Overview

This package implements Expectation-Maximization (EM) and Monte Carlo EM (MCEM) algorithms for missing data imputation under a multivariate and Poisson assumption.

The MCEM is available under multivariate assumption only, which allows Monte Carlo sampling when the analytical expectation in the E-step becomes intractable.

Installation instruction.

Step	Action	Code
1	Install required library	**library("ggplot2")**
2	Install the package from GITHUB.	remotes::install_github("ngamihimihi/DATA501_Project", subdir = "DATA501Package", build_vignettes = TRUE, INSTALL_opts = c("--install-tests"))
3	Load required package for testing.	library(“DATA501Package”)
4	Run available test.	testthat::test_package("DATA501Package", "tap")

Test plan

Objective

To test and validate the functionality of an R package that implements the EM and MCEM algorithms for missing data imputation under the multivariate normal and poisson assumption. Please note that MCEM algorithms is only avaialble for multivariate assumption.

Test scope

Function ready for testing.

- e_step_nvnorm_em
- e_step_nvnorm_mcem
- e_step_poisson_em
- em_engine (available distribution: poisson and mvnorm)

- initialise_parameters
- initialise_parameters_poisson
- initilise_parameters_mvnorm
- log_likelihood_nvnorm
- log_likelihood_poisson
- m_step_nvnorm
- m_step_poisson
- run_em_algorithm

Method ready for testing:

- plot(result,what="loglik")
- summary(result)

Test Categories and Cases

a. Initialization functions.

- Functions: initialize_parameters_nvnorm()

Test ID	Desciption	Expected outcome
1.1	Input valid numeric matrix with some NAs	Returns valid mu and sigma, no error
1.2	Input with a column entirely NA	Throws error
1.3	Input with a row entirely NA	Issues warning, continues
1.4	Covariance matrix not PD	Issues message, applies jitter
1.5	Invalid input (e,g.: dataframe or character matrix)	Throws error

b. E-step functions

e_step_nvnorm_em(), e_step_nvnorm_mc(),e_step_poisson_em()

Test ID	Applicable function	Desciption	Expected outcome
2.1	All three	Inputed matrix matches the shape of input	Dimension equal
1.2	All three	Observed values remain unchanged	Same as input
1.3	All three	Missing values are imputed	NA replaced
1.4	e_step_nvnorm_mc()	For MC version,accept_rate attribute exists	Attribute exists and is numeric

c. Log-likelihood function.

- Function: log_likelihood_nvnorm(),log_likelihood_poisson()

Test ID	Applicable functions	Description	Expected outcome
3.1	Both functions	Expected output data type	Output is of double type and is a finite value
3.2	Both functions	No error thrown with correct input	No error
3.3	log_likelihood_nvnorm()	Function is able to handle single row data	Correct data type output and no error thrown
3.4	log_likelihood_poisson()	Function is able to handles NA values	Loglikelihood is calculated successfully
3.5	log_likelihood_poisson()	Function throw errors with mismatched lambda and input data length e.g: lambda vector has length 3 while data matrix has 2 columns.	Error throw
3.6	log_likelihood_poisson()	Function returns NA for invalid lambda	Output returns is NA
3.7	log_likelihood_poisson()	Function returns NA for invalid data values	Output returns is NA
3.8	log_likelihood_poisson	Function works for edge case zeros	Output is calculated successfully

d. Main Engine and EM Algorithm

Function: em_engine(), run_em_algorithm()

Test ID	Applicable function	Description	Expected Outcome
4.1	Both	Run without error for valid data	Return updated em_model
4.2	Both	parameter_history length == number of iterations	Confirmed

4.3	Both	Early stopping triggers	Stop before max_iter if tolerance met
4.4	Both	MCEM accept Monte Carlo parameters	Return updated em_model;

Instruction to test submission.

1. Install dependencies.

Make sure all the following packages are installed:

install.packages(c("devtools", "testthat", "rmarkdown", "knitr"))

2. Install the package.

Install package from GITHUB

3. Run all unit tests:

Current unit tests are prepared for the 2 main object and function.

To run the unit test:

devtools::test()

4. Use Test data

Dependency: dplyr, data needs to be converted to matrix before passing on to run_em_algorithm

Code to import and test:

data<-read.csv("kc_house_data.csv",skip=1,header = FALSE)

head(data,5)

data<-data[,-c(1,2)]

data <- as.matrix(data)

model <- em_model(data,distribution = "nvnorm",method = "EM")

model_em <- em_model(data,distribution = "nvnorm",method = "EM")

model_mcem<- em_model(data,distribution = "nvnorm",method = "EM")

#View result

#Standard EM

```
model_em$data
model_em$method
model_em$early_stop
model_em$loglik_history
model_em$distribution
model_em$parameters
model_em$parameter_history
head(model_em$imputed,5)
#Monte Carlo EM
model_mcem$data
model_mcem$method
model_mcem$early_stop
model_mcem$loglik_history
model_mcem$distribution
model_mcem$parameters
model_mcem$parameter_history
head(model _mcem$imputed,5)
```