PC method to impute missing values for mixed data

Release 2024

Ambre Adjevi and Laura Fuentes

Mar 23, 2024

CONTENTS:

CHAPTER

ONE

ALGORITHMS MODULE

class algorithms.**FAMD**(*data*, *k1*, *k2*, *nb_values_per_cat*, *n_components=2*)

Bases: object

DM()

Function to define D and M according to current values

data_concat()

Redefines the whole dataset (df) after updating df_C0 (continuous variables) and df_categ (categorical variables) with new imputation

ponderation_gsbs()

Weighting step specialized in the gsbs dataset: This function updates: - the standard deviation values (sj) for continous variables - proportion for categorical variables (pj) according to a new imputation update.

run_famd(*verbose=False*)

Function that runs the whole FAMD algorithm

Returns
reconstructed version of (self.XD_moins_sqrt - self.M) according to self.n_components

Return type
Z_p (pd.DataFrame)

step3()

Performs the third step from FAMD algorithm: - Update D, M - Computes the SVD on self.XD_moins_sqrt - self.M

Returns
reconstructed version of (self.XD_moins_sqrt - self.M) according to self.n_components

Return type
Z_p (pd.DataFrame)

class algorithms.**IterativeFAMDImputer**(*data*, *k1*, *k2*, *nb_values_per_cat*, *n_components=2*)

Bases: *FAMD*

impute(*n_it*, *tol=0.0001*, *verbose=False*)

Runs the whole imputation process

Parameters

- **n_it** (*int*) – number of iterations to run the algorithm
- **tol** (*float, optional*) – Threshold to define early stopping criteria. Defaults to 1e-4.
- **verbose** (*bool, optional*) – Defaults to False.

Returns
Imputed dataset after algorithm convergence

Return type
self.df (pd.DataFrame)

`inital_impute()`

Initial imputation: - continuous variables: mean/variable - categories: proportion/category

`ponderation_gsbs()`

Weighting function specialized on the gsbs dataset This function updates: - the standard deviation values (sj) for continous variables, - proportion for categorical variables (pj) according to a new imputation update.

CHAPTER

TWO

UTILS MODULE

utils.**compute_metrics**(*df*, *cat_idx*, *n_it*, *n_components*, *proba_non_missing*)

Computes the falsely classified rate and nrmse over a synthetic dataset for different probabilities of missingness.

Parameters

- **df** (*pd.DataFrame*) – dataframe to impute
- **n_it** (*int*) – maximum number of iterations for iFAMD convergence
- **n_components** (*int*) – number of principal components for reconstruction
- **proba_non_missing** (*list of float*) – List of probabilities that a value is not missing

Returns

Masked dataframe hence containing missing values

Return type

data_missing_raw (pd.DataFrame)

utils.**create_dataset**(*n*, *S*, *K*, *cat*, *cat_idx*, *nb_of_cat_per_var*, *SNR=1.0*)

Generate dataset with continuous (float variables) and categorical variables (string variables) based on parameters. :param n: Sample size. :type n: int :param S: Underlying dimension. :type S: int :param K: K[s] = number of times the variable s (s in {1,...,S}) is duplicated in the dataset :type K: list of ints :param cat: Number of categorical variables :type cat: int :param cat_idx: Indexes of the categorical variables :type cat_idx: list of ints :param nb_of_cat_per_var: Number of categories for each categorical variable :type nb_of_cat_per_var: list of ints :param SNR: Signal to Noise Ratio :type SNR: float

Returns

Generated dataset

Return type

df (pd.Dataframe)

utils.**create_missingness**(*df*, *proba_non_missing*)

Returns dataframe df with missing values

Parameters

- **df** (*pd.DataFrame*) – Dataframe to mask
- **proba_non_missing** (*float*) – probability that a value is not missing

Returns

masked dataframe hence containing missing values

Return type

data_missing_raw (pd.DataFrame)

utils.**create_rare_df**(*f*, *n*, *S=1*, *K=[4]*, *SNR=5*)

Create dataset with rare categories, as described in section 3.3 of the paper

Parameters

- **f** (*float*) – frequency of the rare categories
- **n** (*int*) – sample size
- **S** (*int*) – Underlying dimension.
- **K** (*list of ints*) – K[s] = number of times the variable s (s in {1,...,S}) is duplicated in the dataset
- **SNR** (*float*) – Signal to Noise Ratio

Returns
Generated dataset with rare categories

Return type
df_rare (pd.Dataframe)

utils.**encode_dummy_variables**(*df*, *cat_var_idx*)

Encode dummy variables, returns the updated dataframe, the list of the dummy variables' names and the list of the number of different values in each categories Format of the dummy variable names : name_of_variable + _ + variable_value

Args :
df (pd.DataFrame) cat_var (pd.Index): index of the variables to encode into dummy variables

Returns
encoded dummy variables dummy_var_idx (pd.Index): list of the dummy variables' names nb_values_per_cat (list): number of different values in each categories

Return type
df_dummy (pd.DataFrame)

utils.**generate_random**(*column*, *missing_indices_len*)

Generate a random number between the minimum and maximum of a given variable

Parameters
column (*pd.Series*) – Column values

Returns
np.random.randint(low=column.min(), high=column.max())

Return type
random value (int)

utils.**random_imputation**(*df*)

Perform random imputation with random values per column

Parameters
df (*pd.DataFrame*) – _description_

Returns
Imputed dataset

Return type
df_modif (pd.DataFrame)

CHAPTER

THREE

METRICS_FAMD MODULE

metrics_FAMD.**compute_nrmse_weighted**(*dfpred*, *dftrue*)

Computes de Normalized Root Mean Squared Error (NRMSE)

Parameters

- **dfpred** (*pd.DataFrame*) – dataframe gathering for continuous variables
- **dftrue** (*pd.DataFrame*) – ground truth dataframe gathering for continuous variables

Returns
NRMSE value for each variable

Return type
nrmse (float)

metrics_FAMD.**metric_fc**(*df_categ*, *true_df_categ*)

Computes the proportion of falsely classified individuals per category.

Parameters

- **df_categ** (*pd.DataFrame*) – dataframe gathering only categorical variables
- **true_df_categ** (*pd.DataFrame*) – ground truth dataframe gathering for categorical variables

Returns
rate of falsely classified individuals per category

Return type
fc (float)

CHAPTER

FOUR

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

INDEX