Machine Learning manuscript No.
(will be inserted by the editor)

Multi-view Clustering of Mixed Data with Vine Copulas: Beyond Meta-Gaussian Dependencies

Lavanya Sita Tekumalla · Vaibhav Rajan ·
Chiranjib Bhattacharyya

Received: date / Accepted: date

Abstract Copulas enable parameterization of multivariate distributions in terms of constituent marginals and dependence families. Vine copulas, hierarchical collections of bivariate copulas, can model a wide variety of dependencies in multivariate data including asymmetric and tail dependencies which the more widely used Gaussian copulas, used in Meta–Gaussian distributions, cannot. However, current inference algorithms for vines, cannot fit data with mixed – a combination of continuous, binary and ordinal – features that are common in many domains. We design a new inference algorithm to fit vines on mixed data thereby extending their use to several applications. As an illustration a dependency–seeking multi–view clustering model based on Dirichlet Process mixture of vines is designed that generalizes previous models to arbitrary dependencies as well as to mixed marginals. Empirical results on synthetic and real datasets demonstrate the performance on clustering single–view and multi–view data with asymmetric and tail dependencies and with mixed marginals.

Keywords Vine Copula · Mixed Data · Multi–View · Dependency–Seeking Clustering

1 Introduction

Copulas are increasingly popular in machine learning due to the modular parameterization of multivariate distributions they provide: the choice of arbitrary

Lavanya Sita Tekumalla
Indian Institute of Science
E-mail: lavanya.iisc@gmail.com

Vaibhav Rajan
Xerox Research Centre India
E-mail: vaibhav.rajan@xerox.com

Chiranjib Bhattacharyya
Indian Institute of Science
E-mail: chiru@csa.iisc.ernet.in

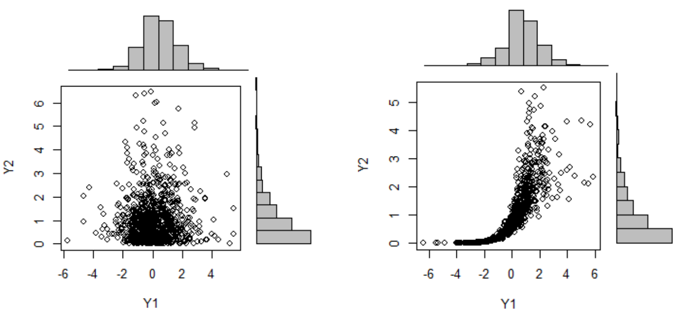


Fig. 1: Bivariate Gaussian copula (left) and Clayton copula (right) samples, both with marginals T distribution with 5 degrees of freedom (Y1) and exponential distribution with rate 1 (Y2), illustrating symmetric and asymmetric dependencies respectively. The Gaussian Copula (with meta–Gaussian dependencies) on the left cannot model data with asymmetric dependencies shown on the right.

marginal distributions that are independent of the *dependency models* from different copula families. With the Gaussian copula itself, using different marginals, many different, including multimodal, joint distributions can be constructed, called *meta–Gaussian* distributions, that have been used in several applications (Eban et al. 2013; Bayestehtashk and Shafran 2013; Letham et al. 2014; Eickhoff et al. 2015). However, meta–Gaussian dependencies from the Gaussian copula do *not* include asymmetric and tail dependencies that are captured by other copula families (Joe 2014): see figure 1.

Finding dependencies in data is also one of the goals of techniques like mixture models and canonical correlation analysis (CCA). Copulas have been used with CCA–based models like dependency–seeking clustering, by Rey and Roth (2012), for multi–view data with arbitrary continuous marginals. But their use is again restricted to Gaussian copulas (and so, only meta–Gaussian dependencies). Further, they are limited to modeling data with continuous–valued features only.

Real world data often has *mixed* (continuous, binary and ordinal valued) features as well as asymmetric and tail dependencies. Among the copula families, vine copulas provide a flexible and intuitive way of pair–wise dependency modeling using hierarchical collections of bivariate copulas, each of which can belong to any copula family thereby capturing a wide variety of dependencies (Aas et al. 2009). Many standard copulas, like Gaussian, can be considered as special cases of vines. However there are no existing techniques to fit vine copulas on mixed data.

The use of copulas with discrete data has remained difficult – the copula is not margin–free and may not be identifiable – nevertheless they can be effectively used (Genest and Neslehova 2007); in the case of vines, the challenge lies mainly in parameter inference. Two previous approaches have been proposed, both for only discrete (not mixed) features, and both requiring expensive estimation of marginals: one by Panagiotelis et al. (2012) and another by Smith and Khaled (2012). The latter can be extended to mixed data but their MCMC algorithm requires computations that are exponential in data dimensions per sampling step, making it practically infeasible.

We propose a new MCMC–based inference algorithm for vines that can fit mixed data and is quadratic in data dimensions per sampling step. Empirically, it is faster than the algorithm of Panagiotelis et al. (2012) for discrete marginals and it yields more accurate parameter estimates, in both the continuous and discrete case, than the current best estimators. Our sampling scheme bypasses the costly estimation of marginals using a rank–based likelihood (Hoff 2007) to obtain approximate parameter estimates.

We illustrate the use of our inference technique in the task of multi-view dependency seeking clustering. We propose a Dirichlet Process (DP) Mixture of Vine copulas that aims to cluster multi-view data based on both intra– and inter–view dependencies. Our model extends the model of Kim et al. (2013) and Rey and Roth (2012) to arbitrary mixed marginals and dependencies beyond meta–Gaussian. The flexibility of the model comes with its challenges in fitting mixed data and non–conjugacy of priors for the latent variables in our model. We design an inference algorithm that overcomes both these hurdles by extending our inference technique for vines. Our empirical comparisons with Rey and Roth's method show that on data with asymmetric or tail dependencies (e.g. figure 1 (right)) our model outperforms baselines.

To summarize, our contributions are:

1. Vines enable modeling asymmetric and tail dependencies beyond the capability of the Meta-Gaussian. We take the first step to fit vines for mixed data (with arbitrary continuous, ordinal and binary marginals) by proposing a new MCMC inference algorithm with time complexity (per sampling step) quadratic in the data dimensions and linear in the number of bivariate copulas used.
2. A dependency–seeking multi–view clustering model based on DP-mixture of vines that models both inter-view and intra-view dependencies and generalizes the model of Rey and Roth (2012) to arbitrary dependencies (beyond meta-Gaussian) as well as to mixed marginals.
3. Empirical results on synthetic and real datasets demonstrating the scalability and accuracy of our inference algorithm and performance on clustering single–view and multi–view data with asymmetric and tail dependencies and with mixed marginals.

2 Background and Related Work

Copula: An M–dimensional copula is a multivariate distribution function $C : [0,1]^M \mapsto [0,1]$. A theorem by Sklar (1959) proves that copulas can uniquely characterize continuous joint distributions. It shows that for every joint distribution with continuous marginals, $F(X_1, \ldots, X_M)$, there exists a unique copula function C such that

$$F(X_1, \ldots, X_M) = C(F_1(X_1), \ldots, F_M(X_M))$$

as well as the converse. In the discrete case, the copula is uniquely determined on $Ran(F_1) \times \ldots \times Ran(F_p)$, where $Ran(F_j)$ is the range of marginal F_j. The joint density function p can be expressed as:

$$p(X_1, \ldots, X_M) = c(F_1(X_1), \ldots, F_M(X_M)).p_1(X_1) \ldots p_M(X_M)$$

for strictly increasing and continuous marginals F_j and copula density c.

The Gaussian copula, for a correlation matrix $\Sigma \in \mathbb{R}^{M \times M}$, is given by $c(u_1, \ldots, u_M; \Sigma) = \Phi_\Sigma\left(\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_M)\right)$, where $u_j = F_j(X_j)$, Φ^{-1} is the inverse CDF of a standard normal and Φ_Σ is the joint CDF of a multivariate normal with mean zero and correlation matrix Σ. To see why Gaussian copulas cannot model tail dependencies (Joe 2014), consider the bivariate case with marginal variables, $X_1 \sim F_1, X_2 \sim F_2$. The upper tail dependence coefficient is given by $\lambda_U = \lim\limits_{\alpha \to 1} P(X_2 > F_2^{-1}(\alpha)|X_1 > F_1^{-1}(\alpha))$. Similarly we have the lower tail dependence coefficient $\lambda_L = \lim\limits_{\alpha \to 0} P(X_2 \leq F_2^{-1}(\alpha)|X_1 \leq F_1^{-1}(\alpha))$. For the Gaussian copula, $\lambda_U = \lambda_L = 2 \lim\limits_{x \to -\infty} \Phi\left(x \frac{\sqrt{1-\rho}}{\sqrt{1+\rho}}\right) = 0$, that is, irrespective of the correlation ρ chosen, tail events occur independently in X_1, X_2 and the tail dependency cannot be modeled. Also, since $\lambda_U = \lambda_L$, it can only model symmetric dependencies.

An example of a copula that can capture such dependencies is the Clayton copula, $C(u_1, u_2; \alpha) = \max((u_1^{-\alpha} + u_2^{-\alpha} - 1)^{-1/\alpha}, 0)$ which exhibits lower tail dependence. Parameter α controls the dependence with perfect dependence if $\alpha \to \infty$ and independence if $\alpha \to 0$. See figure 1 for a comparison. See Joe (2014) for a comprehensive treatment of copulas and Elidan (2013) for a machine learning perspective.

Vine Copula: Vines are hierarchical collections of bivariate copulas. They model complex dependencies in multivariate data with well-studied bivariate copulas as building blocks. Since the number of pair copula decompositions is very large for high dimensions, special graphical models have been introduced that constrain the structure of the decompositions. Any multivariate density, is decomposable into conditional densities: $p(X_1, \dots, X_M) = p(X_M).p(X_{M-1}|X_M) \dots p(X_1|X_2, \dots X_M)$, thereby can be written as functions of bivariate copula densities by expanding the conditional using the following identity for any set of random variables $\tilde{Q}, Q_1, \dots, Q_L$:

$$p(\tilde{Q}|Q_1, \dots, Q_L) = c_{\tilde{Q}, V_j|V_{-j}}(F(\tilde{Q}|Q_{-j}), F(Q_j|Q_{-j})).p(\tilde{Q}|Q_{-j})$$

where Q_{-j} denotes the set $\{Q_1, \dots, Q_{j-1}, Q_{j+1}, \dots, Q_L\}$ (Aas et al. 2009). This forms the basis of the hierarchical vine structure. A **D-vine** has $M-1$ hierarchical trees and $M(M-1)/2$ bivariate copulas for M dimensional data.

The general expression for the density $p(X_1, \dots X_M)$ of a D-vine is given by

$$\prod_{j=1}^{M} p(X_j) \prod_{t=1}^{M-1} \prod_{s=1}^{M-t} c_{s,s+t|s+1,\dots,s+t-1}(F^1_{s,t}, F^2_{s,t}) \tag{1}$$

comprising of $\binom{M}{2}$ bivariate copulas $\{c_{s,s+t|s+1,\dots,s+t-1}\}$ where index s identifies the trees and t iterates over the edges in each tree; $F^1_{s,t} = F(X_s|X_{s+1}, \dots, X_{s+t-1})$, $F^2_{s,t} = F(X_{s+t}|X_{s+1}, \dots, X_{s+t-1})$. The conditional distributions in the pair copula constructions, can be recursively evaluated using *h-functions* (Aas et al. 2009) for any set of random variables $\tilde{Q}, Q_1, Q_2, \dots, Q_L$:

$$F(\tilde{Q}|Q_1, \dots, Q_L) = \frac{\partial C_{\tilde{Q}, Q_j|Q_{-j}}(F(\tilde{Q}|Q_{-j}), F(Q_j|Q_{-j}))}{\partial F(Q_j|Q_{-j})}. \tag{2}$$

For example, $p(X_1|X_2, X_3) = c_{13|2}(F(X_1|X_2), F(X_3|X_2))p(X_1|X_2)$ and $p(X_2|X_3) = c_{23}(F(X_2), F(X_3)).p(X_2)$.

Figure 2 shows a D-Vine for the three dimensional case with density,

$$p(X_1, X_2, X_3) = f_1(X_1).f_2(X_2).f_3(X_3)c_{12}(F_1(X_1), F_2(X_2)).c_{23}(F_2(x_2), F_3(x_3)).$$
$$c_{13|2}(F(X_1|X_2), F(X_3|X_2))$$

At the lowest level, each input variable is associated with a node (1,2 and 3) and edges represent bivariate copulas (c_{12}, c_{23}). Nodes at subsequent levels (12 and 23) represent conditional distributions obtained from the nodes of the previous level and edges represent conditional copulas ($c_{13|2}$) which are evaluated using the appropriate h-functions. During estimation the data at the lowest level are the transformed input data (transformed via rank or CDF transformations) and at each subsequent level they are obtained using h-functions.

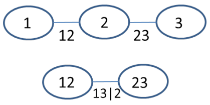


Fig. 2: 3-dimensional D-Vine structure with 2 trees. See text for more details.

Analytic expressions for h-functions have been derived for commonly used copulas; see Aas et al. (2009) for more details and an introduction to vines. Thus all the densities may be expressed in terms of univariate marginals and bivariate copulas. The advantage of such a model is that not all the bivariate copulas have to belong to the same family thus enabling us to model different kinds of bivariate dependencies. See Panagiotelis et al. (2012) for an example on how different copula families, even in a discrete D-vine, has a substantial impact on the joint probabilities of the multivariate distribution. In this paper we describe our models using D-Vines but the techniques are readily extended to other regular vines.

Parameter Estimation for Discrete Vines: For vines with discrete margins, Smith and Khaled (2012) propose a sampling scheme, that is extensible to mixed data, but requires $\mathcal{O}(2^M)$ computations per sampling step of their MCMC algorithm which uses a data augmentation approach to compute the probability mass function (pmf). Panagiotelis et al. (2012) derive a decomposition of the pmf that requires only $2M(M-1)$ evaluations of bivariate copula functions in the vine. But their method cannot be used with vines with mixed margins. Further, their recommended estimation method is the two–step IFM approach (Joe 2014; Panagiotelis et al. 2012) where the marginals are estimated first and then ML estimates of parameters are obtained using non–linear maximization methods such as gradient ascent that are fraught with problems due to local maxima.

Multi–view Dependency Seeking Clustering: Finding linear inter–view dependencies through CCA has been extended in several ways in recent years to capture non–linear dependencies (eg. kernelized CCA (Shawe-Taylor and Cristianini 2004)) and non–normal distributions (eg. exponential CCA (Klami et al. 2010)). Dependency–seeking multi-view clustering aims to cluster co–occurring samples in multiple views so that given the clustering, views become independent. The model of Klami and Kaski (2008), for two views X and Y is

$$Z \sim \text{Mult}(\theta), (X,Y)|Z \sim \mathcal{N}_{p+q}(\mu_z, \Psi_z)$$

where Ψ_z has a block diagonal structure which implies that given the cluster assignment, the views are independent. To address the problem of non–normally distributed data (that results in model mismatch and erroneously large number of uninterpretable clusters) this model is extended by Rey and Roth (2012) who use a Gaussian copula in place of $\mathcal{N}_{p+q}$ thus enabling discovery of non-normally distributed clusters. Their final model is a Dirichlet Process (DP) mixture of Gaussian copulas. However, their method (and others mentioned above) is limited to capturing meta–Gaussian dependencies and cannot be used on mixed data. Model based clustering techniques such as that in Yerebakan et al. (2014) attempt to capture more complex continuous densities by modeling each mixture component with a multimodal densities based on an Infinite Gaussian mixture. However their technique is neither suited for a multi-view setting, nor for handling complex dependencies in mixed data, that constitutes the focus of our paper.

Clustering mixed data: Recent model–based clustering methods to fit mixed data have been designed by McParland and Gormley (2015) and Browne and McNicholas (2012) that use latent variable approaches, similar to ours, but assume Gaussian distribution; and by McParland et al. (2014) who use a mixture of factor analyzers model. Recent copula–based models include a mixture of D–Vines by Kim et al. (2013) that can only fit continuous data. A more general mixture of copulas by Kosmidis and Karlis (2015) mentions possible extensions to discrete and mixed data. For several copula families their algorithm scales exponentially with dimensions rendering them impractical. For vines, that capture more complex dependencies and constitute our main focus, they do not discuss mixed data extensions and for discrete vines they suggest the same PMF decomposition of Panagiotelis et al. (2012) that we compare with and significantly outperform in performance and accuracy (fig 3) in our experiments.

Correlation clustering also attempts to find clusters based on dependencies and is typically PCA–based. E.g. INCONCO Plant and Böhm (2011) that can be used with mixed data but models dependencies by distinct Gaussian distributions for each category of each discrete feature. While SCENIC (Plant 2012), that is empirically found to outperform INCONCO, is not as restrictive in the dependencies, it also is limited by the fact that it assumes a Gaussian distribution to find a low–dimensional embedding of the data. Note that these methods are not suited for multi-view clustering; we use SCENIC and ClustMD (McParland and Gormley 2015) as baselines in single-view settings only.

3 D-Vines for Mixed Data

Our approach involves a generative formulation for D-vines where we explicitly introduce marginals for each datapoint as latent variables.

Generative formulation for D-vines. Consider N observations of M-dimensional data $\mathbf{X} = \{X_{i,j}\}$. Let $\mathbf{U} = \{U_{i,j}\} \in [0,1]^{N \times M}$ be a set of continuous latent variables. A generative formulation for D-vine can be defined as follows. We first sample $U_{i,j}, \forall i, j$ from a D-Vine with *uniform marginals.* The observed data $X_{i,j}, \forall i, j$ is generated by invoking the quantile function of the corresponding marginal variable $U_{i,j}$. We note that the actual marginal distributions $\{F_j\}$ need not be continuous, which enables us to model mixed data. Further, to facilitate Bayesian inference on the parameters Θ and Σ of the D-vine, we introduce appropriate priors. (Summarized in equation 3.)

$$\begin{gathered}\forall 1 \leq s < t \leq M, \theta_{s,t} \sim \text{Unif}(1:T), \quad \sigma_{s,t}|\theta_{s,t} \sim \text{Prior}(\sigma_{s,t}) \\ \forall i \in [N], U_{i,.} \sim DVine_{Unif}(\Sigma, \Theta) \\ \forall j \in [M], \forall i \in [N], X_{i,j} = F_j^{-1}(U_{i,j}) \end{gathered} \tag{3}$$

$\Theta = \{\theta_{s,t} \in [T] : 1 \leq s < t \leq M\}$ denotes the set of $\binom{M}{2}$ bivariate pair-copula families, chosen from a set of T families. We place a uniform prior on $\theta_{s,t}, \forall s, t$ to select each copula family with a probability $\frac{1}{T}$. Σ is the collection of parameters of all the constituent bivariate copulas in the D-vine definition. We place a uniform prior over the support of the parameters in $\Sigma_{s,t} \forall s, t$. We also note that alternate priors exploiting conjugacy are preferable where permissible. For instance, for bivariate Gaussian copula, we place an inverse Wishart prior exploiting conjugacy.

3.1 Inference

Exact inference for this problem is intractable and we propose an approximate inference algorithm for vines for mixed data based on Gibbs sampling using the *extended rank likelihood* (Hoff 2007) approach that bypasses the estimation of margins and thus can accommodate both continuous and discrete ordinal margins. Further, due to the non-conjugacy of priors, our Gibbs Sampling steps are interspersed with Metropolis Hastings steps, similar to the sampling approaches found in Neal (2000) and Meeds et al. (2007).

Consider data $\mathbf{X} = \{X_{i,j}\}$ and latent variables $\{U_{i,j}\}$ introduced in our D-vine generative model. Without any knowledge of the marginals $\{F_j\}$ and without observing $\mathbf{U}$, (which may be discrete or continuous), observing $\mathbf{X}$ tells us that $\mathbf{U}$ must lie in the set (following the same rank constraints as in $\mathbf{X}$):

$$D = \{\mathbf{U}' \in [0,1]^{N \times M} : \forall i \in [N], j \in [M],$$
$$max\left\{{U'}_{rj} : X_{rj} < X_{ij}; r \in [N]\right\} < {U'}_{ij} < min\left\{{U'}_{rj} : X_{ij} < X_{rj}, r \in [N]\right\}\}$$

since marginals are non-decreasing. The occurrence of this event is considered as our data. The rank likelihood is given by:

$$P(\mathbf{U} \in D | \Sigma, F_1, \ldots F_M) = \int_D P(\mathbf{U}|\Sigma) d\mathbf{U} = P(\mathbf{U} \in D | \Sigma)$$

Since the rank likelihood function is based on the marginal probability of an event that is a superset of observing the ranks (i.e. the event D), it is also referred to as the extended rank likelihood.

Our Gibbs sampling scheme is as follows. The latent variables for which to compute Gibbs sampling updates during inference are $\{U_{i,j}\}$, Θ and Σ. Our strategy comprises of first sampling $\{U_{i,j}\}$ from D-vine with uniform marginals subject to rank based constraints that follow from the extended rank likelihood methodology, followed by sampling Σ and Θ conditioned on the $\{U_{i,j}\}$ random variables.

An important aspect of this inference process is rank constrained sampler from a D-Vine with uniform marginals, that we now discuss. Consider the set

$$D_{i,j} = \{u \in [0,1] : max\left\{U_{rj} : X_{rj} < X_{ij}, r \in [N]\right\} < u < min\left\{U_{rj} : X_{ij} < X_{rj}, r \in [N]\right\}\} \tag{4}$$

Let $D_{i,.}$ denote the set $D_{i,1} \times D_{i,2} \times \ldots \times D_{i,M}$. We block sample the random variables $U_{i,.}$ from $p(U_{i,.}|\Sigma, \Theta, U_{-i,.}, U_{i,.} \in D_{i,.})$ which is a truncated D-vine distribution due to rank constraints. However sampling directly from this distribution is hard, and so we use the Metropolis Hastings (MH) algorithm to draw a sample

Algorithm 1: Rank based sampler for D-vines with Uniform Marginals

```
for each $i = 1, \dots, N$ do
    $U^{old}i, . = U_{i,.}$
    $U^{Low}_{i,j} = max\{U_{r,j} : X_{r,j} < X_{i,j}, r \in [N]\}$ , $U^{High}_{i,j} = min\{U_{r,j} : X_{i,j} < X_{r,j}, r \in [N]\}$
    Generate an MH sample as follows:
    for each $j = 1, \dots, M$ do
        if $j == 1$ then
            $U^{new}_{i,1} \sim unif(U^{Low}_{i,j}, U^{High}_{i,j})$
        else
            $R^{low} = F(U^{Low}_{i,j}|U^{new}_{i,1}, \dots, U^{new}_{i,j-1}), R^{high} = F(U^{High}_{i,j}|U^{new}_{i,1}, \dots, U^{new}_{i,j-1}),$
            $R \sim unif(R^{low}, R^{high})$
            for $l$ in $2 : j-1$ do
                $R = h^{-1}(R, F(U^{new}_{i,l-1}|U^{new}_{i,l}, \dots, U^{new}_{i,j-1})$
            $U^{new}_{i,j} = h^{-1}(R, U^{new}_{i,j-1})$
    Accept $U^{new}_{i,j}$ if $unif(0,1) < \Pi_{j=2}^{M} \frac{F(U^{High}_{i,j}|U^{new}_{i,1},\dots,U^{new}_{i,j-1}) - F(U^{Low}_{i,j}|U^{new}_{i,1},\dots,U^{new}_{i,j-1})}{F(U^{High}_{i,j}|U^{old}_{i,1},\dots,U^{old}_{i,j-1}) - F(U^{Low}_{i,j}|U^{old}_{i,1},\dots,U^{old}_{i,j-1})}$
```

using a proposal that is a close approximation to this desired distribution. Our proposal distribution is $p(U_{i,1}|\Sigma, \Theta, U_{i,1} \in D_{i,1}) \prod_{j=2}^{M} p(U_{i,j}|\Sigma, \Theta, U_{i,1} \dots U_{i,j-1}, U_{i,j} \in D_{i,j})$. To sample the random vector $U_{i,.}$ from this proposal, we first sample $U_{i,1}$ from $p(U_{i,1}|\Sigma, \Theta, U_{i,1} \in D_{i,1})$, then sample from $p(U_{i,2}|\Sigma, \Theta, U_{i,1}, U_{i,2} \in D_{i,2})$ and so on, until we finally sample from conditional $p(U_{i,M}|\Sigma, \Theta, U_{i,1}, \dots, U_{i,M-1}, U_{i,M} \in D_{i,M})$. The cumulative distributions for each conditional in this procedure are the h-functions (Aas et al. 2009) (see equation 2), that are invertible in closed form for most bivariate copula families. Hence we use inverse transform sampling to sample from these h-functions, subject to the rank constraint $D_{i,j}$. Drawing a single sample $U_{i,.}$ from the proposal for a single datapoint involves $O(M^2)$ h-function inversions. We empirically observe a high acceptance ratio with this proposal leading to almost no rejected samples, thereby leading to a complexity of $O(M^2)$ for the Gibbs update of $U_{i,.}$. Details of this MH procedure are described in appendix B. Our algorithm is summarized in algorithm 1.

To draw samples for latent variables Θ and Σ, we use the Metropolis–Hastings algorithm owing to the non-conjugacy of their priors. We also note that it is possible to collapse Θ for faster mixing and work with a mixture of families for each pair copula. However, we did not encounter issues with convergence in our experiments for sampling Θ and proceed as follows. We draw a sample of $\{\sigma_{s,t}, \theta_{s,t}\}$, $\forall 1 \leq s \leq t \leq M$ using random walk metropolis hastings to sample from:

$$(\sigma_{s,t}, \theta_{s,t}|W^{s,t}) \propto p(\theta_{s,t})p(\sigma_{s,t})p(W^{s,t}|\sigma_{s,t}, \theta_{s,t}) \tag{5}$$

The conditioning set $W^{s,t}$ is constructed as follows. When sampling $\{\theta_{s,t}, \sigma_{s,t} : t = s+1\}$, the parameters of the first level bi-variate copulas depend directly on a subset of the sampled marginal variables $\{U_{i,j}\}$ (refer to eqn 1). Hence, the update for $\theta_{s,t}, \sigma_{s,t}$ for the first level bi-variate copulas is conditioned on the set of pairs $W^{s,t} = \{U_{i,s}, U_{i,t}, \forall i\}$. The parameters of higher level bi-variate copulas $(t > s+1)$ depend on pairs of higher order conditionals (again, refer to eqn 1). Hence, for these, the set $W^{s,t}$ is constructed as $W^{s,t} = \{F(U_{i,s}|U_{i,s+1}, \dots, U_{i,s+t-1})$, $F(U_{i,t}|U_{i,s+1}, \dots, U_{i,s+t-1}), \forall i\}$. Note that the conditional distributions $W^{s,t}, \forall t > s+1$ above are evaluated once again using the h-functions recursively.

Computational Complexity. Drawing a single sample from a rank constrained D-vine with uniform marginals using Metropolis Hastings algorithm entails time complexity of $O(M^2)$ with the chosen proposal. Hence, time complexity for a single Gibbs sweep in our algorithm is $O(M^2N)$ due to the quadratic complexity of sampling the $U_{i,.}$ variables for each of the N samples and the sampling for parameters and families of $\binom{M}{2}$ pair copulas.

4 Vines for Multi-View Dependency Seeking Clustering of Mixed data

Consider data $\{X_{i,v,j}\}$, N data points with $i \in [N]$, collected from V views with $v \in [V]$, where $j \in [M_v]$ denotes the dimension in the specific view. Our goal is to cluster the data simultaneously from all the views, while modeling intra–view dependencies in each view.

We model the data in each view v in each cluster k with a D-Vine with the appropriate pair copula families denoted by $\Theta = \{\Theta_{k,v}\}$, and the corresponding parameters $\Sigma = \{\Sigma_{k,v}\}$ by extending the generative definition in equation 3 with a Dirichlet Process (DP) mixture model Teh (2010) in equation 6. We note that each $\Theta_{k,v} = \{\theta_{k,v,s,t} : 1 \leq s < t \leq M_v\}$, representing the families for set of all pair copulas for cluster k, view v. Similarly, we have $\Sigma_{k,v} = \{\sigma_{k,v,s,t} : 1 \leq s < t \leq M_v\}$. To adaptively choose the number of mixture components from the data, we place a DP prior on our mixture distribution. Hence, we draw the mixture weights $\pi \sim GEM(\alpha)$ using the stick breaking process Aldous (1985)Teh (2010) with a concentration parameter α in turn with a gamma prior. The generative process proceeds by selecting cluster indices $\mathbf{Z} = \{Z_i\}$ for each observation i and generating the marginal latent variables $\mathbf{U} = \{U_{i,v,j}\}$ from a D-Vine with uniform marginals followed by the inverse transformation to obtain $\mathbf{X} = \{X_{i,v,j}\}$ similar to equation 3, in a multiview clustering setting. This generative process is shown in eqn 6.

$$\begin{gathered}
\alpha \sim \text{Gamma}(a,b), \quad \pi \sim \text{GEM}(\alpha) \\
\forall k,v,s,t, \quad \theta_{k,v,s,t} \sim \text{Unif}(1:T) \\
\forall k,v,s,t, \quad \sigma_{k,v,s,t}|\theta_{k,v,s,t} \sim \text{Prior}(\sigma_{k,v,s,t}) \\
\forall i \in i,\ldots,N, \quad Z_i|\pi \sim \pi \\
\forall i,v, \quad U_{i,v,.}|Z_i = k, \theta, \mathbf{\Sigma} \sim \text{DVine}(\mathbf{\Sigma_{k,v}}, \mathbf{\Theta_{k,v}}) \\
\forall i,v,j, \quad X_{i,v,j} = F^{-1}_{v,j}(U_{i,v,j}) \qquad (6)
\end{gathered}$$

Inference. We explore approximate inference for our model using Gibbs sampling based on the D-vine inference technique outlined in section 3. We sample random variables $\mathbf{U}$, Σ, θ, $\mathbf{Z}$ and α while π is integrated out due to conjugacy (Aldous 1985).

Notation: A set with a subscript starting with a hyphen($-$) indicates the set of all elements except the index following the hyphen. Let $n_k = |\{\mathbf{X}_i : Z_i = k\}|$.

For sampling α, we follow the standard technique in Escobar and West (1995). Sampling U, Σ, Θ follows from section 3 due to our modeling assumption that data in each view and each cluster is independently generated from a D-vine. Hence, for each cluster k, for each view v, sampling the random variables corresponding to the marginal distributions $\mathbf{U}^{k,v} = \{U_{i,v,.} : i \in [N], Z_i = k\}$, the pair copula parameters $\Sigma_{k,v}$ and the families $\Theta_{k,v}$ independently follow the same steps as outlined in the Gibbs sampling iteration in algorithm 1.

Sampling the cluster assignment, Z, is based on CRP (Aldous 1985), the predictive distribution arising from a DP. However, it differs from the standard approach due to the rank constraint in the algorithm. The probability of Z_i taking a particular value k can be expressed as a product of two terms, $p(Z_i = k|Z_{-i})$ arising from the CRP and $P(U_{i,.,.}|Z_i = k, \Sigma, \Theta)$, the likelihood term. However the support for Z_i is constrained to the set C_i (defined below) for selecting an existing cluster to satisfy the rank constraints within the cluster. Hence, for any $k \in [K]$, Z_i being set to k is permissible if $\mathbf{U}^k \cup U_{i,.,.}$ meets the rank constraints. We define the set of permissible clusters as

$$C_i = \{k : (U_{i,.,.}^{k,Low}) < U_{i,.,.} < (U_{i,.,.}^{k,High})\} \tag{7}$$

The update for Z_i is given as follows.

$$p(Z_i = k|Z_{-i}, U, \Sigma, \Theta, D_{i,.,.}) \propto \frac{n_k}{N+\alpha} p(U_{i,.,.}|Z_i = k, \Sigma, \Theta)\delta(k \in C_i) \tag{8}$$

Computing the probability of $Z_i = k_{new}$, for a new component requires integrating over the prior distributions of the set of parameters $\Sigma_{k_{new},v}$ of the new component and the corresponding D-Vine families $\Theta_{k_{new},v}$. We follow the technique proposed by Neal (2000), by finding a Monte Carlo estimate of the probability of selecting a new cluster.

Algorithm 2: Gibbs Sampling: Multiview Dependency Seeking clustering with D-vines for mixed data

```
for each $i = 1, \ldots, N$ do
    $p(Z_i = k|U, \Sigma, \Theta) \propto \frac{n_k}{N+\alpha} p(U_{i,.,.}|Z_i = k, \Sigma, \Theta)\delta(k \in C_i)$
    Where $C_i$ is defined in equation 7
for each $i = 1, \ldots, N$ do
    for each $v = 1, \ldots, V$ do
        for each $j = 1, \ldots, M_v$ do
            $U_{i,v,.} \sim DVine(\Theta_{k,v}, \Sigma_{k,v})|U_{i,v,j} \in D_{i,v,j}, \forall j$ , //Refer to our algorithm 1 for details of rank based D-vine sampler
for k=1 to K do
    for each $v = 1, \ldots, V$ do
        for t=1 to $M_v$-1 do
            for s=1 to $M_v$-t do
                //Sample D-Vine Parameters with MH as in equation 5
                $\sigma_{k,v,s,t}, \theta_{k,v,s,t} \sim p(\sigma_{k,v,s,t}, \theta_{k,v,s,t}|W_{k,v,s,t})$
```

5 Experiments

5.1 Parameter Estimation

To evaluate how well our inference algorithm estimates parameters of a D-Vine, we simulate 500 samples from a 6–dimensional D-Vine with continuous marginals (Gaussian, exponential and gamma) with known parameters and estimate the parameters using our algorithm **Ext-DVine** and the Maximum Likelihood method of Aas et al. (2009), *MLE*, as well as the method of Panagiotelis et al. (2012).

Table 1 shows the average RMSE of the original parameters with respect to the estimated parameters obtained by Ext-DVine and MLE. Our estimates are closer

Algorithm	RMSE
Cont: Ext-DVine	0.0389
MLE (Aas et al)	0.0395
Discrete: Ext-DVine	0.06
Panagiotelis et al	0.106
Mixed: Ext-Dvine	0.0429

Table 1: RMSE from original parameters; top: Ext-Dvine vs MLE estimate (Aas et al. 2009) for continuous data, center: Ext-Dvine vs (Panagiotelis et al. 2012) for discrete data, bottom: Ext-Dvine with mixed data - mean over 25 runs

GOF Method	Kendall's Tau	Pearson	Spearman Rho
MLE	0.031	0.071	0.046
Ext-DVine	0.016	0.066	0.023

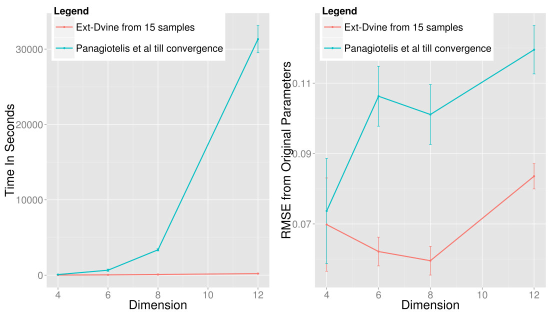
Table 2: Goodness of Fit: RMSE between correlation values on original data & data simulated using parameter estimates from MLE and our method.

to the true parameters than those obtained by MLE. We repeat this experiment with mixed marginals (Gaussian, gamma, negative binomial, Poisson) and obtain a low RMSE of the estimated parameters from the original parameters (there are no available baselines for mixed data).

For continuous data, we also perform a *goodness of fit* test comparing Ext-DVine inference with the popular MLE technique of Aas et al. (2009), from the R package CDVine of Brechmann and Schepsmeier (2013), by estimating the parameters of the D-Vine and simulating new data with these parameters and comparing difference in correlations (measured by Kendall's Tau, Spearman's Rho and Pearson's correlation coefficients) between the original dataset and the re–simulated dataset. Table 2 shows that the differences in correlations are lesser when parameters are estimated using our method thus implying a better fit with our Bayesian inference algorithm for Ext-DVine, as compared to the differences in correlations when simulation parameters are ML estimates.

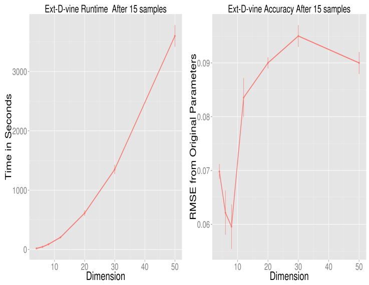
5.2 Time Complexity

We empirically evaluate the time complexity and accuracy for discrete marginals by plotting time taken for inference varying the dimension (M) for a fixed datasize of N=500 points, with parameters generated from priors. Since there is no baseline for mixed data, we restrict this evaluation to discrete data and use the baseline of Panagiotelis et al. (2012), the most efficient method known for discrete vines. We use 15 sampling sweeps while the method of Panagiotelis et al. (2012) takes significantly more time to run till convergence (with between 10-20 iterations) and obtains less accurate parameter estimates. (Results shown over 25 runs with error bars in figure 3). While our inference method analytically leads to complexity quadratic in M (and linear in the number of pair-copulas), in figure 3, it almost looks linear in M, in comparison with Panagiotelis et al. (2012) due to significantly higher runtime of the baseline. In fact, the baseline did not complete its run to convergence after running for a day even for 20 dimensional data. In figure 4, we show a standalone plot of the runtime and accuracy of our technique (without the discrete baseline) for upto M=50 dimensions. We observe quadratic complexity of O(M^2N), linear in the number of pair copulas, for a fixed datasize N as discussed.



(a) Runtime (b) Accuracy (RMSE)

Fig. 3: Discrete Data: Comparison with Panagiotelis et al. (2012), Ext-Dvine much faster with higher accuracy: Bars indicate 5 times sd over 25 runs.



(a) Runtime (b) Accuracy

Fig. 4: Performance on Discrete Data, Bars indicate 5 times SD over 25 runs.

5.3 Dependency Seeking Clustering

Multi-view Setting: We evaluate our model for Multi-view dependency seeking clustering on synthetic datasets containing asymmetric and tail dependencies.

Baselines: For continuous features, we compare with Rey and Roth's model Rey and Roth (2012), the state of the art for dependency seeking multi-view clustering that uses Gaussian copulas (GC–MVC) and can only be used with continuous data. For mixed data, since there are no existing baselines, we implement an extended rank likelihood based inference on Rey and Roth's model (ext–GC–MVC). This method does not exist in previous literature, but inference follows the straightforward sampling scheme of Hoff (2007) and does not face the difficulties that we address, for inference with vines. Note that while this can fit mixed data, it can only model meta–Gaussian dependencies. Our vine–based algorithm to handle mixed data is denoted by **DP–Vine–MVC**.

Evaluation Metrics. We evaluate the ability of GC–MVC and our method DP–Vine–MVC to identify the correct number of clusters. We also evaluate the clustering performance of DP–Vine–MVC and Ext-GC–MVC when the number of clusters is given as input. Clustering performance is measured by Adjusted Rand Index (ARI) (Hubert and Arabie 1985), Variation of Information (VI) (Meilă 2007), Normalized Mutual Information (NMI) (Vinh et al. 2010) and the classification accuracy obtained by fixing the labels of the inferred clusters. Note that lower VI is better while higher values in other metrics indicate better performance. All results shown are averages over 25 simulations.

Simulations. We generate data with two views, with three dimensions in each view. The pairwise dependencies for each view, with a different dependency structure, is shown in figure 5. For mixed datasets we generate two datasets, one with continuous marginals (gamma, normal and exponential) in each view and one with mixed marginals (gamma, negative binomial and Poisson) in each view. Exact parameters and settings of the simulations are detailed in appendix A.

Results. Figure 6 shows proportion of times, out of 25 runs, when algorithms DP–Vine–MVC and DP-GC–MVC obtain a specific number of clusters. We observe

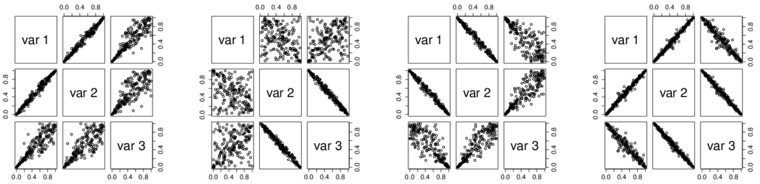
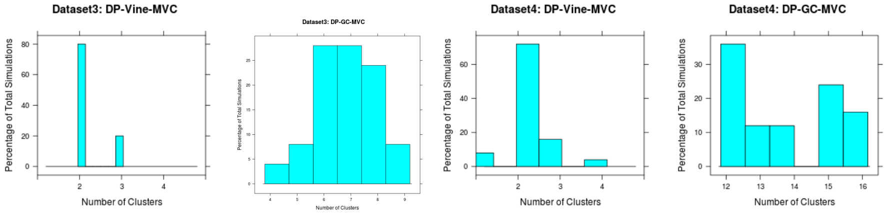


Fig. 5: Pairwise dependencies for each view in simulations. Above: Cluster1-View1 (L), Cluster2-View2 (R); Below: Cluster1-View2 (L), Cluster2-View2 (R).

that DP-GC–MVC does not infer the right number of clusters (fig 6b, 6d). In the continuous case, our method infers the right number 80% of the times and in the remaining cases, the deviation is not large – it infers 3 instead of 2 (fig 6a). In comparison, DP–GC–MVC has to compensate for the model mismatch by increasing the number of clusters. In most cases, the number of clusters inferred is more than 6 (fig 6b). In the case of mixed data, the results of ext–DP–GC–MVC are worse. The inferred number of clusters range from 12 to 16 (fig 6d). DP–Vine–MVC does much better, inferring the right number of clusters in 65% of cases and the deviation ≤ 2 (fig 6c). Table 3 shows the clustering performance of DP–Vine–MVC and GC–MVC for continuous data. DP–Vine–MVC obtains better clustering performance than the other two methods.



(a) MultiviewCont (b) MultiviewCont (c) MultiviewMixed (d) MultiviewMixed

Fig. 6: Histograms of number of clusters found by DP–Vine–MVC (left) and GC–MVC (right). Above: continuous marginals, Below: mixed marginals.

Table 3: Multi-View Clustering on synthetic datasets with Continuous marginals

Algorithm	DP–Vine	GC
Measure	MVC	MVC
ARI	**0.346**	0.110
NMI	**0.308**	0.117
VI	**0.936**	1.128
Accuracy	**0.795**	0.661

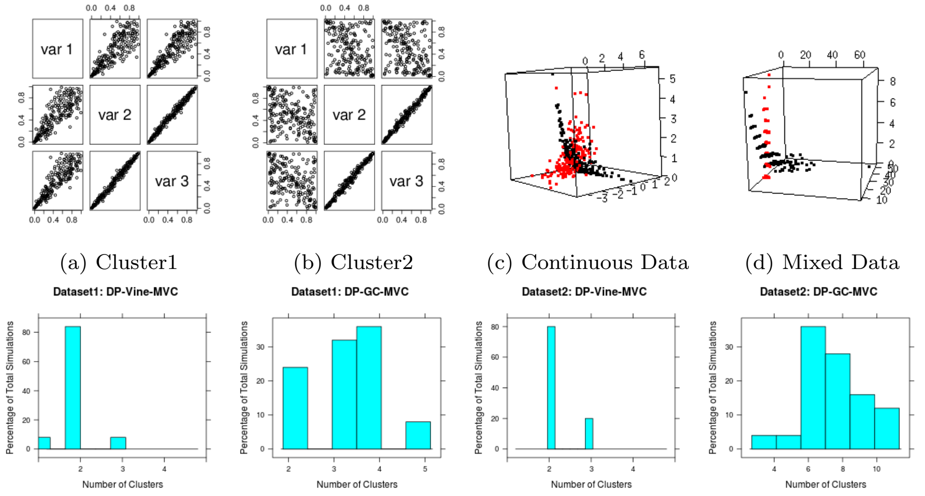
Table 4: Multi-View Clustering on synthetic datasets with Mixed marginals

Algorithm	DP–Vine	ext–GC
Measure	MVC	MVC
ARI	**0.252**	0.167
NMI	**0.207**	0.138
VI	**1.095**	1.185
Accuracy	**0.729**	0.692

Table 4 shows the clustering performance of DP–Vine–MVC and ext–GC–MVC for mixed data. Note that ext–GC–MVC is not able to discriminate between clusters with non-metaGaussian dependencies and hence has worse performance. Best results in both tables are in bold.

Single–View Setting While our focus application is multiview dependency seeking clustering, we also run our algorithm in the special case of single-view setting to demonstrate our algorithm for datasets with more complex dependencies like combination of asymetric and tail dependencies. We generate data with pairwise tail dependencies and asymmetric dependencies as shown in figure 7 (a-d). In dataset 1 we use gamma, normal and exponential marginals and in dataset 2 we use gamma, negative binomial and poisson marginals. Note that cluster 1 has asymmetric dependencies and cluster 2 has tail dependencies.

Figure 7 (e-h) shows proportion of times, out of 25 runs, when algorithms DP–Vine–MVC and GC–MVC obtain a specific number of clusters in the single view setting showing how our model fits the data compared to baseline for data generated from a known number of clusters. In the continuous case, our method infers the right number 80% of the times and in the remaining cases, the deviation is not large (fig 7e). GC–MVC has to compensate for the model mismatch by increasing the number of clusters (fig 7f). In the case of mixed data, ext–GC–MVC infers the number of clusters to be more than 5 in 90% of the cases, erroneously (fig 7h). DP–Vine–MVC does much better, inferring the right number of clusters in 80% of the cases and the deviation is ≤ 1 (fig 7g). Table 5 shows the performance of DP–Vine–MVC in comparison with GC–MVC and GMM for dependency seeking clustering on continuous data. Table 6 compares DP–Vine–MVC vs baselines Ext-GC–MVC, SCENIC (Plant 2012) and ClustMD (McParland and Gormley 2015), the state of the art for clustering mixed data in a single-view setting.



(a) Cluster1 (b) Cluster2 (c) Continuous Data (d) Mixed Data

(e) Cont:DP-Vine (f) Cont:DP-GC (g) Mixed:DP-Vine (h) Mixed:DP-GC

Fig. 7: a-b: Pairwise scatter plots for each cluster of generated synthetic data. 3D scatterplot of data generated from dataset1 (c) and dataset2 (d). e-h: Single View Setting - Histograms of number of clusters found - continuous (e-f), mixed (g-h) marginals.

Table 5: Results: single view, continuous

Algorithm Measure	DP–Vine MVC	GC MVC	GMM
ARI	**0.220**	0.065	0.017
NMI	**0.195**	0.056	0.021
VI	**1.084**	1.295	1.354
Accuracy	**0.738**	0.634	0.572

Table 6: Results: single view, mixed marginals

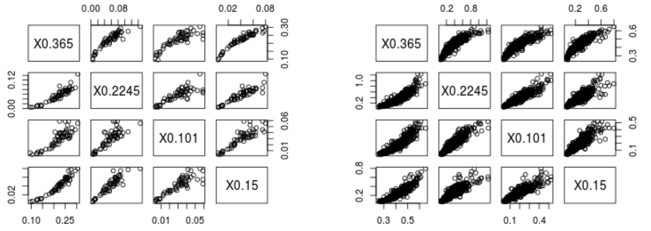
Algorithm Measure	DP–Vine MVC	Ext-GC MVC	SCENIC	ClustMD
ARI	**0.124**	0.075	0.006	0.058
NMI	**0.101**	0.074	0.014	0.083
VI	1.237	1.215	1.366	**1.153**
Accuracy	**0.664**	0.635	0.508	0.602

Age:3–20	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Cluster1(DP-Vine)	1	10	8	16	11	3	2	5	0	0	0	0	0	0	0	0	0	0
Cluster2(DP-Vine)	0	0	2	17	34	59	78	52	59	45	19	25	16	6	10	6	7	5

Age:3–20	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Cluster1(DP-GC)	0	1	1	11	14	26	35	18	20	21	9	11	3	3	2	2	3	1
Cluster2(DP-GC)	1	5	5	8	9	13	9	12	16	10	2	5	10	0	6	2	3	1
Cluster3(DP-GC)	0	3	2	10	15	9	18	13	19	8	1	6	2	2	0	0	1	2
Cluster4(DP-GC)	0	1	2	4	7	14	18	14	4	6	7	4	1	1	2	2	0	1

Table 7: Abalone dataset: DP-Vine-MVC produces two overlapping clusters: younger abalones of age <= 7 and older abalones of age >= 7. Four clusters are produced by DP-GC-MVC which do not show discernible relation to age.

Real Datasets. *Abalone dataset:* This dataset, from the UCI repository (Bache and Lichman 2013), contains six continuous attributes of abalones with different ages. Our algorithm finds two clusters also shown in table 7, that meaningfully represents younger and older abalones. The dependency structures in the two clusters are different, younger abalones have asymmetric dependencies that our model captures through a Clayton pair copula. Figure 8 shows pairwise correlations in older and younger abalones. Younger abalones have asymmetric correlations where there is high correlation for smaller values and low correlation for larger values: accurately captured by our model. We also show clustering results with our baseline DP-GC-MVC that cannot model meta–Gaussian dependencies present.



(a) Cluster1: Young Abalones (b) Cluster2: Older Abalones

Fig. 8: Pairwise correlations in older and younger abalones.

Mortality Dataset: This dataset from physionet (MIMIC II database) Goldberger et al. (2000 (June 13)), comprises of 800 ICU patient patient records where each record contains the last collected readings for 8 features, from 2 views. (1) View 1 features: BUN, Creatinine, HCO3, PaO2 (from blood tests) (2) View 2:

Algorithm: Measure	DP–Vine MVC	ext–GC MVC
ARI	**0.20**	0.02
NMI	**0.20**	0.009
VI	**0.90**	1.27
Accuracy	**0.734**	0.58

Table 8: Results: Mortality dataset

GCS, HR, Weight, Age (external measurements). The data also contains target binary label (mortality status) indicating whether or not the patient survived. We cluster this data with DP-Vine-MVC and obtain two clusters that are indicative of patient mortality status with accuracy shown in table 8. We outperform the baseline ext–GC–MVC in both the measures: clustering metrics and mortality prediction accuracy.

5.4 Summary of Results:

- Ext-Dvine obtains more accurate parameter estimates than the MLE method of Aas et al. (2009) for continuous margins as well as the method of Panagiotelis et al. (2012) for discrete margins. In runtime it is faster than the latter and is the first method to fit vines on mixed margins.

- DP-Vine-MVC, our DP mixture model for dependency seeking clustering in multi-view and single view settings, is evaluated on simulated continuous and mixed data containing asymmetric and tail dependencies. We show superior performance over baselines in
  1. clustering accuracy in a finite mixture setting,
  2. detecting the correct number of clusters in a non-parametric setting.

- DP-Vine-MVC significantly outperforms Ext-GC-MVC (that is limited to modeling meta-Gaussian dependencies) on clustering real world datasets.

6 Conclusion

We design a new MCMC inference algorithm to fit vines on mixed data that runs in $O(M^2N)$ time per sampling step (M dimensions, N observations). Our model, a DP mixture of vines, can fit mixed margin distributions and arbitrary dependencies. Empirically we demonstrate the benefits of our model in dependency seeking clustering, extending state–of–the–art multi– and single– view models by modeling asymmetric and tail dependencies and fitting mixed data.

References

Aas, Kjersti, Claudia Czado, Arnoldo Frigessi, and Henrik Bakken. 2009. Pair-copula constructions of multiple dependence. *Insurance: Mathematics and Economics* 44 (2): 182–198.

Aldous, David J. 1985. In *École d'été de probabilités de Saint-Flour, XIII—1983. Lecture Notes in Mathematics*, 1–198. Springer.

Bache, K., and M. Lichman. 2013. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml.

Bayestehtashk, Alireza, and Izhak Shafran. 2013. Parsimonious multivariate copula model for density estimation. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5750–5754.

Brechmann, Eike Christian, and Ulf Schepsmeier. 2013. Modeling dependence with C- and D-vine copulas: The R package CDVine, Technical report.

Browne, Ryan P, and Paul D McNicholas. 2012. Model-based clustering, classification, and discriminant analysis of data with mixed type. *Journal of Statistical Planning and Inference* 142 (11): 2976–2984.

Eban, Elad, Gideon Rothschild, Adi Mizrahi, Israel Nelken, and Gal Elidan. 2013. Dynamic copula networks for modeling real-valued time series. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, 247–255.

Eickhoff, Carsten, Arjen P de Vries, and Thomas Hofmann. 2015. Modelling term dependence with copulas. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 783–786.

Elidan, Gal. 2013. Copulas in machine learning. In *Copulae in Mathematical and Quantitative Finance. Lecture Notes in Statistics*, 39–60.

Escobar, Michael, and Mike West. 1995. Bayesian density estimation and inference using mixtures. In *Journal of American Statistical Association.*

Genest, Christian, and Johanna Neslehova. 2007. A primer on copulas for count data. *Astin Bulletin* 37 (2): 475.

Goldberger, A. L., L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C. K. Peng, and H. E. Stanley. 2000 (June 13). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation* 101 (23): 215–220.

Hoff, Peter D. 2007. Extending the rank likelihood for semiparametric copula estimation. *The Annals of Applied Statistics* 1 (1): 265–283.

Hubert, Lawrence, and Phipps Arabie. 1985. Comparing partitions. *Journal of classification* 2 (1): 193–218.

Joe, Harry. 2014. *Dependence Modeling with Copulas.* CRC Press.

Kim, Daeyoung, Jong-Min Kim, Shu-Min Liao, and Yoon-Sung Jung. 2013. Mixture of D-vine copulas for modeling dependence. *Computational Statistics & Data Analysis* 64: 1–19.

Klami, Arto, and Samuel Kaski. 2008. Probabilistic approach to detecting dependencies between data sets. *Neurocomputing* 72 (1): 39–46.

Klami, Arto, Seppo Virtanen, and Samuel Kaski. 2010. Bayesian exponential family projections for coupled data sources. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence (UAI)*, 286–293.

Kosmidis, Ioannis, and Dimitris Karlis. 2015. Model-based clustering using copulas with applications. In *Statistics and Computing.* Springer.

Letham, Benjamin, Wei Sun, and Anshul Sheopuri. 2014. Latent variable copula inference for bundle pricing from retail transaction data. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, 217–225.

McParland, Damien, and Isobel Claire Gormley. 2015. Model based clustering for mixed data: clustMD. *arXiv preprint arXiv:1511.01720.*

McParland, Damien, Isobel Claire Gormley, Tyler H McCormick, Samuel J Clark, Chodziwadziwa Whiteson Kabudula, and Mark A Collinson. 2014. Clustering South African households based on their asset status using latent variable models. *The Annals of Applied Statistics* 8 (2): 747.

Meeds, Edward, Zoubin Ghahramani, Radford Neal, and Sam Roweis. 2007. Modeling dyadic data with binary latent factors. *Advances in Neural Information Processing Systems (NIPS)* 19.

Meilă, Marina. 2007. Comparing clusterings: an information based distance. *Journal of Multivariate Analysis* 98 (5): 873–895.

Neal, Radford M. 2000. Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics* 9 (2): 249–265.

Panagiotelis, Anastasios, Claudia Czado, and Harry Joe. 2012. Pair copula constructions for multivariate discrete data. *Journal of the American Statistical Association* 107 (499): 1063–1072.

Plant, Claudia. 2012. Dependency clustering across measurement scales. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 361–369.

Plant, Claudia, and Christian Böhm. 2011. INCONCO: interpretable clustering of numerical and categorical objects. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1127–1135.

Rey, M., and V. Roth. 2012. Copula mixture model for dependency-seeking clustering. In *International Conference on Machine Learning (ICML)*.

Shawe-Taylor, John, and Nello Cristianini. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press.

Sklar, A. 1959. Fonctions de rpartition n dimensions et leurs marges. *Publ. Inst. Statist. Univ. Paris* 8: 229–231.

Smith, Michael S, and Mohamad A Khaled. 2012. Estimation of copula models with discrete margins via Bayesian data augmentation. *Journal of the American Statistical Association* 107 (497): 290–303.

Teh, Y. W. 2010. Dirichlet processes. In *Encyclopedia of Machine Learning*. Springer.

Vinh, Nguyen Xuan, Julien Epps, and James Bailey. 2010. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *The Journal of Machine Learning Research* 11: 2837–2854.

Yerebakan, Halid Z., Bartek Rajwa, and Murat Dundar. 2014. The infinite mixture of infinite Gaussian mixtures. In *Advances in Neural Information Processing Systems (NIPS)*.

APPENDIX A

7 Experiments: Generation of Simulated Data

We generate data demonstrating pairwise tail dependencies. We do this by simulating data from a D-Vine with suitable dependency structure and performing the inverse transform with the appropriate marginals. The data for each cluster is generated independently through this process.

Multiview Setting: We generate two views for each cluster with three dimensions in each view. We use copula famalies Clayton, Gaussian and Student-T bivariate copulas with parameters shown in table 9. For the discrete multiview dataset, we then generate the marginals with distributions and their parameters shown in table 10. Similarly, for the continuous multiview dataset, we generate the parameters shown in table 11.

Cluster1, view1	Gaussian (0.99)	Clayton (4)	T-Copula (0.99,3)
Cluster1, view2	Gaussian (-0.99)	Clayton (4)	Gaussian (0.99)
Cluster2, view1	T-Copula (0.45,3)	Gaussian (-0.99)	T-Copula (0.5,3)
Cluster2, View2	T-Copula (0.99,3)	Gaussian (-0.99)	T-Copula (0.99,3)

Table 9: Copula parameters for generating synthetic MultiView data for datasets3 and dataset4

Singleview Setting: The data generation process for the single view case is similar to the multi view case with only 1 view. We use copula families Clayton, Gaussian and Student-T bivariate copulas with parameters shown in table 12. For the discrete singleview dataset, we then generate the marginals with distributions and their parameters shown in table 13. Similarly, for the continuous multiview dataset, we generate the parameters shown in table 14.

Cluster1, view1	Gamma (2,2)	Normal (3,2)	Poisson (2)
Cluster1, view2	Gamma (4,1)	Normal (0,3)	Poisson (3)
Cluster2, view1	Gamma (4,1)	Normal (4,1)	Poisson (3)
Cluster2, View2	Gamma (4,1)	Normal (0,3)	Poisson (3)

Table 10: Marginals for multiview mixed dataset

Cluster1, view1	Gamma (2,4)	Normal (2,2)	Gamma (2,2)
Cluster1, view2	Gamma (4,3)	Normal (4,3)	Gamma (2,4)
Cluster2, view1	Gamma (3,2)	Normal (2,1)	Gamma (2,1)
Cluster2, View2	Gamma (4,2)	Normal (4,1)	Gamma (1,1)

Table 11: Marginals for Multiview Continuous dataset

Cluster1	Clayton (5)	Gaussian (-0.99)	Clayton (4)
Cluster2	T-Copula (-0.55,3)	Clayton (5)	T-Copula (.25,3)

Table 12: Copula Parameters for Single View data

Cluster1	Gamma (.5,1)	NegBinomial (25,.5)	Poisson (2)
Cluster2	Gamma (1,1)	Normal (25,1)	Poisson (2)

Table 13: Marginals for Singleview mixed dataset

Cluster1	Gamma (2,1)	Normal (0,.1)	Exponential (1)
Cluster2	Gamma (4,3)	Normal (0,1)	Exponential (1)

Table 14: Marginals for Singleview Continuous dataset

APPENDIX B

8 Metropolis Hastings for sampling from a Rank constrained D-Vine

An important step of our Gibbs sampling inference procedure in section 3.1 comprises of sampling $\{U_{i,.}\}$ from a D-vine with uniform marginals subject to rank based constraints that follow from the extended rank likelihood methodology. Let $D_{i,j} = \{u \in [0,1] : max\{U_{rj} : X_{rj} < X_{ij}, r \in [N]\} < u < min\{U_{rj} : X_{ij} < X_{rj}, r \in [N]\}\}$. and $D_{i,.}$ denote the set $D_{i,1} \times D_{i,2} \times \ldots \times D_{i,M}$. Our target is to block sample the random variables $U_{i,.}$ from a target distribution $t(U_{i,.})$ that is a truncated D-vine with rank constraints $p(U_{i,.}|\Sigma, \Theta, U_{-i,.}, U_{i,.} \in D_{i,.})$. One way to sample from this distribution is to sample values from the unconstrained D-vine $p(U_{i,.}|\Sigma, \Theta)$ and reject samples that do not satisfy rank constraints. However, this could lead to excessive rejections -instead we use Metropolis Hastings with a proposal that is an approximation of our target distribution, and satisfies the rank constraints.

Consider the following proposal distribution:

$$r(U) = p(U_{i,1}|\Sigma, \Theta, U_{i,1} \in D_{i,1}) \prod_{j=2}^{M} p(U_{i,j}|\Sigma, \Theta, U_{i,1} \ldots U_{i,j-1}, U_{i,j} \in D_{i,j}) \qquad (9)$$

To sample the random vector $U_{i,.}$ from this proposal, as mentioned in section 3.1, we first sample $U_{i,1}$ from $p(U_{i,1}|\Sigma, \Theta, U_{i,1} \in D_{i,1})$, then sample from $p(U_{i,2}|\Sigma, \Theta, U_{i,1}, U_{i,2} \in D_{i,2})$ and so on, until we finally sample from conditional $p(U_{i,M}|\Sigma, \Theta, U_{i,1}, \ldots, U_{i,M-1}, U_{i,M} \in D_{i,M})$. The cumulative distributions for each step in this procedure are the h-functions (Aas et al. 2009) (see equation 2), that are invertible in closed form for most bivariate copula families. Hence we use inverse transform sampling to sample from these h-functions, subject to the rank

Table 15: Acceptance ratio of Metropolis Hastings: Mean over 25 Ext-Dvine inference runs with 30 samples in each run (after discarding burn-in of 20 samples) with 500 datapoints of 6 dimensions with various marginals. The continuous dataset was generated from Gamma, normal, exponential maginals, discrete dataset from a combination of poisson and negative binomial marginals and mixed dataset from Gamma, negative binomial and Poisson marginals

DataSet	Acceptance Ratio
Continuous	**0.9983**(sd=0.0003)
Discrete	**0.9973**(sd=0.0003)
Mixed	**0.9737**(sd=0.0026)

constraint $D_{i,j}$. Drawing a single sample $U_{i,.}$ from the proposal for a single datapoint involves $O(M^2)$ h-function inversions.

We now compute the acceptance ratio for this sampling scheme. Let $t(U_{i,.})$ be the target distribution described above.

$$\begin{aligned} t(U_{i,.}) &= p(U_{i,.}|\Sigma, \Theta, U_{-i,.}, U_{i,.} \in D_{i,.}) = p(U_{i,.}|\Sigma, \Theta, U_{i,.} \in D_{i,.}) \\ &= \frac{p(U_{i,1}, U_{i,2}, \ldots, U_{i,M}|\Sigma, \Theta) \prod_{j=1}^{M} \delta(U_{i,j} \in D_{i,j})}{p(U_{i,1} \in D_{i,1}, \ldots, U_{i,M} \in D_{i,M}|\Sigma, \Theta)} \end{aligned} \tag{10}$$

Now, consider a single term $p(U_{i,j}|\Sigma, \Theta, U_{i,1} \ldots U_{i,j-1}, U_{i,j} \in D_{i,j})$ in the proposal distribution from equation 9.

$$\begin{aligned} &p(U_{i,j}|\Sigma, \Theta, U_{i,1} \ldots U_{i,j-1}, U_{i,j} \in D_{i,.}) \\ &= \frac{p(U_{i,1}, \ldots, U_{i,j}, U_{i,j} \in D_{i,.}|\Sigma, \Theta)}{p(U_{i,1}, \ldots, U_{i,j}, U_{i,j-1} \in D_{i,.}|\Sigma, \Theta)} = \frac{p(U_{i,1}, \ldots, U_{i,j}|\Sigma, \Theta)\delta(U_{i,j} \in D_{i,j})}{\int_{U_{i,j} \in D_{i,j}} p(U_{i,1}, \ldots, U_{i,j}|\Sigma, \Theta)} \\ &= \frac{p(U_{i,1}, \ldots, U_{i,j}|\Sigma, \Theta)\delta(U_{i,j} \in D_{i,j})}{p(U_{i,1}, \ldots, U_{i,j-1}|\Sigma, \Theta) \int_{U_{i,j} \in D_{i,j}} p(U_{i,j}|U_{i,1}, \ldots, U_{i,j-1}|\Sigma, \Theta)} \\ &= \frac{p(U_{i,j}|\Sigma, \Theta, U_{i,1}, \ldots, U_{i,j-1})\delta(U_{i,j} \in D_{i,j})}{F(U_{i,j}^{High}|\Sigma, \Theta, U_{i,1}, \ldots, U_{i,j-1}) - F(U_{i,j}^{Low}|\Sigma, \Theta, U_{i,1}, \ldots, U_{i,j-1})} \end{aligned} \tag{11}$$

$$\text{Hence, } r(U_{i,.}) = \frac{p(U_{i,1}, \ldots, U_{i,M}|\Sigma, \Theta) \prod_{j=1}^{M} \delta(U_{i,j} \in D_{i,j})}{\prod_{j=1}^{M} F(U_{i,j}^{High}|\Sigma, \Theta, U_{i,1}, \ldots, U_{i,j-1}) - F(U_{i,j}^{Low}|\Sigma, \Theta, U_{i,1}, \ldots, U_{i,j-1})} \tag{12}$$

The acceptance ratio can be computed from equation 12 and 10 as

$$\begin{aligned} &acceptance(U_{i,.}^{new}, U_{i,.}^{old}) = \frac{t(U_{i,.}^{new})r(U_{i,.}^{old})}{t(U_{i,.}^{old})r(U_{i,.}^{new})} \\ &= \prod_{j=2}^{M} \frac{F(U_{i,j}^{High}|\Sigma, \Theta, U_{i,1}^{new}, \ldots, U_{i,j-1}^{new}) - F(U_{i,j}^{Low}|\Sigma, \Theta, U_{i,1}^{new}, \ldots, U_{i,j-1}^{new})}{F(U_{i,j}^{High}|\Sigma, \Theta, U_{i,1}^{old}, \ldots, U_{i,j-1}^{old}) - F(U_{i,j}^{Low}|\Sigma, \Theta, U_{i,1}^{old}, \ldots, U_{i,j-1}^{old})} \end{aligned} \tag{13}$$

We empirically observe a high acceptance ratio with this proposal leading to almost no rejected samples with our proposal. Table 15 shows the acceptance ratio averaged over 25 runs with datasets generated with 500 points with 6 dimensions. The complete inference algorithm is summarized in algorithm 1 of the main paper.