

Jupyter Notebookを使ったプログラミング模擬演習の評価

○桑田 喜隆^{†1} 石坂 徹^{†1} 小川祐紀雄^{†1} 政谷好伸^{†2}
長久勝^{†2} 横山重俊^{†2†3} 浜元信州^{†3}

^{†1} 室蘭工業大学

^{†2} 国立情報学研究所

^{†3} 群馬大学

Evaluating Jupyter Notebook in a trial class of programming

Yoshitaka Kuwata^{†1}, Toru Ishizaka^{†1}, Yukio Ogawa^{†1}, Yoshinobu Masatani^{†2},
Masaru Nagaku^{†2}, Shigetoshi Yokoyama^{†2†3}, and Nobukuni Hamamoto^{†3}

^{†1} Muroran Institute of Technology, Japan

^{†2} National Institute for Informatics, Japan

^{†3} Gumma University, Japan

概要

筆者らはJupyter Notebookを使ったプログラミング教育を提案している。Jupyter Notebookの実行履歴を解析することで、学生の状況を把握し適切なアドバイスを行うことが可能になる。本稿では、14人の学生および4人の教員が参加したプログラミングの模擬演習の分析により得られた知見を述べる。

Abstract

The authors proposed to use Jupyter Notebook for programming exercise. By analyzing the execution record of student's Notebooks, we can recognize their status of exercises. Then we can provide useful advice for helping to understand the programming. In this paper, we report the analysis of data obtained from a trial class, in which 14 students and four teachers are involved.

1. はじめに

1.1 プログラミング教育における課題

プログラミング教育の重要性が再認識されている。例えば、2020年度より小学校でプログラミングに関する教育が必修化されることになっている。高等教育においても、その重要性は認識されており、IT教育の一環として一般教養として必修化している大学も存在する。

筆者らは、一般教養科目としてプログラミングを一部として取り入れた授業を実施する中で、以下の点が課題としてあげられる。

- 学生の知識が異なる
- 学生の能力にばらつきがある

1.2 プログラミング教育に関する3つの仮説

筆者らは参考文献2で述べたように、プロ

プログラミング教育について以下の仮説を設定している。

- (1) 座学に加え演習を実施することで学生の理解度が向上する
- (2) 一般的に、演習が進むほど学生の理解度が低下する
- (3) 理解度に応じた指導を行うことで、理解度の低下を防ぐことができる

上記の検証のため、教授方法や演習の実施方法の効果を定量的に測定するが重要であると考えられる。定量評価結果をもとに、継続的に改善を行うことが可能になる。

従来は試験結果や課題の正解率などで理解度を測る方法がとられていた。しかし、手作業で実施する場合、採点や分析に手間および時間がかかり、学生へのフィードバックが遅くなる点が

¹ Yoshitaka Kuwata
室蘭工業大学
北海道室蘭市水元町2-7-1

kuwata@mmm.muroran-it.ac.jp

問題であった。そこで、筆者らは Jupyter Notebook に注目して定量的なデータをリアルタイムに取得する方法を検討した。

2. Jupyter Notebook を活用したプログラミング演習

上記の課題に対し、筆者らは Jupyter Notebook を使ってプログラミング演習を実施することを提案した。(参考文献 2)

Jupyter Notebook の実行履歴をリアルタイムで収集し、分析することが可能であるため、タイムリーに学生にフィードバックを行うことができる。例えば、授業の演習時間の間に、多くの学生がわからない箇所がわかれば、その場で補足説明をすることで理解を助けることができる。

教材や教授方法に関しても、定量的に効果測定も可能になり、より効率の良い教育が可能になると考えられる。また学生ごとに進捗や理解が異なり、それに合わせたアドバイスが可能になるため、より効率の高い教育が可能になる。

3. 模擬授業を使った評価

Jupyter Notebook を利用したプログラミングの授業を実施するにあたり、少人数の学生を対象にした模擬授業を実施した。本稿では、模擬授業の実施の詳細について示したのち、実施結果および得られた知見について述べる。

3.1 模擬授業の目的

「プログラミング入門」は新たに全コースの必修科目として設けられた授業であり、1年の後期に開設される。プログラミングに必要な概念を理解し、基礎的なプログラムを作成することができるようになることを目標としている。

コースの開設に先立ち、以下の目的で模擬授業を実施した。

- (1) 教科書や教材の内容確認
 - (2) 授業方法に関する評価
 - (3) 授業内容のレベル確認
- なお、本稿で報告する内容は(2)に関連する。

3.2 模擬授業の内容

表 1 に模擬授業 7 回の内容を示す。

プログラミング入門で予定している15回のうち、前半の7回を実施した。実際の授業では、模擬授業の内容に加えて、グラフの利用、アルゴリズム2、シミュレーション1, 2, および総合演習4回を行う。

第 1 回から第 4 回までは本来の授業と同様に週 1 回とした。第 5 回から第 7 回は日程確保の都合で、集中講義として 1 日で実施した。

授業時間は各回とも 90 分であるが、授業後に 30 分を宿題のための時間として割り当てた。

表 1 模擬授業の内容

回	実施日	模擬授業内容
第 1 回	6 月 13 日	イントロダクション
第 2 回	6 月 20 日	プログラミングの基本概念
第 3 回	6 月 27 日	条件判断
第 4 回	7 月 4 日	制御構造, 繰り返し
第 5 回	7 月 6 日	データ構造
第 6 回	7 月 6 日	関数
第 7 回	7 月 6 日	アルゴリズム 1

表 2 に参加者の役割と人数を示す。

表 2 参加者の役割と人数

アクター	人数	実施内容	選定条件
教員	4	講義, 演習/課題支援, 課題採点	授業の担当教員
TA	2	演習の支援, 採点補助	情報系の大学院生
学生	12	授業を受講	非情報系の学部 2 年生

学生役は学内でアルバイトとして募集した。単位取得とは関係はないが、できるだけ実際の授業を受けるつもりで参加して欲しいと依頼している。

TA 役もアルバイトで募集したが、プログラミングに関する知識が前提のため、情報系の大学院生に依頼した。

教員役は、実際の授業を担当する予定の教員 4 名が全回参加した。講義は交代で実施し、演習および課題の支援は全員で実施した。

3.3 模擬授業の進め方

授業の流れと役割を図 1 に示す。

左側の流れ図は、授業のアクティビティを示している。また、図の右側の黒丸は主として取り組むアクターを示している。

授業は事前学習、講義、演習、課題およびその評価から構成される。各項目の詳細については表 3 にまとめた。

なお、事前学習もアルバイトの一部として実施しており、簡易な報告書を提出してもらった。また、欠席する場合には、次の回までに自習してもらった。

なお、模擬授業では課題を実施する時間を授業後に設けたため、宿題となった課題はなかった。

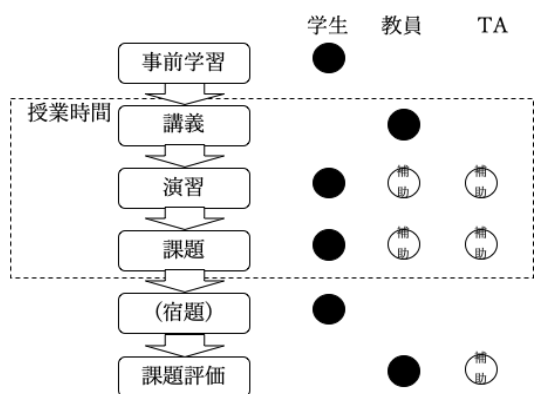


図 1 授業の流れと役割

表 3 授業の進め方

項目	想定時間	内容
事前学習	-	●学生が、教科書の該当項目を読んで、内容を理解しておく
講義	15 分	●教員が、ポイントとなる項目を説明する。
演習	45 分	●予め用意しておいた演習用 Notebook を学生が各自で解くことで理解を深める。 ●TA および教師は学生からの質問を受ける。
課題	30 分	●演習内容をまとめた、課題用 Notebook を学生が各自で解いて、提出する。 ●教員または TA が課題用 Notebook を評価し、次回以降学生にフィードバックする。
(宿題)	-	●授業時間内に課題が終わらなかった場合に宿題とする。
課題評価	-	●教員または TA が課題を評価してフィードバックを行う。

3.4 利用した教材

使用した教材および構成を表4に示す。

- 教科書
事前学習のため、参考文献 6 のドラフト版の該当する章を事前に印刷し配布した。
- 説明スライド
教科書の重要なポイントを講義で説明するために用意した。学生がパソコンから参照できるように学内の LMS(Moodle)で配布した。
- 演習用 Notebook および課題用 Notebook
教科書に合わせて新たに Notebook 準備した。学

生がすぐに演習を行えるように、各学生のフォルダ上に直接配布した。

表 4 教材の構成

回	教科書 (文献 6)	説明スライド	演習用 Notebook	課題用 Notebook
第 1 回	第 1 章	27 ページ	10 セル	課題なし
第 2 回	第 2 章	17 ページ	27 セル	1 セル
第 3 回	第 3 章	19 ページ	25 セル	1 セル
第 4 回	第 4 章	14 ページ	20 セル	3 セル
第 5 回	第 5 章	16 ページ	31 セル	5 セル
第 6 回	第 6 章	10 ページ	21 セル	5 セル
第 7 回	第 8 章	17 ページ	13 セル	5 セル

なお、Notebook の規模を示す指標として、Code セル数を示している。例えば、10 セルの演習に対して学生は 10 回の回答が必要になる。但し、セルによって難易度が異なる。例えば、Notebook には既に記入済みで評価するだけのセルや、数行のプログラムを作成するセルなどが含まれているが、どちらも 1 セルとして集計している。

3.5 演習環境

図 3 に演習環境の構成を示す。

今回の模擬授業は CoursewareHub (参考文献 4, 5) と学内の LMS の両方を利用した。

- 学生は、学内の教育パソコン(シンクライアント)から演習を実施した。
- スライド教材の配布には学内の LMS(Moodle)を使用した。
- Jupyter Notebook の環境は、SINET 経由で NII の CoursewareHub を使用した。

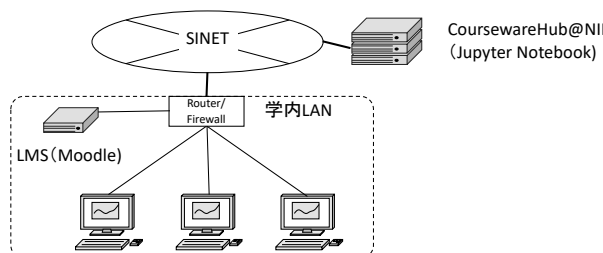


図 3 演習環境の構成

CoursewareHub は Jupyter Hub の機能を拡張しており、教材の配布や回収などが容易に実施できる。また評価の履歴やエラーなどの情報を取得することが可能になっている。クラウド上の仮想マシン上で実行されるため、履歴を一括して取得してリアルタイムに分析することも可能である。

4. 取得した情報

模擬授業で取得したデータは次の通りである。

- (1) 事前アンケート, 事後アンケート(各 1 回)
コース開始時と終了時に, Moodle を使いプログラミングに関する知識などの主観的な情報を収集した
 - (2) 事前学習レポート(7 回)
毎回, 事前学習の中での不明点などを簡単なレポートとして提出してもらった
 - (3) 模擬授業に関するアンケート(7 回)
Moodle を使い各回の授業に対する主観評価を収集した
 - (4) Notebook の実行履歴(7 回分)
模擬授業 7 回で, 約 11,000 件の Notebook の実行履歴データを CoursewareHub で収集した
 - (5) 演習 Notebook および課題 Notebook(7 回分)
CoursewareHub の各学生のフォルダに保存されている Notebook を収集した
- 本稿では(4)で最も参加者が多い第 4 回の解析結果について説明する。また, (3)の主観的なデータについては, 考察としてその一部を述べる。

5. Jupyter Notebook から取得した情報の評価

5.1 進捗グラフによる演習状態の把握

既に文献 2 で提案したように, Notebook の実行履歴を利用した進捗グラフによって, 演習の進捗状況を把握することができる。

図 4 に第 4 回の進捗グラフを示す。

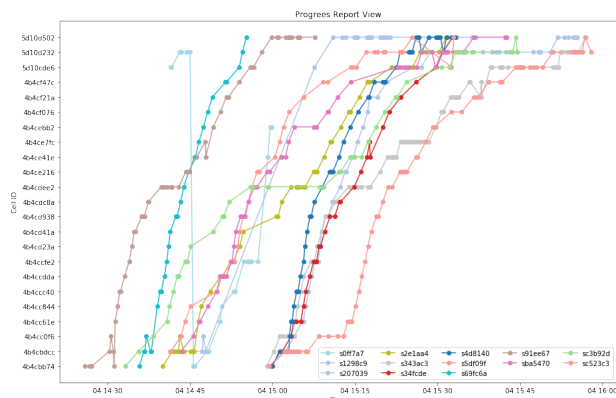


図 4 進捗グラフの例 (第 4 回)

進捗グラフの横軸は「セルの評価時刻」を, 縦軸は「セル ID」を示している。Notebook の上位に位置するセルは, グラフの下に示される。Notebook のセルを上から順番に評価すると右上がりのグラフとなる。

色の異なるグラフは各学生の履歴を示してお

り, 第 4 回は TA1 名を含めて 13 人のログが記録されていた。進捗グラフでは, 約半数の学生が演習開始の 15:00 より前(講義中)に演習を開始していることが分かる。

また, グラフ最上位の 3 セルは課題に対応しており, 授業終了時刻である 16:05 までに全員が課題まで取り掛かっていることが分かる。

学生の進捗グラフを表示する Notebook を再評価することにより, 教員は授業の途中で学生の進捗状況を手軽に参照することができる。これにより, 必要に応じて進捗状況の遅い学生に対して支援を行うなどの対応が可能である。

なお, 模擬授業では教員が進捗グラフを参照したものの, 学生の人数が少なかったこともあり, 進捗グラフを使ったフィードバック等はない。

5.2 セルの評価数による Notebook の評価

セルごとの評価数を使うと, Notebook 教材の評価をすることができる。

図 5 に第 4 回の演習用 Notebook および課題用 Notebook のセルごとの評価数, エラー総数を示す。

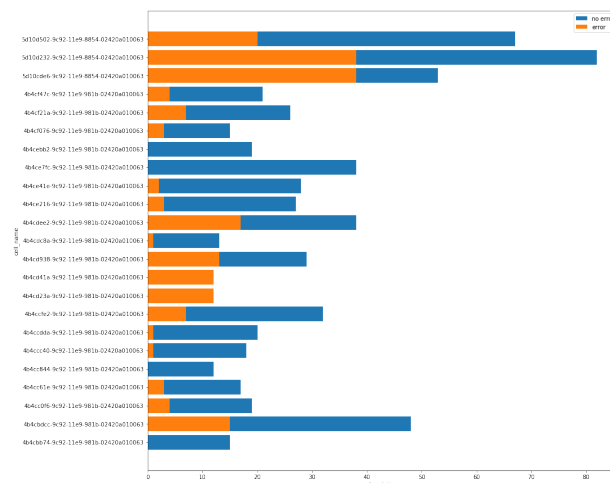


図 5 セルごとの評価数,エラー数(第 4 回)

縦軸はセル ID を, 横軸はその評価数およびエラー数を示している。

第 4 回の Notebook の最初のセル(グラフ一番下)は予めプログラムを入力済みで評価するだけであったため, エラーは出ていない。2 番目のセルでは, 1 番目のセルを改造して試す問題としたため, 評価回数やエラーも多くなっていることが分かる。

更に, グラフの最上位の 3 セルは課題に相当しており, 評価数, エラー数とも多いことが分

かる。これは、演習に比べ課題の難易度を高く設定しているためであり、出題者の意図と一致する。

このように、評価回数およびエラー回数を確認することで、出題時に期待した操作を行なっているかを確認することができる。もし演習中に期待と異なる挙動が観測された場合、教員から学生全体に補足説明を加えることができる。また、必要に応じて授業後に Notebook を修正して、次回以降の授業時に改善することも考えられる。

5.3 利用者ごとの進捗パタンの把握

学生ごとの進捗状況の特徴を分析するため、利用者ごとに進捗グラフを作成した。

図 6 に利用者ごとの進捗グラフを示す。図 4 に示したデータを利用者(13 人)ごとにグラフ表示している。横軸が時刻を、縦軸がセル ID を示していることは図 4 と同じである。

利用者ごとの進捗グラフを利用することで、その特徴を詳細に分析することが可能である。例えば、演習を開始した時刻と終了までにかかった時間の 2 象限で学生分類する方法が考えられる。表 5 に分類例とそれぞれのパターンに含まれる人数²を示す。

表 5 利用者の分類例

開始 進捗	早い (15:00 以前)	普通 (15:00 以降)
早い (40 分以内)	(パターン a) 2 人	(パターン b) 2 人
普通 (40 分以上)	(パターン c) 5 人	(パターン d) 3 人

模擬授業は、まず講義でポイントを教えた後に演習を開始することを想定していた。しかし、講義中に演習を開始することを、あえて禁止なかった。これは、学生の知識レベルにばらつきがあり、予習によって内容を十分理解し、授業を聞く必要がないと考える学生の可能性を考えたためである。実際に、7 名が講義中に演習を開始している。(パターン a, c)ただし、講義中に演習を開始しても、講義を全く聞かないわけではなく、聞きながら、演習を実施していたケースもあった。

しかし、この分析から、早く演習を開始した学生が必ずしも早く終了しているわけではないことが分かった。(パターン c)講義をよく聞かずに演

習を開始したため、途中で分からなくなって再度教科書などを見直したことも考えられる。これに対して、明示的に講義中の演習を禁止する、または演習を実施できなくしておくなどの方が、効率が良い可能性もある。ただし、本評価だけでは不十分であり、更に分析が必要であると考えられる。

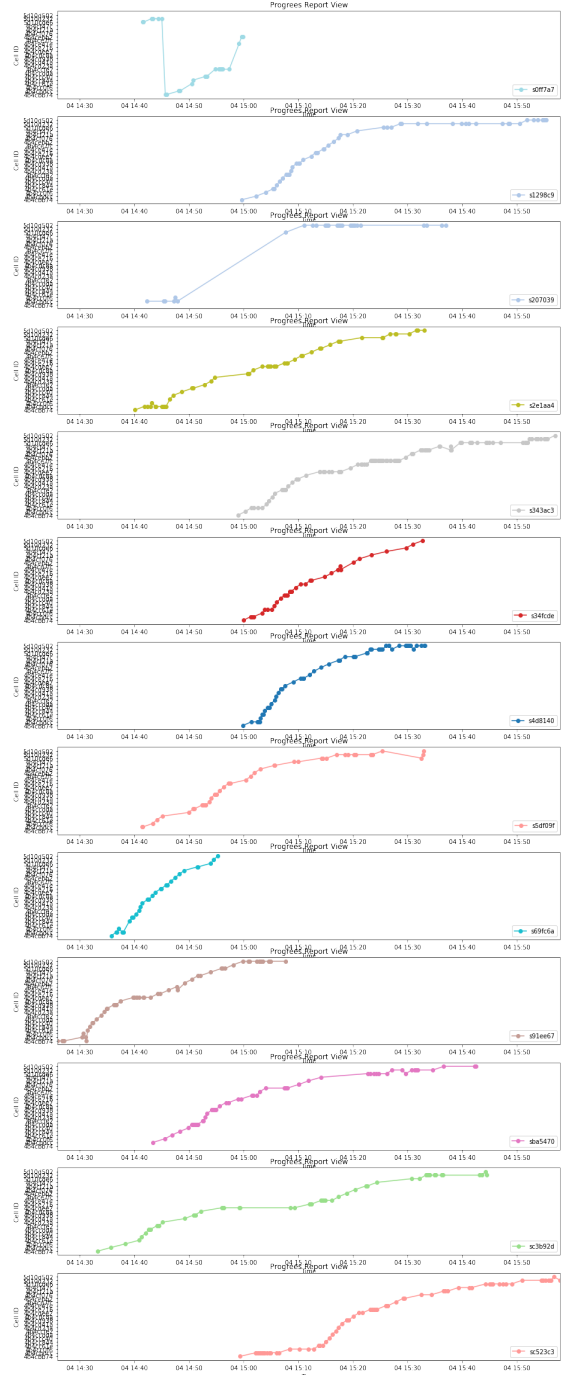


図 6 利用者ごとの進捗グラフ(第 4 回)

² 進捗グラフの一番上の利用者は TA の 1 人であるため今回の分析対象から外した。

5.4 相対進捗グラフ

図 7 に開始時刻からの相対時刻で示した進捗グラフを示す。

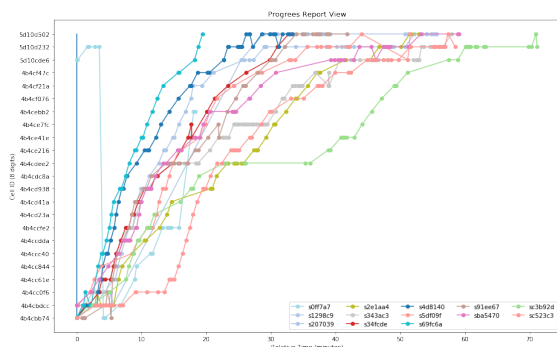


図 7 相対進捗グラフ

開始後 30 分程度で演習を終えて、課題に取り掛かっている学生が多いことが分かる。それ以上時間がかかっている学生は課題にも時間がかかっているため、個別指導の実施を検討する必要があると考える。

なお、相対進捗グラフは異なる時間帯に開催されたクラスや異なる回の比較に役に立つと考えている。

6. 考察(主観評価)

模擬授業に関するアンケートの分析結果について述べる。模擬授業では、各回の終了時に理解度についての質問をしている。

これまでの授業内容は、どの程度理解していますか？

- 1: 全くわからない
- 2: ほとんど理解していない
- 3: どちらかという、理解していない
- 4: どちらかという、理解している
- 5: ほとんど理解している
- 6: 完璧に理解している

図 8 に、理解度指数の平均値を示した。

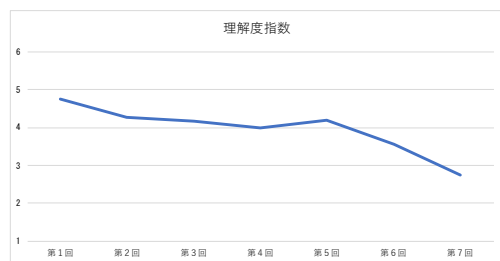


図 8 理解度指数の変化

仮説(2)で想定したように、授業の回が進むごとに理解度指数は低下していることが主観評価によって確認できた。ただし、第 5 回までの低下は緩やかであり、それに比べて第 6 回および第 7 回で急激に低下していることが分かる。特に第

7 回はアルゴリズムの回であり、それまでに比べ難しくなっている。また、第 5 回から第 7 回は集中講義のため、後半になって学生の集中力が低下し、理解の不足につながった可能性もある。いずれにしても、何らかの対策が必要であると考える。

7. まとめと今後の課題

本稿では、Jupyter Notebook を使ったプログラミングの模擬授業で得られた知見について述べた。想定通りに学生の進捗状況や誤りを授業の途中でも把握することが可能であることが確認できた。本手法は学生の状態を把握する方法として有効であると考えられる。

また、模擬授業に参加した学生の主観評価から、授業が進むに従い理解度が低下することも、確認できた。途中でわからない概念が出てきてその後の授業が分からなくなることが一つの原因であると考えられる。理解度の低下を防ぐため、授業の途中で復習のための「総合演習」を設ける予定である。

なお、本稿では第 4 回の結果を中心に述べたが、他の回を含む解析結果については別に報告する予定である。

本研究は JSPS 科研費 (JP18K11561) の「クラウドを活用したプログラミング演習環境に関する研究」の助成を受けたものである。

A. 参考文献

1. Project Jupyter, Project Jupyter Homepage, <http://jupyter.org/> (2019/8/8 参照)
 2. 桑田喜隆, 石坂 徹, 政谷好伸, 長久勝, 横山重俊, 浜元信州, クラウドを利用した対話的なプログラミング教育環境と その評価手法の提案, 第 23 回 人工知能学会 知識流通ネットワーク研究会, 2018 年 9 月 28 日
 3. 桑田喜隆, 石坂 徹, 政谷好伸, 長久勝, 横山重俊, 浜元信州, Jupyter Notebook の実行履歴を活用したプログラミング演習の状況把握, 第 24 回 人工知能学会 知識流通ネットワーク研究会, 2019 年 3 月 8 日
 4. 長久勝, 政谷好伸, 合田, 憲人, Notebook による講義・演習環境の開発, 情報処理学会 教育学習支援情報システム研究会報告 2019-CLE-27, 2019 年 03 月 13 日
 5. Literate Computing for Reproducible Infrastructure, <https://literate-computing.github.io/> (2019/2/18 参照)
 6. 室蘭工業大学情報教育研究会編, Jupyter Notebook で始めるプログラミング 2019, 学術図書出版社, 2019 年 9 月
- ※ 記載されている会社名, 商品名, 又はサービス名は, 各社の商標又は登録商標です。