Pattern Recognition (1)

Project 01

Iftekhar Ahmed
Hasan Mosharrof
Farhad Ali
Selin Sezer

Overview

Task 1.1
Selin Sezer

Task 1.2
Selin Sezer

Task 1.3
Iftekhar Ahmed

Task 1.4
Hasan Mosharrof

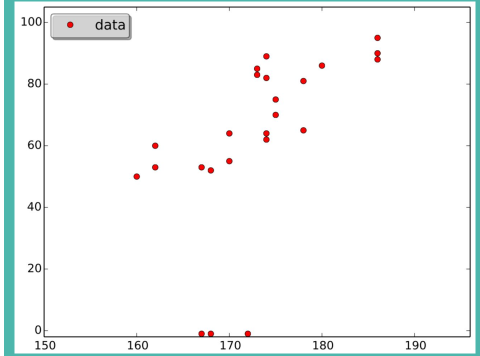
Task 1.5
Farhad Ali

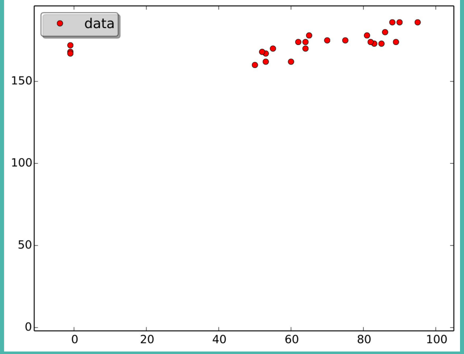
Task 1.1

acquaint yourself with python for pattern recognition

Problem:

* Eliminating missing data (outliers)





Task 1.1

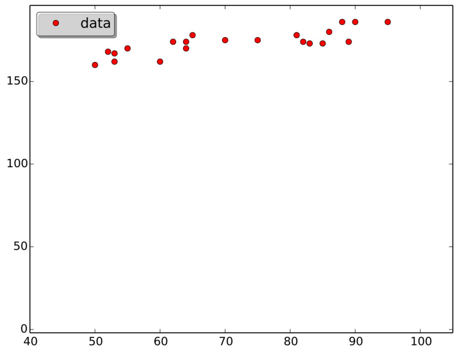
acquaint yourself with python for pattern recognition

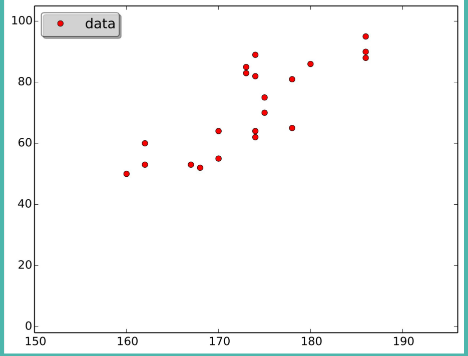
Solution:

Ideas:

* Post-processing the data after reading

**** Processing the data while reading***





Task 1.1

acquaint yourself with python for pattern recognition

Code:

```
def is_outlier(point):
    if point[0] < 0 or point[1] < 0:
        return True
    else:
        return False
```

```
ws = np.array([d[0] for d in data if not is_outlier(d)])
hs = np.array([d[1] for d in data if not is_outlier(d)])
gs = np.array([d[2] for d in data if not is_outlier(d)])
```

Overview

Task 1.1
Selin Sezer

Task 1.2
Selin Sezer

Task 1.3
Iftekhar Ahmed

Task 1.4
Hasan Mosharrof

Task 1.5
Farhad Ali

Task 1.2

fitting a Normal distribution to 1D data

Problem:

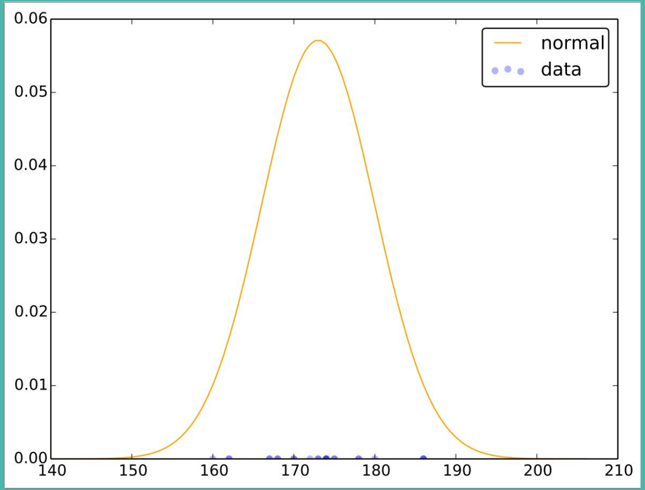
* Computing the mean and the standard deviation of a given data

* Plotting the normal distribution characterizing its density among the data

Task 1.2

fitting a Normal distribution to 1D data

Solution/Code:




```python
# calculate the mean and the standard deviation
mean, standard_deviation = norm.fit(heights)

# create a range where the probability density will be calculated
x = np.linspace(min(heights)-20, max(heights)+20, 100)

# create the probability density function using the parameters we obtained above
fitted_pdf = norm.pdf(x, loc=mean, scale=standard_deviation)
```

Overview

Task 1.1
Selin Sezer

Task 1.2
Selin Sezer

Task 1.3
Iftekhar Ahmed

Task 1.4
Hasan Mosharrof

Task 1.5
Farhad Ali

Weibull Distribution fitting

task 1.3:

Weibull Distribution

$$f(x \mid \kappa, \alpha) = \frac{\kappa}{\alpha} \left(\frac{x}{\alpha}\right)^{\kappa-1} e^{-\left(\frac{x}{\alpha}\right)^{\kappa}}$$

K = shape, α = scale

We estimate the maximum likelihood of the params to fit the distribution to the histogram of dataset

task 1.3:

Given a data sample D = $\{di\}_{i=1}^{N}$, the log-likelihood for the parameters of the Weibull distribution is

$$L(\alpha, \kappa \mid D) = N\left(\log \kappa - \kappa \log \alpha\right) + (\kappa - 1) \sum_i \log d_i - \sum_i (d_i/\alpha)^\kappa.$$

Deriving L with respect to α and κ leads to a coupled system of partial differential equations for which there is no closed form solution.

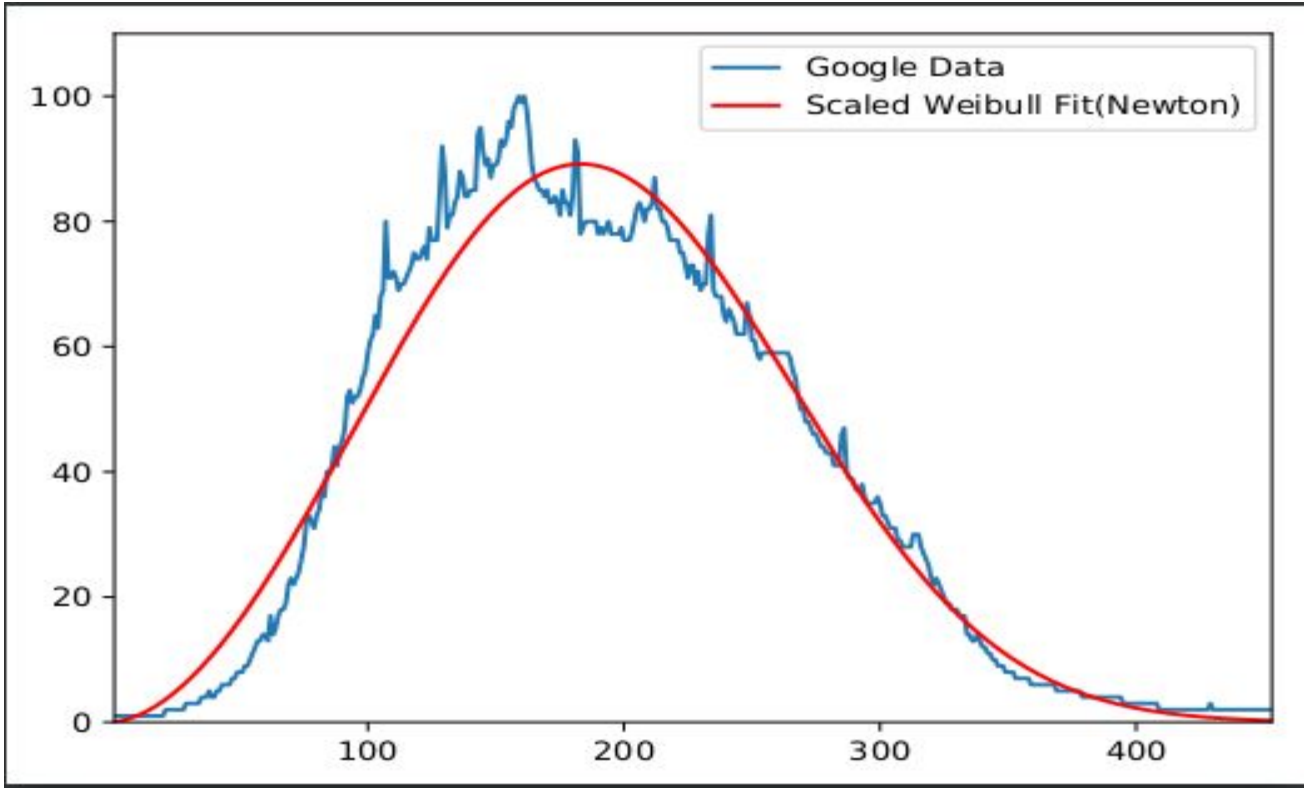
task 1.3:

Therefore, resort to Newton’s method for simultaneous equations and compute

```
def newtonParametersCalculator(k, a, histogramData):
    # We Input parameters 'k' and 'a' (alpha) into the function.
    N = len(histogramData)
    sum_log_di = np.sum(np.log(histogramData))
    sum_di_a_k = np.sum((histogramData / a) ** k)

    # Calculate all the matrix elements of the Newtonian Method.
    dl_dk = N / k - N * math.log(a) + sum_log_di - np.sum(
        ((histogramData / a) ** k) * np.log(histogramData / a))
    dl_da = (k / a) * (sum_di_a_k - N)
    d2l_dk = -N / (k ** 2) - np.sum(((histogramData / a) ** k) *
                                   (np.log(histogramData / a)) ** 2)
    d2l_da = (k / ((a) ** 2)) * (N - (k + 1) * sum_di_a_k)
    d2l_dkda = M21 = (1 / a) * sum_di_a_k + (k / a) * np.sum(
        ((histogramData / a) ** k) * np.log(histogramData / a)) - N / a
    return np.array(np.matmul(np.linalg.inv(np.matrix([[d2l_dk, d2l_dkda],
                                                      [d2l_dkda, d2l_da]])), np.array([-dl_dk, -dl_da])) +
                    np.array([k, a]))[0]
```

task 1.3:

Output



Overview

Task 1.1
Selin Sezer

Task 1.2
Selin Sezer

Task 1.3
Iftekhar Ahmed

Task 1.4
Hasan Mosharrof

Task 1.5
Farhad Ali

Drawing unit circles

Task 1.4

The Lp norm of *x* is given by:

Here:

$$||\boldsymbol{x}||_p = \left(\sum_{i=1}^{m} |x_i|^p\right)^{\frac{1}{p}}$$

$\boldsymbol{x} \in \mathbb{R}^m$ and $p \in \mathbb{R}, p \geqslant 1$

considering the space $\mathbb{R}^2$, what is a *unit circle* ?

answer

the set

$$C = \left\{ \boldsymbol{x} \in \mathbb{R}^2 \;\middle|\; d\left(\boldsymbol{x}, \boldsymbol{0}\right) = 1 \right\}$$

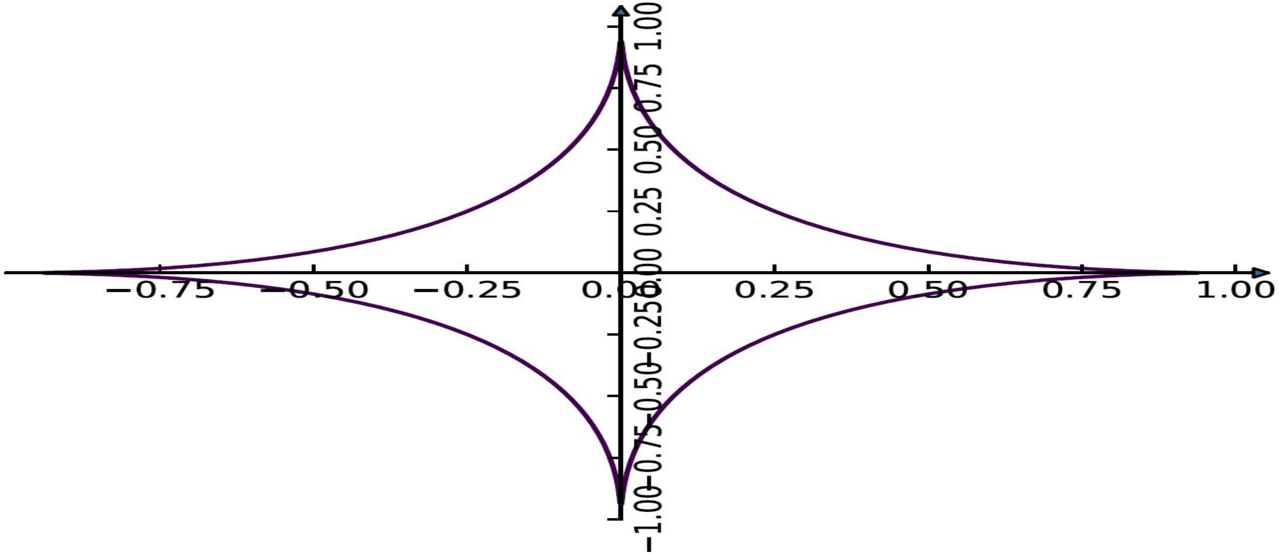
Plotting 2D vector with p-norm

```
def plotUnitCircle(p):

    """ plot some 2D vectors with p-norm < 1 """
    fig = plt.figure(1)
    ax = SubplotZero(fig, 111)
    fig.add_subplot(ax)
    #print(fig)
    #print(ax)
    for direction in ["xzero", "yzero"]:
        ax.axis[direction].set_axisline_style("-|>")
        ax.axis[direction].set_visible(True)

    for direction in ["left", "right", "bottom", "top"]:
            ax.axis[direction].set_visible(False)

    x = np.linspace(-1.0, 1.0, 1000)
    y = np.linspace(-1.0, 1.0, 1000)
    X, Y = np.meshgrid(x, y)
    F = (((abs(X) ** p + abs(Y) ** p) ** (1.0 / p))-1)
    ax.contour(X, Y, F, [0])
```

Output

Consider the Lp norm for p = 1/2 and plot the corresponding R^2 unit circle.Is this really a norm or not?

P = ½ is not a norm cause it violates the triangle inequality rules.



Overview

Task 1.1
Selin Sezer

Task 1.2
Selin Sezer

Task 1.3
Iftekhar Ahmed

Task 1.4
Hasan Mosharrof

Task 1.5
Farhad Ali

Estimating the dimension of fractal objects in an image

Step1: Convert the image to binary image

Step2: *Produce box scaling sets S and compute corresponding box count, S obviously from $1/2^i$ where i={1 to L-2}, while L =$\log_2$(height/width of image). For every level we are showing the box in separate image output.*

Step3: *Use linear least square, estimate it's slope D and fit a line.Line plot is saved as "Task_1_5_"+image_name+".pdf" into current directory.*

Q&A