mixedgcImp Vignette

Yuxuan Zhao

11/16/2019

Introduction

The package *mixedgcImp* is for imputing missing values of continuous and ordinal mixed dataset. The imputation is done by fitting a Gaussian copula model to the incomplete mixed observation. More details can be found in our paper Zhao, Y., & Udell, M. (2019). In this vignette, we reproduce a simulation setting used in our paper.

Simulate mixed type data observation

We first generate rows of $\mathbf{Z} \in \mathbb{R}^{n\times p}$ as $\mathbf{z}^1, \ldots, \mathbf{z}^n \overset{i.i.d.}{\sim} \mathcal{N}(\mathbf{0}, \Sigma)$, then generate $\mathbf{X} = \mathbf{f}(\mathbf{Z})$ through coordinate-wise monotone functions $\mathbf{f}$. $\mathbf{X}$ consists of n observations and p variables. We let $p = 15$ and $n = 2000$ and randomly generate Σ. Use $\mathbf{f}$ such that $\mathbf{X}_1, \ldots, \mathbf{X}_5$ have exponential distributions with rate parameter 1/3, $\mathbf{X}_6, \ldots, \mathbf{X}_{10}$ are binary and $\mathbf{X}_{11}, \ldots, \mathbf{X}_{15}$ are ordinal with 5 levels. Then we randomly remove 50% of the entries of $\mathbf{X}$.

```r
library(mixedgcImp)
library(MASS)
set.seed(424)
n = 2000
p = 15
Sigma = matrix(rnorm(p*p), nrow = p)
Sigma = cov2cor(Sigma %*% t(Sigma)) # generate random correlation matrix
Z = mvrnorm(n, mu = rep(0,p), Sigma = Sigma) # generate copula observation
X = Z # generate observation with specified marginals
# five continuous columns with exponential marginal
X[,1:5] = qexp(pnorm(Z[,1:5]), rate = 1/3)
# five binary columns
for (i in 6:10) X[,i] = continuous2ordinal(Z[,i], k = 2)
# five ordinal columns with five levels
for (i in 11:15) X[,i] = continuous2ordinal(Z[,i], k = 5)
 # mask 50% of the original observation
X_obs = X
loc = sample(1:prod(n*p), size = floor(prod(n*p)*0.5))
X_obs[loc] = NA
```

Fit Gaussian copula model from incomplete mixed observation

Implementation

Implementing our algorithm is easy, since it does not involve any tuning parameter. Just pass your observation into function *impute_mixedgc*.

```r
fit = impute_mixedgc(X_obs)# takes around 20 secs
```

Imputation evaluation

We then evaluate the imputation error for each data type using the scaled mean absolute error (SMAE) proposed in our paper to measure the imputation error on columns in I:

$$\text{SMAE} := \frac{1}{|I|}\sum_{j\in I}\frac{||\hat{\mathbf{X}}_j - \mathbf{X}_j||_1}{||\mathbf{X}_j^{\text{med}} - \mathbf{X}_j||_1}$$

where $\hat{\mathbf{X}}_j, \mathbf{X}_j^{\text{med}}$ are the imputed values and observed median for j-th column, respectively. For each data type, we compute the SMAE on corresponding columns. The estimator's SMAE is smaller than 1 if it outperforms column median imputation. We now compute the SMAE for each data type.

```
err_imp = cal_mae_scaled(xhat = fit$Ximp, xobs = X_obs, xtrue = X) # SMAE for each column
mean(err_imp[1:5]) # SMAE across continuous columns
```

```
## [1] 0.7902322
```

```
mean(err_imp[6:10]) # SMAE across binary columns
```

```
## [1] 0.7614535
```

```
mean(err_imp[11:15]) # SMAE across ordinal columns
```

```
## [1] 0.8108516
```

Fitted copula correlation matrix

You can also extract the fitted copula correlation matrix to describe the correlation relationship among columns. To evaluate the estimated correlation, we use relative error $||\hat{\Sigma} - \Sigma||_F/||\Sigma||_F$, where $\hat{\Sigma}$ is the estimated correlation matrix.

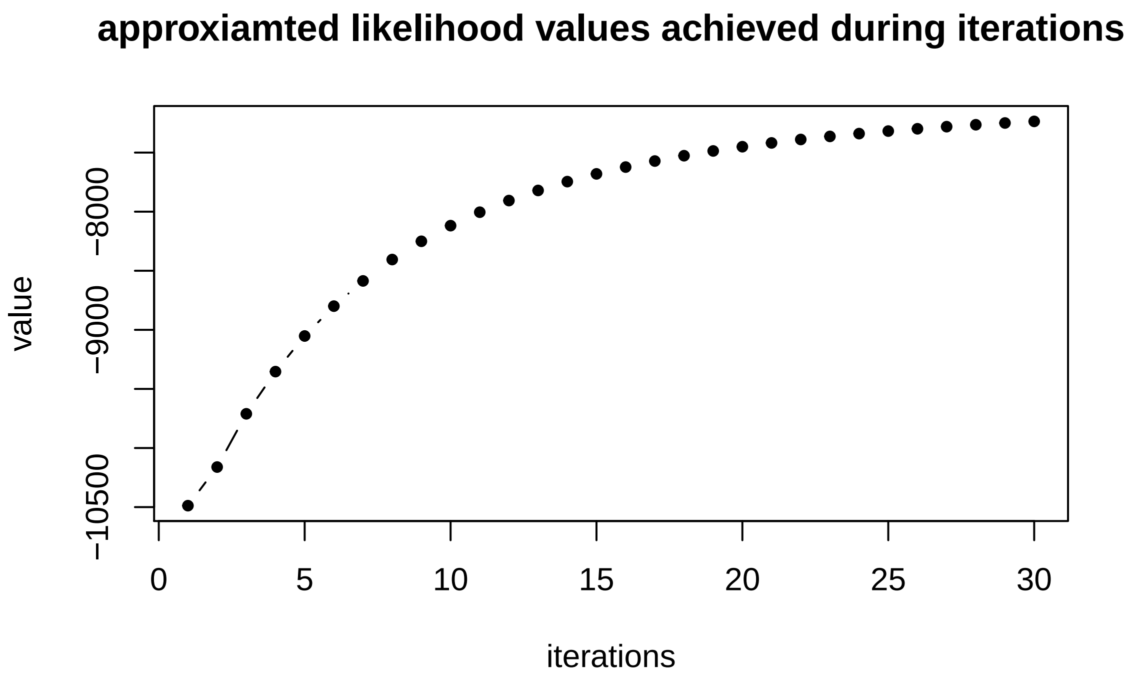
```
# relative Frobeinus error of imputed correlation matrix
norm(fit$R - Sigma, type = 'F')/norm(fit$R, type = 'F')
```

```
## [1] 0.1758019
```

Monitor the convergence

The likelihood function or objective function we are maximizing is hard to evaluate. We provide an approximation of the likelihood evaluation during iterations, which can be used to monitor the fitting process. If a Gaussian copula model is well fitted to the data, we expect to see monotonically increasing likelihood values during iterations.

```
plot(fit$loglik, ylab = 'value', type = 'b', pch = 20,
     xlab = 'iterations', main = 'approxiamted likelihood values achieved during iterations')
```



In empirical study, we find early stop can help improve the imputation performace sometimes, since it can help avoid overfitting. Our suggestion is to first implement with default tolerence threshold $1e-3$, which usually returns good performance in our experiments. With fitted model, one can check the provided likelihood values during iterations. If the likelihood values increase slowly for many iterations at last, we recommand fit again with larger tolerance threshold. In the future, we will add explicit regularization to our algorithm to avoid overfitting.