Missing Value Imputation using Generative Adversarial Networks

Shubhangi Kishore
dept. name of Computer Science
Stony Brook University
Stony Brook, United States
shubhangi.kishore@stonybrook.edu

***Abstract*—We explore techniques for imputing missing values using machine learning. The presence of missing values is a frequent problem encountered in data analysis and machine learning applications. The purpose of this work is to create a data imputation framework to impute values for a numerical dataset in particular using Generative Adversarial Networks. The approach discussed attempts to provide an understandable solution through detailed error analysis. Our analysis assesses the strength and weakness of GAIN and demonstrates the potential and utility of using GANs in multimodal and diverse UCI datasets. The framework is based on optimization of GAIN algorithm to impute missing data with mixed continuous and categorical variables for multiple imputation with a visualisation toolkit to analyse errors.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

Missing data typically have a negative impact on machine learning models. With the rise of generative models in deep learning, recent studies proposed solutions to the problem of imputing missing values based various deep generative models. Previous experiments with Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) showed promising results in this domain. In recent years there has been a growing focus on explainable AI and a need for transparency in ML models. This includes all steps that involve preparing and preprocessing the data. Data with missing values can decrease model quality and even lead to wrong insights. Initially, these results focused on imputation in image data, e.g. filling missing patches in images. Recent proposals addressed missing values in tabular data. For these data, the case for deep generative models seems to be less clear. between imputation and software data quality. Missingness mechanism used is - Missing Completely at Random (MCAR) where occurrence of missing data dose not depend on both observed and missing values. Missing data deletion commonly used may lead to significant information loss and result in biased models. To help researchers impute scores using GAN based imputation methods and visualize the technique, two python subroutines subroutines were written. The a wrapper was created for the gain algorithm and an explainable GAIN toolkit was developed to visualize the results and errors with a scatter-plot matrix, PCA plots to observe which data points have more imputation errors and how the error varies across attributes.

Mentored by Professor Klaus Mueller, Stony Brook University.

II. Related Work

Below we provide some preliminary introduction to missing data imputation mechanisms based on GANs and denoising autoencoders.

A. GAN

GANS can be used to generate distributions with a different dimension that provides a solution for data imputation. Generative deep models are trained through an adversarial process with two networks - a generator that maps from latent space to the data distribution and a discriminator that distinguishes candidates produced by the generator from the true data distribution. There is a feedback loop between the generator and discriminator where a generator imputes missing data to deceive the discriminator which discriminates between the imputation and fake complete data generated.The advantage of using GAN is that the model learns the hidden data distribution.

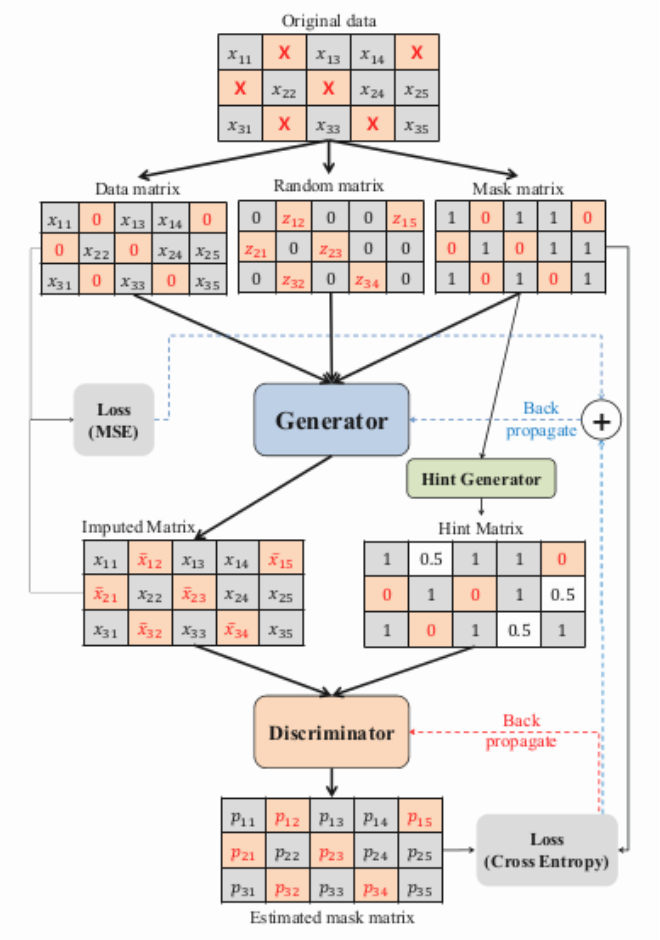
B. GAIN

GAIN: Missing Data Imputation using Generative Adversarial Nets proposed imputing missing data by adapting the GAN framework. [1] The generator receives noise and mask as input and observes components of a real data vector and imputes the missing components, and outputs a completed vector. The discriminator then takes a completed vector and attempts to determine which components were actually observed and which were imputed. GAIN uses a hint mechanism where the discriminator gets some additional information in the form of a hint vector. Both imputed matrix and hint matrix are input data for a discriminator. The authors solve the minimax optimization problem in an iterative manner. Both G and D are modeled as fully connected neural nets. The data type is categorical and numeric value.

C. MisGAN

MisGAN is another GAN framework handles missing data. It makes use of 2 Generators and 2 Discriminators. The Generator Gx generates complete data and the Generator Gm produces the mask for missing data . It is then compared in Discriminators Dx and Dm to check whether it can be distinguished from the matrix of real data x and the mask matrix of real missing values m. To create a missing data

Fig. 1. GAIN framework [1]



Imputet the pre-trained Generator Gx generates complete data and Generator G acts as an Imputer. It is fed up to a single Discriminator D and checked whether G produces satisfactory results from data with missing values. The authors experiment on the MNIST Data set and three others and use FDI as the evaluation metric. [3]

D. VIGAN

VIGAN approaches the problem of missing data where there exists multimodal data that lacks the one to one mapping between the datasets from different sources. VIGAN is the model for imputing missing views based on generative adversarial networks which combines cross-domain relations given unpaired data with multi-view relations given paired data. The evaluation of VIGAN in conducted on a genetic study of substance use disorders dataset. [2]

III. Theoretical Analysis

A. GAIN Based Framework

GAIN maximizes the probability of correctly predicting values that are imputes and the ones that are observed. The figure below shows the overall architecture of the GAIN algorithm. [**?**]

B. Imputation Strategy

d ← Dimensions, X ← Data Vector, M ← Masked Vector, Z ← Noise Vector, H ← Hint Vector, X_M ← generated incomplete dataset. [1]

- The generator's goal is to accurately impute missing data, and the discriminator's goal is to distinguish between observed and imputed components. The calculation of the reconstructed observed values ensures that the values G is generating for the observed values approximate observed values. G also maps noise values to the data space.
  $G : \mathcal{Z} \times \mathcal{X} \times \mathcal{M} \times \mathbb{R}^n \rightarrow \mathcal{X}$

- The discriminator is trained to minimize the classification loss when classifying which components were observed and imputed. X_M and H is passed to the Discriminator the output of D is an estimated mask matrix Mˆ that gives the probability of each value in X_M being observed.
  $D : \mathcal{X} \times \mathcal{H} \times \mathbb{R}^n \rightarrow [0,1]^k$

- Hint Mechanism is a random variable H that depends on M and for each (imputed) sample (xˆ, m), we draw h according to the distribution $H|M = m$. H supports D to distinguish whether a value is imputed or observed by informing D about a percentage of values and whether they are imputed or observed. $H = B \odot M + 0.5(1 - B)$ where $B \in \{0,1\}^d$.

1) Learning Objective: The objective is maximizing the probability of correctly predicting which value is imputed and which one is observed. This results in the following minmax problem - The loss function is defined as $\mathcal{L}(D,G) = \mathbb{E}\left[M^T \log \hat{M} + (1-M)^T \log(1-\hat{M})\right]$ where $\hat{M} = D\left(\hat{X}, H; \theta_d\right)$.

IV. Experimental Analysis

A. Dataset Distribution

These diverse UCI Datasets are used in this study - Heart Dataset, Spam Dataset, Letters Dataset, Breast Cancer. Three missingness percentages were used (5, 10 and 15). Here are some attributes of the breast cancer dataset for which we have discussed the results in detail in this work - Diagnosis, real-valued features are computed for each cell nucleus: radius, texture (standard deviation of gray-scale values), perimeter, area, smoothness (local variation in radius lengths), compactness, concavity (severity of concave portions of the contour), concave points (number of concave portions of the contour), symmetry, fractal dimension. [**?**] This figure represents the distribution of some of the attributes -

B. Hyperparameters

Hyperparemeters modified and tested include (grid search) - miss rate: probability of missing components, batch size, hint rate, Alpha: learning rate, number of iterations. For the D and G we have these parameters - Number of layers, Size

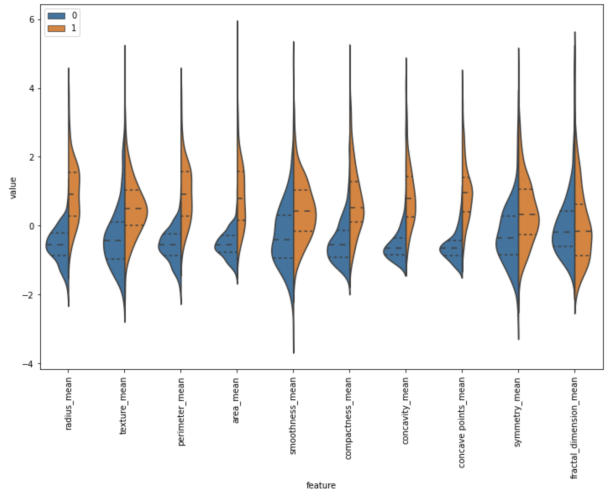


Fig. 2. Feature distribution of Breast Cancer Dataset

of layers, Activation function, Learning rate and Momentum. ReLU activation function was used.

C. Training

The purpose of training the model is to reduce the gap between the reconstructed data and the original data. The process are shown as follows: The input vector is created by introducing missingness in our complete datasets. We set the percentage of missing values in our dataset and create a binary mask. Values from the masked data part is equal to 0, and the rest values are keeping still the original values. With different sizes of masks, we generate datasets with varying Miss Rate. In Gain we first optimize the discriminator D with a fixed generator G using mini-batches, we optimize the generator G using the newly updated discriminator D. G and D are updated using stochastic gradient descent as described in section 3. This is repeated till the training loss has not converged. Imputation method - GAIN was applied. Imputation results and accuracy of imputation methods are compared using sum of root mean squared error and average absolute error calculated per attribute on the test set, given as $RMSE(\hat{x}) = \sqrt{MSE(\hat{\boldsymbol{x}})} = \sqrt{\mathbb{E}\left((\hat{\boldsymbol{x}} - \boldsymbol{x})^2\right)}$.

V. Results

The prediction performance for the following datasets given different missing rates was measured -

Dataset	RMSE	Missing Data Percentage
Heart	0.185	0.2
	0.11	0.1
Letter	0.24	0.2
	0.140	0.1
Spam	0.1687	0.2
	0.0661	0.1
Breast Cancer	0.1354	0.2
	0.1014	0.1

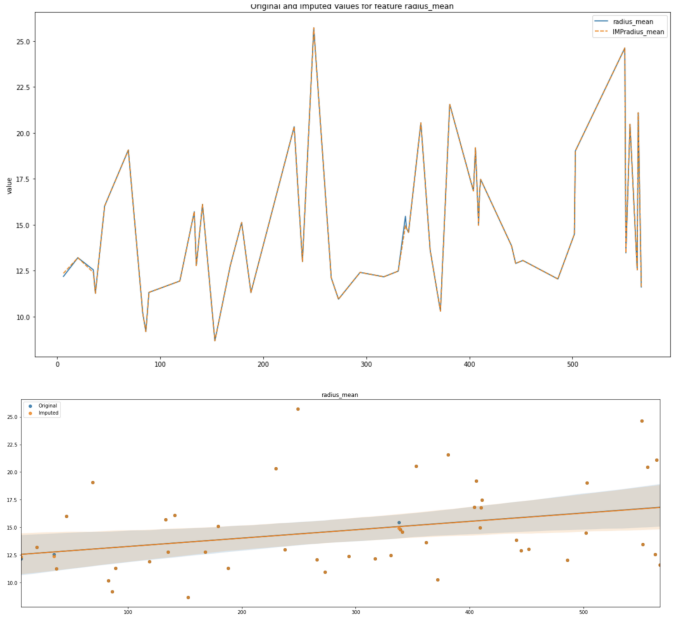


Fig. 3. Line plot for observed and generated mean radius values.

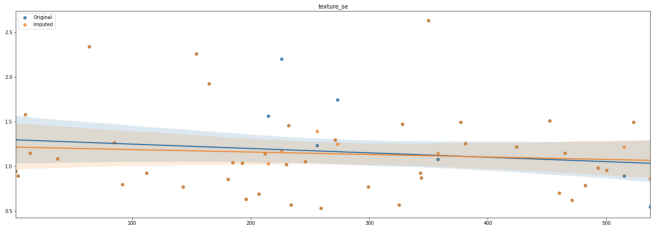


Fig. 4. Observed and generated texture values.

The variance and mean absolute error reported were 0.0853 , 0.0815 respectively. The specific results and error analysis is discussed in this section on the real life breast cancer data-set containing mixed, continuous variables to better understand the algorithm and make the results more explainable.

A. Absolute Error Analysis

To observe how the error varies for different attributes, the imputation model is also evaluated by measuring the effects on the feature-label relationships after the imputation.

To understand the error in a multivariate sense via appropriate data visualizations. We pick two variables from the dataset, var 1 and var 2, and make a bivariate scatterplot with all data points. Scatterplots are essential for diagnosing relationships, to find out how to summarize it best and whether the tools you intend to use to summarize the relationship are applicable.

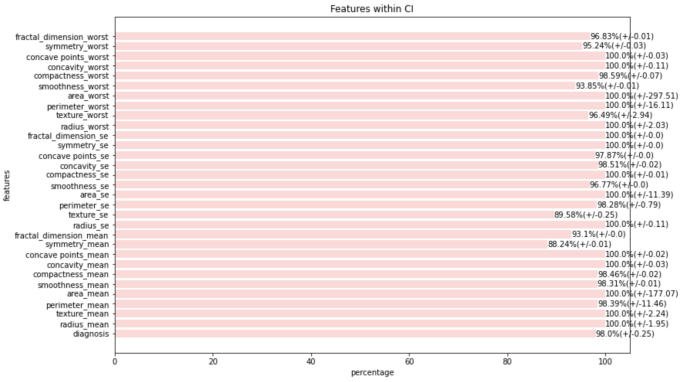
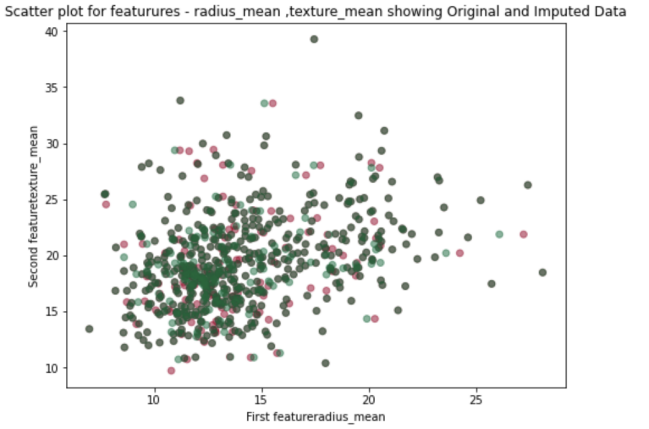
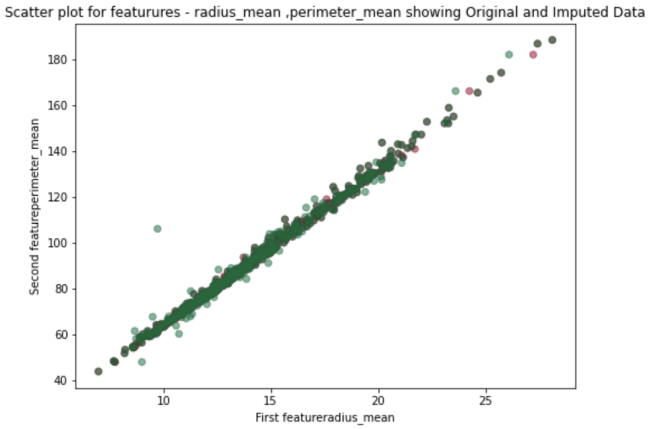


Fig. 5. Confidence Intervals for different attributes.



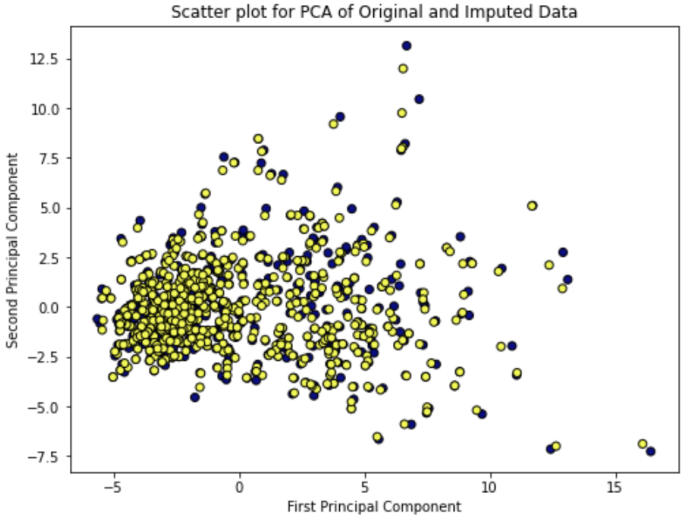


This figure exhibits the features and corresponding error bounds.

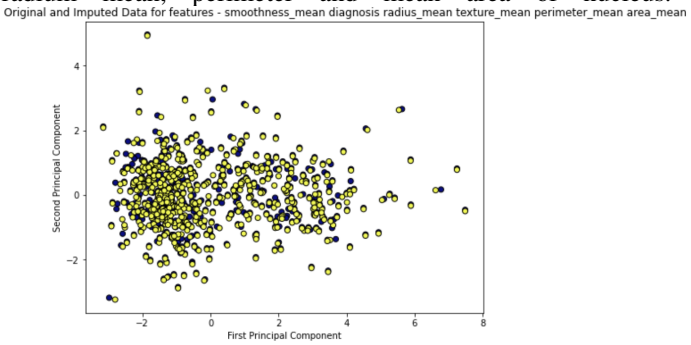
The GAN works better for some features than others. The dependent column was predicted with no error (categorical). Certain columns like texture se showed a comparatively

B. PCA Analysis

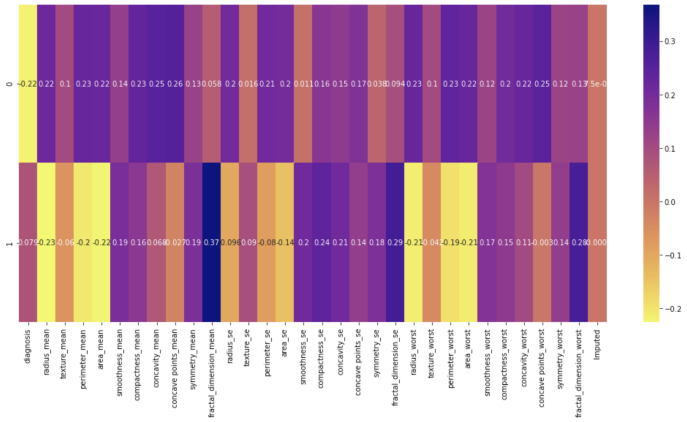
Here we visualize the results and errors with a scatter-plot matrix and a PCA plot and observe which data points have more imputation errors. Observations are made as to how sets of features translate to PCA plots. For instance, we were able to compare representations of features in both the original and imputed datasets so that factors that could contribute to how GAIN model operates and makes its predictions are better understood. Yellow is used for the Imputed Dataset and Blue for the original. Here we have used all features for the PCA -



We then picked sets of 4 features and repeated the experiment. In the figure below PCA is performed for smoothness, radium mean, perimeter and mean area of nucleus.



The first principal components, a linear combination of original predictor variables which captures the maximum variance in the data set. Larger the variability captured in the first component, larger the information captured by the component. We were able to consider how much of the total variance was explained by each of our principal components. The order of PCA components is represented in the heatmap here.



C. Future Work

Some of the concerns and drawbacks of gain include - GAIN training is slow and the loss of the Discriminator is difficult to stabilize and can result in vanishing gradients as well, this work does not improve on existing results by adding different layers and vary the number of nodes in each layer.In this work the variation in error due to changes in hyper parameters were quite interesting but were not fully explored. Another meaningful experiment would be to look at features that have a higher influence on the data than others and see how it influences the performance when these features are missing rather than less meaningful features.Another aspect to be considered when furthering this research is how the algorithm can be modified based on the type of attributes present in the dataset - categorical vs continuous. The current work is made open source and availaible on GitHub.

ACKNOWLEDGMENT

I would first like to thank my supervisor, Professor Klaus Mueller, whose expertise was invaluable in this body of research. Your feedback directed this work and research questions increased my understanding of GAN algorithms. I would like to acknowledge my colleagues at Stony Brook University for their wonderful collaboration.

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] Shang, Chao, et al. ”VIGAN: Missing view imputation with generative adversarial networks.” 2017 IEEE International Conference on Big Data (Big Data). IEEE, 2017.

[3] Li, Steven Cheng-Xian, Bo Jiang, and Benjamin Marlin. ”Misgan: Learning from incomplete data with generative adversarial networks.” arXiv preprint arXiv:1902.09599 (2019).

[4] Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[5] Wang, Yufeng, et al. ”PC-GAIN: Pseudo-label Conditional Generative Adversarial Imputation Networks for Incomplete Data.” arXiv preprint arXiv:2011.07770 (2020).