Transductive Learning for Multi-Task Copula Processes

-Technical Report-

Markus Schneider M.SCHNEIDER@ACFR.USYD.EDU.AU
Institute for Artificial Intelligence
Ravensburg-Weingarten University of Applied Sciences, Germany
formerly affiliated with
Australian Centre for Field Robotics
The University of Sydney, Australia

Fabio Ramos F.RAMOS@ACFR.USYD.EDU.AU
Australian Centre for Field Robotics,
School of Information Technologies
The University of Sydney, Australia

Abstract

We tackle the problem of multi-task learning with copula process. Multivariable prediction in spatial and spatial-temporal processes such as natural resource estimation and pollution monitoring have been typically addressed using techniques based on Gaussian processes and co-Kriging. While the Gaussian prior assumption is convenient from analytical and computational perspectives, nature is dominated by non-Gaussian likelihoods. Copula processes are an elegant and flexible solution to handle various non-Gaussian likelihoods by capturing the dependence structure of random variables with cumulative distribution functions rather than their marginals. We show how multi-task learning for copula processes can be used to improve multivariable prediction for problems where the simple Gaussianity prior assumption does not hold. Then, we present a transductive approximation for multi-task learning and derive analytical expressions for the copula process model. The approach is evaluated and compared to other techniques in one artificial dataset and two publicly available datasets for natural resource estimation and concrete slump prediction.

1 Introduction

Multi-task learning is valuable in many areas of research such as spatial-temporal modeling, environmental sciences, numerical optimization and data fusion. In these problems it is advantageous to predict more than one quantity at a time (in contrast to single-task learning) to exploit inter-dependencies. Kernel-based algorithms achieve this by the use of an appropriate *multi-task* kernel. *Gaussian process* (GP) [15] based regression, as a simple and fully probabilistic model, is often the tool of choice for such problems. The GP framework supports an easy specification of a regression prior using a mean function and a kernel and delivers closed form solutions at inference stage for the predictive mean and variance. However, in many cases the assumption of a Gaussian likelihood of the data is incorrect, but accepted because of the mathematical elegance of the GP framework and the lack of alternatives with comparable performance.

Copulas, with roots in statistics [17] are models that separate the dependence structure of two or more random variables from their marginal distribution, thus possessing the flexibility of using a different probability distribution function for each variable. Informally, they perform a transformation that maps each variable through its cumulative distribution function (cdf) to the unit interval and captures the dependence between the variables using a coupling term. This methodology can handle complex joint distributions between random variables offering tractable solutions for conditional and marginal operations. Copula distributions can be extended to stochastic processes [8] with the help of kernels. It can be shown that the Gaussian process model is just a special case of the copula process if Gaussian marginal distributions are used. This makes copula processes an appealing replacement for GPs in cases where the Gaussian assumption is not appropriate. In this paper, we address the computational costs of copula processes, which make their application to multi-task problems difficult. We introduce a general *transductive* approximation and provide analytical expressions for multi-task copula processes.

1.1 Related Work

Copula processes are relatively new in machine learning. After the fundamental work on copula processes [8], an alternative to GARCH models for finance applications using a copula based stochastic volatility model was proposed [23]. It can also be shown [22] that a heavy-tailed process, derived from copula theory, can provide robustness against outliers in the data. In geostatistics the copula process is called *copula based Kriging estimator* [9] and had been introduced as a possible improvement over Gaussian random fields.

Multi-task learning is a more general form of co-Kriging where predictions for multiple quantities are made at the same time. Several different methods had been proposed for multi-task Gaussian processes: The task dependence can be introduced with shared hyper parameters [12] or an appropriate prior on the covariance matrix as, for example an inverse-Wishart distribution [25]. It is also possible to construct new kernel functions [4, 5] if the GP is considered as a convolution of a continuous white noise process with a smoothing kernel [7].

The *Bayesian committee machine* (BCM) [20] is a *local* approximation for general probabilistic learning algorithms. The BCM divides the complete training data set into smaller subsets, which are trained individually and then re-combined again for predictions. It belongs to the family of transductive algorithms because the predictive distribution depends on the number and location of the query points. The algorithm is very popular for Gaussian process regression with its high demand for memory and computational time, however it is in general not straight forward how to divide the training data set. Other *global* methods for Gaussian processes such as *Deterministic Training Conditional* [16], *Fully Independent Training Conditional* [18] and *Partially Independent Training Conditional* [6] approximation generate a sparse covariance matrix by identifying and using only a representative subset of the training data while ignoring or approximating the other samples in the set. A framework and an excellent discussion about these methods can be found in [6] and we will show later how our approximation can fit into it. Recently a structured noise covariance that is independent of the inputs but captures residual correlation between tasks was proposed in [14]. Gaussian process regression networks [24] model related tasks with an adaptive mixture of GPs.

The novelty of this work lies in the derivation of a transductive approximation for Bayesian multi-task problems. We show how the computational complexity can be handled for a large number of tasks, which normally would grow significantly, the more variables are estimated simultaneously. Furthermore we show the practical consequences on Gaussian copula processes with a multi-task kernel obtained through process convolution.

2 Copula Processes

Copulas are a statistical framework to decompose a joint distribution of random variables $H(y_1, \ldots, y_n)$ into their univariate marginal cumulative distribution functions (cdf) $F_1, \ldots, F_n$ and a coupling term, the actual copula. Hereby each random variable gets mapped though its marginal distribution into the $[0, 1]$ interval, called probability integral transformation $u_i = F_i(y_i)$. The task is then to find a copula $\mathcal{C}$, such that

$$\begin{aligned} H(y_1, \ldots, y_n) &= \mathcal{C}\left(F_1(y_1), \ldots, F_n(y_n)\right) \\ &= \mathcal{C}\left(u_1, \ldots, u_n\right). \end{aligned} \tag{1}$$

The distribution $\mathcal{C}$ has to meet certain requirements [10, 13], but it can be proven [17], that a decomposition as in Eq. (1) exists for every joint distribution H.

Selecting one specific copula, it is also possible to create a huge set of different multivariate distributions by varying different marginal distribution functions. A copula with favorable analytical properties is the *Gaussian copula* C which can be constructed from the multivariate Gaussian cdf $\Phi_{\mu,\Gamma}$ with mean μ and covariance matrix Γ as

$$C_{\mu,\Gamma}(u) = \Phi_{\mu,\Gamma}\left(\Phi^{-1}_{\mu_1,\Gamma_{11}^{1/2}}(u_1), \ldots, \Phi^{-1}_{\mu_n,\Gamma_{nn}^{1/2}}(u_n)\right),$$

where $u = (u_1, \ldots, u_n)$, $\Phi_{\mu_i,\Gamma_{ii}^{1/2}}$ is the ith univariate Gaussian cdf with $\Gamma_{ii}^{1/2}$ as the square root of the i-th diagonal element of matrix Γ and μ_i as the i-th element of vector μ. Its density c can be derived as

$$c_{\mu,\Gamma}(u) = \frac{\mathcal{N}_{\mu,\Gamma}\left(\Phi^{-1}_{\mu_1,\Gamma_{11}^{1/2}}(u_1), \ldots, \Phi^{-1}_{\mu_n,\Gamma_{nn}^{1/2}}(u_n)\right)}{\prod_{i=1}^{n} \mathcal{N}_{\mu_i,\Gamma_{ii}^{1/2}}\left(\Phi^{-1}_{\mu_i,\Gamma_{ii}^{1/2}}(u_i)\right)}, \tag{2}$$

where we used $\mathcal{N}$, the Gaussian density, as the derivative of Φ. A *Gaussian copula process* can be created [8] if the Gaussian distribution $\Phi_{\mu,\Gamma}$ gets replaced by though a Gaussian process.

A Gaussian process [15] $\{Z_x\}$ is a collection of Gaussian random variables indexed by x with mean function $m(x)$ and a positive definite kernel $k(x, x')$. For a finite set of input locations $X = (x_1, \ldots, x_n)$ and corresponding outputs $y = (y_1, \ldots, y_n)$ we write the density of the finite dimensional subset of $\{Z_x\}$ as $h_{Z_X}(y) = p(Z_X) = \mathcal{N}_{\mu(X),K(X,X)}(y)$ and mean the multivariate normal density with kernel matrix $[K(X, X)]_{i,j} = k(x_i, x_j)$ and mean vector $[\mu(X)]_i = m(x_i)$. To ease notation we follow

[22] and assume the mean function to be constant zero and $k(x,x) = \gamma^2, \forall x$ from now on.

Given these notations we can construct a Gaussian copula process $\{Y_x\}$ with marginal distribution function $F_1, \ldots, F_n$ as

$$C_{0,K(X,X)}(F_1(y_1), \ldots, F_n(y_n))$$

$$p(Y_X) = c_{0,K(X,X)}(F_1(y_1), \ldots, F_n(y_n)) \cdot \prod_{i=1}^{n} \frac{\partial F_i(y_i)}{\partial y_i}. \tag{3}$$

Notice that we can recover the warped Gaussian process [19] if we use $\Phi_{0,\gamma}^{-1}(F_i(y_i))$ as warping functions and can also get the Gaussian process as a special case if we set $F_i = \Phi_{0,\gamma}$.

The predictive distribution $p(Y_{X^*}|Y_X)$ for $y^* = (y_1^*, \ldots, y_m*)$ at $X^* = (X_1^*, \ldots, X_m^*)$ with margins $F_1^*, \ldots, F_m^*$ can be obtained as

$$p(Y_{X^*}|Y_X) = c_{\hat{\mu},\hat{\Gamma}}(F_1^*(y_1^*), \ldots, F_m^*(y_m^*)) \cdot \prod_{i=1}^{m} \frac{\partial F_i^*(y_i^*)}{\partial y_i^*} \tag{4}$$

$$\hat{\mu} = K(X, X^*)^T K(X, X)^{-1} w$$
$$\hat{\Gamma} = K(X^*, X^*) - K(X, X^*)^T K(X, X)^{-1} K(X, X^*)$$

and $w_i = \Phi_{0,\gamma}^{-1}(F_i(y_i))$.

2.1 Making Predictions

In the inference step we normally want to provide an estimate, a single number with uncertainty bounds rather than a full predictive distribution. In machine learning this is often the mean and the variance. This can be problematic for the Gaussian copula process, since (depending on the marginal distribution function) these quantities may not exist. Furthermore, since the predictive distribution can be asymmetric, the variance may not be informative enough. Hence we suggest to calculate the *median* and the *quantiles* of the predictive distribution and provide the expressions next. In order to get a quantile $Q(p)$ at input X_i^* we use

$$Q(p) = F_i^{*-1}\left(\Phi_{\mu_i,\Gamma_{ii}^{1/2}}\left(\Phi_{\hat{\mu}_i,\hat{\Gamma}_{ii}}^{-1}(p)\right)\right),$$

where $\hat{\mu}_i$, $\hat{\Gamma}_{ii}$ are the i-th entry from $\hat{\mu}$, $\hat{\Gamma}$ as in Eq. (4) and F_i^{*-1} is the quantile distribution of the corresponding cdf[1]. The median as the 0.5 quantile is then, using the equation above, given by

$$\begin{aligned} Q(0.5) &= F_i^{*-1}\left(\Phi_{\mu_i,\Gamma_{ii}^{1/2}}\left(\Phi_{\hat{\mu}_i,\hat{\Gamma}_{ii}}^{-1}(0.5)\right)\right) \\ &= F_i^{*-1}\left(\Phi_{\mu_i,\Gamma_{ii}^{1/2}}\left(\hat{\mu}_i\right)\right), \end{aligned}$$

since the 0.5 quantile, the median, of a Gaussian distribution is its mean.

[1] Notice, that the dependence on X_i^*, X and y is introduced implicitly by these variables.

2.2 Multi-Task Copula Processes

In contrast to single-task learning, where the objective is to estimate a scalar valued quantity, the aim of multi-task learning is to estimate more than one variable at a time. The applications of multi-task learning are broad, but very often the estimation of a primary variable of interest can be improved if we take other correlated variables (secondary or co-variables) into account. Fig. 1 illustrates this concept. A typical situation is the case where only a small sample set of the primary variable is available, but a larger data set for the secondary variables. This can happen, for example if the primary variable is much more difficult or expensive to estimation and occurs frequently in geology or environmental setting.

The challenge to extend a kernel-based algorithm (such as GPs or the Gaussian copula process) to a multi-task version gets reduced to the problem of defining an appropriate multi-task kernel. Some multi-task kernels are inspired from co-Kriging theory [21] as, for example the *intrinsic correlation model* (ICM) and *linear model of corregionalization* (LMC). Others are more recent such as the convolutional kernel [7]. Given kernels for the individual tasks, the convolutional kernel attempts to find a kernel for all cross-task dependencies such that the resulting kernel matrix is still positive definite. For example, the cross-task kernel between a *squared exponential*, $k(r) = \exp(-r^2/l_{SE}^2)$, and a *Matérn* kernel with smoothness $\nu = 3/2$, $k(r) = (1 + \sqrt{3}r/l_M)\exp(-\sqrt{3}r/l_M)$ is

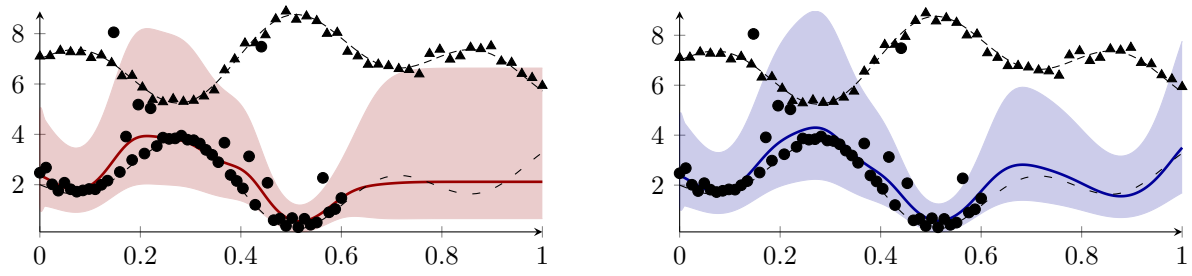
$$\begin{aligned} k^{dq}(x,x') = k^{dq}(r) = \sqrt{\lambda}\left(\frac{\pi}{2}\right)^{1/4} e^{\lambda^2}\Bigg[2\cosh\left(\frac{\sqrt{3}r}{l_M}\right) \\ - e^{\frac{\sqrt{3}r}{l_M}}\,\mathrm{erf}\left(\lambda + \frac{r}{l_{SE}}\right) - e^{\frac{\sqrt{3}r}{l_M}}\,\mathrm{erf}\left(\lambda - \frac{r}{l_{SE}}\right)\Bigg] \end{aligned}$$

where $\lambda = \frac{\sqrt{3}}{2}\frac{l_{SE}}{l_M}$, $\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-z^2}\,\mathrm{d}z$, $r = ||x - x'||^2$ and l_{SE}, l_M are the length scales for the squared exponential (task d) and the Matérn (task q) respectively [11].

We do not have to change anything on the equations for the copula processes, but use a multi-task kernel instead of the ordinary kernel function. We can then merge the inputs and outputs from different tasks into the sets X and y respectively and do the same for test inputs X^* and outputs y^*.

2.3 Parameter Estimation

The copula process model is not entirely parameter free as the kernel and the univariate marginal distributions are usually parameterized in some way. We will denote the set



(a) Single-Task Gaussian Copula

(b) Multi-Task Gaussian Copula

Figure 1: Comparison between single-task (left) and multi-task (right) Gaussian copula processes for an artificial dataset. The dashed lines representing the (latent) true functions with noisy samples marked as circles and triangles. The observations for the primary task (circles) are corrupted noisy versions of the true function. The noise was sampled from a generalized extreme value distribution. For the observations of the co-task (triangles) we simply added a zero mean Gaussian noise. The solid lines are the predictive median, whereas the shaded region is the area in between the 0.05 and 0.95 quantiles.

of all these parameters with θ. One of the advantages of Bayesian methods is the ability to estimate such parameters from data rather than using cross-validation. We follow the standard procedure and using a maximum likelihood approach. More specifically, we are going to minimize the negative log-likelihood

$$L(\theta) = -\log(P(Y_X;\theta)),$$

where $P(Y_X;\theta)$ is as in Eq. (3), but we now explicitly annotate the dependence on θ. A common approach to minimize this non-convex function is to use conjugate gradient optimization with random restarts or simulated annealing. This requires numerous evaluations of L. As it can be seen from Eq. (3), the biggest computational costs are introduced by $\mathcal{N}_{0,K(X,X)}(y)$ from the numerator in Eq. (2) which is given by

$$\frac{1}{(2\pi)^{n/2}|K(X,X)|}\exp\left(-\frac{1}{2}y^T K(X,X)^{-1}y\right),$$

where costs are dominated by the inversion of $K(X,X)$ with $\mathcal{O}(n^3)$. Especially for multi-task problems this becomes rapidly troublesome. Recall, that n is the number of elements in X and y and that, in the case of multi-task learning, we collected the data from all tasks in these two sets. If we assume that each of our t tasks has roughly the same number of training examples, say $\bar{n}$, then the complexity is $\mathcal{O}((t\bar{n})^3)$. This makes multi-task learning computationally very difficult and we introduce approximation scheme next to attack this problem.

3 Transductive Multi-Task Learning

As mentioned in the previous section, many learning algorithms, such as the ones we used in this work, can only handle a limited number training data efficiently. This makes it even harder to apply to multi-task problems, since each task carries additional data. In Kriging, Gaussian processes and Gaussian copula processes we have to do a covariance (kernel) matrix inversion, which scales cubic with the number of training data. In this section we present a transductive approach for multi-task algorithms inspired by the Bayesian committee machine [20].

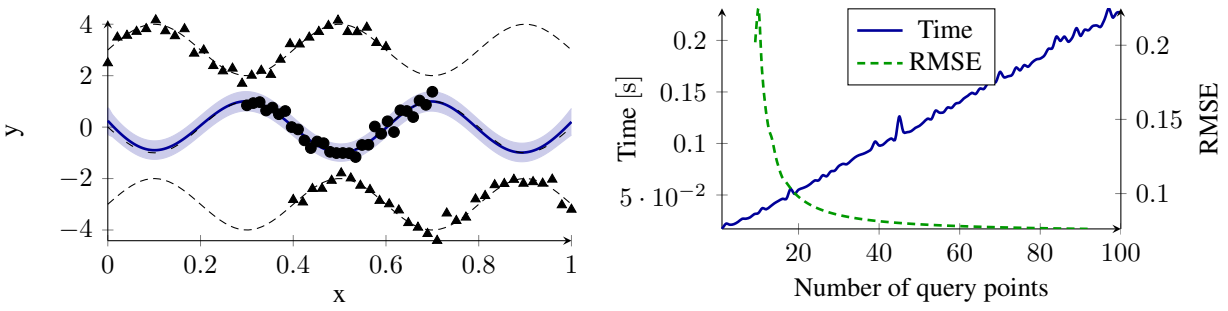
Informally speaking, we are going to perform multi-task learning with the primary variable of interest and each of the secondary variables individually and combine the results at the end. This will reduce the computational costs to $\mathcal{O}(t\bar{n}^3)$.

Theorem 1. *Let $Y_{X_1},\ldots,Y_{X_t}$ be the random variables modeling each of the t tasks and we assume without the loss of generality that we want to make predictions for the primary variable $Y_{X_1^*}$ for task 1. Using the assumption that any two Y_{X_i}, Y_{X_j} with $i \neq j \in \{2\ldots,t\}$ are conditionally independent given Y_{X_1} and $Y_{X_1^*}$, we can approximate the full multi-task model as*

$$P(Y_{X_1^*}|Y_{X_1},\ldots,Y_{X_t}) \approxeq \frac{\prod_{i=2}^{t} P(Y_{X_1^*}|Y_{X_1},Y_{X_i})}{P(Y_{X_1^*}|Y_{X_1})^{t-2}} \cdot const.$$

Proof. With the help of the Bayes' rule and chain rule we write

$$P(Y_{X_1^*}|Y_{X_1},\ldots,Y_{X_t}) = \frac{P(Y_{X_1^*})P(Y_{X_1}|Y_{X_1^*})\cdots P(Y_{X_t}|Y_{X_1},\ldots,Y_{X_{t-1}},Y_{X_1^*})}{P(Y_{X_1},\ldots,Y_{X_t})},$$



(a) Experiment Setup

(b) Time vs. Accuracy

Figure 2: a) shows the setup of the toy example. The primary variable to estimate is the black dashed line in the middle and the black circles are the noisy samples. Above and below are the co-variables and their samples (black triangles). The copula median estimate for the primary variable is the blue solid line in the middle nearly on top of the primary variable. The shaded region denotes the area between the 0.05 and 0.95 quantiles. b) shows how the prediction time and the estimation error behaves if the number of query points gets enlarged. It can be seen, that the predictions getting better with more query points. This is a property of transductive algorithms.

which is true in general. In the next step we are making our independence assumptions which yields

$$\begin{aligned}
&P(Y_{X_1^*}|Y_{X_1},\ldots,Y_{X_t}) \\
&\quad\approxeq \frac{P(Y_{X_1^*})P(Y_{X_1},|Y_{X_1^*})\prod_{i=2}^t P(Y_{X_i}|Y_{X_1},Y_{X_1^*})}{P(Y_{X_1},\ldots,Y_{X_t})} \\
&\quad= \frac{P(Y_{X_1^*})P(Y_{X_1}|Y_{X_1^*})^{t-1}\prod_{i=2}^t P(Y_{X_i}|Y_{X_1},Y_{X_1^*})}{P(Y_{X_1},\ldots,Y_{X_t})P(Y_{X_1}|Y_{X_1^*})^{t-2}} \\
&\quad= \frac{P(Y_{X_1^*})\prod_{i=2}^t P(Y_{X_i},Y_{X_1}|Y_{X_1^*})}{P(Y_{X_1},\ldots,Y_{X_t})P(Y_{X_1}|Y_{X_1^*})^{t-2}} \\
&\quad= \frac{\prod_{i=2}^t P(Y_{X_1^*}|Y_{X_1},Y_{X_i})}{P(Y_{X_1^*}|Y_{X_1})^{t-2}}\cdot \text{const},
\end{aligned}$$

where we multiplied in the second step the whole equation with $P(Y_{X_1}|Y_{X_1^*})^{t-1}/P(Y_{X_1}|Y_{X_1^*})^{t-1}$ and used Bayes' rule again in the last step. □

Notice, that with this approximation, we never have to learn a model for more than two tasks at a time, which gives the computational speedup and also provides a way to easily distribute the computation to several machines.

If we apply the approximation to Gaussian copula processes, the numerator and denominator are conditional Gaussian copula densities of the form as in Eq. (4). This is advantageous since we only have to deal with products and quotients of Gaussian distributions introduced by Eq. (2), for which analytical solutions are available. More precisely, the approximate predictive distribution for the Gaussian copula process is then

$$\begin{aligned}
P(Y_{X_1^*}|Y_{X_1},\ldots,Y_{X_t}) &\approxeq c_{\dot\mu,\dot\Gamma}(F_1^*(y_1^*),\ldots,F_m^*(y_m^*)) \\
&\quad\cdot\prod_{i=1}^m \frac{\partial F_i^*(y_i^*)}{\partial y_i^*},
\end{aligned}$$

where $\dot\mu$ and $\dot\Gamma$ can be obtained from

$$\mathcal{N}_{\dot\mu,\dot\Gamma} = \prod_{t=2}^t \frac{\mathcal{N}_{\hat\mu_{1,i},\hat\Gamma_{1,i}}}{\mathcal{N}_{\hat\mu_{1,i},\text{diag}(\hat\Gamma_{1,i})}}\left(\frac{\mathcal{N}_{\hat\mu_1,\text{diag}(\hat\Gamma_1)}}{\mathcal{N}_{\hat\mu_1,\hat\Gamma_1}}\right)^{t-2}, \quad (5)$$

and $\hat\mu_1,\hat\Gamma_1$, $\hat\mu_{1,i},\hat\Gamma_{1,i}$ are defined as in Eq. (4) if we calculate the predictive distribution for $P(Y_{X_1^*}|Y_{X_1})$ and $P(Y_{X_1^*}|Y_{X_1},Y_{X_i})$ respectively. For example $\hat\Gamma_{1,i}$ would be obtained as

$$\begin{aligned}
\hat\Gamma_{1,i} &= K(X_1^*,X_1^*) - K([X_1,X_i],X_1^*)^T \\
&\quad\cdot K([X_1,X_i],[X_1,X_i])^{-1}K([X_1,X_i],X_1^*),
\end{aligned}$$

which is also the main contributor to the complexity of $\mathcal{O}(8(t-1)\bar n^3) = \mathcal{O}(t\bar n^3)$. Eq. 5 above can be further reduced with the rules for products and quotients of Gaussian distributions which can be found in standard textbooks and in [20], but we omit it here due to paucity of space. Please note also that all $y_1^*,\ldots,y_m^*$ are from the primary task and so are their univariate marginal distributions $F_1^*,\ldots,F_m^*$.

If we follow [6], we can also see our transductive approximation as an inducing approach, where the so called *inducing variables* are defined to be Y_{X_1} and $Y_{X_1^*}$. Using this point of view, it may be easier to see that the quality of prediction can depend on the number of query points $Y_{X_1^*}$ used. As in general for transductive algorithms, the prediction becomes better, the more query points are used (see

Fig. 2). As a consequence, even if only a few estimations are needed, one should include artificial dummy test inputs in the prediction step and then discard them. In most cases this is not a serious problem, since the training/parameter estimation phase is the one, which takes an order of magnitude more time than the prediction phase.

4 Experiments

The experiments in this section share the following setup: In the first step, we train a standard Gaussian copula process for each task individually and determine which combination of kernel and marginal function is the best for each specific task. This procedure yields good results in general. We use either the squared exponential or the Matérn kernel and take the convolutional multi-task kernel approach to calculate the cross-kernels. For the marginal distribution functions we choose one from the following set: The *normal, log-normal, exponential, generalized extreme value (GEV), gamma, t*-distribution or a *parzen window* estimator. In the second step we use maximum likelihood on the full multi-task Gaussian copula process to optimize the parameters for the kernel and the marginal distributions alltogether.

4.1 A Toy Dataset

The experiment is done on an artificial dataset to demonstrate the methodology. The dataset consists of three highly dependent tasks (see Fig. 2(a)), where only noisy samples of the true (latent) functions are available to the algorithm. The transductive Gaussian copula process is used as described in and can be summarized as follows: First, we learn two distinct multi-task copula processes, the primary variable (middle) together with each of the secondary variables (top and bottom). In the second step we merge the results again to obtain a single multi-task copula process. In this toy example, the full model is omitted, since it is nearly identical to the approximation.

As mentioned in the previous section, the performance of transductive algorithms can depend on the number and location of the query inputs. Fig. 2(b) illustrates this property. We slowly increasing the number of query inputs, starting with only one and calculate the root-mean-squared-error (RMSE) and the time needed for the prediction averaged over 20 trials. The prediction time increases as expected with the number of query points while the prediction accuracy improves. As we will see, the choice of query points does not have a significant impact on the other two datasets.

4.2 The Jura Dataset

The second experiment is performed on the Jura dataset which contains 359 samples of two categorical variables

	Cd [Ni, Zn]	Cu [Pb,Ni,Zn]
StGCP	0.56 ± 0.07	14.43 ± 2.62
MtGCP	$\mathbf{0.42 \pm 0.06}$	$\mathbf{6.57 \pm 1.04}$
TransGCP	0.44 ± 0.09	6.96 ± 1.43
CK	0.51	7.8
StGP	0.57	15.8
MtGP	0.44	7.5
D200	~ 0.46	-
F359	~ 0.47	-
P200	~ 0.45	-

Table 1: A comparison between various algorithms for the elements Cd and Cu. The table shows the mean of the absolute error and corresponding standard deviation. The first three rows are our implementation of the single-task Gaussian copula process (StGCP), multi-task Gaussian copula process (MtGCP) and the transductive approximation (TransGCP). The other rows are the numbers for the single-task Gaussian process (StGP) and multi-task Gaussian process (MtGP) from [1] and co-Kriging (CK) from [3]. The last three entries are from [2, Fig. 8] and are the DTC, FITC and PITC approximations with 200, 359 and 200 inducing points respectively.

	Opt. Time	Time/Eval.
MtGCP Cd [Ni, Zn]	898 s	0.517 s
TransGCP Cd [Ni, Zn]	429 s	0.363 s
MtGCP Cu [Pb,Ni,Zn]	1046 s	0.625 s
TransGCP Cu [Pb,Ni,Zn]	621 s	0.409 s
D200 Cd [Ni, Zn]	185 s	-
F359 Cd [Ni, Zn]	691 s	-
P200 Cd [Ni, Zn]	385 s	-

Table 2: The table shows the comparison between the full multi-task copula process (MtGCP) and the transductive approximation (TransGCP) for Cadmium (Cd) and Copper (Cu). The first column indicates the algorithm followed by the primary variable and the secondary variables in brackets. The second column shows the total time needed for the marginal likelihood optimization (Opt. Time) and the last column show the time needed per marginal likelihood function evaluation (Time/Eval). The last three entries are from [2, Table 8] and the algorithm did not run on the same machine as our results. We just provide the figures for completeness and a rough baseline.

(land uses and rock type) and the concentration of seven chemical elements (Cadmium, Cobalt, Chromium, Copper, Nickel, Lead and Zinc) from a 14.5 km^2 region of the Swiss Jura. As in the previous experiment the primary variable has fewer samples than the secondary variables. This can occur in real datasets if, for example, the concentration of one element is harder or more expensive to estimate or the dataset contains missing values. For comparison reasons we use exactly the same setup as in [3, 1]:

- the dataset is divided into 259 training samples and 100 test samples for the primary variable, but all 359 samples are used for the secondary variables;
- for Cadmium (Cd) as the primary variable, the secondary variables are Nickel (Ni) and Zinc (Zn);
- for Copper (Cu) as the primary variable, the secondary variables are Lead (Pb), Nickel (Ni) and Zinc (Zn).

Furthermore we are using the Matérn kernel for Cd, Ni and Cu and the squared exponential kernel for Zn and Pb. We are modeling the marginal distribution functions for Cd, Ni and Cu with a generalized extreme value distribution and for Zn and Pb a Gamma distribution is used.

We compare the *mean absolute error* (MAE) for various algorithms in Table 1 and show the comparison between the full multi-task copula process and the transductive approximation in Table 2. The number of test inputs did not have a significant influence (less than 1%) on the prediction results and therefore the numbers were omitted.

We also included results from [1, 2] showing the approximation for convolved multi-task Gaussian processes. Note that the convolved Gaussian processes approximation uses less inducing inputs and is therefore not expected to be as good as our transductive approach. Furthermore we did not run the Gaussian process approximations on the same machine and therefore the runtime (in seconds) is not directly comparable, but were included for completeness.

4.3 The Concrete Slump Dataset

Our last experiment is performed on the concrete slump dataset with 103 data points, 7 input variables (cement, slag, fly ash, water, SP, Coarse Aggr. and Fine Aggr.) and 3 output variables (slump (cm), flow (cm) and 28-day compressive strength (mpa)). The goal is to estimate the variable slump with flow and compressive strength as secondary variables. We split the dataset randomly into 83 training and 20 evaluation points and calculate the average over 100 runs (see Table 3). We found that the combination of Matérn kernel and the generalized extreme value distribution yielded the best results for all three variables. Interestingly the predictions of the transductive approximation are (on average) better than the predictions of the full model. This can happen since the optimization problem for the transductive Gaussian copula process is slightly easier than the one from the full model and the optimizer can sometimes find a better solution. As for the Jura dataset we did not found any significant changes in predictions if we vary the number of query points.

	MtGCP	TransGCP
RMSE	5.65 ± 2.15	5.47 ± 1.89
MAE	4.08 ± 1.47	3.97 ± 1.29
Opt. Time	386 ± 150 s	320 ± 105 s
Time/Eval	0.18 ± 0.09 s	0.09 ± 0.06 s

Table 3: The table shows the comparison between the full multi-task copula process (MtGCP) and the transductive approximation (TransGCP) for the slump dataset. The rows are the *root mean squared error* (RMSE), *mean absolute error* (MAE), total time needed for the marginal likelihood optimization (Opt. Time) and the last row shows the time needed per marginal likelihood function evaluation (Time/Eval). All results are averaged over 100 trials $\pm$ standard deviation.

5 Conclusions

This work introduced a new transductive approximation methodology for multi-task learning to solve the computational challenges in copula processes. Copula processes are extremely useful in cases where the assumptions of Gaussian processes are invalid. They allow a different marginal distribution for each query variable while capturing the inter-task dependencies. We showed how the Gaussian copula process framework could be extended to multi-task learning with appropriate kernels and addressing computational challenges. We derived closed-form expressions for the transductive approximation which reduces the computational costs from $\mathcal{O}((t\bar{n})^3)$ to $\mathcal{O}(t\bar{n}^3)$. Furthermore we investigate experimentally, on one synthetic, and two real public datasets, the different properties of the transductive learning approach.

5.1 Acknowledgments.

This work was supported by the Rio Tinto Centre for Mine Automation and the Australian Centre for Field Robotics.

References

[1] M. A. Alvarez and N. D. Lawrence. Sparse convolved gaussian processes for multi-output regression. In *NIPS*, pages 57–64, 2008.

[2] M. A. Alvarez and N. D. Lawrence. Computationally efficient convolved multiple output gaussian

processes. *Journal of Machine Learning Research*, 12:1425–1466, 2011.

[3] R. S. Bivand, E. J. Pebesma, and V. Gómez-Rubio. *Applied Spatial Data Analysis with R*. Springer, 2008.

[4] P. Boyle and M. Frean. Dependent gaussian processes. In *Advances in neural information processing systems 17: proceedings of the 2004 conference*, volume 17, page 217. The MIT Press, 2005.

[5] P. Boyle and M. Frean. Multiple output gaussian process regression. Technical report, Victoria University Of Wellington, 2005.

[6] J. Q. Candela and C. E. Rasmussen. A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 2005.

[7] D. Higdon. A process-convolution approach to modelling temperatures in the north atlantic ocean. *Environmental and Ecological Statistics*, 5(2):173–190, 1998.

[8] S. Jaimungal and E. K. H. Ng. Kernel-based copula processes. *Machine Learning and Knowledge Discovery in Databases*, pages 628–643, 2009.

[9] H. Kazianka and J. Pilz. Copula-based geostatistical modeling of continuous and discrete data including covariates. *Stochastic Environmental Research and Risk Assessment*, 24(5):661–673, 2010.

[10] N. Kolev, U. dos Anjos, and B.V.M. Mendes. Copulas: A review and recent developments. *Stochastic models*, 22(4):617–660, 2006.

[11] A. Melkumyan and F. Ramos. Multi-kernel gaussian processes. In *International Joint Conference on Artificial Intelligence*, pages 1408–1413, 2011.

[12] T.P. Minka and R.W. Picard. Learning how to learn is learning with point sets, 1997.

[13] R. B. Nelsen. *An Introduction to Copulas (Lecture Notes in Statistics)*. Springer, 1 edition, October 1998.

[14] B. Rakitsch, C. Lippert, K. M. Borgwardt, and O. Stegle. It is all in the noise: Efficient multi-task gaussian process inference with structured residuals. In *NIPS*, pages 1466–1474, 2013.

[15] C. E. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[16] M. Seeger, C. K. I. Williams, and N. Lawrence. Fast forward selection to speed up sparse gaussian process regression. In *Ninth International Workshop on Artificial Intelligence and Statistics*. Society for Artificial Intelligence and Statistics, 2003.

[17] A. Sklar. Random variables, distribution functions, and copulas: A personal look backward and forward. *Lecture Notes-Monograph Series*, pages 1–14, 1996.

[18] E. Snelson and Z. Ghahramani. Sparse gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems*. The MIT Press, 2006.

[19] E. Snelson, C.E. Rasmussen, and Z. Ghahramani. Warped gaussian processes. In *Advances in neural information processing systems 16: proceedings of the 2003 conference*, volume 16, page 337. The MIT Press, 2004.

[20] V. Tresp. A bayesian committee machine. *Neural Computation*, 12(11):2719–2741, 2000.

[21] H. Wackernagel. *Multivariate geostatistics: an introduction with applications*. Springer-Verlag, 2nd edition, 2003.

[22] F. Wauthier and M. Jordan. Heavy-tailed process priors for selective shrinkage. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 2406–2414. 2010.

[23] A. Wilson and Z. Ghahramani. Copula processes. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 2460–2468. 2010.

[24] A. Wilson, D. A. Knowles, and Z. Ghahramani. Gaussian process regression networks. In J. Langford and J. Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning (ICML)*, Edinburgh, June 2012. Omnipress.

[25] K. Yu, V. Tresp, and A. Schwaighofer. Learning gaussian processes from multiple tasks. In *Proceedings of the 22nd international conference on Machine learning*, pages 1012–1019. ACM, 2005.