COPULAS FOR MACHINE LEARNING

Artem Sobolev

March 11, 2016

Higher School of Economics

TABLE OF CONTENTS

THEORY

COPULAS

Definition

A **Copula** is a multivariate probability distribution with uniform marginals.

$$C(u_1, \dots, u_d) \overset{\text{def}}{:=} \mathbb{P}(U_1 \leq u_1, \dots, U_d \leq u_d)$$

Equivalent definition

A **Copula** is a joint distribution of $(F_1(X_1), \dots, F_d(X_d))$.

$$C(u_1, \dots, u_d) \overset{\text{def}}{:=} \mathbb{P}(F_1(X_1) \leq u_1, \dots, F_d(X_d) \leq u_d)$$

For a set of (dependent) random variables $\{X_i\}_{i=1}^d$ with c.d.f. $\{F_i\}_{i=1}^d$

WHY COPULAS ARE INTERESTING

- Copulas can be used to describe dependencies between random variables
- A copula is preserved under monotone transformation of variables

Sklar's theorem

For any joint cumulative distribution function (c.d.f.) F with marginals $F_1, \ldots, F_d$ there exists a copula C such that

$$F(x_1, \ldots, x_n) = C(F_1(x_1), \ldots, F_d(x_d))$$

The copula is unique if marginals are continuous.

COPULA-FU

Density:

$$p(x_1,\dots,x_d) = \underbrace{c(F(x_1),\dots,F(x_d))}_{\text{copula density}}\, p(x_1)\dots p(x_d)$$

Marginalization:

$$p(x_1,\dots,x_{d-1}) = \frac{\partial^{d-1} C(F(x_1),\dots,F(x_{d-1}),1)}{\partial F(x_1)\dots\partial F(x_{d-1})} p(x_1)\dots p(x_{d-1})$$

Conditioning:

$$p(x_1|x_2,\dots,x_d) = \underbrace{\frac{c(F(x_1),F(x_2),\dots,F(x_d))}{\frac{\partial^{d-1} C(F(x_1),\dots,F(x_d))}{\partial F(x_2)\dots\partial F(x_d)}}}_{\text{conditional copula density}} p(y)$$

WHERE TO GET COPULAS

- Construct from existing multivariate distributions

$$C(u_1, \ldots, u_d) = F(F_1^{-1}(u_1), \ldots, F_d^{-1}(u_d))$$

- Use known copulas
  Example: **Archimedean copulas**

$$C(u_1, \ldots, u_d) = \psi(\psi^{-1}(u_1) + \cdots + \psi^{-1}(u_d))$$

- Build from bivariate copulas
  There are many well-known and researched bivariate copulas

GAUSSIAN COPULA

One can extract a copula from a multivariate Gaussian distribution $\mathcal{N}(0, R)$:

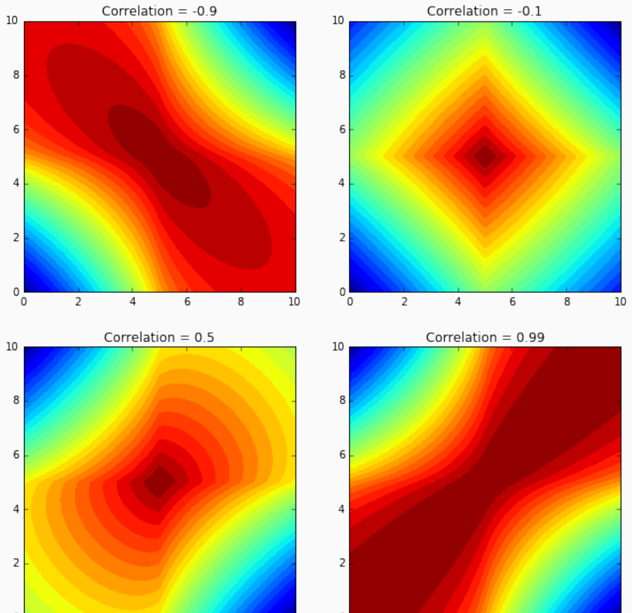
$$c(u_1, \ldots, u_d) = \frac{1}{\sqrt{\det R}} \exp\left(-\frac{1}{2}\xi^T(R^{-1} - I)\xi\right)$$

Where

- $\xi_k = \Phi^{-1}(u_k)$
- R is a correlation (not covariation!) matrix
- $\Phi(x)$ is a c.d.f. of the standard normal distribution $\mathcal{N}(0, 1)$

Can capture correlations between random variables, but doesn't have tail dependence

GAUSSIAN COPULA WITH LAPLACE MARGINS



VINE MODELS

We can build complex multivariate copula from a set of bivariate copulas

Definition

A **Regular Vine** is a set of trees $\{T_1, \ldots, T_{d-1}\}$ such that

1. Tree T_j has $d - j + 1$ nodes and $d - j$ edges
2. Edges $\mathcal{E}_j$ in tree T_j correspond to nodes in tree T_{j+1}
3. Nodes $\mathcal{V}_j$ of tree T_j are joined with an edge only if corresponding edges in tree T_{j-1} share a node

VINE COPULAS

Each edge $e(i,j) \in \mathcal{E}_k$ in a tree T_k has a corresponding bivariate copula

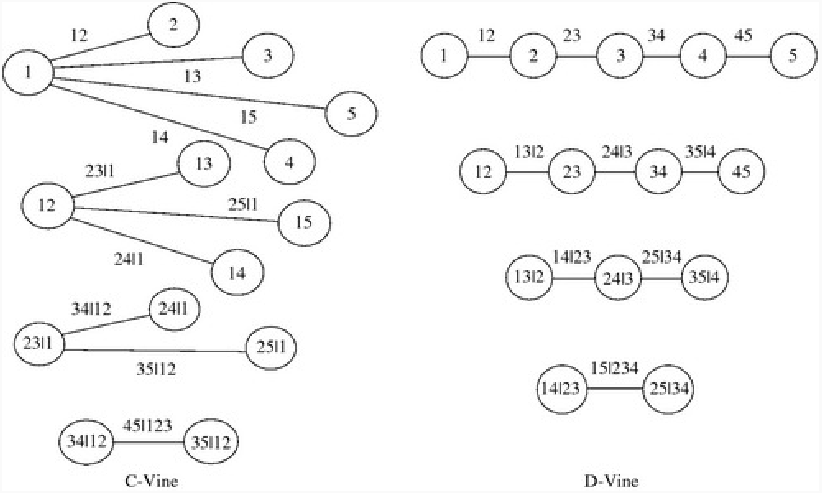
$$c_e \overset{\text{def}}{:=} c_{D(e)}\left(Q(u_i|u_{D(e)}), Q(u_j|u_{D(e)})\right)$$

Where $D(e)$ is a *conditioning set.*

Vine copula density is

$$c(u_1, \ldots, u_d) \overset{\text{def}}{:=} \prod_{i=1}^{d-1} \prod_{e \in \mathcal{T}_i} c_e$$

C-VINES AND D-VINES



FITTING COPULAS

- Maximum Likelihood Estimation still works
- It's common to fit marginals first (Inference functions for margins, IFM)
    - Equivalent to MLE only in special case of everything being Gaussian
- For vines you need structure, bivariate copulas and parameters
    - Greedy tree construction guided by some edge weighting scheme (Kendall's τ, Spearman's ρ, distance metrics, etc)
    - Various tests for bivariate copula selection

APPLICATIONS

COPULA VARIATIONAL INFERENCE

Recall the mean-field variational inference

$$q(z;\lambda) \overset{\text{def}}{:=} q(z_1;\lambda)\dots q(z_d;\lambda)$$

Let's augment it with a copula to introduce some dependencies

$$q(z;\lambda,\eta) \overset{\text{def}}{:=} c\,(Q(z_1;\lambda),\dots,Q(z_d;\lambda);\eta)\cdot q(z_1;\lambda)\dots q(z_d;\lambda)$$

OPTIMIZATION

Maximize the **ELBO** w.r.t. λ and η:

$$\mathcal{L}(x;\lambda,\eta) \overset{\text{def}}{:=} \mathbb{E}_{q(z;\lambda,\eta)}\left[\log \frac{p(x,z)}{q(z;\lambda,\eta)}\right]$$

- Alternating stochastic optimization
- For discrete z score function estimator is used

$$\nabla_\lambda \mathcal{L} = \mathbb{E}_{q(z;\lambda,\eta)}\left[\nabla_\lambda \log q(z;\lambda,\eta) \log \frac{p(x,z)}{q(z;\lambda,\eta)}\right]$$

- For continuous z stochastic gradients are obtained using the reparametrization trick: $z = z(u;\lambda,\eta)$ for $u \sim \text{Unif}([0,1]^d)$

$$\nabla_\lambda \mathcal{L} = \mathbb{E}_u\left[\left(\nabla_z \log p(x,z) - \nabla_z \log q(z;\lambda,\eta)\right) \nabla_\lambda z(u;\lambda,\eta)\right]$$

COPULA REPARAMETRIZATION TRICK

How to compute $z(u; \lambda, \eta)$ given a uniform noise $u \sim \text{Unif}([0,1]^d)$:

1. Calculate $v = (v_1, \dots, v_d)$ using (multivariate) inverse CDF sampling

$$\begin{aligned} v_1 &= u_1 \\ v_2 &= C_{2|1}^{-1}(u_2|v_1) \\ &\dots \\ v_d &= C_{d|1\dots d-1}^{-1}(u_d|v_1, \dots, v_{d-1}) \end{aligned}$$

2. Calculate $z = (Q_1^{-1}(v_1), \dots, Q_d^{-1}(v_d))$

Use the chain rule to obtain the gradient.

EXPERIMENTS

Mixture of Gaussians

$$p(x, z, \mu, \Lambda^{-1}, \pi) = p(\pi) \prod_{k=1}^{K} p(\mu_k, \Lambda_k^{-1}) \prod_{n=1}^{N} p(x_n | z_n, \mu_{z_n}, \Lambda_{z_n}^{-1}) p(z_n | \pi)$$

Model approximate posterior $q(z, \mu, \Lambda, \pi)$

Method	Likelihood	Runtime
Mean-field	-383.2	15 min.
LRVB	-330.5	38 min.
Copula VI (2 steps)	-303.2	32 min.
Copula VI (5 steps)	-80.2	1 hr. 17 min.
Copula VI (converged)	-50.5	2 hr.

COPULA BAYESIAN NETWORKS

In a Bayesian Network change in marginal distribution affects margins of all descendants. Let's use copulas to separate marginal distributions from dependencies.

Recall conditional copula density

$$R_c(F(y), F(x_1), \ldots, F(x_n)) = \frac{c(F(y), F(x_1), \ldots, F(x_n))}{\frac{\partial^n C(1, F(x_1), \ldots, F(x_n))}{\partial F(x_1) \ldots \partial F(x_n)}}$$

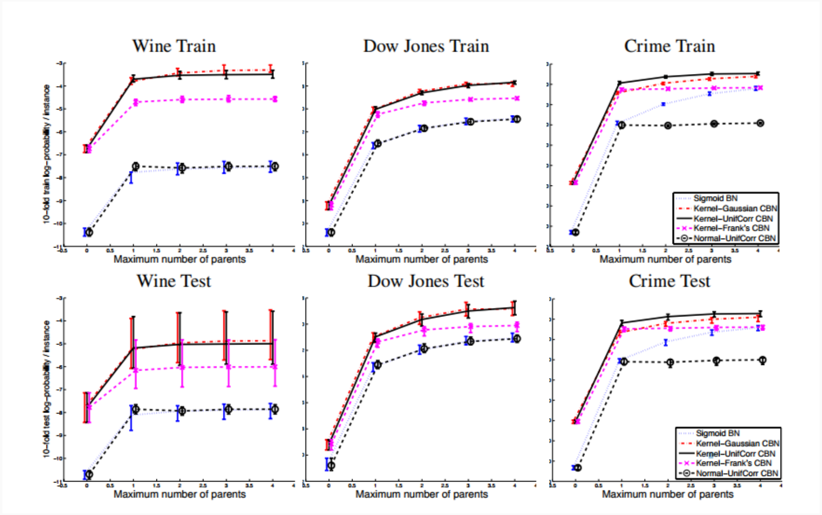
Can rewrite BN joint density:

$$p(x_1, \ldots, x_n) = \prod_{i=1}^{n} R_c(F(x_i), \{F(\mathrm{pa}_{ik})\}) p(x_i)$$

LEARNING

- Estimate margins first (IFM approach)
- Tree is learned using greedy search guided by Bayesian Information Criterion

RESULTS


Wine Train
Dow Jones Train
Crime Train
Wine Test
Dow Jones Test
Crime Test
10−fold train log−probability / instance
10−fold test log−probability / instance
Maximum number of parents
Sigmoid BN
Kernel−Gaussian CBN
Kernel−UnifCorr CBN
Kernel−Frank's CBN
Normal−UnifCorr CBN

QUESTIONS?