Expert Systems With Applications 115 (2019) 656–672

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A flexible transfer learning framework for Bayesian optimization with convergence guarantee

Tinu Theckel Joy*, Santu Rana, Sunil Gupta, Svetha Venkatesh

Centre for Pattern Recognition and Data Analytics, Deakin University, Geelong, Australia

ARTICLE INFO

Article history:
Received 6 May 2018
Revised 12 August 2018
Accepted 13 August 2018
Available online 13 August 2018

Keywords:
Bayesian optimization
Transfer learning
Gaussian process

ABSTRACT

Experimental optimization is prevalent in many areas of artificial intelligence including machine learning. Conventional methods like grid search and random search can be computationally demanding. Over the recent years, Bayesian optimization has emerged as an efficient technique for global optimization of black-box functions. However, a generic Bayesian optimization algorithm suffers from a "cold start" problem. It may struggle to find promising locations in the initial stages. We propose a novel transfer learning method for Bayesian optimization where we leverage the knowledge from an already completed source optimization task for the optimization of a target task. Assuming both the source and target functions lie in some proximity to each other, we model source data as noisy observations of the target function. The level of noise models the proximity or relatedness between the tasks. We provide a mechanism to compute the noise level from the data to automatically adjust for different relatedness between the source and target tasks. We then analyse the convergence properties of the proposed method using two popular acquisition functions. Our theoretical results show that the proposed method converges faster than a generic no-transfer Bayesian optimization. We demonstrate the effectiveness of our method empirically on the tasks of tuning the hyperparameters of three different machine learning algorithms. In all the experiments, our method outperforms state-of-the-art transfer learning and no-transfer Bayesian optimization methods.

1. Introduction

Experimental optimizations are ubiquitous in many areas of Artificial Intelligence (AI). An example from machine learning is tuning the hyperparameters of a deep neural network on a large data that can consume a significant amount of computational time and memory for training. The hyperparameters here are architectural parameters like number of neurons in a hidden layer, number of hidden layers and model parameters like learning rate of the stochastic gradient descent algorithm that learns the model. Conventional strategies such as grid search and random search become inefficient with a large number of hyperparameters.

Recently, Bayesian optimization has become popular as an efficient framework for tuning hyperparameters (Snoek, Larochelle, & Adams, 2012). Bayesian optimization offers efficient solutions for global optimization problems especially when function evaluation is expensive (Brochu, Cora, & De Freitas, 2010; Mockus, 1994; Shahriari, Swersky, Wang, Adams, & de Freitas, 2016). Other applications of Bayesian optimization include sequential experimental design (Brochu et al., 2010), learning optimal robot mechanics (Lizotte, Wang, Bowling, & Schuurmans, 2007), optimal sensor placement (Garnett, Osborne, & Roberts, 2010), environmental monitoring (Marchant & Ramos, 2012), synthetic gene design (González, Longworth, James, & Lawrence, 2015) and synthesizing polymer fibre materials (Li et al., 2017). Bayesian optimization can also be used in optimizing any expert systems that have hyperparameters. It has recently been applied in tuning the hyperparameters of a credit scoring system (Xia, Liu, Li, & Liu, 2017), and building an autonomous system for recommending new materials (Ohno, 2018).

Bayesian optimization uses a probabilistic framework to model the objective function. A non-parametric Gaussian process (GP) (Williams & Rasmussen, 2006) is often the default choice as a prior over the unknown function. Bayesian optimization then employs a surrogate utility function namely acquisition function to decide the next point for evaluation. Acquisition function strategically trades off exploration and exploitation to find the next point. It "explores" the regions where epistemic uncertainty about the function is high, and "exploits" regions where function values are expected to be

* Corresponding author.
E-mail addresses: ttheckel@deakin.edu.au (T. Theckel Joy), santu.rana@deakin.edu.au (S. Rana), sunil.gupta@deakin.edu.au (S. Gupta), svetha.venkatesh@deakin.edu.au (S. Venkatesh).

https://doi.org/10.1016/j.eswa.2018.08.023
0957-4174/

higher in a weighted manner. Unlike the original objective function, acquisition functions are analytic and cheap functions. This makes them amenable to the usual global optimization algorithm.

However, a generic Bayesian optimization may suffer from a "cold start" problem when it tackles a new optimization function especially if the input space is high dimensional or the objective function landscape is complex. Due to the absence of proper knowledge, it might struggle in the beginning and require more function evaluations before converging to promising locations. Initial samples thus add cost to the optimization without contributing much to the process. In AI applications like robotics, Bayesian optimization might struggle in the initial stages and therefore take more time to generalize to a good configuration. Similarly, hyperparameter optimization in machine learning can also be costly when the model is complex, and data is large. *Reducing the cold start time hence remains an important problem to solve.*

Bayesian optimization operates by balancing two strategies, exploration of unknown region and exploitation of predicted good region. Most of the functions have a small good region and a large swath of low value region. Initially when we start with random samples, they will be low value with high probability, and hence there will not be much to exploit. Therefore, initially, Bayesian optimization algorithm mostly performs exploration, which is more commonly known as the cold start problem. One can largely reduce this cold start problem by providing knowledge from related tasks. Using this knowledge, one can incorporate better idea about the good areas of the function, and hence avoid the cold start problem to a large extend.

There are different models developed in this context. Bardenet, Brendel, Kégl, and Sebag (2013) developed a transfer learning method where a surrogate ranking scheme is used to optimize similar tasks. A Gaussian process is used to build a common ranking scheme for hyperparameters from different tasks. Bardenet et al. (2013) assume strong similarity in ranking function across the tasks. Yogatama and Mann (2014) developed a method that utilizes the knowledge from the source tasks by modeling the deviations from the average performance of different hyperparameters per task. Their method also assumes higher similarity in the deviations from the means of the previous tasks.*Additionally, none of them has provided theoretical guarantees on convergence. Hence, transfer learning for Bayesian optimization, which can handle differently related tasks and provide theoretical guarantees, is still an open problem.*

Addressing this, we develop a new framework for transfer learning. We assume the source task and target task lie within some proximity to each such that they become similar within an appropriate noisy envelope. Both of the functions are assumed to be same within the noise envelope. This practically allows us to use source data as noisy measurements for target function. We adjust the width of the envelope to be smaller when the tasks are closely related. We stretch the envelope further when tasks are only mildly related. We visualize this idea of the envelope in Fig. 1. We show two scenarios where the source and target task differ in relatedness. When the tasks are similar, the width of the envelope is small as shown in Fig. 1a. One can notice that this envelope is enough to encompass both the tasks. However, when the tasks are only mildly related, we accommodate the tasks by increasing the width of the envelope as shown in Fig. 1b. This way, we envisage a scheme where we adjust the envelope to accommodate source and target tasks.

When information is correct (tasks are similar), then Bayesian optimization would recommend better samples, providing faster convergence. Our method ensures that the information added remains correct by providing a mechanism to make that zero when tasks are different. When the tasks are totally different, the envelopes will be infinitely wider and the observations from the source task will be ignored for optimization and it will roll back to a generic Bayesian optimization scheme. This adaptive behavior underpins the flexibility of our framework to address different relatedness across tasks and reach a decision on either transferring or discarding the knowledge from the source task.

A constructive example could be handwritten digit recognition and the varying difficulty of distinguishing between digits. For example, 1 vs 2 or 1 vs 5 may have similar complexity requirement of the classifiers (similar sets of hyperparameters) as the digits are quite distinct, and hence require simpler models. On the contrary 5 vs 6 may require more complex models (different sets of hyperparameters). When the two tasks are similar, our method uses a smaller noise envelope that reflects the similarity between the two tasks. When we have to use different sets of hyperparameters (different tasks), we use a higher noise envelope in our method. Basically, the noise envelope helps in adding the observations from the source task with some level of uncertainty that reflects our belief on the similarity between the source and target task.

To realize our proposed framework, we model source task as noisy observations of the target task. We modify the covariance matrix of the Gaussian process where source points are added with more noise. We then estimate the noise variance for the source envelope from the observational data in a Bayesian setting. Joy, Rana, Gupta, and Venkatesh (2016) have reported a preliminary study of the proposed method. Current paper ushers in deriving theoretical guarantees on the convergence of the proposed method.

We analyse the convergence of our algorithm using both Gaussian process upper confidence bound (GP-UCB) (Srinivas, Krause, Kakade, & Seeger, 2010) and Expected improvement (EI) (Mockus, Tiesis, & Zilinskas, 1978) acquisition functions. Srinivas et al. (2010) and Wang and de Freitas (2014) provide theoretical guarantees for both GP-UCB and EI in a no-transfer setting respectively. They derive an upper bound on the cumulative regret and show that the growth in regret is sublinear. Cumulative regret is the sum of instantaneous regret which is the difference between the global optimum and the current observation. We derive a tighter upper bound on the cumulative regret for both the acquisition functions when our proposed transfer learning algorithm is used. Our bounds show improved convergence properties of the proposed algorithm.

We demonstrate the flexibility of our method simulating scenarios where the tasks are either very similar or only mildly related. We further employ our algorithm in tuning the hyperparameters of three machine learning algorithms. We develop a novel hyperparameter tuning setup where we select a small fraction of the training data for the source task and the whole for the target. Both of these tasks are evaluated on a held out validation data. The observations for the source can be generated cheaply since it uses only a small fraction of the training data. We then utilize this knowledge to tune the hyperparameters for the target task. Here the tasks differ in functional complexity even though they are from the same data distribution. In the context of hyperparameter tuning, we also evaluate our method on the tasks where the source and target data are from different data distributions. We select two state-of-the-art transfer learning methods (Bardenet et al., 2013; Yogatama & Mann, 2014) and the generic no-transfer Bayesian optimization method as the baselines for our experiments.

The sketch of the paper is as follows: we present related background on Bayesian optimization in Section 2. Section 3 describes the proposed method and analyze its convergence properties. We further detail the experimental set-up and results in Section 4. We finally conclude our work in Section 5.

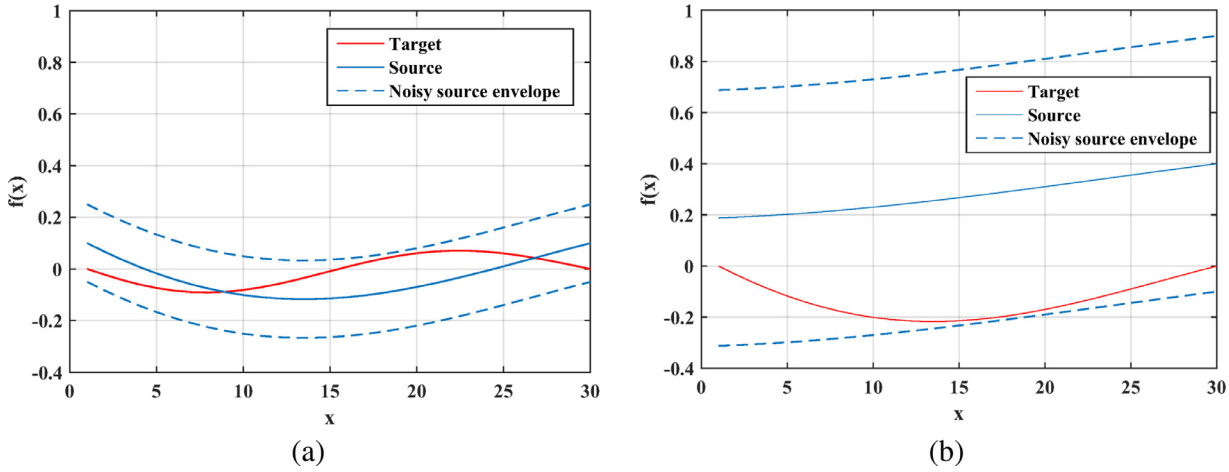

Fig. 1. An illustration of proposed method using two example scenarios: (a) tasks are closely related and (b) tasks are only mildly related. Red line represents target task, blue line represents source task, and blue dashed line represents the stretched envelope of the source task. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Symbols and description.

Symbol	Description
$\mathbf{x}^*$	Global maximum
$k(\mathbf{x}, \mathbf{x}')$	Covariance between $\mathbf{x}$ and $\mathbf{x}'$
θ	Length-scale of SE kernel
$\mathbf{K}$	Covariance matrix
$\mu(\mathbf{x})$	Posterior predictive mean
$\sigma^2(x)$	Posterior predictive variance
σ^2	Noise variance
α	Acquisition function
$\mathbf{x}^+$	Current best observation in EI
μ^+	Maximum predictive posterior mean over data D
r_t	Instantaneous regret at any iteration t
R_T	Cumulative regret after iteration T
s	Source task
σ_s^2	Noise variance for source task

2. Related background

Bayesian optimization is an efficient method for the global optimization of unknown objective functions $f : \mathcal{X} \to \mathbb{R}$, where $\mathcal{X} \subset \mathbb{R}^d$. Formally, it solves:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$$

where $\mathcal{X}$ is a compact and convex set. One can often access only the perturbed evaluations of $f(.)$ which makes the optimization further difficult. We provide a brief description of Bayesian optimization below and introduce the symbols in Table 1.

2.1. Gaussian process

In Bayesian optimization, the unknown objective function is modeled using a Gaussian process (GP) (Williams & Rasmussen, 2006). GP is a powerful tool for specifying our assumptions over the space of smoothly varying functions. A finite collection of GP sample path can be modeled as a multivariate Gaussian distribution. The properties of the Gaussian distribution allow us to compute the predictive means and variances in closed form. It is specified by a mean function, $\mu(\mathbf{x})$ and covariance function, $k(\mathbf{x}, \mathbf{x}')$. A sample from a Gaussian process is a function given as:

$$f(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \tag{1}$$

where the function value at an arbitrary point $\mathbf{x}$ is a Gaussian distributed random variable. Without any loss in generality, the prior mean function can be assumed to be a zero function making the Gaussian process fully defined by the covariance function. A popular choice of covariance function is squared exponential (SE) function, given as:

$$k(\mathbf{x}, \mathbf{x}') = exp\left(-\frac{1}{2\theta}||\mathbf{x} - \mathbf{x}'||^2\right) \tag{2}$$

where θ is the length-scale parameter. We assume the length-scale to be isotropic in our method. Other popular covariance functions include Matérn kernel, rational quadratic kernel etc.

Let us denote a set of initial observations as $\mathcal{D}_0 := \{\mathbf{x}_i, y_i\}_{i=1}^{t_0}$. Often, the observations can be perturbed by a noise as, $y = f(\mathbf{x}) + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma_{msr}^2)$ where σ_{msr}^2 is a measurement noise. The function values $\mathbf{y}_{1:t_0}$ jointly follow a multivariate Gaussian distribution as, $\mathbf{y}_{1:t_0} \sim \mathcal{N}(\mathbf{0}, \mathbf{K})$, where covariance matrix $\mathbf{K}$ is given as:

$$\mathbf{K} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \dots & k(\mathbf{x}_1, \mathbf{x}_{t_0}) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_{t_0}, \mathbf{x}_1) & \dots & k(\mathbf{x}_{t_0}, \mathbf{x}_{t_0}) \end{bmatrix} \tag{3}$$

where diagonal elements are always equal to 1 because of our choice of the covariance function. For a new data point $\mathbf{x}_t$, let the function value be y_t. Using the properties of GP, $\mathbf{y}_{1:t_0}$ and y_t are jointly Gaussian and can be expressed as:

$$\begin{bmatrix} \mathbf{y}_{1:t_0} \\ \mathbf{y}_t \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \begin{bmatrix} \mathbf{K} + \sigma_{msr}^2\mathbf{I} & \mathbf{k} \\ \mathbf{k}^T & k(\mathbf{x}_t, \mathbf{x}_t + \sigma_{msr}^2\mathbf{I}) \end{bmatrix} \tag{4}$$

where $\mathbf{k} = [k(\mathbf{x}_1, \mathbf{x}_t)\, k(\mathbf{x}_2, \mathbf{x}_t)\, \dots\, k(\mathbf{x}_{t_0}, \mathbf{x}_{t_0})]$. The predictive distribution at a new point $(\mathbf{x}_t)$ can be expressed as:

$$p(y_t|\mathbf{x}_t, \mathbf{x}_{1:t_0}, \mathbf{y}_{1:t_0}) \sim \mathcal{N}(\mu(\mathbf{x}_t), \sigma^2(\mathbf{x}_t)) \tag{5}$$

where the predictive mean and the variance is given as:

$$\mu(\mathbf{x}_t) = \mathbf{k}^T[\mathbf{K} + \sigma_{msr}^2\mathbf{I}]^{-1}\mathbf{y}_{1:t_0} \tag{6}$$

$$\sigma^2(\mathbf{x}_t) = k(\mathbf{x}_t, \mathbf{x}_t) - \mathbf{k}^T[\mathbf{K} + \sigma_{msr}^2\mathbf{I}]^{-1}\mathbf{k} \tag{7}$$

2.2. Acquisition function

The next stage of Bayesian optimization employs a strategy similar to the expected utility in sequential experimental design and decision theory (Shahriari et al., 2016). The role of the utility functions is estimating the amount of information that a particular observation can provide (Lindley, 1956; Shahriari et al., 2016). In Bayesian optimization, these utility functions are known as

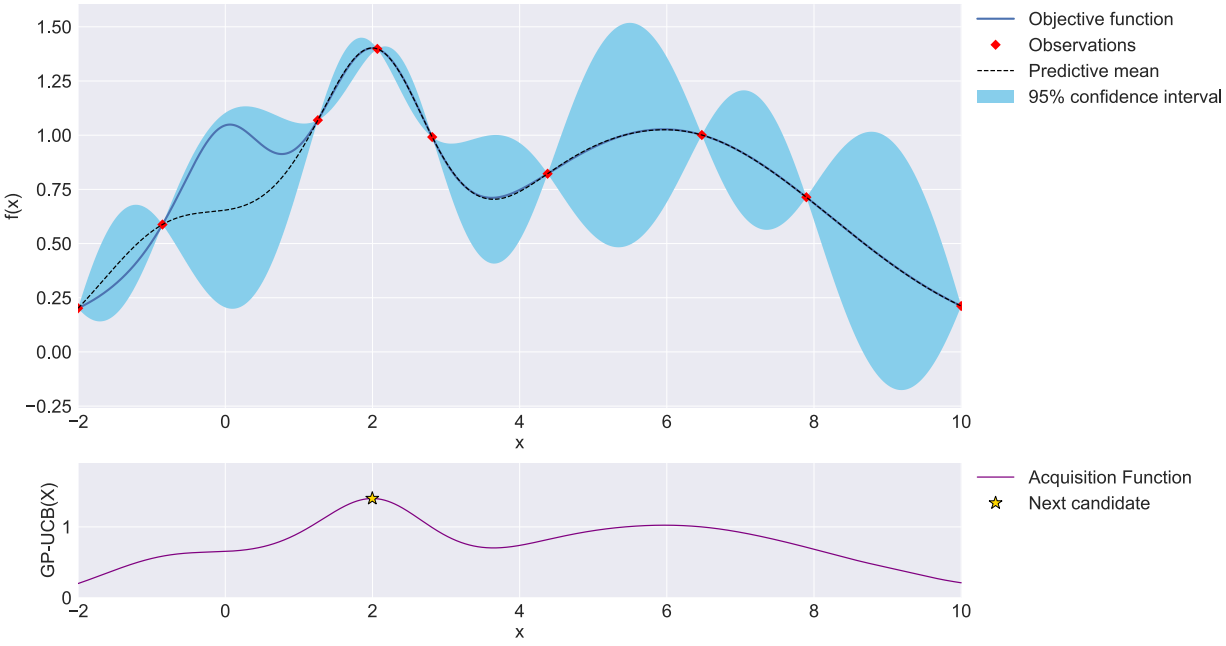

Fig. 2. An illustration of GP-UCB based Bayesian optimization in an example scenario.

acquisition functions. Acquisition function guides us to reach the optimum of the underlying function by exploring the areas where uncertainty about the function is high and exploiting the areas where the expected function values can be high. Essentially, one can maximize the acquisition function to extract the next point for evaluation.

Acquisition functions can be defined either using improvement based criteria or using confidence based criteria. Improvement based Probability of Improvement (PI) (Kushner, 1964) maximizes the probability of improvement over the current best observation. Similarly, another improvement based criteria, Expected Improvement (EI) (Mockus et al., 1978) maximizes the improvement over the current best observation in an expected sense. Confidence based GP-UCB (Srinivas et al., 2010) uses the upper confidence bound of the GP predictive distribution as an acquisition function. Nonetheless, the work of (Brochu et al., 2010) suggest using a mix of these acquisition functions. We use both GP-UCB and EI acquisition functions in the proposed method for their simplicity and guarantee on the convergence. However, proposed framework can be used with all the acquisition functions.

Gaussian process upper confidence bound (GP-UCB)

Srinivas et al. (2010) define GP-UCB as the upper confidence bound of Gaussian process. GP-UCB selects a point where posterior predictive mean (exploitation) and variance (exploration) are high. GP-UCB is defined as:

$$\alpha_t(\mathbf{x}) = \mu_{t-1}(\mathbf{x}) + \beta_t^{1/2}\sigma_{t-1}(\mathbf{x}) \tag{8}$$

where $\beta_t = 2\log(t^2 2\pi^2/(3\delta)) + 2d\log\left(t^2 dbr\sqrt{\log(4da/\delta)}\right)$, $\sum_{t\geq 1}\pi_t^{-1} = 1$, $\pi_t \geq 0$, a, b are constants and d is dimension of the problem and are given as, $a > 0$, $b > 0$, $d > 0$, $r > 0$, $\delta \in (0, 1)$, $t \geq 1$. The constants a, b are related to the Lipschitz constant of the objective function f as, $\Pr\left\{sup_{x\in\mathcal{D}_t} \mid \frac{\partial f}{\partial x_j} \mid > L\right\} \leq ae^{-bjL^2}$ for all $j = 1, 2, \ldots, d$.

We visualize both GP posterior and acquisition function in Fig. 2. We consider a simple function in one dimension. We plot the posterior mean with 95% confidence interval. Fig. 2 shows that GP-UCB selects a point where posterior predictive mean and variance are high.

Expected improvement (EI)

Let us assume our task is optimizing an unknown function $f(.)$. Let the initial observations be $\mathcal{D}_0 \equiv \{\mathbf{x}_{1:t_0}, \mathbf{y}_{1:t_0}\}$ and the current best observation be $\mathbf{x}^+ = \arg\max_{\mathbf{x}_i\in\mathcal{D}_0} f(\mathbf{x}_i)$. Please note that $\mathbf{x}^*$ indicates the global maximum of the objective function. Now, the improvement function is defined as:

$$I(\mathbf{x}) = \max\{0, f(\mathbf{x}) - f(\mathbf{x}^+)\} \tag{9}$$

EI is defined as the expected value of $I(\mathbf{x})$ (Mockus et al., 1978) as:

$$\alpha(\mathbf{x}) = \max_{\mathbf{x}\in\mathcal{X}} \mathbb{E}(I(\mathbf{x})|D)$$

The analytic form of $E(I(\mathbf{x}))$ can be obtained as (Jones, Schonlau, & Welch, 1998; Mockus et al., 1978):

$$\alpha_t(\mathbf{x}) = \begin{cases} (\mu_{t-1}(\mathbf{x}) - f(\mathbf{x}^+))\Phi(z) + \sigma_{t-1}(\mathbf{x})\phi(z) & \text{if } \sigma_{t-1}(\mathbf{x}) > 0 \\ 0 & \text{if } \sigma_{t-1}(\mathbf{x}) = 0 \end{cases} \tag{10}$$

where $z = (\mu(\mathbf{x}) - f(\mathbf{x}^+))/\sigma(\mathbf{x})$, $\Phi(.)$ and $\phi(.)$ are the CDF and PDF of a standard normal distribution respectively.

There is another variant of EI mentioned in the work of Wang and de Freitas (2014). Their work uses the maximum predictive mean instead of the function value of current best observation to define the improvement over the past samples. It is given as $\mu^+ = \max_{\mathbf{x}\in\mathcal{X}} \mu_{t-1}(\mathbf{x})$. This choice is reasonable especially when the optimization is carried out in a noisy setting. Based on this, a modified EI acquisition function is given as:

$$\alpha_t(\mathbf{x}) = \begin{cases} (\mu_{t-1}(\mathbf{x}) - \mu^+)\Phi(z) + \sigma_{t-1}(\mathbf{x})\phi(z) & \text{if } \sigma_t(\mathbf{x}) > 0 \\ 0 & \text{if } \sigma_t(\mathbf{x}) = 0 \end{cases} \tag{11}$$

where $z = (\mu_{t-1}(\mathbf{x}) - \mu^+)/\sigma_{t-1}(\mathbf{x})$. We use this modified acquisition function for the proposed method. To explain further, we visually illustrate EI acquisition function in an example scenario in Fig. 3. Current best observation, $\mathbf{x}^+$ and the expected improvement is also shown in Fig. 3. Essentially, EI recommends a point that offers maximum expected improvement over the current best observation. A generic Bayesian optimization algorithm without any transfer learning is presented in Algorithm 1.

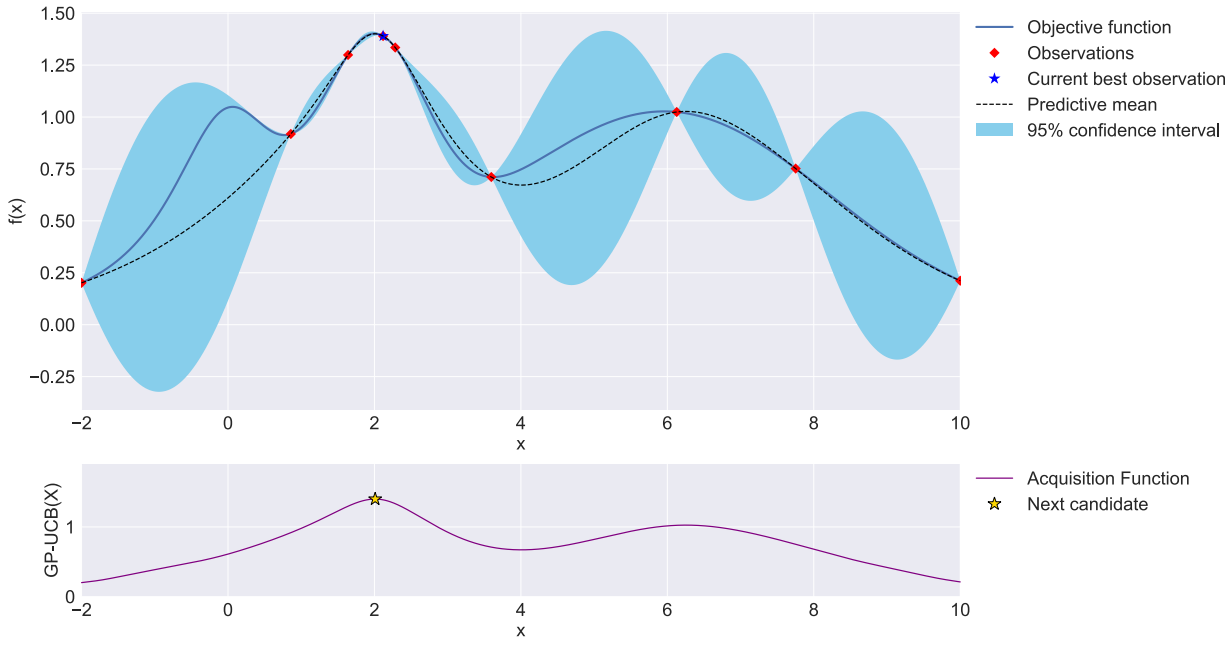

Fig. 3. An illustration of EI based Bayesian optimization in an example scenario.

Algorithm 1 A generic Bayesian optimization algorithm.

1: Initial observations: $\mathcal{D}_0 := \{\mathbf{x}_i, y_i\}_{i=1}^{t_0}$
2: Total budget: T iterations
3: **for** $t = 1, 2, \ldots, T$ **do**
4: $\mathbf{x}_t = \arg\max_{\mathbf{x}\in\mathcal{X}} \alpha_t(\mathbf{x}|\mathcal{GP}(\mathcal{D}_{t-1}))$ ▷ Maximizing acquisition function $\alpha(.)$
5: $y_t =$ Evaluate $f(.)$ at $\mathbf{x}_t$ ▷ Evaluating objective function
6: $\mathcal{D}_t := \mathcal{D}_{t-1} \cup \{\mathbf{x}_t, y_t\}$ ▷ Augmenting observation set
7: Update GP model: $\mathcal{GP}(\mathcal{D}_t)$
8: **end for**
9: Output: $\mathbf{x}^* = \arg\max_{\mathbf{x}_i\in\mathcal{D}_{i=1:T}} f(\mathbf{x}_i)$ ▷ Global optimum

2.3. Convergence analysis of a generic Bayesian optimization

In this section, we analyse the convergence guarantees of both GP-UCB and EI in a generic no-transfer setting. We now familiarize the terms used in the convergence analysis.

Regret

Instantaneous regret at any iteration t is defined as the difference between the best function value and the current observation which is given as:

$$r_t = f(\mathbf{x}^*) - f(\mathbf{x}_t)$$

where $\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$ and $\mathbf{x}_t$ is the observation at t. The cumulative regret till iteration T is defined as:

$$R_T = \sum_{t=1}^{T} r_t$$

Generally, cumulative regret is used to analyse the convergence of Bayesian optimization algorithms. This is useful since it accounts for information collected in each step towards optimizing an unknown expensive function. A no regret setting is an essential property of any global optimization algorithm and is given as, $\lim_{T\to\infty} R_T/T = 0$ (Srinivas et al., 2010). This means that the average cumulative regret becomes zero as T increases. This also shows that the growth in regret is sublinear, which is the best rate possible for a global optimization algorithm.

Information gain

In Bayesian optimization, information gain allows us to formulate the amount of information that one has gained about an unknown function. Information gain can generally be defined as the reduction in uncertainty about objective function $f(.)$ from observations $\mathbf{y}_\mathcal{A}$ corresponding to a set of points $\mathcal{A} \in \mathcal{X}$. It is given as:

$$\mathcal{I}(\mathbf{y}_\mathcal{A}; f(.)) = H(y_\mathcal{A}) - H(y_\mathcal{A} \mid f(.)) \tag{12}$$

where H is the entropy. Srinivas et al. (2010) and Wang and de Freitas (2014) used maximum information gain to establish an upper bound on cumulative regret. We denote maximum information gain as:

$$\gamma_T = \max_{\mathcal{A}\subset\mathcal{X}:|\mathcal{A}|=T} \mathcal{I}(\mathbf{y}_{1:T}; f(.)) \tag{13}$$

where T is number of iterations. Maximum information gain for Gaussian distribution can be derived as:

$$\gamma_T = \max_{\mathcal{A}\subset\mathcal{X}:|\mathcal{A}|=T} \frac{1}{2} \log |\, \mathbf{I} + \sigma^{-2}\mathbf{K}_\mathcal{A} \,| \tag{14}$$

We now provide a sketch for convergence analysis of GP-UCB and EI based Bayesian optimization.

2.3.1. Convergence analysis of GP-UCB

Srinivas et al. (2010) proved that GP-UCB follows a sublinear growth in the cumulative regret as the number of iterations increase. We recall the lemmas of Srinivas et al. (2010) to show a sketch of the proofs. We quickly recall the expression for GP-UCB as, $\alpha_t(\mathbf{x}) = \mu_{t-1}(\mathbf{x}) + \beta_t^{1/2}\sigma_{t-1}(\mathbf{x})$.

Lemma 1. *Using GP-UCB, for compact and convex set $\mathcal{X}$, the instantaneous regret at any iteration t is given as, $r_t \leq 2\beta_t^{1/2}\sigma_{t-1}(x_t) + \frac{1}{t^2}$ with high probability $\geq 1-\delta$ where $0 < \delta < 1$.*

Proof. We refer to the Lemmas 5.1, 5.2, 5.5 and 5.8 of Srinivas et al. (2010) for the detailed proof. Although being a probability, the inequality $\geq 1-\delta$ is an algebraic form and it is proved in Lemmas 5.1, and 5.5 of Srinivas et al. (2010). This has been further used as a probability in establishing the regret bound. The same reasoning has been followed in convergence analysis of EI acquisition function (Wang & de Freitas, 2014). □

Lemma 2. *The information gain for the points selected can be derived in terms of the predictive variances as:*

$$I(y_T;\mathbf{f}_T)=\frac{1}{2}\sum_{t=1}^{T}\log(1+\sigma^{-2}\sigma_{t-1}^2(\mathbf{x}_t))$$

Proof. Proof is detailed in Lemma 5.3 in the work of Srinivas et al. (2010). □

Lemma 3. *For* $\delta\in(0,1)$, *and* $\beta=2\log(t^2 2\pi^2/(3\delta))+2d\log\left(t^2 dbr\sqrt{\log(4da/\delta)}\right)$, *then the following holds with probability greater than* $1-\delta$*:*

$$\sum_{t=1}^{T}r_t^2\le\beta_T C_1 I(\mathbf{y}_T;f_T)\le C_1\beta_T\gamma_T$$

where $\forall T\ge 1$ *and* $C_1=\frac{8}{\log(1+\sigma^{-2})}$.

Proof. Proof is detailed in Lemma 5.4 of Srinivas et al. (2010). □

From these Lemmas, Theorem 2 of Srinivas et al. (2010) derives an upper bound on the cumulative regret as:

$$Pr\left(R_T\le\sqrt{C_1 T\beta_T\gamma_T+\frac{\pi}{\sqrt{6}}}\right)\ge 1-\delta$$

where C_1 is a term relating to noise variance σ^2, given as, $C_1=\frac{8}{\log(1+\sigma^{-2})}$ and $\sum\frac{1}{t^2}=\frac{\pi}{\sqrt{6}}$. This upper bound on the cumulative regret is for a generic Bayesian optimization without any knowledge from the previous tasks. With high probability, Srinivas et al. (2010) show that GP-UCB algorithm is a no regret method as $\lim_{T\to\infty}R_T/T=0$. For a squared exponential kernel, β_T grows with $O(\log(T))$ and γ_T follows an order of $O((\log(T))^{d+1})$. This establishes the sublinear property of GP-UCB based Bayesian optimization method.

2.3.2. Convergence analysis of EI

Wang and de Freitas (2014) derives an upper bound on the cumulative regret for EI based acquisition function. Their proof is similar to the work of Srinivas et al. (2010). We now recall the additional lemmas used by Wang and de Freitas (2014). Additionally, we select length-scale θ such that $\theta_L=\theta=\theta_U$ in order to satisfy the conditions in Lemma 6 and Theorem 1 of Wang and de Freitas (2014). Here *L, U* denote lower and upper bound on the length-scale.

Lemma 4. *For* $\mathcal{X}\subset\mathbb{R}^d$, *instantaneous regret for any sample* $\mathbf{x}$ *using EI acquisition function is defined as:*

$$r_t\le\left((2C_3+1)\varphi_t+\left(C_3+\sqrt{\log(\frac{t+\sigma^2}{\sigma^2})}\right)\nu_t\right)\sigma_{t-1}(\mathbf{x})$$

where $C_3\ge\frac{\iota(\varphi_{t-1}/\nu_t)}{\iota(-\varphi_{t-1}/\nu_t)}$, $\iota(z):=z\Phi(z)+\phi(z)$, $z\ge 0$, $(\varphi_t)^2:=C_2\parallel f\parallel_{\mathcal{H}(\mathcal{X})}+2\gamma_{t-1}+\sqrt{8}\log^{1/2}(2t^2\pi^2/3\delta)\left(\sqrt{C_2}\parallel f\parallel_{\mathcal{H}(\mathcal{X})}+\sqrt{\gamma_{t-1}}\right)+2\sigma\log(2t^2\pi^2/3\delta)$, $C_2:=\prod_{i=1}^{d}\frac{\theta_i^U}{\theta_i^L}$, $\nu_t=\Theta\left(\gamma_{t-1}+\log^{1/2}(2t^2\pi^2/3\delta)\sqrt{\gamma_{t-1}}+\log(t^2\pi^2/3\delta)\right)$, $t\ge 1$ *and* $\parallel f\parallel_{\mathcal{H}}$ *denotes the norm of a function* $f(.)$ *in Hilbert space.*

Proof. We refer to the results of Lemmas 9 and 10 of Wang and de Freitas (2014). In our formulation C_2 becomes, $C_2=1\because\theta_L=\theta=\theta_U$. □

Wang and de Freitas (2014) then combine the results from their Lemmas 6, 7 and 8 to derive an upper bound on the cumulative regret. A detailed proof is available in Wang and de Freitas (2014). With a probability of $1-\delta$, Theorem 1 of Wang and de Freitas (2014) shows that an EI based Bayesian optimization achieves an upper bound derived as:

$$Pr\left(R_T\le\beta_T\sqrt{T\gamma_T}\right)\ge 1-\delta \tag{15}$$

where $\beta_T=2\log(\frac{T}{\sigma^2})\gamma_{T-1}+\Lambda_T+\sqrt{\gamma_{T-1}}+C_3\parallel f\parallel_{\mathcal{H}(\mathcal{X})}$, $f(.)\in\mathcal{H}(\mathcal{X})$, $\Lambda_T=\sqrt{8}\log(\frac{T}{\sigma^2})\log^{1/2}(4T^2\pi^2/6\delta)\sqrt{C_3}\parallel f\parallel_{\mathcal{H}}$ and $T\ge 1$. Wang and de Freitas (2014) prove the convergence of the generic Bayesian optimization irrespective to the choice of a fixed length-scale and guarantee a sublinear growth similar to GP-UCB.

3. Proposed method

Let us assume that there is an already completed source optimization task with observations $\{\mathbf{x}_i^s,y_i^s\}_{i=1}^m$ where m denotes the total number of source observations. Let $f(.)$ be the target task and the objective is to optimize it as,

$$\mathbf{x}^*=\arg\max_{\mathbf{x}\in\mathcal{X}}f(\mathbf{x})$$

where $\mathcal{X}\subseteq\mathbb{R}^d$. We utilize the informations from the source task by modeling them as noisy measurements of the target function, and it is given as,

$$y_i^s=f(\mathbf{x}_i^s)+\epsilon_i^s,\forall i=1,\ldots,m \tag{16}$$

where $\epsilon^s\sim\mathcal{N}(0,\sigma_s^2)$ is the source noise with variance σ_s^2. This implies that we assume that the source function to lie within $3\sigma_s$ ball of the target function values with probability close to 1.

Let us denote the initial observations from the target task as $\{\mathbf{x}_i,y_i\}_{i=1}^{t_0}$. We combine data from both the source and target and create a combined observation set: $\mathcal{D}_0:=\{\{\mathbf{x}_i^s,y_i^s\}_{i=1}^m,\{\mathbf{x}_i,y_i\}_{i=1}^{t_0}\}$. The target GP is built using the combined observation. Without any loss in generality, we can model these observations using a prior zero mean function, and covariance function as, $\mathbf{f}\sim\mathcal{N}(\mathbf{0},\mathbf{K})$. We compute the covariance matrix $\mathbf{K}$ for the combined observations as,

$$\mathbf{K}=\begin{bmatrix}k(\mathbf{x}_1^s,\mathbf{x}_1^s) & \ldots & k(\mathbf{x}_1^s,\mathbf{x}_m^s) & k(\mathbf{x}_1^s,\mathbf{x}_1) & \ldots & k(\mathbf{x}_1^s,\mathbf{x}_{t_0})\\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots\\ k(\mathbf{x}_m^s,\mathbf{x}_1^s) & \ldots & k(\mathbf{x}_m^s,\mathbf{x}_m^s) & k(\mathbf{x}_m^s,\mathbf{x}_1) & \vdots & k(\mathbf{x}_m^s,\mathbf{x}_{t_0})\\ k(\mathbf{x}_1,\mathbf{x}_1^s) & \vdots & \vdots & k(\mathbf{x}_1,\mathbf{x}_1) & \vdots & k(\mathbf{x}_1,\mathbf{x}_{t_0})\\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots\\ k(\mathbf{x}_{t_0},\mathbf{x}_1^s) & \ldots & \ldots & k(\mathbf{x}_{t_0},\mathbf{x}_1) & \ldots & k(\mathbf{x}_{t_0},\mathbf{x}_{t_0})\end{bmatrix}$$

where $k(\mathbf{x}_1^s,\mathbf{x}_1^s)$ is self-covariance, and $k(\mathbf{x}_1^s,\mathbf{x}_1)$ denotes the covariance between source point $\mathbf{x}_1^s$ and target point $\mathbf{x}_1$. We now update the covariance matrix $\mathbf{K}$ to incorporate the source noise as,

$$\mathbf{K}_*=\mathbf{K}+\begin{bmatrix}\sigma_s^2\mathbf{I}_{m\times m} & \mathbf{0}\\ \mathbf{0}^T & \sigma_{msr}^2\mathbf{I}_{t_0\times t_0}\end{bmatrix} \tag{17}$$

where source noise variance σ_s^2 models the closeness between the source and the target function and σ_{msr}^2 is the measurement noise for the target. We add a higher noise variance to source points along the diagonal elements of the kernel matrix and these source points now act as noisy target observations. This allows us to use the same covariance function to capture the similarity between the observations from source and target tasks. Source noise variance, σ_s^2, reflects the relatedness between the tasks, and it should be set wisely based on the expected similarity between the source and the target. Intuitively, addition of source noise variance helps us to maintain more uncertainty over the source points than the target points.

Using the property of GP, the posterior predictive mean and variance for a new point $\mathbf{x}_t$ are given as,

$$\mu(\mathbf{x}_t)=\mathbf{k}^T\mathbf{K}_*^{-1}\mathbf{y} \tag{18}$$

$$\sigma^2(\mathbf{x}_t) = k(\mathbf{x}_t, \mathbf{x}_t) - \mathbf{k}^T \mathbf{K}_*^{-1} \mathbf{k} \tag{19}$$

where $\mathbf{K}_*$ is the modified covariance matrix. Every iteration, covariance matrix is recomputed using Eq. (17) and the Bayesian optimization is performed using the GP with this covariance matrix.

We now analyse how source noise variance σ_s^2 and target noise variance σ^2 are related to tasks of varying similarity.

Source Noise Variance and Relatedness across the Tasks

In this section, we analyse our algorithm under different values of σ_s^2. The value of σ_s^2 can significantly affect the efficiency of the proposed framework. When two tasks are the same, we set $\sigma_s^2 = \sigma^2$ and all the observations are treated equally. When source task and target task are different, we use $\sigma_s^2 > \sigma^2$ and σ_s^2 increases as the similarity between the tasks decreases. We use this setting to reflect the notion that these two set of observations are not derived from the same task. When the two tasks are not all related, source noise takes a very high value such that $\sigma_s^2 = \infty$. In this case, the data from the source task will be completely ignored while optimizing the target task and the proposed method rolls back to a generic no-transfer Bayesian optimization scheme.

In essence, the value of σ_s^2 is important to achieve optimal transfer learning. We further provide a data-driven approach to estimate its value from the observations.

Estimation of source noise variance (σ_s^2)

A naive solution is to use a fixed noise variance throughout the optimization procedure. However, this leads to poor performance since the relatedness can vary as we observe more observations from the target task and it needs to be modeled accordingly. This motivates us to devise a scheme for estimating the source noise automatically. We realize this by formulating a Bayesian framework that can estimate the source noise variance from the data.

Using the conjugate property, inverse gamma distribution is a conjugate prior to the variance of a normal distribution. Using this fact, we start by placing an inverse gamma distribution with parameters τ_0 and ν_0 as a prior distribution of σ_s^2, given as,

$$\sigma_s^2 \sim InvGamma(\tau_0, \nu_0) \tag{20}$$

We start with a wide prior and then update the posterior from the observation of output value of the target (y) and the predicted source value ($\hat{y}^s$) every time the method recommends a target sample $\mathbf{x}$. The source function is modeled with a Gaussian process to predict $\hat{y}_t^s$ at $\mathbf{x}_t$. The posterior is also an inverse gamma distribution with updated parameters τ_t and ν_t as,

$$p(\sigma_s^2 \mid \{y_i - \hat{y}_i^s\}_{i=1}^t) \sim InvGamma(\tau_t, \nu_t) \tag{21}$$

Assuming the mean of the difference in Eq. (21) to be zero, the parameters τ_t and ν_t is updated as,

$$\tau_t = \tau_0 + t/2 \tag{22}$$

$$\nu_t = \nu_0 + \frac{\sum_{i=1}^{t} (y_i - \hat{y}_i^s)^2}{2} \tag{23}$$

We use the mode of the posterior distribution as the value of source noise variance and it is given as,

$$\sigma_s^2 = \frac{\nu_t}{\tau_t + 1} \tag{24}$$

We present the proposed method in Algorithm 2.

We now establish convergence properties for the proposed using both GP-UCB and EI acquisition functions.

Algorithm 2 Proposed transfer learning algorithm.

1: Source observations: $\mathcal{D}^s := \{\mathbf{x}_i^s, y_i^s\}_{i=1}^m$
2: Initial target observations: $\{\mathbf{x}_i, y_i\}_{i=1}^{t_0}$
3: $\mathcal{D}_0 := \{\{\mathbf{x}_i^s, y_i^s\}_{i=1}^m, \{\mathbf{x}_i, y_i\}_{i=1}^{t_0}\}$
4: Fit $\mathcal{GP}(\mathcal{D}_0)$, $\mathcal{GP}(\mathcal{D}^s)$
5: **for** $t = 1, 2, \ldots, T$ **do**
6: $\quad \mathbf{x}_t = \arg\max_{\mathbf{x} \in \mathcal{X}} \alpha_t(\mathbf{x} | \mathcal{GP}(\mathcal{D}_{t-1}))$
7: $\quad y_t =$ Evaluate $f(.)$ at $\mathbf{x}_t$
8: $\quad \mathcal{D}_t := \mathcal{D}_{t-1} \cup \{\mathbf{x}_t, y_t\}$
9: $\quad$ Compute $\hat{y}_t^s$ at $\mathbf{x}_t$ ▷ Using $\mathcal{GP}(\mathcal{D}^s)$
10: $\quad$ Update σ_s^2 ▷ Using Eq. (24)
11: $\quad$ Update $\mathbf{K}$ ▷ Using Eq. (17)
12: $\quad$ Update GP model: $\mathcal{GP}(\mathcal{D}_t)$
13: **end for**
14: Output: $\mathbf{x}^* = \arg\max_{\mathbf{x}_i \in \mathcal{D}_{i=1:T}} f(\mathbf{x}_i)$

3.1. Theoretical analysis of the proposed method

We analyse the convergence of our proposed transfer learning method. We develop our theoretical results from the works of Srinivas et al. (2010) and Wang and de Freitas (2014) where authors have established convergence guarantees for both GP-UCB and EI in a no-transfer setting. We now recall the convergence analysis of GP-UCB and EI from Section 2.3. Both our Theorems 1 and 2 follow the general intuition that additional source points reduce the variance in the system.

We start with our analysis on GP-UCB and use all the assumptions made by Srinivas et al. (2010). Here objective function f is continuous in $\mathcal{X}$ and $k(\mathbf{x}, \mathbf{x}) = 1$ for $\forall \mathbf{x} \in \mathcal{X}$. We recall Lemma 1 from the previous section which establishes the instantaneous regret for a sample $\mathbf{x}$ using GP-UCB acquisition function. We now list our specific Lemmas which establish that predictive variance for the proposed transfer learning will be less than a no-transfer method.

Lemma 5. *For a compact and convex set $\mathcal{X}$ and when number of source observations $\mathcal{X}$, $\max_{x \in \mathcal{X}} \sigma'^2_{t-1}(\mathbf{x}_t) < \max_{x \in \mathcal{X}} \sigma^2_{t-1}(\mathbf{x}_t)$, where $t \geq 1$, $\sigma'^2_{t-1}(\mathbf{x}_t)$ denotes variance in the proposed transfer learning method and $\sigma^2_{t-1}(\mathbf{x}_t)$ denotes the variance for a generic Bayesian optimization approach.*

Proof. This lemma can be proved using the intuition that the source observations act as additional points that contribute to the reduction in the variance of the system. A detailed proof is given in the Appendix. □

We now use Lemma 5 to relate the significance of reduced variances in the system and maximum information gain.

Lemma 6. *At any iteration $T \geq 1$, maximum information gain for the proposed transfer learning method and a generic Bayesian optimization method is given as:*

$$\gamma_T' \leq \gamma_T$$

where γ_T' denotes the maximum information gain for proposed transfer learning method, γ_T for a no-transfer Bayesian optimization method.

Proof. The proof follows from Lemmas 1 and 2 using some algebraic manipulations. The proof is detailed in the Appendix. □

We now recall Lemma 3 and combine the results from previous lemmas.

Theorem 1. *For any compact and convex $\mathcal{X}$, where $\mathcal{X} \subset \mathbb{R}^d$, the cumulative regret for the proposed transfer learning method has a*

tighter upper bound than the generic Bayesian optimizations method in the context of GP-UCB acquisition function with a probability of $\geq 1-\delta$, *given as:*

$$\sqrt{C_1 T\beta_T \gamma_T^{'} + \frac{\pi}{\sqrt{6}}} \leq \sqrt{C_1 T\beta_T \gamma_T + \frac{\pi}{\sqrt{6}}} \tag{25}$$

where $\delta \in (0,\ 1)$, $\beta_t = 2\log(t^2 2\pi^2/(3\delta)) + 2d\log(t^2 dbr\sqrt{log(4da/\delta)})$, $C_1 = 8/\log(1+\sigma^{-2})$

Proof. This can be proved using Lemmas 1, 2, 3, 6, 7 and the results from Theorem 2 of (Srinivas et al., 2010). Further proof is given in the Appendix. □

Remark. Our theorem shows that we achieve a tighter upper bound on the cumulative regret than a generic Bayesian optimization method. The improved bound implies that the proposed transfer learning method converges faster than a generic Bayesian optimization method. With high probability, it achieves a no regret state, $\lim_{T\to\infty} R_T/T = 0$ faster. Further, it has to be noted that β_T grows with $O(\log(T))$ and γ_T follows an order of $O((\log(T))^{d+1})$ while using a squared exponential kernel. This shows the essential sublinear property of a Bayesian optimization method.

Similar convergence analysis can be performed for EI acquisition function also, which leads to our Theorem 2. We use all the assumptions made in the work of Wang and de Freitas (2014) in stating our Theorem 2. We recall the lemmas 4, 5 and Theorem 1 of Wang and de Freitas (2014) from the section. We now recall our Lemmas 6 and 7 to establish the fact that maximum information gain for the proposed transfer learning method is less than that for a generic no-transfer method.

Theorem 2. *For a compact set* $\mathcal{X}$, *the cumulative regret for the proposed transfer learning method achieves a tighter upper bound than a generic Bayesian optimization method using EI acquisition function with a probability* $\geq 1-\delta$ *as:*

$$\beta_T^{'}\sqrt{T\gamma_T^{'}} \leq \beta_T\sqrt{T\gamma_T} \tag{26}$$

where $0<\delta<1$, $T\geq 1$, $\beta_T^{'} = 2\log(\frac{T}{\sigma^2})\gamma_{T-1}^{'} + \Lambda_T + \sqrt{\gamma_{T-1}^{'}} + \| f \|_{\mathcal{H}(\mathcal{X})}$ *is for proposed transfer learning method and* $\beta_T = 2\log(\frac{T}{\sigma^2})\gamma_{T-1} + \Lambda_T + \sqrt{\gamma_{T-1}} + \| f \|_{\mathcal{H}(\mathcal{X})}$ *is for the generic approach,* $f(.) \in \mathcal{H}(\mathcal{X})$ *and* $\Lambda_T = \sqrt{8}log(\frac{T}{\sigma^2})\log^{1/2}(4T^2\pi^2/6\delta)\sqrt{C_4}\| f \|_{\mathcal{H}}$

Proof. Lemma 6 proves that $\gamma^{'} \leq \gamma$. From Lemmas 5 and 6, it is straightforward to see that $\beta_T^{'} < \beta_T$. The required inequality can be proved based on these inequalities. A detailed proof is given in Appendix. □

Remark. We achieve an improved upper bound on the cumulative regret similar to the bound in Theorem 1. Our bound is finite and it ensures that the proposed transfer learning method is a no regret method as, $\lim_{T\to\infty} R_T/T = 0$. This finite bound also has a sublinear growth in regret where β_T and γ_T grow identical to the one in Theorem 1.

We thus show that the proposed transfer learning algorithm achieves tighter upper bound than a no-transfer Bayesian optimization method using both EI and GP-UCB acquisition functions. These tighter bounds establish improved convergence characteristics of the proposed algorithm.

4. Experiments

We consider a variety of tasks to evaluate the effectiveness of our method in the context of transfer learning. In the first set of experiments, we evaluate the proposed method of maximizing a synthetic function. In this experiment, we evaluate the flexibility of the proposed method by simulating two transfer learning cases wherein the source and target functions are either similar or only mildly related. The objective is to find the optimum of the target function. We also conduct experiments on the task of tuning the hyperparameters of three machine learning algorithms using benchmark real-world classification data. We examine two transfer learning scenarios in this experiment. First, we create source task from the same data distribution and for the latter case we use an entirely different data as the source to tune the hyperparameters on a target task.

We compare the proposed method with the following baselines:

- *Collaborative-BO* (Bardenet et al., 2013)**:** Transfer learning method which uses a latent ranking function to transfer the knowledge across the tasks. A GP has been used to learn the rank the observations across the tasks.
- *Efficient-BO* (Yogatama & Mann, 2014)**:** Transfer learning method that uses a common function for source and target is learnt where the common function is represented as deviations from their respective means.
- *Generic-BO*: no-transfer generic Bayesian optimization, Algorithm 1, is used only for the target task.

We label the proposed method as Envelope-BO. We conduct experiments using both EI and GP-UCB acquisition functions. We evaluate all the baselines based on the number of iterations and time taken to achieve the best performance. Computation time is reported on an Intel Xeon machine with 3.46 GHz Dual processor, 96 GB RAM and NVIDIA Tesla M2070 GPU. An implementation of the proposed method can be found here[1].

4.1. Experiment with synthetic data

We generate two use cases of synthetic functions: one where target function that is highly similar to the source function and the other where the target function is only mildly related to the source function. We vary the target function in both use cases whilst the source function remains the same. We choose the source function as a 2-variate normal probability distribution function with mean at [0, 0] and covariance as $\begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}$. The target function is also a 2-variate normal probability distribution with same covariance and different mean. We randomly sample 25 data points for source function in the range $[-3, 3]$ in both the dimensions. The experiments are repeated across 10 different initial configurations. The percentage of the maximum value achieved against the number of iterations are plotted.

Case 1: When the tasks are closely related

In the first use case, mean of the target task is set as (0.1,0.1) which is very close to the source function. Source and target functions are plotted in Fig. 4. The results are plotted in Fig. 5.

All the transfer learning methods, Envelope-BO, Efficient-BO and Collaborative-BO, perform equally well for EI acquisition function. This outlines the significance of the transfer learning methods when the tasks are closely related. However, for GP-UCB acquisition function, Envelope-BO performs better than all the other baselines. All the other transfer learning methods perform equally well among themselves. The no-transfer generic Bayesian optimization, Generic-BO takes more iterations than the two transfer learning methods.

We also plot the source noise variance in this scenario. Fig. 6 shows how source noise varies across different iterations. In this scenario, since the tasks are related, the variance starts with a

[1] http://bit.ly/2pXR5Ui.

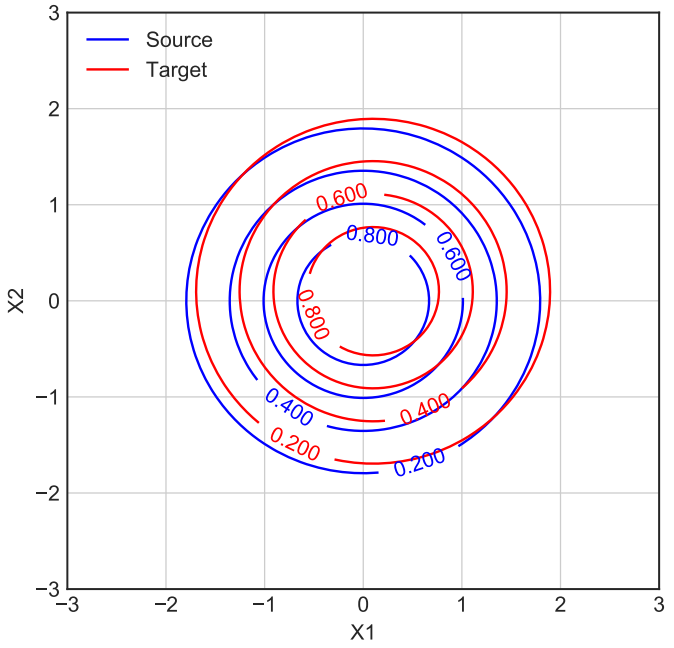

Fig. 4. Synthetic data Case 1: closely related source (blue) and target function (red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

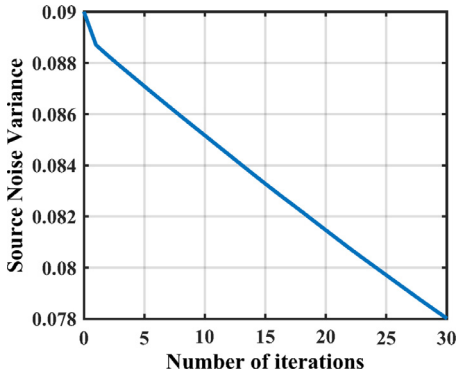

Fig. 6. Source noise variance σ_s^2 vs number of iterations when source and target are closely related.

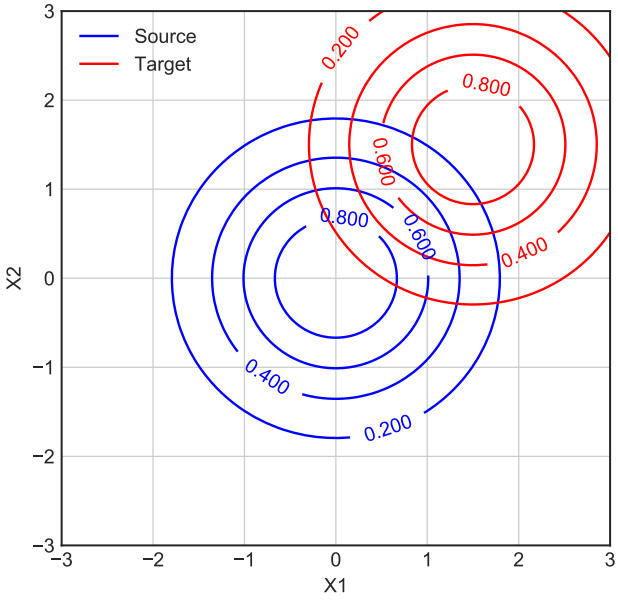

Fig. 7. Synthetic data Case 2: mildly related source (blue) and target function (red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

wide initial prior and starts reducing as the number of iterations increase. The low source noise variance indicates the closeness between the tasks.

Case 2: When the tasks are only mildly related

For this use case, the target task is another 2-variate normal probability distribution with mean at (1.5,1.5) and same covariance. Fig. 7 plots source and target functions. The results for maximizing the target function are plotted in Fig. 8.

The average of the results with standard errors across 10 different initial configurations is reported. Fig. 8a clearly shows Envelope-BO is able to achieve a significant maximum value compared to the other baselines. Generic Bayesian optimization performs better compared to Efficient-BO and Collaborative-BO. In this case, since the source and the target functions are quite different,

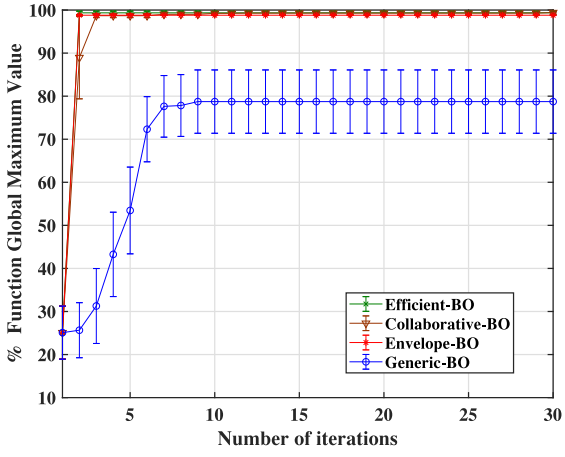

Fig. 5. Synthetic data Case 1 when tasks are closely related: percentage of the maximum value achieved as a function of number of iterations for the three different methods: using (a) EI and (b) GP-UCB. Average of the results with standard error is reported.

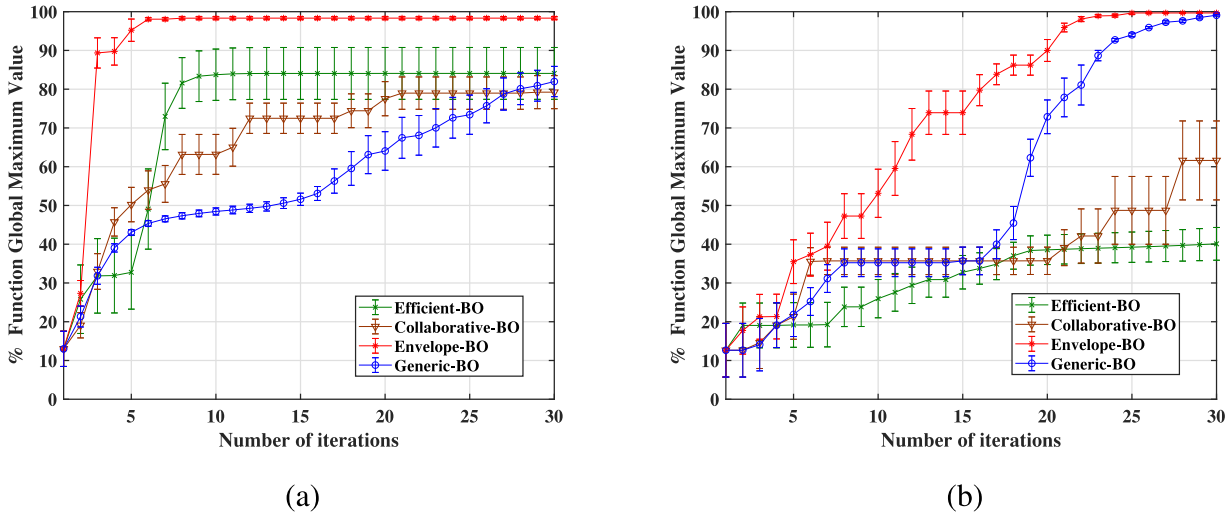

Fig. 8. Synthetic data Case 2 when tasks are only mildly related: percentage of the maximum value achieved as a function of number of iterations for the three different methods: using (a) EI and (b) GP-UCB. Average of the results with standard error is reported.

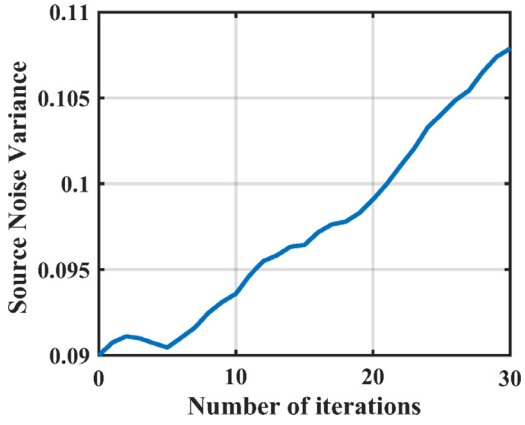

Fig. 9. Source noise variance σ_s^2 vs number of iterations when source and target are only mildly related.

Efficient-BO and Collaborative-BO fail to reach a significant maximum. In Fig. 8b, using GP-UCB, Envelope-BO converges to the maximum of the target function faster compared to the other baselines. Later the Generic-BO performs better than Efficient-BO and converges to the maximum.

The source noise variance is plotted in Fig. 9. When the tasks are different, the source noise variance also tends to increase to adjust the source-target relatedness. In Fig. 9, it clearly shows that the source noise variance increases as the number of iterations increases.

4.2. Experiment on tuning the hyperparameters of machine learning algorithms

We tune the hyperparameters of three machine learning algorithms: Support Vector Machines (SVM) with radial basis function (RBF) kernel (Schölkopf & Smola, 2002), Elastic net (Zou & Hastie, 2005) and Multi-layer perceptron (MLP) (Ruck, Rogers, Kabrisky, Oxley, & Suter, 1990) in three real-world classification problems. We give a brief description of the data in Table 2. We consider two use cases where information from the source task can be leveraged to tune the hyperparameters on the target data. In the first case, we consider a scenario where source task and target task are coming from the same data distribution but function complexities are different. In the latter case, we consider source task from a different data distribution where function complexities are also different. Each iteration of Bayesian optimization consists of learning the model on the training data using the recommended hyperparameters and evaluating the model on the test data. Different hyperparameters and their search bounds are listed in Table 3. We perform Bayesian optimization on the logarithmic of the values of the hyperparameters where the ranges are high. The experiments are repeated with 10 different samples of train and test data where each experiment is allowed 30 iterations.

Table 2
data used in the experiments.

Data	#Data points	#Features
Mushroom	8124	112
Madelon	2600	500
a8a	32,561	123
USPS	9298	256
MNIST	70,000	684

Table 3
Search bounds for different hyperparameters.

Model	Hyperparameter	Bound
SVM RBF	**Cost *C***	10^{-3}–10^{3}
	Kernel width	10^{-5} – 10^{0}
Elastic Net	**L_1 penalty**	10^{-5}–10^{-1}
	L_2 Penalty	10^{-5}–10^{-1}
MLP	**Hidden layers**	1–3
	Number of neurons	100–800
	Learning rate	10^{-3}–10^{0}
	Batch size	100–1000
	Momentum	10^{-3}–10^{0}
	Drop-out weight	0.1–0.5

4.2.1. Case 1: Source and target task from the same data distribution

In this experiment, we assume that the source and target task are coming from the same data distribution. We divide 60% of the whole data into training and use rest as test data. We then create source task from a small fraction of the whole training data and use whole training data for the target task. We use whole training data for the target task and 30% of the training data for the source task. Model is evaluated on a separate test data.

SVM with RBF kernel

For SVM with RBF kernel, the hyperparameters are the regularization parameter (C) of the SVM formulation and length-scale

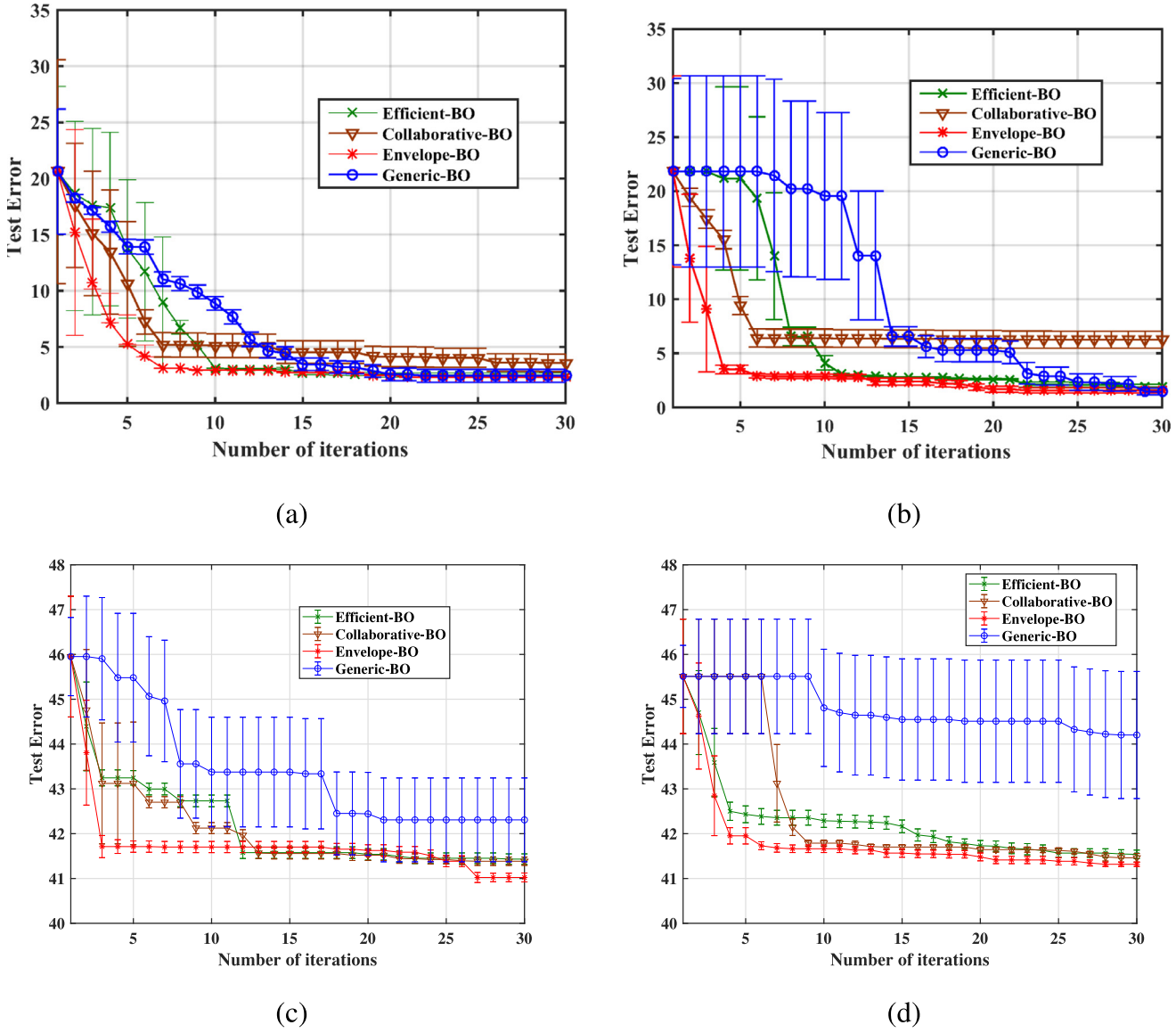

Fig. 10. Source and target task from the same data data distribution: results for tuning the hyperparameters for SVM with RBF kernel on two data, mushrooms (first row) and madelon (second row) using EI (on left) and GP-UCB (on right) acquisition functions. Average of the current best model performance with standard error is plotted.

of RBF kernel. We collect two benchmark data, mushrooms and madelon, from LIBSVM repository (Chang & Lin, 2011). We conduct experiments for SVM using the publicly available software, LIBSVM (Chang & Lin, 2011). The objective is to achieve minimum test error on the test data within the stipulated number of iterations using both EI and GP-UCB acquisition functions.

We conduct experiments on mushrooms and madelon data in this experiment. The results are presented in Fig. 10. In all the experiments, the proposed method, Envelope-BO achieves the minimum test error in the least number of evaluations compared to both the methods. The transfer learning based methods Efficient-BO and Collaborative-BO follow closely but have never been able to reach to the optimal hyperparameters faster than the proposed method. The Generic-BO without any transfer learning performs the slowest and mostly not being able to reach the best within 30 iterations.

Elastic net

Elastic net (Zou & Hastie, 2005) is a logistic regression algorithm with L_1 and L_2 regularization. We tune L_1 and L_2 penalty weights of elastic net on mushrooms and madelon data. We use AUC as the performance measure for elastic net. Here the objective is to achieve the maximum AUC within 30 iterations.

The results for the hyperparameter tuning on the different data are presented in Fig. 11. The results show that the proposed method achieves the maximum accuracy in the least number of evaluations compared to both the methods. Other two transfer learning based methods Efficient-BO and Collaborative-BO also perform well but not better than the proposed method. The Generic-BO without any transfer learning performs the slowest and mostly not being able to reach to the best within 30 iterations in all the experiments.

Multi-layer perceptron

We tune 6 parameters of a Multi-Layer Perceptron on a multi-class classification problem. We use MNIST (LeCun, Bottou, Bengio, & Haffner, 1998) data, which is the bench-marked real-world multi-class classification data of handwritten digit recognition. This data contains a training set of 60,000 and test set of 10,000 instances. We tune both the architectural parameters and the parameters of stochastic gradient descent algorithm that learns MLP. We use an implementation from a package which is publicly

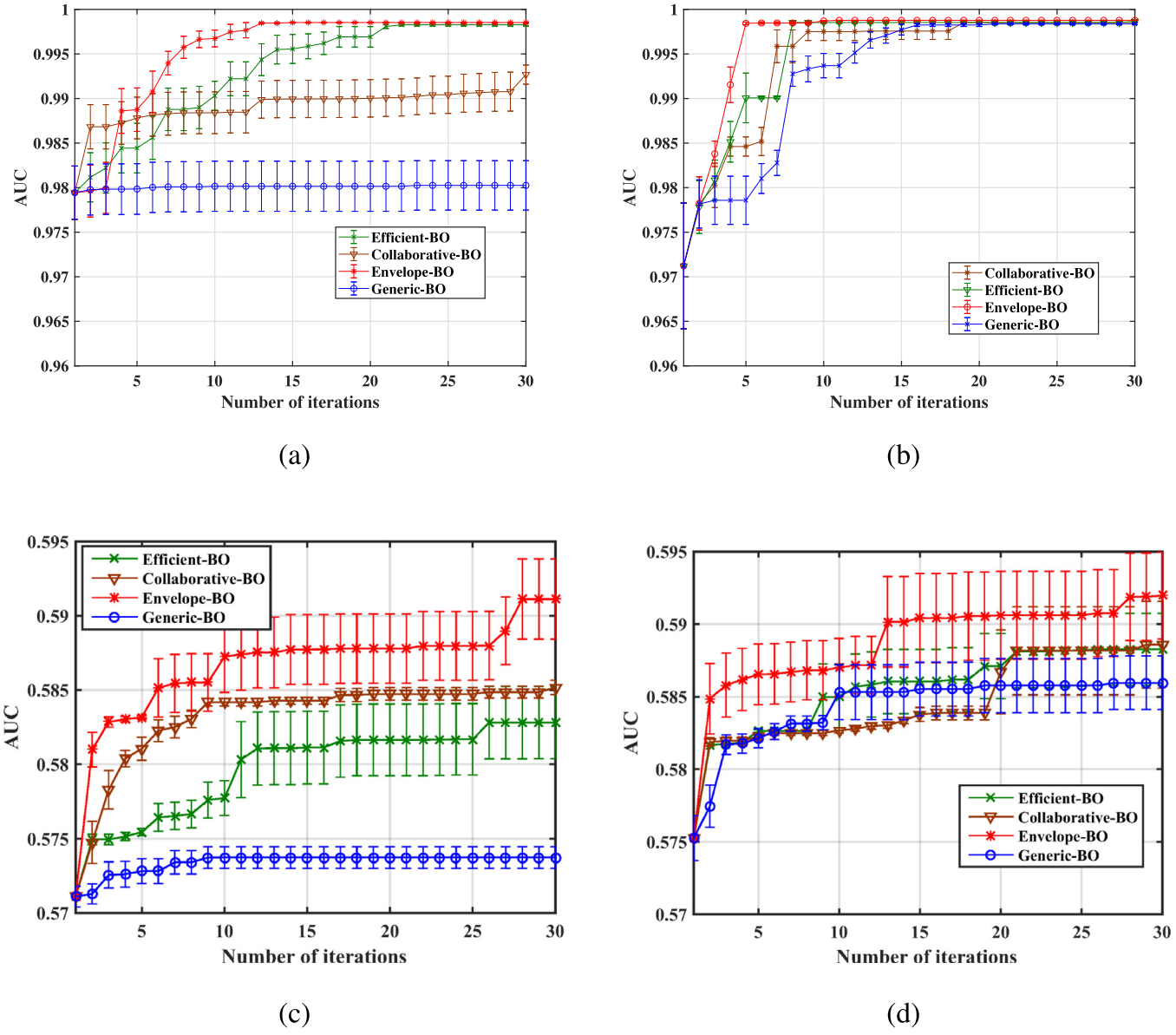

Fig. 11. Source and target task from the same data data distribution: results for tuning the hyperparameters for Elastic net on two data, mushrooms (first row) and madelon (second row) using EI (on left) and GP-UCB (on right) acquisition functions. Average of the current best model performance with standard error is plotted.

available here[2]. All the baselines are evaluated based on the test error achieved.

The results are reported in Fig. 12. Envelope-BO clearly outperforms all the other baselines and achieves a significant test error within the stipulated number of iterations. Efficient-BO and Collaborative-BO perform equally among themselves but both of these transfer learning methods are not able to reach the performance level of Envelope-BO. The no-transfer method Generic-BO exhibits cold start problem in both the cases. However, it converges later after the initial poor performance.

4.2.2. Case 2: Source and target task from different data distributions

We further evaluate our method on scenarios where both the tasks are from different data distributions. In this experiment, we select smaller data sets as source task and relatively bigger data sets as the target task. We assume that the hyperparameters have been optimized on the source data. The objective now becomes tuning the hyperparameters of different models on target data leveraging information from the source task. Models are trained on 60% of the data and evaluated on rest of the data.

SVM with RBF kernel

In this experiment, we use mushrooms data as a source and a8a data as target task. We collect a8a data from LIBSVM repository (Chang & Lin, 2011) and it is detailed in Table 2. We plot the results in Fig. 13. The results show that both EI and GP-UCB based Envelope-BO are able to achieve the best possible model performance within the stipulated number of iterations. Efficient-BO also performs reasonably well in this task.

Elastic net

Similar to SVM, we consider mushrooms data as a source for tuning the hyperparameters of elastic net on a8a. We plot the results in Fig. 14. For both EI and GP-UCB, Envelope-BO significantly outperforms all the other baselines.

Multi-layer perceptron

We consider USPS data from LIBSVM repository (Chang & Lin, 2011) as the source data. USPS data is relatively small, and

[2] https://github.com/skaae/DeepLearnToolbox.

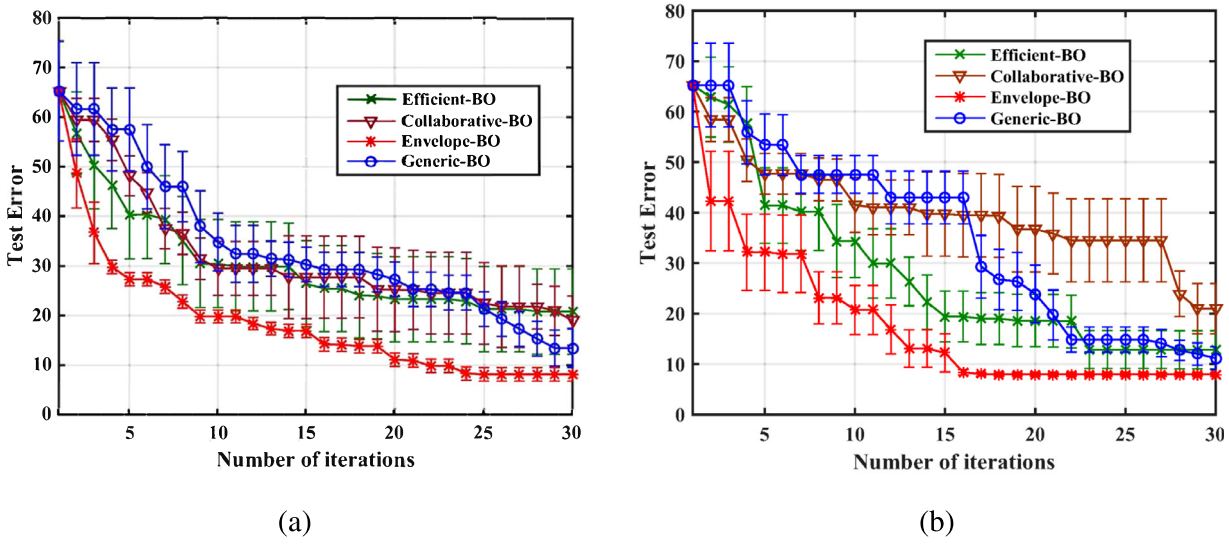

Fig. 12. Source and target from the same data data distribution: results for tuning the hyperparameters of MLP, using (a) EI and (b) GP-UCB. Average of the current best model performance with standard error is plotted.

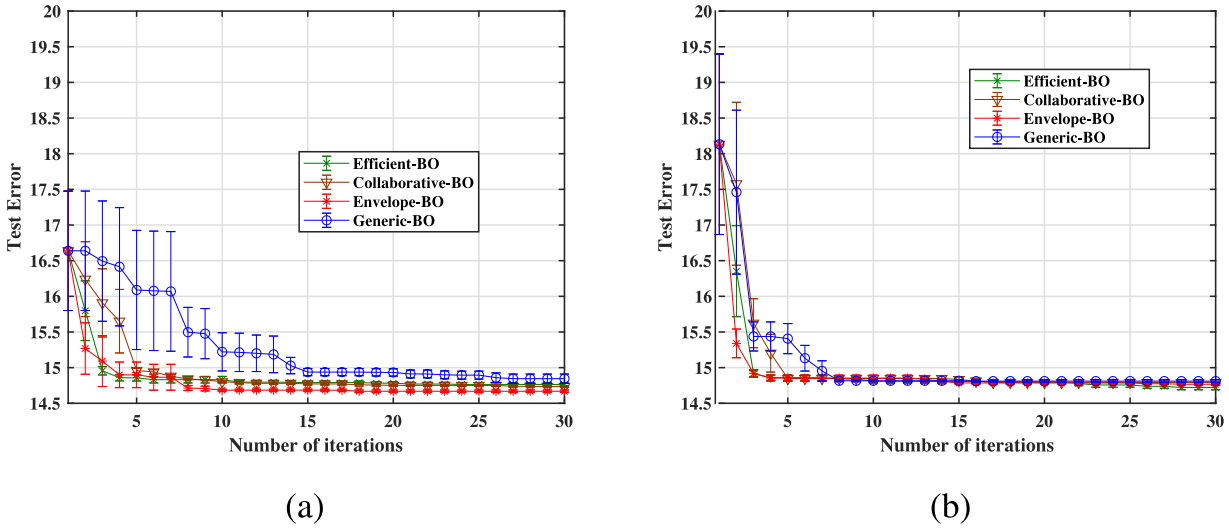

Fig. 13. Source and target from different data distributions: results for tuning the hyperparameters of SVM on a8a (target) using mushrooms data as source, using (a) EI and (b) GP-UCB. Average of the current best model performance with standard error is plotted.

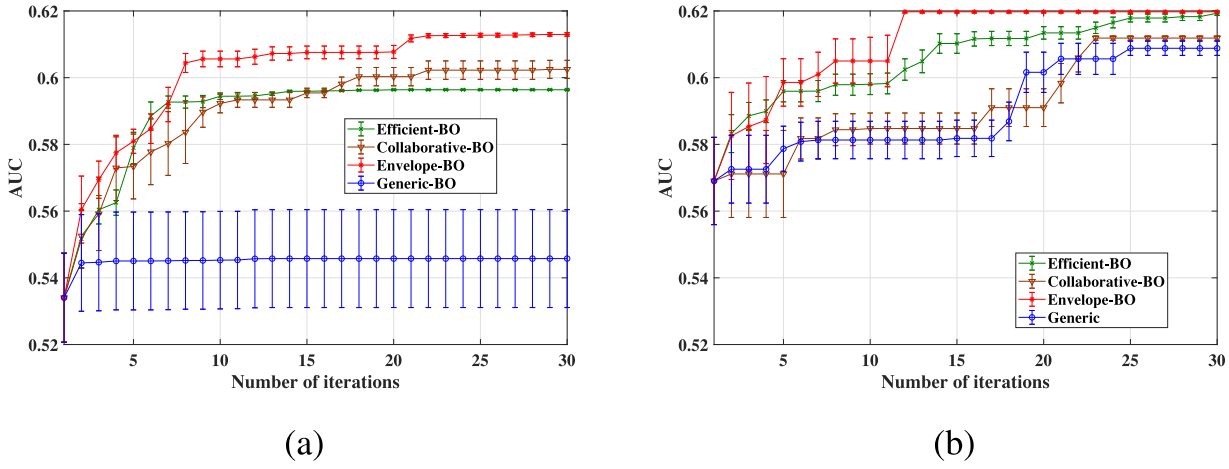

Fig. 14. Source and target from different data distributions: results for tuning the hyperparameters of Elastic net on a8a (target) using mushrooms data as the source, using (a) EI and (b) GP-UCB. Average of the current best model performance with standard error is plotted.

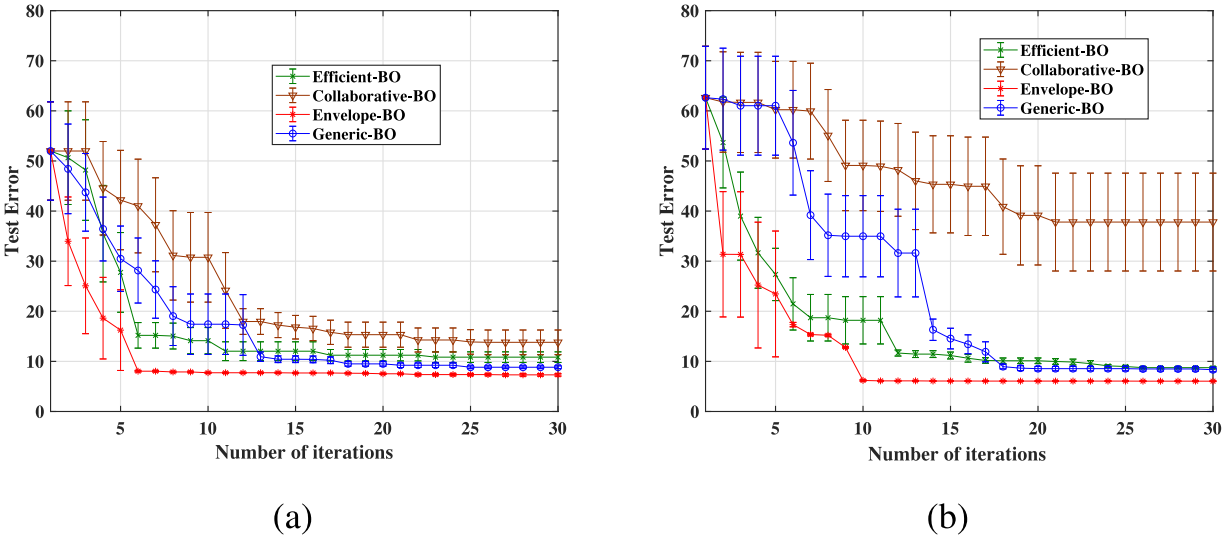

Fig. 15. Source and target from different data distributions: results for tuning the hyperparameters of MLP on MNIST (target) using USPS data as the source. Average of the current best model performance with standard error is plotted.

Table 4
Time taken in cpu seconds to achieve the best model performance.

Baselines	Elastic net			SVM			MLP
	Mushrooms	Madelon	a8a	Mushrooms	Madelon	a8a	MNIST
Envelope-BO	55.65	90.87	1132.05	43.25	51.32	1353.17	4256.43
Efficient-BO	109.25	173.28	1322.76	65.37	102.91	1429.25	10273.65
Collaborative-BO	125.83	179.71	1378.32	81.53	113.74	1631.47	13483.26
Generic-BO	131.37	193.12	1758.59	97.14	240.3	2031.81	11855.69

one can easily optimize the hyperparameters. We leverage this information to tune the hyperparameters of MLP on MNIST data. We plot the results in Fig. 15. Envelope-BO performs better than all the other baselines. Efficient-BO performs better than Collaborative-BO and Generic-BO. This further outlines the flexibility of our method to leverage information from a different data distribution.

4.3. Empirical analysis of computational complexity

In Table 4, we report the computational time taken to find the best hyperparameters in CPU seconds. Time is reported for only all the baselines using only EI acquisition function across different data and machine learning algorithms.

The proposed transfer learning method achieves the best model performance within shortest time in all the experiments. The no-transfer Generic-BO takes more time than the other two transfer learning methods. In the experiment for tuning the hyperparameters of SVM and elastic net, Envelope-BO performs almost two to three times better than Efficient-BO and Collaborative-BO and four to five times better than Generic-BO. In the same experiment, all methods take considerably long time for a8a data that is relatively bigger than other two data sets. Envelope-BO further performs well in the experiment for tuning hyperparameters of MLP achieving the best model performance significantly faster than Efficient-BO and Collaborative-BO. When the tasks are not similar, Collaborative-BO spends more time in building the ranking function to model the tasks and thus lags behind other transfer learning methods.

5. Conclusion

In this paper, we proposed a novel transfer learning framework to address cold start problem of Bayesian optimization. We modeled source task as a noisy observation of the target function and use the source observations to avoid the cold start problem for target task optimization. We estimated noise variance from observational data in a Bayesian setting. This enabled us to address the limitations of the existing methods that only work when tasks are closely related. We further analyzed the theoretical guarantees of our method using EI and GP-UCB acquisition functions. We derived tighter regret bounds for the proposed method and showed improved convergence properties than the generic Bayesian optimization. We demonstrated the performance of the proposed method in tuning the hyperparameters of three machine learning algorithms. The proposed method significantly outperforms state-of-the-art transfer learning methods and the generic no-transfer Bayesian optimization.

The proposed method considers the overall similarity between the tasks for transferring the knowledge. In Bayesian optimization, however, similarity can also be guided by the location of the peak rather than the overall function. In our future work, we aim to transfer only a portion of the optimal source observations to the target to further accelerate the target optimization task.

Acknowledgments

This research was partially funded by the Australian Government through the Australian Research Council (ARC). Prof Venkatesh is the recipient of an ARC Australian Laureate Fellowship (FL170100006).

Appendix A. Proof for Lemmas and Theorems

In this section, we detail proofs for the Lemmas and the theorems used in analysing the convergence properties of the pro-

Table A.5
Symbols.

Notation	Description
$\{x, y\}$	Observations
m	Number of source points
$\sigma'^2_{t-1}(x)$	Variance in the proposed transfer learning method
$\sigma^2_{t-1}(x)$	Variance in generic no-transfer Bayesian Optimization
σ^2	Noise variance
$I(y, \mathbf{f})$	Information Gain for no-transfer method
$I'(y, \mathbf{f})$	Information Gain for proposed method
γ'_T	Maximum information gain in proposed method after iteration T
γ_T	Maximum information gain in no-transfer method after iteration T
r_t	Instantaneous regret at any iteration t
R_T	Cumulative regret after iteration T

posed transfer learning algorithm. The symbols that are used in the proofs are given in Table A.5.

Proof for Lemma 5

Proof. This Lemma can be proved using the fact that the source observations act as additional points that contribute to the reduction in the variance of the system. This lemma establishes the fact that the uncertainty associated with the GP model in proposed transfer learning is lesser than that of a model in a generic Bayesian optimization method.

Let $\mathbf{x}_{1:t-1}$ denote the points for a generic no-transfer Bayesian optimization approach without the presence of a source task. Based on these observation the predictive variance at any point $\mathbf{x}_t$ is given as,

$$\sigma^2_{t-1}(\mathbf{x}_t) = k(\mathbf{x}_t, \mathbf{x}_t) - k(\mathbf{x}_{1:t-1}, \mathbf{x}_t)^T \mathbf{K}(\mathbf{x}_{1:t-1}, \mathbf{x}_{1:t-1})^{-1} k(\mathbf{x}_{1:t-1}, \mathbf{x}_t) \tag{A.1}$$

where $\mathbf{K}(\mathbf{x}_{1:t-1}, \mathbf{x}_{1:t-1})^{-1}$ is the covariance matrix. The proposed transfer learning method consists of a sequence of observations from the source task $\{\mathbf{x}^s_i, y^s_i\}_{i=1}^m$ and target task $\{\mathbf{x}_i, y_i\}_{i=1}^{t-1}$. Without any loss in generality, the points from the target acts as the primary sequence. The points from the source task acts as auxiliary points which reduce the variance in the system and thus boost the search for optimum of the target function.

Segmenting the observations into target points $\mathbf{x}_{1:t-1}$ and source points $\mathbf{x}^s_{1:m}$, variance in target task for a new observation $\mathbf{x}_t$ at any iteration t can be expressed as,

$$\begin{aligned}\sigma^{2'}_{t-1}(\mathbf{x}_t) = k(\mathbf{x}_t, \mathbf{x}_t) - \begin{bmatrix} k(\mathbf{x}^s_{1:m}, \mathbf{x}_t) & k(\mathbf{x}_{1:t-1}, \mathbf{x}_t)\end{bmatrix}^T \\ \mathbf{K}_*^{-1}\begin{bmatrix} k(\mathbf{x}^s_{1:m}, \mathbf{x}_t) & k(\mathbf{x}_{1:t-1}, \mathbf{x}_t)\end{bmatrix}\end{aligned} \tag{A.2}$$

where $\mathbf{K}_*$ is a block matrix which is given as,

$$\mathbf{K}_* = \begin{bmatrix} \mathbf{K}(\mathbf{x}^s_{1:m}, \mathbf{x}^s_{1:m}) & \mathbf{K}(\mathbf{x}^s_{1:m}, \mathbf{x}_{1:t-1})^T \\ \mathbf{K}(\mathbf{x}^s_{1:m}, \mathbf{x}_{1:t-1}) & \mathbf{K}(\mathbf{x}_{1:t-1}, \mathbf{x}_{1:t-1})\end{bmatrix}$$

where $\mathbf{K}_*$ is a block matrix with $\mathbf{K}(\mathbf{x}^s_{1:m}, \mathbf{x}^s_{1:m})$ is covariance matrix for source points and $\mathbf{K}(\mathbf{x}_{1:t-1}, \mathbf{x}_{1:t-1})$ is the covariance matrix for target points. We start our derivation with an introduction to the properties of positive definite matrices.We list them below.

(A): Let us consider a positive definite matrix $\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22}\end{bmatrix}$ where its Schur complement is given as, $\mathbf{M}_1 = \mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21}$. If $\mathbf{A}$ is positive definite ($\mathbf{A} \succeq 0$), then its Schur complement ($\mathbf{M}_1$) is also positive definite.

(B): If $\mathbf{A}$ is positive definite, then $\mathbf{A}^{-1}$ is also positive definite.

(C): If $\mathbf{A} \in \mathbb{R}^{n\times n}$ is positive definite and $\mathbf{B} \in \mathbb{R}^{r\times n}$ and of rank r, then $\mathbf{BAB}^T$ is also positive definite.

For a better readability, we now assign $\mathbf{A} = \mathbf{K}_*$,

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22}\end{bmatrix} = \begin{bmatrix} \mathbf{K}(\mathbf{x}^s_{1:m}, \mathbf{x}^s_{1:m}) & \mathbf{K}(\mathbf{x}_{1:t-1}, \mathbf{x}^s_{1:m}) \\ \mathbf{K}(\mathbf{x}^s_{1:m}, \mathbf{x}_{1:t-1}) & \mathbf{K}(\mathbf{x}_{1:t-1}, \mathbf{x}_{1:t-1})\end{bmatrix} \tag{A.3}$$

Now, $\mathbf{K}_*^{-1}$is given as,

$$\begin{aligned}\mathbf{K}_*^{-1} &= \mathbf{A}^{-1} \\ &= \begin{bmatrix} \mathbf{A}_{11}^{-1} + \mathbf{A}_{11}^{-1}\mathbf{A}_{12}\mathbf{M}_2^{-1}\mathbf{A}_{21}\mathbf{A}_{11}^{-1} & -\mathbf{M}_1^{-1}\mathbf{A}_{12}\mathbf{A}_{22}^{-1} \\ -\mathbf{A}_{22}^{-1}\mathbf{A}_{21}\mathbf{M}_1^{-1} & \mathbf{A}_{22}^{-1} + \mathbf{A}_{22}^{-1}\mathbf{A}_{21}\mathbf{M}_1^{-1}\mathbf{A}_{12}\mathbf{A}_{22}^{-1}\end{bmatrix}\end{aligned} \tag{A.4}$$

Let us also assign $\mathbf{B} = \mathbf{K}_*^{-1}$ as,

$$\begin{aligned}\mathbf{B} &= \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22}\end{bmatrix} \\ &= \begin{bmatrix} \mathbf{A}_{11}^{-1} + \mathbf{A}_{11}^{-1}\mathbf{A}_{12}\mathbf{M}_2^{-1}\mathbf{A}_{21}\mathbf{A}_{11}^{-1} & -\mathbf{M}_1^{-1}\mathbf{A}_{12}\mathbf{A}_{22}^{-1} \\ -\mathbf{A}_{22}^{-1}\mathbf{A}_{21}\mathbf{M}_1^{-1} & \mathbf{A}_{22}^{-1} + \mathbf{A}_{22}^{-1}\mathbf{A}_{21}\mathbf{M}_1^{-1}\mathbf{A}_{12}\mathbf{A}_{22}^{-1}\end{bmatrix}\end{aligned}$$

where the terms can be expanded in terms of source and target observations as,

$$\begin{aligned}\mathbf{B}_{11} &= \mathbf{K}(\mathbf{x}^s_{1:m}, \mathbf{x}^s_{1:m})^{-1} \\ &\quad + \mathbf{K}(\mathbf{x}^s_{1:m}, \mathbf{x}^s_{1:m})^{-1}\mathbf{K}(\mathbf{x}^s_{1:m}, \mathbf{x}_{1:t-1})\mathbf{M}_2^{-1} \\ &\quad \mathbf{K}(\mathbf{x}_{1:t-1}, \mathbf{x}^s_{1:m})\mathbf{K}(\mathbf{x}^s_{1:m}, \mathbf{x}^s_{1:m})^{-1}\end{aligned} \tag{A.5}$$

$$\mathbf{B}_{12} = -\mathbf{M}_1^{-1}\mathbf{K}(\mathbf{x}^s_{1:m}, \mathbf{x}_{1:t-1})\mathbf{K}(\mathbf{x}^s_{1:m}, \mathbf{x}^s_{1:m})^{-1} \tag{A.6}$$

$$\mathbf{B}_{21} = -\mathbf{K}(\mathbf{x}^s_{1:m}, \mathbf{x}^s_{1:m})^{-1}\mathbf{K}(\mathbf{x}_{1:t-1}, \mathbf{x}^s_{1:m})\mathbf{M}_1^{-1} \tag{A.7}$$

$$\begin{aligned}\mathbf{B}_{22} &= \mathbf{K}(\mathbf{x}_{1:t-1}, \mathbf{x}_{1:t-1})^{-1} \\ &\quad + \mathbf{K}(\mathbf{x}_{1:t-1}, \mathbf{x}_{1:t-1})^{-1}\mathbf{K}(\mathbf{x}_{1:t-1}, \mathbf{x}^s_{1:m})\mathbf{M}_1^{-1}\mathbf{K}(\mathbf{x}^s_{1:m}, \mathbf{x}_{1:t-1}) \\ &\quad \mathbf{K}(\mathbf{x}_{1:t-1}, \mathbf{x}_{1:t-1})^{-1}\end{aligned} \tag{A.8}$$

where $\mathbf{M}_1 = \mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21}$ and $\mathbf{M}_2 = \mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12}$

Here, the terms in $\mathbf{B}_{11}$, $\mathbf{B}_{12}$, and $\mathbf{B}_{21}$ already bring reduction to the system where there is a direct influence of observations from source task. This can be reasoned using the principle, "information never hurts" (Krause & Guestrin, 2005). However, the term $\mathbf{B}_{22}$ has an indirect influence from source observations. Now, the variance at $\mathbf{x}_t$ is given as,

$$\begin{aligned}\sigma^{2'}_{t-1}(\mathbf{x}_t) &= k(\mathbf{x}_t, \mathbf{x}_t) - \mathbf{k}^T(\mathbf{A}_{22}^{-1} + \mathbf{A}_{22}^{-1}\mathbf{A}_{21}\mathbf{M}_1^{-1}\mathbf{A}_{12}\mathbf{A}_{22}^{-1})\mathbf{k} - \mathbf{S} \\ &= k(\mathbf{x}_t, \mathbf{x}_t) - \mathbf{k}^T\mathbf{A}_{22}^{-1}k(\mathbf{x}_{1:t-1}, \mathbf{x}_t)\mathbf{k} - \mathbf{k}^T\mathbf{A}_{22}^{-1}\mathbf{A}_{21}\mathbf{M}_1^{-1}\mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{k} - \mathbf{S} \\ &= \sigma^2_{t-1}(\mathbf{x}_t) - \mathbf{k}^T\mathbf{A}_{22}^{-1}\mathbf{A}_{21}\mathbf{M}_1^{-1}\mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{k} - \mathbf{S}\end{aligned}$$

where $\mathbf{k} = k(\mathbf{x}_{1:t-1}, \mathbf{x}_t)$ and $\mathbf{S}$ denotes the terms that have direct influence from source observations. Here $\sigma^2_{t-1}(\mathbf{x}_t)$ denotes the variance in a generic Bayesian optimization that has no source observations (using Eqs. (A.1) and (A.3)). Now, we have to show that the following term is non-negative:

$$\mathbf{k}^T\mathbf{A}_{22}^{-1}\mathbf{A}_{21}\mathbf{M}_1^{-1}\mathbf{A}_{12}\mathbf{A}_{22}^{-1})\mathbf{k} \succeq 0 \tag{A.9}$$

This can be further reduced to proving:

$$\mathbf{A}_{22}^{-1}\mathbf{A}_{21}\mathbf{M}_1^{-1}\mathbf{A}_{12}\mathbf{A}_{22}^{-1} \succeq 0 \tag{A.10}$$

We now know that $\mathbf{A}_{21} = \mathbf{A}_{12}^T$ and $\mathbf{A}_{12} \in \mathbb{R}^{m\times t-1}$. If source points and target points are distinct, then rank $(\mathbf{A}_{12}) = \min(m, t-1)$. Using property (C), we have,

$\mathbf{A}_{21}\mathbf{M}_1^{-1}\mathbf{A}_{12} \succ 0$ ($\because \mathbf{M}_1 \succ 0$ using property (A))

$\mathbf{A}_{22}^{-1}$ is positive definite as $\mathbf{A}_{22}$ is positive definite (using property (B)). However, $\mathbf{A}_{22}$ may have higher rank than $\mathbf{A}_{21}\mathbf{M}_1^{-1}\mathbf{A}_{12}$. Hence, we prove:

$$\mathbf{A}_{22}^{-1}\mathbf{A}_{21}\mathbf{M}_1^{-1}\mathbf{A}_{12}\mathbf{A}_{22}^{-1} \succeq 0$$

Hence, for $t \geq 1$ and $m \geq 1$ (in the presence of observations from a source task), it can be shown that,

$$\max_{\mathbf{x}\in\mathcal{X}} \sigma'^{2}_{t-1}(\mathbf{x}_t) < \max_{\mathbf{x}\in\mathcal{X}} \sigma^{2}_{t-1}(\mathbf{x}_t)$$

This concludes the proof. □

Proof for Lemma 6

Proof. From the Lemma 5.3 of Srinivas et al. (2010), we have information gain for a generic no-transfer method, given as:

$$I(\mathbf{y}_T; \mathbf{f}_T) = \frac{1}{2}\sum_{t=1}^{T} \log(1 + \sigma^{-2}\sigma^{2}_{t-1}(\mathbf{x}_t)) \tag{A.11}$$

Let us denote $\Delta_1 = \sigma^{-2}\sigma^{2}_{t-1}(\mathbf{x}_t)$. Similarly the information gain for the proposed transfer learning method can be formulated as:

$$I'(\mathbf{y}_T; \mathbf{f}_T) = \frac{1}{2}\sum_{t=1}^{T} \log(1 + \sigma^{-2}\sigma'^{2}_{t-1}(\mathbf{x}_t)) \tag{A.12}$$

Here we denote $\Delta_2 = \sigma^{-2}\sigma'^{2}_{t-1}(\mathbf{x}_t)$. From Lemma 2, we have, $\max_{\mathbf{x}\in\mathcal{X}} \sigma'^{2}_{t-1}(\mathbf{x}_t) < \max_{\mathbf{x}\in\mathcal{X}} \sigma^{2}_{t-1}(\mathbf{x}_t)$ and it is straightforward to write $\Delta_1 \geq \Delta_2$.

Thus from the above Eqs. (A.11) and (A.12), we can formulate:

$$I'(\mathbf{y}_T; \mathbf{f}_T) \leq I(\mathbf{y}_T; \mathbf{f}_T) \tag{A.13}$$

Maximum information gain for a generic no-transfer Bayesian optimization method is defined as:

$$\gamma_T = \max_{A\subset X:|A|=T} I(\mathbf{y}_T; \mathbf{f}_T(.)) \tag{A.14}$$

Now γ_T for the proposed transfer learning method is given as:

$$\gamma'_T = \max_{A\subset X:|A|=T} I'(\mathbf{y}_T; \mathbf{f}_T)) \tag{A.15}$$

From the above Eqs. (A.14) and (A.15), it can be shown that:

$$\gamma'_T \leq \gamma_T \tag{A.16}$$

This concludes the proof. □

Proof for Theorem 1

Proof. This can be proved using Lemmas 1, 2, 3, 5 and 6. Upper bound on the cumulative regret for a generic no-transfer Bayesian optimization is derived as:

$$Pr\left(R_T \leq \sqrt{C_1 T\beta_T\gamma_T + \frac{\pi}{\sqrt{6}}}\right) \geq 1-\delta$$

We also derive same expression for the cumulative regret for proposed transfer learning method as:

$$Pr\left(R_T \leq \sqrt{C_1 T\beta_T\gamma'_T + \frac{\pi}{\sqrt{6}}}\right) \geq 1-\delta$$

Lemma 6 proves that $\gamma'_T \leq \gamma_T$, where γ'_T is for proposed transfer learning method. We can now establish:

$$\sqrt{C_1 T\beta_T\gamma'_T + \frac{\pi}{\sqrt{6}}} \leq \sqrt{C_1 T\beta_T\gamma_T + \frac{\pi}{\sqrt{6}}} \tag{A.17}$$

This concludes the proof for Theorem 1. □

Proof for Theorem 2

Proof. The proof is similar to our Theorem 1. It follows from Lemmas 4, 5 and 6. An upper bound on the cumulative regret for a no-transfer generic Bayesian optimization using EI can be derived as:

$$Pr\left(\beta_T\sqrt{T\gamma_T}\right) \geq 1-\delta$$

where $0<\delta<1$, $T\geq 1$, $\beta'_T = 2\log(\frac{T}{\sigma^2})\gamma'_{T-1} + \Lambda_T + \sqrt{\gamma'_{T-1}} + \| f \|_{\mathcal{H}(\mathcal{X})}$ is for proposed transfer learning method and $\beta_T = 2\log(\frac{T}{\sigma^2})\gamma_{T-1} + \Lambda_T + \sqrt{\gamma_{T-1}} + \| f \|_{\mathcal{H}(\mathcal{X})}$ is for the generic approach, $f(.) \in \mathcal{H}(\mathcal{X})$ and $\Lambda_T = \sqrt{8}log(\frac{T}{\sigma^2})\log^{1/2}(4T^2\pi^2/6\delta)\sqrt{C_4} \| f \|_{\mathcal{H}}$. From Lemmas 5 and 6, we have $\gamma'_T \leq \gamma_T$. Further, it is straightforward to establish $\beta'_T < \beta_T$. We can now conclude Theorem 2 as:

$$\beta'_T\sqrt{T\gamma'_T} \leq \beta_T\sqrt{T\gamma_T} \tag{A.18}$$

This proves Theorem 2. □

Appendix B. Implementation details

We have used the following setting for implementing our algorithm. While using GP-UCB, we learn the parameter β_t using the standard definition. β_t at any iteration t is given as,

$$\beta_t = 2\log(t^2 2\pi^2/(3\delta)) + 2d\log\left(t^2 dbr\sqrt{log(4da/\delta)}\right)$$

where we use $a=1$, $b=1$, and $\delta = 0.01$. Here d denotes the dimension. We leave the choice of setting the parameters τ_0 and ν_0 in Eqs. (22) and (23) to users. We have used $\tau_0 = 5$ and $\nu_0 = 3$ throughout the experiments.

References

Bardenet, R., Brendel, M., Kégl, B., & Sebag, M. (2013). Collaborative hyperparameter tuning. In *Proceedings of the 30th international conference on machine learning (ICML-13)* (pp. 199–207).

Brochu, E., Cora, V. M., & De Freitas, N. (2010). A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. arXiv preprint arXiv:1012.2599.

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST), 2*(3), 27.

Garnett, R., Osborne, M. A., & Roberts, S. J. (2010). Bayesian optimization for sensor set selection. In *Proceedings of the 9th ACM/IEEE international conference on information processing in sensor networks* (pp. 209–219). ACM.

González, J., Longworth, J., James, D. C., & Lawrence, N. D. (2015). Bayesian optimization for synthetic gene design. arXiv preprint arXiv:1505.01627,.

Jones, D. R., Schonlau, M., & 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., & Venkatesh, S. (2016). Flexible transfer learning framework for bayesian optimisation. In *Advances in knowledge discovery and data mining* (pp. 102–114). Springer.

Krause, A., & Guestrin, C. (2005). Near-optimal Nonmyopic Value of Information in Graphical Models. *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence, Edinburgh, Scotland* (pp. 324–331). Arlington, Virginia, United States: AUAI Press.

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

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278–2324.

Li, C., de Celis Leal, D. R., Rana, S., Gupta, S., Sutti, A., Greenhill, S., et al. (2017). Rapid bayesian optimisation for synthesis of short polymer fiber materials. *Scientific Reports, 7*(1), 5683.

Lindley, D. V. (1956). On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics, 27*(4), 986–1005. https://projecteuclid.org/euclid.aoms/1177728069

Lizotte, D. J., Wang, T., Bowling, M. H., & Schuurmans, D. (2007). Automatic Gait Optimization with Gaussian Process Regression.. In *Proceedings of the IJCAI: 7* (pp. 944–949).

Marchant, R., & Ramos, F. (2012). Bayesian optimisation for intelligent environmental monitoring. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 2242–2249). IEEE.

Mockus, J. (1994). Application of Bayesian approach to numerical methods of global and stochastic optimization. *Journal of Global Optimization, 4*(4), 347–365.
Mockus, J., Tiesis, V., & Zilinskas, A. (1978). The application of Bayesian methods for seeking the extremum. *Towards Global Optimization, 2*(2), 117–129.
Ohno, H. (2018). Empirical studies of gaussian process based Bayesian optimization using evolutionary computation for materials informatics. *Expert Systems with Applications, 96*, 25–48.
Ruck, D. W., Rogers, S. K., Kabrisky, M., Oxley, M. E., & Suter, B. W. (1990). The multilayer perceptron as an approximation to a bayes optimal discriminant function. *IEEE Transactions on Neural Networks, 1*(4), 296–298.
Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT press.
Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., & de Freitas, N. (2016). Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE, 104*(1), 148–175.
Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems* (pp. 2951–2959).
Srinivas, N., Krause, A., Kakade, S., & Seeger, M. (2010). Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the international conference on machine learning (ICML)*.
Wang, Z., & de Freitas, N. (2014). Theoretical analysis of Bayesian optimisation with unknown gaussian process hyper-parameters. arXiv preprint arXiv:1406.7758.
Williams, C. K., & Rasmussen, C. E. (2006). *Gaussian processes for machine learning* p. 4. MIT Press.
Xia, Y., Liu, C., Li, Y., & Liu, N. (2017). A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring. *Expert Systems with Applications, 78*, 225–241.
Yogatama, D., & Mann, G. (2014). Efficient transfer learning method for automatic hyperparameter tuning. In *Proceedings of the seventeenth international conference on artificial intelligence and statistics* (pp. 1077–1085).
Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67*(2), 301–320.