STAT 501 Project 2
PCA for Discrete Distribution with Gaussian Copula

Yifan Zhu

May 7, 2018

Abstract

In the report, we extend the semiparametric scale-invariant PCA method called copula PCA proposed by Han and Liu (2014) to the case when our distribution is descrete. The senators' voting data set was used to apply this method.

1 Introduction

In this project, our goal is to reduce the dimension of catagorical data with PCA. However, we cannot perform PCA directly on the catagorical data, since the sample space only has finite points and we cannot find a proper set of basis (or principle components) for this space like what we did for continuous distribution. Hence we consider the catagorical data with descrete distribution are generated from some latent variables with continuous distribution. Then instead of performing PCA on the observed data, we perform PCA on the latent variables. We also want to transfrom or connect the latent variables to multivariable normal r.v.s. Since for dimension reduction, PCA is an optimal dimension reduction method for multivariate normal data.

2 Method of copula PCA and extension to discrete case

2.1 nonparanormal distribution

Definition 2.1 (nonparanormal distribution, Han and Liu (2014))**.** Let $f^0 = \{f_j^0\}_{j=1}^d$ be a set of strictly increasing univariate functions. We say that a d dimensional random vector $\boldsymbol{X} = (X_1, \ldots, X_d)^T$ follows a nonparanormal distribution $\mathrm{NPN}_d(\boldsymbol{\Sigma}_0, f^0)$, if

$$f^0(\boldsymbol{X}) := (f_1^0(X_1), \ldots, f_d^0(X_d))^T \sim \mathrm{N}_d(\mathbf{0}, \boldsymbol{\Sigma}^0)$$

where $\mathrm{diag}(\boldsymbol{\Sigma}^0) = \mathbf{1}$.

From the definition, the nonparanormal distribution must be a continuous distribution. And it is actually a special case of Gaussian copula family, when the distribution is continuous.

Suppose we have $\boldsymbol{X} \sim \mathrm{NPN}_d(\boldsymbol{\Sigma}^0, f^0)$, then the marginal cdfs are

$$F_j(x_j) = P(X_j \le x_j) = P(f_j^0(X_j) \le f_j^0(x_j)) = \Phi(f_j^0(x_j))$$

And the joint cdf is

$$\begin{aligned}
F(x_1, x_2, \ldots, x_d) &= P(X_1 \le x_1, \ldots, X_d \le x_d) \\
&= P(f_1^0(X_1) \le f_1^0(x_1), \ldots, f_d^0(X_d) \le f_d^0(x_d)) \\
&= \Phi_{\boldsymbol{\Sigma}^0}(f_1^0(x_1), \ldots, f_d^0(x_d))) \\
&= \Phi_{\boldsymbol{\Sigma}^0}(\Phi^{-1}(\Phi(f_1^0(x_1))), \ldots, \Phi^{-1}(\Phi(f_d^0(x_d)))) \\
&= \Phi_{\boldsymbol{\Sigma}^0}(\Phi^{-1}(F_1(x_1)), \ldots, \Phi^{-1}(F_d(x_d))) \\
&= C_{\boldsymbol{\Sigma}^0}(F_1(x_1), \ldots, F_d(x_d))
\end{aligned}$$

where $\Phi_{\mathbf{\Sigma}^0}$ is cdf of $\mathrm{N}_d(\mathbf{0}, \mathbf{\Sigma}^0)$ and $C_{\mathbf{\Sigma}^0}$ is a Gaussian copula with correlation matrix $\mathbf{\Sigma}^0$. So $\boldsymbol{X} \sim \mathrm{NPN}_d(\mathbf{\Sigma}^0, f^0)$ actually means $(F_1(X_1), \ldots, F_d(X_d))^T \sim C_{\mathbf{\Sigma}^0}$ or $(\Phi^{-1}(F_1(X_1)), \ldots, \Phi^{-1}(F_d(X_d)))^T \sim \mathrm{N}(\mathbf{0}, \mathbf{\Sigma}^0)$.

2.2 Estimation of $\mathbf{\Sigma}^0$

For nonparanormal distribution, we already know $F_j(x_j) = \Phi(f_j^0(x_j))$, then one natural way is to first estimate $f_j^0(x_j) = \Phi^{-1}(F_j(x_j))$, and then use the estimated $\hat{f}_j^0$ to transform the data to multivariate normal and estimate $\mathbf{\Sigma}^0$ as if we have data from a multivariate normal distribution. Suppose we have our data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ from nonparanormal distribution, and $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{id})^T$, then Liu et al. (2012) gave an estimation of f_j^0 by

$$\hat{f}_j^0(t) = \Phi^{-1}\left(T_{\delta_n}[\hat{F}_j(t)]\right)$$

where T_{δ_n} is a Winsorization (or truncation) operator defined as $T_{\delta_n}(x) = \delta_n I(x < \delta_n) + xI(\delta_n \leq x \leq 1 - \delta_n) + (1 - \delta_n)I(x > 1 - \delta_n)$ with $\delta_n = 1/(4n^{1/4}\sqrt{\pi \log n})$. And $\hat{F}_j(t) = \frac{1}{n+1}\sum_{i=1}^n I(x_{ij} \leq t)$ is the scaled empirical cdf of X_j. Then

$$\hat{\mathbf{\Sigma}}_{jk}^0 = \frac{\frac{1}{n}\sum_{i=1}^n \hat{f}_j^0(x_{ij})\hat{f}_k^0(x_{ik})}{\sqrt{\frac{1}{n}\sum_{i=1}^n \left(\hat{f}_j^0(x_{ij})\right)^2}\sqrt{\frac{1}{n}\sum_{i=1}^n \left(\hat{f}_k^0(x_{ik})\right)^2}}, \ \hat{\mathbf{\Sigma}}^0 = [\hat{\mathbf{\Sigma}}_{jk}^0]$$

Another way to estimate $\mathbf{\Sigma}^0$ directly without estimating f_j^0 by Liu et al. (2012) uses Spearman's ρ and Kendall's τ statistics. Let r_{ij} be the rank of x_{ij} among $x_{1j}, \ldots, x_{nj}$ and $\bar{r}_j = \frac{1}{n}\sum_{i=1}^n r_{ij}$, then

$$\hat{\rho}_{jk} = \frac{\sum_{i=1}^n (r_{ij} - \bar{r}_j)(r_{ik} - \bar{r}_k)}{\sqrt{\sum_{i=1}^n (r_{ij} - \bar{r}_j)^2 \cdot \sum_{i=1}^n (r_{ik} - \bar{r}_k)}}$$

$$\hat{\tau}_{jk} = \frac{2}{n(n-1)} \sum_{1 \leq i < i' \leq n} \mathrm{sign}\left(x_{ij} - x_{i'j}\right)\left(x_{ik} - x_{i'k}\right)$$

The population version of Spearman's ρ and Kendall's τ are

$$\rho_{jk} = \mathrm{Corr}(F_j(X_j), F_k(X_k)), \ \tau_{jk} = \mathrm{Corr}\left(\mathrm{sign}(X_j - \tilde{X}_j), \mathrm{sign}(X_k - \tilde{X}_k)\right)$$

where $\tilde{X}_j$ and $\tilde{X}_k$ are two independent copies of X_j and X_k.

Lemma 2.1 (Liu et al. (2012))**.** *Assuming $\boldsymbol{X} \sim \mathrm{NPN}(\mathbf{\Sigma}^0, f^0)$, we have*

$$\mathbf{\Sigma}_{jk}^0 = 2\sin\left(\frac{\pi}{6}\rho_{jk}\right) = \sin\left(\frac{\pi}{2}\tau_{jk}\right)$$

By Lemma 2.1, we then estimate $\mathbf{\Sigma}^0$ by

$$\hat{\mathbf{\Sigma}}_{jk}^0 = 2\sin\left(\frac{\pi}{6}\hat{\rho}_{jk}\right) = \sin\left(\frac{\pi}{2}\hat{\tau}_{jk}\right)$$

For $\hat{\mathbf{\Sigma}}^0 = \left[2\sin\left(\frac{\pi}{6}\hat{\rho}_{jk}\right)\right]$, Han and Liu (2014) also showed when $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n \overset{\mathrm{iid}}{\sim} \mathrm{NPN}_d(\mathbf{\Sigma}^0, f^0)$, for any $n > 21/\log d + 2$,

$$P\left(\|\hat{\mathbf{\Sigma}}^0 - \mathbf{\Sigma}^0\|_{\max} \leq 8\pi\sqrt{\frac{\log d}{n}}\right) \geq 1 - \frac{2}{d^2}$$

where $\|\boldsymbol{M}\|_{\max} = \max\{|\boldsymbol{M}_{ij}|\}$.

Then we can use either estimation of $\mathbf{\Sigma}^0$ to perform our PCA. In Han and Liu (2014) and Han and Liu (2013), sparse assumption about the eigenvectors are imposed, and some theoratical properties were discussed about the sparse PCA solution.

2.3 Generalized distributional transform and continuous latent variables

Now we want to extend the above method to catagorical case. The main idea is to define a latent continuous variable and apply the estimation of $\mathbf{\Sigma}^0$ with Spearman's ρ and Kendall's τ (These two estimations both assume data are from continuous distribution).

Definition 2.2 (generalized distributional transformation, Rüschendorf (2013), Chapter 1)**.** Let Y be a real random variable with distribution function F and let V be a random variable independent of Y, such that $V \sim \text{Uniform}(0,1)$. The generalized distributiona function is defined by

$$F(x,\lambda) = P(X < x) + \lambda P(Y = x)$$

and we call

$$U = F(Y,V)$$

the generalized distributional transform of Y.

Theorem 2.2 (Rüschendorf (2013), Chapter 1)**.** *Let $U = F(Y,V)$ be the distributional transform of Y as defined above, then*

$$U \sim \text{Uniform}(0,1) \text{ and } Y = F^{-1}(U) \text{ a.s.}$$

where

$$F^{-1}(t) = \inf\{x \in \mathbb{R} : F(x) \geq t\}$$

is the generalized inverse of F, or the quantile transform of F.

Now suppose our $\boldsymbol{X}$ is from a discrete distribution of Gaussian copula family, then

$$F(x_1,\ldots,x_d) = C_{\mathbf{\Sigma}^0}(F_1(x_1),\ldots,F_d(x_d)),$$

which means

$$(U_1,\ldots,U_d)^T = (F_1(X_1,V_1),\ldots,F(X_d,V_d))^T \sim \text{N}(\mathbf{0},\mathbf{\Sigma}^0)$$

by Skalar's Theorem (see Rüschendorf (2013)).

Thus we can consider $U_j = F_j(X_j,V_j)$, $j = 1,\ldots,d$ to be our latent variables, where $F_j(x_j,1) = P(X_j \leq x_j)$ is the marginal cdf of $\boldsymbol{X}$. And our data is generated by a quantile transformation of U_j, i.e. $X_j = F_j^{-1}(U_j)$

To transform the catagorical data to the latent continuous one, we will use the empirical cdf and empirical pmf. Suppose for X_j there are m possible values which are $c_1 < c_2,\ldots,c_m$, then the empirical pmf is

$$\hat{P}_j(c_l) = \frac{1}{n}\sum_{i=1}^{n} I(x_{ij} = c_l)$$

and the empirical cdf is

$$\hat{F}_j(t) = \sum_{i=1}^{n} I(x_{ij} \leq t)$$

and

$$\hat{F}_j(c_l) = \sum_{k=1}^{l} \hat{P}_j(c_k)$$

Then our transformation would be

$$\hat{F}_j(c_l,V) = \hat{F}_j(c_{l-1}) + \hat{P}_j(c_l)V$$

when $l = 1$, we denote $\hat{F}_j(c_0) = 0$.

And the quantile transform would be

$$\hat{F}_j^{-1}(u) = \sum_{k=1}^{m} c_k I(\hat{F}_j(c_{k-1}) < u \leq \hat{F}_j(c_k))$$

We perform the copula PCA with $\mathbf{\Sigma}^0$ estimated by Spearman's ρ or Kendall's τ from the data we obtained with generalized distributional transformation. Then we can do dimension reduction for $(\Phi^{-1}(U_1), \ldots, \Phi^{-1}(U_d))^T$. We can also transform the data projected to a lower dimension back to the catagorical data and see the difference.

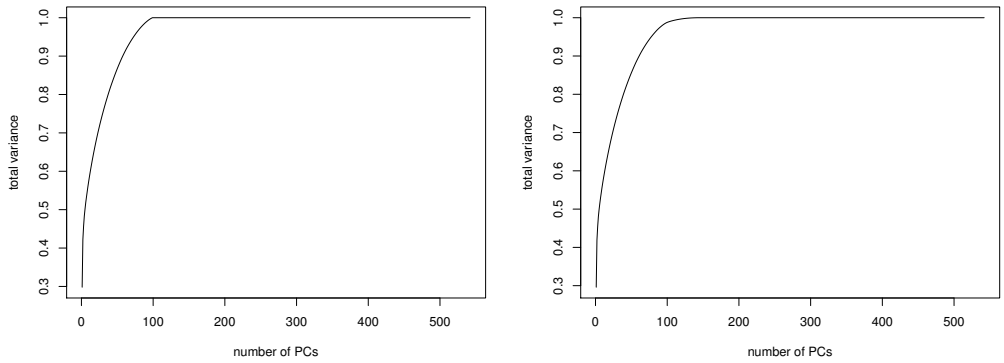
Also, we need to note that the estimated $\hat{\mathbf{\Sigma}}^0$ using this method might not be a correlation matrix (might not be positive and with 1's in the diagnal). Thus in practice we find the nearest correlation matrix of $\hat{\mathbf{\Sigma}}^0$ and use this correlation matrix to perform our PCA. Method used here is by Higham (2002) to find the nearest correlation matrix in terms of ∞ norm.

3 Senators' voting data set

We use senators' voting data to try this method. In this data set, there are 100 observations, each observation has 542 components. Possibles values are $-1, 0, 1$. We first transform the data to continuous one with empirical marginal distribution and then calculate the estimation of $\mathbf{\Sigma}^0$. We can also test the multivariate normality, since a basic assumption we made is that the data is from Gaussian copula family.

We test the multivariate normality of the marginal transformed data by projecting the data to one dimension of some arbitrary directions, e.g. 10000 differeny directions. The p-value is **0.03988561**, which means our assumption that the data are from a Gaussian copula family might not hold. But we will still continue in spite of this.

We used both Spearman's ρ and Kendall's τ estimations. For Spearman's ρ, the first 38 and 58 PCs can account for 80% and 90% of the total variance . And for Kendall's τ, the first 39 PCs and 61 PCs can account for 80% and 90% of the total variance. The cumulative proportion of total variance against number of PCs is shown in Figure 1.



(a) by estimation with Spearman's ρ (b) by estimation with Kendall's τ

Figure 1: Cumulative proportion of variance against number of PCs with two kinds of estimations of correlation matrix

Then we reconstruct the senators' voting data by these few PCs (number of PCs accounting for 80% and 90% of variance with estimation from Spearman's ρ and Kendall's τ) with the help of quantile transformation, and compare how much of our catagorical data matches with the original

one. The percent of matched data is shown in Table 1. We can see we can reconstruct the data pretty well although our assumption is not reallt met here.

Table 1: Percent of matching data between dimension reduced and original

	Spearman's ρ	Kendall's τ
80%	0.9196863	0.9167159
90%	0.9295387	0.9244834

4 Discussion

We can see this method is pretty simple and performs well in the senators' voting data set. Although the assumption is not really met we still get good result: the dimension is reduced a lot and not a lot of information is lost. One reason might be our estimation is based one the rank information of data, and that provided some robustness when deviating from the assumption. We could invetigate about this in the future if possible.

References

Fang Han and Han Liu. Principal component analysis on non-gaussian dependent data. In *International Conference on Machine Learning*, pages 240–248, 2013.

Fang Han and Han Liu. High dimensional semiparametric scale-invariant principal component analysis. *IEEE transactions on pattern analysis and machine intelligence*, 36(10):2016–2032, 2014.

Nicholas J Higham. Computing the nearest correlation matrixa problem from finance. *IMA journal of Numerical Analysis*, 22(3):329–343, 2002.

Han Liu, Fang Han, Ming Yuan, John Lafferty, and Larry Wasserman. The nonparanormal skeptic. *arXiv preprint arXiv:1206.6488*, 2012.

Ludger Rüschendorf. Mathematical risk analysis. *Springer Ser. Oper. Res. Financ. Eng. Springer, Heidelberg*, 2013.

Appendix A R codes

```r
library(readxl)
senators<-read_xls("senate_voting_data.xls")

#b) Plot Andrews' curves

senators.names<-names(senators)[-c(1,2)]
rev.party.state.names<-lapply(X=strsplit(gsub(pattern="[.]",replacement="",x=senators.names),split="
 ↪  "),FUN = rev)

senators.party <- lapply(X = rev.party.state.names, FUN = function(x)(unlist(x)[1]))
senators.party <- unlist(senators.party)

senators.last.names <- lapply(X = rev.party.state.names, FUN =
 ↪  function(x)(unlist(x)[4]))
senators.last.names <- unlist(senators.last.names)


#Create new data.frame for plotting
senators_new <- as.data.frame(t(senators[,-c(1,2)]))

colnames(senators_new) <- NULL
rownames(senators_new) <- NULL

senators_new <- data.frame(senators_new, party = senators.party)


# Use the codes from Canvas
source("ggandrews.R")

# Display the Andrews' curves
ggandrews(senators_new, type = 2, clr = 543, linecol = c("blue", "green", "red"),
 ↪  return_value = FALSE)


epmf <- function(x){
  n <- length(x)
  return(c(sum(x==-1), sum(x==0), sum(x==1))/n)
}

ecdf <- function(epmf){
  return(c(0, epmf[1], epmf[1]+epmf[2], 1))
}

senator_epmf <- sapply(senators_new[,-ncol(senators_new)], epmf)
senator_ecdf <- apply(senator_epmf, MARGIN = 2, ecdf)

library(plyr)

gdtrans <- function(i){
  u1 <- mapvalues(senators_new[,i], from = c(-1,0,1), to = senator_epmf[,i])
  u2 <- mapvalues(senators_new[,i], from = c(-1,0,1), to = senator_ecdf[1:3,i])
  return(u2 + u1*runif(n = length(u1)))
}

inv_gdtrans <- function(u, i){
  -1* (senator_ecdf[1,i] < u[,i] & senator_ecdf[2,i] >= u[,i]) + 1* (senator_ecdf[3,i] <
   ↪  u[,i] & senator_ecdf[4,i] >= u[,i])
}

source("testnormality.R")
```

```r
senators_c <- sapply(1:542, gdtrans)

senators_normal <- qnorm(senators_c, mean = 0, sd = 1)

testnormality(senators_normal)

library(energy)
mvnorm.etest(x = senators_normal, R = 1000)

library(MASS)
library(Matrix)

# rho based estimation
spearmanrho <- cor(senators_c, method = "spearman")
sigmahat_rho <- 2*sin(pi/6*spearmanrho)
sigmahat_rho <- nearPD(sigmahat_rho, corr = T)
decomp_rho <- eigen(sigmahat_rho$mat, symmetric = T)

s <- decomp_rho$values

pvar<-s/sum(s)

#  cumulative proportion of total variance explained
#  by each component

cpvar <- cumsum(s)/sum(s)
plot(x = 1:length(cpvar), y = cpvar, 'n', xlab = 'number of PCs', ylab = 'total
 ↪  variance')
lines(x = 1:length(cpvar), y = cpvar)

npc80 <- min(which(cpvar > 0.8))
npc90 <- min(which(cpvar > 0.9))

senators_normal_80 <-
 ↪  senators_normal%*%decomp_rho$vectors[,1:npc80]%*%t(decomp_rho$vectors[,1:npc80])

senators_u_80 <- pnorm(senators_normal_80, mean = 0, sd = 1)

senators_80 <- sapply(1:542, inv_gdtrans, u = senators_u_80)

sum(senators_80==senators_new[,-ncol(senators_new)])/(542*100)


senators_normal_90 <-
 ↪  senators_normal%*%decomp_rho$vectors[,1:npc90]%*%t(decomp_rho$vectors[,1:npc90])

senators_u_90 <- pnorm(senators_normal_90, mean = 0, sd = 1)

senators_90 <- sapply(1:542, inv_gdtrans, u = senators_u_90)

sum(senators_90==senators_new[,-ncol(senators_new)])/(542*100)

# tau based estimation
kendalltau <- cor(senators_c, method = "kendall")
sigmahat_tau <- sin(pi/2*kendalltau)
sigmahat_tau <- nearPD(sigmahat_tau, corr = T)
decomp_tau <- eigen(sigmahat_tau$mat, symmetric = T)

s <- decomp_tau$values

pvar<-s/sum(s)

#  cumulative proportion of total variance explained
#  by each component

cpvar <- cumsum(s)/sum(s)
```

```r
plot(x = 1:length(cpvar), y = cpvar, 'n', xlab = 'number of PCs', ylab = 'total variance')
lines(x = 1:length(cpvar), y = cpvar)
npc80 <- min(which(cpvar > 0.8))
npc90 <- min(which(cpvar > 0.9))

senators_normal_80 <- senators_normal%*%decomp_tau$vectors[,1:npc80]%*%t(decomp_tau$vectors[,1:npc80])

senators_u_80 <- pnorm(senators_normal_80, mean = 0, sd = 1)

senators_80 <- sapply(1:542, inv_gdtrans, u = senators_u_80)

sum(senators_80==senators_new[,-ncol(senators_new)])/(542*100)


senators_normal_90 <- senators_normal%*%decomp_tau$vectors[,1:npc90]%*%t(decomp_tau$vectors[,1:npc90])

senators_u_90 <- pnorm(senators_normal_90, mean = 0, sd = 1)

senators_90 <- sapply(1:542, inv_gdtrans, u = senators_u_90)

sum(senators_90==senators_new[,-ncol(senators_new)])/(542*100)
```