Efficient Hyperparameter Optimization by Using Bayesian Optimization for Drug–Target Interaction Prediction

Tomohiro Ban[1,2], Masahito Ohue[1,3,4], Yutaka Akiyama[1,2,3,4,5,*]
[1]*School of Computing, Tokyo Institute of Technology, 2-12-1 W8-76 Ookayama, Meguro-ku, Tokyo 152-8550, Japan*
[2]*Education Academy of Computational Life Sciences, Tokyo Institute of Technology,*
2-12-1 W8-93 Ookayama, Meguro-ku, Tokyo 152-8550, Japan
[3]*Advanced Computational Drug Discovery Unit, Institute of Innovative Research, Tokyo Institute of Technology,*
4259 Nagatsutacho, Midori-ku, Yokohama, Kanagawa 226-8501, Japan
[4]*AIST-Tokyo Tech Real World Big-Data Computation Open Innovation Laboratory (RWBC-OIL), National Institute of Advanced Industrial Science and Technology, 1-1-1 Umezono, Tsukuba, Ibaraki 305-8560, Japan*
[5]*Molecular Profiling Research Center for Drug Discovery, National Institute of Advanced*
Industrial Science and Technology, 2-4-7 Aomi, Koto-ku, Tokyo 135-0064, Japan
**Email: akiyama@c.titech.ac.jp; Tel: +81-3-5734-3645; Fax: +81-3-5734-3646*

***Abstract*—A Bayesian optimization technique enables a short search time for a complex prediction model that includes many hyperparameters while maintaining the accuracy of the prediction model. Here, we apply a Bayesian optimization technique to the drug–target interaction (DTI) prediction problem as a method for computational drug discovery. We target neighborhood regularized logistic matrix factorization (NRLMF) (Liu *et al.*, 2016), which is a state-of-the-art DTI prediction method, and accelerated parameter searches with the Gaussian process mutual information (GP-MI). Experimental results with four general benchmark datasets show that our GP-MI-based method obtained an 8.94-fold decrease in the computational time on average and almost the same prediction accuracy when measured with area under the curve (AUC) for all datasets compared to those of a grid parameter search, which was generally used in DTI predictions. Moreover, if a slight accuracy reduction (approximately 0.002 for AUC) is allowed, an increase in the calculation speed of 18 times or more can be obtained. Our results show for the first time that Bayesian optimization works effectively for the DTI prediction problem. By accelerating the time-consuming parameter search, the most advanced models can be used even if the number of drug candidates and target proteins to be predicted increase. Our method's source code is available at https://github.com/akiyamalab/BO-DTI.**

***Keywords*-drug–target interaction (DTI), Bayesian optimization, neighborhood-regularized logistic matrix factorization (NRLMF), Gaussian process mutual information (GP-MI)**

I. Introduction

Information about the interactions between drugs and proteins is important for the derivation of early drug candidate compounds in drug discovery research. Drugs are generally known to interact with one or more proteins and show their efficacy by modulating the operations of proteins [1]. Since the presence of unexpected interacting proteins also causes side effects, information about the interactions between drugs and proteins is regarded as important in drug discovery. However, biochemical experiments require a large cost and time to obtain interaction information; thus, computational techniques for predicting interactions are expected (it is known as the drug–target interaction (DTI) prediction) [2].

Many methods for predicting DTIs have been proposed [2][3][4][5][6][7][8] since Yamanishi *et al.* formalized the DTI prediction problem in 2008 [2]. DTI prediction methods can be roughly categorized into approaches based on the kernel method [2][3][4] and approaches based on matrix factorization [5][6][7][8]. In the early days when DTI prediction was proposed, the kernel method attracted attention; however, several methods based on matrix factorization such as multiple similarities collaborative matrix factorization (MSCMF) [7] and neighborhood-regularized logistic matrix factorization (NRLMF) [8] have recently been proposed. Currently, the state-of-the-art prediction method for DTI is NRLMF [8]. However, learning methods such as NRLMF and MSCMF have many hyperparameters, which are parameters whose values are set before the learning process, to be optimized. Furthermore, model selection has been carried out entirely by a grid search thus far, and a considerable amount of time has been required for learning.

In the machine-learning field, Bayesian optimization has recently attracted attention for the efficient optimization of hyperparameters [9][10][11]. Examples of the wide range applications of Bayesian optimization [12] include recommendation systems [13], robotics [14], and automatic chemical compound design [15]. There are several Bayesian optimization methods, but the Gaussian process mutual information (GP-MI) method proposed by Contal *et al.* has good search efficiency up to the optimum solution [11].

In this study, Bayesian optimization was applied for the first time to the DTI prediction problem with the purpose of reducing the computation time for hyperparameter optimization. By using four benchmark datasets widely used in DTI prediction, the efficiency of hyperparameter optimization was compared for Bayesian optimization and a grid search by cross-validation (CV), and the results show that Bayesian optimization is effective for hyperparameter optimization for the DTI prediction problem.

978-1-5386-2594-1/17/$31.00 ©2017 IEEE

Table I: Statistics of the drug–target interaction datasets.

Statistics	Nuclear receptor	GPCR	Ion channel	Enzyme
No. of drugs	54	223	210	445
No. of targets	26	95	204	664
No. of interactions	90	635	1476	2926

II. Materials and Methods

A. Materials

To evaluate our prediction method, we employ a general benchmark dataset [2] used for drug–protein interaction prediction [2][3][4][5][7][8]. The benchmark consists of four DTI datasets—Nuclear receptor, GPCR, Ion channel, and Enzyme—which are defined by three types of matrices. Each dataset consists of an interaction matrix, a drug similarity matrix, and a target (protein) similarity matrix. The interaction matrix is an adjacency matrix that takes a value of 1 if activity is experimentally confirmed between the drug and the target protein, and takes a value of 0 otherwise. Interaction information is obtained from the KEGG BRITE [16], BRENDA [17], SupterTarget [18], and DrugBank [19] databases. Table I summarizes the statistical information of the interaction matrix in each dataset. Note that even if it takes a value of 0, an interaction may be observed in the future by an experiment, and it can be assigned a value of 1. The chemical structures of the drug are obtained from the KEGG LIGAND database [16], and the Tanimoto coefficient (calculated by SIMCOMP [20]) is used for the similarity between drugs. Amino-acid sequences obtained from the KEGG GENES database [16] are used for the proteins, and the similarity between proteins is calculated using the normalized Smith–Waterman score. These datasets can be obtained at http://web.kuicr.kyoto-u.ac.jp/supp/yoshi/drugtarget/.

B. Problem Formalization

In this paper, we denote the set of drugs as $\mathcal{D} = \{d_i\}_{i=1}^{n_d}$ and set the target to $\mathcal{T} = \{t_j\}_{j=1}^{n_t}$, where n_d and n_t are the number of elements in sets $\mathcal{D}$ and $\mathcal{T}$, respectively. The interaction matrix is represented by the adjacency matrix $\mathbf{Y} \in \{0,1\}^{n_d \times n_t}$. When an interaction between a drug d_i and a target t_j has been confirmed experimentally, $\mathbf{Y}_{ij} = 1$ (to indicate an interaction pair); otherwise, $\mathbf{Y}_{ij} = 0$ (for an unknown pair). In particular, in the interaction matrix, we denote the drugs without interacting target proteins as $\mathcal{D}^- = \{d_i \in \mathcal{D} \mid \forall l, \mathbf{Y}_{il} = 0\}$ (negative drugs). We denote the targets without any interaction drugs as $\mathcal{T}^- = \{t_j \in \mathcal{T} \mid \forall l, \mathbf{Y}_{lj} = 0\}$ (negative targets). Moreover, let $\mathcal{D}^+ = \mathcal{D}\backslash\mathcal{D}^-$ be positive drugs, and let $\mathcal{T}^+ = \mathcal{T}\backslash\mathcal{T}^-$ be positive targets. Let $(\mathbf{S}_d)_{il} \in [0,1]$ be the similarity between drugs d_i and d_l, and let $\mathbf{S}_d \in [0,1]^{n_d \times n_d}$ be the drug similarity matrix. Similarly, let the similarity between targets t_j and t_l be $(\mathbf{S}_t)_{jl} \in [0,1]$ and $\mathbf{S}_t \in [0,1]^{n_t \times n_t}$ be the target similarity matrix. The aim of this problem is to assign scores to unknown pairs and rank drug and target pairs in descending order of the likelihood of interaction.

C. NRLMF

NRLMF [8] is a state-of-the-art method for predicting the interactions between drugs and targets based on matrix factorization.

1) Interaction Probability: The possibility that drug d_i and target t_j interact is evaluated by the interaction probability p_{ij} calculated from latent feature vectors of the drug and target. The latent feature vector of drug d_i is represented by $\mathbf{u}_i \in \mathbb{R}^r$, and the latent feature vector of the protein t_j is represented by $\mathbf{v}_j \in \mathbb{R}^r$, where r is a hyperparameter representing the number of dimensions of the latent feature vector. $\mathbf{U} \in \mathbb{R}^{n_d \times r}$ is a latent feature matrix that has the latent feature vector $\mathbf{u}_i^\top$ as a row vector, and $\mathbf{V} \in \mathbb{R}^{n_t \times r}$ is a latent feature matrix that has the latent feature vector $\mathbf{v}_i^\top$ as a row vector. At this time, the interaction probability between drug d_i and protein t_j is defined as follows:

$$p_{ij} = \frac{\exp(\mathbf{u}_i^\top \mathbf{v}_j)}{1 + \exp(\mathbf{u}_i^\top \mathbf{v}_j)}. \tag{1}$$

2) Prediction Model: We define the likelihood of interaction matrix $\mathbf{Y}$ in the latent feature matrix $\mathbf{U}, \mathbf{V}$ as follows:

$$\Pr(\mathbf{Y} \mid \mathbf{U}, \mathbf{V}) = \prod_{i=1}^{n_d} \prod_{j=1}^{n_t} p_{ij}^{c\mathbf{Y}_{ij}} (1 - p_{ij})^{1 - \mathbf{Y}_{ij}}, \tag{2}$$

where $c > 0$ is a hyperparameter for balancing the number of interaction pairs and unknown pairs. For latent feature matrixes $\mathbf{U}$ and $\mathbf{V}$, the multivariate normal distribution with a mean of $\mathbf{0}$ and the prior distribution are defined as

$$\Pr(\mathbf{U} \mid \mathbf{\Sigma}_d) = \prod_{i=1}^{n_d} \mathcal{N}(\mathbf{u}_i \mid \mathbf{0}, \mathbf{\Sigma}_d), \tag{3}$$

$$\Pr(\mathbf{V} \mid \mathbf{\Sigma}_t) = \prod_{j=1}^{n_t} \mathcal{N}(\mathbf{v}_j \mid \mathbf{0}, \mathbf{\Sigma}_t), \tag{4}$$

where the variance–covariance matrices are expressed as $\mathbf{\Sigma}_d = (\lambda_d \mathbf{I}_d + \alpha \mathbf{L}_d)^{-1}$ and $\mathbf{\Sigma}_t = (\lambda_t \mathbf{I}_t + \beta \mathbf{L}_t)^{-1}$. Moreover, $\lambda_d, \lambda_t, \alpha, \beta > 0$ are hyperparameters, and $\mathbf{I}_d \in \mathbb{R}^{n_d \times n_d}$ and $\mathbf{I}_t \in \mathbb{R}^{n_t \times n_t}$ are identity matrices. $\mathbf{L}_d \in \mathbb{R}^{n_d \times n_d}$; when the K_1-nearest neighbors for the drug $NN_{K_1}(d_i) \subset \mathcal{D}\backslash\{d_i\}$, the matrix that satisfies the following relationship:

$$\sum_{i=1}^{n_d} \sum_{l=1}^{n_d} a_{il} \|\mathbf{u}_i - \mathbf{u}_l\|_2^2 = tr(\mathbf{U}^\top \mathbf{L}_d \mathbf{U}), \tag{5}$$

$$a_{il} = \begin{cases} (\mathbf{S}_d)_{il} & \textit{if } d_l \in NN_{K_1}(d_i) \\ 0 & \textit{otherwise} \end{cases}, \tag{6}$$

where $\|\cdot\|_2$ is the Euclidean norm. $\mathbf{L}_t \in \mathbb{R}^{n_t \times n_t}$ is defined in the same way.

3) Estimation and Scoring: Using Bayes' theorem, the posterior distribution of the latent feature matrices is considered using (7), and MAP estimation is performed using (8), thereby obtaining solutions $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}$.

$$\Pr(\mathbf{U}, \mathbf{V} \mid \mathbf{Y}) \propto \Pr(\mathbf{Y}|\mathbf{U}, \mathbf{V}) \Pr(\mathbf{U}|\mathbf{\Sigma}_d) \Pr(\mathbf{V}|\mathbf{\Sigma}_t) \quad (7)$$

$$\hat{\mathbf{U}}, \hat{\mathbf{V}} = \underset{\mathbf{U},\mathbf{V}}{\operatorname{argmax}} \Pr(\mathbf{U}, \mathbf{V} \mid \mathbf{Y}) \quad (8)$$

The optimization algorithm for obtaining the solution uses the alternating gradient descent method [21]. In addition, AdaGrad [22] with hyperparameter θ corresponding to the gradient coefficient increases the rate of learning. In the estimated solutions to $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}$, the values for negative drugs and negative targets are modified. For a negative drug $d_i \in \mathcal{D}^-$, let the K_2-nearest neighbors of positive drugs be $NN_{K_2}(d_i) \subset \mathcal{D}^+$, where $\hat{\mathbf{u}}_i \in \mathbb{R}^r$ is the transposed row vector of the i-th row of the estimated solution $\hat{\mathbf{U}}$. At this time, the latent feature vector of the drug d_i is corrected by using

$$\tilde{\mathbf{u}}_i = \begin{cases} \hat{\mathbf{u}}_i & \mathit{if}\ \ d_i \in \mathcal{D}^+ \\ \frac{\sum_{d_l \in NN_{K_2}(d_i)} (\mathbf{S}_d)_{il} \hat{\mathbf{u}}_l}{\sum_{d_l \in NN_{K_2}(d_i)} (\mathbf{S}_d)_{il}} & \mathit{if}\ \ d_i \in \mathcal{D}^- \end{cases}. \quad (9)$$

We also modify the latent feature vectors of the negative target $t_j \in \mathcal{T}^-$ in the same way. The score $\tilde{p}_{ij}$ of drug d_i and target t_j is calculated using the modified latent feature vectors as follows:

$$\tilde{p}_{ij} = \frac{\exp(\tilde{\mathbf{u}}_i^\top \tilde{\mathbf{v}}_j)}{1 + \exp(\tilde{\mathbf{u}}_i^\top \tilde{\mathbf{v}}_j)}. \quad (10)$$

D. GP-MI

The GP-MI algorithm [11] is a state-of-the-art method for Bayesian optimization.

1) Bayesian Optimization: $f : \chi \to \mathbb{R}$ is an implicit function that needs to be optimized, where $\chi \subset \mathbb{R}^n$ ($n \in \mathbb{N}$) is a compact convex set. At this time, the purpose of Bayesian optimization is to estimate the optimal solution of the function f through consecutive queries $x_1, x_2, ... \in \chi$. At the $(T+1)$-th iteration, the new query x_{T+1} is chosen from χ on the basis of the previously obtained queries $\mathcal{X}_T = \{x_1, ..., x_T\}$ and the observation values $\mathcal{Y}_T = \{y_1, ..., y_T\}$. The relationship between the observed value $y \in \mathbb{R}$ and the query $x \in \chi$ is defined using the noise ϵ according to Gaussian distribution $\mathcal{N}(\epsilon \mid 0, \sigma^2)$ as follows:

$$y = f(x) + \epsilon. \quad (11)$$

2) Gaussian Process: The function f follows a Gaussian process $GP(m, k)$ [23], where $m : \chi \to \mathbb{R}$ is a mean function, and $k : \chi \times \chi \to \mathbb{R}$ is a kernel function. By assuming a Gaussian process, f is supposed to be smooth implicitly. Let the mean function be zero, i.e., $m : \chi \to 0$. The kernel function k is uses the following squared exponential kernel:

$$k(x, x') = \exp\left(-\frac{1}{2}\|x - x'\|_2^2\right). \quad (12)$$

On the basis of the queries $\mathcal{X}_T$ and the observation values $\mathbf{y}_T = (y_1, ..., y_T)^\top \in \mathbb{R}^T$ up to time T, the expected value $\mu_{T+1}(x)$ and variance $\sigma^2_{T+1}(x)$ of the observation value $y \in \mathbb{R}$ for $\forall x \in \chi$ at time $(T+1)$ are defined as

$$\mu_{T+1}(x) = \mathbf{k}_T(x)^\top \mathbf{C}_T^{-1} \mathbf{y}_T, \quad (13)$$

$$\sigma^2_{T+1}(x) = k(x, x) - \mathbf{k}_T(x)^\top \mathbf{C}_T^{-1} \mathbf{k}_T(x), \quad (14)$$

where $\mathbf{k}_T(x) = [k(x_\tau, x)]_{x_\tau \in \mathcal{X}_T} \in \mathbb{R}^T$, and $\mathbf{C}_T = \mathbf{K}_T + \sigma^2 \mathbf{I}$ ($\mathbf{K}_T = [k(x_\tau, x_{\tau'})]_{x_\tau, x_{\tau'} \in \mathcal{X}_T} \in \mathbb{R}^{T \times T}$). $\sigma^2 \in \mathbb{R}$ is the variance of the noise, and $\mathbf{I} \in \mathbb{R}^{T \times T}$ is the identity matrix.

3) Algorithm: The GP-MI algorithm selects the next query $x_\tau \in \chi$ using $\mu_\tau(x)$ and $\sigma^2_\tau(x)$ calculated using (13), (14), and

$$x_\tau = \underset{x \in \chi}{\operatorname{argmax}}\, \mu_\tau(x) + \phi_\tau(x), \quad (15)$$

where $\phi_\tau : \chi \to \mathbb{R}$ is an increasing function of $\sigma^2_\tau(x)$. Function ϕ_τ is expressed using parameters δ and $\gamma_{\tau-1}$ as

$$\phi_\tau(x) = \sqrt{\log \frac{2}{\delta}} \left(\sqrt{\sigma^2_\tau(x) + \gamma_{\tau-1}} - \sqrt{\gamma_{\tau-1}}\right), \quad (16)$$

where $\gamma_{\tau-1} = \sum_{\tau'=1}^{\tau-1} \sigma^2_{\tau'}(x_{\tau'})$. $\delta > 0$ is a hyperparameter of the GP-MI algorithm, and its termination condition is $x_{\tau+1} = x_\tau$.

III. Results and Discussion

By comparing the calculation time and prediction accuracy of a grid search and Bayesian optimization by CV, we confirm that the calculation time is reduced while maintaining the prediction accuracy.

A. Experimental Settings

Four datasets, Nuclear receptor, GPCR, Ion channel, and Enzyme, as mentioned in Sect. II-A, were used. In order to measure the calculation time and prediction accuracy (the area under the receiver operating characteristic curve (AUC)) of the grid search and Bayesian optimization, 10-fold CV was performed 200 times, and there were 200 samples (one sample consists of the average computational time and average AUC) for each optimization method. Note that there is correspondence between each sample of the grid search and Bayesian optimization. The sample size is 200, where the effect size evaluated by Cohen's d [24] is 0.2, the significance level is 0.05, and the statistical power is 0.8 or more when a t-test with the correspondence relationship is carried out. The definition of Cohen's d is given by

$$d = \frac{\bar{x}_{GS} - \bar{x}_{BO}}{s}, \quad (17)$$

Table II: Computational times [sec] for a grid search and Bayesian optimization (GP-MI).

Dataset	Grid search	GP-MI					
		$\delta = 10^{-100}$	$\delta = 10^{-80}$	$\delta = 10^{-60}$	$\delta = 10^{-40}$	$\delta = 10^{-20}$	$\delta = 10^{0}$
Nuclear receptor	2135 ± 15	250 ± 73	246 ± 66	233 ± 61	207 ± 65	104 ± 37	30 ± 47
GPCR	19066 ± 80	2092 ± 95	2053 ± 86	1974 ± 91	1705 ± 95	965 ± 224	140 ± 68
Ion channel	32687 ± 133	3926 ± 146	3864 ± 138	3743 ± 165	3259 ± 154	1823 ± 403	230 ± 100
Enzyme	225493 ± 2682	23070 ± 751	22714 ± 728	22004 ± 743	19092 ± 818	10845 ± 2184	1521 ± 673

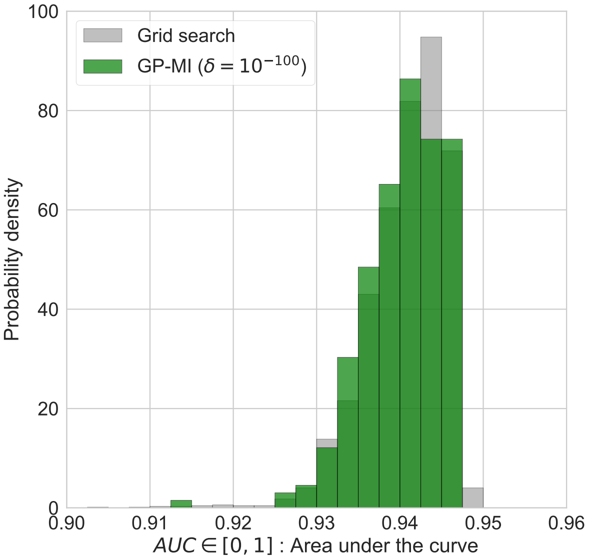

Figure 1: Probability densities of the AUC in the hyperparameter optimization of NRLMF by using a grid search and Bayesian optimization (GP-MI). Each AUC is an average value of the 10-fold CV of a hyperparameter tuple chosen from 2688 combinations iteratively. The GP-MI algorithm is terminated after 264 iterations.

$$s = \sqrt{\frac{(n_{GS} - 1)s_{GS}^2 + (n_{BO} - 1)s_{BO}^2}{n_{GS} + n_{BO} - 2}}, \qquad (18)$$

where n_{GS}, $\bar{x}_{GS}$, and s_{GS} are the sample size, average value, and unbiased standard deviation of the samples for the grid search, respectively; and n_{BO}, $\bar{x}_{BO}$, and s_{BO} are the sample size, average value, and unbiased standard deviation of the samples for Bayesian optimization, respectively.

To divide the data in the CV, we used the CV scenario (CVS1) according to the NRLMF paper [8]. In this scenario, the drug–target pairs were divided into 10 sets with almost the same size, regardless of whether they were interaction pairs or unknown pairs. One of the ten sets was used as test data, and an adjacency matrix in which the values of the test data are replaced with zero was used as training data. Each set became test data iteratively, and the computational time and AUC were measured 10 times. Then, one sample can be obtained from the mean of the computational time and the mean of the AUC for 10-fold CV.

The hyperparameter of NRLMF adopted the settings reported in [8], $c = 5$, $K_1 = K_2 = 5$, $r = \{50, 100\}$, $\alpha = \{2^{-5}, 2^{-4}, ..., 2^{2}\}$, $\beta = \{2^{-5}, 2^{-4}, ..., 2^{0}\}$, $\lambda_d = \lambda_t = \{2^{-5}, 2^{-4}, ..., 2^{1}\}$, and $\theta = \{2^{-3}, 2^{-2}, ..., 2^{0}\}$. At this time, the set of hyperparameters is a subspace in $\mathbb{R}^9$ with 2688 elements, which corresponds to χ for Bayesian optimization. We set the GP-MI parameter to $\delta = \{10^{-100}, 10^{-80}, ..., 10^{0}\}$. In addition, the kernel function in the Gaussian process used the squared exponential kernel in (12), and the noise variance was set to $\sigma^2 = 0.1$.

All calculations were conducted on the TSUBAME 2.5 supercomputing system of the Tokyo Institute of Technology, Japan. We used one CPU core of its thin node, which consists of two Intel Xeon 5670 processors (6 cores) running at 2.93 GHz with 54 GB RAM, on the SUSE Linux Enterprise Server 11 SP3 operating system.

B. *Computational Time*

Table II summarizes the computational times for the grid search and Bayesian optimization. Comparing Bayesian optimization with $\delta = 10^{-100}$ and the grid search, the computational time was decreased by 8.54 times for Nuclear receptor, 9.11 times for GPCR, 8.33 times for Ion channel, and 9.77 times for Enzyme. On the other hand, comparing Bayesian optimization with $\delta = 10^{-20}$ and the grid search, the computational time was decreased by 20.53, 19.76, 17.93, and 20.79 times for the Nuclear receptor, GPCR, Ion channel, and Enzyme datasets, respectively. These results show that Bayesian optimization (GP-MI) can reduce the computational time by a factor of 8.94 or more on average compared to a grid search. Further, if the number of hyperparameters is the same, the same speed increase can be expected regardless of the size of the dataset. As a property of the Bayesian optimization parameter δ, a larger value of δ results in a lower calculation time. (However, a larger value of δ results in a tendency to fall into a local solution; thus, the calculation time is lowered, but the prediction accuracy tends to deteriorate.)

The behavior of a searched space in which histograms compared GP-MI with $\delta = 10^{-100}$ and Grid search using AUC is shown in Fig. 1. This figure shows that the GP-MI algorithm searched the entire hyperparameter space with fewer searches.

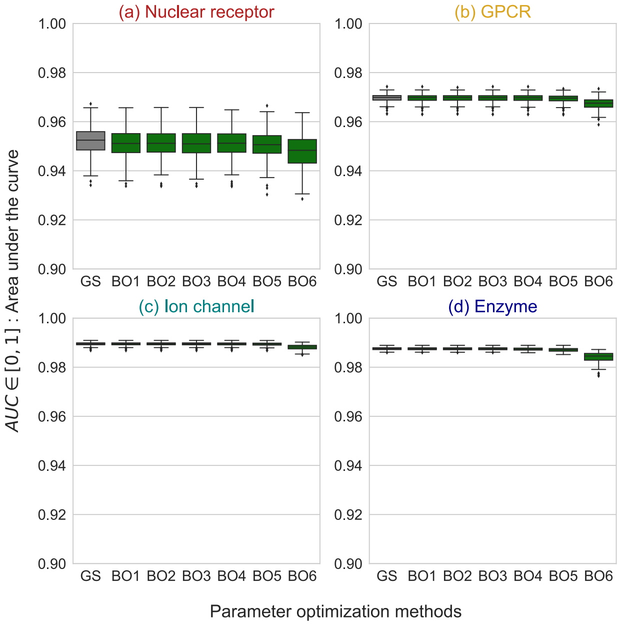

Figure 2: Box plots of the average AUCs (1-time 10-fold CV) consisting of 200 samples for each dataset and optimization scenario. GS indicates grid search; BO1, GP-MI with $\delta = 10^{-100}$; BO2, GP-MI with $\delta = 10^{-80}$; BO3, GP-MI with $\delta = 10^{-60}$; BO4, GP-MI with $\delta = 10^{-40}$; BO5, GP-MI with $\delta = 10^{-20}$; and BO6, GP-MI with $\delta = 10^{0}$.

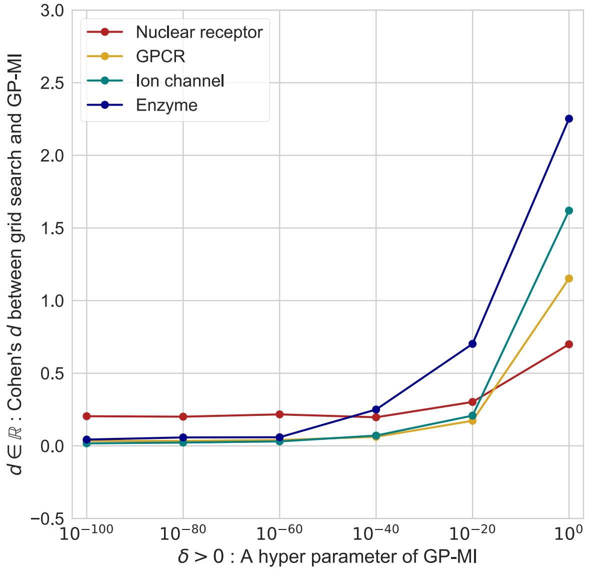

Figure 3: Plot of Cohen's d between the grid search and Bayesian optimization for each dataset.

C. Prediction Accuracy

Fig. 2 shows box plots of the AUCs for grid search and Bayesian optimization. These results show that the prediction accuracy of Bayesian optimization is close to that of the grid search when the parameter δ is smaller. Performance on the Nuclear receptor dataset (Fig. 2(a)) has larger variances than on the other datasets because its size is smaller. Comparing Bayesian optimization with $\delta = 10^{-100}$ and the grid search, the difference in the average AUCs for these methods is 0.0012 for Nuclear receptor, 0.0001 for GPCR, less than 0.0001 for Ion channel, and less than 0.0001 for Enzyme. On the other hand, comparing Bayesian optimization with $\delta = 10^{-20}$ and the grid search, the difference in the average AUCs for these methods is 0.0017, 0.0003, 0.0001, and 0.0004 for the Nuclear receptor, GPCR, Ion channel, and Enzyme datasets, respectively. These results show that the prediction accuracy of Bayesian optimization tends to improve as the size of the dataset becomes larger. As a property of the parameter δ, the prediction accuracy tends to decrease as the value of δ becomes larger, which means that the prediction accuracy has a trade-off relationship with the computational time.

Fig. 3 shows a plot of Cohen's d, to determine the effect size between the grid search and Bayesian optimization. Cohen's d is a statistic for assessing the difference between the mean values of two independent groups, and it can be considered that the difference between the two groups is small as it approaches zero. In particular, $d = 0.01$ is very small (negligible), and $d = 0.2$ is said to be small [25]. For Bayesian optimization with $\delta = 10^{-100}$, it is found that the effect of size between these methods is 0.204, 0.035, 0.017, and 0.043 for the Nuclear receptor, GPCR, Ion channel, and Enzyme datasets, respectively. These results show that the difference in the average values of the AUCs of the grid search and Bayesian optimization is small for all datasets. In particular, for GPCR, Ion channel, and Enzyme, which have large dataset sizes, the difference is negligibly small. As a property of the parameter δ, Cohen's d converges to a certain value greater than zero when δ is sufficiently small.

IV. Conclusion

We proposed an optimization for hyperparameter searches with Bayesian optimization (GP-MI) [11] using the state-of-the-art method NRLMF [8] to predict interaction between drugs and proteins. In the computational experiment, 10-fold CV was performed 200 times for a statistical comparison with four datasets, Nuclear receptor, GPCR, Ion channel, and Enzyme. As a result, it was shown that the computational time for Bayesian optimization was reduced by a factor of 8.94 on average compared to that for a grid search. Although the prediction accuracy (AUC) of Bayesian optimization was inferior to that of the grid search, the Cohen's d for the effect size was shown to be about 0.01–0.2 when the parameter δ was sufficiently small. This means that the prediction accuracy of Bayesian optimization is sufficiently close to that of the grid search. These results enabled us to conclude that the optimization of the hyperparameter search can be accelerated to predict interaction between drugs and proteins by using Bayesian optimization for a conventional grid search.

It is possible to apply Bayesian optimization to prediction methods such as PMF [6] and MSCMF [7], which have a large number of hyperparameters. Moreover, by parallelizing Bayesian optimization [26], it is possible to lower the computational time. This is expected to enable learning to be performed efficiently for large-scale datasets.

Acknowledgment

This work was partially supported by JST CREST "Extreme Big Data" (No. JPMJCR1303), JST Research Complex Program "Wellbeing Research Campus: Creating new values through technological and social innovation," JSPS KAKENHI (Nos. 15J11261, 17H01814, 15K16081, 15H02776), and AMED BINDS *In Silico* Unit.

References

[1] M. A. Yildirim, K.-I. Goh, M. E. Cusick, A.-L. Barabási, and M. Vidal, "Drug–target network," *Nat. Biotechnol.*, vol. 25, pp. 1119–1126, 2007.

[2] Y. Yamanishi, M. Araki, A. Gutteridge, W. Honda, and M. Kanehisa, "Prediction of drug–target interaction networks from the integration of chemical and genomic spaces," *Bioinformatics*, vol. 24, no. 13, pp. i232–i240, 2008.

[3] T. van Laarhoven and E. Marchiori, "Predicting drug–target interactions for new drug compounds using a weighted nearest neighbor profile," *PLOS ONE*, vol. 8, no. 6, Art. no. e66952, 2013.

[4] J.-P. Mei, C.-K. Kwohm, P. Yang, X.-L. Li, and J. Zheng, "Drug–target interaction prediction by learning from local information and neighbors," *Bioinformatics*, vol. 29, no. 2, pp. 238–245, 2012.

[5] M. Gŏnen, "Predicting drug–target interactions from chemical and genomic kernels using Bayesian matrix factorization," *Bioinformatics*, vol. 28, no. 18, pp. 2304–2310, 2012.

[6] M. C. Cobanoglu, C. Liu, F. Hu, Z. N. Oltvai, and I. Bahar, "Predicting drug–target interactions using probabilistic matrix factorization," *J. Chem. Inf. Model.*, vol. 53, no. 12, pp. 3399–3409, 2013.

[7] X. Zheng, H. Ding, H. Mamitsuka, and S. Zhu, "Collaborative matrix factorization with multiple similarities for predicting drug-target interactions," In *Proc. 19th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, pp. 1025–1033, 2013.

[8] Y. Liu, M. Wu, C. Miao, P. Zhao, and X.-L. Li, "Neighborhood regularized logistic matrix factorization for drug–target interaction prediction," *PLOS Comput. Biol.*, vol. 12, no. 2, Art. no. e1004760, 2016.

[9] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *J. Glob. Optim.*, vol. 13, no. 4, pp. 455–492, 1998.

[10] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, "Information-theoretic regret bounds for Gaussian process optimization in the bandit setting," *IEEE Trans. Inf. Theory*, vol. 58, no. 5, pp. 3250–3265, 2012.

[11] E. Contal, V. Perchet, and N. Vayatis, "Gaussian process optimization with mutual information," In *Proc. 31st Int. Conf. Machine Learning (ICML 2014)*, vol. 32, pp. II-253–II-261, 2014.

[12] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proc. IEEE*, vol. 104, no. 1, pp. 148–175, 2016.

[13] O. Chapelle and L. Li, "An empirical evaluation of Thompson sampling," In *Adv. in Neural Inform. Process. Syst. (NIPS)*, vol. 24, pp. 2249–2257, 2011.

[14] D. J. Lizotte, T. Wang, M. Bowling, and D. Schuurmans, "Automatic gait optimization with Gaussian process regression," In *Proc. 20th Int. Joint Conf. on Artificial Intell. (IJCAI'07)*, vol. 7, pp. 944–949, 2007.

[15] R. Gómez-Bombarelli *et al.*, "Automatic chemical design using a data-driven continuous representation of molecules," *arXiv preprint*, Art. no. 1610.02415, 2016.

[16] M. Kanehisa *et al.*, "From genomics to chemical genomics: New developments in KEGG," *Nucleic Acids Res.*, vol. 34 suppl. 1, pp. D354–D357, 2006.

[17] I. Schomburg *et al.*, "BRENDA, the enzyme database: Updates and major new developments," *Nucleic Acids Res.*, vol. 32 suppl. 1, pp. D431–D433, 2004.

[18] S. Günther *et al.*, "SuperTarget and Matador: Resources for exploring drug-target relationships," *Nucleic Acids Res.*, vol. 36 suppl. 1, pp. D919–D922, 2008.

[19] D. S. Wishart *et al.*, "DrugBank: A knowledgebase for drugs, drug actions and drug targets," *Nucleic Acids Res.*, vol. 36 suppl. 1, pp. D901–D906, 2008.

[20] M. Hattori, Y. Okuno, S. Goto, and M. Kanehisa, "Development of a chemical structure comparison method for integrated analysis of chemical and genomic information in the metabolic pathways," *J. Am. Chem. Soc.*, vol. 125, no. 39, pp. 11853–11865, 2003.

[21] C. C. Johnson, "Logistic matrix factorization for implicit feedback data," In *NIPS2014 Workshop on Distributed Machine Learning and Matrix Computations*, 2014.

[22] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Jul. 2011.

[23] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, Cambridge: MIT Press, 2006.

[24] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, 2nd ed., Hillsdale, NJ: Erlbaum, 1988.

[25] S. S. Sawilowsky, "New effect size rules of thumb," *J. Mod. Appl. Stat. Methods*, vol. 8, no. 2, pp. 467–474, 2009.

[26] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," In *Adv. in Neural Inform. Process. Syst. (NIPS)*, vol. 25, 2012.