УДК 004.65:004.8

ПРОГНОЗУВАННЯ ЧАСУ ВИКОНАННЯ SQL-ЗАПИТІВ ІЗ ВИКОРИСТАННЯМ АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ

Озарінська О. В.
Національний університет «Одеська політехніка»
Україна, Одеса
o.v.ozarinska@op.edu.ua

Анотація. *Запропоновано та пілотно перевірено методику прогнозування часу виконання SQL-запитів за характеристиками запиту й плану виконання. На синтетичному наборі SQLite найменшу похибку серед трьох моделей показав випадковий ліс (MAE 0,241 мс; RMSE 0,442 мс).*
Ключові слова: *SQL-запити; машинне навчання; прогнозування часу виконання; плани виконання; реляційні бази даних.*

Вступна частина. Зростання обсягів даних і різноманітності аналітичних навантажень підвищує вимоги до прогнозованості продуктивності реляційних систем керування базами даних (СУБД). Традиційний оптимізатор оцінює альтернативні плани виконання SQL-запиту за внутрішньою вартісною моделлю, однак її оцінка не тотожна фактичному часу виконання. Неточності статистики, кореляції між атрибутами, конкуренція за ресурси та зміна апаратного середовища ускладнюють прогнозування [1].

У попередньому дослідженні автора розглядалися індексування, аналіз планів виконання, кешування й матеріалізовані представлення як інструменти оптимізації SQL-запитів. Подальший розвиток цього напряму пов’язаний із використанням моделей машинного навчання, здатних виявляти нелінійні залежності між характеристиками плану та тривалістю виконання. Водночас такі моделі можуть втрачати точність при перенесенні на інші набори даних або типи запитів [2, 3].

Метою дослідження є розроблення відтворюваної методики порівняльного оцінювання моделей машинного навчання для прогнозування часу виконання SQL-запитів, визначення інформативних ознак плану виконання та критеріїв перевірки узагальнювальної здатності моделей. Методику доповнено пілотним обчислювальним експериментом на синтетичних даних у SQLite.

Основна частина. Об’єктом прогнозування є тривалість виконання запиту t, виміряна засобами СУБД за фіксованої конфігурації середовища. Вхідний вектор ознак x формується з характеристик SQL-запиту, плану виконання та статистики даних. До базових ознак доцільно включити кількість операторів з’єднання JOIN, групування GROUP BY і сортування ORDER BY, оцінену кількість рядків, сумарну вартість плану, кількість операцій послідовного й індексного сканування, обсяг таблиць і вибірковість предикатів. Для запобігання витоку цільової змінної вхідні ознаки слід брати з плану EXPLAIN без фактичних часових показників EXPLAIN ANALYZE; останній застосовують лише для отримання цільового часу.

Запропонований процес складається з п’яти етапів: формування набору запитів; отримання структурованих планів виконання; побудова таблиці ознак; навчання моделей; оцінювання якості на відкладених запитах. Для PostgreSQL плани можна експортувати у форматі JSON; у пілотному експерименті використано EXPLAIN QUERY PLAN середовища SQLite. Кожен запит бажано виконувати багаторазово, фіксуючи медіанний час та розкид вимірювань; умови кешування, паралельного навантаження й конфігурацію сервера необхідно документувати.

Як базову модель обрано лінійну регресію, яка забезпечує інтерпретовану залежність між ознаками та прогнозом. Випадковий ліс (Random Forest) дозволяє моделювати нелінійні взаємодії без явного задавання їхньої форми. Градієнтний бустинг (Gradient Boosting) послідовно уточнює ансамбль дерев і є перспективним для неоднорідних табличних даних.

Вибір цих алгоритмів дає змогу зіставити просту інтерпретовану модель із двома ансамблевими підходами, не стверджуючи наперед перевагу будь-якого з них.

Для оцінювання якості прогнозування використано дві метрики: середню абсолютну похибку (MAE), яка показує середнє відхилення прогнозу від виміряного часу, та корінь із середньоквадратичної похибки (RMSE), чутливіший до великих помилок. Обидва показники наведено в мілісекундах; менші значення відповідають точнішому прогнозуванню.

У повномасштабному дослідженні навчальні й тестові дані слід розділяти за шаблонами запитів, а не за повторними виконаннями одного SQL-виразу. Для пілотної перевірки нижче застосовано інший, обмеженіший сценарій: відкладено параметризовані варіанти чотирьох відомих шаблонів. Такий тест оцінює прогнозування для нових значень параметрів, але не доводить перенесення на принципово незнайомі структури запитів.

Окрему увагу приділено впливу оцінок кардинальності – кількості рядків, яку оптимізатор очікує отримати після кожної операції. Помилки оцінювання кардинальності можуть спричиняти неефективний вибір плану та одночасно погіршувати якість ознак для машинного навчання. Тому варто порівнювати моделі на двох наборах ознак: мінімальному (структура запиту та агреговані оцінки плану) і розширеному (ознаки окремих операторів та статистика таблиць). Різниця в якості прогнозу покаже, чи виправдовує складніший опис плану додаткові витрати на його оброблення.

У працях із машинного навчання для оптимізації запитів запропоновано як спеціалізовані моделі оцінювання продуктивності, так і підходи до інтеграції прогнозів у процес вибору плану [2, 3]. Проте результати таких досліджень залежать від набору запитів, схеми бази даних і умов експерименту. Для практичного впровадження моделі необхідно оцінювати не лише похибку прогнозу, а й час обчислення ознак, швидкість отримання прогнозу та потребу в перенавчанні.

За результатами тестування лінійна регресія досягла MAE 1,061 мс та RMSE 1,377 мс; випадковий ліс – MAE 0,241 мс та RMSE 0,442 мс; градієнтний бустинг – MAE 0,360 мс та RMSE 0,508 мс. Отже, у цьому пілотному сценарії випадковий ліс забезпечив найменшу похибку. Водночас експеримент виконано в оперативній пам’яті на синтетичних даних, без зміни структури запитів між навчанням і тестуванням; результати не характеризують продуктивність PostgreSQL, MySQL чи виробничих навантажень.

Ознаками моделей були тип запиту, параметри фільтрації, а також кількість операцій сканування, індексного пошуку й використання тимчасових структур за EXPLAIN QUERY PLAN. Навчальну вибірку склали 180 варіантів, тестову – 60; усі чотири шаблони були представлені в обох вибірках. Моделі реалізовано засобами Python і scikit-learn з фіксованим початковим значенням генератора випадкових чисел.

Пілотний експеримент. Для перевірки методики створено синтетичну таблицю SQLite з 50 000 записів та індексами за категорією і числовою ознакою. Сформовано 240 варіантів запитів чотирьох типів: агрегування з фільтрацією, обчислення середнього значення, групування та вибірка з сортуванням. Для кожного запиту виконано попередній прогрів і три вимірювання часу; цільовим значенням обрано медіану трьох запусків.

Аналітичний результат роботи полягає у формуванні критеріїв порівняння алгоритмів: точність, стійкість до змін навантаження, інтерпретованість і накладні витрати. Пілотний експеримент дозволив перевірити здійсненність запропонованої схеми на обмеженому синтетичному наборі; його результати не можна безпосередньо переносити на промислові СУБД.

Висновки. Запропоновано та пілотно перевірено методику прогнозування часу виконання SQL-запитів за характеристиками запиту й плану виконання. На синтетичному наборі SQLite найменшу похибку серед трьох моделей показав випадковий ліс (MAE 0,241 мс; RMSE 0,442 мс). Висновок стосується лише описаних умов експерименту.

Показано важливість відокремлення ознак прогнозування від вимірюваного часу, контролю повторних запусків і перевірки на незалежних варіантах запитів. Подальша робота

передбачає тестування на нових шаблонах, різних обсягах даних і в PostgreSQL; лише після цього можна оцінювати узагальнювальну здатність моделей.

Використання інструментів ШІ: не використовувались.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] Leis, V., Gubichev, A., Mirchev, A., Boncz, P., Kemper, A., & Neumann, T. (2015). How good are query optimizers, really? *Proceedings of the VLDB Endowment*, 9(3), 204–215. https://doi.org/10.14778/2850583.2850594

[2] Marcus, R., Negi, P., Mao, H., Zhang, C., Alizadeh, M., Kraska, T., Papaemmanouil, O., & Tatbul, N. (2019). Neo: A learned query optimizer. *Proceedings of the VLDB Endowment*, 12(11), 1705–1718. https://doi.org/10.14778/3342263.3342644

[3] Marcus, R., & Papaemmanouil, O. (2019). Plan-structured deep neural network models for query performance prediction. *Proceedings of the VLDB Endowment*, 12(11), 1733–1746. https://doi.org/10.14778/3342263.3342646

Ozarinska O. V.

Predicting SQL query execution time using machine learning algorithms

Abstract. *A reproducible methodology for predicting SQL query execution time using execution-plan features is proposed. Linear regression, random forest, and gradient boosting are selected for comparison. The study defines evaluation criteria, including MAE, RMSE, robustness to unseen query templates, and computational overhead. A pilot experiment on 240 synthetic SQLite query variants found the lowest test error for random forest (MAE 0.241 ms; RMSE 0.442 ms). The results are specific to this controlled setup.*

Keywords: *SQL queries; machine learning; execution time prediction; query plans; relational databases.*