Focused Multi-task Learning Using Gaussian Processes

Gayle Leen[1,2**], Jaakko Peltonen[1,2], and Samuel Kaski[1,2,3]

[1] Aalto University School of Science, Department of Information and Computer Science
[2] Helsinki Institute of Information Technology HIIT
[3] University of Helsinki, Department of Computer Science
gayle.leen@decode.is, jaakko.peltonen@tkk.fi, samuel.kaski@aalto.fi

Abstract. Given a learning task for a data set, learning it together with related tasks (data sets) can improve performance. Gaussian process models have been applied to such multi-task learning scenarios, based on joint priors for functions underlying the tasks. In previous Gaussian process approaches, all tasks have been assumed to be of equal importance, whereas in transfer learning the goal is *asymmetric*: to enhance performance on a target task given all other tasks. In both settings, transfer learning and joint modelling, *negative transfer* is a key problem: performance may actually decrease if the tasks are not related closely enough. In this paper, we propose a Gaussian process model for the asymmetric setting, which learns to "explain away" non-related variation in the additional tasks, in order to focus on improving performance on the target task. In experiments, our model improves performance compared to single-task learning, symmetric multi-task learning using hierarchical Dirichlet processes, and transfer learning based on predictive structure learning.

Keywords: Gaussian processes, multi-task learning, asymmetric setting, negative transfer

1 Introduction

Analysis of brain signals is a prime example of data analysis tasks which could benefit from successful transfer learning. Functional neuroimaging studies typically suffer from the "small n, large p" problem: the number of subjects n is small but the dimensionality of the data p for each subject, obtained by methods such as functional Magnetic Resonance Imaging (fMRI) is huge. In patient studies of a brain disorder, there are practical limitations on how many patients can be accessed and measured, and in experimental neuroscience the problem is that the larger the number of replications and variants needed, the less new neuroscience can be done. Moreover, when generalizing across subjects, the brain physiology and function are sufficiently similar that different brains can be matched, but the

** now at deCODE Genetics, Reykjavik

matching is only approximate. We study a classification task in which the goal is to predict the stimulus given brain measurements of a certain user, utilizing the measurements of other users on the same and different stimuli.

The task is more general, however. It has been shown that transferring knowledge between several potentially related learning tasks has improved performance. This scenario, termed multi-task learning [6] or transfer learning [14], has gained considerable attention in the machine learning community in recent years (see [11] for a recent review). Sharing statistical strength between tasks can potentially compensate for having very few samples in the desired learning task, and can make the inference more robust to noise.

1.1 Symmetric and Asymmetric Multi-task Learning

Transfer of knowledge between different tasks is useful only when the tasks are related; if tasks are unrelated, *negative transfer* can occur, meaning that the transfer distorts the model learned for a target task rather than providing additional statistical strength. Therefore, a crucial part of multi-task learning algorithms lies in the modelling of task relatedness, through the specification and the learning of the dependency structure between tasks.

In general, existing multi-task learning approaches use a symmetric dependency structure between tasks. This type of set-up, which we term *symmetric multi-task learning*, assumes that all tasks are of equal importance. The set of related tasks is learned jointly, with the aim of improving over learning the tasks separately (the *no transfer* case), averaged over all tasks.

However, a common learning scenario is to learn a specific task (*primary task*), while incorporating knowledge learned through other similar tasks (*secondary tasks*). For instance, in the neuroscience scenario mentioned earlier, we are interested in learning about a specific patient's response to a stimulus, but we can transfer information from other patients' responses to related stimuli to improve learning. This asymmetric case, or *transfer learning*, requires the assumption of an asymmetric dependency structure between tasks. Existing approaches include reweighting-based methods [16, 3, 4] or learning of shared feature spaces. An alternative has been to, in effect, use a symmetric multi-task learning method in an asymmetric mode, by using the model learned from auxiliary tasks as a prior for the target task [9, 12, 17].

Inspired by the Gaussian process (GP) models used earlier for symmetric multi-task learning, we propose a novel and simple dependency structure for asymmetric multi-task learning using GPs. This focuses on learning a target task and learns to avoid negative transfer; this can be done conveniently in the GP formulation, by adding task-specific processes which "explain away" irrelevant properties. At the same time, flexibility of the GP framework is preserved.

2 Dependency Structure in Multi-task Learning with Gaussian Processes

Supervised learning tasks such as classification and regression can be viewed as function approximation problems given the task inputs and targets; accordingly, multi-task learning can be viewed as learning multiple, related functions. The Gaussian process (GP) framework provides a principled and flexible approach for constructing priors over functions. The GP framework has subsequently been applied successfully to multi-task learning problems [20, 5, 1]. A crucial element of these models is the way in which the dependency structure between the multiple functions is encoded through the construction of the covariance function. However, current GP approaches do not address the problem of asymmetric multi-task learning, and only consider symmetric dependency structures, which we review in the following subsection.

2.1 Symmetric Dependency Structure

Suppose that there are N distinct inputs, $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_N]^\top$, and M tasks, such that $y_{t,i}$ is the target for input i in task t. We denote the vector of outputs for task t as $\mathbf{y}^t = [y_1^t, ..., y_N^t]^\top$, and the $N \times M$ vector of outputs for all M tasks, as $\mathbf{y} = [\mathbf{y}_1^\top, ..., \mathbf{y}_M^\top]^\top$. In the GP approach to the problem, it is assumed that there is a latent function underlying each task, $f_1, ..., f_M$. Denoting the latent function evaluated at input i for task t as $f_t(\mathbf{x}_i)$, a (zero mean) GP prior is defined over the latent functions, with a covariance function of the form

$$\langle f_t(\mathbf{x}) f_{t'}(\mathbf{x}') \rangle = k^T(t, t') k^x(\mathbf{x}, \mathbf{x}') \tag{1}$$

where k^T is a covariance function over tasks, specifying the intertask similarities, and k^x is a covariance function over inputs. For regression tasks, the observation model is $y_{i,t} \sim \mathcal{N}(f_t(\mathbf{x}_i), \sigma_t^2)$, where σ_t^2 is the noise variance in task t.

In [5], k^T is defined as a 'free-form' covariance function, where $k^T(i, j) = K^T_{i,j}$ indexes a positive semidefinite intertask similarity matrix K^T. Other methods such as [19] have included a parameterised similarity matrix over task descriptor features, but this could be restrictive in modelling similarities between tasks. These types of priors essentially assume that each of the task latent functions is a linear combination of a further set of latent functions, known as intrinsic correlation models in the geostatistics field (see e.g. [15]). This idea was further generalised in [1] to generating the task latent functions by convolving a further set of latent functions with smoothing kernel functions.

2.2 Predictive Mean for Symmetric Multi-task GP

The predictive mean on a new data point $\mathbf{x}_*$ in task j, for the multi-task GP formulation of [5], is given by

$$\overline{f}_j(\mathbf{x}_*) = (\mathbf{k}_j^T \otimes \mathbf{k}_*^x)^\top \Sigma^{-1} \mathbf{y} \quad \text{where} \quad \Sigma = K^T \otimes k^x(\mathbf{X}, \mathbf{X}) + D \otimes \mathbf{I} \tag{2}$$

where $\mathbf{k}_j^T$ is the jth column of task similarity matrix K^T, $\otimes$ is the Kronecker product, $\mathbf{k}_*^x = [k(\mathbf{x}_*, \mathbf{x}_1), ..., k(\mathbf{x}_*, \mathbf{x}_N)]^\top$ is the vector of covariances between the test input $\mathbf{x}_*$ and the training inputs. The $k^x(\mathbf{X}, \mathbf{X})$ is the matrix of covariance function values between all training input points, and D is an $M \times M$ diagonal matrix where the (j, j)th element is σ_i^2.

To gain intuition into the form of the predictive mean, let us define the $M \times N$ vector $\mathbf{w} = \Sigma^{-1}\mathbf{y}$, and divide it into M blocks of N elements: $\mathbf{w} = [\mathbf{w}_1^\top, ..., \mathbf{w}_M^\top]^\top$. We can then rewrite (2) as

$$\overline{f}_j(\mathbf{x}_*) = \sum_{m=1}^{M} K_{m,j}^T (\mathbf{k}_*^x)^\top \mathbf{w}_m = \sum_{m=1}^{M} K_{m,j}^T \mu_*^m \tag{3}$$

where $\mu_*^m = (\mathbf{k}_*^x)^\top \mathbf{w}_m$ can be interpreted as the posterior mean of the latent function at $\mathbf{x}_*$ for task m, thus (2) is a weighted sum of posterior means for all tasks, and the weights $\{K_{m,i}^T\}_{m=1}^M$ are covariances between task j and all tasks.

Since K^T is positive semidefinite, the sharing of information between tasks is naturally symmetric, and all tasks are treated equally. However, we are interested in an asymmetric setup, where we learn a primary task together with several secondary tasks. Rather than modelling the relationships between secondary tasks, we want to focus on the aspects relevant to learning the primary task.

2.3 Asymmetric Dependency Structure

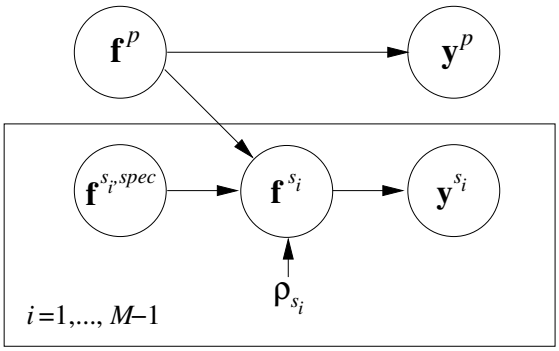


Fig. 1. Graphical model of the focused GP multi-task model, showing the relationship between the function values of the primary and secondary tasks. Parameters of the covariance functions omitted for clarity.

In the previous symmetric learning problem, the tasks were modelled as conditionally independent on a set of M (i.i.d.) underlying functions, which capture the shared structure between all tasks. In this section, we derive an

asymmetric version of a GP framework for multi-task learning, by constraining the secondary tasks to be conditionally independent given the primary task, such that the shared structure between all secondary tasks is due to the primary task.

Similarly to the previous notation, let us denote the inputs to each task as $\mathbf{X}$. Suppose that there is one primary task, with targets $\mathbf{y}^p = [y_1^p, ..., y_N^p]^\top$, with underlying latent function values $\mathbf{f}^p = [f^p(\mathbf{x}_1), ..., f^p(\mathbf{x}_N)]^\top$. Suppose there are $M-1$ secondary tasks, where the targets for the ith secondary task are denoted by $\mathbf{y}^{s_i} = [y_1^{s_i}, ..., y_N^{s_i}]^\top$. The corresponding latent function values are $\mathbf{f}^{s_i} = [f^{s_i}(\mathbf{x}_1), ..., f^{s_i}(\mathbf{x}_N)]^\top$.

We are interested in learning the underlying function f^p for the primary task. Here, potentially related secondary tasks can help to learn f^p; conversely if we know f^p, this could help to learn the functions underlying the secondary tasks $\{f^{s_i}\}$. We can formalise this intuition by examining the GP predictive likelihood on the secondary task function values, after training on the primary task. However, first we need to define a joint prior over the primary and secondary task function values. We start by making the assumption that $\{f^{s_i}\}$ can be decomposed into a 'shared' component (which is shared with the primary task) and a 'specific' component. That is, for the nth input,

$$f^{s_i}(\mathbf{x}_n) = f^{s_i,shared}(\mathbf{x}_n) + f^{s_i,specific}(\mathbf{x}_n) \; . \tag{4}$$

Further we assume that $f^{s_i,shared} = \rho_{s_i} f^p$, that is, the shared component is correlated with the primary task function. This may seem like a restrictive assumption but assuming linear relationships between task functions has been proved to be successful in e.g. [15, 5]. Now we can place a shared prior over each $f^{s_i,shared}$ and f^p. The corresponding graphical model is presented in Figure 1.

Sharing between Primary and Secondary Task Functions. We place a zero mean Gaussian process prior on f^p, with covariance function k^p, such that the prior on the shared function is also a GP, with covariance function

$$\left\langle f^t(\mathbf{x}) f^{t'}(\mathbf{x}') \right\rangle = k^t(t,t') k^p(\mathbf{x},\mathbf{x}') \quad \text{where} \quad k^t(t,t') = \rho_t \rho_{t'} \tag{5}$$

where ρ_t is the correlation of task t with the primary task, and $\rho_p = 1$, and f^t can denote either the primary task or any of the secondary tasks. Denoting the task functions for the $M-1$ secondary tasks as $\mathbf{f}^s = \left[(\mathbf{f}^{s_1})^\top, ..., (\mathbf{f}^{s_{M-1}})^\top\right]^\top$, the joint distribution over the shared function values is given by

$$p(\mathbf{f}^p, \mathbf{f}^{s,shared}) = \mathcal{GP}\left(0, \begin{bmatrix} \mathbf{K}_{pp} & \mathbf{K}_{sp}^\top \\ \mathbf{K}_{sp} & \mathbf{K}_{ss} \end{bmatrix}\right) \tag{6}$$

where $\mathbf{K}_{pp}$ is the matrix of covariance function values from (5) between the primary task points, $\mathbf{K}_{sp}$ evaluated between secondary and primary, and $\mathbf{K}_{ss}$ between secondary task inputs. Given the primary task function values, we can then derive the predictive distribution on the shared components of the secondary tasks using the standard GP equations:

$$p(\mathbf{f}^{s,shared} \mid \mathbf{f}^p) = \mathcal{GP}\left(\mathbf{K}_{sp}\mathbf{K}_{pp}^{-1}\mathbf{f}^p, \Lambda\right) \tag{7}$$

where Λ is a diagonal matrix whose elements are given by the diagonal of $\mathbf{K}_{ss} - \mathbf{K}_{sp}\mathbf{K}_{pp}^{-1}\mathbf{K}_{sp}^{\top}$. This approximation allows us to make a reduced rank approximation, and offers a computationally efficient solution to jointly learning the covariance matrix across a large number of input points.

An interpretation of equation (7) is the posterior distribution of f^p (the primary task function) after observing the primary task function values $\mathbf{f}^p$, evaluated at all the secondary task inputs. This differs slightly from the standard GP predictive equations in that the posterior mean for each secondary task s is weighted by ρ_s, which models the correlation with the primary task. To illustrate this, for secondary task l, the posterior mean $\bar{f}^{1,shared}$ given $\mathbf{f}^p$:

$$\bar{f}^{1,shared} = \rho_l k^p(\mathbf{X}_l, \mathbf{X}_p) k^p(\mathbf{X}_p, \mathbf{X}_p)^{-1} \mathbf{f}^p = \rho_l \mu_l^p$$

where we have used the notation: $\mathbf{X}_i$ is the set of input points for task i, and μ_l^p is the posterior mean given covariance function k^p and observations $\mathbf{f}^p$, evaluated at $\mathbf{X}_l$. Learning ρ_s during training can help to avoid negative transfer from secondary to primary task.

Explaining Away Secondary Task-Specific Variation. We define the covariance function over $\mathbf{f}^{s,specific}$ to be block diagonal in $[\mathbf{K}_1^{spec}, ..., \mathbf{K}_{M-1}^{spec}]$ with respect to the tasks. These covariance functions have parameters specific to each task: $\mathbf{f}^{s,specific} \sim \mathcal{GP}(0, \mathbf{K}^{spec})$. This creates flexible models for the secondary tasks, which can 'explain away' variation that is specific to a secondary task, and unshared with the primary task. Since the primary task function values are unknown, rather than estimating them directly we integrate over them:

$$p(\mathbf{f}^s) = \mathcal{GP}\left(0, \mathbf{K}_{sp}\mathbf{K}_{pp}^{-1}\mathbf{K}_{sp}^{\top} + \Lambda + \mathbf{K}^{spec}\right) . \tag{8}$$

Putting everything together, the resulting prior on all the task functions is

$$p(\mathbf{f}^p, \mathbf{f}^s) = \mathcal{GP}\left(0, \begin{bmatrix} \mathbf{K}_{pp} & \mathbf{K}_{sp}^{\top} \\ \mathbf{K}_{sp} & \mathbf{K}_{sp}\mathbf{K}_{pp}^{-1}\mathbf{K}_{sp}^{\top} + \Lambda + \mathbf{K}^{spec} \end{bmatrix}\right) \tag{9}$$

2.4 Hyperparameter Learning

We can learn the hyperparameters of our model in (9) by optimising the marginal log likelihood with respect to the hyperparameters of the covariance functions, the task similarity vector $[\rho_{s_1}, ..., \rho_{s_{M-1}}]$, and the parameters of the observation model, given the inputs $\mathbf{x}$ and targets y. For regression, the observation model is $y_{i,t} \sim \mathcal{N}(f_t(\mathbf{x}_i), \sigma_t^2)$, where σ_t^2 is the noise variance in task t, and for classification we use a probit noise model $p(y_{i,t} \mid f_{i,t} = \Phi(y_{i,t}(f_{i,t} + b))$, where Φ is the cumulative distribution function for a standard Gaussian $\mathcal{N}(0, 1)$, and b is a bias parameter. For the binary classification experiments in Section 5.2, we make an approximation to the model likelihood using Expectation Propagation [10].

3 Related Work and Discussion

In our focused multi-task GP, the pseudo-input locations are fixed as the inputs to the primary task, such that they can explain the shared variation between the primary and secondary tasks, and also between the secondary tasks. The sparse GP method in [13] bears similarities to our model. This parameterises the covariance function of a GP by learning a set of pseudo-input locations. In that model, the pseudo-inputs summarise the variation of the data through assuming that the function values are conditionally independent given the pseudo-inputs.

Recently there has been interest in asymmetrical GP multi-task learning [7], where generalisation errors for the multi-task GP of [5] were derived for an asymmetrical multi-task case, with one primary and one secondary task. However, this work did not derive a new model for asymmetric multi-task learning, and focused on analysing the symmetric model.

The asymmetric dependency structure that we have presented uses a simple idea to bias the model to learning the underlying function for the primary task, by decomposing the underlying task functions for the secondary tasks as 'shared' and 'specific' components. The shared components are from a joint GP prior with the primary task function. These are conditioned on the primary task function values (7) such that this biases the shared variation between tasks to be due to the primary task function, and a task specific weight, which is learned during training. We additionally assume that the each of the secondary task functions can also be explained by a process specific to it, by defining a block diagonal covariance structure over the secondary tasks. This allows the model to 'explain away' secondary task specific variation and focus the model on learning the primary task.

In this first paper we make the simplifying assumption that the task of interest is entirely composed of the shared function, and that there are no other strong shared functions between other tasks. This model already proves useful in a challenging fMRI task, demonstrating that the idea of asymmetric modelling with explaining-away yields useful results, and it can be extended to more general asymmetric modelling in later stages.

In brief, if there is reason to suspect detrimental shared variation between other tasks, one can add additional GP functions which is shared between other tasks but not with the primary task. The overall model can then learn which shared function is a better explanation. As the number of tasks increases, the number of possible sharing configurations increases (shared functions between 2,3,...,M tasks) and the complexity of the model quickly increases. This will be studied in further work.

4 Examining the Generalisation Error for Asymmetric and Symmetric Models

To examine the effect of the processes that are specific to a secondary task, we look at the generalisation error on the primary task for the asymmetric two tasks

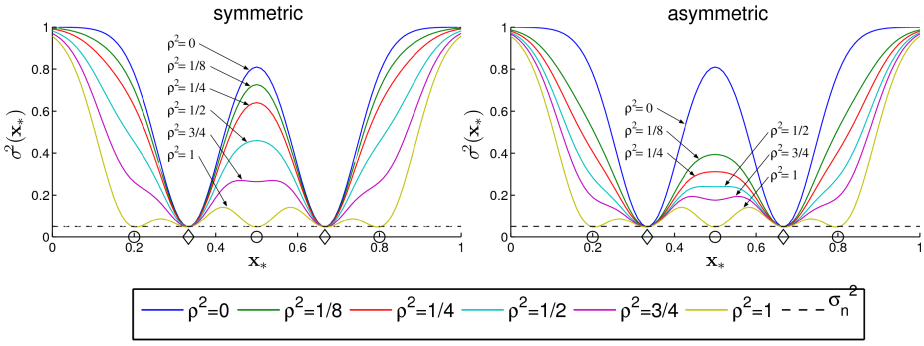


Fig. 2. The posterior variances for the test locations $\mathbf{x}_* \in [0, 1]$ given training points from the primary task ($\mathbf{X}_P = [1/3 \;\; 2/3]$, plotted as $\diamond$) and secondary task ($\mathbf{X}_S = [1/5 \;\; 1/2 \;\; 4/5]$, plotted as $\circ$) for the symmetric case (top) and the asymmetric case (bottom). Each plot uses corresponding values of ρ^2 (see legend).

case in a similar manner to [7]. We investigate the influence of ρ, the degree of "relatedness" between the two tasks. Suppose that we have training inputs $\mathbf{X}_P$ for the primary task, and $\mathbf{X}_S$ for the secondary task. The covariance matrices C_{sym} and C_{asym}, for the symmetric and asymmetric cases respectively, of the noisy training data are given by:

Symmetric case

$$C_{\text{sym}}(\rho) = K^{\text{sym}}(\rho) + \sigma_n^2\mathbf{I} \quad \text{where} \quad K^{\text{sym}}(\rho) = \begin{pmatrix} K^p_{PP} & \rho K^p_{PS} \\ \rho K^p_{SP} & K^p_{SS} \end{pmatrix} \tag{10}$$

Asymmetric case

$$\begin{aligned} C_{\text{asym}}(\rho) &= K^{\text{asym}}(\rho) + \sigma_n^2\mathbf{I} \\ \text{where} \;\; K^{\text{asym}}(\rho) &= \begin{pmatrix} K^p_{PP} & \rho K^p_{PS} \\ \rho K^p_{SP} & \rho^2 K^p_{SS} + (1-\rho^2)K^s_{SS} \end{pmatrix} \end{aligned} \tag{11}$$

where we have used the notation K^p_{AB} to denote the matrix of covariance values, due to k^p, evaluated between $\mathbf{X}_A$ and $\mathbf{X}_B$. For the asymmetric case, the covariance matrix for the secondary task comes from the 'shared' covariance function k^p with the primary task, and a 'specific' covariance function k^s. The relationship between the primary and secondary tasks due to the ρ's comes directly from (1) and (5) for the symmetric and asymmetric cases respectively.

4.1 Generalisation Error for a Test Point $\mathbf{x}_*$

If the GP prior is correctly specified, then the posterior variance for a new test point $\mathbf{x}_*$ for the primary task (due to the noise free f^p) is also the generalisation error for $\mathbf{x}_*$. The posterior variance at $\mathbf{x}_*$ for the primary task is:

$$\textbf{Symmetric case:} \quad \sigma^2_{\text{sym}}(\mathbf{x}_*, \rho) = k_{**} - \mathbf{k}_*^\top C_{\text{sym}}(\rho)^{-1}\mathbf{k}_* \tag{12}$$

$$\textbf{Asymmetric case:} \quad \sigma^2_{\text{asym}}(\mathbf{x}_*, \rho) = k_{**} - \mathbf{k}_*^\top C_{\text{asym}}(\rho)^{-1}\mathbf{k}_* \tag{13}$$

where k_{**} is the prior variance at $\mathbf{x}_*$, $k^p(\mathbf{x}_*, \mathbf{x}_*)$, and $\mathbf{k}_*^\top = (k^p(\mathbf{x}_*, \mathbf{X}_p) \;\; \rho k^p(\mathbf{x}_*, \mathbf{X}_s))$ We note that the target values y do not affect the posterior variance at the test locations, and have omitted the dependence on $\mathbf{X}_P$, $\mathbf{X}_S$ and σ_n^2 in the notation for $\sigma^2_{\text{sym}}(\mathbf{x}_*, \rho), \sigma^2_{\text{asym}}(\mathbf{x}_*, \rho)$ for clarity.

To illustrate the difference between the symmetric and asymmetric cases, we plot the posterior variances as a function of $\mathbf{x}_*$ in Figure 2, given two observations for the primary task, and three observations of the secondary task (see figure for more details). Following the setup in [7], we use a squared exponential covariance function with lengthscale 0.11 for k^p, noise variance $\sigma_n^2 = 0.05$, and, for the asymmetric setup, a squared exponential covariance function with lengthscale 1 for k^s.

Each plot contains 6 curves corresponding to $\rho^2 = [0, 1/8, 1/4, 1/2, 3/4, 1]$, and the dashed line shows the prior noise variance. The training points from the primary task ($\diamond$) create a depression that reaches the prior noise variance for all the curves. However, the depression created by the training points for the secondary task ($\circ$) depends on ρ. For the single task learning case ($\rho = 0$), there is no knowledge transferred from the secondary task. As ρ increases, the generalisation error at the secondary task test points decreases. For the intermediate ρ^2 values (i.e. not 0 or 1 (full correlation)), our asymmetric model gives a smaller posterior variance than the symmetric model at secondary task locations, and therefore suggests better generalisation error.

4.2 Intuition about the Generalisation Errors

Given the illustrative example in the previous section, we sketch the relationship between the generalisation errors for the primary and secondary tasks:

$$\sigma^2_{\text{asym}}(\mathbf{x}_*, \rho) \leq \sigma^2_{\text{sym}}(\mathbf{x}_*, \rho) \tag{14}$$

We show this by considering the covariance matrix at the secondary task points, conditioned on the primary task points. This represents the residual uncertainty about the secondary task points, given that we know the primary task points. Denoting this quantity as $A(\rho)$:

$$A(\rho)_{\text{sym}} = K^p_{SS} + \sigma_n^2\mathbf{I} - \rho^2 K^p_{SP}(K^p_{PP} + \sigma_n^2\mathbf{I})^{-1}K^p_{PS} \tag{15}$$

$$A(\rho)_{\text{asym}} = \rho^2 K^p_{SS} + (1-\rho^2)K^s_{SS} + \sigma_n^2\mathbf{I} - \rho^2 K^p_{SP}(K^p_{PP} + \sigma_n^2\mathbf{I})^{-1}K^p_{PS} \tag{16}$$

If $A(\rho)_{\text{asym}} \preceq A(\rho)_{\text{sym}}$ then:

$$\begin{aligned}
A(\rho)^{-1}_{\text{asym}} &\succeq A(\rho)^{-1}_{\text{sym}} \\
\mathbf{v}(\rho)^\top A(\rho)^{-1}_{\text{asym}}\mathbf{v}(\rho) &\geq \mathbf{v}(\rho)^\top A(\rho)^{-1}_{\text{sym}}\mathbf{v}(\rho) \\
k_{**} - k^p(\mathbf{x}_*, \mathbf{X}_P)(K^p_{PP} + \sigma_n^2\mathbf{I})^{-1}k^p(\mathbf{x}_*, \mathbf{X}_P) - \mathbf{v}(\rho)^\top A(\rho)^{-1}_{\text{asym}}\mathbf{v}(\rho) & \\
\leq k_{**} - k^p(\mathbf{x}_*, \mathbf{X}_P)(K^p_{PP} + \sigma_n^2\mathbf{I})^{-1}k^p(\mathbf{x}_*, \mathbf{X}_P) - \mathbf{v}(\rho)^\top A(\rho)^{-1}_{\text{sym}}\mathbf{v}(\rho) & \\
\sigma^2_{\text{asym}}(\mathbf{x}_*, \rho) &\leq \sigma^2_{\text{sym}}(\mathbf{x}_*, \rho)
\end{aligned} \tag{17}$$

where we have used the Banachiewicz inversion formula to evaluate the matrix inversions in (12) and (13), and we have defined $\mathbf{v}(\rho) = \rho(k^p(\mathbf{X}_S, \mathbf{x}_*) - K^p_{SP}(K^p_{PP} + \sigma_n^2\mathbf{I})^{-1}k^p(\mathbf{X}_P, \mathbf{x}_*))$

The asymmetric model has more flexibility than the symmetric model in the modelling of the secondary task, since it uses both f^p and f^s, rather than just f^p. We expect that $A(\rho)$ for the asymmetric version would be smaller than for the symmetric since the additional flexibility should allow more accurate modelling of the covariances between the secondary task points, and hence the asymmetric generalisation error should be smaller than the symmetric.

5 Experiments

In this section, we demonstrate the performance of the focused multi-task GP model on a synthetic regression problem, and compare it with alternative models on an asymmetric multi-task classification problem on fMRI data. In all experiments, we use squared exponential covariance functions with automatic relevance determination (ARD) prior: $k(\mathbf{x}, \mathbf{x}') = \sigma_s^2 \exp(-\frac{1}{2}\sum_d (\mathbf{x}_d - \mathbf{x}'_d)^2/l_d^2)$, where σ_s^2 is the overall scale and l_d is the lengthscale for the dth input dimension, initialized to 1. This prior is used for both primary and secondary task functions.

5.1 Synthetic Data

Synthetic data are generated as follows (see Fig. 3. All the functions are functions of the same input $\mathbf{x}$, 100 samples evenly spaced on the interval [-5, 5]. The primary task function is generated from $\mathbf{f}_p \sim \mathcal{GP}(0, \mathbf{K}_p)$, where the kernel function is squared exponential with length scale 1. The secondary task functions are generated according to $f_s^m \sim \mathcal{GP}(\alpha_m \mathbf{f}_p, \beta_m \mathbf{K}_s^m)$. Each specific kernel function is squared exponential with lengthscale 1, and α_m is drawn at random from $\mathcal{N}(0, 1)$, β_m at random from $[0, 1]$. We assume a Gaussian observation noise model, since this is a regression problem.

We remove 50 samples from the primary task (see Fig. 4b), and use them as test data. We train the model with different numbers of secondary tasks, ranging from 0 (single task learning) to 24. We repeat the procedure 10 times, randomly drawing the secondary task functions for each run. Figure 4 (b) shows the mean of the posterior distribution (black) over the primary task function for one of the runs, for different numbers of secondary tasks. We also plot the true underlying primary function (blue line), showing that the model can predict the missing part of the primary task function by transferring information from secondary tasks. Figure 4 (a) shows that the mean squared error on the test set decreases as the number of secondary tasks increases.

5.2 fMRI Data

We evaluate the performance of our model on fMRI data, taken from [8]. Six healthy young adults participated in two identical sessions, in which they received a continuous 8-min sequence comprising of auditory, visual and tactile

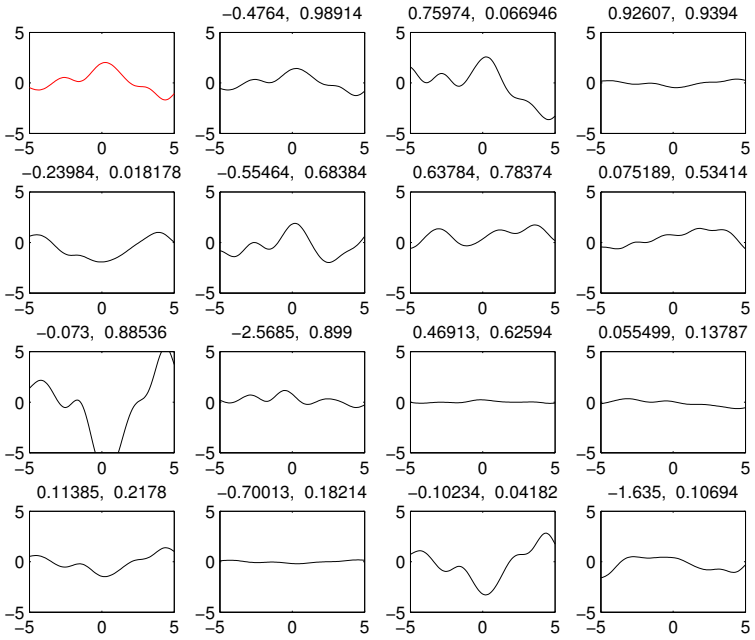
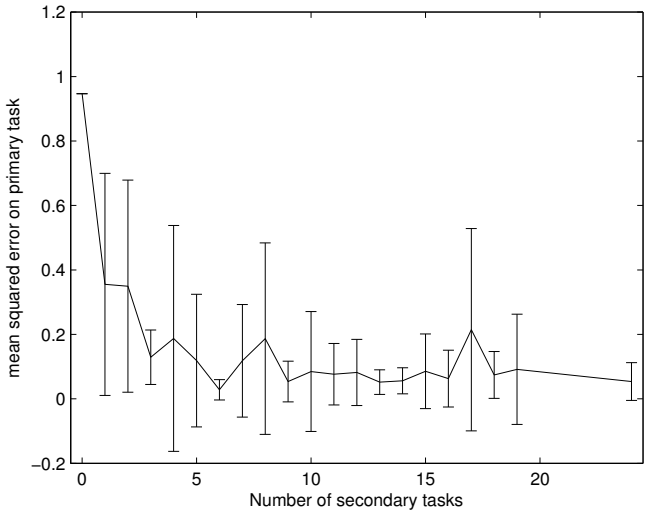


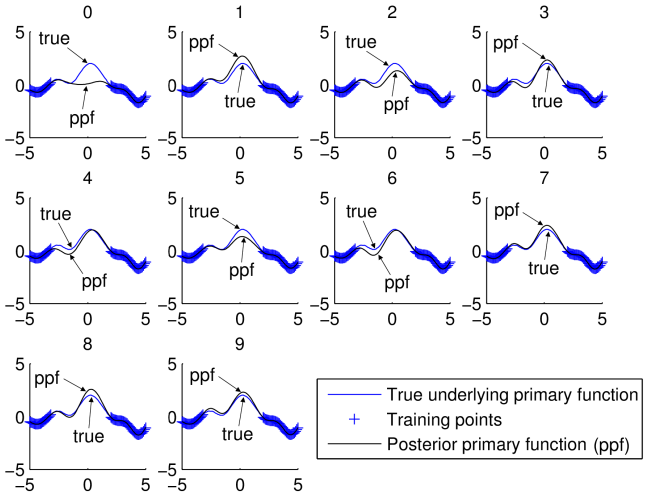
Fig. 3. Synthetic data experiment: experiment setup. We show the functions underlying the generated data: the primary task function (top left, red) and 15 examples of secondary task functions (black). The weights of the shared and specific functions for the secondary tasks are given above each plot.

stimuli in blocks of 6×33s. The stimuli of different senses never overlapped. Whole-head volumes were acquired with a Signa VH/i 3.0 T MRI scanner (General Electric, Milwaukee, WI) using a gradient EPI sequence (TR = 3 s, TE = 32 ms, FOV = 20 cm, flip = 90^o, $64 \times 64 \times 44$ voxels with resolution $3 \times 3 \times 3\text{mm}^3$). In each session, 165 volumes were recorded with the 4 first time points excluded from further analysis. Preprocessing of the fMRI data included realignment, normalization with skull stripping, and smoothing. For additional details on the measurements and applied preprocessing, see [18]. After preprocessing, the dimensionality was reduced to 40 by spatial independent component analysis (ICA) that identified spatial brain activation patterns related to various aspects of the stimuli. For each adult, the resulting data is 161 sets of ICA features (40 dimensional), which can be classified according to one of 6 stimuli ('touch', 'auditory' (tones, history, instruction), 'visual' (faces, hands, buildings)).

We consider the task of predicting whether a subject is reacting to a particular stimulus, 'touch', given the fMRI data. We aim to improve the learning of this primary task by learning it in conjunction with other, related tasks from the



(a)



(b)

Fig. 4. Synthetic data experiment: results of learning with the proposed asymmetric multi-task Gaussian process model. (a) Mean squared error on the primary task test set, over 10 runs, for different numbers of secondary tasks, error bars represent ±1 s.d. (b) Posterior distribution over the primary task function for different numbers of secondary tasks (given above each plot).

other subjects. This can be formulated as 6 one-against-all classification tasks in an asymmetric multi-task setup (see Table 1). For each subject the fMRI measurements were done in two separate sessions; in the experiments we use the first session as training data and the second session as test data.

Table 1. Asymmetrical multi-task set up for fMRI data study

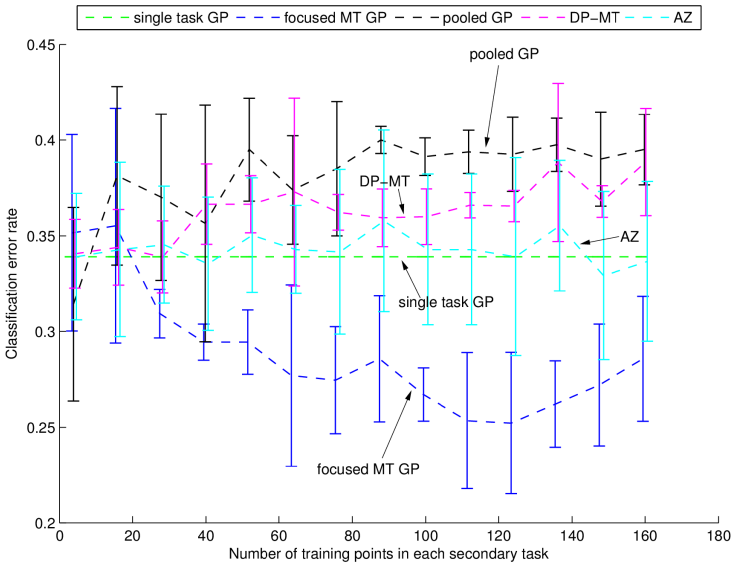
Subject	Classification Task
1 (primary)	'touch' against all
2 (secondary)	'touch' against all
3 (secondary)	'touch' against all
4 (secondary)	'touch' against all
5 (secondary)	'auditory' (instruction) against all
6 (secondary)	'visual' (buildings) against all

We compare the focused multi-task learning approach ('focused MT-GP') with four reference models. The first baseline model is single task learning using GP classification ('single task GP'), trained only on the samples of the primary task. The second ('pooled GP') learns a GP classification model from the training examples from all tasks (i.e. treating all data as a single task) For 'pooled GP' we use a sparse approximation when the number of training examples > 300, using 30 pseudo-inputs. We also compare to two state-of-the-art methods, one developed for transfer learning and the other multi-task learning: the predictive structure learning method of [2] ('AZ'), and the symmetric multi-task learning with Dirichlet process priors method ('DP-MT') from [17]. For the 'AZ' method, the we fix the dimension of the shared predictive structure heuristically to $h = 26$, after performing PCA across all the training samples (primary and secondary) and find the dimension of the subspace that explains 80% of the variance.

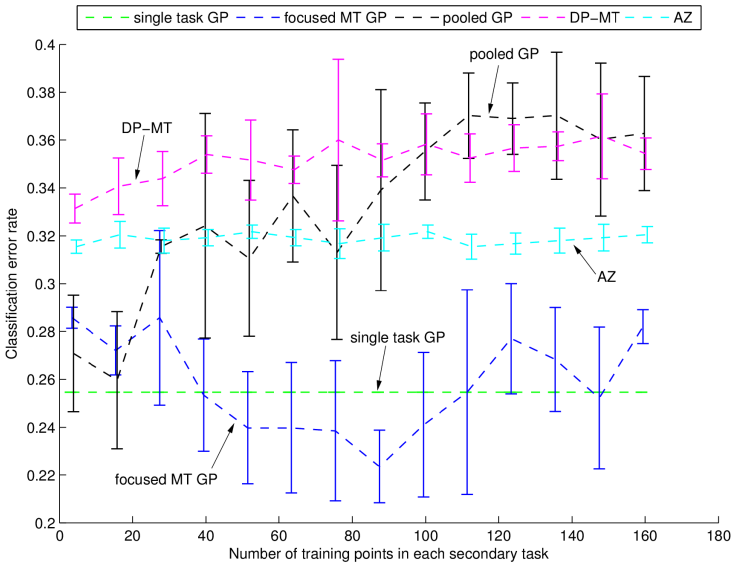
We evaluate the methods using a fixed number of training examples in the primary task (64 and 161), while varying the number of training examples in each secondary task (ranging from 4 to 160)), over 5 repetitions. Due to the class imbalance in the data, when randomly picking a subset of secondary training task examples, we ensure that there is at least one positive and one negative example. For the GP-based methods, we also fix the bias parameter $b = \Phi^{-1}(r)$, where r is the ratio of positive samples to negative samples in the training data.

Figure 5 displays the classification error on the test set for the primary task, over different numbers of training examples for the secondary tasks, for 64 training examples in the primary task (a) and 161 (*i.e.*, all available training examples for the primary task) in (b).

Pooling of samples seems to always be a bad choice on this data and, somewhat surprisingly, DP-MT does not work well either. Both work only roughly equally to single-task learning for small numbers of secondary task data and the performance worsens as amount of secondary data increases. Hence it seems



(a) Number of primary task training examples = 64



(b) Number of primary task training examples = 161

Fig. 5. Classification error on test set for primary task, against number of training examples in each secondary task for different primary task training set size (a: small, b: larger)

that the secondary data here differs from primary data to the extent of causing negative transfer. AZ seems to work better but at most on the same level as single task learning. More work would be needed for model selection, however, which might improve performance.

Focused MT-GP seems able to leverage on the secondary tasks, clearly outperfoming others including single task learning when the amount of data in the primary task is small. Multitask learning is most relevant when the primary task has little data; Focused MT-GP performs well in this scenario. When primary data has more data single task learning improves rapidly, although in Figure 5 Focused MT-GP still outperforms it. Focused MT-GP seems to need more than a few samples in the secondary tasks in order to perform well; the explanation is probably that for this data it is hard to distinguish between useful and negative transfer, and more data is needed to make the choice. Bad performance of pooling and symmetric multi task approaches supports this interpretation.

6 Conclusion

We derived a multi-task Gaussian process learning method, the 'focused multi-task GP', designed for asymmetrical multi-task learning scenarios, to facilitate improved learning on a primary task through the transfer of relevant information from a set of potentially related secondary tasks. The novel dependency structure was formulated based on the GP predictive distribution over the secondary tasks given the primary task, and constraining the secondary tasks to be conditionally independent. After observing the primary task, the primary task function can be used to predict a part of each secondary task, depending on the degree of task relatedness, which is learned during the optimisation. The model also permits each secondary task to have its own task-specific variation which is unshared with the primary task, and this flexibility should cause the model to focus on modelling the primary task function well. We demonstrated the model on synthetic data and an asymmetrical multi-task learning problem with fMRI data, and showed improved performance over baseline approaches, and a state of the art transfer learning and multi-task learning method.

Acknowledgments. The authors belong to the Adaptive Informatics Research Centre, a national CoE of the Academy of Finland. JP was supported by the Academy of Finland, decision number 123983. This work was also supported in part by the PASCAL2 Network of Excellence, ICT 216886, and by the Tekes Multibio project.

References

1. Alvarez, M., Lawrence, N.D.: Sparse convolved Gaussian processes for multioutput regression. In: Advances in Neural Information Processing Systems. vol. 21, pp. 57–64 (2009)

2. Ando, R.K., Zhang, T.: A framework for learning predictive structures from multiple tasks and unlabeled data. Journal of Machine Learning Research 6, 1817–1853 (2005)
3. Bickel, S., Bogojeska, J., Lengauer, T., Scheffer, T.: Multi-task learning for HIV therapy screening. In: McCallum, A., Roweis, S. (eds.) Proceedings of the 25th Annual International Conference on Machine Learning (ICML 2008), pp. 56–63. Omnipress (2008)
4. Bickel, S., Sawade, C., Scheffer, T.: Transfer learning by distribution matching for targeted advertising. In: Koller, D., Schuurmans, D., Bengio, Y., Bottou, L. (eds.) Advances in Neural Information Processing Systems 21. pp. 145–152 (2009)
5. Bonilla, E.V., Chai, K.M.A., Williams, C.K.I.: Multi-task Gaussian Process Prediction. In: Neural Information Processing Systems (2008)
6. Caruana, R.: Multitask learning. Machine Learning 28, 41–75 (1997)
7. Chai, K.M.A.: Generalization errors and learning curves for regression with multi-task gaussian processes. In: Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C.K.I., Culotta, A. (eds.) Advances in Neural Information Processing Systems 22. pp. 279–287 (2009)
8. Malinen, S., Hlushchuk, Y., Hari, R.: Towards natural stimulation in fMRI - issues of data analysis. Neuroimage 35(1), 131– 139 (2007)
9. Marx, Z., Rosenstein, M.T., Kaelbling, L.P.: Transfer learning with an ensemble of background tasks. In: Inductive Transfer: 10 Years Later, NIPS 2005 workshop (2005)
10. Minka, T.: Expectation Propagation for approximative Bayesian inference. In: Breese, J.S., Koller, D. (eds.) Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence. pp. 362–369 (2001)
11. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering (in press)
12. Raina, R., Ng, A.Y., Koller, D.: Transfer learning by constructing informative priors. In: Inductive Transfer: 10 Years Later, NIPS 2005 workshop (2005)
13. Snelson, E., Ghahramani, Z.: Sparse Gaussian Processes using Pseudo-inputs. In: Advances in Neural Information Processing Systems 18 (2006)
14. Thrun, S.: Is learning the n-th thing any easier than learning the first? Advances in Neural Information Processing Systems 8 (1996)
15. Wackernagel, H.: Cokriging versus kriging in regionalized multivariate data analysis. Geoderma 62, 83 – 92 (1994)
16. Wu, P., Dietterich, T.G.: Improving SVM accuracy by training on auxiliary data sources. In: Greiner, R., Schuurmans, D. (eds.) Proceedings of the 21st International Conference on Machine Learning (ICML 2004), pp. 871–878. Omnipress, Madison, WI (2004)
17. Xue, Y., Liao, X., Carin, L.: Multi-Task Learning for Classification with Dirichlet Process Priors. Journal of Machine Learning Research 8, 35–63 (2007)
18. Ylipaavalniemi, J., Savia, E., Malinen, S., Hari, R., Vigário, R., Kaski, S.: Dependencies between stimuli and spatially independent fMRI sources: Towards brain correlates of natural stimuli. Neuroimage 48, 176–185 (2009)
19. Yu, K., Chu, W., Yu, S., Tresp, V., Z, X.: Stochastic Relational Models for Discriminative Link Prediction. In: Advances in Neural Information Processing Systems 19 (2007)
20. Yu, K., Tresp, V.: Learning to learn and collaborative filtering. In: Inductive Transfer: 10 Years Later, NIPS 2005 workshop (2005)