# Python Module for Missing Data Imputation

ECE 298 (SPRING 2019) REPORT

Ambareesh SN Jayakumari | A53268696 | 06/17/2019

# Introduction

Missing data is an important issue, even small proportions of missing data can adversely impact the quality of learning process, leading to biased inference. Many methods have been proposed over the past decades to minimize missing data bias and can be divided into two categories: One that attempt to model the missing data process and use all available partial data for directly estimating model parameters and two that attempt to fill in/impute missing values with plausible predicted values. Imputation methods are preferred for their obvious advantage, that is, providing users with a complete dataset that can be analyzed using user specified models *(Gondara and Wang, 2018)*

As part of literature review for the project different approaches to the problem was studied. The rest of the report is arranged as sections which cover each of the following:

*Section I: SRMI (Sequential Regression Multiple Imputation)*

*Section II: CART (Classification and Regression Trees)*

*Section III: Soft-Impute*

*Section IV: MIDA (Multiple Imputation using Denoising Autoencoders)*

## SECTION I: SRMI

Motive - From a Bayesian perspective, missing values follow a distribution given the observed values, and we are trying to find this predictive distribution (and not a single imputation value)

Idea - Imputing several plausible sets of missing values in the incomplete data set resulting in several completed data sets

Assuming:

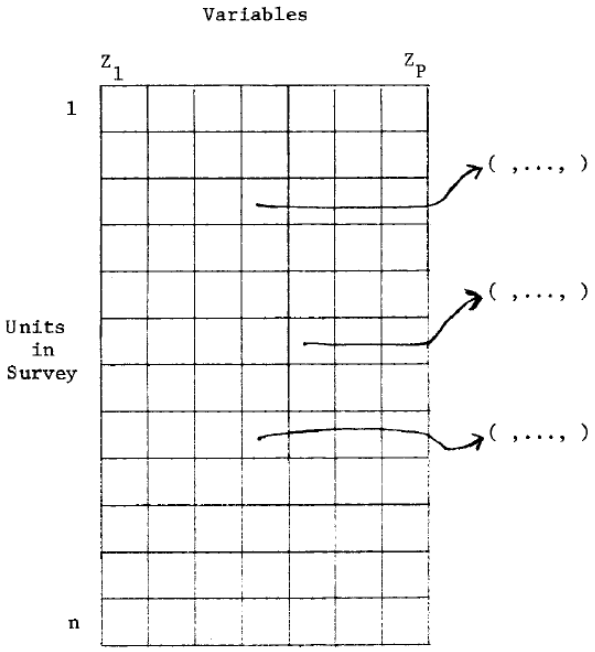
- Population is essentially infinite
- Sample is a simple random sample

*Missing data mechanism is ignorable*

**Multiple Imputations so far:**

- Used a Bayesian framework, where an explicit model is specified for all variables with missing values, conditional on the fully observed variables, unknown parameters, and prior distribution for these parameters
- The model generates a posterior predictive distribution of missing values given the fully observed values

Imputations are draws from this predictive distribution (Rubin (1976))



Model for first imputation = . . .

Model for second imputation = . . .

.

.

Model for Ith imputation = . . .

Data Set with Multiple Imputations for Each Missing Datum

## <u>Multiple Imputations – Challenges</u>

Specifying a Bayesian model is often difficult, because of three reasons:

1. Survey data usually has many variables (hundreds) of varying forms and distributions (counts, categorical, continuous, and within continuous – normal, lognormal and others).

2. Restrictions that arise from nature of certain variables (E.g. "Income from second job", imputations must be restricted to people who indicated they had a second job)

3. Logical bounds that must be maintained while doing imputations (E.g. "Years since quit smoking" should be bound by [0, Age -18] if no evidence of teenage smoking is seen)

*All the above are addressed by SRMI*

## SRMI – Imputation Method

- X: Design matrix (n x p) for all n observations and p variables with no missing values
- $Y_1, Y_2 \,..\, Y_k$ all variables with missing values, ordered from lowest to highest
- Joint Conditional Density equation factored into conditional density functions with parameters $\Theta_1$, $\Theta_2$ .. $\Theta_k$
- Each f is a conditional density function and is modelled according to the type of the variable. Draw values from the corresponding predictive distribution, given X (observed values)

$$f(Y_1,\ Y_2,\ \ldots,\ Y_k \mid X,\ \theta_1,\ \theta_2,\ \ldots,\ \theta_k) = f_1(Y_1 \mid X,\ \theta_1) f_2(Y_2 \mid X,\ Y_1,\ \theta_2) \ldots\ \ f_k(Y_k \mid X,\ Y_1,\ Y_2,\ \ldots,\ Y_{k-1},\ \theta_k)$$

Each imputation consists of c rounds

- Round 1 – Imputation done sequentially on all Y variables starting from the one with the lowest number of missing values. (Explain with e.g.) and updating X
- Round 2 to Round c – Repeatedly impute all Y variables by using everything except that particular variable (Explain with e.g.)
- Impute until Round c/ Convergence

Modifications needed to include restrictions and bounds

- Example of restrictions –Models have to be fitted to appropriate subset of individuals. (No. of pregnancies, No of years smoking cigarettes) + Dummy variables for subsequent regressions
- Truncated normal linear regression/SIR
- Can also be extended to polytomous variables.

Conditional regression $f_1(x)(Y_1|X,\ \theta_1)$ for each type:

- Normal linear regression model if $Y_j$ is **Continuous**
- A logistic regression model if $Y_j$ is **Binary**
- Generalized logit regression model if $Y_j$ **Categorical**
- Poisson loglinear model if $Y_j$ is a **Count**

- Two-stage model (logistic regression, + normal linear regression model) if $Y_j$ is **Mixed**

**Simulation Study:**

Steps:

1. Generate a complete dataset from a hypothetical population
2. Estimate selected regression parameters
3. Delete certain values (Ignorable missing data mechanism)
4. Use SRMI to impute deleted values
5. Obtain new multiply imputed estimated for the same parameters from second step
6. Examine the differences

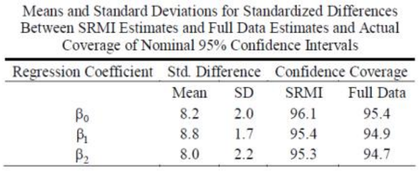
A total of 2,500 complete data sets with three variables ($U$, $Y_1, Y_2$ ) and sample size 100 were generated using the following models:

1. $U$ : Normal (0, 1)
2. $Y_1$ : Gamma with mean $\mu_1 = \exp(U\text{-}1)$ and variance $\mu_1^2/5$
3. $Y_2$ : Gamma with mean $\mu_2 = \exp(-1+0.5U+0.5Y1)$ and variance $\mu_2/2$

Model for $Y_2$ is the primary interest

**Results:**

- For each of the 2,500 simulated data sets with missing values, a total 250 rounds with $M = 5$ different random starts were created using SRMI
- For each replicate, the resulting imputed data sets and the full data set were analyzed by fitting the Gamma model for $Y_2$ using maximum likelihood
- To assess the differences in the point estimates standardized difference between the SRMI and full data estimates
- SRMI data resulted in well calibrated intervals estimates



Means and Standard Deviations for Standardized Differences Between SRMI Estimates and Full Data Estimates and Actual Coverage of Nominal 95% Confidence Intervals

| Regression Coefficient | Std. Difference | | Confidence Coverage | |
|---|---|---|---|---|
| | Mean | SD | SRMI | Full Data |
| $\beta_0$ | 8.2 | 2.0 | 96.1 | 95.4 |
| $\beta_1$ | 8.8 | 1.7 | 95.4 | 94.9 |
| $\beta_2$ | 8.0 | 2.2 | 95.3 | 94.7 |

**Discussion:**

- Describes and evaluates SRMI technique useful for imputing values in wide variety of complex data structures.
- Ideal when the joint distribution of variables with missing values, is too hard to formulate.
- Advantage – Dealing with missing variables on a case by case basis (and fully conditional on all the observed values)
- Imputations from SRMI and Bayesian models are comparable as proved by case study
- Key Assumption – Dataset arises from a simple random sample design
- Disadvantage – Large datasets could be computationally intense
- Issues - How do you decide which data is suitable for SRMI?

## SECTION II: CART

Motive - Primary mission of most national statistical agencies is to disseminate data to the public. However, government agencies are under increasing pressure to limit access to data because of growing threats to data confidentiality. Even stripping obvious identifiers like names, addresses, and identification numbers may not be enough to protect confidentiality. To protect confidentiality in public use datasets, many statistical agencies release data that have been altered to protect confidentiality.

### Synthesis of Data - So far:

- Rubin (1993) and Little (1993) suggested that agencies release partially synthetic data, which comprise the original units surveyed with some collected values replaced with multiple imputations.
- The key to the success of synthetic data approaches, especially when replacing many values, is the data generation model.
- Current practice for generating synthetic data typically employs sequential modeling strategies based on parametric or semiparametric models similar to those for imputation of missing data in Raghunathan et al. (2001).
- The basic idea is to impute $Y_1$ from a regression of $Y_1$ on ($Y_2$, $Y_3$, etc.), impute $Y_2$ from a regression of $Y_2$ on ($Y_1$, $Y_3$, etc.), impute $Y_3$ from a regression of $Y_3$ on ($Y_1$, $Y_2$, etc.), and so on.

### Synthesis of Data- Challenges:

- Specifying these conditional imputation models can be daunting in surveys with many variables. Often datasets include numerical, categorical, and mixed variables, some of which may not be easy to model with standard tools.
- The relationships among these variables may be non-linear and interactive.
- Specifying models for many variables is a resource-intensive task, and statistical agencies simply may not have the time to invest in careful specification of these conditional models for many variables.

### Identification of disclosure risk measures

- Intruder has a vector of information, t, on target units in population.

  $t_o$ : unique identifier of the target

  $d_{jo}$: (not released) unique identifier for record j in D, where j = 1, .. , n
- Intruder seeks to match unit j in D to the target when $d_{jo} = t_o$, and not to match when $d_{jo} \neq t_o$ for any $j \in D$.

- Let J be a random variable that equals j when $d_{jo} = t_o$ for j ∈ D, and equals n+1 when djo = to for some j∉ D.
- Intruder seeks to calculate the Pr (J = j|t,D) for j = 1, . . . , n
- If the identification probabilities are large enough (decided on a threshold) they can be declared an identification.

But the intruder does not know $Y_{rep}$, so for each record in D he computes:

$$\Pr(J = j|\mathbf{t}, D) = \int \Pr(J = j|\mathbf{t}, D, Y_{\text{rep}})\Pr(Y_{\text{rep}}|\mathbf{t}, D)\mathrm{d}Y_{\text{rep}}$$

For comparing the risks of the nonparametric synthesizers, authors assume that the intruder approximates identification probability by treating the simulated values in the released datasets as plausible draws of $Y_{rep}$

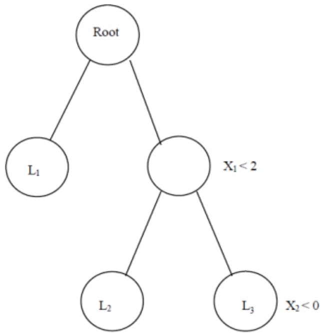
Matching probability for any record j is then:

$$\Pr(J = j|\mathbf{t}, D) = (1/m)\sum_{l}(1/F_{\mathbf{t}})(D_j^{(l)} = \mathbf{t})$$

- $F_t$ is the number of records in the population that satisfy the matching criteria
- Logical expression $(D^{(l)}_j = t)$ equals 1 when the values of variables used for matching by intruders for record j are in accord with the corresponding values in t; else 0
- Assume Intruder doesn't know if target is included in D. If the intruder knows who is in D, $F_t$ is replaced with $N_{(l)}$(the number of records in $D_{(l)}$ that satisfy the matching criteria)

**CART (Classification and Regression Trees)**

- CART seek to approximate the conditional distribution of a univariate outcome from multiple predictors.
- The algorithm partitions the predictor space so that subsets of units formed by the partitions have relatively homogeneous outcomes.
- The partitions are found by recursive binary splits of the predictors. The series of splits can be effectively represented by a tree structure, with leaves corresponding to the subsets of units.
- The values in each leaf represent the conditional distribution of the outcome for units in the data with predictors that satisfy the partitioning criteria that define the leaf



Example for a simple Classification and Regression Tree using just two rules.

**CART for synthesis**

Using only records with $Z_j = 1$, fit the tree of $Y_i$ on $Y_{-i}$ so that each leaf contains at least k records - call this tree $Y^{(i):}$

- For categorical variables, we grow Y(i) by finding the splits that successively minimize the Gini index;

- For numerical variables, we successively minimize the deviance of Yi in the leaves.

Stop splitting any leaf when:

- The deviance in that leaf is less than some agency-specified threshold d, Or

- When we cannot ensure at least k records in each child leaf.

Only records with $Z_j = 1$ are used to ensure that the tree is tailored to the data that will be replaced:

- For any record with $Z_j = 1$, trace down the branches of $Y^{(i)}$ until that record's terminal leaf. Let $L_w$ be the w^th^ terminal leaf in $Y^{(i)}$, and let $Y^{(i)}{}_{Lw}$ be the $n_{Lw}$ values of $Y^{(i)}$ in leaf $L_w$. For all records whose terminal leaf is $L_w$, we generate replacement values of $Y^i{}_j$ by drawing from $Y^{(i)}{}_{Lw}$ using the Bayesian bootstrap (Rubin, 1981)

- Repeating the Bayesian bootstrap for each leaf of $Y^{(i)}$ results in the $l^{th}$ set of synthetic values. Repeat this process m times to generate m datasets with synthetic values of $Y^i$

**Empirical Evaluation: Simulation Study**

Idea:

1. Take a subset of a public dataset (population), repeatedly draw samples from it.

    2. For each sample, construct partially synthetic datasets using different synthesizer techniques.

    3. Use inferential methods to determine point estimates and 95% confidence intervals for 162 estimands of multiple analyses (linear regressions, logistic regressions etc.) and evaluate disclosure risk, for ten replications.

- Data - public use microdata sample from the 2002 Uganda Population and Housing Census

    - 10% systemic sample of Uganda's natives

    - ~2.5 million questionnaire records

    - >100 variables

Steps:

1. From population repeatedly draw 1% simple random samples – this is the original data from which synthetic datasets should be generated

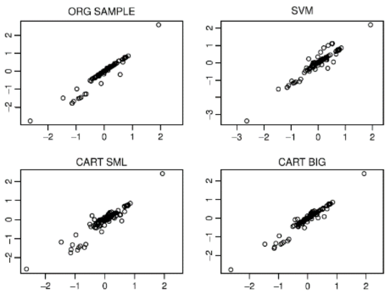
2. Variables chosen: Five - Number of persons in the household, age, marital status, literacy, employment status. Synthetic data generation by replacing all records' values for these variables; all other variables are left unchanged

3. For each drawn observed dataset, synthesize the five variables using the steps for multivariate synthesis in the order: persons, age, marital status, literacy, and employment status

4. For each synthesizer, generate m = 5 partially synthetic datasets

**<u>Results:</u>**

CART shows the best results among all ML Models. Below is the comparison of the True Population Quantities and Average of the point estimates, across the 1000 simulation runs for the original and synthetic datasets

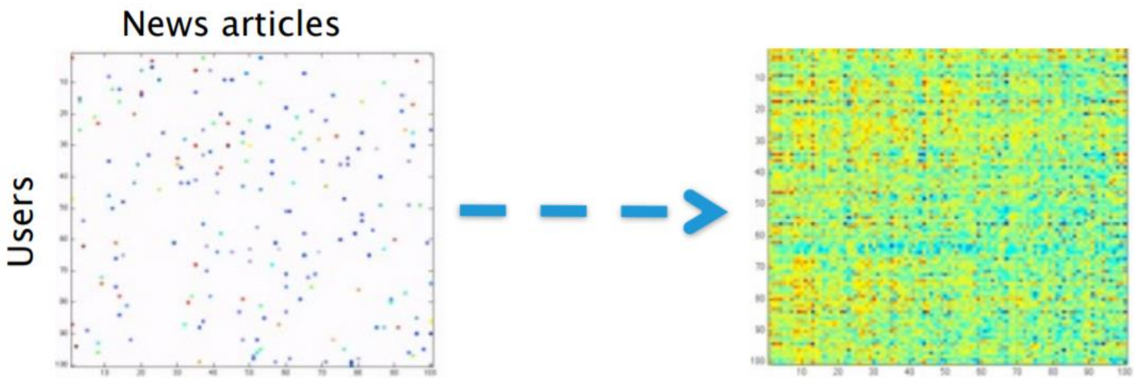


## Discussion

- Empirical evaluations show that it is possible to obtain synthetic datasets that provide reliable estimates paired with low disclosure risk by using nonparametric synthesizers.
- Authors achieved results without any tuning of the different synthesizers (Better results are obtainable if the methods are tailored to the data at hand)
- SVM and CART synthesizers outperform the bagging and random forests synthesizers in terms of analytical validity, albeit for the price of an increased risk of identification disclosure
- SVMs are promising, but implementing is difficult (sensitive to tuning, and it is not obvious which variation should be used for synthesis.)
- CART balances analytical validity and disclosure risks. With appropriate d, it can provide a high level of data utility for potentially acceptable disclosure risks.

## SECTION III: SOFT-IMPUTE

Motive: In many applications measured data can be represented in a matrix $X_{m*n}$ for which only a relatively small number of entries are observed. The task - “Complete” the matrix based on the observed entries. For e.g. a recommender system as below:



Idea:

- Problem rephrased as learning an unknown parameter - a matrix $Z_{m*n}$ with very high dimensionality based on very few observations $(X_{m*n})$
- For inference assume that the parameter $Z$ lies in a much lower dimensional manifold - i.e. assume that $Z$ can be well represented by a matrix of low rank

$$\mathrm{Z} \approx V_{m*k}\; G_{k*n}$$

Here $k \ll \min\,(n, m)$

**Formulation:**

For a matrix $X_{m*n}$let $\Omega \subset \{1, \ldots, m\} \times \{1, \ldots, n\}$ denote the indices of observed entries

Now the optimization problem:

$$\begin{array}{ll} \text{minimize} & \operatorname{rank}(Z) \\ \text{subject to} & \sum\limits_{(i,j)\in\Omega} (X_{ij} - Z_{ij})^2 \leq \delta. \end{array}$$

Such that $\partial \geq 0$ is a regularization parameter controlling the tolerance in training error. Rank constraint makes this problem for general Ω combinatorically hard, and to make this easier we need a modification.

The above can be reformulate in Lagrange form:

$$\underset{Z}{\text{minimize}} \frac{1}{2} \sum_{(i,j)\in\Omega} (X_{ij} - Z_{ij})^2 + \lambda \|Z\|_*$$

## Convergence Analysis

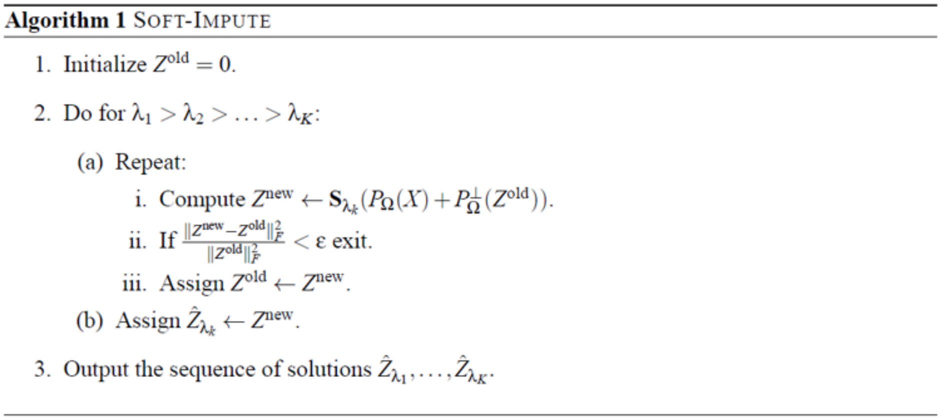
- Unlike generic first-order methods (Nesterov, 2003) including competitive first-order methods for nuclear norm regularization, Soft-Impute does not involve the choice of any additional step-size.
- Algorithm is readily scalable for solving large scale problems.
- Sequence $Z_\lambda^k$ generated via Soft Impute converges asymptotically to a minimizer of the objective function, i.e. , as $k$ ⇥ $\infty$
- Proofs are complex and are based on trace inequalities

## Computational Complexity

- The computationally demanding part of algorithm is in $S_\lambda\ (P_\lambda(X) +\ P_\lambda^{\perp}(Z_\lambda^k))$
- The above requires calculating a low-rank SVD of a matrix, since the underlying model assumption is that rank($Z$) $\ll$ min{$m$, $n$}
- There are very efficient direct matrix factorization methods for calculating the SVD of matrices of moderate size (at most a few thousand in dimensions), from numerical linear algebra literature.
- For large matrices one must resort to indirect iterative methods for calculating the leading singular vectors/values of a matrix
- Authors use PROPACK algorithm (Larsen, 2004, 1998) because of (1) Its low storage requirements, (2) effective flop count (3) structure of the problem which involves sparse matrices and (4) its well documented MATLAB version

## Algorithm for Nuclear Norm Regularization

$$\underset{Z}{\text{minimize}} \quad f_\lambda(Z) := \frac{1}{2}\|P_\Omega(X) - P_\Omega(Z)\|_F^2 + \lambda\|Z\|_*$$



**Algorithm 1** Soft-Impute

```
1. Initialize $Z^{\text{old}} = 0$.
2. Do for $\lambda_1 > \lambda_2 > \ldots > \lambda_K$:
    (a) Repeat:
        i. Compute $Z^{\text{new}} \leftarrow \mathbf{S}_{\lambda_k}(P_\Omega(X) + P_\Omega^\perp(Z^{\text{old}}))$.
        ii. If $\frac{\|Z^{\text{new}} - Z^{\text{old}}\|_F^2}{\|Z^{\text{old}}\|_F^2} < \varepsilon$ exit.
        iii. Assign $Z^{\text{old}} \leftarrow Z^{\text{new}}$.
    (b) Assign $\hat{Z}_{\lambda_k} \leftarrow Z^{\text{new}}$.
3. Output the sequence of solutions $\hat{Z}_{\lambda_1}, \ldots, \hat{Z}_{\lambda_K}$.
```

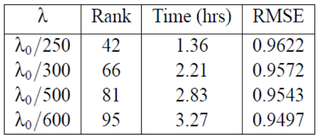
## Simulation Study: Application on Netflix Data

Netflix Prize

- The Netflix training data consists of the ratings of 17,770 movies by 480,189 Netflix customers. Resulting data matrix is extremely sparse, with 100,480,507 or 1% of the entries observed.
- The task was to predict the unseen ratings for a qualifying set and a test set of about 1.4 million ratings each with the true ratings in these data sets held in secret by Netflix.
- Netflix's own algorithm has an RMSE of 0.9525, and the contest goal was to improve this by 10%, or an RMSE of 0.8572 or better.

## Results

- A maximum of 10 iterations were performed for each of the examples
- The results are not competitive with those of the competition leaders, but demonstrate the feasibility of applying SOFT-IMPUTE to such a large data set



| $\lambda$ | Rank | Time (hrs) | RMSE |
|---|---|---|---|
| $\lambda_0/250$ | 42 | 1.36 | 0.9622 |
| $\lambda_0/300$ | 66 | 2.21 | 0.9572 |
| $\lambda_0/500$ | 81 | 2.83 | 0.9543 |
| $\lambda_0/600$ | 95 | 3.27 | 0.9497 |

Results of applying SOFT-IMPUTE to the Netflix data. $\lambda_0 = ||P_\Omega(X)||_2$

## SECTION IV: MIDA

Motive: Missing Data is an important problem and the current imputation models have the following issues:

- Incapability of handling mixed data types (categorical and continuous)
- Strict distributional assumptions
- Often cannot handle arbitrary missing data patterns
- The existing models capable of overcoming issues are limited in:
    - Ability to model highly nonlinear relationships
    - Deal with high volume data and complex interactions while preserving inter-variable dependencies

Idea: Recent advancements in Deep Learning have established state-of-the-art results in many fields (CV, Speech Recognition, Motion Planning) because of:

- Deep architectures have the capability to automatically learn latent representations and complex inter-variable associations, which is not possible using classical models.
- Denoising Autoencoders (DAEs), part of the deep learning framework, are designed to recover clean output from noisy input.
- Missing data is a special case of noisy input, making DAEs ideal as an imputation model.

### Autoencoders

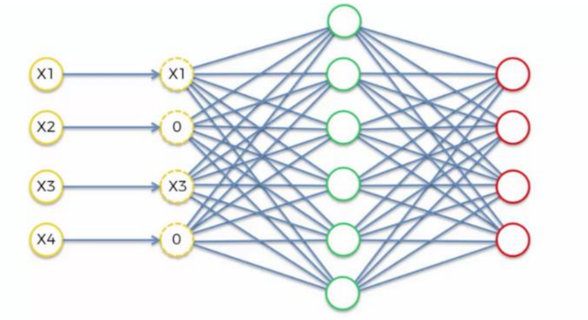
- An Autoencoder takes an input $\boldsymbol{x} \in [0,1]^{d}$ and maps (encodes) it to an intermediate representation **h** $\in [0,1]^{d'}$using an encoder, where **d'** represents a different dimensional subspace.
- The encoded representation is then decoded back to the original **d** dimensional space using a decoder.
- Encoder and decoder are both Artificial Neural Networks, as follows, where **z** is the decoded result and **s** is any nonlinear function, reconstruction error between **x** and **z** is minimized during training phase.

$$h = s(Wx + b)$$

$$z = s(W'h + b')$$

## Denoising Autoencoders (DAEs)

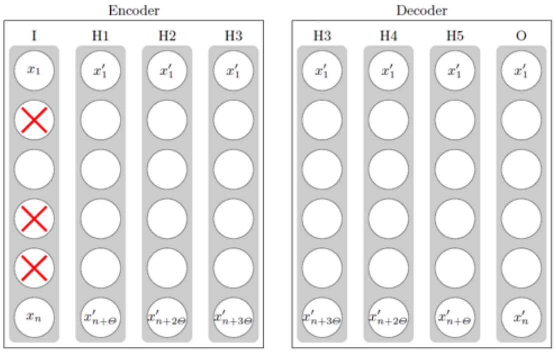
- Denoising autoencoders are a natural extension to autoencoders.
- Approach is to corrupt the input data and force the network to reconstruct the clean output, by making the hidden layers learn robust features.
- Corruption can be applied in different ways, such as randomly setting some input values to zero or using distributional additive noise.



## MIDA Architecture:

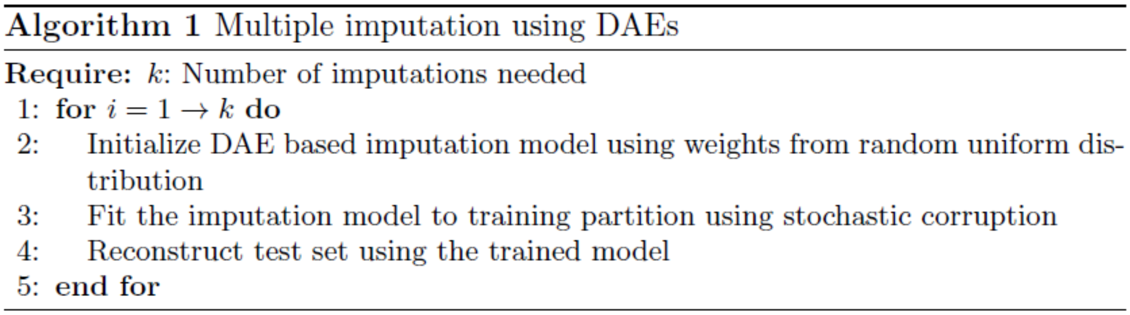
- Employ atypical overcomplete representation of DAEs (i.e. more units in successive hidden layers during encoding phase compared to the input layer)
- Overcomplete DAEs create representations capable of adding lateral connections, aiding in data recovery.
- Start with an initial **n** dimensional input, then at each successive hidden layer, add nodes, increasing the dimensionality to n+Θ
- For experiments, used Θ = 7 (hyperparameter)
- Use inputs standardized between 0 and 1 to facilitate faster convergence for small to moderate sample sizes.

The architecture for Θ hidden layers can be represented as below:



## Usage and Algorithm:

- Initialize the respective column average in case of continuous variables and most frequent label in case of categorical variables as placeholders.
- Training phase initiated with a stochastic corruption process setting random inputs to zero, and the model learns to map corrupted input to clean output.
- Assumption: There is enough complete data to train the model, so the model learns to recover true data using stochastic corruption on inputs, and is not learning to map placeholders as valid imputations.



**Algorithm 1** Multiple imputation using DAEs

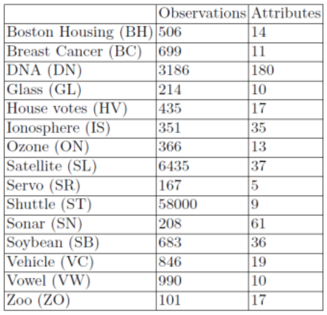
```
Require: $k$: Number of imputations needed
for $i = 1 \to k$ do
    Initialize DAE based imputation model using weights from random uniform distribution
    Fit the imputation model to training partition using stochastic corruption
    Reconstruct test set using the trained model
end for
```

## Simulation Study: Comparative Study with MICE

Competitor - Current state-of-the-art in multiple imputation is Multivariate Imputation by Chained Equations (MICE)

- Fully conditional specification approach which species multivariate model, on variable by variable basis using a set of conditional densities, one for each variable with missing data.
- Draws imputations by iterating over conditional densities.
- Has an added advantage of being able to model different densities for different variables?

Datasets used were the following:



| | Observations | Attributes |
|---|---|---|
| Boston Housing (BH) | 506 | 14 |
| Breast Cancer (BC) | 699 | 11 |
| DNA (DN) | 3186 | 180 |
| Glass (GL) | 214 | 10 |
| House votes (HV) | 435 | 17 |
| Ionosphere (IS) | 351 | 35 |
| Ozone (ON) | 366 | 13 |
| Satellite (SL) | 6435 | 37 |
| Servo (SR) | 167 | 5 |
| Shuttle (ST) | 58000 | 9 |
| Sonar (SN) | 208 | 61 |
| Soybean (SB) | 683 | 36 |
| Vehicle (VC) | 846 | 19 |
| Vowel (VW) | 990 | 10 |
| Zoo (ZO) | 101 | 17 |

**Results:**

- The MIDA model outperforms MICE in 100% of cases with data MCAR and MNAR with *uniform missing pattern*, and in > 70% of cases with *random missing pattern*.

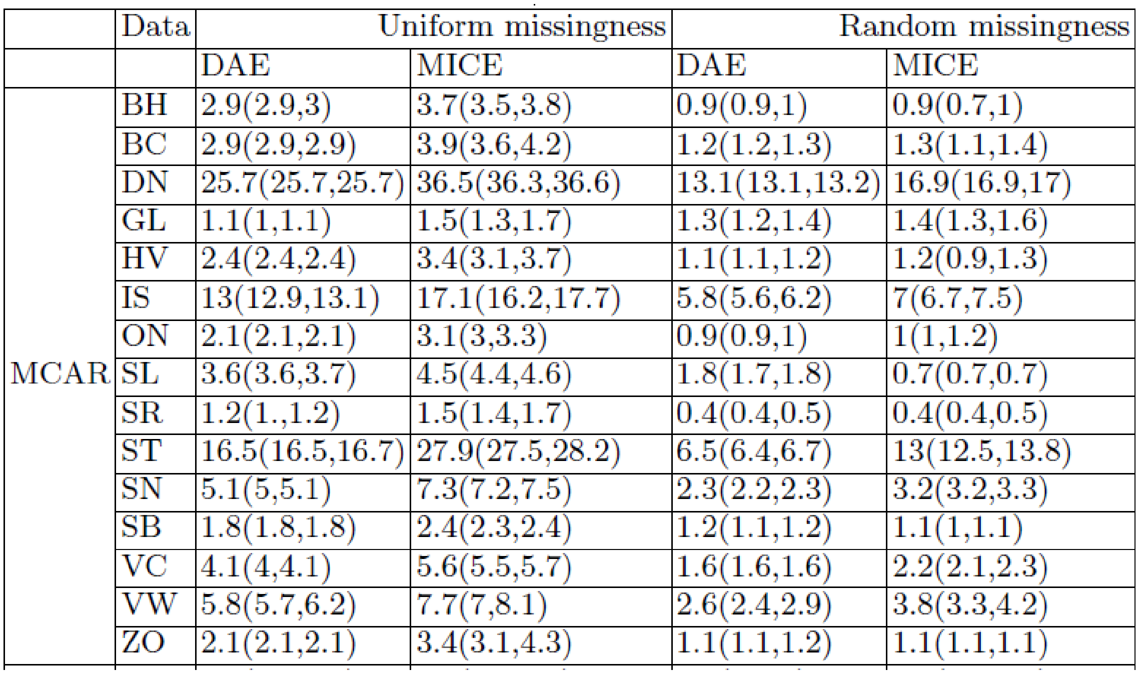
- Model's superior performance in small to moderate dataset sizes with constrained dimensionality, is indicative of its utility when datasets are large and are of higher dimensionality (bottleneck for other multiple imputation models)

- Model does not need a certain proportion of available data to predict missing value.

- Computational cost associated with MIDA is at par or better than imputations based on MICE for small to moderate sized datasets.

- Computational gains are significant when modelling a complete dataset in a single attempt compared to iterative variable by variable imputation in MICE.

The following table summarizes the above:



|  | Data | Uniform missingness |  | Random missingness |  |
|---|---|---|---|---|---|
|  |  | DAE | MICE | DAE | MICE |
| MCAR | BH | 2.9(2.9,3) | 3.7(3.5,3.8) | 0.9(0.9,1) | 0.9(0.7,1) |
|  | BC | 2.9(2.9,2.9) | 3.9(3.6,4.2) | 1.2(1.2,1.3) | 1.3(1.1,1.4) |
|  | DN | 25.7(25.7,25.7) | 36.5(36.3,36.6) | 13.1(13.1,13.2) | 16.9(16.9,17) |
|  | GL | 1.1(1,1.1) | 1.5(1.3,1.7) | 1.3(1.2,1.4) | 1.4(1.3,1.6) |
|  | HV | 2.4(2.4,2.4) | 3.4(3.1,3.7) | 1.1(1.1,1.2) | 1.2(0.9,1.3) |
|  | IS | 13(12.9,13.1) | 17.1(16.2,17.7) | 5.8(5.6,6.2) | 7(6.7,7.5) |
|  | ON | 2.1(2.1,2.1) | 3.1(3,3.3) | 0.9(0.9,1) | 1(1,1.2) |
|  | SL | 3.6(3.6,3.7) | 4.5(4.4,4.6) | 1.8(1.7,1.8) | 0.7(0.7,0.7) |
|  | SR | 1.2(1.,1.2) | 1.5(1.4,1.7) | 0.4(0.4,0.5) | 0.4(0.4,0.5) |
|  | ST | 16.5(16.5,16.7) | 27.9(27.5,28.2) | 6.5(6.4,6.7) | 13(12.5,13.8) |
|  | SN | 5.1(5,5.1) | 7.3(7.2,7.5) | 2.3(2.2,2.3) | 3.2(3.2,3.3) |
|  | SB | 1.8(1.8,1.8) | 2.4(2.3,2.4) | 1.2(1.1,1.2) | 1.1(1,1.1) |
|  | VC | 4.1(4,4.1) | 5.6(5.5,5.7) | 1.6(1.6,1.6) | 2.2(2.1,2.3) |
|  | VW | 5.8(5.7,6.2) | 7.7(7,8.1) | 2.6(2.4,2.9) | 3.8(3.3,4.2) |
|  | ZO | 2.1(2.1,2.1) | 3.4(3.1,4.3) | 1.1(1.1,1.2) | 1.1(1.1,1.1) |

Values in parenthesis: mean (min, max) (RMSE Sum)

**Discussion:**

Positives

- Looks very promising, as results show better performance than MICE.

- Implementation will be an interesting challenge
- Can borrow some ideas from other Deep Learning approaches to increase training speed and accuracy and can test on different datasets.

Negatives

- Interpretability is low, since it is a DL model.
- Might have to one-hot-encode categorical variables!
- How to deal with mixed variables?

## Conclusion and Plan for SRIP

The approach involving MIDA, a Deep Learning solution using Denoising Autoencoders has been chosen after a detailed literature review. The results from the paper by Gondara and Wang (2018) show better performance than the current state of the art methods and hence a general-purpose Python implementation of the same can be a useful addition to the Missing Data Imputation algorithms.

**Future Work**: Implement an open source Python package for general purpose Missing Data Imputation using MIDA as part of Summer Research Internship Program.

## References

- *Raghunathan, Trivellore E., et al. "A multivariate technique for multiply imputing missing values using a sequence of regression models." Survey methodology 27.1 (2001): 85-96.*
- *Rubin, Donald B. "Multiple imputations in sample surveys-a phenomenological Bayesian approach to nonresponse." Proceedings of the survey research methods section of the American Statistical Association. Vol. 1. American Statistical Association, 1978.*
- *Hokayem, Charles, Trivellore Raghunathan, and Jonathan Rothbaum. "Sequential Regression Multivariate Imputation in the Current Population Survey Annual Social and Economic Supplement." Proceedings of the Joint Statistical Meeting. 2015.*
- *Reiter, Jerome P., and Robin Mitra. "Estimating risks of identification disclosure in partially synthetic data." Journal of Privacy and Confidentiality 1.1 (2009)*
- *Reiter, Jerome P. "Using CART to generate partially synthetic public use microdata." Journal of Official Statistics 21.3 (2005): 441.*

- *Rubin, Donald B. "The bayesian bootstrap." The annals of statistics (1981): 130-134.*
- *Raghunathan, Trivellore E., et al. "A multivariate technique for multiply imputing missing values using a sequence of regression models." Survey methodology 27.1 (2001): 85-96.*
- *Rubin, Donald B. "Multiple imputations in sample surveys-a phenomenological Bayesian approach to nonresponse." Proceedings of the survey research methods section of the American Statistical Association. Vol. 1. American Statistical Association, 1978.*
- *Wu, Ting-Fan, Chih-Jen Lin, and Ruby C. Weng. "Probability estimates for multi-class classification by pairwise coupling." Journal of Machine Learning Research 5.Aug (2004): 975-1005.*
- *Mazumder, Rahul, Trevor Hastie, and Robert Tibshirani. "Spectral regularization algorithms for learning large incomplete matrices." (2010)*
- *J. Cai, E. J. Candes, and Z. Shen. "A singular value thresholding algorithm for matrix completion."(2008)*
- *O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman. "Missing value estimation methods for DNA microarrays."(2001)*
- *Gondara, Lovedeep, and Ke Wang. "MIDA: Multiple imputation using denoising autoencoders." Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, Cham, 2018.*
- *Yoshua Bengio, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In Advances in Neural Information Processing Systems, pages 899{907, 2013}*
- R implementation of MIDA (Gondara)
- https://github.com/Oracen/MIDAS
- https://towardsdatascience.com/denoising-autoencoders-explained-dbb82467fc2
- http://sun.stanford.edu/~rmunk/PROPACK/paper.pdf
- https://en.wikipedia.org/wiki/Netflix_Prize
- https://pypi.org/project/fancyimpute/
- https://simplyml.com/benchmarking-the-singular-value-decomposition/