(http://csmatters.org) 6 - 4

0b110 - 0b100

Diversity Makes For Better Solutions

Unit 6. Data Visualization

Revision Date: Feb 06, 2020
Duration: 2 50-minute sessions

Lesson Summary

Pre-lesson preparation

The second session is a programming activity and requires using NetLogo (a tutorial is in the Unit 4 Lesson on Hypothesis Testing with NetLogo) or using Python with Bokeh (for more advanced students).

Summary

Students will be introduced to the topic of diversity and computational problems, learn about and discuss implications of the proliferation of algorithms in society, and reflect on how the quality of input data and small, sometimes unintended biases can lead to low quality solutions.

Outcomes

- Students will consider how diversity leads to better solutions.
- Students will discuss the proliferation of algorithms in society.
- Students will learn about some applications of artificial intelligence and machine learning algorithms.
- Students will explore a computational simulation of bias.
- Students will understand how small changes in input parameters can affect output quality and variety.
- Students will program an agent-based model from social science theory.

Overview

Session 1

1. Getting Started (5 min)
    - Artificial intelligence algorithms in daily life?
2. Discussion (10 min)
    - Algorithms, Input Data, and Diversity
3. Guided Activty (25 min)
    - Parable of the Polygons
4. Wrap-up (10 min)

Session 2

1. Getting Started (5 min)
    - Algorithms, Diversity, and Solutions
2. Programming Activity (40 min)
    - Make your own Parable of the Polygons
3. Wrap-up (5 min)

Learning Objectives

CSP Objectives

- *EU CRD-1 - Incorporating multiple perspectives through collaboration improves computing innovations as they are developed.*
  - LO CRD-1.A - Explain how computing innovations are improved through collaboration.
- *EU CRD-2 - Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking.*
  - LO CRD-2.B - Explain how a program or code segment functions.
  - LO CRD-2.E - Develop a program using a development process.
- *EU DAT-2 - Programs can be used to process data, which allows users to discover information and create new knowledge.*
  - LO DAT-2.C: - Identify the challenges associated with processing data.
- *EU IOC-1 - While computing innovations are typically designed to achieve a specific purpose, they may have unintended consequences.*
  - LO IOC-1.A - Explain how an effect of a computing innovation can be both beneficial and harmful.
  - LO IOC-1.B - Explain how a computing innovation can have an impact beyond its intended purpose.
  - LO IOC-1.D - Explain how bias exists in computing innovations.

Math Common Core Practice:

- MP3: Construct viable arguments and critique the reasoning of others.

Common Core ELA:

- RST 12.2 - Determine central ideas and conclusions in the text
- RST 12.6 - Analyze the author's purpose in providing an explanation, describing a procedure
- RST 12.9 - Synthesize information from a range of sources

NGSS Practices:

- 2. Developing and using models
- 7. Engaging in argument from evidence

NGSS Content:

- HS-ETS1-1. Analyze a major global challenge to specify qualitative and quantitative criteria and constraints for solutions that account for societal needs and wants.
- HS-ETS1-4. Use a computer simulation to model the impact of proposed solutions to a complex real-world problem with numerous criteria and constraints on interactions within and between systems relevant to the problem.

Teacher Resources

Student computer usage for this lesson is: **required**

Supplementary articles:

- "How Diversity Makes Us Smarter"
  - http://www.scientificamerican.com/article/how-diversity-makes-us-smarter/ (http://www.scientificamerican.com/article/how-diversity-makes-us-smarter/)

Lesson Plan

Session 1

Getting Started (5 min)

Think-Pair-Share: Artificial intelligence algorithms in daily life?

Ask your students to list discuss what artificial intelligence algorithms they think affect them in daily life and how they are affected. How would it be different if those algorithms did not recognize them or treated them inappropriately? If an algorithm makes the wrong judgement about them, what would they do, who could they go to for help? It is OK if students are unsure about the answers as it leads into the following discussion.

Discussion: Algorithms, Input Data, and Diversity (10 min)

Algorithms have become prominent and common in daily life. In particular, algorithms that use **artificial intelligence (AI)** or a technique called **machine learning** are employed to make decisions that affect us, sometimes in ways we aren't aware of and cannot control. For example, machine learning is used by Google, Facebook, Flickr, among other companies to detect faces in photographs, apply labels, or automatically tag images with people they recognize. In general, these algorithms are *trained* on input data (such as images of people), and are improved iteratively using statistical methods until they learn how to categorize, differentiate, and reach a correct answer. Another example of companies using AI is in personalized marketing or recommendation systems. For instance, Amazon tailors the products they suggest based on an algorithm trained on similar customers' purchase history. Or a company like Netflix will recommend movies to users who rate similar movies similarly.

At the heart of these algorithms is the fact that their solutions all depend on the initial parameters and input data used to train them. What happens when those parameters and data are not sufficiently diverse? Small changes in those parameters, oversights in the training data, implementer bias (both intended and not), and missing distributions or categories can all negatively prejudice algorithms. All this is important to keep in mind as algorithms start determining aspects of our lives and there is less or no control over the input and decisions they make about and for us. Computing innovations can reflect existing human biases because of biases written into the algorithms or biases in the data used by the innovation. In fact, biases can be embedded at all levels of software development. Thus, all involved in program development - including programmers - should take action to reduce bias in algorithms used for computing innovations as a way of combating existing human biases.

Effective collaboration produces computing innovations that reflect the diversity of talents and perspectives of those who design them. Widespread access to information and public data facilitates the identification of problems, development of solutions, and dissemination of results. A 2016 article in the New York Times (http://www.nytimes.com/2016/06/26/opinion/sunday/artificial-intelligences-white-guy-problem.html?_r=0 (http://www.nytimes.com/2016/06/26/opinion/sunday/artificial-intelligences-white-guy-problem.html?_r=0)) highlights an example problem where lack of diversity among devlopers lead to a lower quality solution. When teams developing various AI and machine learning algorithms lacked diversity they failed to recognize errors from calibrating their software using data based on too limited a population. This lead to significantly bad results, such as not recognizing people with dark skin tones or tagging them inappropriately.

Examples where diversity may greatly influence the quality and real-world implications of algorithms include:

- Software used by police for assessing risk has an unwarranted racial bias (https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing (https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing))
- Targeting and financial exploitation of more vulnerable communities (https://bigdata.fairness.io/data-brokers/ (https://bigdata.fairness.io/data-brokers/))
- Hiring algorithms and software like E-verify, a program to determine legal eligibility for the workforce (https://bigdata.fairness.io/e-verify/ (https://bigdata.fairness.io/e-verify/))
- Online searching and results that are returned (https://bigdata.fairness.io/search-bias/ (https://bigdata.fairness.io/search-bias/))
- Credit scoring algorithms (https://www.thenation.com/article/how-companies-turn-your-facebook-activity-credit-score/ (https://www.thenation.com/article/how-companies-turn-your-facebook-activity-credit-score/))
- Predictive policing and surveillance, such as the NSA's alleged leaked program XKEYSCORE, tracks metrics and metadata about internet users, potentially then fed into various other surveillance apparatuses (https://theintercept.com/2015/07/01/nsas-google-worlds-private-communications/ (https://theintercept.com/2015/07/01/nsas-google-worlds-private-communications/))
- In 2014, the White House publicly recognized that "Big Data" algorithms may foster discrimination and that the federal government would need to begin being more watchful of such practices (https://www.whitehouse.gov/sites/default/files/docs/big_data_privacy_report_may_1_2014.pdf (https://www.whitehouse.gov/sites/default/files/docs/big_data_privacy_report_may_1_2014.pdf))
- In 2016, the FTC published a report specifically on the potential of such algorithms to foster both inclusivity and exclusivity (https://www.ftc.gov/system/files/documents/reports/big-data-tool-inclusion-or-exclusion-understanding-issues/160106big-data-rpt.pdf (https://www.ftc.gov/system/files/documents/reports/big-data-tool-inclusion-or-exclusion-understanding-issues/160106big-data-rpt.pdf))

Guided Activity (25 min)

Parable of the Polygons

A hands-on example of how initial parameters, subtle biases, and lack of diversity can greatly affect an algorithm's execution is in the "Parable of the Polygons" (http://ncase.me/polygons/ (http://ncase.me/polygons/)). The Parable of the Polygon is a simulation based on the mathematical model of how segregated communities may arise from small differences preferences, based on the theory of Thomas Schelling, an economist and game theorist who received the Nobel Prize for Economics in 2005.

Guide students through the website and let them explore dragging polygons, changing settings, and see how they can influence (it is best for them to follow along individually or in groups on their own computer).

An alternative version of the final sandbox with three polygon shapes is also available here: http://ncase.me/polygons-pentagons/play/automatic/automatic_sandbox_frame.html (http://ncase.me/polygons-pentagons/play/automatic/automatic_sandbox_frame.html)

Wrap-up (10 min)

Journal or Homework

Have your students continue to explore the Parable of the Polygons website and answer the following questions using the sandbox at the bottom of the website:

- What settings where the ratio of polygon populations are the same lead to the greatest segregation? Why do you think that is?
- What settings where the ratio of polygon populations are the same lead to the least segregation? Why is that?
- What settings where polygons still have room to move lead to the algorithm being unable to find a solution (it does not terminate)? Why is that?
- Answer the same three questions above using the three-polygon version here: http://ncase.me/polygons-pentagons/play/automatic/automatic_sandbox_frame.html (http://ncase.me/polygons-pentagons/play/automatic/automatic_sandbox_frame.html)

Session 2

Getting Started (5 Min)

Think-Pair-Share: Algorithms, Diversity, and Solutions [5 min]

- Have students consider one or more of the following questions:
    - How can an algorithm be affected by diversity (such as in its input data)?
    - What does it mean for input or training data to be diverse?
    - Can you think of examples where diversity, or the lack of diversity, affects the quality of an algorithm's solution?
    - Can you think of algorithms that benefit from diverse data?

Programming Activity (40 Min)

Two options: guided programming exploration with NetLogo, or more involved Python programming project for advanced students

Option 1: Using NetLogo

- If you have not already used NetLogo in the classroom, you can follow the instructions for downloading and setting up NetLogo in the Unit 4 Lesson "Hypothesis Testing with Simulations in NetLogo"
- Open up NetLogo and go to File->Models Library, then under the Sample Models folder, open the Social Science folder and load the Segregation model
- For information about the model, read the sections in the Info tab
- Have the students experiment with the settings and see what differences and similarities it has to the Parable of the Polygons website
- For the remainder of the activity, have the students make some meaningful changes to the code of the simulation (in the Code tab) and write a short response stating what they did and how their parameters
- Some suggestions include:
    - Add more types of agents to the model (hint, add another color to the line "set color one-of [red green]")
    - Change the sliders to match the Parable of the Polygons slider (agent count based on ratios, ability to have <50% agents)
- Finally, a more developed model is included in the NetLogo models library

Option 2: Using Python with Bokeh **(for advanced students)**

Have students re-create the Parable of the Polygons themselves in Python, time with at least four types of polygon shape/colors. The source code for the Parable website is hosted on GitHub here: https://github.com/ncase/polygons/blob/gh-pages/play/automatic/automatic.js (https://github.com/ncase/polygons/blob/gh-pages/play/automatic/automatic.js) It is in

JavaScript, but it serves as a good pseudocode basis. Alternatively, students could use the NetLogo code from Option 1 as another reference in creating their simulation.

Wrap-up (5 Min)

Journal: Have your students reflect and list 5 examples or ways in which diverse input makes for better solutions.

(http://www.umbc.edu/) (http://www.umd.edu/) (http://www.nsf.gov/)

Authored by: CS Matters in Maryland
Website: csmatters.org (http://csmatters.org)
Email: csmattersinmaryland@gmail.com (mailto:csmattersinmaryland@gmail.com)