

LAPORAN TUGAS KECIL 2 IF2211 STRATEGI ALGORITMA

SEMESTER II TAHUN 2021/2022

Implementasi Convex Hull untuk Visualisasi Tes *Linear Separability Dataset* dengan Algoritma *Divide and Conquer*

Adelline Kania Setiyawan / 13520084 / K-3

I. ALGORITMA *DIVIDE AND CONQUER*

Inisiasi Data

1. Program akan memberi terlebih dahulu dataset yang tersedia. Pada tugas ini, program hanya bisa melakukan analisis Convex Hull pada data Iris, Breast Cancer, dan Wine. Untuk dataset lain, seperti load_diabetes dan load_digits, tidak bisa digunakan karena jumlah targetnya sangat banyak sehingga tidak mencukupi jumlah warna yang tersedia. Sedangkan untuk dataset load_linnerud dan load_boston tidak bisa dibaca oleh program.
2. Program akan meminta input berupa dataset yang ingin digunakan serta 2 index kolom dari dataset untuk dijadikan axis dan ordinat. Setelah itu, program akan membuat sebuah dataframe agar data siap untuk dianalisis dan divisualisasikan

Convex Hull

1. Program akan memanggil fungsi myConvexHull() dengan data yang ingin ditentukan *convex hull* sebagai parameternya. Pada fungsi myConvexHull(), program akan mencari titik dengan nilai axis terendah dan tertinggi dari titik-titik yang terdapat pada data serta menghubungkan 2 titik tersebut, misal titik base 1 (xBase1, yBase1) untuk titik dipaling kiri dan titik base 2 (xBase2, yBase2) untuk titik dipaling kanan, sebagai garis yang membagi wilayah titik-titik tersebut menjadi 2 bagian, bagian atas dan bagian bawah. Selanjutnya, program akan mengiterasi semua titik lainnya pada data tersebut dan menentukan apakah titik tersebut berada di bagian atas atau bawah garis dengan memanfaatkan konsep determinan. Nilai determinan positif menandakan bahwa titik tersebut berada di atas garis, sedangkan nilai determinan negatif menandakan bahwa titik tersebut berada di bawah garis. Index titik-titik yang berada di atas dan bawah garis akan disimpan pada 2 array yang berbeda, misal array topArea dan bottomArea.
2. Setelah 2 titik pada dataset terbagi menjadi 2, program akan memanggil fungsi findHull() untuk menentukan hull yang bisa dibentuk dari 2 wilayah dataset tersebut. Fungsi findHull() akan memiliki parameter, yaitu xBase1, yBase1, xBase2, yBase2, array yang

berisi index-index titik yang berada di satu wilayah, index titik base 1, index titik base 2, *boolean value* yang menandakan apakah wilayah tersebut berada di atas atau bawah garis, array index titik yang menjadi hull (bersifat sebagai input/output), dan dataset.

3. Pada fungsi findHull(), wilayah titik-titik pada array akan dibagi menjadi 2 lagi (konsep *divide and conquer*), yaitu dengan cara:
 - Mencari titik dengan jarak terjauh antara titik tersebut dengan garis yang dibentuk oleh titik base 1 dan base 2.
 - Mengelompokkan titik-titik pada wilayah tersebut menjadi 2 bagian lagi, bagian sebelah kiri dan kanan. Sebuah titik akan berada di sebelah kiri apabila determinan titik tersebut dengan garis yang dibentuk oleh titik base 1 dan titik dengan jarak terjauh bernilai positif (untuk wilayah di atas garis) dan negatif (untuk wilayah di bawah garis), begitu pula sebaliknya. Titik yang berada di sebelah kanan apabila determinan titik tersebut dengan garis yang dibentuk oleh titik dengan jarak terjauh dan titik base 2 bernilai negatif (untuk wilayah di atas garis) dan positif (untuk wilayah di bawah garis)
4. Setelah wilayah terbagi menjadi 2, fungsi findHull() akan dipanggil lagi untuk array kumpulan index titik-titik yang berada di wilayah kiri dan kanan. Fungsi ini akan terus secara rekursif dipanggil. Fungsi akan mencapai basisnya jika array yang dijadikan input parameternya kosong. Hal ini menandakan sudah tidak ada titik lain yang berada di luar titik convex hull yang dibuat, baik di sebelah kiri maupun kanan. Pada akhirnya fungsi findHull() akan menambahkan array berupa index 2 titik pada dataset yang harus dihubungkan sebagai *convex hull*. Fungsi myConvexHull() akan mengembalikan matrix yang berisi array-array yang ditambahkan oleh fungsi findHull() atau dalam kata lain. fungsi myConvexHull() akan mengembalikan matrix yang berisi array-array yang berisi index titik-titik pada dataset yang harus terhubung untuk dapat membuat sebuah *convex hull*.

II. **SOURCE PROGRAM**

Modul Convex Hull:

```
# MODUL CONVEX HULL

import math
from numpy.linalg import norm

# Fungsi untuk menentukan apakah titik (xc,yc) berada di atas/kiri atau bawah/kanan
```

```

# garis yang dibentuk oleh titik (x1,y1) dan (x2,y2) dengan memanfaatkan konsep determinan
def determinant(xc, yc, x1, y1, x2, y2):
    result = (x1 * y2) + (xc * y1) + (x2 * yc) - (xc * y2) - (x2 * y1) - (x1 * yc)
    return result

# Fungsi untuk menentukan jarak titik (xc,yc) dengan garis yang dibentuk
# oleh titik (x1,y1) dan (x2,y2)
def distance(xc, yc, x1, y1, x2, y2):
    p1 = np.array([x1, y1])
    p2 = np.array([x2, y2])
    p3 = np.array([xc, yc])
    result = norm(np.cross(p2 - p1, p1 - p3)) / norm(p2-p1)
    return result

# Fungsi rekursif untuk menentukan hull dari titik-titik suatu bangun ruang
def findHull(xBase1, yBase1, xBase2, yBase2, arrIdx, idxBase1, idxBase2, isTop,
            finalHull, myData):
    # Basis: apabila sudah tidak ada titik-titik di luar garis (arrIdx kosong)
    if len(arrIdx) == 0:
        # Titik (xBase1, yBase1) dan (xBase2, yBase2) membentuk Convex Hull
        matrixHull = [idxBase1, idxBase2]
        finalHull.append(matrixHull)

    # Rekurens: masih terdapat titik-titik di luar garis (arrIdx tidak kosong)
    else:
        # Mencari titik terjauh (xHull,yHull) dari garis yang dibentuk oleh titik (xBase1,
        # yBase1) dan (xBase2, yBase2)
        idx = 0
        maxDist = distance(myData[arrIdx[idx]][0], myData[arrIdx[idx]][1], xBase1, yBase1,
                           xBase2, yBase2)
        for i in range(1, len(arrIdx)):
            xc = myData[arrIdx[i]][0]
            yc = myData[arrIdx[i]][1]
            dist = distance(xc, yc, xBase1, yBase1, xBase2, yBase2)
            if dist > maxDist:
                idx = i
                maxDist = dist

        # Titik terjauh adalah titik yang akan menjadi titik pembentuk ConvexHull
        xHull = myData[arrIdx[idx]][0]
        yHull = myData[arrIdx[idx]][1]

        # Menginisiasi dan mengisi array dengan index titik yang berada di
        # luar (kiri atau kanan) titik-titik pembentuk Convex Hull
        leftArea = []

```

```

rightArea = []

for i in range(len(arrIdx)):
    xVal = myData[arrIdx[i]][0]
    yVal = myData[arrIdx[i]][1]

    # Menghitung determinan titik (xVal, yVal) dengan garis yang dibentuk oleh
    # titik (xBase1, yBase1) dan (xHull, yHull)
    det = determinant(xVal, yVal, xBase1, yBase1, xHull, yHull)
    # Sebuah titik akan dianggap berada pada garis apabila determinannya <=
    # |0.0001|
    if not(det < 0.0001 and det > -0.0001):
        # Apabila garis merupakan garis yang berada di atas garis awal,
        # titik yang berada di luar garis adalah titik berdeterminan positif
        # sedangkan apabila garis merupakan garis yang berada di bawah garis awal,
        # titik yang berada di luar garis adalah titik berdeterminan negatif
        if isTop:
            if det > 0:
                leftArea.append(arrIdx[i])
            else:
                if det < 0:
                    leftArea.append(arrIdx[i])

        # Menghitung determinan titik (xVal, yVal) dengan garis yang dibentuk oleh
        # titik (xHull, yHull) dan (xBase2, yBase2)
        det = determinant(xVal, yVal, xHull, yHull, xBase2, yBase2)
        if not(det < 0.0001 and det > -0.0001):
            if isTop:
                if det > 0:
                    rightArea.append(arrIdx[i])
            else:
                if det < 0:
                    rightArea.append(arrIdx[i])

    if isTop:
        findHull(xBase1, yBase1, xHull, yHull, leftArea, idxBase1, arrIdx[idx], True,
                finalHull, myData)
        findHull(xHull, yHull, xBase2, yBase2, rightArea, arrIdx[idx], idxBase2, True,
                finalHull, myData)
    else:
        findHull(xBase1, yBase1, xHull, yHull, leftArea, idxBase1, arrIdx[idx], False,
                finalHull, myData)
        findHull(xHull, yHull, xBase2, yBase2, rightArea, arrIdx[idx], idxBase2,
                False, finalHull, myData)

# Fungsi utama untuk menentukan Convex Hull bangun ruang

```

```

def myConvexHull(myData):
    # Mencari titik dengan nilai axis terendah dan tertinggi
    minIdx = 0
    minVal = myData[minIdx][0]
    maxIdx = 0
    maxVal = myData[maxIdx][0]

    for i in range(1, len(myData)):
        if myData[i][0] < minVal:
            minIdx = i
            minVal = myData[minIdx][0]
        if myData[i][0] > maxVal:
            maxIdx = i
            maxVal = myData[maxIdx][0]

    # Titik (xBase1, yBase1) adalah titik dengan nilai axis terkecil,
    # berada di paling kiri
    xBase1 = myData[minIdx][0]
    yBase1 = myData[minIdx][1]

    # Titik (xBase2, yBase2) adalah titik dengan nilai axis terbesar,
    # berada di paling kanan
    xBase2 = myData[maxIdx][0]
    yBase2 = myData[maxIdx][1]

    # Menginisiasi dan mengisi array dengan index titik yang berada di
    # luar (atas atau bawah) titik-titik pembentuk Convex Hull
    topArea = []
    bottomArea = []

    for i in range(len(myData)):
        xVal = myData[i][0]
        yVal = myData[i][1]
        det = determinant(xVal, yVal, xBase1, yBase1, xBase2, yBase2)

        # Sebuah titik akan dianggap berada pada garis apabila determinannya <= |0.0001|
        if not(det < 0.0001 and det > -0.0001):
            # Apabila determinan bernilai positif, posisi titik berada di atas
            # garis awal, sedangkan apabila determinan bernilai negatif, posisi
            # titik berada di bawah garis awal
            if det > 0:
                topArea.append(i)
            else:
                bottomArea.append(i)

    finalHull = []

```

```

    findHull(xBase1, yBase1, xBase2, yBase2, topArea, minIdx, maxIdx, True, finalHull,
            myData)
    findHull(xBase1, yBase1, xBase2, yBase2, bottomArea, minIdx, maxIdx, False, finalHull,
            myData)

    return finalHull

```

Main Program

```

# MAIN PROGRAM

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import datasets
from myConvexHull import myConvexHull

# Memberi tahu pengguna dataset yang tersedia
print('Available Datasets: ')

availableDatasets = ['Irish', 'Breast Cancer', 'Wine']
maxRowDatasets = [3, 29, 12]

for i in range(len(availableDatasets)):
    print(i+1, '. ', availableDatasets[i])

# Meminta input nomor dataset kepada pengguna
# Asumsi input selalu valid
inputDataset = int(input('Select dataset (1-3): '))

# Meload dataset
if inputDataset == 1:
    data = datasets.load_iris()
elif inputDataset == 2:
    data = datasets.load_breast_cancer()
else:
    data = datasets.load_wine()

# Membuat dataframe
df = pd.DataFrame(data.data, columns=data.feature_names)
df['Target'] = pd.DataFrame(data.target)

# Meminta input berupa index kolom yang akan dijadikan axis dan ordinat
# Asumsi input selalu berada didalam range yang diberikan
x = int(input(f'Enter column index for axis (0-{maxRowDatasets[inputDataset-1]}): '))
y = int(input(f'Enter column index for ordinate (0-{maxRowDatasets[inputDataset-1]}): '))

```

```

# Visualisasi hasil Convex Hull
plt.figure(figsize = (10, 6))
colors = ['b','r','g']

xName = str.title(data.feature_names[x])
yName = str.title(data.feature_names[y])
title = xName + ' vs ' + yName
plt.title(title)

plt.xlabel(data.feature_names[x])
plt.ylabel(data.feature_names[y])

for i in range(len(data.target_names)):
    bucket = df[df['Target'] == i].iloc[:,[x,y]].values
    myData = df[df['Target'] == i].iloc[:,[x,y]].values
    # Menggunakan fungsi myConvexHull untuk menentukan titik-titik
    # yang membentuk Convex Hull dari dataset yang diberikan
    hull = myConvexHull(myData)
    plt.scatter(bucket[:, 0], bucket[:, 1], label=data.target_names[i])
    for simplex in hull:
        plt.plot(bucket[simplex, 0], bucket[simplex, 1], colors[i])

plt.legend()

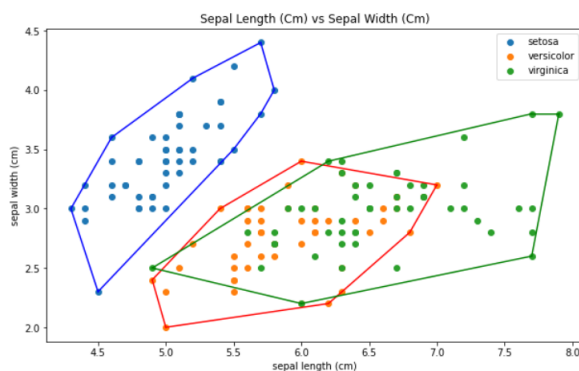
```

III. INPUT DAN OUTPUT PROGRAM

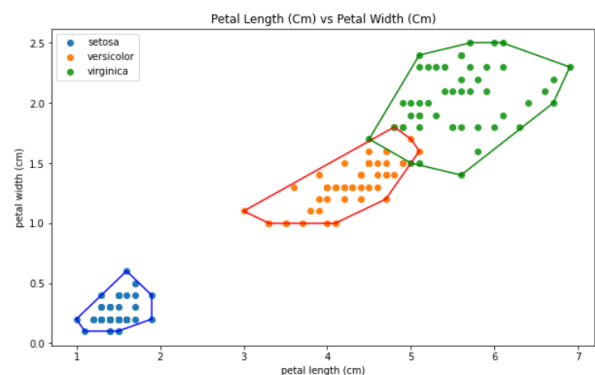
Input dan Output Program

Dataset Iris

Available Datasets:
 1 . Irish
 2 . Breast Cancer
 3 . Wine
 Select dataset (1-3): 1
 Enter column index for axis (0-3): 0
 Enter column index for ordinate (0-3): 1
 <matplotlib.legend.Legend at 0x239504f4250>



Available Datasets:
 1 . Irish
 2 . Breast Cancer
 3 . Wine
 Select dataset (1-3): 1
 Enter column index for axis (0-3): 2
 Enter column index for ordinate (0-3): 3
 <matplotlib.legend.Legend at 0x23951600100>



Dataset Breast Cancer

Available Datasets:

1 . Irish

2 . Breast Cancer

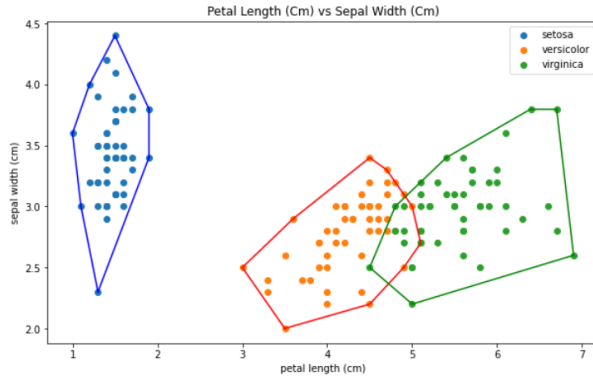
3 . Wine

Select dataset (1-3): 1

Enter column index for axis (0-3): 2

Enter column index for ordinate (0-3): 1

<matplotlib.legend.Legend at 0x239516aa040>



Available Datasets:

1 . Irish

2 . Breast Cancer

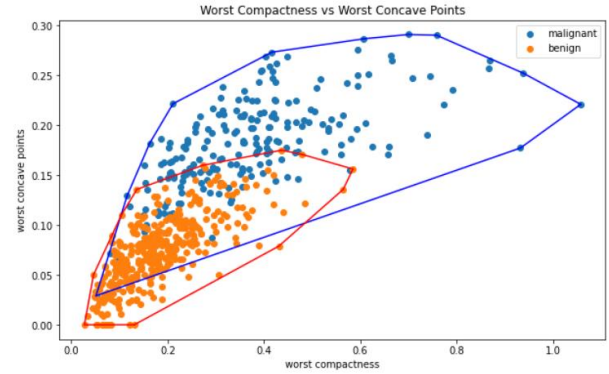
3 . Wine

Select dataset (1-3): 2

Enter column index for axis (0-29): 25

Enter column index for ordinate (0-29): 27

<matplotlib.legend.Legend at 0x23950038eb0>



Dataset Wine

Available Datasets:

1 . Irish

2 . Breast Cancer

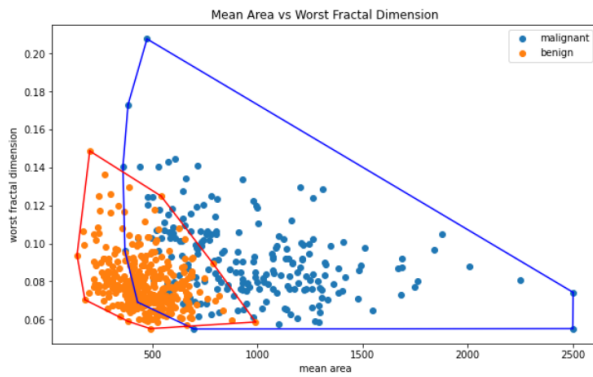
3 . Wine

Select dataset (1-3): 2

Enter column index for axis (0-29): 3

Enter column index for ordinate (0-29): 29

<matplotlib.legend.Legend at 0x2394b0a0550>



Available Datasets:

1 . Irish

2 . Breast Cancer

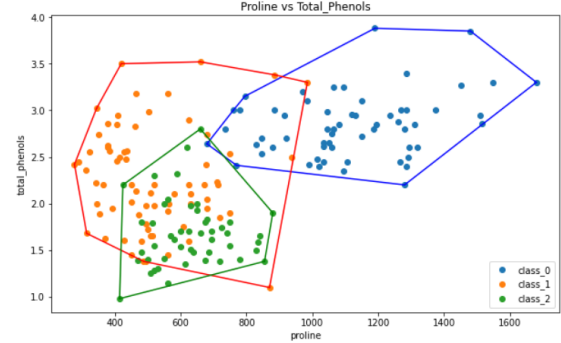
3 . Wine

Select dataset (1-3): 3

Enter column index for axis (0-12): 12

Enter column index for ordinate (0-12): 5

<matplotlib.legend.Legend at 0x23949923580>



IV. LINK CODE PROGRAM

<https://github.com/adellinekania/Tucil2-MyConvexHull>

Poin	Ya	Tidak
1. Pustaka myConvexHull berhasil dibuat dan tidak ada kesalahan	V	
2. Convex Hull yang dihasilka sudah benar	V	

3. Pustaka myConvexHull dapat digunakan untuk menambilkkan Convex Hull setiap label dengan warna yang berbeda	V	
4. Bonus: Program dapat menerima input dan menuliskan output untuk dataset lainnya	V	