Boston

May 17, 2022

[1]:

```
import warnings
warnings.filterwarnings('ignore')
```

[2]:

```
import scrapbook as sb
import pandas as pd
import numpy as np
import seaborn as sns
import numpy as np
from statistics import mean, median
import matplotlib.pyplot as plt
```

1 Baseline

[3]:

```
books = sb.read_notebooks("./BaseLine_Model_Output")
baseLine_data = []
for nb in books.notebooks:
    nbList=[nb.scraps['Stats Model MAE'].data,nb.scraps['Catboost MAE'].data]
    baseLine_data.append(nbList)
df = pd.DataFrame(baseLine_data, columns = ["Stats Model","Catboost"])
baseline_data = np.array(baseLine_data)
stats = median(baseline_data[:,0])
catboost = median(baseline_data[:,1])
display(df)
print(df.median(axis=0))
```

```
   Stats Model  Catboost
0     0.298850  0.192764
1     0.305628  0.201693
2     0.435760  0.270210
3     0.380010  0.262905
4     0.412824  0.267935
5     0.356926  0.213024
6     0.384976  0.209407
7     0.389513  0.199208
8     0.384688  0.238885
9     0.390258  0.247499
```

```
Stats Model    0.384832
```

```
Catboost       0.225954
dtype: float64
```

2 GAN Model

[4]:

```
book = sb.read_notebooks("./GAN_Output")
gan_data = []
gan_mse = []
for nb in book.notebooks:
    metrics = nb.scraps['GAN_1 Metrics'].data
    for i in range(1000):
        gan_mse.append(metrics[0][i])
    nbList = [nb.scraps['GAN Model MSE'].data,
              nb.scraps['GAN Model MAE'].data,
              nb.scraps['GAN Model Euclidean distance'].data,
              nb.scraps['GAN Model Manhattan Distance'].data]
    gan_data.append(nbList)

df = pd.DataFrame(gan_data, columns = ['MSE','MAE','Euclidean␣
 ↪Distance','Manhattan Distance'])
display(df.style)
print("MEDIAN:")
print(df.median(axis = 0))
gan_data = np.array(gan_data)
gan_median = median(gan_data[:,1])
```

```
<pandas.io.formats.style.Styler at 0x7f7a0810b5e0>
```

```
MEDIAN:
MSE                     0.141095
MAE                     0.258360
Euclidean Distance      3.793474
Manhattan Distance     26.352682
dtype: float64
```

3 ABC_GAN Analysis

3.1 ABC Pre-generator - Catboost

[5]:

```
book = sb.read_notebooks("./ABC_GAN_Catboost")
paramVal = [1,0.1,0.01]
abc_mae = [[] for i in range(3)]
abc_mae_skip = [[] for i in range(3)]
abc_mae_mean = [[] for i in range(3)]
abc_mae_skip_mean = [[] for i in range(3)]
abc_weights = [[] for i in range(3)]
prior_model = [[] for i in range(3)]
```

```
abc_pre_generator = [[] for i in range(3)]

for nb in book.notebooks:
    metrics1 = np.array(nb.scraps['ABC_GAN_1 Metrics'].data)
    metrics3 = np.array(nb.scraps['ABC_GAN_3 Metrics'].data)
    paramVar = float(nb.papermill_dataframe.iloc[0]['value'])

    #Divide data according to parameters
    for i in range(3):
        if paramVar == paramVal[i]:
            for j in range(100):
                abc_mae[i].append(metrics1[1,j])
                abc_mae_skip[i].append(metrics3[1,j])
            abc_weights[i].append(nb.scraps['Skip Connection Weight'].data)
            prior_model[i].append(nb.scraps['Prior Model MAE'].data)
            abc_pre_generator[i].append(nb.scraps['ABC Pre-generator MAE'].data)
            abc_mae_skip_mean[i].append(mean(metrics3[1,:]))
            abc_mae_mean[i].append(mean(metrics1[1,:]))
```

```
[6]: data = [[] for i in range(3)]
data_median_catboost = [[] for i in range(3)]
for i in range(3):
    for j in range(len(abc_weights[i])):
        data[i].append([prior_model[i][j],paramVal[i],
                                                                                          ␣
 ↪abc_pre_generator[i][j],abc_weights[i][j],abc_mae_mean[i][j],abc_mae_skip_mean[i][j]])

    df = pd.DataFrame(data[i], columns = ['Baseline','Variance','Prior␣
 ↪Model','Weight','ABC_GAN','Skip_GAN'])
    data_median_catboost[i] = [ df['Baseline'].median(),df['Variance'].
 ↪median(), df['Prior Model'].median(),
                                df['ABC_GAN'].median(), df['Skip_GAN'].
 ↪median(),df['Weight'].median()]
print(data_median_catboost)
```

```
[[0.22586972779979714, 1.0, 0.8373863917342285, 0.2511735610103271,
0.24766350942773416, 0.761081874370575], [0.21455946827434413, 0.1,
0.22733163730941142, 0.21345637898691291, 0.2254030777299982,
0.3514516055583954], [0.2050931549501881, 0.01, 0.2050996628560008,
0.22331805785834863, 0.2063114587503047, 0.019863920286297798]]
```

3.2 ABC Pre-generator - Stats

```
[7]: book = sb.read_notebooks("./ABC_GAN_Stats")
paramVal = [1,0.1,0.01]
abc_mae = [[] for i in range(3)]
abc_mae_skip = [[] for i in range(3)]
```

```
abc_mae_mean = [[] for i in range(3)]
abc_mae_skip_mean = [[] for i in range(3)]
abc_weights = [[] for i in range(3)]
prior_model = [[] for i in range(3)]
abc_pre_generator = [[] for i in range(3)]

for nb in book.notebooks:
    metrics1 = np.array(nb.scraps['ABC_GAN_1 Metrics'].data)
    metrics3 = np.array(nb.scraps['ABC_GAN_3 Metrics'].data)
    paramVar = float(nb.papermill_dataframe.iloc[0]['value'])

    #Divide data according to parameters
    for i in range(3):
        if paramVar == paramVal[i]:
            for j in range(100):
                abc_mae[i].append(metrics1[1,j])
                abc_mae_skip[i].append(metrics3[1,j])
            abc_weights[i].append(nb.scraps['Skip Connection Weight'].data)
            prior_model[i].append(nb.scraps['Prior Model MAE'].data)
            abc_pre_generator[i].append(nb.scraps['ABC Pre-generator MAE'].data)
            abc_mae_skip_mean[i].append(mean(metrics3[1,:]))
            abc_mae_mean[i].append(mean(metrics1[1,:]))
```

[8]:
```
data = [[] for i in range(3)]
data_median_stats = [[] for i in range(3)]
for i in range(3):
    for j in range(len(abc_weights[i])):
        data[i].append([paramVal[i],prior_model[i][j],
                          ␣
 ↪abc_pre_generator[i][j],abc_weights[i][j],abc_mae_mean[i][j],abc_mae_skip_mean[i][j]])

    df = pd.DataFrame(data[i], columns = ['Variance','Baseline','Prior␣
 ↪Model','Weight','ABC_GAN','Skip_GAN'])
    data_median_stats[i] = [df['Baseline'].median(),df['Variance'].median(),␣
 ↪df['Prior Model'].median(), df['ABC_GAN'].median(), df['Skip_GAN'].median(),
                            df['Weight'].median()]
print(data_median_stats)
```

```
[[0.36067681751436, 1.0, 0.8589699324034119, 0.2606692294421806,
0.24868939553009228, 0.9962700307369232], [0.3885386412937586, 0.1,
0.39483391314328864, 0.27145286795697815, 0.2598036163691135,
0.5945071876049042], [0.35825527768550247, 0.01, 0.357203127103188,
0.27940214876403263, 0.2595625594518063, 0.15424392372369766]]
```

[9]:
```
#Output Table
output_data_stats = [[stats, gan_median] for i in range(3)]
for i in range(3):
    for j in range(6):
        output_data_stats[i].append(data_median_stats[i][j])

df_stats = pd.DataFrame(output_data_stats,␣
 ↪columns=['Baseline','GAN','Baseline','Variance','Prior␣
 ↪Model','ABC_GAN','Skip_GAN','Weight'])

output_data_catboost = [[catboost, gan_median] for i in range(3)]
for i in range(3):
    for j in range(6):
        output_data_catboost[i].append(data_median_catboost[i][j])
df_catboost = pd.
 ↪DataFrame(output_data_catboost,columns=['Baseline','GAN','Baseline','Variance','Prior␣
 ↪Model','ABC_GAN','Skip_GAN','Weight'])

display(df_stats)
display(df_catboost)
```

```
   Baseline      GAN  Baseline  Variance  Prior Model   ABC_GAN  Skip_GAN  \
0  0.384832  0.25836  0.360677      1.00     0.858970  0.260669  0.248689
1  0.384832  0.25836  0.388539      0.10     0.394834  0.271453  0.259804
2  0.384832  0.25836  0.358255      0.01     0.357203  0.279402  0.259563

     Weight
0  0.996270
1  0.594507
2  0.154244

   Baseline      GAN  Baseline  Variance  Prior Model   ABC_GAN  Skip_GAN  \
0  0.225954  0.25836  0.225870      1.00     0.837386  0.251174  0.247664
1  0.225954  0.25836  0.214559      0.10     0.227332  0.213456  0.225403
2  0.225954  0.25836  0.205093      0.01     0.205100  0.223318  0.206311

     Weight
0  0.761082
1  0.351452
2  0.019864
```