

Examples of vines and vine copulas
 Harry Joe
 July 24/31, 2015; Graphical models reading group

Outline

- Examples with correlation matrices
 - Example showing continuous variables and non-Gaussian dependence
 - Example with discrete response variables and polychoric correlation matrix
-

Dependence structure

\mathbf{R}_{data} is sample correlation matrix (possibly of normal scores or polychoric correlation matrix).

$\mathbf{R}_{\text{model}}(\boldsymbol{\delta})$ is parsimonious model; $\dim(\boldsymbol{\delta}) < d(d-1)/2$.

Fit (parsimonious) parametric models to correlation matrix \mathbf{R} (variances are considered as univariate parameters); simplest models are: exchangeable, 1-factor, Markov tree.

A common discrepancy measure is:

$$D_{\text{model}} = \log(\det[\mathbf{R}_{\text{model}}(\hat{\boldsymbol{\delta}})]) - \log(\det[\mathbf{R}_{\text{data}}]) + \text{tr}[\mathbf{R}_{\text{model}}^{-1}(\hat{\boldsymbol{\delta}}) \mathbf{R}_{\text{data}}] - d,$$

If the dependence model has some conditional independence relations, then the dimension of $\boldsymbol{\delta}$ is less than $d(d-1)/2$. Other comparisons are the **AIC/BIC based on a Gaussian log-likelihood**. Also useful are the average and max absolute deviations of the model-based correlation matrix from the empirical correlation matrix:

average _{$j < k$} $|\mathbf{R}_{\text{model},jk}(\hat{\boldsymbol{\delta}}) - \mathbf{R}_{\text{data},jk}|$ and max _{$j < k$} $|\mathbf{R}_{\text{model},jk}(\hat{\boldsymbol{\delta}}) - \mathbf{R}_{\text{data},jk}|$.

Non-Gaussian dependence

If bivariate normal scores plot suggest deviations from Gaussian, find a copula extension of the parsimonious dependence (e.g., truncated vine, common factor, Bayesian network) using parametric copulas with tail asymmetry and/or tail dependence.

Useful graphical models have the same form for Gaussian and non-Gaussian:

Gaussian graphical models have partial correlations on edges;

non-Gaussian graphical models have bivariate copulas on edges.

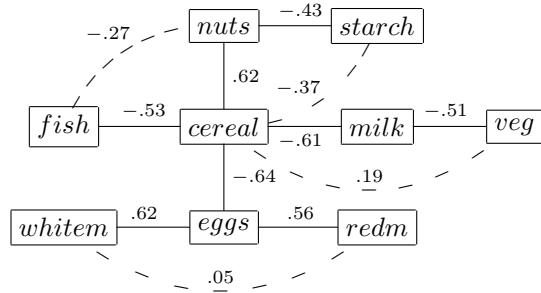
Example in Wagaman and Levina 2009

Correlation of normal scores for protein data $n = 25$ European countries, measurements in grams per capita per day

	RedMeat	WhiteMeat	Eggs	Milk	Fish	Cereals	Starch	Nuts	Fr.Veg
RedMeat	1.0000	0.0552	0.5627	0.559	0.218	-0.500	0.0547	-0.305	-0.1784
WhiteMeat	0.0552	1.0000	0.6163	0.316	-0.155	-0.403	0.2340	-0.598	0.0611
Eggs	0.5627	0.6163	1.0000	0.559	0.212	-0.641	0.3562	-0.471	0.0206
Milk	0.5587	0.3164	0.5591	1.000	0.219	-0.609	0.1430	-0.592	-0.5104
Fish	0.2179	-0.1549	0.2122	0.219	1.000	-0.530	0.4147	-0.275	0.1845
Cereals	-0.4999	-0.4027	-0.6413	-0.609	-0.530	1.000	-0.3747	0.622	0.1889
Starch	0.0547	0.2340	0.3562	0.143	0.415	-0.375	1.0000	-0.435	0.0676
Nuts	-0.3051	-0.5975	-0.4710	-0.592	-0.275	0.622	-0.4349	1.000	0.4049
Fr.Veg	-0.1784	0.0611	0.0206	-0.510	0.185	0.189	0.0676	0.405	1.0000

Markov tree with edges connecting variables with large absolute correlations. Markov tree parsimonious dependence model (if valid) would imply that $\rho_{jk} = \rho_{jm}\rho_{mk}$ if the path from variables j to k in the tree is

through a single variable m . In general, the ρ_{jk} is the product of the correlations along the edges of the path that connects variables j and k .



Markov tree or 1-truncated vine, embedded in graph, minimum spanning tree is best tree inside complete graph, $\log(1 - r_{jk}^2)$ on edges of complete graph.

3-Truncated vine for not as good as 3-factor

	Factor1	Factor2	Factor3	
RedMeat	0.102	0.511	-0.235	factor 1: contrast whitem, eggs vs nuts, cereals
WhiteMeat	0.994			factor 2: contrast fish, redm, eggs, milk, starch vs cereals, nuts
Eggs	0.655	0.432		
Milk	0.357	0.417	-0.575	
Fish		0.835	0.114	
Cereals	-0.463	-0.699	0.283	factor 3: contrast veg, nuts vs milk
Starch	0.269	0.421		
Nuts	-0.630	-0.335	0.477	
Fr.Veg			0.991	

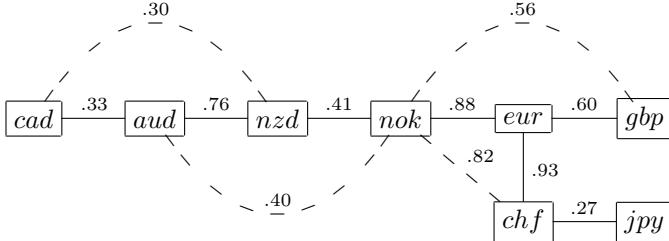
[FX example](#), years 2000–2002, $n = 780$

Correlation matrix of normal scores of log returns of exchange rates (relative to USD).

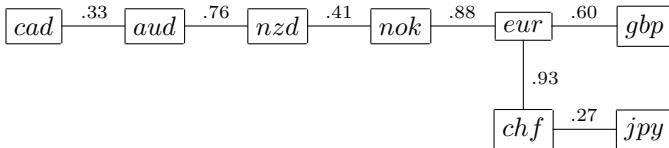
	EUR	AUD	CAD	CHF	GBP	JPY	NOK	NZD
EUR	1.000	0.403	0.134	0.930	0.603	0.239	0.876	0.408
AUD	0.403	1.000	0.332	0.361	0.273	0.148	0.397	0.756
CAD	0.134	0.332	1.000	0.120	0.079	0.127	0.147	0.296
CHF	0.930	0.361	0.120	1.000	0.587	0.274	0.824	0.379
GBP	0.603	0.273	0.079	0.587	1.000	0.242	0.558	0.279
JPY	0.239	0.148	0.127	0.274	0.242	1.000	0.234	0.125
NOK	0.876	0.397	0.147	0.824	0.558	0.234	1.000	0.410
NZD	0.408	0.756	0.296	0.379	0.279	0.125	0.410	1.000

Loading matrix for bi-factor (without JPY); (algebraically independent) partial correlation parametrization

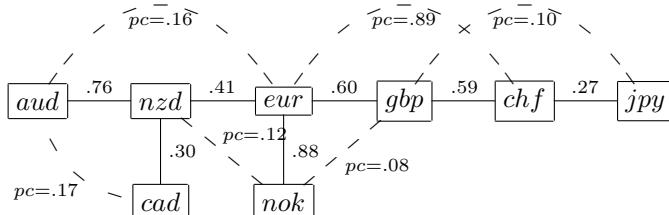
	loading matrix			partial correlations		
EUR	0.914	0.340	0.000	0.914	0.838	0.000
CHF	0.832	0.498	0.000	0.832	0.899	0.000
GBP	0.588	0.195	0.000	0.588	0.241	0.000
NOK	0.908	0.138	0.000	0.908	0.329	0.000
AUD	0.442	0.000	0.806	0.442	0.000	0.899
CAD	0.154	0.000	0.327	0.154	0.000	0.331
NZD	0.448	0.000	0.692	0.448	0.000	0.774



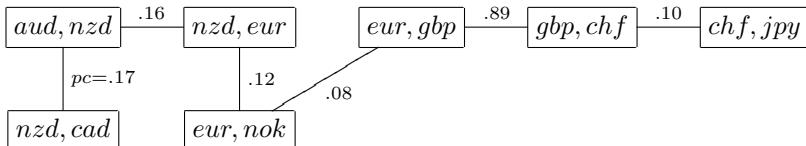
Markov tree or 1-truncated vine, embedded in graph, minimum spanning tree is best tree inside complete graph, $\log(1 - r_{jk}^2)$ on edges of complete graph.



Markov tree or 1-truncated vine, not a good fit (too parsimonious); chf $\perp\!\!\!\perp$ gbp given eur etc.. Note that product along edges not a good approx.



2-truncated vine with additional conditional dependence, 8 currencies, good fit; tree 1 is a modification of minimum spanning tree Partial correlation nzd,nok|eur is 0.12 etc. If 2-truncated vine does not explain all dependence, add more edges of form aud,gbp|nzd,gbp. Solid edges: tree1; dashed edges: tree2, etc. Vine is a sequence of trees satisfying some proximity conditions.



Tree 2 of above 2-truncated vine; edges of tree1 are nodes of tree2. Draw graph of all edges that satisfy proximity condition — also have [nzd,eur]–[eur,gbp], partial correlation: 0.05; [nzd,cad]–[nzd,eur], partial correlation: 0.02

Summary for correlation matrices of 1-factor and Markov tree (the two simplest structures)
1-factor

$$\begin{pmatrix} 1 & \alpha_1\alpha_2 & \alpha_1\alpha_3 & \cdots & \alpha_1\alpha_d \\ \alpha_2\alpha_1 & 1 & \alpha_2\alpha_3 & \cdots & \alpha_2\alpha_d \\ \alpha_3\alpha_1 & \alpha_3\alpha_2 & 1 & \cdots & \alpha_3\alpha_d \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_d\alpha_1 & \alpha_d\alpha_2 & \alpha_d\alpha_3 & \cdots & 1 \end{pmatrix} \quad \text{compare rows } j, k, \text{ in other columns: similarly ordered.}$$

Markov tree

$$\begin{pmatrix} \dots & j & \dots & k & \dots & m & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & 1 & \dots & \rho_{jk} & \dots & \rho_{jm} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & \rho_{kj} & \dots & 1 & \dots & \rho_{km} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & \rho_{mj} & \dots & \rho_{mk} & \dots & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix} \quad \text{for any triple, weakest correlation is product of other 2}$$

Going from Gaussian vine to vine copula: replace correlations in tree1 by bivariate copulas; replace partial correlations (pc) in trees 2 and higher by bivariate copulas with conditional distributions as argument. To get asymmetric tail dependence for all bivariate margins, use BB1 or reflected BB1 on tree 1, and Gaussian or Frank etc for trees 2 and higher.

Optimization for truncated Gaussian vines

Markov tree negative (Gaussian) log-likelihood is: $\sum_{e \in \mathcal{T}_1} \log(1 - \rho_e^2)$. Hence one needs to optimize this over trees.

For ℓ -truncated vine, negative log-likelihood is: $\sum_{e \in \mathcal{T}_1, \dots, \mathcal{T}_\ell} \log(1 - \rho_e^2)$. Hence one needs to optimize this over ℓ -truncated vines.

Proof assumes that variables have been standardized to have means of 0 and variances of 1. (Brechmann and Joe, 2014, CSDA).

Fix a Markov tree, WLOG let the pairs in the tree be $\{1, 2\}$ and $\{j, k(j)\}$ for $j = 3, \dots, d$, where $1 \leq k(j) < j$. The regression model is:

$$\begin{aligned} Z_1 &= \psi_1 \varepsilon_1, \\ Z_2 &= \varphi_2 Z_1 + \psi_2 \varepsilon_2, \\ Z_j &= \varphi_j Z_{k(j)} + \psi_j \varepsilon_j, \quad j = 3, \dots, d, \end{aligned} \quad (1)$$

where $\varphi_1 = 0$ and $\psi_j = \sqrt{1 - \varphi_j^2}$ for all $j = 1, \dots, d$, and $\varphi_j = \rho_{j, k(j)}$.

Based on a product of conditional regression-type densities $f_{Z_j|Z_{k(j)}}$, $j \geq 2$,

$$f_{\mathbf{z}}(\mathbf{z}) = f_{Z_1}(z_1) \prod_{j=2}^d f_{Z_j|Z_{k(j)}}(z_j|z_{k(j)}), \quad \mathbf{z} = (z_1, \dots, z_d)',$$

the log-likelihood of the Markov tree model (1) with correlation matrix $\Sigma = \Sigma(\rho_{j, k(j)}, j = 2, \dots, d)$ has the form

$$\begin{aligned} L(\rho_{j, k(j)}, j = 2, \dots, d) &= -\frac{n}{2} \log |\Sigma| - \frac{1}{2} \sum_{i=1}^n \mathbf{z}'_i \Sigma^{-1} \mathbf{z}_i \\ &= -\frac{n}{2} \sum_{j=2}^d \log(1 - \rho_{j, k(j)}^2) - \frac{1}{2} \left\{ \sum_{i=1}^n z_{i1}^2 + \sum_{j=2}^d \sum_{i=1}^n \frac{(z_{ij} - \rho_{j, k(j)} z_{i, k(j)})^2}{(1 - \rho_{j, k(j)}^2)} \right\}, \end{aligned} \quad (2)$$

where $|\cdot|$ denotes the matrix determinant and constant terms with 2π have been omitted.

Maximizing the above reduces to $d - 1$ separate one-parameter optimizations of the form

$$L_j(\rho) := -\frac{n}{2} \log(1 - \rho^2) - \frac{1}{2} \sum_{i=1}^n (z_{ij} - \rho z_{i,k(j)})^2 / (1 - \rho^2), \quad \rho = \rho_{j,k(j)}.$$

With scaled variables, assume that $1 = \frac{1}{n} \sum_{i=1}^n z_{ij}^2 = \frac{1}{n} \sum_{i=1}^n z_{i,k(j)}^2$ and that correlations are sums of cross-products, this simplifies to

$$L_j(\rho) = -\frac{n}{2} \log(1 - \rho^2) - \frac{n}{2} (1 + \rho^2 - 2\rho r_{j,k(j)}) / (1 - \rho^2)$$

with derivative

$$L'_j(\rho) = \frac{n\rho}{1 - \rho^2} - \frac{n(\rho - r_{j,k(j)})}{(1 - \rho^2)} - \frac{n\rho(1 + \rho^2 - 2\rho r_{j,k(j)})}{(1 - \rho^2)^2}.$$

Hence with a fixed tree, the MLE of (2) has $\hat{\rho}_{j,k(j)} = r_{j,k(j)}$ for $j = 2, \dots, d$. Substitution of $\hat{\rho}_{j,k(j)} = r_{j,k(j)}$ back into the log-likelihood (2) leads to

$$\begin{aligned} L(\hat{\rho}_{j,k(j)}, j = 2, \dots, d) &= -\frac{n}{2} \sum_{j=2}^d \log(1 - r_{j,k(j)}^2) - \frac{n}{2} \left\{ 1 + \sum_{j=2}^d (1 + r_{j,k(j)}^2 - 2r_{j,k(j)}^2) / (1 - r_{j,k(j)}^2) \right\} \\ &= -\frac{n}{2} \sum_{j=2}^d \log(1 - r_{j,k(j)}^2) - \frac{n}{2} d. \end{aligned} \quad (3)$$

Optimizing over all possible Markov trees therefore reduces to finding the tree T with $d - 1$ edges $\{e = \{j, k\}\}$ that minimizes

$$\sum_{e=\{j, k\} \in T} \log(1 - r_{jk}^2). \quad (4)$$

The optimal solution can easily be found using minimum spanning tree algorithms, which is guaranteed to find the optimal solution when edge weights between nodes $1 \leq k < j \leq d$ are given by $\log(1 - r_{jk}^2)$. Estimating Markov tree models (1) boils down to a simple model selection problem.

Same thing for the extension to 2-truncated vines, and ℓ -truncated vines: the negative log-likelihood is: $\sum_{e \in \mathcal{T}_1, \dots, \mathcal{T}_\ell} \log(1 - \rho_e^2)$. For a product of conditional normal (regressions), the exponent part of the likelihood becomes a constant with the MLE for a fixed truncated vine.

Algorithms for Gaussian vines or vine copulas.

1. Algorithm of Dissmann et al. Absolute Kendall's tau is used on edges of vines for sequential minimum spanning tree, and bivariate copulas on edges are optimized over a given set in parametric copulas. (Not completely flexible, user cannot add copula families like bivariate skew-normal and bivariate skew-t)

2. Algorithm in Brechmann and Joe (2015). $\log(1 - \rho_e^2)$ on edges of vine (correlations on tree 1, and partial correlations on remaining trees). Best minimum spanning trees at each level of vine combined with genetic algorithm for a non-greedy approach. After good truncated vines are found, bivariate copulas on edges are optimized over a given set in parametric copulas.

3. Enumeration algorithm in Joe (2014) and in CopulaModel R package. With $\log(1 - \rho_e^2)$ on edges of vine, the best 1-truncated to $(d - 2)$ -truncated Gaussian vines are found by enumeration. Number of vines is $2^{(d-3)(d-2)/2} \cdot d! / 2$ so enumeration is fast for $4 \leq d \leq 7$ and possible for $d = 8$. For $d = 3$, enumerate the 3 possible vines directly.

Vine copula density (with truncation level $\ell > 1$) has the form

$$c_{1:d} = \prod_{e=(j,k) \in \mathcal{T}_1} c_{jk} \prod_{e=[jk;S] \in \mathcal{T}_2, \dots, \mathcal{T}_\ell} c_{jk;S}(C_{j|S}, C_{k|S})$$

or

$$f_{1:d} = f_1 \cdots f_d \prod_{e=(j,k) \in \mathcal{T}_1} c_{jk}(F_j, F_k) \prod_{e=[jk;S] \in \mathcal{T}_2, \dots, \mathcal{T}_\ell} c_{jk;S}(F_{j|S}, F_{k|S})$$

Diagnostics for vine copulas: Chapter 5 of Joe (1014).

After tree \mathcal{T}_1 , one can create pseudo-observations of the form $(v_{ij_1}, v_{ij_2}) = (C_{j_1|k}(u_{ij_1}|u_{ik}; \tilde{\theta}_{j_1k}), C_{j_2|k}(u_{ij_2}|u_{ik}; \tilde{\theta}_{j_2k}))$ for $i = 1, \dots, n$, assuming (j_1, k) and (j_2, k) are edges in tree \mathcal{T}_1 and $\tilde{\theta}_{j_1k}, \tilde{\theta}_{j_2k}$ are estimates based on maximum likelihood with only tree \mathcal{T}_1 . For any such pair of such pseudo-variables, diagnostics such as bivariate normal scores plots and semi-correlations can be used to assess tail asymmetry and strength of dependence in the tails. This can be continued to higher-order trees.

More details for specific D-vine 1-2-3 for data (x_{i1}, x_{i2}, x_{i3}) , $i = 1, \dots, n$. Joint density is:

$$f_{123} = f_1 f_2 f_3 c_{12}(F_1, F_2) c_{23}(F_2, F_3) c_{13;2}(F_{1|2}, F_{3|2})$$

A copula for conditional distributions is:

$F_{13|2}(x_1, x_3|x_2) = C_{13;2}(F_{1|2}(x_1|x_2), F_{3|2}(x_3|x_2); x_2)$. Check conditional Spearman rho $\rho_S(F_{13|2}(\cdot|x_2))$ as a function of x_2 with the method of Gijbels, Veraverbeke et al (2011) [kernel smoothing, 5chapter/gammafactor/rho12gx2.r]

If dependence doesn't vary much, assume the simplifying assumption with a model $C_{13;2}(F_{1|2}(x_1|x_2; \theta_{12}), F_{3|2}(x_3|x_2; \theta_{32}); \theta_{13})$

Fit $F_{12}(x_1, x_2) = C_{12}(F_1(x_1), F_2(x_2); \theta_{12})$, and $F_{32}(x_3, x_2) = C_{32}(F_3(x_3), F_2(x_2); \theta_{32})$, with estimators $\hat{\theta}_{12}, \hat{\theta}_{32}$.

Get uniform pseudo-observations

$$\hat{u}_{i;1|2} = F_{1|2}(x_{i1}|x_{i2}; \hat{\theta}_{12}), \quad \hat{u}_{i;3|2} = F_{3|2}(x_{i3}|x_{i2}; \hat{\theta}_{32}).$$

and look at normal scores plot, semi-correlations etc.

My approach (small d): from tail asymmetry/dependence based on semi-correlations, try a few suitable copulas for edges of vine and take the best ones based on AIC.

VineCopula R package: optimize log-likelihood over a big set of parametric bivariate copula families (Gumbel, survival Gumbel, Gauss, t, Frank etc) at each edge.

Abalone male data, from

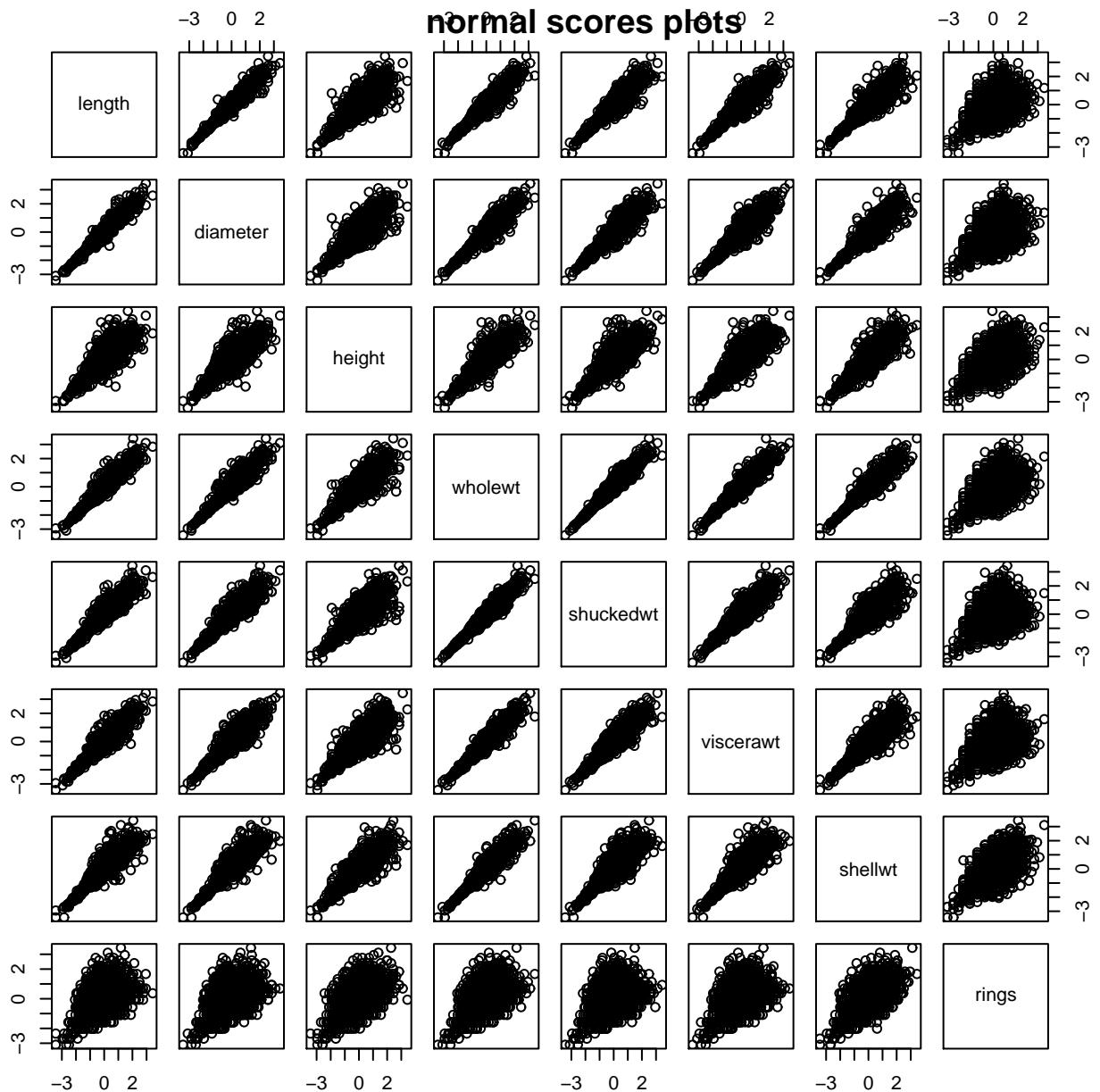
<http://archive.ics.uci.edu/ml/datasets.html>. $d = 7$ variables with sample size $n = 1527$ (after one omission). The variables x_1, \dots, x_7 are the following.

1. Length in mm for the longest shell measurement.
2. Diameter in mm perpendicular to length.
3. Height in mm of the meat in shell.
4. Whole weight in grams for the whole abalone.
5. Shucked weight in grams for the weight of meat.
6. Viscera weight in grams for the gut weight (after bleeding).
7. Shell weight in grams after being dried.
8. Rings: integer, +1.5 gives the age in years

Bivariate normal scores plots: joint tail asymmetry skewed to lower tail with lower tail dependence, so multivariate Gaussian is inappropriate. Univariate models: skew-normal for measurements in mm, generalized gamma for measurements in grams. ?? for age (which is actually discretized). With $f_\Gamma(\cdot; \gamma)$ as the $\text{Gamma}(\gamma, 1)$ density,

$$f_{SN}(y; \xi, \omega, \alpha) = 2\omega^{-1}\phi([y - \xi]/\omega)\Phi(\alpha[y - \xi]/\omega), \quad -\infty < y < \infty, -\infty < \xi < \infty, \omega > 0, -\infty < \alpha < \infty,$$

$$f_{GG}(y; \gamma, \zeta, \sigma) = \frac{\zeta(y/\sigma)^{\zeta\gamma}}{y\Gamma(\gamma)} \exp\{-(y/\sigma)^\zeta\} = (\zeta/\sigma) \cdot (y/\sigma)^{\zeta-1} f_\Gamma((y/\sigma)^\zeta; \gamma), \quad y > 0, \gamma > 0, \zeta > 0, \sigma > 0.$$



Bivariate normal scores plots and semi-correlations show tail asymmetry and lower tail dependence for all pairs.

Diagnostics for tail asymmetry. The semi-correlations (correlations of normal scores in lower/upper quadrants) are included below.

j1	j2	ncor	lcor	ucor	bvnsemic	j1	j2	ncor	lcor	ucor	bvnsemic		
1	2	0.973	0.974	0.891	0.930	tree1	1	7	0.907	0.926	0.687	0.782	tree1
1	3	0.832	0.832	0.515	0.646		2	7	0.918	0.933	0.728	0.805	

```

2 3 0.844 0.845 0.550 0.666      3 7 0.881 0.873 0.625 0.732
1 4 0.957 0.960 0.834 0.892      4 7 0.946 0.950 0.803 0.865 tree1
2 4 0.953 0.957 0.824 0.881      5 7 0.865 0.885 0.591 0.703 tree1
3 4 0.872 0.861 0.633 0.716      6 7 0.889 0.907 0.635 0.748
1 5 0.938 0.940 0.762 0.847      1 8 0.419 0.480 -0.084 0.209
2 5 0.925 0.930 0.726 0.819      2 8 0.444 0.494 -0.016 0.227
3 5 0.818 0.814 0.527 0.625 tree1 3 8 0.500 0.478 0.096 0.269
4 5 0.967 0.956 0.877 0.915      4 8 0.457 0.481 -0.046 0.236
1 6 0.929 0.934 0.762 0.829      5 8 0.328 0.423 -0.158 0.151
2 6 0.918 0.923 0.734 0.806      6 8 0.420 0.489 -0.073 0.210
3 6 0.847 0.839 0.574 0.672      7 8 0.573 0.527 0.166 0.331 tree1
4 6 0.959 0.953 0.850 0.896
5 6 0.922 0.911 0.730 0.814 tree1

```

Best truncated vines (level 1, 2, 3, ...) for Gaussian vines via enumeration of vines applied to correlation matrix of normal scores.

Candidate bivariate 2-parameter copula families for tree 1 of best 3-truncated vine: BB1, reflected BB1, BB7, reflected Gumbel, imitlefA (Archimedean copula based in integrated Mittag-Leffler LT, lower tail dependence, upper intermediate tail dependence).

BB1 is best or near best on all bivariate margins in tree 1.

Checked conditional Spearman's rho; suggest simplifying assumption OK

tree 2 pseudo-observations: semi-correlations do not suggest much tail asymmetry or tail dependence: candidates Gauss, Frank, survival/reflected Gumbel.

Tail dependence results. To get lower tail dependence for all bivariate margins of a vine copula, it suffices to have copulas with lower tail dependence in tree 1.

Some R code for pseudo-observations

```

ab=read.csv("abalone.data",head=F)
abm=subset(ab,V1=="M")
# n=1528
adat=abm[,2:9]; adat=as.matrix(adat)
uu=uscore(adat)
d=8; nn=nrow(adat)

perm=c(5,7,6,1,4,3,2,8)
udat=uu[,perm]

A=matrix(0,d,d)
A[1,]=c(1,1,1,2,2,1,4,2); A[2,2:8]=c(2,2,1,1,3,2,1)
A[3,3:8]=c(3,3,3,2,1,5); A[4,4:8]=c(4,4,4,3,3)
A[5,5:8]=c(5,5,5,4); A[6,6:8]=c(6,6,6)
A[7,7:8]=c(7,7); A[8,8]=8
pcondnames1=rep("pcondbb1",d-1)
qcondnames1=rep("qcondbb1",d-1)
logdcopnames1=rep("logdbb1",d-1)
pcondnames=c("pcondbb1","pcondbvncop","pcondbvncop","pcondbvncop")
qcondnames=c("qcondbb1","qcondbvncop","qcondbvncop","qcondbvncop")
logdcopnames=c("logdbb1","logdbvncop","logdbvncop","logdbvncop")
np=matrix(1,d,d)
np[1,2:d]=2
cpar.bb1=bb1.lm2cpar(c(.7,.3))

```

```

parvec1=rep(cpar.bb1,d-1)

#1-truncated
#BB1 R-vine
mle1=nlm(rvinenllk.trunc,p=parvec1,
  udat=udat,A=A,logdcopnames=logdcopnames[1],pcondnames=pcondnames[1],np,
  ifixed=rep(F,14),parfixed=NULL,
  hessian=T,iterlim=100,print.level=1,LB=rep(c(0,1),7),UB=20)
pseud1=rvinenllkpseud(mle1$estimate,udat,A,logdcopnames[1], pcondnames[1],np)
names(pseud1)
#[1] "nllk"      "condforw"   "condbackw"

zdatf=nscore(pseud1$condforw)
zdatb=nscore(pseud1$condback)
# specific of interest for tree 2 of A
# 76|5 forw[,1] forw[,2]
# 51|7 back[,1] forw[,3]
# 54|7 back[,1] forw[,4]
# 63|5 forw[,2] forw[,5]
# 72|1 back[,3] forw[,6]
# 58|7 back[,1] forw[,7]

# ncorr lcorr ucorr bvnsemicor

par(mfrow=c(3,3))
semic=semicor(zdatf[,c(1,2)]); plot(zdatf[,c(1,2)])
ze=bvnsemic(semic[1])
cat("forw1","forw2",semic,ze,"\n")
# forw1 forw2 0.4924551 0.4047709 0.2433944 0.2628247

semic=semicor(cbind(zdatb[,1],zdatf[,3])); plot(zdatb[,1],zdatf[,3])
ze=bvnsemic(semic[1])
cat("back1","forw3",semic,ze,"\n")
# back1 forw3 0.7061517 0.6139989 0.419965 0.4700388

semic=semicor(cbind(zdatb[,1],zdatf[,4])); plot(zdatb[,1],zdatf[,4])
ze=bvnsemic(semic[1])
cat("back1","forw4",semic,ze,"\n")
# back1 forw4 0.9189503 0.8317676 0.8245751 0.8072359

semic=semicor(zdatf[,c(2,5)]); plot(zdatf[,c(2,5)])
ze=bvnsemic(semic[1])
cat("forw2","forw5",semic,ze,"\n")
# forw2 forw5 0.4112894 0.2189911 0.2186511 0.2039755

semic=semicor(cbind(zdatb[,3],zdatf[,6])); plot(zdatb[,3],zdatf[,6])
ze=bvnsemic(semic[1])
cat("back3","forw6",semic,ze,"\n")
# back3 forw6 0.4022111 0.3156196 0.1996521 0.1979017

semic=semicor(cbind(zdatb[,1],zdatf[,7])); plot(zdatb[,1],zdatf[,7])

```

```

ze=bvnsemic(semic[1])
cat("back1","forw7",semic,ze,"\n")
# back1 forw7 -0.4382905 0.04882022 -0.169017 -0.1232321

# slight negative asymmetry seen in plots 1,2,5
# but for simplicity use Gaussian for tree 2
# =====

#2-truncated

parvec2=c(mle1$estimate,.5,.7,.9,.4,.4,-.4)
mle2=nlm(rvinenllk.trunc,p=parvec2,
          udat=udat,A=A,logdcopnames=logdcopnames[1:2],pcondnames=pcondnames[1:2],np,
          ifixed=rep(F,20),parfixed=NULL,
          hessian=T,iterlim=200,print.level=1,LB=c(rep(c(0,1),7),rep(-.9,6)),
          UB=c(rep(10,14),rep(.99,6)) )
pseud2=rvinenllkpseud(mle2$estimate,udat,A,logdcopnames[1:2],
                      pcondnames[1:2],np)
zdat2f=nscore(pseud2$condforw)
zdat2b=nscore(pseud2$condback)

# specific of interest for tree 3 of A
# 61|75 forw[,1] forw[,2]
# 64|75 forw[,1] forw[,3]
# 73|56 back[,1] forw[,4]
# 52|17 back[,2] forw[,5]
# 48|75 forw[,3] forw[,6]

par(mfrow=c(3,3))
semic=semicor(zdat2f[,c(1,2)]); plot(zdat2f[,c(1,2)])
ze=bvnsemic(semic[1])
cat("forw1","forw2",semic,ze,"\n")
# forw1 forw2 0.2969355 0.1800482 0.1745947 0.1338262

semic=semicor(zdat2f[,c(1,3)])
plot(zdat2f[,c(1,3)]); ze=bvnsemic(semic[1])
cat("forw1","forw3",semic,ze,"\n")
# forw1 forw3 0.6507415 0.4087883 0.4542447 0.4072167
# ...

```

Open problems

Real data example where simplifying assumption for trees 2 and higher is severely violated.

Method to generate all 2-truncated vines, 3-truncated vines without enumerating all vines. Number of 1-truncated vines (trees) with d variables is d^{d-2} (Cayley's theorem, enumeration is possible through Prüfer code).

Vine copulas for making predictions (machine learning). For example, 8 measurements on abalone were obtained simultaneously (as well as M,F,infant); goal is to predict age from the 7 measurements of size and weight. Because of non-Gaussianity and non-homoscedasticity of conditional distributions, a vine copula density can be fitted to the 8 variables followed by a conditional distribution of age given the other variables to get prediction intervals that are not constant length. This approach could be better than linear heteroscedastic approaches.