Efficient recommendation tool of materials by an executable file based on machine learning

Kei Terayama1,2,3*, Koji Tsuda1,4,5*, and Ryo Tamura1,4,5,6*

1RIKEN Center for Advanced Intelligence Project, Tokyo 103-0027, Japan
2Medical Sciences Innovation Hub Program, RIKEN Cluster for Science, Technology and Innovation Hub, Kanagawa 230-0045, Japan
3Graduate School of Medicine, Kyoto University, Kyoto 606-8507, Japan
4Graduate School of Frontier Sciences, University of Tokyo, Chiba 277-8568, Japan
5Research and Services Division of Materials Data and Integrated System, National Institute for Materials Science, Ibaraki 305-0047, Japan
6International Center for Materials Nanoarchitectonics (WPI-MANA), National Institute for Materials Science, Ibaraki 305-0044, Japan
7E-mail: kei.terayama@riken.jp; tsuda@k.u-tokyo.ac.jp; tamura.ryo@nims.go.jp

Received June 12, 2019; revised July 8, 2019; accepted July 22, 2019; published online August 7, 2019

1. Introduction

Materials informatics (MI) research has received tremendous attention.1–6 These studies should accelerate materials discovery from both academic and business perspectives. With the aid of MI techniques, it is now possible to develop novel materials with the desired properties using the smallest number of syntheses or simulations.7–12 In such cases, MI techniques efficiently recommend candidate materials compositions and process parameters to discover materials with the desired properties, even if the amount of materials data is small. Furthermore, other MI techniques can also propose next candidate points to efficiently construct phase diagrams.13,14

Enormous machine learning tools, which can be adopted for MI research, have been developed and released. However, many of these tools are developed as a code of specific programming languages such as Python. Thus, the proper environment should be prepared for specific languages when these tools are performed on computers. This can be burdensome, especially for non-programmers. To proliferate machine learning techniques in materials science, easy-to-use MI tools that can be easily adopted are essential. We developed applications for MI, which can be performed on Windows computers without special settings. We targeted Windows computers due to the ubiquity and the fact that they control many experimental devices.

This paper explains how to use our developed applications (see Fig. 1), the computing time to obtain the next candidate, and the sampling results for a ternary phase diagram. Our applications efficiently perform Bayesian optimization to optimize materials properties and uncertainty sampling to complete a new phase diagram. We introduce the usage of these applications and show the sampling results for a ternary phase diagram. © 2019 The Japan Society of Applied Physics

1. Introduction

Materials informatics (MI) research has received tremendous attention.1–6 These studies should accelerate materials discovery from both academic and business perspectives. With the aid of MI techniques, it is now possible to develop novel materials with the desired properties using the smallest number of syntheses or simulations.7–12 In such cases, MI techniques efficiently recommend candidate materials compositions and process parameters to discover materials with the desired properties, even if the amount of materials data is small. Furthermore, other MI techniques can also propose next candidate points to efficiently construct phase diagrams.13,14

Efficient recommendation tool of materials by an executable file based on machine learning

Kei Terayama1,2,3*, Koji Tsuda1,4,5*, and Ryo Tamura1,4,5,6*

1RIKEN Center for Advanced Intelligence Project, Tokyo 103-0027, Japan
2Medical Sciences Innovation Hub Program, RIKEN Cluster for Science, Technology and Innovation Hub, Kanagawa 230-0045, Japan
3Graduate School of Medicine, Kyoto University, Kyoto 606-8507, Japan
4Graduate School of Frontier Sciences, University of Tokyo, Chiba 277-8568, Japan
5Research and Services Division of Materials Data and Integrated System, National Institute for Materials Science, Ibaraki 305-0047, Japan
6International Center for Materials Nanoarchitectonics (WPI-MANA), National Institute for Materials Science, Ibaraki 305-0044, Japan
7E-mail: kei.terayama@riken.jp; tsuda@k.u-tokyo.ac.jp; tamura.ryo@nims.go.jp

Received June 12, 2019; revised July 8, 2019; accepted July 22, 2019; published online August 7, 2019

1. Introduction

Materials informatics (MI) research has received tremendous attention.1–6 These studies should accelerate materials discovery from both academic and business perspectives. With the aid of MI techniques, it is now possible to develop novel materials with the desired properties using the smallest number of syntheses or simulations.7–12 In such cases, MI techniques efficiently recommend candidate materials compositions and process parameters to discover materials with the desired properties, even if the amount of materials data is small. Furthermore, other MI techniques can also propose next candidate points to efficiently construct phase diagrams.13,14

Enormous machine learning tools, which can be adopted for MI research, have been developed and released. However, many of these tools are developed as a code of specific programming languages such as Python. Thus, the proper environment should be prepared for specific languages when these tools are performed on computers. This can be burdensome, especially for non-programmers. To proliferate machine learning techniques in materials science, easy-to-use MI tools that can be easily adopted are essential. We developed applications for MI, which can be performed on Windows computers without special settings. We targeted Windows computers due to the ubiquity and the fact that they control many experimental devices.

This paper explains how to use our developed applications (see Fig. 1), the computing time to obtain the next candidate, and the sampling results for a ternary phase diagram. Our applications efficiently perform Bayesian optimization to optimize materials properties and uncertainty sampling to complete a new phase diagram. We introduce the usage of these applications and show the sampling results for a ternary phase diagram. © 2019 The Japan Society of Applied Physics

1. Introduction

Materials informatics (MI) research has received tremendous attention.1–6 These studies should accelerate materials discovery from both academic and business perspectives. With the aid of MI techniques, it is now possible to develop novel materials with the desired properties using the smallest number of syntheses or simulations.7–12 In such cases, MI techniques efficiently recommend candidate materials compositions and process parameters to discover materials with the desired properties, even if the amount of materials data is small. Furthermore, other MI techniques can also propose next candidate points to efficiently construct phase diagrams.13,14

Enormous machine learning tools, which can be adopted for MI research, have been developed and released. However, many of these tools are developed as a code of specific programming languages such as Python. Thus, the proper environment should be prepared for specific languages when these tools are performed on computers. This can be burdensome, especially for non-programmers. To proliferate machine learning techniques in materials science, easy-to-use MI tools that can be easily adopted are essential. We developed applications for MI, which can be performed on Windows computers without special settings. We targeted Windows computers due to the ubiquity and the fact that they control many experimental devices.

This paper explains how to use our developed applications (see Fig. 1), the computing time to obtain the next candidate, and the sampling results for a ternary phase diagram. Our applications are composed of two executable files: COMBO.exe and PDC.exe. These are available from our project page (https://tsudalab.org/en/projects/mitools). Both files can be executed on a Windows computer. We performed an operation check on Windows 7, 8.1, and 10 (64-bit version). For ease of use, external parameters are not necessary for these applications.

COMBO.exe is a Bayesian optimization tool based on the COMmon Bayesian Optimization library (COMBO).7 This selects candidate materials, which may have the desired properties, for syntheses or simulations by machine learning. In COMBO, the Gaussian process is approximated using the Bayesian linear model with a random feature map7 to overcome the computational bottleneck of the Bayesian optimization framework. Furthermore, hyperparameters are automatically determined by maximizing the type-II likelihood,16 while the regularization term avoids overfitting. In COMBO.exe, the Thompson sampling is adopted to select the next candidate from the posterior distribution.17

On the other hand, the selection of a candidate point to efficiently complete a phase diagram can be performed by PDC.exe based on the uncertainty sampling for the phase diagram construction (PDC).14 Using uncertainty sampling, the point with the highest uncertainty score, as evaluated by a machine learning based classification model, is chosen as an informative sample. Typically, data with the highest uncertainty are located near phase boundaries. This approach is useful to construct complicated phase diagrams from scratch. In PDC.exe, the label propagation method18 is used as a phase estimation and the uncertainty score is defined by the least confident method.19

2. Application framework

First, the data file to be read by our applications should be prepared. Let $x_1, x_2, \ldots, x_N \in \mathbb{R}^d$ be a set of prepared candidate points, which corresponds to the search space. For example, this $d$-dimensional vector $x_i$ includes materials compositions, materials descriptors, or process parameters. Here, it is assumed that the values of $y(x_i)$, which called objective variable, are known at only some data points among the prepared candidate points, and the number of known points is denoted as $M$. In this problem setting, our applications recommend one unknown point, which should be investigated in the next step from the remaining $N-M$ candidate points. The procedure to create the data file (see Fig. 1) is as follows:

1. The data file, called “data.csv”, should be prepared in the csv format (e.g. using Microsoft Excel).
(2) The first row should denote explanations of each column for objective variable, \( y(x) \), and explanatory variables, \( x_i \).

(3) The objective variable, \( y(x) \), are contained in the first column. Since only the \( M \) values of \( y(x) \) are known, this column may have a lot of blanks \((N-M\) blanks), and our applications recommend the next candidate point from these blanks. For COMBO.exe, \( y(x) \) may be a real value, and this application selects the next candidate, which has a large predicted value of \( y(x) \) in consideration of uncertainty. Here, we discuss the determination of \( y(x) \) in materials science. Let the physical properties of \( i \)th materials be denoted as \( a_i \), \( b_i \), and \( c_i \). The following definitions of objective variable are possible:

**Aim1:** Materials with a large physical property of \( a_i \) are searched:

\[
 y(x_i) = a_i. \tag{1} 
\]

**Aim2:** Materials with a small physical property of \( a_i \) are searched:

\[
 y(x_i) = -a_i. \tag{2} 
\]

**Aim3:** Materials with \( a_i = \alpha \) should be obtained:

\[
 y(x_i) = -(\alpha - a_i)^2. \tag{3} 
\]

The real constant value \( \alpha \) is prepared by the experimenter.

**Aim4:** Materials with large \( a_i \), \( b_i \), and \( c_i \) should be obtained:

\[
 y(x_i) = w_a a_i + w_b b_i + w_c c_i. \tag{4} 
\]

\( w_a \), \( w_b \), and \( w_c \) are positive artificial weights, which express the priorities of the properties to be optimized. These values are prepared by the experimenter.

**Aim5:** Materials with \( a_i = \alpha \) and \( b_i = \beta \) should be obtained:

\[
 y(x_i) = -w_a (\alpha - a_i)^2 - w_b (\beta - b_i)^2. \tag{5} 
\]

\( w_a \) and \( w_b \) are positive artificial weights, which express the priorities of the properties to be optimized. \( \alpha \), \( \beta \), \( w_a \), and \( w_b \) values are prepared by the experimenter.
On the other hand, for PDC.exe, \( y(x_i) \) should be integer values, which express the index of the phases. This application selects the next candidate with the highest uncertainty in the phase diagram.

(4) From the second column, the explanatory variables \( x_i \) for all prepared candidate points are packed. In this part, the blanks are prohibited. Note that as the number of candidate points increases, the selection time increases. Here, any number of dimensions of \( x_i \) can be applied in our applications.

Next, when the preparation of the data file is finished, to perform the Bayesian optimization or uncertainty sampling for the PDC on a Windows computer, the following procedures are performed:

1. Place the prepared data.csv in the same folder containing COMBO.exe or PDC.exe.
2. Open the Command Prompt on a Windows computer.
3. Drag-and-drop COMBO.exe or PDC.exe to the Command Prompt window, and press the Enter key.
4. Accept the next candidate point from our applications (e.g., which is outputted in the Command Prompt) as follows: Next Point: [0.74 0.450 333 21] Row Number: 274
5. Close the Command Prompt.
6. Evaluate the objective variable at the proposed point obtained in (4) by syntheses or simulations for COMBO.exe, or identify the type of phases for PDC.exe.
7. Update the first column in data.csv using the information obtained by (5).
8. Repeat (2)–(7) until the experimenters are satisfied.

3. Computation time

We address the computing time to obtain the next candidate recommended by COMBO.exe and PDC.exe. The computing time should depend on the numbers of candidate points \( N \) and known points (training data) \( M \). Figure 2 shows the \( N \) and \( M \) dependence of computing time as the real time when a Windows 10 computer with an Intel Core i7-8650U CPU is used.

For COMBO.exe, the computation time steadily increases against \( N \) and \( M \). Within a few minutes, an enormous number of candidate points \( N \sim 200,000 \) can be processed by COMBO.exe. This computing time is much shorter than the timescale for syntheses or simulations. On the other hand, PDC.exe takes more time than COMBO.exe. Interestingly, as the number of known points \( M \) increases, the computing time is reduced when \( N \) is fixed. It is considered that the label propagation, which is the algorithm of the phase estimation in PDC.exe, converges quickly due to the large number of known points. For practical use, the candidate points should be reduced within \( N = 25,000 \).

4. Sampling demonstrations

We demonstrate samplings on a ternary phase diagram by COMBO.exe and PDC.exe. The target is the NaF-KF-LiF system, which has a simple calculated phase diagram. Intermediate compounds did not appear in this system. Only three phases characterized by NaF, KF, and LiF exist. The left side of Fig. 3 is the contour plot of the melting temperature, while the right side is the phase diagram of this system.

As a demonstration of COMBO.exe, we searched the composition with the minimum melting temperature for the NaF-KF-LiF system. That is, Eq. (2) when \( a_i \) is used as the melting temperature should be used for the objective variable. The left side of Fig. 3 is the sampling results by COMBO.exe when the parameter space is discretized to 352 candidate points. The top figure indicates the positions of five initial points, which are randomly generated. Beginning from the second row, the sampling points are plotted for every four point increase by COMBO.exe. Note that, practically, its sampling point increases by one point. As the iteration continues, points with low melting temperatures are sampled. In addition, points far from the already sampled points are also chosen by uncertainty.

The right side of Fig. 3 is the sampling results by PDC.exe to efficiently complete the ternary phase diagram. The five initial points are the same with COMBO.exe, and the results for every eight point increase are plotted by PDC.exe. Points around the phase boundaries are actively sampled. In the initial points, the KF phase is not detected. Consequently, the candidate points around the phase boundary between NaF and LiF are intensively sampled at the beginning. In the next step, when the KF phase is found, PDC.exe suddenly recommend points near the other phase boundaries. Note that in general, many more sampling points are necessary in PDC.exe than COMBO.exe due to their different purposes.
In both cases, efficient sampling, which is well suited for the purpose, can be performed by COMBO.exe and PDC.exe. This means that our applications should reduce the trial and error in the materials discovery.

5. Conclusion

We explain how to use our developed MI applications, which can be executed on Windows computers. Our applications are based on machine learning and perform Bayesian optimization and uncertainty sampling on the phase diagram. The sampling results on the ternary phase diagram confirmed that our applications can efficiently sample for the intended purpose.

The important points are that neither additional settings in a Windows computer nor external parameters are required. Thus, we strongly believe that the materials scientists can achieve the benefits of machine learning without stress via our applications. Furthermore, since many experimental devices in materials science are controlled by Windows computers, our applications have a high affinity for such equipment. Therefore, we expect that our developed applications can accelerate materials discovery with the aid of machine learning.

Acknowledgments We thank Naoto Ienaga, Tsuyoshi Ueno, Trevor David Rhone, Zhufeng Hou, Tenyasu Mizoguchi, Yoshitaro Nose, Hidenori Hiramatsu, Hideo Hosono, and Yasushi Okuno for valuable discussions. This article is supported by a project subsidized by the New Energy and Industrial Technology Development Organization (NEDO), the “Materials Research by Information Integration” Initiative (MI2I) project, and Core Research for Evolutional Science and Technology (CREST) (Grant No. JPMICR17J2) from the Japan Science and Technology Agency (JST).