

Marcin GORAWSKI, Paweł JURECZEK
Politechnika Śląska, Instytut Informatyki

ZAPYTANIA WZORCA DO TRAJEKTORYJNEJ HURTOWNI DANYCH

Streszczenie. Artykuł podejmuje temat zapytań wzorca dla Trajektoryjnej Hurtowni Danych, TrDW. W ramach zapytań wzorca został zaprezentowany sposób przekształcania trajektorii obiektów do postaci sekwencji regionów oraz eksploracji tak uzyskanych sekwencji z użyciem funkcji porównujących. Przedstawione funkcje porównujące zostały podzielone na dwie grupy. Pierwsza grupa umożliwia uzyskanie informacji na temat konkretnego, zdefiniowanego przez użytkownika wzorca. Druga natomiast służy pozyskiwaniu informacji podsumowujących dotyczących wszystkich analizowanych sekwencji trajektorii. Informacje uzyskane w wyniku użycia drugiej grupy mogą również posłużyć, jako dane wejściowe grupy pierwszej. W artykule przedstawiono również wpływ różnych parametrów, wykorzystywanych podczas przekształcania trajektorii, na rozmiar składowanych agregatów.

Słowa kluczowe: eksploracja danych, region zainteresowania, agregaty, zapytania wzorca, trajektoryjna hurtownia danych

PATTERN QUERIES FOR TRAJECTORY DATA WAREHOUSES

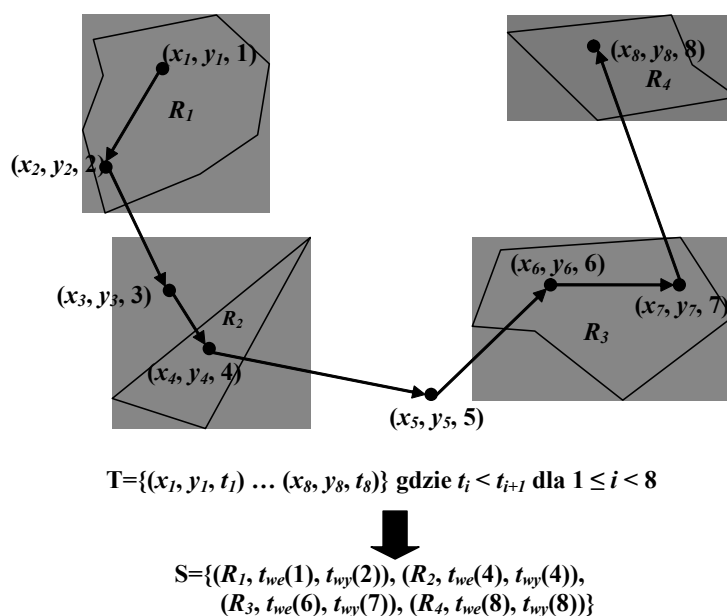
Summary. This paper presents the query model for Trajectory Data Warehouse. The pattern queries in this paper present a way of transforming object trajectories into region sequences, as well as exploring such sequences using comparison methods. Such comparison methods are divided into two groups. The first group makes it possible to collect information about a particular user-defined pattern. The second involves obtaining information summary of all analyzed trajectory sequences. Information obtained as a result of using the second group may be also used as input data for the first group. We also show the impact of different parameters of transformation of trajectories on the size of stored aggregates.

Keywords: data exploration, region of interest, aggregates, pattern queries, trajectory data warehouse

1. Wprowadzenie

Wzorce służą wykrywaniu powtarzających się sytuacji w tym np. kolejnego odwiedzania tych samych miejsc w określonym czasie. Przykładem może być tu sekwencja odwiedzania przez turystów największych atrakcji na danym obszarze. Podobna sytuacja występuje w przypadku analizy zachowań pieszych lub obserwacji przeglądanych przez użytkowników stron internetowych. W przypadku trajektorii wykrycie różnych interesujących zależności przestrzennych byłoby trudne do realizacji z punktu widzenia analizy tylko i wyłącznie współrzędnych, dlatego wzorce trajektorii bazują zazwyczaj na regionach zainteresowania RoI. Regiony zainteresowania mogą być określone na podstawie właściwości geograficznych np. sieci dróg lub popularnych miejsc często odwiedzanych przez ludność.

W przypadku, gdy mamy zdefiniowane regiony zainteresowania (znamy ich kształty), proces wyszukiwania wzorców składa się zazwyczaj z kilku etapów. Najczęściej najpierw przetwarzamy punkty odzwierciedlające trajektorie obiektów na określone sekwencje regionów (mapujemy punkty na regiony – zobacz rys. 1), po czym w procesie eksploracji danych analizujemy konkretne zachowania. Gdy regiony zainteresowania są nieznane możemy je wyznaczyć automatycznie. Automatyczne wyznaczanie RoI opiera się na wyznaczeniu najczęściej odwiedzanych przez obiekty obszarów, co pociąga za sobą konieczność przeliczenia trajektorii.



Rys.1. Przekształcenie trajektorii do postaci RoI

Fig.1. Mapping trajectories into RoI

W trakcie mapowania trajektorii na RoI, każdemu regionowi jest przypisywany czas t_{we} wejścia do i czas t_{wy} wyjścia z regionu. Na podstawie t_{we} i t_{wy} można porównywać czas pobytu obiektu w regionie oraz czas przejścia pomiędzy konkretnymi regionami.

W literaturze możemy odnaleźć kilka odmiennych podejść do zagadnień zapytań wzorca. W [1] autorzy zastępują punkty wyznaczające położenie obiektu albo „postojami”, w których obiekt przebywał określony czas, albo „ruchami” pomiędzy nimi. Typ postojów zależy od wymagań stawianych aplikacji tzn. mogą to być hotele w przypadku aplikacji turystycznej lub sygnalizacja świetlna w przypadku aplikacji kontroli ruchu.

W [2] przedstawiono sposoby znajdowania RoI składających się na sekwencje, w przypadku, gdy nie mamy z góry podanych informacji o miejscach zainteresowania i ich kształtach. Jednym ze sposobów jest wyznaczanie miejsc, w których jest największa gęstość punktów trajektorii. W artykule tym zostały również wprowadzone definicje dotyczące wzorców i ich eksploracji. Tak więc, zgodnie z [2], wzorce dzielimy na FSP (ang. *Frequent Sequential Pattern*) oraz TAS (ang. *Temporally Annotated Sequence*). TAS rozszerza sekwencyjne wzorce FSP o czas przejścia pomiędzy elementami sekwencji.

TAS można przedstawić w następujący sposób:

$$T = s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} \dots \xrightarrow{\alpha_{n-1}} s_n \quad (1)$$

gdzie s_0, \dots, s_n jest zbiorem elementów sekwencji, a a_0, \dots, a_{n-1} jest zbiorem czasów przejścia pomiędzy tymi elementami.

W trakcie wyszukiwania wzorców możemy również uwzględnić czas, w którym sekwencja miała miejsce oraz czas pobytu obiektu w regionie.

Dla Trajektoryjnej Hurtowni Danych (ang. *Trajectory Data Warehouse*, TrDW) [6] zaprojektowaliśmy moduł zapytań wzorca charakteryzujący się następującymi cechami: (1) cała mapa dzielona jest na określoną liczbę rozłącznych regionów o takim samym rozmiarze; (2) na podstawie punktów trajektorii, wygenerowania za pomocą generatora z [3] (dla miasta Oldenburg), wyznaczane są gęste regiony; (3) każda sekwencja kolejnych punktów trajektorii obiektu jest mapowana do sekwencji gęstych regionów – zapisywane są też dodatkowe informacje takie jak czas przejścia pomiędzy elementami sekwencji, czas pobytu obiektu w regionie, agregat punktów w regionie itd.; (4) przeprowadzana jest eksploracja danych.

Organizacja kolejnych rozdziałów jest następująca. W rozdziale 2 pokazano sposób wyznaczania sekwencji regionów na podstawie trajektorii, po czym w rozdziale 3 przedstawiono metody ich eksploracji. Rozdział 4 prezentuje wyniki przeprowadzonych eksperymentów. Rozdział 5 stanowi podsumowanie artykułu.

2. Wyznaczanie sekwencji regionów

Proces wyznaczania sekwencji regionów możemy podzielić na dwa etapy. W pierwszym etapie znajdujemy regiony (reprezentowane są przez rozłączne prostokątne komórki podziału

mapy), które np. posiadają gęstość (liczność) punktów trajektorii większą niż gęstość zadana przez użytkownika. Wyznaczone w ten sposób regiony są nazywane gęstymi. W drugim etapie regiony te wykorzystywane są przez algorytm *ComputeSequence* do wyznaczenia sekwencji regionów.

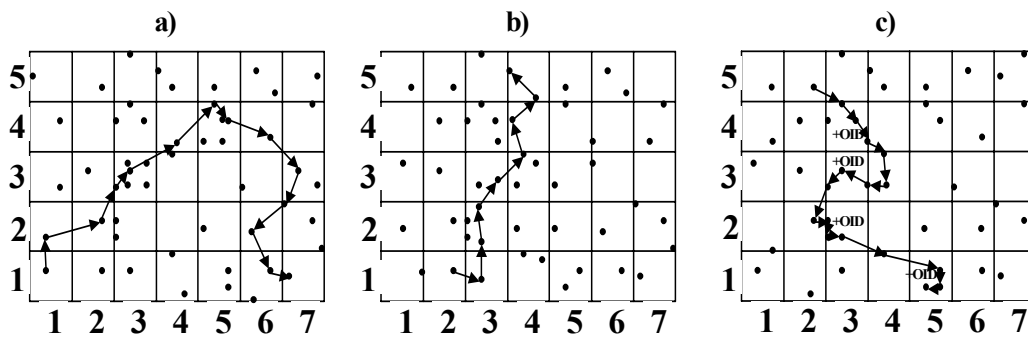
Wyszukanie istotnych ze względu na analizę regionów (etap pierwszy) może być realizowane na 3 różne sposoby. Pierwsza z metod (rys. 2a) zakłada, że obiekty wysyłają informacje o swoim położeniu w równych odstępach czasu. W konsekwencji, długość segmentu trajektorii może być różna w zależności np. od istniejącej infrastruktury drogowej. Przykładowo, obiekt poruszający się po autostradzie może przebyć większy dystans niż obiekt poruszający się w obszarze zabudowanym. Jednocześnie odległość euklidesowa pomiędzy dwoma kolejnymi punktami trajektorii nie musi zawsze odpowiadać rzeczywistej odległości przebytej przez obiekt. W metodzie tej punkty trajektorii są zliczane w regionach.

W metodzie z rys. 2b. położenie obiektu aktualizowane jest w chwili przebycia określonej odległości. Oznacza to, że czas przejścia pomiędzy dwoma kolejnymi punktami może być różny i zależeć, podobnie jak w poprzedniej metodzie, od istniejącej infrastruktury drogowej. Metoda ta sumuje czasy pobytu obiektów w danym regionie.

Ostatnia z metod zlicza kolejne punkty trajektorii obiektu w regionie, a następnie, jeżeli uzyskana liczba punktów trajektorii danego obiektu jest nie mniejsza od zadanego przez użytkownika progu *minCount*, do listy obiektów regionu dodawane jest ID obiektu (liczba unikalnych obiektów w regionie odpowiada gęstości regionu). Wartość progu użytkownika *minCount* = 1 sprawia, że w regionie zliczany jest każdy obiekt, który w nim przebywał. Ponadto, w każdym z regionów każdy z obiektów może zostać zliczony tylko raz. Dzięki zastosowanemu podejściu już na etapie pomiaru gęstości regionów możemy pominąć punkty trajektorii, które są nieistotne. Przykładowo, możemy pomijać pojedyncze punkty trajektorii w regionie, które występują na granicy regionów. Na rys. 2c przyjęto, że ID obiektu jest dodawane do listy danego regionu, jeśli liczba kolejnych punktów trajektorii obiektu jest większa od dwóch – wystąpienie takiej sytuacji oznaczono za pomocą symbolu „+OID”.

Ponadto, można zauważyć, że algorytmy dwóch pierwszych metod są do siebie podobne, a główna różnica polega na tym, że w pierwszym przypadku zliczamy punkty trajektorii przypadające na danych region, w drugim natomiast sumujemy czasy pobytów obiektów w regionie.

Użycie konkretnej metody zależy od sposobu akwizycji pomiarów.



Rys. 2. Trzy różne podejścia służące wyznaczeniu gęstych regionów

Fig. 2. Three different approaches to determining the dense regions

Dla każdej z powyższych metod wyznaczone gęstości regionów mogą zostać dodatkowo przesunięte w celu wyeliminowania z analizy mniej istotnych obszarów – przesuwany jest minimalny próg gęstości decydujący o tym, czy dany region jest gęsty.

Algorytm *ComputeSequence* zlicza wystąpienia kolejnych punktów trajektorii obiektu w odpowiednich regionach. Region jest uznawany za miejsce postoju na trasie obiekt, jeśli w procesie wyznaczania gęstości punktów trajektorii został on uznany za odpowiednio gęsty oraz obiekt odpowiednio długo w nim przebywał – użytkownik określa minimalny czas pobytu obiektu w regionie lub minimalną liczbę kolejnych punktów trajektorii obiektu w regionie za pomocą parametru *StayTime*. Każdy z regionów sekwencji może przyjąć jedno z 4 oznaczeń: *Start*, *Stop*, *End*, *StartStop*. Znacznik *Start* oznacza, że region jest regionem, w którym obiekt rozpoczął ruch. Oznaczenie *Stop* mówi, że region został wyznaczony pomiędzy regionem *Start* i *End*. *End* jest oznaczeniem ostatniego regionu trajektorii (jeśli jednak wystąpił już w sekwencji region *Start*), a znacznik *StartStop* jest stosowany na określenie regionu, w którym znajdowała się cała trajektoria obiektu ruchomego. Ponadto, każdy z regionów (stopów) sekwencji zawiera informację o środku ciężkości kolejnych punktów trajektorii przypadających na dany region – agregat punktów w regionie.

Zaznaczmy, że zwiększenie ograniczenia czasowego *StayTime* powoduje zmniejszenie dokładności odwzorowania trajektorii przez sekwencję (zmniejsza się też średnia długość sekwencji regionów). Z kolei, zmniejszenie parametru *StayTime* zwiększa średnią długości sekwencji regionów oraz dokładność odwzorowania (zwiększa się również liczba wynikowych danych).

Ze względu na długi czas wyznaczania sekwencji regionów dla dużego zbioru trajektorii w TrDW została wprowadzona możliwość materializacji wyników, która jest realizowana za pomocą omówionej poniżej tabeli SEQUENCES.

3. Eksploracja sekwencji regionów

Wyznaczony zbiór sekwencji regionów możemy poddać analizie. W przypadku gdy potrzebujemy podsumowania, dotyczącego wszystkich wyznaczonych sekwencji regionów, możemy skorzystać z algorytmów typu Apriori [4, 5]. Algorytmy te pozwalają wyznaczyć zbiór odpowiednio częstych podsekwencji wszystkich analizowanych sekwencji. W przypadku zaprojektowanej TrDW mogą to być najczęstsze sekwencje regionów trajektorii taksówek. Warto zanotować, że do wyznaczania częstości kandydatów została użyta m.in. – opisana dalej – funkcja SelectiveSearch.

Odnalezione za pomocą algorytmów typu apriori częste sekwencje możemy poddać szczegółowej analizie za pomocą funkcji porównujących. Klasyfikacja podstawowych funkcji porównujących została przedstawiona w tabeli 1. Dwie ostatnie kolumny tabeli odpowiadają charakterystycznym własnościom funkcji porównujących, ich znaczenie jest następujące: druga kolumna tabeli mówi o tym, czy kolejność elementów wzorca musi być zachowana w analizowanej sekwencji; trzecia kolumna natomiast określa, czy kolejne elementy wzorca muszą wystąpić w sekwencji regionów bezpośrednio po sobie.

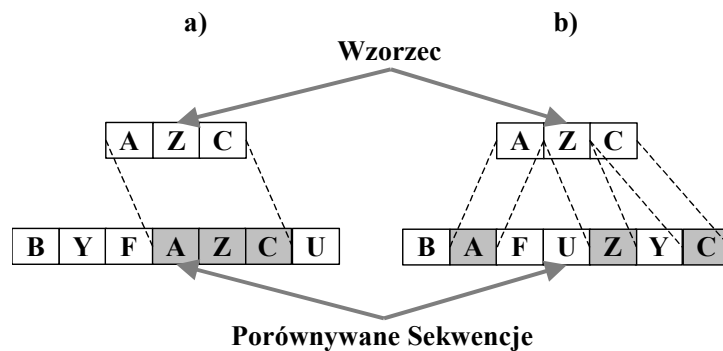
W przypadku funkcji MarkedSearch istnieje możliwość określenia, czy wybrane elementy wzorca mają występować po sobie w porównywanych sekwencjach – funkcja MarkedSearch jest połączeniem funkcji ExactSearch i SelectiveSearch.

Tabela 1

Funkcje porównujące wzorzec z sekwencją

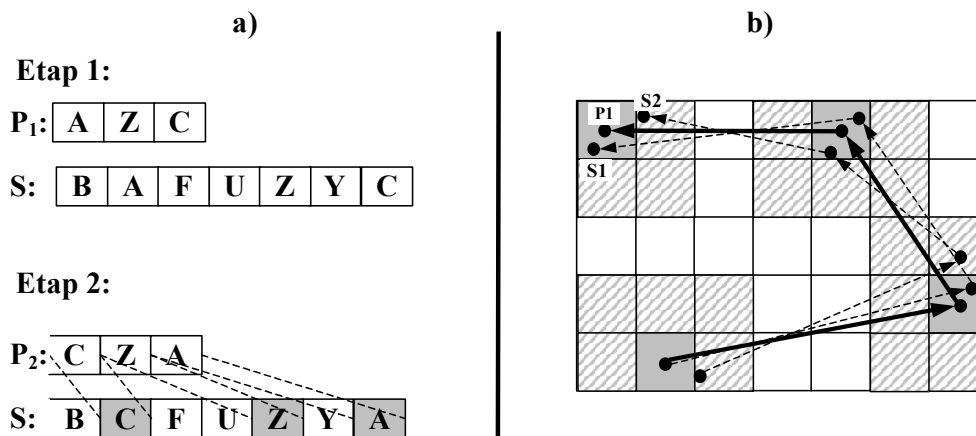
Funkcja	Kolejność elementów wzorca w sekwencji	Występowanie elementów wzorca po sobie w sekwencji
SimpleSearch	NIE	NIE
ExactSearch	TAK	TAK
SelectiveSearch	TAK	NIE
MarkedSearch	TAK	TAK/NIE

Na rys. 3 zostały przedstawione dwie podstawowe funkcje porównujące. Pierwsza z nich (rys. 3a) poszukuje *dokładnego* wystąpienia wzorca w porównywanej sekwencji (ExactSearch). Druga funkcja porównująca (SelectiveSearch) sprawdza, czy kolejne elementy wzorca znajdują się w takiej samej kolejności w sekwencji regionów, przy czym pomiędzy elementami wzorca mogą znajdować się inne elementy.



Rys. 3. Funkcje porównujące: a) ExactSearch, b) SelectiveSearch
 Fig. 3. Comparison functions: a) ExactSearch, b) SelectiveSearch

Podstawowe funkcje porównujące można rozszerzyć zgodnie z tabelą 2. W przypadku porównywania elementu wzorca z elementem sekwencji można wziąć pod uwagę regiony otaczające region wzorca, taki sposób porównywania został nazwany przybliżonym i odpowiada drugiej kolumnie tabeli 2. Na rys. 4b. przedstawiono ideę porównywania przybliżonego.



Rys. 4. Porównywanie: a) dwukierunkowe wyszukiwanie, b) i przybliżone porównywanie
 Fig. 4. Comparison: a) bi-directional comparison, b) approximate comparison

Regiony wzorca zaznaczono na szaro, a ich obszary sąsiedztwa ukośnymi liniami. Pomimo dużego podobieństwa przestrzennego sekwencji regionów S1 i S2 są różne, gdyż składają się z różnych regionów (sekwencje S1 i S2 są w tym przykładzie reprezentowane przez agregaty/centroidy punktów w regionie oraz linie łączące te punkty/agregaty). Żeby w wyniku uwzględnić sekwencje S1 i S2, należy rozszerzyć zakres regionów biorących udział w porównaniu (uwzględnić obszar sąsiedztwa elementów wzorca), co jest realizowane za pomocą funkcji przybliżonego porównywania.

W takim porównywaniu możemy również uwzględnić czas pobytu obiektu w regionie oraz czas przejścia pomiędzy elementami sekwencji (druga kolumna tabeli 2). Trzecia kolumna tabeli 2 odnosi się do dwukierunkowego porównywania, w przypadku którego nie jest brany pod uwagę kierunek wykonywanego wyszukiwania. Działanie algorytmu

dwukierunkowego porównania polega na wyszukiwaniu w porównywanych sekwencjach wystąpienia danego wzorca oraz jego odwróconej postaci (rys. 4a).

Tabela 2

Rozszerzenia funkcji porównujących

Funkcja	Przybliżone porównywanie	Warunki na czas przejścia/czas pobytu
SimpleSearch	TAK	NIE/TAK
ExactSearch	TAK	TAK/TAK
SelectiveSearch	TAK	TAK/TAK
MarkedSearch	TAK	TAK/TAK

Oprócz wymienionych funkcji porównujących została opracowana funkcja StatisticSearch. Zadaniem tej funkcji jest wyznaczanie liczności określonych kombinacji podsekwencji wzorca. Do porównywania podwzorca z sekwencją użyta została funkcja ExactSearch.

4. Struktury agregatów

Poniżej porównano rozmiar agregatów, składowanych w bazie Oracle 10g, w zależności od parametrów StayTime oraz liczby kolumn COLS i wierszy ROWS, na które została podzielona cała przestrzeń. Na początku został przedstawiony sposób składowania sekwencji regionów (agregatów), po czym zostały zaprezentowane wyniki przeprowadzonych eksperymentów.

4.1. Sekwencje regionów dla trajektorii taksówek

W TrDW w celu przechowywania sekwencji RoI (będących agregatami trajektorii obiektów) została stworzona tabela SEQUENCES. Tabela SEQUENCES służy materializacji wyników i posiada następujące atrybuty: (a) ID – identyfikator krotki; (b) OID – identyfikator obiektu ruchomego; (c) SEQID – identyfikator sekwencji regionów; (d) REGROW i REGCOL – współrzędne gridowe regionu; (e) X i Y – współrzędne ciężkości punktów trajektorii, które znajdowały się w regionie; (f) COUNT – liczba punktów trajektorii w regionie; (g) STAYTIME – czas pobytu obiektu w regionie; (h) ENTERTIME – czas, w którym obiekt pojawił się w regionie; (i) EXITTIME – czas wyjścia obiektu z regionu; (j) STATUS – kod typu regionu, typy to np. START, STOP, END itd.; (k) SEQUENCES_SAVE_ID – identyfikator materializacji; (l) CODE jednowymiarowy kod regionu. Atrybuty ID, OID, SEQID, X, Y, COUNT, STAYTIME, ENTERTIME, EXITTIME SEQUENCES_SAVE_ID są typu NUMBER, a atrybuty REGROW, REGCOL, STATUS są typu NUMBER(4). Atrybut CODE jest typu NUMBER(8).

4.2. Testowanie

Poniżej zostało przedstawione zestawienie rozmiarów tabeli SEQUENCES dla różnych wartości parametrów StayTime oraz COLS i ROWS. W tym przypadku parametr StayTime określa minimalną liczbę kolejnych punktów trajektorii obiektu w regionie, które muszą wystąpić, aby region mógł zostać dodany do sekwencji regionów (pierwsza z metod pomiaru gęstości z rozdziału 2). Wszystkie pomiary zostały wykonane za pomocą Oracle Enterprise Manager dla tych samych parametrów składowania tabel.

Parametr W1 (zobacz tabelę na rys. 5) pokazuje, jaką częścią rozmiaru zbioru źródłowego (trajektorie zostały wygenerowane za pomocą [3]; liczba wygenerowanych wierszy: 2394389; liczba trajektorii: 50005; rozmiar składowanych danych: 100316 kB) jest rozmiar danej tabeli agregatów.

StayTime	Tables											
	COLS=ROWS=25				COLS=ROWS=50				COLS=ROWS=75			
	ROWS	Space Used (kB)	Space Allocated(KB)	W1(%)	ROWS	Space Used (kB)	Space Allocated(KB)	W1(%)	ROWS	Space Used (kB)	Space Allocated(KB)	W1(%)
10	143307	10238,8	10240	10,20652	138333	9727,51	10240	9,69689	138546	9718,81	10240	9,68822
9	148403	10742,7	11264	10,70888	142790	10004,9	10240	9,97337	147485	10275,13	11264	10,2428
8	156128	11004	11264	10,96934	151438	10742,7	11264	10,7089	153485	10742,69	11264	10,7089
7	169045	12286,5	12288	12,24783	171710	12286,5	12288	12,2478	158701	11262,62	11264	11,2272
6	184229	13007,4	13312	12,96644	191794	13476,6	14336	13,4341	168605	11766,56	12288	11,7295
5	198674	14014	14336	13,96986	204076	14295,1	14336	14,2501	199696	13858,42	14336	13,8148
4	228987	16382	16384	16,33044	239851	16771,4	17408	16,7187	249524	17405,88	17408	17,3511
3	286818	20477,5	20480	20,41305	313703	22005,3	22528	21,936	303601	21501,38	21504	21,4337
2	354046	25076,9	25600	24,99801	459197	32244,1	32768	32,1426	475029	33027,97	33792	32,924
1	412960	28932,5	29696	28,8414	731327	50673,8	51200	50,5143	971464	66539,93	69632	66,3305

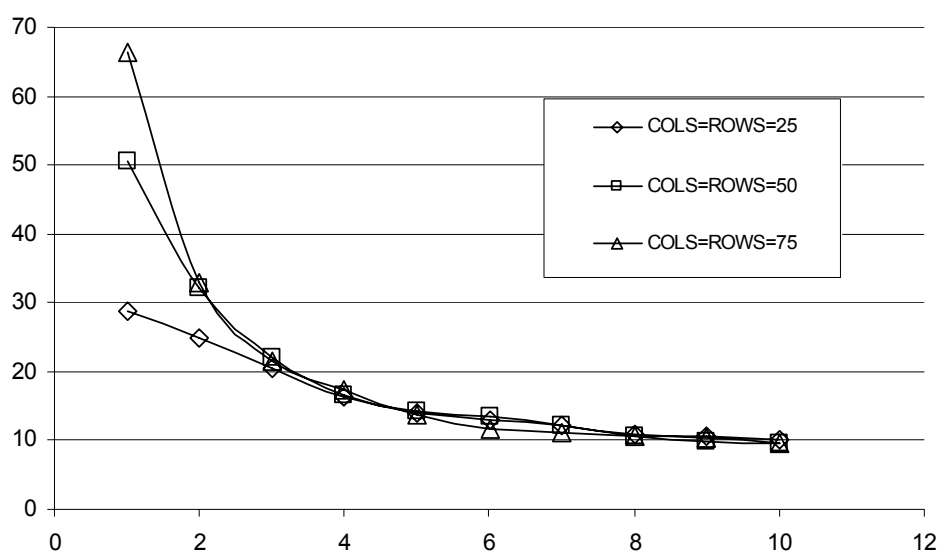
Rys. 5. Tabela zależności rozmiaru agregatów od parametru StayTime

Fig. 5. Table of influence aggregation size from StayTime parameter

Rysunek 6 obrazuje graficznie wyniki z tabeli przedstawionej na rys. 5.

Na podstawie uzyskanych eksperymentalnie wyników możemy wysunąć następujące wnioski:

- Wraz ze zmniejszaniem wartości parametru StayTime wzrasta rozmiar składowanych agregatów. Wzrost jest spowodowany zwiększeniem liczby regionów w sekwencjach, co z kolei jest spowodowane tym, że obiekty ruchome w większej liczbie regionów przebywały dłużej, niż wynosiła wartość parametru StayTime.
- Wraz ze zwiększeniem rozdzielczości podziału mapy wzrasta rozmiar składowanych agregatów. Wzrost był spowodowany zwiększeniem dokładności odwzorowania trajektorii przez regiony, a tym samym zwiększeniem długości sekwencji regionów.
- Minimalna wartość parametru StayTime, dla której rozmiary składowania agregatów dla różnych rozdzielczości nie różniły się znacząco, wyniosła 3.



Rys. 6. Rozmiary agregatów w zależności od parametru StayTime

Fig. 6. The sizes of the aggregates with respect to StayTime parameter

Uzyskane wyniki pokazały, że może istnieć taka wartość parametru StayTime, poniżej której rozmiar agregatów wzrasta niemalże wykładniczo. Istnienie takiej wartości wynika z faktu, że dla dużych wartości parametru StayTime średnia długość sekwencji regionów dąży do wartości z przedziału $1 \div 2$. Znajomość takiej wartości może się okazać przydatna przy wyborze odpowiedniej rozdzielczości podziału mapy dla różnych wartości parametru StayTime.

5. Podsumowanie

W artykule przedstawiono sposób wyznaczania sekwencji regionów, sposoby wyszukiwania wzorców oraz wpływ parametrów transformacji trajektorii na rozmiar agregatów.

W celu przyspieszenia wyszukiwania różnych przestrzenno-czasowych zależności zostały wyznaczone agregaty, reprezentowane za pomocą sekwencji regionów. Po pierwsze, zastosowano mapowanie punktów trajektorii na sekwencje odpowiednich regionów, co pozwoliło ograniczyć liczbę analizowanych danych i przyspieszyć proces eksploracji. Po drugie, dla każdego z regionów wyznaczono centroid znajdujących się w nim punktów trajektorii. Mając równocześnie na uwadze długi czas przetwarzania dużych zbiorów trajektorii, użyto prostej materializacji wyznaczonych sekwencji w postaci zapisu do TrDW. W artykule zostały także przedstawione podstawowe metody porównywania dla wzorców w sekwencji, których zastosowanie nie ogranicza się tylko i wyłącznie do trajektorijnych hurtowni danych.

W końcowej części artykułu pokazano, że użycie modelu obiektowego może przynieść wymierne korzyści w przypadku składowania długich sekwencji regionów.

BIBLIOGRAFIA

1. Alvares L. O., Bogorny V., Kuijpers B., de Macedo J. A. F., Moelans B., Vaisman A.: A Model for Enriching Trajectories with Semantic Geographical Information. GIS 2007: Seattle, Washington, USA
2. Giannotti F., Nanni M., Pedreschi D., Pinelli F.: Trajectory Pattern Mining. KDD 2007, s. 330÷339
3. Brinkhoff T.: A Framework for Generating Network-Based Moving Objects – <http://www.fh-oow.de/institute/iapg/personen/brinkhoff/generator/>.
4. Agrawal R., Srikant R.: Fast Algorithms for Mining Association Rules. VLDB 1994, s. 487÷499.
5. Agrawal R., Srikant R.: Mining Sequential Patterns. ICDE 1995, s. 3÷14.
6. Gorawski M.: Lokalizacja obiektów i trajektoryjne hurtownie danych. Konferencja Inteligentny System Transportowy dla Śląska, Katowice 2008, s. 1÷14.

Recenzent: Prof. dr Jarek Gryz

Wpłynęło do Redakcji 31 stycznia 2009 r.

Abstract

This paper presented various ways to how 'region sequences' can be calculated, 'patterns' can be searched, and the aggregates stored. In order to speed up the search of different spatial-time dependencies, various aggregation methods can be used. Firstly, a mapping of the points of the trajectory into the sequence of the respective regions was used that helped to reduce the number of analyzed sets of data and sped-up the process of exploration. Secondly, for each of the regions a centroid of the trajectory points was determined. Because of the lengthy processing time of large trajectory data sets, a simple materialization of the determined sequences in a form of writing to TrDW was used. In the paper were also presented the basics of 'comparison methods', which can be used for Trajectory Data Warehouse technology among others.

Adresy

Marcin GORAWSKI: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16,
44-100 Gliwice, Polska, Marcin.Gorawski@polsl.pl.

Paweł JURECZEK: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16,
44-100 Gliwice, Polska, Pawel.Jureczek@polsl.pl.