

# 基于跳过程与统计学习的普适排行榜类产品度量模型

程晨<sup>†</sup> 陈子恒<sup>‡</sup> 郝天泽<sup>§</sup>

摘要

关键字:

---

<sup>†</sup>School of Mathematical Sciences, Peking University, Beijing 100871, China. Email address: [moriartycc@pku.edu.cn](mailto:moriartycc@pku.edu.cn), ID: 1500010714

<sup>‡</sup>School of Mathematical Sciences, Peking University, Beijing 100871, China.

<sup>§</sup>School of Mathematical Sciences, Peking University, Beijing 100871, China.

目录

1 引言 3

2 假设 3

3 无跳过的顺序选择模型——Steam 探索队列 4

3.1 模型概述 & 符号约定 4

3.2 度量模型 5

3.3 统计推断 & 优化算法 5

3.4 实验结果 6

4 带跳过的顺序选择模型——美团外卖 6

4.1 模型修正 & 符号约定 6

4.2 度量模型 & 统计推断 & 优化算法 6

4.3 实验结果 6

5 带跳过的投票模型——豆瓣、知乎推荐 6

5.1 模型修正 & 符号约定 6

5.2 度量模型 & 统计推断 & 优化算法 6

5.3 实验结果 6

6 实际产品评价 6

6.1 两种著名的排行榜产品 6

6.2 不同情形下的实验比较 6

7 由此启发得到的基于优化的排行榜算法 6

8 总结 6

# 1 引言

## 2 假设

在开始阐述我们的模型之前，我们先描述这些模型所依赖的一些基本假设，和所建立的概率模型的空间。

**排行榜** 我们所讨论的排行榜为产品状态集合  $\mathcal{P} = \{1, 2, \dots, N\}$  的一个排序，我们将所有排行榜产品构成的集合记为  $\Omega_P$ ，则  $\forall \omega \in \Omega_P$ ，我们就有一个排行榜

$$\omega = (p_1, p_2, \dots, p_N)$$

**产品固有性质** 为了能够度量排行榜类产品的“优劣”，我们假设产品本身有一些固有性质。如果直接定义一个“好坏”的程度，既难以描述和精确定义，又不适合在现实中使用。因此，我们考虑一些能够准确观察到和用统计方法推断的性质，且所有数据都能够用**用户停留时间** $\tau$ 和**用户决策** $a$ 来确定。对于任意一个产品  $i \in \mathcal{P}$ ，我们假设

- **用户停留时间  $\tau$  的分布。**用户停留时间服从一个仅与  $i$  有关的分布  $f_i$ ，或

$$\tau \sim f_i$$

我们说明这样的假设是合理的。尽管对于每一个不同的用户  $u$ ，其所对应的  $f_i^u$  可能不同（我们认为一个确定的用户使用同一份产品所用的时间也是一个随机变量）。但考虑所有用户集合  $\mathcal{U}$ ，以及每一个用户  $u \in \mathcal{U}$  被抽样的概率  $p_u$ ，我们就能得到

$$f_i = \sum_{u \in \mathcal{U}} f_i^u p_u$$

- **用户的决策  $a$  的可能性。**对于排行榜类产品，用户本身可能有多种感官或心理上的体验，但这些体验并不能被开发者观察和收集到，因此并不是我们所关心的。我们关心的是用户所做的决策，而这些决策将被用来衡量排行榜类产品的好坏。具体来说，根据我们研究的不同排行榜类模型，我们将用户的决策分为
  - **接受 (Accepted)**：对于选择型的排行榜，接受表示用户阅读该产品后选择了该产品。
  - **放弃 (Rejected)**：对于选择型的排行榜，放弃表示用户阅读该产品后放弃了该产品。
  - **跳过 (Skipped)**：对于选择、投票型的排行榜，跳过表示用户没有阅读该产品就直接跳过。
  - **结束 (Terminated)**：对于投票型的排行榜，结束表示用户结束了他的阅读过程。
  - **投票 (Voted)**：对于投票型的排行榜，投票表示用户阅读该产品后进行的决策，具体来说可以有三种
    - \* **支持 (Up vote)**
    - \* **反对 (Down vote)**
    - \* **不表态 (No vote)**

**用户** 对于用户来说，我们关心他们的停留时间  $\tau$  和决策  $a$ 。相应的，我们做出这些假设

- $\tau$  为指数分布。即  $f_i = \text{Exp}(\lambda_i)$ 。这样的假设是常用且合理的。
- 用户群体对于接受、放弃、跳过、结束、支持、反对和不表态分别有基于产品的固有概率  $a_i, r_i, s_i, t_i, u_i, d_i, n_i$ 。
- 产品独立性。每个用户使用当前产品时不会受到之前产品的影响，且不考虑个人喜好的因素，即其所做决策的概率完全有产品本身决定。

### 3 无跳过的顺序选择模型——Steam 探索队列

#### 3.1 模型概述 & 符号约定

我们首先考虑一种最为简单，受约束较多的排行榜类推荐模型——无跳过的顺序选择模型。在这个模型中，每一个用户必须依照排行榜的顺序依次阅读有关每一个产品的信息，不能跳过，且必须作出一个选择（我们假设如果到了排行榜队列底端就必须选择最后一个产品）。这样的模型是基于著名的游戏品台 Steam 中的探索队列 (1) 功能简化得到的。



图 1: Steam 探索队列

（请在这里添加一段对于 Steam 探索队列的描述）

具体来说，我们给出无跳过的顺序选择模型的定义。

**定义 3.1.（无跳过的顺序选择模型）** 该模型是一个建立在状态空间为  $\mathcal{P} \cup \{A\}$  上的跳过程。其中  $A$  表示接受状态。其中状态  $i$  处的指数闹钟参数为  $\lambda_i$ ，且其嵌入链的转移概率为

$$\begin{aligned} a_{p_N} &= 1 \\ P_{p_i, A} &= a_{p_i}, \quad 1 \leq i \leq N \\ P_{p_i, p_{i+1}} &= r_{p_i} = 1 - a_{p_i}, \quad 1 \leq i \leq N-1 \end{aligned} \quad (1)$$

相应地，该跳过程的速率矩阵可以描述为

$$\begin{aligned} Q_{p_i, A} &= \lambda_{p_i} a_{p_i}, \quad 1 \leq i \leq N \\ Q_{p_i, p_{i+1}} &= \lambda_{p_i} r_{p_i} = \lambda_{p_i} (1 - a_{p_i}), \quad 1 \leq i \leq N-1 \end{aligned} \quad (2)$$

在图 (2) 中展示了具体的模型。该模型可表述为  $\mathfrak{A}(\lambda, \mathbf{a}, \omega)$ 。

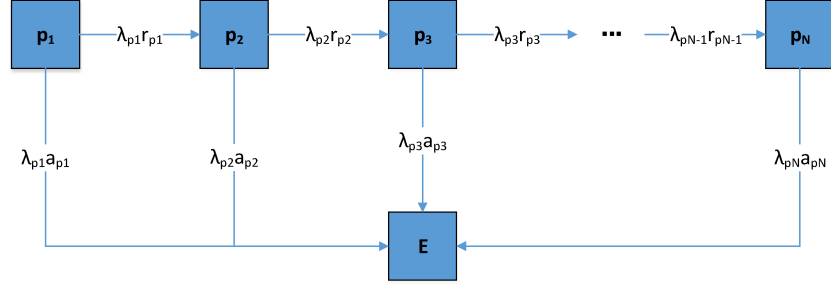


图 2: 无跳过的顺序选择模型结构

(请在这里添加一段对于该模型的感性描述)

### 3.2 度量模型

在这样的模型中，我们有理由设置用户找到其接受的产品的时间  $\tau_E$  为衡量产品优劣的标准。时间越短，那么产品就越有效。而  $\tau_E$  在对应的随机过程中即为到达状态  $E$  的首达时，其分布函数可以表示为

$$F_{\tau_E}(t) = e_{p_1}^T \exp(\mathbf{Q}t) e_E, \quad e_i \in \mathbb{R}^{N \times 1}, i \in \mathcal{P}$$

其方差和期望分别为 (约定  $a_{p_0} = 1$ )

$$\begin{aligned} \mathbb{E}\tau_E &= \sum_{k=1}^N \frac{\prod_{j=0}^{k-1} a_{p_j}}{\lambda_{p_i}} \\ \text{Var}\tau_E &= \sum_{k=1}^N \left( \frac{\prod_{j=0}^{k-1} a_{p_j}}{\lambda_{p_i}} \right)^2 \end{aligned} \quad (3)$$

证明请参见附录。为了综合考虑期望和方差，以及展现出用户对于长队列的不满程度，我们将衡量函数定义为

$$\text{Score}_{\mathfrak{A}(\lambda, \mathbf{a}, \omega)}(x, \alpha) = \sum_{k=1}^N \left( \frac{\prod_{j=0}^{k-1} a_{p_j}}{\lambda_{p_i}} x^k \right)^\alpha, \quad x \in [0, \infty), \alpha \in [1, 2] \quad (4)$$

(请在这里添加一段对于此衡量函数的定性描述，以及合理性分析)

当然，由于参数  $\lambda, \mathbf{a}$  是隐含的，我们需要通过统计推断的方法得到它们，即通过数据得到推断的参数  $\hat{\lambda}, \hat{\mathbf{a}}$ ，然后得到我们的衡量函数

$$\text{Score}_{\mathfrak{A}(\hat{\lambda}, \hat{\mathbf{a}}, \omega)}(x, \alpha)$$

且  $\text{Score}_{\mathfrak{A}(\hat{\lambda}, \hat{\mathbf{a}}, \omega)}(x, \alpha)$  越小，则排行榜产品越好。

### 3.3 统计推断 & 优化算法

对于参数  $\lambda, \mathbf{a}$  的推断我们使用极大似然估计，即假设第  $m$  个用户接受了第  $p_{k_m}$  个产品，前  $k_m$  个产品的浏览时间分别为  $t_{m,i}, 1 \leq i \leq k_m$ ，总共有  $M$  个用户数据被收集，那么似然函数为

$$\mathcal{L}(\lambda, \mathbf{a}, \omega) = \sum_{m=1}^M \ln p_k(\lambda, \mathbf{a}) = \sum_{m=1}^M \ln \left( \left( \prod_{i=1}^{k_m-1} (1 - a_{p_i}) \right) a_{p_{k_m}} \left( \prod_{i=1}^{k_m} \lambda_{p_i} \right) \exp \left( - \sum_{i=1}^{k_m} \lambda_i t_{m,i} \right) \right) \quad (5)$$

容易看出此对数似然函数为凸函数，我们使用成熟的凸优化工具（CVX toolbox）可以得到极大似然估计

$$\left(\hat{\lambda}, \hat{a}\right) = \underset{(\lambda, a)}{\operatorname{argmin}} \mathcal{L}(\lambda, a, \omega)$$

利用推断得到的参数就可以计算衡量函数了。

### 3.4 实验结果

## 4 带跳过的顺序选择模型——美团外卖

### 4.1 模型修正 & 符号约定

### 4.2 度量模型 & 统计推断 & 优化算法

### 4.3 实验结果

## 5 带跳过的投票模型——豆瓣、知乎推荐

### 5.1 模型修正 & 符号约定

### 5.2 度量模型 & 统计推断 & 优化算法

### 5.3 实验结果

## 6 实际产品评价

### 6.1 两种著名的排行榜产品

### 6.2 不同情形下的实验比较

## 7 由此启发得到的基于优化的排行榜算法

## 8 总结