Electronic Preprint for Journal of Information Processing Vol.26

Regular Paper

Curriculum Analysis of Computer Science Departments by Simplified, Supervised LDA

Yoshitatsu Matsuda[1,a)] Takayuki Sekiya[2,b)] Kazunori Yamaguchi[1,c)]

Received: June 27, 2017, Accepted: March 6, 2018

Abstract: The design of appropriate curricula is one of the most important issues in higher educational institutions, and there are many features to be considered. In this paper, the two key features ("locality bias" and "combination of two simple factors") were discovered by investigating the actual computer science (CS) curricula of the top-ranked universities on the basis of Computer Science Curricula 2013 (CS2013), where the CS topics are classified into the 18 Knowledge Areas (KAs). We applied a machine learning method named simplified, supervised latent Dirichlet allocation (ssLDA) to the actual syllabi of the CS departments of the 47 top-ranked universities. ssLDA estimates the relative weights of the KAs of CS2013 in each syllabus. Then, each CS department was characterized as the averaged weights of the KAs over its included syllabi. We applied the three well-known data analysis methods (hierarchical cluster analysis, principle component analysis, and non-negative matrix factorization) to the averaged weights of each department and found the above two key features quantitatively and objectively.

Keywords: syllabus, curriculum, curriculum analysis, CS2013, supervised LDA

1. Introduction

It is needless to say that a curriculum is important for education. A curriculum should represent a characteristic educational activity that each university offers to students. However, it is not easy to design an appropriate CS curriculum with the limited time and resources, due to the rapid expansion of the field of computer science. The ACM and IEEE have been putting much effort into providing the standard curricula series for over 40 years because faculty members and instructors need curricular guidelines to design their own CS curriculum for their universities. Computer Science Curricula 2013 (CS2013) [1] is the latest edition of curriculum guidelines for undergraduate degree programs in CS, which is released by the ACM and IEEE. Our ultimate purpose is to provide some useful and actionable additional guidelines for developing or improving the CS curricula appropriately by utilizing CS2013. In other words, we attempt to propose the guidelines which support to assign the time resources appropriately to the CS topics and to design an attractive curriculum.

There are many previous works such as design tools [27], the repository [24], course-knowledge unit relations [14], [25], the change visualization [10]. On the other hand, we have employed a data science approach using a statistical machine learning method [16], [17], [19]. They collected the actual syllabi of the course curriculum from the CS departments of the top-ranked universities. Then, in order to convert the syllabi into quantitative data, they employed simplified, supervised latent Dirichlet allocation (ssLDA) [19], which can estimate the relative weights of the Knowledge Areas (KAs) of CS2013 in each syllabus without human intervention. KA corresponds to a principal topic of study in CS. **Table 1** shows the names and abbreviations of KAs. Each syllabus is projected to a point in the KA space, where each coordinate represents the strength of the connection between the syllabus and the corresponding KA. Then, each curriculum was represented as the center of the points corresponding to its included syllabi. By applying the well-known data analysis methods to the centers, they could discover some useful facts for designing appropriate curricula.

In this paper, we employed the same data science approach. From the investigation results, we discovered the following two key features from the actual curricula: "locality bias" over the world and "combination of two simple factors" in curriculum design. First, some countries are working hard to develop a national curriculum [15] and it is clear that there is some locality bias. However, the details of the locality bias have not been estimated quantitatively. For example, it is not clear yet which countries are grouped. Our investigation disclosed the properties of the locality bias quantitatively and objectively. Second, the weighting of each KA in a curriculum is quite important for designing an appropriate curriculum, and CS2013 provides the guidelines (named "Core Tier-1," "Core Tier-2" and "Elective"). However, it is not always so easy to utilize the guidelines in practice because it is not based on the actual curricula. Our investigation extracted the two simple factors from the actual curricula, which are easier to use in practice. This paper is an extended and elaborated version of Ref. [19] with more massive datasets and much more intensive

[1] Graduate School of Arts and Sciences, The University of Tokyo, Meguro, Tokyo 153–8902, JAPAN
[2] Information Technology Center, The University of Tokyo, Meguro, Tokyo 153–8902, JAPAN
a) matsuda@graco.c.u-tokyo.ac.jp
b) sekiya@ecc.u-tokyo.ac.jp
c) yamagch@graco.c.u-tokyo.ac.jp

Table 1 KAs of CS2013.

ID	KA
AL	Algorithms and Complexity
AR	Architecture and Organization
CN	Computational Science
DS	Discrete Structures
GV	Graphics and Visualization
HCI	Human-Computer Interaction
IAS	Information Assurance and Security
IM	Information Management
IS	Intelligent Systems
NC	Networking and Communication
OS	Operating Systems
PBD	Platform-Based Development
PD	Parallel and Distributed Computing
PL	Programming Languages
SDF	Software Development Fundamentals
SE	Software Engineering
SF	Systems Fundamentals
SP	Social Issues and Professional Practice

investigation.

This paper is organized as follows. Section 2 describes the related works. Section 3 explains the collected datasets. Section 4 describes the method projecting each syllabus to the KA space (named ssLDA) in detail. The appropriateness of ssLDA and the KA space is verified in Section 5. Investigation results about the actual syllabi and curricula are shown in Section 6, and they are discussed in Section 7. Lastly, Section 8 concludes the paper.

2. Related Works

Since a curriculum is one of the most important assets of higher education, some works developed curriculum design tools and made it public [7], [27]. In many cases, such tools require teachers to define courses with units of knowledge [14], which takes a lot of time and effort. Though Tungare et al. created a repository system for computer science syllabi [24] and developed tools such as Syllabus-Maker for creating and comparing syllabi, they have not developed a technique to grasp characteristics of a whole curriculum. Though a number of studies have been made on methodologies and tools for analyzing curricula by using statistically processed syllabus data [8], [12], [28], they can not compare the different curricula quantitatively. Marshall tried to quantify the changes in the CS structure of the ACM/IEEE curricula series [10] and visualized the structure of a curriculum by modeling the KAs as a network graph. Gluga et al. developed a web-based system called PROGOSS that maps curricula learning goals and mastery levels to individual assessment tasks across entire degree programs [6]. Szabo et al. also developed curriculum analysis framework which supports the identification of prerequisite concepts [20]. Mendez et al. reported that they applied several learning analytic techniques to a curriculum [11], where the long-standing grade data of students in their institution is used for the curriculum analysis. Kawintiranon et al. proposed the curriculum analysis method [9] by mapping course materials to the Computer Engineering Curricular Guideline, CE2016 [2]. They use keywords extracted from course materials using TF-IDF and information of web pages gathered by Google Search API. Our method can directly project syllabi to KA space with a sophisticated method based on Latent Dirichlet allocation (LDA) [5] without any other additional information.

3. Datasets

It is the key feature of our analysis to characterize the actual CS curriculum quantitatively by the KAs of CS2013. Each curriculum is regarded as the set of the included syllabi. Each syllabus is projected to a point in the KA space by ssLDA. Then, each curriculum is characterized as a point in the KA space, which is the center of the points of the included syllabi. Here, we describe the datasets.

3.1 Computer Science Curricula 2013 (CS2013)

The ACM and IEEE Computer Society jointly have been constructing curricular guidelines for undergraduate programs in CS and have been releasing reports roughly every ten years, such as CC1991 [3], CC2001 [22], and CS2008 [21]. In December 2013, "Computer Science Curricula 2013" (CS2013) was released as the latest report in CS. The report includes a set of principles, a redefined Body of Knowledge (BOK), exemplars of actual courses and curricula. According to the CS2013 report, the BOK "does not propose a particular set of courses of curriculum structure," but "In Computer Science terms, one can view the Body of Knowledge as a specification of the content to be covered and a curriculum as an implementation." The BOK consists of a set of 18 Knowledge Areas (KAs), each of which corresponds to a principal topic of study in CS. Each KA contains about 10 Knowledge Units (KUs), where each KU is a short document. Table 1 shows the names and abbreviations of KAs. By applying a simple stemming algorithm to obtain nouns in a singular form, 3,304 words were extracted from the BOK of CS2013.

3.2 Collection of Actual Curricula

We collected manually the actual curricula offered by CS departments of higher educational institutions. In order to obtain such curricula of major universities, we referred to one of the popular university rankings, titled "Times Higher Education (THE) WORLD UNIVERSITY RANKINGS [23], Top 100 universities for engineering and technology 2014–2015." We analyzed the 47 universities of the top 50 ones because the curricula of three remaining universities could not be found on their own web sites. **Table 2** lists the universities and the departments related to CS. We use the IDs in the rightmost column of the table to specify universities hereafter. We manually downloaded web pages or PDF files from each department's website and extracted the syllabi of courses required to take a bachelor of CS. For simplicity, we did not take the mandatory or prerequisite structures into consideration. Though most of the universities in Table 2 offer their curricula in English, six universities (TUDelft, UTokyo, TUM, KUL, Kyoto, and ECOLE) offer their curricula in their own languages. We translated those non-English curricula into English with Google Translate [*1]. Each syllabus is characterized as a bag of words which is used in KUs of CS2013. We eliminated obviously unnecessary words such as HTML tags, header and footer, and stop words. We also applied a simple stemming algorithm to obtain nouns in a singular form. In addition, we excluded

[*1] https://translate.google.com/

Table 2 CS related departments of universities.

Rank	Country	University (Department)	ID
1	us	Massachusetts Institute of Technology (Electrical Engineering and Computer Science)	MIT
2	us	Stanford University (Computer Science Dept.)	Stanford
3	us	California Institute of Technology (Computing + Mathematical Science Dept.)	Caltech
4	uk	Princeton University (Dept. of Computer Science)	Princeton
5	uk	University of Cambridge (Computer Laboratory)	Cambridge
6	uk	Imperial College London (Dept. of Computing)	Imperial
7	uk	University of Oxford (Dept. of Computer Science)	Oxford
8	ch	ETH Zürich - Swiss Federal Institute of Technology Zürich (Dept. of Computer Science)	ETH
9	us	University of California, Los Angeles (Computer Science Dept.)	UCLA
10	us	University of California, Berkeley (Dept. of Electrical Engineering and Computer Sciences)	UCB
11	us	Georgia Institute of Technology (College of Computing)	Georgia Tech
12	ch	École Polytechnique Fédérale de Lausanne (School of Computer and Communication Sciences)	EPFL
13	sg	National University of Singapore (School of Computing)	NUS
14	us	University of Texas at Austin (Computer Science Dept.)	UTAustin
15	us	University of Michigan (Dept. of Electrical Engineering and Computer Science)	Michigan
16	us	Carnegie Mellon University (School of Computer Science)	CMU
17	us	Cornell University (Dept. of Computer Science)	Cornell
18	us	University of Illinois at Urbana-Champaign (Dept. of Computer Science)	Illinois
19	nl	Delft University of Technology	TUDelft
19	us	Northwestern University (Dept. of Electrical Engineering and Computer Science)	Northwestern
21	hk	Hong Kong University of Science and Technology (Dept. of Computer Science and Engineering)	HKUST
22	us	University of California, Santa Barbara (Dept. of Computer Science)	UCSB
24	ca	University of Toronto Scarborough (Dept. of Computer and Mathematical Sciences)	UTSC
25	jp	The University of Tokyo (Dept. of Information Science, School of Science)	UTokyo
27	us	University of Wisconsin-Madison (Dept. of Computer Science)	Wisconsin
28	de	Technical University of Munich (Dept. of Informatics)	TUM
29	sg	Nanyang Technological University (School of Computer Engineering)	NTU
30	se	KTH Royal Institute of Technology	KTH
31	dk	Technical University of Denmark (Dept. of Applied Mathematics and Computer Science)	DTU
32	us	Columbia University (Computer Science Dept.)	Columbia
33	us	University of Washington (Dept. of Computer Science and Engineering)	Washington
34	be	KU Leuven (Dept. of Computer Science)	KUL
35	kr	Seoul National University (Dept. of Computer Science and Engineering)	Seoul
36	hk	The University of Hong Kong (Dept. of Computer Science)	HongKong
37	uk	University of Manchester (School of Computer Science)	Manchester
37	au	University of Melbourne (School of Information)	UNIMELB
39	au	University of Queensland (School of Information Technology and Electrical Engineering)	Queensland
40	us	Rice University (Dept. of Computer Science)	Rice
41	jp	Kyoto University (Informatics and Mathematical Science, Faculty of Engineering)	Kyoto
42	fr	École Polytechnique (Computer Science Department)	ECOLE
43	ca	University of British Columbia (Dept. of Computer Science)	UBC
45	us	Purdue University (School of Electrical and Computer Engineering)	Purdue
46	kr	Pohang University of Science and Technology (Computer Science and Engineering)	POSTECH
46	au	University of Sydney (School of Information Technologies)	Sydney
48	au	Monash University (Faculty of Information Technology)	Monash
49	us	University of Minnesota (Dept. of Computer Science and Engineering)	Minnesota
50	us	University of California, San Diego (Dept. of Computer Science and Engineering)	UCSD

the syllabi consisting of less than 10 words. The averaged ratio of the excluded syllabi over all the curricula is 8.1%. In other words, 91.9% of the actual syllabi include a high enough number of words (namely, $\geq$ 10) in the vocabulary of the BOK of CS2013. It means that the words of CS2013 could largely cover the actual syllabi. The averaged number of syllabi over the 47 universities was 65.5, and the averaged number of words over all the syllabi was 39.2.

4. Method Projecting Syllabi to KA Space

In order to convert each syllabus to a quantitative data, we employ "simplified, supervised latent Dirichlet allocation" (ssLDA), which was originally proposed in Ref. [19]. ssLDA estimates the relative weights of the KAs for a given document (a bag of words), where the sum of the weights is normalized to 1. The weights can be regarded as a point in the KA space. In other words, ssLDA can project each syllabus to the KA space. ssLDA is an extension of the widely-used Latent Dirichlet allocation (LDA) method [5] and its supervised version [4], [26]. Here, the details of ssLDA are described.

4.1 Outline

Here, we will describe the outline of the method. First, every syllabus is regarded as a set of used words (namely, the bag of words model). Then, each syllabus is projected to a point in the topic space defined by CS2013. LDA is employed for this projection. LDA is widely used in natural language processing and machine learning and is known to be useful for extracting the topic space from a set of reference documents and projecting other documents to the extracted space. However, the original LDA is designed to extract the topics automatically. On the other hand, the topic of a reference document is given in advance in our target dataset (namely, the BOK of CS2013). Therefore, we employed ssLDA for giving an approximate model to our target dataset. Supervised LDA (sLDA) has been proposed in a different context [4], [26] where classification labels are given as well

Table 3 Term correspondence table between ssLDA model and curriculum analysis.

ssLDA Model	Curriculum Analysis
document	syllabus
set of documents	curriculum
topic	KA
position of a document in topic space	dominance ratios of KAs of a syllabus
training data	BOK of CS2013
allocated label in training data	KA allocated to each syllabus in CS2013
$\boldsymbol{\theta} = (\theta_i)$	true dominance ratios of KAs
$z_n = (z_{ni})$	KA allocated to word n
$\boldsymbol{\beta} = (\beta_{ij})$	Strength of relationship between KA i and word j
c	most dominant KA

as documents. However, the labels are not directly related to the extracted topics in LDA. ssLDA is a simplified version of sLDA, where each extracted topic is bound to a given topic label. **Table 3** shows the correspondence of terms between the proposed ssLDA model and our curriculum analysis.

It is essential that ssLDA allows a syllabus to belong to multiple KAs because it is based on a probabilistic model. In the training data (namely, CS2013), every syllabus belongs to a single KA. Therefore, many classification methods, such as support vector machines (SVMs), would seem to be suitable for the training phase. However, the syllabus in an actual curriculum is often distributed over many KAs. According to the CS2013 report, "Knowledge Areas are not intended to be in one-to-one correspondence with particular courses in a curriculum:" (Ref. [1] p.27). In addition, more than 60% of course exemplars of CS2013 cover multiple KAs. Therefore, methods that focus on classification are not appropriate. On the other hand, since ssLDA learns a probabilistic model from the training data, the dominance ratios of multiple KAs can be estimated from an actual syllabus.

It is also essential that ssLDA can estimate a continuous probabilistic model even from the sparse data because of the Dirichlet prior. Simple probabilistic models, such as naive Bayes classifiers, give a discontinuous probabilistic estimation because almost all words occur only a few times in CS2013. Therefore, such simple models are not appropriate for syllabi belonging to multiple KAs.

4.2 Generative Model

The probabilistic generative model of a document with a label in ssLDA is given as follows:

(1) The probability of occurrence of topics $\boldsymbol{\theta} = (\theta_i)$ (constrained to $\sum_i \theta_i = 1$) is generated by Dirichlet distribution with a hyper-parameter α. Here, $i = 1, \ldots, K$ and K is the number of topics.

(2) For each word w, a topic assignment $\boldsymbol{z}_n = (z_{ni})$ is given by multinomial distribution with the parameter $(\boldsymbol{\theta} = (\theta_i))$, where $n = 1, \ldots, N$ and N is the number of words in the document. Each $\boldsymbol{z}_n$ is the K-dimensional vector where the assigned topic is 1 and the others are 0. Then, the word is given by the multinomial distribution with the parameter $(\boldsymbol{\beta} = (\beta_{ij}))$, where i is the assigned topic in $\boldsymbol{z}_n$ and j corresponds to the actual word w_n in the vocabulary (where $\sum_j \beta_{ij} = 1$ for every i).

(3) The allocated topic of the document $c \in \{1, \ldots, K\}$ is given by the following softmax distribution with the parameter $\bar{\boldsymbol{z}} = \sum_n \boldsymbol{z}_n / N$ and a hyper-parameter $\eta > 0$: $P(c|\bar{\boldsymbol{z}}, \eta) = \exp(\eta \bar{z}_c) / \sum_{i=1}^{K} \exp(\eta \bar{z}_i)$. Here, $\bar{\boldsymbol{z}} = (\bar{z}_i)$ can be regarded as the empirical distribution on the topic assignments of the document.

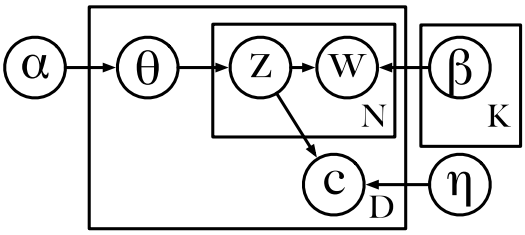

Fig. 1 Generative model of simplified, supervised LDA.

The graphical model of the above generative model is shown in **Fig. 1**.

This is an extension of the original LDA model [5] with c and η. Because each label c corresponds to an internal topic assignment z, the model is much simpler than the previous sLDA models [4], [26].

4.3 Inference and Parameter Estimation

For estimating the variables and parameters in the model, such as $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$, we use the maximum likelihood estimation with a variational EM algorithm. Though the inference process can be regarded as a special case of a more general method on a more complicated model in Ref. [26], it is derived here under the simplified model. The original likelihood of a document with a given topic label is given as $\log P(\boldsymbol{w}, c|\alpha, \boldsymbol{\beta}, \eta)$ where $\boldsymbol{w} = (w_n)$ is a document consisting of words w_n's. The estimation of this original likelihood is intractable. Thus, the following lower bound is derived by the variational Bayesian approach in a way similar to that in the original LDA [5]:

$$
\begin{aligned}
&\log P(\boldsymbol{w}, c|\alpha, \boldsymbol{\beta}, \eta) \\
&= \log \int \mathrm{d}\boldsymbol{\theta} \sum_{\boldsymbol{Z}} P(\boldsymbol{w}, c, \boldsymbol{\theta}, \boldsymbol{Z}|\alpha, \boldsymbol{\beta}, \eta) \\
&= \log \int \mathrm{d}\boldsymbol{\theta} \sum_{\boldsymbol{Z}} \frac{P(\boldsymbol{w}, c, \boldsymbol{\theta}, \boldsymbol{Z}|\alpha, \boldsymbol{\beta}, \eta)\, Q(\boldsymbol{\theta}, \boldsymbol{Z}|\boldsymbol{\gamma}, \boldsymbol{\phi})}{Q(\boldsymbol{\theta}, \boldsymbol{Z}|\boldsymbol{\gamma}, \boldsymbol{\phi})} \\
&\geq \int \mathrm{d}\boldsymbol{\theta} \sum_{\boldsymbol{Z}} Q(\boldsymbol{\theta}, \boldsymbol{Z}|\boldsymbol{\gamma}, \boldsymbol{\phi}) \log \frac{P(\boldsymbol{w}, c, \boldsymbol{\theta}, \boldsymbol{Z}|\alpha, \boldsymbol{\beta}, \eta)}{Q(\boldsymbol{\theta}, \boldsymbol{Z}|\boldsymbol{\gamma}, \boldsymbol{\phi})} \\
&= E_q(\log P(\boldsymbol{w}, \boldsymbol{\theta}, \boldsymbol{Z}|\alpha, \boldsymbol{\beta})) + E_q(\log P(c|\boldsymbol{Z}, \eta)) \\
&\quad - E_q(\log Q(\boldsymbol{\theta}, \boldsymbol{Z}|\boldsymbol{\gamma}, \boldsymbol{\phi})) \qquad (1)
\end{aligned}
$$

where $\boldsymbol{Z} = (\boldsymbol{z}_n)$, $Q(\boldsymbol{\theta}, \boldsymbol{Z}|\boldsymbol{\gamma}, \boldsymbol{\phi}) = q(\boldsymbol{\theta}|\boldsymbol{\gamma}) \prod_n q(\boldsymbol{z}_n|\boldsymbol{\phi}_n)$ is the variational distribution, and $E_q()$ is the expectation operator over Q. $\boldsymbol{\gamma} = (\gamma_i)$ and $\boldsymbol{\phi} = (\boldsymbol{\phi}_n)$ are the free variational parameters. In addition, $\boldsymbol{\gamma}$ and $\boldsymbol{\phi}_n = (\phi_{ni})$ are the Dirichlet parameter and the multinomial one (constrained to $\sum_i \phi_{ni} = 1$), respectively. The lower bound in Eq. (1) is the same as that in the original LDA except for the second term $E_q(\log P(c|\boldsymbol{Z}, \eta))$ in the last form, which is derived from the softmax distribution of the topic label. The lower bound of $E_q(\log P(c|\boldsymbol{Z}, \eta))$ is given as follows in a way similar to

that in Ref. [26]:

$$
\begin{aligned}
& E_q\left(\log P\left(c|\boldsymbol{Z},\eta\right)\right) \\
&= E_q\left(\log \frac{\exp\left(\eta\bar{z}_c\right)}{\sum_i \exp\left(\eta\bar{z}_i\right)}\right) \\
&= \frac{E_q\left(\eta\sum_n z_{nc}\right)}{N} - E_q\left(\log\left(\sum_i \exp\left(\eta\bar{z}_i\right)\right)\right) \\
&\geq \frac{\eta\sum_n E_q\left(z_{nc}\right)}{N} - E_q\left(\log\left(\sum_i \bar{z}_i \exp(\eta)\right)\right) \\
&= \frac{\eta\sum_n E_q\left(z_{nc}\right)}{N} - E_q\left(\log\left(\exp(\eta)\right) + \log\left(\sum_i \bar{z}_i\right)\right) \\
&= \frac{\eta\sum_n \phi_{nc}}{N} - \eta - E_q\left(\log\left(\sum_i \bar{z}_i\right)\right) = \frac{\eta\sum_n \phi_{nc}}{N} - \eta
\end{aligned}
\tag{2}
$$

Here, Jensen's inequality on the exponential function is applied under the conditions $\sum \bar{z}_i = 1$ and $\bar{z}_i \geq 0$. $E_q\left(z_{nc}\right) = \phi_{nc}$ is also utilized. Then, the variational and model parameters $(\boldsymbol{\gamma}, \boldsymbol{\phi}, \boldsymbol{\beta})$ can be estimated by the variational EM algorithm. Only the update equations are shown here. $\boldsymbol{\gamma}$ and $\boldsymbol{\phi}$ are estimated by

$$
\gamma_i = \alpha + \sum_n \phi_{ni}\,, \tag{3}
$$

$$
\phi_{ni} \propto \beta_{iw_n} \exp\left(\Psi\left(\gamma_i\right) - \Psi\left(\sum_k \gamma_k\right) + \frac{\eta\delta_{ic}}{N}\right) \tag{4}
$$

where Ψ is the digamma function and δ_{ic} is the Dirac delta function. $\boldsymbol{\beta}$ is estimated by

$$
\beta_{ij} \propto \sum_d \sum_n 1\left[j = w_n^d\right] \phi_{ni}^d \tag{5}
$$

where $d = 1, \ldots, D$ corresponds to each document in the datasets and $1\left[j = w_n^d\right]$ is the function taking 1 only when the n-th word w_n^d in the document d is equal to the word j (or 0 otherwise). The update equations for $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ are almost exactly the same as those in the original LDA [5]. The only difference is the additional term $\eta\delta_{ic}/N$ in the update for $\boldsymbol{\phi}$. It strengthens ϕ_{ni} only if i is equivalent to the given topic label c. In addition, the weight depends on η and N only. In this paper, α and η are treated as fixed hyperparameters. The settings on them are described in Section 5. Note that η degenerates into 0 if it is optimized by the maximum likelihood estimation because the penalty for misclassification is overestimated. Though it is a kind of overfitting problem, it can be avoided by the cross-validation method described in Section 5.

4.4 Prediction

After the estimation of $\boldsymbol{\beta}$ of the generative model, a given document with no label is projected to a position $\boldsymbol{\theta}^* = \left(\theta_i^*\right)$ in the topic probability space with the expected topic label c^*. In order to estimate $\boldsymbol{\theta}^*$ robustly, $\boldsymbol{\theta}^*$ is estimated as the expectation of $\boldsymbol{\theta}$ over the conditional probability with no label. In other words, $\boldsymbol{\theta}^*$ is the mean over $P\left(\boldsymbol{w}, \boldsymbol{\theta}|\alpha, \boldsymbol{\beta}\right)$. This distribution is approximated as $\sum_{\boldsymbol{Z}} Q\left(\boldsymbol{\theta}, \boldsymbol{Z}|\boldsymbol{\gamma}^*, \boldsymbol{\phi}^*\right) = q\left(\boldsymbol{\theta}|\boldsymbol{\gamma}^*\right)$ by the variational approach, where $\boldsymbol{\gamma}^* = \left(\gamma_i^*\right)$ and $\boldsymbol{\phi}^* = \left(\phi_{nc}^*\right)$ are the parameters estimated by the original LDA with no label. In other words, the additional term $\eta\delta_{ic}/N$ is omitted in the update process of $\boldsymbol{\gamma}^*$ and $\boldsymbol{\phi}^*$ in the same way as in the original LDA. Then, $\theta_i^* = \gamma_i^* / \sum_i \gamma_i^*$ because $\boldsymbol{\gamma}^*$ is the Dirichlet parameter. Regarding c^*, it is given so that $\log P\left(c = c^*|\eta\right)$ is the maximum over c. By Eq. (2), $\log P\left(c|\eta\right)$ is approximated as $\sum_n \phi_{nc}^* \cong \gamma_c^*$ where constant factors are omitted. Therefore, c^* is given so that $\gamma_{c^*}^*$ is the largest over γ_c^*.

5. Verification of ssLDA and KA Space

Here, we describe the experimental results for determining the hyper-parameters of ssLDA and verifying the appropriateness of the KA space generated by ssLDA.

5.1 Determination of Hyper-parameters

Here, the setting of the hyper-parameters α and η in Section 4 is described. Though α is originally given as a K-dimensional vector for the K-dimensional Dirichlet distribution, it is given as a single parameter by assuming that α is the same constant over all the topics. α was set to 1 in the same way as in Sekiya's previous work [18] by using some empirical and theoretical considerations. To determine η we use a cross-validation method that makes use of the classification accuracy of the topic labels. In this paper, we utilize the leave-one-out cross-validation (LOOCV) estimation of the topic classification accuracy in the training data of CS2013. In order to avoid the effects of local maxima, the optimizations of ssLDA were carried out in five trials from random initial values for each η. Then, the result with the largest likelihood estimation was used. The LOOCV estimations of various η are shown in **Table 4**. It can be seen from the table that $\eta = 50$ is the best setting.

5.2 Verification of KA Space Generated by ssLDA

Here, the appropriateness of the generated KA space is verified from the following three viewpoints: the comparison with other classification methods in the training data of CS2013, the classification accuracy in the test data of the actual syllabi (the course exemplars in the appendix of CS2013), and the words with high probability in each KA.

First, we compared the proposed method with other classification methods by the LOOCV estimations of the classification accuracy in the training data of CS2013. **Table 5** shows the results for naive Bayes classifier and SVMs (with the linear, radial, and sigmoid kernels). We used the R e1071 package [*2] which

Table 4 Parameter η and LOOCV classification accuracy.

η	Accuracy
5.0	0.172
10.0	0.399
20.0	0.638
50.0	**0.663**
100.0	0.595
200.0	0.534

Table 5 Comparison of LOOCV classification accuracy with other methods (naive Bayes classifier and SVMs with various kernels).

Classifier	Accuracy
Naive Bayes	0.656
SVM (linear)	0.264
SVM (radial)	0.325
SVM (sigmoid)	0.620
ssLDA (η=50)	**0.663**

[*2] Package 'e1071', http://cran.r-project.org/web/packages/e1071/e1071.pdf (accessed 2017-06-03).

Table 6 The top 10 words with the highest probability in each KA of CS2013.

AL	algorithm, graph, tree, complexity, automatum, solve, implement, algorithmic, class, strategy
AR	instruction, memory, architecture, familiarity, assembly, level, organization, processor, representation, machine
CN	simulation, modeling, science, information, including, datum, model, algorithm, computational, processing
DS	proof, probability, induction, propositional, relation, predicate, usage, bayes, counting, theorem
GV	rendering, visualization, graphic, surface, image, representation, animation, rasterization, light, color
HCI	user, interface, interaction, design, motivation, HCI, evaluation, technology, quantitative, report
IAS	security, attack, secure, forensic, cryptographic, threat, cryptography, familiarity, policy, SE
IM	query, relational, database, information, index, datum, schema, transaction, file, mining
IS	search, agent, reasoning, planning, classification, robot, representation, learning, implement, algorithm
NC	network, platform, social, layer, familiarity, application, allocation, industrial, IP, describe
OS	system, operating, memory, device, access, SF, virtual, OS, file, management
PD	parallel, parallelism, distributed, shared, message, versus, race, algorithm, synchronization, SF
PBD	function, programming, web, mobile, operation, class, constraint, variant, language, event
PL	type, program, language, code, static, analysis, semantic, syntax, memory, optimization
SP	social, professional, privacy, computing, ethical, computer, intellectual, policy, HCI, environmental
SDF	design, program, software, component, principle, coding, programming, error, code, structure
SE	software, requirement, team, risk, project, process, specification, testing, development, validation
SF	performance, logic, scheduling, memory, machine, error, program, simple, resource, figure

was an implementation of naive Bayes classifier and SVMs. The hyper-parameters of SVMs were tuned by the cross validation. It shows that the highest accuracy in ssLDA (0.663 in Table 4) is (more or less) superior to all the other methods. It verifies the validity of the proposed method in CS2013.

Second, we used the course exemplars in the appendix of the CS2013 report as the test dataset. These exemplars were marked with the KAs that they most significantly cover by the lecturers of those courses. There were 15 KA-marked syllabi in the collected dataset. The proposed method calculated the values of each KA for the 15 syllabi. Then, the rankings of the marked (namely correct) KAs in the values were utilized for the estimation. In the eleven syllabi, the rankings are the 1st. The other four syllabi have the 2nd, 7th, 9th, and 13th rankings. The rate of correctness is 73 percentage (11/15). Considering that there are the 18 KAs, it can be regarded to be quite high. The mean reciprocal rank is 0.79, which is quite high also.

Third, we show the words with high probability in each KA, where the probability is estimated as $\boldsymbol{\beta}$ in the proposed method. **Table 6** shows the 10 words with the highest probability in each KA. Intuitively, the extracted words seem to be strongly related to the corresponding KA.

6. Investigation Results

Here, the characteristics of the actual curricula are investigated in the KA space. Each curriculum of the CS department of a university is characterized by the center of gravity (the mean) of its included syllabi in the KA space.

6.1 Investigation of Actual Syllabi by Basic Statistics in KA Space

Here, the basic characteristics of the actual syllabi in the KA space are investigated, and the appropriateness of CS2013 is verified. If the KA space is appropriate, the syllabi should be distributed "widely" with a low-biased center in the KA space. First, **Fig. 2** shows the distributions estimated by their histograms along each axis of KA over all the collected syllabi. The mode (namely, the peak), the mean (the center) and the standard deviation of the distribution are shown also. The means of the distributions seem to be near to the unbiased point $(1/18 \simeq 0.056, 0.056, \cdots, 0.056)$ although some KAs (such as CN, GV, HCI and SP) are a little high. Regarding the modes, they are around 0.03 in every KA. The value of 0.03 is smaller than the unbiased point. In addition, every histogram is a long-tailed distribution. It shows that there are a few dominant KAs in many syllabi. Regarding the standard deviations, they are similar although some KAs are a little high. In order to investigate the correlations among the KAs, **Fig. 3** shows the bar graphs of the rank-ordered eigenvalues (the percentage over the sum) of the covariance matrix among KAs. Note that the rightmost (and the smallest) eigenvalue on each bar graph is always 0 because of the constraint $\sum_i \theta_i^* = 1$ in the KA space. In Fig. 3, the bar graph seems to be flat and the largest value is not extremely high. It shows that the distribution of the syllabi is not concentrated in any directions. In summary, all the syllabi of the actual universities are distributed within an area of the KA space. The distribution in each KA is long-tailed and there is no KA which is always dominant. In addition, the distributed area is not concentrated in any directions. These results suggest that CS2013 is appropriate for analyzing the actual syllabi.

6.2 Investigation of Actual Curricula by Basic Statistics in KA Space

Here, the basic characteristics of the actual curricula are investigated in the KA space, and it is shown that there is some university bias in the actual curricula. We will show that there is a structure in the distribution of the curricula. In **Fig. 4**, the bar graphs of the rank-ordered eigenvalues (the percentage over the sum) of the covariance matrix among KAs for the curricula are shown. Comparing it with Fig. 3, there are several dominant principal components in Fig. 4. On the other hand, **Table 7** shows the averaged Euclidean distance among syllabi within a curriculum and that over all the syllabi. They correspond to the degree of the deviation of syllabi within each curriculum and the total deviation over all the syllabi. There was only a slight difference between them. In other words, each university offers a variety of courses in its CS curriculum to a degree. However, Fig. 4 shows that the curricula are concentrated in a few different directions.

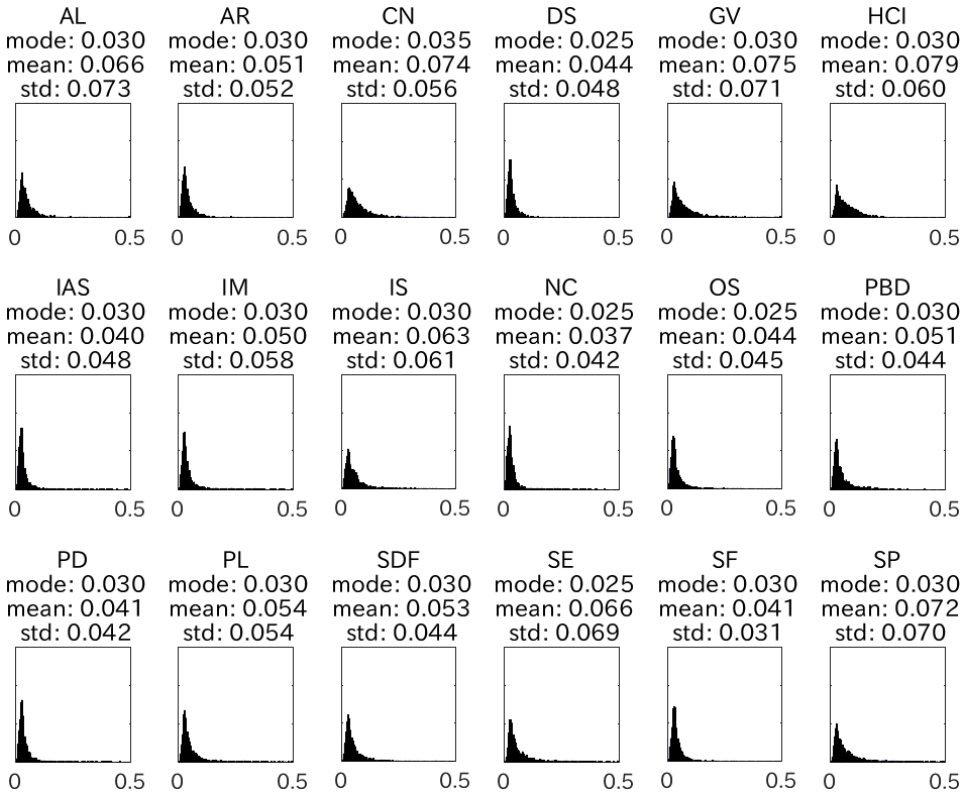

Fig. 2 The distribution of syllabi (with the mode (peak), the mean, and the standard deviation (STD)) along each KA.

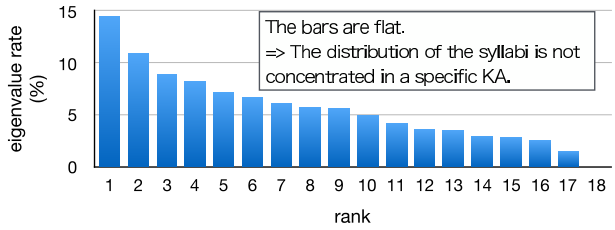

Fig. 3 Bar graphs of rank-ordered eigenvalues of covariance matrix of the distributions for all the syllabi.

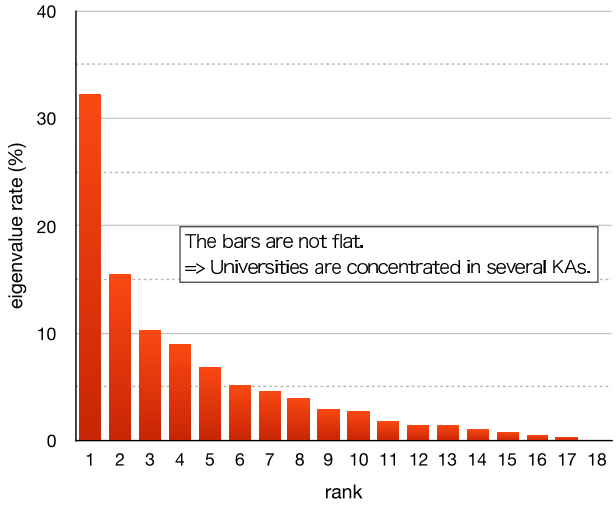

Fig. 4 Bar graphs of rank-ordered eigenvalues of covariance matrix for the universities.

Table 7 Averaged distance within each university and that over all the syllabi.

within university	all syllabi
0.302	0.315

6.3 Discovery of Structure Embedded in Actual Curricula by HCA and PCA

Here, we applied hierarchical cluster analysis (HCA) and principal component analysis (PCA) to the actual curricula in the KA space in order to find out their embedded structure. Regarding HCA, the distance among the curricula is measured as the usual Euclidean one. Then, Ward's method is utilized for constructing the hierarchical cluster tree of the curricula. The four clusters (denoted by C1, C2, C3, and C4) were extracted from the tree. The number of clusters was set experimentally for the analysis. Regarding PCA, the covariance matrix of KAs is calculated over the curricula and is decomposed. The bar graph of the rank-ordered eigenvalues (the percentage over the sum) of the covariance matrix among KAs is shown in Fig. 4, where there are several dominant principal components. The three principal components (denoted by P1, P2, and P3) were investigated in this paper. **Figure 5** shows the extracted hierarchical tree. The four clusters are colored (C1:cyan, C2:red, C3:magenta, C4:green). **Figures 6** and **7** show the one-dimensional and two-dimensional plots of the curricula along the three principal components, where each curricula is colored also by its cluster. They show that the four clusters are divided in the three-dimensional principal component space. C1 is separated from the others along the first principal component (Fig. 6). The other clusters are divided in the two dimensional space along the second and third principal components in Fig. 6, where C2, C3, and C4 are around the upper left, the upper right, and the lower, respectively. **Figure 8** shows the center of each cluster in the KA space. The values along each KA are centered in advance by subtracting their mean from them.

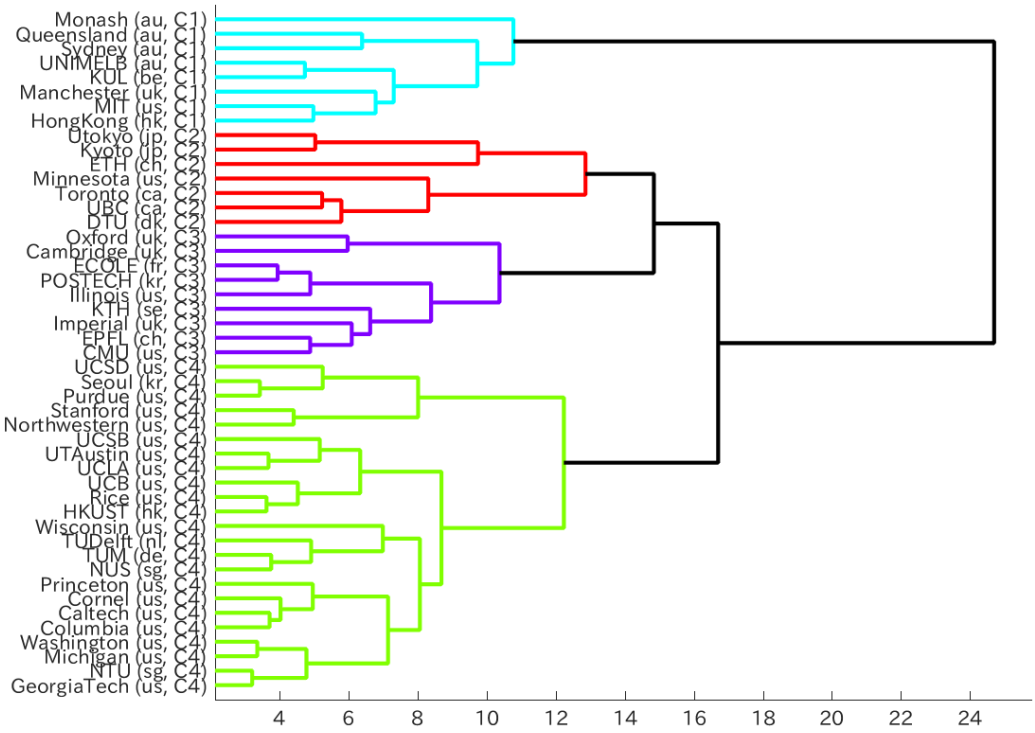

Fig. 5 Hierarchical cluster tree of the universities by Ward's method: Each university is represented by its ID with the country code in Table 2 and its assigned cluster (C1-C4) when they are divided into four clusters. Each cluster is colored (C1:cyan, C2:red, C3:magenta, C4:green).

Then, the center of each cluster in the KA is calculated as the mean over all the curricula in the cluster. The positive or negative value of each KA corresponds to the degree of prioritization or the deprioritization of the KA in the cluster, respectively. Figure 8 suggests the properties of the clusters as follows:

- C1 tends to prioritize HCI, SE, and SP (quite intensively), which are strongly related to the human and social aspects. On the other hand, it tends to deprioritize GV and PL, which are related to the computational aspect. It shows that the universities in C1 give high priority to the human aspect in their education.
- C2 tends to prioritize AL and CN, which are related to the computational and theoretical aspects. On the other hand, it tends to deprioritize GV, HCI, and SP, which are related to the human aspect. It shows that the universities in C2 encourage the theoretical aspect.
- C3 tends to prioritize AL, DS, GV, and PL, which are related to the computational and application aspects. Though C3 is similar to C2, the universities in C3 encourage the application aspect.
- C4 has no prioritized or deprioritized KA. In other words, the universities in C4 tend to include all the KAs evenly. It shows that the universities in C4 provide "balanced" curricula.

Similarly, **Fig. 9** shows the three principal components on the KAs. The three components account for over 50 percentage of the total variance. Figure 9 suggests the properties of the axes as follows:

- P1: The KAs with higher values are HCI, SE, and SP, which are related to the human aspect. The other KAs are related to the computational aspect. Those observations suggest that P1 could be interpreted as the weight between the human aspect and the computational one.
- P2: The KAs with higher values are GV and HCI, which are related to the application aspect. On the other hand, those with lower values are CN, which are related to the theoretical aspect. It shows that P2 could be interpreted as the weight between the application aspect and the theoretical one.
- P3: The KAs with higher values are AL, PL and SP, which are related to the software aspect. On the other hand, those with lower values are AR, which are related to the hardware aspect. Therefore, P3 seems to correspond to the weight between the software aspect and the hardware one.

Table 8 Relation of clusters and countries of locations: This is the contingency table between country codes and the four clusters (C1-C4).

country code	us	uk	au	ca	jp	sg	hk	ch	kr	be	dk	fr	se	nl	de	Total
C1	1	1	4	0	0	0	1	0	0	1	0	0	0	0	0	8
C2	1	0	0	2	2	0	0	1	0	0	1	0	0	0	0	7
C3	2	3	0	0	0	0	0	1	1	0	0	1	1	0	0	9
C4	17	0	0	0	0	2	1	0	1	0	0	0	0	1	1	23
Total	21	4	4	2	2	2	2	2	2	1	1	1	1	1	1	47

Note that the properties of the axes P1 and P2 are consistent with those of the clusters C1, C2, and C3. In addition, we will show that the discovered clusters are strongly related to the countries. **Table 8** shows the contingency table between the countries and the extracted four clusters (C1-C4). The p-value of the Chi-square test of this table was 1.8×10^{-5}. It shows that the clusters significantly depend on the countries. C1 includes all the universities in Australia. C2 includes all the universities in Canada and Japan. C3 and C4 include almost all the universities in UK and US, re-

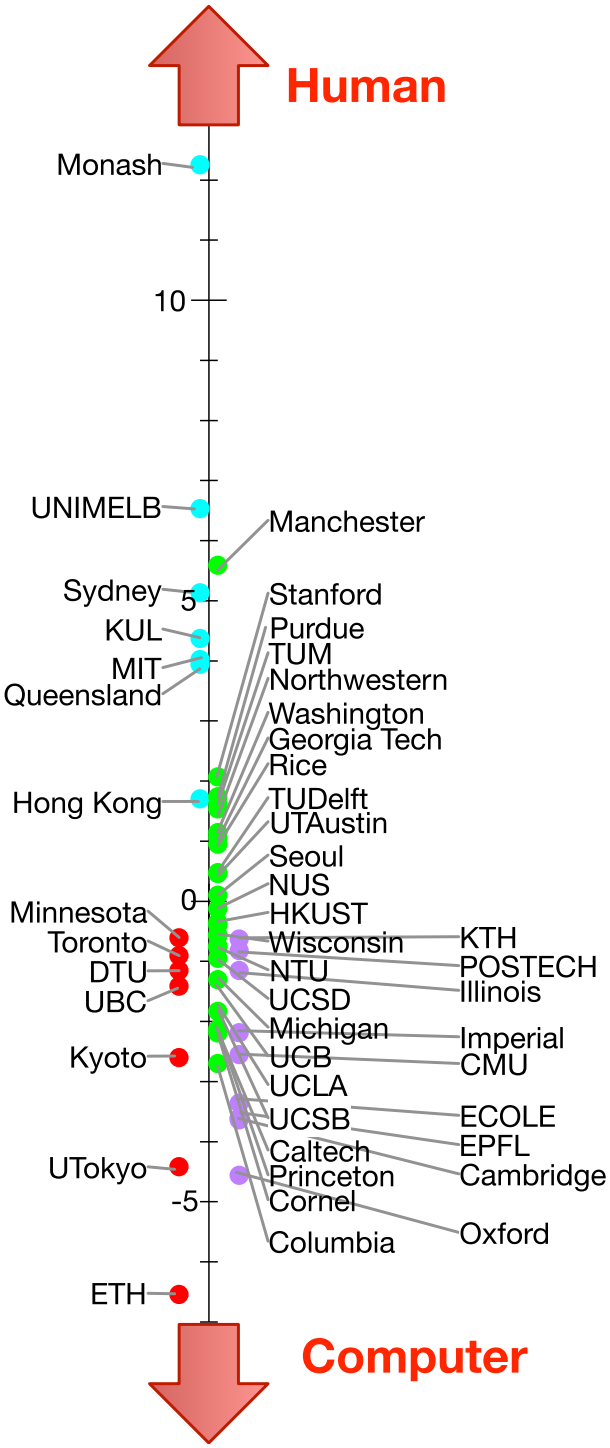

Fig. 6 One-dimensional plot of the universities along the first principal component where the colors correspond to the clusters (C1:cyan, C2:red, C3:magenta, C4:green).

Table 9 Averaged distance within each country and that over the world.

within country	over the world
0.0644	0.0777

spectively. It is surprising because no information about the locations and the countries is used for constructing the hierarchical cluster tree. Regarding the distance among the universities, **Table 9** shows that the averaged distance among the curricula within a country is smaller than the averaged distance over the world. The p-value in the single-tailed paired Student's t-test for the averaged distances was 3.0×10^{-15}. Thus, it was verified statistically that the universities in the same country tend to be localized in the KA space.

6.4 Extraction of Actual Core-Tiers by NMF

Here, the Core-Tiers of KAs are estimated from the actual curricula by the non-negative matrix factorization (NMF) [13]. It is one of the most important processes in the curriculum design to allocate the appropriate weights to the KAs in a curriculum. In contrast to PCA, the factors extracted by NMF are guaranteed to be non-negative. Therefore, they are easily used as the weights of the KAs. The original CS2013 provides two Core-Tiers of KAs (in **Fig. 10**) and recommends that a weighted sum of the two Tiers is utilized in order to design an appropriate curriculum.

The Core-Tiers of CS2013 are constructed by a bottom-up approach where the importance of many topics in each KA is discussed and estimated. Figure 10 shows the core hours for the KAs in the first and second Core-Tiers. For example, CS2013 recommends that a CS curriculum should include the topics in SDF and it should take 43 hours to complete those topics. Though they are derived from detailed discussions, it is hard to employ them in practice because they fluctuate largely and there seems no clear relation among the KAs. Actually, there was no actual curriculum represented suitably by a weighted sum of Core-Tiers of CS2013. Now, the Core-Tiers of KAs are estimated from the actual curricula instead of CS2013 by utilizing NMF, which is given as the following model

$$\boldsymbol{X} \simeq \boldsymbol{W}\boldsymbol{H} \tag{6}$$

where $\boldsymbol{X} = (x_{ij})$ was given as an 47×18 matrix (note that 47 is the number of curricula and 18 is that of KAs). Each row of $\boldsymbol{X}$ was given as a curriculum in the KA space. NMF estimates $\boldsymbol{W} = (w_{ik})$ and $\boldsymbol{H} = (h_{kj})$ so that $\boldsymbol{X}$ is as near to $\boldsymbol{W}\boldsymbol{H}$ as possible under the constraints that all the elements of $\boldsymbol{W}$ and $\boldsymbol{H}$ are non-negative. In order to extract the two factors, $\boldsymbol{W}$ and $\boldsymbol{H}$ were given as 47×2 and 2×18 matrices. Therefore, $\boldsymbol{W}$ and $\boldsymbol{H}$ correspond to the factors on the curricula and those on the KAs, respectively. The best estimation over 100,000 replicates was employed for avoiding local minima. Moreover, the constraints of $\sum_j h_{1j} = 1$ and $\sum_j h_{2j} = 1$ were added. The ordering of the rows of $\boldsymbol{H}$ was determined so that $\sum_i w_{i1}\ (= 30.5) > \sum_i w_{i2}\ (= 16.6)$, which means that the first row of $\boldsymbol{H}$ is dominant. **Figure 11** shows the ratios of KA in the first and second rows of $\boldsymbol{H}$. It shows that there are a few salient KAs in each tier and the other KAs are distributed approximately uniformly. The ratios of AL, CN, GV, IS, and PL are relatively high in the first factor. It suggests that the first factor emphasizes the application-programming aspect. On the other hand, the ratios of HCI, SE, and SP are relatively high in the second aspect. It means that the second factor emphasizes the human-social factor. In summary, the dataset from the world's top-ranked universities shows that the appropriate curricula should include all the KAs and it should emphasize the application-programming and human-social aspects. Note that the factors are consistent with the components of PCA in Fig. 9 though there are slight differences. The weighting of the two factors characterizes each curriculum. These guidelines seem to be utilized much more easily

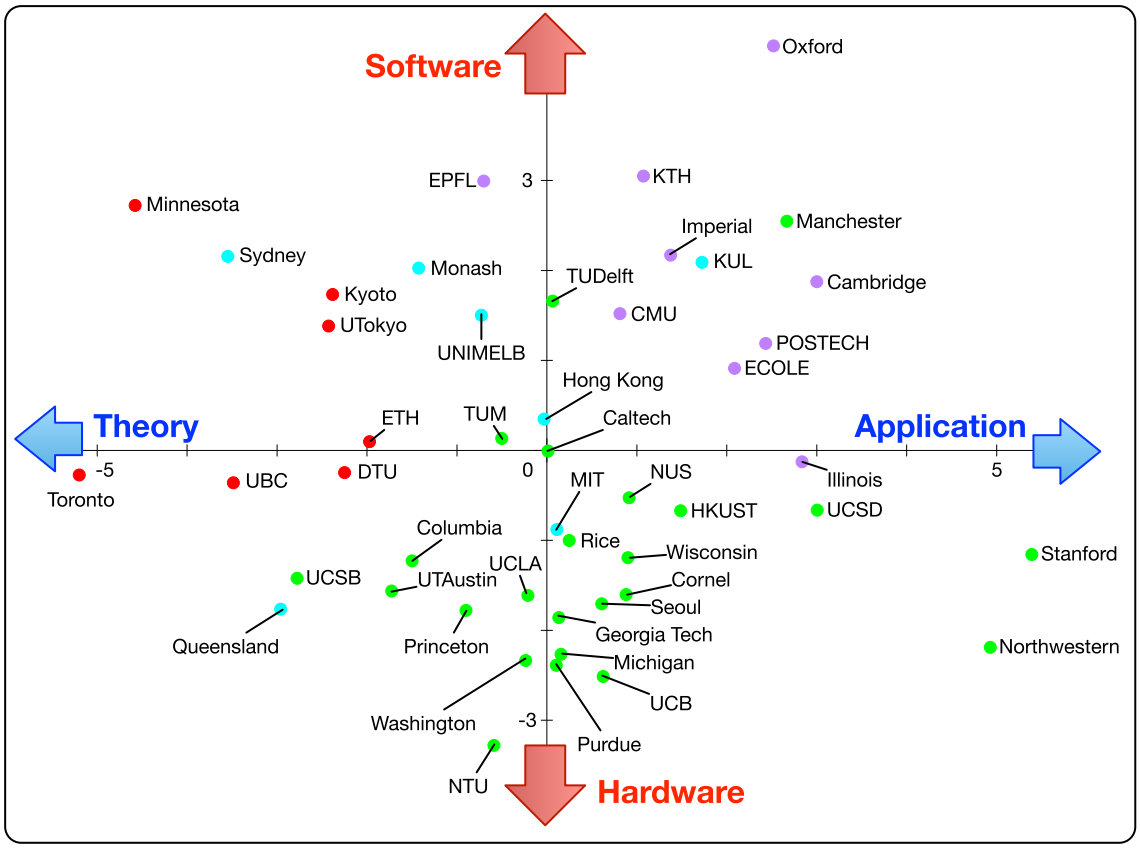

Fig. 7 Two-dimensional plot of the universities along the second and third principal components where the colors correspond to the clusters (C1:cyan, C2:red, C3:magenta, C4:green).

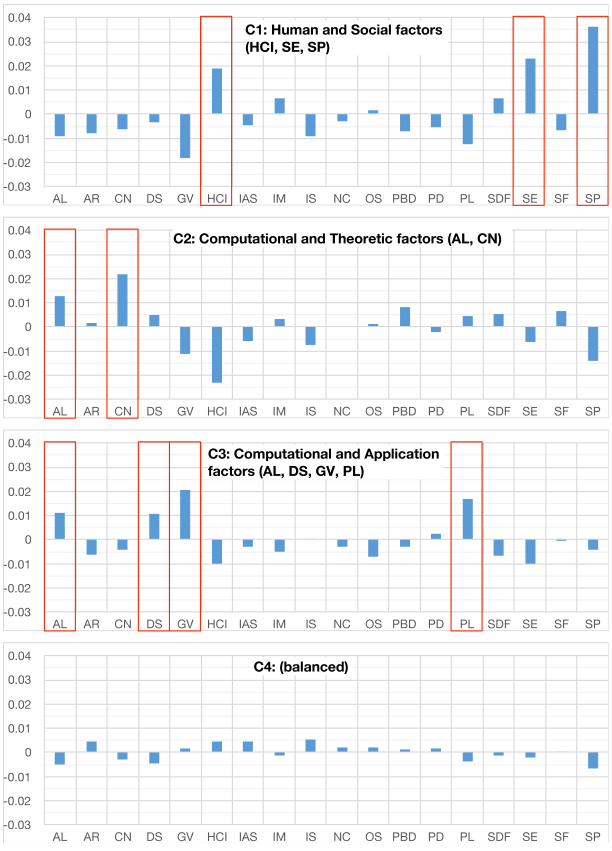

Fig. 8 The center of each cluster in the KA space.

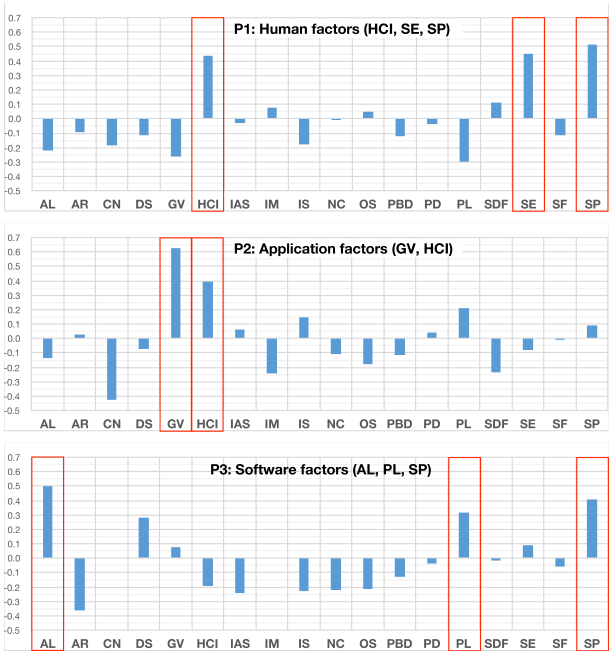

Fig. 9 The three principal components in the KA space.

than the Core-Tiers of CS2013 shown in Fig. 10.

7. Discussions

Here, we discuss the two key features of the actual syllabi, which were discovered quantitatively and statistically from the above observations.

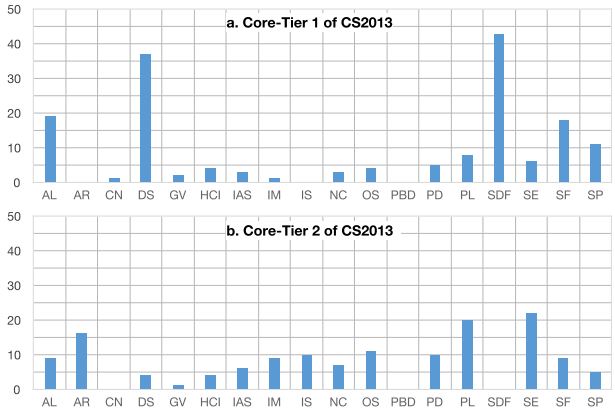

Fig. 10 Bar graphs of the core hours of KAs in Core-Tier1 and Core-Tier2 provided by CS2013.

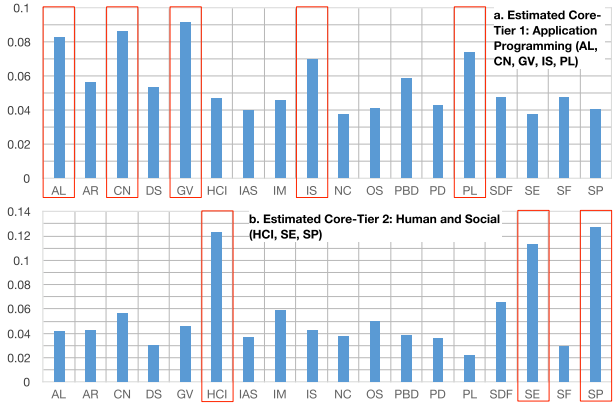

Fig. 11 Bar graphs of the ratios of KAs in Core-Tier1 and Core-Tier2 estimated from actual syllabi.

First, HCA and PCA discovered that there is a strong relation between the clusters of the curricula and the countries. In other words, we could observe a kind of locality bias in the curricula of the leading universities over the world. This observation does not assert that the curricula should be designed to be similar to those in the neighborhood areas. On the one hand, it may be useful for promoting the collaboration in the prioritized fields in each country. From the viewpoint of students, on the other hand, it may narrow their choices because studying abroad is expensive for many students even today. Actually, there were often some exceptions in each country. Nevertheless, this seems to be a key feature to be considered carefully for designing an appropriate curriculum. Though the locality bias may have been found empirically, the proposed approach discovers it quantitatively and statistically significant. Moreover, the weights of the KAs in each area can be estimated quantitatively.

Second, NMF discovered that the actual curricula can be regarded as a "combination of two simple factors." The first and second factors correspond to the "application-programming" and "human-social" aspects, respectively in Fig. 11.

Lastly, we propose the following guidelines for designing a CS curriculum from the above features:

- Determine the weights of the "application-programming" aspect and the "human-social" one. Then, give roughly the weights of all the KAs in the curriculum.
- Investigate the CS departments in the neighborhood area and estimate the corresponding cluster. Then, modify the weights of KAs according to the designing policy which decides whether the curriculum follows the regional tendency or not.

The proposed guidelines may be helpful to curriculum designers utilizing CS2013 in the following two points. One is that the proposed two factors (the "application-programming" and "human-social" aspects) are clearer and easier to use than the Core-Tiers of CS2013. Another is that the proposed guidelines can consider the regional effects which are neglected in CS2013, and helps faculty members and instructors to design a characteristic curriculum.

8. Conclusion

In this paper, we applied a curriculum analysis method to investigating the actual curricula offered by CS departments of the top-ranked universities. The analysis method projects each curriculum to the KA space by ssLDA. By utilizing the three well-known data analysis methods (HCA, PCA, and NMF), we discovered the two important features of the actual curricula: "locality bias" and "combination of several simple aspects." We also proposed the guidelines for designing an appropriate curriculum on the basis of the discovered features. We are now using the analysis method to the set of actual syllabi directly and investigating the interconnections among the KAs.

We are now planning to develop a public web-based tool implementing the proposed analysis method, and to evaluate the usefulness of our proposed guidelines by providing them widely for curriculum designers. We are also planning to collect a larger number of actual curricula in a semi-automated manner.

Acknowledgments This work was supported by JSPS KAKENHI Grant Number 17H01837.

References

[1] ACM/IEEE-CS Joint Task Force on Computing Curricula: Computer Science Curricula 2013, Technical Report, ACM Press and IEEE Computer Society Press (2013).
[2] ACM/IEEE-CS Joint Task Force on Computing Curricula: Computer Engineering Curricula 2016, Technical Report, ACM Press and IEEE Computer Society Press (2016).
[3] Tucker, A.B., Barnes, B.H., et al.: Computing Curricula 1991 Report of the ACM/IEEE-CS Joint Curriculum Task Force (1990).
[4] Blei, D.M. and McAuliffe, J.D.: Supervised Topic Models, *Advances in Neural Information Processing Systems 20* (*NIPS 2007*) Platt, J.C., Koller, D., Singer, Y. and Roweis, S.T. (Eds.), pp.121–128, Curran Associates, Inc. (2008).
[5] Blei, D.M., Ng, A.Y. and Jordan, M.I.: Latent Dirichlet Allocation, *Journal of Machine Learning Research*, Vol.3, pp.993–1022 (2003).
[6] Gluga, R., Kay, J. and Lister, R.: PROGOSS: Mastering the curriculum, *Proc. Australian Conference on Science and Mathematics Education* (*formerly UniServe Science Conference*) (2012).
[7] Harvard Graduate School of Education: The Collaborative Curriculum Design Tool (CCDT), available from ⟨http://learnweb.harvard.edu/ccdt/⟩ (accessed 2010-02-22).
[8] Ida, M.: Textual information and correspondence analysis in curriculum analysis, *Proc. 18th international conference on Fuzzy Systems*, pp.666–669, IEEE Press (2009).
[9] Kawintiranon, K., Vateekul, P., Suchato, A. and Punyabukkana, P.: Understanding knowledge areas in curriculum through text mining from course materials, *2016 IEEE International Conference on Teaching, Assessment, and Learning for Engineering* (*TALE*), pp.161–168 (online), DOI: 10.1109/TALE.2016.7851788 (2016).
[10] Marshall, L.: A Comparison of the Core Aspects of the ACM/IEEE Computer Science Curriculum 2013 Strawman Report with the Specified Core of CC2001 and CS2008 Review, *Proc. 2nd Computer Science Education Research Conference*, *CSERC '12*, pp.29–34, New York, NY, USA, ACM (2012).

[11] Méndez, G., Ochoa, X. and Chiluiza, K.: Techniques for Data-driven Curriculum Analysis, *Proc. 4th International Conference on Learning Analytics And Knowledge*, *LAK '14*, pp.148–157, New York, NY, USA, ACM (2014).

[12] Ota, S. and Mima, H.: Machine Learning-based Syllabus Classification toward Automatic Organization of Issue-oriented Interdisciplinary Curricula, *Procedia - Social and Behavioral Sciences*, Vol.27, pp.241–247 (2011).

[13] Paatero, P. and Tapper, U.: Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values, *Environmetrics*, Vol.5, No.2, pp.111–126 (1994).

[14] Pedroni, M., Oriol, M. and Meyer, B.: A framework for describing and comparing courses and curricula, *Proc. 12th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, *ITiCSE '07*, pp.131–135, New York, NY, USA, ACM (2007).

[15] Pinar, W.F., Reynolds, W.M., Slattery, P. and Taubman, P.M.: *Understanding Curriculum: An Introduction to the Study of Historical and Contemporary Curriculum Discourses*, Peter Lang Pub Inc. (1995).

[16] Sekiya, T., Matsuda, Y. and Yamaguchi, K.: Analysis of Computer Science Related Curriculum on LDA and Isomap, *ITiCSE'10, Proc. 15th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, pp.48–52 (2010).

[17] Sekiya, T., Matsuda, Y. and Yamaguchi, K.: Development of a Curriculum Analysis Tool, *ITHET 2010, 9th International Conference on Information Technology Based Higher Education and Training*, pp.413–418 (2010).

[18] Sekiya, T., Matsuda, Y. and Yamaguchi, K.: Analysis of Computer Science Related Curriculum, *Summer Symposium in Shizukuishi 2013*, *SSS2013*, pp.33–40 (2013). (in Japanese).

[19] Sekiya, T., Matsuda, Y. and Yamaguchi, K.: Curriculum Analysis of CS Departments Based on CS2013 by Simplified, Supervised LDA, *Proc. 5th International Conference on Learning Analytics and Knowledge*, *LAK '15*, pp.330–339, New York, NY, USA, ACM (2015).

[20] Szabo, C. and Falkner, K.: Neo-piagetian Theory As a Guide to Curriculum Analysis, *Proc. 45th ACM Technical Symposium on Computer Science Education*, *SIGCSE '14*, pp.115–120, New York, NY, USA, ACM (2014).

[21] The CS2008 Review Taskforce: Computing Science Curriculum 2008: An Interim Revision of CS2001, available from ⟨http://www.acm.org/education/curricula/ComputerScience2008.pdf⟩.

[22] The Joint Task Force on Computing Curricula IEEE Computer Society/Asccociation for Computing Machinery: Computing Curricula 2001 Computer Science, available from ⟨http://www.acm.org/education/curric_vols/cc2001.pdf/view⟩.

[23] Times Higher Education (THE): World University Rankings, available from ⟨https://www.timeshighereducation.com/world-university-rankings⟩ (accessed 2016-08-08).

[24] Tungare, M., Yu, X., Cameron, W., Teng, G., Perez-Quinones, M.A., Cassel, L., Fan, W. and Fox, E.A.: Towards a syllabus repository for computer science courses, *Proc. 38th SIGCSE Technical Symposium on Computer Science Education*, pp.55–59 (2007).

[25] Walker, H.M. and Rebelsky, S.A.: Using CS2013 for a Department's Curriculum Review: A Case Study, *Journal of Computing Sciences in Colleges*, Vol.29, No.5, pp.138–144 (2014) (online), available from ⟨http://dl.acm.org/citation.cfm?id=2600623.2600650⟩.

[26] Wang, C., Blei, D. and Li, F.-F.: Simultaneous image classification and annotation, *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009*, pp.1903–1910, IEEE (2009).

[27] Wiske, M.S., Sick, M. and Wirsig, S.: New technologies to support teaching for understanding, *International Journal of Educational Research*, Vol.35, No.5, pp.483–501 (2001) (online), available from ⟨http://www.sciencedirect.com/science/article/B6VDF-458P6TG-1/2/6414d7ffbc5d8d016382dc50b02b73f7⟩.

[28] Yu, X., Tungare, M., Fan, W., Yuan, Y., Pérez-Quinones, M., Fox, E., Cameron, W. and Cassel, L.: Automatic syllabus classification using support vector machines, *Handbook of Research on Text and Web Mining Technologies*, Information Science Reference (2008).

Yoshitatsu Matsuda received his Ph.D. from the University of Tokyo, Japan in 2002. He is currently a research fellow in the University of Tokyo. His research interests include independent component analysis, massive data analysis, and self-organizing neural networks. He is a member of IPSJ.

Takayuki Sekiya received his Ph.D. from the University of Tokyo in 2000. Currently, he works as a research associate for Information Technology Center, the University of Tokyo. He is interested in computer systems to support educational activities in higher educational institutions. He is a member of IPSJ.

Kazunori Yamaguchi received his B.S., M.S., and Doctor of Science degrees in information science from the University of Tokyo, in 1979, 1981, and 1985, respectively. Currently, he is a professor of the University of Tokyo. His research interests are in data models and data analysis. He is a member of IPSJ.