HELM: Highly Efficient Learning of Mixed copula networks

Yaniv Tenzer
Department of Statistics
The Hebrew University
yaniv.tenzer@gmail.com

Gal Elidan
Department of Statistics
The Hebrew University
galel@huji.ac.il

Abstract

Learning the structure of probabilistic graphical models for complex real-valued domains is a formidable computational challenge. This inevitably leads to significant modelling compromises such as discretization or the use of a simplistic Gaussian representation. In this work we address the challenge of *efficiently* learning truly expressive copula-based networks that facilitate a mix of varied copula families within the *same* model. Our approach is based on a simple but powerful bivariate building block that is used to highly efficiently perform local model selection, thus bypassing much of computational burden involved in structure learning. We show how this building block can be used to learn general networks and demonstrate its effectiveness on varied and sizeable real-life domains. Importantly, favorable identification and generalization performance come with dramatic runtime improvements. Indeed, the benefits are such that they allow us to tackle domains that are prohibitive when using a standard learning approaches.

1 INTRODUCTION

Probabilistic graphical models [Pearl, 1988] in general and Bayesian networks (BNs) in particular, have become popular as a flexible and intuitive framework for modeling multivariate densities, a central goal of the data sciences. At the heart of the formalism is a combination of a qualitative graph structure that encodes the regularities (independencies) of the domain and quantitative local conditional densities of a variable given its parents in the graph. The result is a decomposable model that facilitates relatively efficient inference and estimation. Unfortunately, learning the structure of such models remains a formidable challenge, particularly when dealing with real-valued domains that are non-Gaussian. The computational bottleneck lies in the need to assess the merit of many candidate structures, each requiring potentially costly maximum likelihood evaluation.

The situation is further compounded in realistic domains where we also want to allow for the combination of different local representations within the same model. Specifically, such a scenario requires that we perform non-trivial local model selection *within* an already challenging structure learning procedure. In practice, with as few as tens of variables, learning any real-valued graphical model beyond the simple linear Gaussian BN can be computationally impractical. At the same time, it is clear that, for many domains, the Gaussian representation is too restrictive. Our goal in this work is to overcome this barrier and to *efficiently* learn the structure of expressive networks that do not only go beyond the Gaussian, but that also allow for a mix of varied local representations.

In the search for expressive representations, several recent works use copulas as a building block within the framework of graphical models [Kirshner, 2007, Elidan, 2010, Wilson and Ghahramani, 2010]. Briefly, copulas [Joe, 1997, Nelsen, 2007] flexibly capture distributions of few dimensions: easy to estimate univariate marginals are joined together using a copula function that focuses solely on the dependence pattern of the joint distribution. Appealingly, regardless of the dependency pattern, any univariate representation can be combined with any copula. In all of the above works, the resulting copula graphical model proved quite effective at capturing complex high-dimensional domains, far surpassing the Gaussian representation.

Recently, Elidan [2012] proposed a structure learning method that is tailored to the so called copula network representation, and that is essentially as efficient as learning a simple linear Gaussian BN. However, an important drawback of the approach is that it constrains all local copulas in the model to be of the same type. Tenzer and Elidan [2013] offer a slight improvement but their method is inherently limited to few (2-3) of *specific* local representations and to tree-structured networks. Clearly, to take advantage of the plethora of dependency patterns captured by different copula families, we would like to have greater flexibility.

Unfortunately, selection of the right copula family, or dependence pattern, can be hard even for just two random variables. Typical approaches (e.g., [Huard et al., 2006, Fermanian, 2005, Hering and Hofert, 2010, Justel et al., 1997, Genest and Rivest, 1993]) require costly computations such as maximum likelihood estimation, Bayesian integration, simulation, etc. (see Section 3 for details). While such methods can be used to perform model selection for a distribution with few variables, they are impractical when faced with a large number of local model selection tasks that underlie global structure learning. In this work we introduce HELM: a method for **H**ighly **E**fficient **L**earning of **M**ixed copula networks.

Intuitively, for the task of model selection, the maximum likelihood density defined by a particular copula family is in fact a nuisance parameter, and we are only interested in detecting the dependency pattern of the copula. Further, since most copulas have a functional form with few parameters (or even just one), identifying between different copulas only requires a crude view of the distribution. Building on this intuition, we build a copula-to-multinomial mapping that is independent of a particular domain. Then, when faced with the model selection task given training samples, we use a comparison of the empirical multinomial signature to the precomputed mapping in order to choose the most promising copula family. Appealingly, for the building block task of choosing a copula family for two variables, our approach is effective, highly efficient, and comes with finite sample guarantees.

With this model selection building block in hand, we are still faced with the task of learning the *global* structure of the model, which in turn requires costly maximum likelihood computations. Fortunately, the same mechanism we use for selection suggests a highly efficient and effective proxy to the exact computation, when learning tree networks. Further, the method also gives rise to a natural heuristic generalization that allows us to highly efficiently learn networks with a general structure.

We demonstrate the benefit of our HELM approach for learning expressive networks that combine a varied set of copulas for several sizeable real-life datasets. Specifically, we show that our procedure is not only accurate in terms of identifying the best copula family, but also leads to learned probabilistic graphical models that generalize well. Importantly, this favorable performance comes with dramatic runtime speedups that facilitate learning of models in domains where maximum likelihood structure learning is computationally impractical.

2 BACKGROUND

In this section we briefly describe copulas, their relationship to Spearman's ρ_s measure of association, and the copula network construction.

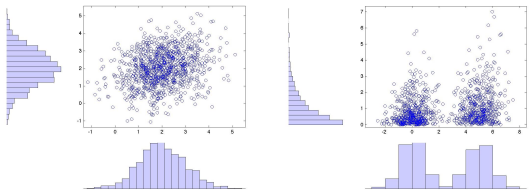


Figure 1: Samples from the bivariate Gaussian copula with $\rho = 0.25$. (left) with Gaussian marginals; (right) with a mixture of Gaussian and Gamma marginals.

Copulas

A copula function joins univariate marginals into a joint real-valued multivariate distributions. Formally, let $U_1, \ldots, U_n$ be random variables marginally uniformly distribution on $[0, 1]$. A copula function $C : [0, 1]^n \rightarrow [0, 1]$ is a joint distribution $C_\theta(u_1, \ldots, u_n) = P(U_1 \leq u_1, \ldots, U_n \leq u_n)$, where θ are the parameters of the copula distribution function.

Now let $\mathcal{X} = \{X_1, \ldots X_n\}$ be an arbitrary set of real-valued random variables. Sklar [1959] states that for *any* CDF $F_\mathcal{X}(\mathbf{x})$, there exists a copula C such that

$$F_\mathcal{X}(\mathbf{x}) = C(F_1(x_1), \ldots, F_n(x_n)).$$

When $F_i(x_i)$ are continuous, C is uniquely defined.

The constructive converse is of particular interest from a modeling perspective: Since $F_i(X_i) \sim U([0, 1])$, *any* copula function taking *any* marginals $\{F_i(X_i)\}$ defines a valid joint cumulative 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.

To derive the joint *density* $f(\mathbf{x}) = \frac{\partial^n F(x_1,\ldots,x_n)}{\partial x_1 \ldots \partial x_n}$ from the copula construction, assuming F has n-order partial derivatives (true almost everywhere when F is continuous), and using the chain rule, we have

$$\begin{aligned} f(\mathbf{x}) &= \frac{\partial^n C(F_1(x_1), \ldots, F_n(x_n))}{\partial F_1(x_1) \ldots \partial F_n(x_n)} \prod_i f_i(x_i) \\ &\equiv c(F_1(x_1), \ldots, F_n(x_n)) \prod_i f_i(x_i), \end{aligned}$$

where we use $c(F_1(x_1), \ldots, F_n(x_n))$ to denote the *copula density function*.

Example 2.1.: The extremely popular Gaussian copula is defined as

$$C_\Sigma(\{U_i\}) = \Phi_\Sigma\left(\Phi^{-1}(U_1), \ldots, \Phi^{-1}(U_N)\right), \quad (1)$$

where Φ is the standard Gaussian, and Φ_Σ is a zero mean Gaussian with correlation matrix Σ.

Copulas and Spearman's Rho
Copulas are intimately connected to many dependence concepts such as Spearman's ρ_s measure of association

$$\rho_s(X_1, X_2) = \frac{cov(F_{X_1}, F_{X_2})}{STD(F_{X_1})STD(F_{X_2})},$$

which is simply Pearson's correlation applied to the cumulative distributions of X_1 and X_2. For the copula associated with the joint $F_{X_1,X_2}(x_1, x_2)$, we have

$$\rho_s(X_1, X_2) = \rho_s(C) \equiv 12 \int \int C(u, v)dudv - 3.$$

Thus, Spearman's ρ_s is monotonic in the copula cumulative distribution function associated with the joint distribution of X_1 and X_2. See [Nelsen, 2007, Joe, 1997] for an in-depth exploration of the framework of copulas and its relationship to dependence measures.

Copula Networks
Similarly to a standard Bayesian network [Pearl, 1988], a copula network uses a directed acyclic graph $\mathcal{G}$ to encode the independencies $I(G) = \{(X_i \perp \mathit{NonDesc}_i \mid \mathbf{Pa}_i)\}$, where $\mathbf{Pa}_i$ are the parents of X_i in $\mathcal{G}$, and $\mathit{NonDesc}_i$ are its non-descendants. $I(G)$ implies a decomposition of the joint density into a product of local conditional densities of each variable given its parents: $f_{\mathcal{X}}(X_1, \ldots, X_n) = \prod_i f_i(X_i \mid \mathbf{Pa}_i)$.

In copula networks, the local densities are defined via the copula ratio

$$f_i(X_i \mid \mathbf{Pa}_i) = \frac{c_\theta(F_i(X_i), \{F_j(X_j)\}_{j \in \mathbf{Pa}_i})}{c_\theta(\{F_j(X_j)\}_{j \in \mathbf{Pa}_i})} f_i(X_i). \quad (2)$$

Appealingly, for copulas the denominator can be easily computed from the numerator without the need for integration. Thus, the representation relies solely on the estimation of joint copulas. See [Elidan, 2010] for more details on the construction and its merits.

3 RELATED WORKS

Broadly speaking, methods for performing copula model selection can be split into three groups. Most commonly, model selection is carried out via (penalized) maximum likelihood estimation, which can be costly due to the need to evaluate the maximum likelihood parameters. In fact, as will be demonstrated in Section 7, even when the maximum likelihood parameters have a simple closed form, the actual computation of the maximum likelihood value can be time consuming in the context of structure learning, where this task is repeated numerous times. A second group of works relies on a measure of deviation between the copula, or some of its statistical properties, from the empirical estimators. Genest and Rivest [1993], for example, use the deviation of Kendall's τ estimates from the population values to select between Archimedean copulas. Unfortunately, for most copulas, characterizing the Kendall distribution requires simulation and can be computationally demanding [Hering and Hofert, 2010]. Another example first employs Rosenblatt's transformation, followed by a deviation measurement relative to the uniform distribution [Justel et al., 1997], a process that can also be computationally intensive.

Fermanian [2005] suggests an alternative in the form of a goodness-of-fit test that is based on kernel density estimation. This, however, still requires tedious numerical integration. Finally, Huard et al. [2006] presents a Bayesian approach that, like our method, is quite generic as it avoids estimation of the maximum likelihood parameters. Posterior computations, however, still require costly integration over the support of Kendall's τ values. In contrast to all of these works, our HELM method uses extremely simple statistics that are easily computed for any copula. As demonstrated in Section 7, this leads to effective performance while offering dramatic speedups.

4 EXPRESSIVE TREE NETWORKS

Our goal is to efficiently learn the structure of copula-based probabilistic graphical models while allowing for different copula families within the same model. We start by considering in this section the building block task of performing selection for bivariate copulas, and show how this building block can be used to learn tree structured networks. Then, after deriving in Section 5 finite sample guarantees for the bivariate case, in Section 6 we propose an extension for learning general networks.

4.1 MULTINOMIAL-BASED SELECTION

Recall that, intuitively, for the purpose of choosing a particular dependence pattern, the distribution is a nuisance parameter and it may be possible to forgo precise estimation. As an example, Figure 2(top) shows the grid frequency of samples from the bivariate Clayton and Gumbel copulas with $\rho_s = 0.5$. The greater emphasis of the Clayton copula on the lower tail is evident as is the converse for the Gumbel copula. Thus, it is possible to choose between the copulas based on simple statistics. Motivated by this example, our selection procedure involves three steps:

1. Precompute a *multinomial signature* for each copula family under consideration.
2. Given a set of training instances, compute an empirical multinomial signature.
3. Choose the copula whose multinomial signature is closest (in some sense) to the empirical one.

We now briefly describe the details involved in each of these three stages.

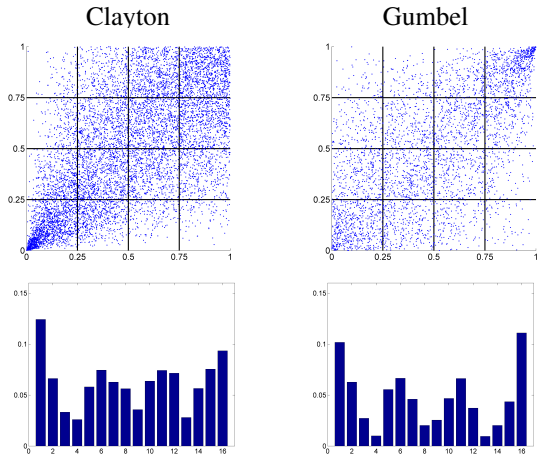


Figure 2: An example of the multinomial mapping for $\rho_s = 0.5$. **(top)** shows the distribution for the Clayton and Gumbel families, overlaid with a 4×4 grid. **(bottom)** shows the corresponding *multinomial signatures* defined by this grid for the two copula families.

Precomputing The Multinomial Signatures

We start by recalling the monotonic relationship, within a copula family, between Spearman's ρ_s and the copula parameter. This allows us to use the notation C_{ρ_s} to refer to a particular instance of the copula family $C_{\theta(\rho_s)}$. To define the copula-to-multinomial mapping, for a copula family C_{ρ_s}, we partition the unit cube into N partitions $\{A_1, \ldots, A_N\}$ and define a multinomial random variable X via

$$P_{C_{\rho_s}}(X = i) = \int_{A_i} c_{\rho_s}(u, v) du dv,$$

where c_{ρ_s} is the corresponding copula density. This mapping defines an N-valued multinomial representation of the copula which we denote by $\pi(c_{\rho_s})$. In principle, computing $P(X = i)$ requires integration. However, for copulas the cumulative distribution function is explicit and, if each A_i is chosen as a rectangular region, then $P(X = i)$ can be readily computed.

For simplicity, we use a generic partition into equal $K \times K$ squares is illustrated in Figure 2(top) for $K = 4$. Finally, since we map the copula to a crude coarsening as it is, in practice we compute the above only for $\rho_s \in \{-1, -0.95, \ldots, 0.95, 1\}$, and use interpolation to define the mapping for intermediary values. This also ensures robustness to small fluctuations in the ρ_s estimate. We use $\pi(c_{\rho_s})$ to denote the resulting multinomial distribution.

Choosing A Copula Family

Given a set of M observations for two random variables X, Y, our task is now to compute the empirical multinomial signature and compare it to the template ones in order to choose the closest copula family. Let $\mathcal{C}$ be the set of candidate copula families from which we wish to choose the most appropriate copula. Omitting the explicit dependence on the data for readability, we use $\hat{\pi}$ and $\hat{\rho}_s$ to denote the empirical multinomial frequency over the $K \times K$ grid and the empirical Spearman's ρ_s estimate, respectively. We choose the copula family $\widetilde{C}$ as the one that minimizes the distance between the empirical and template signatures.[1] That is:

$$\widetilde{C} = \text{argmin}_{C_{\hat{\rho}}} d\left(\hat{\pi} \| \pi(c_{\hat{\rho}_s})\right), \tag{3}$$

where $d(\cdot\|\cdot)$ is a divergence measure between distributions. Several possible choices for this measure come to mind. The KL [Kullback and Leibler, 1951] distance is the divergence of choice between distributions, but can be sensitive to small probabilities which can occur in some of the multinomial grid cells. The L1-norm measure is less sensitive to outliers but does not measure relative deviation. In Section 5 we explore the theoretical properties of both choices, and in Section 7 we demonstrate their empirical merit.

4.2 LEARNING A TREE NETWORK

We now turn to our goal of learning the structure of a high-dimensional tree copula networks. Consider a tree structured model over N variables [Kirshner, 2007, Elidan, 2012] where the joint density can be written as

$$f_{\mathcal{X}}(x_1, \ldots, x_n) = \prod_{(i,j) \in T} c_{ij}(F_i(x_i), F_j(x_k)) \prod_i f_i(x_i),$$

where c_{ij} is the copula associated with the edge (i, j) in the network. Learning the optimal tree structure can be easily carried our using a maximum spanning tree algorithm once the merit of each the $O(N^2)$ candidate edges has been computed. However, even if we have already made the choice of the copula family for each pair of variables, we still need to estimate the maximum likelihood parameters of the copula, and then compute the maximum likelihood score. That is, taking the log of the density, for each pair of variables in the network X_i and X_j, we need to compute

$$\texttt{Score}(i, j) \equiv \sum_{m=1}^{M} \log c_{\hat{\theta}}(F_{X_i}(x_i[m]), F_{X_j}(x_j[m])), \tag{4}$$

where the sum is over instances and $\hat{\theta}$ are the maximum likelihood parameters. As we report in Section 7, the overall computations for the entire network can be demanding.

To overcome this difficulty, we again note that the precise bivariate distribution, defined via $\hat{\theta}$, is a nuisance parameter. In fact, all that we need is a proxy to the above score that will reasonably *rank* candidate edges.[2] Recalling that

[1] In the unlikely case that more than one copula minimizes this mesure, we randomly choose between the minimizing copulas.

[2] We note that the ρ_s-based proxy of Elidan [2012] cannot be used since it assumes the same copula for all edges.

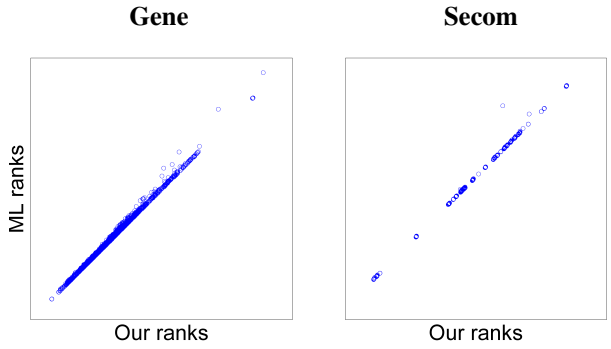


Figure 3: Edge score ranks using maximum likelihood (y-axis) vs. our multinomial proxy score (x-axis) for two of the real-life datasets used in Section 7.

our multinomial mapping roughly approximates the density, it is natural to use the implied multinomial likelihood as a proxy score. Concretely, using $C_{i,j}$ to denote the chosen copula for the variable pair X_i and X_j, instead of Eq. (4), we use

$$\texttt{Score}^{\texttt{Mult}}(i,j) = \sum_k \hat{\pi}_{i,j}[k] \log \pi(c_{i,j})[k], \quad (5)$$

where $\hat{\pi}_{i,j}$ is the empirical multinomial distribution induced by X_i and X_j.

To gauge the quality of this proxy score, Figure 3 compares the ranks of Eq. (5) and Eq. (4) for pairs of variables in two large datasets that we use in our experimental evaluation, **Gene** and **Secom** (see Section 7 for details). It is easy to see that our proxy score is near perfect for the purpose of ranking the benefit of candidate edges. Importantly, computation of $\hat{\pi}$ is linear in the (small) number of cells of the multinomial signature. Consequently, computation of our proxy score is significantly faster than the computation of the likelihood score. As we shall see in Section 7, this results in dramatic speedups of the learning procedure.

5 FINITE SAMPLE BOUNDS

Before describing how our approach can be extended to general structures, in this section we consider the theoretical properties of our building block copula selection method. Clearly, as K is increased in the multinomial mapping, we capture the density at an increasingly better granularity and, assuming continuity, asymptotic recovery follows from standard considerations. In fact, even for fixed small partitions such as $K = 4$, most standard copula families will disagree on some of the $K \times K$ multinomial bins, and asymptotic recovery can be easily guaranteed. We are more interested, however, in providing finite sample guarantees for our algorithm, both when using the Kullback-Leibler divergence and the L1-norm in Eq. (3). To the best of our knowledge ours are the first finite-sample bounds in the context of copula model selection.

We assume ρ_s is known (or has been measured) and omit it from the notation for clarity. Let $\mathcal{C}$ be a finite copula family hypothesis class of cardinality $|\mathcal{C}| = L$ and let $\mathcal{D}$ be a set of M i.i.d training instances sampled from $C^* \in \mathcal{C}$. Denote by $\hat{\pi}$ the empirical multinomial signature of $\mathcal{D}$ and by $\widetilde{C}(\mathcal{D})$ the copula family chosen by our algorithm. The probability of mistakingly identifying the copula family is:

$$\textbf{err}(\mathcal{D}) = P_{C^*}(\widetilde{C}(\mathcal{D}) \neq C^*),$$

where we use P_{C^*} as a shorthand for $P_{\mathcal{D}\sim C^*}$. We will now bound number of instances needed to ensure that the error is below a constant $\textbf{err}(\mathcal{D}) \leq \alpha$.

5.1 KULLBACK-LEIBLER DIVERGENCE

Assume that the data was generated by a specific copula family $C_0 \in \mathcal{C}$. We will later allow C_0 to be any copula in $\mathcal{C}$. We start by observing that deciding between C_0 and some other $C_j \in \mathcal{C}$ based on the KL distance from $\hat{\pi}$ is equivalent to a hypothesis test:

$$\mathcal{H}_0 : \hat{\pi} \sim C_0 \qquad \mathcal{H}_1 : \hat{\pi} \sim C_j,$$

where, using the likelihood ratio test, the rejection region is defined via $\lambda(C_0, C_1; \mathcal{D}) = \frac{P_{\pi(c_j)}(\mathcal{D})}{P_{\pi(c_0)}(\mathcal{D})} > 1$. Thus, classification error can be cast in terms of type I error, giving rise to a finite sample bound:

Lemma 5.1.: *Assume $\mathcal{D} \sim C_0$ or equivalently $\hat{\pi} \sim \pi(c_0)$. There exists a constant $\delta_0(j)$ such that for any $\alpha > 0$ and $C_j \in \mathcal{C}$, if $M \geq log\left(\frac{1}{\alpha}\right)\frac{1}{\delta_0(j)}$ then*

$$P_{C_0}\big(d_{KL}(\hat{\pi}\|\pi(c_j)) \leq d_{KL}(\hat{\pi}\|\pi(c_0))\big) \leq \alpha$$

Proof: By Sanov's theorem [Cover and Thomas, 1991] we have that the type I error is $2^{-Md_{KL}(\pi_0\|\pi(c_0))}$, where π_0 is the closest multinomial to $\pi(c_0)$ that is in the rejection region of the above test. π_0 is given explicitly by:

$$\pi_0[k] = \frac{\pi(c_0)[k]^\lambda \pi(c_j)[k]^{1-\lambda}}{\sum_{k'} \pi(c_0)[k']^\lambda \pi(c_j)[k']^{1-\lambda}}, \quad \lambda \in \mathbb{R},$$

where $[k]$ is the k'th multinomial component, and λ is chosen so that $d_{KL}(\pi_0\|\pi(c_0)) - d_{KL}(\pi_0\|\pi(c_j)) = 0$. Taking $\delta_0(j)$ to be $d_{KL}(\hat{\pi}\|\pi(c_j))$ (see Cover and Thomas [1991] for details on how $\lambda, \delta_0(j)$ can be computed) we get the desired result. Note that π_0 does not depend on α. ∎

Defining $\delta_0 = \min_{j\neq 0}\delta_0(j)$, we then have:

Corollary 5.2.: *Let $\mathcal{D} \sim C_0$. If $M \geq log_2\left(\frac{L-1}{\alpha}\right)\frac{1}{\delta_0}$, then the classification error is bounded from above by α.*

Proof: Using the union bound we have:

$$\begin{aligned} &P_{C_0}(\exists j : d(\hat{\pi}\|\pi(c_j)) \leq d(\hat{\pi}\|\pi(c_0))) \\ &\leq \sum_{C_j \in \mathcal{C}} P_{C_0}(d(\hat{\pi}\|\pi(c_j)) \leq d(\hat{\pi}\|\pi(c_0))) \\ &\leq \sum_{C_j \in C, j\neq 0} \frac{\alpha}{L-1} = \alpha \end{aligned}$$

where, for compactness $d() \equiv d_{KL}()$. The second inequality follows from the above lemma by using $\alpha = \frac{\alpha}{L-1}$ so that $M \geq log\big(\frac{L-1}{\alpha}\big)\frac{1}{\delta_0(j)}$ for all j. ∎

Finally, we can drop the assumption that the specific generating copula family is known and, appealingly, get a bound that grows logarithmically with $\frac{1}{\alpha}$:

Theorem 5.3. : Define $\delta_{\mathcal{C}} = \min_{i:C_i \in \mathcal{C}} \delta_i$. If $M \geq log_2\big(\frac{L(L-1)}{\alpha}\big)\frac{1}{\delta_{\mathcal{C}}}$ then the misclassification error is bounded from above by α.

5.2 L_1 DISTANCE

We now develop parallel bounds for the case of the L1-norm. Denote by $\pi(c_i)[k]$ the k-th component of the multinomial defined by C_i, and define $\delta_0(i)[k] = |\pi(c_0)[k] - \pi(c_i)[k]|$ for $C_0, C_i \in \mathcal{C}$.

Lemma 5.4.: *Let $\mathcal{D} \sim C_0$, and let $\delta_0 = \min_{i \neq 0,k} \delta_0(i)[k]$. If the number of samples satisfies $M \geq log\big(\frac{2(L-1)K^2}{\alpha}\big)\frac{1}{\delta_0^2}$, then the probability of a classification error is bounded from above by α.*

Proof: For compactness define $\Delta_i[k] = |\hat{\pi}[k] - \pi(c_i)[k]|$. Then, since $\hat{\pi} \sim \pi(c_0)$, and using simple union bounds, the probability of misclassification is:

$$
\begin{aligned}
&P_{C_0}\Big(\exists i \neq 0 : \sum_k \Delta_i[k] \leq \sum_k \Delta_0[k]\Big) \\
&\leq \sum_{i \neq 0} P_{C_0}\Big(\sum_k \Delta_i[k]) \leq \sum_k \Delta_0[k]\Big) \\
&\leq \sum_{i \neq 0,k} P_{C_0}(\Delta_i[k] \leq \Delta_0[k])
\end{aligned}
$$

Next, by definition $\Delta_i[k] \leq \Delta_0[k] \Leftrightarrow \Delta_0[k] \geq \delta_0(i)[k]$. Also, since $\mathcal{D} \sim C_0$, we have $E(\hat{\pi}[k]) = \pi(c_0)[k]$. Using Hoeffding's inequality we then have:

$$
\begin{aligned}
P_{C_0}\Big(\Delta_i[k] \leq \Delta_0[k]\Big) &= P_{C_0}\Big(\Delta_0[k] \geq \delta_0(i)[k]\Big) \\
&\leq 2e^{-2M\delta_0^2}.
\end{aligned}
$$

If we now choose the number of samples to be $M \geq log\big(\frac{2(L-1)K^2}{\alpha}\big)\frac{1}{\delta_0^2}$, the result easily follows. ∎

Now, using a similar argument to the KL case, we have

Theorem 5.5. : Define $\delta_{\mathcal{C}} = \min_i \delta_i$. For all α, if $M \geq log_2\big(\frac{L(L-1)K^2}{\alpha}\big)\frac{1}{\delta_{\mathcal{C}}^2}$, then the misclassification error is bounded from above by α.

As an example consider sample data that is distributed according to AMH copula with Spearman's rho equals 0.6. Assuming the copula hypothesis class consists of AMH, Clayton, Gumbel and Plackett copulas. Then using $k = 2$, it is easily verified that $\delta_0 = 0.0713$. Thus, according to 5.4, in order to bound the classification error by $\alpha = 0.05$, at lest 1217 samples are needed.

6 LEARNING GENERAL NETWORKS

We now show how our structure learning approach of Section 4 can be adapted to the more elaborate task of learning a copula graphical model with a general structure. As is commonly done, due to the super-exponential nature of the search space, we learn the structure via a greedy search procedure that involves local modifications to the structure (e.g., add/delete/reverse an edge). Similarly to the case of trees, we start by generalizing the local copula selection building block and then explain how this can be used when learning a global structure.

6.1 CHOOSING THE COPULA

Recall that in order to choose a copula family in the bivariate case, we first evaluate the empirical measure of association $\hat{\rho}_s$, as well as the bivariate statistics of the data, and then choose the copula family signature that is closest to the empirical distribution for $\hat{\rho}_s$. In a nutshell, we will choose a copula family for more than two variables by aggregating bivariate distances.

Before doing so, however, we need to evaluate $\hat{\rho}$. For the Gaussian copula, our path is obvious since each bivariate marginal is characterized by it's own dependence parameters $\Sigma_{i,j}$, and the corresponding measure of association $\hat{\rho}_{i,j}$ can be computed as before. However, the situation is quite different for other copula families. For example, all bivariate marginals of an n-dimensional Archimedean copula have the *same* dependence parameter so that we require $\hat{\rho}_{i,j} = \hat{\rho}$ for all i, j. Thus, a natural choice in the common case of a *one parameter* family is to estimate $\hat{\rho}$ using

$$
\hat{\rho} = \sum_{X_i, X_j, i<j} \binom{n}{2}^{-1} \hat{\rho}_{X_i, X_j}.
$$

Note that this is one of the standard generalizations of Spearman's rho [Schmid and Schmidt, 2007]. Then, with $\hat{\rho}_{i,j} = \hat{\rho}$ in hand, we select the copula family that minimizes the sum of distances between the empirical and template multinomials signatures, similarly to Eq. (3):

$$
\widetilde{C} = \text{argmin}_{C_{\hat{\rho}}} \sum_{X_i, Xj, \, i<j} d\left(\hat{\pi} \| \pi(c_{\hat{\rho}_{i,j}})\right).
$$

6.2 EVALUATING THE STRUCTURE SCORE

As in bivariate case, after choosing the copula family, we still face the challenge of comparing the benefit of different candidate structural changes. Concretely, using $Eq.$ (2), we we need to evaluate the conditional likelihood score:

$$
\begin{aligned}
&\texttt{Score}(i, \mathbf{Pa}_i) \equiv \\
&\sum_{m=1}^{M} \log \frac{c_{\hat{\theta}}(F_i(x_i[m]), \{F_j(x_j[m])\}_{j \in \mathbf{Pa}_i})}{c_{\theta}(\{F_j(x_j[m])\}_{j \in \mathbf{Pa}_i})},
\end{aligned}
$$

	N	F	G	C	A	M
N	**0.87**	0.06	0.02	0.00	0.02	0.02
F	0.05	**0.89**	0.01	0.00	0.02	0.04
G	0.01	0.01	**0.98**	0.00	0.00	0.00
C	0.00	0.00	0.00	**0.97**	0.02	0.01
A	0.02	0.02	0.00	0.03	**0.92**	0.01
M	0.02	0.03	0.00	0.00	0.02	**0.94**

HELM

	N	F	G	C	A	M
N	**0.93**	0.01	0.01	0.00	0.03	0.03
F	0.14	**0.74**	0.00	0.00	0.04	0.08
G	0.13	0.00	**0.87**	0.00	0.00	0.00
C	0.13	0.00	0.00	**0.84**	0.03	0.00
A	0.01	0.01	0.00	0.00	**0.96**	0.02
M	0.00	0.00	0.00	0.00	0.02	**0.98**

Huard

	N	F	G	C	A	M
N	**0.96**	0.02	0.00	0.00	0.00	0.01
F	0.02	**0.94**	0.00	0.00	0.00	0.048
G	0.00	0.00	**1.00**	0.00	0.00	0.00
C	0.00	0.00	0.00	**1.00**	0.00	0.00
A	0.04	0.12	0.00	0.02	**0.79**	0.02
M	0.02	0.03	0.00	0.00	0.00	**0.98**

Costly ML

Figure 4: Copula family selection performance for synthetically generated data with 1000 samples. Methods compared are our HELM, that of Huard [Huard et al., 2006], and time consuming ML estimation. Each confusion matrix shows the percentage of the predicted family (columns) given the generating family (rows).

where $\hat{\theta}$ are the maximum likelihood parameters of the copula associated with X_i and its parents.

Once again, we face the bottleneck of maximum likelihood estimation. Whenever an analytically simple relationship between $\hat{\rho}$ ((or Kendall's τ) and $\hat{\theta}$ exists, we use the heuristic proposed by [P.Embrechts and M.Hofert, 2010] and simply invert the average association measure described above. For other copula families (e.g. Ali-Mikhail), we resort to a standard optimization procedure such as conjugate gradient. Note that even in this case, we perform costly estimation *only* for the chosen copula family, and are thus still significantly more efficient than a full maximum likelihood selection and estimation procedure.

6.3 ADJUSTING THE SCALE OF THE SCORE

To learn a structure that allows for several parents for each variable, all family scores must obviously lie on the same scale. However, the proxy score we use in the case of a single parent is based on a discrete multinomial likelihood (Eq. (5)), while for multiple parents we use a real-valued conditional likelihood (Eq. (6)). Thus, to rank candidate structural changes, we must somehow calibrate these scores relative to each other.

Fortunately, as is clearly evident in Figure 3, our single parent *proxy* scores are almost linearly correlated to the *exact* maximum likelihood scores. Consequently, all that is required in order to accurately approximate the needed scores is to recover this linear transformation via straightforward regression. Concretely, we randomly choose few (e.g. 10%) of the variables pairs, and solve the following regression problem:

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \sum_{i,j} \left(\texttt{Score}(i,j) - \beta_0 - \beta_1 \texttt{Score}^{\texttt{Mult}}(i,j) \right)^2$$

We then use $\hat{\beta}$ to calibrate our bivariate scores:

$$\widetilde{\texttt{Score}}(i,j) = \beta_0 + \beta_1 \cdot \texttt{Score}^{\texttt{Mult}}(i,j).$$

To summarize: starting with the empty graph $G_\emptyset$, we rank the different $O(N^2)$ candidate edges using our multinomial approximation score. Next, we calibrate these scores using the regression coefficients β. These calibrated score are then used together with the multi-parent scores to guide the greedy structure learning procedure.

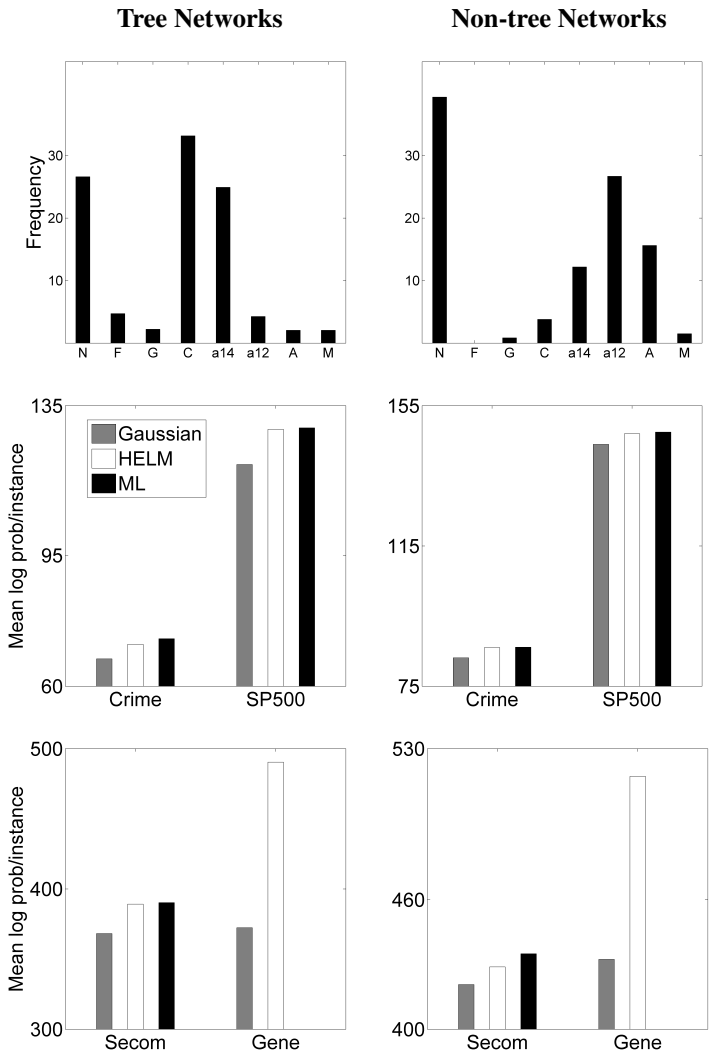
7 EXPERIMENTAL EVALUATION

We now evaluate the ability of our **HELM** approach to efficiently learn expressive copula networks that generalize well. We start by evaluating the merit of the **HELM** model selection building block in the case where the generating distribution is known. We then demonstrate the power of **HELM** when learning high-dimensional structures for sizable real-life domains.

7.1 COPULA MODEL SELECTION

To evaluate our **HELM** copula model selection building block, we synthetically generate i.i.d. instances from different copula families and attempt to identify the generating family from the samples. We compare our **HELM** approach to the standard maximum likelihood (**ML**) approach (using an inversion of the empirical Kendall tau or Spearman's rho for fast estimation where possible), and to a Bayesian approach from the copula community suggested by **Huard** [Huard et al., 2006].

Similarly to Huard et al. [2006], we consider a collection of copula families that exhibit varied dependence patterns: Normal (**N**), Frank (**F**), Gumbel (**G**), Clayton (**C**), Ali-Mikhail-Haq (**A**),and Farlie-Gumbel-Morgenstern (**M**) (see [Joe, 1997, Nelsen, 2007] for details of these copulas). For each family, we precompute its multinomial signature as described in Section 4. To cover a wide range of dependence levels, we generate the synthetic data as follows: for values of Spearman's ρ_s ranging from 0.25 to 0.95, we randomly choose a copula family C, and generate $M = 1000$ i.i.d samples from C_{ρ_s}. We repeat this 1000 times for each value of ρ_s and use the different methods to predict the generating copula family.



Tree Networks

	Gaussian copula	Our method	ML	Speed Factor
Crime	67.31 (0.84)	71.2 (0.79)	72.75 (0.76)	175
Secom	368.19 (3.68)	389.12 (3.73)	390.36 (3.58)	182
SP500	119.36 (3.15)	128.73 (3.18)	129.15 (3.26)	307
Gene	372.43 (4.61)	490.56 (4.83)	–	∞

Non-tree Networks

	Gaussian copula	Our Method	ML	Speed Factor
Crime	83.16 (0.92)	85.79 (0.86)	86.12 (0.89)	78
Secom	420.72 (3.35)	428.96 (3.83)	435.74 (3.79)	76.3
SP500	144.27 (3.45)	146.94 (3.38)	147.52 (3.41)	59.9
Gene	432.42 (4.72)	517.25 (4.63)	–	∞

Figure 5: Comparison of copula networks learned using the different methods for tree and non-tree networks. **(left top)** shows the distribution of the chosen copula families when learning using standard maximum likelihood (**ML**) for the **Secom** dataset. **(left middle/bottom)** summarizes the average test log-probability per instance performance of our method (white bars), **ML** (black bars) and the Gaussian copula baseline (gray bar). **(right)** detailed generalization performance along with standard deviation across random folds (in parentheses) and speedup factor of our method relative to **ML**.

In Figure 4 we report average results in the form of confusion matrices that show the distribution of the predicted copula family (columns) for each generating copula family (rows). Results for our approach are with $K = 8$ and using KL (results were qualitatively similar using $K = 4$ and the L_1 distance). As can be seen, **HELM** surpasses **Huard** on average and is not far beyond the much slower **ML**. This is to be expected since we intentionally took a crude but efficient view of the distribution, a crucial step toward the goal of performing global structure learning.

The above competitiveness comes with substantial computational advantages. A single model selection task when using **ML** took 1.08×10^{-2} seconds on average. Using **HELM** this took only 1.1545×10^{-4} seconds on average, a two orders of magnitude speedup. Advantages are even greater for $K = 4$ since **HELM** is quadratic in K. In this case, **HELM** is close to 200 faster than **ML**, while suffering negligible decrease in predictive performance. Finally, while **Huard** does reasonably well in terms of predictive performance, this comes at an enormous computational cost, taking an average of 0.28 seconds, or 4 orders of magnitude slower than **HELM**.

7.2 LEARNING EXPRESSIVE NETWORKS

We now evaluate the merit of **HELM** for learning expressive copula networks for real-life domains that benefit from a rich mix of local representations. We consider four real-

life datasets that are quite sizable in the context of structure learning of non-Gaussian real-valued models:

- **Crime** (UCI repository). 1994 instances of **100** census variables ranging from household size to fraction of children born out of marriage, for 1994 U.S. communities.
- **Secom** (UCI repository). 1567 instances of **362** variables collected from sensors during a semi-conductor manufacturing process, corresponding to key factors that effect downstream yield.
- **SP500**. End of day changes of the **500** Standard and Poor's index stocks (variables) over a period of close to 2000 trading days (samples).
- **Gene**. A compendium of gene expression. We focus on **999** genes (variables) that have at most one missing experiment, resulting in 2000 samples.

For each domain, we learn a copula network model using **HELM** as well as using standard maximum likelihood (using fast inversion of ρ_s or τ_K where possible). In both cases, we allow for a mix of Gaussian, Frank, Gumbel, Clayton, arch12, arch14, Ali-Mikhail and FGM copulas. To make comparison to the costly ML feasible, we learn networks with up to two parents. For the univariate marginals for both methods, we use a standard kernel-based approach [Parzen, 1962] with the common Gaussian kernel (see, for example, [Bowman and Azzalini, 1997] for details). As an additional baseline, we also consider learning only with a Gaussian copula, which is the strongest of all single family baselines. Finally, we note that due to its significant computational demands, the Bayesian method of **Huard** could not be used in these experiments.

We start by qualitatively demonstrating the real-life need for expressive modeling, or for the combination of different local representations wihin the same model. As an example Figure 5(left top) shows the distribution of the copula families chosen when learning a mixed model using the **ML** method for the **Secom** dataset. Obviously, the learned model is a rich one.

Quantitatively, Figure 5(left middle/bottom) shows that a mixed **ML** model (black bar) also leads to better generalization relative to the best single family baseline (gray bar) in terms of test set log-probability per instance. Also shown is the performance of **HELM** (white bar). As can be seen, **HELM** is competitive with the costly **ML** method. The table on the right includes the average test performance results along with standard deviations (in parentheses) across 10 folds. Importantly, note that the improvement over the single family baseline is significant since the scale of improvement is in bits per instances. Thus, an improvement of, for example, 10 bits per instance is equivalent to each test instance being on average 2^{10} more likely.

Recall that our goal was not simply to learn competitive expressive networks but to do so highly efficiently so as to facilitate scaling up of structure learning. Speed up factors of **HELM** relative to **ML** are reported in the right-hand column of the tables in Figure 5. As can be seen, the runtime improvements are dramatic at over two orders of magnitudes when learning tree networks. To make these numbers concrete, for example, using **HELM** to learn a mixed tree for the **SP500** domain took less than a minute, while for **ML** the average runtime was nearly 5 hours. For the **Gene** data set with 1000 variables, although learning a mixed network with **HELM** took only around 4.5 minutes, **ML** did not terminate after two days. A substantial runtime improvement is also evident for more general structures. For example, learning using **HELM** took around an hour and a half for **SP500**, while learning using **ML** took over three days. Dramatically, although **HELM** was able to learn a mixed **Gene** network model in less than two hours, learning a model for this domain using **ML** proved impractical, and did not terminate within a week.

8 SUMMARY

We presented **HELM**, an algorithm for efficiently learning copula networks that allows for a rich mix of varied copula families within the *same* model. We demonstrated the substantial computational advantages of using our multinomial signature based approach when learning complex models for several varied sizeable real-life domains.

Our contribution is three fold. First, we presented a straightforward but powerful copula model selection building block that, even in the simple bivariate case, is competitive with maximum likelihood and other estimation approaches while offering dramatic runtime improvements. We further derive finite-sample guarantees for this building block. To the best of our knowledge, these are the first such guarantees in the context of copula model selection.

Second, we showed how our building block gives rise accurate and efficient *ranking* of candidate structures, resulting in highly efficient global structure learning. Third, the computational advantages allows us to scale up structure learning and easily cope with domains that are prohibitive if tackled using standard procedures. Indeed, to the best of our knowledge, ours is the first structure learning method that allows for a mix of local real-valued representations and that has been applied to domains of this size.

Acknowledgements

This work was supported in part by an ISF Center of Excellence grant, by an Israeli Ministry of Science center of knowledge and by the Intel Collaborative Research Institute for Computational Intelligence (ICRI-CI). We thank Elad Eban and Elad Mezuman for their valuable comments on different drafts of this work.

References

A. Bowman and A. Azzalini. *Applied Smoothing Techniques for Data Analysis*. Oxford University Press, 1997.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, New York, 1991.

G. Elidan. Lightning-speed structure learning of nonlinear continuous networks. In *Proceedings of the AI and Statistics Conference (AISTATS)*, 2012.

Gal Elidan. Copula Bayesian networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

J.-D. Fermanian. Godness-of-fit tests for copulas. *Journal of Multivariate Analysis*, 95:52–119, 2005.

Christian Genest and L.-P. Rivest. Statistical inference procedures for bivariate archimedean copulas. *Statist.Assoc.*, 88(8):1034 – 1043, 1993.

Chistian Hering and Marius Hofert. Godness-of-fit tests for archimedean copulas in large dimensions. *Working paper*, 2010.

David Huard, Guillaume ívin, and Anne-Catherine Favre. Bayesian copula selection. *Comput. Stat. Data Anal.*, 51 (2):809–822, 2006.

H. Joe. Multivariate models and dependence concepts. *Monographs on Statistics and Applied Probability*, 73, 1997.

A. Justel, D. Pena, and R. Zamar. A multivariate kolmogorov-smornov test of goodness of fit. *Statistical Probability Letters*, 35:251–259, 1997.

S. Kirshner. Learning with tree-averaged densities and distributions. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.

S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:76–86, 1951.

R. Nelsen. *An Introduction to Copulas*. Springer, 2007.

E. Parzen. On estimation of a probability density function and mode. *Annals of Math. Statistics*, 33:1065–1076, 1962.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.

P.Embrechts and M.Hofert. Statistical inference for copulas in high dimension: a simulation study. *Statistical Probability Letters*, 35:251–259, 2010.

Friedrich Schmid and Rafael Schmidt. Multivariate extensions of spearman's rho and related statistics. *Statistics and Probability Letters*, 77(4):407 – 416, 2007.

A. Sklar. Fonctions de repartition a n dimensions et leurs marges. *Publications de l'Institut de Statistique de L'Universite de Paris*, 8:229–231, 1959.

Y. Tenzer and G. Elidan. Speedy model selection (SMS) for copula models. In *Uncertainty in Artificial Intelligence (UAI)*, 2013.

Andrew Wilson and Zoubin Ghahramani. Copula processes. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.