International Journal of Informatics Society

09/18 Vol.10 No.2 ISSN 1883-4566

Informatics Society

Aims and Scope

The purpose of this journal is to provide an open forum to publish high quality research papers in the areas of informatics and related fields to promote the exchange of research ideas, experiences and results.

Informatics is the systematic study of Information and the application of research methods to study Information systems and services. It deals primarily with human aspects of information, such as its quality and value as a resource. Informatics also referred to as Information science, studies the structure, algorithms, behavior, and interactions of natural and artificial systems that store, process, access and communicate information. It also develops its own conceptual and theoretical foundations and utilizes foundations developed in other fields. The advent of computers, its ubiquity and ease to use has led to the study of informatics that has computational, cognitive and social aspects, including study of the social impact of information technologies.

The characteristic of informatics' context is amalgamation of technologies. For creating an informatics product, it is necessary to integrate many technologies, such as mathematics, linguistics, engineering and other emerging new fields.

Guest Editor's Message

Makoto Imamura

Guest Editor of Twenty-ninth Issue of International Journal of Informatics Society

We are delighted to have the twenty-ninth issue of the International Journal of Informatics Society (IJIS) published. This issue includes selected papers from the Ninth International Workshop on Informatics (IWIN2017), which was held at Zagreb, Croatia, Sept. 3-6, 2017. The workshop was the eleventh event for the Informatics Society, and was intended to bring together researchers and practitioners to share and exchange their experiences, discuss challenges and present original ideas in all aspects of informatics and computer networks. In the workshop 33 papers were presented in eight technical sessions. The workshop was successfully finished with precious experiences provided to the participants. It highlighted the latest research results in the area of informatics and its applications that include networking, mobile ubiquitous systems, data analytics, business systems, education systems, design methodology, intelligent systems, groupware and social systems.

Each paper submitted IWIN2017 was reviewed in terms of technical content, scientific rigor, novelty, originality and quality of presentation by at least two reviewers. Through those reviews 16 papers were selected for publication candidates of IJIS Journal, and they were further reviewed as Journal papers. We have two categories of IJIS papers, Regular papers and Industrial papers, each of which were reviewed from the different points of view. This volume includes five papers among the accepted papers, which have been improved through the workshop discussion and the reviewers' comments.

We publish the journal in print as well as in an electronic form over the Internet. We hope that the issue would be of interest to many researchers as well as engineers and practitioners over the world.

Makoto Imamura He received his M.E. degree from Kyoto University of Applied Mathematics and Physics in 1986 and his Ph.D. degree from Osaka University of the Information Science and Technology in 2008. From 1986 to 2016, he worked for Mitsubishi Electric Corp. In April 2016, he has moved to the school of Information and Telecommunication Engineering at Tokai University as a Professor. His research interests include machine learning, pervasive computing, model-based design and their applications to PHM (prognostics and health management) and cyber-physical system. He won an IPSJ (Information Processing Society of Japan) Journal Best Paper Award, a JEMA (The Japan Electrical Manufactures' Association) Development Award and an IPSJ Yamashita SIG (Information Fundamentals) Research Award. He is a member of IEEE, IPSJ, IEEJ, JSME and JSAI.

Regular Paper

Maintaining Information in Differential Privacy by Using Insensitive Relationships between Personal Attributes

Takayasu Yamaguchi[†‡] and Hiroshi Yoshiura[‡]

[†]Research Laboratories, NTT DOCOMO, Inc.
[‡]Department of Informatics, University of Electro-Communications
yamaguchitaka@nttdocomo.com, yoshiura@uec.ac.jp

Abstract - Statistics generated from collections of personal data are used in both the public and private sectors, but there is a risk of the personal data being inferred from the statistics. To prevent such inference and protect the privacy of the individuals represented by the statistics, anonymization is used to modify the statistics. Among the methods for anonymization, those based on differential privacy are more promising because of their generality and rigid mathematical basis. Modification to achieve differential privacy, however, degrades the accuracy of the statistics. Because the degree of modification is proportional to the number of an individual's attributes in the statistics, the larger the number of attributes, the more degraded the statistics. On the other hand, the smaller the number, the less useful the statistics. We propose optimizing this trade-off by making use of the relationships between personal attributes. The proposed method uses only some of the attributes in the personal data while indirectly using the remaining ones by estimating their values from the values of the attributes used. The relationships between attributes are used for this estimation. However, if the relationships themselves are sensitive, i.e. they reflect personal data, additional anonymization is needed, which could degrade the effectiveness of the method. Thus, the key to effectiveness is identifying the relationships between personal attributes that are precise and insensitive. The effectiveness of our method was demonstrated by implementing it with a Laplace mechanism, a representative method for implementing differential privacy, and by evaluating the implemented system using the MovieLens 1M dataset.

Keywords: security, big data, data mining, anonymization, differential privacy

1 INTRODUCTION

Statistics generated from collections of personal data are used in both the public and private sectors, but there is a risk of the personal data being inferred from the statistics. To prevent such inference and protect the privacy of the individuals represented by the statistics, anonymization is used to modify the statistics. Among the methods for anonymization, those based on differential privacy are more promising because of their generality and rigid mathematical basis [1]-[4]. Modification to achieve differential privacy, however, degrades the accuracy of the statistics. Because the degree of modification is proportional to the number of an individual's attributes in the statistics, the larger the number of attributes, the more degraded the statistics. On the other hand, the smaller the number, the less useful the statistics.

We propose optimizing this trade-off by making use of the relationships between personal attributes. The proposed method uses only some of the attributes in the personal data while indirectly using the remaining ones by estimating their values from the values of the attributes used. The relationships between attributes are used for this estimation. For example, given the attributes of height, weight, and gender, height and gender are used and weight is estimated using the relationships between the three attributes, thus reducing the amount of modification by a third while maintaining the usefulness of the statistics. However, if the relationships themselves are sensitive, i.e. they reflect personal data, additional anonymization is needed, which could degrade the effectiveness of the method. Thus, the key to effectiveness is identifying the relationships between personal attributes that are precise and insensitive. The effectiveness of our method was demonstrated by implementing it with a Laplace mechanism, a representative method for implementing differential privacy, and by evaluating the implemented system using the MovieLens 1M dataset.

Section 2 describes related work. Section 3 describes our strategy for reducing the number of attributes, and Section 4 describes the implementation and evaluation. Section 5 summarizes the key points and mentions future work.

2 RELATED WORK

2.1 Anonymization Methods

Anonymization is applied either to records of personal information (called microdata) or to statistics calculated from microdata. The main purpose of anonymizing microdata is to prevent linking records to specific persons and that of anonymizing statistics is to prevent inferring the original microdata from which the statistics were derived. Representative methods for anonymizing microdata are methods based of k-anonymity [5], l-diversity [6], and t-closeness [7]. A set of techniques generically called statistical disclosure control are used for anonymizing statistics [8][9]. Methods based on differential privacy [1] and probabilistic k-anonymity [10][11] are used

for both. Anonymization is implemented by modifying data, and methods for modification include perturbing, swapping, aggregating, and omitting data [12]. Thus, it is inevitable that anonymization degrades the original data and statistics. Two essential issues of anonymization are therefore security and data quality [13]. Security is related to how well the purpose of anonymization (i.e. preventing linkage and inference) can be achieved. Data quality is related to how well the information that can be extracted from the data or statistics is preserved. Because there is a trade-off between security and data quality, our goal is to improve security while maintaining the quality or to improve quality (i.e. minimize information degradation) while maintaining security.

2.2 Differential Privacy and Laplace Mechanism

Differential privacy is a criterion for achieving anonymization security that is particularly promising due to its generality and rigid mathematical basis. It is expressed as an inequality:

$$\forall D_1, \forall D_2 \in D, \forall S \subseteq Range(\mathcal{K})$$
$$\{Pr[\mathcal{K}(D_1) \in S] \leq \exp(\epsilon) \times Pr[\mathcal{K}(D_2) \in S]\} \quad (1)$$

where $\mathcal{K}$ is a mechanism for calculating and anonymizing statistics from microdata D_x, D is a domain of $\mathcal{K}$, i.e. the set of microdata treated by $\mathcal{K}$, and ϵ is a security parameter. Thus equation (1) represents that the ratio between an item being included in the statistics produced from D_1 and the item included in the statistics produced from D_2 is limited by exponential (ϵ) if D_1 and D_2 differ in only one record (i.e. differ for only one person). The smaller the ϵ, the more strict the anonymization, i.e. the greater the modification of the statistics $\mathcal{K}(D_x)$ and the more degraded the statistics.

The Laplace mechanism, a representative mechanism for implementing differential privacy, adds Laplacian noise with a scale parameter (λ) to the statistics. The scale parameter is sometimes referred to as the diversity. The larger the λ, the greater the noise and the more degraded the statistics. The Laplace mechanism achieves differential privacy by satisfying an inequality:

$$\lambda \geq \frac{\Delta f}{\epsilon} \quad (2)$$

where ϵ is the security parameter mentioned above, and Δf is the sensitivity of the statistical value against the microdata. It is the maximum change in the statistics when one record (i.e. one person) in the microdata is changed. The smaller the Δf, the better the quality of the anonymized statistics. This is reasonable considering the meaning of differential privacy mentioned above.

2.3 Issues Concerning Quality Maintenance

The loss of quality due to using the Laplace mechanism can be decreased by reducing Δf, which depends on the type of statistic. For example, Δf is proportional to $1/n$ for the average value of the microdata, where n is the number of records in the microdata. This is because the maximum change in the average value due to the change of one record is proportional to $1/n$. The Δf is also proportional to $1/n$ for the variance. On the other hand, it is independent of n for the maximum and minimum. Thus, average and variance suffer less degradation due to using the Laplace mechanism than maximum and minimum. One approach to maintaining quality is thus to reduce the target statistics to a combination of statistics that have good properties such as average and deviation [2] [14]. Though much improvement has been already done in this approach, the current differential privacy mechanisms are not practical for many applications in the trade-off between security and data quality.

By "sensitivity" we mean the maximum change in a statistic when any possible change in one record is considered in the context of any possible combination of other records [15]-[17]. Thus, another approach to reducing Δf is to ignore rare cases [18]-[20] that make Δf large. This approach, however, degrades anonymization security because anonymization does not work if rare cases occur.

The fewer the attributes of a record (i.e. of a person), the lower the sensitivity [21] [22]. For example, assume a record has three attributes: gender, height, and weight. The statistics to be calculated are the frequencies of tall men (men taller than 180 cm) and heavy men (men heavier than 80 kg). The sensitivity is 2 because a man of 190 cm and 90 kg could change into a woman of 150 cm and 50 kg[1]. If we consider only gender and height, the sensitivity is reduced to 1 because we can count only the frequency of tall men. Thus, a third approach is to consider a subset of the record attributes. However, straightforward methods in this direction simply use less information to calculate the statistics [23], which is less useful for such applications as marketing of sporting goods.

As shown in Table 1, increasing the number of attributes increases the richness of the information provided by the statistics, but it also increases the privacy risk. It results in the increased sensitivity (Δf) and more modification, which reduces the usefulness of the statistics. Reducing the number of attributes reduces the privacy risk and the amount of modification needed, but the information provided by the statistics becomes poorer, and the statistics become less useful. Thus, we need a way to generate statistics from fewer attributes that are information-rich.

Using primary component analysis (PCA) is one way to reduce the number of attributes while maintaining information quality [24]-[26]. However, PCA may reveal private informa-

Table 1: Technical requirements

	Number of attributes	
	Many	Few
Information in statistics	Rich	Poor → Rich
Danger to privacy Necessary modification	Large	Small
Usefulness	Bad	Bad → Good

[1]This example is a little bit more complex than that described in the abstract.

tion in itself, e.g. how the values of the attributes are related.

3 STRATEGY FOR REDUCING NUMBER OF ATTRIBUTES

We maintain the quality of the anonymized statistics by using the third approach mentioned above; i.e. we use a subset of attributes in the microdata to calculate the statistics. We maintain the quality of information by using a subset of the attributes and estimating the values of the other attributes by using the relationships between the used attributes and the other attributes. We derive the relationships between the attributes from public information, thereby preventing the revelation of private information. If we cannot derive the relationships from public information only, we derive them by using statistics for the microdata such as the average and deviation, which are less sensitive than the target statistics.

Our strategy for anonymizing the statistics with less degradation comprises seven steps.

1. Calculate insensitive statistics, which suffer less from anonymization.
2. Anonymize insensitive statistics.
3. Use anonymized statistics to establish relationships between attributes.
4. Create statistics for subset of attributes without anonymization.
5. Anonymize these statistics with small noise.
6. Use relationships to estimate values of unused attributes.
7. Calculate target statistics from used attributes and values of unused attributes.

We explain our strategy by using the example shown in Fig. 1. The microdata are shown at the top-left. The target statistics are the frequencies of tall men and heavy men which are represented by a frequency table. A conventional method to obtain an anonymized frequency table is to generate a frequency table and anonymize it. However, the sensitivity in this case is 2 (as mentioned above), and may be too high to maintain the quality of the anonymized statistics. We thus use only values of gender and height in the microdata (at the upper-right in Fig. 1) and generate a frequency table from the reduced microdata. We anonymize this reduced frequency table, which reduces the sensitivity to 1.

We then estimate the frequency of heavy men by using the relationships between gender, height, and weight. Ideally, these relationships would be precisely obtained from public information such as statistics published by the Ministry of Health, Labour and Welfare. Otherwise, we can obtain the relationships as a combination of insensitive statistics such as average and deviation.

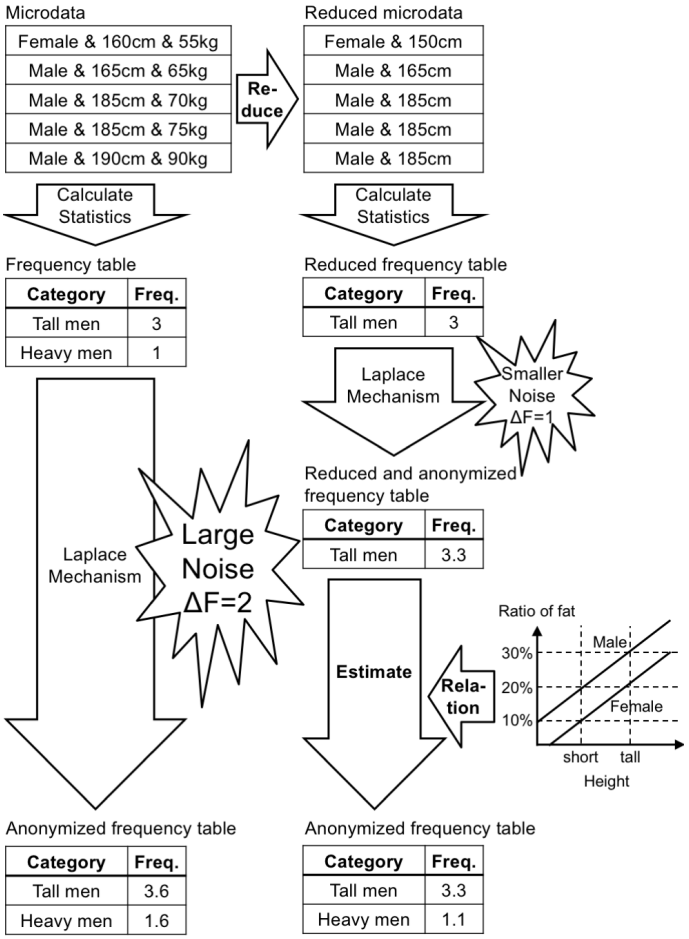


Figure 1: Example implementation of proposed strategy

4 EXAMPLE IMPLEMENTATION AND EVALUATION

Our use case is the construction of a recommendation system for movies. The system finds the top m movie categories that are best suited for the age, gender, and occupation of the user. A post-processing system selects movies from those categories [27] [28].

4.1 Dataset

We used the MovieLens 1M dataset [29] as example microdata for this system. The dataset consists of 1,000,209 records, each of which is a user's rating of a movie. It contains data for 6,040 users and 3,952 movies. Table 2 shows a part of this dataset. Note that there are 18 basic movie categories such as Action, Adventure and Comedy, and there are $2^{18} - 19$ combinational categories such as "Action & Adventure," "Comedy & Romance," and "Action & Adventure & Sci-Fi."[2] Categories combining two basic categories, e.g. "Action & Adventure," are called second-order categories, those combining three are called third-order categories, and, in general, those combining i basic categories are called ith-

[2]There are $2^{18} - 19$ combinations of basic categories except for the basic categories themselves; a null combination is excluded.

Table 2: Part of MovieLens 1M dataset

User ID	Profile	Movie (category)	Rating
1	Male, 18–24, programmer	Waterworld, (Action & Adventure)	5
1	Male, 18–24, programmer	Beverly Hills Cop, (Action & Comedy)	4
2	Female, 25–34, writer	Sabrina, (Comedy & Romance)	3
3	Male, 18–24, programmer	Star Trek, (Action & Adventure & Sci-Fi)	5
4	Female, 25–34, artist	Sound of Music, (Musical)	4

order categories. Thus, we have $2^{18}-1$ basic and combinational categories. The user ratings range from 1 to 5, with 5 being the highest.

We took the rating data as sales records, i.e. we took that a user bought a kind of categories of movie if user rated it level 4 or 5. We assume that recommender can find which categories of movie are popular for user's profiles from the dataset as top m rankings of categories. Since purchased movie categories are represented by combinations of 18 basic categories, these features could change in $2^{18}-1$ patterns and reveal privacy and, thus, should be modified.

4.2 System Design

The microdata of movie purchases are shown at the top-left in Fig. 2. These microdata are linked to user profiles and are used to generate a frequency table for each user class (e.g. "Female & 25-34 & writer"). They were anonymized, as shown at the bottom-left. The top ten categories in the anonymized table were recommended by the system with respect to user class. In the anonymization, however, the sensitivity was $2^{18}-1$ because one person could purchase all $2^{18}-1$ categories or purchase none of them.

To reduce the sensitivity, we used only 18 basic categories to generate the frequency table shown at the middle-right of Fig. 2 and anonymize it. The sensitivity was thereby reduced to 18. The top ten categories were recommended from the anonymized frequency table. However, this frequency table is information-poor because it contains only basic categories. We call this method the reduced-attributes method.

To make the reduced-attributes method information-rich, we propose using the relationships between the basic categories and the higher-order categories, which are shown in the table below the reduced and anonymized frequency table in Fig. 2. This table represents the rates of the ith-order categories appearing in the top ten; i.e. the rates of the first, second, and third-order categories are $3:5:2$. Thus, the three most frequent basic categories are included in the top ten list. Similarly, the five most frequent second-order categories are included as well as the two most frequent third-order categories. The anonymized frequency table (shown at bottom-right in Fig. 2is generated from the frequency table created by the reduced-attributes method and the table of relationships.

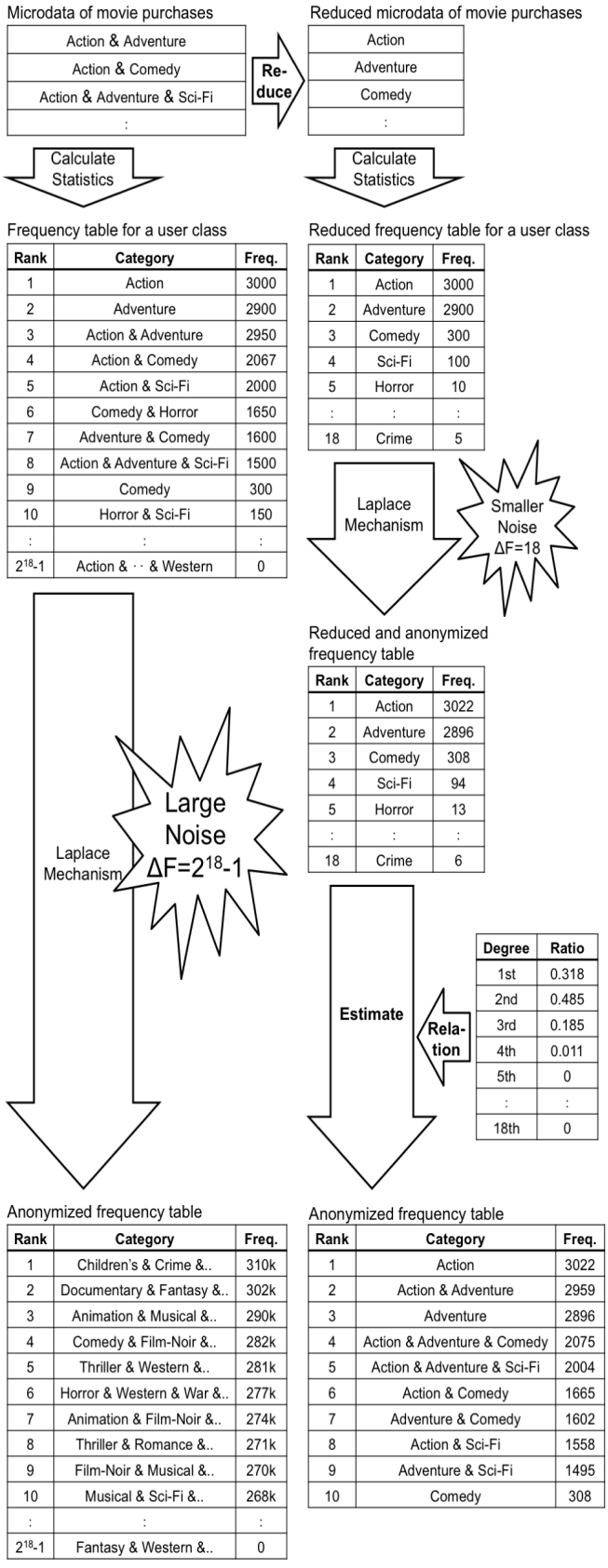


Figure 2: Overview of example implementation

To estimate the frequencies of the second- and third-order categories, we assume that the more frequent the basic

categories, the more frequent the second-order category consisting of these basic categories. We thus estimate the frequency of a second-order category by using the sum of frequencies of the basic element categories, e.g. frequency of "Action & Adventure" is (frequency of Action + frequency of Adventure) / 2. We estimate the frequency of third-order categories similarly.

The relationships between the attributes reflects the original microdata and thus should be anonymized. We use the security parameter (ϵ) for both anonymization of the frequency table and anonymization of the relationships by using $r\epsilon$ and $(1-r)\epsilon$, respectively. In our implementation, we set $r = 0.1$.

Since the proposed method uses only 18 basic categories to generate the frequency table, the sensitivity of the frequency table is 18. The sensitivity of each value in the relationship table is also 18 because we have only 18 kinds of categories, e.g. basic through the 18th category.

4.3 Evaluation

We first used our recommendation system without anonymization and obtained the top ten recommendations for each user class. We used these recommendations as correct recommendations. We then used the system with anonymization. Three anonymization methods were used. The first one was a conventional method that added Laplacian noise with a sensitivity of $2^{18} - 1$. The second one was a reduced-attributes method that used up to ith movie categories and ignored other categories. The second one was evaluated with i = 1, 2, and 3. The third one was the proposed method.

Users were classified by gender and 7 age classes (i.e. 14 classes) in the first experiment, by occupation (21 classes) in the second experiment, and by gender, age, and occupation (294 classes) in the third experiment. For each experiment, two values of ϵ were used: 2.0 for weak anonymization, 1.0 for strong anonymization.

The recommendation accuracy (in percent) was measured in terms of the number of recommendations included in both the correct recommendations and the recommendations provided

by the system. It was represented as precision at m (P@m) [30] and measured for each method and each parameter value (order i and security parameter ϵ).

Table 3: Results for using gender and age (P@10)

Required Security Level	Without Anonymization	Conventional Method	Reduced-Attributes Method (Order) 1st	2nd	3rd	Proposed Method
None	100%	100%	28%	78%	99%	48%
Weak	NG	0%	28%	51%	1%	48%
Strong	NG	0%	28%	34%	0%	48%

Table 4: Results for using gender and age (P@20)

Required Security Level	Without Anonymization	Conventional Method	Reduced-Attributes Method (Order) 1st	2nd	3rd	Proposed Method
None	100%	100%	28%	78%	99%	51%
Weak	NG	0%	28%	67%	4%	51%
Strong	NG	0%	28%	43%	1%	51%

Table 5: Results for using occupation (P@10)

Required Security Level	Without Anonymization	Conventional Method	Reduced-Attributes Method (Order) 1st	2nd	3rd	Proposed Method
None	100%	100%	30%	82%	100%	50%
Weak	NG	0%	30%	37%	2%	49%
Strong	NG	0%	30%	18%	1%	46%

Table 6: Results for using occupation (P@20)

Required Security Level	Without Anonymization	Conventional Method	Reduced-Attributes Method (Order) 1st	2nd	3rd	Proposed Method
None	100%	100%	30%	82%	100%	50%
Weak	NG	0%	30%	54%	4%	50%
Strong	NG	0%	30%	37%	3%	50%

Table 7: Results for using gender, age, and occupation (P@10)

Required Security Level	Without Anonymization	Conventional Method	Reduced-Attributes Method (Order) 1st	2nd	3rd	Proposed Method
None	100%	100%	28%	77%	93%	36%
Weak	NG	0%	26%	10%	1%	36%
Strong	NG	0%	24%	8%	1%	27%

Table 8: Results for using gender, age, and occupation (P@20)

Required Security Level	Without Anonymization	Conventional Method	Reduced-Attributes Method (Order) 1st	2nd	3rd	Proposed Method
None	100%	100%	29%	78%	96%	40%
Weak	NG	0%	28%	18%	2%	40%
Strong	NG	0%	27%	15%	2%	34%

Table 9: Detailed results for using gender and age (P@10, $\epsilon = 2.0$)

Gender	Age	Reduced-Attributes Method (Order) 1st	2nd	3rd	Proposed Method
Female	–18	30%	0%	0%	50%
	18–24	30%	20%	0%	50%
	25–34	30%	40%	10%	50%
	35–44	30%	40%	0%	50%
	45–49	30%	30%	0%	60%
	50–55	30%	0%	0%	50%
	56–	20%	0%	0%	40%
Male	–18	30%	0%	0%	40%
	18–24	30%	60%	0%	50%
	25–34	30%	70%	0%	50%
	35–44	20%	60%	0%	40%
	45–49	30%	30%	0%	50%
	50–55	30%	40%	0%	50%
	56–	30%	40%	0%	50%
Weighted Average		28%	51%	1%	48%

Table 10: Detailed results for using gender and age (P@10, $\epsilon = 1.0$)

Gender	Age	Reduced-Attributes Method (Order) 1st	2nd	3rd	Proposed Method
Female	–18	30%	0%	0%	50%
	18–24	30%	10%	0%	50%
	25–34	30%	20%	0%	50%
	35–44	30%	10%	0%	50%
	45–49	30%	10%	0%	60%
	50–55	30%	0%	0%	50%
	56–	20%	0%	0%	30%
Male	–18	30%	0%	0%	20%
	18–24	30%	40%	0%	50%
	25–34	30%	60%	0%	50%
	35–44	20%	30%	0%	40%
	45–49	30%	20%	0%	50%
	50–55	30%	10%	0%	50%
	56–	30%	30%	0%	50%
Weighted Average		28%	34%	0%	48%

Table 11: Detailed results for using occupation (P@10, $\epsilon = 2.0$)

Job	Reduced-Attributes Method (Order) 1st	2nd	3rd	Proposed Method
academic / educator	30%	40%	0%	50%
artist	30%	30%	0%	50%
clerical / admin	30%	10%	0%	50%
college / grad student	30%	70%	0%	50%
customer service	30%	0%	0%	50%
doctor / health care	30%	10%	0%	50%
executive / managerial	30%	70%	10%	50%
farmer	20%	10%	0%	20%
homemaker	20%	10%	0%	50%
K–12 student	30%	10%	0%	50%
lawyer	30%	10%	0%	50%
programmer	30%	40%	10%	50%
retiree	30%	30%	0%	50%
sales / marketing	30%	10%	0%	50%
scientist	30%	10%	0%	50%
self-employed	30%	30%	0%	50%
technician / engineer	30%	20%	0%	50%
tradesman / craftsman	30%	0%	0%	40%
unemployed	20%	10%	0%	20%
writer	30%	20%	0%	40%
other / not specified	0%	50%	0%	50%
Weighted Average	30%	37%	2%	49%

Table 12: Detailed results for using occupation (P@10, $\epsilon = 1.0$)

Job	Reduced-Attributes Method (Order) 1st	2nd	3rd	Proposed Method
academic / educator	30%	20%	0%	50%
artist	30%	20%	0%	50%
clerical / admin	30%	0%	0%	50%
college / grad student	30%	60%	0%	50%
customer service	30%	0%	0%	50%
doctor / health care	30%	10%	0%	40%
executive / managerial	30%	20%	0%	50%
farmer	20%	10%	0%	20%
homemaker	20%	0%	0%	40%
K–12 student	30%	0%	0%	50%
lawyer	30%	10%	0%	40%
programmer	30%	20%	10%	50%
retiree	30%	20%	0%	50%
sales / marketing	30%	0%	0%	50%
scientist	30%	0%	0%	40%
self-employed	30%	20%	0%	40%
technician / engineer	30%	0%	0%	50%
tradesman / craftsman	30%	0%	0%	30%
unemployed	10%	10%	0%	10%
writer	30%	0%	0%	30%
other / not specified	30%	20%	0%	50%
Weighted Average	30%	18%	1%	46%

4.4 Results

Table 3 (P@10) shows the results of the first experiment when m equalled ten. The reduced-attributes method used up to the first-, second-, and third-order movie categories. The conventional anonymization method was the best without anonymization but could not produce any correct recommendations after anonymization. The accuracy of the reduced-attributes method was lower than that of the conventional method without anonymization because it used only some attributes. It had better accuracy than the conventional method after anonymization. The reduced-attributes method was the best with weak anonymization ($\epsilon = 2.0$). The proposed method was better than the reduced-attributes method for most parameter values but worse than the second-order reduced-attributes method with weak anonymization. The advantage of the proposed method over the reduced-attributes method was larger when anony-mization was strong ($\epsilon = 1.0$).

Table 4 (P@20) shows the results of the first experiment when m equaled twenty. The results in all cases were better than those when m equaled ten (Table 3). The conventional anonymization method was the best without anonymization but the worst with anonymization. The reduced-attributes method was the best with weak anonymization ($\epsilon = 2.0$). The proposed method was the best with strong anonymization ($\epsilon = 1.0$)

Tables 5, 6, 7, and 8 show the results of the second and third experiments. Again, the conventional method could not produce any correct recommendations after anonymization. The proposed method was better than the reduced-attributes method for the case using strong anonymization ($\epsilon = 1.0$). It was better than the reduced-attributes method for the case using weak anonymization ($\epsilon = 2.0$) except for the result for using occupation (P@20) (Table 6).

Table 9 shows detailed results for the first experiment with $\epsilon = 2.0$. Accuracy is shown for each user class, anonymization method, and parameter value. The results for the conventional method are omitted because all values were 0%; i.e. none of the recommendations produced was correct. The proposed method was more accurate than the first- and third-order applications of the reduced-attributes method and partly less accurate than the second-order application. Its accuracy was stable across user classes while that of the reduced-attributes method greatly depended on the user class.

Table 10 shows detailed results for the first experiment with $\epsilon = 1.0$. The advantage of the proposed method over the reduced-attributes method was larger than that with $\epsilon = 2.0$ (Table 9).

Tables 11 and 12 show detailed results for the second experiment with $\epsilon = 2.0$ and $\epsilon = 1.0$, respectively. As in the other experiments, the advantage of the proposed method over the reduced-attributes method was larger when $\epsilon = 1.0$ than when $\epsilon = 2.0$.

4.5 Discussion

The conventional method was the best without anonymization but the worst with anonymization. Comparing Tables 3, 4, 5, 6, 7, and 8, we see that the proposed method outperformed the reduced-attributes method when strong security was required. This is because the proposed method is degraded less than the reduced-attributes method when strong security is required.

Table 13 extends Table 9 by showing the number of records and focusing on the case in which the proposed method performed worse than the reduced-attributes method. From this table, we can see that the proposed method works well when there is a small number of records.

The features of the proposed and the reduced-attributes methods are summarized in Fig. 3.

Table 13: Detailed results for using gender and age with number of records (P@10, $\epsilon = 2.0$)

Gender	Age	Number of Records	Reduced-Attributes Method (Order) 1st	2nd	3rd	Proposed Method
Female	–18	5,337	30%	0%	0%	50%
	18–24	23,910	30%	20%	0%	50%
	25–34	53,537	30%	40%	10%	50%
	35–44	29,785	30%	40%	0%	50%
	45–49	14,677	30%	30%	0%	60%
	50–55	11,835	30%	0%	0%	50%
	56–	6,498	20%	0%	0%	40%
Male	–18	10,273	30%	0%	0%	40%
	18–24	**76,889**	30%	**60%**	0%	**50%**
	25–34	**169,017**	30%	**70%**	0%	**50%**
	35–44	**86,908**	20%	**60%**	0%	**40%**
	45–49	34,799	30%	30%	0%	50%
	50–55	33,249	30%	40%	0%	50%
	56–	18,567	30%	40%	0%	50%
Weighted Average			28%	51%	1%	48%

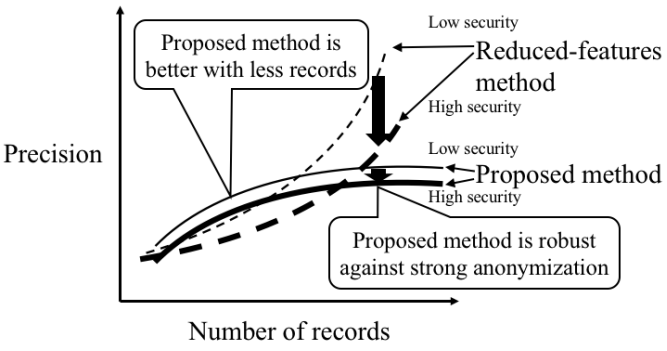


Figure 3: Features of proposed and reduced-attributes methods

5 CONCLUSION

The more personal features used in statistics, the strong anonymization needed, making statistics useless, while less personal features makes statistics information-poor. We have proposed a strategy for implementing differential privacy to

reduce Laplacian noise and maintain the quality of anonymized statistics. A subset of attributes in the microdata is used to generate statistics, reducing noise to add on the statistics with reduced attributes, and restoring information in the statistics by using relationships between the selected attributes and the other attributes. Our method uses less features to make information-poor statistics and uses knowledge to make it information-rich. Our method has been implemented for differential privacy and evaluated with MovieLens 1M dataset, demonstrating its advantage when security requirement is strong and the number of records is small.

The dataset used in this paper has data categories containing a small number of basic categories, which does not hold generally. Future work thus includes extending the proposed method to cope with different kinds of datasets that have different kinds of categories or no categories.

REFERENCES

[1] C. Dwork, Differential Privacy, ICALP, Vol. 4052, pp. 1–12 (2006).

[2] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating Noise to Sensitivity in Private Data Analysis," TCC, pp. 265–284 (2006).

[3] C. Dwork and A. Roth, The Algorithmic Foundations of Differential Privacy, Foundations and Trends in Theoretical Computer Science Vol. 9, No. 3–4, pp. 211–407 (2014).

[4] K. Nissim, S. Raskhodnikova, and A. Smith, "Smooth Sensitivity and Sampling in Private Data Analysis," STOC, pp. 75–84 (2007).

[5] L. Sweeney, "K-anonymity: A Model for Protecting Privacy," IJUFKS, Vol. 10, No. 5, pp. 557–570 (2002).

[6] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "l-Diversity: Privacy Beyond k-Anonymity," TKDD, Vol. 1, No. 1 (2007).

[7] L. Ninghui, L. Tiancheng, and V. Suresh, "t-Closeness; Privacy Beyond k-Anonymity and l-Diversity," ICDE, pp. 106–115 (2007).

[8] C. Skinner, "Statistical Disclosure Control for Survey Data," Handbook of Statistics, Vol. 29, pp. 381–396 (2009).

[9] A. Hundepool, J. Domingo-Ferrer, L. Franconi, S. Giessing, E. S. Nordholt, K. Spicer, and P.-P. de Wolf, Statistical Disclosure Control, A John Wiley & Sons (2012).

[10] J. Soria-Comas and J. Domingo-Ferrer, "Probabilistic k-Anonymity through Microaggregation and Data Swapping," ICFS, pp. 1–8 (2012).

[11] D. Ikarashi, R. Kikuchi, K. Chida, and K. Takahashi, "Probabilistic k-Anonymity through Microaggregation and Data Swapping," IWSEC (2015).

[12] C. C. Aggarwal and P. S. Yu, "Privacy-Preserving Data Mining Models and Algorithms," Vol. 34, Springer (2008).

[13] D. Kifer and B. Lin, "Towards an Axiomatization of Statistical Privacy and Utility," PODS, pp. 147–158 (2010).

[14] C. Dwork and A. Smith, "Differential Privacy for Statistics: What we Know and What we Want to Learn," JPC, Vol. 1, No. 2, pp. 135–154 (2009)

[15] C. Dwork, "Differential Privacy in New Settings," SODA, pp. 174-183 (2010).

[16] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum, "Differential Privacy under Continual Observation," STOC, pp. 715–724 (2010).

[17] G. Kellaris, S. Papadopoulos, X. Xiao, and D. Papadias, "Differentially Private Event Sequences over Infinite Streams," VLDB Endowment, pp. 1155–1166 (2014).

[18] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our Data, Ourselves: Privacy via Distributed Noise Generation," EUROCRYPT, Vol. 4004, pp. 486-503 (2006).

[19] S. P. Kasiviswanathan and A. D. Smith, "A Note on Differential Privacy: Defining Resistance to Arbitrary Side Information," IACR (2008).

[20] N. Li, W. Qardaji, and D. Su, "On Sampling, Anonymization, and Differential Privacy or, K-anonymization Meets Differential Privacy," ASIACCS, pp. 32–33 (2012).

[21] G. Cormode, M. Procopiuc, D. Srivastava, and T. T. L. Tran "Differentially Private Publication of Sparse Data," ICDT (2011).

[22] N. Shlomo, L. Antal, and M. Elliot, "Measuring Disclosure Risk and Data Utility for Flexible Table Generators," JOS, Vol. 31, No. 2, pp. 305-324 (2015).

[23] F. McSherry and I. Mironov, "Differentially Private Recommender Systems, Building Privacy into the Net," SIGKDD, pp. 627–636 (2009).

[24] S. Zhou, K. Ligett, and L. Wasserman, "Differential Privacy with Compression," ISIT, pp. 2718–2722 (2009).

[25] K. Chaudhuri, A. D. Sarwate, and K. Sinha, "A Near-Optimal Algorithm for Differentially-Private Principal Components," JMLR, (2013).

[26] C. Dwork, K. Talwar, A. Thakurta, and L. Zhang, "Analyze Gauss: Optimal Bounds for Privacy-preserving Principal Component Analysis," STOC, pp. 11–20 (2014).

[27] J. B. Schafer, J. A. Konstan, and J. Riedl, "E-commerce Recommendation Applications," DMKD, Vol. 5, pp. 115-153 (2001).

[28] D. K. Agarwal and B.-C. Chen, "Statistical Methods for Recommender Systems," Cambridge Univ. Press (2016).

[29] GroupLens, "MovieLens 1M Dataset," available from <http://grouplens.org/datasets/movielens/1m> (2003).

[30] Nick Craswell, Precision at n, Encyclopedia of Database Systems, pp. .2127-2128 (2009).

(Received October 20, 2017)
(Revised April 26, 2018)

Takayasu Yamaguchi received his B.E. and M.E. degrees from the University of Electro-Communications in 1999 and 2001. He is currently a Senior Research Engineer at Research Laboratories, NTT DOCOMO, Inc. and also a student of doctoral program in Department of Informatics, the University of Electro-Communications. His research interests, these days, lie in the area of privacy preserving data mining, statistical machine learning, and big data applications. He is a member of IPSJ.

Hiroshi Yoshiura received his B.Sc. and D.Sc. degrees from the University of Tokyo, Japan in 1981 and 1997. He is currently a Professor in Department of Informatics, the University of Electro-Com-munications. Before joining UEC, he had been at Systems Development Laboratory, Hitachi, Ltd. He has been engaged in research on information security and privacy. He is a member of IEEE, IEICE, IPSJ, JSAI, and JSSM.

Industrial Paper

Evaluation of Databases for Enterprise Systems Dealing with Images

Tsukasa Kudo†, and Yuki Furukawa†

†Faculty of Informatics, Shizuoka Institute of Science and Technology, Japan
kudo.tsukasa@sist.ac.jp

Abstract -

Nowadays, since various sensors such as surveillance cameras, wearable devices are used, various large amount of data became to be stored in the databases. To manipulate such a data, NoSQL databases have been put to practical use. Especially, MongoDB provides GridFS interface, and it was shown that the performance of such a data manipulation exceeded MySQL which is the conventional relational database. Here, to apply MongoDB to the enterprise systems, it is noted that there is the problem that the join operation is not provided. However, as for the business systems dealing with a large amount of data that takes time, it is expected that the effect of using MongoDB exceeds this problem. In this study, we conduct the comparative evaluation between MongoDB and MySQL for the actual production management system, which is a type of enterprise system. As a result, we show MongoDB is superior to MySQL in the case to manipulate a large amount of data even if the join operations are performed. Furthermore, we show it is possible to construct the configuration that takes each advantage of both by using program language such as Java. In this configuration, MongoDB manipulates a large amount of data; MySQL manipulates the other data including the join operation.

Keywords: MongoDB, database, GridFS, join operation, production management system

1 INTRODUCTION

Nowadays, various devices are spreading rapidly, such as smartphones, surveillance cameras, and various wearable devices. As a result, it is becoming possible to enter various data efficiently into the systems using such an inexpensive entry device [5], [7]. Therefore, even as for the enterprise systems, it is expected that the system can be operated more efficiently by utilizing not only conventional character and numeric data but also various kinds of sensing data such as images and videos.

In order to store and manipulate such a data, various kinds of NoSQL databases have been proposed and put to practical use [15]. Among them, there is MongoDB [3] which is a kind of the document-oriented NoSQL database: it stores the data as documents of semi-structured data model expressed by JSON; particularly, it equips GridFS interface to treat the enormous data efficiently [14]. Here, these documents correspond to the records of the relational database (RDB). And, as for a large amount of data, we had also confirmed that its performance of MongoDB had been superior to MySQL, especially in the insertion [11]. However, to apply MongoDB to the enterprise systems, it is noted that there is the problem that the join operation is not provided.

So, in order to expand its application area in the enterprise systems, it is necessary to solve the problem about the above-mentioned join operation. Here, since the large amount of data manipulation performance of MongoDB is superior to MySQL as above-mentioned, it is expected that the throughput in the entire system can be enhanced by using it, even if the performance deteriorates in the join operation. However, we could not find the study that has evaluated such a performance on the premise of the actual enterprise systems.

In this study, we explore the join operation from the view of MongoDB application. Its aims are as followings. First, we show the application field where MongoDB is superior to RDB, though the join operation is used. Second, we show the program structure, by which the join operation and a large amount of data manipulation can be performed efficiently.

So, our goal in this study is to evaluate such a performance and to show the requirements to apply MongoDB to the enterprise systems. As a target system of this evaluation, we used an actual production management system, which is a type of enterprise system. It was mainly configured to execute SQL statements directly by using batch files, though the complex processing was implemented by using the stored procedure [12]. And, we attempt comparative evaluations between MySQL and MongoDB in two cases.

In the first case, we migrate the above-mentioned SQL statements directly to MongoDB's statements and show the comparative evaluation results on this migration productivity and performance. In addition, we show that the manipulation including the numerous large amount of data deteriorates the performance of MongoDB in this case. In the second case, we use the programing language Java and its database drivers and show the comparative evaluation results of the performance. And, we show that the above-mentioned deterioration can be solved and MongoDB's performance to manipulate a large amount of data is superior to MySQL in this case. Furthermore, we show it is possible to construct the configuration that takes each advantage of both databases: MongoDB manipulates a large amount of data; MySQL manipulates the other data including the join operation.

The remainder of this paper is organized as follows. Section 2 shows the related work, and Section 3 shows the abstract of the target system to clarify the precondition. Section 4 shows the correspondence of data manipulation between MySQL statements used in the system and Mongo shells, which is the data manipulation statements of MongoDB, and we show the implementation of the system by using Mongo

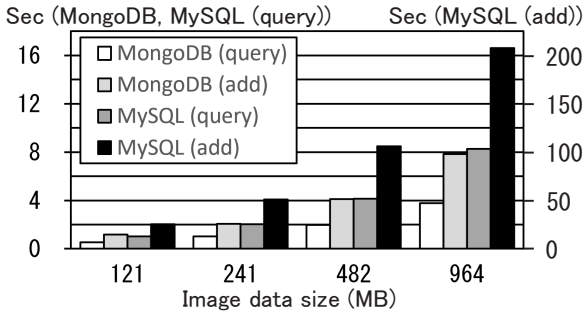


Figure 1: Evaluation result of performance comparison

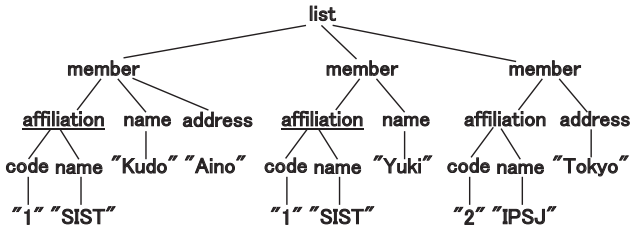


Figure 2: MongoDB structure with embedded document

shells. In Section 5, we show the results of comparative evaluations between MongoDB and MySQL in the case of direct migration from SQL statements to Mongo shells. Similarly, Section 6 shows the comparative performance evaluations in the case of utilizing programing language Java. And, we discuss these results in Section 7. Lastly, Section 8 concludes this paper.

2 RELATED WORK

The comparative evaluations between RDB and NoSQL databases have been performed on such as the data modeling, data manipulation, and performance.

Firstly, for a large amount of data manipulation, MySQL and Oracle, which are the relational database management system (RDBMS), provide BLOB data type [13]; MongoDB, which is a kind of document-oriented NoSQL database, provides GridFS interface [10]. And, it was shown that MongoDB excelled about the performance to manipulate such a data. Moreover, we have already conducted the comparative evaluations on the performance between MongoDB and MySQL for the video data, and we found MongoDB was much more efficient as shown in Fig. 1. Especially, as for the insertion, in this case, MongoDB was 25 times faster than MySQL [11].

Incidentally, there is a method of saving a large amount of data by using the file system without using the databases. However, as for this method, it is pointed out that there are some problems: the authority to access the data cannot be managed; it is difficult to perform the automatic backup such as the replication [18].

Similarly, the comparative performance evaluations of the CRUD operations (insertion, query, update, and deletion) were performed for the data that had been handled by RDB traditionally, such as text and numerical data. As a result, it was shown that the performance of MongoDB is superior to RDB [4], [6].

On the other hand, it has been pointed out that there were two problems to apply it to the enterprise systems: first, it does not maintain the ACID properties of the transaction; second, it does not equip the join operation for the plural collections which correspond to the tables of RDB [16].

As for the first problem, namely the transaction, we had shown a solution in our previous study. Here, the transaction of MongoDB can maintain the ACID properties only on the manipulation of an individual document, which corresponds to the table in RDB. So, firstly, we developed the transaction processing method, by which the ACID properties could be maintained even on the manipulation of plural documents [8]. Next, to evaluate this method, we applied it to the prototype of the actual production management system. As a result, we showed that this method could maintain the ACID properties even on the plural documents manipulation of a large amount of data in addition to the conventional character and numeric data [9].

Furthermore, we showed the application field where MongoDB's large amount of data manipulation is effective through these studies. Concretely, we had been advancing its application study for the production management system utilizing images and videos in order to improve the efficiency of the inventory management work. Here, conventionally, the inventory quantity of various kinds of parts must be counted, and it makes the workload higher. So, we had conceived the method, in which the inventory manager judges visually whether there had been the necessary inventory quantity by using the images and videos [9]. As a result, the manager could perform this business at the office based on the inventory plan calculated beforehand, instead of counting the inventory at the field.

Here, the join operation is not provided by MongoDB. Instead, it is recommended to use the data model of the non-first normal form, called As for the second problem, namely the join operation, it is recommended to use the data model of the non-first normal form, called "embedded documents". Since MongoDB is based on the semi-structured data model shown in Fig. 2, each document of the collection is able to have the individual data structure. So, for example, the attribute "affiliation" in Fig. 2, which is usually saved in the different table by the normalization in the case of RDB, can be saved in the document "member" as the embedded document. So, they can be queried without using the join operation. The comparative performance evaluation between MongoDB with this method and RDB with the join operation was performed, and it has been shown that MongoDB was superior to RDB [6].

However, with this method, it needs to have the same data in the plural documents as the embedded document, and it arises the problem like the update anomaly of RDB. For example, in the case where the name of affiliation in Fig. 2 is changed, many records must be updated. For this problem, by using a programming language, the join operation can be composed even of MongoDB [1]. That is, if the query result of one collection includes the data of foreign key to refer another collection, then it can be utilized to query another collection. Then, the joined data can be composed of both query

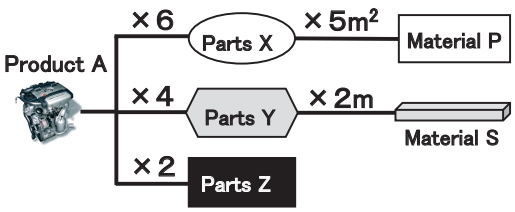


Figure 3: Structure of BOM of target system.

results. Here, in the case of using such a data manipulation, it is expected that its performance will be deteriorated.

On the other hand, as above-mentioned, the performance of the large amount data manipulation of MongoDB is so superior to RDB. So, it is expected that high performance can be obtained by applying MongoDB to the enterprise systems that manipulate a large amount of image data, even in the case where the join operations are implemented by this method.

However, we could not find the study, which evaluated the performance of the image data manipulation being accompanied by the join operation, on the premise of the actual enterprise systems.

3 TARGET SYSTEM

3.1 Target Function of Production Management System

The target enterprise system of this study is an actual production management system of some company which our laboratory is supporting. And, some of their functions have been already in operation; the others are currently under development. We use MySQL for RDBMS, and the calculation processing of each function is executed collectively by batch processing, then the results are stored into the database. We use Excel to entry the source data or to output the processing results as forms. We show the outline of the target system below.

The first is the material requirement calculation function, which has been already in operation and manages the bill of material (BOM) [17] as shown in Fig. 3. In this figure, Product A consists of 6 of part X, 4 of Y, and 5 of Z. And, parts X and Y are manufactured from $5m^2$ board material P and $2m$ stick material S respectively. As for the part Z, the commercial goods are purchased. In this way, by managing the BOM, it is possible to calculate the material cost of A based on the unit price of P, S and the price of Z.

The configuration of this processing is as follows. The data for the calculation consists of the BOM, products, parts, materials as shown in Fig. 3. And, they are stored in the tables of MySQL, and it is changed if necessary by using MySQL for Excel which is a linkage tool between MySQL and Excel. Then, the calculation processing is executed for all the data in a lump sum, and it is not necessary to specify the parameters. So, its process is described only by SQL statements, and they are executed as a batch file for Windows. Lastly, by using the view tables, the calculation results are converted to the various forms to be handled easily. Then, they are output by using the above-mentioned MySQL for Excel.

Spec_id	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
001A				1			1				1	
002A	4	5	4	2	1	4	5	3	2	2	2	
002B						2	1			3		
001A	4	2	2	3	2	7	4	4	4	4	2	1
003H	1		3	1	5	2	2					
109H	4	2	2	3	2	7	4	4	4	4	2	1

Figure 4: Monthly total of each spec

		Mar.								Apr.			
		22	23	24	27	28	29	30	31	1	3	4	5
Order_id	Spec_id	Wed	Thu	Fri	Mon	Tue	Wed	Thu	Fri	Sat	Mon	Tue	Wed
AP000001-01	001A	▼	●					■			▲		
AP000002-01	002A		▼	●					■			▲	
AP000003-01	003H		▼	●					■			▲	
AP000002-02	002A			▼	●					■			▲
AP000004-01	104A			▼	●					■			▲

Remarks) ▼: Manufactured; ●: Start to prepare shipment; ■: Delivery; ▲: Used

Figure 5: Manufacturing schedule

The second is the production planning function, and the plan is made based on received or expected receipt orders. The contents of the order are designated by the specification (spec) sheet composed of each product type and its quantity, which has spec identifier (ID). And, this system targets the common products specified by the spec ID. That is, though the order includes the custom ordered products which are individually specified in each order, they are not administered by this system. We show the sample of the output documents in Fig. 4 and 5. Figure 4 shows the monthly total number of each spec, and it is used to grasp the long-term order status. Figure 5 shows the monthly work plan which is made per order, and it is used to grasp the daily milestone. We are currently conducting the operational test of this function and preparing the necessary data.

In this function, since the parameter must be specified such as the target month, we composed this in Excel and Excel VBA (Visual Basic for Excel applications). Concretely, parameters are entered from Excel sheet, then by Excel VBA, SQL statements are created and the corresponding batch file is started to execute these statements. Lastly, its results are processed to output forms by Excel VBA similar to the first. Moreover, we composed some MySQL data manipulations by the stored procedure or stored function [12]. For example, to make the production planning, we must estimate the number of business days excluding holidays. So, it is necessary to ensure that the calendar to show the number of each business day from the beginning of the year. And, since such processing includes the iterative processing, it cannot be done by only the simple SQL statements.

The third is the inventory management function that saves the status of each product shelf as the images and videos. Figure 6 shows the inventory images, and these are stored in the database. The aim of this function is to provide the inventory image with its necessary quantity information to the manager at the office to confirm the satisfaction of its inventory. So, it is necessary that the manager can set the inventory status based on the confirmation results, and the images of the spec-

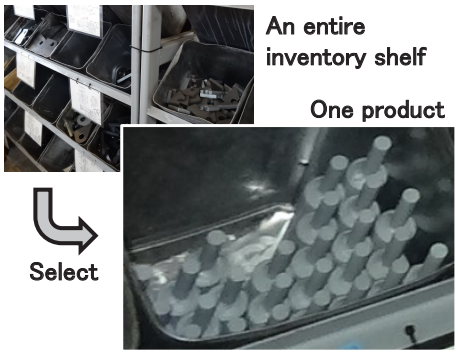


Figure 6: Inventory management utilizing images

ified products must be queried based on the manufacturing schedule shown in Fig. 5. To introduce this function, we are currently conducting the evaluation of its prototype.

Incidentally, the actual system is composed of various functions besides the above: it calculates the MRP (Material Requirement Planning) [17], which is the necessary quantity of the parts and materials, by the linkage of the production plan and BOM; the various management documents are output, and so on. However, since the data manipulation patterns are covered by the above-mentioned cases, we conduct the comparative study on only those in this study.

3.2 Database Structure and Data Manipulations

Figure 7 shows the ER diagram of the target system; below, we indicate the table name and attribute name in the ER diagram in italic. In addition, we show only the main tables and attributes for the sake of simplicity. (1) of Fig. 7 corresponds to the material requirement calculation function, and parts ($parts$) and products ($product$) are associated with BOM (BOM). And, the following price is set to the unit price ($price\_unit$) of parts: $price\_unit$ of $part\_price$ is set in the case where the part is purchased; price per 1kg ($price\_kg$) of $material$ is set in the case where the parts are manufactured from the material associated with material ID (mat_id).

(2) of Fig. 7 corresponds to the production planning function, and the data of $calendar$ such as the number of business days is calculated by the holiday information ($holiday$); the order and product are associated with specification composed of $specification$ and $spec$. The schedule data shown in Fig. 5 is calculated and saved into $manufacture\_plan$, in which each milestone date is included: manufacturing completion date (m_date_num), start date to prepare shipment (c_date_num), delivery date (d_date_num) and used date at the ordering company (u_date_num).

(3) of Fig. 7 corresponds to the inventory management function, and saves the inventory status of products as images and videos. $stock\_shelf$ indicates product shelf, in which the products are stored. And, $stock$ shows the status of the inventory: the image or video name (doc_name) saved in $image$; the correspondence data between them and product shelf is saved in $stock(p\_id, doc\_name)$; the inventory quantity ($quantity$) of $stock$ is set if necessary. Here, since the images and videos are captured at any time, their capture time (chk_time) is included in the primary key of $stock$, and the relationship between $stock$ and $stock\_shelf$ becomes many-to-one. Firstly, the image and video data is saved from the camera into the folder of the PC, then saved into $image$; And, the necessary data for this processing is downloaded from the database to the work folder of the PC when necessary.

We extracted the data manipulation patterns of these tables from the functions mentioned in Section 3.1, and got patterns shown below. Incidentally, the basic CRUD data manipulations of the single table are excluded.

(a) Join operation: in (1), each part is joined with the products which use the part respectively, and the results are saved into $parts\_cost$. So, the join operation between BOM and $parts$ is performed.

(b) Iterative operation: in (2), to set the number of business day ($date\_num$) of $calendar$, the division of the business day and holiday are set to the column $holiday$ of $calendar$ firstly. Then, the numbers of business date are set sequentially from January 1st, that is, it constitutes the iterative operation. Incidentally, it is implemented by the stored procedure in MySQL.

(c) Grouped aggregation operation: in (1), the product of material cost and quantity, which are expressed by $cost$ and $p\_quantity$ of $parts\_cost$ and calculated in (a), are aggregated for each product, and stored into $producrt\_price$.

(d) Selection of record with the self-join operation: in (1), since $part\_price$ has a history on the estimated date (est_ymd), the record having the max estimate date must be queried for each $part\_id$. In the SQL statement, this is expressed by the subquery with the self-join operation as shown in Fig. 8.

(e) Images and videos operation: in (3), the images and videos of the inventory shelves are stored into $image$, so these data must be inserted and queried. In MySQL, this is executed by "load_file" function to insert, and "select into dumpfile" statement to query.

4 IMPLEMENTATION OF DATA MANIPULATION USING MONGODB

4.1 Implementation Policy

In MongoDB, the Mongo shell is provided for methods for the CRUD operations and the interactive data manipulations which have JavaScript interface. And, similar to the SQL statements, JavaScript files can be executed as the batch file, or as a function like the stored procedure in SQL. In this study, we implemented the Mongo shell as a batch file on Windows as shown in Fig. 9. In Fig. 9, "JSfile.js" is the JavaScript file including the Mongo shell methods; and, it is executed by inputting to "mongo" command; then the execution results are output to "out.csv" file by a print statement of JavaScript.

Below, we show the implementation of each operation mentioned in Section 3.2. Incidentally, we describe only the main

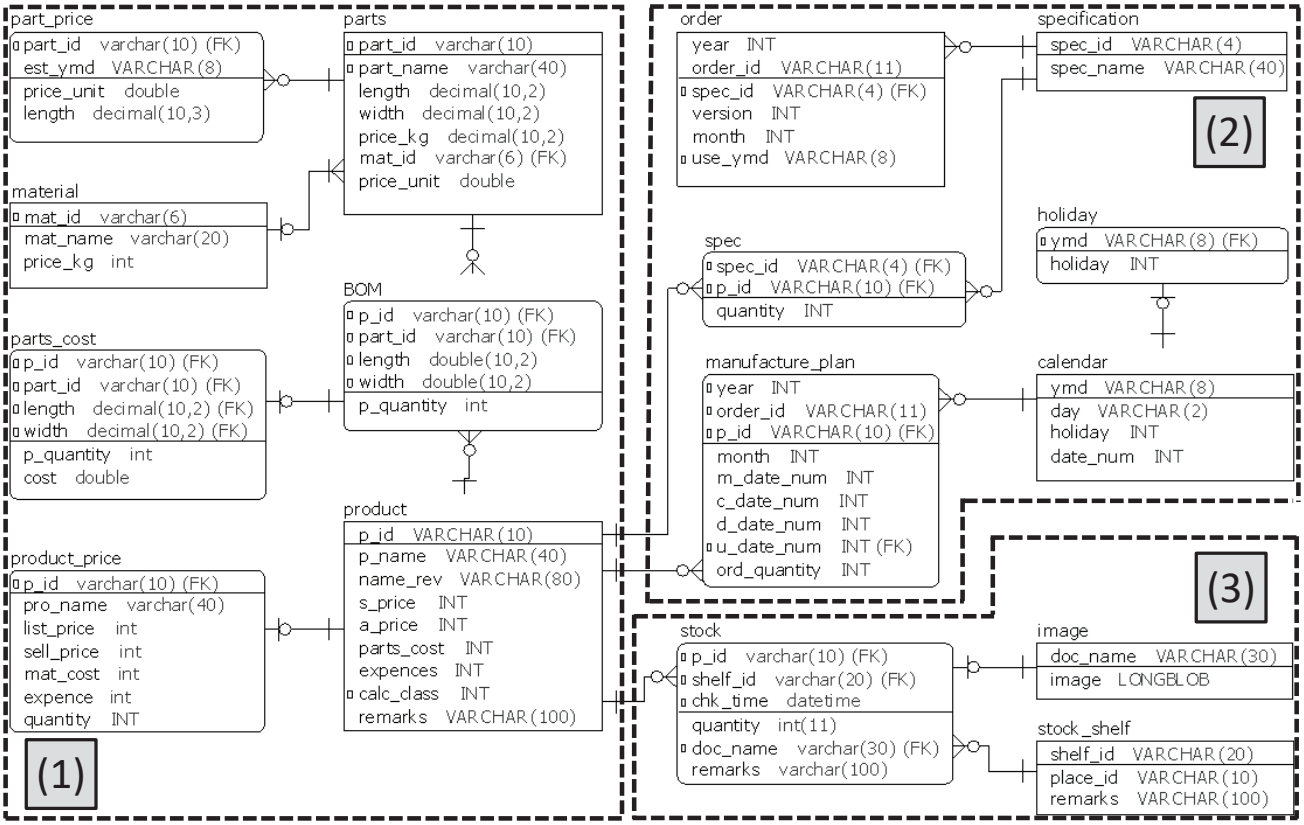


Figure 7: ER diagram of database

```
SELECT * FROM part_price AS a
  WHERE a.est_ymd = (SELECT
    MAX(b.est_ymd) FROM part_price AS b
    WHERE a.part_id = b.part_id);
```

Figure 8: Max value query by self-join

```
> mongo < JSfile.js > out.csv
```

Figure 9: Batch file of Mongo shell

attributes and operations for the sake of simplicity. In the actual system, related attributes and operations are added to the following logic.

4.2 Implementation of Join and Iterative Operation

In the Mongo shell, the join operation is not provided. So, the collections were joined as follows: firstly, we copied the collection corresponding to "many" of many-to-one into the temporary new collection; then, we added the fields of the collection corresponding to "one" to the above collection. In this way, we could create the result collection of the join operation.

In Fig. 10, we show the case of (a) in Section 3.2, in which $parts$ and BOM are joined to create $parts\_cost$. In (1) of Fig. 10, $parts\_cost$ is created by copying BOM, then $cost$

```
// (1) copy BOM to parts_cost
db.BOM.copyTo("parts_cost");
// (2) update parts_cost to join parts
var partRec;  // variable (document of parts)
var part=db.parts.find();  // (3) find method
while(part.hasNext()){
  partRec=part.next();   // get next document
  db.parts_cost.update(  // (4) update method
    {zairyou_id:partRec.zairyou_id,…
     haba:partRec.width}, // query condition
    {$set:{cost:part.price_unit}}, // set cost
    {multi:true});   // update multi documents
}
```

Figure 10: Join operation procedure of MongoDB

field of $parts\_cost$ is set to $price\_unit$ field of $parts$ in (2) as follows. Firstly, by using find method in (3), which corresponds to select statement of SQL, all the documents of $parts$ are queried. Here, documents are sequentially set to $partRec$ as same as the cursor operation of SQL. Next, update method at (4), which corresponds to update statement of SQL, updates the value of $cost$ attribute of all the documents that match the query condition shown by the first parenthesis "{ }" which expresses the pair as of "{field name:field value}". Here, the query condition is such that all these attribute values equal to the specified attribute values. In addition, if $parts\_cost$ collection does not have $cost$ field, then it is inserted.

```
// (1) total cost per p_id
Var mat_cost=db.parts_cost.aggregate
  ({$group:{_id:{p_id:"$p_id"},
   cost:{$sum:"$cost"}}});

// (2) document with latest date
var part_price=db.part_price.aggregate
  ({$match:{part_id:"P0001"},
   {$group:{_id:"$part_id",
      latest:{$max:{est_ymd:"$est_ymd"}}}}});
var partRec=part_price.next();
var laRec=db.part_price.findOne
  ({part_id:"P0001",est_ymd:partRec.latest});
```

Figure 11: Aggregation method of MongoDB

Incidentally, in the case where the join operation is performed for only some of documents matching the specified query condition, only the target documents are inserted at (1) in Fig. 10. For this operation, insert method corresponding to insert statement of SQL is used.

Next, the iterative statement, which is while statement and so on, can be used in the Mongo shell. So, we implemented the iterative operation to create $calendar$ shown in (b) of Section 3.2 by using these statements, like the stored procedure in SQL.

4.3 Implementation of Aggregation and Self-Join Operation

Mongo shell provides the aggregate method, which corresponds to the aggregation operator and group by clause of SQL. So, as for the aggregation operation of material cost for each part in (c) of Section 3.2, it can be executed by this method. In this method, as shown in (1) of Fig. 11, *$group* expression shows the fields to be aggregated, and *$sum* expression shows the aggregation method of summation like the SQL statement. Incidentally, the aggregation results can be got by the cursor operation like Fig. 10.

Similarly, as shown in (2) of Fig. 11, the selection operation of record having the max value shown in (d) of Section 3.2 can be performed by the aggregate method, and the latest estimated date was queried from $part\_price$ in this case. Here, since MongoDB does not provide the join operation, we configured the operation to query the target document again from $part\_price$ using the queried estimated date and $part\_id$ value. In Fig. 11, *$match* expression in aggregate method specifies the query condition. Also, findOne method queries only the single document, and in this figure, it queries as of the query condition that $parts\_id$ is "P0001" and est_ymd is the queried estimated date.

4.4 Image and Video Data Manipulation

The upper limit of the document size of MongoDB is 16 MB, and the GridFS interface is provided for data exceeding this limit. Using this interface, the image and video data is saved into GridFS collection divided from other attributes.

```
REM (1) insert operation of image from file
mongofiles -d iwin2017
    put 2017_5_A-1-3-1.JPG -l A-1-3-1.JPG

REM (2) query operation of image into file
mongofiles -d iwin2017
    get 2017_5_A1-1-1.JPG -l A-1-3-1.JPG
```

Figure 12: Image insertion and query command

Table 1: Comparison of CRUD operation

MySQL	MongoDB	Class
SELECT	find(), findOne()	CRUD
INSERT	insert()	
UPDATE	update()	
DELETE	remove()	
(*Join operation*)		(a)
	[many].copyTo()	
JOIN syntax	[one].find() (use cursor)	
	[many].update()	
Stored procedure	JavaScript	(b)
Stored function	JavaScript	
Group BY clause	aggregate()	(c)
(*Query record with max value)*		(d)
self-JOIN operation	aggregate()	
+ subquery	findOne()	
(*Image and video operation*)		(e)
INSERT	MONGOFILES	
+ LOAD_FILE()	command (put)	
SELECT INTO	MONGOFILES	
DUMPFILE	command (get)	

And, since the data insertion and query are performed by utilizing mongofiles command, not the Mongo shell, we configured to perform this command in batch files which are separated from the JavaScript files.

We show the examples of these commands in Fig. 12. Here, "-d" indicates the database, and "-l" indicates the file name on the disk. That is, we can save the image data into the database with the different name from the name as of disk file. Incidentally, since this command is a utility executed in Windows command line, connection and disconnection with the database is performed at each its execution.

Finally, in Table 1, we show the summary of the implementation method comparison between MySQL and MongoDB. Here, the column "Class" indicates the classification of these data manipulations. "CRUD" shows the basic data manipulation, and others indicate the number in Section 3.2.

5 IMPLEMENTATION OF SYSTEM AND COMPARATIVE EVALUATIONS

In order to demonstrate the target production management system can be constructed by using MongoDB, we implemented the principal part of this system by using MongoDB

according to the correspondence of CRUD operation shown in Table 1. Then, we conducted the comparative evaluations of the program volume and execution performance between MongoDB and MySQL.

5.1 Implementation Using MongoDB

First, as for the material requirement calculation function shown in (1) of Fig. 7, we implemented the process to create $product\_price$. Here, cost data of $parts\_price$ and $material$ is reflected into $parts$, then $product\_price$ is created from $parts$ via $parts\_price$. We implemented this processing using the Mongo shell as the batch file shown in Fig. 9.

Second, as for the production planning function shown in (2) of Fig. 7, we implemented the following processing: one creates $calendar$ from $holiday$; the other makes the csv files for the aggregation and schedule document shown in Fig. 4 and Fig. 5 respectively. We implemented this processing using the above-mentioned batch file, and we embedded the parameters in the JavaScript program directly without linking with Excel for the sake of simplicity.

Third, as for the inventory management functions shown in (3) of Fig. 7, we implemented the following two processing shown in Fig. 12. Incidentally, these implementation methods are same as MySQL except the execution command as shown below.

One is the processing to save the pictures and videos of the product shelves into $image$, and to insert the correspondence data between $stock\_shelf$ and $image$ into $stock$, that is, this creates the correspondence between the shelves and the images or videos. To save the images and videos data, their file name in the camera must be grasped in the insertion program. So, we implemented this processing using Excel VBA to make the insertion batch file, in which insertion is executed by mongofiles command. And, we implemented the correspondence data insertion program by using the Mongo shell, which is executed by the batch file.

The other is processing to query the image and video data. Similar to above, we implemented this processing to be executed by mongofiles command, which was made by using Excel VBA based on the given query condition: specified shelves, or the product shipment date and so on.

5.2 Comparative Evaluations of Program Volume

In order to perform comparative evaluations of productivity between MongoDB and MySQL, we counted the number of source lines of the programs respectively. Table 2 shows these results, and (1) shows the material calculation; (2) shows the production planning function. Incidentally, since the user interface programs of the inventory management system operations were made by using Excel VBA as mentioned in Section 5.1, and they were common to both databases. So, we omitted their evaluation. Here, (2) was divided into the three processings: Plan(C) shows the creation processing of $manufacture\_plan$ in (2) of Fig. 7; Plan(O) shows the processing to output the query results into csv files for the forms shown in Fig. 4 and Fig. 5; Plan(P) shows the output processing of these forms from the csv files. Moreover, since the statement other than the SQL statement and Mongo shell is necessary, we show their individual volume in this table.

Table 2: Comparison of program volume (line)

	MySQL			MongoDB		
	SQL	Else	Total	Shell	Else	Total
(1) Material	15	0	15	24	29	53
(2) Plan(C)	8	64	72	12	91	103
(2) Plan(O)	11	7	18	8	24	32
Total	34	71	105	44	144	188
(2) Plan(P)	0	180	180	0	180	180

(1): Material requirement calculation function
(2): Production planning function
C: create data; O:output to csv file; P: print document

The function indicated by (1) in Table 2 could be configured only by the SQL statements in MySQL. However, in MongoDB, it was necessary to use JavaScript in addition to the Mongo shell. In this case, the number of source lines of the latter was about 3.5 times that of the former. On the contrary, the processing indicated by Plan(C) and Plan(O) in (2) could not be described only by SQL statements in MySQL, so it had to be described with the stored procedures and stored functions; MongoDB was the same as MySQL. In this case, the former number of source lines was about 2 times that of the latter.

The processing indicated by Plan(P) in (2) is the printing of a form, and there was no database access. So, this processing was common to MySQL and MongoDB. That is, the processing to output the form consists of data extraction from the database indicated by Plan(O) and printing as of Plan(P). In this case, the ratio of Plan(P) was 91% in MySQL and 85% in MongoDB.

5.3 Comparative Evaluations of Elapsed Time

We executed each processing mentioned in Section 3.1 on the standalone PC environment to evaluate the elapsed time comparatively. Here, we modified each processing to execute only the database access including the data format operations as the batch file, that is, the following processings were excluded: defining the parameters, printing of forms and so on. The execution environment is as follows. CPU is i7-6700 (3.41GHz); memory is 16GB; the disk is SSD memory of 512GB; OS is Windows 10. We adopted MySQL (Ver. 5.7.12), MongoDB (Ver. 3.4.3) for the database.

We show the evaluation results in Table 3. “Material” shows the elapsed time of the material calculation shown by (1) of Table 2, and it includes the manipulation (a), (c) and (d) mentioned in Section 3.2. The elapsed time of MongoDB was approximately 11 times that of MySQL. “Calendar” and “Plan” are parts of the production planning function: the former creates $calendar$, and as for MySQL, it was executed by stored procedure with the iterative manipulation shown in (b) of Section 3.2; similarly, the latter created $manufacture\_plan$ by

Table 3: Comparison of elapsed time (second)

Processing	MySQL	MongDB	Ratio	Manipulate
Material	1.07	11.90	11.1	(a), (c), (d)
Calendar	9.58	5.10	0.5	(b)
Plan	0.32	26.84	83.0	(a) (3-join)
Image(in)	23.48	260.78	11.1	(e)
Image(out)	0.24	77.76	324.0	(e)

Remarks: Ratio=MongoDB/MySQL

SQL statement to join the 3 tables, $calendar$, $order$ and $spec$. The elapsed time of MongoDB was about 0.5 and 83 times that of MySQL respectively. "Image(in)" shows the case to insert the image data of actual product shelves into database from the disk: the number of images is 340, and the size of each image is from 0.9 MB to 2.9 MB; "Image(out)" shows the opposite case to the previous case, that is, the same data is queried from the database to store into the disk as files.

As a result, in this case, the elapsed time of MongoDB was about 11 and 324 times that of MySQL respectively. That is, MongoDB was degraded despite being more efficient than MySQL in manipulating a large amount of data. The reason for this was as follows. Firstly, since the mongofiles command must be executed for each file as the Windows command individually, the connect operation occurs for each file insertion and database query. So, the delay occurred in this process. On the contrary, MySQL could execute all the data manipulations by connecting once similar to the other operations.

By the way, we separated the fields of the images and videos from the inventory table ($inventory$) and composed the individual table ($image$) as shown in Fig. 7 even in MySQL. This was due to the results of the preliminary study: in the case of gathering all these fields to one table, the extreme delay occurred. That is, to confirm the inventory, firstly we saved the inventory image shot in the order of the shelf ID to this table; then, updated shelf ID ($shelf\_id$) according to the image order. However, this update operation took more than 20 seconds for the above-mentioned 340 data, and it was too long for the operations. Here, it was pointed out that the instances of LONGBLOB should not in the query results if it was not really necessary [13]. However, as a result of this preliminary study, we found that the extreme latency occurred not only in this case but also in the case where the images and video column was not included in the data manipulation.

6 PERFORMANCE EVALUATION USING JAVA

6.1 Implementation Using Program Language

As shown in "Image(in)" and "Image(out)" of Table 3, the performance to manipulate the image data of MongoDB, in this case, is inferior to RDB, though the performance to manipulate the single large amount of video data is better than RDB as shown in Fig. 1. Here, as mentioned in Section 4.4, the connection to, and disconnection from the database are performed for each execution of mongofiles command, and a large number of image data manipulations were performed by this command in this evaluation. As a result, this performance deterioration occurred.

```
// (1) Insert statements of MongoDB
InputStream st = new FileInputStream(new File("fName"));
ObjectId fileId = gridFSBucket.uploadFromStream("doc_name", st);

// (2) Select statements of MongoDB
FileOutputStream st = new FileOutputStream("fName");
gridFSBucket.downloadToStream(fileId, st);

// (3) Insert SQL statement of MySQL
insert into image values ('doc_name', load_file('fName'));

// (4) Select SQL statement of MySQL
select image into dumpfile 'fName' from image
    where doc_name = 'doc_name';
```

Figure 13: Image data manipulation statements

On the other hand, for example, MongoDB Java driver provides GridFSBucket class for GridFS interface, and the multiple image data manipulation can be performed after connecting once. Therefore, in order to prevent the performance degradation shown in Table 3, we implemented this manipulation by using Java. And, we also performed the comparative performance evaluation between MongoDB and MySQL. Here, as for MySQL, this manipulation was implemented by using Java, too.

As for the implementation environment, we used Java Platform Standard edition 8, MongoDB Java driver Ver.3.5.0 and MySQL Connector/J Ver.5.1.1.41. Other environments were the same as in Section 5.3. And, the implementation target was the inventory management function using images shown in (3) of Fig. 7.

Firstly, we evaluated the basic function, that is, the individual performance of the join processing and image manipulation. Next, we evaluated the combination processing, that is, the performance of the case where both of the join operation and image data manipulation are performed. Furthermore, we also evaluated the combination structure of MySQL for the join operation and MongoDB for the image data manipulation. Hereinafter, we indicate this structure by "mix" structure, and its detail is as follows.

The manipulations of a large amount of data such as images are performed by the streaming in both of MongoDB and MySQL. Figure 13 shows the examples of the statement of image data manipulations. Here, "fName" shows the image file name; "doc_name" shows the image name in the database. (1) and (2) show the insert and select statements of MongoDB to manipulate image respectively: In each first line, the streaming of Java is defined; in each second line, upload and download of the image is executed by using GridFSBucket class respectively. Incidentally, "ObjectID" is the identifier of the document in MongoDB, and it can be queried by using the image name "doc_name" in advance the select statement (2). On the other hand, as for SQL statement of MySQL, the image file name to be inserted is specified by using "load_file" function; the destination image file name of the image data to be queried is specified by using "dumpfile" clause.

Therefore, as for the implementation of the image data manipulation in Java, following composition is possible: firstly, the target image name "doc_name" is obtained and saved into the variable of type String; then, the statements to manipulate the image in Fig. 13 are executed. Moreover, in this composition, the image name can be queried by using MySQL and the image data manipulation can be executed by using MongoDB. In this way, by using this mix construction, it was expected that the superior operations of each database could be combined.

6.2 Evaluations of Basic Functions

To evaluate the basic functions, we implemented the following 4 cases by using both of MySQL and MongoDB.

(A) Join operation on three tables (3-Join): this queries the data of three tables that matches the designated query condition of p_id of $product$ by joining these three tables in Fig. 7: $product$, $stock$ and $stock\_shelf$.

(B) Self-join operation (Shelf-Join): this queries only the latest data of $stock$ that matches the designated query condition of p_id. That is, only records having the latest chk_time are queried for the pair $\{p\_id, shelf\_id\}$. Here, it is composed of the subquery with the self-join operation in RDB.

(C) Image insertion: this inserts all the image data existing in the designated folder into the database.

(D) Image download: this downloads all the image data existing in the designated table of the database into the designated folder.

As for (A), though we implemented it by cursor operation in both databases, their structure was different. That is since MySQL has the SQL statement of the join operation, it is possible to query the join results as a cursor. On the other hand, MongoDB has no statement of the join operation. So, we implemented the following processing: firstly, we queried the target data of $product$ including p_id; then, we queried the target data of $stock$ including $shelf\_id$ by this p_id; lastly, we queried the target data of $stock\_shelf$ by this $shelf\_id$, and joined all the query results.

As for (B), we implemented the query by using the subquery and self-join operation shown in Fig. 8 in RDB. On the other hand, as shown in Table 1, MongoDB had the aggregate statement corresponding to the group by clause in RDB. So, we queried the max value of chk_time of $stock$ for the pair p_id, $shelf\_id$, then queried the target data by using these data.

As for (C) and (D), we implemented by the similar structure in both databases. For (C), we queried all the image data of the designated folder and inserted them sequentially into the table. On the contrary, for (D), we queried the image data sequentially by utilizing the cursor and saved into the designated folder. Here, in MySQL, we implemented by utilizing the insert and select statements of SQL. And, in MongoDB, though we implemented by utilizing GridFS interface, the database connection could be maintained by using Java as above-mentioned.

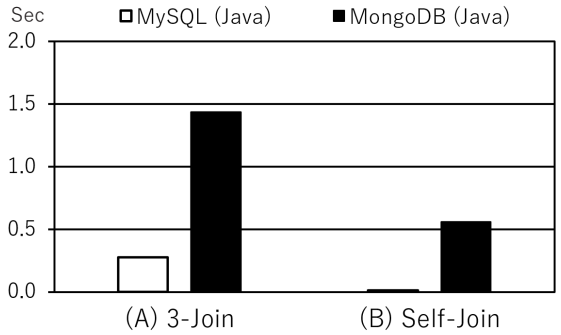


Figure 14: Elapsed time of join operation

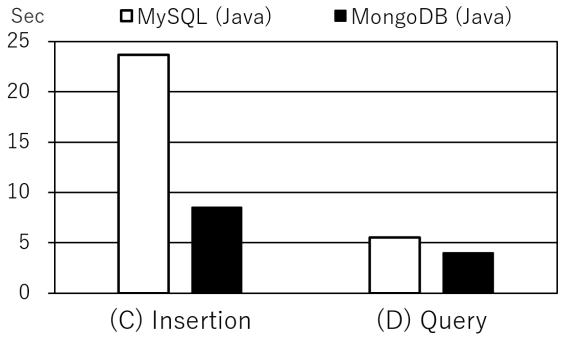


Figure 15: Elapsed time of image manipulation

In Fig. 14, we show the evaluation results of (A) and (B). The number of data of the three tables was 1063, 2000 and 845 respectively, and the numbers of result data of (A) and (B) were 1,770 and 339. As shown in Fig. 14, the elapsed time of MySQL was 0.276 seconds in (A), which is 5 times faster than 1.435 seconds of MongoDB; the one of MySQL was 0.017 seconds in (B), which is 33 times faster than 0.561 seconds of MongoDB.

And, in Fig. 15, we show the evaluation results of (C) and (D). Contrary to the previous results, the elapsed time of MongoDB was 18 times and 9 times faster than the one of MySQL respectively in these cases. The elapsed time of both was 8.5 and 23.7 seconds in (C) respectively, and 4.0 and 5.5 seconds in (D).

6.3 Evaluations of Combination Processing

Next, we performed the comparative performance evaluations for the practical processing by combining the above-mentioned operations. That is, we implemented and evaluated the following processing: firstly, we queried the target image name (doc_name) by the join operation in (A) or (B); then we inserted these image data into the database in (C); lastly, we queried these image data from the database and saved into another folder in (D). Here, we implemented three cases: the first was implemented by only MySQL; in the second, (A) and (B) were implemented by MySQL, and (C) and (D) were implemented by MongoDB; the third was implemented by only MongoDB. Here, (A), (B) and so on is shown in Section 6.2, and hereinafter, it is same.

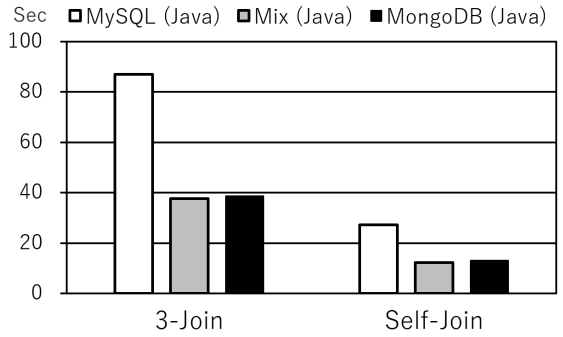


Figure 16: Elapsed time of combination processing

Table 4: Breakdown time of combination processing (second)

Processing	Construct	Join	Insertion	Query
3-Join	MySQL	0.285	68.967	17.658
	Mix	0.280	25.819	11.506
	MongoDB	1.461	25.807	11.419
Self-join	MySQL	0.015	21.758	5.531
	Mix	0.021	8.304	3.894
	MongoDB	0.998	8.196	3.911

We show the evaluation results in Fig. 16, and the breakdown of the elapsed time of each operation in Table 4. Here, since the query results of (A) and (B) were same as those of Section 6.2, the number of image data that was manipulated was also 1,770 and 339 respectively. And, each elapsed time is indicated as follows: "Join" indicates the join operation; "Insertion" indicates the image data insertion; "Query" shows the image data query. As shown in Table 4, since the time to manipulate image data was longer than the time of joining operation, the elapsed time as of MongoDB was shorter than MySQL. Especially, a large difference was observed in the elapsed time of the image insertion.

In addition, as for the mix structure, which is shown by "Mix" in Table 4, it was constructed by using MySQL's join operation and MongoDB's image data manipulation. So, for example, the elapsed time of the join operation is similar to MySQL; the one of image manipulation is the same as MongoDB. That is, we obtained the superior performance for each operation as mentioned in Section 6.1. As a result, the best performance was achieved by the mix structure.

7 DISCUSSIONS

We discuss the evaluation results. First, as for the target production management system, we found that all the functions implemented by using MySQL could be implemented by using MongoDB. As the results of the productivity comparative evaluations, though the number of MongoDB's data manipulation commands increased, the ratio of the description of data manipulation was very small in the actual systems as shown in the Table 2. Therefore, from the viewpoint of the overall system development man-hour, we consider that the importance of the selection concerning the both will be small.

Second, we found some note points to maintain the performance of a large amount of data. As shown in the last paragraph of section 5.3, it was necessary to separate such a data column to the individual table even in MySQL as same as MongoDB. On the other hand, in MongoDB, the connection to the database should be maintained as shown in Table 3 and Fig. 15, that is, in the case to access such a data many times, it should be composed by using programing language and so on.

Lastly, by using programing language, we could use both of MySQL for the join operation and MongoDB for the image data manipulation as shown in the last paragraph of Section 6.1. In the case of manipulating a large amount of data, by using MongoDB, we could obtain better performance than MySQL. However, as shown in Fig. 7, there are many processes that use no image in the enterprise system. On the contrary, they utilize the join operation. So, we currently consider that there is a solution to use both as above-mentioned.

8 CONCLUSIONS

In order to manipulate a large amount of data, the application of NoSQL database is spreading. However, to apply NoSQL databases to the enterprise systems, there is the challenge that the join operation must be implemented efficiently. In this study, we conducted the comparative evaluations between MySQL and MongoDB for the actual enterprise system in two cases: the implementation by using Mongo shells and the one by using programming language Java.

In the first case, we found the functions of general SQL statements could be implemented by using only Mongo shells, though the performance degraded in the case of manipulating many large amounts of data such as images.

In the second case, we found that the above-mentioned deterioration of performance could be solved, and the elapsed time to manipulate a large amount of data was longer than the one of the join operation. That is, better performance was obtained at the whole data manipulations by using MongoDB. Furthermore, we showed it was possible to construct the configuration that took each advantage of both databases: MongoDB manipulated a large amount of data; MySQL manipulated the other data including the join operation.

For the future challenge, we will expand the application area of MongoDB by using sharding and improving the data structure.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 15K00161.

REFERENCES

[1] R. Arora, and R. R. Aggarwal, "Modeling and querying data in MongoDB," International Journal of Scientific and Engineering Research, Vol. 4, No. 7, pp. 141–144 (2013).

[2] M. Bach, and A. Werner, "Document-Oriented Data Stores of Vision Objects," Proc. of Innovative Control

Systems for Tracked Vehicle Platforms, pp. 163–174 (2014).
[3] K. Banker, “MongoDB in Action,” Manning Pubns Co. (2011).
[4] A. Boicea, F. Radulescu, and L.I. Agapin,“MongoDB vs Oracle–database comparison,” Proc. of Third International Conference on Emerging Intelligent Data and Web Technologies, pp. 330–335 (2012).
[5] M. Chen, S. Mao, and Y. Liu, “Big data: A survey,” Mobile Networks and Applications, Vol. 19, No. 2, pp. 171–209 (2014).
[6] C. Győrödi, R. Győrödi, G. Pecherle, and A. Olah, “A comparative study: MongoDB vs. MySQL,” 13th International Conference on EMES, pp. 1–6 (2015).
[7] S. Hiremath, G. Yang, and K. Mankodiya, “Wearable Internet of Things: Concept, architectural components and promises for person-centered healthcare,” EAI 4th International Conference on Wireless Mobile Communication and Healthcare, pp. pp. 304–307 (2014).
[8] T. Kudo, M. Ishino, K. Saotome, and N. Kataoka, “A Proposal of Transaction Processing Method for MongoDB,” Procedia Computer Science, Vol 96, pp. 801–810 (2016).
[9] T. Kudo, Y. Ito, and Y. Serizawa, “An Application of MongoDB to Enterprise System Manipulating Enormous Data,” Proceedings of International Workshop on Informatics (IWIN2016), pp. 277–284 (2016).
[10] MongoDB, Inc., “Welcome to the MongoDB Docs,” https://docs.mongodb.com/ (reffered Oct. 16, 2017).
[11] K. Nagasawa, and T. Kudo, “Development of Mobile Quotation System Utilizing Tablet and MongoDB,” Proc. of 2017 IEICE General conference, D-9-23, p. 113 (2017) (In Japanese).
[12] Oracle Corp., “Chapter 23 Stored Programs and Views,” https://dev.mysql.com/doc/refman/5.7/en/stored-programs-views.html (reffered June 6, 2017).
[13] Oracle Corp., “11.4.3 The BLOB and TEXT Types,” https://dev.mysql.com/doc/refman/5.7/en/blob.html (reffered June 6, 2017).
[14] D.R. Rebecca, and I. E. Shanthi, “A NoSQL Solution to efficient storage and retrieval of Medical Images,” International Journal of Scientific & Engineering Research, Vol. 7, No. 2, pp. 545–549 (2016).
[15] E. Redmond, and J.R. Wilson, “Seven Databases in Seven Weeks: A guide to Modern Databases and the NoSQL Movement,” Pragmatic Bookshelf (2012).
[16] K. Seguin, “The Little MongoDB Book” (2011), http://openmymind.net/mongodb.pdf (reffered May 5, 2017).
[17] S. P. Singh, “Production and Operation Management,” Vikas Publishing House Pvt Ltd (2014).
[18] M.P. Stevic, P., B. Milosavljevic, and B.R. Perisic, “Enhancing the management of unstructured data in e-learning systems using MongoDB,” Program, Vol. 49, No. 1, pp. 91–114 (2015).

(Received October 20, 2017)
(Revised December 26, 2017)

Tsukasa Kudo received the B.S. and M.E. from Hokkaido University in 1978 and 1980, and the Dr.Eng. from Shizuoka University in 2008. In 1980, he joined Mitsubishi Electric Corp. He was a researcher of parallel computer architecture and engineer of business information systems. Since 2010, he is a professor of Shizuoka Institute of Science and Technology. Now, his research interests include database application and software engineering. He is a member of IEIEC and IPSJ.

Yuki Furukawa is currently working toward a B.I. degree at Shizuoka Institute of Science and Technology. Her research interests include production management system and comparative evaluations between MySQL and MongoDB.

Regular Paper

Effective Derivation of a Mapping of Variables in a Loop Structure

Kozo Okano†, Shinji Kusumoto‡, and Yukihiro Sasaki‡
†Faculty of Engineering, Shinshu University, Japan
‡Graduate School of Information Science and Technology, Osaka University, Japan
okano@cs.shibshu-u.ac.jp, kusumoto@ist.osaka-u.ac.jp

Abstract - Static program analysis enables us to analyze a program without performing an actual execution run, but the analysis of loops is, however, difficult in general. In order to solve this problem, one of existing techniques derives a mapping between variables using regression analysis on data obtained by multiple executions of a program (run history). The technique is a kind of a hybrid approach and when we analyze a complicated loop using the technique, it may derive an incorrect mapping. Our new proposed technique overcomes this problem using recurrence relations. It first obtains a run history and then performs regression analysis on loop iterations and variables based on the run history. It finally derives a recurrence relation on the variables occurring in the loop body. Experiments confirm that it can derive useful mappings that we cannot derive by the existing technique.

Keywords: loops, static analysis, recurrence relation, run history, mapping

1 INTRODUCTION

In software engineering, especially in the maintenance phase of software, engineers need to understand software by reading or analyzing code. In such situations, program analysis methods can be helpful. Program analysis methods will produce an abstract summary of the behavior of a given fragment of code, usually a function, or a method (in object-oriented programming language). The abstract summary is usually in the form of a formal specification such as Java Modelling Language [1].

Program analysis methods are divided into two categories: (1) static program analysis types and (2) dynamic program analysis types. Static program analysis methods do not need to execute the target program while dynamic program analysis methods will.

Static program analysis methods use many concrete methods such as symbolic execution [2] and model checking [3]. Recently other approaches have emerged. *e.g.*, heuristic methods [4], and automatic predicate abstraction based methods, such as SLAM [5], BLAST [6]. Static program analysis methods using logic sometimes utilize SAT/SMT solvers [7]. SAT/SMT solvers are enhanced SAT solvers with background theories. A SAT solver is a simple solver for satisfiability problems on logical expressions over propositional variables (boolean variables). Some examples of background theories include decision problems on integer expressions and expressions over arrays and tuples (record types). Thus, SAT/SMT solvers can solve decision problems on programs. There are many SAT/SMT solvers including popular solvers are such as Z3 [8] and Yices [9].

For dynamic program analysis, Daikon [10] is a well-known tool. It derives program assertions from data obtained from execution logs of the target programs. The execution is usually performed many times in order to infer accurate assertions. Recently approaches based on regression analysis [11] have been proposed [12]. Le [12] proposed a method that derives a mapping from a family (or a set of sets) of variables to a set of variables before and after a target loop structure using regression analysis. It first executes the target loop multiple times with varying input values. Based on the data obtained by the execution, it then performs regression analysis. The analysis infers a mapping between variables before and after a target loop.

Therefore, it can easily analyze programs with loop structures. However, it can deal with only linear and quadratic mappings. Consequently, it cannot infer an exponent mapping which represents Fibonacci sequences.

Our proposed approach overcomes this problem as follows. First, we perform dynamic program analysis, and then we obtain data on the number n of loop iterations and the program variables. Next, we perform regression analysis on the data and obtain a relation between n and the program variables. Then we construct a recurrence formula on the program variables by static analysis on the loop body. The recurrence formula represents a relation between the program variables for the $n+1$-th and n-th loops. We can obtain their closed-form solution of the recurrence formula, which represents the program variables for n. By combining the closed-form solution and the results of the regression analysis, we obtain a final mapping representing the mapping between variables before and after a target loop.

The remainder of this paper is organized as follows. Section 2 presents disadvantages of the existing methods as preliminaries. Section 3 gives the proposed method. Sections 4 and 5 show experimental results and discussion, respectively. Finally, Section 6 summarizes this paper.

2 PRELIMINARIES

In general, static analysis approaches sometimes have trouble handling loop structures. In model checking, we overcome this problem by using several techniques such as loop unwinding and Craig interpolation for the approximation of loop invariants. In general, such techniques are not omnipotent due to memory limitations and calculating complexity. For this reason, some of the existing methods contrive sev-

eral methods, such as bounded unwinding [13], user-specified time-out mechanisms [14], and so on. In [15], the S2E tool utilizes Path Selection function which enables us to control the termination of a loop with multiple criteria. For example, PathKiller can stop loop iteration up to a user specified number. In another approach, Xie *et al.* [16] proposed a method which returns an *unknown* value for a variable that cannot be analyzed. Le [12] proposed a method based on regression analysis. It can derive a mapping between variables before and after a given loop structure, it claims that the method can derive more accurate mapping than others.

2.1 Regression Analysis

Regression analysis is an estimating method for the relationships among variables. It includes many techniques for modeling and analyzing several variables, when the focus is on the relationship between a dependent variable and one or more independent variables.

Regression analysis is usually based on a regression model. Regression models involve the following parameters and variables:

- The unknown parameters, denoted as β, which may represent scalars or vectors.
- The independent variables, $\mathbf{X}$.
- The dependent variables, $\mathbf{Y}$.

A regression model relates $\mathbf{Y}$ to a function of $\mathbf{X}$ and β.
$\mathbf{Y} \approx f(\mathbf{X}, \beta)$

In a formal manner, the approximation is typically formalized as $E(\mathbf{Y} \mid \mathbf{X}) = f(X, \beta)$. To carry out regression analysis, the form of the function f must be specified in advance.

2.2 Segmented Symbolic Analysis (SSE)

The approach in [12] (SSE for short) uses approximation functions (regression models) shown in Table 1.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \beta_4 x_1^2 + \beta_5 x_2^2$$
(x_1 and $x_2 are$ the input and the output, respectively)

There are many cases in which the functions in Table 1 are not applicable. For example, the program for generating Fibonacci numbers in Listing 1 cannot apply the functions in Table 1.

Table 1: Transformation Table for Loops

Model	example	values for $\beta_0 \sim \beta_5$
Constant	$y = 0$	0
Simple Linear	$y = x_1$	0 except for β_1
Multiple Linear	$y = 2 \cdot x_1 + x_2$	$\beta_4 = 0, \beta_5 = 0$
Polynomial Linear	$y = x_1^2 + x_1 \cdot x_2$ if $x_1 > 0$	
Piece-wise Linear	then $y = x_1$ else $y = 3$	cannot be expressed

Listing 1: Program for Fibonacci Numbers

```
public class TestFibo {
  public int cf(int n) {
    int current = 0;
    int prev = 1;
    int prevprev = 0;
    if(n > 0){
      for(int i = 0; i < n; i++) {
        current = prev + prevprev;
        System.out.print(current + " ");
        prevprev = prev;
        prev = current;
      }
      return current;
    } else {
      System.err.println(
        "Input is less than 1");
      return -1;
    }
  }
}
```

In general, it is hard to apply regression analysis on Fibonacci numbers because its closed-form solution is exponential to the number of loop iterations and it cannot be represented by the functions in Table 1.

SSE requires preparing many input patterns to output execution logs, which contain a large number of execution paths. Insufficient execution paths lessen the accuracy of the approximation. This is another disadvantage of the approach. In other words, the approach is not useful for a program with many branches in its loop structures, which makes the coverage of the test cases low.

3 OUR PROPOSED METHOD

In this section, we describe the differences between the existing method presented in [12] and our proposed method.

Our proposed method derives a mapping between variables before and after the target loop. The approach can easily deal with loop structures.

SSE uses mapping via regression analysis only. It loses precision in the approximation. Our proposed method uses regression analysis only to derive a mapping between arguments (of the given Java method) and the number of iterations of the target loop (of the given Java method). A relation between n, the number of iterations of the target loop, and the variables of the loop, is obtained by static analysis used in cooperation with an analysis tool for mathematics like Mathematica. This combination produces approximation with high accuracy.

3.1 Outline of Our Proposed Method

Here, we give an outline of our proposed method.

The inputs and the outputs of our method are summarized as follows.

- Input: a Java method with a single loop structure

- Output: an approximation of the loop structure in the method, if successfully generated, otherwise a failure is returned.

Note that for a method with multiple loop structures, we can divide the method into several methods where each method has a single loop structure.

For example in Listing 2, a method with multiple loop structure can be translated into several methods where each has only a single loop structure as shown in Listing 3.

In order to convert a nested loop structure into a simple loop structure, we use a new counter vpc which indicates which loop is selected.

Listing 2: Multiple Loops Structure

```
public class MultipleLoops {
  public int ex(int n) {
    int local = 0;
    for (int i=0; i< n; i++)
      local += i;
    int x = local;
    int y = local*2;
    for (int j=0; j < n; j ++)
      for (int k=0; k < n; k ++) {
        x *= 3 + k;
        y += x + j;
      }
    return x + y;
  }
}
```

Listing 3: Method with a Single Loop Structure

```
public class MultipleLoops {

  public int ex1(int n) {
    int local = 0;
    for (int i=0; i< n; i++)
      local += i;
    return local;
  }

  public int ex2(int loc, int n) {
    int x = loc;
    int y = loc*2;
    int i = 0;
    byte vpc = 1;
    while (vpc < 2){
      switch (vpc) {
        case 1;
          if (j < n) {
            vpc = 2;
            j ++;
            k = 0;
          } else {
            vpc = 3;
          }
          break;
        case 2;
          if (k < n) {
            x *= 3 + k;
            y += x + j;
            k ++;
          } else {
            vpc = 1;
          }
          break;
      }
    }
    return x + y;
  }

  public int ex(int n) {
    int tmp = ex1(n);
    return ex2(tmp, n);
  }
}
```

Thus, for a method with a multiple loops structure, our proposed method is also applicable. For nested loops, it is known that such loops can be translated into a single loop. Therefore, in principle, this method is also applicable.

We limit the class of the input Java method as follows because we will use SAT/SMT solvers in later analysis stages. The allowed types are bool, byte, short, int, float, double, and arrays of them. The proposed method cannot deal with String. For control structures, it allows for the use of `if`, `for`, `while` statements.

Through of the section, we use the following variables:
$\boldsymbol{x}$: arguments of the given (target) method
$\boldsymbol{y}$: variables which a user want to analyze
n: the number of iterations of the loop

Figure 1 shows the architecture of a tool proto-type of our proposed method.

The procedure is summarized as follows.

Step 1: Add proper print statements to the target source code in order to store execution logs on $\boldsymbol{y}$ and others.

Step 2: Execute the program with varying $\boldsymbol{x}$ and obtain a sufficient number of execution logs.

Step 3: Analyze the logs and obtain the execution paths, relations between $\boldsymbol{x}$ and the execution paths, and a record on n.

Step 4: Using regression analysis, infer the relation between $\boldsymbol{x}$ and n.

Step 5: As a recurrence relation, obtain a relation between $\boldsymbol{y}_s$ at the entry point and $\boldsymbol{y}_e$ at the exit point of the loop body.

Step 6: Solve the closed-form of the recurrence relation obtained in **Step 5**.

Step 7: Finally, calculate an expression representing a relation between $\boldsymbol{y}$, $\boldsymbol{x}$, and n by integrating **Step 3**, **Step 4**, and **Step 6**.

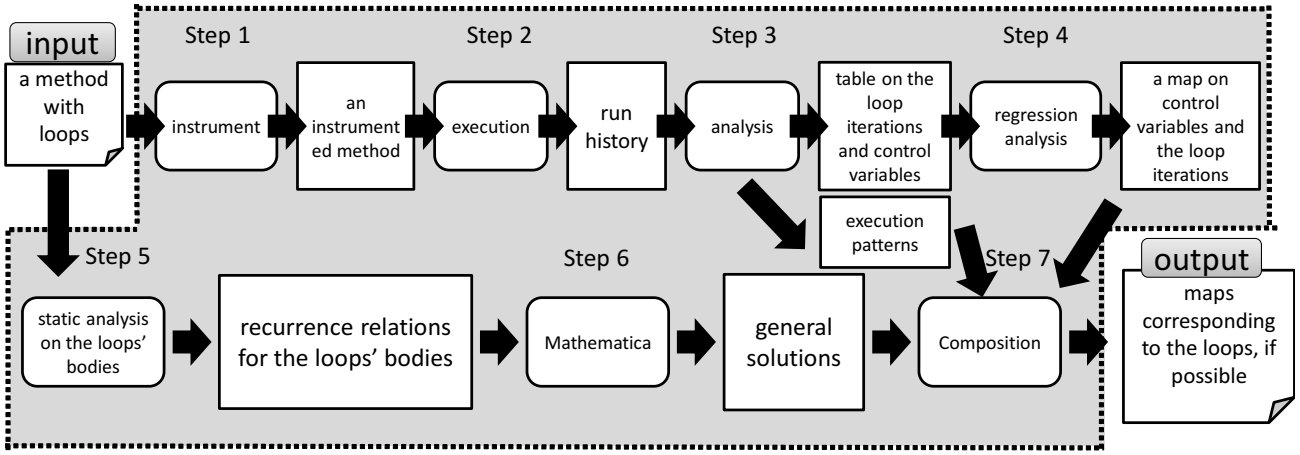


Figure 1: Tool overview

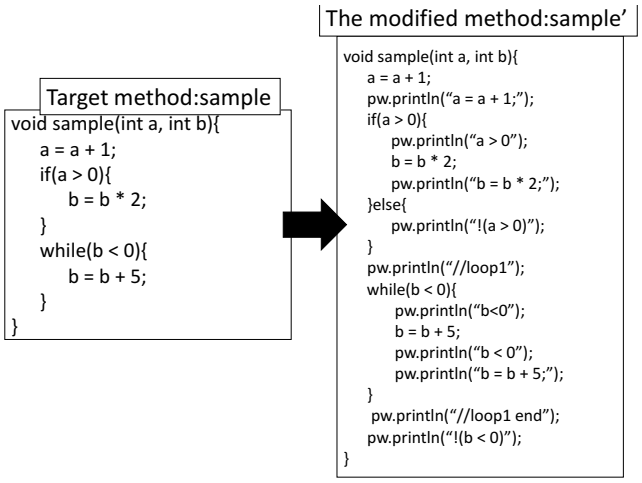


Figure 2: Print statement instrument for obtaining execution logs

In the following subsections, we will explain **Steps 1, 3, 4, 5, 6, and 7** which are important steps of our proposed method.

3.2 Step 1

In order to store execution logs, we add print statements to the target source code (Figure 2).

This step is similar to the Instrument step of Daikon [10], a famous tool for detecting invariants of programs.

In Fig. 2, `pw` is an instance of `PrintWriter` class. `pw` uses its `println` method to output log to a text file. The statement is inserted after assignment statement of the original code. For a control statement, such as an if-statement, it is also inserted to output the information on the condition of the path. For example it will output "$a > 0$" for the path in which the condition holds. For a loop structure, it outputs start and end markers as shown in Fig. 2, as well as the information on the condition.

JDT [17] is used to implement such an instrument.

3.3 Execution paths

Many paths are considered with regard to the conditions in the loop structure. Thus, we have to enumerate every pattern of the paths.

In general, a path in a loop structure is defined as a sequence of sentences executed for a given a concrete set of values of variables in the loop structure.

We call any path enumerated "an execution path (in the loop)." In general, the number of "if-statements" is i, and then there are 2^i execution paths at most.

3.4 Step 3-1

Here, we obtain a relation between $\boldsymbol{x}$ of the given method and the statements in the loop body.

The execution logs contain records on $\boldsymbol{x}$ and the number of occurrences of the execution path.

We explain this more precisely using the example in Fig. 3.

Let us assume that the upper left code is the target method. The method contains two execution paths as shown in Fig. 3.

We can see from the figure, that if the argument i is 25, then Execution Path2 (EXP2) occurs twice and EXP1 occurs three times.

Figure 3 shows only three tuples, but, we actually obtain these tuples with more than 100 executions varying the values of i. The same value 100 is used in the existing method. Many studies including [18] have proposed how to generate efficient values of the inputs (arguments).

3.5 Step 3-2

Next we confirm the order of the executing paths. Figure 3 shows the situation in which EXP2 is first executed twice and then EXP1 is executed three times.

Let us assume that in general executions of a loop body, each execution path occurs repeatedly and successively.

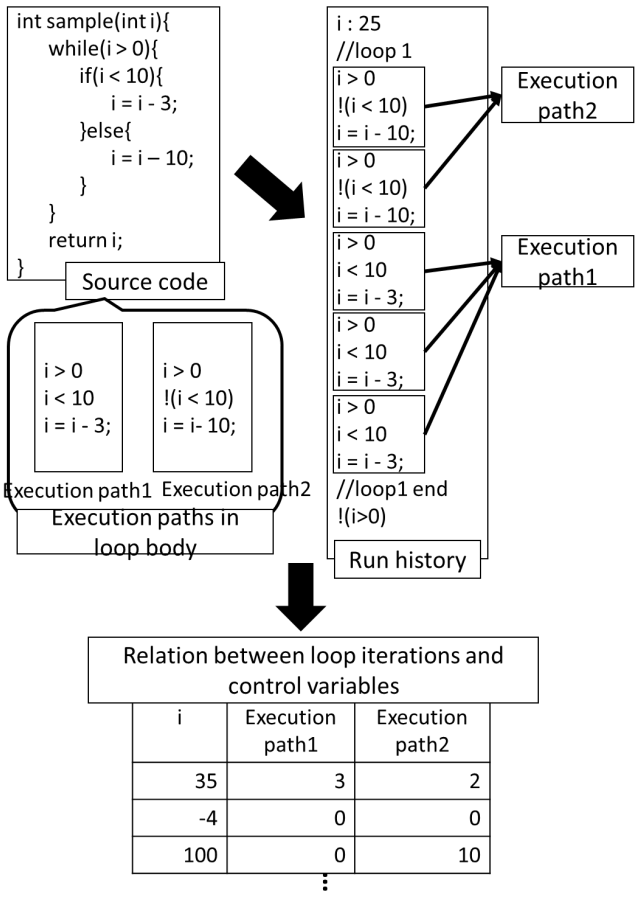


Figure 3: Analysis of Run history

> **Execution path assumption**: For any values of the variables, if an instance of an execution path occurs more than twice, then the execution paths occur successively.

Under this assumption, we can abstract this sequence as the following regular expression.

$$(EXP2)^*(EXP1)^* \tag{1}$$

At **Step 3**-2, we obtain such regular expressions on the execution patterns.

When we recall Fig. 3 we observe that in the sequence, EXP2 is first executed twice and followed by three executions of EXP1. We call such a pattern an execution pattern.

We enumerate every execution pattern from the execution logs. For each execution pattern, we abstract the constants representing the number of occurrences with the Kleene closure symbol $*$. For example, three occurrences of the execution path EXP1 is abstracted as $(EXP1)^*$.

For a loop, we can obtain a set of execution patterns.

For simplicity, hereafter we consider execution patterns in a form of $(EXP1)^*(EXP2)^* \cdots (EXPn)^*(n > 0)$. For other cases, we return failure of analysis.

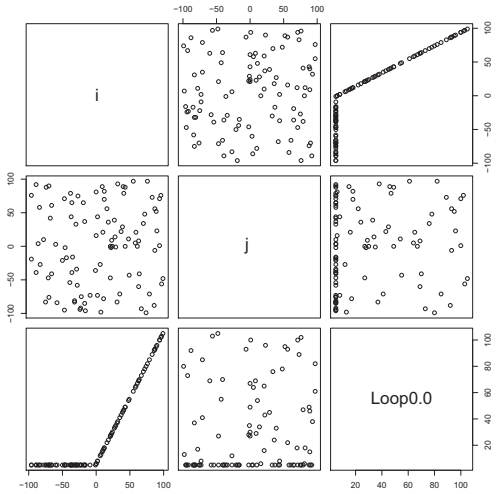


Figure 4: Relation between loop iterations and control variables

3.6 Step 4

Here, we describe how to derive a relation between $\boldsymbol{x}$ and the number of occurrences of an execution path (which is obtained at **Step 3-2**).

We use R [19], a regression analyzer.

Figure 4 shows relations between arguments (integers i and j) and an execution path named "loop." Loop0.0 stands for "loop."

A plot located in the first row, second column in Fig. 4 shows a relation between i and j.

In a similar way, plots in the first row, third column, and in the second row, third column show relations between i and the number of executions of the "loop," and between j and the number of executions of the "loop," respectively.

The plot in the first row, second column in Fig. 4 indicates that there is no correlation between i and j due to their randomness.

Additionally, we find that there is no correlation between j and the number of loop executions. However, there is a strong correlation between i and the number of loop executions. For the case where i is negative, the number of executions of "loop" becomes 0. For the case on $i > 0$, the number of executions of "loop" becomes i.

Such a relation can be obtained using the regression analysis for each execution path.

For example, let i and j be arguments.

Let $a_0, a_1, a_2 \cdots$ be coefficients.

The following model (expression) can be used in regression analysis.

$$n = a_0 + a_1 i + a_2 j + a_3 ij + a_4 i^2 + a_5 j^2 \tag{2}$$

For Fig. 4, we obtain a result in which a_1 equals 1 and the other coefficients are 0. Thus, we obtain a relation $n = i$.

Note that n is the number of iterations. Thus, it does not have a negative value. We assume that $n = 0$ when $n < 0$ for later analysis steps.

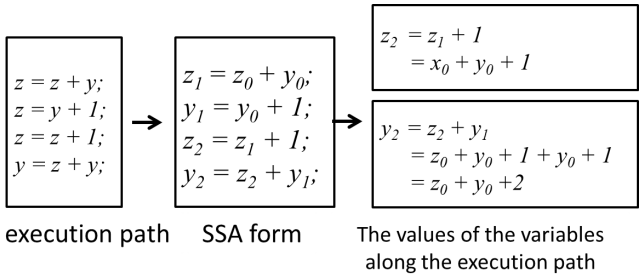


Figure 5: Conversion of Execution int SSA form

For the case of failure of regression analysis, we return analysis failure.

3.7 Step 5

Here, for each execution path i, we derive a relation between variables $\boldsymbol{y}_0^i$ and $\boldsymbol{y}_k^i$, where $\boldsymbol{y}$ is a vector of variables appearing in the execution path i. The suffixes are the same as the SSA form explained bellow.

First, a series of assignment statements of the execution path into SSA (static Single Assignment) form [20]. In SSA form every variable can appear in at most one assignment. In order to satisfy this condition, an original variable is, in general, divided into several variables when it is involved in several assignment statements. For such a case, the divided variables are distinguished by their own suffixes.

For example, an execution path can be translated into the SSA form shown in Fig. 5. The execution path uses variable z and y. Their corresponding variables for the first assignment, are represented as z_0 and y_0. At line 1, $z+y$ is assigned for z. In such a case, variables z_0 and z_1 are used. In a similar way, at line 3, z_2 is used. The suffixes play a role to distinguish variable z at different positions.

Next, using the SSA forms, we derive a recurrence relations on the variables.

In Fig. 5, the first values of the variable z and y are represented as z_0 and y_0. The final values are represented as variables z_2 and y_2. Using the SSA form, we can infer that z_2 equals $z_1 + 1$ and z_1 equals $z_0 + y_0$. Thus, z_2 and y_2 equal z_0+y_0+1 and z_2+y_1, respectively. Because y_1 equals y_0+1, we infer that y_2 equals $z_0 + y_0 + 2$.

The obtained equations can be represented as the following recurrence relations:

$$z[n+1] = z[n] + y[n] + 1,\ y[n+1] = z[n] + y[n] + 2 \quad (3)$$

Here, $z[0]$ and $y[0]$ stand for the seed values, i.e., z_0 and y_0 for the variables z and y. Symbols $z[n]$ and $y[n]$ stand for the general term of z and y obtained by repeating n-times of the SSA form.

3.8 Step 6

Here, we solve the recurrence relation.

For example, we can obtain the following recurrence relation from an SSA form in Fig. 5.

$$z[n+1] = z[n] + y[n] + 1,\ y[n+1] = z[n] + y[n] + 2 \quad (4)$$

Let us assume that the seed values of z and y are z_0 and y_0, respectively. We can obtain a closed-form solution for the recurrence relation using Mathematica[21], as follows.

$$z[n] = \frac{1}{2}(-4 + 3 \cdot 2^n + 2^n y_0) \quad (5)$$

$$y[n] = \frac{1}{2}(-2 + 3 \cdot 2^n + 2^n z_0) \quad (6)$$

It is difficult to obtain such a complex expression using the existing method [12].

3.9 Step 7

Here we integrate the obtained analysis results in the previous steps and generate the final mapping.

Let us assume that the execution pattern is $(EXP1)^*(EXP2)^* \cdots (EXPk)^*$, and that the number of execution times of EXPi is m_i.

$\boldsymbol{y}_0^i$ stands for the initial values at the entry of EXPi

Let $\boldsymbol{y} = F_i(\boldsymbol{y}_0^i, n)$ be the mapping obtained at **Step 6**. Let $G_i(\boldsymbol{x})$ be the mapping obtained at **Step 4**, where $m_i = G_i(\boldsymbol{x})$.

Let us consider the following cases.

1. $k = 1$ holds
2. $k > 1$ and $\forall i \exists c : 0 < i \leq k, (G_i = c\boldsymbol{x}$ or $G_i = c)$ holds
3. $\forall i, j : 0 < i, j \leq k, G_i = G_j$ holds
4. otherwise

For the first three cases, we can obtain the result as follows.

For case (1), the final mapping is $\boldsymbol{y} = F_1(\boldsymbol{y}_0^1, G_1(\boldsymbol{x}))$.

For cases (2) and (3), the final mapping is $\boldsymbol{y} = F_k(\cdots F_2(F_1(\boldsymbol{y}_0^1, G_1(\boldsymbol{x})), G_2(\boldsymbol{x})), \ldots, G_k(\boldsymbol{x}))$.

For case (4), we conclude that the mapping cannot be generated and failure is returned.

We shows an example for the case (2):

Let $(EXP1)^*(EXP2)^*$ be the execution pattern.

Let us consider a situation where when EXP1 is executed n times, then the value of variable y increases by n, and if EXP2 is executed n times. Then, the value of variable y increases by $10n$.

In such a case, equations $F_1(y_0^1, n) = y_0^1 + n$ and $F_2(y_0^2, n) = y_0^2 + 10n$ holds. Additionally let us assume that when $i > 0$ EXP1 and EXP2 are executed i and 1 times, respectively; and that when $i \leq 0$ EXP1 and EXP2 are not executed.

By integrating all of the above, we can obtain the final mapping $y = y_0^1$ for $i \leq 0$, and $y = y_0^1 + x + 10$ for $i > 0$.

4 EXPERIMENTS

The setup of the experiments is summarized as follows.

- OS: Windows 7 Enterprise 64bit
- CPU: Intel Xeon E5-2609 2.40GHz $\times$ 2
- Memory: 48.0GB
- Java: JRE7
- R: version 3.0.2
- Mathematica 9.0.0
- Z3: z3-4.3.0

4.1 Overview of the experiment

The research questions are summarized as follows.

RQ1 Mapping quality: Is the obtained mapping accurate?

RQ2 SAT/SMT applicability: Is the obtained mapping applicable to SAT/SMT solvers?

RQ3 Range of Capability: Can more types of mapping be obtained from as compared to the existing method?

We use Z3 for criteria for RQ2

Table 2 summarizes the target programs and each program contains loop structures.

4.2 Results

Tables 3 and 4 show the execution times and results of our experiments.

For RQ2, a ✓ mark means that the output values, that are obtained from the mapping using random inputs varying from 0 to 200, have relative errors within 10% against the true values. A $\times$ mark stands for other cases.

5 DISCUSSION

5.1 RQ1

With Mathematica, we can correctly derive closed-form solutions of the recurrence formulae obtained from the programs. For some programs not shown in Table 2, we cannot derive their closed-form solutions. The reasons are that 1) Mathematica cannot deal with them, and 2) in general, not every recurrence relation has a closed-form solution. For such cases, the existing methods also cannot be applied.

5.2 RQ2

There are mappings that are not applicable to SAT/SMT solvers. For example, for Newton, our method derives an expression $\sqrt{a}\cosh(2^n \cosh^{-1}\sqrt{a})$. Z3 cannot deal with the expression.

For such a case concolic testing [22], [23] might be a solution for further analysis.

5.3 RQ3

The existing method, in principle, cannot derive a correct mapping for a Fibonacci generator. The existing method approximates it as quadratic equations. It, however, has large relative errors for a large input. Thus, advantage of our proposed method is confirmed.

5.4 Execution Times

A large proportion of execution times are occupied by Mathematica computation. Particularly, solving the closed-form solutions is highly time consuming.

5.5 Other Discussions

The proposed method in this work uses regression analysis for obtaining a mapping between the number of loop iterations and arguments. In general, obtaining a relation between variables before and after the target loop is a complex task. For this reason, relative errors arising from regression analysis become small. Consequently, our approach has the advantage for cases in which (1) a mapping between the number of loop iterations and arguments is linear or quadratic, and (2) a relation between variables before and after the target loops is complex.

5.6 Limitation of the Methods

The closed-form solution is obtained using Mathematica in this work; thus the ability of obtaining the solution depends on Mathematica.

We assume that the patterns of execution paths are in the restricted form shown earlier in the execution path assumption. If the code violates the assumption, then the proposed method cannot be applied.

5.7 Treats to Validity

The programs used in the experiments are small. The number of the programs is also small. The results, however, show that the proposed method can be applicable to programs that cannot be handled by the existing methods.

The class of variable types for variable is restricted, mainly due to the limitations of SAT/SMT solvers.

6 CONCLUSION

This paper proposed a new method for inferring a mapping between variables before and after a given loop structure. The experimental results show that our proposed method can derive complex mapping, which the existing methods cannot successfully derive.

Our future work includes the application of concolic testing on our derived mappings, in order to perform efficient and further analysis.

Acknowledgments

The research is also being partially conducted as Grant-in-Aid for Scientific Research C (16K00094) and S(25220003).

Table 2: Target Programs

Program	What to process	LOC	Number of loop structures	Number of if statements
Fibonacci	Fibonacci numbers	20	1	1
Newton	calculation of square root by Newton method	30	1	1
DrawPict	draw pictures	39	1	1
DrawPara	draw parabola	77	3	10
Summation	calculation of summation	20	1	1
Power	calculation of power series	20	1	1

Table 3: Execution Times (sec.)

Program	Step 1	Step 2	Step 3	Step 4	Step 5	Steps 6 and 7
Fibonacci	2.6	0.62	0.1	4.4	4.4	0.72
Newton	2.3	0.36	0.0	2.4	1.3	0.64
DrawPict	2.7	0.59	0.0	3.0	2.2	0.75
DrawPara	2.5	1.1	0.0	2.9	2.9	0.65
Summation	2.4	0.57	0.05	2.4	2.4	0.66
Power	2.3	0.38	0.0	3.5	3.5	0.75

Program	total
Fibonacci	8.4
Newton	5.8
DrawPict	7.0
DrawPara	7.2
Summation	6.0
Power	6.9

Table 4: Experimental Results

Program	the existing method	our method	RQ2
Fibonacci	×	✓	×
Newton	×	✓	×
DrawPict	✓	✓	✓
DrawPara	×	✓	✓
Summation	✓	✓	✓
Power	×	✓	×

Funding from Mitsubishi Electric Corporation is gratefully acknowledged.

REFERENCES

[1] G. T. Leavens, A. L. Baker, and C. Ruby: "JML: a Java modeling language," Formal Underpinnings of Java Workshop (at OOPSLA'98) pp.404–420 (1998).

[2] P. Godefroid, N. Klarlund, and K. Sen: "DART: directed automated random testing," ACM SIGPLAN Notices, Vol.40, No.6, pp.213–223 (2005).

[3] W. Visser, K. Havelund, G. Brat, S. Park, and F. Lerda: "Model checking programs," Automated Software Engineering, Vol.10, No.2, pp.203–232 (2003).

[4] P. Cousot: "Proving program invariance and termination by parametric abstraction lagrangian relaxation and semidefinite programming," Proceedings of the 6th International Conference of VMCAI 2005, Vol. 3385, pp.1–24, Lecture Notes in Computer Science (2005).

[5] T. Ball and S.K. Rajamani: "The slam project: Debugging system software via static analysis," Proceedings of the 29th ACM SIGPLAN-SIGACT POPL'02, pp.1–3 (2002).

[6] T.A. Henzinger, R. Jhala, R. Majumdar, G.C. Necula, G. Sutre, and W. Weimer: "Temporal-safety proofs for systems code," Proceedings of the 14th International Conference on Computer Aided Verification, CAV 2002, pp.526–538 (2002).

[7] A. Biere, M. Heule, H. Van Maaren, and T. Walsh: "Handbook of Satisfiability," IOS press (2009).

[8] L. deMoura and N. Bjørner: "Z3: An efficient smt solver," Proceedings of Tools and Algorithms for the Construction and Analysis of Systems 2008, Vol.4963, pp.337–340, Lecture Notes in Computer Science (2008).

[9] B. Dutertre: "Yices 2.2," Computer-Aided Verification (CAV'2014), Vol.8559, pp.737–744, Lecture Notes in Computer Science (2014).

[10] M.D. Ernst, J. Cockrell, W.G. Griswold, and D. Notkin: "Dynamically discovering likely program invariants to support program evolution," IEEE TSE, Vol.27, pp.1–25 (2001).

[11] M. Younger: "Handbook for Linear Regression," Duxbury Resource Center (1979).

[12] W. Le: "Segmented Symbolic Analysis," Proceedings of the 2013 International Conference on Software Engineering, pp.212–221 (2013).

[13] D.R. Cok and J.R. Kiniry: "Esc/java2: Uniting esc/java and jml: Progress and issues in building and using esc/java2 and a report on a case study involving the use of esc/java2 to verify portions of an internet voting tally system," Proceedings of Construction and Analysis of Safe, Secure and Interoperable Smart Devices: International Workshop, CASSIS 2004, Vol.3362, pp.108–128, Lecture Notes in Computer Science (2005).

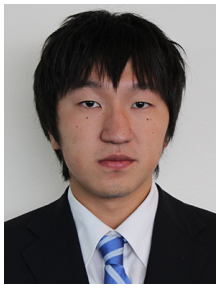
[14] C. Cadar, D. Dunbar, and D. Engler: "KLEE: Unassisted and Automatic Generation of High-coverage Tests for Complex Systems Programs," Proceeding of the 8th USENIX Conference on Operating Systems Design and Implementation, pp.209–224 (2008).

[15] V. Chipounov, V. Kuznetsov, and G. Candea: "S2E: A Platform for In-vivo Multi-path Analysis of Software Systems," Proceedings of the 16th International Conference on Architectural Support for Programming Languages and Operating Systems, pp.265–278 (2011).

[16] Y. Xie, A. Chou, and D. Engler: "ARCHER: Using Symbolic, Path-sensitive Analysis to Detect Memory Access Errors," Proceedings of the 9th European Software Engineering Conference Held Jointly with 11th ACM SIGSOFT International Symposium on Foundations of Software Engineering, pp.327–336 (2003).
[17] "Eclipse Java development tools (JDT)," (accessed 2015-05-05). http://www.eclipse.org/jdt/.
[18] K. Kobayashi, Y. Sasaki, K. Okano, and S. Kusumoto: "Automated assertion generation using PDF and SMT-Solver," IEICE Transaction on Information and Systems, JD, Vol.96, No.11, pp.2657–2668 (2013) (In Japanese).
[19] "the R statistical package," (accessed 2015-05-05). http://cran.r-project.org/.
[20] E. Clarke, D. Kroening, and F. Lerda, "A Tool for Checking ANSI-C Programs," Proceedings of the 10th International conference on Tools and Algorithms for the Construction and Analysis of Systems, pp.168–176 (2004).
[21] "Mathematica: Wolfram Research," (accessed 2015-05-05). https://www.wolfram.com/.
[22] K. Sen, D. Marinov, and G. Agha: "CUTE: A Concolic Unit Testing Engine for C," SIGSOFT Software Engineering Notes, Vol.30, No.5, pp.263–272 (2005).
[23] R. Majumdar and K. Sen: "Hybrid Concolic Testing," Proceeding of the 29th International Conference on Software Engineering, pp.416–426 (2007).

(Received October 20, 2017)
(Revised December 27, 2017)

Kozo Okano received his BE, ME, and PhD degrees in Information and Computer Sciences from Osaka University in 1990, 1992, and 1995, respectively. From 2002 to 2015, he was an Associate Professor at the Graduate School of Information Science and Technology of Osaka University. In 2002 and 2003, he was a visiting researcher at the Department of Computer Science of the University of Kent in Canterbury, and a visiting lecturer at the School of Computer Science of the University of Birmingham, respectively. Since 2015, he has been an Associate Professor at the Department of Computer Science and Engineering, Shinshu University. His current research interests include formal methods for software and information system design. He is a member of IEEE, IEICE, IPSJ.



Yukihiro Sasaki received his BI and MI degrees from Osaka University in 2012 and 2014, respectively. He currently works for Mitsubishi Electric Corporation. His research interests include dynamic generation of assertion for programs.

Shinji Kusumoto received his BE, ME, and DE degrees in Information and Computer Sciences from Osaka University in 1988, 1990, and 1993, respectively. He is currently a Professor at the Graduate School of Information Science and Technology of Osaka University. His research interests include software metrics and software quality assurance techniques. He is a member of the IEEE, the IEEE Computer Society, IPSJ, IEICE, and JFPUG.

Regular Paper

A Fast Online Algorithm for Analyzing Magnitude Fluctuation of Time Series

Makoto Imamura* Junji Tsuda* Daniel Nikovski ** Masato Tsuru ***

* School of Information and Telecommunication Engineering, Tokai University, Japan
** Mitsubishi Electric Research Laboratories, USA
*** Department of Computer Science and Electronics, Kyushu Institute of Technology, Japan
Imamura@tsc.u-tokai.ac.jp, 7bjnm017@mail.u-tokai.ac.jp
nikovski@merl.com, tsuru@cse.kyutech.ac.jp

Abstract - Equipment condition monitoring (ECM) has attracted much attention recently in industrial domains, especially as the Internet of Things (IoT) has been emerging and growing rapidly. Monitoring the fluctuations of sensor data generated by industrial equipment is an important issue when trying to detect equipment anomalies. This paper proposes a new fast online algorithm for analyzing a novel magnitude fluctuation feature computed from unsteady time series. The magnitude fluctuation is defined by a convex-shaped pattern which consists of an upward trend leg and a downward trend leg. This definition enables the extraction of anomalous spikes and operational regimes in sensor data of equipment by using the amplitude and duration of an extracted convex-shaped pattern. We also show that the computational complexity of our proposed algorithm is O(n), where n is the length of the input time series; this complexity enables real-time sensor data processing with a sampling period at the microsecond level.

Keywords: Magnitude Fluctuation Analysis, Anomaly Detection, Feature Extraction, Time Series Datamining, Equipment Condition Monitoring, Online Algorithm

1 INTRODUCTION

As the Internet of Things (IoT) [1] has been emerging and growing, sensor big data that is streamed from various kinds of equipment in power plants, industrial facilities, and buildings can be made available for monitoring, diagnosis, energy-saving, productivity improvement, quality management, and marketing. As a result, industry has paid much attention to the use of big sensor data generated from equipment or facilities in order to create a smart society.

Equipment Condition Monitoring (ECM) is a commonly used service based on sensor big data, and data mining techniques are key components in making ECM smarter [2]. This paper proposes a new magnitude fluctuation feature for unsteady and random time-series as a tool for a data mining technique, and also describes an efficient online algorithm for computing it from sensor big data.

After mechanical equipment has been operated for a long time, convex-shaped spikes are often observed in sensor data such as the torque current of motors or the pressure inside a pipe, because of frictional wear, adhesion of foreign substances, etc. Therefore, extracting convex-shaped spikes in sensor data is useful for detecting anomaly or degradation of equipment. However, convex-shaped patterns occur in sensor data not only as symptoms of degradation but also of controlled operating patterns or random noise (Figure 1). In most cases, the heights of convex-shaped patterns are different depending on whether it is a control operating pattern, a degradation symptom, or noise. As shown in Figure 1, the height of an operational pattern is often larger than that of a degradation symptom pattern. In its turn, the height of a degradation symptom pattern is often larger than that of noise. In this work, we define *extended maximal convex curves* to represent a convex-shaped pattern in a time series, along with its height, which we call *amplitude*. Furthermore, we propose a fast online algorithm to extract extended maximal convex curves from a time series, by introducing a novel operation "leg reduction", which will be explained later in Section 2.2. Online algorithms are typically a requirement for realizing real-time equipment condition monitoring.

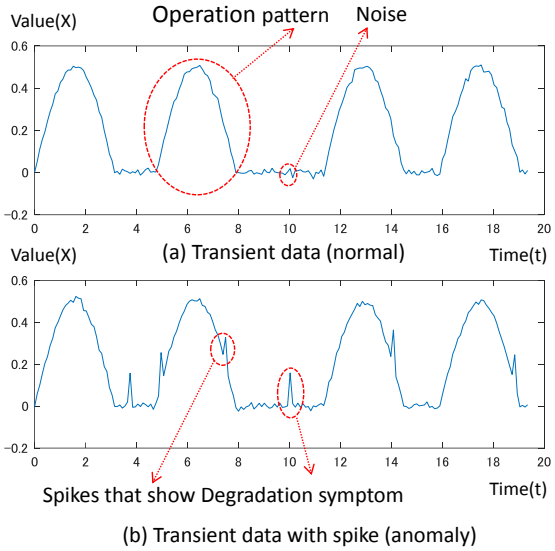


Figure 1: Transient data with spikes

Magnitude fluctuation for unsteady data has been studied from the perspective of data mining by Fink et al [3]. They proposed the concept of a leg, and its associated search method to find a global trend in a time-series including small variations such as noise. The dotted lines in Figure 2 are examples of legs. Both lines show the global upward trend that includes local up-down segments. However, their method treats only single legs, finding an upward or downward trend,

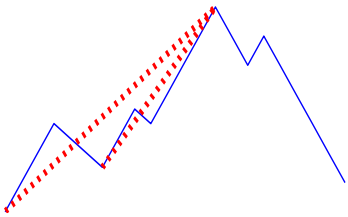


Figure 2: Leg

but can't determine the magnitude of fluctuations. A convex curve could be defined as the continuous occurrence of upward and downward trends. However, in the case of a trapezoidal subsequence with noise, it is non-trivial to select convex curves from several candidates. This is the reason for the need of a new notion of an "extended maximal convex curve", and it will be discussed later in Figure 6 in section 2.1. Furthermore, a naïve application of Fink's algorithm to extract a convex curve needs a computation proportional to $n \times l$, where n is the length of a given time series and l is the average length of legs. In contrast, the computational complexity of our proposed algorithm is $O(n)$, and it doesn't depend on the length of the convex curve.

There are also related works on time series processing with legs [4] [5]. These previous works of ours proposed the computation of the frequency of fluctuations in time series where upward trends and downward trends appear alternately and iteratively. A leg frequency is defined for a given window size and an amplitude. Regarding the difference between our previous works and this research, whereas the leg frequency has window size as a parameter that should be optimally selected by users, the amplitude of a convex curve in this research has no parameter. This means a higher usability of the amplitude of a convex curve. This paper is the extended version of our earlier work [6]. The main difference with [6] is that in this paper we show the proof and the evaluation of our proposed algorithm, while [6] only suggested its possibility.

Related works on extracting pattern from time series include motif discovery [7][8], discord discovery [9] [10] and autoregression [11]. The difference between these existing works is whether an algorithm has window size as a parameter or not. This means that our work does not need to decide an appropriate window size.

Related work on finding a subsequence that includes a distinctive pattern in an online setting is online segmentation [12]. The difference between that existing work and our work is that the former is exclusive segmentation, but the latter is overlap segmentation. Our segmentation problem is how to extract all of the convex curves included in a time series, while a convex curve may include some other convex curves. When a larger fluctuation includes smaller fluctuations as shown in the bottom graph in Figure 1, the smaller convex curve is included by the upward or downward trend in the larger convex curve.

The rest of our paper is organized as follows. Section 2 describes the definition of maximal convex curve and its mathematical properties. Section 3 shows a maximal convex curve amplitude calculating algorithm, and analyzes its order of complexity. Section 4 evaluates our proposed algorithm empirically. First, we show that it can extract convex-shaped spikes in transient data by experimental means. Second, we show that the execution time of our algorithm is linear in *n*. Section 5 provides conclusions and directions for future work.

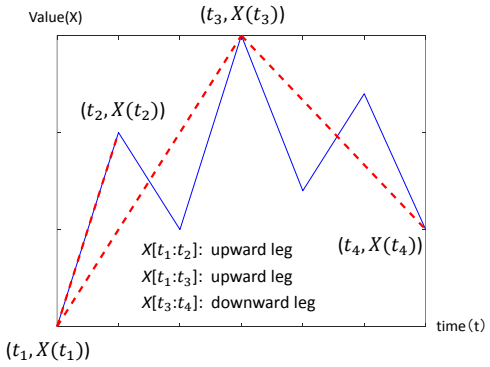


Figure 3: Upward and downward legs

2 MAXIMAL CONVEX CURVE

This section defines the amplitude of the maximal convex curve at each time of a given time series as a feature which shows the degree of magnitude fluctuation of the time series. The merit of our proposed feature is that it is parameter free. It means that it is not necessary to tune parameters when we use this feature. On the other hand, most of the features of time series proposed in the existing studies depend on at least the window size, so how to select an optimal window size is often a problem.

2.1 Definition of a Maximal Convex Curve

The maximal convex curve at each time *t* is defined as a pair of the maximal leg from t toward left and that toward right. However, a naive definition of a maximal leg makes its amplitude unstable. Therefore, we introduce an extended maximal leg from t toward left or right to obtain a robust definition.

Definition: time series X, subsequences X[p:q]

A *Time Series* $X=[x_1, \cdots, x_m]$ is a continuous sequence of real values. The value of the i-th time point is denoted by

$$X[i] = x_i.$$

A *subsequence* $S = [x_p, x_{p+1}, \ldots, x_q] = X[p{:}q]$ is a continuous subsequence of X starting at position p and ending at position q. We denote the length of a subsequence S by len:

$$len(S) \equiv q - p + 1$$

Definition: Leg

Let X be a time series. We define a leg by a subsequence $L = X[l{:}r]$ that satisfies the conditions below.

$$\forall i.\ l < i < r \quad (X[r] - X[i])(X[i] - X[l]) > 0$$

That is, a subsequence $X[l{:}r]$ has a maximum and a minimum at the terminal points *l*, *r*.

If $X[r] - X[l] > 0$, a leg L is called an upward leg.
If $X[r] - X[l] < 0$, a leg L is called a downward leg.

Figure 3 shows examples of upward and downward legs.

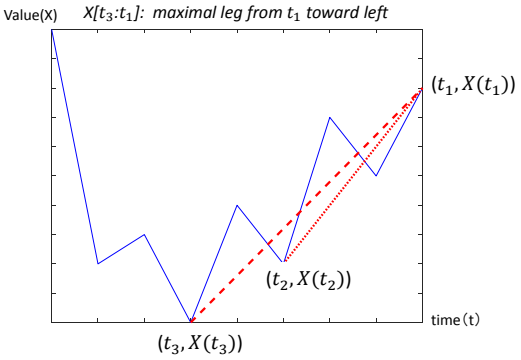


Figure 4: Maximal leg from t toward left

Definition: Sign and amplitude of a leg

We define the *sign* and *amplitude* of a leg $L = X[l:r]$ by the functions below. We denote them by amp and $sign$ respectively:

$$amp\ (L) = abs\ (X[r] - X[l]).$$

The absolute function $abs(a)$ means the absolute value of a.

$$\begin{aligned} sign\ (L) &= 1, && \text{if } (X[r] - X[l]) > 0 \\ &= 0, && \text{if } (X[r] - X[l]) = 0 \\ &= -1, && \text{if } (X[r] - X[l]) < 0 \end{aligned}$$

By the above definition, the sign of an upward leg is plus and the sign of a downward leg is minus.

Definition: A Maximal Leg from t toward left

Let *X* be a time series and *t* be a time point in X.

We define *a Maximal Leg from t toward left* by a leg $L = X[l:t]$ that satisfies the following condition.

For any $l' < l$, $X[l':t]$ is not a leg the amplitude of which is larger than that of $X[l:t]$.

That is,

for any l' such that $l' < l$ and

$sign(L)\ (X[t] - X[l]) \leq sign(L)\ (X[t] - X[l'])$

some j such that $l' \leq j < l$ exists and j satisfies

$sign(L)\ (X[t] - X[j]) \leq 0$

Figure 4 shows an example of a maximal leg from t_1 toward left. $X[t_3:t_1]$ is a maximal leg from t_1 toward left. On the other hand, $X[t_2:t_1]$ is a leg, but not a maximal leg from t_1 toward left.

Definition: A Maximal Leg from t toward right

We define *a Maximal Leg from t toward right* by a leg $L = X[t:r]$ that satisfies the following, with everything else is the same as "*Maximal Leg from t toward left*":

For any $r < r'$, $X[t:r']$ is not a leg the amplitude of which is larger than that of $X[t:r]$.

Definition: Convex curve at t in X

Let *X* be a time series and *t* be a time point in X.

We define *a convex curve at t in X* by a subsequence $S = X[l:r]$ that satisfies the following conditions.

(i) $l < t < r$

(ii) $X[l:t]$ is a leg.

(iii) $X[t:r]$ is a leg.

(iv) $\text{sign}(X[l:t])\ \text{sign}(X[t:r]) < 0$

We denote it by $X[l:t:r]$, and call *t* the *vertex* of the convex curve. l and r are called *left terminal* and *right terminal* of the convex curve, respectively. We call an interval $[l:r]$ *support* of the convex curve.

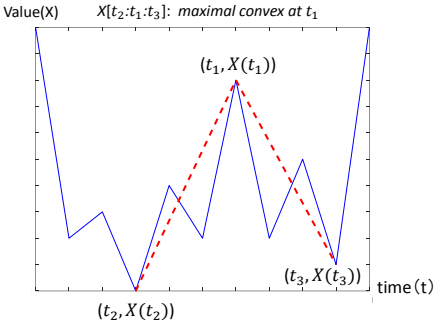


Figure 5: Maximal convex curve

Definition: Maximal Convex Curve at t in X

We define *a maximal convex curve at t in X* by a subsequence $S = X[l:r]$ that satisfies the following conditions.

(i) $l < t < r$

(ii) $X[l:t]$ is a maximal leg from t toward left.

(iii) $X[t:r]$ is a maximal leg from t toward right.

(iv) $\text{sign}(X[l:t])\ \text{sign}(X[t:r]) < 0$

Figure 5 shows the example of a maximal convex curve. $X[t_2:t_1:t_3]$ is a maximal convex curve at t_1.

Definition: *Signed Amplitude of a maximal convex curve*

Let $C = X[l:t:r]$ be a maximal convex curve.

We define *the amplitude* $amp_c(X,t)$, *sign* $sign_c(X,t)$ *and signed amplitude* $signedAmp_c(X,t)$ *of a maximal convex curve at t in X*, respectively, by the functions below:

$amp_c(X,t) \equiv \min(amp(X[l:t]), amp(X[t:r]))$

$sign_c(X,t) \equiv sign(X[l:t])$

$signedAmp_c(X,t) \equiv sign_c(X,t) \times amp_c(X,t)$

If *t* is not a vertex of a maximal convex curve, we define the amplitude at *t* to be zero.

The above definition of a maximal convex curve is not robust in the sense that even a small change of the value at the vertex of a convex curve can lead to a large change of amplitude value. For example, in Figure 6, suppose that $X(t_2) = X(t_4)$ and $X(t_1) = X(t_5)$. If $X(t_2) = X(t_3) = X(t_4)$, a maximal convex curve at t_2 is $X(t_1:t_2:t_5)$ and its amplitude is $(X(t_2) - X(t_1))$. If $X(t_3)$ is just a little less than $X(t_2)$, a maximal convex curve at t_2 is $X(t_1:t_2:t_3)$, and a maximal convex curve at t_4 is $X(t_3:t_4:t_5)$. And, the amplitude of each convex curve is $\big(X(t_2) - X(t_3)\big)$. That is, just a small change of X can make a great change of the amplitude.

In real sensor data, even real-valued data may have discrete values by the limitation of a sensor or a measuring device, such that they often might have the same values. Therefore, the above is not an atypical contrived case.

We introduce the concept of an extended leg in order to obtain a robust definition of a maximal convex curve.

Definition: Extended Leg

We define an *Extended Leg* by the sequence $L = X[l:r]$ that satisfies the condition below.

$$\forall i.\ l < i < r \quad (X[r] - X[i])(X[i] - X[l]) > 0 \\ \vee\ X[i] = X[r]$$

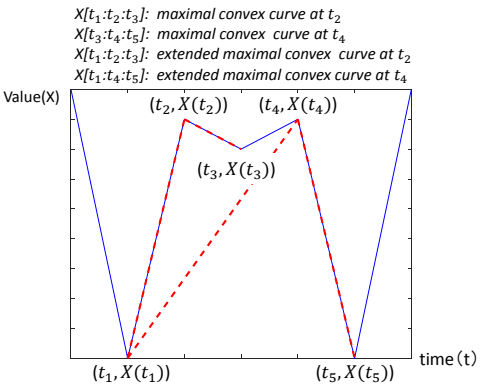


Figure 6: Extended maximal convex curve

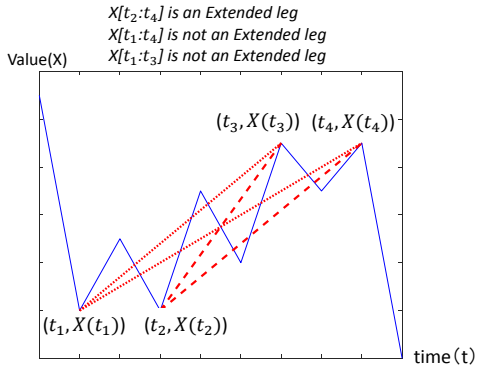


Figure 7: Examples of extended legs

Figure 7 shows the example of an extended leg. When we assume that $X[t_1] = X[t_2]$ and $X[t_3] = X[t_4]$, $X[t_2{:}\,t_3]$ and $X[t_2{:}\,t_4]$ are extended legs, but $X[t_1{:}\,t_4]$ and $X[t_1{:}\,t_3]$ are not extended legs.

"Maximal Extended Leg from t toward left and right" can be defined by the same way as *"a Maximal Leg from t toward left and right"*, respectively.

Definition: Extended Maximal Convex Curve at t in X
Let X be a time series and t be a time point in X.
We define an extended maximal convex curve at t in X by a subsequence $C = X[l{:}\,r]$ that satisfies the following conditions. We denote it by $X[l{:}\,t{:}\,r]$
(i) $l < t < r$
(ii) $X[l{:}\,t]$ is a maximal extended leg from t toward left.
(iii) $X[t{:}\,r]$ is a maximal leg from t toward right.
The *amplitude*, *sign* and *signed amplitude* of an extended maximal convex curve at t are defined similarly to those of a maximal convex curve.

Please note that the left side is extended but the right side is not extended in the above definition, so that we do not count the larger amplitude twice. For example, in Figure 6,$X[t_1{:}\,t_2{:}\,t_3]$ and $X[t_1{:}\,t_4{:}\,t_5]$ are extended maximal convex curves, but $X[t_1{:}\,t_2{:}\,t_5]$ is not an extended maximal convex curve.

Definition: Amplitude function, Positive amplitude function
Let X be a time series. *Amplitude function* $amp_X(t)$ is a function from each t in X to the signed amplitude of the extended maximal convex curve at t. If t is not vertex, its amplitude is defined to be 0. *Positive amplitude function* is defined to be $max(amp_X(t), 0)$. *Negative amplitude function* is defined to be $min(amp_X(t), 0)$.

2.2 Properties of Maximal Convex Curves

This section discusses the amplitude property of a maximal convex curve for deriving a fast online algorithm to calculate a maximal convex curve at each time t. A maximal convex curve can be defined at the point at which X is a locally maximal or minimal value. Hereafter, we assume that X is a locally maximal time series that consists of only locally maximal or minimal values.

Overlap segmentation makes it difficult to extract a maximal convex curve. The reason is that an online algorithm needs to know the time when the maximal amplitude is decided for each point, while reading data in order. In worst case, the convex curve that has the largest amplitude might not be decided until the last data is read. We introduce a novel operation "*leg reduction*" for searching the time when the maximal convex curve is decided. Leg reduction decides the convex curve and simplifies time series by removing the points which are the vertex of decided convex curves. Leg reduction is classified into three types, which are middle, left and right leg reductions, depending on the positions where maximal convex curves are decided. This section describes the definition and mathematical property of leg reductions.

2.2.1 Local Maximal Preserving Transformation

Before describing leg reductions, we introduce *local maximal preserving transformation* to reduce the problem simpler. Local maximal preserving transformation is an operation to remove the points that are not the vertex of convex curves from given time series. We note that the amplitude of the point that is not the vertex of a convex cure is defined to be zero. We will define *locally maximal time series* and *positioned time series* for the preparation to define local maximal preserving transformation.

Definition: Locally maximal time series
For any t, X is a locally maximal time series if it satisfies the following condition:

$$\forall t.\ (\mathrm{X[t+2]} - \mathrm{X[t+1]})\,(X[\mathrm{t+1}] - \mathrm{X[t]}) < 0$$

Locally maximal time series can be obtained from a given time series by removing the points that are neither a local maximum nor a local minimum (Figure 8).

Definition: Positioned time series
Positioned time series is a two-dimensional array $\mathrm{Y} = [\mathrm{X}, \mathrm{P}]$, which consists of time series $\mathrm{X} = [x_1, \ldots, x_i, \ldots, x_n]$ and position series $\mathrm{P} = [1, \ldots, i, \ldots, n]$. And we call $[\mathrm{X}, \mathrm{P}]$ *a positioned time series generated from* X.

Definition: Local maximum preserving transformation
Let X be a time series, and $[\mathrm{X}, \mathrm{P}]$ be a positioned time series generated from X. A local maximal preserving transformation is defined as the repeated below procedures from $\mathrm{E} = [\mathrm{X}, \mathrm{P}]$ until the following procedure from E_1 to E_2 can no longer be applied.

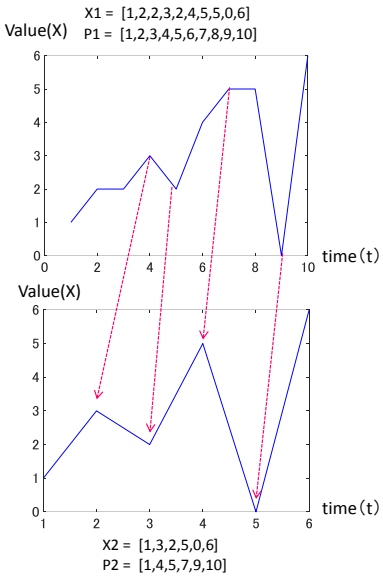


Figure 8: Local maximum preserving trans

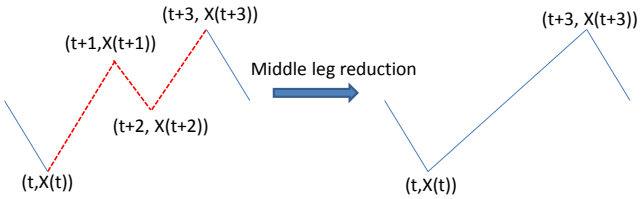


Figure 9: Middle leg reduction

Let $E_1 = [X_1, P_1]$ be a positioned time series.
If $X_1[t] = X_1[t+1]$ or
$(X_1[t+1] - X_1[t])(X_1[t+2] - X_1[t+1]) > 0$,
then we get $E_2 = [X_2, P_2]$
by removing $X_1[t+1]$ and $P_1[t+1]$ from X_1 and P_1, respectively, by the following.
For $0 \le i < t$, $X_2[i] := X_1[i]$, $P_2[i] := P_1[i]$
For $t \le i \le len(X_1) - 1$, $X_2[i] := X_1[i+1]$, $P_2[i] := P_1[i+1]$

Proposition 1: Conservation of convex amplitude in local maximum preserving transformation
Let X_1 be a time series, and $E_2 = [X_2, P_2]$ be a positioned locally maximal time series obtained from X_1 by a local maximum preserving transformation. If a maximal convex curve C_1 is defined at i in X_1, then there is a maximal convex curve C_2 that is defined at j in X_2 such that $P_2[j] = i$, with the same amplitude of C, that is, $amp_c(X_2, j) = amp_c(X_1, i)$, and vice versa.
[Proof] A maximal convex is only defined at a local maximal or minimal point. And the amplitude of a leg is not changed by a local maximum preserving transformation. Therefore, a maximal convex curve in X_1 has the corresponding convex curve in X_2, and vice versa.

2.2.2 Middle Leg Reduction

Definition: Middle leg reduction (Figure 9)
Let $t, t+1, t+2, t+3$ be time points at a locally maximal time series X. If $X[t: t+3]$ satisfies the following conditions (we call them *middle leg conditions*), the procedure to remove $X[t+1]$ and $X[t+2]$ from X is called a *middle leg reduction*.

$$abs(X[t+1] - X[t]) > abs(X[t+2] - X[t+1]) \quad \ldots(1)$$

$$abs(X[t+3] - X[t+2]) \ge abs(X[t+2] - X[t+1]) \quad \ldots(2)$$

More concretely, a middle leg reduction from $E_1 = [X_1, P_1]$ to $E_2 = [X_2, P_2]$ is described by the following.
Let $E_1 = [X_1, P_1]$ be a positioned time series, and X_1 be a locally maximal one that satisfies middle leg conditions. For
$1 \le i \le t$, $X_2[i] := X_1[i]$, $P_2[i] := P_1[i]$
For $t+1 \le i \le len(X_1) - 2$,
$X_2[i] := X_1[i+2]$, $P_2[i] := P_1[i+2]$

We note that inequality (1) does not contain an equal sign, whereas the inequality (2) contains an equal sign. It corresponds to the definition of extended maximal convex curve.

Proposition 2: *Conservation of convex amplitude in a middle leg reduction*
Let $E_1 = [X_1, P_1]$ be a positioned local maximal time series, and $E_2 = [X_2, P_2]$ be a positioned time series obtained from E_1 by a middle leg reduction.
(i) For $i \le t$, $amp_c(X_2, i) = amp_c(X_1, i)$
For $i \ge t+1$ $amp_c(X_2, i) = amp_c(X_1, i+2)$
(ii) $amp_c(X_2, t+1) = amp_c(X_1, t+2)$
$= abs(X_1[t+2] - X_1[t+1])$
[Proof]
(i) When $X_1[t+1]$ and $X_1[t+2]$ are removed from X_1, a subsequence $X_1[t: t+3]$ is a leg the sign of which is the same as $X_1[t: t+1]$ and $X_1[t+2: t+3]$. $X_1[t]$ and $X_1[t+3]$ keep being local maxima. Therefore, E_2 keeps being a positioned local maximal time series. Therefore, the amplitude of every convex curve at a time point t in X_1 except for $t+1$ and $t+2$ is not changed after a middle leg reduction, because of the definition of a maximal leg from t.
(ii) If $X_1[t: t+3]$ satisfies the middle leg conditions, a leg $X_1[t: t+1]$ is a maximal leg from $t+1$ toward left and $X_1[t+1: t+2]$ is a maximal leg from t+1 toward right. Therefore,
$amp_c(X_2, t+1)$
$= \min(amp(X_1[t: t+1]), amp(X1[t+1: t+2])$
$= abs(X_1[t+2] - X_1[t+1])$
$amp_c(X_2, t+2) = abs(X_1[t+2] - X_1[t+1])$ is proved similarly.

2.2.3 Left Leg Reduction

Definition Left leg reduction (Figure 10)
Let 1, 2,3 be time points at a locally maximal time series X. If $X[1: 3]$ satisfies the following conditions, a procedure to remove $X[1]$ from X is called a *left leg reduction*.

$$abs(X[2] - X[3]) \ge abs(X[2] - X[1])$$

The above condition is called a *left leg condition*.

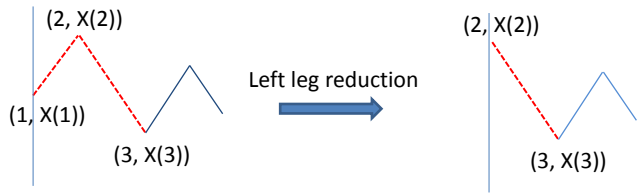


Figure 10: Left leg reduction

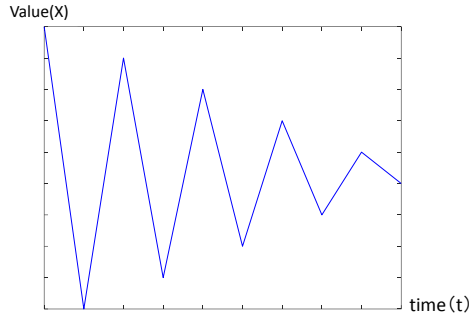


Figure 11: Shrinking time series

More concretely, a left leg reduction from $E_1 = [X_1, P_1]$ to $E_2 = [X_2, P_2]$ is described by the following:
Let End_1=len(X_1).
For i $\leq End_1 - 1$, X_2 [i] := X_1 [i+1], P_2 [i] := P_1 [i+1]

Proposition 3: Conservation of convex amplitude in left leg reduction
Let X be a time series, $E_1 = [X_1, P_1]$ be a positioned local maximal time series generated from X, and $E_2 = [X_2, P_2]$ be a positioned time series obtained from E1 by a left leg reduction.
(i) For i $\leq$ len(X_1) $- 1$, $amp_c(X_2, i) = amp_c(X_1, i+1)$
(ii) $amp_c(X_2, 2) =$ abs($X_1[2] - X_1[1]$)
[Proof] The proof for Proposition 3 is similar to that of Proposition 2.

We define shrinking time series for describing proposition 4.
Definition: Shrinking time series (Figure 11)
The locally maximal time series X is called shrinking if it satisfies the following condition.
$\forall t$. abs(X(t+1) – X(t)) > abs(X(t+2) – X(t+1)) ...(3)

Proposition 4: By repeating middle and left leg reductions until they can no longer be applied, we get a shrinking time series.
[Proof]
Let *R* be the time series that is gotten by repeating middle and left leg reductions until they can no longer be applied.
The first 3 points of *R* satisfy the below inequality, because left reduction cannot be applied.
$abs(R[2] - R[1]) > abs(R[3] - R[2])$
Therefore, the first three points satisfy inequality (3) in the definition of shrinking time series.
The next three points satisfy the below inequality, because middle leg reduction cannot be applied to the first 4 points of *R*.
$abs(R[3] - R[2]) > abs(R[4] - R[3])$
Therefore, the first four points satisfy inequality (3) in the definition of shrinking time series.
By repeating the same operation until the end of time series *R*, we get that all the points in time series *R* satisfy inequality (3) based on mathematical induction.
Note that if the sign is " $\geq$ " in inequality (1) of middle leg reduction as with that in inequality (2), this proposition is not valid. This shows that an extended maximal curve is necessary not only for robust definition but also for fast algorithm.

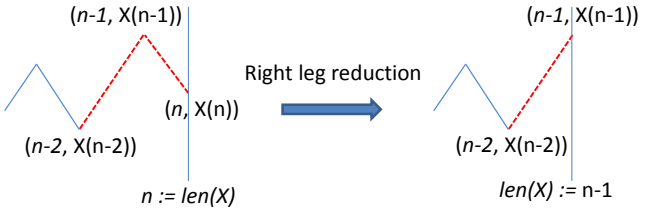


Figure 12: Right leg reduction

2.2.4 Right Leg Reduction

Definition: Right leg reduction (Figure 12)
Let "End" be $len(X)$. If $X[\text{End} - 2 : \text{End}]$ satisfies the following condition,
$abs(X[End] - X[End-1]) \geq abs(X[End-1] - X[End-2])$
A procedure to remove $X[End]$ from X is called a *right leg reduction*. The above condition is called *right leg condition*.
More concretely, a right leg reduction from $E_1 = [X_1, P_1]$ to $E_2 = [X_2, P_2]$ is described by the following:

For i $\leq$ len(X_1) $- 1$, X_2[i] := X_1 [i] , P_2[i] := P_1 [i]

Proposition5: *Conservation of convex curve amplitude in right leg reduction*
If a positioned local maximal time series E = [X, P] is shrinking,
(i) $amp_c(X, End-1) =$ abs($X[End] - X[End-1]$)
where $End = len(X)$.
[Proof] It follows directly from the definitions of a locally maximal time series and a shrinking time series.

2.2.5 Main Theorem

Theorem: Let X be a time series. The signed amplitude and length of an extended maximal convex curve at *t* in X can be calculated by middle, left and right leg reductions. In other words, the amplitude function for X can be also calculated.
[Proof]
If we apply middle and left leg reductions repeatedly until they can no longer be applied, we get a shrinking time series by proposition 4. For all the local maxima and minima that are removed by middle or left leg reduction, their amplitudes are calculated by proposition 2 and 3. For the remaining local maxima and minima, their amplitudes are calculated by proposition 5. Therefore, all the amplitudes of maximal convex curves in X are calculated by middle, left and right leg reductions.

3 MAXIMAL CONVEX CURVE AMPLITUDE CALCULATING ALGORITHM

This section shows an online algorithm to calculate the amplitude function for a given time series. The computational complexity of a naive algorithm by the definition of a maximal convex curve is $O(n^2)$, but that of our proposed one is $O(n)$ based on the results in the preceding section 2.2.
The values of an amplitude function do not depend on the order of the applications of a middle leg reduction and a left leg reduction, by virtue of the propositions in section 2.2. Therefore, we can get an online algorithm by the repetition of

executable reductions while scanning the values from the beginning.

Figure 13 shows an algorithm that computes an amplitude function. In Figure 13, "X" is an input time series and "A" is the values of the amplitude function at time *t* for "X". Both are implemented as a one-dimensional array.

Line 1-4 initializes the output "A", variable "S" and "F". "S" is a stack that stores the vertexes that are used for leg reductions. "F" is a flag that decides when the while-loop execution terminates.

Line 6-32 is a main loop that terminates when all the lines are scanned. Line 7 pushes a local maximum or minimum to the top of stack "S". The first and the last points at "X" are treated as local maximum or minimum.

Line 8-29 is a while-loop that executes middle and left leg reductions until those cannot be applied any more. Line 10-14 corresponds to a middle leg reduction. If a middle leg reduction is executed, then a flag "F" is set to be 1 at line 15, else "F" is set to be 0 at line 17. Line 20-22 corresponds to a left leg reduction. If a left reduction is executed, then a flag "F" is set to be 1 at line 23, else "F" is set to be 0 at line 25. If neither a middle leg reduction nor a left leg reduction is executed, "F" is set to be 0 at line 28. If "F" equals 0, the while-loop terminates. Then a function "getNextLocal-Maximum" will search the next maximum or minimum point at "X" and set it to "i" in line 30. If "i" is the last point at "X", main for-loop ends.

In line 10-14, line 10-11 checks the middle leg coditions. Line 12 and 13 correspond to the equations (ii) in the proposition 2. Line 14 corresponds to the operation obtaining X_2 from X_1 by a middle leg reduction. A function "pop(S, [2,3])" pops the second and the third values of a stack "S" and fills them with the values followed the fourth and fifth value in order.

In line 20-22, line 20 checks a left leg condition. Line 21 corresponds to the equation (ii) in the proposition 3. Line 22 corresponds to the operation obtaining X_2 from X_1 by a left leg reduction. A function "pop(S, [3])" pops the third value of a stack "S" and fill them with the values followed the fourth value in order.

Line 34-37 is the repeated execution of right leg reductions. The main theorem ensures that the remaining values in stack "S" can be reduced to "S" whose size is 2 by repeated application of right leg reductions. Line 36 corresponds to the equation (i) in the proposition 5. A function "pop(S, [1])" pops the first value of a stack "S" and fill them with the followed values in order.

When all the values of "X" are scanned and the size of stack "S" becomes 2, "maximalConvexAmplitude" ends and return output values "A" in line 38.

Lastly, we will show the computational complexity of this algorithm. Let *n* be the length of time series "X". The lines that depend on *n* are for-loop (line 6-32), the first while-loop(line 8-29), "getNextLocalMaximum" (line 30) and the second while-loop (line 34-37).

The repeat count of the for-loop starting at line 6 equals to the number of the local maxima and minima of "X" (We call the number *f*). And the total number of reductions in the first while-loop starting at line 8 is also smaller than the number *f*.

Algorithm: maximalConvexAmplitude *(X)*

[Input] X: time series
[Variable] S:Stack, F: Flag, i: Maximum or minimum point at X
[Output] A: the values of amplitude function for X

```
01  // (1) Initilization
02  A:= zeros(len(X));     // All the value of A is zero.
03  S : = [];              // Initialize a stack
04  F := 1;                // Flag that decides while-loop execution
05  // (2) Main loop
06  for i:=1 to len(X)     //  len(X) is the length of X
07    S := push(S,X[i]) ;
08    while F == 1
09      if len(S) >= 4      // Prop. 2: Middle leg reduction
10        if abs(X(S[3] – X(S[4])) >abs(X(S[2]) – X(S[3]) and
11           abs(X(S[1])–X(S[2])) ≥ abs(X(S[2]) – X(S[3]))
12           A(S[3]) := X(S[3]) – X(S[2]);
13           A(S[2]) := X(S[2]) – X(S[3]);
14           S := pop(S, [2,3]);
15           F := 1;
16        else
17           F:=0;
18        end if
19      elseif len(S) >= 3    //Prop. 3: Left leg reduction
20        if abs(X(S[1]) – X(S[2])) ≥abs(X(S[2]) – X(S[3]))
21          A(S[2]) := X(S[2]) – X(S[3])
22          S := pop(S, [3]);
23          F := 1;
24        else
25          F:=0;
26        end if
27      else
28        F:=0;
29    end while
30    i := getNextLocalMax (X,i) //Prop.1:Local maximal transformation
31    F: = 1;
32  end for
33  // (3) Post process
34  while len(S) >= 3       // Prop.4 and 5: Right leg reduction
35     A(S[2]) := X(S[2])  – X(S[1]);
36     S := pop(S, [1]);
37  end while
38  return A;
```

Figure 13: Maximal convex amplitude calculation

Therefore, the computational complexity of the first while-loop is order O(*f*), that is, at most order O(*n*). We note that *f* is smaller than *n*. Furthermore, the computational complexity of the total execution of "getNext-LocalMaximum" is order O(*n*), because it scans the values of "X" just once. As a result, the computational complexity of the for-loop is O(*n*).

Next, the total repeat count of the second while-loop is also smaller than *f*. In conclusion, we get the result that the computational complexity of "maximalConvex-Amplitude" is order O(*n*).

4 EVALUATION

The preceding section showed that our proposed algorithm to calculate "amplitude function (the signed amplitude of a maximal convex curve at a time)" is online and its computational complexity is order O(*n*).

This section shows that our proposed algorithm can extract convex-shaped spikes in transient data and it enables real-time data processing with a sampling period at the microsecond level by the experiment with simulated data and real data [13].

4.1 Convex-Shaped Pattern Extraction

First, we show that our algorithm can extract convex-shaped spikes from noisy sine data. Second, we confirm that it can extract spikes in transient data shown in Fig. 1. Last, we show that it can extract various convex patterns by giving the levels of amplitudes for space shuttle Marotta Valve data.

(1) Noisy sine wave with spike

In Figure 14, the top graph is a sine curve with noise, and the bottom graph shows positive amplitude function. This figure shows that the positive amplitude at a local maximal point represents the magnitude fluctuation of convex-shaped patterns even during the transient period. Furthermore, amplitude function can distinguish noisy convex patterns than from main convex patterns whose amplitude is approximately 2.5.

In Figure 15, the top graph is an anomalous transient time series mixed with convex-shaped spikes, and the bottom graph is the positive amplitude function. It shows that our algorithm can extract not only an operational patterns with 0.5 scale amplitude, but also convex-shaped spikes with 0.1 scale amplitude.

(3) Space telemetry: space shuttle Marotta Valve

Figure 16 shows an example of a Space Shuttle Marotta Valve time series that are annotated as normal [11]. Marotta Valve is a fuel supply valve for airplane or rocket.

In Figure 17, the top graph is a Space shuttle Marotta Valve time series that contains 5 cycles. The bottom graph shows the amplitudes that are larger than 3. Red circles in the top graph mean the vertexes of convex-shaped patterns. Dotted lines in the top graph are left or right terminals of the convex-shaped patterns whose vertexes are red circles. This shows that our algorithm can extract normal operation patters by giving an amplitude as a value that is a little smaller than a maximum of one normal cycle.

Figure 18 is an enlarged view of Figure 17 during times from 351 to 390. It shows that the vertexes at around 390 are seen as one vertex in Figure 17, but there are two convex-shaped patterns whose amplitudes are larger than 5 in them.

In Figure 19, the top graph shows amplitudes that are larger than 1.2 and smaller than 3. The red circles and red dotted lines have the same meaning as in Figure 17. Each up and down spike in energizing phases is extracted from each cycle, but the first and the second cycles have other convex-

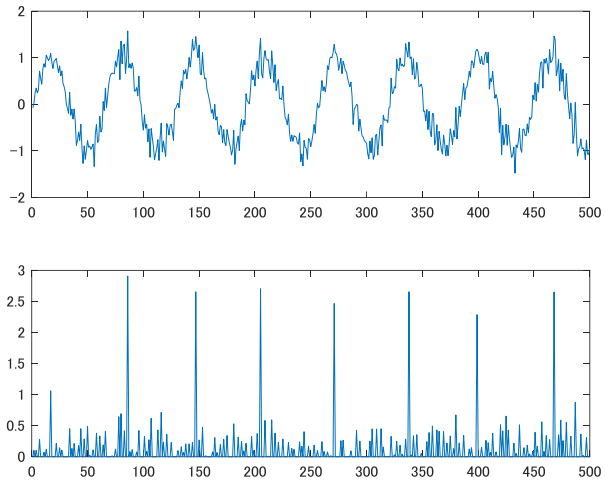


Figure 14: Convex amplitude of noisy sine wave

(2) Transient data with spike

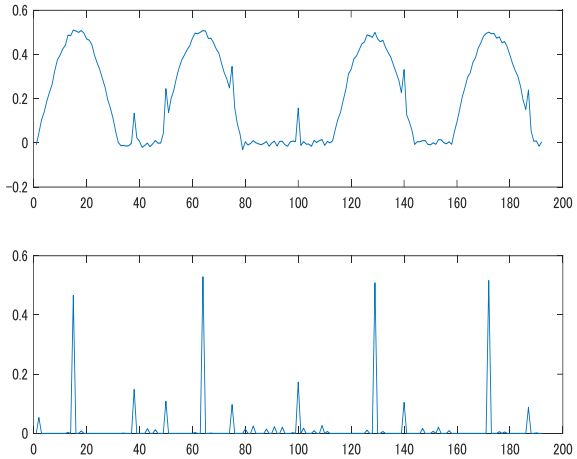


Figure 15: Transient data with spike

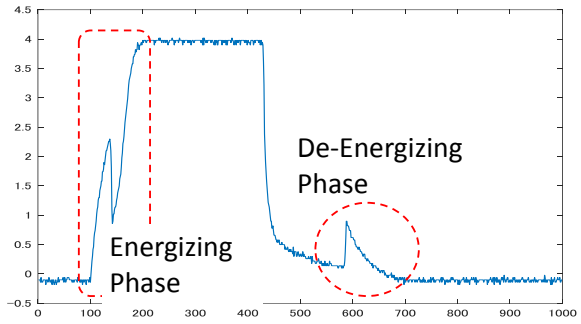


Figure 16: An example of a Space Shuttle Marotta Valve time series that are annotated as normal

shaped patterns except for a normal energizing phase. Figure 20 is an enlarged view of Figure 19 during times from 95 to 180. Similarly, Figure 21 is an enlarged view of Figure 19 during times from 3101 to 3186. The pattern shown by Figure 19 is a continuously convex-shaped so that it is different from normal energizing phase pattern such as Figure 21.

In Figure 22, the top graph shows amplitudes that are larger than 0.46 and smaller than 0.9. The red circles and red dotted lines have the same meaning as in Figure 17. Each

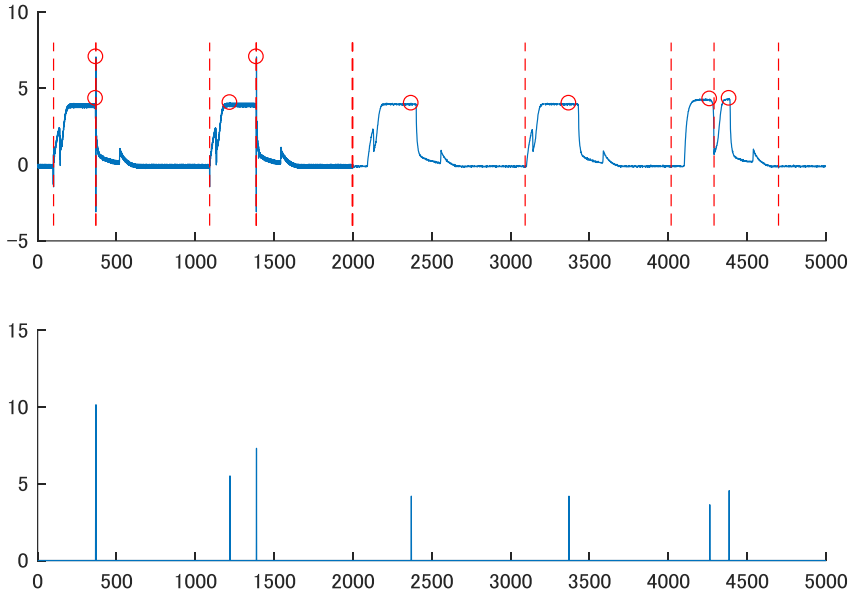


Figure 17: Convex-shaped patterns whose amplitudes are larger than 3

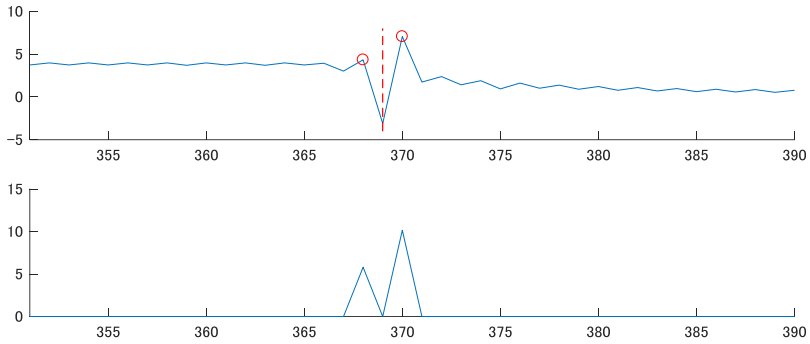


Figure 18: Enlarged view from 351 to 390 in Figure 17

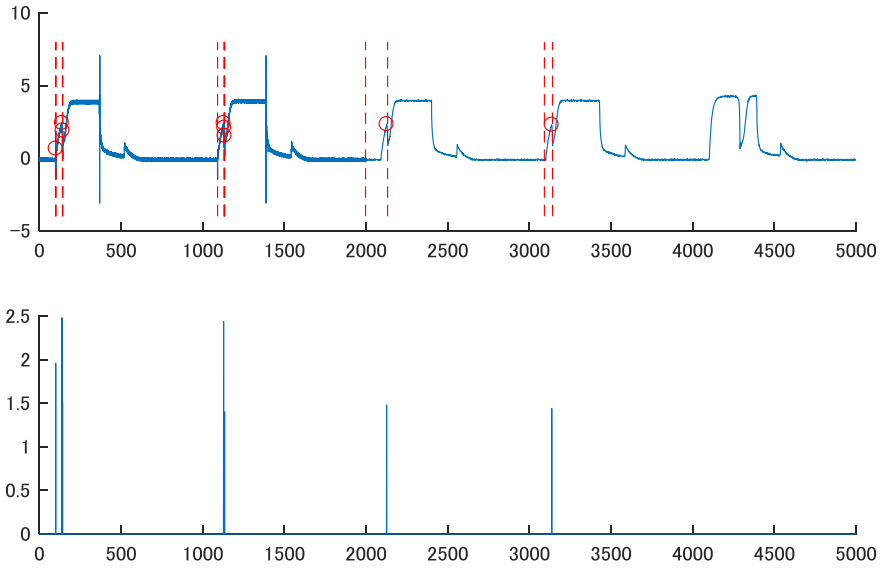


Figure 19: Convex-shaped patterns whose amplitudes are larger than 1.2 and smaller than 3.0

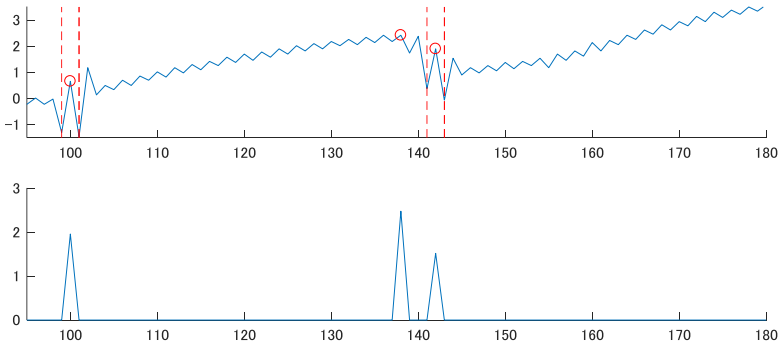


Figure 20: Enlarged view from 95 to 180 in Figure 19 (abnormal)

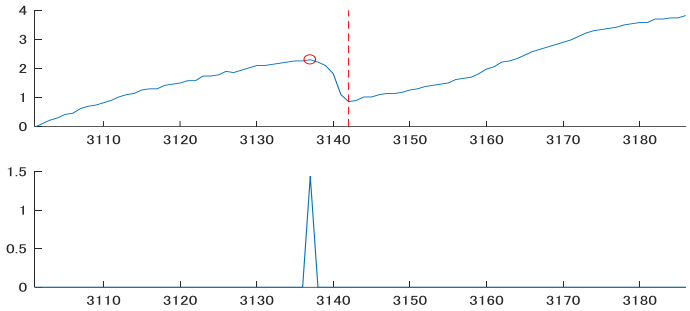


Figure 21: Enlarged view from 3101 to 3186 in Figure 19 (normal)

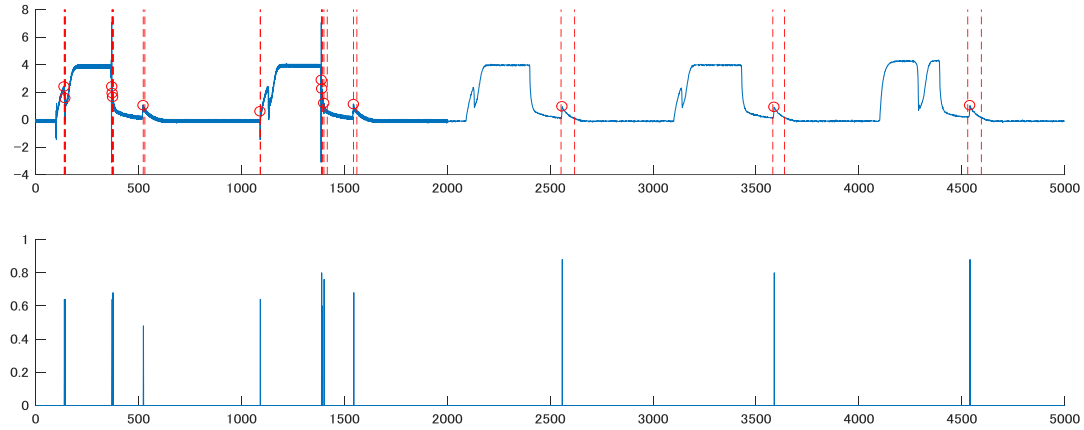


Figure 22: Convex-shaped patterns whose amplitudes are larger than 0.46 and smaller than 0.9

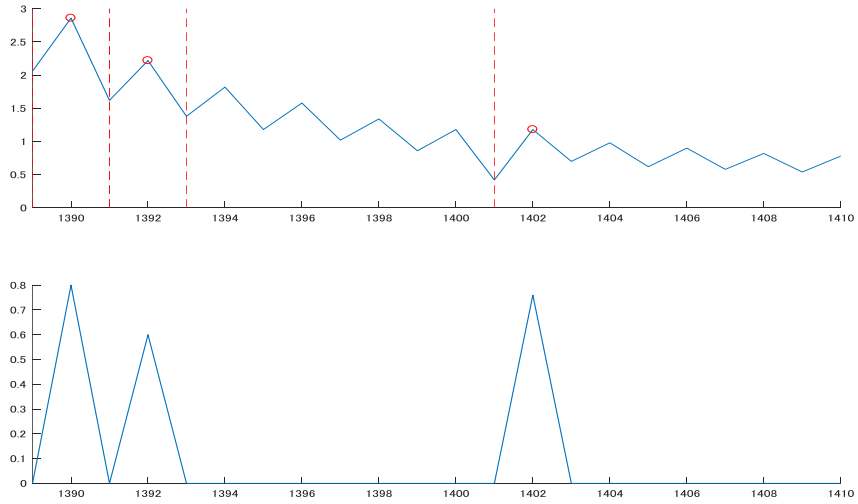


Figure 23: Enlarged view from 1389 to 1410 in Figure 22 (abnormal)

Table 1: Environment for evaluation

CPU	Intel® Core™ i5-6600 CPU 3.30GHz
Memory(RAM)	32GB®
OS	Windows 7 Professional
Language	MATLAB

de-energizing phase patterns is extracted from each cycle, but the first and second cycles have other convex-shaped patterns except in de-energizing phase patterns. Fig. 23 is an enlarged view of Fig. 17 during times from 1389 to 1410. It shows that there are 3 convex-shaped patterns whose amplitudes are the same as a normal de-energizing phase pattern.

Those experimental results above showed that amplitude function can extract anomalous convex-shaped patterns by giving the amplitude value that we wanted to find.

4.2 Performance

This section shows the dependency of the computational time of our algorithm on the length of time series, for various sensor data sets. Table 1 shows the environment for evaluation.

Figure 24 shows the trend graphs of experimental time series. Data labels "ECG", "Power", "Respiration" and "Valve" are electrocardiogram qtdb/se102, Dutch power demand dataset, a patient's respiration nprs44, and space shuttle Marotta Valve TEK16 in the UCR time series classification archive [13], respectively. NoisySine is the simulated time series shown in Figure 14.

The lengths of experimental time series are between 0 and 20000, at a step increase of 2000. The length of TEK16 is shorter than 20000, so we obtained a time series of length 20000 by concatenating the original time series multiple times.

Figure 25 shows the execution times for the 5 time series data sets. Each time is an average of the times of 10 trials, in order to reduce the effect of variance. The figure shows that the execution times depend on the behavior of data, but they are linear in n, where n is the length of the time series. The lengths of those time series are from 0 to 20000 at an increment of 2000 samples. The execution times for the length of 10000 are between 0.01 and 0.04 sec. It means that our algorithm can process one data point per between 10^{-6} and 4×10^{-6}. In other words, our algorithm enables real-time processing of time series with sampling periods between 1 and 4 microseconds.

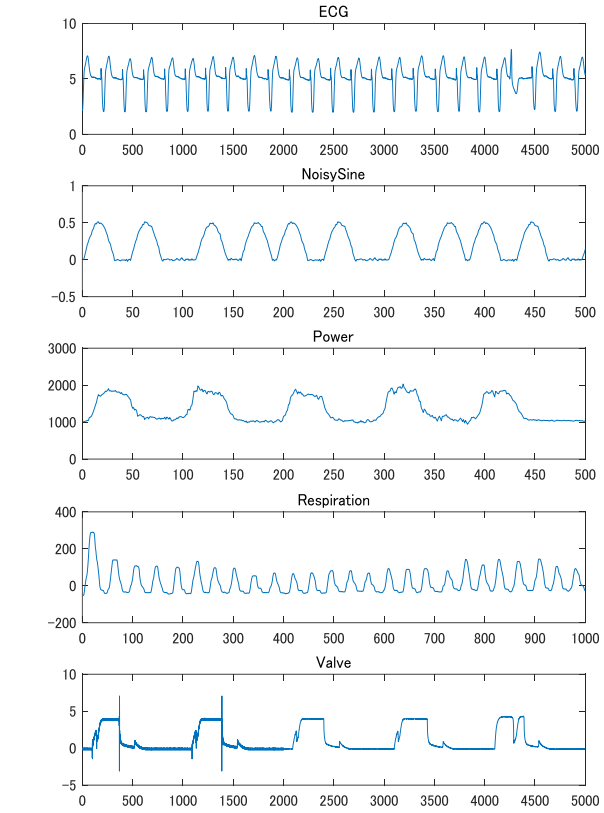


Figure 24: The trend graphs of experimental time series

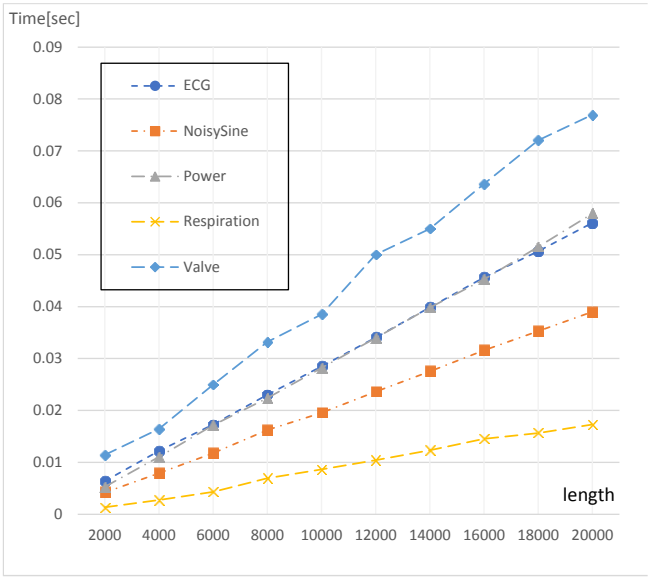


Figure 25: The execution times for experimental time series

5 CONCLUSIONS

We have proposed a new parameter-free online algorithm that calculates the amplitude of a maximal convex curve for extracting convex-shaped spikes in transient sensor data. We also showed that the computational complexity of our algorithm is O(n), where n is the length of input time series, and it enables real-time processing with sampling period ranging from 1 to 4 microseconds.

In future work, we will apply our algorithm to the following problems:

- Segmenting time series in order to identify operational regimes of equipment.
- Anomaly detection for equipment condition monitoring.

This work is supported by JSPS KAKENHI Grant Number 17K00161.

REFERENCES

[1] J. Zheng, D. Simplot-Ryl, C. Bisdikian, H. T. Mouftah: “The Internet of Things [Guest Editorial]”, Communications Magazine, IEEE, Vol.49, No.11, pp.30-31 (2011).

[2] M. Imamura, D. Nikovski, Z. Sahinoglu, M. Jones: "A Survey on Machine Learning for Equipment Condition Monitoring Using Sensor Big Data", IIEEJ Transactions on Image Electronics and Visual Computing Vol.2 No.2, pp. 112-121 (2014).

[3] E. Fink, B. P. Kevin: "Indexing of Compressed Time series", DATA MINING IN TIME SERIES DATABASES, World Scientific, pp. 43-65 (2004).

[4] M. Imamura, T. Nakamura, H. Shibata, N. Hirai, S. Kitagami, T. Munaka: "Leg Vibration Analysis for Time Series", IPSJ Journal, vol. 57, No.4, pp.1303-1318 (2016). (in Japanese).

[5] M. Imamura, D. Nikovski, M. Jones: "An Anomaly Detection System for Equipment Condition Monitoring", International Journal of Informatics Society (IJIS) VOL.8, NO.3, pp. 161-169 (2016).

[6] M. Imamura, H. Watanabe, D. Nikovski, A. Farahmand: "Online magnitude fluctuation analysis for anomaly detection", 10th International Workshop on Informatics (IWIN), p.269-275 (2017).

[7] P. Patel, E. Keogh, J. Lin and S. Lonardi, "Mining motifs in massive time series databases," 2002 IEEE International Conference on Data Mining, 2002. Proceedings., 2002, pp. 370-377.

[8] Y. Zhu et al., "Matrix Profile II: Exploiting a Novel Algorithm and GPUs to Break the One Hundred Million Barrier for Time Series Motifs and Joins," 2016 IEEE 16th International Conference on Data Mining (ICDM), Barcelona, 2016, pp. 739-748.

[9] E. Keogh, J. Lin, A. Fu: "HOT SAX: finding the most unusual time series subsequence: algorithms and applications". The Fifth IEEE international conference on data mining, pp. 226–233, www.cs.ucr.edu/eamonn/discords/ (2005).

[10] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh : "Querying and Mining of Time Series Data: Experimental Comparison of Representations and Distance Measures", VLDB 2008, pp.1542-1552 (2008).

[11] G. Box, P. Edward and G. Jenkins: Time Series Analysis, Forecasting and Control (fifth edition), WILEY, p. 54-67. (2016).

[12] E. Keogh, S. Chu, D. Hart and M. Pazzani, "An online algorithm for segmenting time series," Proceedings 2001 IEEE International Conference on Data Mining, San Jose, CA, 2001, pp. 289-296.

[13] Y. Chen, E. Keogh, B. Hu, and N. Begum, A. Bagnall, A. Mueen, and G. Batista: The UCR Time Series Classification Archive, (2015). <available from (http://www.cs.ucr.edu/~eamonn/time_series_data/>

(Received December 15, 2017)
(Revised February ,27 2018)

Makoto Imamura He received the M.E. degree from Kyoto University of Applied Mathematics and Physics in 1986 and the Ph.D. degree from Osaka University of the Information Science and Technology in 2008. From 1986 to 2016, he worked for Mitsubishi Electric Corp. In April 2016, he has moved to the school of Information and Telecommunication Engineering at Tokai University as a Professor. His research interests include machine learning, model-based design and their applications in prognostics and health management and cyber-physical system.

Junji Tsuda He received a B.E. degree from Tokai University, Japan in 2017. He is a master course student of the school of Information and Telecommunication Engineering at Tokai University. His research interests include IoT systems and data analytics.

Daniel Nikovski He received a PhD in robotics from Carnegie Mellon University in 2002, and is presently a senior member of research staff and group manager of the Data Analytics group at Mitsubishi Electric Research Laboratories. He has worked on probabilistic methods for reasoning, learning, planning, and scheduling, and their applications to hard industrial problems. He is a member of IEEE.

Masato Tsuru He received M.E. degree from Kyoto University, Japan in 1985, and then received his D.E. degree from Kyushu Institute of Technology, Japan in 2002. He worked at Oki Electric Industry Co., Ltd., Nagasaki University, and Japan Telecom Information Service Co., Ltd. In 2003, he moved to the Department of Computer Science and Electronics, Kyushu Institute of Technology as an Associate Professor, and then has been a Professor in the same department since April 2006. His research interests include performance measurement, modeling, and management of computer communication networks. He is a member of the ACM, IEEE, IEICE, and IPSJ.

Regular Paper

Best-Time Estimation Method using Information Interpolation for Sightseeing Spots

Masaki Endo*, Masaharu Hirota**, Shigeyoshi Ohno*, and Hiroshi Ishikawa***

*Division of Core Manufacturing, Polytechnic University, Japan
{endou, ohno}@uitec.ac.jp
**Department of Information Science, Okayama University of Science, Japan
hirota@mis.ous.ac.jp
***Graduate School of System Design, Tokyo Metropolitan University, Japan
ishikawa-hiroshi@tmu.ac.jp

Abstract – With the spread of SNS, many data are transmitted in real time. Some data with position information are included in these data. A benefit of analysis using data with position information is that they can extract an event accurately from a target area to be analyzed. However, because data with position information are scarce among all social media data, the amount to analyze is insufficient in almost all areas. In other words, most events cannot be fully extracted. Therefore, efficient analytical methods must be devised for accurate extraction of events with position information, even in areas with few data. For this study, we estimate the time of biological season observation in particular areas and sightseeing spots by information interpolation using tweet location information. Herein, we explain the analysis results obtained using interpolation of information related to cherry blossoms and autumn leaves as an example.

Keywords: information interpolation, phenological observation, trend estimation, Twitter

1 INTRODUCTION

In recent years, sightseeing has come to be regarded as an extremely important growth field to revive Japan's powerful economy [1]. Tourism, with its strong economic ripple effect, is expected to benefit regional revitalization and employment opportunities through accommodation of world tourism demand, including that from rapidly growing Asia. In addition, people around the world can discover and disseminate the charm of Japan and can promote mutual understanding among countries.

In addition to the promotion of tourism to Japan, the progress of domestic travel is important. It is necessary for a nation with modern tourism to build a community society by which regional economies are well-served, attracting tourists widely. Moreover, it is necessary to cultivate tourist areas full of individuality and to promote their charm positively.

According to a survey study of information technology (IT) tourism and services to attract customers [2] by the Ministry of Economy, Trade and Industry (METI), tourists want real-time information and local unique seasonal information posted on websites. Current websites provide similar information in the form of guidebooks. Nevertheless, information of that medium is not frequently updated. Because each local government, tourism association, and travel company independently provides information about local travel destinations, it is difficult for tourists to collect information for "now" tourist spots. Therefore, the travel industry demands that current, useful, real-world information be provided for travelers by capturing the change of information in accordance with the season and time zone of the tourism region.

We consider a method to estimate the best time for phenological observations for tourism such as the best time for viewing cherry blossoms and autumn leaves in each region by particularly addressing phenology observations assumed for "now" in the real world. We define "now" information as that intended for tourism and disaster prevention required by travelers during travel, such as best flower-viewing times, festivals, and locally heavy rains.

Tourist information for best times requires a peak period: the best time is not a period after or before falling flowers, but a period that is best to view blooming flowers. Furthermore, the best times differ among regions and locations. Therefore, it is necessary to estimate the best time of phenological observation for particular regions and locations. Estimating best-time viewing requires collection of large amounts of information with real-time properties.

For this study, we use Twitter data obtained from many users throughout Japan. Twitter [3], a typical microblogging service, has some geotagged tweets that include position information sent in Japan. We use the data to ascertain the best time (peak period) in biological season observation by region. We proposed a low-cost estimation method [4] by which prefectures and municipalities showing a certain number of tweets with geotags can be estimated with a relevance rate of about 80% compared to the flowering day / full bloom day of cherry blossoms observed by the Japan Meteorological Agency. The geotagged tweets that are used with this method are useful as social indicators that reflect the real world situation. They are a useful resource supporting a real-time regional tourist information system in the tourism field. Therefore, our proposed method might be an effective means of estimating the best time to view events other than biological seasonal observations.

Nevertheless, geotagged tweets are extremely few among all tweets. Therefore, a difficulty exists that the data are insufficient for analysis with finer granularity. For this reason, it is necessary to improve the method of interpolating the information of geotagged tweets to conduct further detailed

analyses in areas such as sightseeing spots. For this research, we propose a method of estimating the best time for a particular tourist spot by performing information interpolation based on amounts of regional information. This paper presents results of verification by experimentation using cherry blossoms and autumn leaves.

The remainder of the paper is organized as follows. Chapter 2 presents earlier research related to this topic. Chapter 3 describes our proposed method for estimating the best time for phenological observations by information interpolation using regional amounts. Chapter 4 explains experimentally obtained results for our proposed method and a discussion of the results. Chapter 5 summarizes the contributions and future work.

2 RELATED WORK

Along with rising SNS popularity, real-time information has increased. Analysis using real-time data has become possible. Many studies have examined efficient methods for analyzing large amounts of digital data. Some studies have been conducted to predict real world phenomena using large amounts of social data.

Phithakkitnukoon et al. [5] analyzed the behavior of travelers such as departure place, destination, and traveling means on a personal level in detail based on massive mobile phone GPS location records. Mislove et al. [6] developed a system that infers a Twitter user's feelings from Twitter text and visualizes changes of emotion in space–time. After research to detect events such as earthquakes and typhoons, Sakaki et al. [7] proposed a method to estimate real-time events from Twitter tweets. Cheng et al. [8] estimated Twitter users' geographical positions at the time of their contributions, without the use of geotags, by devoting attention to the geographical locality of words from text information in Twitter-posted articles. Although various studies have analyzed spatiotemporal data, research to estimate the viewing period using information interpolation is a new field.

3 OUR PROPOSED METHOD

This section presents a description of an analytical method for target data collection. It presents best-time estimation to obtain a guide for phenological change from Twitter in Japan. Our proposal is portrayed in Fig. 1.

We describe the best-time estimation method of organisms by analysis using a moving average method applied to geotagged tweets that include organism names. Section 3.1 describes how to collect geotagged tweets to be analyzed, whereas 3.2 describes preprocessing for conducting analysis, and 3.3 describes the best-time estimation method. In our proposed method up to now, the number of geotagged tweets has been small. It was possible to estimate the best time in a prefecture unit or municipality, but we were unable to analyze fine grain size. Therefore, using the method with information interpolation proposed in this paper, it is possible to estimate the best time to visit sightseeing spots with finer granularity. Section 3.4 presents an explanation of the information interpolation method, whereas 3.5 presents the output of the estimation result.

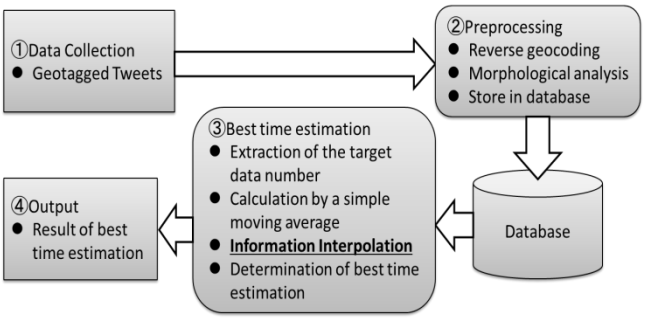


Figure 1: Our proposal summary.

3.1 Data Collection

This section presents a description of the Method of (1) data collection presented in Fig. 1. Geotagged tweets sent from Twitter are a collection target. The range of geotagged tweets includes the Japanese archipelago (120.0°E ≤ longitude ≤ 154.0°E and 20.0°N ≤ latitude ≤ 47.0°N) as the collection target. Collection of these data was done using a streaming API [9] provided by Twitter Inc.

Next, we explain the number of collected data. According to a report presented by Hashimoto et al. [10], among all tweets originating in Japan, about 0.18% are geotagged tweets: they are rare among all data. However, the geotagged tweets we collected are an average of 500 thousand tweets per day. We used about 250 million geotagged tweets from 2015/2/17 through 2017/5/13. We calculated the best time for flower viewing, as estimated using the processing described in the following sections using these data.

3.2 Preprocessing

This section presents a description of the method of (2) preprocessing presented in Fig. 1. Preprocessing includes reverse geocoding and morphological analysis, as well as database storage for data collected through the processing described in Section 3.1.

From latitude and longitude information in the individually collected tweets, reverse geocoding identified prefectures and municipalities by town name. We use a simple reverse geocoding service [11] that is available from the National Agriculture and Food Research Organization in this process: e.g., (latitude, longitude) = (35.7384446°N, 139.460910°E) by reverse geocoding becomes (Tokyo, Kodaira City, Ogawanishi-cho 2-chome).

Morphological analysis divides the collected geo-tagged tweet morphemes. We use the "Mecab" morphological analyzer [12]. By way of example, "桜は美しいです" (in English "Cherry blossoms are beautiful.")" is divided into "(桜 / noun), (は / particle), (美しい / adjective), (です / auxiliary verb), and (。 / symbol)".

Preprocessing accomplishes the necessary data storage for the best-time viewing, as estimated based on results of the processing of the data collection, reverse geocoding, and morphological analysis. Data used for this study were the tweet ID, tweet post time, tweet text, morphological analysis result, latitude, and longitude.

3.3 Estimating Best-Time Viewing

This section presents a description of the method of (3) best-time estimation presented in Fig. 1. Our method for estimating best-time viewing processes the target number of extracted data and calculates a simple moving average, yielding an inference of the best time to view the flowers. The method defines a word related to the best-time viewing, estimated as the target word. The target word is a word including Chinese characters, hiragana, and katakana, which represents an organism name and seasonal change.

Next, we describe the simple moving average calculation, which uses a moving average of the standard of the best-time viewing judgment. It calculates a simple moving average on a daily basis using aggregate data by the target number of data extraction described above. Fig. 2 presents an overview of the simple moving average of the number of days.

We calculate the simple moving average in formula (1) using the number of data going back to the past from the day before the estimated date of the best-time viewing.

$$X(Y) = \frac{P_1+P_2+\cdots+P_Y}{Y} \quad (1)$$

$X(Y)$: Y day moving average
P_n: Number of data of n days ago
Y: Calculation target period

The standard lengths of time we used for the simple moving average were a seven-day moving average and one-year moving average. A seven-day moving average is based on one week because tweets tend to be more numerous on weekends than on weekdays. In addition, phenological observations, which are the current experiment subjects, are targeting "events" that happen once a year (e.g., appreciation of cherry blossoms, viewing of autumn leaves, moon viewing). Such events are therefore based on a one-year moving average.

Next, we describe a simple moving average of the number of days specified for each organism to compare the seven-day moving average and a one-year moving average. In this study, the best time to view the period varies depending on the specified organism, the individual organism, and the number of days from the biological period.

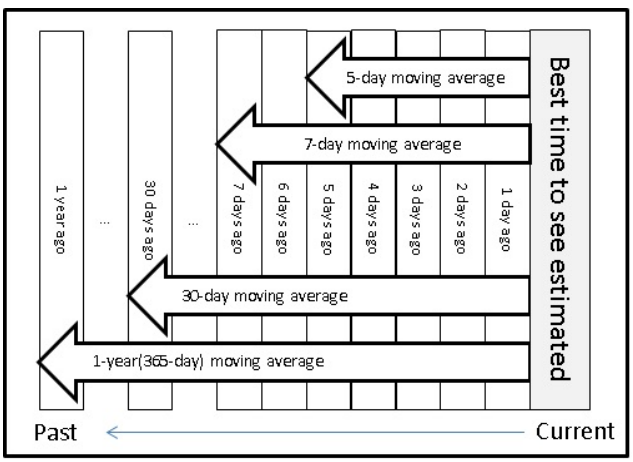


Figure 2: Number of days simple moving average.

As an example, we describe cherry blossoms. The Japan Meteorological Agency [13] carries out phenological observations of "Sakura," which yields two output items of the flowering date and the full bloom date observation target. The "Sakura flowering date" [14] is the first day on which blooming of 5–6 or more wheels of flowers occur on a specimen tree. The "Sakura in full bloom date" is the first day on which about 80% or more of the buds in the specimen tree are open. In addition, "Sakura" is the number of days from general flowering until full bloom: about five days. Therefore, "Sakura" in this study uses a five-day moving average as the standard.

Next, we describe an estimated judgment of the best time for viewing, as calculated using the simple moving average (seven-day moving average, one-year moving average, and another biological moving average). It specifies the two conditions as a condition of an estimated decision for the best time for viewing.

Condition 1 uses the number of tweets a day prior and a one-year moving average. Condition 1 is assumed to be satisfied when the number of tweets a day prior exceeds the one-year moving average, as shown in Formula 2.

Condition 2 uses a seven-day moving average and a biological moving average. The biological moving average varies depending on the organism that is estimated. It is five days in the case of cherry blossoms. For autumn leaves, it is 30 days. Therefore, in equation 3, we compare the seven-day moving average with the biological moving average, letting A be the long number of days, and letting B be the short number of days. In the case of estimation of cherry blossoms, A is 7 days; B is 5 days. For autumn leaves, A is 30 days; B is 7 days. Then we evaluate the moving average of A and B as shown in Equation 3. Furthermore, if the day on which Equation 3 holds lasts more than half of the number of days in A, Condition 2 is satisfied. In the case of cherry blossoms, A is 5 days. Therefore, condition 2 requires continuation for more than 3 days.

$$P_1 \geqq X(365) \quad (2)$$
$$X(A) \geqq X(B) \quad (3)$$

Finally, an estimate is produced using conditions 1 and 2. Using the proposed method, a day satisfying both condition 1 and condition 2 is estimated as best-time viewing.

3.4 Information Interpolation Method

Herein, the information interpolation method will be described. Conventionally, we estimated the best time by application of the estimation method shown in the following estimated judgment using the moving average value described above. As a result, for analysis of a wide area such as a prefecture unit, the recall rate can be estimated as about 80%. However, with an estimate of granularity such as by sightseeing spots, an inability to estimate the viewing period because of a lack of data is a problem. Therefore, in this paper, we propose a method of using regional quantities that newly use information interpolation to compensate for the lack of data volume. The proposed method uses the result of reverse geocoding performed during preprocessing in the

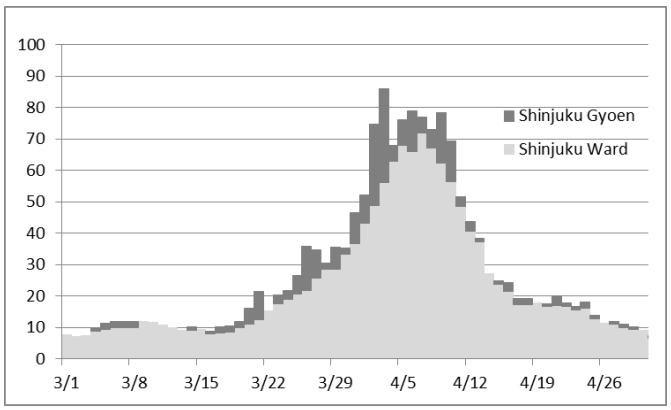


Figure 3: Example of information interpolation.

previous section. Tweets that were judged as the same municipality by reverse geocoding are totaled for each day by city, town, or village. Then, considering the characteristic by which the tweets move on a weekly basis, we obtain a seven-day moving average and set the seven-day moving average of the municipalities as the regional quantity of each region. To estimate the best time for viewing, use the value obtained by adding the regional quantity of the municipality where the sightseeing spot is located to the tweet amount of the sightseeing spot to be estimated.

As an example, we describe Shinjuku Gyoen, which is a cherry blossom sightseeing spot, and Shinjuku Ward, within which the spot is located. The dark gray area in Fig. 3 shows cherry blossom tweets related to Shinjuku Gyoen. An estimate might not be possible with just the number of tweets related to each sightseeing spot. For this reason, interpolation is performed using the seven-day moving average of tweets about cherry blossoms in Shinjuku Ward indicated by light gray in the city unit within which each sightseeing spot is located. In the proposed method, the best estimate is made using the number of tweets related to cherry blossoms at each tourist spot and the sum of the seven-day moving average of city unit.

However, if no tweet is related to sightseeing spots with the proposed method, estimation results of city unit are applied, so there are cases in which there is no difference depending on sightseeing spots in the same area. In the preliminary experiments, we succeeded in ascertaining the difference from nearby sightseeing spots if there are small tweets in the sightseeing spots.

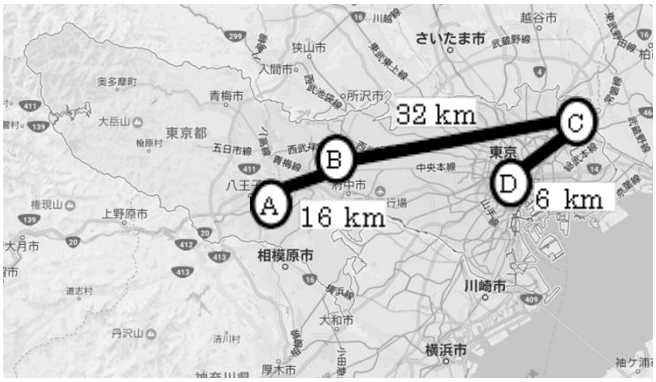


Figure 4: Position of target area.

3.5 Output

This section presents a description of the method of (4) output presented in Fig. 1. Output can be visualized using a best-time viewing result, as estimated by processing explained in the previous section. A time-series graph presents the inferred results for best-time viewing. The graph presents the number of data and the date, respectively, on the vertical axis and the horizontal axis. We are striving to develop useful visualization techniques for travelers.

4 EXPERIMENTS

This chapter presents a description of the experiment to infer the best time to view cherry blossoms and autumn leaves for the proposed method described in Chapter 3. Section 4.1 describes the dataset used for optimal time reasoning. As an estimation result by sightseeing spot, section 4.2 presents the estimation result without using information interpolation, with the best estimation result obtained using information interpolation in section 4.3. Section 4.4 presents a comparison of the experimentally obtained results in Section 4.2 and Section 4.3.

4.1 Dataset

Datasets used for this experiment were collected using streaming API, as described for data collection in Section 3.1. Data are geotagged tweets from Japan during 2015/2/17 – 2017/8/31. The data include about 280 million items.

The estimation experiment to ascertain the best-time viewing of cherry blossoms uses the target word “cherry blossom,” which can be written as“桜” and“さくら”and “サクラ”in Japanese. For the experiment of autumn leaves, the target words are“紅葉,”“黄葉,”“コウヨウ,”“こうよう,”“モミジ,”and“もみじ”. We analyzed tweets that included a target word in the tweet text.

The following two experiments were conducted. The first is an experiment using the number of tweets including the target word and the sightseeing spot name without information interpolation. The second is an experiment using information interpolation. We use these datasets to estimate the optimum time for the sightseeing spots in Tokyo by experiments without information interpolation, (shown in Section 4.2) and experiments using information interpolation (shown in Section 4.3).

The subjects of the experiment were set as tourist spots in Tokyo. This report describes“Takao Mountain,”“Showa Memorial Park,”“Shinjuku Gyoen,”and“Rikugien.” Fig. 4 portrays the target area: A, B, C, and D in the figure respectively denote“Takao Mountain,”“Showa Memorial Park,”“Rikugien,”and“Shinjuku Gyoen.” A and B are separated by about 16 km straight-line distance. B and C are about 32 km apart. C and D are about 6 km apart.

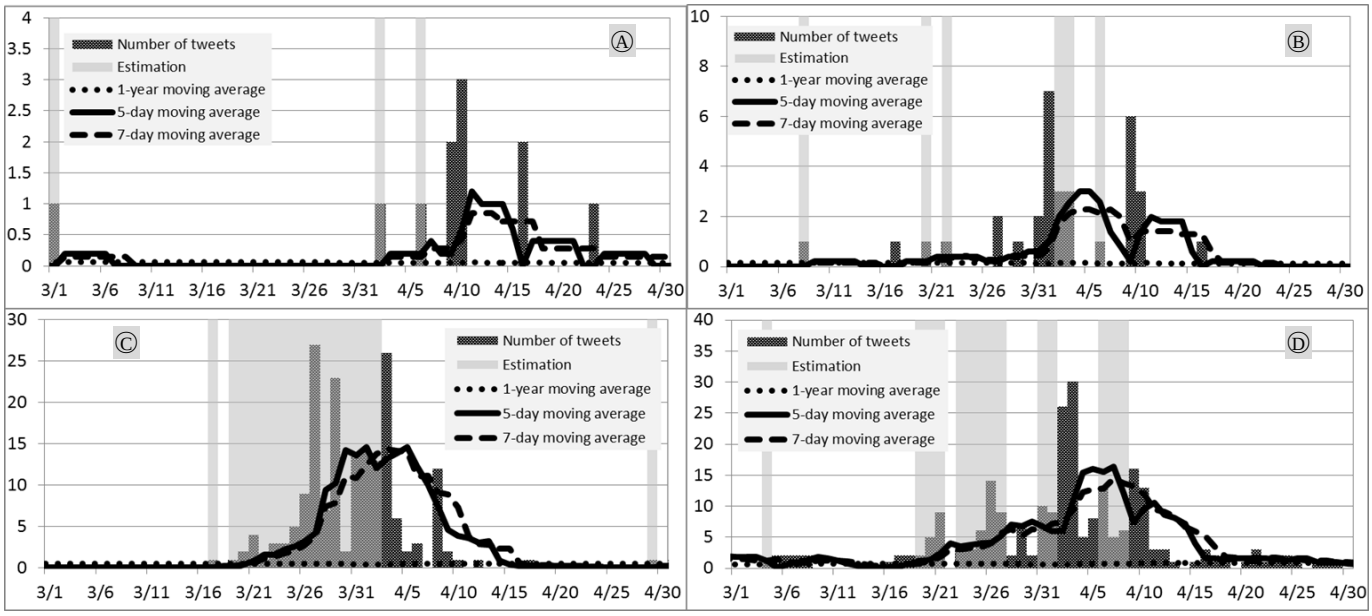


Figure 5: Experimental results obtained using tweets including the target word and the tourist spot name without interpolation (Cherry blossom).

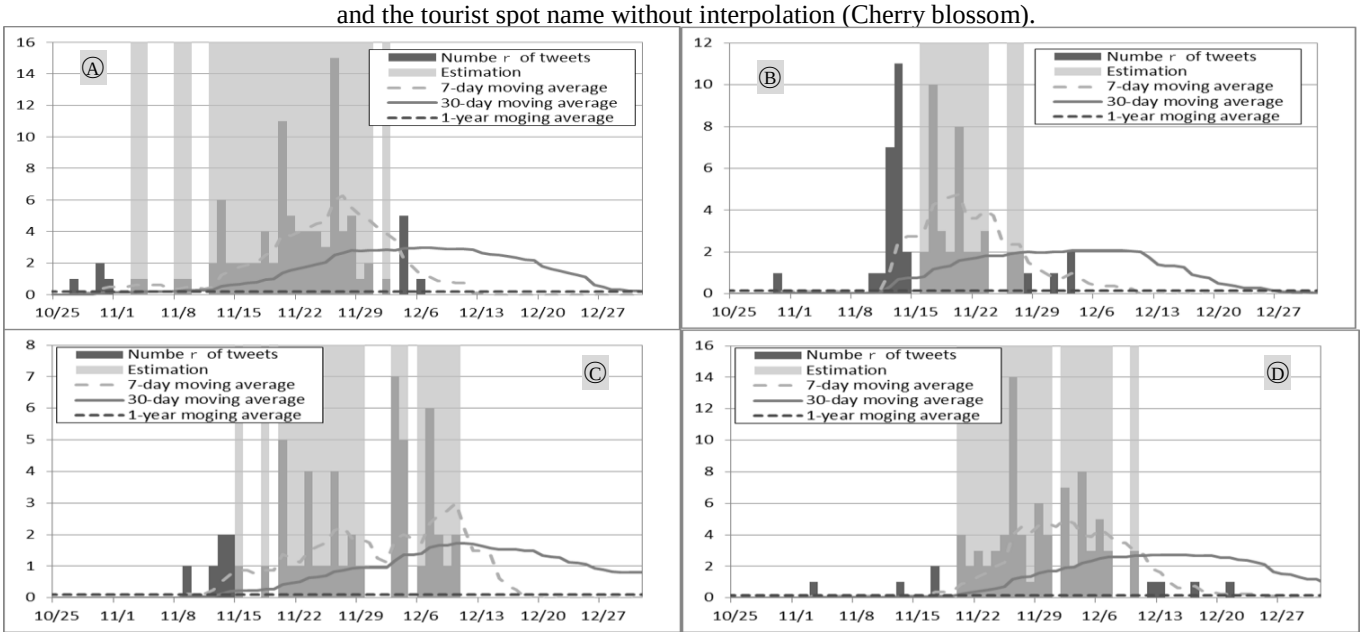


Figure 6: Experimental results obtained using tweets including the target word and the tourist spot name without interpolation (Autumn leaves)

4.2 Estimation Experiment for Best-Time Viewing without Information Interpolation

In this section, we present experimentally obtained results from estimating the best time without using information interpolation from tweets containing a target word and sightseeing spot name. Figure 5 presents results for the estimated best-time viewing in 2016 using the target word 'cherry blossoms' in the target tourist spots. The dark gray bar in the figure represents the number of tweets. The light gray part represents best-time viewing as determined using the proposed method. In addition, the solid line shows a five-day moving average. The dashed line shows a seven-day moving average. The dotted line shows a one-year moving average.

At tourist spots targeted for the experiment in 2016, as portrayed in Fig. 5, many data were obtained for C and D. The maximum number of tweets per day was about 30. These results confirmed that some estimation can be done using near-site estimation without interpolation. However, best-time viewing cannot be done in A and B because of the very small number of tweets.

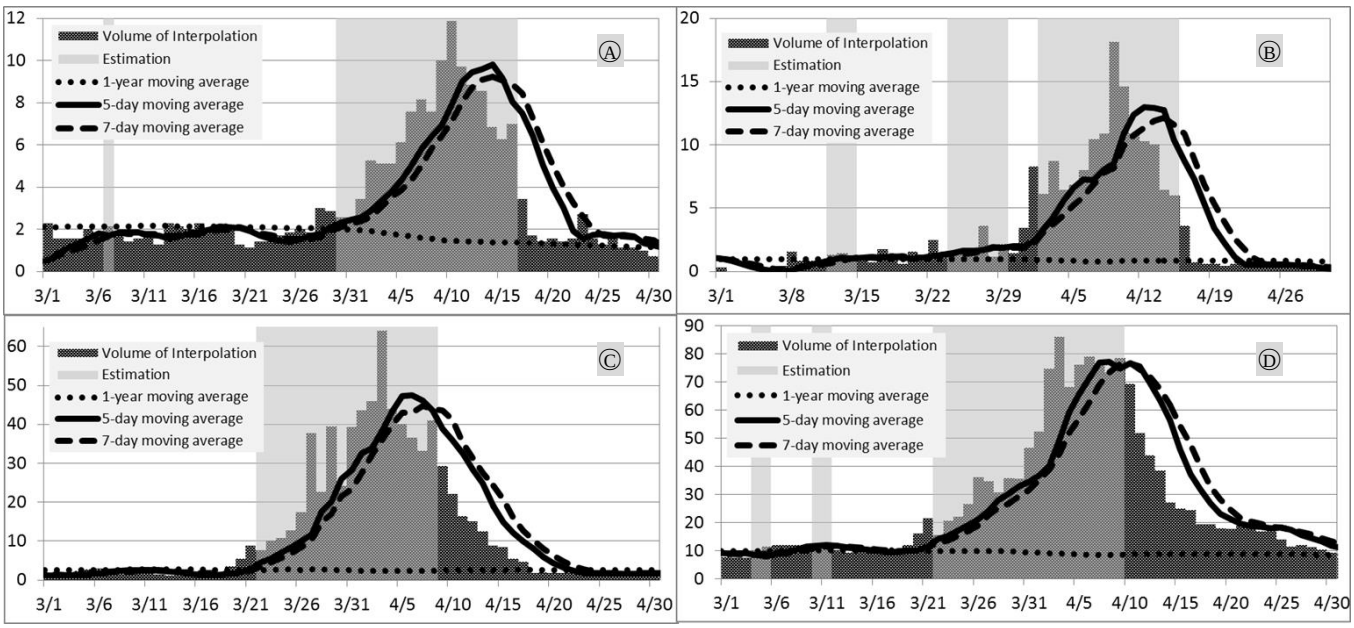


Figure 7: Experimental results obtained using tweets including the target word and the tourist spot name by interpolation

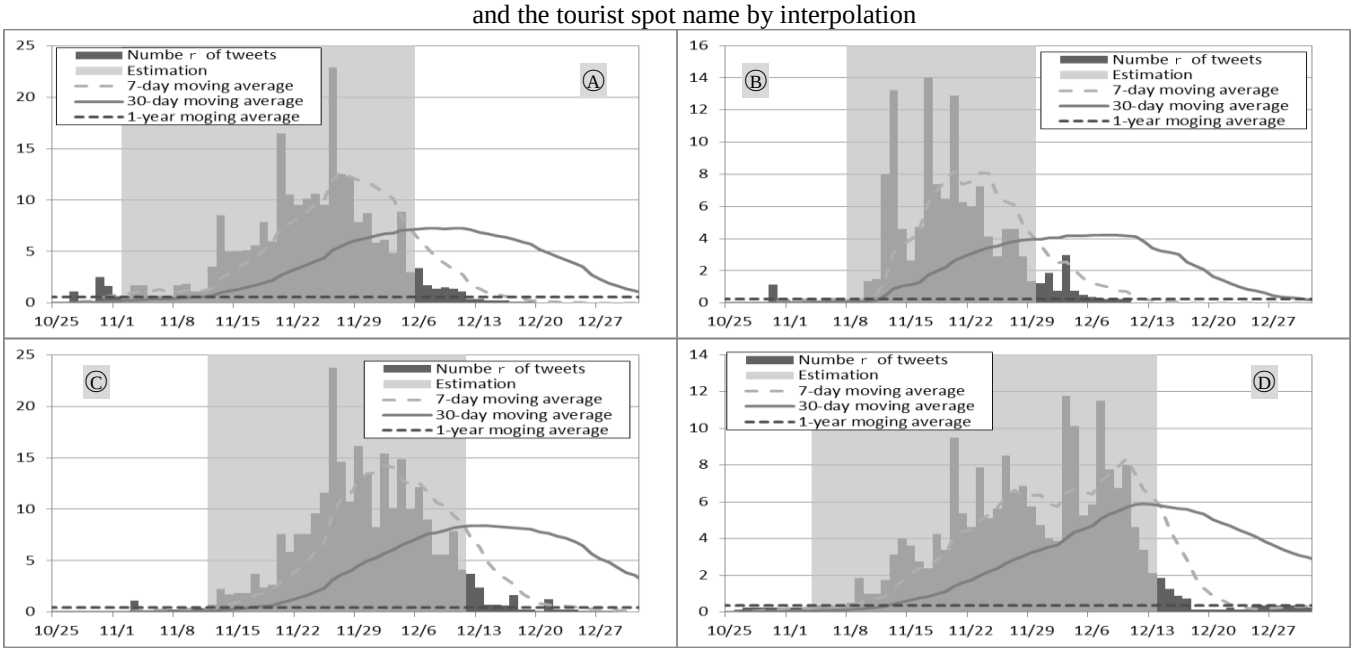


Figure 8: Experimental results obtained using tweets including the target word and the tourist spot name by interpolation

Next, we present experimentally obtained results of autumn leaves. Figure 6 portrays the estimated best time viewing results estimated using the word target word 'autumn leaves' in 2016. The notation in the figure is the same as that in Fig. 5. However, for cherry blossoms, the solid line is the five-day moving average, whereas the autumn leaves use a 30-day moving average. As shown in Fig. 6, the autumn leaves experiment has been estimated for each sightseeing spot because the viewing period is longer than that of cherry blossoms shown in Fig. 5. However, some parts cannot be estimated as continuous periods.

These results clarified that the method we proposed earlier cannot be predictive for detailed areas such as sightseeing spots. This result is attributable to the insufficient information volume.

4.3 Estimation Experiment for Best-Time Viewing with Information Interpolation

This section presents experimentally obtained results of estimation using information interpolation with regional quantities, which is the method proposed in this paper. Figure 7 presents results obtained using information interpolation for cherry blossom estimation. The notation is the same as the notation used in the previous section. Apparently, A and B can produce an estimate using the proposed method by increasing the number of tweets using information interpolation with surrounding tweets. In C and D, there are days that can be determined more accurately by interpolating the number of tweets.

Next, Fig. 8 presents results of information interpolation in autumn leaves estimation. Autumn leaves can be estimated as a continuous period using information interpolation. From this result, it can be inferred that the period estimation can be performed more accurately than without information interpolation.

These results demonstrate the possibility of resolving the difficulty of insufficient information when using sightseeing spot tweet data of the tourist spot area along with interpolation. One can then estimate the peak period for a particular tourist spot.

4.4 Comparing Best Time for Viewing Estimation and Observed Data

Table 1 presents a comparison of experimentally obtained results of estimating the best time using information interpolation. As the table shows, Experiment 1 used co-occurring words in tweets, including the sightseeing spot name coexisting with the target word "Sakura," without using the interpolation shown in 4.3. For Experiment 2, we used interpolation based on the information amount of the area including the tourist spots shown in 4.4. Numerical values in the table are the number of tweets including subject words and co-occurring words in Experiment 1. In Experiment 2, it is the sum of the number of tweets in Experiment 1 and the interpolation value of the regional information amount.

The light gray area in the table presents the date when the satiety prediction was made using the proposed method. In addition, confirming the flowering day and full bloom period of each sightseeing spot using JMA data is difficult.

Table 1: Comparison result in target areas of the best time to see the estimated and the observed data (Cherry blossom)

	Takao Mountain		Showa memorial park		Rikugien		Shinjuku gyoen	
	Exp.1	Exp.2	Exp.1	Exp.2	Exp.1	Exp.2	Exp.1	Exp.2
3/18	0	2.00	0	1.00	0	1.86	2	10.57
3/19	0	2.00	0	0.57	1	3.57	2	11.86
3/20	0	1.29	1	1.57	2	5.43	5	16.00
3/21	0	1.14	0	1.14	4	8.86	9	21.43
3/22	0	1.43	1	2.43	1	7.57	0	15.43
3/23	0	1.43	0	1.43	3	10.00	3	20.43
3/24	0	1.71	0	1.57	3	10.71	3	21.86
3/25	0	1.86	0	1.43	5	12.57	6	26.57
3/26	0	2.00	0	1.71	9	17.43	14	35.86
3/27	0	1.86	2	3.57	27	37.57	9	34.71
3/28	0	3.00	0	1.14	7	22.71	2	30.57
3/29	0	2.86	1	2.14	23	39.43	7	35.57
3/30	0	2.57	0	1.43	2	24.14	2	35.29
3/31	0	2.57	2	3.43	14	39.29	10	46.57
4/1	0	3.43	7	8.29	14	43.57	9	52.14
4/2	1	5.29	3	6.14	13	45.86	26	74.71
4/3	0	5.14	3	8.71	26	64.00	30	86.00
4/4	0	5.14	0	6.43	6	44.00	5	68.00
4/5	0	6.14	0	6.86	2	40.00	8	76.00
4/6	1	7.57	1	8.00	3	36.57	13	79.00
4/7	0	8.14	0	10.43	0	33.14	5	76.86
4/8	0	7.57	0	10.86	12	41.00	6	73.14
4/9	2	10.00	6	18.14	2	29.29	16	78.43
4/10	3	11.86	3	14.57	1	22.00	13	69.29
4/11	0	9.71	0	10.86	0	16.43	3	51.57
4/12	0	8.86	0	10.29	1	15.00	3	43.71
4/13	0	8.57	0	10.00	0	12.43	1	38.29
4/14	0	6.86	0	6.43	0	8.86	0	27.00
4/15	0	6.29	0	6.00	0	8.29	1	24.86
4/16	2	7.00	1	3.57	0	5.43	3	24.43
4/17	0	3.43	0	0.71	1	4.71	2	19.14
4/18	0	1.71	0	0.57	0	1.71	2	19.14
4/19	0	1.43	0	0.57	0	1.86	0	17.86
4/20	0	1.57	0	0.43	0	1.71	1	17.57
4/21	0	1.43	0	0.57	0	1.71	3	20.00
4/22	0	1.57	0	0.57	0	1.71	1	17.71
Precision	0.51	0.77	0.57	0.74	0.95	0.84	0.80	0.82
Recall	0.06	0.58	0.22	0.44	0.39	0.58	0.56	0.83

Nevertheless, this experiment to evaluate SNS data for flowering is valid also for weather forecasting companies [15] and for public service organizations [16] to evaluate optimum viewing times based on services and blogs that are used. Arrows indicating the flowering time can be checked manually at tourist sites. Recall and precision using the observed data and the best time to view estimated results are calculated for each target area for 2016 from 3/1 through 4/30 using formula (4) and formula (5).

$$Precision = \frac{Number\ of\ days\ to\ match\ the\ observed\ data}{Number\ of\ days\ in\ best\ time\ to\ see\ estimated} \quad (4)$$

$$Recall = \frac{Number\ of\ days\ to\ match\ the\ observed\ data}{Number\ of\ days\ of\ observation\ data} \quad (5)$$

We can explain the method using the example of Experiment 1 of Mt. Takao. The arrow portion of the flowering state is confirmed by hand as correct data, 1; the others are 0. In addition, the day estimated as the best time using the proposed method is set to 1; otherwise a day is 0. Furthermore, the percentage of days coinciding during 3/1 to

Table 2: Comparison result in target areas of the best time to see the estimated and the observed data (Autumn leaves)

	Takao mountain		Showa memorial park		Rikugien		Shinjuku gyoen	
	Exp.1	Exp.2	Exp.1	Exp.2	Exp.1	Exp.2	Exp.1	Exp.2
11/1	0	0.625	0	0.125	0	0	0	0.25
11/2	0	0.625	0	0.125	0	0	0	0.25
11/3	1	1.75	0	0.125	1	1.125	0	0.125
11/4	1	1.75	0	0.125	0	0.125	0	0.375
11/5	0	0.75	0	0.125	0	0.125	0	0.375
11/6	0	0.75	0	0.125	0	0.125	0	0.375
11/7	0	0.625	0	0.125	0	0.25	0	0.375
11/8	1	1.75	0	0.25	0	0.25	0	0.5
11/9	1	1.875	0	0.25	0	0.375	1	1.875
11/10	0	1.125	1	1.375	0	0.375	0	1
11/11	0	1.125	1	1.5	0	0.375	0	1
11/12	2	3.5	7	8	0	0.75	1	1.75
11/13	6	8.5	11	13.25	1	2.25	2	3.125
11/14	2	5	2	4.625	0	1.75	2	4
11/15	2	5	0	2.625	0	1.875	1	3.625
11/16	2	5.125	2	4.75	0	1.875	0	2.75
11/17	2	5.625	10	14	2	3.75	0	2.375
11/18	4	7.875	3	7.375	0	2.375	1	4.25
11/19	2	6	2	6.5	0	2.625	0	3.375
11/20	11	16.5	8	12.875	4	7.625	5	9.5
11/21	5	10.5	2	6.25	2	5.875	1	5.375
11/22	4	9.5	2	6	3	7.625	1	4.625
11/23	4	10.125	3	7.25	2	7.625	4	7.875
11/24	4	10.625	0	4.125	3	9.625	1	5.125
11/25	3	9.5	0	2.875	4	11.625	1	5.625
11/26	15	22.875	2	4.625	14	23.75	4	8.5
11/27	4	12.5	2	4.625	4	14.625	1	6.5
11/28	5	12.25	1	2.875	1	10.75	2	6.875
11/29	1	7.875	0	1.375	6	16.125	1	5.75
11/30	2	8.75	0	1.25	4	13.375	0	4.75
12/1	0	5.875	1	1.875	0	8.25	0	4
12/2	1	6.125	0	0.75	7	15.375	0	3.875
12/3	0	4.875	2	3	2	10.125	7	11.75
12/4	5	8.875	0	0.75	8	14.875	5	10.125
12/5	0	3	0	0.5	3	10	0	5.25
12/6	1	3.375	0	0.375	5	12.125	1	5.875
12/7	0	1.75	0	0.25	3	9	6	11.5
12/8	0	1.375	0	0.25	0	5.625	2	7.75
12/9	0	1.5	0	0.25	0	5.625	1	6.75
12/10	0	1.375	0	0.25	3	7.875	2	8
12/11	0	1.125	0	0	0	4.125	0	4.625
12/12	0	0.375	0	0	1	3.75	0	3.375
12/13	0	0.375	0	0	1	2.375	0	2.125
12/14	0	0.25	0	0	0	0.75	0	1.875
12/15	0	0.25	0	0	0	0.75	0	1.25
12/16	0	0.25	0	0	0	0.625	0	0.875
12/17	0	0.125	0	0	1	1.625	0	0.75
12/18	0	0.125	0	0	0	0.25	0	0.125
12/19	0	0.125	0	0	0	0.25	0	0.125
12/20	0	0	0	0	0	0.125	0	0.125
12/21	0	0	0	0	1	1.25	0	0.125
12/22	0	0	0	0	0	0.25	0	0.25
12/23	0	0	0	0	0	0.25	0	0.125
12/24	0	0	0	0	0	0.25	0	0.125
12/25	0	0	0	0	0	0.125	0	0.375
12/26	0	0	0	0	0	0.125	0	0.375
12/27	0	0	0	0	0	0.125	0	0.375
12/28	0	0	0	0	0	0.125	0	0.375
12/29	0	0	0	0	0	0	0	0.375
12/30	0	0	0	0	0	0	0	0.25
12/31	0	0	0	0	0	0	0	0.25
Precision	0.72	0.85	0.87	0.93	0.93	0.87	0.62	0.97
Recall	0.94	1.00	0.56	1.00	0.82	1.00	0.45	0.95

4/30 is shown. In Experiment 1 for Mt. Takao, except during the arrows, they match, but only 4/2 and 4/6 match during the arrow period. The Precision is 0.51 because the number of days matching the observed data is 31 days and the number of days of the best time for viewing is estimated is 61 days. In addition, because Recall is the ratio of matched days during the arrow, the number of days to match the observed data is 2 days and the number of days of observation data is 31 days. Therefore, it is 0.06.

Experimentally obtained results confirmed the tendency by which the relevance ratio and the recall rate became higher in Experiment 2 than in Experiment 1. In addition, A and B, which are at higher altitudes than C and D, exhibited regional features: the best viewing time occurs later. These results confirmed the usefulness of the proposed method for best-time estimation for sightseeing spots using information interpolation along with regional data.

Table 2 presents a comparison of results of experiments for autumn leaves. The notation is the same as that of Table 1. The period was October 1 through December 31, 2016. The accuracy and recall ratio of experiment results obtained using information interpolation improved without information interpolation in each spot. Results confirm the effectiveness of the method proposed in this paper.

5 CONCLUSION

As described herein, to improve the best-time estimation accuracy and thereby enhance tourist information related to phenological observation, we proposed an information interpolation method. The proposed method showed the optimal time to view flowers at sightseeing spots by interpolating information using the seven-day moving average of the number of tweets of municipalities, including those of sightseeing spots. This method can estimate the best time for sightseeing spots with fine granularity, giving predictions in units required for sightseeing.

The results of cherry blossoms and autumn leaves experiments conducted for tourist spots in Tokyo in 2016 using the proposed method confirmed improvement of the estimation accuracy when using information interpolation. The proposed method using information interpolation for tweets related to target word might improve the real-world accuracy of estimating the best times. We confirmed the possibility of applying this proposed method to the estimation of viewpoints and lines of sight in areas and sightseeing spots with few tweets and little location information.

Although the proposed method showed success in interpolation of information and highly accurate estimation, it is necessary as a future task to verify whether the same result is obtainable also in biological seasonal observations other than those for cherry blossoms or autumn leaves. Future studies must also examine automatic extraction of target words and a method to perform future predictions in real time.

ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Numbers 16K00157 and 16K16158, and by a Tokyo Metropolitan University Grant-in-Aid for Research on Priority Areas “Research on Social Big Data.”

REFERENCES

[1] Japan Tourism Agency, “Tourism Nation Promotion Basic Law,” http://www.mlit.go.jp/kankocho/en/kankorikkoku/index.html (1, 2007).

[2] Ministry of Economy, Trade and Industry, “Study of landing type IT tourism and attract customers service,” http://www.meti.go.jp/report/downloadfiles/g70629a01j.pdf (3, 2007) (in Japanese).

[3] Twitter, “Twitter,” https://Twitter.com/ (4, 2014).

[4] M. Endo, Y. Shoji, M. Hirota, S. Ohno, and H. Ishikawa, “On best time estimation method for phenological observations using geotagged tweets,” IWIN2016 (2016).

[5] S. Phithakkitnukoon, T. Teerayut Horanont, A. Witayangkurn, R. Siri, Y. Sekimoto, and R. Shibasaki, “Understanding tourist behavior using large-scale mobile sensing approach: A case study of mobile phone users in Japan,” Pervasive and Mobile Computing (2014).

[6] A. Mislove, S. Lehmann, Y.Y. Shn, J.P. Onnela, and Rosenquist, “Understanding the Demographics of Twitter Users,” Proceeding. Fifth International AAAI Conference on Weblogs and Social Media (ICWSM’11), pp.133-140 (2011).

[7] T. Sakaki, M. Okazaki, and Y. Matsuo, “Earthquake shakes Twitter users: real-time event detection by social sensors,” WWW 2010, pp.851-860 (2010).

[8] Z. Cheng, J. Caverlee, and K. Lee, “You are where you tweet: a content-based approach to geo-locating twitter users,” Proceedings of the 19th ACM International Conference on Information and Knowledge Management (2010).

[9] Twitter Developers, “Twitter Developer official site,” https://dev.twitter.com/ (4, 2014).

[10] Y. Hashimoto and M. Oka, “Statistics of Geo-Tagged Tweets in Urban Areas (<Special Issue>Synthesis and Analysis of Massive Data Flow),” JSAI, Vol. 27, No. 4, pp. 424-431 (2012) (in Japanese).

[11] National Agriculture and Food Research Organization, “Simple reverse geocoding service,” http://www.finds.jp/wsdocs/rgeocode/index.html.ja (4, 2014).

[12] MeCab, “Yet Another Part-of-Speech and Morphological Analyzer,” http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html (9, 2012).

[13] Japan Meteorological Agency, “Disaster prevention information XML format information page,” http://xml.kishou.go.jp/ (12, 2011).

[14] Japan Meteorological Agency, “Observation of Sakura,” http://www.data.jma.go.jp/sakura/data/sakura2012.pdf (3, 2016).

[15] Weathernews Inc., “Sakura information,” http://weathernews.jp/sakura (3, 2016).

[16] Japan Travel and Tourism Association, “Whole country cherry trees,” http://sakura.nihon-kankou.or.jp (4, 2015).

(Received October 30, 2017)
(Revised April ,17 2018)

Masaki Endo earned a B.E. degree from Polytechnic University, Tokyo and graduated from the course of Electrical Engineering and Computer Science, Graduate School of Engineering Polytechnic University. He received an M.E. degree from NIAD-UE, Tokyo. He earned a Ph.D. Degree in Engineering from Tokyo Metropolitan University in 2016. He is currently an Associate Professor of Polytechnic University, Tokyo. His research interests include web services and web mining. He is also a member of DBSJ, NPO STI, IPSJ, and IEICE.

Masaharu Hirota received a Doctor of Informatics degree in 2014 from Shizuoka University. After working for National Institute of Technology, Oita College, he has been working as Associate Professor in Faculty of Informatics, Okayama University of Science from April, 2017. His research interests include photograph, GIS, multimedia, and visualization. He is a member of ACM, DBSJ, and IPSJ.

Shigeyoshi Ohno earned M.Sci. and Dr. Sci. degrees from Kanazawa University and a Dr. Eng. degree from Tokyo Metropolitan University. He is currently a full Professor of Polytechnic University, Tokyo. His research interests include big data and web mining. He is a member of DBSJ, IPSJ, IEICE and JPS.

Hiroshi Ishikawa earned B.S. and Ph.D. degrees in Information Science from The University of Tokyo. After working for Fujitsu Laboratories and becoming a full Professor at Shizuoka University, he became a full Professor at Tokyo Metropolitan University in April, 2013. His research interests include databases, data mining, and social big data. He has published actively in international refereed journals and conferences such as ACM TODS, IEEE TKDE, VLDB, IEEE ICDE, and ACM SIGSPATIAL. He has authored several books: *Social Big Data Mining* (CRC Press). He is a fellow of IPSJ and IEICE and is a member of ACM and IEEE.

Submission Guidance

About IJIS

International Journal of Informatics Society (ISSN 1883-4566) is published in one volume of three issues a year. One should be a member of Informatics Society for the submission of the article at least. A submission article is reviewed at least two reviewer. The online version of the journal is available at the following site: http://www.infsoc.org.

Aims and Scope of Informatics Society

The evolution of informatics heralds a new information society. It provides more convenience to our life. Informatics and technologies have been integrated by various fields. For example, mathematics, linguistics, logics, engineering, and new fields will join it. Especially, we are continuing to maintain an awareness of informatics and communication convergence. Informatics Society is the organization that tries to develop informatics and technologies with this convergence. International Journal of Informatics Society (IJIS) is the journal of Informatics Society.

Areas of interest include, but are not limited to:

- Internet of Things (IoT)
- Smart Cities, Communities, and Spaces
- Big Data, Artificial Intelligence, and Data Science
- Network Systems and Protocols
- Computer Supported Cooperative Work and Groupware
- Security and Privacy in Information Systems
- Intelligent Transportation System
- Distributed Computing
- Multi-media communication
- Information systems
- Mobile computing
- Ubiquitous computing

Instruction to Authors

For detailed instructions please refer to the Authors Corner on our Web site, http://www.infsoc.org/.

Submission of manuscripts: There is no limitation of page count as full papers, each of which will be subject to a full review process. An electronic, PDF-based submission of papers is mandatory. Download and use the LaTeX2e or Microsoft Word sample IJIS formats.

http://www.infsoc.org/IJIS-Format.pdf

LaTeX2e

LaTeX2e files (ZIP) http://www.infsoc.org/template_IJIS.zip

Microsoft Word™

Sample document http://www.infsoc.org/sample_IJIS.doc

Please send the PDF file of your paper to secretariat@infsoc.org with the following information:

Title, Author: Name (Affiliation), Name (Affiliation), Corresponding Author. Address, Tel, Fax, E-mail:

Copyright

Publisher

Address: Informatics Laboratory, 3-41 Tsujimachi, Kitaku, Nagoya 462-0032, Japan

E-mail: secretariat@infsoc.org

CONTENTS