Systematic study of generative models

Adrien Pavao

August 2018

Issues, correcting soon:

- Need to optimize wGAN hyperparameters for better results.
- Need to re-run MDA plots because the same_size parameter was “False”.

1 Study description

The goal of this comparative study is to fairly compare the different generators. We applied various comparison metrics on original and generated datasets, with systematic settings.

Datasets

Name	Size	Variable number	Variable type	Problem type
Iris	150	4	Numerical	Multi-class classification
Mushrooms	6499	23	Categorical	Binary classification
Boston housing	506	14	Mixed, mostly numerical	Regression
Adult	48842	15	Mixed	Binary classification
MIMIC (Medical applications)	26927	343	Mixed, lots of binary var	Binary classification

Table 1: Datasets description

Each dataset has a train and a test set. The train set represent 70% of the entire dataset.

Data is available on GitLab on a private repository[1].

Processing

SDV pre-processing.

The target variable is then decoded.

Invert pre-processing : de-code the pre-processing to restore categorical variables.

re-scale and shift (in case continuous variables were scaled)

Generative models

Models are trained with the real data **train** set.

- **Multivariate gaussian** as a baseline.
- **Parzen windows** kernel density estimation. Bandwidth = 0.2, gaussian kernel, euclidean metric.
- **Random Forest imputation** probability = 1.
- **Wasserstein GAN** (parameters? how many epochs?)

[1] `http://gitlab.com/didayolo/data`

Number of sample generated for each generator:

For Iris, Mushrooms, Boston: 10,000 samples. For MIMIC: 20,000 samples. For Adult: 30,000 samples.

Except for Random Forest which gives synthetic datasets the same size as the original ones. It is not interesting to sample more points with this algorithm if the proportion of replacement parameter p is equal to 1, because this method is deterministic.

Metrics

All metrics are computed on real data **test** set and generated data (except classification/regression performance).

- **Principal component analysis** (PCA) plots. We overlay the scatter plot of the real data test set and generated data.
- **Minimum distance accumulation** (MDA). Nearest neighbor criterion obtained by plotting the distribution of distances of one example in D1 with its nearest neighbor in D2 (and vice versa: symmetrize). Vary the privacy threshold. Plot privacy (area over curve below threshold) vs. resemblance (area under curve beyond threshold). Quantitative metric = area under privacy vs. resemblance curve. Use the TEST set (triangular comparisons also possible but make at least a test/gene comparison)
- **Classification/regression performance**. The model is a Random Forest with 50 estimators. There are three experience: train on real data then test on real data, train on real data then test on synthetic data, train on synthetic data and test on synthetic data. The scoring metric is **accuracy** for classification and **r2** for regression. r^2 best possible score is 1 while bad scores can be arbitrarily negative.
- **Discriminant score**: a classifier is trained to output the probability of a given sample being real or fake. The model trainined is a MultiLayer Perceptron with layers size (100, 200), ReLU activation function and a constant 0.001 learning rate.

For PCA, MDA and discriminant score datasets are re-sampled at the same size.

2 Results

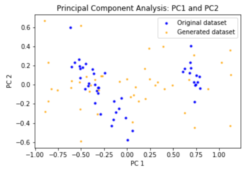
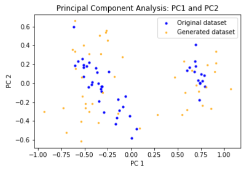
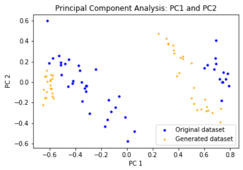
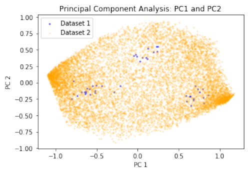
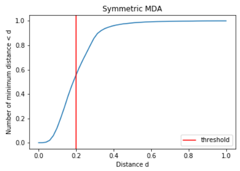
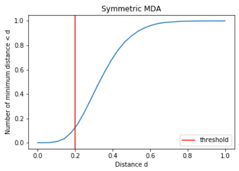
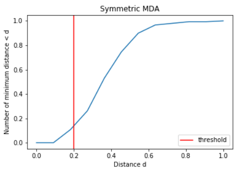
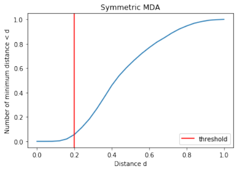
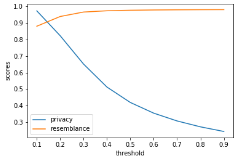
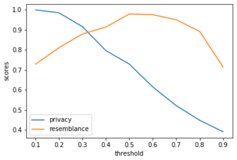
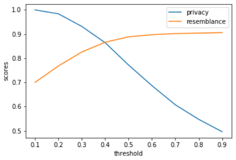
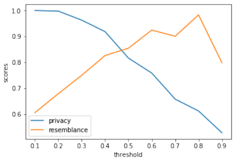
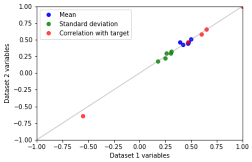
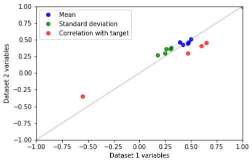
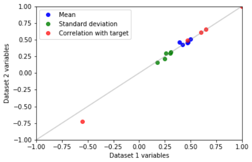
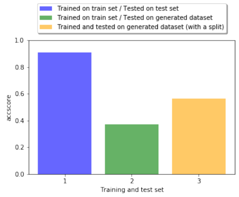
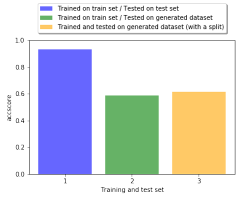
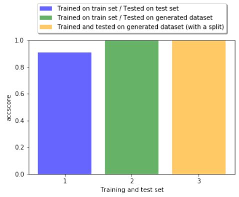
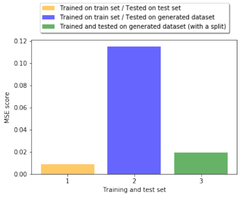
Generator	Multivariate Gaussian	Parzen Windows	Random Forest	Wasserstein GAN
PCA plot				
MDA plot				
Threshold variation				
Marginals				TODO
Task score				
Discriminant score	precision recall Original dataset 0.78 0.01 Generated dataset 0.67 1.00 avg / total 0.71 0.67	precision recall Original dataset 0.89 0.01 Generated dataset 0.68 1.00 avg / total 0.75 0.68	precision recall Original dataset 0.56 0.38 Generated dataset 0.62 0.76 avg / total 0.59 0.60	TODO

Table 2: **Iris results**

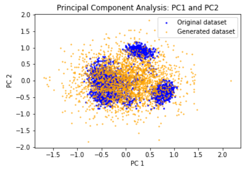
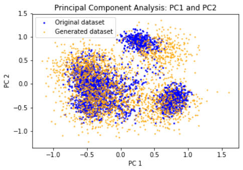
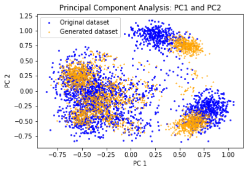
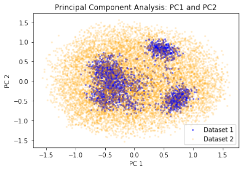
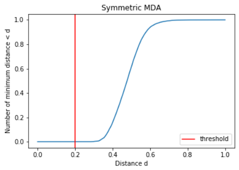
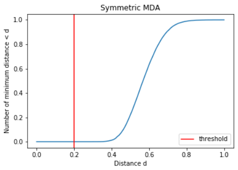
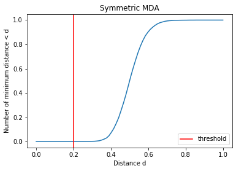
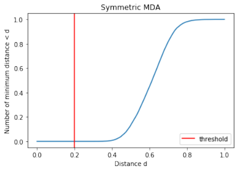
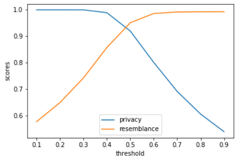
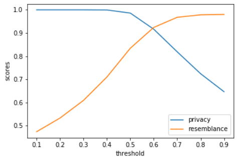
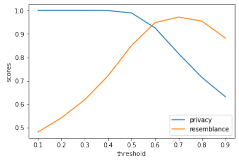
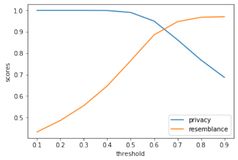
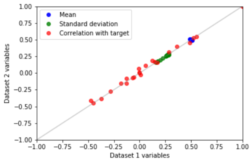
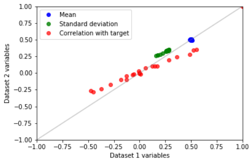
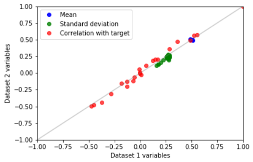
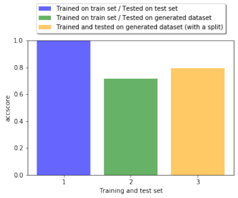
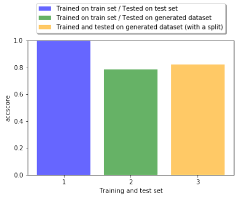
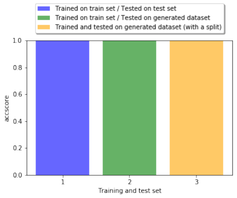
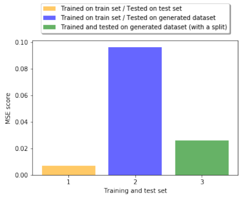
Generator	Multivariate Gaussian	Parzen Windows	Random Forest	Wasserstein GAN
PCA plot				
MDA plot				
Threshold variation				
Marginals				TODO
Task score				
Discriminant score	precision recall Original dataset 0.97 0.80 Generated dataset 0.94 0.99 avg / total 0.95 0.95	precision recall Original dataset 0.89 0.60 Generated dataset 0.85 0.97 avg / total 0.86 0.86	precision recall Original dataset 0.77 0.68 Generated dataset 0.85 0.89 avg / total 0.82 0.82	TODO

Table 3: **Mushrooms results**

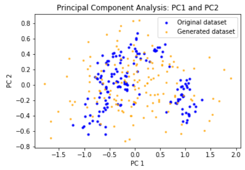
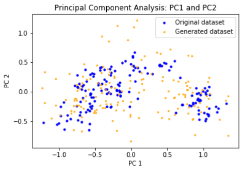
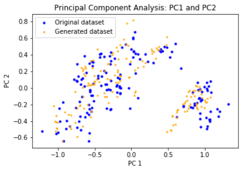
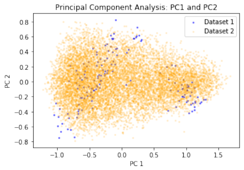
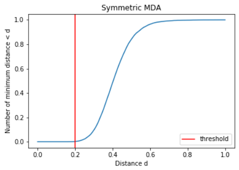
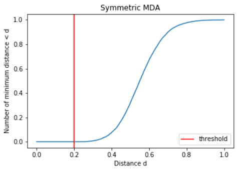
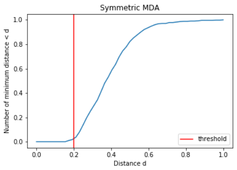
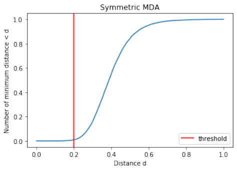
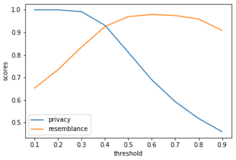
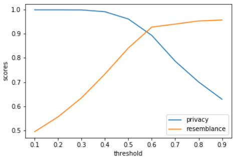
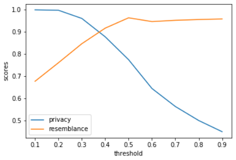
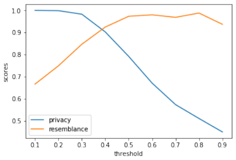
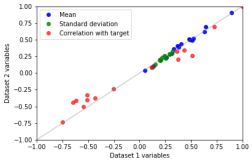
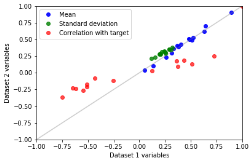
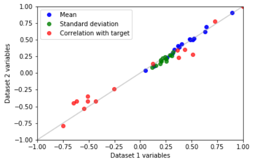
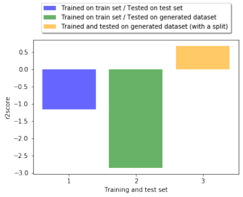
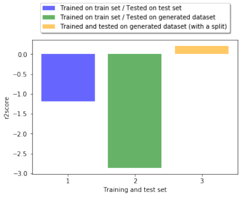
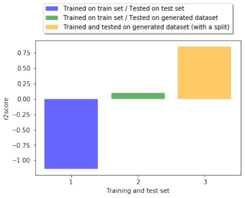
Generator	Multivariate Gaussian	Parzen Windows	Random Forest	Wasserstein GAN
PCA plot				
MDA plot				
Threshold variation				
Marginals				TODO
Task score				TODO
Discriminant score	precision recall Original dataset 0.87 0.04 Generated dataset 0.68 1.00 avg / total 0.74 0.68	precision recall Original dataset 0.90 0.06 Generated dataset 0.78 1.00 avg / total 0.81 0.78	precision recall Original dataset 0.67 0.42 Generated dataset 0.60 0.81 avg / total 0.63 0.62	TODO

Table 4: **Boston results**

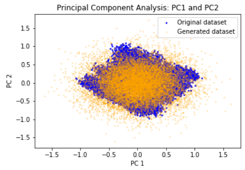
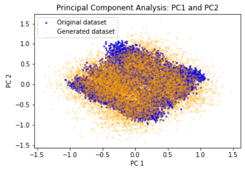
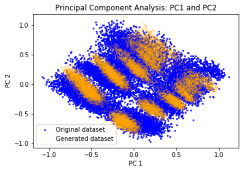
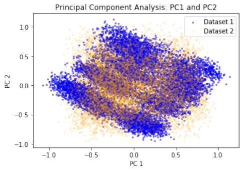
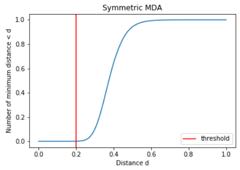
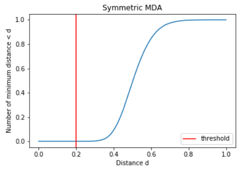
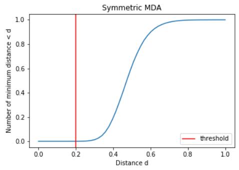
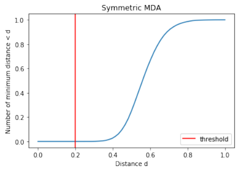
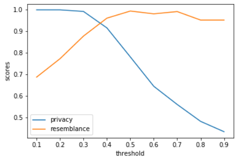
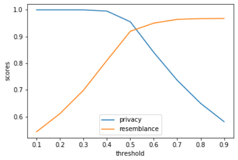
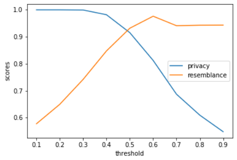
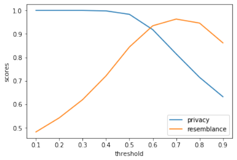
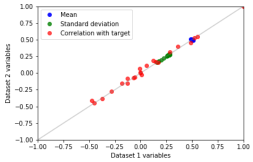
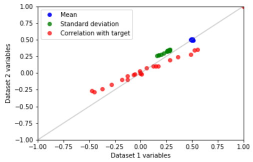
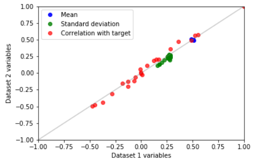
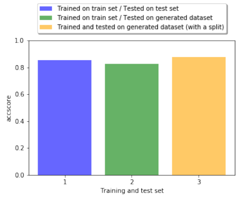
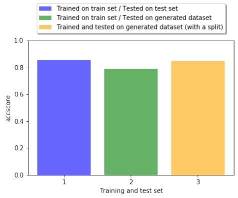
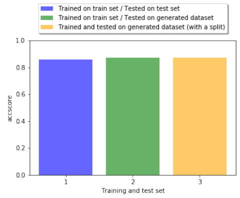
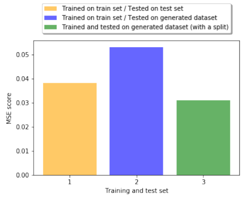
Generator	Multivariate Gaussian	Parzen Windows	Random Forest	Wasserstein GAN
PCA plot				
MDA plot				
Threshold variation				
Marginals				TODO
Task score				
Discriminant score	precision recall Original dataset 0.92 0.75 Generated dataset 0.85 0.96 avg / total 0.88 0.87	precision recall Original dataset 0.90 0.78 Generated dataset 0.88 0.95 avg / total 0.89 0.89	precision recall Original dataset 0.68 0.49 Generated dataset 0.69 0.84 avg / total 0.69 0.69	TODO

Table 5: **Adult results**

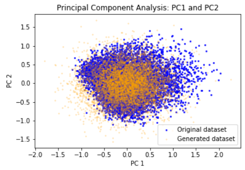
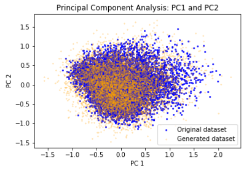
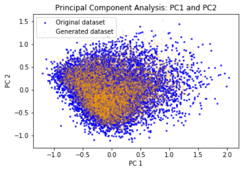
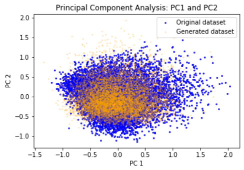
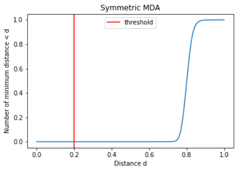
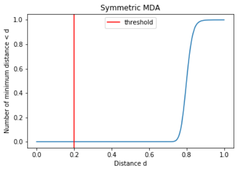
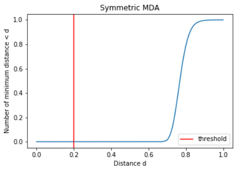
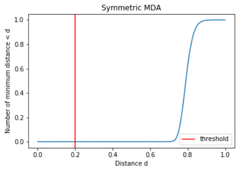
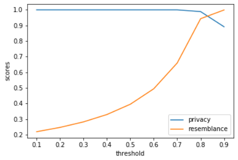
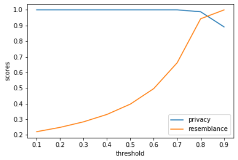
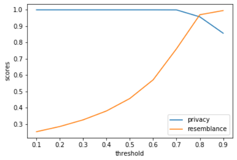
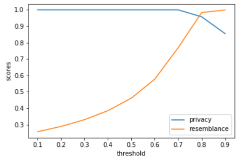
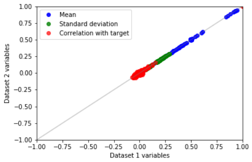
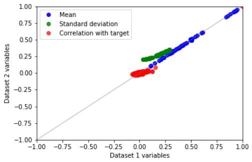
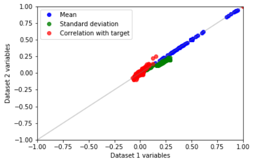
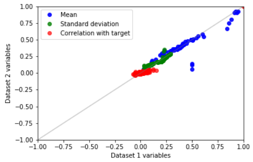
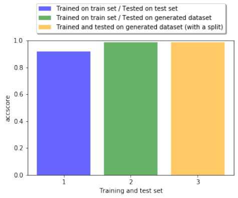
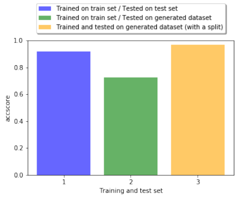
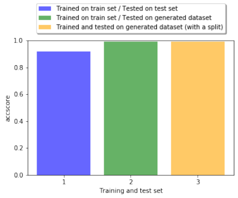
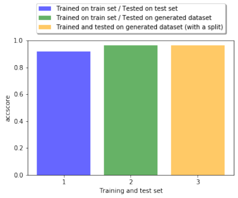
Generator	Multivariate Gaussian	Parzen Windows	Random Forest	Wasserstein GAN
PCA plot				
MDA plot				
Threshold variation				
Marginals				
Task score				
Discriminant score	precision recall Original dataset 0.56 0.33 Generated dataset 0.54 0.75 avg / total 0.55 0.54	precision recall Original dataset 0.81 0.66 Generated dataset 0.83 0.92 avg / total 0.82 0.82	precision recall Original dataset 0.55 0.35 Generated dataset 0.55 0.74 avg / total 0.55 0.55	precision recall Original dataset 0.77 0.22 Generated dataset 0.78 0.98 avg / total 0.78 0.78

Table 6: **MIMIC results**

Analysis

1) UTILISER LES COULEURS POUR LES CLASSES ET LA FORME POUR LE DATASET D'ORIGINE ???

2) Il serait peut-être mieux d'utiliser LDA (les directions qui séparent le mieux les classes) et de montrer des symboles différents pour les classes. Mettez des couleurs différentes pour échantillons reels et échantillons artificiels.

3) Variez les paramètres du GAN !!!