Multi-Task Learning with Task Relations

Zhao Xu and Kristian Kersting
Fraunhofer IAIS
Sankt Augustin, Germany
{firstname.lastname}@iais.fraunhofer.de

***Abstract*—Multi-task and relational learning with Gaussian processes are two active but also orthogonal areas of research. So far, there has been few attempt at exploring relational information within multi-task Gaussian processes. While existing relational Gaussian process methods have focused on relations among entities and in turn could be employed within an individual task, we develop a class of Gaussian process models which incorporates relational information across multiple tasks. As we will show, inference and learning within the resulting class of models, called relational multi-task Gaussian processes, can be realized via a variational EM algorithm. Experimental results on synthetic and real-world datasets verify the usefulness of this approach: The observed relational knowledge at the level of tasks can indeed reveal additional pairwise correlations between tasks of interest and, in turn, improve prediction performance.**

***Keywords*-Relational Learning; Multi-task Learning; Link-based Analysis; Nonparametric Bayesian Models**

I. Introduction

Multi-task learning, see e.g. [4], [7], [12], [1], [2], is a natural and widely successful setting to collaboratively solve a set of related learning tasks. Intuitively, by learning a joint model for all tasks, the method has to fit the observed data from all tasks simultaneously. This allows for information between different tasks to be shared and can significantly alleviate problems associated with sparse training data such as overfitting and unstable search and in turn often result in significant performance gains compared to single task models. For instance, within preference elicitation, an important subtask of many recommendation systems, we can share information from modeling the preferences of different users, i.e., learning the preference of one user is viewed as a task.

This sharing of information between related tasks is close in spirit to statistical relational learning, see e.g [9], [14], that combines aspects of relational logic and statistical reasoning and learning. For example, when eliciting preferences of users, it is often helpful to consider the social network among them. If two users A and B are friends, their preferences are likely to be similar. They would give similar ratings on an item more likely than users without social relations. Known preference of a user can provide useful information about unknown preferences of related users. Thus the observable relations also allow for information between entities (users) to be shared and to improve performance.

Despite these commonalities, there are some fundamental differences among existing multi-task and relational learning approaches. Most existing multi-task learning methods leverage common *latent* relations between *high-level* entities, namely tasks, whereas relational learning methods focus on the *observed* relations between *low-level* entities such as items. Consequently, it is natural to ask the questions

- How do we explicitly employ observed relations among tasks within multi-task learning?
- Can observed relational knowledge reveal additional correlations between tasks of interest?

An investigation of these questions was the seed that grew into our main contribution: relational multi-task Gaussian processes (RMTGPs).

Specifically, relational multi-task Gaussian processes introduce for each task one latent function f^q, on which we condition the observations. These latent functions are drawn from a common Gaussian process prior for all tasks. However, in contrast to multi-task Gaussian processes (MTGPs), the latent functions in RMTGPs are not independent of each other given the shared prior but coupled by the observable task relations. Technically, we introduce an additional random variable $r_{q,q'}$ representing a relation between tasks q and q'. Its value (true / false) is conditioned on the latent functions f^q and $f^{q'}$. Intuitively, tasks with a true relation between them will have more similar latent functions than those false relations. Viewing functions as nodes in a graphical model, this establishes a network of inter-linked tasks so that information of individual tasks can propagate through the whole network. Consequently, RMTGPs not only leverage common *latent* relations across tasks, but also make use of *observed* relations between tasks to reveal additional correlations between the tasks of interest[1].

The rest of the paper is organized as follows. We start off by touching upon related work in Sec. II. Then we introduce the relational multi-task models in Sec. III and Sec. IV. We develop the corresponding inference, learning, and (transductive) prediction methods in Sec. V. Before concluding, we present our experimental evaluation in Sec. VI.

[1] For the sake of simplicity, we do not model relations among low-level entities. This can easily be accomplished by applying relational hierarchical Gaussian process framework to relational GPs instead of GPs.

II. Related Work

The relational multi-task Gaussian process model unifies two lines of research within the Gaussian process community, namely relational and multi-task learning.

Statistical relational learning, see e.g. [9], [14] for overviews, investigates how to employ relations among entities within probabilistic models. This is mainly motivated by the growing need in analyzing data that is best represented as a graph, such as the World Wide Web, social networks, social media, biological networks, communication networks, and physical network systems. To incorporate links and relations into probabilistic kernel methods such as Gaussian processes (GPs), there are essentially two approaches. One is encoding relations as random variables conditioned on the latent function values of entities involved in relations [8], [28], [26]. The other is to encode relations in the covariance matrixes [32], [23]. Intuitively, this represents relational information as hidden common causes and condition the outputs on the relations.

Multi-task learning [24], [7], [3], [1], [31] is to collaboratively learn a set of related tasks such that prediction about one query can leverage information from all other queries. Lawrence and Platt [19] first leveraged Gaussian process in multi-task learning, where the covariance matrix is task-specific block diagonal structure, the covariances between different tasks are zero. Yu *et al.* [29] introduced a hierarchical GP model, where the information is transferred by a “informed prior” that is learned from the individual tasks. One recent extension is proposed by Birlutiu *et al.* [5]. Additionally, Bonilla *et al.* [6] introduced a GP model for multi-task learning, which generalized over task attributes with a Kronecker product based method. Most recently, Deshpande *et al.* [10] and Landwehr *et al.* [18] considered multi-task learning in relational domains. In contrast to the present paper, they employed relations within tasks and not across tasks. Besides Gaussian processes, other techniques are employed for multi-task learning, e.g., the task clustering approaches [25], [3], [27], the regularization methods [12], [2], and neural networks [7]. Several relational approaches have been explored. For example, Evgeniou *et al.* proposed a regularization methods for multi-task learning with task relations [11]. Kato *et al.* introduced a similar method with a different penalization term [17]. Sheldon extended their works to incorporate non-linear kernels [22]. In contrast to the regularization-based methods, this paper proposes a novel approach which models the prediction uncertainty in multi-task learning and incorporates the observable task relations into a flexible probabilistic model. The method encodes both latent and observed relations between tasks within a relational GP framework. The knowledge propagation between tasks is based on the relational likelihood distributions. The proposed framework can also naturally be applied to model multi-relational tasks.

III. Relational Linear Multi-task Models

Let us start with a simple linear regression model for the relational multi-task learning, then extend it to a more flexible Gaussian process model. Assume that there are (1) a set of n items $E = \{e_1, \dots, e_n\}$ with attributes $X = \{x_i : x_i \in \mathbb{R}^D, i = 1, \dots, n\}$, and (2) a set of Q tasks with relations $R = \{r_{q,q'} : q, q' \in 1, \dots, Q\}$, as well as (3) real-valued observations on the items for each task q, $y^q = \{y_i^q : y_i^q \in \mathbb{R}, i = 1, \dots, n\}$.

A. Linear Model for Single-task Learning

In Bayesian analysis, a linear regression model for a single task can be represented as:

$$\begin{aligned} &\omega|\mu, K \sim \mathcal{N}(\mu, K) \\ &f(x_i) = x_i^T \omega, \quad i = 1, \dots, n \\ &y_i|f(x_i), \sigma^2 \sim \mathcal{N}(f(x_i), \sigma^2). \end{aligned} \tag{1}$$

The observations y_i are modeled as noisy linear combination of attributes with a D-dimensional weight ω, which is drawn from a Gaussian with mean μ and covariance matrix K. $f(x_i)$ denotes the true value of the data point i. σ^2 is variance of the noise.

B. Linear Model for Multi-task Learning

In multi-task learning, there is an underlying assumption that the tasks are distinct, but related with each other. One task can borrow strength from the information extracted from another tasks. In Bayesian framework, it can be modeled with hierarchical method:

$$\begin{aligned} &\omega^q|\mu, K \sim \mathcal{N}(\mu, K), \quad q = 1, \dots, Q \\ &f^q(x_i) = x_i^T \omega^q, \quad i = 1, \dots, n \\ &y_i^q|f^q(x_i), \sigma^2 \sim \mathcal{N}(f^q(x_i), \sigma^2). \end{aligned} \tag{2}$$

There is one distinct weight vector ω^q for each task q, but they are drawn from a common prior $\mathcal{N}(\mu, K)$. The shared prior parameters μ and K model the common properties in the different tasks. In the hierarchical Bayesian framework, each task is distinguished via personalized parameters, but closely connected with the common prior.

C. Linear Model for Relational Multi-task Learning

Most existing multi-task learning approaches do not consider observable relations among tasks. However, link and relational information in general have been proved to be a promising way to improve performance. For example, in a social media website, songs are entities, users are tasks. Ratings of users on songs are observations. Additionally we have the social relations between users (tasks). Users being friends would give similar rating values on a song more likely than users with no social relations. Knowing such kinds of relational information will reduce the uncertainty on learning and prediction. To explicitly incorporate the information into the hierarchical linear model, we introduce

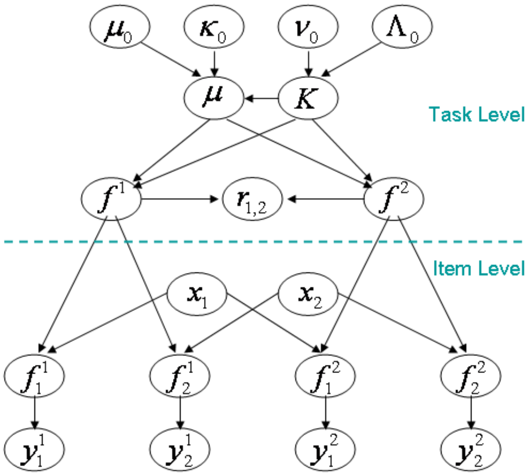


Figure 1. The RMTGP model for a simple example with two tasks and two items. f^1 and f^2 are *random functions* drawn from a GP prior, one for each task. The mean and covariance matrix of the GP is drawn from a Normal-Inverse-Wishart distribution with parameters ν_0, κ_0, μ_0 and Λ_0. The two functions are coupled with the relation $r_{1,2}$ between tasks. x_1 and x_2 are attributes of the two items.

an additional random variable $r_{q,q'}$ for each relation between tasks q and q'. The variable $r_{q,q'}$ is conditioned on ω^q and $\omega^{q'}$ with the probability

$$P(r_{q,q'}|\omega^q, \omega^{q'}, \lambda) = \exp(-\lambda(\omega^q - \omega^{q'})^T(\omega^q - \omega^{q'})), \quad (3)$$

where $\lambda > 0$ is the rate parameter. The probability naturally captures the property in relational multi-task learning. The less the difference between weight vectors of the two tasks, the more likely there are relations between them. In the social media example, the weight vector ω^q represents intrinsic preference of a user on songs. If two users are friends, then they likely have similar preferences (i.e. weights ω^q), and vice versa. The complete linear multi-task model with observable task relations is defined as

$$\begin{aligned} &\omega^q|\mu, K \sim \mathcal{N}(\mu, K), \quad q = 1, \ldots, Q \\ &r_{q,q'}|\omega^q, \omega^{q'}, \lambda \sim \exp(-\lambda(\omega^q - \omega^{q'})^T(\omega^q - \omega^{q'})) \\ &\qquad\qquad q, q' = 1, \ldots, Q \\ &f^q(x_i) = x_i^T \omega^q, \quad i = 1, \ldots, n \\ &y_i^q|f^q(x_i), \sigma^2 \sim \mathcal{N}(f^q(x_i), \sigma^2). \end{aligned} \quad (4)$$

IV. Relational Non-linear Multi-task Models with Gaussian Processes

So far, we have assumed that the latent functions are linear. Each function is characterized by its weight vector. The parameterization of functions limits the flexibility of the model, since the mathematic form of functions is not necessarily linear, there exists uncertainty. To overcome the limitation, we extend the linear case with nonparametric techniques, and in turn offer the relational multi-task Gaussian process model (RMTGP).

The RMTGP model is graphically represented in Fig. 1. For each task, we introduce a *random function* f^q, and assume that all functions are drawn from a Gaussian process (GP), which is a distribution over functions. Additionally the functions are not independent of each other given the GP prior, instead, they are linked together due to some relations. The tasks having relations between them will likely have similar functions. To model these dependencies, we again introduce for each relation $r_{q,q'}$ an additional random variable with the conditional likelihood $P(r_{q,q'}|f^q, f^{q'})$. Intuitively, the more similar functions f^q and $f^{q'}$ are, the more likely there is a relation between the tasks q and q'. Let us now introduce the prior distributions, the likelihood function for task relations, and the generative process of the RMTGP model.

A. Prior Distributions

We define a GP prior over the attribute-wise latent functions, shared by all tasks. Specifically, for a random function $f^q(\cdot)$ of a task q, the function values $\{f^q(x_1), f^q(x_2), \ldots\}$ at an infinite number of data points can be represented as an infinite dimensional vector, i.e., the i'th dimension is the function value $f^q(x_i)$ (shortened as f_i^q). We assume that the infinite dimensional random vector follows a Gaussian process prior with mean function $m(x_i)$ and covariance function $k(x_i, x_j)$. In turn, any finite set of function values $\{f_i^q : i = 1, \ldots, n\}$ has a multivariate Gaussian distribution with mean μ and covariance matrix K defined in terms of the mean and covariance functions of the GP [20]. Formally, the prior distribution over functions of item attributes is defined as follows:

$$\begin{aligned} P(f^q|\mu, K) &= \mathcal{N}(\mu, K) \\ &= \frac{1}{(2\pi)^n |K|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(f^q - \mu)^T K^{-1}(f^q - \mu)\right), \end{aligned} \quad (5)$$

where f^q denotes the function values $(f_1^q, \ldots, f_n^q)$ of the items for the task q. K is the $n \times n$ covariance matrix.

One is tempted to use some covariance function with a finite number of hyperparameters to compute K. However the parameterized kernel function limits the flexibility of the model as e.g. Yu *et al.* pointed out [29]. Therefore, we assume that the prior parameters μ and K are directly drawn from a conjugate hyperprior, i.e. Normal-Inverse-Wishart (NIW) distribution [13]:

$$\begin{aligned} P(\mu, K|\kappa_0, \nu_0, \mu_0, \Lambda_0) &\propto |K|^{-\frac{\nu_0+n+2}{2}} \exp(v), \\ v = -\frac{1}{2} tr(\Lambda_0 K^{-1}) &- \frac{\kappa_0}{2}(\mu - \mu_0)^T K^{-1}(\mu - \mu_0). \end{aligned} \quad (6)$$

The parameters $\nu_0 > n$ and $\kappa_0 > 0$ are the degrees of freedom for K and the number of prior measurements for μ. Λ_0 represents our prior belief on the covariance matrix before seeing any observations $\{y^q\}$. For that we can use any Mercer kernel function. A typical choice is the squared exponential covariance function with isotropic distance measure:

$$k(x_i, x_j) = \tau^2 \exp(-\frac{\rho^2}{2} \sum\nolimits_d^D (x_{i,d} - x_{j,d})^2) \tag{7}$$

where τ and ρ are parameters of the covariance function, and $x_{i,d}$ denotes the d-th dimension of the attribute vector x_i. Now μ and K are samples of a NIW distribution, they are flexible enough to reflect any possible covariance structure shared among all tasks. By the two parameters μ and K, the latent correlations between tasks are represented in an elegant way.

B. Likelihood for Task Relations

We now define the likelihood distribution for the observable task relations. As already argued earlier, the related tasks generally show similar intrinsic properties, i.e., their functions are likely to be close to each other. We extend the likelihood definition in linear model, and have

$$P(r_{q,q'}|\lambda, f^q, f^{q'}) = \exp(-\lambda d(f^q, f^{q'})), \tag{8}$$

where $d(f^q, f^{q'})$ denotes the distance between two functions. This encodes the natural assumption: The less the difference between the functions f^q and $f^{q'}$, the more likely is it that the task q is related/linked to the task q'. There are many choices for the distance function used. The typical ones include the L_1 and L_2 norm. In this paper, we leverage the L_1 norm:

$$\begin{aligned} d(f^q, f^{q'}) &= ||f^q - f^{q'}||_1 \\ &= \sum_i |f_i^q - f_i^{q'}|. \end{aligned} \tag{9}$$

C. The Generative Model

Finally we complete the relational multi-task GP model with the generative procedure. Give the hyperparameters $\theta = \{\kappa_0, \nu_0, \mu_0, \Lambda_0, \sigma^2, \lambda\}$, the data is generated as follows:

$$\begin{aligned} &K|\nu_0, \Lambda_0 \sim IW_{\nu_0}(\Lambda_0^{-1}), \\ &\mu|\mu_0, K, \kappa_0 \sim \mathcal{N}(\mu_0, \frac{1}{\kappa_0} K) \\ &f^q|\mu, K \sim \mathcal{N}(\mu, K), \quad q = 1, \dots, Q \\ &y_i^q|f_i^q, \sigma^2 \sim \mathcal{N}(f_i^q, \sigma^2), \quad i = 1, \dots, n \\ &r_{q,q'}|\lambda, f^q, f^{q'} \sim \exp(-\lambda||f^q - f^{q'}||_1). \end{aligned} \tag{10}$$

Where $IW_{\nu_0}(\Lambda_0)$ denotes the Inv-Wishart distribution with freedom degree ν_0. Generating a matrix from the Inv-Wishart distribution amounts to the following two steps:

- Sampling ν_0 vectors from a Gaussian distribution

$$\alpha_t|\Lambda_0 \sim \mathcal{N}(0, \Lambda_0), \text{ for } t = 1, \dots, \nu_0$$

- Compute the covariance matrix with

$$K = \left(\sum_{t=1}^{\nu_0} \alpha_t \alpha_t^T \right)^{-1}.$$

V. Inference and Learning

The key inferential problem in the relational nonparametric hierarchical model is to compute the joint posterior distribution of the unknown variables given the multi-task observations $Y = \{y^q\}_{q=1}^Q$ and the relations between tasks $R = \{r_{q,q'}\}_{q,q'=1}^Q$, as well as the entity attributes $X = \{x_i\}_{i=1}^n$. The unknown variables in the model are the latent functions $f = \{f^q\}_{q=1}^Q$, one for each task. Thus the posterior is proportional to:

$$\begin{aligned} P(f|Y, R, X, \theta) \propto\; & P(K|X, \nu_0, \Lambda_0^{-1})\; P(\mu|\mu_0, K, \kappa_0) \\ & \times \prod_q P(f^q|\mu, K) P(Y^q|f^q, \sigma^2 I) \\ & \times \prod_{q,q'} P(r_{q,q'}|f^q, f^{q'}) \end{aligned} \tag{11}$$

where θ denotes hyperparameters of the model, $\theta = (\kappa_0, \nu_0, \Lambda_0^{-1}, \mu_0, \lambda, \sigma^2)$. It is clear that the equation is intractable, since unknown functions are coupled together by the task relations R. To address the problem, we consider to decouple the dependencies with a variational approximation algorithm [15]. In particular, we expect to find a variational distribution $\hat{P}(f)$ to approximate the true posterior $P(f|Y, R, X, \theta)$ as close as possible. For computational efficiency, a family of fully-factorized distributions are assumed,

$$\hat{P}(f^1, \dots, f^Q) = \prod_{q=1}^{Q} \hat{P}(f^q), \tag{12}$$

and for each f^q, the variational distribution is defined as a Gaussian:

$$\hat{P}(f^q) = \mathcal{N}(f^q|\hat{\mu}^q, \hat{K}^q), \tag{13}$$

where $\hat{\mu}^q$ and $\hat{K}^q$ are the mean and covariance matrix of the variational distribution. The difference between the variational distribution and the true posterior distribution is measured via *Kullback-Leibler* (KL) divergence, i.e.,

$$\begin{aligned} KL(\hat{P}||P) = \mathbb{E}_{\hat{P}}[\log \hat{P}(f)] - \mathbb{E}_{\hat{P}}[\log P(Y, R, f|X, \theta)] \\ + \log P(Y, R|\theta). \end{aligned}$$

Permuting the equation, we get an inequality about the log-likelihood of the data:

$$\begin{aligned} &\log P(Y, R|\theta) \\ &\quad = \mathbb{E}_{\hat{P}}[\log P(Y, R, f|X, \theta)] - \mathbb{E}_{\hat{P}}[\log \hat{P}(f)] + KL(\hat{P}||P) \\ &\quad \geq \mathbb{E}_{\hat{P}}[\log P(Y, R, f|X, \theta)] - \mathbb{E}_{\hat{P}}[\log \hat{P}(f)]. \end{aligned} \tag{14}$$

The right terms define a *lower bound* of the log-likelihood of the data. It can also be derived from *Jensen's inequality*. The difference between the lower bound and the log-likelihood is the KL divergence. The larger the lower bound is, the smaller the divergence is, and the closer the variational distribution approximates the true posterior. Thus the probabilistic inference problem is now converted into an optimization problem: maximize the lower bound of the log-likelihood with respect to the variational parameters. In details, the lower bound is written as:

$$\begin{aligned}\mathcal{L} = &\sum_{q,q'} \mathbb{E}_{\hat{P}}[\log P(r_{q,q'}|f^q, f^{q'}, \lambda)] \\ &+ \sum_{q} \mathbb{E}_{\hat{P}}[\log P(y^q|f^q, \sigma^2)] \\ &+ \sum_{q} \mathbb{E}_{\hat{P}}[\log P(f^q|\mu, K)] \\ &+ \log P(\mu, K|X, \nu_0, \kappa_0, \mu_0, \Lambda_0^{-1}) \\ &- \sum_{q} \mathbb{E}_{\hat{P}}[\log \hat{P}(f^q)]. \end{aligned} \tag{15}$$

The first two terms are about the expectations of likelihoods: task relations and observations per tasks. The third term is related to the expectation of the prior. The fourth term is about the mean and covariance matrix of the prior. The last term is the entropy of the variational distribution. Since the variational distributions $\hat{P}(f^q)$ are factorized and defined as Gaussian, the computation of (15) is relatively straightforward, and we have

$$\begin{aligned}&\mathbb{E}_{\hat{P}}[\log P(r_{q,q'}|f^q, f^{q'}, \lambda)] \\ &\quad = -\lambda\left[tr(\hat{K}^q + \hat{K}^q) + (\hat{\mu}^q - \hat{\mu}^{q'})^T(\hat{\mu}^q - \hat{\mu}^{q'})\right] \\ &\mathbb{E}_{\hat{P}}[\log P(y^q|f^q, \sigma^2)] \\ &\quad = -\frac{1}{2}\left[n \log \sigma^2 + tr(\hat{K}^q) + (y^q - \hat{\mu}^q)^T(y^q - \hat{\mu}^q)\right] \\ &\mathbb{E}_{\hat{P}}[\log P(f^q|\mu, K)] \\ &\quad = -\frac{1}{2}\left[\log det(K) + tr(K^{-1}\hat{K}^q)\right] \\ &\qquad - \frac{1}{2}(\hat{\mu}^q - \mu)^T(\hat{\mu}^q - \mu) \\ &\mathbb{E}_{\hat{P}}[\log \hat{P}(f^q)] \\ &\quad = -\frac{1}{2}\log det(\hat{K}^q) \\ &\log P(\mu, K|X, \nu_0, \kappa_0, \mu_0, \Lambda_0^{-1}) \\ &\quad = -\frac{1}{2}[(\nu_0 + n + 2)\log det(K) + tr(\Lambda_0 K^{-1}) \\ &\qquad + \kappa_0(\mu - \mu_0)^T K^{-1}(\mu - \mu_0)]. \end{aligned} \tag{16}$$

Where $tr(\cdot)$ and $det(\cdot)$ denote the trace and determinant of a matrix respectively. Note that the constant terms (e.g. $\log 2\pi$) do not appear in the equations.

Now we learn the variational parameters $\{\hat{\mu}^q, \hat{K}^q\}$ by maximizing the lower bound. Since the prior mean μ and covariance matrix K are unknown as well, we leverage variational EM algorithm to learn the two sets of parameters together. In the E-step, we maximize the lower bound with respect to the variational parameters $\{\hat{\mu}^q, \hat{K}^q\}$. The step actually optimizes the variational distribution to approximate the true posterior distribution given the current prior parameters. In the M-step, we maximize the lower bound with respect to the prior parameters μ and K. In each step, we use coordinate ascent to solve the optimization problem. Specifically, we take the derivative with respect to $\hat{\mu}^q$ (resp. $\hat{K}^q$, μ, and K), set it to zero, and solve the equation, then get the update formula for E- and M-steps. The partial derivative equations are defined as follows

$$\begin{aligned}0 = \frac{\partial \mathcal{L}}{\partial \hat{\mu}^q} &= K^{-1}(\mu - \hat{\mu}^q) + \frac{1}{\sigma^2}(y^q - \hat{\mu}^q) \\ &\quad - 2\lambda \sum_{q'}(\hat{\mu}^q - \hat{\mu}^{q'}) \\ 0 = \frac{\partial \mathcal{L}}{\partial \hat{K}^q} &= -\frac{1}{2}K^{-1} - \frac{1}{2\sigma^2}I - \lambda Q^q I + \frac{1}{2}(\hat{K}^q)^{-1} \\ 0 = \frac{\partial \mathcal{L}}{\partial \mu} &= \sum_{q=1}^{Q} K^{-1}(\hat{\mu}^q - \mu) + \kappa_0 K^{-1}(\mu_0 - \mu) \\ 0 = \frac{\partial \mathcal{L}}{\partial K} &= \frac{1}{2}\sum_{q=1}^{Q} K^{-1}[\hat{K}^q + (\hat{\mu}^q - \mu)(\hat{\mu}^q - \mu)^T]K^{-1} \\ &\quad - \frac{Q}{2}K^{-1} - \frac{\nu_0 + n + 2}{2}K^{-1} + \frac{1}{2}K^{-1}\Lambda_0 K^{-1} \\ &\quad + \frac{\kappa_0}{2}K^{-1}(\mu - \mu_0)(\mu - \mu_0)^T K^{-1} \\ 0 = \frac{\partial \mathcal{L}}{\partial \sigma^2} &= \frac{1}{2\sigma^4}\sum_{q=1}^{Q}[tr(\hat{K}^q) + (y^q - \hat{\mu}^q)^T(y^q - \hat{\mu}^q)] - \frac{nQ}{2\sigma^2}. \end{aligned} \tag{17}$$

Where q' denotes tasks which have relations with the task q. Q^q is the number of related tasks of q.

Putting everything together, the variational EM method consists of:

- **E step:**

$$\begin{aligned}\hat{\mu}^q &= (I + (\sigma^{-2} + 2\lambda Q^q)K)^{-1} \\ &\quad \times (\mu + \sigma^{-2}Ky^q + 2\lambda K \sum_{q'} \hat{\mu}^{q'}), \\ \hat{K}^q &= (K^{-1} + (\sigma^{-2} + 2\lambda Q^q)I)^{-1}. \end{aligned}$$

- **M step:**

$$\begin{aligned}\mu &= a\,\mu_0 + \sum_q b\,\hat{\mu}^q, \\ K &= c\,[\Lambda_0 + \kappa_0(\mu - \mu_0)(\mu - \mu_0)^T \\ &\quad + \sum_q \hat{K}^q + (\hat{\mu}^q - \mu)(\hat{\mu}^q - \mu)^T], \\ \sigma^2 &= d\sum_q tr(\hat{K}^q) + (y^q - \hat{\mu}^q)^T(y^q - \hat{\mu}^q) \end{aligned}$$

where a,b,c and d are coefficients: $a = \kappa_0/(Q + \kappa_0)$, $b = 1/(Q + \kappa_0)$, $c = 1/(Q + \nu_0 + n + 2)$, and $d = 1/nQ$.

Iteratively run the E- and M-steps until convergence, then we can get the approximate posterior distribution $\mathcal{N}(\hat{\mu}^q, \hat{K}^q)$ for each task q and the common prior $\mathcal{N}(\mu, K)$ shared by all tasks. The convergence can be monitored by tracing the difference of the optimized parameters (e.g. μ and K) between two iterations. The procedure can be initialized with $\mu = 0$ and $K = \Lambda_0$.

From the E step, we can obviously observe how the tasks are mixed by observed task relations. Updating the variational mean $\hat{\mu}^q$ for the task q takes into account three components:

1) the prior information $K^{-1}\mu$,
2) the observations within the task $1/\sigma^2 y^q$,
3) the variational means $\hat{\mu}^{q'}$ of the tasks having relations with q.

In updating variational covariance matrix $\hat{K}^q$, the term $\lambda Q^q I$ comes from the related tasks as well. The M step is similar to that in non-relational multi-task learning. It is reasonable, since the prior parameters (μ and K) are independent of task relations given the latent functions ($\hat{\mu}^q$ and $\hat{K}^q$).

A. Transductive Prediction

The prediction inference in multi-task learning is to predict values of unobserved entities. In this section we consider the predictive inference in a transductive setting, i.e. there is no new entity introduced in prediction. It can also be viewed as a learning problem with missing values [30]. In particular, we still use the variational EM method to address the problem, but the E step is modified such that it learns not only the variational parameters, but also the expectation of unseen values for each task. Let $\mathbb{I}$ and $\mathbb{U}$ denote the indexes of entities with and without observations. Then the new E step is as follows:

$$\begin{aligned}\hat{\mu}^q &= (I + \frac{1}{\sigma^2}K_{\mathbb{I}} + 2\lambda Q^q K)^{-1} \\ &\quad \times (\mu + \frac{1}{\sigma^2}K_{\mathbb{I}} y^q_{\mathbb{I}} + 2\lambda K \sum_{q'} \hat{\mu}^{q'}), \qquad (18)\end{aligned}$$

$$\hat{K}^q = (K^{-1} + \frac{1}{\sigma^2} I_{\mathbb{I}} + 2\lambda Q^q I)^{-1}, \qquad (19)$$

where $K_{\mathbb{I}}$ denotes the prior covariance matrix K, but only keeping the columns $\mathbb{I}$, all others being zeros. In comparison with the E step in the last section, updating variational mean and covariance matrix for each task will only consider the data points having observations. $\hat{\mu}^q_{\mathbb{U}}$ will be the expectation of the missing values, which computation depends on the observations within the same tasks, and the observations from the relational tasks.

In the M step, the update of μ and K remain unchanged, since the two parameters are independent of observations given the latent functions. Only σ^2 is updated differently:

$$\sigma^2 = \sum_q \frac{1}{n^q} \left[tr(\hat{K}^q_{\mathbb{I}}) + (y^q_{\mathbb{I}} - \hat{\mu}^q_{\mathbb{I}})^T (y^q_{\mathbb{I}} - \hat{\mu}^q_{\mathbb{I}}) \right], \qquad (20)$$

where n^q is the number of observations in the task q. σ^2 is the “variance” averaged over all observed data points.

Here we introduce transductive prediction, the proposed model can also work on inductive setting. In the inductive setting, the covariance between the new entities and the known ones can be computed via the nyström method [21] or similarity matching [33].

VI. Empirical Analysis

Our intention in the empirical analysis is to investigate the following questions:

- **(Q1)** Does RMTGP perform better than single task GPs and multi-task Gaussian processes (MTGP) without relations?
- **(Q2)** Is its performance more stable for smaller number of observed training examples?
- **(Q3)** How does its performance depend on the informativeness of the task relations provided?

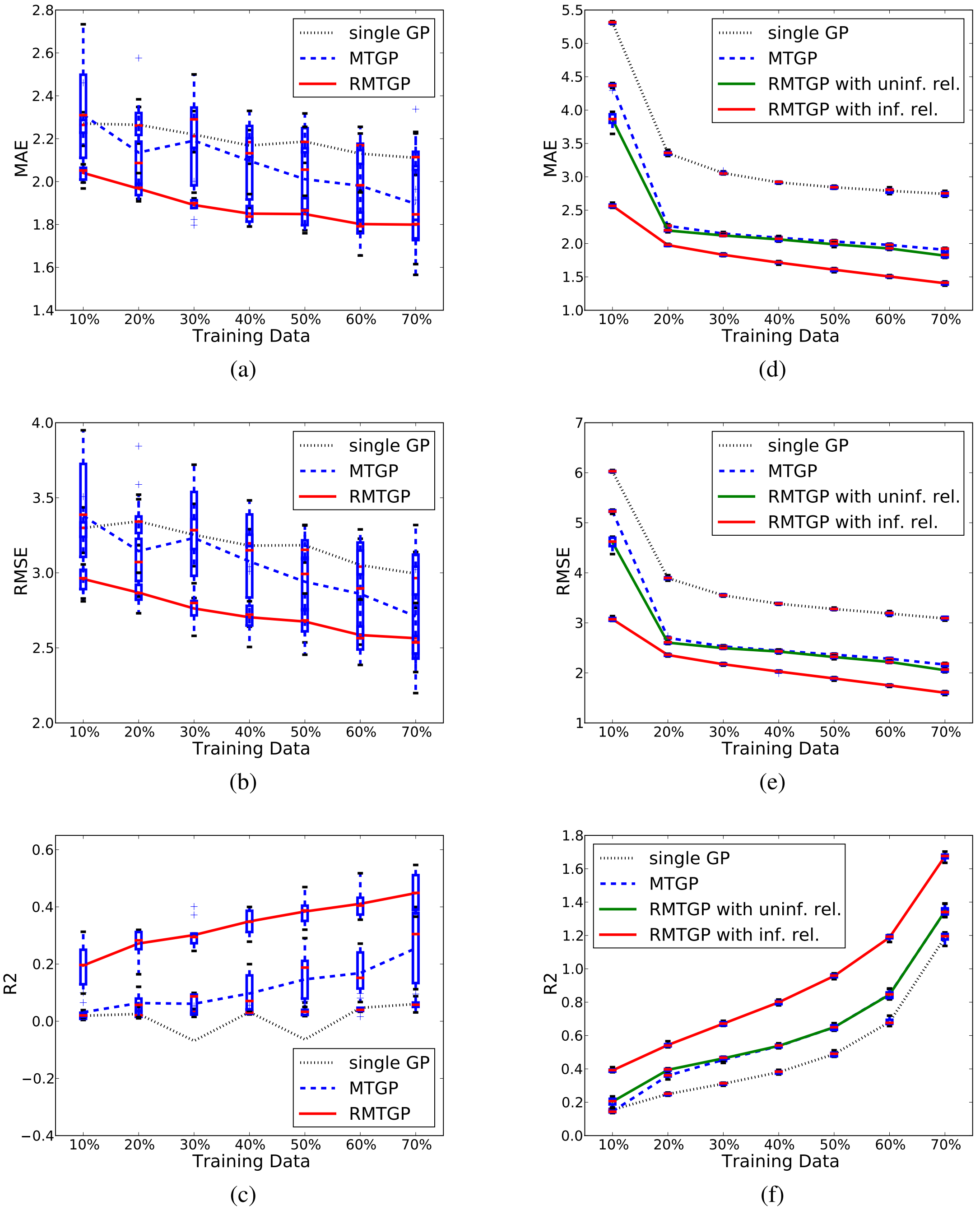
To this aim, we implemented RMTGPs as well as single-task GPs and MTGPs within Python and evaluated them on two datasets, a synthetic dataset and Kamishima’s Sushi data [16]. For (R)MTGP, we conducted the experiments within a transductive setting and measured performance using three commonly used metrics:

- the mean absolute error
$$MAE = \frac{1}{n}\sum_i |y_i - f_i|,$$
- the root mean squared error
$$RMSE = \sqrt{\frac{1}{n}\sum_i (y_i - f_i)^2},$$
- the coefficient of determination
$$R^2 = \left(\frac{\sum_i (y_i - \bar{y})(f_i - \bar{f})}{(n-1)\sigma_y \sigma_f} \right)^2.$$

Where MAE and RMSE measure the difference between predicted and real values, i.e., the smaller, the better, the coefficient of determination R^2 measures the generalization performance of a model, i.e., the larger, the better. For all experiments, we randomly selected 10% (20%,...,70%) observations of each task for training and the rest for testing. For each setting (10%,...,70%), the selection was repeated 10 times to get the average performance.

A. Data Description

The synthetic dataset is generated using the generative procedure described earlier. That is, we uniformly sample $n = 100$ data points with 1-dimensional attributes $x_i \in (-15, +15)$. The hyperparameters μ_0 are computed assuming $\mu_0 = \cos(x)$ and Λ_0 using a squared exponential kernel. Then, we draw μ and K from the NIW distribution with parameters μ_0, Λ_0, $\nu_0 = n + 10$, and $\kappa_0 = 2$.



(a)

(d)

(b)

(e)

(c)

(f)

Figure 2. Experimental results on the synthetic (left) and Sushi (right) datasets averaged over 10 random reruns, each on predicting unseen values given different percentages of observations. MAE (top) and RMSE (middle); the lower the two measures, the better the performance. R^2 (bottom); the larger, the better. The results show that exploiting relations among tasks can reveal additional correlations and in turn improve the prediction performance.

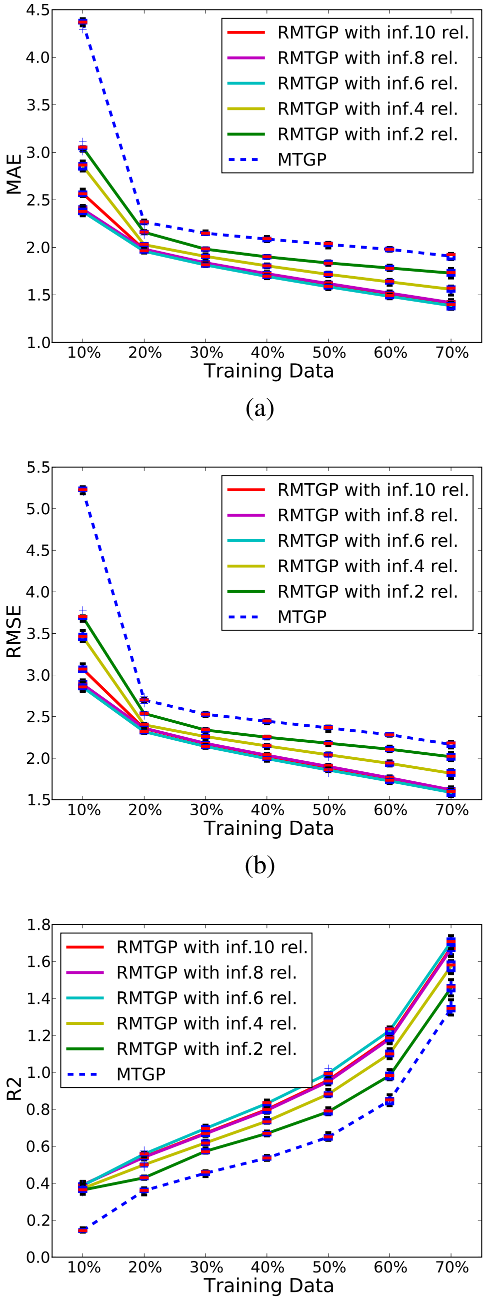
Next, 10 functions are sampled from $\mathcal{N}(\mu, K)$, and we add some Gaussian noise with $\sigma^2 = 0.01$. Finally, the relations between tasks are sampled based on the L_1 norm.

The Sushi dataset was collected by Kamishima [16]. It is about preferences of users on 10 different sushi variants: ebi, anago, maguro, ika, uni, sake, tamago, toro, tekka-maki, and kappa-maki. Each sushi type is described in terms of the following attributes: style, major group, minor group, heaviness/oiliness, consumption frequency, normalized price, and sell frequency. In total, there are 5000 users, and we randomly select 1000 users for our experiments. Each user provides a full ordering of the ten sushi types according to her preference. The ratings range from 1 to 10 where the most preferred sushi gets the rating 10. Additionally, each user is described in terms of attributes: gender, age, and others that compile regional information. As there are no task relations (i.e. relations between users) in the sushi dataset, we computed artificial relations with two methods. (1) We established *informative* relations using the averaged L_1 norm of user ratings thresholded at 0.3. (2) We computed *un-informative* relations using the cosine similarities of user attributes thresholded at 0.8 (i.e. there is a relation between two users only if they have the same age, gender, living region in early life and living region currently). The latter relations are un-informative since users with the same attributes do not necessarily have the same preferences on sushi. We further note that, although informed relations are based on ratings, they do not provide much information about the test set since the resulting relations (exist/non-exist) only compile whether the preferences of users are similar or not, no more information. In turn, it is still sensible to compare performance between RMTGP and (MT)GP. Moreover, informative relations are not transitive: u_1 and u_2 respectively u_2 and u_3 linked does not imply u_1 and u_3 linked.

B. Experimental Results

Fig. 2 (left) summarizes the results on the synthetic dataset. As one can see, in all cases (different percentages of known observations for each task), RMTGPs outperform non-relational MTGPs and single-task GPs. A Wilcoxon rank sum test (p-value 0.01) shows that this difference is significant. Moreover, we see that the RMTGP model performs particularly well when the number of known observations is small. Overall, the RMTGP improved prediction performance between 5.10% and 11.72%. Furthermore, the variability of RMTGP's performance across multiple tasks is substantially smaller than for non-relational MTGP.

The experimental results on the Sushi dataset are shown in Fig. 2 (right). When the relations between users are informative, RMTGPs outperform non-relational MTGPs and single-task GPs, especially for low percentages of known observations. Furthermore, RMTGPs also achieve good performance if the relations are not informative. The results



(a)

(b)

(c)

Figure 3. Experimental results about the impact of relations being informative on the overall performance. The analysis is conducted with the Sushi datasets where task relations are computed with the ratings on 2 (4, 6, 8, 10) sushi types. MAE (top) and RMSE (middle); the lower the two measures, the better the performance. R^2 (bottom); the larger, the better.

are very similar but still slightly better than non-relational MTGPs and substantially better than single-task GPs. What is the reason that RMTGPs did not substantially outperform MTGP when considering less informative relations? We find that the L_1 norm of the preference differences averaged over all users is 0.5709. The norm averaged over users having the un-informative relations only is 0.5660. So, there is a difference but not a significant one. In other words, users linked with such relations do not necessarily have similar sushi preferences. Therefore, the benefit of employing these relations within RMTGPs is low and RMTGP essentially coincides with MTGP without task relations. In other words, RMTGPs are a proper generalization of MTGPs and automatically balance between explicit task relations and latent dependencies across tasks learned from observations: if the task relations are uninformative, RMTGP employs just the latent dependencies; if the task relations are informative, RMTGP employs the revealed additional correlations to improve the predictive performance.

To further investigate the impact of relations being informative on the overall performance, we ran the same experiment with relations of different informativeness levels, namely relations computed using the ratings on the first 2 (4, 6, 8, 10) sushi types only. Fig. 3 summarizes the experimental results. In all cases RMTGPs perform significantly better than MTGPs. Already using the L1 norm on the first two of the ten sushi preferences to establish task relations results in significantly better performance.

To summarize, the experimental results clearly show that all questions **(Q1)**-**(Q3)** can be answered affirmatively: exploiting relations among tasks can reveal additional correlations and in turn improve the prediction performance.

VII. Conclusion

Most existing multi-task Gaussian process approaches treat all tasks a priori the same: fully related or totally irrelevant, the Gaussian process has to figure this out by learning. In reality, however, observable relations among tasks fall everywhere along this spectrum and we may know this a priori. In this work, we have shown how to incorporate relations among tasks within a Gaussian process framework. The observed relations among tasks reveal additional correlations that in turn can result in performance gains. Specifically, we have developed a Bayesian framework to multi-task learning based on Gaussian processes that exploits observed relations among tasks. On synthetic and real-world datasets, we have shown that the resulting class of non-parametric models can yield significantly better predictive performance than single- and multi-task Gaussian processes.

While this paper has focused on regression, the proposed relational multi-task Gaussian process framework can be used for classification as well as for preference learning. Furthermore it can be generalized for directed relations as well as multiple classes of relations. These are promising avenues for future work.

Acknowledgments

This work was funded by the Fraunhofer ATTRACT Fellowship ”Statistical Relational Activity Mining” (STREAM), and by the European Commission under contract number FP7-248258-First-MM, as well as by the German Federal Ministry of Economy and Technology (BMWi) under the THESEUS project.

References

[1] R. K. Ando, T. Zhang, and P. Bartlett. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, 2005.

[2] A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In B. Schölkopf, J. Platt, and T. Hofmann, editors, *Advances in Neural Information Processing Systems 19*, pages 41–48. MIT Press, 2007.

[3] B. Bakker and T. Heskes. Task clustering and gating for bayesian multitask learning. *Journal of Machine Learning Research*, 4, 2003.

[4] J. Baxter. A bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine Learning*, pages 7–39, 1997.

[5] A. Birlutiu, P. Groot, and T. Heskes. Multi-task preference learning with an application to hearing aid personalization. *Neurocomputing*, 73(7-9):1177–1185, 2010.

[6] E. V. Bonilla, K. M. A. Chai, and C. K. I. Williams. Multi-task Gaussian process prediction. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*. MIT Press, 2008.

[7] R. Caruana. Multitask learning. *Machine Learning*, 28:41–75, 1997.

[8] W. Chu, V. Sindhwani, Z. Ghahramani, and S. Keerthi. Relational learning with gaussian processes. In B. Schölkopf, J. Platt, and T. Hofmann, editors, *Advances in Neural Information Processing Systems 19*, pages 289–296. MIT Press, 2007.

[9] L. De Raedt. *Logical and Relational Learning*. Springer, 2008.

[10] A. Deshpande, B. Milch, L. S. Zettlemoyer, and L. P. Kaelbling. Learning probabilistic relational dynamics for multiple tasks. In *Proceedings of the Twenty-Third Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-07)*, pages 83–92. AUAI Press, 2007.

[11] T. Evgeniou, C. A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 2005.

[12] T. Evgeniou and M. Pontil. Regularized multi–task learning. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '04)*, pages 109–117. ACM, 2004.

[13] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis*. CRC Press, 2004.

[14] L. Getoor and B. Taskar, editors. *An Introduction to Statistical Relational Learning*. MIT Press, 2007.

[15] M. I. Jordan, Z. Ghahramani, T. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.

[16] T. Kamishima. Nantonac collaborative filtering: recommendation based on order responses. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '03)*, pages 583–588. ACM, 2003.

[17] T. Kato, H. Kashima, M. Sugiyama, and K. Asai. Multi-task learning via conic programming. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*. MIT Press, 2008.

[18] N. Landwehr, A. Passerini, L. De Raedt, and P. Frasconi. Fast learning of relational kernels. *Machine Learning*, 78(3):305–342, 2010.

[19] N. D. Lawrence and J. C. Platt. Learning to learn with the informative vector machine. In *Proceedings of the Twenty-First International Conference on Machine learning (ICML '04)*, 2004.

[20] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[21] A. Schwaighofer, V. Tresp, and K. Yu. Learning gaussian process kernels via hierarchical bayes. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*. MIT Press, 2005.

[22] D. Sheldon. Graphical multi-task learning. In *NIPS 2008 Workshop on Structured Input and Structured Output*, 2008.

[23] R. Silva, W. Chu, and Z. Ghahramani. Hidden common cause relations in relational learning. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*. MIT Press, 2008.

[24] S. Thrun. Is learning the n-th thing any easier than learning the first? In D. S. Touretzky, M. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8*, pages 640–646. MIT Press, 1996.

[25] S. Thrun and J. O'Sullivan. Discovering structure in multiple learning tasks: The TC algorithm. In L. Saitta, editor, *Proceedings of the 13th International Conference on Machine Learning (ICML '96)*. Morgen Kaufmann, 1996.

[26] Z. Xu, K. Kersting, and V. Tresp. Multi-relational learning with gaussian processes. In C. Boutilier, editor, *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI–09)*, Pasadena, CA, USA, July 11–17 2009.

[27] Y. Xue, X. Liao, L. Carin, and B. Krishnapuram. Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research*, 8:35–63, 2007.

[28] K. Yu, W. Chu, S. Yu, V. Tresp, and Z. Xu. Stochastic relational models for discriminative link prediction. In B. Schölkopf, J. Platt, and T. Hofmann, editors, *Advances in Neural Information Processing Systems 19*, pages 1553–1560. MIT Press, 2007.

[29] K. Yu, V. Tresp, and A. Schwaighofer. Learning gaussian processes from multiple tasks. In *Proceedings of the 22nd International Conference on Machine learning (ICML '05)*, pages 1012–1019, 2005.

[30] S. Yu, V. Tresp, and K. Yu. Robust multi-task learning with t-processes. In *Proceedings of the 24th International Conference on Machine learning (ICML '07)*, pages 1103–1110, 2007.

[31] Y. Zhang and D.-Y. Yeung. Multi-task learning using generalized t process. In Y. W. Teh and M. Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS '10)*, pages 964–971, 2010.

[32] X. Zhu, J. Kandola, J. Lafferty, and Z. Ghahramani. Graph kernels by spectral transforms. In O. Chapelle, B. Schoelkopf, and A. Zien, editors, *Semi-Supervised Learning*. MIT Press, 2005.

[33] X. Zhu, J. Lafferty, and Z. Ghahramani. Semi-supervised learning: From gaussian fields to gaussian processes. Technical report, Technical Report CMU-CS-03-175, 2003.