Znajdowania maksymalnego podgrafu spójnego - opis algorytmu przyblizonego

Pamela Krzypkowska, Michał Kortała

November 6, 2018

1 Założenia

Mamy dwa grafy spojne, które charakteryzuja sie zbiorem wierzchołków i krawedzi.
Można te grafy opisać w poniższy sposób: $G1 = (V1, E1), G2 = (V2, E2)$.
Reprezentacja grafu to macierz sasiedztwa M[,] o wymiarach nxn, gdzie n - liczba wierzchołków grafu.
Jeżeli istnieje krawedz miedzy wierzchołkiem v i u, to w macierzy sasiedztwa M[v,u] = 1 oraz M[u,v] = 1, gdyż grafy nie sa skierowane.

2 Algorytm przybliżony

Prezentowany przez nas algorytm przybliżony opiera sie na wykorzystaniu algorytmu Dijkstry i pózniejszym wykorzystaniem zwroconym przez algorytm odleglosci od kolejnych wierzcholkow, jako funkcji wagowej.

Teraz pytanie jak wybrac pierwszy wierzcholek od ktorego bedziemy liczyc odleglosci za pomoca algorytmu Dijkstry. Pierwszy wierzcholek to bedzie ten ktory ma najwyzszy stopien w naszym grafie. Znajdziemy go przeszukujac cala macierz sasiedztwa.

3 Opis dokladny

Dla dwoch danych grafow G1 i G2 przechodzimy przez wszystkie wiersze albo kolumny (nie ma tutaj roznicy, wynika to z symetrii macierzy sasiedztwa) i znajdujemy wierzcholek o maksymalnym stopniu.

Od wierzcholka o maksymalnym stopniu (w obu grafach) puszczamy algorytm Dijkstry, jako waga każdej krawedzi dajemy 1. Dzieki temu otrzymujemy odleglosc (liczona w ilosci krawedzi) od naszego wierzcholka o maksymalnym stopniu do kazdego innego wierzcholka w grafie.

Budowanie naszego maksymalnego podgrafu zaczynamy od wierzcholka o maksymalnym stopniu. Dodajemy jako pierwszy wierzcholek podgrafu - wierzcholek o maksymalnym stopniu ze stopniem mniejszym z dwoch grafow. Dla dwoch grafow o maksymalnych stopniach kolejno 4 i 7 wybieramy oba te wierzcholki ze stopniem 4 (czyli dodajemy 4 krawedzie z obu grafow do podgrafu wyjsciowego).

Jak wybrac teraz owe krawedzie gdy stopnie wierzcholkow nie sa takie same? Dla wierzcholka wejsciowego (pierwszego wierzcholka w podgrafie) nie ma to znaczenia gdyz nasza heurystyka bedzie odleglosc wierzcholkow od wierzcholka poczatkowego wyliczone za pomoca algorytmu Dijkstry. Mozemy wiec w pierwszej iteracji wybrac dowolne krawedzie. Wierzcholki do ktorych prowadza wybrane krawedzie dodajemy do podgrafu i wrzucamy na stos (albo do jakiejkolwiek innej struktury danych) by zajac sie nimi w nastepnych iteracjach.

W kazdej kolejnej iteracji wyjmujemy kolejny wierzcholek dodany do podgrafu ale nie przerobiony jeszcze (w drugiej iteracji przerobiony jest tylko wierzcholek o najwyzszym stopniu) i dokonujemy dodania kolejnych krawedzi i wierzcholkow do podgrafu. Mapowanie kolejnych wierzcholkow odbywa sie za pomoca odleglosci obliczonych za pomoca algorytmu Dijkstry. Mapujemy na siebie wierzcholki ktore maja te sama odleglosc od wierzcholka o maksymalnym stopniu. W przypadku gdy nie ma wierzcholkow o dokladnie tej samej odleglosci, wybieramy takie o jak najblizszych odleglosciach.

Iterujemy i dodajemy krawedzie i wierzcholki tak dlugo az nie bedziemy mogli znalezc zadnych wierzcholkow spelniajacych kryterium mapowania.

4 Złożonosc

Zlożonoc policze w przypadku operacji na jednym z grafów, potem wystarczy pomnożyć ilosc operacji przez dwa (zeby wykonac je tez dla drugiego grafu). Znalezienie maksymalnego stopnia wierzcholka wymaga jednego przejscia przez macierz sasiedztwa, czyli zlozonosc jest na poziomie n^2 ,gdzie n jest liczba wieczolkow, bo musimy przejsc przez wszystkie krawedzie, w grafie gestym zaleznosc krawedzi i wierzcholkow to wlasnie $E \approx n^2$. Algorytm Dijkstry ktory musimy wykonac by dostac wynikowa macierz odleglosci od naszego wierzcholka, nazwijmy go v, do wszystkich pozostalych to w najgorszym wypadku n^2, jesli nie uzyjemy kolejki priorytetowej a tylko nieposortowanej tablicy czyli naszej wejsciowej prezentacji grafu.

Nastepnie gdy mamy juz wszystkie te poczatkowe operacje wykonane, przechodzimy do glównej czesci algorytmu czyli znajdowania maksymalnego podgrafu. W jednej iteracji zajmujemy sie jednym wierzcholkiem wiec maksymalnie mozemy mieć n iteracji zewnetrznej petli dodajacej kolejne wierzcholki do naszego podgrafu. Potem dla wybranego wierzcholka (wyjetego ze stosu), patrzymy czy mozemy dodac jego sasiadow do naszego podgrafu, mapujac na siebie wierzcholki o tej samej odleglosci od wierzcholka o maksymalnym stopniu (tutaj znowu n operacji, bo trzeba przejsc cala tablice odleglosci wyznaczona przez algorytm Dijkstry). Dla kazdego nowego wierzcholka ktory chcemy dodac musimy sprawdzic czy nie psuje on nam podgrafu ktory juz zbudowalismy, musimy przejsc wiec przez wszystkie wierzcholki istniejacego podgrafu i zobaczyc czy sa one np. juz dodane, mozna przyjac ze w najgorszym wypadku bedzie to n operacji, gdzie n jest iloscia wierzcholkow grafu.

Gdy nie mozemy dodac wierzcholka o takim samej odleglosci, dodajemy taki o najblizszej mozliwej odleglosci ze wszystkich, jesli nie narusza on struktury naszego grafu.

Iteracje prowadzimy az nie bedziemy w stanie dodac zadnego nowego wierzcholka (wszystkie juz sprawdzimy).

W sytuacji algorytmu maksymalizujacego V + E, w momencie znalezienia podgrafu i wyjscia z petli, bedziemy probowac dodawac krawedzie miedzy dodanymi juz i zmapowanymi wierzcholkami. Zlozonosc tej operacji jest okreslana liczba krawedzi czyli n^2.

Finalna zlozonosc to w takim razie:
$C = n^2 + n^2 + n * (n * n) + n^2 = 2 * n^2 + n^3 + n^2$
$C = O(n^3)$