

Ad fontes

Волоський Максим і Федорів Святослав

Постановка задачі та загальний опис підходу

На вхід у вигляді даних подається колекція точок інтересу, з перевагами даних точок та вагами ребер між ними. Користувач хоче знайти K шляхів, що задовільнили би його запити до переваг точок через які будуть прокладені маршрути, що будуть запропоновані користувачеві. Данні маршрути повинні починатися з точки, де знаходиться користувач і закінчуватись в точці вказаній ним. Одним з перших викликів є персоналізація вимог, поза як кожен користувач має власні вимоги, щодо переваг точок(вузлів) маршруту. Іншою проблемою є великий об'єм точок, а одже велика кількість альтернативних (а іноді майже ідентичних між собою) маршрутів.

З вище переліченого зрозуміло, що данні розглядаються, як зв'язаний, зважений граф з певними індексами(властивостями) точок, які репрезентують переваги кожної з точок. На вхід користувач подає пару точок, початок та кінець маршруту, вектор індексів, які репрезентують індивідуальні вимоги до маршруту користувача, константа бюджету користувача та залежність між кількістю та різноманітністю маршрутів, які користувач отримає в результаті.

Отже кожен шлях може бути охарактеризований функцією $F = \sum W(h) \cdot \Phi(h)(p_h)$; де $W(h)$ індекс відповідної переваги h в векторі індексів переваг, $\Phi(h)$ функція залежності між кількістю та різноманітністю для даної переваги, $p(h)$ множина точок, які мають цю перевагу. Тобто потрібно вибрати такі шляхи, з найбільшою F та найменшим бюджетом

Власний опис алгоритму

Якщо функція F представляє собою оцінку маршруту відносно переваг точок через які проходить граф, то зрозуміло що наша задача зводиться до пошуку оптимального варіанту між вартістю та вигодами, які пропонують об'єкти, крізь які проходить знайдений маршрут.

1) Сформулювати проблему, як пошук маршруту в зваженому графі з персональними вимогами користувача стосовно різноманітності за допомогою функції субмодульної агрегації Φ_h , як параметра запиту.

2) Створити таку структуру даних(на основі абстракції графу), яка б дала можливість визначати точки з допомогою індексів вартості подорожей між точками, це б значно прискорило алгоритм, поза-як дало б змогу уникнути цих обчислень в алгоритмі.

3) Алгоритм загалом складається з онлайн та оф-лайн частини.
Зрозуміло, що перед початком роботи данні завантажуються підчас онлайн частини та записуються до структури даних.

4) Перший етап нашого алгоритму, не беручи до уваги звісно завантаження файлів з інтернету це переписування в нашу структуру даних, а також обрахування вартості шляхів між точками, а потім знайдення топ шляхів.

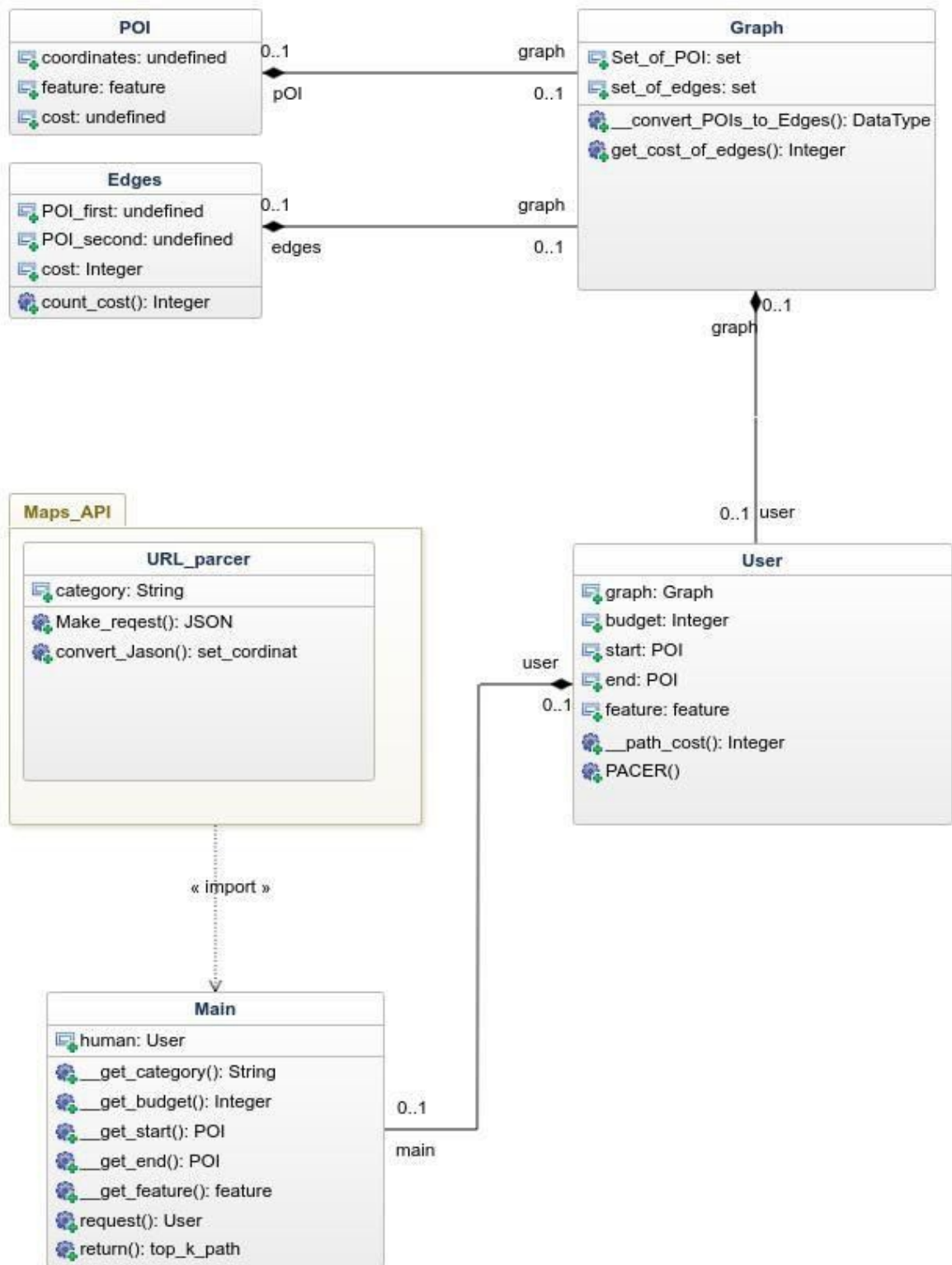
Оцінка складності алгоритму

В повнозв'язному графі є N точок, отже з кожної точки виходить $N-1$ ребро. Нехай початковий шлях від Start до End = SE. Тоді розташуємо всі ребра, що виходять з S в порядку зростання. Перебираємо ці ребра, якщо

$$\text{cost}(i) \leq \text{budget} - (\text{cost}(i-1) - \text{edge}).$$

Якщо цей шлях нам підходить, ми дивимось чи $\text{Gain}(i-1)$ менший за шлях який ми розглядаємо, якщо це так тоді добавляємо цей шлях у чергу з пріоритетом де пріоритет цього шляху результат роботи Gain. Очевидно що так як ми розглядаємо N ребер складність роботи $O(N)$ в найгіршому випадку. Так як наша програма працює з API, потрібен ключ, якщо список шляхів пустий це значить, що ви повністю використали ліміт ключа. Зверніться до розробника.

Діаграма розроблених класів:



Алгоритм пошуку шляхів

1) Головною ідеєю даного алгоритму, є в суті своїй обрізання найменш перспективних варіантів шляхів, які точно не можуть становити цінності для користувача. Для пояснення даного алгоритму вводяться такі поняття. Не завершений шлях це такий набір POI, що не містить в собі кінцевого пункту призначення, а містить декілька POI. Завершений шлях це такий, що містить в собі останню точку, тобто кінцевий пункт призначення. Далі наведемо формули.

2) $PV = P - VU\{j\}$ де PV маршрут дорівнює неповному маршруту + кінцевий пункт призначення $Gain(PV) = Phwh\Phi h(PV)$ цю формулу ми згадували вище, вона стосується пропозицій на всьому шляху $end(P) = j$ кінцева точка маршруту $cost(P) = cost(P-) + Ti,j + sj$ вартість маршруту, для користувача.

3) Замість просто побудувати найкоротший зі шляхів ми викреслюємо ті шляхи, що не підходять нам по параметрах, і залишаємо лише ті, які задовільняють вимоги користувача. На вихід подаються шляхи в порядку спадання переваг та бюджету, що найкраще відповідають заданим вимогам.

