COMBINED TASK BAYESIAN OPTIMIZATION: A SMART SOLUTION TO SCALE-UP PROBLEM

Ryosuke Yoshizaki and Manabu Kano*

Department of Systems Science, Kyoto University, Sakyo-ku, Kyoto 606-8501, Japan

Abstract

We aim to develop a new method that can optimize operating conditions of commercial-scale equipment to achieve scale-up from pilot-scale equipment even when only a small number of experiments can be performed with commercial-scale equipment. The proposed method, combined task Bayesian optimization (CTBO), uses not only data of a target task, e.g., a commercial-scale plant, but also data of a source task, e.g., a pilot-scale plant. CTBO determines new operating conditions in the target task sequentially by BO while information of the source task is exploited by transfer learning. CTBO was compared with BO and LW-PLS + jDE (locally weighted partial least squares + self-adaptive differential evolution) through their applications to a pharmaceutical granulation process. CTBO remarkably outperformed the other methods. CTBO is expected to be useful not only for scale-up but also for technology transfer.

Keywords

Scale-up, Bayesian optimization, Gaussian process regression, Transfer learning.

Introduction

Operating condition optimization is crucial to assure product quality and reduce operation cost in any industry. In particular, one of the most challenging problems is to optimize operating conditions of commercial-scale equipment when only a small number of data are available just after scale-up from pilot-scale equipment. Such a situation is quite common because the experimental cost is significantly higher with commercial-scale equipment than pilot-scale equipment. In the present work, we aim to solve this problem and propose a new operating condition optimization method, which is referred to as combined task Bayesian optimization (CTBO).

A conventional method builds a model that relates product qualities with operating conditions, and then optimizes the operating conditions by using the model so that the target product qualities are realized and operation cost is minimized under various constraints. Unfortunately, this method does not work well when only little data is available at a commercial-scale plant, because it is difficult to build an accurate model. Such a situation always occurs just after scale-up from a pilot-scale plant. The optimization performance would be improved if data of the pilot-scale plant could be used jointly with data of the commercial-scale plant. However, combining data of different scales is not straightforward because pilot-scale and commercial-scale plants have different numbers and types of sensors, which are operated under different conditions. To resolve such difficulties, joint-Y partial least squares (JYPLS) was proposed for scale-up and product transfer (García Muñoz et al., 2005). JYPLS assumes that output variables are common in both plants and can be jointly used while input variables are different and need to be used separately. JYPLS was applied to scaling-up processes (García Muñoz et al., 2005; Liu et al., 2011). However, JYPLS does not function well when the number of data is limited.

The goal of this research is to develop a new method that can efficiently and accurately derive optimal operating conditions by using data obtained from a commercial-scale plant and a pilot-scale plant even when only little data is available at the commercial-scale plant. The proposed method uses Bayesian optimization (BO) (Brochu et al., 2009; Snoek et al., 2012) and transfer learning (Pan and Yang, 2010).

*To whom all correspondence should be addressed *manabu@human.sys.i.kyoto-u.ac.jp*

BO can systematically determine a plan for new operating conditions to be evaluated for further optimization. Thus, without experimental design, BO can find a better solution through fewer experiments than conventional methods.

Transfer learning aims to exploit knowledge from one or more source tasks and to apply the knowledge to the target task (Pan and Yang, 2010). A key idea of the present work is that transfer learning is useful for solving the scale-up problem by regarding the source task and the target task as the pilot-scale plant and the commercial-scale plant, respectively.

Conventional transfer learning algorithms assume that the number of input variables in the source task is the same as that in the target task. In practice, however, the number and types of sensors of a pilot-scale plant are different from those of a commercial-scale plant. Thus, we introduce a transformation matrix, which transforms data of the pilot-scale plant into data of the commercial-scale plant so that data from both can be used for modeling the commercial-scale plant.

In the present work, combined task Bayesian optimization (CTBO) is proposed by integrating BO, transfer learning, and the transformation matrix. To validate the effectiveness of the proposed method, CTBO was compared with BO and LW-PLS + jDE (locally weighted partial least squares + self-adaptive differential evolution) through their applications to a pharmaceutical granulation process.

Bayesian Optimization

Bayesian optimization (BO) can efficiently optimize a nonlinear objective function $f(\boldsymbol{x})$.

$$\max_{\boldsymbol{x}} \; f(\boldsymbol{x}) \tag{1}$$

BO is especially useful when it is expensive to evaluate $f(\boldsymbol{x})$, e.g., drug trials, destructive tests, or financial investment (Brochu et al., 2009). Operating condition optimization of a complex industrial process is a problem that BO is effective at solving. In practice, the objective function $f(\boldsymbol{x})$ is often corrupted by Gaussian noise

$$\varepsilon \sim \mathcal{N}(0, \sigma_{\text{noise}}^2) \; . \tag{2}$$

Hence, the objective function with noise is treated as

$$y = f(\boldsymbol{x}) + \varepsilon \; . \tag{3}$$

In BO, Gaussian process regression (GPR) and an acquisition function play important roles. GPR can grasp characteristics of the objective function from past data. The acquisition function can be evaluated more easily than the objective function and therefore is used to determine a next point, at which the acquisition function is evaluated.

Gaussian Process Regression

Gaussian process regression (GPR) is a nonlinear regression method, which maps input variables $\boldsymbol{x}$ onto a feature space with an explicit function $\phi(\cdot)$ and conducts linear regression in the feature space, i.e.,

$$y = \boldsymbol{w}^{\mathrm{T}} \phi(\boldsymbol{x}) + m \tag{4}$$

where $\boldsymbol{w}$ is a weight vector and m is a mean of the output variable y (Bishop, 2006). A distribution of y is estimated under the assumption that the weight vector follows a multivariate Gaussian distribution;

$$\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \sigma_w^2 \boldsymbol{I}) \tag{5}$$

where σ_w is a parameter and $\boldsymbol{I}$ is a unit matrix.

Given N samples, Eq. (4) is expressed as

$$\boldsymbol{y} = \boldsymbol{\Phi} \boldsymbol{w} + \boldsymbol{m} \tag{6}$$

where

$$\boldsymbol{y} = [\; y_1, \; y_2, \; \ldots, \; y_N \;]^{\mathrm{T}} \tag{7}$$

$$\boldsymbol{\Phi} = [\; \phi(\boldsymbol{x}_1), \; \phi(\boldsymbol{x}_2), \; \ldots, \; \phi(\boldsymbol{x}_N) \;]^{\mathrm{T}} \tag{8}$$

$$\boldsymbol{m} = m \boldsymbol{1}_N \tag{9}$$

and $\boldsymbol{1}_N \in \Re^N$ is a vector of ones. Under the assumption of Eq. (5), a mean vector and a covariance matrix are yielded as

$$E[\boldsymbol{y}] = \boldsymbol{m} \tag{10}$$

$$\mathrm{cov}[\boldsymbol{y}] = \sigma_w^2 \boldsymbol{\Phi} \boldsymbol{\Phi}^{\mathrm{T}} = \boldsymbol{K} \; . \tag{11}$$

The covariance matrix $\boldsymbol{K} = \{K_{ij}\}$, which is also known as a Gram matrix, consists of kernel functions $k(\cdot, \cdot)$.

$$K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sigma_w^2 \phi(\boldsymbol{x}_i)^{\mathrm{T}} \phi(\boldsymbol{x}_j) \; . \tag{12}$$

The prior of the output variable follows a multivariate Gaussian distribution

$$p(\boldsymbol{y} | \boldsymbol{X}, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{m}, \boldsymbol{K})$$

$$\boldsymbol{X} = [\; \boldsymbol{x}_1, \; \boldsymbol{x}_2, \ldots, \; \boldsymbol{x}_N \;]^{\mathrm{T}} \tag{13}$$

where $\boldsymbol{\theta}$ is a vector of GPR hyperparameters used for calculation of the covariance matrix. The mean m needs to be estimated, therefore m is added to $\boldsymbol{\theta}$.

The covariance matrix with the Gaussian noise in Eq. (2) is expressed as

$$\boldsymbol{C} = \boldsymbol{K} + \sigma_{\mathrm{noise}}^2 \boldsymbol{I} \tag{14}$$

where the parameter σ_{noise} needs to be estimated, thus it is also added to $\boldsymbol{\theta}$. The prior of $\boldsymbol{y}$ is described as

$$p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{m}, \boldsymbol{C}) \ . \tag{15}$$

The posterior of the output variable y_{new} in GPR also follows the Gaussian distribution (Rasmussen and Williams, 2006).

$$p(y_{\mathrm{new}}|\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\theta}) = \mathcal{N}(\mu(\boldsymbol{x}), \sigma^2(\boldsymbol{x})) \tag{16}$$

where

$$\mu(\boldsymbol{x}) = m + \boldsymbol{k}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{C}^{-1}(\boldsymbol{y} - \boldsymbol{m}) \tag{17}$$

$$\sigma^2(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{C}^{-1}\boldsymbol{k}(\boldsymbol{x}) \tag{18}$$

$$\boldsymbol{k}(\boldsymbol{x}) = \left[\ k(\boldsymbol{x}_1, \boldsymbol{x}),\ k(\boldsymbol{x}_2, \boldsymbol{x}),\ \ldots,\ k(\boldsymbol{x}_N, \boldsymbol{x})\ \right]^{\mathrm{T}} . \tag{19}$$

The choice of kernel function is significant for GPR to build an accurate model. In this work, the ARD Matérn 5/2 kernel with a hyperparameter θ_0 is adopted since it has great flexibility in capturing the smoothness (Brochu et al., 2009).

The posterior of the output variable in GPR described in Eq. (16) needs to be repeatedly updated for every experiment; in other words, hyperparameters of GPR are required to be tuned iteratively. The hyperparameters are determined so that their posterior is maximized. Within the Bayesian framework, the hyperparameters are automatically determined by using past data. Hence, users are freed from bothersome trial-and-error tuning. In the present work, Markov Chain Monte Carlo (MCMC) is used to determine the hyperparameters since it can determine the hyperparameters without the gradient and therefore greatly contributes toward the effectiveness of the proposed CTBO.

Acquisition Function

The acquisition function $a(\boldsymbol{x})$ is one of the most important factors to determine optimization performance of BO, and it is selected so that the cost of evaluating $a(\boldsymbol{x})$ is much lower than that of the original objective function $f(\boldsymbol{x})$. The acquisition function has two important roles: exploration and exploitation. Figure 1 describes the relationship between the objective function and the acquisition function. Exploration is to search a point at which posterior uncertainty is expected to be large, i.e., $\sigma(\boldsymbol{x})$ is emphasized. On the other hand, exploitation is to search a point at which $f(\boldsymbol{x})$ is expected to be good, i.e., $\mu(\boldsymbol{x})$ is emphasized. BO can efficiently search a good point in a wide area by using the acquisition function that combines exploration and exploitation.

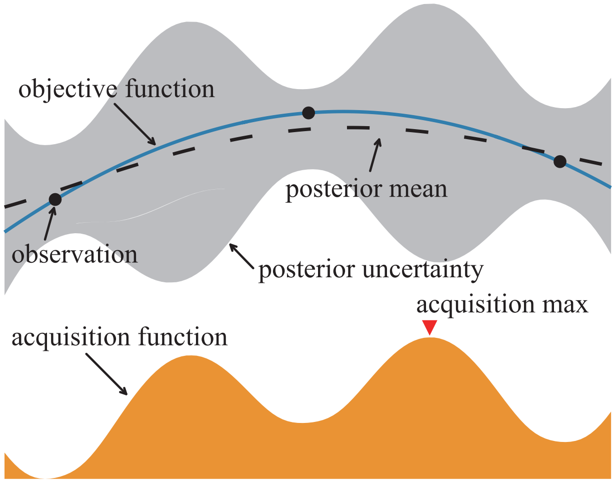

Figure 1. One-dimensional objective function, GPR model, and acquisition function in Bayesian optimization (BO).

In the present work, the mutual information (MI) algorithm is adopted since Contal et al. (2014) theoretically proved that MI improves upper bounds for cumulative regret compared with the conventional algorithms and empirically demonstrated its practicability through numerical examples.

After the choice of the acquisition function, it is required to find a solution that maximizes the acquisition function. It is hard to find a global optimal solution. Thus, local optimal solutions for different initial values are typically obtained with nonlinear programming, and the best solution among them is adopted. Latin hypercube sampling (Mckay et al., 2000) is an effective method to generate solution candidates, at which the acquisition function is calculated.

Transfer Learning

Transfer learning aims to exploit knowledge from one or more source tasks and apply the knowledge to the target task (Pan and Yang, 2010). In the proposed method, transfer learning is modified to solve the scale-up problem; a pilot-scale plant and a commercial-scale plant are regarded as the source task and the target task, respectively. Although there are various algorithms for transfer learning, adaptive transfer learning (Cao et al., 2010), which is established for GPR, is adopted in the present work.

A fundamental idea of adaptive transfer learning is to multiply a kernel function by a weight when samples come from different tasks:

$$\tilde{\boldsymbol{K}} = \{\tilde{k}(\boldsymbol{x}_i, \boldsymbol{x}_j)\} = \begin{cases} \lambda k(\boldsymbol{x}_i, \boldsymbol{x}_j), & \text{when } \zeta(\boldsymbol{x}_i, \boldsymbol{x}_j) = 1 \\ k(\boldsymbol{x}_i, \boldsymbol{x}_j), & \text{otherwise} \end{cases} \tag{20}$$

$$\lambda = 2\left(\frac{1}{1+q^2}\right)^{r^2} - 1 \tag{21}$$

where $\zeta(\boldsymbol{x}_i, \boldsymbol{x}_j) = 1$ if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ come from different tasks, and q and r are hyperparameters.

Let subscripts a and b denote the source task and the target task, respectively. Input and output variables are expressed as

$$\boldsymbol{X}_a = \left[\ \boldsymbol{x}_{a1},\ \boldsymbol{x}_{a2},\ \ldots, \boldsymbol{x}_{aN_a}\ \right]^{\mathrm{T}} \tag{22}$$

$$\boldsymbol{X}_b = \left[\ \boldsymbol{x}_{b1},\ \boldsymbol{x}_{b2},\ \ldots, \boldsymbol{x}_{bN_b}\ \right]^{\mathrm{T}} \tag{23}$$

$$\boldsymbol{y}_a = \left[\ y_{a1},\ y_{a2},\ \ldots,\ y_{aN_a}\ \right]^{\mathrm{T}} \tag{24}$$

$$\boldsymbol{y}_b = \left[\ y_{b1},\ y_{b2},\ \ldots,\ y_{bN_b}\ \right]^{\mathrm{T}}. \tag{25}$$

The covariance matrix is described as

$$\tilde{\boldsymbol{K}} = \begin{bmatrix} \tilde{\boldsymbol{K}}_{aa} & \tilde{\boldsymbol{K}}_{ab} \\ \tilde{\boldsymbol{K}}_{ba} & \tilde{\boldsymbol{K}}_{bb} \end{bmatrix} = \begin{bmatrix} \boldsymbol{K}_{aa} & \lambda\boldsymbol{K}_{ab} \\ \lambda\boldsymbol{K}_{ba} & \boldsymbol{K}_{bb} \end{bmatrix} \tag{26}$$

where $\boldsymbol{K}_{ab} = \boldsymbol{K}_{ba}^{\mathrm{T}}$.

In GPR with adaptive transfer learning, the prior of the output variable in the target task is described as

$$p(\boldsymbol{y}_b | \boldsymbol{X}_a, \boldsymbol{y}_a, \boldsymbol{X}_b, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{C}) \tag{27}$$

$$\boldsymbol{\mu} = \boldsymbol{m}_b + \tilde{\boldsymbol{K}}_{ba}(\tilde{\boldsymbol{K}}_{aa} + \sigma^2_{\text{noise},a}\boldsymbol{I})^{-1}(\boldsymbol{y}_a - \boldsymbol{m}_a) \tag{28}$$

$$\boldsymbol{C} = (\tilde{\boldsymbol{K}}_{bb} + \sigma^2_{\text{noise},b}\boldsymbol{I}) - \tilde{\boldsymbol{K}}_{ba}(\tilde{\boldsymbol{K}}_{aa} + \sigma^2_{\text{noise},a}\boldsymbol{I})^{-1}\tilde{\boldsymbol{K}}_{ab} \tag{29}$$

$$\boldsymbol{m}_a = m_a \boldsymbol{1}_{N_a} \quad \boldsymbol{m}_b = m_b \boldsymbol{1}_{N_b} \tag{30}$$

where $\boldsymbol{1}_{N_a} \in \Re^{N_a}$ and $\boldsymbol{1}_{N_b} \in \Re^{N_b}$ are the vectors of ones, and $\boldsymbol{\theta}$ is a vector of hyperparameters. The posterior of the output variable in the target task is described as

$$p(y_{b,\text{new}} | \boldsymbol{X}_a, \boldsymbol{y}_a, \boldsymbol{X}_b, \boldsymbol{y}_b, \boldsymbol{\theta}) = \mathcal{N}(\mu(\boldsymbol{x}_b), \sigma^2(\boldsymbol{x}_b)) \tag{31}$$

$$\mu(\boldsymbol{x}_b) = m_b + \tilde{\boldsymbol{k}}^{\mathrm{T}}(\boldsymbol{x}_b)\boldsymbol{C}^{-1}(\boldsymbol{y} - \boldsymbol{m}) \tag{32}$$

$$\sigma^2(\boldsymbol{x}_b) = k(\boldsymbol{x}_b, \boldsymbol{x}_b) - \tilde{\boldsymbol{k}}^{\mathrm{T}}(\boldsymbol{x}_b)\boldsymbol{C}^{-1}\tilde{\boldsymbol{k}}(\boldsymbol{x}_b) \tag{33}$$

$$\boldsymbol{y} = \left[\ \boldsymbol{y}_a^T,\ \boldsymbol{y}_b^{\mathrm{T}}\ \right]^{\mathrm{T}} \tag{34}$$

$$\boldsymbol{m} = \left[\ \boldsymbol{m}_a^T,\ \boldsymbol{m}_b^{\mathrm{T}}\ \right]^{\mathrm{T}} \tag{35}$$

$$\tilde{\boldsymbol{k}}(\boldsymbol{x}_b) = \left[\ \tilde{\boldsymbol{k}}_a^{\mathrm{T}}(\boldsymbol{x}_b),\ \tilde{\boldsymbol{k}}_b^{\mathrm{T}}(\boldsymbol{x}_b)\ \right]^{\mathrm{T}} \tag{36}$$

$$\tilde{\boldsymbol{k}}_a(\boldsymbol{x}_b) = \left[\ \tilde{k}(\boldsymbol{x}_{a1}, \boldsymbol{x}_b),\ \ldots,\ \tilde{k}(\boldsymbol{x}_{aN_a}, \boldsymbol{x}_b)\right]^{\mathrm{T}} \tag{37}$$

$$\tilde{\boldsymbol{k}}_b(\boldsymbol{x}_b) = \left[\ \tilde{k}(\boldsymbol{x}_{b1}, \boldsymbol{x}_b),\ \ldots,\ \tilde{k}(\boldsymbol{x}_{bN_b}, \boldsymbol{x}_b)\ \right]^{\mathrm{T}}. \tag{38}$$

Transformation Matrix for Scale-up

Conventional transfer learning algorithms assume that the number of input variables in the source task is the same as that in the target task. In practice, the number and types of sensors of pilot-scale equipment are different from those of commercial-scale equipment; that is, $\boldsymbol{X}_a \in \Re^{N_a \times M_a}$ and $\boldsymbol{X}_b \in \Re^{N_b \times M_b}$ are different in size along both directions.

To use the transfer learning algorithms for scale-up, a transformation matrix $\boldsymbol{W}_{tr} \in \Re^{M_a \times M_b}$ is introduced in the present work. Assuming that $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$ have a linear relationship, $\boldsymbol{X}_a$ of the pilot-scale equipment (the source task) is transformed to $\boldsymbol{X}'_a \in \Re^{N_a \times M_b}$ of the commercial-scale equipment (the target task) as follows:

$$\boldsymbol{X}'_a = \boldsymbol{X}_a \boldsymbol{W}_{tr}\ . \tag{39}$$

The most important and challenging issue is how to determine $\boldsymbol{W}_{tr}$ appropriately. MCMC tunes the transformation matrix $\boldsymbol{W}_{tr}$ together with the other GPR hyperparameters in the proposed algorithm. In other words, augmented hyperparameters are defined as

$$\widetilde{\boldsymbol{\theta}} = \left[\ \boldsymbol{\theta}^{\mathrm{T}},\ vec(\boldsymbol{W}_{tr})^{\mathrm{T}}\ \right]^{\mathrm{T}} \tag{40}$$

where $\boldsymbol{\theta}$ is the vector of GPR hyperparameters and $vec(\cdot)$ is an operator that concatenates all the columns of a matrix into a vector.

Case Study

Optimization performance of combined task Bayesian optimization (CTBO) is compared with those of BO and LW-PLS + jDE. LW-PLS + jDE is a conventional efficient method that combines locally weighted partial least squares (LW-PLS) (Kim et al., 2011; Kano and Fujiwara, 2013) and self-adaptive differential evolution (jDE) (Brest et al., 2006). It was proposed by Yoshizaki et al. (2015), and its practicability was demonstrated through its application to a pharmaceutical granulation process. In this section, these methods are applied to another pharmaceutical granulation process.

In LW-PLS + jDE, Latin hypercube sampling is used to generate samples of various operating conditions at which experiments are conducted. Then, the objective function is calculated for each set of operating conditions. The measurements of the operating conditions and the calculated values of the objective function are stored in a database; $\mathcal{D}_{lw} = \{\boldsymbol{x}_n, y_n\}_{n=1}^{N_b}$. By using the database $\mathcal{D}_{lw}$, the optimal operating conditions $\boldsymbol{x}_{\text{best}}$ is obtained with LW-PLS + jDE. Finally, the objective function y_{best} for $\boldsymbol{x}_{\text{best}}$ is calculated.

On the other hand, BO and CTBO determine new operating conditions one by one. The operating conditions and the objective function are stored in a database until the number of samples reaches N_b. Uniform random sampling chooses initial operating conditions in the target task since both BO and CTBO require at least one set of data to develop a GPR model. The optimal operating conditions and the corresponding objective function are determined as follows:

$$\boldsymbol{x}_{\text{best}} = \boldsymbol{x}_{N_{\text{best}}} \ , \ y_{\text{best}} \quad = y_{N_{\text{best}}} \ , \ N_{\text{best}} = \arg\min_{n \in [1, N_b]} y_n \ . \tag{41}$$

To verify the practicability of the proposed method, CTBO, BO, and LW-PLS + jDE were applied to a pharmaceutical granulation process. In this industrial case study, operation data were obtained from a pilot-scale plant and a commercial-scale plant of a pharmaceutical company. The pilot-scale plant and the commercial-scale plant were regarded as the source task and the target task, respectively. The same product was manufactured in both plants, but operating conditions were different since the equipment was different especially in size. To optimize operating conditions in the commercial-scale plant by conducting a small number of experiments, it is desirable to utilize operation data of the pilot-scale plant. The objective function in this study is the deviation of the product quality $\boldsymbol{y}_b$ from its target $\boldsymbol{y}_{b,\text{target}}$, i.e.,

$$\min_{\boldsymbol{x}_b} \ ||\boldsymbol{y}_b(\boldsymbol{x}_b) - \boldsymbol{y}_{b,\text{target}}|| \ . \tag{42}$$

The numbers of available samples (provided by the company) were 40 in the pilot-scale plant and 32 in the commercial-scale plant. All the past data of the pilot-scale plant were used for scale-up, i.e., $N_a = 40$. On the other hand, a neural network (NN) model of the commercial-scale plant was developed by using real operation data. The NN model was used to generate data for testing CTBO, BO, and LW-PLS + jDE, because it was impossible to conduct many experiments at the real plant. The NN model had 10 hidden nodes, four input nodes corresponding to operating conditions (input variables), and three output nodes corresponding to product quality (output variables). To set a realistic problem, the number of samples was limited: $N_b = 3$, 5, or 10 for the commercial-scale plant.

The optimization was conducted 100 times for different initial values to evaluate the influence of the initial values on the results. LW-PLS + jDE and BO used data of the commercial-scale plant only, while CTBO used data of both the pilot-scale plant and the commercial-scale plant.

Table 1. Optimization results of the commercial-scale pharmaceutical granulation equipment.

N_b	Measure	LW-PLS	BO	CTBO
3	Median	2.45	2.28	1.97
	Std Dev	1.16	0.62	0.63
5	Median	2.14	1.74	1.61
	Std Dev	1.14	0.45	0.47
10	Median	2.18	1.32	1.19
	Std Dev	1.24	0.37	0.37

The optimization results of the commercial-scale pharmaceutical granulation process are shown in Table 1. BO improved the optimization performance as the number of samples N_b increased; in fact, both the median and the standard deviation became better (smaller) as N_b became larger. On the other hand, the optimization performance of LW-PLS + jDE deteriorated in some cases even when the number of samples N_b increased. The results of LW-PLS + jDE indicated that an increase of data chosen randomly did not assure an improvement in the optimization performance. In addition, the standard deviations of LW-PLS + jDE were much larger than those of BO. The results clarified the disadvantage of LW-PLS; that is, it is difficult to construct an accurate regression model when the number of samples is limited.

The proposed method, CTBO, was superior to LW-PLS + jDE and BO. In particular, since CTBO outperformed BO, it was confirmed that the proposed method appropriately exploited the information from the pilot-scale plant and used it for optimizing the operating conditions in the commercial-scale plant. The results have demonstrated that CTBO significantly improved the optimization performance in the situation that only little data was available at the commercial-scale plant.

Conclusions

We have developed a new method that can optimize operating conditions of a commercial-scale plant with a small number of experiments. The proposed method, referred to as combined task Bayesian optimization (CTBO), integrates BO and transfer learning. BO was used to determine new operating conditions sequentially in the target task, i.e., a commercial-scale plant; transfer learning was used to combine data of both the source task and the target task, i.e., a pilot-scale plant and a commercial-scale plant. Conventional transfer learning algorithms assume that the number of input variables in the source task is the same as that in the target task. In practice, however, the number and types of sensors of pilot-scale equipment are different from those of commercial-scale equipment. Thus, transfer learning algorithms cannot be applied directly. To deal with this problem, a transformation matrix was introduced. In addition, MCMC was used to tune the transformation matrix and GPR hyperparameters simultaneously.

The proposed CTBO was verified through its application to the pharmaceutical granulation process. The results demonstrated that CTBO outperformed LW-PLS + jDE and BO and therefore CTBO used data of the pilot-scale plant effectively to optimize the operating conditions of the commercial-scale plant.

CTBO will be applicable not only to the scale-up problem but also to the problem of technology transfer from a mother plant to a copy plant; that is, when two plants are the same but operating conditions need to be optimized at the copy plant due to various uncertainties.

Acknowledgements

This work was partially supported by JSPS KAKENHI Grant Number 15K06554.

References

Bishop, C. (2006). *Pattern recognition and machine learning.* Springer.

Brest, J., Greiner, S., Bošković, B., Mernik, M., and Zumer, V. (2006). Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation*, 10(6):646–657.

Brochu, E., Cora, V., and De Freitas, N. (2009). A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *Technical Report TR-2009-023, UBC.*

Cao, B., Pan, S., Zhang, Y., Yeung, D.-Y., and Yang, Q. (2010). Adaptive transfer learning. In *Proc. of the National Conference on Artificial Intelligence*, volume 1, pages 407–412.

Contal, E., Perchet, V., and Vayatis, N. (2014). Gaussian process optimization with mutual information. In *31st International Conference on Machine Learning (ICML)*, volume 2, pages 1515–1523.

García Muñoz, S., MacGregor, J., and Kourti, T. (2005). Product transfer between sites using joint-Y PLS. *Chemometrics and Intelligent Laboratory Systems*, 79(1-2):101–114.

Kano, M. and Fujiwara, K. (2013). Virtual sensing technology in process industries: Trends and challenges revealed by recent industrial applications. *Journal of Chemical Engineering of Japan*, 46(1):1–17.

Kim, S., Kano, M., Nakagawa, H., and Hasebe, S. (2011). Estimation of active pharmaceutical ingredients content using locally weighted partial least squares and statistical wavelength selection. *International Journal of Pharmaceutics*, 421(2):269–274.

Liu, Z., Bruwer, M.-J., MacGregor, J., Rathore, S., Reed, D., and Champagne, M. (2011). Scale-up of a pharmaceutical roller compaction process using a joint-Y partial least squares model. *Industrial and Engineering Chemistry Research*, 50(18):10696–10706.

Mckay, M., Beckman, R., and Conover, W. (2000). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61.

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

Rasmussen, C. and Williams, C. (2006). Gaussian processes for machine learning. *MIT Press.*

Snoek, J., Larochelle, H., and Adams, R. (2012). Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems (NIPS)*, volume 4, pages 2951–2959.

Yoshizaki, R., Kano, M., Tanabe, S., and Miyano, T. (2015). Process parameter optimization based on lw-pls in pharmaceutical granulation process. In *IFAC Proceedings Volumes: Int'l Symp. on Advanced Control of Chemical Processes (ADCHEM)*, volume 48, pages 303–308.