FORUM

MobiliT.AI

What last year taught us: the magical seven plus minus two

Giuliano (Giulio) Antoniol – antoniol@ieee.org

One year ago SEMLA 2018

- Bridge the gap between software engineers and machine learning experts
  - Architecture and software design
  - Model/data verification and validation
  - Change management
  - User experience evaluation and adjustment
  - Privacy, safety, and security issues
  - Ethical concerns

AI Has Become a Clique

Cliques are commonly formed among researchers.

- They read, write, and meet only within the clique. [illegible]
- They are often unaware of results, tools, and techniques from outside.
- A huge amount of duplication results.
- Bad solutions are used where good ones exist.
- Vocabulary differences make communication impossible.

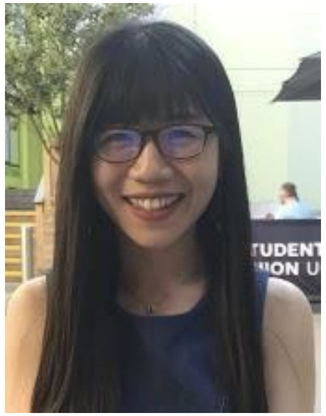
CS has become a clique. We are isolated from computer users.

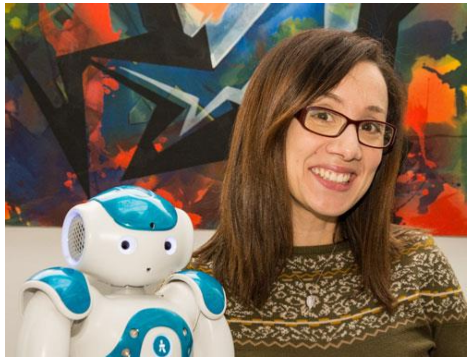
CS is being divided into cliques. We don't talk across disciplines enough.

AI has become a clique on its own.

- Example: Vision researcher who was unaware of signal processing progress.
- Example: Neural net not compared with least squares method.
- Example: Researchers not familiar with algorithm and software design results.

May 23-24, 2019 – 2d SEMLA Event: semla.polymtl.ca





Hands-on Session: Metamorphic Testing of Deep Neural Networks

ML/AI - SEMLA

- Eliza (J Weizenbaum 1966) demonstrates we can be easily fooled believing an intelligent behavior even if it is just pattern matching and pattern substitutions

- Fast forward to early 80’s first attempts to integrate pattern recognition, machine learning, vision, spoken and natural language processing into “intelligent” platforms

- The dream is still valid create systems that learn

Deep learning — SEMLA

- Countless possibilities but:
  - How do we cope with robustness ?
  - How do we deploy in mission critical systems ?
  - How do explain model decision ?
  - How do we adapt current regulations ?

Why Worry

Self driving car crash

ML/AI should it help us to:

- Imitate human behavior ?

- Play game well ?

- Build programs that use the same methods that human use?

Caveat: ML/AI a panacea?

- Not all task are well suited for ML
- We can often solve the same or similar problem with traditional coding
- If we have physical laws and mathematical models why should we learn from data ?
- Find the right problem for the right tool is “a huge challenge”
  - 2011 IBM started its AI initiative for health: no result so far

ML/AI for mobility

- We are somehow used to human errors
- A program (or human !) failure may have catastrophic effects
- The user should be aware of what is under the hood and the associated risks or at least be warned
  - 737 MAX training and manual, was it sufficient?

Testing course

- White box and black box
  - Boundary value analysis
    - MCAS limit was 2.5 not 0.6! It was classified as major failure no death risk
- MC/DC aka RCC coverage criterion
- Testing process and documentation
- The testing team is not the developer team

Trusting software

- Software runs the world we need to build more and more applications BUT we need to trust software: we depend on it
- Quality assurance and testing need complete, precise, non ambiguous, non vague specifications
- If specifications are not complete or non ambiguous how can we define an the expected result?

Non testable programs

- Pseudo-oracles:
  - If we cannot hope to have a full, non vague, precise specification
  - If we cannot reasonably check the output
  - If we do not have the “answer”

ML/AI Testing Contradiction

- If we write a program to compute an answer it implies we have not such an answer

- If we do not know what the answer is, how can we write an oracle and test the program?

- If we have an ML/AI component it implies we do not know the answer

E. J. Weyuker, "On testing non-testable programs," The Computer Journal, vol. 25, no. 4, pp. 465–470, 1982

ML/AI QA a new problem?

- Not at all !
  - The Pseudo-oracle problem was there long before ML and AI
- Untestable programs are just more common
- ML/AI are data intensive: what matter the most are data
- Without the data it may be hard or impossible to interpret, explain, introspect or validate results

No Oracle – Pseudo-Oracle

- We cannot hope to have the oracle
- Even If we do not know the answer it may not be so catastrophic
  - Get rid of the idea of absolute oracle use a differential oracle
- Apply the concept of N-version programming
  - If two or more systems are trained on the same data they must give the same answer, right ?

Late 90$_s$ - metamorphic testing

- If we use supervised ML the pseudo-oracle problem can be lessened

- If we have labeled data it imply we know the answer for a subset of the data

- Why do not leveraging such knowledge ?

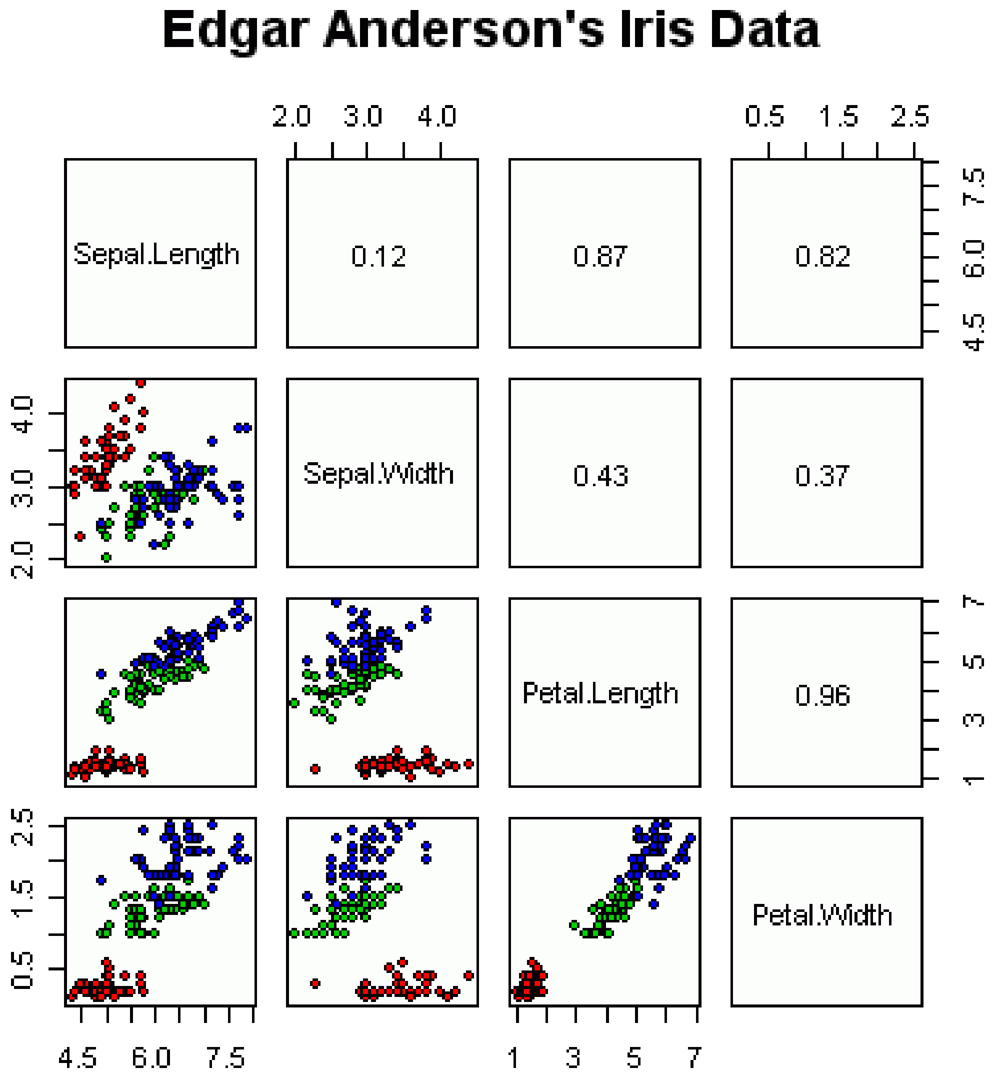
Shifting the focus

- We no longer need the oracle
- We need the metamorphic relations
- It may not ensure “corner” cases aka catastrophic events will never happen
  - Search based software testing: search guided by a cost function risky inputs

One example: DEEPTEST

- Clever use of a set of “reasonable” image transformation:
  - add rain, fog, lens distortion, blur
- Greedy combination of transformation to increase neurons coverage
- Enforce metamorphic relations
  - “recycle” the labels but change the data
    - rain or snow the road stretch is the same output should be the same but different people drive differently thus impose output are just very close (!)

Beyond models: Software 2.0



Software 2.0

- Simply learn the desired behavior
- There are domains where we have plenty of labeled data (a switch or light controllers, car engines, …)
- If you have understanding of the problem and physical laws but the coding task is difficult while data are abundant software 2.0 can be the answer
- Will traditional software disappear?

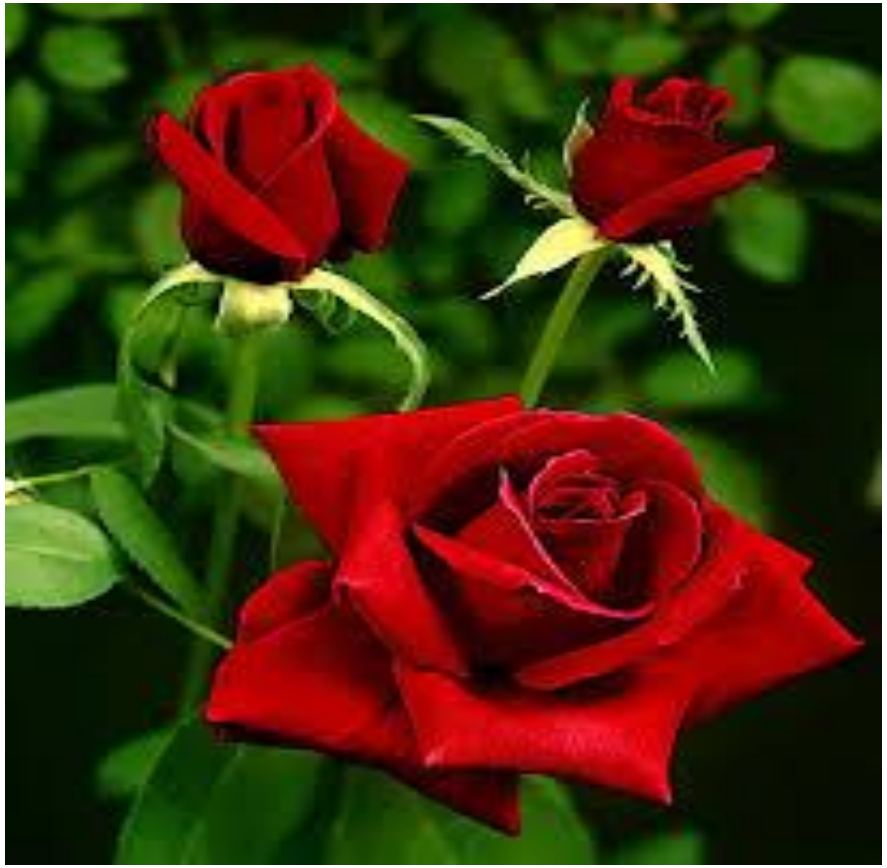
Anchoring effect -- Daniel Kahneman

- Base current judgment on previously heard numbers
- The price of a house: people tend to settle for higher house prices if the starting number is larger
  - overshooting
- It worked before, so it should work again
  - Arianne accident

ML/AI Components

- **ML/AI Code is not really relevant for QA:**
  - **They are data intensive**
- A ML/AI component will be integrated into an environment
- Training data must reflect the deployment environment – all possible environments
  - If training data do not represent context X we cannot expect the “right” behavior

How many roses?



Miller G.A. (1956) The magical number seven, plus or minus two: Some limits on our capacity for processing information . Psychological Review. 63 (2): 81–97

How many timbers?

Daniel Kahneman: The law of small numbers -- Brains are bad at dealing with large numbers

Conclusion

- Although the horizon is changing at a faster pace the problem was known long ago
- We have initial and promising testing theories tools
  - more efficient and cost effective approaches/tools are needed
- We lack explainability, introspection and scalable exploratory data analysis
  - Why did the ML/AI component take that decision ?
- There is a urgent need to address data: quality, management, process, certification
- Be aware of risks — make the user aware of risks

META - Conclusion

… Geometrica ideo demonstramus, quia facimus, physica si demonstrare possemus, faceremus… G. Vico 1708. Lib. Methaph. Chap III

… Wir müssen wissen — wir werden wissen! … Hilbert 1930

They were wrong: the system cannot demonstrate its own consistency … Goedel 1931

Please read Parnas paper:

The Real Risks of Artificial Intelligence: Communication of ACM, Oct 2017, Vol 60 No 10