Information-theoretic Transfer Learning framework for Bayesian Optimisation

Anil Ramachandran, Sunil Gupta, Santu Rana, Svetha Venkatesh

Centre for Pattern Recognition and Data Analytics (PRaDA), Deakin University, Australia
{aramac, sunil.gupta, santu.rana, svetha.venkatesh}@deakin.edu.au

Abstract. Transfer learning in Bayesian optimisation is a popular way to alleviate "cold start" issue. However, most of the existing transfer learning algorithms use overall function space similarity, not a more aligned similarity measure for Bayesian optimisation based on the location of the optima. That makes these algorithms fragile to noisy perturbations, and even simple scaling of function values. In this paper, we propose a robust transfer learning based approach that transfer knowledge of the optima using a consistent probabilistic framework. From the finite samples for both source and target, a distribution on the optima is computed and then divergence between these distributions are used to compute similarities. Based on the similarities a mixture distribution is constructed, which is then used to build a new information-theoretic acquisition function in a manner similar to Predictive Entropy Search (PES). The proposed approach also offers desirable "no bias" transfer learning in the limit. Experiments on both synthetic functions and a set of hyperparameter tuning tests clearly demonstrate the effectiveness of our approach compared to the existing transfer learning methods.

1 Introduction

Experimental optimisation is a widely used technique in scientific studies and engineering design to find optimal solutions to a variety of problems via experimentation. In its most common form it is used to seek globally optimal solutions of unknown black-box functions, which are often expensive to evaluate. Bayesian optimisation offers a sample efficient solution to these kinds of problems. For example, it has been used in the scientific community for synthetic gene design [1], alloy optimisation [2] and fiber yield optimisation [3]. In machine learning it is often used to find the optimal hyperparameters for the learning algorithms and the optimisation routines [4]. Bayesian optimisation requires a probabilistic model of the function that is being optimised. Gaussian process is often a popular choice as the prior for the function model. Posterior computed based on the existing observations is then used to build a computationally cheap *acquisition function* to seek the next evaluation point. There are a variety of acquisition functions such as Probability of Improvement [5], Expected Improvement [6], GP-Upper Confidence Bound (UCB) [7], Predictive Entropy Search (PES) [8] etc. Nearly, all the acquisition functions address the trade off between sampling the regions where the posterior mean is high (*exploitation*) and sampling the regions where uncertainty is high (*exploration*) [9]. The maximiser of the acquisition function offers the best chance of being

the optima. Since the acquisition functions are computationally cheap they admit standard global optimisation routines such as DIRECT [10]. Bayesian optimisation runs in a loop with experiments sequentially being performed at the locations of the respective acquisition functions optima until an acceptable solution is found or the iteration budget is exhausted. However, the generic Bayesian optimisation approach is susceptible to the "cold start" problem where it may recommend several points with low function values before reaching a high function value region. For experiments which are highly expensive, for example, hyperparameter tuning of a large deep network on a massive training data that takes weeks to get trained on a many clusters of GPUs, the cost due to "cold start" can be quite substantial, and we like to avoid that.

A principled approach to alleviate the "cold start" issue is to utilize the knowledge acquired in any previous function (source) optimisations to achieve faster convergence in the optimisation of a new related function (target) via transfer learning. State of the art work include, [11], in which the authors assume high similarity in the rank-space of function values and develop a model that transfers function ranking between source and target. Similarly, [12] assumes high similarity between all the functions when their mean functions are subtracted out. Both these methods do not model task to task relatedness, and are hence susceptible to negative transfer in the presence of a very different source function. In [13], the authors assume that the task relationships are already known and utilize the knowledge as per source target relationship in a multi-task setting. In [14,15], the authors propose to compute overall similarity between functions based on all the observations and [16,17] uses meta-features to compute similarities between functions. Whilst the former can get tricked by scaling of the functions, the latter depends crucially on the availability of right kind of meta-features. Most of them assume function space similarity, instead of a more aligned similarity measure in terms of the location of the optima, and hence may fail to find the global optima if the functions have different scaling or relatedness between them. For this reason, a robust transfer learning framework for Bayesian optimisation which transfer knowledge based only on the proximity of the function optima is still an open problem.

In this paper, we propose a transfer learning framework for Bayesian optimisation that performs knowledge transfer based on the location of the optima in the input space and thus invariant to scaling and robust to noise in function. Since the location of global optima is not known perfectly with finite observations (be it source or target), we use probabilistic knowledge of the optima. Thompson sampling can be used to obtain the optima distribution for both sources and the target. Following that we propose a novel measure of relatedness by computing the divergence between these distributions. These distributions are then merged into a mixture distribution based on the similarities and is used to build the acquisition function. We use Predictive Entropy Search (PES) as a vehicle to build our new acquisition function. Our method offers a "no bias" algorithm, in a sense that in the limit ($T \to \infty$) the similarity to any random source tends to zero with probability 1, making Bayesian optimisation for the target free from any source induced bias in the limit. We validate our framework through application to optimisation of both synthetic functions and real world experiments. We compare our method with three well known transfer learning methods as well as with the generic Bayesian optimisation algorithm and demonstrate the superiority of our method.

2 Background

2.1 Gaussian Process

Gaussian process is an effective method for regression and classification problems and has received considerable attention in machine learning community. It serves as a probabilistic prior over smooth functions and is a generalization of infinite collection of normally distributed random variables. A Gaussian process can be completely specified by a mean function, $\mu(\mathbf{x})$ and covariance function, $k(\mathbf{x}, \mathbf{x}^{'})$. Due to these properties, we can model a smooth function as a draw from a Gaussian process. Formally,

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

where the function value at a point $\mathbf{x}$, i.e. $f(\mathbf{x})$ is a normal distribution and the relation between the function values at any two points $\mathbf{x}$ and $\mathbf{x}^{'}$ is modeled by covariance function $k(\mathbf{x}, \mathbf{x}^{'})$. Without loss in generality, the mean function can be defined to be zero thus making the Gaussian process fully specified by the covariance function [18,19]. Popular choices of covariance functions include squared exponential kernel, Matérn kernel, linear kernel, etc. We assume the function measurements are noisy, i.e. observations $\mathbf{y}_i = f(\mathbf{x}_i) + \epsilon_i$, where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ is the measurement noise. Collectively, we denote the observations as $\mathcal{D}_n = \{\mathbf{x}_{1:n}, \mathbf{y}_{1:n}\}$. The function values $[f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)]$ follow a multivariate Gaussian distribution $\mathcal{N}(0, \mathbf{K})$, where

$$\mathbf{K} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \ldots & k(\mathbf{x}_1, \mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & \ldots & k(\mathbf{x}_n, \mathbf{x}_n) \end{bmatrix} \tag{2}$$

Given a new point $\mathbf{x}$, $\mathbf{y}_{1:n}$ and $f(\mathbf{x})$ are jointly Gaussian, then by the properties of Gaussian process we can write

$$\begin{bmatrix} \mathbf{y}_{1:n} \\ f(\mathbf{x}) \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} \mathbf{K} + \sigma^2\mathbf{I} & \mathbf{k} \\ \mathbf{k}^T & k(\mathbf{x}, \mathbf{x}) \end{bmatrix}\right) \tag{3}$$

where $\mathbf{k} = [k(\mathbf{x}, \mathbf{x}_1)\, k(\mathbf{x}, \mathbf{x}_2) \ \ldots \ k(\mathbf{x}, \mathbf{x}_n))]$. Using Sherman-Morrison-Woodbury formula [18] we can write the predictive distribution at any $\mathbf{x}$ as

$$p(y \mid \mathcal{D}_n, \mathbf{x}) = \mathcal{N}(\mu_n(\mathbf{x}), \sigma_n^2(\mathbf{x})) \tag{4}$$

where the predictive mean $\mu_n(\mathbf{x})$ is given as

$$\mu_n(\mathbf{x}) = \mathbf{k}^{\mathbf{T}} \left[\mathbf{K} + \sigma^{\mathbf{2}}\mathbf{I}\right]^{-\mathbf{1}} \mathbf{y}_{1:n} \tag{5}$$

and the predictive variance $\sigma_n^2(\mathbf{x})$ is given as

$$\sigma_n^2(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - \mathbf{k}^{\mathbf{T}} \left[\mathbf{K} + \sigma^{\mathbf{2}}\mathbf{I}\right]^{-\mathbf{1}} \mathbf{k}. \tag{6}$$

2.2 Bayesian Optimisation

Bayesian optimisation is an elegant approach for the global optimisation of expensive, black-box functions. Given a small set of observations from previous function evaluations, Bayesian optimisation proceeds by building a probabilistic model of the function generally using a Gaussian process (GP). However, other methods are also used for function modeling, e.g. random forests and Bayesian neural networks [20,21]. The model of the function is then combined with the existing observations to derive a posterior distribution. This distribution is then used to construct a surrogate utility function called *acquisition function*, which is cheap to evaluate and finally optimised to recommend the next function evaluation point while keeping a trade-off between exploitation and exploration [9,22].

Several popular acquisition functions are available in literature: Probability of improvement (PI), which takes into account the improvement in the probability over the current best function value [5], Expected improvement (EI), which considers the expected improvement over the current best [6] and GP-UCB, which selects the evaluation point based on the upper confidence bound [7]. These functions are based on predictive mean and variance of the model posterior. An alternative acquisition function that maximizes the expected posterior information gain about the location of the global optima over an input space grid is proposed in [23,24]. Another information-based acquisition function called Predictive Entropy Search (PES) extended this approach to continuous search spaces [8].

2.3 Transfer Learning for Bayesian Optimisation

Transfer learning methods in Bayesian optimisation utilize ancillary information acquired from previous function (source) optimisations to achieve faster optimisation for a new related function (target). The crucial requirement is to determine the source function which is highly similar to the target. Limited work exist for transfer learning in Bayesian optimisation and most of them have made assumption regarding similarity between the source and the target. For example, Bardenet et al. [11] proposed the first work on transfer learning for Bayesian optimisation by transferring the source knowledge via a ranking function, which was assumed to be applicable for the target function as well. Another similar approach proposed in [12] assumes that the deviations of a function from its mean scaled through the standard deviation are transferable from source to target. Both these methods have strong assumption regarding source-target similarity and hence experience difficulty to find the global optima if among many sources, some have different function shapes or different optima compared to the target. To handle any potential differences between the source and the target, an alternate transfer learning framework was proposed by Joy et al. [14] modeling source data as noisy observations of the target function. The noise envelope is estimated by taking difference between any available source/target data and can be used to distinguish a related source from an unrelated one. Using the previous method as a base, Ramachandran et al. [25] proposed another transfer learning method for Bayesian optimisation that select sources which are highly related to the target. The authors use multi-armed bandit formulation for the selection of optimal sources. However, these methods would not be

able to leverage from a related source having its output scale different from that of the target even though both source and target have their optima located at the same place. Meta-learning approaches proposed in [16,17] can estimate source/target similarities, however it requires meta features for source and target functions, which may not be available in general.

3 Proposed Method

We propose a new information-theoretic transfer learning framework for Bayesian optimisation that utilizes data from source and target functions to maximize the information about the global minima of the target function. We first discuss an information-theoretic framework for Bayesian optimisation known as Predictive Entropy Search (PES) and then present our proposed transfer learning model.

3.1 Optimisation of black-box functions using Predictive Entropy Search (PES)

Let $f(\mathbf{x})$ be an expensive black-box function and we need to find its global minimizer $\mathbf{x}_* = \underset{\mathbf{x}\in\mathcal{X}}{\operatorname{argmax}} f(\mathbf{x})$ over some domain $\mathcal{X} \subset \mathbb{R}^d$. Let $\mathcal{D}_n = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ denote the noisy observations from the function $f(\mathbf{x})$ under the observation model $y_i = f(\mathbf{x}_i) + \epsilon_i$ where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ is the measurement noise. Predictive entropy search is an acquisition function for Bayesian optimisation that recommends the next function evaluation point with an aim to maximize the information about $\mathbf{x}_*$, whose posterior distribution is $p(\mathbf{x}_* \mid \mathcal{D}_n)$. The posterior distribution represents the likelihood of a location being the function global minimum after observing $\mathcal{D}_n$ observations. The information about $\mathbf{x}_*$ is measured in terms of the differential entropy between $p(\mathbf{x}_* \mid \mathcal{D}_n)$ and the expected value of $p\left(\mathbf{x}_* \mid \mathcal{D}_n \cup \{(\mathbf{x}, y)\}\right)$. Formally, the PES acquisition function selects a point $\mathbf{x}_{n+1}$ that maximizes the information about $\mathbf{x}_*$ as

$$\mathbf{x}_{n+1} = \underset{\mathbf{x}\in\mathcal{X}}{\operatorname{argmax}} \alpha_n(\mathbf{x}) = \mathbb{H}\left[p(\mathbf{x}_* \mid \mathcal{D}_n)\right] - \mathbb{E}_{p(y|\mathcal{D}_n,\mathbf{x})}\left[\mathbb{H}\left[p\left(\mathbf{x}_* \mid \mathcal{D}_n \cup \{(\mathbf{x}, y)\}\right)\right]\right] \tag{7}$$

where $\mathbb{H}\left[p(\mathbf{x})\right] = -\int p(\mathbf{x}) \log p(\mathbf{x}) d\mathbf{x}$ is the differential entropy and the expectation is with respect to the posterior predictive distribution of y given $\mathbf{x}$. Evaluation of the acquisition function in (7) is intractable and requires discretisation [24]. Noting that mutual information is a symmetric function, an easier yet equivalent formulation is as below:

$$\mathbf{x}_{n+1} = \underset{\mathbf{x}\in\mathcal{X}}{\operatorname{argmax}} \alpha_n(\mathbf{x}) = \mathbb{H}\left[p(y \mid \mathcal{D}_n, \mathbf{x})\right] - \mathbb{E}_{p(\mathbf{x}_*|\mathcal{D}_n)}\left[\mathbb{H}\left[p\left(y \mid \mathcal{D}_n, \mathbf{x}, \mathbf{x}_*\right)\right]\right] \tag{8}$$

The first term in (8) involves the predictive distribution $p(y \mid \mathcal{D}_n, \mathbf{x})$, which is Gaussian under Gaussian process modeling of $f(\mathbf{x})$. Therefore, we have

$$\mathbb{H}\left[p(y \mid \mathcal{D}_n, \mathbf{x})\right] = 0.5 \log\left[2\pi e\left(v_n\left(\mathbf{x}\right) + \sigma^2\right)\right]$$

where $v_n\left(\mathbf{x}\right)$ is the variance of $p(y \mid \mathcal{D}_n, \mathbf{x})$ and σ^2 is the variance due to measurement noise. The second term involves $p\left(y \mid \mathcal{D}_n, \mathbf{x}, \mathbf{x}_*\right)$, which is the posterior distribution for y given the observed data $\mathcal{D}_n$ and the location of the global minimizer

Algorithm 1 Bayesian optimisation using PES as acquisition function

1. **Input:** Initial observations $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n_0}$
2. **Output:** $\{\mathbf{x}_n, y_n\}_{n=1}^{T}$
3. **for** $n = n_0, \ldots, T$ **do**
   (a) Draw M samples of $\mathbf{x}_*$ from the posterior Gaussian process of the target function.
   (b) Use $\mathbf{x}_*$ samples to compute $\alpha_n(\mathbf{x})$ and maximize it as in (9) to recommend a new point $\mathbf{x}_n$.
   (c) Evaluate the target function at $\mathbf{x}_n$: $y_n = f(\mathbf{x}_n) + \epsilon_n$ where $\epsilon_n \sim \mathcal{N}\left(0, \sigma^2\right)$.
   (d) Augment $(\mathbf{x}_n, y_n)$ to the target observations and update the posterior Gaussian process.
4. **end for**

of f. An exact form for the distribution $p\left(y \mid \mathcal{D}_n, \mathbf{x}, \mathbf{x}_*\right)$ is intractable and its entropy $\mathbb{H}\left[p\left(y \mid \mathcal{D}_n, \mathbf{x}, \mathbf{x}_*\right)\right]$ for a given $\mathbf{x}_*$ sample is usually computed using expectation propagation [26]. The expected value of the entropy with respect to the distribution $p(\mathbf{x}_* \mid \mathcal{D}_n)$ is approximated by averaging the entropy for Monte Carlo samples of $\mathbf{x}_*$. A well known approach called Thompson sampling is typically used to draw the $\mathbf{x}_*$ samples [27]. Using the estimated acquisition function, PES recommends the next evaluation point by the following maximization

$$\mathbf{x}_{n+1} = \underset{\mathbf{x}\in\mathcal{X}}{\operatorname{argmax}} \alpha_n(\mathbf{x}) = 0.5 \log\left[v_n\left(\mathbf{x}\right) + \sigma^2\right] - \frac{1}{M}\sum_{i=1}^{M} 0.5 \log\left[v_n^{(i)}\left(\mathbf{x} \mid \mathbf{x}_*^{(i)}\right) + \sigma^2\right] \tag{9}$$

where M is the number of $\mathbf{x}_*$ samples drawn, $v_n\left(\mathbf{x}\right)$ and $v_n^{(i)}\left(\mathbf{x} \mid \mathbf{x}_*^{(i)}\right)$ are the predictive variances given i-th sample of $\mathbf{x}_*$. A pseudo-code for the Bayesian optimisation using PES acquisition function is presented in **Algorithm 1**. For further details on PES, we refer the reader to [8].

3.2 The Proposed Transfer Learning Method

Since the goal of the Bayesian optimisation is to find the optima (or minima) of a function, we develop a transfer learning method that directly transfers knowledge about the location of global minima from source functions to the target. Let $\left\{\{\mathbf{x}_i^s, y_i^s\}_{i=1}^{N_s}\right\}_{s=1}^{S}$ be the observations from S sources under the observation model $y_i^s = f^s(\mathbf{x}_i^s) + \epsilon_i^s$ where $\epsilon_i^s \sim \mathcal{N}(0, \sigma^2)$ and $p^s(\mathbf{x}_*)$ be the global minima distribution of the global minimizer $\mathbf{x}_*$ from each source s. Similarly, let $\{\mathbf{x}_j, y_j\}_{j=1}^{n_0}$ be the target observations under the observation model $y_j = f(\mathbf{x}_j) + \epsilon_j$ up to iteration n_0, where $\epsilon_j \sim \mathcal{N}(0, \sigma^2)$ and $p(\mathbf{x}_*)$ be the global minima distribution of its global minimizer $\mathbf{x}_*$. Our proposed transfer learning scheme intervenes into the distribution of $\mathbf{x}_*$ and modifies it to become a mixture distribution of $p(\mathbf{x}_*)$ from the target and $p^s(\mathbf{x}_*)$ from each source s. The proposed mixture distribution can be formally written as

$$p^{\text{TL}}(\mathbf{x}_*) = \pi_0 p(\mathbf{x}_*) + \pi_1 p^1(\mathbf{x}_*) + \ldots + \pi_S p^S(\mathbf{x}_*) \tag{10}$$

where $\pi_0, \pi_1, \ldots, \pi_S$ are the mixture coefficients such that $\sum_{s=0}^{S} \pi_s = 1$. Our model sets these mixture coefficients in proportion to the similarity between the target and a source. We first define a similarity measure ψ_s between the target $p(\mathbf{x}_*)$ and a source $p^s(\mathbf{x}_*)$. ψ_0 is assumed to be the similarity of $p(\mathbf{x}_*)$ with itself and is set to 1. We define $\pi_0, \pi_1, \ldots, \pi_S$ as

$$\pi_s = \frac{\psi_s}{\sum_{s=0}^{S} \psi_s}. \tag{11}$$

Given the proposed mixture distribution $p^{\text{TL}}(\mathbf{x}_*)$, our proposed information-theoretic transfer learning maximizes the following acquisition function

$$\mathbf{x}_{n+1} = \underset{\mathbf{x} \in \mathcal{X}}{\text{argmax}} \alpha_n(\mathbf{x}) = \mathbb{H}\left[p(y \mid \mathcal{D}_n, \mathbf{x})\right] - \mathbb{E}_{p^{\text{TL}}(\mathbf{x}_* | \mathcal{D}_n)}\left[\mathbb{H}\left[p\left(y \mid \mathcal{D}_n, \mathbf{x}, \mathbf{x}_*\right)\right]\right] \tag{12}$$

The entropies in the above equation are computed as in (9).

Source/Target Similarity Measure Given two probability distributions $p(\mathbf{x}_*)$ and $p^s(\mathbf{x}_*)$, we measure a divergence between p and p^s using their Kullback-Leibler (KL) divergence. KL-divergence takes non negative values and is unbounded. After measuring the divergence (denoted as $D(p^s||p)$), we map it to a similarity measure as $\psi_s = \exp(-\frac{D(p^s||p)}{\eta})$, where $\eta > 0$ is a model hyperparameter. Any other divergence measures such Hellinger distance, total variation distance or χ^2-divergence could also be used [28].

Since we have access to only samples of $\mathbf{x}_*$ and no direct access to the closed form of the probability distributions $p(\mathbf{x}_*)$ and $p^s(\mathbf{x}_*)$, we need to estimate these density functions before computing the KL-divergence. A naive way to estimate probability density is via histograms with uniform binning, however, this method quickly becomes inefficient in the number of samples. To avoid this inefficiency, we estimate the KL-divergence based on nearest neighbor (NN) distances [29,30], an approach that relies on non-parametric density estimation and then uses it to compute the KL-divergence. We refer to this estimate as NN-divergence. Let $\left\{\mathbf{x}_*^{s,1}, \ldots, \mathbf{x}_*^{s,n}\right\}$ and $\left\{\mathbf{x}_*^1, \ldots, \mathbf{x}_*^m\right\}$ be the d-dimensional samples drawn from $p^s(\mathbf{x}_*)$ and $p(\mathbf{x}_*)$ respectively. The NN divergence is estimated as

$$D_{n,m}\left(p^s(\mathbf{x}_*) \parallel p(\mathbf{x}_*)\right) = \frac{d}{n} \sum_{i=1}^{n} \log \frac{\tau_m(i)}{\rho_n(i)} + \log \frac{m}{n-1} \tag{13}$$

where

$$\tau_m(i) = \min_{j=1,\ldots,m} \parallel \mathbf{x}_*^{s,i} - \mathbf{x}_*^j \parallel$$

is the distance of $\mathbf{x}_*^{s,i}$ to its nearest neighbor in the target sample set $\left\{\mathbf{x}_*^1, \ldots, \mathbf{x}_*^m\right\}$ and

$$\rho_n(i) = \min_{j=1,\ldots,n, j \neq i} \parallel \mathbf{x}_*^{s,i} - \mathbf{x}_*^{s,j} \parallel$$

is the distance of $\mathbf{x}_*^{s,i}$ to its nearest neighbor within the source sample set $\left\{\mathbf{x}_*^{s,1}, \ldots, \mathbf{x}_*^{s,n}\right\}$.

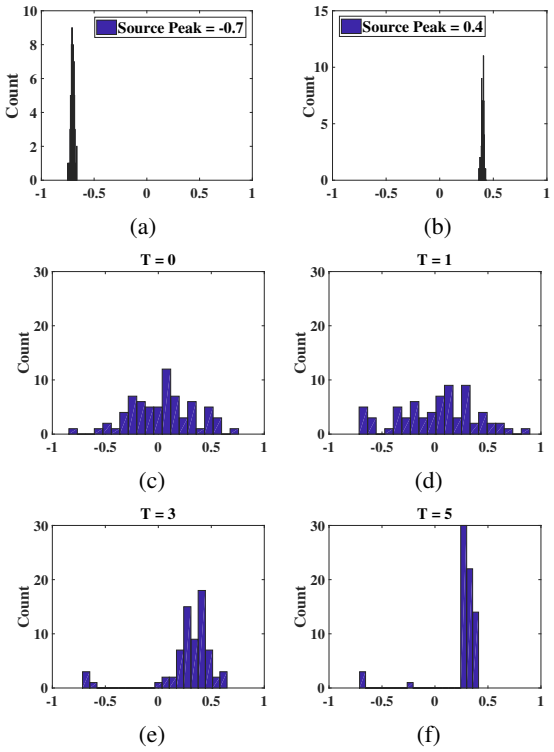


Fig. 1: Evolution of $p^{\text{TL}}(\mathbf{x}_*)$ samples with increasing iterations for a one-dimensional target function (minimum at 0.3); (a) - (b) show histogram representation of $p^s(\mathbf{x}_*)$ for two one-dimensional source functions with minima at -0.7 and 0.4 respectively; (c) shows the histogram representation of $p(\mathbf{x}_*)$ at the start of optimisation; (d) - (f) show the histogram representation of $p^{\text{TL}}(\mathbf{x}_*)$ at $T = 1, 3,$ and 5 respectively. The contribution of the source farther from the target reduces quickly while that of the related source increases.

Discussion Since the target does not have many observations available in the initial iterations, the global minima samples from target, $p(\mathbf{x}_*)$ are distributed widely over the domain $\mathcal{X}$. Therefore the NN divergence of each source with the target will have similar values and this in turn causes our transfer learning framework to choose almost equal number of samples from each source in the initial iterations. As the iterations increase, the NN divergence estimate improves as more target observations are made. This increases the contribution of related sources and decreases the contribution of unrelated sources in the mixture distribution $p^{\text{TL}}(\mathbf{x}_*)$. Asymptotically, the contribution of the $p^s(\mathbf{x}_*)$ distribution from an unrelated source becomes zero. This makes our transfer learning algorithm capable of preventing negative transfer from unrelated sources. In the limit when our algorithm has densely sampled the target function, $p(\mathbf{x}_*)$ becomes nearly an impulse function and its KL-divergence with any source becomes extremely

Algorithm 2 The Proposed Transfer Learning Algorithm

1. **Input:** Source observations: $\left\{\{\mathbf{x}_i^s, y_i^s\}_{i=1}^{N_s}\right\}_{s=1}^{S}$ under the model $y_i^s = f^s(\mathbf{x}_i^s) + \epsilon_i^s$, Initial target observations: $\{\mathbf{x}_j, y_j\}_{j=1}^{n_0}$ under the model $y_j = f(\mathbf{x}_j) + \epsilon_j$.
2. **Output:** $\{\mathbf{x}_n, y_n\}_{n=1}^{T}$.
3. Draw M samples of $\mathbf{x}_*$ from the posterior Gaussian process of each source. Denote them as $\{\mathbf{x}_*^{s,(i)}\}_{i=1}^{M}$ for s-th source.
4. **for** $n = n_0, \ldots, T$ **do**
   (a) Draw M samples of $\mathbf{x}_*$ from the posterior Gaussian process of the target function. Denote them as $\{\mathbf{x}_*^{(i)}\}_{i=1}^{M}$.
   (b) Compute the NN-divergence $D(p^s||p)$ between s-th source and the target using samples $\{\mathbf{x}_*^{s,(i)}\}_{i=1}^{M}$and $\{\mathbf{x}_*^{(i)}\}_{i=1}^{M}$ as in (13). Next compute π_s using (11).
   (c) Draw $\mathbf{x}_*$ samples from $p^{\text{TL}}(\mathbf{x}_*)$ by re-sampling $\{\mathbf{x}_*^{(i)}\}_{i=1}^{M}$ and $\{\mathbf{x}_*^{s,(i)}\}_{i=1}^{M}$ in the proportion of $\pi_0, \pi_1, \ldots, \pi_S$.
   (d) Use $\mathbf{x}_*$ samples to compute $\alpha_n(\mathbf{x})$ and maximize it as in (12) to recommend a new point $\mathbf{x}_n$.
   (e) Evaluate the target function at $\mathbf{x}_n$: $y_n = f(\mathbf{x}_n) + \epsilon_n$ where $\epsilon_n \sim \mathcal{N}\left(0, \sigma^2\right)$.
   (f) Augment $(\mathbf{x}_n, y_n)$ to the target observations and update the posterior Gaussian process.
5. **end for**

large implying that $p^{\text{TL}}(\mathbf{x}_*) \rightarrow p(\mathbf{x}_*)$ and becomes free of any bias from the sources. Figure 1 provides an illustration of the evolution of the mixture distribution $p^{\text{TL}}(\mathbf{x}_*)$. We can see that the contribution of unrelated source reduces quickly whilst that of the related source increases. We considered two source functions as:

$$f(\mathbf{x}) = 1 - a * \exp\left(-\frac{1}{2}(\mathbf{x} - \mu)(\mathbf{x} - \mu)^T\right)$$

For source 1, $\mu = -0.7$, $a = 2$ and for source 2, $\mu = 0.4$, $a = 2$. The target function has a similar form with $\mu = 0.3$, $a = 1$. For each source, 20 data points are sampled randomly from the support $[-1.5, 1.5]$ and $p^s(\mathbf{x}_*)$ samples are drawn using their posterior Gaussian process models. Figures (1a) and (1b) show the histogram counts of source $p^s(\mathbf{x}_*)$ samples. We also show the histogram count of $p(\mathbf{x}_*)$ samples drawn using the posterior Gaussian process models of initial target data (see Figure (1c)). Figures (1d) - (1f) show the evolution of $p^{\text{TL}}(\mathbf{x}_*)$ with increasing iterations. Initially $p^{\text{TL}}(\mathbf{x}_*)$ samples are widely distributed. With increasing iterations, the mass of $p^{\text{TL}}(\mathbf{x}_*)$ samples near the global minima location increases as it selects more samples from the closer source (minimum at 0.4). This example illustrates the typical behavior of our transfer learning algorithm in relying more on related sources.

Our proposed information-theoretic transfer learning algorithm for Bayesian optimisation is summarized in **Algorithm 2.**

4 Experiments

We perform experiments using both synthetic and real optimisation tasks. Through synthetic experiments, we analyze the behavior of our proposed transfer learning method

in a controlled setting. Through real data experiments, we show that our method can tune the hyperparameter for support vector machine and elastic net efficiently. For both synthetic and real experiments, we compare our proposed method with three other well known transfer learning methods and with Bayesian optimisation that does not use any knowledge from sources. The following baselines are used:

- **Env-GP:** This algorithm [14] models a source function as noisy measurements of the target function. The noise for each source is estimated separately and then the observations from each source are merged.
- **SMBO:** This algorithm [12] transfers deviation of a function from its mean scaled through the standard deviation. The observations from each source are first standardized and then data from all sources are merged.
- **SCoT:** This algorithm [11] transfers function ranking from a source to the target in latent space. The observations from each source are first adjusted and then data from all sources are merged.
- **Generic-BO (No Transfer):** This baseline is a PES based Bayesian optimisation algorithm (see Algorithm 1) and does not use any information from source functions. This is used to assess the gain in optimisation performance due to transfer learning.

4.1 Experimental setting

We use the square-exponential kernel as the covariance function in Gaussian process (GP) modeling. All GP hyperparameters are estimated using maximum a posteriori (MAP) estimation. In all our experiments, the hyperparameter η (used in similarity measure) is set to 10. All the results are averaged over 10 runs with random initializations.

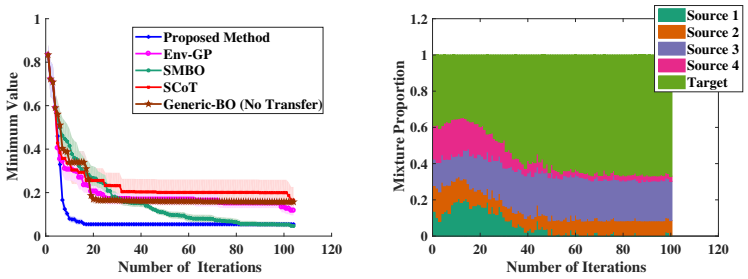
4.2 Synthetic Experiments

We consider 4 source functions as:

$$f(\mathbf{x}) = 1 - a * \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^T\right)$$

For source 1, $\boldsymbol{\mu} = [-1.8, -1.8, -1.8]$, $a = 2$. For source 2, $\boldsymbol{\mu} = [-0.7, -0.7, -0.7]$, $a = 2$. For source 3, $\boldsymbol{\mu} = [0.4, 0.4, 0.4]$, $a = 2$. For source 4, $\boldsymbol{\mu} = [1.5, 1.5, 1.5]$, $a = 2$. The target function has a similar form with $\boldsymbol{\mu} = [0.3, 0.3, 0.3]$, $a = 1$. For each source, 50 data points were sampled randomly from $[-2, 2]$ along each dimension. Our transfer learning method selects $p^{\text{TL}}(\mathbf{x}_*)$ samples from any of the four sources based on the estimated source/target similarity. Figure $2a$ shows the minimum function value obtained with respect to iterations for the proposed method and the baselines. Each method starts with the same four random observations. Our method outperforms the baselines by achieving 95% of the minimum value in the 8^{th} iteration. The performance of Env-GP method is poor because it estimates source and target similarity by measuring the difference between the source and the target functions, not their optima location. The other two transfer learning baselines also show poor performance as some of the sources

are not similar to the target. Figure2b shows the mixture proportions of sources and the target distribution with respect to increasing iterations. As seen from the Figure, the source with minima at $[\mathbf{0.4, 0.4, 0.4}]$ contributes maximally in the mixture distribution. As the iterations increase, the contribution of sources with distant minimum (from target's minimum) reduces to small values.



(a) (b)

Fig. 2: Synthetic Experiments: (a) Minimum value vs optimisation iterations (b) Proportions of various sources and the target in the mixture distribution with respect to optimisation iterations.

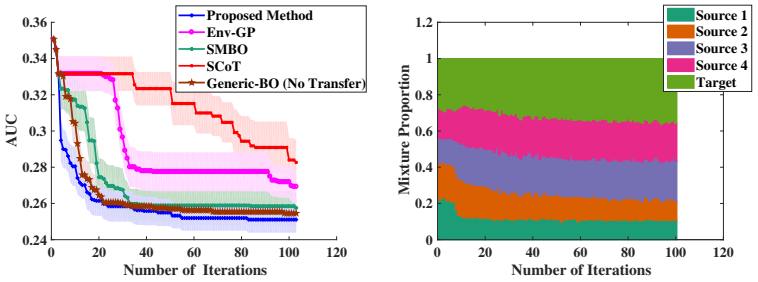
4.3 Hyperparameter Tuning

We tune hyperparameters of two classification algorithms: Support Vector Machine (SVM) and Elastic net. We consider 5 binary classification datasets - 'banana', 'breast cancer', 'colon cancer' , 'german numer' and 'diabetes' (LibSVM repository [31]). A brief summary about these datasets is provided in **Table 1**. We train a classifier for each dataset. For each dataset, 70% data is chosen randomly for training and the rest 30% used for validation. Hyperparameter tuning involves optimising validation performance (measured via AUC) as a *function* of hyperparameter values. We consider the first 4 hyperparameter tuning functions ('banana', 'breast cancer', 'colon cancer' and 'german numer') as source functions and the hyperparameter tuning function of the 'diabetes' as the target function. We assume that the several samples from the source functions are already available.

SVM with RBF kernel has two hyperparameters to tune: cost parameter (C) and kernel parameter (γ). The range for γ is set as $\left[10^{-3}, 10^{3}\right]$ and the same for C is $\left[2^{-3}, 2^{3}\right]$. We run our proposed method and other baseline methods and report the results in Figure 3. Figure $3a$ shows the AUC performance on the held-out validation set. The baseline, Generic-BO (No Transfer) shows better performance than other three transfer learning baselines. In contrast, the proposed method is able to outperform Generic-BO (No Transfer) converging faster than all the baselines. Figure $3b$ shows the proportion of contributions from different sources versus iterations.

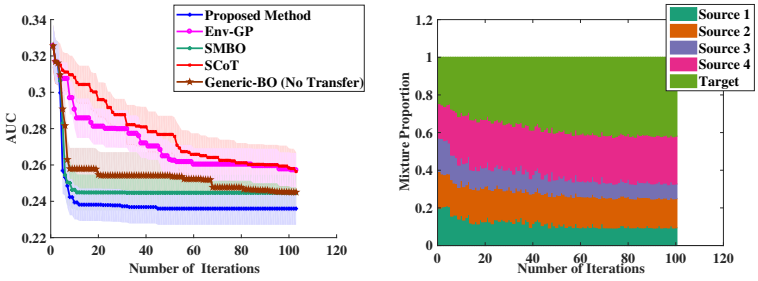
Dataset	Number of data points	Number of features
Diabetes	768	8
Banana	5300	2
Breast Cancer	683	10
Colon Cancer	62	2000
German Numer	1000	24

Table 1: Binary datasets used in our experiments.



(a) (b)

Fig. 3: Hyperparameter tuning (SVM): (a) AUC vs optimisation iterations (b) Proportions of sources and the target in the mixture distribution with respect to optimisation iterations.



(a) (b)

Fig. 4: Hyperparameter tuning (Elastic net): (a) AUC vs optimisation iterations (b) Proportions of sources and the target in the mixture distribution with respect to optimisation iterations.

Elastic net has two hyperparameters to tune: L_1 and L_2 penalty weights. The ranges for both these hyperparameters are set as $\left[10^{-2}, 10^{0}\right]$. The performance in terms of AUC on the held-out validation set is shown in Figure $4a$. Our method performs significantly better than all the baselines. A plot depicting the proportions of contributions from different sources versus iterations is shown in Figure $4b$. The source code used for all these experiments are available at https://github.com/AnilRamachandran/ITTLBO.git and the datasets are available at https://doi.org/10.7910/DVN/LRNLZV.

5 Conclusion

We propose a novel information-theoretic transfer learning algorithm for Bayesian optimisation. Our algorithm is based on constructing a mixture distribution of optima from both sources and the target combining them by the divergence between the optima distributions. This biased distributions is then used to formulate a new information theoretic acquisition function in a manner similar to the Predictive Entropy Search. In the limit $(T \rightarrow \infty)$ the optimisation becomes free from any random sources influence with probability 1. We evaluate our algorithm with diverse optimisation tasks and show that it outperforms other well known transfer learning methods.

The framework proposed in this paper is the first attempt to build a novel information-theoretic transfer learning framework for Bayesian optimisation. There are several possibilities for applying this idea in other related frameworks such as optimal sensor placements in monitoring systems [32], optimal experimental design for reservoir forecasting [33] and automatic emulator constructor for radiative transfer models (RTMs) [34].

Acknowledgment

This research was partially funded by the Australian Government through the Australian Research Council (ARC) and the Telstra-Deakin Centre of Excellence in Big Data and Machine Learning. Professor Venkatesh is the recipient of an ARC Australian Laureate Fellowship (FL170100006).

References

1. González, J., Longworth, J., James, D.C., Lawrence, N.D.: Bayesian optimization for synthetic gene design. arXiv preprint arXiv:1505.01627 (2015)
2. Li, C., Gupta, S., Rana, S., Nguyen, V., Venkatesh, S., Shilton, A.: High dimensional Bayesian optimization using dropout. arXiv preprint arXiv:1802.05400 (2018)
3. Li, C., de Celis Leal, D.R., Rana, S., Gupta, S., Sutti, A., Greenhill, S., Slezak, T., Height, M., Venkatesh, S.: Rapid Bayesian optimisation for synthesis of short polymer fiber materials. Scientific reports **7**(1) (2017) 5683
4. Snoek, J., Larochelle, H., Adams, R.P.: Practical Bayesian optimization of machine learning algorithms. In Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q., eds.: Advances in Neural Information Processing Systems 25. Curran Associates, Inc. (2012) 2951–2959
5. Kushner, H.J.: A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. Journal of Basic Engineering **86**(1) (1964) 97–106
6. Močkus, J., Tiesis, V., Žilinskas, A.: The Application of Bayesian Methods for Seeking the Extremum. In: Toward Global Optimization. Volume 2. Elsevier (1978) 117–128
7. Srinivas, N., Krause, A., Kakade, S.M., Seeger, M.W.: Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. IEEE Transactions on Information Theory **58**(5) (2012) 3250–3265
8. Hernández-Lobato, J.M., Hoffman, M.W., Ghahramani, Z.: Predictive entropy search for efficient global optimization of black-box functions. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q., eds.: Advances in Neural Information Processing Systems 27. Curran Associates, Inc. (2014) 918–926
9. Brochu, E., Cora, V.M., De Freitas, N.: A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. arXiv preprint arXiv:1012.2599 (2010)
10. Jones, D.R., Perttunen, C.D., Stuckman, B.E.: Lipschitzian optimization without the Lipschitz constant. Journal of Optimization Theory and Applications **79**(1) (1993) 157–181
11. Bardenet, R., Brendel, M., Kégl, B., Sebag, M.: Collaborative hyperparameter tuning. In: ICML (2). (2013) 199–207
12. Yogatama, D., Mann, G.: Efficient transfer learning method for automatic hyperparameter tuning. Transfer **1** (2014) 1
13. Swersky, K., Snoek, J., Adams, R.P.: Multi-task Bayesian optimization. In: Advances in neural information processing systems. (2013) 2004–2012
14. Joy, T.T., Santu, R., Gupta, S.K., Venkatesh, S.: Flexible transfer learning framework for Bayesian optimisation. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer (2016) 102–114
15. Shilton, A., Gupta, S., Rana, S., Venkatesh, S.: Regret bounds for transfer learning in Bayesian optimisation. In: Artificial Intelligence and Statistics. (2017) 307–315
16. Feurer, M., Springenberg, J.T., Hutter, F.: Initializing Bayesian hyperparameter optimization via meta-learning. In: AAAI. (2015) 1128–1135
17. Feurer, M., Letham, B., Bakshy, E.: Scalable meta-learning for Bayesian optimization. arXiv preprint arXiv:1802.02219 (2018)
18. Rasmussen, C., Williams, C.: Gaussian processes for machine learning. Gaussian Processes for Machine Learning (2006)
19. Williams, C.K.I., Barber, D.: Bayesian classification with Gaussian processes. IEEE Transactions on Pattern Analysis and Machine Intelligence **20**(12) (1998) 1342–1351
20. Hutter, F., Hoos, H.H., Leyton-Brown, K.: Sequential model-based optimization for general algorithm configuration. In: International Conference on Learning and Intelligent Optimization, Springer (2011) 507–523

21. Snoek, J., Rippel, O., Swersky, K., Kiros, R., Satish, N., Sundaram, N., Patwary, M., Prabhat, M., Adams, R.: Scalable Bayesian optimization using deep neural networks. In: International Conference on Machine Learning. (2015) 2171–2180
22. Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., de Freitas, N.: Taking the human out of the loop: A review of Bayesian optimization. Proceedings of the IEEE **104**(1) (2016) 148–175
23. Villemonteix, J., Vazquez, E., Walter, E.: An informational approach to the global optimization of expensive-to-evaluate functions. Journal of Global Optimization **44**(4) (2009) 509
24. Hennig, P., Schuler, C.J.: Entropy search for information-efficient global optimization. Journal of Machine Learning Research **13**(Jun) (2012) 1809–1837
25. Ramachandran, A., Gupta, S.K., Santu, R., Venkatesh, S.: Selecting optimal source for transfer learning in Bayesian optimisation. In: Pacific Rim International Conference on Artificial Intelligence, Springer (2018) (To Appear)
26. Minka, T.P.: A family of algorithms for approximate Bayesian inference. PhD thesis, Massachusetts Institute of Technology (2001)
27. Chapelle, O., Li, L.: An empirical evaluation of Thompson sampling. In: Advances in neural information processing systems. (2011) 2249–2257
28. Liese, F., Vajda, I.: On divergences and informations in statistics and information theory. IEEE Transactions on Information Theory **52**(10) (2006) 4394–4412
29. Wang, Q., Kulkarni, S.R., Verdú, S.: A nearest-neighbor approach to estimating divergence between continuous random vectors. In: Information Theory, 2006 IEEE International Symposium on, IEEE (2006) 242–246
30. Pérez-Cruz, F.: Kullback-Leibler divergence estimation of continuous distributions. In: Information Theory, 2008. ISIT 2008. IEEE International Symposium on, IEEE (2008) 1666–1670
31. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM transactions on intelligent systems and technology (TIST) **2**(3) (2011) 27
32. Krause, A., Singh, A., Guestrin, C.: Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. Journal of Machine Learning Research **9**(Feb) (2008) 235–284
33. Busby, D.: Hierarchical adaptive experimental design for Gaussian process emulators. Reliability Engineering & System Safety **94**(7) (2009) 1183–1193
34. Martino, L., Vicent, J., Camps-Valls, G.: Automatic emulator and optimized look-up table generation for radiative transfer models. In: Proceedings of IEEE international geoscience and remote sensing symposium (IGARSS). (2017)