Gaussian Process Vine Copulas for Multivariate Dependence

José Miguel Hernández–Lobato\textsuperscript{1,2},
David López–Paz\textsuperscript{3,2} and
Zoubin Ghahramani\textsuperscript{1}

June 27, 2013

\textsuperscript{1}University of Cambridge
\textsuperscript{2}Equal Contributor
\textsuperscript{3}Max-Planck-Institute for Intelligent Systems
Copulas and Multivariate Probabilistic Models

Copulas link univariate marginal distributions into joint multivariate probabilistic models.

Copulas specify dependencies between the random variables.
Separating Marginals and Dependencies Using Copulas

\[ p(x_1, \ldots, x_d) = c(u_1, \ldots, u_d) \prod_{i=1}^{d} p_i(x_i), \]

where \( u_i := P_i(x_i) \) and \( P_i \) denotes the cumulative distribution function (cdf) for the variable \( x_i \).

We can model each of the \( d \) marginal distributions and the copula function independently.

There exists a broad catalog of two-dimensional copula models

Gaussian  Clayton  Frank  tCopula  Gumbel  Joe

However, for the high-dimensional regime, the number of copula models available and their expressiveness is more limited.
More than Two Dimensions: Vine Factorizations

Parameters: bivariate copula families, tree construction method.
Example of Vine Factorizations: Three Variables

We can factorize \( c(u_1, u_2, u_3) \) using the product rule of probability as

\[
c(u_1, u_2, u_3) = p(u_3|u_1, u_2)p(u_2|u_1)
\]

and we can express each factor in terms of bivariate copula functions

\[
p(u_3|u_1, u_2) = p(u_3, u_1|u_2)[p(u_1|u_2)]^{-1}
= c_{31|2}(u_3|\{u_2\}, u_1|\{u_2\})p(u_3|u_2),
\]

\[
p(u_3|u_2) = p(u_3, u_2) = c_{32}(u_3, u_2),
\]

\[
p(u_2|u_1) = p(u_2, u_1) = c_{21}(u_2, u_1),
\]

where \( u_i|\{u_j\} \) is the conditional cdf of \( u_i \) given \( u_j \). Finally, we obtain

\[
c(u_1, u_2, u_3) = c_{31|2}(u_3|\{u_2\}, u_1|\{u_2\})c_{32}(u_3, u_2)c_{21}(u_2, u_1).
\]
Example of Vine Factorizations: Four Variables

Using vines, we can factorize \( p(x_1, x_2, x_3, x_4) \) as

\[
p(x_1, x_2, x_3, x_4) = p_1(x_1)p_2(x_2)p_3(x_3)p_4(x_4) \\
\times c_{13}(u_1, u_3)c_{23}(u_2, u_3)c_{34}(u_3, u_4) \\
\times c_{12|3}(u_1|\{u_3\}, u_2|\{u_3\})c_{14|3}(u_1|\{u_3\}, u_4|\{u_3\}) \\
\times c_{24|13}(u_2|\{u_1, u_3\}, u_4|\{u_1, u_3\}) 
\]

We need to use conditional copulas and conditional cdfs!

The conditional cdfs can be recursively computed using

\[
u_i|v = \frac{\partial C_{ij|v_{-j}}(u_i|v_{-j}, u_j|v_{-j})}{\partial u_j|v_{-j}}.
\]

The Vine Simplifying Assumption: conditional copulas are independent from their conditioning variables.

Can we do better?
A Semi-parametric Model for Conditional Copulas

Assumption: the conditional copula is parametric and the conditioning variables have a direct effect on the copula parameter, specified in terms of Kendall’s tau\(^4\) \(\tau \in [-1, 1]\):

**Example:** The Gaussian copula.

\[
c(u_1, u_2 | \tau) = \frac{\phi_2[\Phi^{-1}(u_1), \Phi^{-1}(u_2) | \rho = \sin(\tau \pi 0.5)]}{\phi(\Phi^{-1}(u_1))\phi(\Phi^{-1}(u_2))},
\]

\(\phi\) and \(\Phi\) are the standard Gaussian pdf and cdf. 
\(\phi_2\) is a 2D Gaussian pdf with zero mean, correlation \(\rho\) and variances 1.

Let \(z\) be a vector of conditioning variables.

Then we assume the conditional dependence \(\tau = \sigma[f(z)]\), where \(f\) is a real function and \(\sigma(x) = 2\Phi(x) - 1\) is a sigmoid function.

\(^4\)Many one-parameter copulas can be specified in terms of Kendall’s \(\tau\) rank correlation coefficient.
Estimation of Conditional Copulas with GP+EP

Given the data $\mathcal{D}_{u,v} = \{(u_i, v_i)\}_{i=1}^n$ and $\mathcal{D}_z = \{z_i\}_{i=1}^n$, we place a Gaussian process prior on $f$ and do Bayesian inference.

The posterior for $\mathbf{f} = (f_1, \ldots, f_n)^T$, where $f_i = f(z_i)$, is

$$p(\mathbf{f} | \mathcal{D}_{u,v}, \mathcal{D}_z) = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}} \propto p(\mathbf{f} | \mathcal{D}_z)p(\mathcal{D}_{u,v} | \mathbf{f}),$$

where

$$p(\mathbf{f} | \mathcal{D}_z) = \mathcal{N}(\mathbf{f} | \mathbf{m}, \mathbf{K}) \quad \text{GP prior}$$

and

$$p(\mathcal{D}_{u,v} | \mathbf{f}) = \prod_{i=1}^n c(u_i, v_i | \tau = \sigma(f_i)) \quad \text{copula likelihood model}.$$
**Ingredient 1: The Gaussian Process Prior**

\[
p(f|\mathcal{D}_{u,v}, \mathcal{D}_z) \propto \mathcal{N}(f|m, K) \prod_{i=1}^{n} c(u_i, v_i | \tau = \sigma(f_i)).
\]

The prior distribution of any set of values of \( f \) is jointly Gaussian. A **covariance function** specifies the value of \( K \in \mathbb{R}^{n \times n}:

\[
K_{ij} = k(z_i, z_j) = s \exp(-(z_i - z_j)^T \text{diag}() (z_i - z_j)) + s_0. \tag{1}
\]

This imposes smoothness properties on the possible prior values of \( f \).

In the GP prior \( m \) is given by a constant mean function.
Ingredient 2: The Copula Likelihood

\[ p(f|D_{u,v}, D_z) \propto \mathcal{N}(f|0, K) \prod_{i=1}^{n} c(u_i, v_i | \tau = \sigma(f_i)) . \]

If several copula families are available as candidates, build one posterior for each one and perform **Bayesian model selection**.
To make predictions at $z_{n+1}$ for $u_{n+1}$ and $v_{n+1}$ average over all possible functions $f$, weighting each one by its posterior probability:

$$p(u_{n+1}, v_{n+1}|z_{n+1}, D_{UV}, D_z) =$$

$$\int c(u_{n+1}, v_{n+1}|\tau = \sigma[f_{n+1}]) p(f_{n+1}|f, z_{n+1}, D_z) p(f|D_{UV}, D_z) df df_{n+1}.$$  

**Problem 1:** The likelihood is not Gaussian. We cannot perform exact inference (solve the integral analytically).

**Problem 2:** Working with GPs is expensive. It often requires $O(n^3)$ operations (compute Cholesky decompositions of $n \times n$ matrices).
Ingredient 4: Expectation Propagation

Solution for 1: Approximate likelihood factors with Gaussians.

\[
\mathcal{P}(f|\mathcal{D}_U, \mathcal{D}_Z) \propto \mathcal{P}(U_1, V_1|\tau = \sigma[f_1]) \cdots \mathcal{P}(U_n, V_n|\tau = \sigma[f_n]) \mathcal{N}(f|m_0, K)
\]

EP tunes \( \hat{m}_i \) and \( \hat{v}_i \) by minimizing \( \text{KL}[q_i(f_i)Q(f)[\hat{q}_i(f_i)]^{-1}|Q(f)] \).

After convergence, the new (approximate) posterior is Gaussian. An approximation of the evidence and its gradient is returned.
Solution for 2: Use a covariance matrix $K'$ with a simple form:

$$ K \approx K' = Q + \text{diag}(K - Q), \quad Q = K_{nn_0} K_{n_0n_0}^{-1} K_{n_0n}, $$

where a subset of $n_0 \ll n$ pseudo-inputs is used.

The $n_0$ pseudo-inputs are tuned by maximizing the evidence returned by EP. New cost of $O(nn_0^2)$. 
All Together: The GPCC Method

Input:
▷ Data \{(u_i, v_i, z_i)\}_{i=1}^{n}.
▷ Parametric copula density function \(c(u, v|\tau)\).
▷ Number of pseudo-inputs \(n_0\).

Algorithm:
▷ Until Convergence:
  ▷ Use EP to get approximate Gaussian posterior, evidence and evidence gradient.
  ▷ Follow evidence gradient to tune pseudo-inputs and the parameters of the covariance function.
▷ Prediction of \((u_*, v_*)\) at \(z_*\): sample \(f_* = f(z_*)\) from the EP-posterior and average over copulas with \(\tau_* = \sigma(f_*)\).

Output:
▷ Conditional copula model.
Experimental Results: Synthetic Test

$Z$ is uniform in $[-6, 6]$

$(U, V)$ are Gaussian with correlation $3/4 \sin(Z)$

We sample the data \{$(\hat{P}_U(u_i), \hat{P}_V(v_i), \hat{P}_Z(z_i))$\}$\!_{i=1}^{50}$.

\begin{center}
\begin{tabular}{c c c}
\hline
$\tau_{U,V|Z}$ & GPVINE & MLLVINE & TRUE \\
\hline
0.0 & 0.2 & 0.4 & 0.6 \\
-0.6 & -0.2 & 0.2 & 0.6 \\
\hline
\end{tabular}
\end{center}

MLLVINE is a nearest-neighbor algorithm proposed by Acar (2011).

Experimental Results: Real-World Data

Real-world data: UCI datasets, meteorological data, mineral concentrations and financial data

Average test log likelihood when limited to two trees in the vine:

<table>
<thead>
<tr>
<th>Data</th>
<th>SVINE</th>
<th>MLLVINE</th>
<th>GPVINE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synthetic</td>
<td>$-0.005 \pm 0.012$</td>
<td>$0.101 \pm 0.162$</td>
<td>$0.298 \pm 0.031$</td>
</tr>
<tr>
<td>Uranium</td>
<td>$0.006 \pm 0.006$</td>
<td>$0.016 \pm 0.026$</td>
<td>$0.022 \pm 0.012$</td>
</tr>
<tr>
<td>Cloud</td>
<td>$8.899 \pm 0.334$</td>
<td>$9.013 \pm 0.600$</td>
<td>$9.335 \pm 0.348$</td>
</tr>
<tr>
<td>Glass</td>
<td>$1.206 \pm 0.259$</td>
<td>$0.460 \pm 1.996$</td>
<td>$1.264 \pm 0.303$</td>
</tr>
<tr>
<td>Housing</td>
<td>$3.975 \pm 0.342$</td>
<td>$4.246 \pm 0.480$</td>
<td>$4.487 \pm 0.386$</td>
</tr>
<tr>
<td>Jura</td>
<td>$2.134 \pm 0.164$</td>
<td>$2.125 \pm 0.177$</td>
<td>$2.151 \pm 0.173$</td>
</tr>
<tr>
<td>Shuttle</td>
<td>$2.552 \pm 0.273$</td>
<td>$2.256 \pm 0.612$</td>
<td>$3.645 \pm 0.427$</td>
</tr>
<tr>
<td>Weather</td>
<td>$0.789 \pm 0.159$</td>
<td>$0.771 \pm 0.890$</td>
<td>$1.312 \pm 0.227$</td>
</tr>
<tr>
<td>Stocks</td>
<td>$2.802 \pm 0.141$</td>
<td>$2.739 \pm 0.155$</td>
<td>$2.785 \pm 0.146$</td>
</tr>
</tbody>
</table>

Results are averages on 50 random train/test splits.

SVINE are Vines that use the simplifying assumption.
Experimental Results: Using More Trees

- **GPVINE**
- **SVINE**

Data plots for various datasets including: cloud, glass, housing, jura, shuttle, weather, stocks.
Experimental Results: Weather in Barcelona

\[ \tau(\text{atmospheric pressure, cloud percentage} | \{\text{latitude, longitude}\}). \]
GP Conditional Copulas and Financial Time Series

We learn time-changing dependencies in financial returns.

Three parametric copulas:

\begin{align*}
\text{Gaussian Copula} & \\
\text{Student's t Copula} & \\
\text{Symmetrized Joe Clayton Copula} & 
\end{align*}

The marginals are described using AR-GARCH processes:

\begin{align*}
  x_t &= \phi_0 + \phi_1 x_{t-1} + \sigma_t \epsilon_t, \\
  \sigma_t &= \omega + \alpha \sigma_{t-1} (|\epsilon_{t-1}| - \gamma \epsilon_{t-1}) + \beta \sigma_{t-1} .
\end{align*}
Experimental Results: Conditioning on time: NASDAQ

<table>
<thead>
<tr>
<th>Method</th>
<th>HD HON</th>
<th>AXP BA</th>
<th>CNW CSX</th>
<th>ED EIX</th>
<th>HPQ</th>
<th>BARC</th>
<th>RBS</th>
<th>RBS HSBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPCC-G</td>
<td>0.1247</td>
<td>0.1133</td>
<td>0.1450</td>
<td>0.2072</td>
<td>0.1536</td>
<td>0.2424</td>
<td>0.3401</td>
<td>0.1860</td>
</tr>
<tr>
<td>GPCC-T</td>
<td><strong>0.1289</strong></td>
<td><strong>0.1187</strong></td>
<td><strong>0.1499</strong></td>
<td>0.2059</td>
<td><strong>0.1591</strong></td>
<td>0.2486</td>
<td><strong>0.3501</strong></td>
<td><strong>0.1882</strong></td>
</tr>
<tr>
<td>GPCC-SJC</td>
<td>0.1210</td>
<td>0.1095</td>
<td>0.1399</td>
<td>0.1935</td>
<td>0.1462</td>
<td>0.2342</td>
<td>0.3234</td>
<td>0.1753</td>
</tr>
<tr>
<td>HMM</td>
<td>0.1260</td>
<td>0.1119</td>
<td>0.1458</td>
<td>0.2040</td>
<td>0.1511</td>
<td><strong>0.2486</strong></td>
<td>0.3414</td>
<td>0.1818</td>
</tr>
<tr>
<td>TVC</td>
<td>0.1251</td>
<td>0.1119</td>
<td>0.1459</td>
<td>0.2011</td>
<td>0.1511</td>
<td>0.2449</td>
<td>0.3336</td>
<td>0.1823</td>
</tr>
<tr>
<td>DSJCC</td>
<td>0.0935</td>
<td>0.0750</td>
<td>0.1196</td>
<td>0.1721</td>
<td>0.1163</td>
<td>0.2188</td>
<td>0.3051</td>
<td>0.1582</td>
</tr>
<tr>
<td>CONST-G</td>
<td>0.1162</td>
<td>0.1027</td>
<td>0.1288</td>
<td>0.1962</td>
<td>0.1325</td>
<td>0.2307</td>
<td>0.2979</td>
<td>0.1663</td>
</tr>
<tr>
<td>CONST-T</td>
<td>0.1239</td>
<td>0.1091</td>
<td>0.1408</td>
<td>0.2007</td>
<td>0.1481</td>
<td>0.2426</td>
<td>0.3301</td>
<td>0.1775</td>
</tr>
<tr>
<td>CONST-SJC</td>
<td>0.1175</td>
<td>0.1046</td>
<td>0.1307</td>
<td>0.1891</td>
<td>0.1373</td>
<td>0.2268</td>
<td>0.2992</td>
<td>0.1639</td>
</tr>
</tbody>
</table>

- Data is preprocessed using AR(1)-GARCH(1,1) models.
- GPCC-G: GPs on $\tau(t)$ for Gaussian Copula.
- GPCC-T: GPs on $\nu(t)$ and $\tau(t)$ for tCopula.
- GPCC-SJC: GPs on $\tau_{L}(t)$ and $\tau_{U}(t)$ for Sym. Joe Clayton.
- HMM: High/Low Correlated tCopula (Jondeau 2006).
- TVC: Dynamic tCopula (Tse 2002).
- DSJCC: Dynamic SJC Copula (Patton 2006).
Experimental Results: Conditioning on time: FOREX

Method AUD CAD JPY NOK SEK EUR GBP NZD
GPCC-G 0.1260 0.0562 0.1221 0.4106 0.4132 0.8842 0.2487 0.1045
GPCC-T 0.1319 0.0589 0.1201 0.4161 0.4192 0.8995 0.2514 0.1079
GPCC-SJC 0.1168 0.0469 0.1064 0.3941 0.3905 0.8287 0.2404 0.0921
HMM 0.1164 0.0478 0.1009 0.4069 0.3955 0.8700 0.2374 0.0926
TVC 0.1181 0.0524 0.1038 0.3930 0.3878 0.7855 0.2301 0.0974
DSJCC 0.0798 0.0259 0.0891 0.3994 0.3937 0.8335 0.2320 0.0560
CONST-G 0.0925 0.0398 0.0771 0.3413 0.3426 0.6803 0.2085 0.0745
CONST-T 0.1078 0.0463 0.0898 0.3765 0.3760 0.7732 0.2231 0.0875
CONST-SJC 0.1000 0.0425 0.0852 0.3536 0.3544 0.7113 0.2165 0.0796

- Data against USD from 02/01/1990 to 15/01/2013 (6011 points).
- Decorrelations at Fall 2008 (onset of global recession) and Summer 2010 (worsening European sovereign debt crisis)
Summary and Conclusions

Vines are flexible dependence models. They factorize a copula density into a product of conditional bivariate copulas.

In practical implementations of vines, some of the conditional dependencies in the bivariate copulas are usually ignored.

To avoid this, we have proposed a method for the estimation of fully conditional vines using Gaussian processes (GPVINE).

GPVINE outperforms a baseline that ignores conditional dependencies (SVINE) and other alternatives (MLLVINE).

The building block of GPVINE are GP Conditional Copulas (GPCC).

GPCC obtain state-of-the-art performance in the task of predicting time-changing dependencies in financial data.
Thanks!