Содержание

Введение

Целью данной работы является исследование распределения квадратичной формы случайного вектора методами статистического моделирования. При исследовании, разработке алгоритмов часто требуется определить качество их работы. Можно, например, использовать для этого реальные данные, но это не всегда возможно, так как не всегда понятны их свойства, поэтому по этим данным, вообще говоря, нельзя дать оценку эффективности алгоритмов. Здесь можно применить моделирование данных по уже известным законам распределения так, что в итоге можно оценить качество работы алгоритма. Принято считать, что в случае, если алгоритм работает на тестовых данных, то его можно применить к реальным данным. Единственная тонкость состоит в том, что это относится лишь к непараметрическим алгоритмам.

Чаще всего используется моделирование данных, распределенных по нормальному закону. В MS Excel и статистических пакетах (SPSS, Statistica) встроенными функциями возможно моделировать только одномерные статистические распределения. Можно составить многомерное распределение из нескольких одномерных, но лишь когда переменные независимы. Если же нужно исследовать данные с зависимыми переменными, то приходится реализовывать в виде программ.

Кроме того, как правило в математических пакетах отсутствует функция распределения Колмогорова-Смирнова. И даже в случае наличия, не всегда критерий Колмогорова-Смирнова можно применить к многомерным данным или же к неким параметрам.

Поэтому в качестве задания была выбрана задача разработки метода и реализации его для исследования многомерных законов распределения, и моделирования статистики Колмогорова. В частности проверка того, что статистика для проверки многомерной нормальности имеет хи-квадрат распределение с помощью критерия Колмогорова.

Моделирование статистики Колмогорова

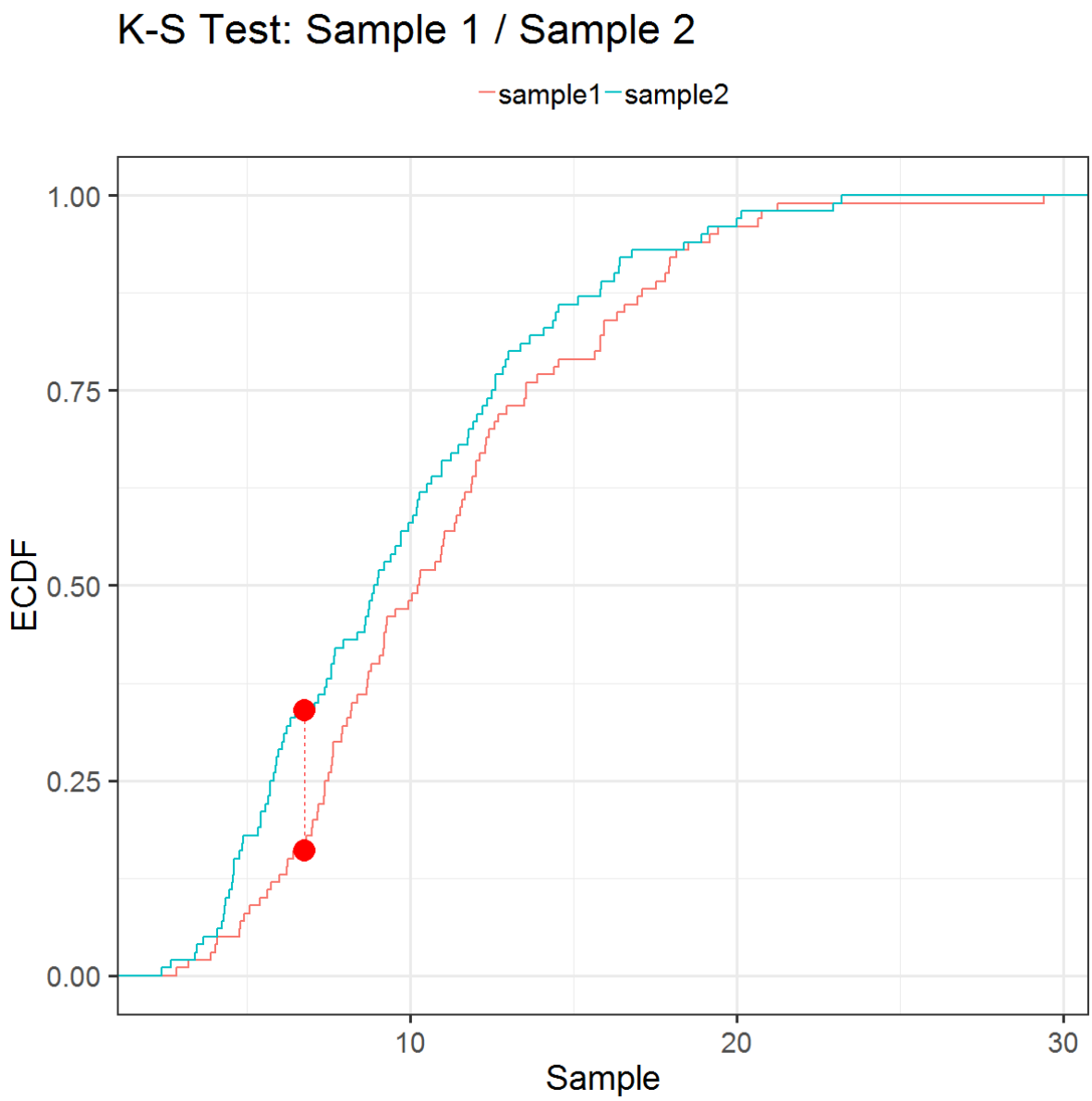

Рис. 1. D_n-статистика Колмогорова

Функция распределения Колмогорова или критерий Колмогорова, красным выделена статистика Колмогорова (рис. 1), используется для проверки гипотез о принадлежности уже известному закону распределения. Статистика Колмогорова или информация выражается в виде наибольшей разности двух функций распределения:

$$D_n = sup\ |F_n(x) - F(x)|$$

$F_n(x)$–эмпирическая функция распределения;
$F(x)$–некая «истинная» функция распределения;

Многомерное нормальное распределение $X = (X_1, \ldots, X_p)$ описывается вектором математических ожиданий $\mu = (\mu_1, \ldots, \mu_p)$ и положительно определенной ковариационной матрицей

$$\Lambda = ||\delta_{i,j}|| \quad i,j = 1, \ldots, p,$$

где

$$\delta_{i,j} = Cov(X_i, X_j) = \Lambda, \qquad i, j = 1, \ldots, p$$

ковариация случайных величин X_i и X_j.

Рис. 2. Функции плотности

«Теорема. Если X из нормального распределения, то его квадратичная форма Y^2 имеет хи-квадрат распределение с p степенями свободы.»

Исходя из этого, можно смоделировать многомерное распределение хи-квадрат из многомерного нормального распределения с данными параметрами, то есть, ковариационной матрицей $\sum$ и вектором математических ожиданий μ, а p-размерность нормального вектора и также размерность ковариационной матрицы и размерность нормального распределения, степени свободы распределения хи-квадрат. Матрицу ковариаций будем искать в виде:

$$\Lambda = AA^T$$
$$X = A\eta + \mu$$
$$\eta = (\eta_1, \ldots, \eta_p)$$

Линейное преобразование вектора η, для моделирования X, где компоненты вектора нормальные распределенные случайные величины с параметрами дисперсии, равной единице, и математическим ожиданием, равным нулю. Оценка ковариационной матрицы:

$$\sum = 1/(1 - p)A^TA$$

Исходя из этого будет моделироваться квадратичная форма, которая будет иметь вид (см. рис. 2):

$$Y^2 = (\vec{X} - \vec{\mu})'\Lambda^{-1}(\vec{X} - \vec{\mu})$$

«Истинное» распределение хи-квадрат известно (см. рис. 2):

$$F_n = Y^2$$

$$F_{x(k)^2}(x) = \frac{\gamma\left(\frac{k}{2}, \frac{x}{2}\right)}{\Gamma\left(\frac{k}{2}\right)}$$

где Γ и γ обозначают соответственно полную и неполную гамма-функции, k-степень свободы

Проверка полученного распределения

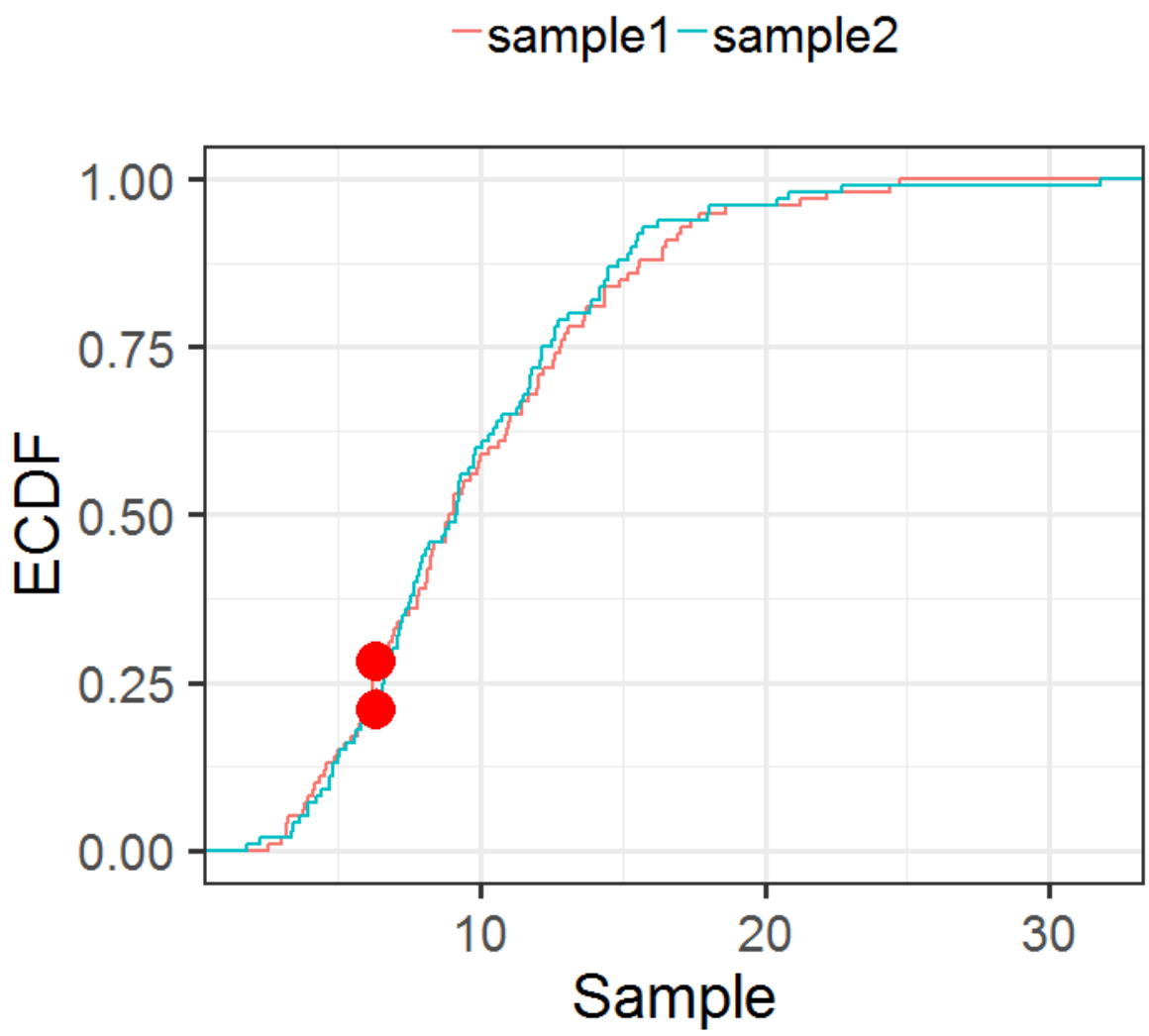

Рис. 3. Достижение максимума статистики D_n при $p = 10$

Исходя из теоремы, степени свободы p будут определять верно ли получилось смоделировать статистику колмогорова. Для того чтобы выяснить это, надо лишь изменять степени свободы и проверять, при каких p статистика D_n будет достигать оптимальных значений (т.е. зафиксируем p у смоделированной квадратичной формы и будем изменять p у «истинного» распределения, при этом будем фиксировать минимум D_n), при достижении оптимальных (см. рис. 3) значений степени свободы будут принимать значения согласно с теоремой.

Таблица 1

Оптимальная статистика D_n при различных p

D_n	p
0.30750000	2
0.35000000	3
0.33000000	4
0.33428571	5
0.30687500	6
0.24012346	7
0.25000000	8
0.13371901	9
0.17527778	10
0.21473373	11
0.17551020	12
0.09444444	13
0.19859375	14
0.32356401	15
0.27074074	16
0.12177285	`17
0.27750000	18
0.13961451	19
0.20942149	20

Заключение

В результате выполнения данной работы было сделано следующее:

1) Было смоделировано многомерное нормальное распределение.

2) Смоделировано многомерное распределение хи-квадрат.

3)Сравнение многомерных смоделированных и “истинных” функций распределения хи-квадрат.

4) Спроектировано и запрограммировано распределение Колмогорова.

Поставленная цель – исследование распределения квадратичной формы случайного вектора методами статистического моделирования – была достигнута.

Список литературы

1. Симушкин С. В. Методические разработки по специальному курсу: многомерный статистический анализ / С. В. Симушкин. Казань, 2006. – ч.1. – с. 71-77.
2. Лемешко Б.Ю., Лемешко С.Б., Миркин Е.П. Исследование критериев проверки гипотез, используемых в задачах управления качеством // Материалы VII международной конференции “Актуальные проблемы электронного приборостроения” АПЭП-2004. Новосибирск, 2004. – т. 6. – с. 269-272.
3. Смирнов Н.В. Оценка расхождения между эмпирическими кривыми распределения в двух независимых выборках // Бюллетень МГУ, серия А. – 1939. – Т.2. №2. – С.3-14.

Приложение

```
library(fitdistrplus)
library(MASS)
library("matlab")

quadraticForm <- function(p){
  yl <- c()
  for (i in seq(1,p*p,1)) {
    y<-c()
    mu <- rnorm(p) #мат ожидание
    mcov <- matrix(data = rnorm(p*p,0,1), nrow = p, ncol = p, byrow = T)
    mcov <- 1/(p-1) *t(mcov) %*% mcov #оценка ковариационной
матрицы
    A<-t(chol(mcov))
    nu<-rnorm(p,0,1) #возможно она(с.в.) должна быть в (0,1]???
    x <- A %*% nu + mu #линейное преобразование вектора nu в
    y <- t(x-mu) %*% ginv(mcov) %*% (x-mu) #квадратичная форма
    yl<-c(yl,y)
  }
  return(yl)
}
plot(ecdf(quadraticForm(10)))
lines(ecdf(rchisq(100,10)), col = "red")

pkolmogorov1x <- function(x, n) {
  if (x <= 0)
    return(0)
  if (x >= 1)
    return(1)
  j <- seq.int(from = 0, to = floor(n * (1 - x)))
  1 - x * sum(exp(lchoose(n, j) + (n - j) * log(1 - x - j/n) + (j - 1) * log(x +
j/n)))
}
```

```
kolmogor <- function(x) {
  eps<- 10^(-10)
  s  <- 0
  a <- 1
  n <- 0
  k <- 0
  da<- 100000
  ga<- 100000
  if(x <= 0) return(0)
  while((Mod(da)>= eps) ) {
    da<-a
    s <- s + a
    n <- n + 1
    a <- (-1)^n * exp((-2)*(n^2) * (x^2))
    da <- Mod(da)-Mod(a)
  }
  n <- -1
  a <- 1
  while((Mod(ga)>= eps) ) {
    ga <- a
    a <- (-1)^n * exp((-2)*(n^2) * (x^2))
    s <- s + a
    ga <- Mod(ga)-Mod(a)
    n <- n - 1
  }
  return(s)
}

dn <- function(yl,drch) {
  #Dn - minMax статистика Колмогорова
  require("ggplot2")
  sample1 <- yl # моя квадратичная форма
  sample2 <- drch #rchisq
  group <- c(rep("sample1", length(sample1)), rep("sample2",
length(sample2)))
```

```
  dat <- data.frame(KSD = c(sample1,sample2), group = group)
  # create ECDF of data
  cdf1 <- ecdf(sample1)
  cdf2 <- ecdf(sample2)
  # find min and max statistics to draw line between points of greatest
distance
  minMax <- seq(min(sample1, sample2), max(sample1, sample2),
length.out=length(sample1))
  x0 <- minMax[which( abs(cdf1(minMax) - cdf2(minMax)) ==
max(abs(cdf1(minMax) - cdf2(minMax))) )]
  y0 <- cdf1(x0)
  y1 <- cdf2(x0)
  return(abs(y1-y0))
 }

 drch <- rchisq(100,df = 10)
 ysqr<-quadraticForm(10)

 plot(density(rchisq(100,10)), main = "Ф-ии плотности хи-квдрат:
'истинная' и смоделированная")
 lines(density(ysqr), col = 'red')

 pkolmogorov1x <- function(x, n) {
  if (x <= 0)
   return(0)
  if (x >= 1)
   return(1)
  j <- seq.int(from = 0, to = floor(n * (1 - x)))
  1 - x * sum(exp(lchoose(n, j) + (n - j) * log(1 - x - j/n) + (j - 1) * log(x +
j/n)))
 }

 kd<-c()
 for (i in seq(0,1,0.05)) {
  kd<-c(pkolmogorov1x(i,20),kd)
```

```
}

plot(density(kd))
plot(density(yl), col = "red")
lines(density(rchisq(100,df = p)))
plot(kd)
######
sy<-sort(yl)
sch<-sort(rchisq(100,df=p))
dn <- sch - yl

plot((dn))



kl<-c()
for(i in seq(0,1,0.01)) {
  kl<-c(kolmogor(i),kl)
}
plot(density(kl))

seq(0,1,0.01)#эмперическая функция распределения
n <- length(yl)
x <- sort(yl); vals <- unique(x)
rval <- approxfun(vals, cumsum(tabulate(match(x, vals)))/n,
                  method = "constant", yleft = 0, yright = 1, f = 0,
                  ties = "ordered")
plot(ecdf(yl))
yl <- quadraticForm(10)
plot(yl)
plot(pkolmogorov1x(ecdf(yl),length(yl)))
drch <- rchisq(40,df = 10)
yl<-quadraticForm(10)
#Dn - minMax статистика Колмогорова
require(ggplot2)
sample1 <- yl # моя квадратичная форма
```

```
sample2 <- drch #rchisq
group <- c(rep("sample1", length(sample1)), rep("sample2",
length(sample2)))
dat <- data.frame(KSD = c(sample1,sample2), group = group)
# create ECDF of data
cdf1 <- ecdf(sample1)
cdf2 <- ecdf(sample2)
# find min and max statistics to draw line between points of greatest
distance
minMax <- seq(min(sample1, sample2), max(sample1, sample2),
length.out=length(sample1))
x0 <- minMax[which( abs(cdf1(minMax) - cdf2(minMax)) ==
max(abs(cdf1(minMax) - cdf2(minMax))) )]
y0 <- cdf1(x0)
y1 <- cdf2(x0)

# png(file = "c:/temp/ks.png", width = 1024, height = 768,
type="cairo-png")
ggplot(dat, aes(x = KSD, group = group, color = group))+
  stat_ecdf(size=1) +
  theme_bw(base_size = 28) +
  theme(legend.position ="top") +
  xlab("Sample") +
  ylab("ECDF") +
  #geom_line(size=1) +
  geom_segment(aes(x = x0[1], y = y0[1], xend = x0[1], yend = y1[1]),
               linetype = "dashed", color = "red") +
  geom_point(aes(x = x0[1] , y= y0[1]), color="red", size=8) +
  geom_point(aes(x = x0[1] , y= y1[1]), color="red", size=8) +
  ggtitle("Dn: Y / Chi-square, p = 10 ") +
  theme(legend.title=element_blank())

library(mvtnorm)

# Some mean vector and a covariance matrix
```

```
mu <- colMeans(iris[1:50, -5])
cov <- cov(iris[1:50, -5])

# genrate n = 100 samples
sim_data <- rmvnorm(n = 100, mean = mu, sigma = cov)

# visualize in a pairs plot
pairs(sim_data)

t<-c()
tn<- c()
ind<-c()
 for (i in seq(2,22,1)) {

  for (j in seq(0,2,0.1)) {
    t <- dn(quadraticForm(i+j), rchisq(100,i))
  }
  tn<- c(tn, min(t))
  ind<-c(ind,which.min(t)/100 +i)
  t<-c()
 }

plot(ecdf(tn))
plot(ecdf(kd))
ks.test(tn,)
as.array(tn)
```