Variational Inference for high dimensional factor copulas

Hoang Nguyen

joint work with M. Concepcion Ausin, Pedro Galeano
Universidad Carlos III de Madrid

Tuesday 7th November, 2017
# Outline

1. **Introduction to copulas**
   - Bivariate copula families
   - Vine copulas
   - Factor copulas

2. **Variational Inference**
   - Approximate posterior inference
   - Variational Objective function
   - Black Box VI
   - Variance reduction technique

3. **Simulation**

4. **Empirical Illustration**
   - Financial return dependence

5. **Conclusion**
Introduction to Copulas

A multivariate copula is a multivariate cdf defined on $[0, 1]^d$ with uniform $U(0, 1)$ marginals.

Consider a $n$-dimensional joint cdf $F$ with marginals $F_1, ..., F_d$. There exists a copula $C$, such that

$$F(x_1, ..., x_d) = C(F_1(x_1), ..., F_d(x_d))$$

for all $x_i$ in $[-\infty, \infty], i = 1, ..., d$. 
Elliptical copulas

\[ C^G_R(u_1, \ldots, u_d) = \Phi^n_R(\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_d)) \]

\[ C^{St}_{R,\nu}(u_1, \ldots, u_d) = F^{MSt}_R(F^{-1}_t(\nu(u_1)), \ldots, F^{-1}_t(\nu(u_d))) \]

**Figure:** Contours of bivariate distributions with the same marginal standard normal
Archimedean copulas

Common Bivatiate Archimedean Copulas:
\[ C(u_1, u_2) = \varphi^{-1}(\varphi(u_1) + \varphi(u_2)) \]

Clayton (1978)
\[ \alpha \geq 0 \]
\[ \varphi(t) = t^{-\alpha} - 1 \]

Frank (1979)
\[ \alpha \geq 0 \]
\[ \varphi(t) = -\ln \frac{e^{-\alpha t} - 1}{e^{-\alpha} - 1} \]

Gumbel (1960)
\[ \alpha \geq 1 \]
\[ \varphi(t) = (-\ln t)^\alpha \]

**Figure:** Contours of bivariate distributions with the same marginal standard normal
Vine copulas

Vine copula: C-vine, D-vine, R-vine (Aas et al., 2009)

Figure: D-vine and Canonical vine copula
Factor copulas

Figure: One factor and two factor copula models (Krupskii and Joe, 2013)
Bifactor and nested factor copulas

Figure: Bifactor copulas with $d = 12$ and $G = 3$ (Krupskii and Joe, 2015)

Figure: Nested factor copulas with $d = 12$ and $G = 3$ (Krupskii and Joe, 2015)
Posterior inference

Assuming that we have specify a factor copula structure together with bivariate linking copula in each tree layers.

- We are interested in the inference on the collection of latent variables and copula parameters \( \{v, \theta\} \) based on the observables \( \{u\} \)
- The posterior is

\[
p(v, \theta | u) = \frac{p(v, \theta, u)}{p(u)}
\]

- One factor copula, for example

\[
p(v_0, \theta | u_1, \ldots, u_d) \propto \prod_{i=1}^{d} \frac{p(u_i, v_0 | \theta)}{p(v_0)} p(v_0) p(\theta)
\]

\[
\propto \prod_{i=1}^{d} c_{u_i, v_0}(u_i, v_0 | \theta) p(\theta)
\]
Posterior inference

For bifactor copula, we derive the posterior using the properties for vine copula,

\[
p(v_0, v_1, \ldots, v_G, \theta | u_1, \ldots, u_d) \propto \prod_{g=1}^{G} \prod_{i=1}^{d_g} c(u_{ig}, v_0, v_g | \theta) p(\theta)
\]

\[
\propto \prod_{g=1}^{G} \prod_{i=1}^{d_g} c_{u_{ig}, v_0}(u_{ig}, v_0 | \theta)
\]

\[
\times \prod_{g=1}^{G} \prod_{i=1}^{d_g} c_{u_{ig}, v_g | v_0}(u_{ig} | v_0, v_g | \theta) p(\theta)
\]

where \(u_{ig} | v_0 = F(u_{ig} | v_0)\). Thus, it is computational expensive. We approximate the posterior by a proposal \(q(v, \theta | \lambda^*)\).

\[
q(v, \theta | \lambda^*) \approx p(v, \theta | u)
\]
Kullback Leibler divergence

Variational Inference measures the different between two distributions using Kullback Leibler divergence:

\[ KL(Q\|P) = \int q(x) \log \frac{q(x)}{p(x)} \, dx \geq 0 \]

Note that: \( KL(Q\|P) \neq KL(P\|Q) \geq 0 \)
Objective function

We specify a family $Q$ of densities as the proposal distribution

$$q(v, \theta | \lambda^*) = \arg \min_{\lambda} KL(q(v, \theta) || p(v, \theta | u))$$

$$KL(q(v, \theta) || p(v, \theta | u)) = E_q[\log q(v, \theta)] - E_q[\log p(v, \theta | u)]$$

$$KL(q(v, \theta) || p(v, \theta | u)) = E_q[\log q(v, \theta)] - E_q[\log p(v, \theta, u)] + \log p(u)$$
Objective function

We specify a family $Q$ of densities as the proposal distribution

$$q(v, \theta | \lambda^*) = \arg \min_{\lambda} KL(q(v, \theta) \| p(v, \theta | u))$$

$$KL(q(v, \theta) \| p(v, \theta | u)) = \mathbb{E}_q[\log q(v, \theta)] - \mathbb{E}_q[\log p(v, \theta | u)]$$

$$KL(q(v, \theta) \| p(v, \theta | u)) = \mathbb{E}_q[\log q(v, \theta)] - \mathbb{E}_q[\log p(v, \theta, u)] + \log p(u)$$

Because we cannot compute the KL, we optimize an alternative objective (Evidence lower bound) that is equivalent to the KL up to an added constant:

$$ELBO(q) = \mathbb{E}_q[\log p(v, \theta, u)] - \mathbb{E}_q[\log q(v, \theta)]$$

$$= \log p(u) - KL(q(v, \theta) \| p(v, \theta | u)) \leq \log p(u)$$

when $q(v, \theta) = p(v, \theta | u)$, we obtain $ELBO = \log p(u)$
Mean field variational family

In mean-field variational family, the latent variables are mutually independent and each governed by a distinct factor in the variational density.

\[ q(v, \theta) = \prod_{l=1}^{\text{#latents}} q(v_l) \prod_{i=1}^{\text{#}\theta} q(\theta_i) \]
Black Box Variational Inference

We specify a family $Q$ of densities over the latent variables.

$$
\lambda^* = \arg \max_{\lambda} \mathbb{E}_{q(v, \theta)}[\log p(v, \theta, u)] - \mathbb{E}_{q(v, \theta)}[\log q(v, \theta)]
$$

such that $\text{supp}(q(v, \theta | \lambda)) \subseteq \text{supp}(p(v, \theta | u))$

- We could propose directly a density approximation $q(v, \theta | \lambda)$ and take the derivative wrt. $\lambda$
- Update $\lambda = \lambda + \text{Step} \times \text{Gradient}$

However, this direct approach produces noisy evaluations of the gradient, $\nabla_\lambda \left( \mathbb{E}_{q(v, \theta)}[\log p(v, \theta, u)] - \mathbb{E}_{q(v, \theta)}[\log q(v, \theta)] \right)$. 
Black box variational inference

An automated algorithm (ADVI) to solve the optimization problem based on continuous transformations of the parameters (Kucukelbir, 2016).

- Define a one-to-one differentiable function.

\[ T : \text{supp}(p(v, \theta | u)) \rightarrow \mathbb{R}^K \]

- Any continuous transformation could be possible:
  - Correlation constrain: \[ T(\theta) = \text{atanh} \theta = \frac{1}{2} \log \left( \frac{1+\theta}{1-\theta} \right) \]
  - Positive constrain: \[ T(\theta) = \log(\theta) \]
  - Lower constrain: \[ T(\theta) = \log(\theta - L) \]
  - Lower and upper bound constrain: \[ T(\theta) = \text{logit} \left( \frac{\theta - L}{U - L} \right) \]
Variance reduction technique

The optimization becomes:

$$\mu^*, \sigma^* = \arg \max \mathbb{E}_{N(\mu, \sigma)}[\log p(v, \theta, u)] - \mathbb{E}_{N(\mu, \sigma)}[\log q(v, \theta)]$$

- Draw M samples $\eta \sim N(0, I)$.
- Obtain $x_k = \mu_k + \eta_k \sigma_k$.
- Obtain $(v_k, \theta_k) = T^{-1}(x_k)$
- Average over M samples for the ELBO.
- Similar approach to calculate the gradient of ELBO. Update $\mu, \sigma$
- This algorithm is guaranteed to converge to a local maximum of the ELBO under certain conditions on the step-size sequence.
- Because $\sigma > 0$, we optimize over $\omega = \log \sigma$ instead.
Automatic Differentiation Variational Inference in Stan

**Algorithm 1:** Automatic differentiation variational inference

**Data:** Copula Data $U = \{ u_i \}$

**Result:** The value $\mu, \omega$

**Initialization** $\mu^{(0)} = 0, \omega^{(0)} = 0$;

**while** *Any change in copula types do*

  **while** *Change in ELBO is above some threshold do*

  Draw M samples $\eta_m \sim N(0, 1)$;

  Invert the standardized $x_m = \mu^{(j)} + \exp(\omega^{(j)})\eta_m$;

  Approximate the noisy gradient $\nabla_\mu \text{ELBO}$ and $\nabla_\omega \text{ELBO}$;

  Update $\mu^{(i+1)} \leftarrow \mu^{(j)} + \varrho^{(j)} \nabla_\mu \mathcal{F}$;

  Update $\omega^{(i+1)} \leftarrow \omega^{(j)} + \varrho^{(j)} \nabla_\omega \mathcal{F}$;

  Incremental iteration (i) ;

  **end**

**Select best bivariate copula $u_i$ and $v$ based on AIC,BIC ;**

**end**

Reassign the copulas and estimate ;

**end**

Return Copula structure and the parameters of proposal distribution;
One factor copula model

We generate a sample of $d = 100$ variables with $T = 1000$ time observations. Bivariate copula types are Gaussian, Student, Clayton, Gumbel, Frank, Joe (and their rotation 90, 180, 270 degree) and Mix copulas. Time is reported in seconds using one core Intel i7-4770 processor.

Table: Time of Computation and Copula selection

<table>
<thead>
<tr>
<th>Copula type</th>
<th>Gaussian</th>
<th>Student</th>
<th>Clayton</th>
<th>Gumbel</th>
<th>Frank</th>
<th>Joe</th>
<th>Mix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial at correct structure</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time estimated (s)</td>
<td>6</td>
<td>322</td>
<td>18</td>
<td>24</td>
<td>5</td>
<td>9</td>
<td>59</td>
</tr>
<tr>
<td>ELBO</td>
<td>31181</td>
<td>35490</td>
<td>78769</td>
<td>67530</td>
<td>58375</td>
<td>76254</td>
<td>58438</td>
</tr>
<tr>
<td>Initial at random structure</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time estimated (s)</td>
<td>303</td>
<td>625</td>
<td>325</td>
<td>258</td>
<td>316</td>
<td>308</td>
<td>382</td>
</tr>
<tr>
<td>Selection iteration</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>% correction</td>
<td>98%</td>
<td>78%</td>
<td>62%</td>
<td>100%</td>
<td>100%</td>
<td>57%</td>
<td>88%</td>
</tr>
<tr>
<td>ELBO</td>
<td>31191</td>
<td>35410</td>
<td>78767</td>
<td>67539</td>
<td>58383</td>
<td>76277</td>
<td>58449</td>
</tr>
</tbody>
</table>

(about 100 - 200 parameters / 100 bivariate copulas / 1 latent factor)
One factor copula model

Figure: Posterior means of ν and θ versus true values
One factor copula model

Figure: Mixed copula estimation with a correct vs random initial structure
Nested factor copula model

We generate the nested factor copula with $d = 100$ variables, $N = 1000$ observations and $G = 5$ groups of latent factors.

Table: Time of Computation and Copula selection

<table>
<thead>
<tr>
<th>Copula type</th>
<th>Gaussian</th>
<th>Student</th>
<th>Clayton</th>
<th>Gumbel</th>
<th>Frank</th>
<th>Joe</th>
<th>Mix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial at correct structure</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time estimated (s)</td>
<td>7</td>
<td>334</td>
<td>18</td>
<td>27</td>
<td>9</td>
<td>11</td>
<td>80</td>
</tr>
<tr>
<td>ELBO</td>
<td>24731</td>
<td>25351</td>
<td>69358</td>
<td>59615</td>
<td>47988</td>
<td>69989</td>
<td>41796</td>
</tr>
<tr>
<td>Initial at random structure</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time estimated (s)</td>
<td>379</td>
<td>1045</td>
<td>354</td>
<td>417</td>
<td>333</td>
<td>380</td>
<td>481</td>
</tr>
<tr>
<td>Selection iteration</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>% correction</td>
<td>72%</td>
<td>72%</td>
<td>70%</td>
<td>97%</td>
<td>97%</td>
<td>58%</td>
<td>79%</td>
</tr>
<tr>
<td>ELBO</td>
<td>22595</td>
<td>25209</td>
<td>68966</td>
<td>57550</td>
<td>46157</td>
<td>69993</td>
<td>41804</td>
</tr>
</tbody>
</table>

(about 105 - 210 parameters / 105 bivariate copulas / 6 latent factors)
Nested factor copula model

Figure: Posterior means of $v_0$, $v_g$ and $\theta$ versus true values
Bifactor copula model

We generate the bifactor copula with \( d = 100 \) variables, \( T = 1000 \) time observations and \( G = 5 \) groups of latent factors.

**Table: Time of Computation and Copula selection**

<table>
<thead>
<tr>
<th>Copula type</th>
<th>Gaussian</th>
<th>Student</th>
<th>Clayton</th>
<th>Gumbel</th>
<th>Frank</th>
<th>Joe</th>
<th>Mix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial at correct structure</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time estimated (s)</td>
<td>59</td>
<td>1212</td>
<td>119</td>
<td>102</td>
<td>56</td>
<td>100</td>
<td>515</td>
</tr>
<tr>
<td>ELBO</td>
<td>50413</td>
<td>83977</td>
<td>136734</td>
<td>117332</td>
<td>96655</td>
<td>135002</td>
<td>93867</td>
</tr>
<tr>
<td>Initial at random structure</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time estimated (s)</td>
<td>1589</td>
<td>4317</td>
<td>857</td>
<td>1028</td>
<td>743</td>
<td>718</td>
<td>1025</td>
</tr>
<tr>
<td>Selection iteration</td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>4</td>
<td>6</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>% correction Tree 1</td>
<td>99%</td>
<td>69%</td>
<td>82%</td>
<td>100%</td>
<td>99%</td>
<td>48%</td>
<td>77%</td>
</tr>
<tr>
<td>% correction Tree 2</td>
<td>97%</td>
<td>79%</td>
<td>76%</td>
<td>57%</td>
<td>98%</td>
<td>44%</td>
<td>66%</td>
</tr>
<tr>
<td>ELBO</td>
<td>51260</td>
<td>83917</td>
<td>136508</td>
<td>111419</td>
<td>99575</td>
<td>134895</td>
<td>96287</td>
</tr>
</tbody>
</table>

(about 200 - 300 parameters / 200 bivariate copulas / 6 latent factors)
Bifactor copula model

Figure: Posterior means of $\nu_0$, $\nu_g$ and $\theta$ versus true values
Financial return dependence

We illustrate an empirical example using $d = 100$ stock returns divided into $G = 10$ groups from 01/01/2010 to 31/12/2013 of the companies listed in S&P 500 index. The daily data contain $T = 1000$ observation days. We use AR(1)-GARCH(1,1) to marginalize each stock returns:

$$
\begin{align*}
    r_{it} &= c_i + \phi_i r_{i,t-1} + a_{it} \\
    a_{it} &= \sigma_{it} \eta_{it} \\
    \sigma_{it}^2 &= \omega_i + \alpha_{i1} a_{i,t-1}^2 + \beta_{i1} \sigma_{i,t-1}^2
\end{align*}
$$

with skewed Student-t innovation, $\eta_{it}$. Then, the dependence structure of innovations is modelled by a factor copula function

$$
\begin{align*}
    \eta_{1t}, \ldots, \eta_{dt} &\sim F(\eta_{1t}, \ldots, \eta_{dt}) \\
    &\sim C(F(\eta_{1t}), \ldots, F(\eta_{dt})|\theta, \nu)
\end{align*}
$$
Financial return dependence

**Table:** Time of Computation and Copula selection

<table>
<thead>
<tr>
<th>Structure</th>
<th>One factor</th>
<th>Nested factor</th>
<th>Two factor</th>
<th>Bifactor copula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time estimated (s)</td>
<td>1559</td>
<td>2225</td>
<td>4812</td>
<td>5059</td>
</tr>
<tr>
<td>ELBO</td>
<td>33340</td>
<td>34232</td>
<td>35051</td>
<td>36070</td>
</tr>
<tr>
<td>Selection iteration</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td># bivariate links</td>
<td>100</td>
<td>110</td>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>% Gaussian</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>% Student</td>
<td>94</td>
<td>90</td>
<td>71</td>
<td>92</td>
</tr>
<tr>
<td>% Clayton (rotated)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>% Gumbel (rotated)</td>
<td>6</td>
<td>16</td>
<td>29</td>
<td>13</td>
</tr>
<tr>
<td>% Frank (rotated)</td>
<td>0</td>
<td>0</td>
<td>95</td>
<td>61</td>
</tr>
<tr>
<td>% Joe (rotated)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>% Independence</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>12</td>
</tr>
</tbody>
</table>
Conclusion

- Fast variational inference for factor copula model in high dimensions.
- Copula bivariate selection based on VI estimation performs well with simulation data.
- Compared to MCMC, variational inference tends to be faster and easier to scale to large data.
- VI generally underestimates the variance of the posterior density. However, the relative accuracy of variational inference and MCMC is still unknown. But we obtain quite reasonable result with factor copula models with limited time.
Sensitivity to Transformations

Consider a posterior density in the Gamma family, with support over \( \mathbb{R}_{>0} \). Figure 9 shows three configurations of the Gamma, ranging from Gamma\((1, 2)\), which places most of its mass close to \( \theta = 0 \), to Gamma\((10, 10)\), which is centered at \( \theta = 1 \). Consider two transformations \( T_1 \) and \( T_2 \)

\[
T_1 : \theta \mapsto \log(\theta) \quad \text{and} \quad T_2 : \theta \mapsto \log(\exp(\theta) - 1),
\]

both of which map \( \mathbb{R}_{>0} \) to \( \mathbb{R} \). \textit{ADVI} can use either transformation to approximate the Gamma posterior. Which one is better?

Figure 9 shows the \textit{ADVI} approximation under both transformations. Table 2 reports the corresponding KL divergences. Both graphical and numerical results prefer \( T_2 \) over \( T_1 \). A quick analysis corroborates this. \( T_1 \) is the logarithm, which flattens out for large values. However, \( T_2 \) is almost linear for large values of \( \theta \). Since both the Gamma (the posterior) and the Gaussian (the \textit{ADVI} approximation) densities are light-tailed, \( T_2 \) is the preferable transformation.

![Figure 9: \textit{ADVI} approximations to Gamma densities under two different transformations.](image)

\textbf{Figure 9:} \textit{ADVI} approximations to Gamma densities under two different transformations.