



Politechnika Wrocławska

## Wydział Informatyki i Zarządzania

kierunek studiów: Informatyka

specjalność: Inżynieria Oprogramowania

### Praca dyplomowa - magisterska

### Algorytmy ewolucyjne w zadaniu klasteryzacji.

Jarosław Wojtasik

słowa kluczowe:

Klasteryzacja

Obliczenia ewolucyjne

Data mining

krótkie streszczenie:

1 linia

2 linia

3 linia

4 linia

5 linia

6 linia

Promotor:	.....	.....	.....
	<i>imię i nazwisko</i>	<i>ocena</i>	<i>podpis</i>

Do celów archiwalnych pracę dyplomową zakwalifikowano do: \*

a) kategorii A (akta wieczyste)

b) kategorii BE 50 (po 50 latach podlegające ekspertyzie)

\* niepotrzebne skreślić

Pieczętka instytutu, w którym  
student wykonywał pracę

Wrocław 2011

## Spis treści

1	Wstęp.....	1
1.1	Klasteryzacja a Klasyfikacja .....	1
1.2	Klasteryzacja – jak to działa.....	2
1.3	Reprezentacja rekordu .....	3
1.4	Miara podobieństwa .....	4
1.5	Techniki Klastrowania.....	5
2	Algorytmy ewolucyjne w klasteryzacji.....	5
2.1	Funkcje Oceny .....	6
2.2	Wielokryterialność.....	6
3	Model .....	8
3.1	Reprezentacja Chromosomu .....	8
3.2	Inicjalizacja populacji .....	8
3.3	Funkcja Oceny .....	8
3.3.1	Gęstość .....	8
3.3.2	Łączność.....	8
3.3.3	Rozłączność.....	9
3.3.4	Poprawność jednostki.....	9
3.4	Operatory Ewolucyjne .....	9
3.4.1	Krzyżowanie.....	9
3.4.2	Mutacja.....	9
4	Plan Pracy.....	10
5	Bibliografia.....	1
6	Spis Ilustracji.....	2

## Streszczenie

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sapien ipsum, fermentum eu malesuada ac, malesuada sed magna. Fusce mollis tempor eros, sed feugiat ante auctor ut. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed tincidunt, enim vel pulvinar porttitor, sem velit scelerisque nisi, ut aliquet nulla nunc non justo. Mauris eu ipsum risus, eu iaculis lorem. Ut ut nisl ipsum. Nulla facilisi. Maecenas nec quam justo. Vestibulum id libero lectus, sit amet sodales diam. Maecenas pulvinar, ante vel mollis porta, risus sem molestie purus, facilisis lacinia ipsum ligula ac orci. Ut nec elit tincidunt nisi lobortis bibendum eget sit amet odio.

Fusce rutrum elementum felis quis viverra. Donec blandit mattis consectetur. Vestibulum vitae magna at velit sodales placerat et sed risus. Aenean mollis urna non eros bibendum convallis. Duis in turpis ante, vel viverra urna. Duis in velit tellus. Cras suscipit imperdiet justo, semper pellentesque ligula laoreet a. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla sapien urna, gravida non pellentesque volutpat, sagittis porttitor mi. Etiam aliquam risus vel enim venenatis a elementum arcu lacinia. Nam et enim id mauris dictum elementum. Phasellus id porta diam. Nullam ornare odio sagittis ligula ultricies convallis. Integer ac suscipit libero. Fusce sed dolor urna, eu adipiscing metus. Fusce commodo nulla ac diam varius eu adipiscing mi egestas. Donec nibh ante, rutrum sit amet semper a, eleifend quis libero. Vivamus sed posuere nunc.

## Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sapien ipsum, fermentum eu malesuada ac, malesuada sed magna. Fusce mollis tempor eros, sed feugiat ante auctor ut. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed tincidunt, enim vel pulvinar porttitor, sem velit scelerisque nisi, ut aliquet nulla nunc non justo. Mauris eu ipsum risus, eu iaculis lorem. Ut ut nisl ipsum. Nulla facilisi. Maecenas nec quam justo. Vestibulum id libero lectus, sit amet sodales diam. Maecenas pulvinar, ante vel mollis porta, risus sem molestie purus, facilisis lacinia ipsum ligula ac orci. Ut nec elit tincidunt nisi lobortis bibendum eget sit amet odio.

Fusce rutrum elementum felis quis viverra. Donec blandit mattis consectetur. Vestibulum vitae magna at velit sodales placerat et sed risus. Aenean mollis urna non eros bibendum convallis. Duis in turpis ante, vel viverra urna. Duis in velit tellus. Cras suscipit imperdiet justo, semper pellentesque ligula laoreet a. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla sapien urna, gravida non pellentesque volutpat, sagittis porttitor mi. Etiam aliquam risus vel enim venenatis a elementum arcu lacinia. Nam et enim id mauris dictum elementum. Phasellus id porta diam. Nullam ornare odio sagittis ligula ultricies convallis. Integer ac suscipit libero. Fusce sed dolor urna, eu adipiscing metus. Fusce commodo nulla ac diam varius eu adipiscing mi egestas. Donec nibh ante, rutrum sit amet semper a, eleifend quis libero. Vivamus sed posuere nunc.

## 1 Wstęp

Aktualny postęp w technologii, oraz ogrom usług które dzięki niej są dostarczane, generują z dnia na dzień duże ilości danych. Aplikacje takie jak wyszukiwarki, sklepy, blogi, czy miliardy stron internetowych to zaledwie garstka źródeł z jakich napływają. Powszechne stało się kolekcjonowanie danych o powiązaniu biznesowym czy naukowym. A przy dzisiejszych możliwościach zbierania i magazynowania danych, gromadzone są coraz większe zbiory wielowymiarowych danych. „Dane rozrastają się aby zapełnić miejsce przeznaczone na składowanie” (Prawo Parkinsona o danych). Faktem jest, że co rok wielkości te ulegają podwojeniu przy równoczesnym spadku przechowywanej informacji (1). Bez specjalistycznych narzędzi, ludzki ekspert nie jest w stanie wyczytać jakichkolwiek informacji z tak dużej porcji nieprzetworzonych danych. Dziedzina wiedzy, o odkrywaniu wiedzy w bazie (Knowledge Discovery on Database – KDD), powstała w ostatniej dekadzie aby sprostać temu problemowi. I nie jest to tylko dziedzina badań, ale ma także duży potencjał i zastosowanie w prawdziwym świecie (1) (2) (3)

Wzrost ilości jak i różnorodności danych wymaga zaawansowanych metodologii aby automatycznie zrozumieć, przetworzyć i podsumować dane. Techniki analizy danych można ogólnie podzielić na dwa główne typy (4): (I) Badawczy lub opisowy, co oznacza, że badacz nie ma z góry określonych modeli lub hipotez, ale chce zrozumieć ogólne właściwości lub strukturę wielo-wymiarowych danych, oraz (II) potwierdzający lub wnioskujący, co oznacza, że badacz chce potwierdzenia ważności hipotezy, modelu lub zestaw założeń. Wiele technik statystycznych zostało zaproponowanych do analizy danych. Takich jak: analiza wariacji, regresji liniowej, analizy dyskryminacyjnej, korelacji kanonicznej, skalowanie wielowymiarowe, analiza czynnikowa, analiza głównego składnika oraz analiza skupień.

### 1.1 Klasteryzacja a Klasyfikacja

Ważne jest, aby zrozumieć różnicę pomiędzy klasteryzacją (klasyfikacją bez nadzoru), a analizą dyskryminacyjną (klasyfikacją nadzorowaną). W klasyfikacji nadzorowanej posiadamy zbiór rekordów, tzw. Treningowy, który został wstępnie oznaczony etykietami przez eksperta. Zadanie polega na tym, aby sklasyfikować wszystkie nowe i nieoznaczone rekordy etykietami, na podobieństwo oznaczeń ze zbioru treningowego. Robi się to poprzez wygenerowanie serii hipotez na podstawie zbioru treningowego, które danym o odpowiednich wartościach przypisują właściwe im etykiety. Natomiast klasteryzacja (3; 5) (klasyfikacja bez nadzoru) polega na pogrupowaniu kolekcji nieoznaczonych wektorów w klastry posiadające jakieś znaczenie. Tutaj także są etykiety, ale wywodzą się one czysto z danych, które grupowano.

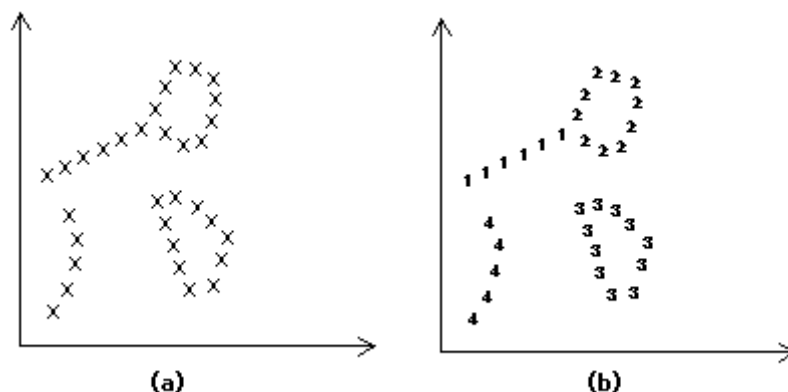
Klasteryzacja jest przydatna w kilku sytuacjach. Takich jak: odkrywcza analiza wzorów, grupowanie, podejmowanie decyzji, uczenie maszynowe. W tym data-mining, wyszukiwanie dokumentów, segmentacja obrazu oraz klasyfikacja rekordów. Na ogół w takich sytuacjach, niewiele jest informacji wstępnych (np. modeli statystycznych) na temat dostępnych danych. To w ramach tych ograniczeń, tworzenie klastrów jest szczególnie odpowiednią metodologią badań współzależności między punktami danych, aby dokonać oceny (być może wstępnej) ich struktury.

## 1.2 Klasteryzacja – jak to działa

Celem grupowania (klasteryzacji) danych, znanego także jako analizy skupień, jest odkrycie naturalnych ugrupowań zbioru rekordów, punktów lub obiektów (6; 7).

Webster (8) definiuje analiza skupień jako „Statystyczna technika klasyfikacji dla odkrywania czy jednostki populacji przynależą do różnych grup, poprzez porównania ilościowe wielu charakterystyk.”

Przykład klasteryzacji przedstawiono na rys 1. Celem jest opracowanie automatycznego algorytmu, który odkryje naturalne ugrupowania (rys 1b) w nie oznaczonych danych (rys 1a). Operacyjną definicję klastrow można sformułować następująco (4): Mając  $n$  obiektów, znajdź  $K$  grup w oparciu o miarę podobieństwa, tak aby podobieństwo pomiędzy obiektami tej samej grupy było wysokie, natomiast niskie dla obiektów z różnych grup. Lecz czym jest miara podobieństwa? Jaka jest definicja klastra? Z rys 1 wynika, że klastry mogą różnić się pod względem kształtu, wielkości oraz gęstości. Obecność szumu w danych czyni wykrywanie klastrow jeszcze trudniejszym. Idealny кластер może zostać zdefiniowany jako zbiór punktów, zwarty i odizolowany. Jednakże w rzeczywistości, кластер to subiektywny byt będący w oku patrzącego, którego znaczenie i interpretacja wymaga wiedzy na temat domeny. Gdy jednak ludzie są doskonali w wyszukiwaniu klastrow w dwóch i prawdopodobnie w trzech wymiarach, potrzebujemy zautomatyzowanych algorytmów dla danych wielowymiarowych. To i niewiadoma ilość klastrow stały się wyzwaniem, które wynikło tysiącami algorytmów, które zostały opublikowane i nadal są.



rys 1 Przykład zbioru danych i wyniku jego klasteryzacji.

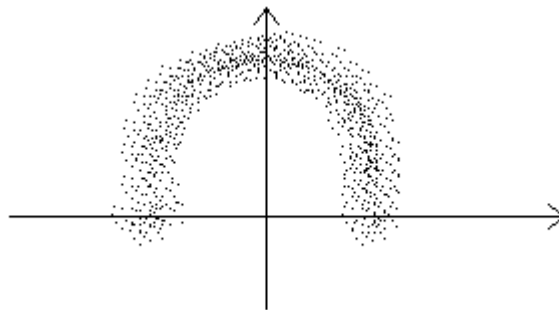
Następujące notacje używane są w dalszej treści dokumentu:

- *Rekord* ( lub *wektor cech*, *obiekt* lub *punkt*)  $\mathbf{x}$  jest pojedynczym elementem danych, użytym w algorytmie klasteryzacji. Zwykle jest wektorem  $d$  miar:  $\mathbf{x} = (x_1, \dots, x_d)$ .
- Indywidualne komponenty  $x_i$  rekordu  $\mathbf{x}$  zwane są *cechami* (lub *atrybutami*).
- $d$  to *wymiarowość* rekordu lub przestrzeni rekordów.
- *Zbiór rekordów* oznaczony jako  $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ . W wielu przypadkach, zbiór który ma podlegać klasteryzacji jest postrzegany jako macierz  $n \times d$  rekordów.
- *Klasa*

- *Twarde* techniki klasteryzacji każdemu rekordowi  $x_i$  przypisują etykietę klasy  $l_i$  Identyfikując jego klasę. Zbiorem wszystkich etykiet dla zbioru rekordów  $\mathcal{X}$  jest  $\mathcal{L} = \{l_i, \dots, l_n\}$ , z  $l_i \in \{1, \dots, k\}$   $l_i \in \{1, \dots, k\}$ , gdzie  $k$  jest liczbą klastrów.
- Procedury *rozmytej* klasteryzacji przypisują każdemu rekordowi stopień przynależności  $f_{ij}$  do danego klastra wyjściowego  $j$
- *Miękkie* techniki klasteryzacji przypisują każdemu rekordowi  $x_i$  stan przynależności  $l_{ij}$  do danego  $j$ -tego klastra. Przy czym  $l_{ij} \in \{0, 1\}$ , gdzie 0- nie należy, 1-należy do klastra.
- *Miara odległości* jest miarą określającą podobieństwo rekordów w przestrzeni ich atrybutów

### 1.3 Reprezentacja rekordu

Brak konkretnych przewodników na temat poprawnej reprezentacji rekordów i doboru cech użytych dla specyficznych sytuacji. Na ogół proces generowania rekordów nie podlega bezpośredniej kontroli. Rolą użytkownika jest gromadzenie faktów i przypuszczeń, ewentualnie dokonać wyboru i ekstrakcji cech. Na pewno dokładna analiza dostępnych cech i dostępnych przekształceń (nawet tych prostych) może dać znacznie lepsze wyniki. Może to zadecydować czy otrzymamy proste i dobrze zrozumiałe klastry, czy skomplikowaną strukturę, której prawdziwą naturę trudno odgadnąć.



rys 2 . ukazuje prosty przykład. Punkty, w tej przestrzeni 2D cech, są zorganizowane w kształt mniej więcej równo oddalony od jednego punktu środka. Jeśli wybrać współrzędne Kartezjańskie, większość algorytmów klastrowania prawdopodobnie podzieli tę figurę na dwa lub więcej klastrów. Jeśli by jedna użyć reprezentacji współrzędnych polarnych, uzyskanie pojedynczego klastra ma większe prawdopodobieństwo.

Rekord może mierzyć tak fizyczny obiekt (np. krzesło) jak i abstrakcyjny (np. styl pisania). Jak wspomniano powyżej, każdy obiekt reprezentowany jest, jako wektor, gdzie każdy wymiar jest pojedynczą cechą. Cechy te można podzielić na ilościowe i jakościowe. Dla przykładu, jeśli waga i kolor były by dwiema użytymi cechami, wtedy (20, black) jest reprezentacją czarnego obiektu o wadze 20 jednostek. Cechy można podzielić na następujące typy.

- (1) Cechy ilościowe
  - A. Wartości ciągłe (np. waga)
  - B. Wartości dyskretne (np. liczba komputerów)
  - C. Wartości przedziału (np. czas przebiegu wydarzenia)
- (2) Cechy jakościowe

- A. Nominalne lub nieposortowane (np. kolor)
- B. Porządkowy (np. ranga wojskowa, temperatura („gorąco”, „zimno”), czy intensywność głośności („cicho”, „głośno”)

Można także używać cech ustrukturyzowanych, reprezentowanych przez drzewo, gdzie rodzic jest generalizacją swoich dzieci. Dla przykładu, węzeł rodzic może być generalizacją dla dzieci oznaczonych, jako „samochody”, „autobusy”, „ciężarówki” i „motory”. Dalej, węzeł „samochody” może być generalizacją samochodów typu „Toyota”, „Ford”, „Mercedes”

Ważne jest, aby wyizolować tylko te najbardziej wartościowe, opisowe i dyskryminujące cechy z zestawu wejściowego, a następnie indywidualnie poddać kolejnym przetwarzaniom analizy. Techniki wyboru cech wyznaczają podzbiory istniejących cech. Natomiast techniki ekstrakcji obliczają nowe cechy z już istniejących. W obu przypadkach celem jest polepszenie jakości klasyfikacji i/lub efektywności obliczeniowej. Dobór cech to bardzo obszerny temat, jednakże w klasteryzacji często zmuszeni jesteśmy na dokonywanie wyborów „na wycucie” a nie rzadko jest to proces prób i błędów, gdzie wybierane są różne zestawy cech obiektów i wybór jest oceniany po zakończeniu procesu klasteryzacji.

## 1.4 Miara podobieństwa

Jako, że podobieństwo jest podstawą definicji klastra, pomiar podobieństwa dwóch rekordów z tej samej przestrzeni cech, jest bardzo ważne dla większości procedur klasteryzacji. Z powodu różnorodności typów wartości cech i ich skalowania, miara (lub miary) podobieństwa, muszą być dobierane bardzo ostrożnie. Najbardziej powszechną praktyką jest obliczanie *miary niepodobieństwa* pomiędzy dwoma rekordami używając miary odległości zdefiniowanej na przestrzeni cech. Najczęściej w przypadku cech o wartościach ciągłych stosowana jest odległość Euklidesowa.

$$d(\mathbf{x}_i, \mathbf{x}_j) = \left( \sum_{k=1}^d (\mathbf{x}_{i,k} - \mathbf{x}_{j,k})^p \right)^{1/p} = \|\mathbf{x}_i - \mathbf{x}_j\|_p$$

Odległość Euklidesowa jest intuicyjnie używane do obliczeń odległości obiektów w dwu i trzy wymiarowych przestrzeniach. Sprawuje się doskonale tak z wyizolowanymi jak i złożonymi klastrami. Wadą używania tej miary jest fakt, że szybko cechy o mniejszej skali tracą na wartości zdominowane przez cechy o większej skali. Rozwiązaniem tego problemu jest normalizacja używanych wartości lub stosowanie wag.

Obliczanie odległości pomiędzy rekordami, których część lub wszystkie cechy nie są wartościami ciągłymi, staje się o wiele trudniejsze. W ekstremalnych przypadkach wartość odległości jest określana binarnie. I tutaj jednak istnieje wiele prac, które starają się rozwiązać ten problem.

Rekordy mogą być także reprezentowane używając stringów czy struktur drzewiastych. Wymaga to wtedy syntaktycznego lub statystycznego podejścia do miary odległości. Ale są one zdecydowanie mniej przydatne.

Są także miary odległości, który pod uwagę biorą otoczenie lub sąsiednie punkty zwanych kontekstem. Miarę tej odległości można określić wzorem:

$$MND(\mathbf{x}_i, \mathbf{x}_j) = NN(\mathbf{x}_i, \mathbf{x}_j) + NN(\mathbf{x}_j, \mathbf{x}_i)$$

Gdzie  $NN(\mathbf{x}_i, \mathbf{x}_j)$  jest ilością sąsiadów punktu  $\mathbf{x}_i$  w odniesieniu do punktu  $\mathbf{x}_j$ . Tzn. funkcja ta określa, którym z kolei najbliższym sąsiadem  $\mathbf{x}_i$  jest  $\mathbf{x}_j$ . Oznacza to, że im więcej obiektów pomiędzy dwoma punktami, tym większa odległość wynikowa. Mimo, że nie miara ta nie spełnia zasad nierówności trójkątnej, została z powodzeniem zastosowana w kilku pracach i pokazuje, że *niepodobieństwo* nie musi być miarowe.

## 1.5 Techniki Klastrowania

Według (9) mamy cztery główne techniki klasteryzacji: partycjonowanie, hierarchiczne, gęstościowe, i sieć(grid).

- Technika partycyjna organizuje dane w  $k$  klastry tak, że ich odległości do punktu reprezentującego dany klaster (k-means) jest najmniejsza. Mimo, że jest to jedna z najpopularniejszych technik: (a) nie nadają się do wykrywania klastrów nie wypukłych, (b) są wrażliwe na odstające punkty i początkową selekcję centrów klastrów, (c) z góry wymaga określenia ilości klastrów, (d) nie radzi sobie z klastrami o różnych rozmiarach i gęstościach, (e) generuje kiepskiej jakości deskryptory klastrów.
- Techniki hierarchiczne produkują zagnieżdżoną sekwencję klastrów. Hierarchia może być formowana „z góry” lub „z dołu”. Każda iteracja polega na łączeniu lub rozdzielaniu klastrów na podstawie odpowiednich miar podobieństwa pomiędzy klastrami. Główną wadą tej techniki jest niemożność korekcji wyników po wykonaniu operacji (dzielenie lub łączenie) oraz wysoka skomplikowość obliczeń. Można tu wymienić takie algorytmy jak BIRCH, CURE czy CHAMELEON
- Techniki Gęstościowe grupują sąsiednie punkty w klastry bazując na lokalnej gęstości zamiast odległości pomiędzy punktami. Popularniejsze algorytmy to: DBSCAN, OPTICS, DENCLUE
- Techniki sieci (grid) organizuje przestrzeń danych w wielowymiarową sieć i agreguje punkty w komórki. Podstawowy pomysł polega na tym, aby odrzucać komórki o małej gęstości, a następnie łączyć komórki o wysokiej gęstości i formować klastry. Metoda ta jest znacząco szybka dla nisko do średnio wymiarowych danych i nieźle radzą sobie z wykrywaniem klastrów o niestandardowych kształtach. Jednakże, jako że ilość komórek rośnie gwałtownie wraz z kolejnymi wymiarami, staje się to trudne i kosztowne dla wielowymiarowych danych. Znaczącymi metodami są tu: CLIQUE, MAFIA, WaveCluster.

## 2 Algorytmy ewolucyjne w klasteryzacji

Jeśli spojrzeć na problem klasteryzacji, jako na problem optymalizacji. Czyli przy zastosowaniu dowolnej z powyższych technik, chcemy uzyskać jak najlepsze wyniki, a nierzadko niektóre z nich bardzo są wrażliwe na startowe parametry. Takie jest główne założenie Algorytmów Ewolucyjnych w zadaniu klasteryzacji. Z taką drobną różnicą, że techniki zostały zredukowane do prostych funkcji oceny.

Najprostszy podział EA można wprowadzić na podstawie ostatecznych wyników, jakie dostarczają na wyjściu (10). Są to klastry *twarde*, *miękkie* oraz *rozmyte*. Działają one dokładnie w ten sam sposób jak opisano wcześniej w tym dokumencie.



Drugi podział to algorytmy o *stałej* liczbie klastrów oraz *dynamicznej*. W pierwszym wypadku liczba klastrów jest z góry określona i nie zmienia się przez cały proces wykonywania algorytmu. Wartość tą użytkownik musi ustalić sam używając wiedzy o domenie, w której dokonuje analizy, lub metodą prób i błędów przeprowadzając kilka analiz na tym samym zbiorze danych. Dynamiczne algorytmy natomiast starają się same określić ilość klastrów i to również podlega procesowi optymalizacji.

Kolejną dziedziną podziału algorytmów Ewolucyjnych w klasteryzacji jest typ zastosowanego chromosomu do opisu pojedynczego rozwiązania (10) :

- Kodowanie binarne – jest to ciąg  $n$  bitów, gdzie  $n$  to ilość rekordów podlegających analizie. W ciągu tym kolejne bity przypisane są do kolejnych rekordów, a bit pozytywny oznacza, że ten rekord jest prototypem klastra.
- Kodowanie całkowitymi – Dzieli się na dwa możliwe zastosowania. Pierwszy to ciąg  $n$  liczb, każda przypisana kolejnemu obiektowi zbioru danych zawiera numer z przedziału  $\{1, \dots, k\}$  opisujący, do jakiego klastra dany obiekt przynależy. Ten sposób zapisu niesie ze sobą niebezpieczeństwo powtarzalności, ponieważ pojedyncze rozwiązanie może być zapisane na wiele sposobów. Innym sposobem na zastosowanie zapisu w postaci liczb całkowitych to użycie wektora  $k$ -elementowego, gdzie każdy opisuje numer obiektu ze zbioru danych użytego jako medoidów.
- Kodowanie rzeczywistymi – w postaci tej, kolejne wartości to współrzędne prototypów w metodologii bazującej na centroidach. W takim wypadku, genotyp będzie miał długość  $k \cdot d$ , gdzie  $k$  to liczba klastrów, a  $d$  to liczba wymiarów.
- Kodowanie hybrydowe lub niestandardowe -

## 2.1 Funkcje Oceny

W praktyce, każde relatywne kryterium oceniające poprawność klastra, nie będące monotoniczne względem liczby klastrów, może zostać użyte jako funkcja oceny. Jednakże dobór tej części algorytmu jest bardzo ważny jako, że kreuje on przebieg procesu jak i ostateczny wynik działania. Samych funkcji jest wiele i wymienię tutaj tylko ich mały podzbiór.

- Funkcja gęstości – średnia odległość obiektów do przypisanych im centroidów.
- Funkcja rozłączności – czyli odległość pomiędzy centroidami
- Funkcja łączności –funkcja ta sprawdza izolację klastrów średnią liczbę sąsiadów, którzy należą do tej samej grupy.

## 2.2 Wielokryterialność

W przeciwieństwie do uczenia się nadzorowanego, jak np. klasyfikacja, klasteryzacja, jako metoda nie nadzorowana, nie posiada „bazowej prawdy”, mówiącej czy rozwiązanie jest poprawne czy optymalne. To sugeruje, aby jakość rozwiązania była określana przez różnorodny zestaw kryteriów, zamiast pojedynczy. Aby likwidować złe wyniki w dowolnym kryterium poprawności.

Aby móc z powodzeniem zastosować wielokryterialność w algorytmie klasteryzacji, kryteria muszą zostać dobrane z pewną drobiazgowością. Rozpatrując różne aspekty jakości klasteryzacji. Często praktyką jest definiowanie kryteriów wręcz zaprzeczających się nawzajem. Wtedy tworzy się pojedynczą regułę, gdzie każde kryterium posiada swoją wagę ważności. Wadą tu jest fakt, że

samo obliczenie stosowanych wag, może wymagać dodatkowych przebiegów aby znaleźć najlepsze rozwiązanie.

W idealnym algorytmie wielo-kryterialnym stosuje się dominację Pareto. Polega to, w skrócie, na odnajdywaniu członków populacji, nie zdominowanych przez nikogo innego. Tzw. „Pareto front”. Dany członek populacji dominuje, jeśli we wszystkich kryteriach jest równy lub w co najmniej jednym lepszy.

### 3 Model

#### 3.1 Reprezentacja Chromosomu

Opis pojedynczego chromosomu zbudowany jest z trzech części:

1. Atrybuty  
 $A = [b, g]$   
 $b$  – liczba sąsiadów wzięta pod uwagę przy ocenie funkcją łączności.  
 $g$  – liczba medoidów jaka maksymalnie może przypadać na jeden kluster.
2. Wektor medoidów  
 $M = [m_1, m_2, \dots, m_K]$   
 $K$  – stały parametr algorytmu. Określa wielkość wektora medoidów.  
 $m_i$  – pojedynczy medoid, numer obiektu danych, który ma posłużyć jako prototyp klastra.
3. Wektor grup (klastrow)  
 $P = [p_1, p_2, \dots, p_l]$   
Określa przynależność medoidów do kolejnych klastrow  
 $l$  – liczba klastrow „proponowana” w danym rozwiązaniu. Dynamiczna i charakterystyczna tylko dla danego rozwiązania  
 $p_j$  – liczba kolejnych medoidów należących do  $j$ -tego klastra. Gdzie  $1 \leq p_j \leq g$  oraz  
 $\sum_{j=1}^l p_j = K$

#### 3.2 Inicjalizacja populacji

Kolejno wszystkie wartości są losowane. Sprawdzany jest warunek by zawartość wektora  $M$  nie powtarzała się.

#### 3.3 Funkcja Oceny

##### 3.3.1 Gęstość

Miara odległości do najbliższego prototypu klastra

$$Den = \sum_{x \in X} \delta(x, m_d)$$

Gdzie:

$x$  – pojedynczy punkt danych ze zbioru danych  $X$

$\delta(\dots)$  - funkcja odległości

$m_x$  – medoid położony „najbliżej” punktu danych

##### 3.3.2 Łączność

Miara ilości sąsiadów z tego samego klastra

$$Con = \sum_{x \in X} \gamma(d, b)$$

Algorytmy ewolucyjne w zadaniu klasteryzacji.

$$\gamma(x, b) = \sum_{j=1}^b \begin{cases} 1/b, & \text{jeśli } x \in C_k \wedge s_j \in C_k \\ 0 & \end{cases}$$

Przy czym:

$C_k$  jest dowolnym klastrem spełniającym ten warunek.

$s_j$  – jest j-tym najbliższym obiektem do danego obiektu  $x$

### 3.3.3 Rozłączność

odległość klastrów.

Wyrażane w sumie odległości każdego medoida do najbliższego medoida z obcego klastra.

### 3.3.4 Poprawność jednostki

- Powtarzanie się prototypu w jednym lub kilku klastrach

## 3.4 Operatory Ewolucyjne

### 3.4.1 Krzyżowanie

- Krzyżowanie kilku punktów wektora  $M$ , wraz podziałem na grupy

### 3.4.2 Mutacja

- Przesunięcia granicy jednej losowej grupy
- Z małym prawdopodobieństwem podmiana medoidu
- Z bardzo małym prawdopodobieństwem zmiana jednego z atrybutów

## 4 Plan Pracy

- **Kwiecień 14**
  - Zakończona aplikacja
- **Kwiecień 18**
  - Przeprowadzone testy na aplikacji
- **Maja 2**
  - Zakończenie aplikacji CUDA
- **Maja 6**
  - Przeprowadzenie testów i porównanie z poprzednią aplikacją
- **Maj 13**
  - Podsumowanie pracy i wersja ostateczna.

## 5 Bibliografia

1. *Data Mining: Introductory and Advanced Topics*. **Dunham, M. H.** brak miejsca : Prentice-Hall, 2003.
2. *Advances in Knowledge Discovery and Data Mining*. **Fayyad, U., i inni.** 1996, AAAI Press/The MIT Press.
3. *Data Mining: Concepts and Techniques*. **Han, J. i Kamber, M.** brak miejsca : Kaufmann Publishers, 2000.
4. *Data clustering: 50 years beyond K-means*. **Jain, Anil K.** brak miejsca : Elsevier, 2009, Pattern Recognition Letters.
5. *A Scatter Search Algorithm for the Automatic Clustering Problem*. **Abdule-Wahab, Rasha S., i inni.** brak miejsca : Springer, 2006, Advances in Data Mining.
6. *Data clustering: a review*. **Jain, A. K., Murty, M. N. i Flynn, P. J.** 3, brak miejsca : ACM, 1999, Comput. Surveys, Tom 31, strony 264-323.
7. *Survey of clustering algorithms*. **Xu, R. i Wunsch, D. I.I.** 16, brak miejsca : IEEE Trans, 3 2005, Neural Networks.
8. **www.merriam-webster.com.**
9. *NOCEA: A rule-based evolutionary algorithm for efficient and effective*. **Sarafis, Ioannis A., Trinder, Phil W. i Zalzala, Ali M.S.** brak miejsca : Elsevier, 2006, Science Direct.
10. *A Survey of Evolutionary Algorithms for Clustering*. **Hruschka, Eduardo Raul, i inni.**
11. *A genetic algorithm with gene rearrangement for K-means clustering*. **Changa, Dong-Xia, Zhang, Xian-Da i Zheng, Chang-Wen.** brak miejsca : Elsevier, 2008, Pattern Recognition.
12. *An Evolutionary Approach to Multiobjective Clustering*. **Handl, Julia i Knowles, Joshua.** brak miejsca : IEEE, 2007, IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION.
13. *High-dimensional data clustering*. **Bouveyron, C., Girard, S. i Schmid, C.** brak miejsca : Elsevier, 2007, Science Direct.
14. *Asurvey of kernel and spectral methods for clustering*. **Filippone, Maurizio, i inni.** brak miejsca : Elsevier, 2008, Science Direct.
15. *Apertitional clustering algorithm validated by a clustering tendency*. brak miejsca : Elsevier, 2006, Pattern recognition.
16. *Clustering data with measurement errors*. **Kumar, Mahesh i Patel, Nitin R.** brak miejsca : ELSEVIER, 2007, Computational Statistics & Data Analysis.
17. *Dynamic Clustering Using Multi-objective Evolutionary Algorithm*. **Chen, Enhong i Wang, Feng.** brak miejsca : Springer, 2005, Computational Intelligence and Security.
18. *An artificial bee colony approach for clustering*. **Zhang, Changsheng, Ouyang, Dantong i Ning, Jiaxu.** brak miejsca : Elsevier, 2010, Expert Systems with Applications.
19. *A method of relational fuzzy clustering based on producing feature vectors using FastMap*. **Brouwer, Roelof Kars.** brak miejsca : Elsevier, 2009, Information Sciences.
20. *Parallelizing Evolutionary Algorithms for Clustering Data*. **Kwedlo, Wojciech.** brak miejsca : Springer, 2006, Parallel Processing and Applied Mathematics.
21. *Performance evaluation of density-based clustering methods*. **Aliguliyev, Ramiz M.** brak miejsca : Elsevier, 2009, Information Sciences.
22. *Parallel Evolutionary Algorithms*. **Osmera, Pavel, Lacko, Bronislav i Petr, Masopust.** brak miejsca : IEEE, 2003, Computational Intelligent in Robotics and Automation.

23. *Multi-objective Optimization Technique Based on Co-evolutionary Interactions in Multi-agent System.* **Dreżewski, Rafał i Siwik, Leszek.** brak miejsca : Springer, 2007, Applications of Evolutionary Computing.

24. *An overview of evolutionary algorithms: practical issues and common pitfalls.* **Whitley, Darrell.** brak miejsca : Elsevier, 2001, Information and Software Technology.

## 6 Spis Ilustracji

rys 1 Przykład zbioru danych i wyniku jego klasteryzacji. ....	2
rys 2 . ukazuje prosty przykład. Punkty, w tej przestrzeni 2D cech, są zorganizowane w kształt mniej więcej równo oddalony od jednego punktu środka. Jeśli wybrać współrzędne Kartezjańskie, większość algorytmów klastrowania prawdopodobnie podzieli tę figurę na dwa lub więcej klastrów. Jeśli by jedna użyć reprezentacji koordynatów polarnych, uzyskanie pojedynczego klastra ma większe prawdopodobieństwo. ....	3