Eliciting Dependent Distributions using Multivariate Normal Copulas

Jeremy Oakley

2016-07-06

Introduction

We illustrate the process of incorporating dependence between uncertain quantities using a multivariate (bivariate) normal copula.

As an example, suppose a clinical trial is to be conducted for a new treatment. The trial will take place at two centres, with treatment intended to last for one year. It is believed that a proportion of patients recruited at each centre will not complete the treatment and hence drop out of the trial. Denote these two uncertain proportions by X_1 and X_2. We suppose that patient characteristics are believed to be different at the second centre, such that the expert is expecting a slightly higher drop-out rate (although she is not certain that $X_2 > X_1$).

Eliciting the marginal distributions

We first elicit marginal distributions for X_1 and X_2. We suppose the experts states

$$P(X_1 \leq 0.12) = 0.25, P(X_1 \leq 0.15) = 0.5, P(X_1 \leq 0.20) = 0.75,$$

$$P(X_2 \leq 0.15) = 0.25, P(X_2 \leq 0.2) = 0.5, P(X_1 \leq 0.25) = 0.75.$$

We fit distributions to each of these set of judgements.

```
library(SHELF)
p <- c(0.25, 0.5, 0.75)
v1 <- c(0.12, 0.15, 0.2)
v2 <- c(0.15, 0.2, 0.25)
myfit1 <- fitdist(vals = v1, probs = p, lower = 0, upper = 1)
myfit2 <- fitdist(vals = v2, probs = p, lower = 0, upper = 1)
```

We choose to use beta distributions for each marginal, and suppose that we have gone through the process of feedback with the expert, so that she is satisfied with the two fitted distributions. The two fitted distributions are plotted below. (To aid comparison, lower and upper axes limits `xl` and `xu` are specified in the `plotfit` commands.)

```r
plotfit(myfit1, d = "beta", xl = 0, xu = 0.5)
```

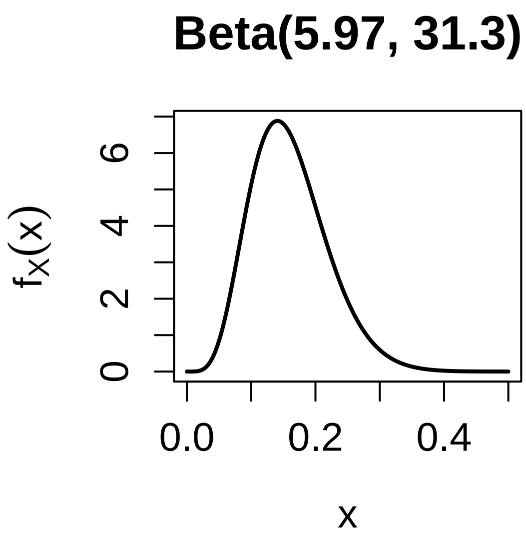

Figure 1: The fitted marginal distribution for X_1.

```r
plotfit(myfit2, d = "beta", xl = 0, xu = 0.5)
```

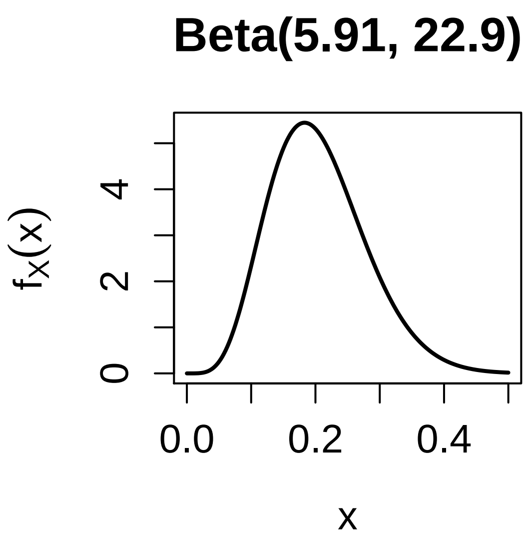

Figure 2: The fitted marginal distribution for X_2.

Incorporating dependence

We can incorporate dependence through the use of a bivariate normal copula. The general form of the joint density function of X_1 and X_2 is a little complex, but simulating from the joint distribution is more straightforward. The idea is as follows.

1. We can simulate a random value of X from any univariate probability distribution using inversion: we sample U from the $U[0,1]$ distribution, and then set our generated value of X to be the solution x of

$$P(X \leq x) = U.$$

2. We can sample *dependent* values of X_1 and X_2 from two separate marginal distributions by sampling *dependent* uniforms U_1 and U_2, and then setting the generated values of X_1 and X_2 to be the solutions x_1 and x_2 of

$$P(X_1 \leq x_1) = U_1, P(X_2 \leq x_2) = U_2.$$

3. In the bivariate normal copula method, we generate dependent uniforms U_1 and U_2 by sampling z_1, z_2 from the bivariate normal distribution

$$\begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} \sim N\left\{\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & r \\ r & 1 \end{pmatrix}\right\},$$

(with the choice of r discussed shortly), and then setting $U_1 = P(Z_1 \leq z_1)$ and $U_2 = P(Z_2 \leq z_2)$.

To obtain r, the expert is asked to consider her quadrant probability

$$p = P(X_1 > 0.15, X_2 > 0.2),$$

i.e. her probability that both uncertain proportions are above their medians. (We suggest using elicited rather fitted medians, although the difference should be small.) This probability must be between 0 and 0.5 with the values of 0, 0.25 and 0.5 corresponding to perfect negative correlation, independence, and perfect positive correlation respectively. Suppose she judges $p = 0.4$, in that if one proportion is higher than her median, she believes the other is likely to be also. The correlation parameter r can be obtained as

$$r = \sin(2\pi(p - 0.25)) = 0.81.$$

The function `copulaSample` can be used to sample from the required distribution. The function generalises to d dependent variables $X_1, \ldots, X_d$, and takes as input a matrix of quadrant probabilities, where element i, j of this matrix is the probability $P(X_i > m_i, X_j > m_j)$, where m_i and m_j are the corresponding elicited medians[1]. It is only necessary to specify the upper triangular elements of this matrix.

```
quadprob <- matrix(0, 2, 2)
quadprob[1, 2] <- 0.4
X <- copulaSample(myfit1, myfit2, qp = quadprob,
                  n = 1000, d = c("Beta", "Beta"))
```

The object `X` is a 1000×2 matrix, with i-th column corresponding to X_i. We verify that the sample has the right properties, checking the sample quartiles and quadrant probability.

[1]With $d > 2$, the elicited pairwise quadrant probabilities may not result in a positive-definite variance matrix. Currently the function copulaSample will simply return an error in this case.

```r
quantile(X[, 1], probs = c(0.25, 0.5, 0.75))
```

```
##     25%     50%     75%
## 0.11775 0.15300 0.19525
```

```r
quantile(X[, 2], probs = c(0.25, 0.5, 0.75))
```

```
##     25%     50%     75%
## 0.14900 0.19500 0.24425
```

```r
mean(X[, 1] > 0.15 & X[, 2] > 0.2)
```

```
## [1] 0.382
```

We plot the sample below.

```r
library(ggplot2)
ggplot(data.frame(X), aes(x = X1, y = X2)) +
  geom_point(alpha = 0.1, colour = "red") +
  geom_hline(yintercept = 0.2) +
  geom_vline(xintercept = 0.15) +
  labs(x=expression(X[1]), y = expression(X[2]))
```

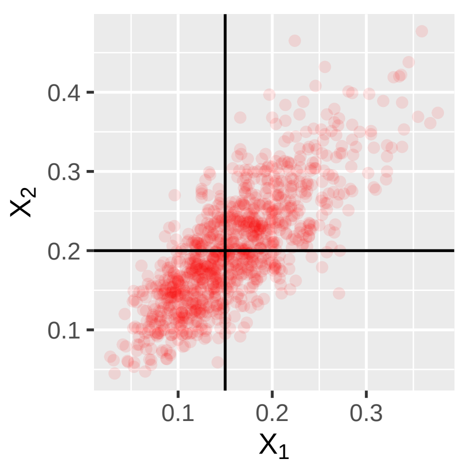

Figure 3: A sample from the joint distribution of X_1, X_2. The horizontal and vertical lines indicated the elicited medians. Note that expert has judged a probability of 0.4 of both X_1 and X_2 being above their median value, and so approximately 40% of the points are in the top right quadrant.

Interactive mode

We can use the function `elicitQuadProb` to open a browser (using `shiny`) in which we can adjust the quadrant probability and view a corresponding joint sample.

```
elicitQuadProb(myfit1, myfit2, m1 = 0.15, m2 = 0.2, d = c("Beta", "Beta"))
```

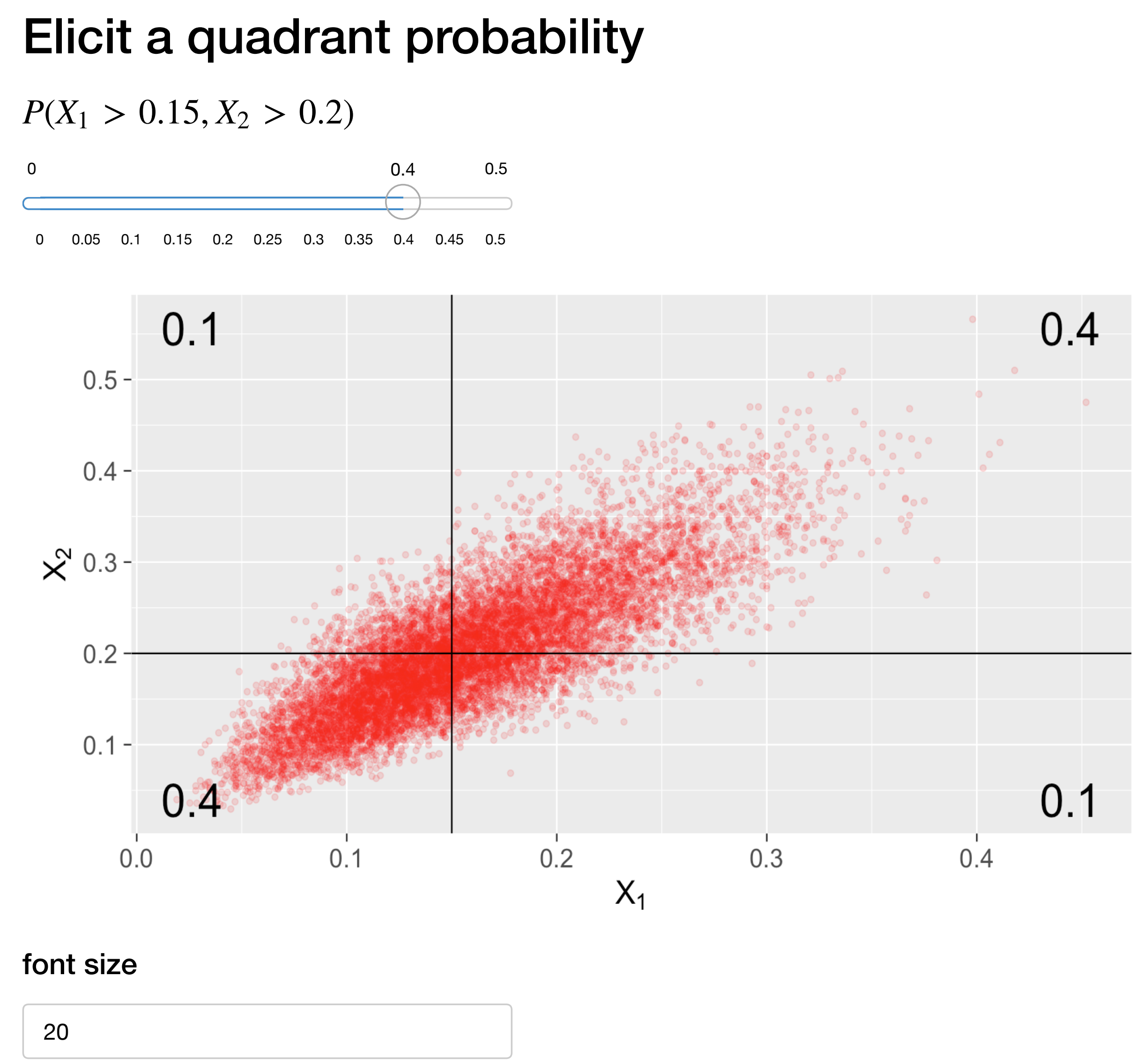

Figure 4: Screenshot of the interactive tool for specifying quadrant probabilities. A new sample is generated each time the (upper right) quadrant probability changes. All four quadrant probabilities are shown.