Fitting Gaussian One Factor Copula

2024-10-20

```
library(cubature)
unif_df = read.csv('unif_df.csv')
library(pracma)
```

```
gaussian_derivative_single_obs_R = function(u1, rho, nl_x2, wl){
  d = length(u1)
  nodes = length(wl)
  rho_squared = rho^2
  one_minus_rho_squared = 1 - rho_squared
  u1 = as.numeric(u1)
  x1 = qnorm(u1, mean = 0, sd = 1)
  x2 = matrix(rep(nl_x2, d), ncol = nodes, nrow = d, byrow = TRUE)

  numerator = 2 * rho * x1 * x2 - rho_squared * (x1^2 + x2^2)
  denominator = 2 * one_minus_rho_squared
  exponent = numerator / denominator
  out = exp(exponent) / sqrt(one_minus_rho_squared)
  prod_vec = apply(out, 2, prod)
  ##########calcualte numerator
  # we have a d by |nl| matrix (columns are the derivative values for each nl)
  x2 = matrix(rep(nl_x2, d), ncol = nodes, nrow = d, byrow = TRUE)
  deriv_mul_vec = (rho * one_minus_rho_squared + (1 + rho_squared) * x1 * x2 - rho * (x1^2 + x2^2)) / (
  num = deriv_mul_vec%*%diag(prod_vec) %*% diag(wl)
  num = rowSums(num)
  ##########calcualte denominator
  denom = sum(prod_vec * wl)
  result = num/denom
  return(list(log_likelihood = log(denom),  derivatives = result))
}

#this function returns a vector of derivatives for a all observations, and the log likelihood for all o
gaussian_derivative_R = function(unif_df, rho, nl_x2, wl){
  out = t(apply(unif_df, 1, gaussian_derivative_single_obs_R, rho = rho, nl_x2 = nl_x2, wl = wl))
  log_likelihood = sum(sapply(out, function(x) x$log_likelihood))
  derivatives = rowSums(sapply(out, function(x) x$derivatives))
  return(list(log_likelihood = log_likelihood, derivatives = derivatives))
}

neg_log_gaussian_likelihood_R = function(unif_df, rho, nl_x2, wl){
  # print("rho: ")
  # print(rho)
  if ((max(rho) > 0.9999999) | (min(rho) < -0.9999999) | rho[1]<0){
    log_likelihood = 9e5
    attr(log_likelihood, "gradient") = 1e6*sign(rho)
    return(log_likelihood)
```

```
  }
  result = gaussian_derivative_R(unif_df, rho, nl_x2, wl)
  log_likelihood = -1 * result$log_likelihood
  gradient = result$derivatives
  attr(log_likelihood, "gradient") = -1 * gradient
  return(log_likelihood)
}

fit_gaussian_copula_R = function(unif_df, rho, n_nodes){
  gl = gaussLegendre(n_nodes, 0, 1)
  nl_x2 = qnorm(gl$x)
  wl = gl$w
  d = ncol(unif_df)
  result_test = nlm(f = neg_log_gaussian_likelihood_R, p = rho, unif_df = unif_df, nl_x2 = nl_x2, wl = w
  return(result_test)
}
```

```
start = Sys.time()
model_R = fit_gaussian_copula_R(unif_df, rho = rep(0.5, ncol(unif_df)), n_nodes = 20)
end = Sys.time()
print(cat('R code runtime:', end-start))
```

```
## R code runtime: 0.865397NULL
```

```
Rcpp::sourceCpp("FitGaussianFactorCopula.cpp")
```

```
## Registered S3 methods overwritten by 'RcppEigen':
##   method               from
##   predict.fastLm       RcppArmadillo
##   print.fastLm         RcppArmadillo
##   summary.fastLm       RcppArmadillo
##   print.summary.fastLm RcppArmadillo
```

```
print(ncol(unif_df))
```

```
## [1] 6
```

```
rho = rep(0.5, ncol(unif_df))
nl_x2 = seq(0.1, 0.6, by = 0.1)
wl = seq(0.1, 0.6, by = 0.1)

start = Sys.time()
print(gaussian_derivative_single_obs(as.numeric(unif_df[1,]), rho, nl_x2, wl))
```

```
## $log_likelihood
## [1] 0.4068418
##
## $derivatives
##            [,1]
## [1,] -0.2938731
## [2,] -0.5227494
## [3,]  0.3671216
## [4,] -0.6892824
## [5,]  0.2568133
## [6,]  0.2859252
```

```r
end = Sys.time()
cat('R code runtime:', end-start, "\n")
```

```
## R code runtime: 0.00199914
```

```r
start = Sys.time()
print(gaussian_derivative_single_obs_R(as.numeric(unif_df[1,]), rho, nl_x2, wl))
```

```
## $log_likelihood
## [1] 0.4068418
##
## $derivatives
## [1] -0.2938731 -0.5227494  0.3671216 -0.6892824  0.2568133  0.2859252
```

```r
end = Sys.time()
cat('CPP code runtime:', end-start, "\n")
```

```
## CPP code runtime: 0.0005040169
```

```r
start = Sys.time()
print(gaussian_derivative_R((unif_df), rho, nl_x2, wl))
```

```
## $log_likelihood
## [1] 567.318
##
## $derivatives
## [1] -190.4316 -289.8604 -314.7263 -287.0782 -160.1335 -312.2942
```

```r
end = Sys.time()
print(gaussian_derivative(as.matrix(unif_df), rho, nl_x2, wl))
```

```
## $log_likelihood
## [1] 567.318
##
## $derivatives
##           [,1]
## [1,] -190.4316
## [2,] -289.8604
## [3,] -314.7263
## [4,] -287.0782
## [5,] -160.1335
## [6,] -312.2942
```

```r
cat('R code runtime:', end-start, "\n")
```

```
## R code runtime: 0.02927208
```

```r
start = Sys.time()
print(gaussian_derivative(as.matrix(unif_df), rho, nl_x2, wl))
```

```
## $log_likelihood
## [1] 567.318
##
## $derivatives
##           [,1]
## [1,] -190.4316
## [2,] -289.8604
## [3,] -314.7263
```

```
## [4,] -287.0782
## [5,] -160.1335
## [6,] -312.2942
```

```
end = Sys.time()
print(gaussian_derivative(as.matrix(unif_df), rho, nl_x2, wl))
```

```
## $log_likelihood
## [1] 567.318
##
## $derivatives
##           [,1]
## [1,] -190.4316
## [2,] -289.8604
## [3,] -314.7263
## [4,] -287.0782
## [5,] -160.1335
## [6,] -312.2942
```

```
cat('R code runtime:', end-start, "\n")
```

```
## R code runtime: 0.002675056
```

```
start = Sys.time()
model_cpp = fit_gaussian_copula(as.matrix(unif_df), rho = rep(0.5, ncol(unif_df)), n_nodes = 20)
end = Sys.time()
print(cat('CPP code runtime:', end-start))
```

```
## CPP code runtime: 0.02250504NULL
```