Reasoning about Hypothetical Agent Behaviours and their Parameters

Stefano Albrecht and Peter Stone
Introduction
Motivation: Ad Hoc Teamwork

Design individual agent which can collaborate effectively with other agents, without pre-coordination

- Flexibility – ability to collaborate with different teammates
- Efficiency – find effective policy quickly
- AAAI 2010 Challenge Paper (Stone et al.)
Motivation: Ad Hoc Teamwork

Design individual agent which can collaborate effectively with other agents, without pre-coordination

Multiagent Interaction without Prior Coordination

JAAMAS Special Issue on MIPC
AAMAS’17 Workshop on MIPC
mipc.inf.ed.ac.uk
Type-Based Method

Hypothesise possible types of other agents:

- Each type $\theta_j \in \Theta_j$ is blackbox behaviour specification

$$P(a_j|H^t, \theta_j)$$
Type-Based Method

Hypothesise possible **types** of other agents:

- Each type $\theta_j \in \Theta_j$ is blackbox behaviour specification

\[
P(a_j|H^t, \theta_j)
\]

- Compute belief over types based on interaction history $H^t$

\[
P(\theta_j|H^t) \propto P(H^t|\theta_j)P(\theta_j)
\]
Type-Based Method

Hypothesise possible types of other agents:

- Each type $\theta_j \in \Theta_j$ is blackbox behaviour specification

Compute belief over types based on interaction history $H^t$

$$P(\theta_j|H^t) \propto P(H^t|\theta_j)P(\theta_j)$$

Plan own action with respect to belief over types
Type-Based Method

- Type 1 → Action
- Type 2 → Action
- Type 3 → Action

Belief → Planning

History → Action
Agent → Own Action
Type-Based Method

- **HBA** (Albrecht & Ramamoorthy, AIJ’16)
- **PLASTIC** (Barrett & Stone, AIJ’16)
Type-Based Method and Parameters

Type-based method useful for ad hoc teamwork:

- Flexible – can hypothesise any types
- Efficient – can learn true type with few observations
- But...

Limitation: method does not recognise parameters in types!
Complex behaviours often have parameters
If we want to reason about $n$ parameter settings, have to store $n$ copies of same type with different parameter settings
⇒ Inefficient, does not scale
Type-Based Method and Parameters

Type-based method useful for ad hoc teamwork:
- Flexible – can hypothesise any types
- Efficient – can learn true type with few observations
- But...

Limitation: method does not recognise parameters in types!
- Complex behaviours often have parameters
- If we want to reason about $n$ parameter settings, have to store $n$ copies of same type with different parameter settings

$\Rightarrow$ Inefficient, does not scale
Goal in this work

Devise method which allows agent to reason about both:
- Relative likelihood of types \textit{and}
- Values of \textbf{bounded continuous parameters} in types
Type-Based Method and Parameters

Goal in this work

Devise method which allows agent to reason about both:
- Relative likelihood of types \textit{and}
- Values of \textit{bounded continuous parameters} in types

- Keep blackbox nature of types (can be any model)
- Work with any continuous parameters in types
Approach
Approach

For each $\theta_j \in \Theta_j$, maintain parameter estimate $p \in [p^{\text{min}}, p^{\text{max}}]^n$

Update estimates after new observations
Approach

For each $\theta_j \in \Theta_j$, maintain parameter estimate $p \in [p^\text{min}, p^\text{max}]^n$

Update estimates after new observations

Updating estimate incurs two computational costs:
Approach

For each $\theta_j \in \Theta_j$, maintain parameter estimate $p \in [p^{\text{min}}, p^{\text{max}}]^n$

Update estimates after new observations

Updating estimate incurs two computational costs:

- Computing new parameter estimate
  Types are blackboxes: must sample effects of parameters
  $\Rightarrow$ Need general, efficient estimation methods
Approach

For each $\theta_j \in \Theta_j$, maintain parameter estimate $p \in [p_{\text{min}}, p_{\text{max}}]^n$

Update estimates after new observations

Updating estimate incurs two computational costs:

- Computing new parameter estimate
  Types are blackboxes: must sample effects of parameters
  $\Rightarrow$ Need general, efficient estimation methods

- Adjusting internal state of type
  May depend on history of observations and parameter values
  $\Rightarrow$ New estimate may introduce model inconsistency
Approach: Selective Parameter Updating

Observe action $a_{j}^{t-1}$ of agent $j$
Approach: Selective Parameter Updating

Observe action $a_{j}^{t-1}$ of agent $j$

Select types $\Phi \subset \Theta_j$ for updating
Approach: Selective Parameter Updating

- Observe action $a_{j}^{t-1}$ of agent $j$
- Select types $\Phi \subset \Theta_j$ for updating
- For each $\theta_j \in \Phi$, update estimate $p^{t-1} \rightarrow p^t$
Approach: Selective Parameter Updating

Observe action $a_{j}^{t-1}$ of agent $j$

Select types $\Phi \subset \Theta_j$ for updating

For each $\theta_j \in \Phi$, update estimate $p^{t-1} \rightarrow p^t$

Update beliefs:

$$P(\theta_j|H^t) \propto P(a_{j}^{t-1}|H^{t-1}, \theta_j, p^t) P(\theta_j|H^{t-1})$$
Approach: Selective Parameter Updating

- Observe action $a_{j}^{t-1}$ of agent $j$
- Select types $\Phi \subset \Theta_j$ for updating
- For each $\theta_j \in \Phi$, update estimate $p^{t-1} \rightarrow p^t$
- Update beliefs:

$$P(\theta_j | H^t) \propto P(a_{j}^{t-1} | H^{t-1}, \theta_j, p^t) P(\theta_j | H^{t-1})$$

Plan own action
Approach: Selective Parameter Updating

1. Observe action $a_{j}^{t-1}$ of agent $j$
2. Select types $\Phi \subset \Theta_j$ for updating
3. For each $\theta_j \in \Phi$, update estimate $p^{t-1} \rightarrow p^t$
4. Update beliefs:
   
   $P(\theta_j|H^t) \propto P(a_{j}^{t-1}|H^{t-1}, \theta_j, p^t) \cdot P(\theta_j|H^{t-1})$

Plan own action
Updating Parameter Estimates

Given type $\theta_j$, update parameter estimate $p^{t-1} \rightarrow p^t$

Type defines action likelihoods

$$P(a_j^{t-1} | H^{t-1}, \theta_j, p)$$
Approximate Bayesian Updating (ABU)

Idea: construct Bayesian update using polynomials

- Maintain prior $P(p|H^{t-1}, \theta_j)$, represented as polynomial
Approximate Bayesian Updating (ABU)

Idea: construct Bayesian update using polynomials

- Maintain prior $P(p|H^{t-1}, \theta_j)$, represented as polynomial
- Approximate likelihood $f(p) = P(a_j^{t-1}|H^{t-1}, \theta_j, p)$ as polynomial by sampling over $p$
Approximate Bayesian Updating (ABU)

Idea: construct Bayesian update using polynomials

- Maintain prior $P(p|H^{t-1}, \theta_j)$, represented as polynomial
- Approximate likelihood $f(p) = P(a_j^{t-1}|H^{t-1}, \theta_j, p)$ as polynomial by sampling over $p$
- Take convolution of prior and likelihood, refit to original degree, normalise to get posterior $P(p|H^t, \theta_j)$
Approximate Bayesian Updating (ABU)

Idea: construct Bayesian update using polynomials

- Maintain prior $P(p|H^{t-1}, \theta_j)$, represented as polynomial
- Approximate likelihood $f(p) = P(a_j^{t-1}|H^{t-1}, \theta_j, p)$ as polynomial by sampling over $p$
- Take convolution of prior and likelihood, refit to original degree, normalise to get posterior $P(p|H^t, \theta_j)$
- Get parameter estimate $p^t$ by taking maximum or sampling from posterior
Approximate Bayesian Updating (ABU) – Example

\[ P(p|H_{t-1}^i, \theta_j) \]

\[ f \]

\[ \text{Fitted } \hat{f} \]

\[ \text{Samples from } \hat{g} \]

\[ \text{Fitted } \hat{h} \]

\[ P(p|H_t^i, \theta_j) \]

\[ p^* \]
Approximate Bayesian Updating (ABU) – Example

\[ P(p|H_{t-1}^i, \theta_j) \]

\[ p^* \]

Belief density

0 0.2 0.4 0.6 0.8 1

Fitted \( \hat{f} \)

Samples from \( \hat{g} \)

Posterior (blue)

Generate estimate

Likelihood of \( f \) from posterior \( p^{t-1}_a \)

action \( a_{t-1}^j \)

Past

Stefano Albrecht, Peter Stone
Reasoning about Hypothetical Agent Behaviours and their Parameters
Approximate Bayesian Updating (ABU) – Example

\[ f(p) = P(a_{j}^{t-1} | H^{t-1}, \theta_j, p) \]

### Prior

- **Belief density**: \( P(p|H_{i}^{-1}, \theta_j) \)
- **Belief density**: \( p^* \)

### Likelihood of \( a_{j}^{t-1} \) given type \( \theta_j \)

- **Samples from \( f \)**
- **Fitted \( \hat{f} \)**

### Past action

\( a_{j}^{t-1} \)

---

Stefano Albrecht, Peter Stone
Reasoning about Hypothetical Agent Behaviours and their Parameters
Approximate Bayesian Updating (ABU) – Example

\[ f(p) = P(a_j^{t-1} | H^{t-1}, \theta_j, p) \]

Past action \( a_j^{t-1} \)

Prior

Posterior (blue)

Stefano Albrecht, Peter Stone

Reasoning about Hypothetical Agent Behaviours and their Parameters
Approximate Bayesian Updating (ABU) – Example

Prior

\[ P(p|H^{t-1}, \theta_j) \]

\[ p^* \]

Posterior (blue)

\[ P(p|H^t, \theta_j) \]

\[ p^* \]

\[ \text{Past action } a_j^{t-1} \]

Likelihood of \( a_j^{t-1} \) given type \( \theta_j \)

\[ f(p) = P(a_j^{t-1}|H^{t-1}, \theta_j, p) \]

\[ \text{Generate estimate } p^t \text{ from posterior} \]
Exact Global Optimisation (EGO)

Estimation as **Global Optimisation**:

$$\arg\max_p \prod_{\tau=1}^{t} P(a_j^{\tau-1}|H^{\tau-1}, \theta_j, p)$$

Solve with Bayesian Optimisation
Approach: Selective Parameter Updating

Observe action $a_{j}^{t-1}$ of agent $j$

Select types $\Phi \subset \Theta_j$ for updating

For each $\theta_j \in \Phi$, update estimate $p^{t-1} \rightarrow p^t$

Update beliefs:

$$P(\theta_j|H^t) \propto P(a_{j}^{t-1}|H^{t-1}, \theta_j, p^t) P(\theta_j|H^{t-1})$$
Selecting Types for Parameter Updates

Expensive to update all types after each observation...

Idea: let agent decide which types to update
- Focus on types which are “most useful” to update

Two selection methods:
- Posterior selection
- Bandit selection
Posterior Selection

Focus on types which are believed to be most likely
- Don’t waste time on unlikely types

But: can lead to premature convergence of belief to wrong type...
- Occasionally update types which are less likely
Posterior Selection

Focus on types which are believed to be most likely
- Don’t waste time on unlikely types

But: can lead to premature convergence of belief to wrong type...
- Occasionally update types which are less likely

Tradeoff: sample $\Phi$ from belief $P(\theta_j|H^{t-1})$
Bandit Selection

Assumption: parameter estimates converge
- Focus on types which are expected to make largest leap toward convergence
- Don’t waste time on estimates that wouldn’t change much
Bandit Selection

Assumption: parameter estimates converge

- Focus on types which are expected to make largest leap toward convergence
- Don’t waste time on estimates that wouldn’t change much

Frame as multi-armed bandit problem:

- Each type $\theta_j$ is an arm
- Pulling arm (= updating type) $\theta_j$ gives reward

$$r^t = \eta^{-1} \sum_{k=1}^{n} |p_k^t - p_k^{t-1}|, \quad \eta = \sum_{k=1}^{n} p_{k}^{\text{max}} - p_{k}^{\text{min}}$$

- Can solve efficiently using bandit algorithm (e.g. UCB1)
Experiments
Level-Based Foraging

Blue = our agent, red = other agent

Goal: collect all items in minimal time

Agents and items have skill levels $\in [0, 1]$
⇒ Have to coordinate skills
Level-Based Foraging

Red has one of 4 types:

\( \theta^L_1 \): Search for item, try to load

\( \theta^L_2 \): Search for feasible item, try to load

\( \theta^F_1 \): Search for agent, load item closest to agent

\( \theta^F_2 \): Search for agent, load closest feasible item
Level-Based Foraging

Red has one of 4 types:

$\theta_j^{L1}$: Search for item, try to load

$\theta_j^{L2}$: Search for feasible item, try to load

$\theta_j^{F1}$: Search for agent, load item closest to agent

$\theta_j^{F2}$: Search for agent, load closest feasible item

Each type has 3 parameters:

- level $p_1$
- view radius $p_2$
- view angle $p_3$
Level-Based Foraging

Red has one of 4 types:

\( \theta_{j}^{L1} \): Search for item, try to load

\( \theta_{j}^{L2} \): Search for feasible item, try to load

\( \theta_{j}^{F1} \): Search for agent, load item closest to agent

\( \theta_{j}^{F2} \): Search for agent, load closest feasible item

Each type has 3 parameters:

- level \( p_1 \)
- view radius \( p_2 \)
- view angle \( p_3 \)

Blue does not know true type, parameter values, or meaning of parameters

Uses MCTS to plan own actions
Videos

2 agents, 5 items, 10x10 world
Starting with random parameter estimates
First video without updating
Second video with updating, using bandit selection and EGO

3 agents, 10 items, 15x15 world
Starting with random parameter estimates
First video without updating
Second video with updating, using bandit selection and EGO
Results

15x15 world, 10 items, 3 agents
Averaged over 500 random instances

% Completed

30  40  50  60  70  80

AGA  ABU  EGO

→ knows parameter values for true type

← random estimates, no updating
Results

15x15 world, 10 items, 3 agents
Averaged over 500 random instances

% Completed

AGA  ABU  EGO

knows parameter values for true type

updating one type after each observation

random estimates, no updating

Stefano Albrecht, Peter Stone
Reasoning about Hypothetical Agent Behaviours and their Parameters
Results

Average seconds (log-scale) needed per parameter update for single type
Results

Mean error in estimates of view radius $p_2$ for true type in 15x15 world (updating all types in each time step)
Results

Average belief $P(\theta_j^*|H^t)$ for true type $\theta_j^*$ in 10x10 world
(updating all types in each time step)
Conclusion

- Updating single type after each observation already achieves substantial improvements over random estimates.
Conclusion

- Updating single type after each observation already achieves substantial improvements over random estimates.

- Posterior selection tends to select more greedily than Bandit selection, premature convergence of beliefs.
Conclusion

- Updating single type after each observation already achieves substantial improvements over random estimates.

- Posterior selection tends to select more greedily than Bandit selection, premature convergence of beliefs.

- EGO best estimation, can detect parameter correlation, but also most expensive.
Conclusion

- Updating single type after each observation already achieves substantial improvements over random estimates.

- Posterior selection tends to select more greedily than Bandit selection, premature convergence of beliefs.

- EGO best estimation, can detect parameter correlation, but also most expensive.

- **Future work:** improved methods for type selection; theoretical understanding of interaction between parameter estimates and belief evolution.
Thank you
Algorithm: Selective Parameter Updating

**Given:** type space $\Theta_j$, initial belief $P(\theta_j|H^0)$ and parameter estimate $p^0$ for each $\theta_j \in \Theta_j$

**Repeat** for each $t > 0$:

1. Observe action $a_{j}^{t-1}$ of agent $j$
2. Select a subset $\Phi \subset \Theta_j$ for parameter updates
3. For each $\theta_j \in \Phi$:
   4. Obtain new parameter estimate $p^t$ for $\theta_j$
   5. Adjust internal state of $\theta_j$ wrt $p^t$
6. Set $p^t = p^{t-1}$ for all $\theta_j \not\in \Phi$
7. For each $\theta_j \in \Theta_j$, update belief:

\[
P(\theta_j|H^t) \propto P(a_{j}^{t-1}|H^{t-1}, \theta_j, p^t) P(\theta_j|H^{t-1})
\]