Friedman2

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     0.287941  0.100321
1     0.188694  0.132098
2     0.279464  0.124398
3     0.326746  0.120304
4     0.208246  0.122657
5     0.356061  0.151503
6     0.227851  0.147860
7     0.235551  0.090044
8     0.274094  0.181755
9     0.261315  0.090944

Stats Model    0.267705
```

```
Catboost        0.123528
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 0x7fdd7cade850>
```

```
MEDIAN:
MSE                    0.005936
MAE                    0.060762
Euclidean Distance     0.342953
Manhattan Distance     1.215234
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)
```

```
[[0.13046564373870745, 1.0, 0.7795531598265417, 0.0740105579700321,
0.06983858133107423, 0.12741650640964508], [0.12747633715256595, 0.1,
0.16171777631953266, 0.0884809817271307, 0.07142530800169333,
0.22983835637569427], [0.13139145441351893, 0.01, 0.13224829098477203,
0.09366990771517159, 0.12402306892140769, 1.1432555766077712e-05]]
```

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)
```

```
[[0.2803421524470827, 1.0, 0.828875747312106, 0.08724931804221123,
0.06896554022626952, 0.9255139827728271], [0.2548423315714845, 0.1,
0.2790772676804481, 0.0892581378724426, 0.04919019117071294,
0.21587934345006943], [0.2916185794096122, 0.01, 0.2931337617804155,
0.08614225113354623, 0.07015415302729233, 0.16751329600811005]]
```

[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	0.267705	0.060762	0.280342	1.00	0.828876	0.087249	0.068966	
1	0.267705	0.060762	0.254842	0.10	0.279077	0.089258	0.049190	
2	0.267705	0.060762	0.291619	0.01	0.293134	0.086142	0.070154	

	Weight
0	0.925514
1	0.215879
2	0.167513

	Baseline	GAN	Baseline	Variance	Prior Model	ABC_GAN	Skip_GAN	\
0	0.123528	0.060762	0.130466	1.00	0.779553	0.074011	0.069839	
1	0.123528	0.060762	0.127476	0.10	0.161718	0.088481	0.071425	
2	0.123528	0.060762	0.131391	0.01	0.132248	0.093670	0.124023	

	Weight
0	0.127417
1	0.229838
2	0.000011