Raport Badawczy
Research Report

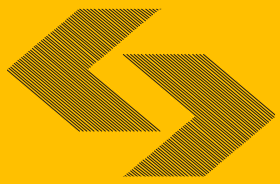
RB/65/2010

Zastosowanie algorytmów grupowania danych opartych na gęstości do odkrywania ważnych miejsc ze śladów GPS

A. Stojanowski

Instytut Badań Systemowych
Polska Akademia Nauk

Systems Research Institute
Polish Academy of Sciences



ZASTOSOWANIE ALGORYTMÓW GRUPOWANIA DANYCH OPARTYCH NA GĘSTOŚCI DO ODKRYWANIA WAŻNYCH MIEJSC ZE ŚLADÓW GPS

Adam Stojanowski
Studia Doktoranckie IBS PAN

Artykuł opisuje wykorzystanie algorytmów grupowania danych opartych na gęstości do odkrywania ważnych miejsc na podstawie śladów GPS. Dane pozyskano z wielu pojazdów ciężarowych kierowanych przez wielu kierowców w różnym czasie. Głównie skupiono się na działaniu algorytmów DBSCAN oraz DENCLUE.

Słowa kluczowe: *data mining, grupowanie danych, DENCLUE, system GPS, systemy GIS, systemy monitoringu floty pojazdów, transport, spedycja, logistyka, drążenie danych*

Wstęp

Firmy z branży TSL (transport – spedycja – logistyka) podobnie jak firmy innych branż bardzo dotkliwie odczuły skutki ostatniego kryzysu. Nie bez znaczenia dla tego biznesu są również niestabilne ceny ropy (np. 2008.07.03 – 146,14 USD / baryłka; 2008.12.26 – 38,55 USD / baryłka, 2010.03.09 – 78,95 USD / baryłka). Zarządy firm zostały zmuszone do szukania możliwości obniżenia kosztów nie tylko zużycia paliwa lecz w każdym obszarze prowadzonej działalności. Do powszechnego użytku weszły systemy monitorowania floty online przy użyciu technologii GPS/GPRS. Ręczna analiza ogromnej ilości danych dostarczanych przez te systemy jest praktycznie niemożliwa. Danych jest tym więcej im więcej pojazdów posiada firma transportowa.

Analiza danych pozyskanych z urządzeń GPS daje ogromy obszar do działania dla metod sztucznej inteligencji. Wyznaczenie miejsc najczęściej odwiedzanych przez pojazdy firmy transportowej a następnie wyznaczenie najczęstszych połączeń pomiędzy tymi miejscami otwiera drogę do możliwości wykonania zaawansowanych analiz, które z kolei mogą doprowadzić do optymalizacji procesu transportowego.

W artykule tym krótko zostaną opisane najważniejsze znane podejścia do grupowania danych. Więcej miejsca zostanie poświęcone algorytmom grupo-

wania opartym na gęstości. Następnie przedstawione zostaną wyniki eksperymentów wykonanych na rzeczywistych danych firmy transportowej. Zastosowane algorytmy to DBSCAN [2] oraz DENCLUE [1]. Przedmiotem eksperymentów będzie wyznaczenie ważnych z punktu widzenia transportu ciężarowego miejsc odwiedzanych przez pojazdy (np. miejsca rozładunku, załadunku, najczęstsze miejsca postojów, itp.)

1. Najważniejsze podejścia do grupowania danych

Grupowanie danych jest jedną z technik drążenia danych (ang. *data mining*), które to techniki opierają się na dziedzinach nauki takich jak statystyka (analiza wielowymiarowa) czy maszynowe uczenie (ang. *machine learning*). Metody drążenia danych wywodzą się z obszaru badań nad sztuczną inteligencją.

Metody partycjonujące

Metody partycjonujące należą do grupy metod iteracyjno-optymalizacyjnych. Zasada ich działania polega na podziale bazy obiektów na k podzbiorów będących klastrami (grupami). Parametrem wejściowym jest wartość k, która oznacza ilość grup. Konieczność podania na wejściu ilości grup jest w wielu przypadkach największą wadą metod partycjonujących gdyż często właśnie ta informacja jest przedmiotem poszukiwań. Inną niedogodnością tych algorytmów jest brak odporności na szum czy też „wartości oddalone" (ang. *outliers*). Metody te nie nadają się również do odkrywania klastrów o dowolnych kształtach.

Algorytmy: K-MEANS [3], K-MEDOIDS [4], CLARANS [5]

Metody hierarchiczne

Efektem działania metod hierarchicznych jest dendrogram będący drzewem grup. Liście reprezentują obiekty a węzły – grupy. Obcięcie drzewa na odpowiednim poziomie daje odpowiadającą temu poziomowi ilość klastrów. Dendrogram pozwala także poznać związki pomiędzy grupami na różnym poziomie. Zasada działania zależy od wybranego podejścia grupowania hierarchicznego: metoda aglomeracyjna – początkowo każdy obiekt jest osobną grupą; w każdej iteracji grupy są łączone, metoda deaglomeracyjna – w pierwszej iteracji wszystkie obiekty stanowią jedną grupę a następnie są dzielone na mniejsze grupy.

Algorytmy: BIRCH [6], CHAMELEON [7]

Metody oparte na gęstości

Podejście algorytmów opartych na gęstości polega na generowaniu grup w miejscach o dużym zagęszczeniu obiektów. Są bardzo skuteczne w grupowaniu danych przestrzennych. Równie dobrze radzą sobie z odfiltrowaniem szumu jak i punktów oddalonych. Metody oparte na gęstości mogą formować klastry o dowolnym kształcie. Algorytmy te zostaną dokładniej opisane w dalszej części tego dokumentu.

Algorytmy: DBSCAN [2], DENCLUE [1], OPTICS [8],

Metody gridowe

Metody używające siatkowych struktur danych o wielu poziomach dokładności. Dzielą one przestrzeń danych na skończoną ilość komórek o strukturze siatkowej - na niej są wykonywane wszystkie obliczenia. Główną zaletą tych metod jest szybki czas przetwarzania, który zależy od ilości danych oraz komórek w poszczególnych wymiarach przestrzeni danych.

Algorytmy: STING [9], WaveCluster [10], CLIQUE [11]

2. Algorytmy grupowania oparte na gęstości

2.1. DBSCAN: Density Based Spatial Clustering of Applications with Noise

Algorytm buduje klastry jako zbiór punktów gęstościowo połączonych (*ang. density-connected*). Na wejściu należy podać dwa parametry: *Eps*(ε) – maksymalny promień sąsiedztwa, *MinPts* – minimalna ilość punktów w sąsiedztwie ograniczonym przez *Eps*.

Autorzy tego podejścia opisali je w kilku definicjach:

- ε – sąsiedztwo punktu p oznaczone jako $N_{Eps}(p)$ jest zdefiniowane przez:

$$N_{Eps}(p) = \{q \in D / \text{dist}(p,q) \leq Eps\} \quad (1)$$

- punkt p jest punktem rdzennym (*ang. core point*) jeżeli jego ε – sąsiedztwo zawiera przynajmniej *MinPts* punktów,

$$| N_{Eps}(p) | \geq MinPts \quad (2)$$

- punkt p jest bezpośrednio gęstościowo osiągalny (*ang. directly density-reachable*) z punktu q jeżeli punkt p znajduje się w ε – sąsiedztwie q oraz punkt q jest punktem rdzennym,

- punkt p jest gęstościowo osiągalny z punktu q z uwzględnieniem ε i *MinPts* jeżeli istnieje łańcuch punktów p_1, ..., p_n, p1 ==q, p_n = p takich, że p_{i+1} jest bezpośrednio gęstościowo osiągalny z punktu p_i,
- punkt p jest gęstościowo połączony (*ang. density-connected*) z punktem q z uwzględnieniem *Eps* oraz *MinPts* jeżeli istnieje punkt o, że punkty p oraz q są gęstościowo osiągalne z punktu o z uwzględnieniem ε oraz *MinPts*,
- klaster C z uwzględnieniem ε oraz *MinPts* jest podzbiorem zbioru punktów, który spełnia następujące warunki:
  - ∀p,q: jeżli p ∈ C oraz q jest gęstościowo osiągalne z p z uwzględnieniem *Eps* oraz *MinPts*, wtedy q ∈ C (maksymalność),
  - ∀p,q∈ C: p jest gęstościowo połączone do q z uwzględnieniem ε oraz *MinPts*, (połączeniowość),
- szum jest zbiorem punktów, które nie należą do żadnego klastra,

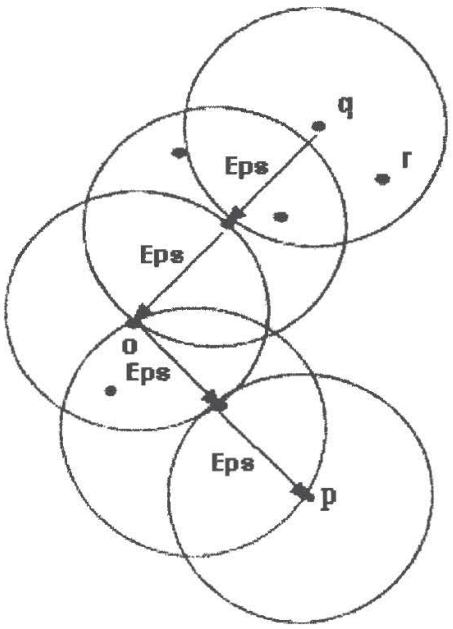
Algorytm DBSCAN wyznacza klastry poprzez analizę wszystkich punktów w bazie. Dla każdego obiektu analizuje *Eps* – sąsiedztwo pod względem zagęszczenia obiektów (*MinPts*). Algorytm rozpoczyna swoją pracę od dowolnego punktu bazy. Wyznacza on nowy klaster jeśli w *Eps* – sąsiedztwie danego punktu znajduje się minimum *MinPts* innych punktów. DBSCAN dodaje do klastra wszystkie punkty bezpośrednio gęstościowo osiągalne i gęstościowo połączone. Warunkiem zakończenia działania algorytmu jest przetworzenie wszystkich punktów w bazie.

2.2. DENCLUE

Metoda DENCLUE wyszukuje grupy poprzez wyznaczenie tzw. atraktorów gęstości (*ang. density-atracttors*), które są lokalnymi maksymami ogólnej funkcji gęstości. Do ich wyznaczenia autorzy proponują algorytm *hill-climbing* wspomagany przez gradient funkcji. Funkcja gęstości jest sumą funkcji wpływu (*ang. influence function*) wszystkich punktów. Funkcja wpływu natomiast, jak sugeruje nazwa, określa wpływ punktu na jego sąsiedztwo. Jako funkcję wpływu można użyć między innymi funkcji Gaussa.

Autorzy formalnie definiują kilka kluczowych pojęć dla swojego podejścia.

Niech x i y będą punktami w F^d d-wymiarowej przestrzeni.



Rysunek 1. Punkty gęstościowo osiągalne. Punkt *r* jest bezpośrednio gęstościowo osiągalny z punktu *q*; punkt *o* jest gęstościowo osiągalny z punktu *q*; punkt *q, o, p* są gęstościowo połączone; punkt *q* jest punktem rdzennym; punkt *p* jest punktem granicznym.

- funkcja wpływu (*ang. influence function*) punktu $y \in F^d$ jest to funkcja $f_B^y : F^d \to R_0^+$ zdefiniowana jako podstawowa funkcja wpływu f_B

$$f_B^y(x) = f_B(x, y) \tag{3}$$

W szczególności gdy B jest rozkładem Gaussa:

$$f_{Gauss}(x, y) = e^{-\frac{d(x,y)^2}{2\sigma^2}} \tag{4}$$

- funkcja gęstości (*ang. density function*) jest sumą funkcji wpływu wszystkich punktów. Dla danych *n* punktów, $D=\{x_1, \ldots, x_n\} \subset F^d$, funkcja gęstości w punkcie x zdefiniowana jest jako:

$$f_B^D(x) = \sum_{i=1}^{n} f_B^{x_i}(x) \tag{5}$$

W szczególności gdy B jest rozkładem Gaussa:

$$f_{Gauss}^D(x) = \sum_{i=1}^{n} e^{-\frac{d(x,x_i)^2}{2\sigma^2}} \tag{6}$$

- gradient funkcji $f_B^D(x)$:

$$\nabla f_B^D(x) = \sum_{i=1}^{n} (x_i - x) * f_B^{x_i}(x) \tag{7}$$

Dla funkcji wpływu Gaussa:

$$\nabla f_{Gauss}^D(x) = \sum_{i=1}^{n} (x_i - x) * e^{\frac{-d(x,x_i)^2}{2\sigma^2}} \tag{8}$$

- punkt $x \in F^d$ należy do atraktora gęstości x*, jeżeli $\exists k \in N : d(x^k, x^*) \leq \varepsilon$ oraz:

$$x^0 = x, x^i = x^{i-1} + \delta * \frac{\nabla f_B^D(x^{i-1})}{\left\| \nabla f_B^D(x^{i-1}) \right\|} \tag{9}$$

Analogicznie dla funkcji Gaussa mamy:

$$x^0 = x, x^i = x^{i-1} + \delta * \frac{\nabla f_{Gauss}^D(x^{i-1})}{\left\| \nabla f_{Gauss}^D(x^{i-1}) \right\|} \tag{10}$$

- klaster typu *„center-definded"* dla atraktora gęstości x* to podzbiór $C \in D$, zawierający takie punkty $x \in C$, które przynależą do atraktora gęstości x* oraz $f_B^D(x) \geq \xi$.

- klaster typu *„arbitrary-shape"* dla zbioru atraktorów gęstości X to podzbiór $C \in D$, gdzie:

- $\forall x \in C \exists x^* \in X : f_B^D(x) \geq \xi$, punkt x przynależy do atraktora gęstości x* oraz

- $\forall x_1^*, x_2^* \in X : \exists$ ścieżka $P \subset F^d$ z x_1^* do x_2^* oraz $\forall p \in P : f_B^D(x) \geq \xi$

Działanie algorytmu rozpoczyna się od wyznaczenia minimalnego prostokąta zawierającego całą przestrzeń danych. Następnie należy ten prostokąt podzielić na d-wymiarowe kostki. Wyznaczone powinny zostać tylko kostki zawierające punkty ($\|C_p\|$). Grupy mogą się rozciągać na kilka kostek, dlatego w celu ułatwienia dostępu do nich należy także połączyć sąsiednie kostki zawierające punkty. Aby zredukować czas niezbędny na łączenie, algorytm przewiduje

wyznaczenie najgęściej wypełnionych kostek (C_{sp}) a następnie podłączenie do nich sąsiadów (zbiór C_r).

W kolejnym etapie należy działać na zbiorze kostek najgęściej wypełnionych z podłączonymi do nich sąsiadami (C_r). Wyznaczamy gęstość każdego punktu, korzystając z wcześniej przedstawionej funkcji gęstości. Następnie wyznaczamy dla każdego punktu x ataraktor gęstości, korzystając z algorytmu *hill-climbing* wspomaganego przez gradient. Jeżeli znaleziony atraktor gęstości spełnia warunek $f_B^D(x^*) \geq \xi$ punkt x dołączany jest do klastra należącego do x^*. Dla zwiększenia wydajności algorytm przewiduje zapamiętanie wszystkich pobliskich punktów w trakcie wykonywania procedury *hill-climbing* oraz podłączenia ich do klastra reprezentowanego przez x^*.

3. Eksperyment

Przygotowanie danych

Dane na których zostały wykonane badania są rzeczywistymi danymi pochodzącymi z 80 pojazdów, które przesyłają informacje dotyczące między innymi swojego położenia. Częstotliwość przesyłania pozycji odbywa się co 5 kilometrów. Pod uwagę wzięto okres około 2 lat. Pojazdy realizowały w tym okresie zlecenia na terenie całej Europy.

Algorytm 1

Set stop time threshold ***stt***

Set distance threshold ***dt***

Order objects by *TruckID* and *ObjectDate*

while exist unprocessed object ***O*** in objects base ***B*** **do**

Compute time difference ***t*** between object O_{i+1} and **O**

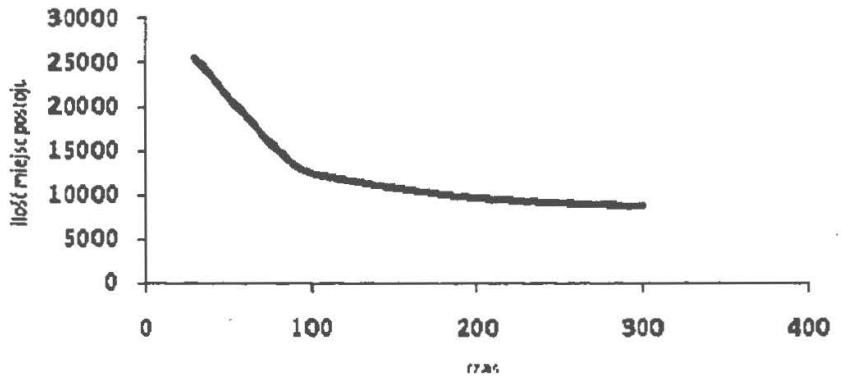
Compute distance ***d*** between object O_{i+1} and **O**

if ***t*** is bigger than ***stt*** **and** ***d*** is bigger than ***dt*** **then**

Add object **O** to StopsTable

end while

Return StopsTable(O_0, O_1, O_2, … , O_n)



Rysunek 2. Wpływ czasu postoju na ilość znalezionych miejsc.

Celem badań jest wykrycie ważnych miejsc. Zakładamy, że ważne miejsce to takie w którym pojazdy zatrzymują się i pozostają w nim przez określony czas. W celu ich wyznaczenia posłużono się algorytmem przedstawionym na następnej stronie.

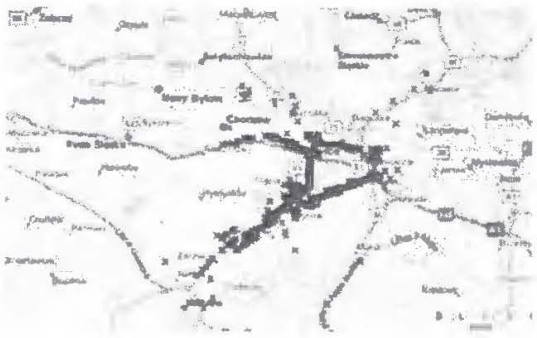
Tak jak można było się spodziewać im dłuższy czas postoju tym mniej punktów zostało znalezionych. Wynikowy zbiór punktów zawiera zarówno przypadkowe miejsca postoju jak i miejsca ważne, które należy wyznaczyć. Do tego celu zostaną użyte algorytmy grupowania oparte na gęstości: DBSCAN oraz DENCLUE. Algorytmy te zostały zaimplementowane w języku C#. Natomiast wyznaczony zbiór punktów posłużył jako zestaw danych wejściowych.

Do przeprowadzenia dalszych eksperymentów wykorzystano próbę 10081 punktów uzyskanych z populacji zawierającej 768881 punktów, którą uzyskano dzięki zastosowaniu parametru *stt* = 180 min.

Wynik działania algorytmów DBSCAN i DENCLUE

Klastry zaznaczone na rysunku nr 3 zostały uzyskane za pomocą algorytmu DBSCAN przy ustawieniu parametrów Eps = 2 km oraz MinPts = 40. Stosując algorytm DENCLUE podobny efekt został uzyskany przy parametrach $\xi = 40$ oraz $\delta = 5km$.

Rysunek 3. Przykładowe grupy znalezione za pomocą algorytmu DBSCAN



Rysunek 4. DBSCAN

Rysunek 5. DENCLUE

Klaster w rejonie Katowic wykryty za pomocą algorytmu DBSCAN i DENCLUE

Grupy uzyskiwane z wykorzystaniem algorytmów DBSCAN i DENCLUE są podobne (jak widać na rysunkach 4 i 5). Uzyskane efekty są zadowalające i zgodne ze stanem faktycznym. Wyznaczone w ten sposób grupy mogą posłużyć dalszej analizie.

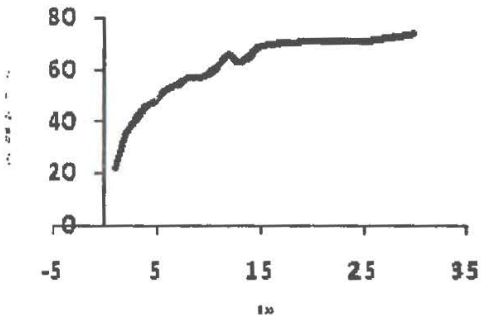
Uwagi końcowe i dalsze kierunki rozwoju

W dokumencie tym przedstawiono sposób wyznaczania ważnych miejsc z punktu widzenia firmy transportowej. Zostały one wykryte przy użyciu algorytmów grupowania danych opartych na gęstości. Miejsca te mogą posłużyć

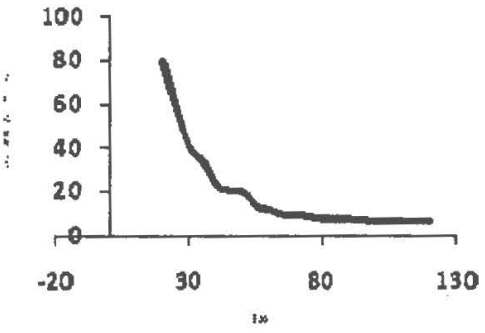
jako baza do wyznaczenia najczęstszych tras przejazdów pomiędzy nimi. Miejsca i trasy mogą zostać poddane dalszej analizie, która doprowadzi do optymalizacji procesu transportowego i pozwoli firmie osiągać lepsze wyniki finansowe.

Analiza może obejmować np. wykrywanie anomalii (nieuzasadnione opuszczenie trasy, która najczęściej wybierana jest przez kierowców), detekcja obszarów poprzez które przejazd niekorzystnie wpływa na transport (częste utrudnienia ruchu, korki, częste pomyłki w wyborze trasy). Przedmiotem analizy może być także zużycie paliwa z uwzględnieniem np. marki pojazdu, kierowcy, wagi przewożonego towaru czy też pory roku.

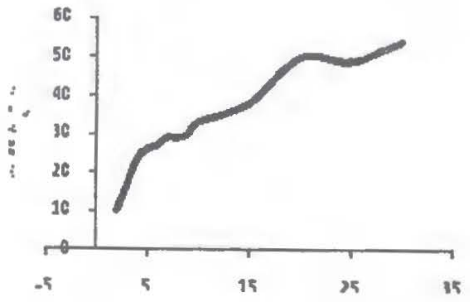
Uzasadnionym rozwinięciem przedstawionych podejść mógłby być algorytm uwzględniający analizę czasu spędzonego w danym rejonie. Bardzo prawdopodobne jest, że istnieją ważne miejsca, w których czas postoju jest krótszy niż zastosowany do uzyskania danych wejściowych dla algorytmów grupowania (ponieważ niektóre firmy ustalają okna czasowe w których pojazd ma się podstawić pod załadunek/rozładunek – co skraca czas postoju).



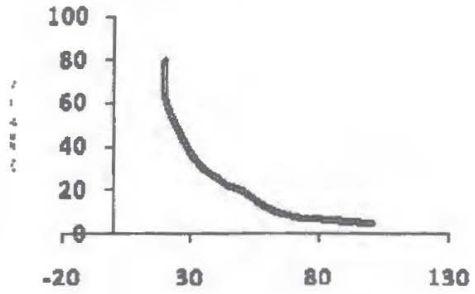
Rysunek 6. DBSCAN. Zależność parametru Eps na ilość znalezionych klastrów (MinPts = 40).



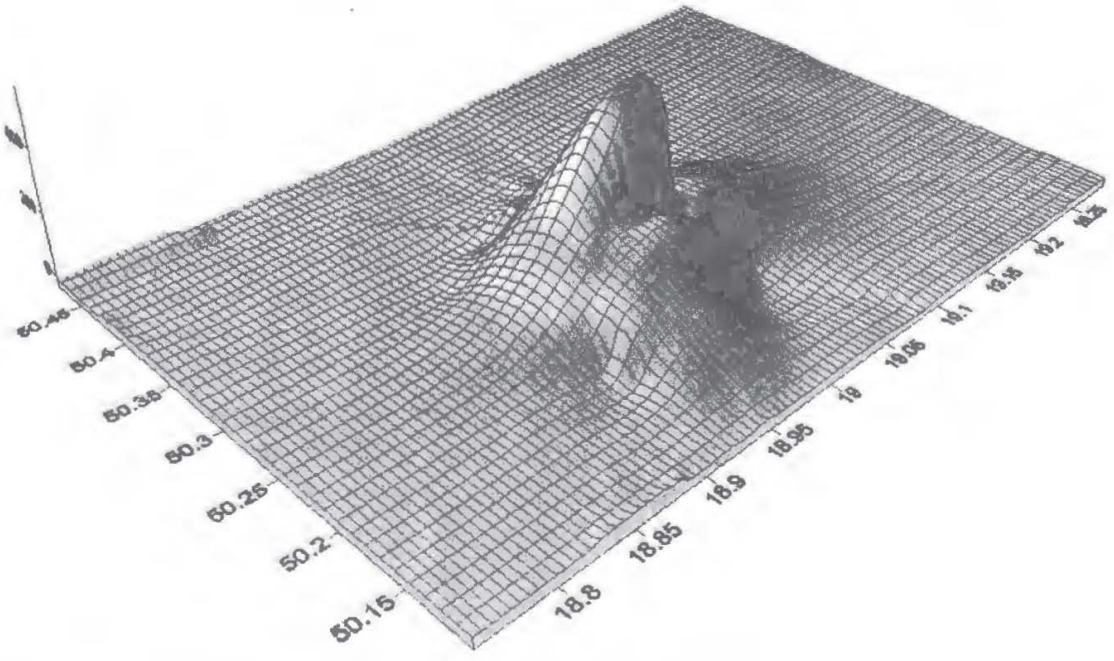
Rysunek 7. DBSCAN. Zależność parametru MinPts na ilość znalezionych klastrów (Eps = 2 km).



Rysunek 8. DENCLUE. Zależność parametru δ na ilość znalezionych klastrów ($\xi = 40$).



Rysunek 9. DENCLUE. Zależność parametru ξ na ilość znalezionych klastrów ($\delta = 5km$).



Rysunek 10. DENCLUE. Przykład funkcji gęstości dla klastra znalezionego w rejonie Katowic.

Literatura

[1] Hinneburg, D. Keim (1998): *An Efficient Approach to Clustering in Large Multimedia Databases with Noise*, Proceedings od KDD Conference.

[2] M. Ester, H.-P. Krigel, J. Sander, X. Xu (1996): *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noice*, Proc. 3rd Int. Conf. on Knowledge Discovery and Data Mining, AAAI Press.

[3] J. MacQueen (1967): *Some methods of classification and analysis of multivariate observations.*

[4] L. Kaufman, P.J. Russeeuw (1990): *Finding groups in Data: An Introduction to Cluster Analysis*, John Wiley & Sons.

[5] R. Ng, J. Han (1995): *Efficient and effective clustering method for spatial data mining*, In Proc. 1994 Intl. Conf. Very Large Data Bases (VLDB'94), Santiago, Chile, 144–155.

[6] T. Zhang, R. Ramakrishnan, M. Livny (1996): *BIRCH: an efficient data clustering method for very large databases*, In Proc. 1996 ACM-SIGMOD Intl. Conf. Management of Data, Montreal, Canada, 103–114.

[7] G. Karypis, E.-H. Han, V. Kumar: "CHAMELEON: A hierarchical clustering algorithm using dynamic modeling", Computer, 32:68–75.

[8] M. Ankerst, M. Breunig, H.-P. Kriegel, J. Sander (1999): *OPTICS: Ordering points to identify the clustering structure*, In Proc. 1999 ACM-SIGMOD Intl. Conf. Management of Data (SIGMOD'99), Philadelphia, PA, 49–60, 1999.

[9] W. Wang, J. Yang, R. Muntz (1997): *STING: A statistical information grid approach to spatial data mining*, In Proc. 1997 Intl. Conf. Very Large Data Bases (VLDB'97), Athens, Greece, 186–195.

[10] G. Sheikholeslami, S. Chatterjee, A. Zhang (1998): *WaveCluster: A multiresolution clustering approach for very large spatial databases*, In Proc. 1998 Intl. Conf. Very Large Data Bases (VLDB'98), New York, 428–439.

[11] R. Agrawal, J. Gehrke, D. Gunopulos, P. Raghavan (1998): *Automatic subspace clustering of high dimensional data for data mining applications*, In Proc. 1998 ACMSIGMOD Intl. Conf. Management of Data (SIGMOD'98), Seattle, 94–105.