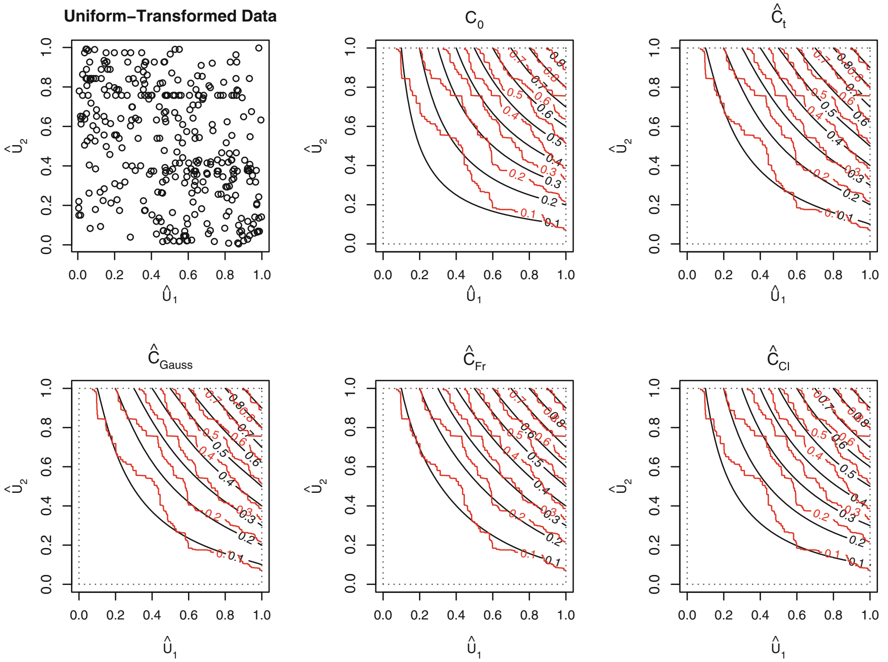

Fig. 8.9. *Uniform-transformed flows for pipeline data. Scatterplot; independence copula contours and four fitted copula contours via parametric models, versus the empirical copula contours.*

fatally flawed—way to assess risk" (Salmon 2009); in particular, the model does not include tail dependence. Duffie and Singleton's (2003, Section 10.4) also discusses copula-based methods for modeling dependent default times.

8.9 R Lab

8.9.1 Simulating from Copula Models

Run the R code that appears below to generate data from a copula. Line 1 loads the `copula` package. Lines 2–3 defines a copula object. At this point, nothing is done with the copula object—it is simply defined. However, the copula object is used in line 5 to generate a random sample from the specified copula model. The remaining lines create a scatterplot matrix of the sample and print its sample Pearson correlation matrix.

```
1 library(copula)
2 cop_t_dim3 = tCopula(dim = 3, param = c(-0.6,0.75,0),
3                         dispstr = "un", df = 1)
4 set.seed(5640)
```

```
5 rand_t_cop = rCopula(n = 500, copula = cop_t_dim3)
6 pairs(rand_t_cop)
7 cor(rand_t_cop)
```

You can use R's help to learn more about the functions `tCopula()` and `rCopula()`.

Problem 1 *Consider the* R *code above.*

(a) What type of copula model has been sampled? Give the copula family, the correlation matrix, and any other parameters that specify the copula.
(b) What is the sample size?

Problem 2 *Examine the scatterplot matrix (generated by line 6) and answer the questions below. Include the scatterplot matrix with your answer.*

(a) Components 2 and 3 are uncorrelated. Do they appear independent? Why or why not?
(b) Do you see signs of tail dependence? If so, where?
(c) What are the effects of dependence upon the plots?
(d) The nonzero correlations in the copula do not have the same values as the corresponding sample correlations. Do you think this is just due to random variation or is something else going on? If there is another cause besides random variation, what might that be? To help answer this question, you can get confidence intervals for the Pearson correlation: For example,

```
8 cor.test(rand_t_cop[,1],rand_t_cop[,3])
```

will give a confidence interval (95 percent by default) for the correlation (Pearson by default) between components 1 and 3. Does this confidence interval include 0.75*?*

Lines 9–10 in the R code below defines a normal (Gaussian) copula. Lines 11–13 define a multivariate distribution by specifying its copula and its marginal distributions—the copula is the one just defined. Line 15 generates a random sample of size 1,000 from this distribution, which has three components. The remaining lines create a scatterplot matrix and kernel estimates of the marginal densities for each component.

```
9  cop_normal_dim3 = normalCopula(dim = 3, param = c(-0.6,0.75,0),
10                                dispstr = "un")
11 mvdc_normal = mvdc(copula = cop_normal_dim3, margins = rep("exp",3),
12                    paramMargins = list(list(rate=2), list(rate=3),
13                                        list(rate=4)))
14 set.seed(5640)
15 rand_mvdc = rMvdc(n = 1000, mvdc = mvdc_normal)
16 pairs(rand_mvdc)
17 par(mfrow = c(2,2))
18 for(i in 1:3) plot(density(rand_mvdc[,i]))
```

Problem 3 *Run the* `R` *code above to generate a random sample.*

(a) What are the marginal distributions of the three components in `rand_mvdc`*? What are their expected values?*
(b) Are the second and third components independent? Why or why not?

8.9.2 Fitting Copula Models to Bivariate Return Data

In this section, you will fit copula models to a bivariate data set of daily returns on IBM stock and the S&P 500 index.

First, you will fit a model with univariate marginal t-distributions and a t-copula. The model has three degrees-of-freedom (tail index) parameters, one for each of the two univariate models and a third for the copula. This means that the univariate distributions can have different tail indices and that their tail indices are independent of the tail dependence from the copula.

Run the following `R` code to load the data and necessary libraries, fit univariate t-distributions to the two components, and convert estimated scale parameters to estimated standard deviations:

```
1  library(MASS)      #  for fitdistr() and kde2d() functions
2  library(copula)    #  for copula functions
3  library(fGarch)    #  for standardized t density
4  netRtns = read.csv("IBM_SP500_04_14_daily_netRtns.csv", header = T)
5  ibm = netRtns[,2]
6  sp500 = netRtns[,3]
7  est.ibm = as.numeric( fitdistr(ibm,"t")$estimate )
8  est.sp500 = as.numeric( fitdistr(sp500,"t")$estimate )
9  est.ibm[2] = est.ibm[2] * sqrt( est.ibm[3] / (est.ibm[3]-2) )
10 est.sp500[2] = est.sp500[2] * sqrt(est.sp500[3] / (est.sp500[3]-2) )
```

The univariate estimates will be used as starting values when the meta-t-distribution is fit by maximum likelihood. You also need an estimate of the correlation coefficient in the t-copula. This can be obtained using Kendall's tau. Run the following code and complete line 12 so that `omega` is the estimate of the Pearson correlation based on Kendall's tau.

```
11 cor_tau = cor(ibm, sp500, method = "kendall")
12 omega =
```

Problem 4 *How did you complete line 12 of the code? What was the computed value of* `omega`*?*

Next, define the t-copula using `omega` as the correlation parameter and 4 as the degrees-of-freedom (tail index) parameter.

```
cop_t_dim2 = tCopula(omega, dim = 2, dispstr = "un", df = 4)
```

Now fit copulas to the uniform-transformed data.

```
data1 = cbind(pstd(ibm, est.ibm[1], est.ibm[2], est.ibm[3]),
              pstd(sp500, est.sp500[1], est.sp500[2], est.sp500[3]))
n = nrow(netRtns) ; n
data2 = cbind(rank(ibm)/(n+1), rank(sp500)/(n+1))
ft1 = fitCopula(cop_t_dim2, data1, method="ml", start=c(omega,4) )
ft2 = fitCopula(cop_t_dim2, data2, method="ml", start=c(omega,4) )
```

Problem 5

(a) Explain the difference between methods used to obtain the two estimates `ft1` *and* `ft2`*.*
(b) Do the two estimates seem significantly different (in a practical sense)?

The next step defines a meta-t-distribution by specifying its t-copula and its univariate marginal distributions. Values for the parameters in the univariate margins are also specified. The values of the copula parameter were already defined in the previous step.

```
mvdc_t_t = mvdc( cop_t_dim2, c("std","std"), list(
           list(mean=est.ibm[1],sd=est.ibm[2],nu=est.ibm[3]),
           list(mean=est.sp500[1],sd=est.sp500[2],nu=est.sp500[3])))
```

Now fit the meta t-distribution. Be patient. This takes awhile; for instance, it took one minute on my laptop. The elapsed time in minutes will be printed.

```
start = c(est.ibm, est.sp500, ft1@estimate)
objFn = function(param) -loglikMvdc(param,cbind(ibm,sp500),mvdc_t_t)
tic = proc.time()
ft = optim(start, objFn, method="L-BFGS-B",
           lower = c(-.1,0.001,2.2, -0.1,0.001,2.2,  0.2,2.5),
           upper = c( .1,   10, 15,  0.1,   10, 15,  0.9, 15) )
toc = proc.time()
total_time = toc - tic ; total_time[3]/60
```

Lower and upper bounds are used to constrain the algorithm to stay inside a region where the log-likelihood is defined and finite. The function `fitMvdc()` in the `copula` package does not allow setting lower and upper bounds and did not converge on this problem.

Problem 6

(a) What are the estimates of the copula parameters in `fit_cop`*?*
(b) What are the estimates of the parameters in the univariate marginal distributions?

(c) Was the estimation method maximum likelihood, semiparametric pseudo-maximum likelihood, or parametric pseudo-maximum likelihood?
(d) Estimate the coefficient of lower tail dependence for this copula.

Now fit normal (Gaussian), Frank, Clayton, Gumbel and Joe copulas to the data.

```
fnorm = fitCopula(copula=normalCopula(dim=2),data=data1,method="ml")
ffrank = fitCopula(copula = frankCopula(3, dim = 2),
                     data = data1, method = "ml")
fclayton = fitCopula(copula = claytonCopula(1, dim=2),
                     data = data1, method = "ml")
fgumbel = fitCopula(copula = gumbelCopula(3, dim=2),
                    data = data1, method = "ml")
fjoe = fitCopula(copula=joeCopula(2,dim=2),data=data1,method="ml")
```

The estimated copulas (CDFs) will be compared with the empirical copula.

```
Udex = (1:n)/(n+1)
Cn = C.n(u=cbind(rep(Udex,n),rep(Udex,each=n)), U=data1, method="C")
EmpCop = expression(contour(Udex, Udex, matrix(Cn, n, n),
                            col = 2, add = TRUE))
par(mfrow=c(2,3),  mgp = c(2.5,1,0))
contour(tCopula(param=ft$par[7],dim=2,df=round(ft$par[8])),
        pCopula, main = expression(hat(C)[t]),
        xlab = expression(hat(U)[1]), ylab = expression(hat(U)[2]) )
eval(EmpCop)
contour(normalCopula(param=fnorm@estimate[1], dim = 2),
        pCopula, main = expression(hat(C)[Gauss]),
        xlab = expression(hat(U)[1]), ylab = expression(hat(U)[2]) )
eval(EmpCop)
contour(frankCopula(param=ffrank@estimate[1], dim = 2),
        pCopula, main = expression(hat(C)[Fr]),
        xlab = expression(hat(U)[1]), ylab = expression(hat(U)[2]) )
eval(EmpCop)
contour(claytonCopula(param=fclayton@estimate[1], dim = 2),
        pCopula, main = expression(hat(C)[Cl]),
        xlab = expression(hat(U)[1]), ylab = expression(hat(U)[2]) )
eval(EmpCop)
contour(gumbelCopula(param=fgumbel@estimate[1], dim = 2),
        pCopula, main = expression(hat(C)[Gu]),
        xlab = expression(hat(U)[1]), ylab = expression(hat(U)[2]) )
eval(EmpCop)
contour(joeCopula(param=fjoe@estimate[1], dim = 2),
        pCopula, main = expression(hat(C)[Joe]),
        xlab = expression(hat(U)[1]), ylab = expression(hat(U)[2]) )
eval(EmpCop)
```

Problem 7 *Do you see any difference between the parametric estimates of the copula? If so, which seem closest to the empirical copula? Include the plot with your work.*

A two-dimensional KDE of the copula's density will be compared with the parametric density estimates (PDFs).

```
par(mfrow=c(2,3),  mgp = c(2.5,1,0))
contour(tCopula(param=ft$par[7],dim=2,df=round(ft$par[8])),
        dCopula, main = expression(hat(c)[t]),
  nlevels=25, xlab=expression(hat(U)[1]),ylab=expression(hat(U)[2]))
contour(kde2d(data1[,1],data1[,2]), col = 2, add = TRUE)
contour(normalCopula(param=fnorm@estimate[1], dim = 2),
        dCopula, main = expression(hat(c)[Gauss]),
  nlevels=25, xlab=expression(hat(U)[1]),ylab=expression(hat(U)[2]))
contour(kde2d(data1[,1],data1[,2]), col = 2, add = TRUE)
contour(frankCopula(param=ffrank@estimate[1], dim = 2),
        dCopula, main = expression(hat(c)[Fr]),
  nlevels=25, xlab=expression(hat(U)[1]),ylab=expression(hat(U)[2]))
contour(kde2d(data1[,1],data1[,2]), col = 2, add = TRUE)
contour(claytonCopula(param=fclayton@estimate[1], dim = 2),
        dCopula, main = expression(hat(c)[Cl]),
  nlevels=25, xlab=expression(hat(U)[1]),ylab=expression(hat(U)[2]))
contour(kde2d(data1[,1],data1[,2]), col = 2, add = TRUE)
contour(gumbelCopula(param=fgumbel@estimate[1], dim = 2),
        dCopula, main = expression(hat(c)[Gu]),
  nlevels=25, xlab=expression(hat(U)[1]),ylab=expression(hat(U)[2]))
contour(kde2d(data1[,1],data1[,2]), col = 2, add = TRUE)
contour(joeCopula(param=fjoe@estimate[1], dim = 2),
        dCopula, main = expression(hat(c)[Joe]),
  nlevels=25, xlab=expression(hat(U)[1]),ylab=expression(hat(U)[2]))
contour(kde2d(data1[,1],data1[,2]), col = 2, add = TRUE)
```

Problem 8 *Do you see any difference between the parametric estimates of the copula density? If so, which seem closest to the KDE? Include the plot with your work.*

Problem 9 *Find AIC for the t, (Gaussian), Frank, Clayton, Gumbel and Joe copulas. Which copula model fits best by AIC? (Hint: The* `fitCopula()` *function returns the log-likelihood.)*

8.10 Exercises

1. Kendall's tau rank correlation between X and Y is 0.55. Both X and Y are positive. What is Kendall's tau between X and $1/Y$? What is Kendall's tau between $1/X$ and $1/Y$?

2. Suppose that X is Uniform(0,1) and $Y = X^2$. Then the Spearman rank correlation and the Kendall's tau between X and Y will both equal 1, but the Pearson correlation between X and Y will be less than 1. Explain why.
3. Show that an Archimedean copula with generator function $\varphi(u) = -\log(u)$ is equal to the independence copula C_0. Does the same hold when the natural logarithm is replaced by the common logarithm, i.e., $\varphi(u) = -\log_{10}(u)$?
4. The co-monotonicity copula C_+ is not an Archimedean copula; however, in the two-dimensional case, the counter-monotonicity copula $C_-(u_1, u_2) = \max(u_1 + u_2 - 1, 0)$ is. What is its generator function?
5. Show that the generator of a Frank copula

$$\varphi_{\mathrm{Fr}}(u|\theta) = -\log\left\{\frac{e^{-\theta u} - 1}{e^{-\theta} - 1}\right\}, \quad \theta \in \{(-\infty, 0) \cup (0, \infty)\},$$

satisfies assumptions 1–3 of a strict generator.
6. Show that as $\theta \to \infty$, $C_{\mathrm{Fr}}(u_1, u_2|\theta) \to \min(u_1, u_2)$, the co-monotonicity copula C_+.
7. Suppose that $\varphi_1, \ldots, \varphi_k$ are k strict generator functions and define a new generator φ as a convex combination of these k generators, that is

$$\varphi(u) = a_1\varphi_1(u) + \cdots + a_k\varphi_k(u),$$

in which $a_1, \ldots, a_k$ are any non-negative constants which sum to 1. Show that $\varphi(u)$ is a strict generator function. For the case in which $k = 2$, what is the corresponding copula function for $\varphi(u)$?
8. Let $\varphi(u|\theta) = (1 - u)^\theta$, for some $\theta \geq 1$, and show that for the two-dimensional case this generates the copula

$$C(u_1, u_2|\theta) = \max[0, 1 - \{(1 - u_1)^\theta + (1 - u_2)^\theta\}^{1/\theta}].$$

Further, show that as $\theta \to \infty$, $C(u_1, u_2|\theta) \to \min(u_1, u_2)$, the co-monotonicity copula C_+, and that as $\theta \to 1$, $C(u_1, u_2|\theta) \to \max(u_1 + u_2 - 1, 0)$, the counter-monotonicity copula C_-.
9. A convex combination of k joint CDFs is itself a joint CDF (finite mixture), but is a convex combination of k copula functions a copula function itself?
10. Suppose $\mathbf{Y} = (Y_1, \ldots, Y_d)$ has a meta-Gaussian distribution with continuous marginal distributions and copula $C^{Gauss}(\cdot|\mathbf{\Omega})$. Show that if $\rho_\tau(Y_i, Y_j) = 0$ then Y_i and Y_j are independent.

References

Cherubini, U., Luciano, E., and Vecchiato, W. (2004) *Copula Methods in Finance*, John Wiley, New York.