ILLINOIS INSTITUTE OF TECHNOLOGY

ECE-505: APPLIED OPTIMIZATION FOR ENGINEERS

PROJECT

FALL 2022

CHIBUZO VALENTINE NWADIKE

A20480163

Tools Used:

- Python 3.11.0
- Google Collab

Attached files:

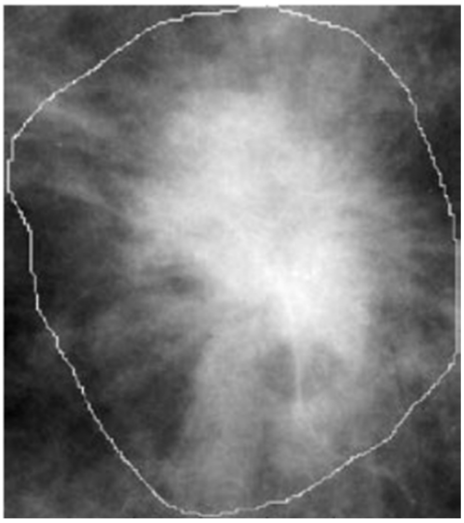
Python script

Dataset: image_samples.txt

GitHub link to the project: https:/github.com/LeoVal1/ECE_505_Project-Optimization-.git

PROBLEM DESCRIPTION

Assume the intensity value (i.e., brightness) of a pixel in a cancerous region in a mammogram image can be modeled by a Gaussian $N(\mu_1, \sigma_1^2)$, where μ_1, σ_1^2 are the mean and variance of the intensity, respectively, while that in a harmless background is modeled by a Gaussian $N(\mu_1, \sigma_1^2)$. A uniform random sampling of mammogram image (shown below) yields 200 intensity values from the image (listed in a separate spreadsheet). Based on this information, estimate the proportion of cancerous pixels in the image.



PROBLEM FORMULATION:

In this problem I employed the application of the optimization algorithm to identify regions of cancerous and non-cancerous pixels by estimating mean and variance for each region. Let P_1 denote the proportion of cancerous pixels in the image. Then a randomly chosen pixel from the image has the following distribution:

$$p(x;\mu_1,\sigma_1^2,\mu_2,\sigma_2^2,P) = P_1 \cdot \frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}} + (1-P_1) \cdot \frac{1}{\sqrt{2\pi\sigma_2^2}} e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}}$$ ------------------------------------1.0

The problem then is to estimate the unknown $\Theta = \left(\mu_1, \sigma_1^2, \mu_2, \sigma_2^2, P_1\right)$ parameters in equation 1.0 from the given image samples.

Then I consider finding the maximum likelihood estimate (MLE)

$$\left(\mu_1,\sigma_1^2,\mu_2,\sigma_2^2,P_1\right) = \arg\max_{\Theta} \left(\prod_{i=1}^{200} p\left(x_i;\mu_1,\sigma_1^2,\mu_2,\sigma_2^2,P\right) \right),$$ --2.0

Where $x_i, i = 1, 2, \cdots, 200,$ denote the image samples.

PROBLEM SOLUTION

Expectation Maximization Algorithm

EM Algorithm (Pages 438 and 439 has been used for reference) is part of maximum likelihood estimation (MLE) used when the marginal probabilities of data are hard to be optimized.

Since the log likelihood of equation 1.0 does not converge, a new random variable Z is introduced such that Z = 0, 1 and by using joint distribution and posterior probability a new algorithm called EM algorithm is introduced which reduces the optimization to maximizing Q or minimizing Q of equation 3.0. The expectation step of EM algorithm includes calculating equation 4.0 using the parameters calculated in Maximization/Minimization step.

$$Q(\Theta,\Theta_{old}) = \Sigma_{i=1}^{200}\Sigma_{k=1}^{2}((ln(P_k)+\frac{1}{2}ln(2\pi\sigma_k^2)-\frac{(x_i-\mu_k)^2}{2\sigma_k^2})(p_{ik}(z/x_i;\Theta_{old})) \qquad \text{----------------------3.0}$$

$$p_{ik}(z/x_i;\Theta_{old}) = \frac{P_k * GaussPDF(x_{ik};\Theta_{old})}{\Sigma_{i=1}^{200}\Sigma_{k=1}^{2}P_k * GaussPDF(x_{ik;\Theta_{old}})} \qquad \text{---4.0}$$

Assumption: Cancerous and Non-Cancerous pixels are independent i.e., p is zero.

step1: Initialize parameters with seed value.

step2: Find Expectation P(Z=z/x;θ).

step3: Maximize Q with respect to parameters or Minimize Q w.r.t parameters

step4: check for convergence of parameters of log likelihood function if yes exit else continue

The code fragment below shows the EM algorithm. Here the image dataset (image_sample.txt) is loaded, and the SGD optimization method is applied. Before the EM algorithm, I defined different function that would used in computing the SGD method in the EM algorithm.

```
def E_M_Algorithm(file_name,seed_theta):
    x=np.loadtxt(file_name,dtype=int,delimiter='\n')
    optimize=unconstrained_optimizer()
    seed_theta=np.array(seed_theta)
    inital_p_x_given_theta=marginal_log_likelihood_funtion_x(x,seed_theta)

    if len(seed_theta)!=5:
        sys.exit("E_M can handle only vector of size 5")

    for i in range(100):
        # Expectation step
        gamma=Expection_function(x,seed_theta)

        # Maximization Step/ minimization step


        [new_theta,graval,alval,crit]=optimize.Steepest_Gradient_Descent_Method(Q_function,grad_Q_function,
                                                                                Newton_line_search,linearfd_Q_func_alpha,
                                                                                Armijo_Condition,
                                                                                 1e-5,1000,seed_theta,linearsd_Q_func_alpha,
                                                                                 1e-4,1e-3,x,gamma)


        new_p_x_given_theta=marginal_log_likelihood_funtion_x(x,new_theta[-1])
        seed_theta=new_theta[-1]
        c=np.sqrt((new_p_x_given_theta-inital_p_x_given_theta)**2)
        if(c<=1e-5):
            print("Convergence of Marginal probability has happened")
            return seed_theta
        else:
            inital_p_x_given_theta=new_p_x_given_theta

    print("Convergence of Marginal probability has not happened")
    return seed_theta
```

Output

```
Mean1:
120.26131051825456
Var1:
196.28040276252787
Mean2:
201.6590417843259
Var2:
218.19920850398717
P:
0.6165765961562276
```