# Copulas in Machine Learning

M V D Satya Swaroop & Sarthi Shah

# Introduction

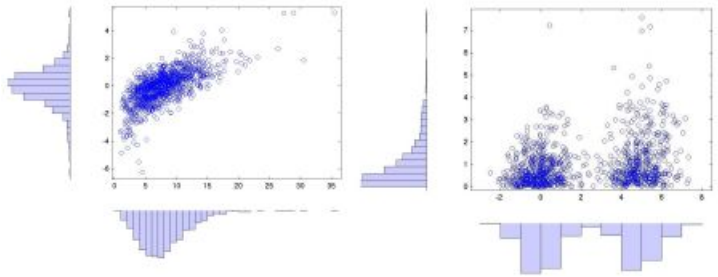
- Authors Motivation
- Background on select topics
- Copula based machine learning models
- Our Results

# Motivation

- Multivariate modeling is of fundamental interest in diverse complex domains
- Machine learning aims to solve this through probabilistic graphical models
- Copulas also allow for multivariate modelling framework
- Therefore, a union between the two fields was pursued
- Innovations in copula-based machine learning techniques:
  - Tree Structured Models
  - Undirected Structured Learning
  - Copula Bayesian Networks
  - Copula processes

# Background: Copulas

- Copula $C : [0,1]^n \rightarrow [0,1]$
  - $C(u_1,\ldots,u_n) = P(U_1 \leq u_1,\ldots,U_n \leq u_n)$
- Joint distribution as copula:
  - $F_X(x_1,\ldots, x_n) = C(F_1(x_1),\ldots,F_n(x_n))$
- Joint density can be derived from the copula function:
  - $$f_{\mathbf{X}}(\mathbf{x}) = \frac{\partial^n C(F_1(x_1),\ldots,F_n(x_n))}{\partial F_1(x_1)\ldots\partial F_n(x_n)} \prod_i f_i(x_i) \equiv c(F_1(x_1),\ldots,F_n(x_n)) \prod_i f_i(x_i)$$

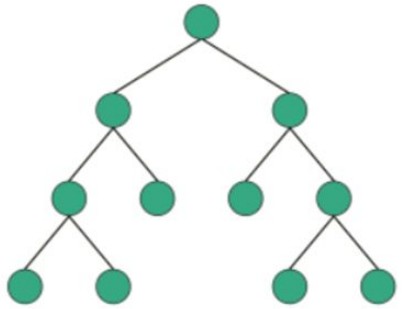


# Background: Probabilistic Graphical Models

- Framework for representing and reasoning about high-dimensional densities
- Directed Graph
- Directed acyclic graph (DAG) (G)
- Graphical models or Bayesian networks (BNs)
  - Use a DAG whose nodes correspond to the random variables of interest $X_1,\ldots,X_n$ to encode the independencies $I(G) = \{(X_i \perp ND_i \mid Pa_i)\}$
  - Joint density decomposes into a product of local conditional densities
    - $f_{\mathbf{X}}(\mathbf{x}) = \prod_{i=1}^{n} f_{X_i|\mathbf{Pa}_i}(x_i \mid \mathbf{pa}_i)$
- Undirected graphical models or Markov Networks (MN)
  - Use an undirected graph H that encodes the independencies $I(H)=\{(X_i \perp X\backslash\{X_i\} \cup Ne_i | Ne_i)\}$
  - Joint density decomposes according to the graph structure
    - $f_{\mathbf{X}}(\mathbf{x}) = \frac{1}{Z}\prod_{c\in\mathcal{C}} \phi_c(\mathbf{x}_c)$

# Tree Structured Models: Tree Structured Copulas

- Let T be an undirected tree structured graph
- Joint density function:
  - $$f_{\mathbf{X}}(\mathbf{x}) = \left[\prod_i f_i(x_i)\right] \prod_{(i,j)\in\mathcal{E}} \frac{f_{ij}(x_i,x_j)}{f_i(x_i)f_j(x_j)}$$



- Resulting Copula:
  - $$c_T(\cdot) = \frac{f_{\mathbf{X}}(\mathbf{x})}{\prod_i f_i(x_i)} = \prod_{(i,j)\in\mathcal{E}} \frac{f_{ij}(x_i,x_j)}{f_i(x_i)f_j(x_j)} = \prod_{(i,j)\in\mathcal{E}} c_{ij}(F_i(x_i),F_j(x_j))$$

# Tree Structured Models: Tree-Averaged Copulas

- Let β be a symmetric n×n matrix with non-negative entries and zero on the diagonal
- Let T be the set of all spanning trees over $X_1,\ldots,X_n$
- The probability of a spanning tree T is defined as:
  - Using a generalization of the Laplacian matrix: $P(T \in \mathcal{T} \mid \beta) = \frac{1}{Z}\prod_{(u,v)\in\mathcal{E}_T} \beta_{uv}$
- Density of the average of all copula spanning trees: $L_{uv}(\beta) = \begin{cases} -\beta_{uv} & u \neq v \\ \sum_w \beta_{uw} & u = v \end{cases}$

$$\sum_{T\in\mathcal{T}} P(T \mid \beta) c_T(\cdot) = \frac{1}{Z}\sum_{T\in\mathcal{T}}\left[\prod_{(u,v)\in\mathcal{E}_T} \beta_{uv} c_{uv}(F_u(x_u), F_v(x_v))\right] = \frac{|L^*(\beta \circ c_T(\cdot))|}{|L^*(\beta)|}$$

# Tree Structured Models: Bayesian Mixtures of Copula Trees

- Present the model here as the limit as $K \to \infty$ of a finite mixture model with K components
- Let X be a set of random variables, z be an index of the set of all trees T over these variables, and $\Theta$ be the set of copula parameters
- Standard Bayesian mixture model:

$$\begin{aligned} \Lambda &\sim f_\Lambda & T_z &\sim T_0(z) \\ \pi &\sim \text{Dirichlet}(\alpha/K,\ldots,\alpha/K) & \Theta_z &\sim f_\Theta \\ z \mid \pi &\sim \text{Discrete}(\pi_1,\ldots,\pi_K) & \mathbf{X}|z,\mathcal{T},\Theta,\Lambda &\sim f(X \mid T_z,\Theta_z,\Lambda). \end{aligned}$$

- The density for a sample $f(X \mid T_z, \Theta_z, \Lambda)$ is constructed using a copula tree

# Undirected Structure Learning: Parametric Undirected Graph Estimation

- Let H be an undirected graph whose nodes correspond to real-valued random variables $X_1,\ldots,X_n$
- The interdependencies between random variables are characterized by $\Omega = \Sigma^{-1}$
- $X_i$ is independent of $X_j$ given all other variables, denoted by $X_i \perp X_j \mid X_{\setminus\{i,j\}}$ if and only if $\Sigma^{-1}_{ij} = 0$
- $\Sigma$ estimated by finding the solution to the following regularized likelihood objective:
  - $$\widehat{\Omega} = \min_{\Omega} -\frac{1}{2}\left(\log|\Omega| - tr(\Omega\hat{S})\right) + \lambda \sum_{j \neq k} |\Omega_{jk}|$$

# Undirected Structure Learning: Nonparanormal Estimation

- A real-valued random vector X is said to have a nonparanormal distribution, X~NPN(μ,Σ,g), if there exist functions $\{g_i\}^n_{i=1}$ such that $(g_1(X_1),\ldots,g_n(X_n))\sim N(\mu,\Sigma)$
- Define
  - $h_i(x) = \Phi^{-1}(F_i(x_i))$
  - let Λ be the covariance matrix of h(X)
  - independence properties discussed above for the multivariate Gaussian hold so that $X_i \perp X_j | X_{\setminus\{i,j\}}$ if and only if $\Lambda^{-1}_{i,j}=0$
  - Empirical marginal distribution function: $\hat{F}_i(t) \equiv \frac{1}{m}\sum_{l=1}^{m} \mathbf{1}_{\{x_i[l]\leq t\}}$
  - Winsorized estimator:

$$\tilde{F}_i(x) = \begin{cases} \delta_m & \text{if } \hat{F}_i(x) < \delta_m \\ \hat{F}_i(x) & \text{if } \delta_m \leq \hat{F}_i(x) \leq 1-\delta_m \\ (1-\delta_m) & \text{if } \hat{F}_i(x) > 1-\delta_m, \end{cases}$$

  - Transformation function:

$$\tilde{g}_i(x) \equiv \hat{\mu}_i + \hat{\sigma}_i \tilde{h}_i(x)$$

# Undirected Structure Learning: Properties of the Estimator

- The main technical result is an analysis of the covariance of the Winsorized estimator
- Under appropriate conditions:
  - $$\max_{i,j} \left| S_m(\tilde{g})_{ij} - S_m(g)_{ij} \right| = O_P\left(m^{-1/4}\right)$$

# Copula Bayesian Networks: The CBN Network

- We consider the lemma that if the independencies encoded in G hold in *f* X(x), then the joint copula decomposes into a product of local copula ratio terms *Rci* . However, the converse is only partially true.
- Lemma 3.1. *Let f* (*x* | y), *with* y = {*y*1,...,*yk*}, *be a conditional density function. There exists a copula density function c* (*F* (*x*),*F*1(*y*1),...,*FK*(*yK*)) *such that*
- *The converse is also true: for* any *copula, Rc* (*F*(*x*),*F*1(*y*1),...,*FK*(*yK*))*fX*(*x*) *defines a valid conditional density.*

$$f(x \mid \mathbf{y}) = R_c(F(x), F_1(y_1), \ldots, F_K(y_K)) f_X(x),$$

***where $R_c$ is the copula ratio***

$$R_c(F(x), F_1(y_1), \ldots, F_K(y_K)) \equiv \frac{c(F(x), F_1(y_1), \ldots, F_K(y_K))}{\frac{\partial^K C(1, F_1(y_1), \ldots, F_K(y_K))}{\partial F_1(y_1) \ldots \partial F_K(y_K)}},$$

# Copula Bayesian Networks: The CBN Network

- A Copula Bayesian Network (CBN) is a triplet C = (G , Θ*C* , Θ *f* ) that defines *f*X(x). G encodes the independencies {(*Xi* ⊥ ND*i* | Pa*i*)}, assumed to hold in *f*X(x).Θ*C* is a set of local copula functions {*Ci* (*F* (*xi*),*F* (pa*i*1),...,*F* (pa*i ki*))} that are associated with the nodes of G that have at least one parent. In addition, Θ*f* is the set of parameters representing the marginal densities *fi* (*xi*) (and distributions *Fi* (*xi*)).
- The joint density *f* X(x) then takes the form

$$f_{\mathbf{X}}(\mathbf{x}) = \prod_{i=1}^{n} R_{c_i}\left(F(x_i), F(\mathbf{pa}_{i1}), \ldots, F(\mathbf{pa}_{ik_i})\right) f_i(x_i).$$

# Copula Bayesian Networks: The CBN Network

- The above product $\prod_i Rci(\cdot)\, fi(xi)$ *always* defines a valid joint density. However, the product $\prod_i Rci$ , when each copula ratio is constructed independently, does not always define a valid copula.In this case, the marginals of the *valid* joint distribution do not necessarily equal to $Fi(xi)$.
- Assuming the marginals are estimated first, estimation of the entire CBN model decomposes into independent estimation of local copulas. Building on the same decomposability, standard greedy methods for structure learning can also be employed.

# Copula Bayesian Networks: Lightning-speed Structure Learning

- Tackles the challenge of automated structure learning of CBNs in a high dimensional settings. The graph G is constrained to be a tree, the optimal structure can be learned using a maximum spanning tree procedure.
- Guided by model selection score like BIC, building block of all score based models is the evaluation of merit of an edge in the network. In case of CBN model it involves computation of

$$\sum_{l=1}^{m} \log c_{\hat{\theta}}(F_X(x[l]), F_Y(y[l])),$$

# Copula Bayesian Networks: Lightning-speed Structure Learning

- But computing $\hat{\theta}$, as well as the actual computation of the log-likelihood function can be difficult. In fact, for non-Gaussian real-valued models, even the learning of a tree structure can be prohibitive.
- Elidan proposes an alternative that builds on the fact that as *m* grows, the above expression approaches the negative (differential) entropy:
- However, computation of the copula entropy can also be difficult since for most copula families the above integral does not have a closed form.

$$-H(C_\theta(U,V)) = \int c_\theta(u,v)\log c_\theta(u,v)dudv,$$

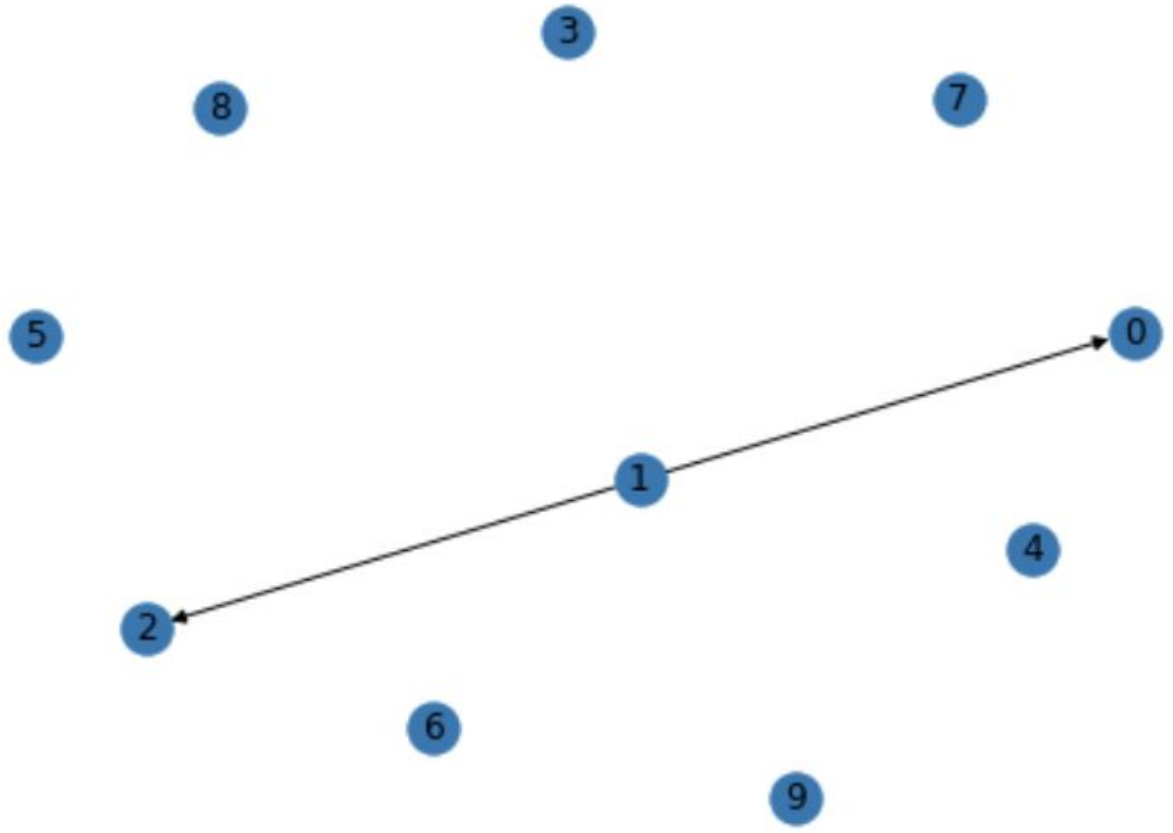
# Copula Bayesian Networks: Lightning-speed Structure Learning

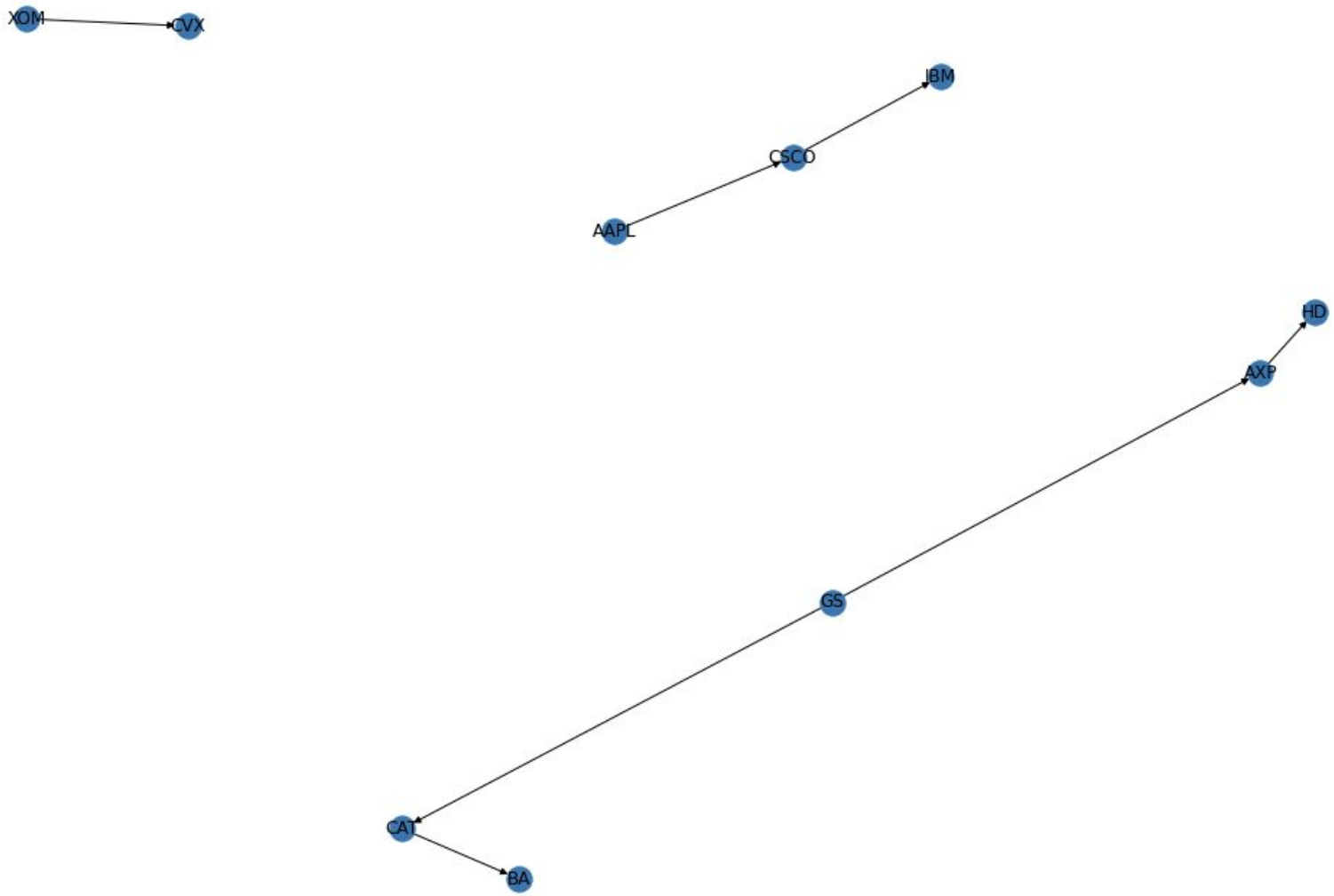
- Instead and efficient compute proxy is The relationship between Spearman's rho rank correlation measure of association and copula function are well known.
- Further, the vast majority of copula families define a concordance ordering where θ2 > θ1 implies $C_{\theta 2}(u, v) > C_{\theta 1}(u, v)$ for all $u, v$. Thus, for most copula families, Spearman's rho is monotonic in the dependence parameter θ .

$$\rho_s(X,Y) = \rho_s(C_\theta) \equiv 12\iint C_\theta(U,V)dudv - 3.$$

- A 100 variable structure, for example, is learned in essentially the same time that it takes to learn the structure of a naive Gaussian BN

# Implementation of CBN Model.




XOM
CVX
IBM
CSCO
AAPL
HD
AXP
GS
CAT
BA

# Copula Processes

- This considers the problem of measuring the dependencies between real-valued measurements of a continuous process.
- These quantities are naturally on different scales and have different marginal distributions. Thus, it is desirable to separate the univariate effect from the dependence structure. Toward this goal, they define a copula process which can describe the dependence between *arbitrarily* many random variables.

# Copula Processes

- Let $\{Xt\}$ be a collection of random variables indexed by $t$ with marginal distributions $Ut \equiv Ft(Xt)$. Let $Gt$ be the marginal distributions of a base process, and let $H$ be the base joint distribution. $Xt$ is a *copula process* with $Gt,H$, denoted $Xt \sim \mathrm{CP}(Gt,H)$, if for every finite set of indices $\mathrm{I} = \{t1,\ldots,tn\}$

$$P\left(\cap_{i=1}^{n}\{G_{t_i}^{-1}(U_{t_i}) \leq a_i\}\right) = H_{t_1,\ldots,t_n}(a_1,\ldots,a_n),$$

- As an example, consider the case where the base measure is a Gaussian process (GP). $Xt$ is a GP if for every finite subset of indices I , the set $\{Xti\}ti \in \mathrm{I}$ has a multivariate Gaussian distribution. To allow for a variable size set I , a GP is parameterized by a mean function $m(t)$ that determines the expectation of the random variable $Xt$, and a kernel function $k(t,t')$ that determines the covariance of $Xt$ and $Xt'$.

# Copula Processes

- When the base measure is chosen to be a GP, we say that $Xt$ has a Gaussian copula process (GCP) distribution. This is equivalent to the existence of a mapping $\Psi$ such that $\Psi(Xt)$ is a GP. We denote this by $Xt \sim \mathrm{GCP}(\Psi, m(t), k(t,t'))$.
- In principle, given complete samples and a known mapping, one can estimate a GCP by simply transforming the data and using black box procedures for GP estimation
- Wilson and Ghahramani, however, consider a more challenging application setting that requires further algorithmic innovation. Concretely, they introduce a volatility model where the unobserved standard deviations of the data follow a GCP distribution

$$\sigma_t \sim \mathrm{GCP}(g^{-1}, 0, k(t,t')).$$

# Copula Processes

- Observations Xt are assumed to be normally distributed.
- Let θ be the parameters that define both the GP covariance function and the warping function. Further, using a different notation from Wilson and Ghahramani to maintain consistency, let $zt = g-1\ (\sigma t\ )$ be the latent function values that have a GP distribution.
- The central components involved in estimating θ from samples *xt* and making prediction at some unrealised time *t*⋆ are:
  - A Laplace approximation for the posterior $f\ (fZ(zt\star)|y,\theta)$.
  - A Markov Chain Monte Carlo technique to sample from this posterior, specifically the elliptical slice sampling method .
  - A flexible parametric as well as nonparametric warping functions to transform the samples into standard deviation space.

# Comparative Summary

| Model | References | variables | Structure | Copula | Comments |
|---|---|---|---|---|---|
| Vines | [3, 1, 25] | $< 10$ in practice | conditional dependence | any bivariate | well understood general purpose framework |
| Nonparametric BBN | [24, 16] | 100s | BN + vines | Gaussian in practice | mature application |
| Tree-averaged | [21, 46] Section 3.1 | 10s | Mixture of trees | any bivariate | requires only bivariate estimation |
| Nonparanormal | [26] Section 3.2 | 100-1000s | MN | Gaussian | high-dimensional estimation with theoretical guarantees |
| Copula networks | [8, 10] Section 3.3 | 100s | BN | any | flexible at the cost of partial control over marginals |
| Copula processes | [53, 18] Section 3.4 | $\infty$ (replications) | - | multivariate | nonparametric generalization of Gaussian processes |

**Table 1** Summary of the different copula-based multivariate models

# Summary

- In the introduction it was argued that, in the context of multivariate modeling, the complementing strengths and weaknesses of the fields of machine learning and that of copulas offer opportunities for symbiotic constructions.
- While discrete high-dimensional modelling has been studied extensively, real-valued modelling for more than a few dimensions is still in its infancy. There exists no framework that is as general and as flexible as copulas for multivariate modelling.
- Thus, it is inevitable that machine learning researchers who aim to stop discretizing data, will have to pay serious attention to the power of copulas. Conversely, if researchers in the copula community aim to cope with truly high-dimensional challenges, algorithmic prowess, a focus of the machine learning community, will have to be used.