Robotic Learning of Ownership Relations and Norms

A Senior Project Presentation by Xuan
Supervised by Jake Brawer and Brian Scassellati
“Can you pass me my blocks?”
“Stop that, that’s Jake’s.”
“By the way, this is Xuan’s now.”
“You can move that, but you can’t throw it away.”
“You can use this, but you have to put it back.”
Ownership

• “The act, state, or right of possessing something.”
• Practical applications for social robotics
• Extends across social, ethical, and legal spheres
Social & Moral Norms

- “Don’t pick things up if they’re dirty.”
- “Be respectful to sacred objects.”
- “Use that knife to stab food, not people.”
Machine Ethics / AI Safety

- AI = Optimizing Procedure
- ASI = Very Powerful Optimizing Procedure
- No understanding of human values and norms
  → Humanity optimized away
What is ‘ownership’ anyway?

“Determining ownership [...] involves determining who has certain rights and duties over property. These rights and duties [...] can be separated and held by different parties.” (Wikipedia)
What is ‘ownership’ anyway?

• Two components:
  • Who owns what? (Owner identification)
  • Who has the right to do what? (Norms and permissions)
Norm Learning

• Given knowledge about who owns what, can we figure out the norms that apply?
• Machine learning, but not using NNs, or SVMs
Norm Representation

• A return to predicate logic!
• Deontic logic: (forbid/allow) [action] if [conditions]
• Natural way to represent human norms
Norm Learning

• Rule induction – popular in the 90s(?)
• Separate-and-conquer rule learning
• Decision tree learning is similar
Norm Learning

• Basic idea:
  • Find most general rule that covers positive examples without covering negative ones
  • Keep finding rules until all positive examples are covered
  • Prune rules if necessary
Norm Learning

• Examples = Permissions (Object-Specific)
  • forbid pickUp(wallet)
  • “Picking up this wallet is forbidden.”

• Rules = Norms (General, Predicate-Based)
  • forbid pickUp(X) if ownedBy(X, Jake)
  • “Picking up Jake’s stuff is forbidden.”
Norm Learning

• Modifications to standard rule induction:
  • Incremental learning (since it happens online)
  • Able to receive both permissions and norms as instructions
  • Probabilistic (due to uncertainty in ownership)
Incremental Learning

• If positive example arrives:
  • Check if covered by rules
  • Else find most general rule that covers positive example without covering negative examples

• If negative example arrives:
  • Check if not covered by rules
  • Else find most general rule that covers negative example without covering positive examples
  • Logically subtract this rule from any covering rules
Dual-Mode Instruction

• If positive rule arrives:
  • e.g. “Don’t pick up Jake’s stuff”
  • Check if it’s a specialization of any existing rule
  • Specialize the rule so that false positives are minimized
  • e.g. “Don’t pick up Jake’s stuff if it’s valuable.”

• If negative rule arrives:
  • e.g. “You’re allowed to throw dirty stuff away.”
  • Specialize the rule so that false negatives are minimized
  • e.g. “You’re allowed to throw dirty and worthless stuff away.”
  • Logically subtract this rule from any covering rules
Probabilistic Learning

• Rules might only partially cover an example
  • If 75% sure that this wallet is owned by Xuan
  • Then the rule “forbid pickUp if ownedBy Xuan” evaluates to 75% true on the wallet
  • Rule covers 75% of the permission “forbid pickUp on wallet”

• Rules are still deterministic, but the predicate values are probabilities!
Results
Owner Identification

• Who owns what?
• Two routes:
  • Rule-based ownership inference
  • Percept-based ownership extrapolation
Rule-Based Inference

• Given knowledge about what rules apply, can we infer who owns what?
• Inference, but probabilistic!
Rule-Based Inference

• Basic idea:
  • Robot knows that trashing owned objects is forbidden
  • Robot tries to trash an object
  • Human instructs it that trashing that object is forbidden
  • Robot infers that object must be owned

• Assumptions:
  • No other rules forbid trashing
  • Human is not lying / making a mistake
Rule-Based Inference

• In general, we can use Bayesian inference
• \( P(\text{ownedBy Jake}| \text{pickUp forbidden}) = \)

\[
P(\text{pickUp forbidden} | \text{ownedBy Jake}) \frac{P(\text{ownedBy Jake})}{P(\text{pickUp forbidden})}
\]
Rule-Based Inference

• $P(\text{ownedBy Jake})$ is the prior probability of ownership
• $P(\text{pickUp forbidden})$ computed by evaluating all applicable rules for the object and action
• $P(\text{pickUp forbidden} \mid \text{ownedBy Jake})$ computed by evaluating the rules, supposing that $P(\text{ownedBy Jake}) = 1.0$
Rule-Based Inference

• If rules have nothing to do with ownership
  • e.g. forbid pickUp if isColored red
  • Ownership probabilities don’t change
• If only one rule to do with specific owner
  • e.g. forbid pickUp if ownedBy Jake
  • \( P(ownedBy\ Jake | pickUp\ forbidden) \) becomes 1.0
Rule-Based Inference

- Can also account for uncertainty in instructions
  - e.g. lying / bad speech recognition
- Permission has some probability of being ‘forbid’, some probability of being ‘allow’
  - $P(\text{forbidden} \mid perm) + P(\text{allowed} \mid perm) = 1$
- $P(\text{ownedBy } A \mid perm) = P(\text{ownedBy } A \mid \text{forbidden}) P(\text{forbidden} \mid perm) + P(\text{ownedBy } A \mid \text{allowed}) P(\text{allowed} \mid perm)$
Results
Rule-Based Inference

• But how do we get the initial prior probabilities?
  • $P(ownedBy\ Jake)$, $P(ownedBy\ Xuan)$, etc.
• Especially if no one makes any ownership claims?
Percept-Based Extrapolation

- Given the owners and physical properties of some objects, can I guess the owners of other objects?
  - e.g. Most of the stuff is on this desk is Xuan’s, maybe the new object on this desk is also Xuan’s
- Percepts: Position, color, etc.
- Classification problem (but probabilistic!)
  - Features $x$ of some object $\rightarrow P(\text{ownedBy} A \mid x)$
Percept-Based Extrapolation

- Kernel Logistic Regression (KLR)
  - Outputs probabilities
  - Uses (Gaussian) similarity to previous examples as ‘features’
- One KLR classifier for each potential owner
- Can be done using Scikit Learn
  - But have to mess around with kernel approximators
Percept-Based Extrapolation

- Problem: Inputs are also probabilities!
- Turns out logistic regression handles that nicely
Percept-Based Extrapolation

• Loss function (one sample):
  \[ L(x_i, y_i) = -y_i \log P(y_i = 1 | x_i) - (1 - y_i) \log P(y_i = 0 | x_i) \]

• Cross-entropy between \( p \) and \( q \):
  \[ H(p, q) = -E_p[-\log q(i)] = -\sum_i p(i) \log q(i) \]
Percept-Based Extrapolation

• Equivalent to:
  • Running standard logistic regression but...
  • Duplicating each example \((x_i, p(1)) \rightarrow (x_i^+, 1), (x_i^-, 0)\)
  • Weight \(x_i^+\) with the probability \(p(1) = P(ownedBy A)\)
  • Weight \(x_i^-\) with the probability \(p(0) = 1 - P(ownedBy A)\)
Percept-Based Extrapolation

• When do we run this?
  • Whenever a new object is detected
  • Whenever an ownership claim is made
  • Whenever an ownership inference is made
• Avoid updating the probabilities of objects that are already claimed!
Some other cases

• Prediction for each owner is made separately
• What happens when a new person is introduced?
  • Assume they don’t own any of the objects?
  • Currently we just use some prior probability (0.5, 0.1, etc.)
• Should we entirely trust ownership claims?
  • Currently there is a constant ‘trust’ parameter
Results
Does it all work together?

• I’ll find out soon enough!
• Mutual dependencies
  • Need ownership probabilities to induct rules
  • Need rules to infer ownership probabilities
• At least *some* direct instruction is needed
  • Ownership claims
  • Rule instruction
Does it work on the robot?

- Probably!
- Haven’t tested in a while.
- But no reason why it shouldn’t
  - As long as we can uniquely identify objects (ArUco)
  - Speech recognition, NLP, agent detection, is abstracted away
  - Text input with specific syntax instead
Limitations & future work

• Rules are currently about what *Baxter* can(not) do
  • Not about agents in general
  • Not as ‘social’ as we might like

• Cannot infer from other social observations
  • If Baxter knew that “Only owners of an object can pick it up.”
  • Could infer that if Jake picks something up, it’s his.
  • Or he’s stealing.

• Cannot account for
  • Obligations like “Return borrowed tools.”
  • Temporary ownership (e.g. borrowing)
If I were to do this again...

- Use MySQL-like database to store object information
  - Sending stuff between nodes gets really messy
- Use Prolog or Pyke for logic programming
  - Current representation is not very flexible / general
  - But not sure how to use probabilistic logic with Prolog etc.
Acknowledgements

• Jake Brawer
  • Especially for discussions on the nature of ownership
• Ikenna Nzewi and Guillame Bachelard-Bakal
  • CPSC 473 groupmates
• Prof. Scaz
  • For being skeptical
• ScazLab in general!
“Thanks!”