Akademie věd České republiky
Ústav teorie informace a automatizace, v.v.i.

Academy of Sciences of the Czech Republic
Institute of Information Theory and Automation

RESEARCH REPORT

Shane Nugent, Anthony Quinn

Transferring Improved Local Kernel Design in Multi-Source Bayesian Transfer Learning, with an application in Air Pollution Monitoring in India

No. 2392 2021

GAČR 18-15970S

ÚTIA AV ČR, P.O.Box 18, 182 08 Prague, Czech Republic
Tel: (+420)266052422, Fax: (+420)286890378, Url: http://www.utia.cas.cz, E-mail: utia@utia.cas.cz

Any opinions and conclusions expressed in this report are those of the authors and do not necessarily represent the views of the Institute.

Abstract

Existing frameworks for multi-task learning [1],[2] often rely on completely modelled relationships between tasks, which may not be available. Recent work [3], [4] has been undertaken on approaches to fully probabilistic methods for transfer learning between two Gaussian Process (GP) tasks. There, the target algorithm accepts source knowledge in the form of a probabilistic prior from a source algorithm, without requiring the target to model their interaction with the source. These strategies have offered robust improvements on current state of the art algorithms, such as the Intrinsic Coregionalization Model.

The Bayesian Transfer Learning algorithm proposed in [4], was found to provide robust, positive transfer. This algorithm was then extended to accommodate knowledge transfer from multiple source modellers [5]. Improved predictive performance was observed from increases in the number of sources.

This report reviews the multi-source transfer findings in [5] and applies it to a real world problem of pollution modelling in India, using public-domain data.

Contents

1 Introduction

The purpose of this report is to explore multi-source Bayesian Transfer Learning (BTL), using expert source knowledge to improve the performance of a non-parametric target. The source and target learning tasks adopt the Gaussian Process Regression model [6]. We apply this to a pollution modelling problem where the target task of estimating PM2.5 levels at hold-out locations in India are to be improved by knowledge from expert source modellers of other pollutants, namely CO, NO_2, and SO_2.

In the BTL framework, the target becomes a global modeller of both the source and target processes. Complete modelling approaches process raw, source observations or their statistics. Conversely, the target global modeller in the BTL framework takes an incompletely modelled form which is conditioned on *probabilistic* source knowledge, computed and transferred from the locally modelled source. This conditioning on the source's probability distribution, rather than on the source's observations, is intrinsic to the Bayesian Transfer Learning approach.

We can define an expert model very loosely as one which can consistently provide a high predictive performance under varying conditions and has some knowledge of the nature of the underlying synthesis model or some privileged access to the local data. We will first explore a common parametric machine learning approach, the k-Nearest Neighbours (kNN)[7] algorithm, and then we will look at how we can construct an expert Gaussian Process Regression model through exploration of various covariance kernel structures.

Once we have built our expert models for each source pollutant, CO, NO_2, and SO_2, we want to see how best to transfer knowledge from these source tasks to the target PM2.5 task. We will evaluate both the Intrinsic Coregionalization Model (ICM)[8], which represents the state of the art, as well as the Bayesian Transfer Learning approach. The success of these two approaches will be judged on their ability to provide increased predictive performance over the isolated PM2.5 target model.

In this application, ICM processes four raw data channels, (CO, NO_2, SO_2, and PM2.5), being a standard multivariate inference approach. In the BTL framework, however, the target only processes its own raw data channel, PM2.5, while processing probabilistic knowledge representations of the three source channels. The BTL approach does not require transfer and processing of the raw source data. Despite this compression, we will show that BTL provides positive and robust transfer, outperforming ICM.

2 Literature Review: Transfer Learning for multi-output observation processes

Transfer learning is a powerful tool in machine learning and is an evolving area of research [9]. In isolated learning, data (i.e. observations) from several processes are jointly modelled to make future predictions. With transfer learning, however, knowledge obtained in disparate domains, for disparate learning tasks, are processed to inform the beliefs (learning) of the target task [10].

In the multi-task learning context, multiple tasks are trained in parallel, while using a shared representation, i.e. they are assumed to share a mutual structure. The appeal of multi-task learning is that each model can generalize (i.e. avoid over-fitting) better than tasks that are learnt in isolation [1],[2].

As noted in [1], however, incorrect or ill-informed assumptions about the relationships between tasks can have detrimental effects on the performance of multi-task inference. This is referred to as negative transfer, whereby the knowledge learnt in a source task reduces the model performance when it is transferred to the target task. It is, of course, desirable that knowledge transfer improves target model performance under favourable conditions, while avoiding transfer under unfavourable conditions, i.e. that it achieves positive, robust transfer. It is, therefore, a very relevant issue to build a robust framework for the specification of the relationship between tasks.

Gaussian Processes (GPs) are flexible, non-parametric processes, which provide a probabilistic approach to learning, able to capture complex dependencies. They provide a well founded framework for learning and model selection [6]. They been have successfully implemented in a range of applications, such as medical time-series analysis [11] and terrain modelling [12]. Gaussian Processes have been effective in modelling dependencies between tasks in the multi-task context, often able to outperform baseline models [13], [14].

In [2], [15], a GP prior is placed over the latent functions, so that correlations are induced between the tasks with fixed or input-dependent, varying coefficients . These complete modelling approaches require that the relationships between the source and target tasks be known and specified when inference is carried out. However, a complete model of the relationships between nodes is often not available and so sensitivity to model choice is a problem for completely modelled approaches.

In [3][4] transfer learning between two Gaussian Process Regression tasks makes use of Fully Probabilistic Design (FPD), whereby the stochastic dependence between the source and target tasks does not need to be specified. This has been shown to provide better performance than ICM. FPD conditions the unknown target quantities on the knowledge source, in the form of a probability distribution. This method optimizes the target posterior predictor of unlabelled function evaluations conditioned on the source posterior predictor of unlabelled outputs.

While this FPD approach has been able to achieve robust, positive transfer with improvements on more traditional methods, it has been done in the single-source, single-target context. In this report, it is adapted to accommodate for multiple sources of knowledge [5], and this multi-source framework is shown to provide increased performance over the single-source framework.

3 Modelling PM2.5

The Indian pollution data set [16] contains hourly readings of multiple pollutants (in $\mu g/m^3$), including PM2.5, from 172 data gathering stations across the country, for the year 2019. In this work, however, we will focus on the geo-spatial data in a fixed temporal snapshot. PM2.5 are airborne particles with a diameter less than 2.5 microns.They constitute an extremely harmful pollutant which can cause serious health issues in a population. However, due to their small size, they are difficult to measure and measuring instruments are extremely sensitive and prone to error [17],[18].

Readings of PM2.5 levels, y, at some location, $x = (x_i, x_j)$, can be expressed as the realization of some underlying function output, f, in the presence of additive noise, ϵ. This noise is assumed to be additive white Gaussian noise (AWGN) with zero mean, such that:

$$y(x) = f(x) + \epsilon \tag{1}$$

where

$$\epsilon \sim \mathcal{N}(o, \sigma^2) \tag{2}$$

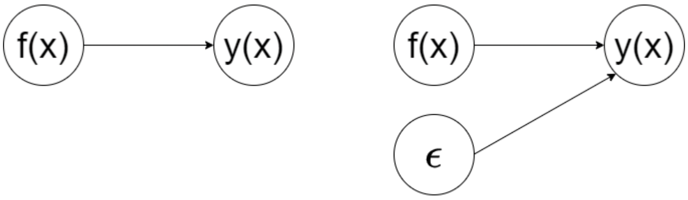


Figure 1: Noiseless and noisy observation processes, y(x), at location x.

Due to the limitations involved in recording PM2.5 concentrations, stations may be gathering inaccurate measurements. Some stations may not have PM2.5 reading capabilities at all (i.e. missing data). Another issue is that the pollution measurement stations tend to be located in cities, leaving people living in more rural settings without knowledge of pollution in their area, inhibiting them from making decisions that could protect their health. Pollution in India leads to over 2 million estimated early deaths every year [19].

What we would like to do is construct an appropriate inference model which can be used to accurately predict PM2.5 concentrations at hold out locations. A successful model could also be expected to provide some de-noising capabilities, should a reading be an outlier, ie. greatly outside of the inference confidence interval.

A model will be assessed on its ability to predict accurately the PM2.5 snapshot concentration at hold-out stations, using a standard measurement of error, the root mean squared error (RMSE). The error for predictions, $\hat{y}$, of N hold-out values, y, is expressed as follows:

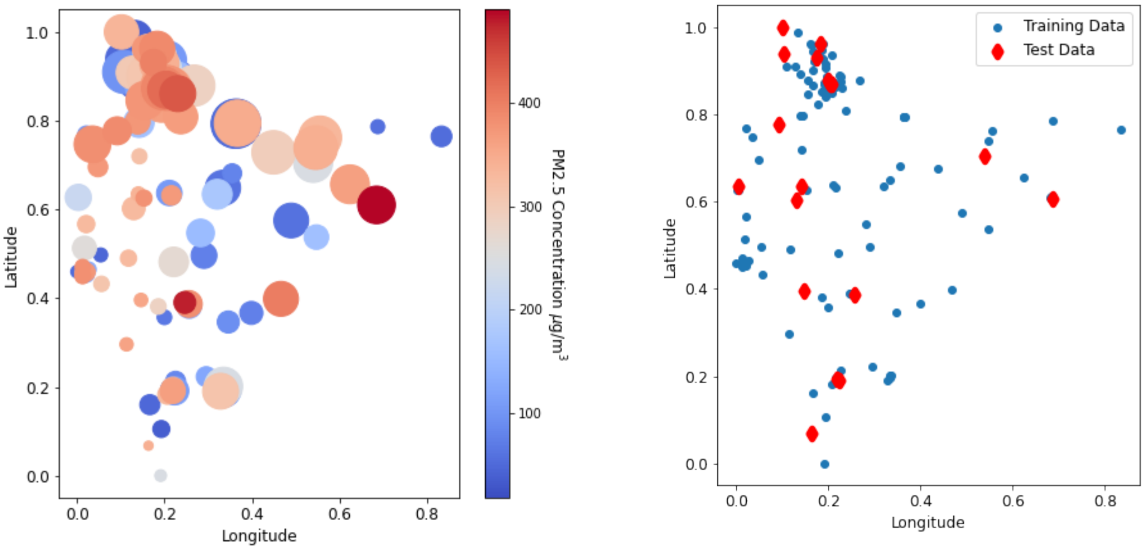
$$RMSE = \sqrt{\frac{\sum_{n=1}^{N} (\hat{y_n} - y_n)^2}{N}} \tag{3}$$

In order to address the different scales of the longitude and latitude inputs, x_i and x_j, respectively, they are both scaled by the same factor to [0,1]. A more general approach to this is Automatic Relevance

Determination (ARD)[20] but our simple approach here will achieve satisfactory results. When working with longitude and latitude, area is treated as a flat surface; the curvature of the earth is not accounted for.

Prior inspection showed that the data towards the end of the year were more rich than those earlier in the year. As such, the first half of the year was pruned from the data. Hourly readings of each pollutant were averaged to yield an output for the day. When querying the data for missing entries, it was found that measurements for PM10, NH_3, and O_3 were sparse. As such, these pollutants were dropped from the data. Days with missing measurements for PM2.5, CO, NO_2, or SO_2 were then dropped. The days were ranked in order of the number of stations providing full data. Models were then tested independently on the top 10 days of data.

Figure 2 shows the readings for the PM2.5 concentration at the pollution monitoring stations across India on the 4th November, 2019, as well as the splitting of the stations into training and test sets.



(a) PM2.5 concentrations at the pollution monitoring stations

(b) Locations of the pollution monitoring stations illustratively split into training and hold-out points.

Figure 2: (a): Average PM2.5 concentrations recorded on 4th November, 2019. Higher concentrations of PM2.5 are denoted by larger bubble size and by the colour gradient. (b): The split of the data-gathering stations into those providing training and test, i.e. hold-out data. Latitude and Longitude are normalised to unity in these plots.

3.1 k-Nearest Neighbours (kNN)

A common machine learning approach to the problem of inferring values at locations across a 2-dimensional spatial plane with non-uniform sampling is the kNN regression model [7]. We use this model to attempt to infer the PM2.5 concentrations at the hold-out locations. Figure 3 shows the RMSE (3) obtained for a distance-weighted kNN regression model for varying number of neighbours, i.e. the number of nearest data gathering stations included in the regression model. It is evaluated several times with random initial values for the hyperparameters and for several selections of hold-out locations, as well as for several of the day-average snapshots.

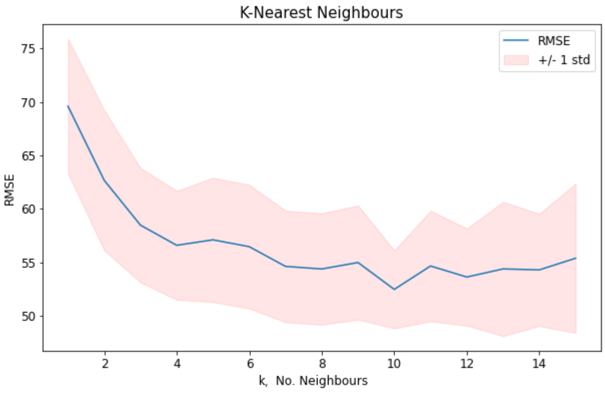


Figure 3: k-Nearest Neighbours regression: varying k, the number of neighbours

We can see that taking 10 as the number of neighbours minimizes error in our simulations, with a RMSE (3) of 52.5 ± 3.6. As such, we will move forward with $k = 10$ for our optimized kNN model. However, we will explore some other approaches to reducing predictive error in the following sections.

3.2 Gaussian Process Regression (GPR)

Gaussian Process Regression [6] is a principled non-parametric approach to probabilistic modelling of functions observed under noisy conditions. The flexibility of GPRs means that they can provide robust predictive performance in a range of environments and for a multitude of data synthesis contexts.

In GPR models, the Bayesian predictor of $y*$ at a hold-out location (i.e. domain value), x^*, given noisy observations, y, at training locations, x, has a Gaussian distribution,

$$F(y^*|y) = \mathcal{N}(m^*, v^*) \tag{4}$$

with mean and variance, respectively,

$$m^* = \mu^* + K(x^*, x)(K(x, x) + \sigma^2 I)^{-1}(y - \mu) \tag{5}$$

$$v^* = (K(x^*, x^*) + \sigma^2 I) - K(x^*, x)(K(x, x) + \sigma^2 I)^{-1} K(x, x^*). \tag{6}$$

Here, the observation model is (1), where $f(x) \sim GP(\mu(x), k(x, x'))$ is the Gaussian Process (GP) prior [6] for the noiseless, unobserved state process, F(x), and $e \sim \mathcal{N}(0, \sigma^2)$ is the additive white Gaussian noise (AWGN) of known variance, σ^2.

It is expected that for two domain values, x_i and x_j, which are close together, the GP values, $f(x_i)$ and $f(x_j)$, will have similar values. For the GP, the known covariance function, $k(x_i, x_j)$ specifies the covariance between $f(x_i)$ and $f(x_j)$, i.e. the dependence between the underlying GP samples as a function of domain seperation. In (5) and (6), $K(x_i, x_j)$ is the matrix of evaluations of $k(x, x')$ at all hold-out domain values, x_i (rows), and all training domain values, x_j (columns). Finally, μ^* is the vector of the known mean function, $\mu(x)$, of the GP, F(x), evaluated at hold-out domain values, x_i^*.

The GP covariance kernel function, $k(x, x') > 0, \quad x, x' \in \mathbb{R}^+$ can be flexibly defined by the researcher.

Choosing an appropriate covariance function is extremely important as it is the key factor which dictates how predictor values can relate to each other. As such, we will evaluate the following kernel functions [6] for their ability to perform the task of predicting PM2.5 readings in the India pollution data set:

Squared-Exponential(SE):

$$k(x, x') = \sigma^2 \exp(-\frac{1}{2}\frac{|x - x'|^2}{l^2}) \tag{7}$$

Rational Quadratic (RQ):

$$k(x, x') = \sigma^2 (1 + \frac{|x - x'|^2}{2\alpha l^2})^{-\alpha} \tag{8}$$

Matern 3/2 (MA3H):

$$k(x, x') = \sigma^2 (1 + \frac{\sqrt{3}|x - x'|}{l}) \exp(-\frac{\sqrt{3}|x - x'|}{l}) \tag{9}$$

Matern 5/2 (MA5H):

$$k(x, x') = \sigma^2 (1 + \frac{\sqrt{5}|x - x'|}{l} + \frac{5(|x - x'|)^2}{3l^2}) \exp(-\frac{\sqrt{5}|x - x'|}{l}) \tag{10}$$

Here, $\sigma^2 > 0$ is the variance of the GP, $l > 0$ is the length scale, and $\alpha > 0$ is the scale mixture rate. These are the hyperparameters of the covariance function and need to be optimised using training data in order to fit the GPR model with a sufficient degree of accuracy. A common and practical approach is to use some method of gradient descent in the hyperparameter space. Adam optimization [21] is a commonly adopted approach. It provides robust, accurate results, and is computationally efficient. As such, it will be the method of optimization, going forward.

GPR models using each covariance kernel function listed above, after the hyperparameters of which have been optimised, are assessed on their performance in predicting the PM2.5 concentrations at 20 hold-out locations, having made observations at 152 training points. This is the same arrangement under which the kNN model was assessed (Figure 3). The following results were obtained.

Kernel Function	RMSE	Standard Deviation
Squared Exponential	56.94	19.3
Rational Quadratic	51.47	17.7
Matern 3/2	53.56	18.5
Matern 5/2	54.56	18.9

From the results, we can see that the RQ kernel function (8) outperforms the other functions which were tested, having a significantly lower prediction RMSE for hold-out locations than the other three which were tested. It also has a slightly lower RMSE than the optimal value (52.5) obtained from the kNN(k=10) model (Figure 3).

The idea of an expert kernel can be pursued through the construction of mixture kernels and non-stationary kernels, as explained further in [22] and [5]. For the India pollution data, however, these more complex constructions were found to provide no benefit over the single, stationary RQ kernel (8), as shown in the table below. It appears that the learning generalization that the single, stationary RQ kernel achieves gives better performance than these flexible, but more complex, kernel structures. The latter are likely over-fitting the training data.

Kernel Function	RMSE	Standard Deviation
Stationary RQ	51.47	17.7
RQ + RQ	52.69	18.9
RQ × RQ	52.44	18.8
Non-Stationary RQ	56.18	20.4

For further details of these kernel combinations, see [22], [5]. For the hyperparameter-randomized runs, the initial hyperparameter values were drawn from a gamma distribution [23] with a high degree of uncertainty. We can reduce further the mean and variance of the prediction error through better initialization of the hyperparameter values. This can be done through repeated simulation and cross validation [24] to concentrate the gamma distribution around optimum values.

To conclude, the unmixed, stationary RQ covariance function (8), with hyperparameters optimized through Adam optimization [21], will be adopted in the GPR modelling of the PM2.5 data layer going forward. As we will see next, this PM2.5 GPR learning task will constitute the source task in our Bayesian transfer learning scheme.

4 Transfer Learning from a Single PM2.5 GPR Source Task

Now that we have explored the way in which, ‘expert’ (i.e. well chosen) kernel structures positively influence the performance of predictive inference for well-trained GPR learning tasks, we want to explore how these expert GPR learning tasks can be used to improve the performance of a poorly designed model. This is a task in Bayesian transfer learning (BTL).

Transfer learning is the use of knowledge gained from one learning task (source task) and applying the knowledge to a similar task (target task) to improve its predictive performance. The time that the target task spends learning and the amount of data needed for generalization can be greatly reduced via transfer learning. A good framework will improve the target’s performance for fixed time and data budgets. This is referred to as *positive* transfer, while a source task which decreases the target’s performance is said to induce *negative* transfer. Our aim, indeed is for *robust* transfer learning in which the target can isolate itself from deleterious source transfer, while also being able to accept a transfer which improves performance.

Returning to the Indian pollution data introduced in Section 3, we hypothesise that measurements of other pollutants, such as CO, NO_2, and SO_2 which can more easily be measured, may be used in transfer learning to build an accurate predictive inference model of $PM2.5$ levels, outperforming an isolated $PM2.5$ learner. The intuition is that these other, easily measured pollutants are positively correlated with PM2.5, but we do not explicitly model this correlation. In this sense, BTL is an approach to inference with incomplete models.

We will first assess this approach using a state-of-the-art multi-task learning method, the Intrinsic Coregionalization model (ICM)[8]. Then, we will adopt the Bayesian transfer learning (BTL) algorithm described in [4] for a single source task transfer. Finally, the BTL approach, which has been used to date for probabilistic knowledge transfer from a single source to a single target, will be extrapolated to account for multiple sources of knowledge in Section 5. This will allow us to explore transfer from three knowledge sources, i.e. the GPR learners for CO, NO_2, and SO_2 in the India pollution data context.

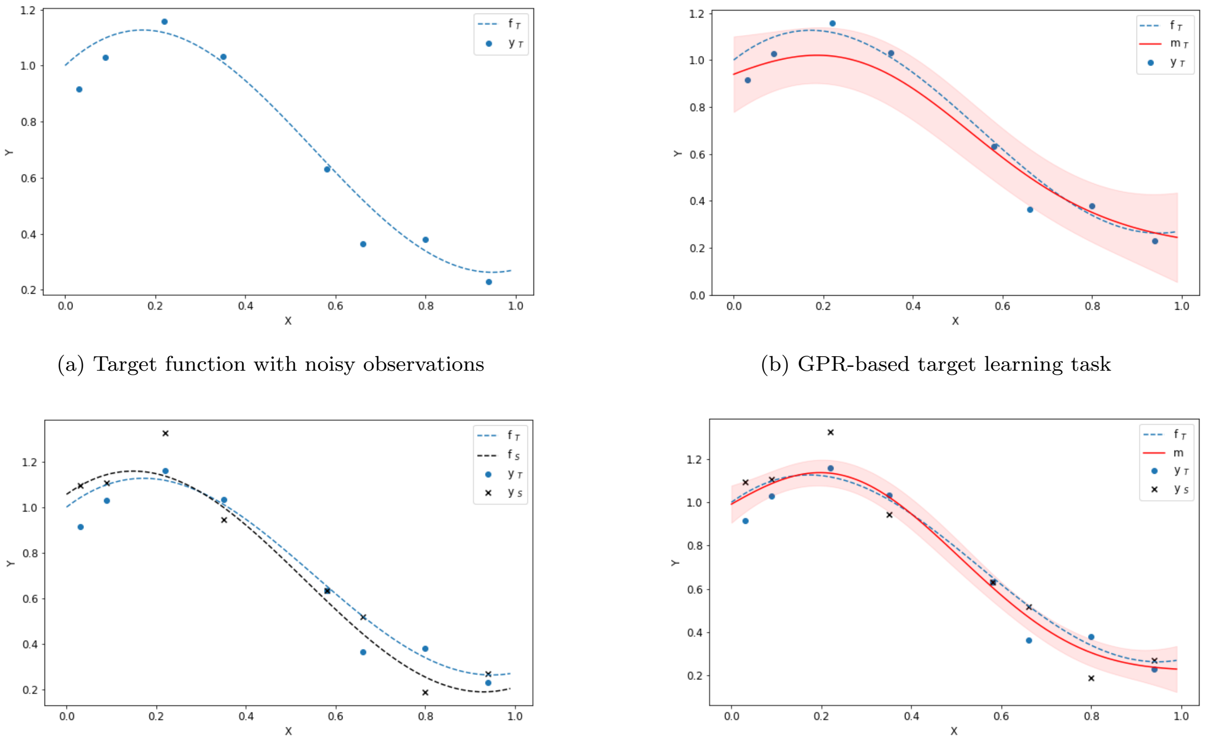
4.1 Intrinsic Coregionalization Model (ICM)

ICM [8] is a distributed inference scheme, which assumes that the source and target share the same underlying GP structure, $u(x)$, but one is a scaled version of the other. The observation noises of the source and target tasks are also assumed to be independent and on the shared domain, x. The outputs of the source and target are therefore modelled as follows:

$$y_S \sim \mathcal{N}(a_S u(x), \sigma_S^2) \tag{11}$$

$$y_T \sim \mathcal{N}(a_T u(x), \sigma_T^2) \tag{12}$$

where a_S and a_T are the known (or estimated) source and target scaling factors, respectively, for the common GP, $u(x) \sim GP(\mu_u(x), k_u(x, x'))$. For example, if we take $a_S = a_T = 1$, the target treats the observations made by the source as observations pertaining to its own task, but with potentially different known additive noise variances, σ_T^2 and σ_S^2. The target’s GPR model can thus be trained on a larger pool of data, now augmented by the source data, $y_S(x_S)$. This idea is depicted in Figure 4.



(a) Target function with noisy observations

(b) GPR-based target learning task

(c) Source and target tasks with noisy observations

(d) ICM GPR learning using source and target data

Figure 4: In the top row, the target modeller makes observations of its own underlying process, $y_T(x_T)$ and then attempts to fit a GPR model to the data. Here, $f_T(x) \equiv a_T u(x)$ denotes the true underlying function and m denotes the inferred mean via GPR. In the bottom row, observations of another, similar process, the source task, are introduced. These observations from the source task, with $f_S(x) \equiv a_S u(x)$ are processed in the the target's GPR model, via the ICM method, with $a_S = a_T = 1$. Improved performance is evident.

4.2 Bayesian Transfer Learning (BTL)

Here, the BTL algorithm described in [4] is briefly summarised. In this framework, the target becomes a global GPR modeller of both the source and target processes. A complete modelling scheme would take the form, $F(f_S, f_T|y_S, y_T)$, which processes (i.e. conditions on) the raw source data, y_S, as well as the target's, y_T (i.e. it is the joint a posteriori probability model of the source and target GPs). Conversely, the target global modeller in our transfer learning framework does not require access to y_S, but instead, receives an appropriate probabilistic inference, F_S, that is computed and transferred in the source. The objective of BTL is for the target to update its knowledge of the source-target system by conditioning on F_S in an optimal manner, and without requiring a complete model of the (f_S, f_T, F_S) system. We call this incomplete modelling, a property that confers robustness on BTL. This scheme is illustrated in Figure 5, showing, as output, the optimal design of the source-knowledge-conditional inference, $M^o(f_S, f_T|y_T, F_S)$. This is the defining characteristic of *Bayesian* transfer learning, as defined in [4]: the target processes the sufficient statistics of y_S, projected into the transferred source predictor, $F_S(y_S^*|y_S)$. It does not require the transfer and target-processing of the raw source data, y_S, as required in conventional multi-task / distributed inference schemes, such as ICM, as noted in Section 4.1.

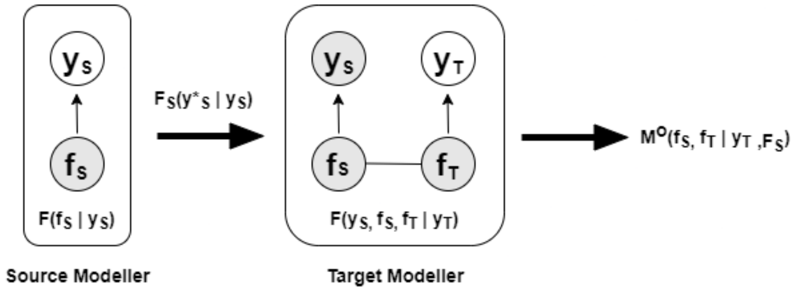


Figure 5: The source isolated modeller transfers its posterior predictive distribution, $F_S(y_S^*|y_S)$, which the global target modeller uses to build the F_S-conditional joint model, $M^o(f_S, f_T|y_T, F_S)$, given its local data, y_T. Grey nodes represent unobserved quantities, while white nodes represent observed quantities.

The target GPR task must, therefore, elicit its auto-covariance structure for $f_S(x_S)$ *and* $f_T(x_T)$, via the kernel functions, $k_{SS}(x_S, x_S)$ and $k_{TT}(x_T, x_T)$, respectively (see Section 3.2). It must also specify the cross-covariance kernel function, $k_{ST}(x_S, x_T) = k_{TS}(x_T, x_S)$, eliciting the cross-covariance structure between $f_S(x_S)$ and $f_T(x_T)$. When these are evaluated at domain values (i.e. locations) $\mathbf{x}_S$ and $\mathbf{x}_T$, respectively, the covariance matrix, K, has the following block structure:

$$K = \begin{bmatrix} K_{SS} & K_{ST} \\ K_{ST}^T & K_{TT} \end{bmatrix} \tag{13}$$

K_{SS} and K_{TT} are the auto-covariance matrices of the source and target GPs respectively, as modelled by the global target, via k_{SS} and k_{TT}, respectively. K_{ST} is the covariance matrix between the source and target. Here, also, K_{ST}^T denotes the transpose of K_{ST}. In this work, the target adopts the ICM complete model as described in Section 4.1. In this case, the block covariance matrix (13) specializes to the following simpler structure:

$$K = B \otimes k_u(\boldsymbol{x}, \boldsymbol{x}) \tag{14}$$

$$B = \begin{bmatrix} a_S \\ a_T \end{bmatrix} \begin{bmatrix} a_S \\ a_T \end{bmatrix}^T = \begin{bmatrix} b_{SS} & b_{ST} \\ b_{TS} & b_{TT} \end{bmatrix} \tag{15}$$

i.e. $b_{SS} = a_S^2$, $b_{TT} = a_T^2$ and $b_{ST} = b_{TS} = a_S a_T$ (11),(12), and $\otimes$ denotes the Kronecker product. $k_u(x, x)$ is the target's covariance function for $u(x)$, and it is further assumed that $x_S = x_T \equiv x$, i.e. the N source and target domain values (locations) are coincident. The values for a_S and a_T, the ICM coefficients, need to be set a priori, requiring (in real data contexts, such as ours) their optimization as hyperparameters with respect to the data, y_S and y_T, to optimise the performance of the algorithm.

In BTL [4], as stated above, the source independently models its local data, y_S, and transfers its data predictor, (4),

$$F_S(y_S^*|y_S) = \mathcal{N}(m_S, R_S), \tag{16}$$

to the target. A key feature of this multiple modeller approach is that the *source's* covariance function for $f_S(x_S)$ can be different (indeed, more expert) in modelling its *local* data, y_S. The statistics, m_S and R_S in (16) are therefore evaluated as in (5),(6), via this local (source) expert knowledge.

Recall that the target must now condition on F_S (16), in this incompletely modelled context. The resulting F_S-conditional target inference of $u(x)$ is not uniquely specified in the absence of a complete model of dependence between the target and F_S. Fully probabilistic design (FPD) [25], [26] chooses the optimum, $M^o(u|y_T, F_S)$, via an appropriate Kullback-Leibler divergence (i.e. Bayesian risk) minimization, yielding:

$$M^o(u(x)|y_T, F_S) = \mathcal{N}(\frac{1}{a_T} m_T^o, \frac{1}{a_T^2} k_{TT}^o) \tag{17}$$

$$m_T^o = \boldsymbol{k}_T(K + blkdiag(R_S, \sigma_T^2 I_N))^{-1} \begin{bmatrix} m_S \\ y_T \end{bmatrix} \tag{18}$$

$$k_{TT}^o(x, x') = k_{TT}(x, x') - \boldsymbol{k}_T(K + blkdiag(R_S, \sigma_T^2 I_N))^{-1} \boldsymbol{k}_T^T. \tag{19}$$

Here, $\boldsymbol{k}_T$ is the length-$2N$ row vector of cross-covariances between the target's points of interest, x (i.e. the test points where $u(x)$ is to be inferred), and the source predictive transfer points, $\boldsymbol{x}_S$, concatenated with the target's auto-covariances between x and $\boldsymbol{x}_T = \boldsymbol{x}_S \in \mathbb{R}^N$:

$$\boldsymbol{k}_T = [k_{TS}(x, \boldsymbol{x}_S) \; k_{TT}(x, \boldsymbol{x}_T)]. \tag{20}$$

k_{TS} and k_{TT} are the joint (ICM) target's cross-covariance and auto-covariance functions (14).

5 Multi-Source Transfer

The algorithm described in Section 4.2 can readily be extended to the multiple task case. Here we focus on the multiple source, single target case relevant to the BTL from multiple pollutant (source) learning tasks to the PM2.5 (target) learning task.

5.1 Constructing the multiple source, single target BTL Algorithm

We assume the same ICM structure that has been imposed upon the source and target models in the single source and target case. Extending (11) to $n \geq 1$ sources, $S_q \in (S_1, S_2, ...S_n)$, the target's conditionally independent, identically distributed models for these are:

$$y_{S_q} \sim \mathcal{N}(a_{S_q} u(x), \sigma^2_{S_q}), \quad q = 1, ..., n \tag{21}$$

Define $\bar{f_S} = (f_{S_1}, f_{S_1}, ...f_{S_n})$ to be the vector of function values of each source and compute $F_{S_q}(y^*_{S_q}|y_{S_q})$, $q = 1, ..., n$, the posterior predictive distribution of each isolated, transferred source in turn (16). We seek the target's joint model of the latent functions, conditioned on these n transferred predictors and on the target's local data, y_T:

$$M(\bar{f_S}, f_T|\bar{F_S}, y_T) \tag{22}$$

In common with the FPD-optimal BTL framework [4], the target does not specify a joint model for f_T and F_{S_q}, and so the conditional (22) is non-unique. Its Bayesian minimum risk design (i.e. FPD-optimal design) is the one which minimizes the Kullback-Leibler Divergence from F_{S_q}-constrained candidates (22) to an ideal (i.e. zero-loss) choice specified by the FPD-optimal target. The details are available in [4], and lead to the following $n \geq 1$ generalization of (17), (18), (19):

$$M^o(\bar{f_S}, f_T|\bar{F_S}, y_T) = \mathcal{N}(m^o, k^o) \tag{23}$$

where

$$m^o = \begin{bmatrix} \bar{m^o_S}(x) \\ m^o_T(x) \end{bmatrix}, \quad k^o = \begin{bmatrix} \bar{k}^o_{SS}(x, x') & \bar{k}^o_{ST}(x, x') \\ \bar{k}^o_{TS}(x, x') & k^o_{TT}(x, x') \end{bmatrix}.$$

Here:

$$m^o_T = \mathbf{k}_T (K + blkdiag(R_{S_1}, R_{S_2}, ..., R_{S_n}, \sigma^2_T I_N))^{-1} \begin{bmatrix} m_{S_1} \\ m_{S_2} \\ \vdots \\ m_{S_n} \\ y_T \end{bmatrix} \tag{24}$$

$$k^o_{TT}(x, x') = k_{TT}(x, x') - \mathbf{k}_T (K + blkdiag(R_{S_1}, R_{S_2}, ..., R_{S_n}, \sigma^2_T I_N))^{-1} \mathbf{k}^T_T \tag{25}$$

Adopting the previously established convention of the target joint modeler assuming a complete ICM analysis model (21) with common $u(x)$ for sources and target, the K block-matrix (13) is constructed as outlined in (14), using a *known* coefficient, a_T, for the target and n, *known* coefficients for the source, $\boldsymbol{a}_S \equiv [a_{S1}, a_{S2}, ..., a_{Sn}]$. Hence:

$$K = \begin{bmatrix} K_{SS} & K_{ST} \\ K_{ST}^T & K_{TT} \end{bmatrix} = B \otimes k_u(\boldsymbol{x}, \boldsymbol{x}^T) \tag{26}$$

$$B = \begin{bmatrix} \boldsymbol{a}_S \\ a_T \end{bmatrix} \begin{bmatrix} \boldsymbol{a}_S^T & a_T \end{bmatrix} = \begin{bmatrix} \boldsymbol{a}_S \boldsymbol{a}_S^T & \boldsymbol{a}_S a_T \\ a_T \boldsymbol{a}_S^T & a_T^2 \end{bmatrix} \tag{27}$$

With n sources, we can expand K_{SS} and K_{ST} block-matrices,

$$K_{SS} = \begin{bmatrix} K_{S_1S_1} & K_{S_1S_2} & ... & K_{S_1S_n} \\ K_{S_1S_2}^T & K_{S_2S_2} & ... & \vdots \\ \vdots & ... & \ddots & \vdots \\ K_{S_1S_n}^T & ... & ... & K_{S_nS_n} \end{bmatrix} \tag{28}$$

$$K_{ST} = \begin{bmatrix} k_{S_1T}(\boldsymbol{x}, x) \\ \vdots \\ k_{S_nT}(\boldsymbol{x}, x) \end{bmatrix} \tag{29}$$

Finally, the *row* vector, $\mathbf{k}_T$, in (24) and (25) has the expanded form:

$$\mathbf{k}_T = [k_{TS_1}(x, \boldsymbol{x}_{S_1}^T)\ k_{TS_2}(x, \boldsymbol{x}_{S_2}^T)\ ...\ k_{TS_n}(x, \boldsymbol{x}_{S_n}^T)\ k_{TT}(x, \boldsymbol{x}_T^T)]. \tag{30}$$

5.2 Transferring knowledge from CO, NO$_2$, and SO$_2$ to PM2.5

Our focus here will be on the implementation of the multi-source FPD-optimal BTL method (Section 5.1) for the Indian pollution data (Section 3), specifically, the transfer of $n = 3$ source predictors (for CO, NO$_2$, and SO$_2$) to the target PM2.5 inference task, in an effort to achieve positive transfer learning for the target task. We will detail each step of the implementation, including some pseudo-code snippets. Note that each of the $n + 1 = 4$ data channels (i.e. pollutant concentration in $\mu g/m^3$) was pre-processed to yield zero-mean channels with variance 1.

The target modeller builds their model of scaled PM2.5 concentrations using a rational quadratic kernel (8). Then, using this target model's covariance function, $k_u(x, x)$ (14) optimized with respect to the target training points, along with the optimized ICM coefficients (27) for all four channels, we construct the block matrix, K (26).

To optimize the ICM coefficients for each data channel, data from the training stations for each of several days, with known PM2.5 concentrations were used. Gradient descent was repeatedly performed on the coefficient of each data channel, with varying initialized values. The set of final (post gradient descent) coefficients that minimized the predictive RMSE (3) for PM2.5 was used as the set of optimum coefficients for that day. This process was repeated for several days of data and then the mean of these optimum coefficient sets was adopted in (26) and (27). This pre-processing improves numerical stability and takes the place of methods such as automatic relevance determination (ARD) [20]. Adopting the ordering notation of $S_1 \equiv CO$, $S_2 \equiv NO_2$, $S_3 \equiv SO2$, $T \equiv PM2.5$, the resulting optimal settings for the ICM coefficients were $a_{S_1} = 0.56$, $a_{S_2} = 0.52$, $a_{S_3} = 0.3$, $a_T = 1$, having normalised with respect to a_T.

We assess 172 pollution measurement stations, partitioned into 152 training stations and 20 randomised hold-out stations. The experiment, along with the random allocation of measurement stations into training and hold-out sets (i.e. cross-validation) is repeated through Monte-Carlo simulations to mitigate any bias that could arise from the partition of the stations into their respective sets. Therefore, the target's covariance matrix, $k_u(\boldsymbol{x}, \boldsymbol{x}^T)$ (26), has dimension [152×152] and so the target's ICM block covariance matrix, K, has dimension (26) is $(4 \times 152) \times (4 \times 152) = [608 \times 608]$.

In the following snippet of pseudo-code, we build the K block matrix given the target's covariance kernel, k_{uu}, and the ICM coefficients, $\boldsymbol{a}_S$ and a_T.

```
GPR_PM25_training = Build_GPRM(x_train, PM25_train)
k_uu = GPR_PM25_training.covariance
B = [as_1, as_2, as_3, at].Transpose * [as_1, as_2, as_3, at]
K = kronecker(B, k_uu)
```

Next, using the same split of training and test stations as the target, each of the three source modellers builds their respective posterior predictive model (16) of CO, NO_2, and SO_2. Several kernel functions are assessed in the same manner as in Section 3.2, each now trained independently on the source tasks' respective local data, conferring *expertise* on these source tasks. The results are shown in the table below.

CO:

Kernel Function	RMSE	Standard Deviation
Squared Exponential	21.32	3.5
Rational Quadratic	19.8	3.5
Matern 3/2	20.39	3.6
Matern 5/2	20.72	3.6

NO_2:

Kernel Function	RMSE	Standard Deviation
Squared Exponential	31.88	9.5
Rational Quadratic	30.85	10.1
Matern 3/2	31.27	9.9
Matern 5/2	31.47	10

SO_2:

Kernel Function	RMSE	Standard Deviation
Squared Exponential	14.38	5.9
Rational Quadratic	14.57	5.8
Matern 3/2	14.54	5.8
Matern 5/2	14.49	5.9

Now that each source has constructed its own isolated model, it transfers its own mean function, m_{S_q}, and covariance, R_{S_q}, of the posterior predictor at the transfer points.

```
m_s1 = GPR_CO_training.mean_function
m_s2 = GPR_NO2_training.mean_function
m_s3 = GPR_SO2_training.mean_function

R_s1 = GPR_CO_training.covariance
```

```
R_s2 = GPR_NO2_training.covariance
R_s3 = GPR_SO2_training.covariance
```

We will focus first on constructing the FPD-optimal target mean (a scalar), m_T^o (24), at a single target domain value, x, after multi-source transfer. It is rewritten here for our specific case of $n = 3$ sources:

$$m_T^o = \mathbf{k}_T (K + blkdiag(R_{S_1}, R_{S_2}, R_{S_3}, \sigma_T^2 I_N))^{-1} \begin{bmatrix} m_{S_1} \\ m_{S_2} \\ m_{S_3} \\ y_T \end{bmatrix}$$

Using each source's predictive covariance matrix, R_{S_q}, and the target model's estimate of its noise variance, σ_T^2, we can construct the block-diagonal term above. σ_T^2 was treated in the same way as the GPR covariance kernel hyperparameters, i.e. its MLE was found via Adam optimization. Each block-matrix has dimension [152 × 152], resulting in a [608 × 608] block diagonal matrix. With the mean function of each source, m_{S_q} and the target's observations, y_T, each of length $N = 152$, we can construct the column vector of length 4 × 152 = 608 . We build the block-diagonal matrix and m_T^o here:

```
target_noise_var_matrix = sigma_T * identity_matrix(152)
block_diagonal = blkdiag(R_s1, R_s1, R_s1, target_noise_var_matrix)

vertical_block = [m_s1, m_s2, m_s3, y_t].Transpose
```

For $\mathbf{k}_T$ (30), we need to consider at what predictive points (i.e. locations) we want to infer $y*$, the PM2.5 readings. We could assess only the hold-out points, $x*$, or we could assess both the hold-out and training points, x_T, together. Assessing both would allow us not only to infer at the unobserved points but also attempt to denoise at the training locations. It is important to realise, however, that we do not have access to any noise-free measurements, and so we cannot quantify the accuracy of filtering. Nonetheless, we will move forward with including both the hold-out and the training points, from which inference at the hold-out points can be ascertained. Rewriting (30) for $n = 3$ sources:

$$\mathbf{k}_T = [k_{TS_1}(x, \boldsymbol{x}_{S_1}^T)\ k_{TS_2}(x, \boldsymbol{x}_{S_2}^T)\ k_{TS_3}(x, \boldsymbol{x}_{S_3}^T)\ k_{TT}(x, \boldsymbol{x}_T^T)] \tag{31}$$

First, we need to construct the K block-matrix (26) again, using the same ICM coefficients, and once again assuming $\mathbf{x}_{S_q} = \mathbf{x}_T = \mathbf{x}$, each of length $N = 152$. As such, the u-kernel sub-matrix, $k_u(\boldsymbol{x}, \boldsymbol{x}^T)$, has dimensions [152×152]. This then yields the covariance structure between all $\mathbf{x}_{S_q}$, $q \in \{1, 2, 3\}$, and $\mathbf{x}_T$, via the same Kronecker product form in (26). The corresponding K block-matrix is thus [608×608]:

$$K = \begin{bmatrix} K_{S_1S_1} & K_{S_1S_2} & K_{S_1S_3} & K_{S_1T} \\ K_{S_2S_1} & K_{S_2S_2} & K_{S_2S_3} & K_{S_2T} \\ K_{S_3S_1} & K_{S_3S_2} & K_{S_3S_3} & K_{S_3T} \\ K_{TS_1} & K_{TS_2} & K_{TS_3} & K_{TT} \end{bmatrix}$$

The FPD-optimal mean (column) vector, $\mathbf{m}_T^o(\mathbf{x}^*)$, of $f_T(\mathbf{x}^*)$ (23), at *all* 20 hold-out domain values (i.e. points), $\mathbf{x}^*$, in the target is formed by stacking m_T^o (24) for each hold-out point, x_i^*, $i = 1, \ldots, 20$:

$$\mathbf{m}_T^o(\mathbf{x}^*) = \mathbf{K}_T (K + blkdiag(R_{S_1}, R_{S_2}, ..., R_{S_n}, \sigma_T^2 I_N))^{-1} \begin{bmatrix} m_{S_1} \\ m_{S_2} \\ \vdots \\ m_{S_n} \\ y_T \end{bmatrix}, \tag{32}$$

where

$$\mathbf{K}_T \equiv \begin{bmatrix} \mathbf{k}_T(x_1^*) \\ \vdots \\ \mathbf{k}_T(x_{20}^*) \end{bmatrix},$$

and where $\mathbf{k}_T(x_i^*)$, $i = 1, \ldots, 20$, are each given by (31), evaluated at the respective hold-out point, $x = x_i^*$.

Correspondingly, the 20×20 FPD-optimal covariance matrix of $f_T(\mathbf{x}^*)$ (23) at $\mathbf{x}^*$ is the matricized form of (25), as follows:

$$\mathbf{K}_{TT}^o(\mathbf{x}^*, \mathbf{x}^{*T}) = \mathbf{K}_{TT}(\mathbf{x}^*, \mathbf{x}^{*T}) - \mathbf{K}_T(K + blkdiag(R_{S_1}, R_{S_2}, ..., R_{S_n}, \sigma_T^2 I_N))^{-1}\mathbf{K}_T^T. \tag{33}$$

The target's FPD-optimal mean, $m_T^o(x_i^*)$ (24), and variance, $k_{TT}^o(x_i^*, x_i^*)$ (25) (being the ith element of $\mathbf{m}_T^o$ (32), and the (i,i)th element of $\mathbf{K}_{TT}^o$ (33), respectively) constitute the uncertainty-equipped estimate of the PM2.5 concentration at each of the 20 hold-out locations, x_i^*, $i = 1, \ldots, 20$.

6 Comparative Performance with India Pollution Data

In these simulations, we assess the performance of our standard inference model, k-Nearest Neighbours Regression (3.1) against our two transfer learning approaches; the state of the art, ICM and the multi-source Bayesian Transfer Learning framework. An isolated GPR model of PM2.5, the target task is also used as a benchmark against which to judge our transfer learning approaches. All pollutants go through standard scaling, where their mean is scaled to zero and a variance of one is induced. Errors are averaged over several sets of training and hold-out locations. The RMSE(3) is used for our error measure. The results for 5 days and their average are given below.

Model	Day 1	Day 2	Day 3	Day 4	Day 5	Avg.
kNN	0.48±0.06	0.63±0.20	0.43± 0.06	0.42±0.14	0.47±0.04	0.49±0.1
TNT	0.51±0.09	0.50±0.07	0.61± 0.12	0.54±0.08	0.44±0.07	0.52±0.09
ICM	0.61±0.05	0.57±0.05	0.75± 0.08	0.71±0.07	0.5±0.05	0.63±0.06
BTL	0.43±0.12	0.43±0.09	0.55± 0.15	0.49±0.09	0.41±0.09	0.46±0.11

We note the following

- Our Bayesian Transfer Learning (BTL) algorithm, which makes use of transferred predictive probabilistic knowledge from multiple sources, consistently outperforms the isolated target model (TNT). It also vastly outperforms ICM under these settings and marginally outperforms the kNN model.
- Our Bayesian Transfer Learning (BTL) algorithm is robust, meaning it has the ability to reject poor-quality source data, while the ICM approach does not exhibit this robustness. In general, transfer that results in reduced performance is referred to as negative transfer, while improved trasnfer-based performance is referred to as positive transfer. As such, BTL consistently delivers positive transfer in this application, while the performance of ICM is heavily reliant on the quality of source knowledge.

7 Final Remarks

Throughout these experiments, Monte Carlo runs were carried out to randomize for some of the settings. However, due to the computational complexity of working with, and optimizing, GPR models and the number of GPR models involved in these experiments, a large number of Monte Carlo runs was often not feasible. Efforts were made to allocate large amounts of time to running simulations, but ultimately, many simulations involved only about 100-200 runs. This undermined, somewhat, the estimation of the underlying performance errors. Nevertheless, the main comparative properties of the algorithm were revealed.

If measurements of other pollutants that were dropped in early analysis (PM10, NH_3, and O_3) were available, they could readily be incorporated into our BTL framework and used as supplementary knowledge sources, prospectively improving further the BTL target performance. As shown with the simulated data in [5], as the number of sources increases, positive transfer improves if the sources are correlated with (i.e. are predictive of) the target. If they are not, they are rejected, without undermining the isolated target learner, a core feature of our robust transfer scheme. In this way, increasing the number of source learners can be a way to converge to the target performance one would obtain from using optimally chosen source analysis models.

The India pollution data provides both geo-spatial and temporal data. For this experiment, only geo-spatial information was considered on individual days, without any inter-day dynamic modelling. We expect that exploiting temporal knowledge (dynamics) would induce more accurate performance of the BTL-optimized PM2.5 target learner.

7.1 Conclusion

The multi-source Bayesian Transfer Learning (BTL) algorithm is a robust transfer learning algorithm, consistently providing significant positive transfer compared to the isolated PM2.5 target learner (i.e. the isolated target GPR task).

It overcomes the fragilitiy of the common, GP-based, state-of-the-art, the Intrinsic Coregionalization Model (ICM) approach and it provides marginally better results than the standard kNN approach. The BTL algorithm is readily extensible to $n > 3$ sources, and we expect further improved performance in this case, as well as in the case where the source and target learners adopt temporal dynamics across multiple days of data.

In future work, focus could be put on investigating mismatch of the joint interaction model between the sources and target, which inevitably affects complete modelling approaches such as ICM. We accept that the dominant factor explaining the improved performance of BTL over ICM in the experiments reported here is the multi-model feature of BTL and its ability to transfer local source modelling expertise. However, even when higher-rank ICM modelling is adopted [1], with locally optimized GP state processes, $f_{S_q}(x)$, for each source - thereby emulating BTL (Figure 5) - ICM must still instantiate a joint prior interaction model between these GPs, something our BTL approach avoids.
Work is ongoing in BTL to allow optimal inference of the transfer weights between the sources and the target. The current framework sets these at unity *a priori*. Recall that it is the induced dependence of the FPD-optimal target inference (24),(25) on the source covariance matrices, R_{S_q}, that allows poor-quality sources to be rejected (i.e. robust transfer). We expect this rejection (robustness) property to be enhanced once the transfer weights are included as inference parameters of our BTL scheme.

References

[1] R. Caruna, “Multitask learning,” *Machine Learning*, vol. 28, pp. 41–75, 1997.

[2] K. C. E. Bonilla and C. Williams, “Multi-task gaussian process prediction,” *Proceedings of the 20th Annual Conference on Neural Information Processing Systems*, pp. 153–160, 2008.

[3] M. Papež and A. Quinn, “Bayesian transfer learning between gaussian process regression tasks,” *Proc. 19th IEEE, International Symposium on Signal Processing and Information Technology (ISSPIT)*, vol. 6pp, 2019.

[4] M. Papež and A. Quinn, “Transferring model structure in bayesian transfer learning for gaussian process regression,” *arXiv:2101.06884*, 2021.

[5] S. Nugent, “Bayesian transfer learning in a network of gaussian process regression nodes,” Master’s thesis, Trinity College Dublin, 2021.

[6] C. R. . C. Williams, *Gaussian Processes for Machine Learning.* MIT Press, 2006.

[7] O. Kramer, “K-nearest neighbors,” *Dimensionality Reduction with Unsupervised Nearest Neighbors*, 2013.

[8] M. Alvarez and N. Lawrence, “Computationally efficient convolved multiple output gaussian processes,” *Journal of Machine Learning Research*, vol. 12, p. 1459–1500, 2011.

[9] L. Torrey and J. Shavlik, *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques.* 2010.

[10] S. Pan and Q. Yang, “A survey on transfer learning,” *IEEE, Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[11] L. Cheng, G. Darnell, B. Dumitrascu, C. Chivers, M. Draugelis, K. Li, and B. Engelhardt, “Sparse multi-output gaussian processes for online medical time series prediction,” *BMC Medical Informatics and Decision Making*, vol. 20, no. 152, 2020.

[12] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Whyte, “Nonstationary dependent gaussian processes for data fusion in large-scale terrain modeling,” *IEEE International Conference on Robotics and Automation*, vol. 20, pp. 1875 – 1882, 2011.

[13] M. Alvarez and N. Lawrence, “Sparse convolved gaussian processes for multi-output regression,” *Advances in Neural Information Processing Systems*, 2009.

[14] T. Nguyen and E. Bonilla, “Collaborative multi-output gaussian processes,” *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, pp. 643–652, 2014.

[15] T. Nguyen and E. Bonilla, “Efficient variational inference for gaussian process regression networks,” *Artificial Intelligence and Statistics*, pp. 472–480, 2013.

[16] O. G. D. P. India, “Air pollution data,” 2019.

[17] E. Commission, “Guidance on pm2.5 measurement under directive 1999/30/ec,” 1999.

[18] W. H. O. (WHO), “Review of methods for monitoring of pm2.5 and pm10,” *Report on a WHO Workshop Berlin, Germany*, 2004.

[19] G. A. on Health and P. (GAHP), “Pollution and Health Metrics,” 2019.

[20] D. Husmeier, *Neural Networks for Conditional Probability Estimation. Perspectives in Neural Computing.* Springer, 1999.

[21] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 2015.

[22] D. Duvenaud, *Automatic Model Construction with Gaussian Processes.* PhD thesis, University of Cambridge, 2014.

[23] E. Lukacs, "A Characterization of the Gamma Distribution," *Annals of Mathematical Statistics*, vol. 26, no. 2, pp. 319–324, 1955.

[24] M. W. Browne, "Cross-validation methods," *Journal of Mathematical Psychology*, vol. 44, pp. 108–132, 2000.

[25] M. Karny and T. Kroupa, "Axiomatisation of fully probabilistic design," *Information Sciences*, vol. 186, pp. 105 – 113, 2012.

[26] C. M. Bishop, *Pattern recognition and machine learning.* Springer, 2006.