Issues Concerning AI Transparency

Tom Dietterich
Why do we need transparency?

• Trust
• Effective human-machine interaction
• Effective machine-to-machine collaboration
• Software engineering
Goal: Trustable AI Systems

• Trustable AI in high-stakes applications
  – Self-driving cars
  – Autonomic power grid
  – Medical devices and surgical robots
  – Cyber defense
  – Weapons systems
  – etc.

• Trustable human augmentation
  – Trustable information source
  – Trustable personal assistant
  – Trustable augmented reality

• Trust that the system has not been compromised by cyberattack
Interaction and Collaboration

• In order for a human to interact successfully with an AI system...
  – human needs to understand
    • what the AI system knows and does not know
    • what the system can and cannot do
    • situations under which the system can and cannot be trusted
    • good predictive model of how the system will behave; when it will engage in clarifying dialogues

• In order for multiple AI systems to collaborate they need
  – models of each other: knowledge, capabilities, preferences, costs
Related Needs

• Software Engineering Tools
  – Training: building the software using machine learning
  – Testing: finding failure pathways
  – Debugging: fixing errors
Explanation Paradox

• In the 1980s, Expert Systems vendors found that the market demanded that there be an “Explanation Facility”
  – But no one ever used these in practice
• People are surprisingly willing to trust a system after only a small amount of experience with it
  – Over-estimate how general the system’s knowledge is?
  – Assume that the system is based on broad fundamental principles rather than thousands of memorized cases?
Fundamental Issues

• Issue 1: Some AI knowledge or behaviors may not have short descriptions
  – Deep learning may produce a hierarchical cross product of local generalizations
  – Theorems may have no short proofs
    • many different information sources
    • many interlocking steps
    • Example: latest results on Pythagorean coloring and the Busy Beaver problem, Erdös discrepancy conjecture
• Issue 2: How do we ensure that explanations are faithful to the actual mechanism of the system?
  – What kinds of intervention or feedback are supported by the explanation?
    • Inserting a breakpoint at an inference step?
    • Manipulating a component (e.g., feature or parameter value) and observing its consequences?
    • Feedback on the explanation results in changes in the system behavior?
Example: End-User Feedback to Learning Systems

- Can the user edit the explanation and cause the “right” changes in the classifier?

User edits the explanation (e.g., adds/deletes features)
• Issue 3: How do we validate “generality”?
  – Test that similar inputs always exercise similar components and produce similar outputs
    • Validate this formally?
    • Test experimentally? What is the smallest change that can produce an output difference greater than $\Delta$?
  – Enforce “smoothness” of the input-output mapping
    • detect and remove fractal behavior?
    • ensure that the influence of every subcomponent is limited?
  – Ensure that there are no hidden back doors
• Issue 4: How can we easily test complex situations?
  – Need ways of generating complex scenarios
  – Need to ensure that there are no “tells” that are exploited by the AI system
    • “Tank in the trees” problem
    • UTF-16 ^@p^@r^@o^@b^@l^@e^@m
  – Need to be able to manage augmented reality tests
    • Real queries to the web?
    • Real sensor data with overlaid augmentations?
    • Example: simulated weapons in carry-on bags
  – Simulated Human-in-the-loop?
• Issue 5: How do we evaluate explanation quality?
  
  – What metrics?
    • User satisfaction (??)
    • User takes appropriate actions based on the explanations
    • User develops appropriate trust (knows when and when not to trust)
    • User can predict future behaviors
      – actions the system will take
      – when the system will ask for help
  
  – Human user studies are expensive
  – Cheaper proxies?
• Issue 6: Transparency for Multiple AI Collaboration
  – Metrics for successful collaboration?
  – Accuracy of each system’s model of the other systems?
    • probability that delegation succeeds?
    • probability that agent A proactively takes a step that will help agent B? Including warning B of potential trouble?
• Issue 7: Transparency for Software Engineering
  – Adversarial testing: Engineer defines bad outcomes and then applies AI search methods to find high probability paths to those outcomes
  – Debugging
    • Can the engineer easily find the cause of failures discovered through testing?
    • Can the cause be easily fixed?
    • Without introducing new failures?