Panel 1: AI Assurance: Small and Large

Tom Dietterich
Collaborative Robotics and Intelligent Systems (CoRIS)
Oregon State University
Assurance for Machine Learning

• Assurance by Construction
• Assurance by Run-time Monitoring
Assurance by Construction

• Robust training
  • Adversarial training can improve robustness
    • (Goodfellow, et al., 2015; Madry, et al., 2018)

• Robust query processing
  • Post-processing by stability testing can guarantee robustness
    • (Li, Chen, Wang & Carin, 2019, arXiv 1809.03113)
    • Requires stationarity assumption
Run-Time Assurance

- Rejection
  - Reject queries for which the ML system has low confidence
    - Requires fitting a confidence function or rejection function
    - Calibrated probabilities (Nicolescu-Mizil & Caruana, 2005)
    - Rejection functions (Cortes, DeSalvo & Mohri, 2018)
  - Requires stationarity assumption
Data Shift Detection

• Data Shift:
  • Changes in class probabilities (e.g., increase in cyberattacks)
  • Changes in input distribution (e.g., network traffic shifts)
  • Changes in the decision boundary (e.g., attackers try to hide)
  • New classes to predict (e.g., new kind of cyberattack)

• Methods:
  • For single queries: Anomaly detection (Liu, Garrepalli, et al. ICML 2018)
    • Provides guarantees
High Reliability Organizations
Todd LaPorte, Gene Rochlin, and Karlene Roberts

• Preoccupation with failure
  • Fundamental belief that the system has unobserved failure modes
  • Treat anomalies and near misses as symptoms of a problem with the system

• Reluctance to simplify interpretations
  • Comprehensively understand the situation

• Sensitivity to operations
  • Maintain continuous situational awareness

• Commitment to resilience
  • Develop the capability to detect, contain, and recover from errors. Practice improvisational problem solving

• Deference to expertise
  • During a crisis, authority migrates to the person who can solve the problem, regardless of their rank
Designing AI Systems to be HROs

• Maintain Situational Awareness
  • AI methods are very good at integrating data from multiple sensors and effectors to estimate a probability distribution over states

• Detect Anomalies and Near Misses
  • Anomalies: Yes
  • Near Misses: Research needed

• Generate Candidate Explanations for Anomalies & Near Misses
  • Very little work: Research needed

• Improvise Solutions
  • Improvisational problem solving that extends or operates outside the system model
## Assessment: Designing AI as an HRO

<table>
<thead>
<tr>
<th></th>
<th>Assessment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Situational Awareness</td>
<td>A  mature methods</td>
</tr>
<tr>
<td>Detect Anomalies and Near Misses</td>
<td>B  high-dimension, dynamics</td>
</tr>
<tr>
<td>Explain Anomalies and Near Misses</td>
<td>D  only basic techniques</td>
</tr>
<tr>
<td>Improvise Solutions</td>
<td>F</td>
</tr>
</tbody>
</table>
Designing a Human + AI Team as an HRO

• Even very powerful AI systems will be surrounded by a human team

• Situational Awareness
  • AI can track the situation, but humans and AI must establish a shared mental model of the situation: Research needed
  • Humans must be aware of what version of the AI system they are using. When was it last updated/retrained? Research needed

• Detect Anomalies and Near Misses
  • AI system must understand and predict behavior of human team
  • AI and Humans must work together: interactive anomaly detection

• Generate Candidate Explanations for Anomalies & Near Misses
  • Very little work: Research needed

• Improvise Solutions
  • AI should support human improvisational problem solving: Research Needed
  • Example: mixed-initiative planning
## Assessment: Human + AI HROs

<table>
<thead>
<tr>
<th></th>
<th>Assessment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Situational Awareness</td>
<td>C  poor UI, poor communication</td>
</tr>
<tr>
<td>Detect Anomalies and Near Misses</td>
<td>C  user feedback to anomaly detection</td>
</tr>
<tr>
<td>Explain Anomalies and Near Misses</td>
<td>D  only basic techniques</td>
</tr>
<tr>
<td>Improvise Solutions</td>
<td>D  mixed-initiative planning</td>
</tr>
</tbody>
</table>
Backup Material
Assurance by Construction

• Let \( f(x; \theta) \) be a predictive model parameterized by \( \theta \)

• Training data \( \{(x \downarrow 1, y \downarrow 1), \ldots, (x \downarrow N, y \downarrow N)\} \)

• Standard training

\[
\theta^{\uparrow*} := \arg\min_{\theta} \sum_{i=1}^{N} L(f(x \downarrow i; \theta), y \downarrow i)
\]

where \( L(y, y) \) is the loss function for predicting \( y \) when the true answer was \( y \)

• Robust (adversarial) training

\[
\theta^{\uparrow*} := \arg\min_{\theta} \max_{\delta \downarrow i \in \Delta} \sum_{i=1}^{N} L(f(x \downarrow i + \delta \downarrow i; \theta), y \downarrow i)
\]

where \( \Delta \) is a set of allowed perturbations (Goodfellow, et al., 2015; Madry, et al., 2018)

Equivalent, in some cases, to regularization methods
Assurance by Post Processing

• Given a trained $f$, post-process it to guarantee robustness

• Example: Stability Testing
  • Given query $x\downarrow q$, sample perturbations and predict $y$ using majority vote
  • $f(x\downarrow q;\theta) = \text{orange}$
  • but the majority of perturbed points have $f( x\downarrow q + \delta) = \text{blue}$
  • so $y := \text{blue}$

• First method to give a guarantee on ImageNet (1000 classes)
  • Li, Chen, Wang & Carin, 2019, arXiv 1809.03113
Assurance by Rejection

• Construct a rejection function $g$
• Example: $g$ produces a calibrated probability. If the maximum probability is too small, then reject
• This is a type of competence model

Classifier $f$

$p = \max_{j} g(p \downarrow ij)$

$p < \tau$

Argmax $\max_{j} g(p \downarrow ij)$
Assurance by Runtime Monitoring

• Construction-time guarantees assume test queries come from the same distribution as training queries

• This assumption rarely holds in practice
  • Changes in class probabilities (e.g., increase in cyberattacks)
  • Changes in input distribution (e.g., network traffic shifts)
  • Changes in the decision boundary (e.g., attackers try to hide)
  • New classes to predict (e.g., new kind of cyberattack)

• Data shift detection
  • Compare recent queries \{x_{\uparrow q1}, x_{\uparrow q2}, ..., x_{\uparrow qm}\} to training points \{x_{\uparrow 1}, ..., x_{\uparrow N}\}
  • Use two-sample tests:
    • typical sets, kernel maximum mean discrepancy, old-vs-new classifier

• Anomaly detection
  • \(A(x_{\downarrow q}) := -\log P(x_{\downarrow q})\), where \(P\) is the distribution of training points
  • Operates on single points => generates many false alarms
Open Category Guarantee

- Assume we know (a bound on) the proportion $\alpha$ of test queries that correspond to new classes “aliens”
- Then we can estimate a threshold $\tau$ that with high probability will detect $1-\epsilon$ of the aliens on new test queries
- Liu, Garrepalli, et al. ICML 2018

\[
P\downarrow m = (1-\alpha)P\downarrow 0 + \alpha P\downarrow \alpha
\]