Friedman3

May 19, 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     0.559211  1.198509
1     0.539465  0.849191
2     0.527114  0.608345
3     0.481079  1.070543
4     0.576510  1.389425
5     0.449442  0.884359
6     0.644467  0.898389
7     0.691582  0.831902
8     0.549939  0.544486
9     0.555921  0.831253

Stats Model    0.552930
```

```
Catboost        0.866775
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 0x7feb44b31e50>
```

```
MEDIAN:
MSE                    0.260455
MAE                    0.397598
Euclidean Distance     2.281899
Manhattan Distance     7.951955
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)
```

```
[[0.3871505244052966, 1.0, 1.0387623669986596, 0.37262011703010645,
0.3880258403783664, 0.204212486743927], [0.3779443393632169, 0.1,
0.3735413580582606, 0.3673721093726344, 0.355285068531055, 0.28923986852169037],
[0.42389601949318334, 0.01, 0.4225384156657258, 0.423910861749202,
0.42417094468504823, 0.0]]
```

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)
```

```
[[0.5499411668699867, 1.0, 1.0697676428179261, 0.4583901086981408,
0.37041094949703374, 0.9976971745491028], [0.6053406625702988, 0.1,
0.625143836017054, 0.5839598947566003, 0.5740661852596269, 0.3009869158267975],
[0.5069318524069576, 0.01, 0.5065895870617352, 0.5247535722646862,
0.4882658434525132, 0.19477925449609756]]
```

[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   0.55293  0.397598  0.549941      1.00     1.069768  0.458390  0.370411
1   0.55293  0.397598  0.605341      0.10     0.625144  0.583960  0.574066
2   0.55293  0.397598  0.506932      0.01     0.506590  0.524754  0.488266

     Weight
0  0.997697
1  0.300987
2  0.194779
```

```
   Baseline       GAN  Baseline  Variance  Prior Model   ABC_GAN  Skip_GAN  \
0  0.866775  0.397598  0.387151      1.00     1.038762  0.372620  0.388026
1  0.866775  0.397598  0.377944      0.10     0.373541  0.367372  0.355285
2  0.866775  0.397598  0.423896      0.01     0.422538  0.423911  0.424171

     Weight
0  0.204212
1  0.289240
2  0.000000
```