Informatyka stosowana, studia dzienne, II st. semestr I

Obliczenia ewolucyjne **2019/2020**

Prowadzący: dr inż. Łukasz Chomątek środa, 14:00

Bartosz Jurczewski 234067 234067@edu.p.lodz.pl

Karol Podlewski 234106 234106@edu.p.lodz.pl

Zadanie 3: Klastrowanie przy pomocy algorytmu genetycznego

1. Cel

Celem zadania była implementacja algorytmu genetycznego rozwiązującego problem podziału zbioru danych na klastry - problemy do rozwiązania obejmowały dobranie funkcji celu, sposobu reprezentacji osobników, sposób selekcji, mutacji oraz krzyżowania.

2. Opis zagadnienia na podstawie literatury

Odpowiednio dostosowany algorytm genetyczny może posłużyć jako nowe narzędzie w klasteryzacji. Algorytm operuje na populacji o liczebności **p**, a każdy z osobników poprzez swój chromosom reprezentuje przypisanie **x** punktów do **k** klastrów. Co nową generację, populacja przechodzi przez proces *selekcji* (wybieramy najlepiej przystosowane osobniki do reprodukcji), a następnie ich genotypy są poddawane operatorom ewolucyjnym: *krzyżowaniu* oraz *mutacji*. Algorytm genetyczny dobiega końca kiedy osiąga warunek stopu, na przykład z góry określoną liczbę generacji. [1] [2]

2.1. Kodowanie osobnika

Każdy osobnik ma swój własny chromosom który składa się z genów. W przypadku klasteryzacji, każdy z genów reprezentuje jeden z centrów klastra. Moc tego zbioru równa jest ilości klastrów.

Mając zbiór **k** punktów, możemy ułożyć je oraz oznaczyć indeksami od 0 do **k**. Gen musi przyjmować wartości z pewnego zbioru wartości dopuszczalnych czyli 0 do **k**.

2.2. Funkcja celu

Funkcja celu służy jako narzędzie do obliczenia poziomu przystosowania osobnika. Odpowiedni dobór funkcji celu jest jednym z zagadnień podczas tworzenia algorytmu genetycznego i ma bezpośrednie przełożenie na jakość wypracowanego rozwiązania.

2.3. Selekcja

Podczas każdego kolejnego pokolenia wybierana jest część istniejącej populacji do wyhodowania nowego pokolenia. Indywidualne osobniki są wybierane w procesie opartym na przystosowaniu, w którym wybierane są bardziej przystosowane osobniki (mierzone funkcją dopasowania). Niektóre metody selekcji oceniają przydatność każdego osobnika i preferencyjnie wybierają najlepsze rozwiązania. Inne metody oceniają tylko losową próbę populacji, ponieważ pierwszy proces może być bardzo czasochłonny.

Popularnymi metoda selekcji są strategia elitarności (ang. *Elite Selection*) oraz metoda ruletki (ang. *Roulette Wheel Selection*). [3]

2.4. Krzyżowanie

Krzyżowanie jest operatorem genetycznym używanym do zróżnicowania kodowania chromosomu z jednej generacji do kolejnej. Jest to analogiczne

do rozmnażania i krzyżowania biologicznego, na którym oparte są algorytmy genetyczne. Krzyżowanie to proces polegający na pobieraniu rozwiązań z więcej niż jednego rodzica i tworzeniu od nich rozwiązania podrzędnego.

2.5. Mutacja

Mutacja jest operatorem genetycznym używanym do utrzymania różnorodności genetycznej od jednego pokolenia populacji chromosomów algorytmu genetycznego do następnego. Jest to analogiczne do mutacji biologicznej. Mutacja zmienia jedną lub więcej wartości genów w chromosomie ze stanu początkowego. W przypadku mutacji rozwiązanie może się całkowicie zmienić w stosunku do poprzedniego rozwiązania. Stąd algorytm genetyczny może dojść do lepszego rozwiązania poprzez użycie mutacji. Mutacja zachodzi podczas ewolucji zgodnie z prawdopodobieństwem mutacji definiowanym przez użytkownika. Prawdopodobieństwo to powinno być niskie. Jeśli jest ustawiony zbyt wysoko, wyszukiwanie zmieni się w wyszukiwanie losowe.

Celem mutacji jest zachowanie i wprowadzenie różnorodności. Mutacja powinna pozwolić algorytmowi na uniknięcie lokalnych minimów poprzez zapobieganie zbytniego podobieństwa populacji chromosomów do siebie, spowalniając w ten sposób lub nawet zatrzymując ewolucję. [4]

3. Opis implementacji

Zadanie zostało zrealizowane przy użyciu frameworka *.NET Core* w wersji 3.1, języka *C#* , z wykorzystaniem bibliotek: *GeneticSharp* (algorytm genetyczny) oraz *CommandLineUtils* (przekazywanie argumentów z wiersza poleceń).

Wykorzystany został zbiór danych *MallCustomers.csv*, który został zredukowany do dwóch wymiarów: jego cechy jakościowe zostały przekształcone w cechy ilościowe, następnie zbiór został znormalizowany oraz poddany analizie głównych składowych w celu redukcji wymiarów. Zbiór nie posiada z góry przypisanych klastrów, dlatego w celu ewaluacji poprawnego podziału można wykorzystać jedynie kryteria wewnętrzne.

3.1. Funkcje celu

W przygotowanym rozwiązaniu użytkownik może wybrać jedną z dwóch funkcji celu:

1. **Współczynnik zarysu** (ang. *Silhouette coefficient*) - obliczany jako średnia różnica między odległością do najbliższego oraz drugiego najbliższego skupienia podzielona przez większą z tych dwóch wartości. [5]
2. **Współczynnik kary** - autorski współczynnik, który ocenia dopasowanie w zależności od liczy punktów leżących bliżej niż podana odległość, a należących do innego klastra oraz punktów należących do danego klastra,

które leżą dalej od podanej odległości. Szerokość kary dla danego punktu x_i oraz danego klastra c z podaną odległością D wynosi:

$$p(x_i, c, D) = \begin{cases} 2 & \text{jeżeli } x_i \in c \text{ oraz } d(x) > D \\ 1 & \text{jeżeli } x_i \ni c \text{ oraz } d(x) < D \\ 0 & \text{w przeciwnym wypadku} \end{cases} \quad (1)$$

W celu obliczenia współczynnika kary dla odległości D musimy wyliczyć średnią wartość współczynnika dla każdego klastra, którą obliczamy odejmując od liczby punktów sumę szerokości kary dla każdego punktu, co następnie dzielimy przez liczbę punktów:

$$\text{Penalty}(D) = \text{mean}_{\ c \in C} \ \frac{|x| - \sum_{i=1}^{|x|} p(x_i, c, D)}{|x|} \quad (2)$$

Współczynnik kary nakłada większą karę na punkty należące do klastra znajdujące się poza wyznaczoną odległością niż na punkty spoza klastra leżące bliżej niż podana odległość - chcemy w ten sprawić by algorytm nie promował zbyt dużych klastrów, ale wciąż miał na uwadze jak ważna jest integralność wewnątrz klastra.

Wartości dopasowania dla każdej z tych funkcji mieszczą się w zakresie <-1, 1> - nie oznacza to jednak, że możemy bezpośrednio porównywać wyniki uzyskane jedną oraz drugą metodą.

3.2. Kodowanie oraz generacja populacji porządkowej

W zaproponowanym rozwiązaniu każdy chromosom składa się z określonej liczby genów (od 2 do 8) odpowiadającej liczbie klastrów. Każdy gen zawiera informacje o środku klastra w postaci identyfikatora numeru obserwacji. Podczas tworzenia chromosomu nie jest możliwe, by posiadał on dwa identyczne geny. Poszczególne obserwacje przypisywane są do klastrów w zależności od ich odległości - dana obserwacja należy do klastra, którego środek znajduje się najbliżej niej samej.

3.3. Selekcja

W przygotowanym rozwiązaniu użytkownik może wybrać jedną z dwóch metod selekcji znanych z literatury:

1. **Elite Selection** - wybiera najlepsze chromosomy pod względem dopasowania,
2. **Roulette Wheel Selection** - metoda koła ruletki, która polega na przypisaniu każdemu osobnikowi prawdopodobieństwa selekcji (im większe przystosowanie, tym większe prawdopodobieństwo), a następnie wylosowaniu chromosomów z całej puli.

3.4. Mutacja

W przygotowanym algorytmie wykorzystaliśmy **autorską mutację**. Zasada jej działania jest następująca: jeśli liczba klastrów jest większa niż 6, usuń jeden klaster. Jeśli jest mniejsza niż 4 dodaj jeden klaster.

3.5. Krzyżowanie

W przygotowanym rozwiązaniu użytkownik może wybrać jedną z dwóch metod krzyżowania znanych z literatury, ale dostosowanych do chromosomów o zmiennej liczbie genów:

1. Modyfikacja **Uniform Crossover** - tworzone są dwie kopie losowych rodziców. Krzyżowanie następuje z prawdopodobieństwem zdefiniowanym przez użytkownika, a krzyżowanych jest tyle genów, ile posiada mniejszy z rodziców. Dłuższy rodzic przekazuje swoje niekrzyżowane geny jednemu potomkowi, dzięki czemu każde dziecko zawiera tyle samo genów co jeden z rodziców.
2. Modyfikacja **Three Parent Crossover** - selekcja, która polega na wybraniu trzech losowych rodziców - chromosom będzie miał tyle genów, ile genów ma trzeci z rodziców. Po kolei porównywany jest każdy gen pierwszego oraz drugiego rodzica - jeżeli jest identyczny, przechodzi on do potomstwa, jeżeli się różni - brany jest gen od trzeciego rodzica. W momencie, kiedy porównanie nie jest możliwe, ze względu na brak genów do porównania u któregokolwiek z rodziców, dalsze krzyżowanie jest przerwane, a nowy chromosom uzupełniany jest ostatnimi genami trzeciego z rodziców.

Dodatkowo, krzyżowanie pojedynczych genów nie następuje, kiedy miałoby ono doprowadzić do posiadana przez chromosom dwóch identycznych genów.

4. Wyniki

Dla każdego wariantu zostały wykonana seria 5 pomiarów, z których w wynikach uwzględnia była wartość środkowa funkcji dopasowania.

4.1. Wpływ funkcji celu

Argumenty stałe dla tej sekcji badań:
— **Wielkość populacji:** 1000.
— **Ilość generacji:** 1000.
— **Prawdopodobieństwo mutacji:** 0,1

Tabela 1. Porównanie funkcji celu - Współczynnik zarysu, dla różnych metod krzyżowania oraz selekcji.

Krzyżowanie	**Selekcja**	**Przystosowanie**	**Liczba klastrów**
Uniform	Elite	0,617	2
Uniform	Roulette Wheel	0,590	2
Three Parent	Elite	0,578	4
Three Parent	Roulette Wheel	0,580	3

Tabela 2. Porównanie wpływu krzyżowania oraz selekcji dla funkcji celu - Współczynnik kary.

Krzyżowanie	**Selekcja**	**Przystosowanie**	**Liczba klastrów**
Uniform	Elite	0,975	4
Uniform	Roulette Wheel	0,975	4
Three Parent	Elite	0,938	4
Three Parent	Roulette Wheel	0,975	4

4.2. Wpływ liczby generacji

Argumenty stałe dla tej sekcji badań:
— **Wielkość populacji:** 50.
— **Prawdopodobieństwo mutacji:** 0,1
— **Krzyżowanie:** Uniform
— **Selekcja:** Elite

Tabela 3. Wpływ liczby generacji na przystosowanie oraz liczbę klastrów dla autorskiej współczynnika zarysu

Liczba generacji	**Przystosowanie**	**Liczba klastrów**
50	0,541	2
100	0,558	4
250	0,567	3
500	0,58	5
1000	0,586	4
2000	0,579	4

Tabela 4. Wpływ liczby generacji na przystosowanie oraz liczbę klastrów dla autorskiej funkcji celu.

Liczba generacji	**Przystosowanie**	**Liczba klastrów**
50	0,912	4
100	0,892	4
250	0,909	5
500	0,924	4
1000	0,928	4
2000	0,939	4

4.3. Wpływ liczby populacji

Na podstawie wyników z poprzedniego punktu badania kontynuowaliśmy dla funkcji celu - Współczynnik kary.

Argumenty stałe dla tej sekcji badań:
— **Liczba generacji:** 1000.
— **Prawdopodobieństwo mutacji:** 0,1
— **Krzyżowanie:** Uniform
— **Selekcja:** Elite

Tabela 5. Wpływ liczby populacji na przystosowanie oraz liczbę klastrów dla współczynnika zarysu.

Liczba populacji	**Przystosowanie**	**Liczba klastrów**
5	0,55	4
10	0,574	4
50	0,548	4
100	0,581	4
500	0,608	2
1000	0,587	2

Tabela 6. Wpływ liczby populacji na przystosowanie oraz liczbę klastrów dla autorskiej funkcji celu.

Liczba populacji	**Przystosowanie**	**Liczba klastrów**
5	0,788	6
10	0,773	5
50	0,908	4
100	0,967	4
500	0,972	4
1000	0,975	4

4.4. Wpływ prawdopodobieństwa mutacji

Argumenty stałe dla tej sekcji badań:
— **Wielkość populacji:** 1000.
— **Liczba generacji:** 1000.
— **Krzyżowanie:** Uniform
— **Selekcja:** Elite

Tabela 7. Wpływ prawdopodobieństwa mutacji na przystosowanie oraz liczbę klastrów dla współczynnika zarysu.

Prawdopodobieństwo mutacji	**Przystosowanie**	**Liczba klastrów**
0,05	0,611	2
0,1	0,585	3
0,2	0,608	2
0,5	0,606	2
0,8	0,611	2
0,9	0,616	2

Tabela 8. Wpływ prawdopodobieństwa mutacji na przystosowanie oraz liczbę klastrów dla autorskiej funkcji celu.

Prawdopodobieństwo mutacji	**Przystosowanie**	**Liczba klastrów**
0,05	0,975	4
0,1	0,975	4
0,2	0,975	4
0,5	0,975	4
0,8	0,975	4
0,9	0,972	4

5. Dyskusja

Przeprowadzone przez nas badanie wykazało, że zastosowanie krzyżowania *Uniform* było korzystne dla obydwu funkcji celu. Wybór metody selekcji nie miał żadnego wpływu na przystosowanie dla autorskiej funkcji celu. Zastosowanie krzyżowania *Uniform* korzystnie wpłynęło za to na wyniki uzyskane przy wykorzystaniu współczynnika zarysu jako funkcji celu. Różnice te jednak są niewielkie i przy wykorzystaniu innego zbioru danych wykorzystanie krzyżowania *Three Parent* czy selekcji *Roulette Wheel* może korzystanie wpłynąć na wartość dopasowania.

Badania dotyczące wpływu liczby generacji jasno pokazały, że sukcesywne zwiększanie liczby generacji zwiększa wartość przystosowania. Jednocześnie liczba klastrów pozostała bez zmian. Porównując badanie dla 50 oraz 2000 generacji, można zauważyć kilku procentowy wzrost. Należy jednak podkreślić że zwiększanie liczby generacji wpływa na wydłużenie czasu pracy programu.

Kolejna sekcja pokazuje wpływ liczby populacji na przystosowanie oraz liczby klastrów. Jednoznacznie wartości 5 oraz 10 skutkowały w zadowalającym przystosowaniu (ok. 0,7). Wartość 50 zwiększyła przystosowanie do około 0,9, jednakże to wartości od 100 w górę pozwoliły osiągnąć pułap przystosowania na poziomie 0,97. Równocześnie liczba klastrów od poziomu przystosowania 0,9 pozostała bez większych zmian - dla współczynnika zarysu zazwyczaj 2, dla współczynnika kary zazwyczaj 4.

Ostatnią sekcją badań była ocena naszej autorskiej mutacji. Dane jasne pokazują brak jakiekolwiek wpływu na przystosowanie oraz liczbę klastrów. Jej obecność była obojętna, nawet dla prawdopodobieństwa mutacji rzędu 0,9 - tak wysoka mutacja jedynie lekko obniżała przystosowanie.

6. Wnioski

— Wykorzystanie znanych z literatury metod selekcji i krzyżowania nie ma znaczącego wpływu dla przystosowania w klastrowaniu za pomocą algorytmów genetycznych w wykorzystanym przez nas zbiorze.
— Zwiększenie liczby generacji zwiększa wartość przystosowania, przy jednoczesnym wydłużeniu czasu pracy programu.
— Zwiększanie liczby populacji również zwiększa wartość przystosowania.
— Zwiększanie liczby populacji oraz generacji od pewnego momentu nieznacznie wpływa na uzyskane wyniki, znacznie wpływając na czas potrzeby do zakończenia działania algorytmu genetycznego.
— Zastosowana przez nas mutacja nie ma żadnego wpływu na proces klastrowania.

Bibliografia

[1] *A genetic algorithm approach to cluster analysis,* M.C. Cowgill, R.J. Harvey, L.T. Watson, https://www.sciencedirect.com/science/article/pii/S0898122199000905

[2] *Using Genetic Algorithm for Selection of Initial Cluster Centers for the K-Means Method,* W. Kwedlo, P. Iwanowicz, https://link.springer.com/chapter/10.1007/978-3-642-13232-2_20

[3] *A review of selection methods in genetic algorithm,* R. Sivaraj, http://www.ijest.info/docs/IJEST11-03-05-190.pdf

[4] *A Faster Genetic Clustering Algorithm,* L Meng,Q H Wu,Z Z Yong, https://link.springer.com/chapter/10.1007/3-540-45561-2_3

[5] *Performance Evaluation of the Silhouette Index,* A. Starczewski, A. Krzyżak, https://link.springer.com/chapter/10.1007/978-3-319-19369-4_5

[6] *Genetic algorithm-based clustering technique,* Ujjwal Maulika, Sanghamitra Bandyopadhyay, https://www.sciencedirect.com/science/article/abs/pii/S0031320399001375