Stacked Gaussian Process Learning

Conference Paper · December 2009

DOI: 10.1109/ICDM.2009.56 · Source: DBLP

4 authors, including:

Marion Neumann
Washington University in St. Louis
15 PUBLICATIONS **61** CITATIONS

SEE PROFILE

Kristian Kersting
University of Bonn
304 PUBLICATIONS **3,323** CITATIONS

SEE PROFILE

Stacked Gaussian Process Learning

Marion Neumann,[1,2] Kristian Kersting,[2] Zhao Xu,[2] and Daniel Schulz[2]

[1]*Institute of Applied Information Processing*
University of Ulm, Germany
neumann_marion@gmx.de

[2]*Fraunhofer IAIS*
Sankt Augustin, Germany
{firstname.lastname}@iais.fraunhofer.de

***Abstract*—Triggered by a market relevant application that involves making joint predictions of pedestrian and public transit flows in urban areas, we address the question of how to utilize hidden common cause relations among variables of interest in order to improve performance in the two related regression tasks. Specifically, we propose stacked Gaussian process learning, a meta-learning scheme in which a base Gaussian process is enhanced by adding the posterior covariance functions of other related tasks to its covariance function in a stage-wise optimization. The idea is that the stacked posterior covariances encode the hidden common causes among variables of interest that are shared across the related regression tasks. Stacked Gaussian process learning is efficient, capable of capturing shared common causes, and can be implemented with any kind of standard Gaussian process regression model such as sparse approximations and relational variants. Our experimental results on real-world data from the market relevant application show that stacked Gaussian processes learning can significantly improve prediction performance of a standard Gaussian process.**

***Keywords*-Statistical Relational Learning, Gaussian Processes, Bayesian Regression, Stacked Learning**

I. Introduction

Data mining systems that process noisy sensory input often treats tasks involved independently of each other. Although this makes the systems comparatively easy to assemble, and helps to contain computational complexity, it comes at a high price: information from one task cannot be utilized to help processing the other so that errors made in one task cannot be corrected, see e.g. [1]. Examples of this can be found in information extraction, natural language processing, speech recognition, vision, robotics, etc. In this paper, we are triggered by a market relevant application of pricing outdoor poster sites.

The German outdoor advertising market has a yearly turnover of 1,200 million dollars. The 'Fachverband Außenwerbung e.V.' (FAW), which is the governing organization of German outdoor advertising, provides performance indicators on which pricing of poster sites is based. The value of each site is composed of a quantitative measure, namely the number of passing vehicles, pedestrians, and public transport, cf. Fig. 1, and of a qualitative measure which specifies the expected notice of passers-by. This is a very rare if not unique case where a spatial data mining model [2], [3], [4] is business critical for a whole branch of industry, comprising a large number of companies. Specifically, the spatial data mining model proposed has become the basis for pricing of posters for that branch of industry and can be used for selecting new poster sites and for planning campaigns. Moreover, the outdoor advertising sector considers the model to be a milestone for their business; the reliability of the prediction result seems to be accepted by dozens of companies in this area.

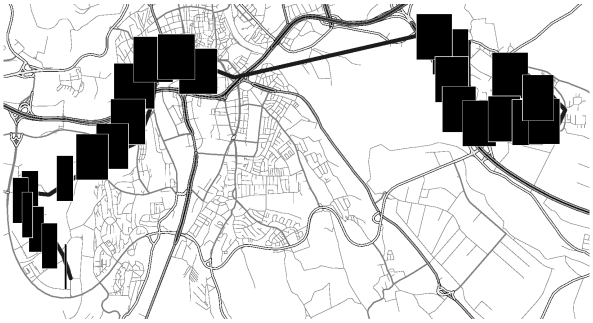


Figure 1. Flow predictions (black boxes) for a bus line in the German city of Ulm using Gaussian processes. The larger the boxes, the more predicted passengers. The bus line is indicated as black line.

Although much effort has gone into developing the model for pricing posters and indeed it has improved steadily over the years, it still has important drawbacks. In the present paper, we focus on the following ones:

- Regression tasks involved are treated independently. Reconsider our motivating application. We are asked to predict how many people will walk by a location and how many people will pass by a location using public transit such as buses. Indeed, when we observe a high number of pedestrians, it is very likely that we also observe a larger number of people on buses, and vice versa.
- Furthermore, there is a lot of uncertainty involved which is not modelled explicitly. Consider e.g. the pedestrian flow data. For each poster site to be evaluated, video measurements have been made. The measurements at a site have been taken manually at 4 different days and 4 different hours lasting 6 minutes. Although, over the years, more than 100.000 pedestrian

flow measurements have been collected this way, data collection is not an easy task: Some locations might be measured more often than others; some not at all. Some measurements might be taken at rush time, others not. For some location, we might have only one value of interest, say the pedestrian flow, measured but not the other.

So, to address both problems, we would ideally like to perform joint Bayesian inference for both tasks simultaneously. Current systems, however, treat the regression tasks independently and employ non-Bayesian regression approaches. Although recent progress in probabilistic inference and machine learning has begun to make joint inference possible. For example, multi-output and multi-task regression approaches [5], [6], [7], [8], transfer learning [9], [10], [11], [12], and the emerging field of statistical relational learning (SRL) [13], [14] are essentially concerned with learning from data points that are not independent and identically distributed. However, setting up a joint inference model is usually rather complex, and the computational cost of running it can be prohibitive. This is exactly the problem we will address in this paper.

Specifically, we propose *stacked Gaussian process learning*, a novel meta-learning scheme in which a base Gaussian process is enhanced by adding the posterior covariance functions of other related tasks to its covariance function in a stage-wise optimization. The benefits are three-fold:

- In terms of the application, Gaussian processes are attractive because of their flexible non-parametric nature and the predictive distributions they provide. Moreover, they also allow one to elegantly make use of relational information available [15], [16], [17]. Reconsider our application. We should be able to propagate our prediction at one location to locations on neighbouring streets, i.e., along the underlying transportation or street network.
- Sometimes, however, it can be quite difficult – if not impossible – to come up with relations, for instance if there is no domain expert around. Stacked Gaussian process learning provides us with a simple but effective way to uncover the relations from the related tasks. The basic idea is that stacked posterior covariance functions encode the hidden common causes among variables of interest that are shared across related regression tasks. As we will show in our experiments on real-world data from our application, this can indeed improve performance on each task.
- Finally, stacked Gaussian process learning is very efficient as it can be implemented virtually with no overhead using any kind of off-the-shelves Gaussian process model such as sparse approximations and relational variants.

We proceed as follows. We start of by touching upon related work. Then, we review regression with standard Gaussian processes in Section III and relational ones in Section IV. In Section V, we develop stacked Gaussian process learning. Afterwards, Section VI presents our experimental evaluation on several real-world dataset. Then, we conclude.

II. Related Work

Stacked Gaussian process learning is the first application of relational Gaussian processes to a market relevant regression problem, namely flow prediction in public transportation and street networks. Scheider *et al.* [3] proposed to use off-the-shelves linear regression and model tree approaches whereas May *et al.* [4] investigated non-stationary k-nearest-neighbour approaches. In contrast to stacked Gaussian processes, they neither provide a principled non-linear Bayesian framework nor cross-domain/joint learning of both tasks.

Algorithmically, stacked Gaussian process learning is related to several recent machine learning and data mining approaches. Probably the closest work is stacked graphical learning [18], [19], [20]. So far, this line of research has focused on stacking Markov networks. There is, however, no reason why Markov networks should be the only representation of choice for symmetric dependence, i.e., relational structures. In this paper, we extend stacking to the case when relationships are postulated to exist due to hidden common causes [16]. As Silva *et al.* [16] have shown, this corresponds to a graphical representation called directed mixed graph (DMG), with bi-directed edges representing the relationship of having hidden common causes between a pair of vertices. However, regression has not been considered for the mixed-graph Gaussian process framework so far.

Finally, stacked Gaussian processes are related to multi-output and multi-task regression approaches [5], [6], [7], [8], transfer learning [9], [10], [11], [12], collaborative learning [21], [22], and the emerging field of statistical relational learning (SRL) [13], [14]. Here, the work of Yu *et al.* [5] is probably the closest. Whereas Yu *et al.* propose an hierarchical Bayes approach, learning the shared hyperparameters using EM, we turn the EM into a boosting like stage-wise optimization of the intra-task covariance matrices. This simplifies the approach and allows one to implement a simple version using any off-the-shelves Gaussian process approach, even relational ones.

III. Gaussian Processes for Regression

The flow prediction problems can be formulated as a Bayesian regression task. In this section, we will show how to encode it with Gaussian process (GP) regression models; here using local attributes only and then utilizing relations in Sec. IV. For a comprehensive introduction and further details on GPs, we refer to [23].

Assume that each segment, i.e., a street or segment of a public transportation line (represented as the center of its bounding box) at which we want to predict a flow is

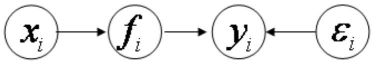


Figure 2. The GP regression model.

associated with attributes $x_i \in \mathbb{R}^D$ and an observation $y_i \in \mathbb{R}$. The attributes include, e.g., the numbers of public buildings, touristic sites, restaurants near the segment, and so on. The observation y_i is some log-transformed flow measurement, i.e., numbers of people passing the segment. The log-transformation of the measurements is necessary since flows are restricted to be positive. Fig.2 illustrates the GP regression model. In a GP framework, the observation is modeled as a noise contaminated function value $f(x_i)$ (shortened as f_i in the rest of the paper) of its attributes x_i, that is

$$y_i = f_i + \epsilon_i, \tag{1}$$

where ϵ_i is Gaussian noise with variance σ^2. The function value f_i is the real, i.e., noise-free flow, which is unobservable. Since the form and parameters of the function is unknown and random, f_i is also unknown and random. Now, the function values $\{f_1, f_2, \ldots\}$ of an infinite number of segments can be represented as an infinite dimensional vector, i.e., the i'th dimension is the function value f_i. We assume that the infinite dimensional random vector follows a Gaussian process prior with mean function $m_a(x_i)$ and covariance function $k_a(x_i, x_j)$. Here, the subscript a emphasizes that the mean and covariance functions are attribute-wise. In turn, any finite set of function values $\{f_i : i = 1, \ldots, n\}$ has a multivariate Gaussian distribution with mean vector and covariance matrix defined in terms of the mean and covariance functions of the GP, see e.g. [23]. Without loss of generality, we assume zero mean so that the GP is completely specified by the covariance function. A typical choice is the squared exponential covariance function with isotropic distance measure:

$$k_a(x_i, x_j) = \kappa^2 \exp\left(-\frac{\rho^2}{2} \sum\nolimits_d^D (x_{i,d} - x_{j,d})^2\right), \tag{2}$$

where κ and ρ are parameters of the covariance function. $x_{i,d}$ denotes the d-th dimension of the attribute vector x_i.

Formally, for a set of n segments with attributes $X = \{x_1, \ldots, x_n\}$, the multivariate Gaussian prior distribution of the function values $f = (f_1, \ldots, f_n)^T$ is written as:

$$\begin{aligned} P(f|X) &= \mathcal{N}(0, K_a) \\ &= \frac{1}{(2\pi)^{\frac{n}{2}} |K_a|^{\frac{1}{2}}} \exp\left(-\frac{f^T K_a^{-1} f}{2}\right). \end{aligned} \tag{3}$$

Here, K_a denotes the $n \times n$ covariance matrix whose ij-th entry is computed in terms of the covariance function with the corresponding attributes x_i and x_j.

Since the observations $Y = (y_1, \ldots, y_n)^T$ are the function values with noise, cf. Eq. (1), their joint distribution is:

$$P(Y|X, \sigma^2) = \mathcal{N}(0, K_a + \sigma^2 I), \tag{4}$$

where I is $n \times n$ identity matrix.

For a new segment with attributes x_*, the predictive distribution of the function value f_* given the noisy observations Y can be computed as follows. Due to the properties of GPs, $(f, f_*)^T$ is still Gaussian with zero-mean and covariance matrix

$$\begin{bmatrix} K_a & k_a(X, x_*) \\ k_a(x_*, X) & k_a(x_*, x_*) \end{bmatrix}, \tag{5}$$

where $k_a(x_*, x_*)$ is the variance of the new function values f_*, $k_a(X, x_*)$ is a n-dimensional column vector, which i'th entry is $k_a(x_i, x_*)$, and $k_a(x_*, X)$ is the transpose of $k_a(X, x_*)$, i.e. $k_a(x_*, X) = k_a(X, x_*)^T$. Introducing the noise terms ϵ_i, we can write the joint distribution of the observations Y and the new function values f_* as

$$\begin{bmatrix} Y \\ f_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K_a + \sigma^2 I & k_a(X, x_*) \\ k_a(x_*, X) & k_a(x_*, x_*) \end{bmatrix}\right). \tag{6}$$

Finally, the predictive distribution of f_* is computed in terms of the formula on conditional Gaussian distributions. Specifically, we have

$$P(f_*|x_*, X, Y) = \mathcal{N}(\mathbb{E}(f_*), \mathrm{cov}(f_*)), \tag{7}$$

where

$$\begin{aligned} \mathbb{E}(f_*) &= k_a(x_*, X)(K_a + \sigma^2 I)^{-1} Y \\ \mathrm{cov}(f_*) &= k_a(x_*, x_*) \\ &\quad - k_a(x_*, X)(K_a + \sigma^2 I)^{-1} k_a(X, x_*). \end{aligned}$$

IV. Gaussian Processes with Relations

Public transportation systems can elegantly be represented using entities and relations. Specifically, they are collections of segments (entities) that are interconnected by streets or public transit lines (relations). Intuitively, we would like to make use of this additional knowledge: our prediction at one segment should help us to reach conclusions about flows at other, related segments. Reconsider Fig. 1. There, the bus line denoted as thick, black line connects several segments.

Several GP variants exactly doing this have recently been proposed [24], [25], [16], [17]. In this paper, we extend Silva *et al.*'s mixed graph Gaussian processes (XGP) [16] to model flows in public transportation system. The XGP model, which represents the relational information as hidden common causes, is a straightforward way to incorporate relations into Gaussian processes, and has successfully been applied to classification and ranking problems.

Assume there are not only segment attributes X and flow observations Y but also relations $R = \{r_{i,j} : i, j \in$

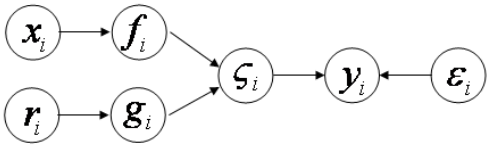


Figure 3. The XGP regression model for relational data.

$1, \ldots, n\}$ such as bus lines or streets between segments. All relations the segment i participates in are denoted as r_i. It is natural to assume that the flow of a segment depends on both the attributes x_i of the segment and the relations r_i the segment participates in. To include r_i into the GP model, we introduce another function value $g(r_i)$ (shorten as g_i) for each segment shown as Fig.3. Now, a segment is associated with two function values: f_i representing the flow predicted based on segment attributes, and g_i representing the flow based on relations. The overall function value (noise-free flow) ζ_i of a segment is a weighted sum of the two components

$$\zeta_i = f_i + \omega_0 g_i, \tag{8}$$

where ω_0 is a mixture weight. The function value g_i encodes some *hidden common causes* from relations. The term *hidden* is due to the fact that the relational information encoded over segments is only *implicit* in contrast to the commonly used information on segment attributes. Now, the observation y_i is the overall, noisy function value

$$\begin{aligned} y_i &= \zeta_i + \epsilon_i \\ &= f_i + \omega_0 g_i + \epsilon_i, \end{aligned} \tag{9}$$

Similarly to the attribute-wise function values, we place a zero-mean GP prior over $\{g_1, g_2, \ldots\}$. Again, $\{g_i : i = 1, \ldots, n\}$ follow a multivariate Gaussian distribution. In contrast to the attribute-wise GPs, however, the covariance function $k_r(r_i, r_j)$ should represent correlation of segments i and j due to the relations. There are essentially two strategies to define such kernel functions. The simplest way is to represent the known relations of segment i as a vector. The kernel function $k_r(r_i, r_j)$ can then be any Mercer kernel function, and the computations are essentially the same as for the attributes. Alternatively, we notice that segments and relations form a graph, and we can naturally employ graph-based kernels to obtain the covariances, see e.g. [26], [27], [16]. The simplest graph kernel might be the regularized Laplacian kernel

$$K_r = [\beta(\Delta + I/\iota^2)]^{-1}, \tag{10}$$

where β and ι are two parameters of the graph kernel. Δ denotes the combinatorial Laplacian, which is computed as $\Delta = D - W$, where W denotes the adjacency matrix of a weighted, undirected graph, i.e., $w_{i,j}$ is taken to be the weight associated with the edge between i and j. D is a diagonal matrix with entries $d_{i,i} = \sum_j w_{i,j}$.

The function values $f = (f_1, \ldots, f_n)^T$ and $g = (g_1, \ldots, g_n)^T$ are both Gaussian, thus their weighted sum $\zeta = (\zeta_1, \ldots, \zeta_n)^T$ also follows a multivariate Gaussian distribution. Its mean is the weighted sum of the means of the two independent Gaussian distributions. Since we assume zero means for f and g, the mean of ζ is also zero. The covariance between ζ_i and ζ_j is computed as

$$\begin{aligned} \text{cov}(\zeta_i, \zeta_j) &= \mathbb{E}\left[(f_i + \omega_0 g_i)(f_j + \omega_0 g_j)\right] \\ &= \text{cov}(f_i, f_j) + \omega_0^2 \text{cov}(g_i, g_j). \end{aligned} \tag{11}$$

Thus, the covariance matrix for ζ is

$$K = K_a + \omega_0^2 K_r. \tag{12}$$

Here, K_a (resp. K_r) denotes the $n \times n$ covariance matrix whose ij-th entry is computed with the corresponding covariance function. Finally, the prior distribution of the overall function value ζ is defined as

$$\begin{aligned} P(\zeta|X, R) &= \mathcal{N}(0, K) \\ &= \frac{1}{(2\pi)^{\frac{n}{2}} |K|^{\frac{1}{2}}} \exp\left(-\frac{\zeta^T K^{-1} \zeta}{2}\right). \end{aligned} \tag{13}$$

The distribution for the noisy observations Y is

$$P(Y|X, R, \sigma^2) = \mathcal{N}(0, K + \sigma^2 I). \tag{14}$$

Compared to Eq. (4), the only difference is replacing K_a with K, the enhanced covariances between interrelated segments, Eq. (12).

In the XGP framework, making flow predictions on segments is considered in a transductive learning setting, i.e. there is no new segment introduced in prediction[1]. In particular, the flows of some of the n segments are missing and the task is to predict them given the observed segments.

Let $\zeta = (\zeta_L, \ \zeta_U)^T$ be the vector composed of ζ_L and ζ_U, i.e., the function values (noise-free flows) for the observed and unobserved segments, respectively. In analogy to predictive inference in standard GP regression models, see Eq. (6), we introduce the noise terms ϵ_i and obtain the joint distribution of the observations Y and the function values ζ_U of the unobserved segments as follows

$$\begin{bmatrix} Y \\ \zeta_U \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K_{L,L} + \sigma^2 I_{|L|} & K_{L,U} \\ K_{U,L} & K_{U,U} \end{bmatrix}\right), \tag{15}$$

where $K_{U,L}$ are the corresponding entries of K between the unobserved segments and observed ones. $K_{L,L}$, $K_{U,U}$, and $K_{L,U}$ are defined equivalently. $I_{|L|}$ is an identity matrix, where the dimensionality is the number of observed segments.

[1] If there are new segments available, the graph of the segments will change so that the relation-wise covariance matrix K_r has to be re-computed.

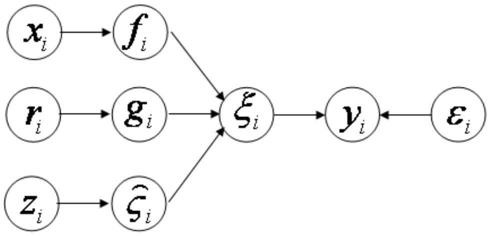


Figure 4. The stacked GP model propagating information among related tasks.

Finally, the predictive distribution of ζ_U given Y is computed as

$$P(\zeta_U | Y, X, R, \sigma^2) = \mathcal{N}\left(\mathbb{E}\left(\zeta_U\right),\ \mathrm{cov}(\zeta_U)\right), \tag{16}$$

where

$$\begin{aligned}\mathbb{E}\left(\zeta_U\right) &= K_{U,L} C\, Y \\ \mathrm{cov}(\zeta_U) &= K_{U,U} - K_{U,L} C\, K_{L,U} \\ C &= (K_{L,L} + \sigma^2 I_{|L|})^{-1}.\end{aligned}$$

V. Stacked Gaussian Process Learning

So far, we have assumed that the relations are given. But where do the relations come from? Often, a domain expert determines the relations. This can be difficult and expensive. Collecting additional data that can be used to learn the relations, however, is often easier. Specifically, we often have data for several, related regression tasks. In our application of pricing posters, for example, we not only have observed public transit flows but also pedestrian flows: when we predict a high number of pedestrians at one location, it is very likely that we also observe a larger number of people on buses, and vice versa. In this Section, we will show how to extract automatically hidden common cause relations from Gaussian process models of the related regression tasks.

Specifically, we extend the XGP framework to *stacked Gaussian processes* (SGPs). SGPs are a meta-learning scheme in which a base Gaussian process is enhanced by adding the posterior covariance functions of other related tasks to its covariance function in a stage-wise optimization. The idea is that the stacked posterior covariances encode the hidden common causes among variables of interest that are shared across the related regression tasks. In other words, all tasks influence each other so that the learning and prediction on one task is based on the other ones. The process is iterated until convergence.

Assume that for each segment, there is pedestrian information z_i associated, including pedestrian attributes $\hat{x}_i$ and pedestrian relations $\hat{r}_i$. Additionally we have a flow observation $\hat{y}_i$. We introduce a new function value $\hat{\zeta}(z_i)$ (shorten as $\hat{\zeta}_i$) to encode the additional hidden common causes from the pedestrian data. Thus, the noise-free flow ξ_i on public transportation is the weighted sum of all causes (Fig.4):

$$\xi_i = f_i + \omega_0 g_i + \omega_1 \hat{\zeta}_i, \tag{17}$$

where ω_1 is the mixture weight for the new cause.

The distributions of f_i and g_i are given. To obtain the distribution of ξ_i, however, we need the distribution of $\hat{\zeta}_i$. To solve the problem, we start off with an XGP model for the pedestrian data without considering the public transportation data. As discussed in Section IV, $\hat{\zeta}_i$ can be decomposed as

$$\hat{\zeta}_i = \hat{f}_i + \hat{\omega}_0 \hat{g}_i,$$

where $\hat{f}_i$ and $\hat{g}_i$ denote the function values of $\hat{x}_i$ and $\hat{r}_i$, i.e., $\hat{f}_i = f(\hat{x}_i)$, $\hat{g}_i = g(\hat{r}_i)$, which follow zero-mean Gaussian with covariance matrices $\hat{K}_a$ and $\hat{K}_r$, respectively. For the computations of $\hat{K}_a$ and $\hat{K}_r$, we refer back to Section IV. The main idea for exploiting the observations on the pedestrian flow is now as follows:

> Use the posterior covariance of the pedestrian GP model as (weighted) hidden common cause relation within the public transit GP model.

More formally, the posterior of $\hat{\zeta}_i$ is used to compute the distribution of ξ_i (Eq. (17)). In this way, all the information on pedestrian attributes $\hat{X}$, pedestrian relations $\hat{R}$, and pedestrian observations $\hat{Y}$ is transfered to the public transportation task.

To be more precise, the posterior distribution of the function values $\hat{\zeta} = (\hat{\zeta}_1, \ldots, \hat{\zeta}_n)^T$ (noise-free pedestrian flows) can be found to be:

$$\begin{aligned}P(\hat{\zeta}|\hat{X}, \hat{R}, \hat{Y}, \hat{\sigma}^2) &= \frac{P(\hat{\zeta}|\hat{X}, \hat{R}) \prod_{i=1}^{n} P(\hat{y}_i|\hat{\zeta}_i, \hat{\sigma}^2)}{P(\hat{Y}|\hat{X}, \hat{R}, \hat{\sigma}^2)} \\ &= \mathcal{N}\left(\hat{\mu},\ \hat{\Sigma}\right),\end{aligned} \tag{18}$$

where $\hat{\sigma}^2$ is the variance of the noise for the pedestrian observations. The likelihood $P(\hat{y}_i|\hat{\zeta}_i, \hat{\sigma}^2)$ is a Gaussian with mean $\hat{\zeta}_i$ and variance $\hat{\sigma}^2$. $P(\hat{\zeta}|\hat{X}, \hat{R})$ is the prior of $\hat{\zeta}$, which is zero-mean Gaussian with the covariance matrix $\hat{K} = \hat{K}_a + \hat{\omega}_0^2 \hat{K}_r$ (referring to Eq.12). $\hat{\mu}$ and $\hat{\Sigma}$ are the mean and covariance matrix of the posterior distribution

$$\begin{aligned}\hat{\mu} &= \begin{bmatrix}\hat{\mu}_L \\ 0\end{bmatrix} \\ \hat{\Sigma} &= \left(\hat{K}^{-1} + \hat{\sigma}^{-2} I\right)^{-1},\end{aligned} \tag{19}$$

where

$$\hat{\mu}_L = \hat{\sigma}^{-2} \hat{\Sigma}_{L,L} \hat{Y}.$$

Since the function values f, g and $\hat{\zeta}$ are three independent Gaussians, their sum ξ is still Gaussian with mean m and

	Task 1	Task 2
Input:		
Prior C.M. of ζ and $\hat{\zeta}$	$K = K_a + \omega_0 K_r$	$\hat{K} = \hat{K}_a + \hat{\omega}_0 \hat{K}_r$
Prior mean of ζ and $\hat{\zeta}$	0	0
Noisy observations	Y	$\hat{Y}$
Variance of noise	σ^2	$\hat{\sigma}^2$
Initialization:		
Prior C.M. of ξ and $\hat{\xi}$	$\Lambda = K$	$\hat{\Lambda} = \hat{K}$
Prior mean of ξ and $\hat{\xi}$	$m = 0$	$\hat{m} = 0$
Iterate until both tasks converge (probably discounting mixture weights over time):		
Posterior C.M. of ξ and $\hat{\xi}$	$\Sigma = (\Lambda^{-1} + \sigma^{-2} I)^{-1}$	$\hat{\Sigma} = (\hat{\Lambda}^{-1} + \hat{\sigma}^{-2} I)^{-1}$
Posterior mean of ξ and $\hat{\xi}$	$\mu = \Sigma(\Lambda^{-1} m + \sigma^{-2} Y)$	$\hat{\mu} = \hat{\Sigma}(\hat{\Lambda}^{-1} \hat{m} + \hat{\sigma}^{-2} \hat{Y})$
Prior C.M. of ξ and $\hat{\xi}$	$\Lambda = K + \omega_1^2 \hat{\Sigma}$	$\hat{\Lambda} = \hat{K} + \hat{\omega}_1^2 \Sigma$
Prior mean of ξ and $\hat{\xi}$	$m = \omega_1 \hat{\mu}$	$\hat{m} = \hat{\omega}_1 \mu$
Prediction of ξ_u and $\hat{\xi}_u$:		
Predictive C.M.	$C = (\Lambda_{L,L} + \sigma^2 I_{\lvert L \rvert})^{-1}$	$\hat{C} = (\hat{\Lambda}_{L,L} + \hat{\sigma}^2 I_{\lvert L \rvert})^{-1}$
	$\text{cov}(\xi_U) = \Lambda_{U,U} - \Lambda_{U,L} C \Lambda_{L,U}$	$\text{cov}(\hat{\xi}_U) = \hat{\Lambda}_{U,U} - \hat{\Lambda}_{U,L} \hat{C} \hat{\Lambda}_{L,U}$
Predictive mean	$\mathbb{E}\,(\xi_U) = \Lambda_{U,L} C Y$	$\mathbb{E}\,(\hat{\xi}_U) = \hat{\Lambda}_{U,L} \hat{C} \hat{Y}$

Table I

Stacked Gaussian Process (SGP) learning. For our flow prediction problems, Task 1 would be the flow prediction for public transportation and Task 2 the pedestrian flow prediction. The term *C.M.* is an abbreviation for “covariance matrix”.

covariance Λ

$$\begin{aligned} m &= \omega_1 \hat{\mu} \\ \Lambda &= K_a + \omega_0^2 K_r + \omega_1^2 \hat{\Sigma} \\ &= K + \omega_1^2 \hat{\Sigma}, \end{aligned} \tag{20}$$

where K_a and K_r are covariances on transportation attributes and relations, respectively. The matrix $\hat{\Sigma}$ is the posterior covariance of the pedestrian data.

Making predictions is straightforward. Given some observed public transportation frequencies Y, the predictive distribution of the unobservable noise-free frequencies ξ_u is essentially computed as in Eq. (16). We only have to replace the covariance matrix K by Λ. Again, the prediction exploits transportation attributes (K_a), transportation relations (K_r), and the additional hidden common cause relations ($\hat{\Sigma}$) transfered from the related task.

Indeed, nothing prevents us from also ‘boosting’ the public transportation hidden common causes into the pedestrian model in an equivalent way. As the new posterior covariances of both GPs might change, stacked Gaussian process (SGP) learning iterates the processes until convergence (probably discounting mixture weights over time) as summarized in Table I.

Note that in the iteration step, the update of transportation flow prior (Λ and m) in Task 1 uses the pedestrian posterior ($\hat{\Sigma}$ and $\hat{\mu}$) in Task 2, and vice versa, by which the information propagates between the tasks. Thus, SGP essentially implements a simple form of joint learning. For numerical stability, one can use the formula $(A^{-1} + B^{-1})^{-1} = A - A\,(A + B)^{-1} A$ to compute the covariance matrices.

The computational complexity of the proposed stacked learning algorithm is $\mathcal{O}(n^3)$ per task and iteration, where n is the number of training cases. Thus, stacked Gaussian processes scale as standard Gaussian processes with a linear overhead depending on the number of iterations. Experiments have shown that already one iteration of stacking often yields significant improvement.

VI. Experimental Evaluation

Our intention is to investigate the following question:

(Q) Does stacked Gaussian process learning improve upon the predictive performance of unstacked learning?

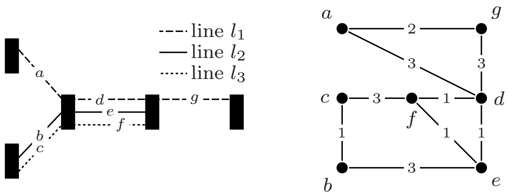
To answer the question, we implemented the stacked Gaussian process learning in Matlab using Rasmussen and Williams’ Gaussian Process library [23] and compared base models with their stacked counterparts.

A. Real-World Datasets

We used the datasets described in detail by Scheider and May [3] and by May *et al.* [4]. They consist of pedestrian flow respectively public transport flow observations of

Dataset	Public Transport						Pedestrian					
	y_{min}	y_{max}	$\bar{y}$	σ_y	N_L	N_U	y_{min}	y_{max}	$\bar{y}$	σ_y	N_L	N_U
Frankfurt	0	2908	268	460	747	1107	0	4578	134	355	678	1176
Kaiserslautern	0	205	48	49	353	465	1	387	54	65	101	717
Ulm	1	853	177	187	193	694	0	213	41	49	147	740
Konstanz	1	407	79	78	164	516	5	500	114	98	109	571

Table II
CHARACTERISTICS OF THE PUBLIC TRANSPORT AND PEDESTRIAN FLOW DATASETS FOR THE 4 GERMAN CITIES USED IN THE EXPERIMENTS. THE VALUES y_{min} AND y_{max} DENOTE THE MINIMAL AND MAXIMAL COUNTS/FLOWS OBSERVED, $\bar{y}$ AND σ_y THE MEAN VALUE AND THE STANDARD DEVIATION OF OBSERVATIONS, AND N_L RESP. N_U DENOTES THE NUMBER OF OBSERVED RESP. UNOBSERVED DATA POINTS.



(a) network (b) relational graph

Figure 5. The relational graph in panel (b) shows the segment-based graph representation with weights $w \in \{1, 2, 3\}$ for the public transport network shown in panel (a) with bus lines $a, b, .., g$.

several German cities together with the underlying network information, i.e., attributes describing stops and segments as well as the involved bus and tram lines. Sufficient observed data for sound evaluation is only available for a few cities. So, we considered the subset containing the 4 German cities Frankfurt, Kaiserslautern, Ulm, and Konstanz.

Each segment is described in terms of a set of attributes that are deduced from the characteristics of the segments' bounding stops and the respective line information. The most important feature is the schedule frequency of lines which measures the traffic pressure between stops. An additional attribute marks the type of the line, i.e., bus, tram, or underground. Furthermore, there are attributes characterising the attractiveness of stops measured by the spatial density of several points of interest (POI) like touristic and cultural facilities, public utilities, restaurants, and hotels. In particular, the attribute values are aggregated number of POI within a POI specific spatial radius around stops. We refer to [3], [4] for more details. Note, however, that the pedestrian data provided slightly different attributes than the public transportation data such as street type and boolean values for footpath and pedestrian area.

Additionally, we also extracted graphs/relations among the segments from the available network information. More precisely, for the pedestrian data, we compute an $\epsilon -$ nearest neighbour graph encoding relations among segments on the basis of the pedestrian data in the underlying street network. The public transportation network, however, had to be first turned into a segment-based representation. Consider Fig. 5. It shows the original network on the left and the segment-based representation on the right. Essentially, each edge in the original representation becomes an node in the new, segment-based representation. Edges between two segments i and j exist due to the following expert-defined relations weighted with w_{ij}:

$$w_{ij} = \begin{cases} 1 & i \text{ and } j \text{ are of different lines} \\ & \text{and have both stops in common} \\ 2 & i \text{ and } j \text{ are of the same line} \\ & \\ 3 & i \text{ and } j \text{ are of the same line} \\ & \text{ and have only one stop in common} \end{cases} \tag{21}$$

The general statistics of both the public transport and the pedestrian dataset for the different cities are summarised in Table II, which shows range (y_{min}, y_{max}), mean $\bar{y}$ and standard deviation σ_y of the noisy observations and the number of labeled and unlabeled data (N_L, N_U). Frankfurt is the only considered city having an underground. In Ulm we predict passenger frequencies for trams and buses considering the joint public transport network and in the other two cities only buses serve as public transport means.

B. Methodology

On the resulting 4 datasets, we compared several models. Specifically, we trained GPs taking no relational information into account (denoted as GP) as well as relational variants (denoted as XGP) as base learners. For both pedestrian and public transport models, we used the squared exponential covariance function with different length-scales for each input dimension. To treat the relational information, we used a regularized Laplacian kernel [26]. The mixture weights were selected using cross-validation on the training set. Hyperparameter optimization was carried out using Rasmussen and Williams' conjugate gradient maximization of the log marginal likelihood. Since traffic flows are constrained to be non-negative, a log-transformation of the observation space was used. Each dataset was split randomly into 2/3 training and 1/3 test set. On the test set, we computed several evaluation measures to gain a complete picture of the predictive performance of the trained models. Specifically, we used:

Coefficient of Determination:

$$R^2 = \left(\frac{\sum_i (y_i - \bar{y}) \left(f_i - \bar{f}\right)}{(n-1)\ \sigma_y \sigma_f} \right)^2$$

Root Mean Squared Error:

$$RMSE = \sqrt{\frac{1}{m} \sum_i (y_i - f_i)^2}$$

Mean Absolute Error:

$$MAE = \frac{1}{m} \sum_i |y_i - f_i|$$

Negative Log Predictive Density:

$$\begin{aligned} NLPD &= -\log P(\mathbf{y}) \\ &= \frac{1}{m} \sum_i \left(\frac{1}{2} \log(2\pi\sigma_i^2) + \frac{(y_i - f_i)^2}{2\sigma_i^2} \right). \end{aligned}$$

Then, stacked learning was initiated using the learned base models yielding their stacked counterparts, denoted as SGP and SXGP. As a baseline approach, we also evaluated a distance-weighted 5-nearest-neighbour method. Each experiment was ran 10 times and we report the averaged values. Statistical significance was assessed using a two-sample t-tests with 95% confidence intervals. All experiments were conducted on a standard Windows laptop PC.

C. Results

Table III summarizes the experimental results. Additionally to the performance measures averaged over 10 reruns with random initialization, it also indicates whether the results are significant or not. Specifically, we tested on significant performance improvements for the following pairs of learners: kNN vs. GP, GP vs. SGP, GP vs. XGP, and XGP vs. SXGP.

As one can see, standard GPs significantly improve the kNN method by 3 error measures in 3 out of 4 cases for the task of predicting public transport passenger flows. Moreover, the results clearly show that GP regression benefits from relational information. In particular, stacked GPs performed significantly better than standard GPs for half of the considered datasets on both tasks. The XGP using 'expert' relations yields significantly better predictive performance on nearly all evaluation measures. Moreover, stacking XGP clearly shows a significant performance gain, too. In 3 out of 4 cases the stacked XGP significantly outperformed unstacked XGP on both tasks, the prediction of public transport passenger and pedestrian flows. Table IV summarizes this overall cumulative 'significant win' statistics for all considered pairs of learners (see above).

	kNN	GP	XGP
GP	4	—	—
SGP	5	5	—
XGP	6	7	—
SXGP	7	8	6

Table IV
NUMBER OF 'SIGNIFICANT WINS', I.E., SIGNIFICANTLY BETTER PREDICTIVE PERFORMANCE ON THE TEST SET (PAIRED SAMPLED T-TEST, $p = 0.05$) FOR THE 8 EXPERIMENTS OF TABLE III (4 DATASETS, 2 TASKS) IN AT LEAST ONE ERROR MEASURE.
COMPARED METHODS: GP → KNN, RELATIONAL AND STACKED GP → STANDARD GP, SXGP → XGP.

To summarize, the extensive set of experiments clearly answers **(Q)** affirmatively.

VII. CONCLUSIONS

Joint inference is currently an area of great interest in information extraction, natural language processing, statistical relational learning, and other fields. Despite its promise, joint inference is often difficult to perform effectively, due to its complexity, computational cost, and sometimes mixed effect on accuracy.

In this paper, we have shown how these problems can be addressed in a market relevant application of pricing outdoor poster sites. Specifically, we have introduced stacked Gaussian process learning, a meta-learning scheme in which a base Gaussian process is enhanced by adding the posterior covariance functions of other related tasks to its covariance function in a stage-wise optimization. The idea is that the stacked posterior covariances encode the hidden common cause relations among variables of interest that are shared across the related tasks. The evaluations on several real-world datasets indicate that regression with stacked Gaussian processes can indeed improve upon the performance of non-joint approaches significantly.

Compared to other joint Bayesian learning approaches, stacked Gaussian process learning is efficient as it can be implemented with virtually no overhead using of-the-shelves Gaussian process models. This property allows it to be easily used even by non-experts and to be very competitive in applications where efficient Bayesian inference algorithms are important.

Future work will compare stacked Gaussian process learning to other joint Gaussian process models and (relational) graphical models such as Markov logic networks and relational Markov networks. We are also considering further applications of stacked Gaussian process learning to joint inference problems in domains such as information retrieval and natural language processing.

From the application perspective, we modelled an important and interesting aspect. Nevertheless, there are a lot of other open machine learning and data mining problems as

City	Model	Public Transport				Pedestrian			
		R^2	$RMSE$	MAE	NLPD	R^2	$RMSE$	MAE	NLPD
FRANKFURT									
	kNN	0.75	232	112	—	0.63	210	76	—
	GP	0.75	233	108	-0.73	0.68 $\diamond$	188	78	-1.78
	SGP	0.76	232	108	-0.73	0.70 $\bullet$	180	77	-1.81 $\bullet$
	XGP	0.76	228	107	-0.74	0.71 $\bullet$	171 $\bullet$	75 $\bullet$	-1.23 $\bullet$
	SXGP	0.77 $\star$	224	108	-1.10 $\star$	0.74 $\star$	163 $\star$	73	-1.12
KAISERSLAUTERN									
	kNN	0.58	32	22	—	0.50	53	27	—
	GP	0.80 $\diamond$	22 $\diamond$	14 $\diamond$	-1.08	0.51 $\bullet$	51 $\bullet$	28 $\bullet$	-0.20
	SGP	0.81 $\bullet$	21 $\bullet$	14	-1.10 $\bullet$	0.55 $\bullet$	49 $\bullet$	27 $\bullet$	-0.38
	XGP	0.84 $\bullet$	19 $\bullet$	13	-1.07	0.57	48	27	-0.33
	SXGP	0.85 $\star$	19 $\star$	13 $\star$	-1.05	0.60 $\star$	48 $\star$	26 $\star$	-0.56
ULM									
	kNN	0.57	127	91	—	0.45	35	24	—
	GP	0.79 $\diamond$	86 $\diamond$	59 $\diamond$	-0.73	0.48	34	21	-0.48
	SGP	0.84 $\bullet$	77 $\bullet$	54 $\bullet$	-0.85	0.56 $\bullet$	32 $\bullet$	19 $\bullet$	-0.44
	XGP	0.80 $\bullet$	84 $\bullet$	59 $\bullet$	-0.76 $\bullet$	0.51 $\bullet$	33 $\bullet$	21 $\bullet$	-0.49
	SXGP	0.84 $\star$	76 $\star$	54 $\star$	-0.79	0.58 $\star$	31 $\star$	19 $\star$	-0.48
KONSTANZ									
	kNN	0.35	65	46	—	0.46	67	45	—
	GP	0.57 $\diamond$	52 $\diamond$	34 $\diamond$	6.16	0.49	62	44	-0.56
	SGP	0.60	51	33	5.87	0.50	68	47	-0.52
	XGP	0.73 $\bullet$	42 $\bullet$	28 $\bullet$	-0.78	0.53 $\bullet$	57 $\bullet$	41	-0.54
	SXGP	0.74	41	28	-0.78	0.56	57	40	-0.46
TOTAL									
	kNN	0.56	107	68	—	0.51	91	43	—
	GP	0.73	98	52	0.91	0.54	84	43	-0.76
	SGP	0.75	95	52	0.80	0.58	82	42	-0.79
	XGP	0.78	93	52	-0.84	0.58	77	41	-0.65
	SXGP	0.80	90	51	-0.93	0.62	75	39	-0.66

Table III

RESULTS OF THE MODELS KNN, GP , SGP, XGP, SXGP FOR 4 GERMAN CITIES. THE EVALUATION IS BASED ON HE COEFFICIENT OF DETERMINATION R^2, THE ERROR MEASURES $RMSE$ AND MAE AS WELL AS THE NEGATIVE LOG PREDICTIVE DENSITY $NLPD$. THE FOLLOWING NOTATION IS USED FOR INDICATING SIGNIFICANT IMPROVEMENTS: $\diamond$ GP IMPROVES KNN; $\bullet$ SGP / XGP IMPROVE GP; $\star$ SXGP IMPROVES XGP.

for instance how to extrapolate flows to cities where we do not have observations for one or even both of the prediction tasks.

ACKNOWLEDGMENT

The authors would like to thank Michael May for the constructive discussions and the anonymous reviewers for their helpful comments. We also thank the German professional association for outdoor advertising 'Fachverband Außenwerbung e.V.' (FAW) for kindly providing the data. Kristian Kersting and Zhao Xu were supported by the Fraunhofer ATTRACT Fellowship STREAM.

REFERENCES

[1] H. Poon and P. Domingos, "Joint inference in information extraction," in *Proceedings of AAAI-07*, 2007.

[2] W. Klösgen and M. May, "Spatial subgroup mining integrated in an object-relational spatial database," in *Proceedings of PKDD-02*, 2002, pp. 275–283.

[3] S. Scheider and M. May, "A method for inductive estimation of public transport traffic using spatial network characteristics," in *Proceedings of 10th AGILE International Conference on Geographic Information Science*, 2007.

[4] M. May, S. Scheider, R. Rösler, D. Schulz, and D. Hecker, "Pedestrian flow prediction in extensive road networks using biased observational data," in *Proceedings of GIS-08*, 2008, p. 67.

[5] K. Yu, V. Tresp, and A. Schwaighofer, "Learning gaussian processes from multiple tasks," in *Proceedings of ICML-05*, 2005, pp. 1012 – 1019.

[6] A. Argyriou, T. Evgeniou, and M. Pontil, "Multi-task feature learning," in *Proceedings of NIPS-06*, 2006.

[7] A. Argyriou, C. Micchelli, M. Pontil, and Y. Ying, "A spectral regularization framework for multi-task structure learning," in *Proceedings of NIPS-07*, 2007.

[8] E. Bonilla, K. Chai, and C. Williams, "Multi-task gaussian process prediction," in *Proceedings of NIPS-08*, 2008.

[9] S.-I.Lee, V. Chatalbashev, D. Vickrey, and D. Koller, "Learning a meta-level prior for feature relevance from multiple related tasks," in *Proceedings of ICML-07*, 2007, pp. 489 – 496.

[10] D. Roy and L. Kaelbling, "Efficient bayesian tasklevel transfer learning," in *Proceedings of IJCAI-07*, 2007, pp. 2599 – 2604.

[11] M. Taylor and P. Stone, "Cross-domain transfer for reinforcement learning," in *Proceedings of ICML-07*, 2007, pp. 879 – 886.

[12] J. Devis and P. Domingos, "Deep transfer via second-order markov logic," in *Proceedings of ICML-09*, 2009.

[13] L. Getoor and B. Taskar, Eds., *Introduction to Statistical Relational Learning*. The MIT Press, 2007.

[14] L. De Raedt, P. Frasconi, K. Kersting, and S. Muggleton, Eds., *Probabilistic Inductive Logic Programming*, ser. Lecture Notes in Computer Science. Springer, 2008, vol. 4911.

[15] K. Yu and W. Chu, "Gaussian process models for link analysis and transfer learning," in *Neural Information Processing Systems*, 2007.

[16] R. Silva, W. Chu, and Z. Ghahramani, "Hidden common cause relations in relational learning," in *Neural Information Processing Systems*, 2007.

[17] Z. Xu, K. Kersting, and V. Tresp, "Multi-relational learning with gaussian processes," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-09)*, C. Boutilier, Ed., 2009, to appear.

[18] W. Cohen and V. Rocha de Carvalho, "Stacked sequential learning," in *Proceedings of IJCAI-05*, 2005, pp. 671 – 676.

[19] Z. Kou and W. Cohen, "Stacked graphical models for efficient inference in markov random fields," in *Proceedings of SDM-07*, 2007.

[20] Z. Kou, W. Cohen, and R. Murphy, "A stacked graphical model for associating sub-images with sub-captions," in *Proceedings of Pacific Symposium on Biocomputing (PSB-07)*, 2007, pp. 257 – 268.

[21] S. Chakrabarti, B. Dom, and P. Indyk, "Enhanced hypertext categorization using hyperlinks," in *Proceedings of SIGMOD-98*, 1998.

[22] S. Macskassy and F. Provost, "Classification in networked data: A toolkit and a univariate case study," *Journal of Machine Learning Research (JMLR)*, vol. 8, pp. 935–983, 2007.

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

[24] W. Chu, V. Sindhwani, Z. Ghahramani, and S. Keerthi, "Relational learning with gaussian processes," in *Neural Information Processing Systems*, 2006.

[25] K. Yu, W. Chu, S. Yu, V. Tresp, and Z. Xu, "Stochastic relational models for discriminative link prediction," in *Neural Information Processing Systems*, 2006.

[26] A. J. Smola and I. Kondor, "Kernels and regularization on graphs," in *Annual Conference on Computational Learning Theory*, 2003.

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