Szymon Nowak-Trzos 411246

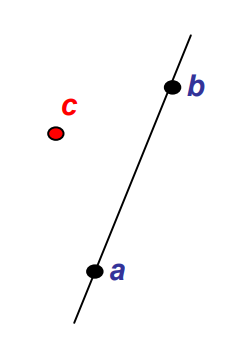
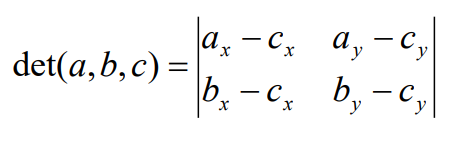
Informatyka rok II, gr.2 środa nieparzysta 13:00-14:30

Algorytmy Geometryczne – Ćwiczenie 1 Sprawozdanie

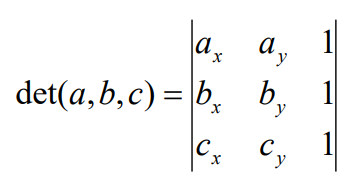
1. Wstęp
   1. Cel Ćwiczenia

Celem ćwiczenia była implementacja algorytmu sprawdzającego położenie punktów w stosunku do prostej przeprowadzonej przez dwa punkty oraz przeprowadzenie analizy wyników obliczeń na kilku zbiorach danych.

* 1. Teoria

Aby określić, czy punkt **c** leży po prawej, po lewej stronie prostej **ab** czy na niej należało określić wyznacznik którejkolwiek z poniższych macierzy:

**1)**



**2)**

Wyznacznik równy 0 wskazuje, że punkt **c** jest współliniowy z **a** i **b**. W przeciwnym wypadku znak wyznacznika rozróżnia punkty po lewej i po prawej stronie **ab.**

* 1. Użyte Narzędzia

Ćwiczenie zostało wykonane dzięki zmodyfikowanemu narzędziu graficznemu załączonego na stronie UPEL. Do niezbędnych obliczeń został wykorzystany język programowania Python oraz biblioteka numpy. Ponadto wykresy zostały wygenerowano przy pomocy biblioteki matplotlib.

1. Wykonanie Ćwiczenia
   1. Zbiory Danych

Wygenerowane zostały cztery zbiory danych:

1. 10^5 losowych punktów o współrzędnych z przedziału [-1000, 1000],
2. 10^5 losowych punktów o współrzędnych z przedziału [-10^14, 10^14],
3. 1000 losowych punktów leżących na okręgu o środku (0,0) i promieniu R=100,
4. 1000 losowych punktów o współrzędnych z przedziału [-1000, 1000] leżących na prostej wyznaczonej przez wektor ([-1, 0], [1, 0.1]).

* 1. Wyznaczniki

Do określenia położenia punktów w każdym zbiorze danych zostały wykorzystane cztery różne algorytmy liczenia wyznaczników:

* „self\_3x3” – zaimplementowany samodzielnie wyznacznik macierzy 3x3,
* „self\_2x2” – zaimplementowany samodzielnie wyznacznik macierzy 2x2,
* „np\_3x3” – algorytm wyznaczania wyznacznika macierzy 3x3 z biblioteki numpy,
* „np\_2x2” – algorytm wyznaczania wyznacznika macierzy 2x2 z biblioteki numpy.
  1. Tolerancja – **ε**

Ze względu na niedoskonałą precyzję obliczeń w komputerze zastosowano kilka różnych granic tolerancji **ε**:

* **ε1** = 10^(-13)
* **ε2** = 10^(-14)
* **ε3** = 10^(-15)
* **ε4** = 10^(-16)
  1. Liczba wykresów

Wszystkie kombinacje powyższych czynników zostały przedstawione na 256 wykresach.

Ze względu na wysoką liczbę wykresów nazwy zostały skrócone używając kodu:

Algorytm-wyznacznika\_Rodzaj-tolerancji\_Nazwa-zbioru-danych,

czyli dla zbioru A, dla wyznacznika „self\_2x2” oraz dla tolerancji 10^(-15) wykres będzie nosił nazwę: „self\_2x2 **ε**: 10^(-15) [-1000, 1000] -> 10^5”

1. Wykresy

Wykresy zostały przedstawione w kolejności: tolerancja->zbiór->wyznacznik.

Kolorem czerwonym zostały zaznaczone punkty po lewej od prostej, kolorem fioletowym po prawej, a kolorem czarnym te, które leżą na prostej.

* 1. Tolerancja **ε1**
     1. Wykresy całościowe

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

* + 1. Wykresy częściowe

|  |  |  |
| --- | --- | --- |
| Prawa Strona | Współliniowe | Lewa Strona |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

* 1. Tolerancja **ε2**
     1. Wykresy całościowe

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

* + 1. Wykresy częściowe

|  |  |  |
| --- | --- | --- |
| Prawa Strona | Współliniowe | Lewa Strona |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

* 1. Tolerancja **ε3**
     1. Wykresy całościowe

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

* + 1. Wykresy częściowe

|  |  |  |
| --- | --- | --- |
| Prawa Strona | Współliniowe | Lewa Strona |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

* 1. Tolerancja **ε4**
     1. Wykresy całościowe

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

* + 1. Wykresy częściowe

|  |  |  |
| --- | --- | --- |
| Prawa Strona | Współliniowe | Lewa Strona |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. Analiza Danych
   1. Różnice w podziale punktów w zbiorach danych
      1. Różnice w zbiorach A i C

|  |  |
| --- | --- |
|  |  |

* + 1. Różnice w zbiorach B i D

|  |  |
| --- | --- |
|  |  |

* 1. Analiza Różnic

Różnice pomiędzy wyznacznikami są zauważalne tylko dla zbiorów **B** i **D**, a pomiędzy zakresem tolerancji jedynie dla zbioru **D**, więc to właśnie na nim się skupimy w dalszej części analizy.

Można zauważyć, że algorytm **„self\_3x3”** z największą dokładnością wskazuje prawidłową współliniowość, bo dopiero przy **ε3** zaczyna się znacząco mylić. Obie implementacje wyznacznika macierzy 2x2 już przy **ε1** źle klasyfikują punkty, aczkolwiek przez cały czas ~70% punktów wskazują dobrze. Prawdopodobieństwo dobrego zaklasyfikowania punktu poprzez algorytm z biblioteki numpy do obliczania wyznacznika macierzy 3x3 wraz ze zmniejszaniem się tolerancji dąży do 33%, a więc nie różni się on od zwykłego zgadywania.

1. Wnioski

Komputerowe sprawdzanie współliniowości punktów, choć proste i przystępne, nie jest jednak w pełni oczywiste. Nawet podczas tak prostego problemu małe różnice implementacyjne mogą prowadzić do ogromnych różnic w wynikach analizy. Wprawdzie na ¾ zbiorów danych nie było zauważalnej różnicy, to na sytuacji granicznej wszystkie zaimplementowane algorytmy miały swoje wady. Samodzielnie zaimplementowany algorytm do obliczania wyznacznika 3x3 bardzo dokładnie wyznaczył współliniowość punktów, aczkolwiek to algorytm „self\_2x2” miał największe prawdopodobieństwo dobrej kwalifikacji punktów w mniejszych przedziałach tolerancji.