

# Бандиты в Query Selection

Шестаков Владимир

## Аннотация

В работе исследуются способы решения задачи Query Optimization. Рассматриваются конкретные решения, использующие метод многоруких бандитов. Среди таких решений выделяется одно конкретное с названием Bao [1], использующее семплирование Томпсона. Данное решение требует проверки на практике, а также исследования на возможность доработки.

**Keywords:** Query Optimization, Multiarmed Bandits, Thompson sampling

## 1 Введение

В современном мире требуются новые инструменты для эффективной обработки данных. В основе подобных систем лежат базы данных, но с увеличением сложности возникает задача оптимизации запросов. Методы, использующие статические правила и эвристики, оказываются не способными выдержать динамические изменения. Современные подходы, использующие машинное обучение, сталкиваются с рядом ограничений: длительное время обучения; неспособность адаптации к изменениям в данных; проблема "хвостовых" случаев, связанная с тем, что запросы исполняются быстро в среднем, но даже редкие случаи длительного исполнения неприемлемы в реальном мире. Таким образом, многие методы становятся непрактичными для их использования.

Среди различных методов выделяется Bao (Bandit Optimizer) [1] — один из способов оптимизации запросов, который успешно преодолевает указанные ограничения. В отличие от других подходов, Bao не заменяет традиционный оптимизатор, а дополняет его, работая как надстройка над уже существующим. Основная идея, заложенная в Bao, это выделение нескольких различных подсказок — изменений в текущем запросе, — которые оптимизатор может применить. Некоторые конечные подмножества таких подсказок используются как "руки" в методе многорукого бандита. Таким образом, сформулировав задачу как контекстного многорукого бандита, Bao использует семплирование Томпсона для нахождения лучшего набора подсказок среди предложенных.

В статье будет воспроизведена работа Bao, проверена его эффективность на тестовых данных и предложены возможные улучшения. Цель работы — показать, на-

сколько различные стратегии решения задачи контекстного многорукого бандита эффективны для оптимизации запросов.

## 2 Постановка задачи

Для начала введём несколько определений, относящихся к самой задаче Query Selection — выбор оптимальной стратегии для определённого запроса.

Каждый набор подсказок является функцией, сопоставляющей запросу  $q \in Q$ , где  $Q$  — множество всевозможных запросов, дереву плана запроса  $t \in T$ , где  $T$  — все деревья плана запроса, какие только могут быть. Обозначим соответствующий набор подсказок как  $HSet_i$ . Все такие наборы подсказок должны содержаться в одном определённом семействе  $F$ , фиксированном изначально. Обозначим метрику, по которой будет происходить оптимизация, как функцию  $P : T \rightarrow R$ , т.е. действующую из множества деревьев плана запроса в множество действительных чисел. Также обозначим функцию выбора набора подсказок как  $B : Q \rightarrow F$ . Нам нужно минимизировать значение функции, так что определим потерю  $R_q$  как  $(P(B(q)(q)) - \min_i P(HSet_i))^2$ .

Соответственно, получаем уже выпуклую функцию, которую проще оптимизировать. Все обозначения, что были в исходной статье, здесь повторяются с таким же смыслом.

Теперь определим то, как выбирается стратегия. Пусть дано множество моделей  $M$ , предсказывающих по запросу, какую из рук дёрнуть, т.е. на запрос  $q$  выдают определённый набор подсказок (условно будем считать его числом от 1 до  $|F|$  и сопоставлять числу  $i$  набор подсказок  $HSet_i$ ). Проверив выбранный план запроса, значения параметров модели обновляются. Среди таких моделей рассмотрим несколько фиксированных, соответствующих конкретным стратегиям многорукого контекстного бандита. Заодно получаем ответ, каким образом получается задача о контекстном бандите: контекстом является сам запрос, множество стратегий (наборы подсказок) служат руками такого бандита.

Одним из кандидатов в такие модели, предложенный в статье, является стратегия, основанная на семплировании Томпсона. Для этого внутри стратегии используется модель, сопоставляющая деревьям плана запроса значения метрики, оптимизируемой в задаче, а затем среди таких значений выбирается минимальное. Как конкретно устроена данная стратегия можно прочитать в исходной статье[1]. Разберём кроме этого несколько иных стратегий.

TODO: Добавить другие стратегии и их описание.

## Список литературы

- [1] Ryan Marcus, Parimarjan Negi, Hongzi Mao, Nesime Tatbul, Mohammad Alizadeh, and Tim Kraska. Bao: Making learned query optimization practical. *Proceedings of the 2021 International Conference on Management of Data*, June 2021.