Copula Network Classifiers (CNCs)

Gal Elidan
Department of Statistics, The Hebrew University

Abstract

The task of classification is of paramount importance and extensive research has been aimed at developing general purpose classifiers that can be used effectively in a variety of domains. Network-based classifiers, such as the tree augmented naive Bayes model, are appealing since they are easily interpretable, can naturally handle missing data, and are often quite effective. Yet, for complex domains with continuous explanatory variables, practical performance is often sub-optimal. To overcome this limitation, we introduce Copula Network Classifiers (CNCs), a model that combines the flexibility of a graph based representation with the modeling power of copulas. As we demonstrate on ten varied continuous real-life datasets, CNCs offer better overall performance than linear and nonlinear standard generative models, as well as discriminative RBF and polynomial kernel SVMs. In addition, since no parameter tuning is required, CNCs can be trained dramatically faster than SVMs.

1 Introduction

Learning general purpose classifiers that can make accurate predictions in varied complex domains is one of the core goals of machine learning. Continuous domains abound in real-life and in this work we address the problem of learning effective classifiers in domains with continuous explanatory variables.

Generative networks-based classifiers such as the widely used naive Bayes (NB) model (e.g., [Duda and Hart, 1973]) are appealing as they are easily interpretable and can naturally handle missing data. As demonstrated by Ng and Jordan [2002] (via a comparison to the logistic regression model), such classifiers can also compete with discriminative approaches, depending on the number of training instances, and the particular characteristics of the domain. Using the framework of Bayesian networks (BNs) [Pearl, 1988], the tree-augmented naive Bayes (TAN) model relaxes the independence assumption of the NB model by allowing for a tree structure over the explanatory features [Friedman et al., 1997]. The TAN model can be learned efficiently and, in some domains, can lead to substantial gains in predictive performance. More generally, complex dependency structures in BN based classifiers sometimes offer additional advantages, though at the cost of computational efficiency (e.g., [Grossman and Domingos, 2004]).

In continuous domains, due to practical considerations, network-based classifiers typically rely on simple parametric forms (e.g., linear Gaussian, sigmoid Gaussian), and their predictive accuracy can be sub-optimal. Our goal in this work is to overcome this limitation and develop network-based classifiers for continuous domains that are competitive, and even superior to state-of-the-art discriminative approaches such as the SVM model [Cortes and Vapnik, 1995].

In an unconditional setting, a copula function [Nelsen, 2007] links *any* univariate marginals (e.g. nonparametric) into a coherent multivariate distribution. This can result in a model that is easier to estimate and less prone to over-fitting than a fully nonparametric one, while at the same time avoiding the limitations of a fully parameterized distribution. In practice, copula constructions often lead to significant improvement in density estimation. Indeed, there has been a dramatic growth of academic and practical interest in copulas in recent years, with applications ranging from mainstream economics (e.g., Embrechts et al. [2003]) to hydrologic flood analysis [Zhang and Singh, 2007]. Recently, Elidan [2010] introduced Copula (Bayesian) Networks (CNs) that integrate the cop-

Appearing in Proceedings of the 15^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2012, La Palma, Canary Islands. Volume XX of JMLR: W&CP XX.

ula and Bayesian network frameworks. CNs allow for the construction of high-dimensional graph-based distributions while taking advantage of the flexibility of copulas. In the context of multivariate density estimation, the construction has led to appealing gains in terms of generalization performance. In this work we show that an adaptation of this model results in an effective general purpose classifier.

We present Copula Network Classifiers (CNCs) for classification in domains with continuous explanatory variables. Our building block is the *conditional copula* function (e.g., [Patton, 2006, Hotta and Palaro, 2006]) that allows us to seamlessly construct context specific copulas which depend on the value of the class variable. Similarly to the construction of Elidan [2010], we then build a multivariate classifier by combining such conditional copulas. While conditional copulas have been used in a fully continuous context (e.g., [Hotta and Palaro, 2006, Patton, 2006]), and copulas have been used (though indirectly) in specific discriminative applications (e.g., [Stitou et al., 2009, Krylov et al., 2011]), to the best of our knowledge conditional copulas have not been used to construct high-dimensional classifiers based on continuous explanatory variables.

Theoretically, after replacing the univariate marginals with their conditional counterparts, the validity of our global decomposable construction follows easily along lines similar to the work of Elidan [2010]. The central contribution of the model is in the practical merit of the proposed classifier. Although applicable to general network structures, for concreteness we restrict our attention to the popular TAN structure, and examine ten varied real-life continuous domains. Even with this simple structure, our model offers performance that is superior to standard network-based classifiers, *as well as* a cross validated SVM model using radial basis function or polynomial kernels. At the same time, our model does not require any tuning of parameters, and can be trained dramatically faster (by orders of magnitude) than an SVM model.

2 Background

In this section we briefly review copulas and the recently introduced Copula Network (CN) model [Elidan, 2010]. We start with the necessary notation. Let $\mathcal{X} = \{X_1, \ldots, X_N\}$ be a finite set of real-valued random variables and let $F_{\mathcal{X}}(\mathbf{x}) \equiv P(X_1 \leq x_1, \ldots, X_n \leq x_N)$ be a (cumulative) distribution over $\mathcal{X}$, with lower case letters denoting assignment to variables. For compactness, we use $F_i(x_i) \equiv F_{X_i}(x_i) = P(X_i \leq x_i, X_{\mathcal{X}/X_i} = \infty)$ and $f_i(x_i) \equiv f_{X_i}(x_i)$. When there is no ambiguity we sometimes use $F(x_i) \equiv F_{X_i}(x_i)$, and similarly for densities and for sets of variables.

2.1 Copulas

A copula function [Sklar, 1959] links marginal distributions to form a multivariate one. Formally,

Definition 2.1: Let $U_1, \ldots, U_N$ be real random variables marginally uniformly distributed on $[0, 1]$. A copula function $C : [0, 1]^N \to [0, 1]$ with parameter(s) θ is a joint distribution function

$$C_\theta(u_1, \ldots, u_N) = P(U_1 \leq u_1, \ldots, U_N \leq u_N),$$

where θ are the parameters of the copula function. ∎

Sklar's seminal theorem states that *any* joint distribution $F_{\mathcal{X}}(\mathbf{x})$ can be represented as a copula function C of its univariate marginals. Further, when the marginals are continuous, C is unique. The constructive converse, which is of central interest from a modeling perspective, is also true: *any* copula function taking *any* marginal distributions $\{F_i(x_i)\}$ as its arguments, defines a valid joint distribution with marginals $\{F_i(x_i)\}$. Thus, copulas are "distribution generating" functions that allow us to separate the choice of the univariate marginals and that of the dependence structure expressed in C. This flexibility often results in a real-valued construction that is beneficial in practice.

Assuming C has Nth order partial derivatives (true almost everywhere when the distribution is continuous), we can derive the joint density from the copula function via the derivative chain rule:

$$\begin{aligned} f(\mathbf{x}) &= \frac{\partial^N C_\theta(\{F_i(x_i)\})}{\partial F_1(x_1) \ldots \partial F_N(x_N)} \prod_i f_i(x_i) \\ &\equiv c_\theta(\{F_i(x_i)\}) \prod_i f_i(x_i), \end{aligned} \quad (1)$$

where $c_\theta(\{F_i(x_i)\})$ is called the *copula density*. See Nelsen [2007], Joe [1997] for further details on copulas.

Example 2.2: A simple widely used copula (particularly in the financial community) is the Gaussian copula, which is constructed directly by inverting Sklar's theorem [Embrechts et al., 2003]:

$$C_\theta(\{F_i(x_i)\}) = \Phi_\Sigma\left(\Phi^{-1}(F_1(x_1)), \ldots, \Phi^{-1}(F_N(x_N))\right),$$

where Φ is the standard normal distribution, and Φ_Σ is the zero mean normal distribution with correlation matrix Σ. To get a sense of the power of copulas, Figure 1 shows samples generated from this copula using two different families of univariate marginals. As can be seen, even with a simple elliptical copula, a variety of markedly different and multi-modal distributions can be constructed. More generally, and without any added computational difficulty, we can use different marginals for each variable, and mix and match marginals of different forms with *any* copula function.

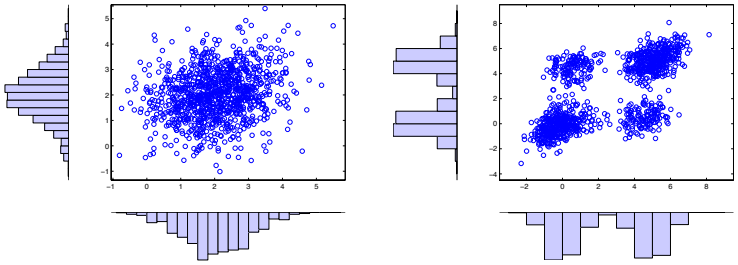


Figure 1: Samples from the bivariate Gaussian copula with correlation $\theta = 0.25$. (left) with unit variance Gaussian marginals; (right) with a mixture of Gaussians marginals.

Normal marginals Mixture of Gaussians marginals

2.2 Copula Networks

Elidan [2010] defines a multivariate density with an explicit univariate representation using a construction that fuses the copula and Bayesian networks (BNs)[Pearl, 1988] formalisms:

Let $\mathcal{G}$ be a directed acyclic graph whose nodes correspond to the random variables $\mathcal{X} = \{X_1, \ldots, X_N\}$, and let $\mathbf{Pa}_i = \{\mathbf{Pa}_{i1}, \ldots, \mathbf{Pa}_{ik_i}\}$ be the parents of X_i in $\mathcal{G}$. As for standard BNs, we use $\mathcal{G}$ to encode the independence statements $I(\mathcal{G}) = \{(X_i \perp ND_i \mid \mathbf{Pa}_i)\}$, where $\perp$ denotes the independence relationship and ND_i are nodes that are not descendants of X_i in $\mathcal{G}$.

Definition 2.3 : A Copula Network (CN) is a triplet $\mathcal{C} = (\mathcal{G}, \Theta_C, \Theta_f)$ that encodes a joint density $f_{\mathcal{X}}(\mathbf{x})$. $\mathcal{G}$ encodes the independence statements $(X_i \perp ND_i \mid \mathbf{Pa}_i)$, that are assumed to hold in $f_{\mathcal{X}}(\mathbf{x})$; Θ_C is a set of local copula functions $C_i(F(x_i), F(\mathbf{pa}_{i1}), \ldots, F(\mathbf{pa}_{ik_i}))$ that are associated with the nodes of $\mathcal{G}$ that have at least one parent; Θ_f is the set of parameters representing the marginal densities $f_i(x_i)$ (and distributions $F_i(x_i)$). The joint density $f_{\mathcal{X}}(\mathbf{x})$ is then takes the form

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

where, if X_i has at least one parent in the graph $\mathcal{G}$, the term R_{c_i} is defined as

$$R_{c_i}(\cdot) \equiv \frac{c_i(F(x_i), F(\mathbf{pa}_{i1}), \ldots, F(\mathbf{pa}_{ik_i}))}{\frac{\partial^K C_i(1, F(\mathbf{pa}_{i1}), \ldots, F(\mathbf{pa}_{ik_i}))}{\partial F(\mathbf{pa}_{i1}) \ldots \partial F(\mathbf{pa}_{ik_i})}}.$$

When X_i has no parents, $R_{c_i}(F(x_i), \emptyset) \equiv 1$. ∎

The term $R_{c_i}(F(x_i), F(\mathbf{pa}_{i1}), \ldots, F(\mathbf{pa}_{ik_i}))f_i(x_i)$ is simply a conditional density $f(x_i \mid \mathbf{pa}_i)$, and can be easily computed. Specifically, whenever the copula density $c_\theta(\cdot)$ has an explicit form, so does this term.

Elidan [2010] shows that a CN defines a coherent joint density, and further that the product of local ratio terms R_{c_i} defines a joint copula over $\mathcal{X}$. Thus, like other probabilistic graphical models, a CN takes advantage of the independence assumptions encoded in $\mathcal{G}$ to represent $f_{\mathcal{X}}(\mathbf{x})$ compactly via a product of local terms. Differently from a regular BN, a CN has an explicit marginal representation leading to practical advantages (see Elidan [2010] for more details).

3 Copula Network Classifiers

While the CN construction of Elidan [2010] offers a general tool for constructing effective multivariate continuous distributions, it cannot be used directly for the important task of classification. This is because copula distributions cannot be defined over categorical variables. To overcome this difficulty, we propose a construction that resembles that of Elidan [2010], but that relies on *conditional* copulas. As in the case of CNs, the building block of our construction is a copula-based parameterization of a variable in the graph G given its parents. Differently, each continuous variable is also allowed to have one or more discrete class variables as parents, thereby defining a distribution that can be used for classification. After some preliminaries, we describe the local conditional copula building block, and then show how it can be used to define our copula-based multivariate classifier network.

3.1 Conditional Copulas

We start by defining the conditional copula construction. For any continuous random variable X and a set of random variables $\mathbf{W}$, $F_X(x|\mathbf{w}) \equiv P(X \leq x|\mathbf{w})$ is distributed uniformly on $[0, 1]$. Thus, a standard copula function $C : [0, 1]^N \to [0, 1]$, if given conditional univariate marginals $\{F_i(x_i|\mathbf{w})\}$ as arguments, defines a distribution conditioned on $\mathbf{W} = \mathbf{w}$. As in the unconditional case, a parallel to Sklar's theorem allows us to represent any continuous multivariate conditional distribution as a copula function of its conditional univariate marginals:

$$C_\theta(\{F_i(x_i|\mathbf{w})\}) = P(X_1 \leq x_1, \ldots, X_N \leq x_N \mid \mathbf{w}).$$

Note that the arguments of the copula function are the *conditional* univariate marginals, and that the parameters of copula function itself may also depend on $\mathbf{W} = \mathbf{w}$. The conditional joint density is derived from

the conditional copula function using the derivative chain rule

$$f(x_1,\ldots,x_n \mid \mathbf{w}) = c_\theta(\{F_i(x_i|\mathbf{w})\} \mid \mathbf{w}) \prod_i f_i(x_i \mid \mathbf{w})$$

where, similarly to the unconditional case, the conditional copula density is defined as

$$c_\theta(\{F_i(x_i|\mathbf{w})\} \mid \mathbf{w}) \equiv \frac{\partial^N C_\theta(\{F_i(x_i|\mathbf{w})\} \mid \mathbf{w})}{\partial F_1(x_1|\mathbf{w}) \ldots \partial F_N(x_N|\mathbf{w})}.$$

3.2 A Copula-based Conditional Representation

The conditional copula has been used in a *continuous* context to facilitate modeling of dynamics (e.g., [Hotta and Palaro, 2006, Patton, 2006]), or to facilitate multivariate constructions inspired by the chain rule (e.g., [Bedford and Cooke, 2002]). In both cases $\mathbf{W}$ are continuous and, in general, copulas are not used to define distributions that include categorical variables. However, the above definition can also easily be used when $\mathbf{W}$ defines discrete events. This allows us to parameterize a conditional density $f(x \mid \mathbf{y}, \mathbf{w})$, where $\mathbf{Y}$ are continuous random variables and $\mathbf{W}$ are discrete as follows:

Lemma 3.1 : *Let $X, \mathbf{Y}$ be continuous random variables and let $\mathbf{W}$ be a set of discrete random variables. There exists a conditional copula $C_\theta(F(x|\mathbf{w}), F_1(y_1|\mathbf{w}), \ldots, F_K(y_K|\mathbf{w}) \mid \mathbf{w})$ such that*

$$f(x \mid \mathbf{y}, \mathbf{w}) = R_{c|\mathbf{w}}(F(x|\mathbf{w}), F_1(y_1|\mathbf{w}), \ldots, F_K(y_K|\mathbf{w}))f(x|\mathbf{w})$$

where $R_{c|\mathbf{w}}$ is the ratio

$$R_{c|\mathbf{w}}(\cdot) \equiv \frac{c_\theta(F(x|\mathbf{w}), F_1(y_1|\mathbf{w}), \ldots, F_K(y_K|\mathbf{w}) \mid \mathbf{w})}{\frac{\partial^K C_\theta(1, F_1(y_1|\mathbf{w}), \ldots, F_K(y_K|\mathbf{w})|\mathbf{w})}{\partial F_1(y_1|\mathbf{w}) \ldots \partial F_K(y_K|\mathbf{w})}},$$

and where $R_{c|\mathbf{w}}$ is defined to be 1 when $\mathbf{Y} = \emptyset$. The converse is also true: for any *conditional copula function $C_\theta(\cdot)$ and* any *univariate marginals conditioned on $\mathbf{w}$, the expression $R_{c|\mathbf{w}}(F(x|\mathbf{w}), F_1(y_1|\mathbf{w}), \ldots, F_K(y_K|\mathbf{w}))f(x \mid \mathbf{w})$ defines a valid conditional density $f(x \mid \mathbf{y}, \mathbf{w})$.*

Note that the Kth order derivative in the denominator of $R_{c|\mathbf{w}}$ is actually simpler to compute than the copula density itself (a (K+1)th order derivative). Thus, computation of this term does not require the costly integration that we would expect in a normalization term. The proof of the above result is similar to the unconditional parallel in Elidan [2010].

3.3 A Multivariate Network Classifier

We can now define our copula-based network classifier, aimed at predicting one or more discrete class variables $\mathcal{W}$ given continuous explanatory variables $\mathcal{X}$. For simplicity of notation, below we use $\mathbf{pa}_{ik}$ to denote the assignment of the k'th continuous parent of X_i in the graph $\mathcal{G}$, and use $\mathbf{w}_i$ to denote the value of the set of discrete class variables that are parents of X_i in $\mathcal{G}$.

Definition 3.2: A Copula Network Classifier (CNC) probabilistic graphical model encodes a joint density $f_{\mathcal{X},\mathcal{W}}(\mathbf{x}, \mathbf{w})$ using four components:

- $\mathcal{G}$ is a directed acyclic graph over $\mathcal{X}$ and $\mathcal{W}$ that encodes $I(\mathcal{G})$. The graph is constrained so that each class variable $W \in \mathcal{W}$ is not a descendant of any continuous variables $X \in \mathcal{X}$.
- Θ_C is a set of local conditional copula functions $C_i(F_i(x_i|\mathbf{w}_i), \{F_{\mathbf{Pa}_{ik}}(\mathbf{pa}_{ik}|\mathbf{w}_i)\} \mid \mathbf{w}_i)$ that are associated with the nodes of $\mathcal{G}$ that have at least one continuous parent.
- Θ_f parameterize the univariate marginal densities. These include $f_i(x_i \mid \mathbf{w}_i)$ of each variable given its discrete class parents in $\mathcal{G}$. In addition, for each child Y of X_i in $\mathcal{G}$, the univariate marginal of X_i given the discrete parents of its child Y is also explicitly represented.
- Θ_W is a discrete parameterization $P(w \mid \mathbf{pa}_w)$ (e.g., table) of each $W \in \mathcal{W}$ given its (possibly empty) set of discrete class variable parents.

The joint density $f_{\mathcal{X},\mathcal{W}}(\mathbf{x}, \mathbf{w})$ is defined as

$$f_{\mathcal{X},\mathcal{W}}(\mathbf{x}, \mathbf{w}) = \prod_{l \in \mathcal{W}} P(w_l \mid \mathbf{pa}_{w_l}) \prod_{i \in \mathcal{X}} f_i(x_i|\mathbf{w}_i) \prod_{i \in \mathcal{X}} R_{c_i|\mathbf{w}_i}(F_i(x_i|\mathbf{w}_i), \{F_{\mathbf{Pa}_{ik}}(\mathbf{pa}_{ik}|\mathbf{w}_i)\}) \quad (2)$$

∎

Theorem 3.3: Let $\mathcal{C}$ be a Copula Network Classifier (CNC) as defined above. Then:

1. The product of conditional copula ratios $R_{c_i|\mathbf{w}_i}$ (last term in Eq. (2)) defines a valid joint conditional copula of $\mathcal{X}$ *given* $\mathcal{W}$.
2. $f_{\mathcal{X},\mathcal{W}}(\mathbf{x}, \mathbf{w})$ as defined in Eq. (2) is a valid joint density over $\mathcal{X}$ *and* $\mathcal{W}$.

Proof: The proof of the first claim is similar to the parallel unconditional case in Elidan [2010]. The second claim follows directly from the first claim by applying Eq. (2) to the joint density. ∎

We note that a converse decomposition theorem also holds: If $I(\mathcal{G})$ hold in $f_{\mathcal{X},\mathcal{W}}(\mathbf{x},\mathbf{w})$, then the joint density decomposes as in Eq. (2).

To summarize, a CNC defines a coherent joint distribution over the explanatory variables $\mathcal{X}$ and the class variables $\mathcal{W}$, while allowing for an explicit parameterization of the univariate marginals. Similarly to the unconditional CN model, this allows us to flexibly compose *any* local copulas with *any* univariate forms. We can then easily use Bayes' rule to infer the most likely label(s) given an assignment to the explanatory variables. Importantly, as we demonstrate in our experimental evaluation on real-life data, this results in consistently competitive predictive performance.

4 Learning

Our Copula Network Classifier (CNC) model falls into the broad category of probabilistic graphical models and as such facilitates the use of available estimation and structure learning techniques. For lack of space, we describe these standard techniques briefly.

Univariate Marginal Estimation
To estimate $f_i(x_i|\mathbf{w})$ we use a standard kernel-based approach [Parzen, 1962]. Given $x[1],\ldots,x[M]$ i.i.d. samples of a random variable X in the context $\mathbf{W}=\mathbf{w}$, the density estimate

$$\hat{f}_h(x|\mathbf{w}) = \frac{1}{Mh(\mathbf{w})}\sum_{i=1}^{M} K\left(\frac{x-x_i}{h(\mathbf{w})}\right)$$

where K is a kernel function and h is the bandwidth parameter. Qualitatively, the method approximates the distribution by placing small "bumps" (determined by the kernel) at each data point. We use the standard Gaussian kernel and the common heuristic of choosing h that is optimal under normality assumptions (e.g., [Bowman and Azzalini, 1997]).

Parameter Estimation
Given a dataset $\mathcal{D}$ of M instances where all of the variables are observed in each instance, the log-likelihood of the m'th instance given a CNC model $\mathcal{C}$ is

$$\ell(\mathcal{D}:\mathcal{C})[m] = \sum_{X_i}\left(\log f_i(x_i[m]|w[m]) + \log R_{c_i|\mathbf{w}_i}[m]\right) + \sum_{w} logP(w[m]|\mathbf{pa}_w[m])\Big], \quad (3)$$

where $R_{c_i|\mathbf{w}_i}[m]$ is a shorthand for the value that the conditional copula ratio $R_{c_i|\mathbf{w}_i}$ takes in the m'th instance. While the log-likelihood objective (the sum of Eq. (3) over instances) appears to fully decompose according to the structure of the graph $\mathcal{G}$, each marginal distribution $F_i(x_i|\mathbf{w})$ actually appears in several copula terms. A solution commonly used in the copula community is the Inference Functions for Margins approach [Joe and Xu, 1996], where the marginals are estimated first. Given $\{F_i(x_i|\mathbf{w})\}$, we can estimate the parameters of each local copula *independently*, e.g., using a standard conjugate gradient approach.

Structure Learning and Model Selection
To learn the structure $\mathcal{G}$ of a CN, we rely on a model selection score that balances the likelihood of the model with its complexity, such as the Bayesian Information Criterion (BIC) of Schwarz [1978]:

$$\text{score}(\mathcal{G} \;:\; \mathcal{D}) = \ell(\mathcal{D}:\hat{\theta},\mathcal{G}) - \frac{1}{2}\log(M)|\Theta_{\mathcal{G}}|, \quad (4)$$

where $\hat{\theta}$ are the maximum likelihood parameters, and $|\Theta_{\mathcal{G}}|$ is the number of free parameters associated with the model. The learning task is then to find the structure that maximizes the score.

In the case of a TAN model (used in our evaluation), the optimal tree over the explanatory variables can be learned efficiently via a maximum spanning tree algorithm (see Friedman et al. [1997] for details in the context of standard BNs). More generally, a greedy search procedure that is based on local modifications to the graph (e.g., add/delete/reverse and edge) is commonly used (see Koller and Friedman [2009] for more details).

Finally, the BIC score of Eq. (4) (or any similar model selection criteria) can also be used to perform automated selection between the different copula families that parameterize the CNC model.

5 Experimental Results

Experimental Setup
To assess the merit of our (CNC) model, we consider the prevalent scenario where $\mathbf{W}$ is a single class variable, and learn a tree augmented naive Bayes (TAN) structure. We compare our copula based model to two network-based classifiers: a TAN with a standard linear Gaussian conditional distribution where $X_i|W=w \sim N(\beta_0+\beta x_j,\sigma)$, where X_j is the parent of X_i in the structure; a sigmoid nonlinear TAN with $X_i|W=w \sim N(\alpha_0+\alpha_1\frac{1}{1+e^{\beta_0+\beta x_j}},\sigma)$. Note that in both cases, though not made explicit for readability, the parameters depend on w. When learning the optimal structure over the explanatory variables in all of these models, we use the BIC score of Eq. (4). We consider two copula-based TAN variants: one that uses only the Gaussian copula and one that, based *only* on training data, used the BIC score to select between the Gaussian and Clayton copula (a representative of an Archimedean copula [Nelsen, 2007]).

Table 1: The ten datasets from the UCI and Statlog machine learning repositories

Name	Attr	Classes	Instances	Prediction Goal
Heart	13	2	270	presence or absence of heart disease
Iris	4	3	150	the classic Iris sub-type identification task
Pima	8	2	768	presence of diabetes in Pima indians
Cardio	36	3	2126	heart pathology level from fetal cardiograms
Magic	10	2	19020	distinguish gamma from hadron particles
Wine	11	6	1599	quality based on chemical characteristics
Parkin	22	2	195	presence of Parkinson's based on speech signal
Mini	50	2	2500	distinguish electron from muon neutrinos
Glass	9	6	214	glass identification based on chemical components
Shuttle	9	7	4506	one of seven shuttle status characterization

We also compare to a strong discriminative model: an SVM trained using the SVMLight package [Thorsten, 1999]. To estimate both the cost parameter β that balances training error and margin and the kernel parameter, we use 5-fold cross-validation on the training set. Following the widely cited protocol of Hsu and Lin [2002], we consider $\beta \in \{2^{-12}, 2^{-11}, \ldots, 2^{-2}\}$. For the width parameter γ of the RBF kernel, we allow $\gamma \in \{2^4, 2^2, 2^2, \ldots, 2^{-10}\}$, for a total of 225 parameter combinations. For the polynomial kernel, we consider degrees $d \in \{2, 3, \ldots, 10\}$ for a total of 135 settings. We note that in the experiments below, the full range of the parameter settings was selected for different repetitions of different domains. Thus, without a-priori knowledge, the space of parameters considered cannot be substantially smaller. Further, for some settings in four domains (Cardio, Wine, Glass, Shuttle), the SVM optimization was hopelessly slow, with a single run taking more than 10^4 times longer than the network-based learning. Thus, to facilitate learning of the SVM model for all domains, for each parameter setting, each of the 5 cross-validation evaluations was limited to 10 minutes on an Intel Xeon X5550 Processor (in partial, one fold, experiments, results were essentially the same with a 30 minute limit).

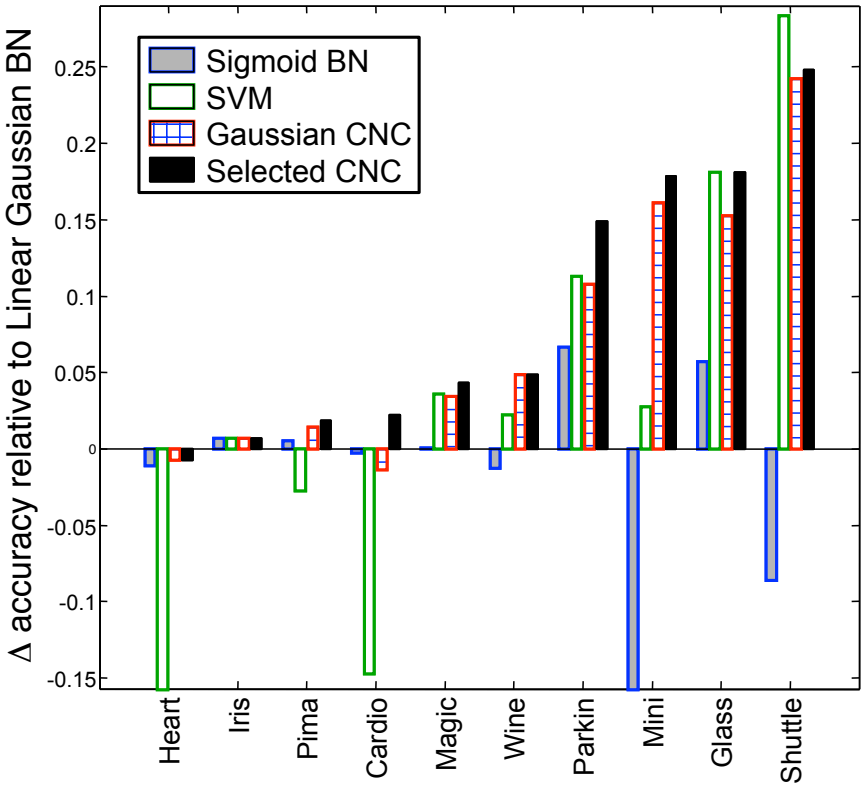
We evaluate all methods on ten varied datasets from the UCI and Statlog machine learning repository [Frank and Asuncion, 2010] that include continuous explanatory variables. The properties of the different datasets are summarized in Table 1. For each dataset, we use a 5-fold random train/test split and report average and range test results over these folds.

Prediction Accuracy

Figure 2 summarizes the test prediction accuracy of the different methods. The graph shows the average accuracy on a 5-fold random train/test partitioning relative to the linear Gaussian BN baseline. The table below provides more details where for each model and dataset, the minimum (across random folds), average and maximum performance is shown.

We start by assessing the merit of our CNC model when compared to the two network-based competitors. The nonlinear sigmoid Gaussian BN (gray fill bar) is better than the baseline in 5/10 domains but can also fail miserably (Mini and Shuttle domains) and should generally be used with caution. In contrast, our Gaussian CNC model (grid red bar) is always superior to the sigmoid Gaussian model, is better than the baseline in 8/10 domains, and is only slightly inferior than the baseline in the other 2 domains. For 5/10 domains the advantage of the Gaussian CNC model relative to the baseline is substantial with an improvement in accuracy from 5% to 25%. If we also allow automated selection between the Gaussian and Clayton copulas (solid black bar), the performance of the CNC model further improves. In this case our model is slightly inferior to the baseline only in a single domain, and is superior to the baseline in 9/10 domains.

Next, we compare our CNC network-based model to the discriminative cross-validated SVM model with a radial basis function kernel (performance with a polynomial kernel was almost always worse and is not reported). When compared to the linear Gaussian BN baseline, performance of the SVM model (green no-fill bar) is mixed: it is better in 7/10 domains but can also fail miserably (Heart and Cardio domains). This is consistent with the comparative evaluation of Ng and Jordan [2002], where simpler generative and discriminative models were evaluated against each other. When compared to our CNC model, the SVM classifier is clearly inferior: our Gaussian CNC model dominates the SVM model on average and significantly so in 4/10 domains. In addition, our stronger selected CNC model is equal or better than the SVM model in 9/10 domains, loses to the SVM model only in a single domain, and is better than the SVM model by more than 5% in 4/10 domains. In summary, the selected CNC model clearly dominates the baseline network-based classifiers, as well as the discriminative SVM competitor, and is overall best in 8/10 domains.



Model		Heart	Iris	Pima	Cardio	Magic	Wine	Parkin	Mini	Glass	Shuttle
Linear	Min	0.72	0.90	0.66	0.93	0.76	0.52	0.64	0.68	0.43	0.64
Gaussian	Avg	**0.84**	0.96	0.74	0.96	0.77	0.54	0.70	0.70	0.52	0.71
BN	Max	0.94	1.00	0.78	0.98	0.77	0.57	0.82	0.71	0.67	0.79
Sigmoid	Min	0.76	0.90	0.70	0.93	0.76	0.50	0.69	0.28	0.48	0.36
Gaussian	Avg	0.83	**0.97**	0.74	0.96	0.77	0.53	0.77	0.30	0.58	0.63
BN	Max	0.94	1.00	0.77	0.97	0.78	0.55	0.85	0.33	0.67	0.81
	Min	0.59	0.93	0.66	0.79	0.80	0.50	0.69	0.71	0.57	1.00
SVM	Avg	0.66	**0.97**	0.71	0.81	0.81	0.57	0.82	0.72	**0.70**	**1.00**
	Max	0.74	1.00	0.74	0.84	0.81	0.61	0.87	0.74	0.86	1.00
Gaussian	Min	0.78	0.93	0.71	0.92	0.80	0.56	0.74	0.80	0.52	0.95
CNC	Avg	0.83	0.97	0.75	0.95	0.81	**0.59**	0.81	0.86	0.68	0.95
	Max	0.89	1.00	0.79	0.97	0.81	0.61	0.90	0.90	0.81	0.96
Gauss/	Min	0.78	0.93	0.73	0.97	0.80	0.56	0.79	0.86	0.52	0.96
Clayton	Avg	0.83	**0.97**	**0.76**	**0.98**	**0.81**	**0.59**	**0.85**	**0.88**	**0.70**	0.96
CNC	Max	0.89	1.00	0.79	0.99	0.82	0.61	0.95	0.90	0.86	0.96

Figure 2: 5-fold test class prediction accuracy of the different models for the 10 datasets. The graph shows the average improvement relative to the linear Gaussian BN. The table shows the minimum, average and maximum prediction performance over the 5 test folds. The best result on average for each dataset appears in bold.

Running Time

The obvious question is whether the performance of our CNC model comes with a computational price. In fact, the opposite is true. Learning a Gaussian CNC can be carried out in closed form and requires no tuning of any parameters. Learning a selected CNC also requires learning of the parameters of the Clayton copula via estimation of the Kendall's τ statistics [Nelsen, 2007], but still involves no parameter tuning. In contrast, SVM learning always involves some form of parameter tuning using cross-validation. As noted, the full range of the parameter settings described above was selected for different repetitions of different domains so that without a-priori knowledge, the space of parameters considered cannot be substantially smaller.

Figure 3 shows the average learning time of the different models as a factor of the running time of the Gaussian CNC model on a logarithmic scale . A single run of the SVM model for a *given* value of the cost and

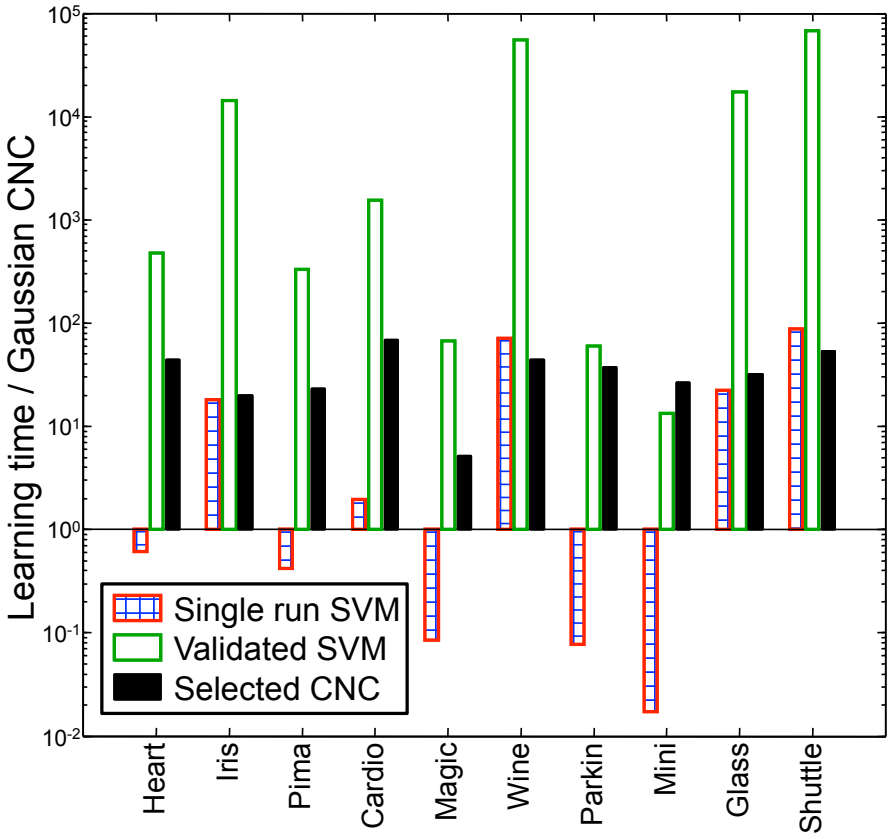


Figure 3: Learning time as a factor of the learning time of a Gaussian CNC model averaged over the 5 random folds. Shown is the learning time factor of: a single SVM run given specific width and cost parameter values (red grid bar); a cross-validated SVM (green no-fill bar); a selected CNC model (solid black bar).

width parameters (grid red bar) is faster (below the black line at 10^0) for 5/10 domains and slower for the other 5/10 domains. Learning a cross-validated SVM (green no-fill bar), however, requires significantly more time than a Gaussian CNC or a selected CNC (solid black bar), often by several orders of magnitude. In fact, learning the SVM model took (slightly) less time than the selected CNC model only in a single domain (Mini). The performance of the SVM model in this case was significantly worse (by 16%). In 8/10 other domains the CNC model performed better while requiring significantly less running time. In the single domain where the SVM performed better by almost 4% (Shuttle), this came at a cost of an increased learning time by a factor of over 1000.

6 Conclusions and Future Work

We have presented the Copula Network Classifier (CNC) model for performing classification given continuous explanatory variables with non-Gaussian interactions. Based on a fusion of the conditional copula and the Bayesian networks frameworks, our model allows for the incorporation of categorical target variables within a copula-based model. We demonstrated the consistent predictive effectiveness of our model relative to baseline network-based classifiers, as well as a strong discriminative SVM model. At the same time, since no parameter tuning is needed, our model can be trained significantly faster than an SVM model.

While our experimental evaluation was focused on a TAN structure due to the simplicity and popularity of this model, our construction can accomodate more complex dependency structures. In addition, one of the important benefits of a copula-based model is that a large number of copula families and univariate parameterizations could be considered for different variables in the graph, without significantly increasing the computational requirements. We expect that both of these extensions will improve performance in sufficiently challenging domains. More importantly, our formal construction also allows for multiple categorical class variables, so that high-dimensional structured prediction could naturally be incorporated. We plan to explore these possibilities in future work.

Acknowledgements

Part of the research was support by a Google Research Award. G. Elidan was supported by an Alon fellowship. I thank Elad Eban for useful comments on a draft of this work.

References

T. Bedford and R. Cooke. Vines - a new graphical model for dependent random variables. *Annals of Statistics*, 2002.

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

C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20, 1995.

R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, 1973.

G. Elidan. Copula bayesian networks. In *Neural Information Processing Systems (NIPS)*, 2010.

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

A. Frank and A. Asuncion. UCI machine learning repository, 2010. URL `http://archive.ics.uci.edu/ml`.

N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29:131–163, 1997.

D. Grossman and P. Domingos. Learning bayesian network classifiers by maximizing conditional likelihood. In *Internationl Conference on Machine Learning*, pages 361–368, 2004.

L. Hotta and H. Palaro. Using conditional copulas to estimate value at risk. *Journal of Data Science*, 4 (1):930–115, 2006.

C. Hsu and C. Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 2002.

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

H. Joe and J. Xu. The estimation method of inference functions for margins for multivariate models. Technical Report 166, Department of Statistics, University of British Columbia, 1996.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

V. Krylov, G. Moser, S. Serpico, and J. Zerubia. Supervised high resolution dual polarization sar image classification by finite mixtures and copulas. *IEEE Journal of Selected Topics in Signal Processing*, 5 (3):554–566, 2011.

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

A. Ng and M. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Neural Information Processing Systems (NIPS)*, 2002.

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

A. Patton. Modelling asymmetric exchange rate dependence. *International Economic Review*, 47(2): 527–556, 2006.

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

G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

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. Stitou, N. Lasmar, and Y. Berthoumieu. Copulas based multivariate gamma modeling for texture classification. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1045–1048, 2009.

J. Thorsten. *Making large-scale support vector machine learning practical*, pages 169–184. MIT Press, 1999.

L. Zhang and V. Singh. Trivariate flood frequency analysis using the gumbel-hougaard copula. *Journal of Hydrologic Engineering*, 12, 2007.