УДК 004.65

ОПТИМИЗАЦИЯ ГИБРИДНЫХ АНАЛИТИЧЕСКИХ ЗАПРОСОВ НА ОСНОВЕ ВСТРОЕННЫХ МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ В ПОЛИМОРФНЫХ СУБД

Андрей Владиславович Веревкин
студент
mr_and_55@mail.ru
Лариса Ивановна Никонорова
кандидат сельскохозяйственных наук, доцент
lenaniknrva@rambler.ru
Мичуринский государственный аграрный университет
г. Мичуринск, Россия

Аннотация. В статье рассматривается подход к выполнению аналитических запросов, сочетающих реляционные операции SQL и предиктивную аналитику машинного обучения (МО) непосредственно в системе управления базами данных. Предлагается модель глубокой интеграции ML-моделей в механизм выполнения запросов СУБД за счёт введения полиморфного оператора, учитываемого оптимизатором. Предложенный подход позволяет исключить накладные расходы на передачу данных и повысить эффективность планирования запросов.

Ключевые слова: машинное обучение в СУБД, гибридные запросы, полиморфная СУБД, HTAP, оптимизация запросов, PostgreSQL.

Экспоненциальный рост объёмов данных и распространение задач аналитики в реальном времени обусловили необходимость более тесной интеграции средств хранения данных и машинного обучения. Одним из первых и наиболее известных примеров такой интеграции является библиотека MADlib, реализующая алгоритмы аналитики непосредственно на уровне СУБД [1]. Традиционная архитектура, основанная на экспорте данных из СУБД во внешние ML-фреймворки и последующем импорте результатов, приводит к значительным задержкам, усложняет сопровождение и создаёт дополнительные риски безопасности. Данный разрыв между управлением данными и аналитикой становится критическим для интерактивных и транзакционно-аналитических систем [2-3].

В условиях развития отечественных цифровых платформ и ориентации на open-source-решения особую роль играют СУБД семейства PostgreSQL. Однако их стандартные механизмы не обеспечивают эффективной поддержки сложных аналитических сценариев, совмещающих SQL-агрегации и предиктивные вычисления. Актуальность исследований, направлена на разработку архитектур глубокой интеграции машинного обучения в СУБД.

Целью работы является анализ и практическая оценка подхода к оптимизации гибридных SQL+ML-запросов на основе встроенных моделей машинного обучения. Научная новизна исследования заключается в формализации концепции полиморфного оператора и экспериментальном подтверждении его эффективности.

Существующие решения интеграции ML и СУБД можно условно разделить на несколько классов. Наиболее распространённым является внешний ETL/ELT-подход, при котором данные выгружаются из базы данных и обрабатываются во внешних фреймворках. Несмотря на гибкость, данный подход характеризуется высокими накладными расходами и слабой пригодностью для оперативной аналитики.

Более тесная интеграция достигается за счёт использования пользовательских функций (UDF), реализованных, например, в библиотеке

MADlib. В этом случае ML-алгоритмы вызываются из SQL, однако оптимизатор запросов не обладает информацией о внутренней структуре и стоимости таких функций, что ограничивает возможности оптимизации.

Современные СУБД всё чаще реализуют нативные механизмы машинного обучения, при которых модель становится объектом базы данных и управляется декларативными средствами SQL [4]. Тем не менее даже в этом случае ML-операции часто рассматриваются как изолированные вычислительные шаги и не участвуют полноценно в построении оптимального плана выполнения запроса. Предлагается рассматривать обученную модель машинного обучения как особый тип предиктивного индекса, интегрированного в план выполнения SQL-запроса. Для этого вводится полиморфный оператор, который принимает на вход поток кортежей, применяет к ним загруженную в память ML-модель и формирует расширенный выходной поток с результатами предсказаний.

Ключевым отличием предлагаемого подхода является то, что полиморфный оператор учитывается оптимизатором запросов наравне с традиционными реляционными операциями. Его стоимость оценивается на основе метаданных модели, что позволяет оптимизатору изменять порядок выполнения операций, в том числе применять предиктивные фильтры до выполнения ресурсоёмких соединений и агрегаций. Таким образом, императивная последовательность действий «выборка — экспорт — вычисление — импорт» заменяется единым декларативным планом выполнения. Это снижает накладные расходы, упрощает разработку и повышает управляемость аналитического контура.

Для проверки эффективности предложенной модели был реализован экспериментальный прототип на базе PostgreSQL с расширением механизма выполнения запросов. В качестве тестового набора данных использовался модифицированный бенчмарк TPC-H, дополненный синтетическими признаками для задач классификации и регрессии. Сравнение проводилось между классическим ETL-подходом, решением на основе UDF (MADlib) и предлагаемой интегрированной моделью. Рассматривались сценарии

предиктивной фильтрации и корректировки агрегированных показателей на основе прогнозных значений. Основными метриками являлись полное время выполнения запросов и объём передаваемых данных.

Результаты экспериментов показали, что использование полиморфного оператора позволяет сократить время выполнения гибридных запросов более чем на 40% по сравнению с классическим ETL-подходом. Достигнутый выигрыш обусловлен исключением межпроцессного обмена данными и более эффективным планированием запросов.

Помимо количественного прироста производительности, предложенный подход демонстрирует ряд качественных преимуществ. Логика аналитики концентрируется в одном SQL-запросе, что упрощает разработку и сопровождение. Исключение этапов передачи данных повышает безопасность и согласованность вычислений, а представление ML-моделей в виде объектов базы данных улучшает управляемость и повторное использование аналитических компонентов.

С практической точки зрения результаты подтверждают, что глубокая интеграция машинного обучения в СУБД является перспективным направлением развития HTAP-систем, особенно в контексте построения суверенных и экономически эффективных аналитических платформ.

На основе полиморфного оператора, встроенного в механизм выполнения запросов СУБД предложена и экспериментально апробирована модель оптимизации гибридных SQL+ML-запросов. Показано, что такой подход позволяет преодолеть ограничения классической двухконтурной архитектуры и существенно повысить производительность аналитических вычислений.

Дальнейшие исследования могут быть направлены на разработку полнофункционального расширения для PostgreSQL.

Список литературы:

1. Hellerstein J., Ré C., Schoppmann F., et al. The MADlib Analytics Library: or MAD Skills, the SQL // Proceedings of the VLDB Endowment. 2012. Vol. 5, № 12. P. 1700–1711.
2. Kumar A., Boehm M., Yang J. Data Management in Machine Learning: Challenges, Techniques, and Systems // SIGMOD. 2017. P. 1717–1722.
3. Karanasos K., Interlandi M., Xin D., et al. Extending Relational Query Processing with ML Inference // Proceedings of the 10th Biennial Conference on Innovative Data Systems Research (CIDR). 2020. P. 1–7.
4. Гольчевский Ю. В., Захаров И. Д. Анализ развития систем управления базами данных // Вестник Сыктывкарского университета. Серия 1: Математика. Механика. Информатика. 2025. № 2. С. 45–53.

UDC 004.65

OPTIMIZATION OF HYBRID ANALYTICAL QUERIES BASED ON EMBEDDED MACHINE LEARNING MODELS IN POLYMORPHIC DBMS

Andrei V. Verevkin
student
mr_and_55@mail.ru
Larisa Iv. Nikonorova
candidate of agricultural sciences, associate professor
lenaniknrva@rambler.ru
Michurinsk State Agrarian University
Michurinsk, Russia

Abstract. This article explores a hybrid approach to executing analytical tasks that combine SQL relational operations with predictive machine learning (ML)

analytics directly within a DBMS. A model is proposed that overcomes the limitations of the classic two-circuit architecture (DBMS + external ML framework) by introducing a "polymorphic operator" into the query planner. The execution and optimization of hybrid plans are enforced through deep integration of ML models as database objects.

Keywords: machine learning in databases, polymorphic DBMS, Hybrid Transactional/Analytical Processing (HTAP), query optimization, predictive analytics, PostgreSQL, MADlib, query execution.

Статья поступила в редакцию 25.02.2026; одобрена после рецензирования 20.03.2026; принята к публикации 31.03.2026.

The article was submitted 25.02.2026; approved after reviewing 20.03.2026; accepted for publication 31.03.2026.