artificial Intelligence*, Hindawi Publishing, Aug, 2009
- [43] Hal R. Varian, Position auctions. *International Journal of Industrial Organization*, 25:1163 – 1178, 2007
- [44] Erik Vee, Sergei Vassilvitskii, and Jayavel Shanmugasundaram, Optimal online assignment with forecasts. *Proc. of the ACM Conf. on Electronic Commerce*, 2010
- [45] Xuerui Wang, Andrei Broder, Marcus Fontoura, and Vanja Josifovski, A search-based method for forecasting ad impression in contextual advertising, *WWW'09*: 491-500, Apr. 2009
- [46] Steven Euijong Whang and Hector Garcia-Molina, Indexing Boolean Expressions, *VLDB'09*, Aug, 2009
- [47] Matei Zaharia, Mosharaf Chowdhury, Justin Ma, Michael J. Franklin, Scott Shenker, and Ion Stoica, Spark: Cluster Computing with Working Sets.