

# Primeiros Resultados Estado de SP

O presente documento apresenta os primeiros resultados gerados ao aplicar as novas implementações aos dados do estado de SP. Para todos os testes, os dados sendo utilizados são de casos confirmados de COVID-19.

## 1 - Detecção de novas ondas

### 1.1 - Sem alterações ( $threshold = 3e-5$ )

A Figura 1 apresenta os resultados do algoritmo de detecção de novas ondas para Jerusalém (como base para comparação) e três cidades do estado de São Paulo.

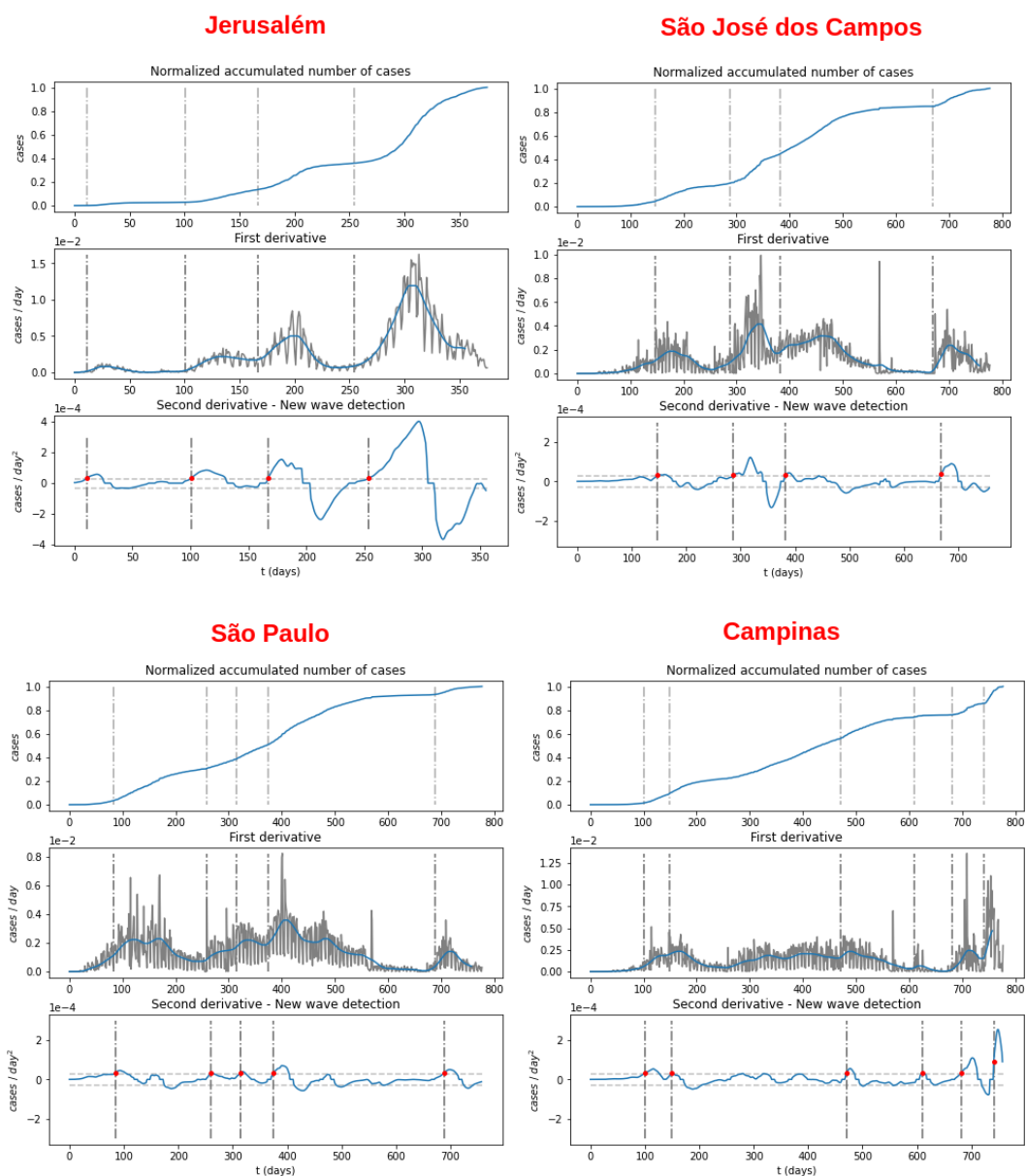


Figura 1 - Comparação do algoritmo de detecção de novas ondas

### 1.2 - Testando um $threshold$ menor para Campinas

Para a cidade de Campinas, foi testado um *threshold* menor, de forma a detectar ondas menos evidentes. A Figura 2 apresenta o resultado desse teste.

Pode-se observar que a versão com menor *threshold* foi capaz de detectar 3 novas transições entre os dias 200 e 400. Assim, o algoritmo sugere a utilização de um modelo mais complexo, contendo 10 sigmóides ao invés de 7.

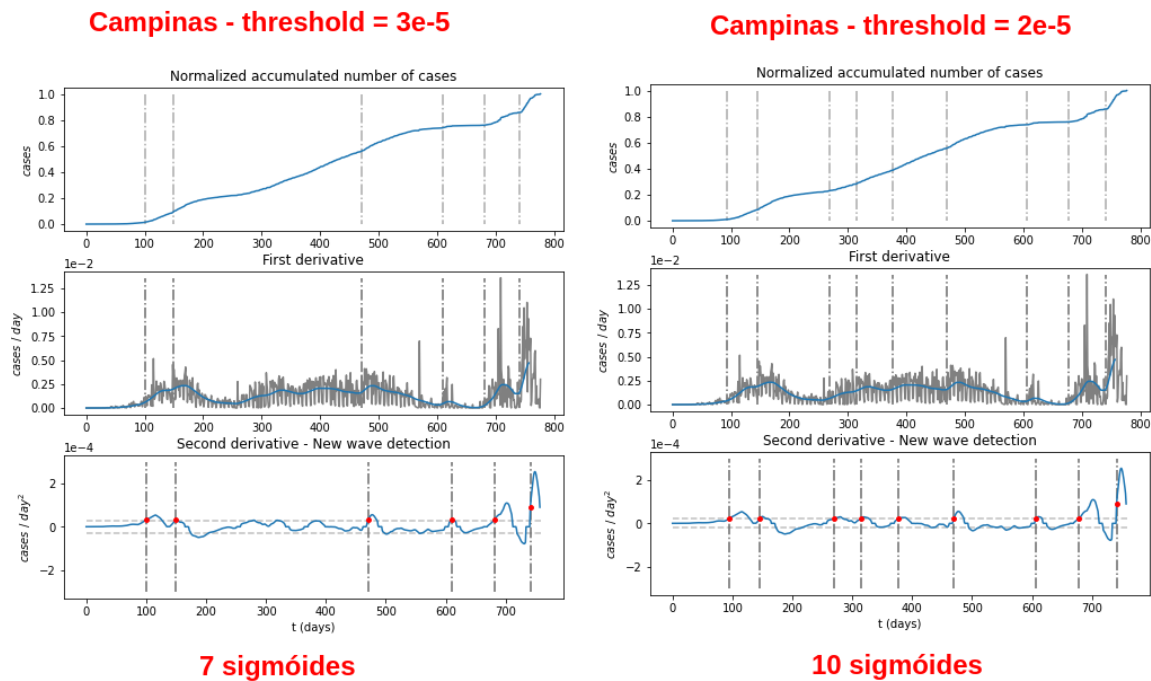


Figura 2 - Comparação de diferentes *thresholds* para a detecção de novas ondas nos dados de Campinas

## 2 - Ajuste do modelo

### 2.1 - Sem alterações

A Figura 3 apresenta a aplicação do algoritmo implementado aos dados de São José dos Campos, sem alterações na lógica. A única mudança feita foi no vetor  $x_{nw}$ , que contém os pontos de transição entre as ondas. Nota-se que o ajuste é bom para as ondas epidemiológicas, mas ruim para os casos com mais de três.

São José dos Campos

$x_{\text{new\_wave}} = [147, 287, 382, 669]$

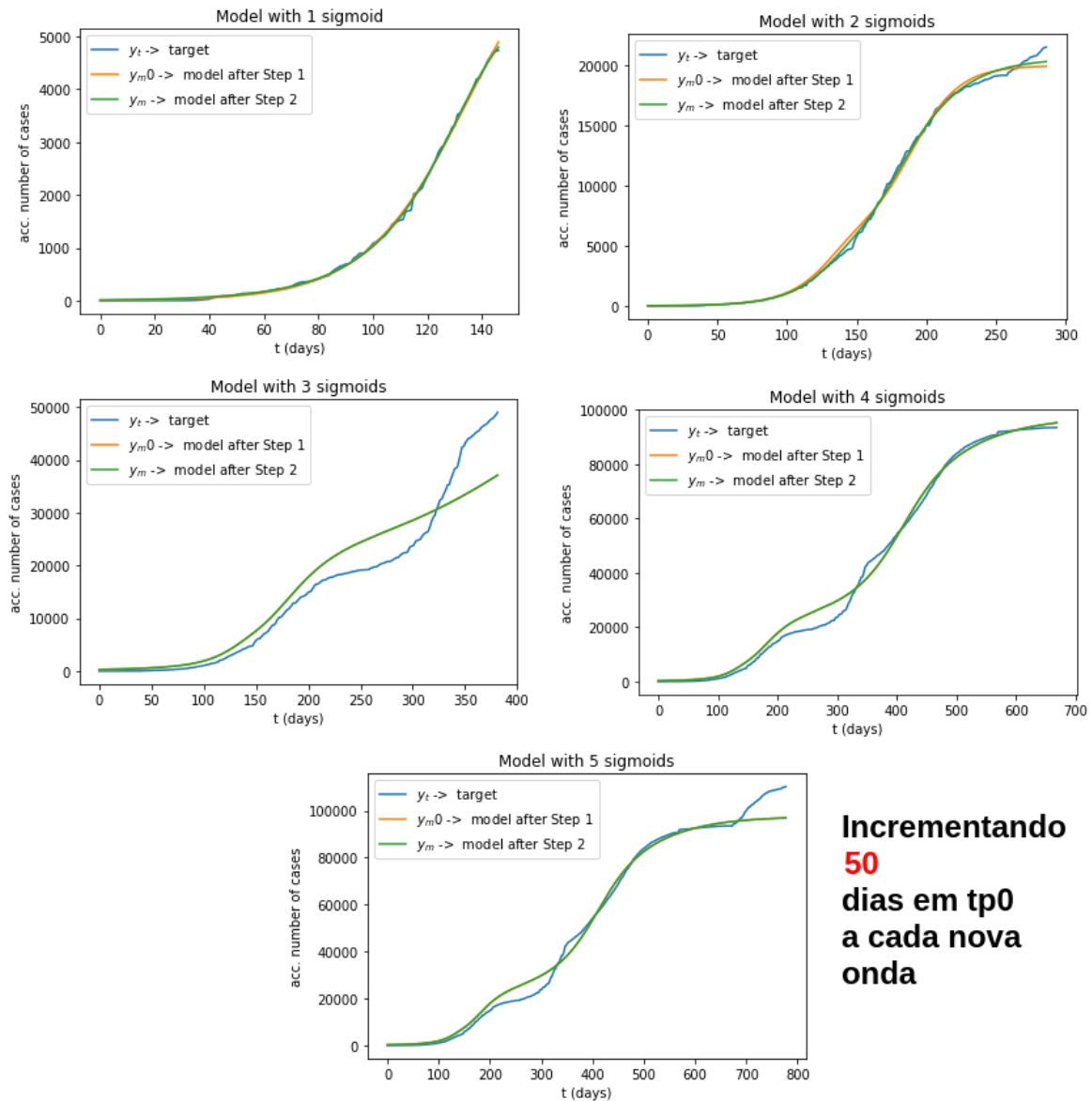


Figura 3 - Ajuste do modelo aos dados de São José dos Campos, sem alterações no código

## 2.2 - Mudando incremento de 100 dias em $t_{p0}$ a cada nova onda

Tendo em vista os resultados da Figura 3, foi feito um teste aumentando de 50 para 100 o incremento no valor inicial do parâmetro  $t_{p0}$  a cada nova sigmoide adicionada ao modelo. A mudança se justifica pelo fato dos dados de Jerusalém irem até o dia 376 e os dados das cidades paulistas, até 778. O resultados desse teste estão apresentados na Figura 4.

Uma melhoria na implementação poderia fazer com que o algoritmo escolha esse incremento automaticamente com base no número total de dados disponíveis.

São José dos Campos

$x_{\text{new\_wave}} = [147, 287, 382, 669]$

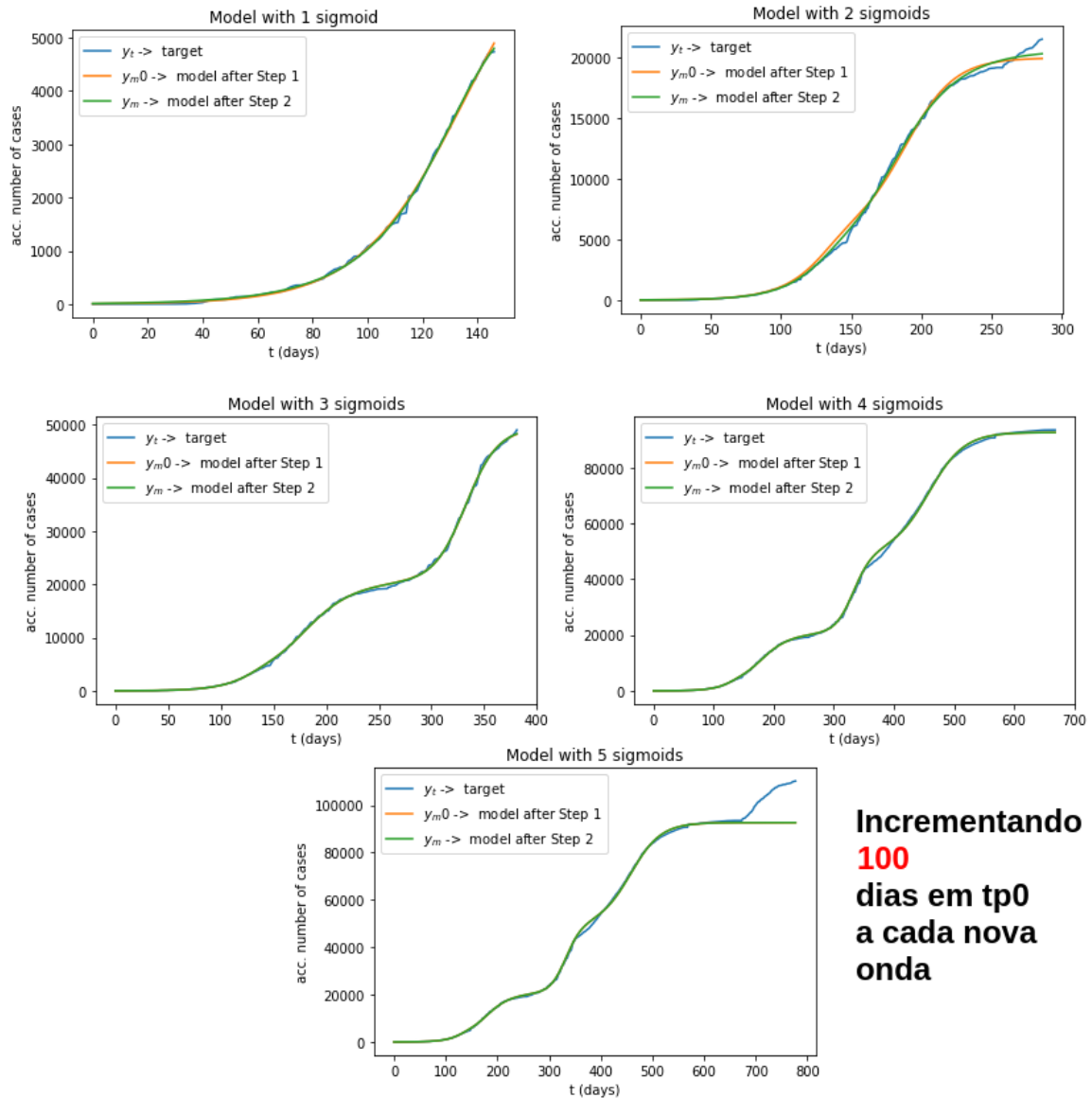


Figura 4 - Ajuste do modelo aos dados de São José dos Campos, aumentando o incremento em  $tp_0$  a cada nova onda

### 3 - Ajuste por cidade (sem alterações)

A Figura 5 apresenta o ajuste do modelo para as cidades de São José dos Campos, São Paulo e Campinas, apenas ajustando os pontos de transição ( $x_{nw}$ ) e utilizando um incremento de 100 dias para  $t_{p0}$  a cada nova onda.

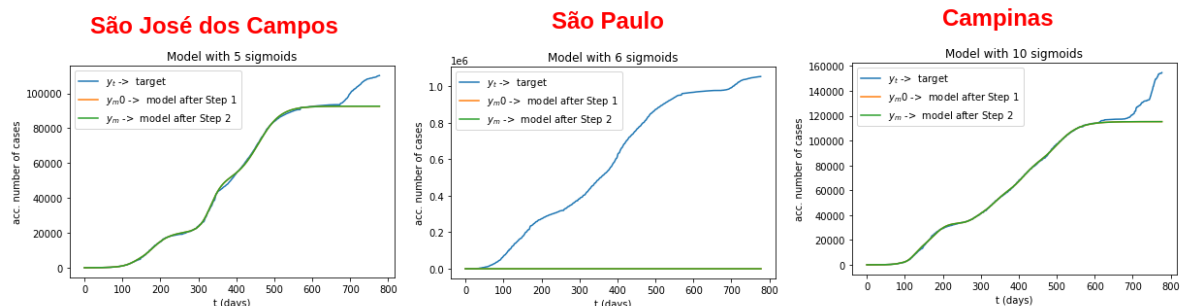


Figura 5 - Ajuste do modelo aos dados das cidades de São José dos Campos, São Paulo e Campinas

Pode-se notar que o ajuste para as cidades de São José dos Campos e Campinas foi bom, com exceção das ondas mais tardias. Já para a cidade de São Paulo, o modelo não conseguiu ser ajustado. Isso pode ser explicado pela grande diferença na magnitude dos dados para a capital, que possui muito mais casos, como mostra a comparação da Figura 6.

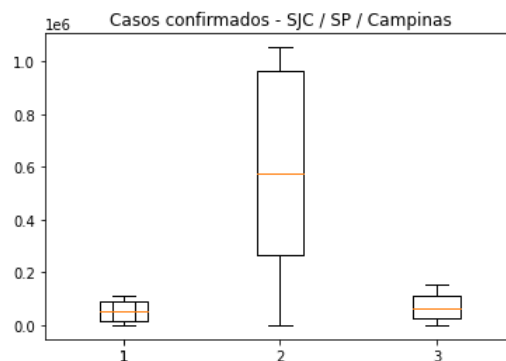


Figura 6 - Boxplot dos dados de casos acumulados de COVID-19 nas cidades de São José dos Campos (1), São Paulo (2) e Campinas (3)

## 4 - Comentários Finais

Com base nos resultados, pode-se notar que a implementação do ajuste do modelo para múltiplas ondas está funcionando bem, sendo capaz de descrever os dados do estado de São Paulo, mesmo sem um processo de ajuste manual dedicado para cada cidade. Se esse processo for realizado, é provável que ajustes ainda mais precisos possam ser obtidos.

Para melhorar a adaptação do modelo a regiões com magnitudes muito diferentes, alterações na lógica de determinação do ponto inicial de cada sigmoide podem vir a ser necessárias. Essa lógica está apresentada na Figura 7 e também pode ser observada no [notebook](#) que contém as implementações.

```
# Step 1 - Optimize a symmetric sigmoid (nu = 1)
# Initial values
if(i == 0):
    y_t = acc_n_cases[:n_days]
    A0 = 2*max(y_t)
    tp0 = (2/3)*len(y_t)
    delta0 = (1/4)*len(y_t)
    nu0 = 1
else:
    tp0 += 100
    A0 *= 1.2
```

Figura 7 - Lógica para determinação do ponto inicial para as sigmoides do modelo