project

Yaqiong Yao

11/24/2018

Numerical simulation

Univariate case

If we choose uniform kernel, the CDF of KDE should be represented as:

$$F_K(x|\mathcal{D}) = \frac{1}{n}\left(\sum_{i=1}^{n}\frac{x - x_i + b}{2b}I(x_i - b < x < x_i + b) + \sum_{i=1}^{n} I(x \geq x_i + b)\right).$$

If we set this equation to be u, which is probablity of $x \leq X$. We can hardly have the close form solution of x. Thus, if the support of x is not $\mathcal{R}$, then no close form can be obtained. Meanwhile, if we select more complicated kernels, such as normal distribution, the CDF of KDE becomes:

$$F_K(x|\mathcal{D}) = \frac{1}{n}\Phi(\frac{x - x_i}{b}).$$

Obviouly, no explicity closed form of x, either.

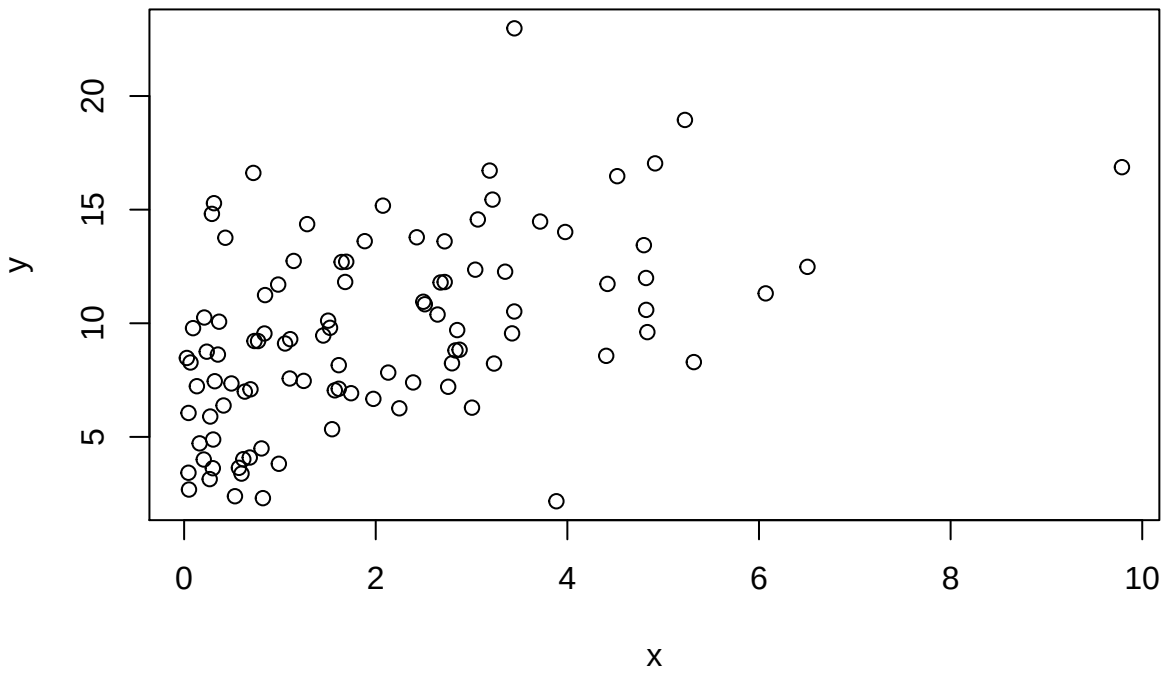
Bivariate case

Suppose we have a gaussian copula with two random variables, one follows gamma distribution with parameter 1 and 2, and another one is from chi-squared distribution with degree of freedom 10. The first method is mapping the cumulative probabilities from the generated gaussian copula with the quantiles of gamma and chi-square distribution respectively.

```
require(mvtnorm)
```

```
## Loading required package: mvtnorm
```

```
n <- 100
sigma <- matrix(c(1, 0.5, 0.5, 1), nrow = 2)
set.seed(098)
dat <- rmvnorm(n, sigma = sigma)
dat <- pnorm(dat)
x <- qgamma(dat[,1], shape = 1, scale = 2)
y <- qchisq(dat[,2], df = 10)
plot(x, y)
```

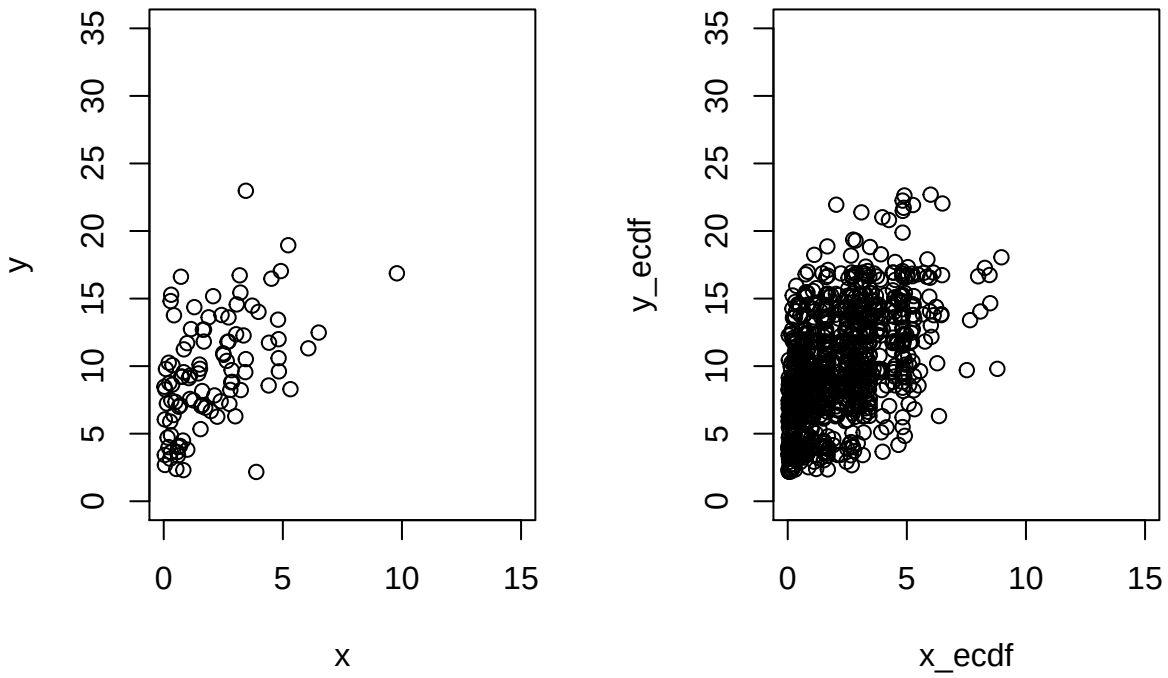


The second method is generate random numbers use the empirical cdf of last method and resampling from the samples of last method.

```
## generate random number with empirical cdf
n <- 1000
sigma <- matrix(c(1, 0.5, 0.5, 1), nrow = 2)
set.seed(456)
dat <- rmvnorm(n, sigma = sigma)
dat <- pnorm(dat)

x_ecdf <- quantile(x, dat[,1])
y_ecdf <- quantile(y, dat[,2])

par(mfrow = c(1,2))
plot(x, y, xlim = c(0, 15), ylim = c(0, 35))
plot(x_ecdf, y_ecdf, xlim = c(0, 15), ylim = c(0, 35))
```



The Third method is using kernel density estimator(KDE). To draw a value from the kernel density estimate: (1) draw a value from the kernel density Y; (2) independently select one of the data points at random X. Adding X and Y gives the value of the sample point.
Consider X is draw from the empirical distribution of data $(x_1, x_2, ..., x_n)$, Y is from kernel $U(0, 1)$. Therefore,

$$F_{X+Y}(x) = Pr(X + Y \leq x) \tag{1}$$

$$= \sum_{i=1}^{n} Pr(X + Y \leq x | X = x_i) Pr(X = X_i) \tag{2}$$

$$= \sum_{i=1}^{n} Pr(x_i + Y \leq x) Pr(X = X_i) \tag{3}$$

$$= \frac{1}{n} \sum_{i=1}^{n} Pr(Y \leq x - xi) \tag{4}$$

$$= \frac{1}{n} \sum_{i=1}^{n} F_K(x - x_i) \tag{5}$$

coincide with the CDF of the KDE.

```
## generate random number from KDE
## check the influence of bandwidth to the random number generating
sample.kernel <- function(n, x, y, adj){
  bw_x <- adj * density(x)$bw
  bw_y <- adj * density(y)$bw
  ind <- sample(1:100, n, replace = TRUE)
```

```
  x_kde <- x[ind] + runif(n, -bw_x, bw_x)
  y_kde <- y[ind] + runif(n, -bw_y, bw_y)
  cbind(x_kde, y_kde)
}
adj = 1
xy_kde <- sample.kernel(n, x, y, adj)

par(mfrow = c(1,2))
plot(x, y, xlim = c(0, 15), ylim = c(0, 35))
plot(xy_kde[,1], xy_kde[,2], xlim = c(0, 15), ylim = c(0, 35))
```