# Эффективное обучение кликовых моделей

Шугаепов Ильнур

научный руководитель: PhD И.Е. Марков

СПб АУ НОЦНТ РАН

13 июня 2017 г.

# Кликовые модели

Кликовые модели описывают поведение пользователя на SERP[1] с точки зрения последовательности наблюдаемых и скрытых случайных величин.

**Определение**

*Кликовая модель $M$, обученная на множестве сессий $\mathcal{S}$ алгоритмом $\mathcal{A}$, есть пара $\langle Q, I \rangle$, где $Q$ - набор запросо-зависимых параметров, $I$ - набор запросо-независимых параметров.*

[1] Search Engine Result Page

# Кликовые модели. Пример

Position-Based Model (PBM):

- Случайные величины
  - $E_r = \mathcal{I}(\text{пользователь изучил сниппет ранга } r)$
  - $A_{uq} = \mathcal{I}(\text{пользователь привлечен } u, \text{ при запросе } q)$
- Параметры

$$P(E_r = 1) = \gamma_r$$
$$P(A_{uq} = 1) = \alpha_{uq}$$

# Обучение кликовых моделей

*Обучение* есть процесс вычисления параметров модели по *click log*.

Методы обучения кликовых моделей:

- Метод максимального правдоподобия (MLE) (все случайные величины наблюдаемы)
- EM-алгоритм (есть скрытые случайные переменные)

# Обучение кликовых моделей

Кликовые модели, в описании которых участвуют скрытые случайные переменные, позволяют описать более сложные сценарии поведения пользователя на SERP.

# EM-алгоритм

**Недостатки:**

- Весь набор параметров модели и *click log* зачастую не помещается целиком в оперативной памяти современного компьютера;
- Итеративное вычисление параметров является довольно медленным.

# Цели и задачи

**Цель:** Сделать существующий подход к EM-алгоритму обучения кликовых моделей более эффективным

**Задачи:**

1. Распределить параметры кликовой модели и данные о поисковых сессиях между узлами кластера таким образом, чтобы обеспечить корректность работы EM-алгоритма
2. Сделать обучение в рамках узла кластера более эффективным
3. Уменьшить насколько возможно комуникационную сложность распределенного алгоритма обучения

# Распределенная кликовая модель

**Определение**

*Пусть $\mathcal{Q} = \{q_1, q_2, \ldots, q_l\}$ - есть множество уникальных поисковых запросов из множества $\mathcal{S}$. Тогда распределенная кликовая модель есть вектор $\mathcal{M} = (M_{q_1}, M_{q_2}, \ldots, M_{q_l})$, где $M_q$ - кликовая модель обученная на $\mathcal{S}_q$ алгоритмом $\mathcal{A}$, и $\mathcal{S}_q$ - множество поисковых сессий инициированных запросом $q$.*
*И $\forall i,j \ (i \neq j \Rightarrow Q_{q_i} \cap Q_{q_j} = \emptyset)$.*

# EM-алгоритм. MapReduce

Подготовка данных:
Представление $\mathcal{S}$ в виде $(\mathcal{S}_{q_1}, \mathcal{S}_{q_2}, \ldots, \mathcal{S}_{q_l})$ .
Общая схема $(t+1)$-ой итерации алгоритма $\mathcal{A}_{MR}$:

Fit (Map)

$$(M_q^{(t)}, \mathcal{S}_q) \quad \underset{\mathcal{A}}{\rightarrow} \quad M_q'^{(t+1)}$$

Extract (Map)

$$M_q'^{(t+1)} \quad \rightarrow \quad I_q'^{(t+1)}$$

Merge (Reduce)

$$(I_{q_1}'^{(t+1)}, I_{q_2}'^{(t+1)}, \ldots, I_{q_l}'^{(t+1)}) \quad \rightarrow \quad I^{(t+1)}$$

Update (Map)

$$(M_q'^{(t+1)}, I^{(t+1)}) \quad \rightarrow \quad \langle Q_q^{(t+1)}, I^{(t+1)} \rangle =: M_q^{(t+1)}$$

# Векторно-матричное представление PBM

Обучение в рамках узла кластера можно сделать более эффективным если представить EM-алгоритм в виде операций над матрицами.

# Векторно-матричное представление PBM

**Предположение**

$$\forall q \in \mathcal{Q}\, \forall r = \overline{1,n}\, \forall s, s' \in \mathcal{S}_q \left( u_r^{(s)} = u_r^{(s')} \right)$$

Предположения не верно для примерно $15\%$ уникальных запросов в Yandex Relevance Prediction Challenge [2].

[2] https://academy.yandex.ru/events/data_analysis/relpred2011/

# Векторно-матричное представление PBM

- Получено представление EM-алгоритм для PBM в виде операций над матрицами как в частном так и в общем случаях.
- EM-алгоритм для частного случая обладает меньшей пространственной и временной сложностью.

# Агрегация по рангу

Агрегация по рангу:
Пусть $\mathcal{S} := \{s_1, s_2, \ldots, s_T\}$, тогда построим множества

$$\mathcal{R}_r := \left\{ \left( q_1, u_r^{(s_1)}, c_r^{(s_1)} \right), \left( q_2, u_r^{(s_2)}, c_r^{(s_2)} \right), \ldots, \left( q_T, u_r^{(s_T)}, c_r^{(s_T)} \right) \right\},$$

$r = 1, 2, \ldots, n,$

где $n$ - максимальный ранг.

# Агрегация по рангу

Для PBM, UBM[3] верно, что:

**Теорема**

1. *Модель $\mathcal{M} = (M_1, M_2, \ldots, M_n)$ - есть допустимая распределенная модель;*
2. *Построение параметров модели $M_r^{(t+1)}$ алгоритмом $\mathcal{A}$ зависит только от $(M_r^{(t)}, \mathcal{R}_r)$.*

Процесс обучения может быть представлен в виде MapReduce фаз.

[3] A. Chuklin, I.Markov and M. de Rijke, Click Models for Web Search, 2015

# Результаты работы

- Реализована Hadoop MapReduce версия EM-алгоритма обучения для PBM,UBM,CCM,DBN[4]
  `https://bitbucket.org/Sh_Nur/pyclick2/branch/hadoop`
- Получено векторно-матричное представление формул для PBM
- Получен альтернативный подход к распределенному обучению PBM, UBM, не требующий обмена данными между узлами кластера
- Получено векторно-матричное представление EM-алгоритма для PBM,UBM в рамках альтернативного подхода

[4] A. Chuklin, I.Markov and M. de Rijke, Click Models for Web Search, 2015

Спасибо за внимание!