Regret Bounds for Transfer Learning in Bayesian Optimisation

Alistair Shilton
Deakin University,
Australia

Sunil Gupta
Deakin University,
Australia

Santu Rana
Deakin University,
Australia

Svetha Venkatesh
Deakin University,
Australia

Abstract

This paper studies the regret bound of two transfer learning algorithms in Bayesian optimisation. The first algorithm models any difference between the source and target functions as a noise process. The second algorithm proposes a new way to model the difference between the source and target as a Gaussian process which is then used to adapt the source data. We show that in both cases the regret bounds are tighter than in the no transfer case. We also experimentally compare the performance of these algorithms relative to no transfer learning and demonstrate benefits of transfer learning.

1 Introduction

Experimentation permeates human endeavour, propelling us towards unexplored frontiers - new understanding, formulation of novel materials, even biological elements of life. The process of experimentation involves conducting an experiment, measuring the quality of output, then repeating the process with insights gained. This process and data acquired is inherently iterative, dynamic, small-scale, expensive and limited by resources of time, cost and even ideas. One of the main characteristics of experimentation is that knowledge is built over time through several sets of experiments that vary in setting - thus “similar” experimental data is often available. For example, in machine learning when hyperparameter tuning has been performed in the past on a particular set of data then this should be transferable to hyperparameter tuning on a similar dataset.

Utilising such past and related (source) data to improve the output of the current experiment (target function) is a natural use for transfer learning. This needs to be incorporated into mechanisms that can handle limited data. Bayesian optimisation is an exciting sub-field of machine learning providing an ideal platform to estimate such functions from limited data, relating inputs to observed outputs via sequential optimisation (Mockus, 2002; Snoek et al., 2012). It uses a Gaussian process (Rasmussen, 2006) to non-parametrically model the black-box function and converts the problem of optimising the unknown function to a problem of optimising a known surrogate function (acquisition function) constructed via the Gaussian process. Transfer of knowledge into such a Bayesian optimisation setting is desirable to solve the problem we are addressing. Though transfer learning is an established research area (Pan and Yang, 2010) for transferring knowledge from past data, limited work has focused on transfer learning for Bayesian optimisation (Bardenet et al., 2013; Yogatama and Mann, 2014).

This paper examines the theoretical properties of transfer learning in Bayesian optimisation. Specifically, we estimate the regret bounds of transfer learning in Bayesian optimisation for two algorithms. The first algorithm, Env-GP (envelope-stretching Bayesian optimisation) (Joy et al., 2016), models any differences between the source and target functions as a noise process, *stretching* the noise envelope in the source data to fit it to the target, where the amount of stretch required is proportional to the difference between them. By contrast the second algorithm, Diff-GP (difference-modelling Bayesian optimisation), *directly models* the difference between the source and target functions and then corrects the source data to match the target function. To analyse the properties of Env-GP and Diff-GP by comparison to the non-transfer context we start with the work of Srinivas et al. (2010), which proves the statistical bound on total regret whilst using GP-UCB as a acquisition function

$$\Pr\left\{R_T \leq \sqrt{C_1 \beta_T \gamma_T T} + c_0 \;\forall T \geq 1\right\} \geq 1 - \delta,$$

where γ_T is the maximum information gain and c_0 is a constant. We prove that both the maximum infor-

Proceedings of the 20^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2017, Fort Lauderdale, Florida, USA. JMLR: W&CP volume 54. Copyright 2017 by the author(s).

mation gain γ_T and the noise dependent term C_1 are decreased in the transfer learning context, and hence both Env-GP and Diff-GP are likely to find the optimum in fewer steps when source data is available.

We demonstrate the efficacy of these algorithms, and show that the Diff-GP outperform the Env-GP as the source and target diverge. Our key contributions are:

- derivation of regret bounds for two transfer learning algorithms in the context of Bayesian optimisation,
- presentation of a new transfer learning algorithm (Diff-GP), and
- experimental verification of algorithms on both simulated and real data.

2 Background

2.1 Gaussian Processes

A Gaussian Process (GP) is a random distribution over the set of smooth functions $f : \mathcal{X} \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ on compact $\mathcal{X}$ denoted

$$f(\mathbf{x}) \sim \mathrm{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')),$$

where $\mu : \mathcal{X} \rightarrow \mathbb{R}$, $\mu(\mathbf{x}) = \mathbb{E}(f(\mathbf{x}))$ is the mean function of the Gaussian process; $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, $k(\mathbf{x}, \mathbf{x}') = \mathbb{E}((f(\mathbf{x}) - \mu(\mathbf{x}))(f(\mathbf{x}') - \mu(\mathbf{x}')))$ is the kernel or covariance function. Without loss of generality we may assume $\mu(\mathbf{x}) = 0$ and $k(\mathbf{x}, \mathbf{x}) = 1$ for all $\mathbf{x} \in \mathcal{X}$. The kernel k is a prior that encodes our underlying assumptions regarding smoothness of the distribution. A popular choice of kernel function is

$$k(\mathbf{x}, \mathbf{x}') = \exp(-\tfrac{1}{2\nu^2} \|\mathbf{x} - \mathbf{x}'\|^2).$$

Given training data $\{ (\mathbf{x}_t^a, y_t^a) | \, t \in \mathbb{Z}_T \}$ generated from $y_t^a = f(\mathbf{x}_t^a) + \epsilon_a$, where $\epsilon_a \sim \mathcal{N}(0, \sigma_a^2)$ and $\mathbb{Z}_T = \{0, 1, \ldots, T-1\}$, the posterior over f is

$$f(\mathbf{x} | \, \mathbf{y}_{A_T}, \mathbf{X}_{A_T}) \sim \mathcal{N}(\mu_T(\mathbf{x}), \sigma_T^2(\mathbf{x})),$$

where

$$\begin{aligned} \mu_T(\mathbf{x}) &= \mathbf{k}_{A_T}^{\mathrm{T}}(\mathbf{x}) \left(\mathbf{K}_{A_T} + \sigma_a^2 \mathbf{I}\right)^{-1} \mathbf{y}_{A_T}, \\ \sigma_T^2(\mathbf{x}) &= k(\mathbf{x}, \mathbf{x}) - \mathbf{k}_{A_T}^{\mathrm{T}}(\mathbf{x}) \left(\mathbf{K}_{A_T} + \sigma_a^2 \mathbf{I}\right)^{-1} \mathbf{k}_{A_T}(\mathbf{x}); \end{aligned} \tag{1}$$

and $\mathbf{y}_{A_T} = [\; y_0^a \; y_1^a \; \ldots \; y_{T-1}^a \;]^{\mathrm{T}}$, $\mathbf{K}_{A_T} = [\; k(\mathbf{x}_i^a, \mathbf{x}_j^a) \;]_{i,j \in \mathbb{Z}_T}$, $\mathbf{X}_{A_T} = [\; \mathbf{x}_0^a \; \mathbf{x}_1^a \; \ldots \; \mathbf{x}_{T-1}^a \;]$ and $\mathbf{k}_{A_T}(\mathbf{x}) = [\; k(\mathbf{x}_i^a, \mathbf{x}) \;]_{i \in \mathbb{Z}_T}^{\mathrm{T}}$.

2.2 Bayesian Optimisation

Let $f(\mathbf{x})$ be a real-valued function over a compact domain $\mathcal{X} \subseteq \mathbb{R}^n$. Consider the problem

$$\underset{\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^n}{\operatorname{argmax}} f(\mathbf{x}), \tag{2}$$

where it is assumed that f is computationally expensive to evaluate, and observations of f may be affected by noise. Examples of such systems include optimising the performance of a machine learning technique for a given set of hyperparameters, or maximising the yield of a chemical reaction given temperature, pressure etc. The aim is to solve (2) using the minimum evaluations of f.

A popular approach to this problem is Bayesian optimisation (Jones et al., 1998). Bayesian optimisation models f using a Gaussian process $f \sim \mathrm{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$ where without loss of generality it is assumed that $\mu = 0$ and hence the Gaussian process is entirely specified by the kernel k. Bayesian optimisation is an iterative method that optimises a surrogate utility function (also known as acquisition function) whose role is to guide the optimiser to the optimum of the underlying function f in as few steps as possible.

The generic Bayesian optimisation function is as follows:

1. Set $t = 0$.
2. Find $\mathbf{x}_t^a = \underset{\mathbf{x} \in \mathcal{X}}{\operatorname{argmax}} \, \alpha(\mathbf{x} | \, \mathbf{X}_{A_t}, \mathbf{y}_{A_t})$.
3. Evaluate $y_t^a = f(\mathbf{x}_t^a)$.
4. Add new observation to A_t as $A_t = A_t \cup \{\mathbf{x}_t^a, y_t^a\}$
5. Set $t = t + 1$ and repeat from step 2 if $t < T$.

where α is the acquisition function and T is the maximum budget on the number of function evaluations. There are various acquisition functions e.g. probability of improvement (Kushner, 1964), expected improvement (Mockus, 2002) and Gaussian process upper confidence bound (GP-UCB) (Srinivas et al., 2012). The GP-UCB is especially amenable to theoretical analysis and is defined as

$$\alpha(\mathbf{x}) = \mu_{t-1}(\mathbf{x}) + \sqrt{\beta_t} \sigma_{t-1}(\mathbf{x}),$$

where β_t are a sequence of constants. The first term in this acquisition function favours exploitation of predicted maxima, and the latter exploration of unknown regions.

2.3 Experimental Design, Information Gain, and Regret Bounds

In experimental design (ED) (Chaloner and Verdinelli, 1995), information gain measures how informative a dataset $\{(\mathbf{x}_i^a, y_i^a) | i \in \mathbb{Z}_T\}$ generated from $y_t^a = f(\mathbf{x}_t^a) + \epsilon_a$, where $\epsilon_a \sim \mathcal{N}(0, \sigma_a^2)$, is about a function f. Information gain is defined to be the mutual information between f and $\mathbf{y}_{A_T}$ - that is,

$$\mathbb{I}(\mathbf{y}_{A_T} | f) = \mathbb{H}(\mathbf{y}_{A_T}) - \mathbb{H}(\mathbf{y}_{A_T} | f).$$

For a Gaussian distribution $\mathbb{H}(\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})) = \frac{1}{2}\log|2\pi e\boldsymbol{\Sigma}|$ (Cover and Thomas, 2012), and hence

$$\mathbb{I}(\mathbf{y}_{A_T} | f) = \tfrac{1}{2}\log\left|\mathbf{I} + \sigma_a^{-2}\mathbf{K}_{A_T}\right|.$$

In Bayesian optimisation, regret measures the distance from the optimal solution. For an optimiser following a sequence of points $\mathbf{x}_0^a, \mathbf{x}_1^a, \ldots$ the instantaneous regret at point t is

$$r_t = f(\mathbf{x}^*) - f(\mathbf{x}_t^a) \geq 0,$$

where $\mathbf{x}^*$ is the true maximum of f. The cumulative regret up to instance T is

$$R_T = \textstyle\sum_{t \in \mathbb{Z}_T} r_t.$$

In Srinivas et al. (2012) a number of statistical bounds are presented for the total regret in a GP-UCB context. These bounds take the form

$$\Pr\left\{R_T \leq \sqrt{C_1 \beta_T \gamma_T T} + c_0 \; \forall T \geq 1\right\} \geq 1 - \delta, \quad (3)$$

where the sequence $\beta_0, \beta_1, \ldots$ is specified, C_1 is a term dependent on measurement noise, and

$$\gamma_T = \max_{A_T \subset \mathcal{X} |\, |A_T| = T} \mathbb{I}(\mathbf{y}_{A_T} | \mathbf{f}_{A_T}) \quad (4)$$

is the maximum information gain where $\mathbf{f} = [f(\mathbf{x}_0^a), f(\mathbf{x}_1^a), \ldots, f(\mathbf{x}_{T-1}^a)]$. In the context of Bayesian optimisation our aim is to get (sufficiently close) to the optimum in the minimal number T of evaluations of f. The cumulative regret R_T provides a measure of closeness to optimality after T steps. Thus the term $\sqrt{C_1 \beta_T \gamma_T T}$ provides a bound on performance. In this paper we will demonstrate that both the maximum information gain γ_T and the term C_1 may be decreased through the use of transfer learning, enabling us to get closer to the optimum in fewer evaluations than in the non-transfer learning case.

3 Transfer Learning in Bayesian Optimisation

Standard Bayesian optimisation starts with no observations of the function f and proceeds with a sequence of test evaluations at points $\mathbf{x}_0^a, \mathbf{x}_1^a, \ldots$ to find the maximum of f. As such it suffers from a cold-start problem (Swersky et al., 2013; Joy et al., 2016). Transfer learning is a means of overcoming this problem and also of speeding up the convergence of the optimisation.

In the transfer learning case it is assumed that there is a set of source data $\{\mathbf{x}_i^s, y_i^s | i \in \mathbb{Z}_{N_s}\}$ given a-priori, where $y_i^s = f'(\mathbf{x}_i^s) + \epsilon'$, $\epsilon' \sim \mathcal{N}(0, \sigma'^2)$. Limited work has focused on transfer learning in Bayesian optimisation. Bardenet et al. (2013) built a transfer learning model under the rigid assumption that if $f'(x) \geq f'(y)$ for some x, y for the source function then $f(x) \geq f(y)$ for the target function. This strong assumption rarely holds in practice, e.g. this assumption is violated for any two functions where one is a lagged version of the other. Yogatama and Mann (2014) built a transfer learning model to utilise past data assuming that the deviations of a function from its mean are transferable. Once again, this assumption holds only for highly similar functions.

We present two approaches to transfer learning in this section. The first, envelope-stretching Bayesian optimisation (Env-GP), was previously presented in Joy et al. (2016). The latter, difference-modelling Bayesian optimisation (Diff-GP), is presented here for the first time. The key difference between Env-GP and Diff-GP is illustrated in Figure 1:

- Env-GP: the Env-GP method models the source data using the target function by treating the difference between the source and target functions as part of the noise model - that is, $y_i^s = f(\mathbf{x}_i^s) + \epsilon_s$, $\epsilon_s \sim \mathcal{N}(0, \sigma_s^2)$, where $\sigma_s^2 > \sigma'^2$ is sufficiently large to incorporate any source/target differences as a noise term.

- Diff-GP: the Diff-GP method explicitly models the difference between source and target functions as a Gaussian process and uses the prior obtained to construct a bias-corrected source data set. This bias-corrected source data may then be used directly for the target without further stretching the envelope.

In sections 3.1 and 3.2, we describe both these methods. Then in section 3.3, following Srinivas et al. (2012) we present a theoretical analysis of both methods in terms of regret bounds, demonstrating how both proposed methods result in a tighter regret bound than the no transfer learning case.

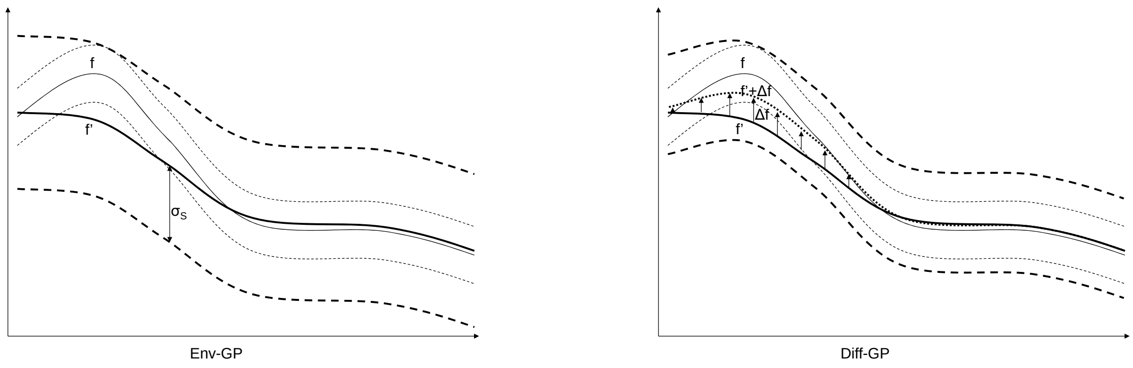

Figure 1: Env-GP and Diff-GP operation. Env-GP (left figure) works by *stretching the envelope* about the source function f' to encompass the target function f so that source data can be modelled in terms of the target function. Diff-GP (right figure), by contrast, constructs a bias-corrected source $f' + \Delta f$ to better match the target.

3.1 Transfer Learning in Bayesian Optimisation 1: Env-GP

By definition $y_i^s = f'(\mathbf{x}_i^s) + \epsilon'$ and $f, f' \sim \mathrm{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$. It follows that there exists $\sigma_s \geq \sigma'$ such that the observations y_i^s may be modelled by $y_i^s = f(\mathbf{x}_i^s) + \epsilon_s$, where $\epsilon_s \sim \mathcal{N}(0, \sigma_s^2)$. The amount of "stretch" required to fit the source data to the target function depends on the magnitude of the difference between $f' - f$ between source and target functions.

Given a set of source data $\{\mathbf{x}_i^s, y_i^s | i \in \mathbb{Z}_{N_s}\}$ generated as described and an appropriate σ_s, the envelope-stretching Bayesian optimisation procedure (Env-GP) is:

1. Set $t = 0$.
2. Find $\mathbf{x}_t^a = \underset{\mathbf{x} \in \mathcal{X}}{\operatorname{argmax}}\, \alpha(\mathbf{x} | [\mathbf{X}_S, \mathbf{X}_{A_t}], [\mathbf{y}_S, \mathbf{y}_{A_t}])$.
3. Evaluate $y_t^a = f(\mathbf{x}_t^a)$.
4. Add new observation to A_t as $A_t = A_t \cup \{\mathbf{x}_t^a, y_t^a\}$
5. Set $t = t + 1$ and repeat from step 2 if $t < T$.

which differs from the standard Bayesian optimisation algorithm through the inclusion of source data at step 2. Note that $f\left(\mathbf{x} \middle| \left[\begin{array}{cc} \mathbf{X}_S & \mathbf{X}_{A_t} \end{array}\right], \left[\begin{array}{cc} \mathbf{y}_S & \mathbf{y}_{A_t} \end{array}\right]\right) \sim \mathcal{N}\left(\tilde{\mu}_t(\mathbf{x}), \tilde{\sigma}_t^2(\mathbf{x})\right)$, where

$$\begin{aligned} \tilde{\mu}_t(\mathbf{x}) &= \left[\begin{array}{c} \mathbf{k}_S(\mathbf{x}) \\ \mathbf{k}_{A_t}(\mathbf{x}) \end{array}\right]^{\mathrm{T}} \mathbf{Q}^{-1} \left[\begin{array}{c} \mathbf{y}_S \\ \mathbf{y}_{A_t} \end{array}\right], \\ \tilde{\sigma}_t^2(\mathbf{x}) &= k(\mathbf{x}, \mathbf{x}) - \left[\begin{array}{c} \mathbf{k}_S(\mathbf{x}) \\ \mathbf{k}_{A_t}(\mathbf{x}) \end{array}\right]^{\mathrm{T}} \mathbf{Q}^{-1} \left[\begin{array}{c} \mathbf{k}_S(\mathbf{x}) \\ \mathbf{k}_{A_t}(\mathbf{x}) \end{array}\right] \qquad (5) \\ \mathbf{Q} &= \left[\begin{array}{cc} \mathbf{K}_S + \sigma_s^2 \mathbf{I} & \mathbf{K}_{SA_t} \\ \mathbf{K}_{SA_t}^{\mathrm{T}} & \mathbf{K}_{A_t} + \sigma_a^2 \mathbf{I} \end{array}\right] \end{aligned}$$

and $\mathbf{y}_S = \left[\begin{array}{cccc} y_0^s & y_1^s & \ldots & y_{N_s-1}^s \end{array}\right]^{\mathrm{T}}$, $\mathbf{K}_S = \left[\begin{array}{c} k(\mathbf{x}_i^s, \mathbf{x}_j^s) \end{array}\right]_{i,j \in \mathbb{Z}_{N_s}}$, $\mathbf{X}_S = \left[\begin{array}{cccc} \mathbf{x}_0^s & \mathbf{x}_1^s & \ldots & \mathbf{x}_{N_s-1}^s \end{array}\right]$, $\mathbf{K}_{SA_t} = \left[\begin{array}{c} k(\mathbf{x}_i^s, \mathbf{x}_j^a) \end{array}\right]_{i \in \mathbb{Z}_{N_s}, j \in \mathbb{Z}_t}$, and $\mathbf{k}_S(\mathbf{x}) = \left[\begin{array}{c} k(\mathbf{x}_i^s, \mathbf{x}) \end{array}\right]_{i \in \mathbb{Z}_{N_s}}$. This method has been presented in Joy et al. (2016), which also proposed a method for estimating σ_s from observations.

3.2 Transfer Learning in Bayesian Optimisation 2: Diff-GP

Let $\{\mathbf{x}_i^s, y_i^s | i \in \mathbb{Z}_{N_s}\}$ be the source data generated by $y_i^s = f'(\mathbf{x}_i^s) + \epsilon'$, where $\epsilon' \sim \mathcal{N}(0, \sigma'^2)$. Then $f'(\mathbf{x} | \mathbf{X}_S, \mathbf{y}_S) \sim \mathcal{N}(\mu_S(\mathbf{x}), \sigma_S^2(\mathbf{x}))$, where

$$\begin{aligned} \mu_S(\mathbf{x}) &= \mathbf{k}_S^{\mathrm{T}}(\mathbf{x}) \left(\mathbf{K}_S + \sigma'^2 \mathbf{I}\right)^{-1} \mathbf{y}_S, \\ \sigma_S^2(\mathbf{x}) &= k(\mathbf{x}, \mathbf{x}) - \mathbf{k}_S^{\mathrm{T}}(\mathbf{x}) \left(\mathbf{K}_S + \sigma'^2 \mathbf{I}\right)^{-1} \mathbf{k}_S^{\mathrm{T}}(\mathbf{x}). \end{aligned}$$

and the same covariance function has been used for both f and f'. Let $\{\mathbf{x}_i^a, y_i^a | i \in \mathbb{Z}_t\}$ be the target data to time t. Defining $g(\mathbf{x}) = f(\mathbf{x}) - f'(\mathbf{x})$ and

$$\begin{aligned} \Delta y_i^a(\mathbf{x}_i^a) &= y_i^a - \mu_S(\mathbf{x}_i^a), \\ \Delta \mathbf{y}_{A_t}(\mathbf{X}_{A_t}) &= \left[\begin{array}{cccc} \Delta y_0^a & \Delta y_1^a & \ldots & \Delta y_{t-1}^a \end{array}\right]^{\mathrm{T}}, \end{aligned}$$

it follows that $\Delta y_i^a(\mathbf{x}) = g(\mathbf{x}) + \epsilon_g(\mathbf{x})$, where $\epsilon_g(\mathbf{x}) \sim \mathcal{N}(0, \sigma_a^2 + \sigma_S^2(\mathbf{x}))$, and $g(\mathbf{x} | \mathbf{X}_{A_t}, \Delta \mathbf{y}_{A_t}) \sim \mathcal{N}(\mu_{D_t}(\mathbf{x}), \sigma_{D_t}^2(\mathbf{x}))$, where

$$\begin{aligned} &\mu_{D_t}(\mathbf{x}) = \mathbf{k}_{A_t}^{\mathrm{T}}(\mathbf{x}) \left(\mathbf{K}_{A_t} + \left(\sigma_a^2 + \sigma_S^2(\mathbf{x})\right) \mathbf{I}\right)^{-1} \ldots \\ &\quad \ldots \Delta \mathbf{y}_{A_t}(\mathbf{X}_{A_t}), \\ &\sigma_{D_t}^2(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) \ldots \\ &\quad \ldots - \mathbf{k}_{A_t}^{\mathrm{T}}(\mathbf{x}) \left(\mathbf{K}_{A_t} + \left(\sigma_a^2 + \sigma_S^2(\mathbf{x})\right) \mathbf{I}\right)^{-1} \mathbf{k}_{A_t}(\mathbf{x}), \end{aligned}$$

and the subscript D_t indicates the value for the "difference" function $g(\mathbf{x})$. This allows us to construct the bias-corrected source data

$$\left\{\mathbf{x}_i^s, y_{i,t}^{cs}(\mathbf{x}_i^s) = y_i^s + \mu_{D_t}(\mathbf{x}_i^s) \middle| i \in \mathbb{Z}_{N_s}\right\}$$

where $y^{cs}_{i,t}(\mathbf{x}) = f(\mathbf{x}) + \epsilon_{s,t}(\mathbf{x})$ is the sum of the (noisy) source sample $y^s_i = f'(\mathbf{x}^s_i) + \epsilon'$ and a correction factor $\mu_{D_t}(\mathbf{x}^s_i)$ equal to the expected (mean) difference between target and source function at $\mathbf{x}^s_i$ and hence can be treated as a noisy sample of the target function; and $\epsilon_{s,t}(\mathbf{x}) \sim \mathcal{N}\left(0, \sigma'^2 + \sigma^2_{D_t}(\mathbf{x})\right)$. This bias-corrected source data may therefore be used for transfer learning in the Bayesian optimisation algorithm without additional stretching of the envelope. We note that the target mean $\mu_{D_t}(\mathbf{x}^s_i)$ will change with each addition of a new target observation, so the bias-corrected source data is a function of t, and moreover that there is some t-dependent envelope stretching built in to the bias-corrected source noise variance - i.e.

$$\sigma^2_{cs,t}(\mathbf{x}) = \sigma'^2 + \sigma^2_{D_t}(\mathbf{x}).$$

Given a set of source data the difference-modelling Bayesian optimisation procedure (Diff-GP) is:

1. Set $t = 0$.
2. Calculate the bias-corrected source data $\left\{ \mathbf{x}^s_i, y^{cs}_{i,t}(\mathbf{x}^s_i) = y^s_i + \mu_{D_t}(\mathbf{x}^s_i) \middle| i \in \mathbb{Z}_{N_s} \right\}$
3. Find $\mathbf{x}^a_t = \underset{\mathbf{x} \in \mathcal{X}}{\operatorname{argmax}} \, \alpha\left(\mathbf{x} | \left[\mathbf{X}_S, \mathbf{X}_{A_t}\right], \left[\mathbf{y}_{CS_t}, \mathbf{y}_{A_t}\right]\right)$, where $\mathbf{y}_{CS_t} = \left[\ y^{cs}_{0,t}(\mathbf{x}^s_0) \ldots y^{cs}_{N_s-1,t}\left(\mathbf{x}^s_{N_s-1}\right)\ \right]^{\mathrm{T}}$
4. Evaluate: $y^a_t = f(\mathbf{x}^a_t)$.
5. Add new observation to A_t as $A_t = A_t \cup \{\mathbf{x}^a_t, y^a_t\}$
6. Set $t = t + 1$ and repeat from step 2 if $t < T$.

Note that $f\left(\mathbf{x} | \left[\ \mathbf{X}_S \quad \mathbf{X}_{A_t}\ \right], \left[\ \mathbf{y}_{CS_t} \quad \mathbf{y}_{A_t}\ \right]\right) \sim \mathcal{N}\left(\breve{\mu}_t(\mathbf{x}), \breve{\sigma}^2_t(\mathbf{x})\right)$, where

$$\begin{aligned}
\breve{\mu}_t(\mathbf{x}) &= \left[\begin{array}{c} \mathbf{k}_S(\mathbf{x}) \\ \mathbf{k}_{A_t}(\mathbf{x}) \end{array}\right]^{\mathrm{T}} \mathbf{R}^{-1} \left[\begin{array}{c} \mathbf{y}_{CS_t} \\ \mathbf{y}_{A_t} \end{array}\right] \\
\breve{\sigma}^2_t(\mathbf{x}) &= k(\mathbf{x}, \mathbf{x}) - \left[\begin{array}{c} \mathbf{k}_S(\mathbf{x}) \\ \mathbf{k}_{A_t}(\mathbf{x}) \end{array}\right]^{\mathrm{T}} \mathbf{R}^{-1} \left[\begin{array}{c} \mathbf{k}_S(\mathbf{x}) \\ \mathbf{k}_{A_t}(\mathbf{x}) \end{array}\right], \\
\mathbf{R} &= \left[\begin{array}{cc} \mathbf{K}_S + \mathbf{\Sigma}^2_{cs,t} & \mathbf{K}_{SA_t} \\ \mathbf{K}^{\mathrm{T}}_{SA_t} & \mathbf{K}_{A_t} + \sigma^2_a \mathbf{I} \end{array}\right]
\end{aligned} \tag{6}$$

and

$$\mathbf{\Sigma}^2_{cs,t} = \operatorname{diag}\left(\sigma^2_{cs,t}(\mathbf{x}^s_0), \sigma^2_{cs,t}(\mathbf{x}^s_1), \ldots, \sigma^2_{cs,t}\left(\mathbf{x}^s_{N_s-1}\right)\right)$$

3.3 Regret Bounds

Our aim in this section is to study the effect of source data in the GP-UCB transfer learning scenario on the regret bound (3). Recall that the regret bound of interest has the form

$$\Pr\{R_T \leq \sqrt{C_1 \beta_T \gamma_T T} + c_0 \ \forall T \geq 1\} \geq 1 - \delta$$

where the sequence $\beta_0, \beta_1, \ldots$ is specified, C_1 is a term dependent on measurement noise, and γ_T is the maximum information gain. We consider the impact of the source data separately on the maximum information gain γ_T and C_1 and demonstrate that both are decreased in the transfer learning case for both Env-GP and Diff-GP, and hence that the bounding term $\sqrt{C_1 \beta_T \gamma_T T}$ on the convergence of the GP-UCB error bound is tighter in the presence of source data.

Theorem 1. *Let γ_T be the maximum information gain of the GP-UCB in the standard (no transfer learning) case and $\tilde{\gamma}_T$ the maximum information gain in the Env-GP (or Diff-GP) case, assuming $N_s > 0$. Then $\tilde{\gamma}_T < \gamma_T$.*

Proof. See appendix A. □

Theorem 2. *Let C_1 be the relevant term in the regret bound (3) of the GP-UCB in the standard (no transfer learning) case and $\tilde{C}_1$ the same term in the Env-GP (or Diff-GP) case, assuming $N_s > 0$. Then $\tilde{C}_1 < C_1$.*

Proof. See appendix A. □

Theorems 1 and 2 together imply that $\sqrt{\tilde{C}_1 \beta_T \tilde{\gamma}_T T} < \sqrt{C_1 \beta_T \gamma_T T}$, and hence, in both the Env-GP and Diff-GP transfer learning algorithms we have a tighter regret bound in the transfer learning case,

$$\Pr\{R_T \leq \sqrt{\tilde{C}_1 \beta_T \tilde{\gamma}_T T} + c_0 \ \forall T \geq 1\} \geq 1 - \delta, \tag{7}$$

compared to the bound (3) for the non-transfer learning case. This implies that both Env-GP and Diff-GP should be able to find the optimal solution more quickly than the standard (no transfer learning) case, on average.

Of course if the source function f' is sufficiently different from the target function f, or the source points are located too far from the region where f is optimal, then it may be assumed that there is no useful speed-up to be gained from the application of transfer learning. For the Env-GP algorithm in this case we may expect that the envelope will need to be stretched significantly (so σ_s will be large). The following theorem shows that in such situations the benefits of transfer learning vanish:

Theorem 3. *Using the notation of Theorems 1 and 2,*

$$\lim_{\tilde{\sigma}_s \to \infty} \sqrt{\tilde{C}_1 \beta_T \tilde{\gamma}_T T} = \sqrt{C_1 \beta_T \gamma_T T},$$

where $\tilde{\sigma}_s = \sigma_s$ in the Env-GP case, $\tilde{\sigma}_s = \sigma_{cs,t}$ in the Diff-GP case.

Proof. See appendix A. □

The response of the Diff-GP algorithm in such circumstances is less clear. It may be that $\sigma_{cs,t}(\mathbf{x})$ (like σ_s in the Env-GP case) will become large, in which case from Theorem 3 we may expect the benefit to vanish. However this depends entirely on whether Diff-GP is (a) able to *learn* the difference between f' and f and (b) whether the source data samples $\mathbf{x}_i^s$ are close to the optimum of the target function. For example, if the difference between f and f' is simple - for example, if $f(\mathbf{x}) = f'(\mathbf{x}) + \text{const}$ - and the source samples are close to the target optimum then the Diff-GP algorithm may be expected to perform well even though $\|f - f'\|_\infty$ may be large.

4 Experimental Results

We consider two sets of experiments here. The first considers a simulated data set where we can control the similarity of the source and target functions. The target and source functions are

$$f(\mathbf{x}) = \exp(-\tfrac{1}{2}|\mathbf{x} - \mu\mathbf{1}|^2) \text{ and}$$
$$f'(\mathbf{x}) = \exp(-\tfrac{1}{2}|\mathbf{x} - \mu'\mathbf{1}|^2),$$

where $\mu' = \mu + \frac{s}{\sqrt{n}}$, s is the shift factor and we have fixed the dimensionality to $n = 2$. By tuning s between 0 and 2 we are able to adjust the similarity between f and f' directly, where $s = 0$ implies identical source and target functions and $s = 2$ gives very dissimilar functions. 20 source observations were generated uniformly randomly. Noise for both source and target measurements was fixed with standard deviations $\sigma = \sigma' = 0.1$. For comparative purposes all simulations in the simulated data were run for $T = 20$ samples. The squared exponential kernel of the form $k(\mathbf{x}, \mathbf{x}') = \exp(-\frac{1}{2\nu^2}\|\mathbf{x} - \mathbf{x}'\|^2)$ has been used for all experiments with the length scale ν set to 0.1.

Results for the simulated dataset are shown in Figure 2. As may be seen from these graphs both the Env-GP and Diff-GP algorithms outperformed the no-transfer-learning case for source shifts up to at least $s = 0.5$. It may be noted that for small s (up to approximately $s = 0.2$) the Env-GP and Diff-GP algorithm performed very similarly. However for moderate shifts ($s = 0.4$ and $s = 0.5$) Diff-GP outperforms Env-GP, which supports our earlier discussion regarding the advantages of Diff-GP when the difference between source and target is moderate but simple in structure (or learnable).

In our second experiment we performed tuning of hyperparameters for a support vector machine. For this experiment we have used the 7-class UCI Image Segmentation Dataset (Lichman, 2013). The data was normalised to zero mean, unit variance on each feature and split into 70/30 ratio for training and validation purposes. Our goal was to learn one-vs-rest classifiers for all the classes. The first learning task (i.e. class 1-vs-rest) was used as a source function and all others (e.g. class 2-vs-rest) were separately treated as target functions. We used the LibSVM (Chang and Lin, 2011) toolbox, which has two hyperparameters for SVM using rbf kernel: kernel scale γ and the cost parameter C. Both were varied from $[10^{-1}, 10^4]$. The functions are learnt in the exponent space, with range $[-1, 4]$ for both the hyperparameters. The source function was sampled on the full integer grid, whilst the target functions were optimized over the continuous space. Figure 3 shows the results of performance on the validation set vs evaluations for two transfer learning and the no-transfer learning (i.e. standard Bayesian optimization using only target data) algorithms. Evidently, within a fixed number of evaluations transfer learning algorithms (Env-GP and Diff-GP) are able to suggest better hyperparameters compared to no-transfer learning algorithm. When comparing the two transfer learning algorithms, we note that in four out of six cases, the Diff-GP algorithm converged significantly faster than the Env-GP algorithm, resulting in higher accuracy performance with the budget of 20 iterations.

5 Conclusion

This paper has derived the regret bounds for two transfer learning algorithms in Bayesian optimisation using GP-UCB as the acquisition function. The first algorithm (Env-GP) models the difference between the source and target functions as a noise process, whereas the second algorithm (Diff-GP) models the difference as a Gaussian process that is in turn used to correct targets in the source data. In addition to the regret bound the algorithms have been verified using both synthetic and real data to demonstrate the utility of these transfer learning methods. Our future work will examine the problem of how to derive similar transfer learning enabled tighter regret bounds for other acquisition functions such as expected improvement, predictive entropy search and so on.

A Proof of Theorems 1-3

Theorem 1: The maximum information gain for the non transfer learning case is given by (4), where $\mathbb{I}(\mathbf{y}_{A_T}|f) = \frac{1}{2}\log|\mathbf{I} + \sigma_a^{-2}\mathbf{K}_{A_T}|$. The distribution of the combined source/target datasets in the transfer learning case is

$$\begin{bmatrix} \mathbf{y}_S \\ \mathbf{f}_{A_T} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{K}_S + \tilde{\mathbf{\Sigma}}_s^2 & \mathbf{K}_{SA_T} \\ \mathbf{K}_{SA_T}^{\mathrm{T}} & \mathbf{K}_{A_T} \end{bmatrix}\right),$$

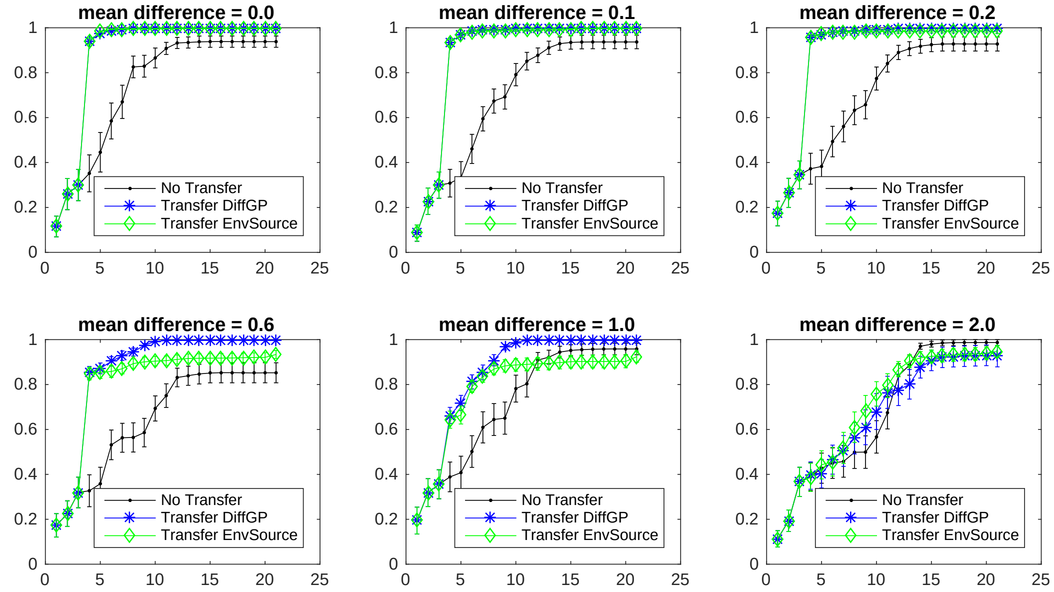

Figure 2: Simulated data results showing efficacy of transfer learning as a function of $f' - f$. In each graph the x-axis shows evaluations and the y-axis shows the "best" solution found to that point. For all the graphs, the results are averaged over 20 optimization trials, each starting with 3 random observations from the target function. Error bars denote the standard errors.

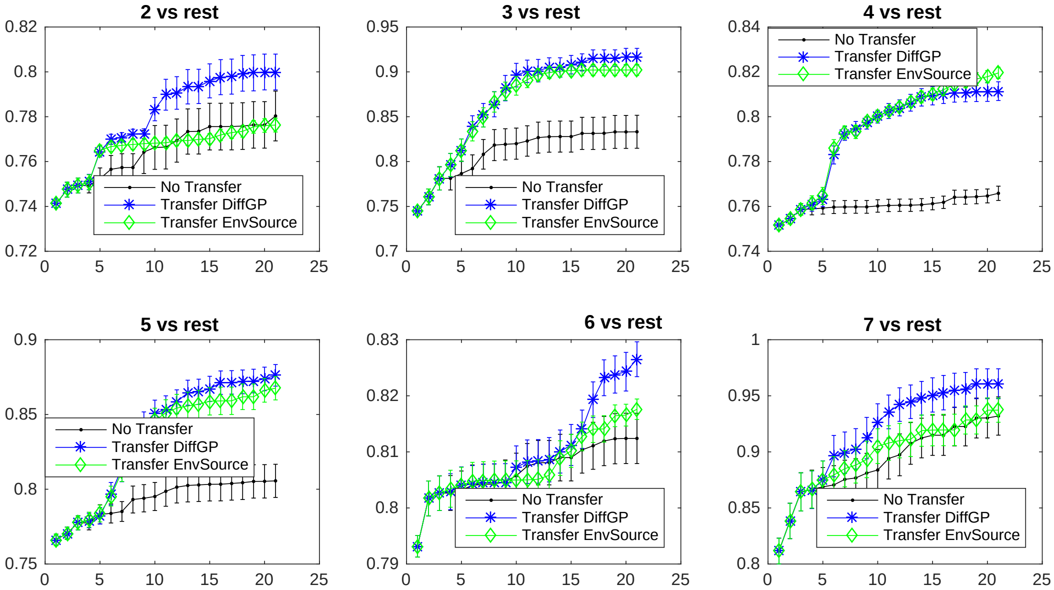

Figure 3: Hyperparameter tuning results for SVM with rbf kernel on UCI Image Segmentation dataset. Hyperparameter-vs accuracy for Class 1 vs rest is used as a source function whilst all other one-vs-rest tuning problems are separately treated as target functions. In each graph the x-axis shows evaluations and the y-axis shows the "best" solution found to that point. For all the graphs, the results are averaged over 20 optimization trials, each starting with 3 random observations from the target function. Error bars denote the standard errors.

where $\tilde{\mathbf{\Sigma}}_s^2 = \sigma_s^2\mathbf{I}$ in the Env-GP case and $\tilde{\mathbf{\Sigma}}_s^2 = \mathrm{diag}\left(\sigma_{cs,t}^2\left(\mathbf{x}_s\right)\right)$ in the Diff-GP case. It follows that, in the transfer learning case, $\mathbf{f}_{A_T} \sim \mathcal{N}(\tilde{\mathbf{m}}_{A_T}, \tilde{\mathbf{K}}_{A_T})$, where

$$\begin{aligned} \tilde{\mathbf{m}}_{A_T} &= \mathbf{K}_{SA_T}^{\mathrm{T}}\left(\mathbf{K}_S + \tilde{\mathbf{\Sigma}}_s^2\right)^{-1}\mathbf{f}_{A_T}, \\ \tilde{\mathbf{K}}_{A_T} &= \mathbf{K}_{A_T} - \mathbf{K}_{SA_T}^{\mathrm{T}}\left(\mathbf{K}_S + \tilde{\mathbf{\Sigma}}_s^2\right)^{-1}\mathbf{K}_{SA_T} \end{aligned} \tag{8}$$

and by definition $\mathbf{K}_A$ and $\mathbf{K}_S$ are both positive semi-definite. Assume without loss of generality that $\tilde{\mathbf{m}}_{A_T} = \mathbf{0}$. The information gain in the transfer learning case is therefore $\tilde{\mathbb{I}}\left(\mathbf{y}_{A_T}; f\right) = \frac{1}{2}\log|\mathbf{I} + \sigma_a^{-2}\tilde{\mathbf{K}}_{A_T}|$, where it should be noted that the sequence of points $\mathbf{x}_0^a, \mathbf{x}_1^a, \ldots, \mathbf{x}_{T-1}^a$ will in general be different from the standard (non transfer learning) case. Using the fact that $|\mathbf{A} + \mathbf{B}| \geq |\mathbf{A}| + |\mathbf{B}|$ for positive definite matrices $\mathbf{A}$, $\mathbf{B}$,

$$\begin{aligned} \tilde{\mathbb{I}}\left(\mathbf{y}_{A_T}; f\right) &= \tfrac{1}{2}\log|\mathbf{I} + \sigma_a^{-2}\mathbf{K}_{A_T} - \ldots \\ &\quad \ldots\sigma_a^{-2}\mathbf{K}_{SA_T}^{\mathrm{T}}\left(\mathbf{K}_S + \tilde{\mathbf{\Sigma}}_s^2\mathbf{I}\right)^{-1}\mathbf{K}_{SA_T}| \\ &\leq \tfrac{1}{2}\log\left|\mathbf{I}+\sigma_a^{-2}\mathbf{K}_{A_T}\right|. \end{aligned} \tag{9}$$

Using (9), it follows that, defining $\tilde{\gamma}_T$ as the maximum information gain in the transfer learning case,

$$\begin{aligned} \tilde{\gamma}_T &= \max_{A_T \subset D : |A_T| = T} \tilde{\mathbb{I}}\left(\mathbf{y}_{A_T}; \mathbf{f}_{A_T}\right) \\ &= \max_{A_T \subset D : |A_T| = T} \tfrac{1}{2}\log|\mathbf{I} + \sigma_a^{-2}\mathbf{K}_{A_T} - \ldots \\ &\quad \ldots\sigma_a^{-2}\mathbf{K}_{SA_T}^{\mathrm{T}}\left(\mathbf{K}_S + \tilde{\mathbf{\Sigma}}_s^2\right)^{-1}\mathbf{K}_{SA_T}| \end{aligned}$$

Using the *Minkowski inequality* on determinants of two positive definite matrices A and B, we have $|A + B| \geq |A| + |B| > |A|$. Assuming $A = \mathbf{I} + \sigma_a^{-2}\mathbf{K}_{A_T} - \sigma_a^{-2}\mathbf{K}_{SA_T}^{\mathrm{T}}\left(\mathbf{K}_S + \tilde{\Sigma}_s^2\right)^{-1}\mathbf{K}_{SA_T}$ and $B = \sigma_a^{-2}\mathbf{K}_{SA_T}^{\mathrm{T}}\left(\mathbf{K}_S + \tilde{\Sigma}_s^2\right)^{-1}\mathbf{K}_{SA_T}$, we can write $\log|A| < \log|A + B|$ and thus conclude that $\tilde{\gamma}_T < \gamma_T$. This completes the proof.

Theorem 2: In the original derivation of (3) in Srinivas et al. (2012) the term C_1 arises in the context of Lemma 5.4 in Srinivas et al. (2012). A key step in the proof of this lemma is the observation that

$$\begin{aligned} \sigma_a^{-2}\sigma_{t-1}^2\left(\mathbf{x}_t\right) &\leq \sigma_a^{-2}k\left(\mathbf{x}_t, \mathbf{x}_t\right) \\ &\leq \sigma_a^{-2}\max\nolimits_{\mathbf{x}\in A} k\left(\mathbf{x}, \mathbf{x}\right) = \sigma_a^{-2}, \end{aligned}$$

which relies on the fact that $k\left(\mathbf{x}, \mathbf{x}\right) = 1$ for all $\mathbf{x} \in \mathcal{X}$. This leads to

$$C_1 = \tfrac{8}{\log\left(1+\sigma_a^{-2}\right)}\max\nolimits_{x\in A_T} k\left(\mathbf{x}, \mathbf{x}\right) = \tfrac{8}{\log\left(1+\sigma_a^{-2}\right)}.$$

In the context of transfer learning $\mathbf{K}_A$ is replaced by $\tilde{\mathbf{K}}_A$ as defined by (8), the diagonals of which are $\tilde{k}\left(\mathbf{x}_t, \mathbf{x}_t\right) = k\left(\mathbf{x}_t, \mathbf{x}_t\right) - \mathbf{k}_S^{\mathrm{T}}\left(\mathbf{x}_t\right)(\mathbf{K}_S + \tilde{\mathbf{\Sigma}}_s^2)^{-1}\mathbf{k}_S\left(\mathbf{x}_t\right) \in (0, 1)$. So, for transfer learning,

$$\begin{aligned} \sigma_a^{-2}\sigma_{t-1}^2\left(\mathbf{x}_t\right) &\leq \sigma_a^{-2}\tilde{k}\left(\mathbf{x}_t, \mathbf{x}_t\right) \\ &\leq \sigma_a^{-2}\max\nolimits_{\mathbf{x}\in A}\tilde{k}\left(\mathbf{x}, \mathbf{x}\right) \\ &< \sigma_a^{-2}, \end{aligned}$$

and hence, using the notation $\tilde{C}_1$ to distinguish from the non transfer learning term C_1,

$$\tilde{C}_1 = \tfrac{8}{\log\left(1+\sigma_a^{-2}\right)}\max\nolimits_{\mathbf{x}\in A_T}\tilde{k}\left(\mathbf{x}, \mathbf{x}\right) < C_1,$$

which completes the proof.

Theorem 3: By (8) we have $\tilde{\mathbf{K}}_{A_T} = \mathbf{K}_{A_T} - \tilde{\mathbf{\Sigma}}_s^{-2}\mathbf{K}_{SA_T}^{\mathrm{T}}(\tilde{\mathbf{\Sigma}}_s^{-2}\mathbf{K}_S + \mathbf{I})^{-1}\mathbf{K}_{SA_T}$. Noting that $\lim_{\tilde{\sigma}_s\to\infty}(\tilde{\mathbf{\Sigma}}_s^{-2}\mathbf{K}_S + \mathbf{I}) = \mathbf{I}$, it follows that $\lim_{\tilde{\sigma}_s\to\infty}\tilde{\mathbf{K}}_{A_T} = \mathbf{K}_{A_T}$. Hence it follows from the proofs of theorems 1 and 2 that $\tilde{C}_1 \to C_1$ and $\tilde{\gamma}_T \to \gamma_T$, and the result follows.

References

Bardenet, R., Brendel, M., Kégl, B., and Sebag, M. (2013). Collaborative hyperparameter tuning. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 199–207.

Chaloner, K. and Verdinelli, I. (1995). Bayesian experimental design: A review. *Statistical Science*, pages 273–304.

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Cover, T. M. and Thomas, J. A. (2012). *Elements of information theory*. John Wiley & Sons.

Jones, D. R., Schonlau, M., and Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492.

Joy, T. T., Rana, S., Gupta, S. K., and Venkatesh, S. (2016). Flexible transfer learning framework for bayesian optimisation. In *Advances in Knowledge Discovery and Data Mining*, pages 102–114. Springer.

Kushner, H. J. (1964). A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86(1):97–106.

Lichman, M. (2013). UCI machine learning repository.

Mockus, J. (2002). Bayesian heuristic approach to global optimization and examples. *Journal of Global Optimization*, 22(1-4):191–203.

Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359.

Rasmussen, C. E. (2006). Gaussian processes for machine learning.

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959.

Srinivas, N., Krause, A., Kakade, S., and Seeger, M. (2010). Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proc. International Conference on Machine Learning (ICML)*.

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. W. (2012). Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265.

Swersky, K., Snoek, J., and Adams, R. P. (2013). Multi-task bayesian optimization. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 26*, pages 2004–2012. Curran Associates, Inc.

Yogatama, D. and Mann, G. (2014). Efficient transfer learning method for automatic hyperparameter tuning. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTATS)*.