Redesigning Notebooks for Data Science Education

April Y. Wang
School of Information
University of Michigan
Ann Arbor, MI, USA
aprilww@umich.edu

Steve Oney
School of Information
University of Michigan
Ann Arbor, MI, USA
soney@umich.edu

Christopher Brooks
School of Information
University of Michigan
Ann Arbor, MI, USA
brooksch@umich.edu

ABSTRACT

Computational notebooks enable data scientists to document their exploration and analysis processes through a combination of code, narrative text, visualizations, and other rich media. In this position paper, we discuss the opportunities to improve computational notebook infrastructure to better support data science education. In particular, we propose to redesign of computational notebook environments along three dimensions: 1) supporting real-time group collaboration, 2) facilitating joint discourse over shared context, and 3) encouraging active learning.

KEYWORDS

computational notebooks, data science education, computational narrative, literate programming

INTRODUCTION

The rise of big data has increased the job demand for data scientists, which has been called "the sexiest job of the 21st century" [2]. In addition to the growth of data science degree programs in colleges, there has been a proliferation of data science Massive Open Online Courses (MOOCs). This has scaled

HusDat @ CHI19, May 04–09, 2019, Glasgow, UK
2019.

the access to quality education for learners who are seeking to gain data science skills. On Coursera, there are currently more than 100 data science courses, including a series of specialized training and degree programs. Many of the data science curricula and degree programs have introduced Python programming in computational notebooks, including Jupyter — the most popular tool for interactive data science [9]. Jupyter's design supports exploratory programming [5], where the implementation can not be decided in advance in an open-ended task. This is particularly helpful for data scientists, who need to frequently inspect the output of parts of the code before knowing how to proceed. In addition, Jupyter notebooks allow users to document their exploration process using a combination of code, output, narrative text, visualizations, and other rich media. This supports sharing and reproducing their results. In online learning environments, instructors often use Jupyter notebooks to demonstrate code and output, and couple the notebooks with video lectures and assessments.

In this paper, we describe a vision of how Jupyter (and other computational notebook environments) could be redesigned to better support data science education. In particular, we suggest that computational notebooks should be augmented to:

- support real-time group collaboration,
- facilitate persistent discourse, and
- encourage active learning of content often provided in instructional videos.

REAL-TIME GROUP COLLABORATION

We believe computational notebooks should be improved to better support *collaborative learning*. Collaborative learning is the process of learners working together to solve a problem [3]. In general, collaborative learning is useful for helping learners construct their understanding into a conceptual framework, internalizing knowledge through social discourse, and motivating learning. We believe that integrating group collaboration in data science education can encourage learners to explore variants of solutions and reason about the benefits and trade-offs. Data science is, in part, a practice of exploratory programming where the task is often open-ended with no clear implementation in advance [5]. Thus, the skills to explore and compare variants of solutions are critical for data science learners. In initial pilot studies of collaborative data science learning activities, learners tended to create better and more diverse solutions to a problem than when working alone, though lamented the lack of tool support for collaborative processes [1].

Although tools like Google Colaboratory[2] allow multiple users to edit the same notebook simultaneously, and this may help to maintain a shared understanding and reduce the communication cost, our pilot study of real-time collaborative editing in Jupyter notebooks[3] has revealed several challenges. For example, two users editing the same notebook may amplify the tension between exploration and keeping a clear explanation of the notebook, which has already been identified as a problem in

[1]The preliminary result showed that learners explored 3.5 times more features in a feature engineering task when working in a collaborative setting.

[2]https://colab.research.google.com

[3]This study was undertaken on a custom version of the Juptyer system.

existing single-authored notebooks [10]. Thus, we believe that real-time collaborative editors should (a) support task management such as planning, assignment, and tracking, (b) improve accountability of code cells and awareness of changes, (c) supplement explanations of the exploration process by retrieving related content from peer discussions.

PERSISTENT DISCOURSE

Discussion forums are one of the most popular mechanisms for connecting learners with their peers and instructors asynchronously. The topics discussed by learners and instructors are sometimes general (e.g., clarification of a concept), problem based (e.g., discussions of alternative solutions for an assignment), or logistic (e.g., grading policy) in nature. However, the participation rate on discussion forums is generally low [1]. When posting problems to discussion forums, learners often fail to capture the complete problem context, leaving helpers to have to invest significant effort in setting up the environment (e.g., libraries or imports to use), retrieving the same or a representative dataset, and even run provided code. Prior work has explored the benefit of sharing context through collaborative media curation for improving participation in online learning [4]. Building off of this, we envision the design of a persistent discussion forum based on computational notebooks, where learners can describe their questions within the notebook environment and instructors and peers can easily reproduce issues through execution of the notebook, explore alternative solutions directly in the notebook, and attach comments to fine grained artifacts (such as cells) within the computational narrative itself. We believe such a mechanism can lower the barrier for building shared context in discussion and thus improve learners' participation rate in discussion forums.

ENCOURAGING ACTIVE LEARNING

An integrated approach that embeds comment threads, assessment [7] or interactive multimedia exercises [6] with lecture videos can improve learning efficiency and better engage learners. To date, the primary delivery environment for these features has been the learning management system as a whole or, in some cases, enhanced video where learners engaged in in-video quizzes. Our position is that the primary tool of the data scientist, the computational notebook, is the appropriate place for an integrated learning environment to be built. Instead of instructors recording demonstrates through screen capture and have learners follow along in another window, we argue that the instructor activities (code written at the keypress level, traversals of documentation, etc.) be captured at a fine grained level and "replayed" in the learners' environments. Learners could then pause the playback at any time by simply clicking into the notebook environment and taking control from the automated environment, exploring alternative methods, or expanding on variables or other elements of the machine state. At any time the learners could then save their "counterfactual exploration" and return to the instructional narrative as recorded, exploring how the instructor solved a given problem. Such

explorations could also be shared with peers, either in real-time (e.g., multiple learners watching a single replay together, pausing it to explore the system) or through asynchronous mechanisms (e.g., posting a "walk-though" of alternative explorations to a discussion forum).

AUTHORS' BACKGROUND

The team has a general interest in exploring approaches and tools that effectively teaching programming on MOOCs. Previously, our team has studied ways to improve remote communication about code [8]. In addition, our team has taught a residential course on introductory Python programming and a MOOC on applied data science, where we have integrated Jupyter notebooks as a pedagogical tool. Through the teaching experience, our team gains empirical evidence of the challenges in adapting computational notebooks in data science education. We hope to contribute our understanding of computational notebooks and share our design insights with the human-centered data science community.

REFERENCES

[1] Carlos Alario-Hoyos, Mar Pérez-Sanagustín, Carlos Delgado-Kloos, Hugo A. Parada G, and Mario Muñoz-Organero. 2014. Delving into Participants' Profiles and Use of Social Tools in MOOCs. *IEEE Transactions on Learning Technologies* 7, 3 (July 2014), 260–266. https://doi.org/10.1109/TLT.2014.2311807

[2] Thomas H. Davenport and D. J. Patil. 2012. Data Scientist: The Sexiest Job of the 21st Century. (2012). Issue October 2012. https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century

[3] Jeanne Marcum Gerlach. 1994. Is this collaboration? 1994, 59 (1994), 5–14. https://doi.org/10.1002/tl.37219945903

[4] William A. Hamilton, Nic Lupfer, Nicolas Botello, Tyler Tesch, Alex Stacy, Jeremy Merrill, Blake Williford, Frank R. Bentley, and Andruid Kerne. 2018. Collaborative Live Media Curation: Shared Context for Participation in Online Learning. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, 555:1–555:14. https://doi.org/10.1145/3173574.3174129

[5] Mary Beth Kery and Brad A. Myers. 2017. Exploring exploratory programming. In *2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)* (2017-10). 25–29. https://doi.org/10.1109/VLHCC.2017.8103446

[6] Juho Kim, Elena L. Glassman, AndrÃľs Monroy-HernÃąndez, and Meredith Ringel Morris. 2015. RIMES: Embedding Interactive Multimedia Exercises in Lecture Videos. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15*. ACM Press, 1535–1544. https://doi.org/10.1145/2702123.2702186

[7] Toni-Jan Keith Palma Monserrat, Yawen Li, Shengdong Zhao, and Xiang Cao. 2014. L.IVE: An Integrated Interactive Video-based Learning Environment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, 3399–3402. https://doi.org/10.1145/2556288.2557368

[8] Steve Oney, Christopher Brooks, and Paul Resnick. 2018. Creating Guided Code Explanations with Chat.Codes. *Proc. ACM Hum.-Comput. Interact.* 2, CSCW, Article 131 (Nov. 2018), 20 pages. https://doi.org/10.1145/3274400

[9] Jeffrey M. Perkel. 2018. Why Jupyter is data scientists' computational notebook of choice. *Nature* 563 (2018), 145. https://doi.org/10.1038/d41586-018-07196-1

[10] Adam Rule, Aurélien Tabard, and James D. Hollan. 2018. Exploration and Explanation in Computational Notebooks. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 32, 12 pages. https://doi.org/10.1145/3173574.3173606