Copulas in Machine Learning

Project Report: Sarthi Shah[114173920] & M V D Satya Swaroop [114388425]

Abstract:

The purpose of this paper is to survey recent copula-based constructions in the field of machine learning, so as to provide a stepping stone for those interested in further exploring this emerging symbiotic research.

1. Introduction:

Despite overlapping goals of multivariate modelling and dependence identification, until recently the fields of machine learning in general and probabilistic graphical models in particular have been ignorant of the framework of copulas. Multivariate modelling Unfortunately, high-dimensional modelling in the context of finite data and limited computational resources can be quite challenging, and susceptible to the curse of dimensionality. Probabilistic graphical models is a framework for coping up with this task. These models are used to represent multivariate densities via a combination of a qualitative graph structure that encodes independencies and local quantitative parameters . The purpose of this paper is to survey these works. Rather than aiming at a complete coverage, the focus is on multivariate constructions.

2. Background:

2.1 Copulas

A copula function links univariate marginal distributions to form a joint multivariate one.

Definition 2.1. Let $U_1,...,U_n$ be real random variables marginally uniformly distributed on [0, 1]. A copula function $C : [0, 1]^n \rightarrow [0, 1]$ is a joint distribution

$$C(u_1,\ldots,u_n) = P(U_1 \leq u_1,\ldots,U_n \leq u_n).$$

We will use $C_\theta(\cdot)$ to denote a parameterised copula function where needed. When the marginals are continuous, $C(\cdot)$ is uniquely defined. Any copula function taking any univariate marginal distributions $\{F_i(x_i)\}$ as its arguments, defines a valid joint distribution with marginals $\{F_i(x_i)\}$. Thus, copulas are "distribution generating" functions that allow us to separate the choice of the univariate marginals and that of the dependence structure, encoded in the copula function $C(\cdot)$.

$$f_{\mathbf{X}}(\mathbf{x}) = \frac{\partial^n C(F_1(x_1),\ldots,F_n(x_n))}{\partial F_1(x_1)\ldots\partial F_n(x_n)} \prod_i f_i(x_i) \equiv c(F_1(x_1),\ldots,F_n(x_n)) \prod_i f_i(x_i), \quad (1)$$

2.2 Probabilistic Graphical Models

A directed graph is a set of nodes connected by directed edges. A directed acyclic graph (DAG) G is a directed graph with no directed cycle. The parents of a node V in a directed graph is the set of all nodes U such that there exists a direct edge from U toV.A node U is an ancestor V in the graph if there is a directed path from U to V.

Directed graphical models or Bayesian networks (BNs), use a DAG G whose nodes correspond to the random variables of interest $X_1,...,X_n$ to encode the in- dependencies $I(G) = \{(X_i \perp ND_i \mid Pa_i)\}$, where $\perp$ denotes the independence relationship, and ND_i are nodes that are not descendants of X_i in G (independencies that follow from I(G) are easily identifiable via an efficient algorithm).

$$f_{\mathbf{X}}(\mathbf{x}) = \prod_{i=1}^{n} f_{X_i|\mathbf{Pa}_i}(x_i \mid \mathbf{pa}_i),$$

where Pa_i are the parents of node X_i in G .Undirected graphical models, or Markov Networks (MNs), use an undirected graph H that encodes the independencies $I(H) = \{(X_i \perp X \setminus \{X_i\} \cup Ne_i \mid Ne_i)\}$, where Ne_i are the neighbours of X_i in H . That is, each node is independent of all others given its neighbours in H . Let C be the set of cliques in H (a clique is set of nodes such that each node is connected to all others in the set).

$$f_{\mathbf{X}}(\mathbf{x}) = \frac{1}{Z} \prod_{c \in \mathscr{C}} \phi_c(\mathbf{x}_c),$$

where X_C are the set of nodes in the clique c, and $\phi_C : R^{|C|} \rightarrow R^+$ is any positive function over the values of these nodes. Z is a normalising constant called the partition function.

3 Multivariate Copula-based Construction

3.1.1 Tree Structured Models

Let T be an undirected tree structured graph (i.e., a graph with no cycles) and let E denote the set of edges in T that connect two vertices. From the Hammersley- Clifford decomposition of , it easily follows that, if the independencies I(T) hold in $f_X(x)$, then it can be written as

$$f_{\mathbf{X}}(\mathbf{x}) = \left[\prod_i f_i(x_i)\right] \prod_{(i,j) \in \mathscr{E}} \frac{f_{ij}(x_i, x_j)}{f_i(x_i) f_j(x_j)}.$$

Using (1), a decomposition of the joint copula also follows

$$c_T(\cdot) = \frac{f_{\mathbf{X}}(\mathbf{x})}{\prod_i f_i(x_i)} = \prod_{(i,j) \in \mathscr{E}} \frac{f_{ij}(x_i, x_j)}{f_i(x_i) f_j(x_j)} = \prod_{(i,j) \in \mathscr{E}} c_{ij}(F_i(x_i), F_j(x_j)), \qquad (4)$$

where $c_T(\cdot)$ is used to denote a copula density that corresponds to the structure T , and $c_{ij}(\cdot)$ is used to denote the bivariate copula corresponding to the edge (i, j). The converse composition also holds: a product of local bivariate copula densities, each associated with an edge of T , defines a valid copula density.

3.1.2 Tree-averaged Copulas

As noted, the main appeal of the tree-structured copula is that it relies solely on bivariate estimation. However, this comes at the cost of firm independence assumptions. 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

$$P(T \in \mathcal{T} \mid \beta) = \frac{1}{Z} \prod_{(u,v)\in\mathcal{E}_T} \beta_{uv},$$

where Z is a normalization constant. Using a generalization of the Laplacian matrix:

$$L_{uv}(\beta) = \begin{cases} -\beta_{uv} & u \neq v \\ \sum_w \beta_{uw} & u = v, \end{cases}$$

it can be shown that the normalisation constant Z is equal to the determinant |L*(β)|, where L*(β) represents the first (n−1) rows and columns of L(β). This result can then be used to efficiently compute the density of the average of all copula spanning trees, which itself is also a copula density:

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

3.1.3 Bayesian Mixtures of Copula Trees

The all tree mixture model described in the previous section overcomes some of the limitations imposed by a single tree model. However, to facilitate computational efficiency, the prior used involves heavy parameter sharing. Let X be a set of random variables, z be an index of the set of all trees T over these variables, and Θ be the set of copula parameters, one for each pair of variables. The following model is a standard Bayesian mixture model, with the novelty that the parameters of the univariate marginals Λ are shared by all mixture components:

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

3.2 Undirected Structure Learning

3.2.1 Parametric Undirected Graph Estimation

Let H be an undirected graph whose nodes correspond to real-valued random variables $X_1,...,X_n$. For multivariate Gaussian distributions, the independencies between the random variables as encoded by the graph's structure are characterised by the inverse covariance matrix $\Omega = \Sigma^{-1}$. Σ be estimated by finding the solution to the follow- ing regularised likelihood objective:

$$\widehat{\Omega} = \min_{\Omega} -\frac{1}{2}\left(\log|\Omega| - tr(\Omega\hat{S})\right) + \lambda \sum_{j\neq k} |\Omega_{jk}|,$$

3.2.2 Non-paranormal Estimation

A real-valued random vector X is said to have a non-paranormal distribution, $X\sim NPN(\mu,\Sigma,g)$, if there exist functions $\{g_i\}^n_{i=1}$ such that $(g_1(X_1),\ldots,g_n(X_n))\sim N(\mu,\Sigma)$, Defined as,

$$h_i(x) = \Phi^{-1}(F_i(x_i)),$$

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 $\Lambda^{-1}{}_{i\,j}=0$. 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:
where $\hat{\mu}_i$ and $\hat{\sigma}_i$ are sample mean and standard deviation of X_i, respectively. The sample covariance matrix $S_m(\tilde{g})$ can now be plugged in (5) in place of $\hat{S}$, defining a two-step estimation procedure for the estimator $\hat{\Omega}_m$:

1. Replace the observations with Winsorized normalised scores as defined in
2. Use the graphical lasso to estimate the undirected graph.

3.3 Copula Bayesian Networks

3.3.1 CBN Model

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 = {y1,...,yk}, be a conditional density function. There exists a copula density function c (F (x),F1(y1),...,FK(yK)) such that.

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

where R_c is the copula ratio

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

The converse is also true: for any copula, Rc (F(x),F1(y1),...,FK(yK))fX(x) defines a valid conditional density.

A Copula Bayesian Network (CBN) is a triplet C = (G , ΘC , Θ f) that defines fX(x). G encodes the independencies {(Xi ⊥ NDi | Pai)}, assumed to hold in fX(x).ΘC is a set of local copula functions {Ci (F (xi),F (pai1),...,F (pai ki))} that are associated with the nodes of G that have at least one parent. In addition, Θf 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

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

The above product ∏*i Rci* (·) *fi*(*xi*) *always* defines a valid joint density. However, the product ∏*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*).

3.3.2.1 CBN Model: Lightning-speed Structure Learning

Tackles the challenge of automated structure learning of CBNs in a high- dimensional settings. When the graph G is constrained to be a tree, the optimal structure can be learned using a maximum spanning tree procedure. The building block of essentially all score-based structure learning methods for graphical models is the evaluation of the merit of an edge in the network. This involves computing the likelihood gain that would result from adding an edge to the network. Which in turn involves estimation of the bivariate maximum likelihood parameters. In the case of the CBN model, this involves computation of

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

Unfortunately, estimating $\hat{\theta}$, 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:

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

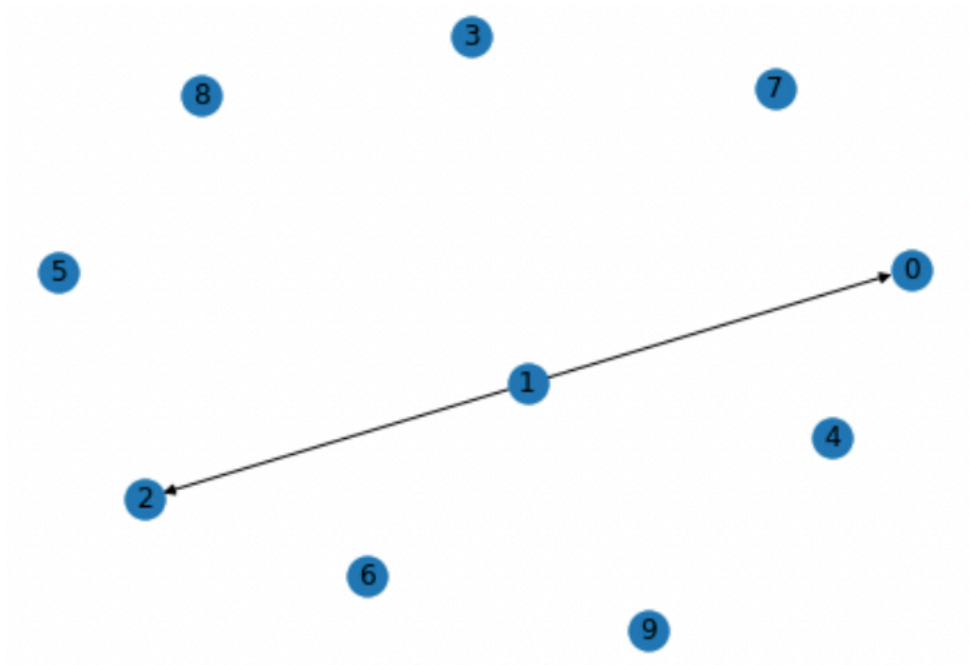
Instead, an efficient to compute proxy is proposed. The relationship between Spearman's rho rank correlation measure of association and the copula function can be easily shown

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

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.

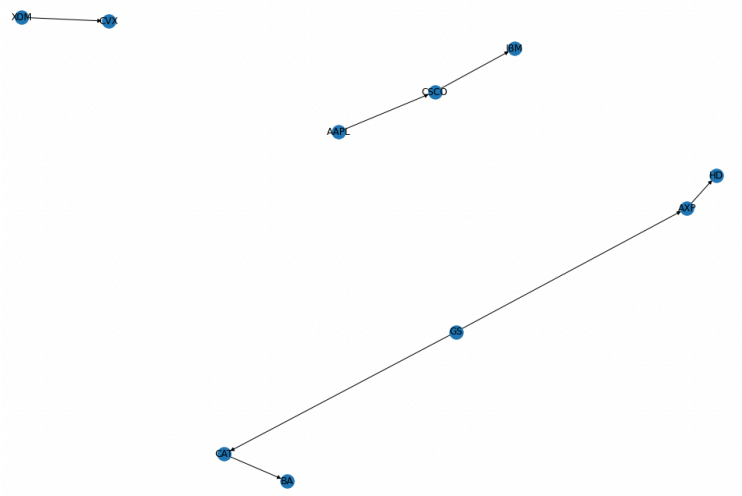
3.3.2.2 Implementation of CBN Model

Basic Implementation of CBN Network using MLE Likelihood Estimator



Structure

After finding automated structure learning using spearman's rho coefficient.



This code was implemented on DowJones 30 stocks with last decade of data.

3.3.2 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.

Let {*Xt* } be a collection of random variables indexed by *t* with marginal distributions *Ut* ≡ *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* ~ CP(*Gt*,*H*), if for every finite set of indices I = {*t*1,…,*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 ∈I has a multivariate Gaussian distribution. 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 Ψ such that Ψ (*Xt*) is a GP. We denote this by *Xt* ~ GCP(Ψ,*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

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

procedures for GP estimation.

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 *z t* = *g*−1 (σ*t*) be the latent function values that have a GP distribution.

The central components involved in estimating θ from samples *x t* and making prediction at some unrealised time $t^{\star}$ are:

- A Laplace approximation for the posterior f (fZ(z t⋆)ly,θ).
- 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.

4. 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	∞ (replications)	-	multivariate	nonparametric generalization of Gaussian processes

Table 1 Summary of the different copula-based multivariate models

5. Summary :

In the introduction it was argued that, in the context of multivariate modeling and information estimation, the complementing strengths and weaknesses of the fields of machine learning and that of copulas offer opportunities for symbiotic constructions. This paper surveyed the main such synergic works that recently emerged in the machine learning community. While discrete high-dimensional modeling has been studied extensively, real- valued modeling 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 modeling.

Thus, it is inevitable that machine learning researchers who aim to stop discretising 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.

6. References

1. Aas,K.,Czado ,C.,Frigessi ,A.,Bakken ,H.:Pair-copula constructions of multiple dependencies. Insurance: Mathematics and Economics 44, 182–198 (2009)

2. Abayomi,K.:Diagnostics for Multivariate Imputation,CopulaBasedIndependentComponent Analysis and a Motivating Example. Columbia University (2008)

3. Bedford,T.,Cooke,R.:Vines-a new graphical model for dependent random variables.Annals of Statistics (2002)

4. Borgwardt,K.M.,Gretton ,A.A.,Rasch ,B.M.J.,Peter Kriegel,H., Schlkopf,A.B.,D,B.A.J.S.: Integrating structured biological data by kernel maximum mean discrepancy. In: Proceedings of Intelligent Systems for Molecular Biology (ISMB) (2006)

5. Brechmann,E.C.,Czado C.,Aas,K.:Truncated regular vines in high dimensions with applications to financial data. Canadian Journal of Statistics 40(1), 68–85 (2012)

6. Chow, C.K., Liu, C.N.: Approximating discrete probability distributions with dependence trees. IEEE Trans. on Info. Theory 14, 462–467 (1968)

7. Darsow, W., Nguyen, B., Olsen, E.: Copulas and Markov processes. Illinois Journal of Mathematics 36, 600–642 (1992)

8. Elidan, G.: Copula Bayesian networks. In: Neural Info. Processing Systems (NIPS) (2010)

9. Elidan,G.:Inference-less density estimation using copula bayesian networks.In:Uncertainty in Artificial Intelligence (UAI) (2010)

10. Elidan,G.:Lightning-speed structure learning of non linear continuous networks.In:Proceedings of the AI and Statistics Conference (AISTATS) (2012)

11. Embrechts,P.,Lindskog, F.,McNeil,A.:Modeling dependence with copulas and applications to risk management. Handbook of Heavy Tailed Distributions in Finance (2003)

12. Ferguson,T.S.:A Bayesian analysis of some non parametric problems . The Annals of Statistics 1(2), 209–230 (1973)