Politechnika Poznańska
Wydział Informatyki
Instytut Informatyki

Praca dyplomowa inżynierska

Implementacja algorytmu eksploracji danych z użyciem CUDA API

Marcin Jabłoński
Łukasz Kosiak
Piotr Kurzawa
Marek Rydlewski

Promotor:
dr inż. Witold Andrzejewski

Poznań, 2017 r.

„Coś się popsuło“
Zbigniew Stonoga

Spis treści

1 Wstęp

1.1 Wprowadzenie

Informatyzacja życia codziennego, jaka dokonała się w ostatnich latach sprawiła, że każdego dnia często nieświadomie zostawiamy po sobie wiele informacji na swój temat. Nawet z pozoru niewinne dane o naszych przyzwyczajeniach typu "z której półki bierzemy bułki w sklepie" są zapisywane w nieznanych nam systemach informatycznych. Dodając do tego inne usługi świadomie przez nas wykorzystywane - chociażby zapisywanie naszej lokalizacji przez prywatny telefon komórkowy - uzyskujemy dość ponury obraz tego, co jesteśmy w stanie po sobie zostawić. Co gorsza, chcąc czy nie chcąc, musimy się pogodzić z faktem, że dane te mogą zostać wykorzystane w różnym celu. Czy mamy jednak czego się obawiać?

Wbrew pozorom, taka błaha na pierwszy rzut oka informacja może mieć jednak istotne znaczenie dla funkcjonowania przemysłu piekarskiego. Przecież takich informacji codziennie my, klienci, zostawiamy ogromne ilości. Nic nie szkodzi na przeszkodzie, aby spróbować z tych danych odczytać preferencje bądź przyzwyczajenia przeciętnego Kowalskiego na temat jego codziennych zakupów, które mogą w przyszłości zaprocentować - zarówno dla właściciela, jak i klienta. Jest to oczywiście tylko przykład, ale oddaje doskonale fakt przydatności z pozoru nie mających znaczenia prostych czynności człowieka, jakie często przypadkiem rejestrują działające wokół nas systemy.

Pozostaje jednak problem przetworzenia takich danych w celu otrzymania interesującej nas informacji, która byłaby potencjalnie użyteczna. Trzeba pamiętać, że rozmiar takich danych nierzadko sięga terabajtów i w praktyce skuteczna analiza takich danych przez człowieka nie jest możliwa. Musi on zatem w tym celu skorzystać z dobrodziejstw, jakie przynosi mu współczesna technologia.

Problem efektywnego przetwarzania zdążył urosnąć do rangi oddzielnego działu w informatyce. W pracy [1] zasugerowano utworzenie nowej dyscypliny mającej na celu opracowanie technik obliczeniowych rozwiązujących takie problemy, zwanej roboczo odkrywaniem wiedzy w bazach danych (ang. KDD – *Knowledge Discovery in Databases*). Techniki te mają na celu odnajdywanie prawidłowych i potencjalnie użytecznych wzorców w dużych zbiorach danych.

Wspominane wyżej techniki w dużej mierze zależą od rodzaju bazy, a ściślej mówiąc - charakteru danych występujących w niej. W przypadku danych zawierających informację o położeniu zazwyczaj mowa jest o odkrywaniu wiedzy w bazach danych przestrzennych (ang. *spatial data mining*). Takie systemy mogą zawierać atrybut lokalizacji obiektu w danym obszarze, jego opis w formie geometrycznej (np. w postaci wielokątów), a także inne atrybuty nieprzestrzenne. Okazuje się, że tradycyjne metody analizy danych przestrzennych zazwyczaj nie radzą sobie z nimi na tyle efektywnie, by było opłacalne ich użycie w praktyce [3], dlatego też zaczęto szukać nowych sposobów na odkrywanie wiedzy w takich bazach.

W pracy [4] zaproponowano odkrywanie *wzorców kolokacji przestrzennych* (lub krócej: *kolokacji*), czyli zbioru cech przestrzennych występujących w niewielkiej odległości od siebie. Łatwo to można sobie wyobrazić na przykładzie przyrody, gdzie osobniki (gatunki) o podobnych cechach zazwyczaj trzymają się razem. Rozumowanie to działa również dla bliższych współczesnemu człowiekowi cech przestrzennych, np. punktach o podobnej funkcji - stacje, kina, piekarnie, itd. Wraz z rosnącą popularnością obliczeń na kartach graficznych (w dużej mierze spowodowana wprowa-

dzeniem technologii *CUDA* autorstwa firmy NVIDIA) pojawiło się wiele gotowych rozwiązań, pozwalających na efektywne wyszukiwanie kolokacji nawet w bardzo rozbudowanych bazach danych. Przegląd niektórych z nich można znaleźć w pracy [5].

Ostatni rok przyniósł kolejną metodę efektywnego przeszukiwania baz danych w celu odnalezienia kolokacji [6]. Wykorzystuje ona autorski algorytm wyszukiwania maksymalnych klik w grafie rzadkim oraz skondensowane drzewa instancji przechowywaczce kliki instancji dla każdego kandydata do kolokacji (patrz Rozdział 2) w celu zmniejszenia czasu obliczeń oraz ograniczenia wymagań co do pamięci operacyjnej. Algorytm ten jest przedmiotem badań niniejszej pracy zbiorowej.

1.2 Cel i zakres pracy

Celem niniejszej pracy jest analiza wydajności zaproponowanych w pracy [6] rozwiązań z zakresu odkrywania kolokacji przestrzennych dla GPU i CPU.

Zakres pracy obejmuje następujące zadania szczegółowe:

1. **Zapoznanie się z literaturą.** Zapoznanie się z podstawowymi pojęciami dotyczącymi odkrywania danych w bazach danych przestrzennych oraz wyszukiwania wzorców kolokacji przestrzennych jest niezbędne do stworzenia działającej implementacji powyższego algorytmu. Dodatkowo należy zwrócić uwagę na dodatkowe zagadnienia związane z teorią grafów.

2. **Opracowanie wersji równoległej algorytmu eksploracji danych.** Konieczne jest przemyślenie wykorzystania algorytmów pomocniczych dla poszczególnych kroków całego rozwiązania oraz zaproponowanie możliwie najkorzystniejszego rozwiązania biorąc pod uwagę dostępną pamięć operacyjną, czas przetwarzania i przesyłania danych między pamięcią operacyjną a pamięcią karty graficznej.

3. **Implementacja wersji sekwencyjnej i równoległej ww. algorytmu.** Rozwiązanie podane w punkcie drugim powinno zostać zaimplementowane w technologii NVIDIA CUDA dla wersji GPU oraz biblioteki OpenMPI w przypadku odmiany dla CPU.

4. **Przeprowadzenie eksperymentów wydajnościowych.** Analiza wyników testów wydajnościowych implementacji z punktu 3 jest głównym celem tej pracy. Należy zbadać efektywność obu rozwiązań pod względem czasu wykonywania oraz zapotrzebowania na dostępną pamięć.

1.3 Charakterystyka źródeł

Jak już wspomniano, niniejsza praca w dużej mierze opiera się o algorytm zaprezentowany w dokumencie [6]. Do jej opracowania była wymagana wiedza zawarta w innych źródłach, często również o charakterze naukowym.

Głównym źródłem wiedzy na temat kolokacji przestrzennych była rozprawda doktorska dr inż. Pawła Boińskiego [5], która w dużym przekroju omawia ideę kolokacji zaprezentowaną przez Shakara i Huanga w pracy [4], a także prezentuje najpopularniejsze techniki ich odkrywania (metody *Co-location Miner*, *iCPI-tree*). Część rozwiązań wykorzystanych w tych technikach została wykorzystana w trakcie realizacji algorytmu.

Oddzielną kwestią jest literatura książkowa, wykorzystana do zapoznania się z technologią CUDA oraz przyjęcia dobrych praktyk optymalizacyjnych i programistycznych. Tutaj szczególnie należy wymienić popularną pozycję *CUDA w przykładach* autorstwa Shane'a Cooke'a [7], a także *Professional CUDA C Programming* [8] będącą również podstawą do wstępu teoretycznego w rozdziale drugim.

1.4 Struktura pracy

W pracy przedstawiono główne pojęcia związane z wyszukiwaniem kolokacji przestrzennych oraz programowaniem równoległym na procesory graficzne i zawarto je w rozdziale 2. Rozdział 3 poświęcony jest algorytmowi będącemu głównym tematem pracy. Rozdział 4 opisuje implementację tego algorytmu w technologii CUDA, natomiast rozdział 5 prezentuje wyniki przeprowadzonych testów.

W tym miejscu zasadniczo będzie można napisać więcej, jeżeli już te rozdziały zostaną ustalone bądź wstępnie uzupełnione. Poza tym należy ustalić, czy w ogóle potrzebujemy takiego działu dla tak małej pracy. Z drugiej strony, zawsze to jednak te pół strony więcej spamu - borewicz

1.5 Podział pracy

Marcin Jabłoński w ramach niniejszej pracy wykonał projekt tego i tego, opracował .....

Łukasz Kosiak wykonał ....., itd.

2 Podstawy teoretyczne

2.1 Charakterystyka danych przestrzennych

2.1.1 Modelowanie danych przestrzennych

Sposób reprezentacji danych przestrzennej w dużej mierze zależy od zastosowań, niemniej najczęściej przybiera jedną z następujących form:

- *model pól* - ma formę funkcji, której dziedzina należy do modelowanej przestrzeni, a jego wynikiem jest cecha przestrzenna;
- *model obiektowy* - dla każdego zjawiska jest tworzony nowy obiekt z odpowiednimi właściwościami (etykietami, atrybutami przestrzennymi i nieprzestrzennymi).

W praktyce model pól używany jest przede wszystkim w metodach opartych na dokonywaniu pomiarów z powietrza - takie dane mają wtedy charakter rastrowy (reprezentacja w postaci pikseli). Model obiektowy stosowany jest natomiast w przypadkach, gdzie występuje duża liczba dodatkowych atrybutów nieprzestrzennych.

2.1.2 Źródła danych przestrzennych

Najogólniej źródła danych przestrzennych można podzielić ze względu na ich format.

Pierwotne źródła danych są opracowane w jednym ze standardowych formatów źródeł (najczęściej dla konkretnego systemu) i nie wymagają jakichkolwiek transformacji. Mają one zazwyczaj postać cyfrową i pochodzą z automatycznych pomiarów dokonanych przez specjalizowane systemy wyposażone w odbiorniki GPS czy tachimetry.

Wtórne dane źródłowe nie zostały zebrane z myślą o wykorzystaniu w systemach typu GIS i dlatego wymagają one odpowiedniej transformacji oraz cyfryzacji (jeżeli są one analogowe). Procedury te są one obarczone pewnym ryzykiem, ponieważ istnieje możliwość wystąpienia błędów w trakcie konwersji i w konsekwencji przekłamaniami w danych wynikowych, które należy ręcznie poprawić.

2.1.3 Relacje

Określenie zachodzących relacji między obiektami w źródłach danych przestrzennych jest ważnym elementem przetwarzania danych przestrzennych. Sposób ich określenia zależy od zastosowanego modelu danych.

W modelu pól relacje determinowane są przez operacje pól (ang. *field operations*, [13]), mogące przybierać różne formy w zależności od zastosowań, natomiast w modelu obiektowym rodzaje relacji przestrzennych zależą od definicji przestrzeni. Według standardu OGC istnieją trzy najpopularniejsze rodzaje związków przestrzennych między obiektami:

- *Relacje metryczne* - wyrażane w postaci predykatów typu "w odległości nie większej niż 10 metrów", oparte na odległości;

- *Relacje kierunkowe* - położenie określone jest względem globalnych kierunków dla przestrzeni (np. na północ, na południe - są to relacje bezwzględne) lub względem innego obiektu/obserwatora (nazywamy takie relacjami względnymi);

- *Relacje topologiczne* - najbardziej skomplikowane, wyrażone przez zależności typu pokrywanie, zawieranie, styczność.

W systemach typu GIS stosuje się głównie relacje topologiczne. Mają one postać predykatów przestrzennych dla operacji filtrowania i połączenia przestrzennego w językach zapytań działających na danych przestrzennych. Najczęściej wykorzystuje się je w tzw. *modelu dziewięciu przecięć* [14], za pomocą którego określa się możliwe relacje zachodzącą dla pary obiektów.

Dla każdego obiektu wyznacza się jego wnętrze, granicę i zewnętrze. Następnie, dokonuje się operacji przecięcia dla danej pary obiektów dla każdej z możliwych kombinacji elementów tego obiektu (np. granica pierwszego obiektu z wnętrzem drugiego). Takich relacji w dwuwymiarowej relacji można wyznaczyć osiem, należą do nich np. rozłączność, styczność, częściowe i całkowite pokrycie itd.

Istnieje również rozszerzenie modelu dziewięciu przecięć, zwanym DE-9IM (ang. *Dimensionally Extended nine-Intersection Model*, [15]), które rozróżnia rodzaj obiektu uzyskanego w wyniku przecięcia (mogą być puste, bezwymiarowe, jednowymiarowe i dwuwymiarowe).

2.2 Metody eksploracji danych przestrzennych

Specyfika danych przestrzennych, a w szczególności fakt, że własności obiektu w danych przestrzennych mogą zależeć od cech jego sąsiadów, powoduje, że stosowanie klasycznych metod eksploracji danych może doprowadzić do nieprawidłowych wyników [16] [17] - stąd też istnieje konieczność korzystania z metod eksploracji dedykowanych dla danych przestrzennych. Wiele z nich jest tak naprawdę rozwinięciem metod opracowanych dla klasycznych zbiorów danych.

2.2.1 Grupowanie przestrzenne

Metoda grupowania przestrzennego (ang. *spatial clustering*) zakłada istnienie przestrzeni m-wymiarowej, w której znajdują się punkty odpowiadające obiektom. Przestrzeń ta ma rozkład niejednorodny, a każdy z obiektów jest opisany przez m atrybutów. Celem grupowania jest poszukiwanie gęstych obszarów punktów używając miary Euklidesowej jako funkcji podobieństwa.

Grupowanie przestrzenne w największym stopniu spośród wszystkich metod eksploracji danych przestrzennych jest podobna do swojego klasycznego odpowiednika - wiele algorytmów grupowania opracowanych dla klasycznych zbiorów danych zadziała również dla danych przestrzennych.

2.2.2 Klasyfikacja przestrzenna

Klasyfikacja przestrzenna (ang. *spatial classification*) działa podobnie jak jego odmiana dla danych klasycznych - przewiduje klasy nowych obiektów w oparciu o tzw. zbiór uczący, składający się ze wcześniejszych obserwacji.

W celu dostosowania klasyfikacji dla danych przestrzennych zaproponowano [18] wykorzystanie *grafu sąsiedztwa* będącego reprezentacją relacji przestrzennych między obiektami, w którym wierzchołki stanowią obiekty przestrzenne, a relacje są krawędziami. Następnie w takim grafie wyznaczane są wszystkie ścieżki, których początkiem jest analizowany obiekt. Dalej analiza przebiega zgodnie z algorytmem *ID3* [19].

Z pojęciem klasyfikacji przestrzennej wiąże się także predykcja położenia (ang. *location prediction*), czyli przewidywanie zdarzeń we wskazanym miejscu w przestrzeni, przy uwzględnieniu autokorelacji przestrzennej. Przydaje się ono np. w określaniu regionów o wysokim ryzyku wystąpienia klęsk żywiołowych czy awarii.

2.2.3 Odkrywanie trendów przestrzennych

Trend przestrzenny definuje się [20] jako regularną zmianę co najmniej jednego atrybutu nieprzestrzennego obiektów wraz z oddalaniem się od innego obiektu. Odkrywanie trendów sprowadza się zazwyczaj do analizy regresji, gdzie odległość od danego obiektu jest zmienną niezależną, natomiast różnica wartości atrybutów do obserwacji - zmienną zależną.

Trendy dzieli się na globalne i lokalne. Pierwsze wskazują na zwiększanie (bądź zmniejszanie) wartości obserwowanych atrybutów przy rozpatrywaniu wszystkich obiektów znajdujących się na ścieżkach wychodzących z punktu początkowego. Typowym przykładem jest wzrost bezrobocia wraz z oddalaniem się od centrum miast. Trend lokalny jest reprezentowany przez pojedyncze ścieżki wykazujące inny kierunek zmian na danym atrybucie niż na sąsiednich ścieżkach.

2.2.4 Przypadki osobliwe w danych przestrzennych

Czasem w danych przestrzennych można znaleźć obiekty, których atrybuty nieprzestrzenne są niespójne z innymi obserwacjami dokonanymi w ich otoczeniu. Noszą one miano *przypadków osobliwych* [21].

Wyszukiwanie takich zjawisk jest trudne, szczególnie gdy istnieje więcej atrybutów nieprzestrzennych - odwzorowanie ich w n-wymiarowej przestrzeni może skutkować *przekleństwem wielowymiarowości* (ang. *curse of dimensionality*) [22], utrudnionym rozróżnianiem obiektów podobnych do siebie.

2.2.5 Asocjacje przestrzenne

Problem odkrywania asocjacji został pierwszy raz zdefiniowany w pracy [23] i w ogólności polega na analizie dostępnych transakcji (zbiorów obiektów, np. koszyka zakupów) oraz wykryciu występujących w nich regularności występowania elementów (typu: klient, który kupując bułki wybrał także masło).

Najczęściej reguły charakteryzuje się miarą *wsparcia* (ang. *support*) i *ufności* (ang. *confidence*). Pierwsza z nich wyraża stosunek występowania transakcji zawierającą lewą i prawą stronę reguły do ilości wszystkich transakcji. Ufność z kolei wskazuje na procentowy udział transakcji zawierających lewą i prawą stronę reguły we wszystkich transakcjach, które zawierają jego lewą stronę (jest to tzw. prawdopodobieństwo warunkowe). W celu ograniczenia ilości wykrytych wzorców wprowadzono także pojęcie *zbioru częstego* (ang. *frequent itemsets*) - zbioru elementów, dla

których wyznaczone wsparcie przekracza pewien ustalony przez użytkownika próg minimalnego wsparcia.

Z pracy [23] pochodzi także popularny algorytm wykorzystywany w wielu metodach odkrywania asocjacji i kolokacji, czyli metoda *Apriori*. Wykorzystuje on ważną cechę miary wsparcia, jaką jest *antymonotoniczność*. Wynika z niej, że zbiór może być zbiorem częstym tylko w przypadku, kiedy jego podzbiory są również zbiorami częstymi.

Na początku algorytmu generowane są jednoelementowe zbiory częste. Następnie iteracyjnie wykonywane są następujące kroki:

- tworzenie zbiorów częstych $(i+1)$-elementowych na podstawie zbiorów o długości i,
- filtrowanie zbiorów kandydujących w oparciu o miarę wsparcia,
- dodanie kandydatów do zbioru wynikowego.

Generowanie kandydatów polega na łączeniu wszystkich par zbiorów częstych o identycznych elementach początkowych, a następnie usuwaniu tych, które nie są zbiorami częstymi w oparciu o własność antymonotoniczności. Algorytm kończy się, gdy zbiór kandydatów będzie pusty.

W celu dostosowania metody odkrywania asocjacji do danych przestrzennych wprowadzono pojęcie *przestrzennej reguły asocjacyjnej* [24]. Zakłada ona istnienie predykatów przestrzennych (mogących wyrażać informacje o odległości czy kierunku), które mogą występować zarówno w części warunkującej (poprzedniku), jak w warunkowanej (następniku). Następnie podczas procesu odkrywania przestrzennych reguł asocjacyjnych dane umieszczone w ciągłej przestrzeni są zamieniane na zbiór transakcji. Metoda ta jest zaliczana do modelu zorientowanego na cechę referencyjną [5]. Istnieje też inne podejście, zwane *odkrywaniem zbiorów częstych klas sąsiadów*, opisane w pracy [25].

2.2.6 Kolokacje przestrzenne

Przedstawiony w pracy [4] problem *odkrywania przestrzennych reguł kolokacyjnych* powstał w odpowiedzi na niedoskonałości asocjacji (w szczególności konieczność wyboru cechy referencyjnej) i zakłada istnienie równorzędnych cech przestrzennych.

Praca wprowadza pojęcie *wzorca kolokacji przestrzennej* (zwanego także kolokacją przestrzenną lub krócej - kolokacją), zbioru cech przestrzennych, których instancje często występują we wzajemnym sąsiedztwie [5]. Stanowi on swego rodzaju odpowiednik zbiorów częstych w asocjacjach przestrzennych. Również miara wsparcia została zastąpiona przez *miarę powszechności* (ang. prevalence), które eliminują wymaganie wiedzy o transakcjach.

Kolokacje przestrzenne są przykładem modelu zorientowanego na zdarzenie (ang. *event-centric model*).

2.3 Odkrywanie kolokacji przestrzennych

Niniejszy rozdział zawiera opisy i definicje pojęć niezbędnych do zrozumienia algorytmu zawartego w rozdziale 3.

2.3.1 Cecha przestrzenna

Kluczową kwestią w procesie odkrywania kolokacji jest odpowiednia klasyfikacja obiektów występujących w bazie danych. Każdy zbiór danych przestrzennych, oprócz informacji o lokalizacji obiektu i opisujących go danych nieprzestrzennych powinien zawierać także właściwość pozwalającą na sklasyfikowanie danego obiektu do określonej klasy. Takie przypisanie nazywane jest cechą przestrzenną (ang. spatial feature) lub rzadziej klasą obiektu (ang. object class).

Jako typowy przykład cechy przestrzennej można podać etykietę przypisaną do obiektu na mapie (np. kościół, szkoła, strzelnica). Pozwala ona na jednoznaczne określenie własności przestrzeni w punkcie, gdzie znajduje się obiekt.

2.3.2 Podstawowe definicje

Definicja 1 (Instancja cechy przestrzennej) *Niech f będzie cechą przestrzenną. Mówimy, że obiekt x jest instancją cechy przestrzennej f, wtedy i tylko wtedy, gdy obiekt x jest typu f oraz jest opisany przez lokalizację i identyfikator.*

Definicja 2 (Wzorzec i instancja kolokacji) *Załóżmy F jako zbiór cech przestrzennych $F = \{f_1, f_2, ..., f_m\}$, a $FI = FI^{f_1} \cup FI^{f_2} \cup ... \cup FI^{f_m}$ niech będzie zbiorem ich instancji. Niech $>_F$ oznacza dowolną relację porządku zdefiniowaną dla zbioru F. Niech f_i oznacza i-tą cechę przestrzenną (ze względu na relację $>_F$), zatem $\forall i, j \in 1, ..., m \; f_i <_F f_j \Leftrightarrow i < j \wedge f_i, f_j \in F$. Mając daną relację sąsiedztwa R (zwrotną i przechodnią) mówimy, że wzorzec kolokacji przestrzennej (w skrócie "kolokacja") jest podzbiorem cech przestrzennych $c \subseteq F$, których instancje $I \subseteq FI$ tworzą klikę ze względu na relację R. Zbiór wszystkich instancji kolokacji przestrzennej c jest oznaczany przez CI^c. Przez długość kolokacji należy rozumieć liczbę elementów w zbiorze cech przestrzennych, który tworzy tę kolokację.*

Definicja 3 (Sąsiedztwo) *Mające daną zwrotną i symetryczną relację sąsiedztwa R, sąsiedztwem lokalizacji l nazywamy zbiór lokalizacji $L = \{l_1, l_2, ..., l_n\}$, gdzie l_i jest sąsiadem l, tzn. zachodzi $R(l, l_i)$ $\forall i \in 1, ..., n$.*

Przykład TODO (oprzeć na przykładzie chińczyków?)

2.3.3 Miary kolokacji

Definicja 4 (Współczynnik uczestnictwa) *Współczynnik uczestnictwa (ang. participation ratio) cechy f i w kolokacji c jest równy procentowemu udziałowi wszystkich instancji cechy f i w instancjach kolokacji c:*

$$pr(f_i, c) = \frac{|\pi^{f_i}(CI^c)|}{FI^{f_i}} \tag{1}$$

gdzie $\pi^{f_i}(CI^c)$ oznacza projekcję relacyjną zbioru instancji CI^c względem cechy f_i (z usuwaniem duplikatów).

Definicja 5 (Indeks uczestnictwa) *Indeks uczestnictwa (ang. participation index) kolokacji c jest równy najmniejszemu ze współczynników uczestnictwa wyznaczonych dla każdej cechy przestrzennej $f_i \in c$:*

$$pi(c) = min_{f_i \in c} pr(f_i, c) \tag{2}$$

Indeks uczestnictwa najczęściej określany jest w literaturze mianem miary powszechności lub krótko powszechnością kolokacji.

Definicja 6 (Maksymalny wzorzec kolokacji przestrzennej) *Niech będzie dana wartość min_prev oznaczająca pewien minimalny próg powszechności. Jeżeli* $c = \{f_1, ..., f_m\}$ *jest kolokacją powszechną (tzn.* $pi(c) \geqslant min\_prev$*) i nie istnieje żaden nadzbiór c taki, że powszechność dla tego nadzbioru jest równa co najmniej min_prev, kolokacja c nazywana jest kolokacją maksymalną.*

2.3.4 Problem

Definicja 7 (Reguła kolokacyjna) *Reguła kolokacyjna to reguła postaci* $c_1 \rightarrow c_2(p, cp)$*, gdzie* $c_1 \subseteq F$*,* $c_2 \subseteq F$ *i* $c_1 \cup c_2 = \emptyset$*. Potencjalna użyteczność reguły może być mierzona przy pomocy jej powszechności p oraz prawdopodobieństwa warunkowego cp.*

Definicja 8 (Prawdopodobieństwo warunkowe) *Prawdopodobieństwem warunkowym* $cp(c_1, c_2)$ *reguły kolokacyjnej* $c_1 \rightarrow c_2$ *nazywamy stosunek liczby instancji wzorca c 1 w sąsiedztwie instancji wzorca* c_2 *do liczby wszystkich instancji wzorca* c_1*:*

$$cp(c_1, c_2) = \frac{|\pi^{c_1}(CI^{c_1 \cup c_2})|}{CI^{c_1}} \tag{3}$$

gdzie $\pi^{c_1}(CI^{c_1 \cup c_2})$ *oznacza projekcję relacyjną instancji wzorca* $CI^{c_1 \cup c_2}$ *względem wzorca* c_1 *(z usuwaniem duplikatów).*

Definicja 9 (Problem odkrywania kolokacji) *Problem odkrywania kolokacji przestrzennych jest zdefiniowany w następujący sposób. Mając dane:*

- *zbiór cech przestrzennych* $F = \{f_1, f_2, ..., f_m\}$
- *zbiór obiektów* $FI = FI^{f_1} \cup FI^{f_2} \cup ... \cup FI^{f_m}$ *, gdzie* $FI^{f_i}, (0 < i \leqslant m)$ *jest zbiorem instancji cechy* f_i*, przy czym każda instancja jest opisana przez lokalizację i identyfikator,*
- *symetryczną i zwrotną relację sąsiedztwa R,*
- *próg minimalnej powszechności min_prev oraz próg minimalnego prawdopodobieństwa warunkowego min_cond,*

znajdź wszystkie poprawne reguły kolokacyjne z powszechnością nie mniejszą niż min_prev i prawdopodobieństwem warunkowym nie mniejszym niż min_cond.

2.4 Przegląd algorytmów odkrywania wzorców kolokacji przestrzennych

W tym podrozdziale zostaną zaprezentowane skrótowo najważniejsze algorytmy odkrywania kolokacji przestrzennych.

2.4.1 Co-location Miner

Wraz z wprowadzeniem pojęcia kolokacji autorzy pracy [4] zaprezentowali także podstawowy obecnie algorytm rozwiązujący problem odkrywania wzorców kolokacji przestrzennych, zwany *Co-location Miner*. W algorytmie tym wyróżnia się następujące fazy:

- generowanie kandydatów na kolokacje przestrzenne (o długości i),
- wyznaczanie instancji dla wygenerowanych kandydatów,
- usuwanie kandydatów, których powszechność wynosi mniej niż przyjęty próg minimalnej powszechności.

Pozostali kandydaci trafiają do zbioru wynikowego, a następnie na ich podstawie są tworzone reguły kolokacyjne. Same reguły również podlegają filtracji - usuwane są te reguły, których prawdopodobieństwo warunkowe jest poniżej określonego progu.

W następnej iteracji algorytm wykonuje dokładnie te same kroki, przy czym generowani kandydaci są o długości o jeden większej. Całość kończy się, gdy nie jest możliwe już wygenerowanie nowych kandydatów.

2.4.2 Multiresolution Co-location Miner

Korzystanie z oryginalnego algorytmu *Co-location Miner* wiąże się niestety z dużymi kosztami obliczeniowymi, głównie ze względu na pracochłonny krok generowania kandydatów na kolokacje. Dlatego też niedługo później w pracy [9] autorzy zaproponowali drobną modyfikację oryginalnego algorytmu, dodając dodatkowy krok filtrowania w opraciu o przybliżoną reprezentację zbioru wejściowego.

W algorytmie *Multiresolution Co-location Miner* zbiór wejściowy zostaje podzielony na obszary (mniejsze fragmenty). Zanim rozpocznie się faza wyznaczania instancji dla wygenerowanych kandydatów, nastepuje szacowanie ich powszechności na podstawie sąsiadujących instancji cech przestrzennych w ramach obszarów. W przypadku zbyt niskiej wartości szacowanej powszechności kandydata, można go wykluczyć z dalszego przetwarzania i tym samym oszczędzić zasoby niezbędne na wyznaczenie jego instancji.

Dalsze kroki przebiegają identycznie jak w przypadku oryginalnego *Co-location Miner*.

2.4.3 Joinless

Celem autorów pracy [10] było stworzenie algorytmu, który omijałby konieczność tworzenia kosztownych połączeń przestrzennych na etapie wyznaczania instancji kandydatów na kolokacje (tak jak np. w przypadku rodziny algorytmów *Co-location Miner*). Nosi on nazwę algorytmu bezpołączeniowego (ang. *joinless*).

Główną różnicą w porównaniu do wcześniejszych algorytmów jest sposób generowania instancji kolokacji. Są one generowane na podstawie sąsiedztw typu gwiazda - zbiorów obiektów, w którego skład wchodzi rozpatrywany obiekt oraz jego sąsiedzi posiadający większą cechę przestrzenną. Wyznacza się je na podstawie oddzielnych algorytmów (np. *plane sweep*), lub korzysta z specjalnych struktur ułatwiających wykrywanie sąsiadów typu *R-drzewo*.

Wygenerowane instancje muszą zostać dodatkowo zweryfikowane (poprawne instancje powinny być kliką, czego nie gwarantuje sąsiedztwo typu gwiazda), a następnie - podobnie jak w algorytmie *Multiresolution Co-location Miner* - dokonuje się ich wstępnego filtrowania pod kątem progu minimalnej powszechności.

2.4.4 iCPI-tree

Drzewo iCPI (*improved Co-location Pattern Instance*, [12]) stanowi zmodyfikowaną odmianę drzewa CPI zawartego w pracy [11]. Struktura ta zawiera informacje o wszystkich zachodzących relacjach sąsiedztwa.

iCPI-tree posiada następującą strukturę:

- Poziom 1 - korzeń drzewa (oznaczony etykietą *NULL*),
- Poziom 2 - cechy elementów centralnych, czyli cechy przestrzenne obiektów centralnych *sąsiedztw typu gwiazda*;
- Poziom 3 - instancje elementów centralnych, dla których ma zostać przechowana informacja o sąsiadach;
- Poziom 4 - cechy sąsiadów,
- Poziom 5 - instancje sąsiadów.

Sąsiedzi uporządkowani są według rosnącej cechy przestrzennej, a w przypadku instancji tej samej cechy - zgodnie z rosnącym identyfikatorem. Takie uporządkowanie nosi nazwę *uporządkowanego zbioru sąsiadów*.

Powyższa struktura drzewiasta jest wykorzystana w algorytmie w celu generowania instancji coraz dłuższych kandydatów w kolejnych iteracjach. Dokonuje się tego poprzez systematyczną ich rozbudowę o kolejne elementy. Na początku wszystkie instancje są jednoelementowe, a w kolejnych iteracjach są one rozbudowywane poprzez wyszukiwanie sąsiadów z odpowiednią cechą i weryfikowane (należy sprawdzić, czy nowo dodany obiekt do instancji jest sąsiadem każdego z obiektów należących do tej instancji).

Pozostałe kroki algorytmu (generowanie kandydatów i reguł, filtrowanie według powszechności) są podobne jak w metodach *Co-location Miner* i *joinless*.

3 Algorytm

Niniejszy rozdział ma za zadanie przybliżenie algorytmu będącego tematem tej pracy - metody odkrywania maksymalnych kolokacji przestrzennych w oparciu o graf rzadki i skondensowane drzewo instancji (ang. *sparse-graph and condensed tree-based maximal co-location algorithm*) przedstawionej w pracy [6].

Poszczególne kroki pierwotnego algorytmu *SGCT* zostaną opisane w kolejnych podrozdziałach. Szczegóły implementacji wraz z zaproponowanymi usprawieniami znajdują się w Rozdziale 4.

3.1 Generowanie tabeli instancji kolokacji o rozmiarze 2

Pierwszy krok algorytmu jest podobny do metody *Co-location Miner* i polega na wygenerowaniu 2-elementowych kandydatów na kolokacje.

Kolokacje o rozmiarze 2 tworzone są na podstawie wygenerowanych w oparciu o cechy przestrzenne jednoelementowych kolokacji. Nie jest do tego wykorzystywana jednak metoda *Apriori*, ponieważ udowodniono w pracy [4], że dla kandydatów dwuelementowych lepszą wydajność można uzyskać w oparciu o algorytm *spatial join*. Wykorzytany został zatem algorytm *sweeping-based spatial join* [26] z dodatkową modyfikacją, usuwającą pary instancji o tej samej cesze przestrzennej.

Przykład 1 *Przykładowe zapytanie tworzące kandydatów na kolokacje o rozmiarze 2 przestawia się następująco:*
select p', p''
from $\{p_1, ..., p_{12}\}p'$, $\{p_1, ..., p_{12}\}p''$
where $p'.feature \neq p''.feature$, p'‘p'', $(p', p'') \in R$

Na podstawie wygenerowanych kolokacji oraz tzw. *progu odległości* (ang. *distance threshold*) tworzona jest dwuwymiarowa tablica z haszowaniem (ang. *hash table*). Jest ona indeksowana cechami przestrzennymi. Każdy element tablicy zawiera wskaźnik do listy zawierającej instancje kandydatów o odpowiadających indeksom cechach przestrzennych. Instancja zostanie dodana do tej listy tylko wtedy, gdy odległość między instancjami przekracza dopuszczalny próg odległości między nimi.

Przykład 2 *Zapytanie* $InsTable_2(A, B)$ *zwróci listę kandydatów* (A_2, B_2), (A_3, B_1), *itd. Nie zwróci* (A_2, B_{100000}), *gdyż odległość między instancjami nie przekracza dopuszczalnego progu odległości.*

3.2 Obliczanie miary powszechności

Również krok obliczania powszechności dla kandydatów nie różni się od tego znanego z *Co-location Miner*.

Dla każdego kandydata w tabeli wyliczany jest współczynnik uczestnictwa (ang. *participation index*). Dokonuje się tego poprzez wybieranie wszystkich unikalnych instancji cech przestrzennych, która są ujęte w danej kolokacji. Następnie zgodnie z definicją miary powszechności z tabeli instancji są usuwani kandydaci, dla których obliczona miara powszechności jest mniejsza niż zadany próg minimalnej powszechności *min_prev* [4].

3.3 Generowanie kandydatów na kolokacje maksymalne

Krok ten wprowadza nową strukturę, zwaną *grafem kolokacji o rozmiarze 2* (ang. *size-2 co-location graph*). Jego definicja brzmi następująco:

Definicja 10 (Graf kolokacji o rozmiarze 2) *Jeżeli przyjąć relacje sąsiedztwa między kolokacjami o rozmiarze 2 jako krawędzie $E = \{e_1, ..., e_u\}$, a cechy przestrzenne występujące w kolokacjach jako wierzchołki $V = \{v_1, ..., v_\lambda\}$, gdzie u i λ są odpowiednio liczbą krawędzi i liczbą wierzchołków, to graf kolokacji o rozmiarze 2 można zamodelować jako graf nieskierowany $G = (E, V)$, przechowywany w listowej strukturze danych uporządkowanej rosnąco. Zbiór N jest zbiorem sąsiedztw wierzchołków i definiuje się go następująco:*

$$N(v_i) = \{W | v_i, w \in E\} \tag{4}$$

Zadaniem tego kroku jest wyszukanie w takim grafie maksymalnych klik, określanych jako *kandydaci na maksymalne kolokacje* (ang. *candidate maximal co-location*).

Definicja 11 (Kandydat na maksymalną kolokację) *Kandydat na maksymalną kolokację C_m składa się z uporządkowanych cech przestrzennych o następujących właściwościach: każda para cech w C_m jest ze sobą połączona krawędzią, a żadne dodatkowe cechy nie mogą być dodane do C_m bez zachowania ich kompletnego połączenia.*

Autorzy pracy [6] udowodnili, że graf kolokacji o rozmiarze 2 można traktować jako graf rzadki. Umożliwia to korzystanie z algorytmu Brona-Kerboscha [28] do wyszukiwania maksymalnych klik w grafie nieskierowanym. Wprowadzone zostały do niego pewne modyfikacje uwzględniające rozproszenie grafu oraz wybieranie *pivotu* w celu usprawienia wyszukiwania kandydatów na kolokacje. Rozproszenie grafu opisuje się miarą *degeneracji grafu* [27]:

Definicja 12 (Degeneracja grafu) *Degeneracja grafu G jest najmniejszą wartością k, taką, że każdy niepusty podgraf G zawiera wierzchołki o stopniu co najwyżej k. Oznacza to, że wielkość maksymalnej kliki nie może przekroczyć $k + 1$.*

Pseudo-kod kroku wygląda następująco:

3.4 Proces odcinania

4 Implementacja

5 Testy efektywnościowe

6 Zakończenie

Bibliografia

[1] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From Data Mining to Knowledge Discovery in Databases. AI Magazine, 17:37–54, 1996.

[2] Ł. Stanisławowski. *Bogactwo i nędza narodów.* O'reilly, 2013.

[3] Harvey J. Miller and Jiawei Han. Geographic Data Mining and Knowledge Discovery. Taylor & Francis, Inc., Bristol, PA, USA, 2001

[4] S. Shekhar and Y. Huang. Discovering Spatial Co-location Patterns: A Summary of Results. In SSTD 2001, pages 236–256, 2001.

[5] Przetwarzanie zbiorów przestrzennych zapytan neksploracyjnych w srodowiskachzograniczonym rozmiarem pamiecioperacyjnej

[6] A fast space-saving algorithm for maximal co-location pattern mining

[7] CUDA by Example: An Introduction to General-Purpose GPU Programming, Jason Sanders, Edward Kandrot

[8] Professional CUDA C Programming, John Cheng, Max Grossman, Ty McKerche

[9] Shashi Shekhar and Yan Huang. The Multi-resolution Co-location Miner: A New Algorithm to Find Co-location Patterns in Spatial Dataset. Technical Report 02-019, University of Minnesota, 2002.

[10] Jin Soung Yoo and Shashi Shekhar. A Joinless Approach for Mining Spatial Colocation Patterns. IEEE Transactions on Knowledge and Data Engineering, 18(10):13231337, 2006.

[11] Lizhen Wang, Yuzhen Bao, Joan Lu, and Jim Yip. A New Join-less Approach for Co-location Pattern Mining. In Qiang Wu, Xiangjian He, Quang Vinh Nguyen, Wenjing Jia, and Mao Lin Huang, editors, Proceedings of the 8th IEEE International Conference on Computer and Information Technology (CIT 2008), pages 197–202, Sydney, July 2008. IEEE.

[12] Lizhen Wang, Yuzhen Bao, and Joan Lu. Efficient Discovery of Spatial Co-Location Patterns Using the iCPI-tree. The Open Information Systems Journal, 3(2):69–80,2009.

[13] Christopher Jones and Mark Hall. A Field Based Representation for Vague Areas Defined by Spatial Prepositions. In Proceedings of the Workshop on Methodologies and Resources for Processing Spatial Language at 6th Language Resources and Evaluation Conference (LREC 2008), 2008.

[14] Max J. Egenhofer and Robert Franzosa. Point-set topological spatial relations. International Journal of Geographic Information Systems, 5(2):161–174, 1991.

[15] Eliseo Clementini, Paolino Di Felice, and Peter van Oosterom. A small set of formal topological relationships suitable for end-user interaction. In Proceedings of the 3rd International Symposium on Advances in Spatial Databases (SSD 1993), pages 277-295, London, UK, UK, 1993. Springer-Verlag.

[16] Harvey J. Miller and Jiawei Han. Geographic Data Mining and Knowledge Discovery. Taylor & Francis, Inc., Bristol, PA, USA, 2001.

[17] John F. Roddick and Myra Spiliopoulou. A Bibliography of Temporal, Spatial and Spatio-Temporal data Mining Research. ACM SIGKDD Exploration Newsletter, 1(1):34–38, 1999.

[18] Martin Ester, Hans-Peter Kriegel, and Jörg Sander. Spatial Data Mining: A Database Approach. In Proceedings of the 5th International Symposium on Advances in Spatial Databases (SSD 1997), pages 47–66, London, UK, UK, 1997. Springer-Verlag.

[19] John R. Quinlan. Induction of Decision Trees. Machine Learning, 1(1):81–106, March 1986.

[20] Martin Ester, Alexander Frommelt, Hans-Peter Kriegel, and Jörg Sander. Algorithms for characterization and trend detection in spatial databases. In Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining (KDD 1998), pages 44–50, 1998.

[21] Shashi Shekhar and Sanjay Chawla. Spatial Databases: A Tour. Prentice Hall, 2003

[22] Richard E. Bellman. Adaptive control processes - A guided tour. Princeton University Press, Princeton, New Jersey, U.S.A., 1961.

[23] Rakesh Agrawal and Ramakrishnan Srikant. Fast Algorithms for Mining Association Rules in Large Databases. In Proceedings of the 20th International Conference on Very Large Data Bases (VLDB 1994), pages 487–499, San Francisco, 1994. Morgan Kaufmann Publishers Inc.

[24] Krzysztof Koperski and Jiawei Han. Discovery of Spatial Association Rules in Geographic Information Databases. In Max J. Egenhofer and John R. Herring, editors, Proceedings of the 4th International Symposium on Advances in Spatial Databases (SSD 1995), volume 951 of Lecture Notes in Computer Science, pages 47–66. Springer Berlin Heidelberg, 1995

[25] Yasuhiko Morimoto. Mining Frequent Neighboring Class Sets in Spatial Databases. In Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2001), pages 353–358, New York, NY, USA, 2001. ACM.

[26] L. Arge, O. Procopiuc, S. Ramaswamy, T. Suel, and J. Vitter. Scalable Sweeping- Based Spatial Join. In Proc. of the Int'l Conference on Very Large Databases, 1998.

[27] Eppstein, D. , Löffler, M. , i Strash, D. (2010). Listing all maximal cliques in sparsegraphs in near-optimal time. In O. Cheong, K. Y. Chwa, & K. Park (Eds.), 21st international symposium on algorithms and computation (pp. 403–414). Berlin, Germany: Springer-Verlag.

[28] Bron, C., & Kerbosch, J. (1973). Algorithm 457: Finding all cliques of an undirected graph. Communications of the ACM, 16 , 575–577. doi: 10.1145/362342.362367.

A Dodatek A

B Dodatek B