UNIVERSIDADE ESTADUAL DE CAMPINAS

Prova de Ea616

Aluno: Bruno Eduardo Freire e Silva
RA:195052
b195052@dac.unicamp.com

Campinas, SP
2020

1-Introdução:

Esse trabalho tem por objetivo tentar explicar a atual propagação do coronavírus através de um modelamento epidêmico do tipo SIR.

Neste modelo observamos a interação entre o número de infectados i(t), o número de pessoas saudáveis e o número de pessoas recuperadas / mortas r(t) durante um determinado periodo. As equações que regem tal interação são dadas por:

$$\frac{dI}{dt} = \frac{\gamma IS}{M} - \alpha I$$

$$\frac{dS}{dt} = -\frac{\gamma IS}{M} \qquad \text{eqs 1,2, 3 e 4}$$

$$\frac{dR}{dt} = \alpha I$$

R + I + S = M

Onde I é o número de infectados, S é o número de pessoas saudáveis, R é o número de pessoas recuperadas/mortas , γ é a taxa de transmissão (quantas pessoas, em média, um infectado infecta), α é taxa de recuperação e , por fim, M que é a população total (no caso do Brasil, 210 milhões de pessoas).

Dessas equações também podemos calcular o número acumulado de infectados a(t), que é dado pela seguinte equação:

$$\frac{dA}{dt} = \frac{\gamma IS}{M} \qquad \text{eq 5}$$

Como o objetivo deste trabalho é explicar o atual cenário de infecção, temos portanto que achar os parâmetros (I inicial, S inicial, A inicial, γ e α) que quando postos sobre tais equações, nos devolvem um curva parecida com a curva de infecção real. Para isso vamos escrever um código que as calcula computacionalmente.

É mais facil trabalhar com o modelamento SIR discretizado, por isso vamos aplicar a transformação de Euler as equações 1,2 e 4 e teremos as seguintes equações:

$$s(k+1) = s(k) - \frac{\gamma i(k)s(k)}{M}$$

$$i(k+1) = (1-\alpha)i(k) + \frac{\gamma i(k)s(k)}{M} \qquad \text{eq 6, 7, 8}$$

$$a(k+1) = a(k) + \frac{\gamma i(k)s(k)}{M}$$

Tendo as equações acima, os dados reais de a(k) e um bom algoritmo de otimização, podemos achar os parâmetros que mais explicam o atual cenario de infecção.

2-Objetivo:

O objetivo desse trabalho é escrever um código python que faça o processo de otimização de hiperparâmetros e com esses hiperparâmetros poder discutir o atual cenário de infecção no Brasil. Alem de claro, poder discutir a qualidade do modelamento.

3-Algoritmo de otimização:

Pretendemos achar os parametros otimos utilizando a função gp_minimize da biblioteca de otimização scikit-optimize, essa biblioteca utiliza-se do processo de otimização com base no erro mínimo quadrático, ou seja, ela tenta achar os valores de alpha, gama, i0, s0 e a0 para os quais minimizam a somatória das diferenças entre o y observado e o y calculado elevados ao quadrado:

$$f(\alpha, \gamma) \;\; t.q \; min \sum_{t=0}^{t=n} [f(\alpha, \gamma) - fobs]^2$$

O algoritmo será:

```
import pandas as pd
import numpy as np
from sklearn.metrics import mean_absolute_error,
mean_squared_error
from skopt import gp_minimize
def sir_model(length, a0, s0, i0, gama, alpha):
 M=210147125
 ak = a0
 sk = s0
 ik = i0
 acumulate = [a0]
 for i in range(1,length):
   skplus1 = sk - (gama * (sk / M) * ik)
   ikplus1 = ik + (gama * (sk / M) * ik) - (alpha * ik)
   akplus1 = ak + (gama * (sk / M) * ik)
   ak = akplus1
   sk = skplus1
   ik = ikplus1
   acumulate.append(ak)

 return np.array(acumulate)
# load data
filepath = '/content/drive/My
Drive/projeto_ea_616/HIST_PAINEL_COVIDBR_24jun2020.xlsx'
data = pd.read_excel(filepath)
# preprocessing data
```

```python
ydata = data[data.regiao == 'Brasil'].casosAcumulado.to_numpy()[1:]

global ydata
def func_to_minimize(params):
 ycalc = sir_model(120, params[0], params[1],
                    params[2], params[3], params[4])
 MAE = mean_absolute_error(ydata, ycalc)
 return MAE ** (1/2)
space = [(400, 495), (150000000, 1835793857), (400, 495), (0.30, 0.32), (0.20, 0.24)]
y = gp_minimize(func_to_minimize, space, n_calls=100, random_state=1234, n_random_starts=20, verbose=True)
```

obs : A variável y contém um método x que retorna os parâmetros ótimos

3.2- Como o algoritmo gp_minimize funciona:

Este algoritmo funciona com base na busca bayesiana de hiperparâmetros. De início criamos vários conjuntos randômicos de parâmetros e para cada conjunto calculamos o erro médio absoluto , com base nesses resultados fazemos um processo gaussiano sobre essas métricas, i.e procuramos por regiões onde mais provavelmente acharemos os melhores parâmetros.

Obs: O erro médio absoluto é gerado pela função python func_to_minimize, que calcula essa métrica usando a função sir_model, ou seja, estamos otimizando os parâmetros do modelo SIR.

3.3- Como foi feito o processo de achar hiperparâmetros:

Como o modelo SIR é extremamente não linear, o processo de busca de hiperparâmetros não pode ser feito de uma vez só, pelo menos com essa metodologia de busca. Então, para conseguir melhores resultados devemos repetir diversas vezes o processo de otimização, e em cada vez, mudar o intervalo em que vamos buscar cada hiperparâmetro.

No meu caso, eu separei a busca em 2 fases, a primeira fase eu tentei achar o melhor intervalo para os parâmetros gama e alfa, enquanto os parâmetros i0, s0 e a0 eu fixei em 1, M-1 e 1. Encontrado os melhores intervalos de gama e alfa, irei procurar os melhores intervalos para i0, s0 e a0.

4- Resultados e discussões:

Nas primeiras iterações percebi que o modelo se comporta muito mal quando definimos um intervalo muito grande para os parâmetros, veja:

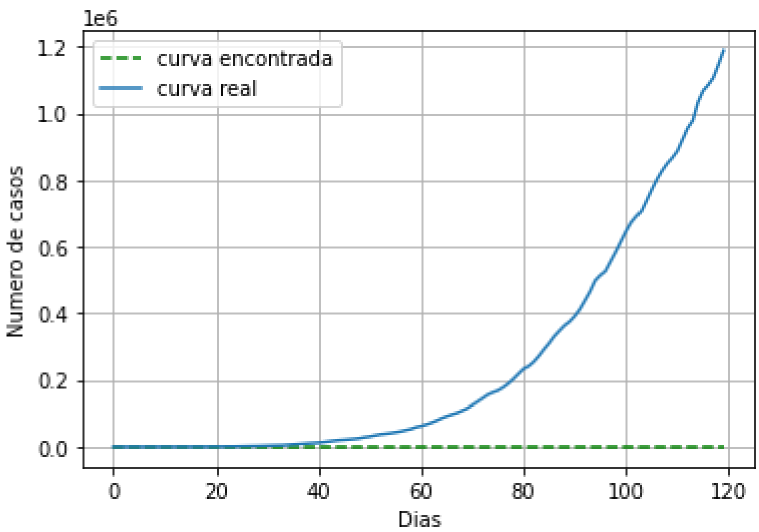

Figura 1: Parâmetros ótimos: a0 = `1944, s0 = 2537285, i0 = 105, gama= 0, alpha= 1 : Metricas: Mean_absolute_error = 242064`

Percebemos que a curva calculada não acompanha em nada a curva real. Vendo isso resolvi procurar por intervalos mais restritos, depois de várias variações, percebi que os melhores valores de de alfa e gama estavam em `(0.30, 0.32) para gama e (0.20, 0.24)` para alfa:

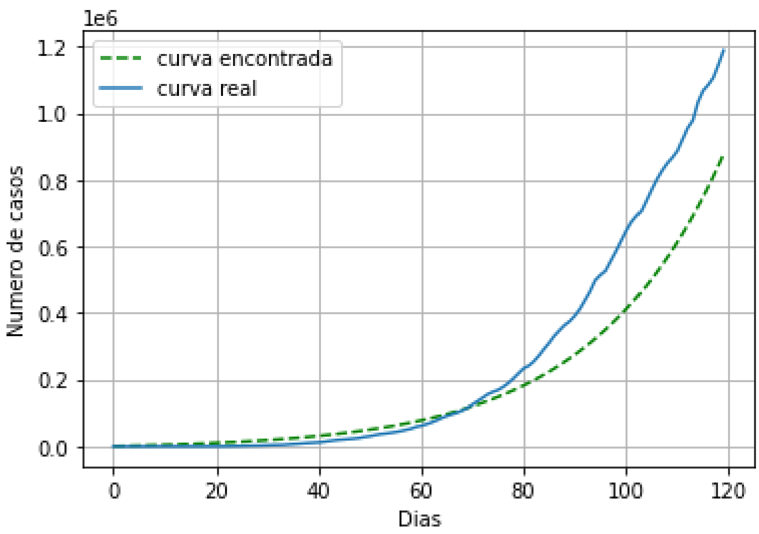

Figura 2: Parâmetros ótimos: a0 = `996, s0 = 171866397, i0 = 1225, gama= 0.3154, alpha= 0.21721339542755375 ; Metricas`: `Mean_absolute_error = 76508`

Agora o trabalho era encontrar os parâmetros i0, s0 e a0. Depois de vários testes chegamos aos valores ótimos de i0=695, s0=183793857, a0=697, gama=0.3092 e alfa=0.2187 e para esse conjunto de parâmetros achamos um erro absoluto médio de 29929, aproximadamente:

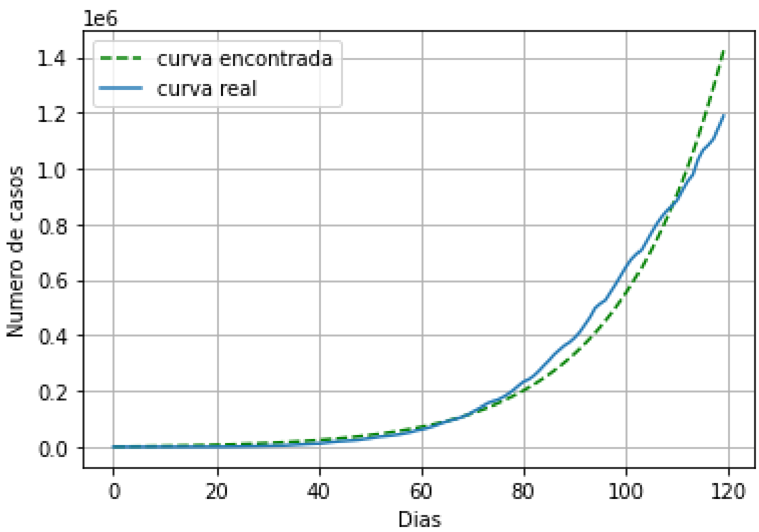

Figura 3: Parâmetros ótimos: a0 = 695, s0 = 183793857, i0 = 697, gama= 0.3092, alpha= 0.2188; Metricas: Mean_absolute_error = 29929.

Como vemos no gráfico acima, conseguimos aproximar bem a curva real da curva calculada e com isso tirar algumas conclusões do comportamento desta. Podemos, por exemplo, discutir o significado dos parâmetros alfa e gama, uma taxa de recuperação de 0.2187 nos indica que o período de infecção ϴ ($\Theta = \frac{1}{\alpha}$) é de aproximadamente 4.57 dias (na literatura encontramos que tal valor vale 5.2 [1]). Já uma taxa de transmissão de 0.3092 nos indica que ,em média, uma pessoa infectada transmite o vírus para 1.41 pessoas (Npessoas = $\frac{\gamma}{\alpha}$) (na literatura vemos que esse valor é aproximadamente 3.2 para o Brasil)

Com relação a predição do comportamento futuro, não posso afirmar nada numericamente, a única certeza é que em um curto prazo de tempo a curva de infectados ficará maior, numa tendência quase exponencial. Essa tendência é fruto de uma série de fatores comportamentais como a negação da gravidade da doença, não testagem em massa da população, medidas de isolamento pouco efetivas...Todos esses fatores influenciam no comportamento da curva e isso o modelo SIR não consegue explicar adequadamente.

5- Bibliografia:

[1] DRA .DANIELA BUSKE. A evolução do epidêmica do covid-19 ? modelo sir. , 2020. Disponivel em: <http://https://wp.ufpel.edu.br/fentransporte/2020/04/09/a-evolucao-epidemica-do-covid-19-modelo-sir/>. Acesso em: 26 jun. 2020.