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.