Inverse reinforcement learning from summary data

Antti Kangasrääsiö, Samuel Kaski

Aalto University, Finland

ECML PKDD 2018 journal track
Published in Machine Learning (2018), 107:1517–1535

September 12, 2018

Modelling human decision-making: Motivation

Our overarching goal is to have accurate white-box models of human decision-making

Modelling human decision-making: Motivation

Our overarching goal is to have accurate white-box models of human decision-making

Applications of high-fidelity user models

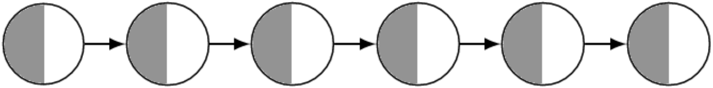
- Replicating demonstrated behavior (imitation learning)
- Optimizing user interfaces (human-computer interaction)
- Estimating cognitive state/goals of humans (chatbots)
- Understanding human cognition (cognitive science)

Modelling human decision-making: Problem

How to infer the parameters of sequential decision-making models when the available observation data is limited?

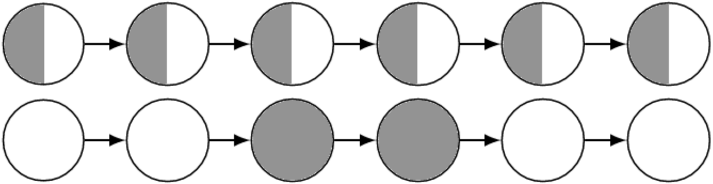
Modelling human decision-making: Problem

How to infer the parameters of sequential decision-making models when the available observation data is limited?



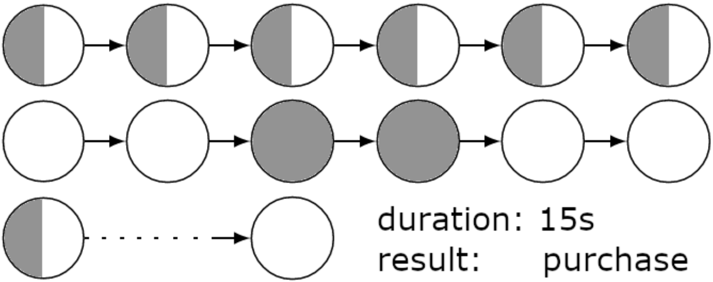
Modelling human decision-making: Problem

How to infer the parameters of sequential decision-making models when the available observation data is limited?



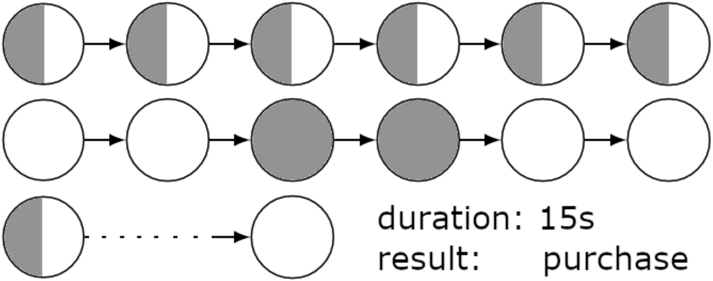
Modelling human decision-making: Problem

How to infer the parameters of sequential decision-making models when the available observation data is limited?



Modelling human decision-making: Problem

How to infer the parameters of sequential decision-making models when the available observation data is limited?



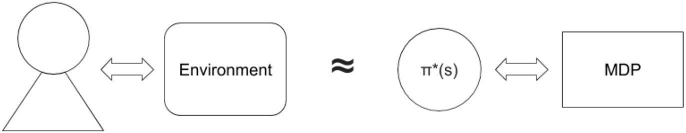
Main contribution: We demonstrate that posterior inference is possible for realistic models of decision-making, even with very limited observations of human behavior

Reinforcement learning models

We use the RL framework for modelling sequential decision-making

The main assumption is that **human decisions** can be approximated by an **optimal policy** trained for a certain **decision problem** (eg. MDP, POMDP)

"Humans make rational decisions within the limitations they have"



Inverse reinforcement learning (IRL)

Inverse reinforcement learning:
Given a set of observations, which MDP has a matching optimal policy?

Inverse reinforcement learning (IRL)

Inverse reinforcement learning:
Given a set of observations, which MDP has a matching optimal policy?

Traditional IRL problem

Given

- an MDP with reward-function $R(s;\theta)$, θ unknown
- a set of state-action trajectories $\Xi = \{\xi_1, \dots, \xi_N\}$ demonstrating optimal behavior, where $\xi_i = (s_0^i, a_1^i, \dots, a_{T_i-1}^i, s_{T_i}^i)$
- a prior $P(\theta)$

Determine a point estimate $\hat{\theta}$ or the posterior $P(\theta|\Xi)$

Existing solutions

Traditional IRL has been gradient descent on the likelihood

$$L(\theta|\Xi) = \prod_{i=1}^{N} P(s_0^i) \prod_{t=0}^{T_i-1} \pi_\theta^*(s_t^i, a_t^i) P(s_{t+1}^i | s_t^i, a_t^i)$$

Tractable when all states and actions are observed
what about when this is not the case?

[1] Activity forecasting, Kitani et al. 2012
[2] EM for IRL with hidden data, Bogert et al. 2016

Existing solutions

Traditional IRL has been gradient descent on the likelihood

$$L(\theta|\Xi) = \prod_{i=1}^{N} P(s_0^i) \prod_{t=0}^{T_i-1} \pi_\theta^*(s_t^i, a_t^i) P(s_{t+1}^i | s_t^i, a_t^i)$$

Tractable when all states and actions are observed
what about when this is not the case?

Previous work: If state observations are corrupted with i.i.d. noise[1] or part of them are missing[2], EM-approach can be used to estimate the true states, after which standard IRL methods apply

[1] Activity forecasting, Kitani et al. 2012
[2] EM for IRL with hidden data, Bogert et al. 2016

Existing solutions

Traditional IRL has been gradient descent on the likelihood

$$L(\theta|\Xi) = \prod_{i=1}^{N} P(s_0^i) \prod_{t=0}^{T_i-1} \pi_\theta^*(s_t^i, a_t^i) P(s_{t+1}^i | s_t^i, a_t^i)$$

Tractable when all states and actions are observed
what about when this is not the case?

Previous work: If state observations are corrupted with i.i.d. noise[1] or part of them are missing[2], EM-approach can be used to estimate the true states, after which standard IRL methods apply

However, this approach is not feasible in the more realistic cases, with complex non-i.i.d. noise or most of the states and actions missing

[1] Activity forecasting, Kitani et al. 2012
[2] EM for IRL with hidden data, Bogert et al. 2016

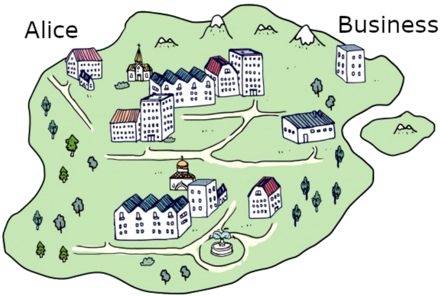
IRL from summary data (IRL-SD)

We ask whether IRL is possible in realistic cases, where the true trajectories ξ_i are filtered through a generic *summarizing function* σ, yielding summaries $\xi_{i\sigma} \sim \sigma(\xi_i)$

IRL from summary data (IRL-SD)

We ask whether IRL is possible in realistic cases, where the true trajectories ξ_i are filtered through a generic *summarizing function* σ, yielding summaries $\xi_{i\sigma} \sim \sigma(\xi_i)$

Example:
Alice walks to work every day along her preferred secret route. Could we infer Alice's scenery preferences given only the **durations of the commutes and the location of her work and home?**



IRL from summary data (IRL-SD)

We ask whether IRL is possible in realistic cases, where the true trajectories ξ_i are filtered through a generic *summarizing function* σ, yielding summaries $\xi_{i\sigma} \sim \sigma(\xi_i)$

Example:
Alice walks to work every day along her preferred secret route. Could we infer Alice's scenery preferences given only the **durations of the commutes and the location of her work and home?**

IRL from summary data (IRL-SD) problem

Given

- an MDP with unknown parameters θ
- a set of **summaries** $\Xi_\sigma = \{\xi_{1\sigma}, \dots, \xi_{N\sigma}\}$ from optimal behavior
- **the summary function** σ
- a prior $P(\theta)$

Determine a point estimate $\hat{\theta}$ or the posterior $P(\theta|\Xi_\sigma)$.

Exact solution

The likelihood corresponding to an IRL-SD problem is

$$L(\theta|\Xi_\sigma) = \prod_{i=1}^{N} \sum_{\xi_i \in \Xi_{ap}} P(\xi_{i\sigma}|\xi_i) P(\xi_i|\theta),$$

where we marginalize over the unobserved true ξ_i

Exact solution

The likelihood corresponding to an IRL-SD problem is

$$L(\theta|\Xi_\sigma) = \prod_{i=1}^{N} \sum_{\xi_i \in \Xi_{ap}} P(\xi_{i\sigma}|\xi_i)P(\xi_i|\theta),$$

where we marginalize over the unobserved true ξ_i

- The set of all plausible true trajectories is $\Xi_{ap} \subseteq S^{T_{max}+1} \times A^{T_{max}}$

Exact solution

The likelihood corresponding to an IRL-SD problem is

$$L(\theta|\Xi_\sigma) = \prod_{i=1}^{N} \sum_{\xi_i \in \Xi_{ap}} P(\xi_{i\sigma}|\xi_i) P(\xi_i|\theta),$$

where we marginalize over the unobserved true ξ_i

- The set of all plausible true trajectories is $\Xi_{ap} \subseteq S^{T_{max}+1} \times A^{T_{max}}$
- $P(\xi_{i\sigma}|\xi_i)$ is determined by the summary function σ

Exact solution

The likelihood corresponding to an IRL-SD problem is

$$L(\theta|\Xi_\sigma) = \prod_{i=1}^{N} \sum_{\xi_i \in \Xi_{ap}} P(\xi_{i\sigma}|\xi_i) P(\xi_i|\theta),$$

where we marginalize over the unobserved true ξ_i

- The set of all plausible true trajectories is $\Xi_{ap} \subseteq S^{T_{max}+1} \times A^{T_{max}}$
- $P(\xi_{i\sigma}|\xi_i)$ is determined by the summary function σ
- The likelihood of a trajectory is as before

$$P(\xi_i|\theta) = P(s_0^i) \prod_{t=0}^{T_i-1} \pi_\theta^*(s_t^i, a_t^i) P(s_{t+1}^i | s_t^i, a_t^i)$$

Exact solution

The likelihood corresponding to an IRL-SD problem is

$$L(\theta|\Xi_\sigma) = \prod_{i=1}^{N} \sum_{\xi_i \in \Xi_{ap}} P(\xi_{i\sigma}|\xi_i) P(\xi_i|\theta),$$

where we marginalize over the unobserved true ξ_i

- The set of all plausible true trajectories is $\Xi_{ap} \subseteq S^{T_{max}+1} \times A^{T_{max}}$
- $P(\xi_{i\sigma}|\xi_i)$ is determined by the summary function σ
- The likelihood of a trajectory is as before

$$P(\xi_i|\theta) = P(s_0^i) \prod_{t=0}^{T_i-1} \pi_\theta^*(s_t^i, a_t^i) P(s_{t+1}^i|s_t^i, a_t^i)$$

Takeaway: $L(\theta|\Xi_\sigma)$ can be evaluated, but it is very expensive to do so due to Ξ_{ap} being generally large or challenging to determine

Monte-Carlo approximation

We can estimate $L(\theta|\Xi_\sigma)$ by solving π^*_θ and then sampling N_{MC} trajectories, Ξ_{MC}, leading to the Monte-Carlo estimate

$$\hat{L}(\theta|\Xi_\sigma) = \prod_{i=1}^{N} \frac{1}{N_{MC}} \sum_{\xi_n \in \Xi_{MC}} P(\xi_{i\sigma}|\xi_n)$$

Monte-Carlo approximation

We can estimate $L(\theta|\Xi_\sigma)$ by solving π^*_θ and then sampling N_{MC} trajectories, Ξ_{MC}, leading to the Monte-Carlo estimate

$$\hat{L}(\theta|\Xi_\sigma) = \prod_{i=1}^{N} \frac{1}{N_{MC}} \sum_{\xi_n \in \Xi_{MC}} P(\xi_{i\sigma}|\xi_n)$$

However

- $P(\xi_{i\sigma}|\xi_n)$ may be 0 for all $\xi_n \in \Xi_{MC}$, forcing $\hat{L}(\theta|\Xi_\sigma)$ to be 0

Monte-Carlo approximation

We can estimate $L(\theta|\Xi_\sigma)$ by solving π_θ^* and then sampling N_{MC} trajectories, Ξ_{MC}, leading to the Monte-Carlo estimate

$$\hat{L}(\theta|\Xi_\sigma) \approx \prod_{i=1}^{N} \left(\frac{1}{N_{MC}} \sum_{\xi_n \in \Xi_{MC}} P(\xi_{i\sigma}|\xi_n) + \eta \right)$$

However

- $P(\xi_{i\sigma}|\xi_n)$ may be 0 for all $\xi_n \in \Xi_{MC}$, forcing $\hat{L}(\theta|\Xi_\sigma)$ to be 0 (can be fixed with a “prior” η)

Monte-Carlo approximation

We can estimate $L(\theta|\Xi_\sigma)$ by solving π^*_θ and then sampling N_{MC} trajectories, Ξ_{MC}, leading to the Monte-Carlo estimate

$$\hat{L}(\theta|\Xi_\sigma) \approx \prod_{i=1}^{N} \Big(\frac{1}{N_{MC}} \sum_{\xi_n \in \Xi_{MC}} P(\xi_{i\sigma}|\xi_n) + \eta \Big)$$

However

- $P(\xi_{i\sigma}|\xi_n)$ may be 0 for all $\xi_n \in \Xi_{MC}$, forcing $\hat{L}(\theta|\Xi_\sigma)$ to be 0 (can be fixed with a "prior" η)
- σ needs to be known as a distribution $P(\xi_{i\sigma}|\xi_n)$

Monte-Carlo approximation

We can estimate $L(\theta|\Xi_\sigma)$ by solving π^*_θ and then sampling N_{MC} trajectories, Ξ_{MC}, leading to the Monte-Carlo estimate

$$\hat{L}(\theta|\Xi_\sigma) \approx \prod_{i=1}^{N} \left(\frac{1}{N_{MC}} \sum_{\xi_n \in \Xi_{MC}} P(\xi_{i\sigma}|\xi_n) + \eta \right)$$

However

- $P(\xi_{i\sigma}|\xi_n)$ may be 0 for all $\xi_n \in \Xi_{MC}$, forcing $\hat{L}(\theta|\Xi_\sigma)$ to be 0 (can be fixed with a "prior" η)
- σ needs to be known as a distribution $P(\xi_{i\sigma}|\xi_n)$

Takeaway: $L(\theta|\Xi_\sigma)$ can be estimated with Monte-Carlo, but there are few technical issues we would like to avoid

Approximate Bayesian computation

ABC also performs inference using on Monte-Carlo sampling

- Instead of estimating the likelihood of each trajectory ξ_i separately, the likelihood of the entire observation set Ξ is estimated together

Approximate Bayesian computation

ABC also performs inference using on Monte-Carlo sampling

- Instead of estimating the likelihood of each trajectory ξ_i separately, the likelihood of the entire observation set Ξ is estimated together

How ABC works:

- Simulate observations using the MC sample: $\Xi_\sigma^{sim} = \{\sigma(\Xi_{MC,n})\}$ (only requires us to sample from σ)

Approximate Bayesian computation

ABC also performs inference using on Monte-Carlo sampling

- Instead of estimating the likelihood of each trajectory ξ_i separately, the likelihood of the entire observation set Ξ is estimated together

How ABC works:

- Simulate observations using the MC sample: $\Xi_\sigma^{sim} = \{\sigma(\Xi_{MC,n})\}$ (only requires us to sample from σ)
- Estimate discrepancy: $\delta(\Xi_\sigma, \Xi_\sigma^{sim}) \to [0, \infty)$ (matches distributions; reduces effect of individual rare observations)

Approximate Bayesian computation

ABC also performs inference using on Monte-Carlo sampling

- Instead of estimating the likelihood of each trajectory ξ_i separately, the likelihood of the entire observation set Ξ is estimated together

How ABC works:

- Simulate observations using the MC sample: $\Xi_\sigma^{sim} = \{\sigma(\Xi_{MC,n})\}$ (only requires us to sample from σ)
- Estimate discrepancy: $\delta(\Xi_\sigma, \Xi_\sigma^{sim}) \to [0, \infty)$ (matches distributions; reduces effect of individual rare observations)

The ε-approximate ABC likelihood: $\tilde{L}_\varepsilon(\theta|\Xi_\sigma) = P(\delta(\Xi_\sigma, \Xi_\sigma^{sim}) \leq \varepsilon|\theta)$

Approximate Bayesian computation

ABC also performs inference using on Monte-Carlo sampling

- Instead of estimating the likelihood of each trajectory ξ_i separately, the likelihood of the entire observation set Ξ is estimated together

How ABC works:

- Simulate observations using the MC sample: $\Xi_\sigma^{sim} = \{\sigma(\Xi_{MC,n})\}$ (only requires us to sample from σ)
- Estimate discrepancy: $\delta(\Xi_\sigma, \Xi_\sigma^{sim}) \rightarrow [0, \infty)$ (matches distributions; reduces effect of individual rare observations)

The ε-approximate ABC likelihood: $\tilde{L}_\varepsilon(\theta|\Xi_\sigma) = P(\delta(\Xi_\sigma, \Xi_\sigma^{sim}) \leq \varepsilon|\theta)$

Intuition: If simulating observations with θ leads to small prediction error, then likelihood of θ is high and vice versa

Approximate Bayesian computation

ABC also performs inference using on Monte-Carlo sampling

- Instead of estimating the likelihood of each trajectory ξ_i separately, the likelihood of the entire observation set Ξ is estimated together

How ABC works:

- Simulate observations using the MC sample: $\Xi_\sigma^{sim} = \{\sigma(\Xi_{MC,n})\}$ (only requires us to sample from σ)
- Estimate discrepancy: $\delta(\Xi_\sigma, \Xi_\sigma^{sim}) \rightarrow [0, \infty)$ (matches distributions; reduces effect of individual rare observations)

The ε-approximate ABC likelihood: $\tilde{L}_\varepsilon(\theta|\Xi_\sigma) = P(\delta(\Xi_\sigma, \Xi_\sigma^{sim}) \leq \varepsilon|\theta)$

Intuition: If simulating observations with θ leads to small prediction error, then likelihood of θ is high and vice versa

Takeaway: The issues with MC (numerical problems with rare observations, σ known as a distribution) can be avoided by using ABC

Inference

Now we can estimate $L(\theta|\Xi)$ at any θ, but how to find the best $\theta \in \Theta$?

- Evaluating the functions is still expensive
- The functions don't have accessible gradients
- Due to limited observability (σ), parameter uncertainty is likely large

Inference

Now we can estimate $L(\theta|\Xi)$ at any θ, but how to find the best $\theta \in \Theta$?

- Evaluating the functions is still expensive
- The functions don't have accessible gradients
- Due to limited observability (σ), parameter uncertainty is likely large

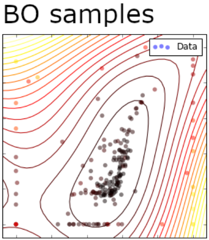
We estimate the log-likelihoods using a GP surrogate model, fit using Bayesian optimization. Mean and shape of distribution estimated from MCMC-samples.

Inference

Now we can estimate $L(\theta|\Xi)$ at any θ, but how to find the best $\theta \in \Theta$?

- Evaluating the functions is still expensive
- The functions don't have accessible gradients
- Due to limited observability (σ), parameter uncertainty is likely large

We estimate the log-likelihoods using a GP surrogate model, fit using Bayesian optimization. Mean and shape of distribution estimated from MCMC-samples.

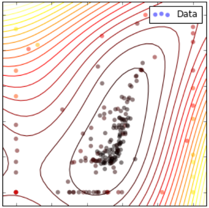


Now we can estimate $L(\theta|\Xi)$ at any θ, but how to find the best $\theta \in \Theta$?

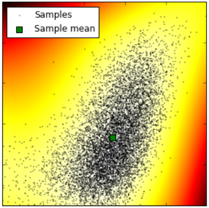
- Evaluating the functions is still expensive
- The functions don't have accessible gradients
- Due to limited observability (σ), parameter uncertainty is likely large

We estimate the log-likelihoods using a GP surrogate model, fit using Bayesian optimization. Mean and shape of distribution estimated from MCMC-samples.

BO samples



MCMC samples

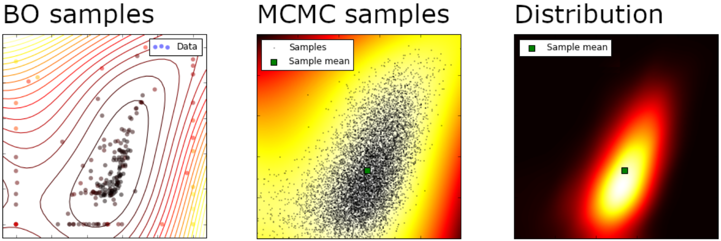


Inference

Now we can estimate $L(\theta|\Xi)$ at any θ, but how to find the best $\theta \in \Theta$?

- Evaluating the functions is still expensive
- The functions don't have accessible gradients
- Due to limited observability (σ), parameter uncertainty is likely large

We estimate the log-likelihoods using a GP surrogate model, fit using Bayesian optimization. Mean and shape of distribution estimated from MCMC-samples.



Simulation experiment

We used grid world environments to validate our approach

- Task was to infer reward weights for state features: $R(s) = \phi(s)^T \theta$

Simulation experiment

We used grid world environments to validate our approach

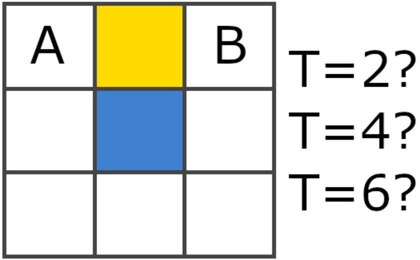
- Task was to infer reward weights for state features: $R(s) = \phi(s)^T \theta$
- We only knew the start and end locations of the agent and the length of the trajectory: $\xi_\sigma = (s_0, s_T, T)$

Simulation experiment

We used grid world environments to validate our approach

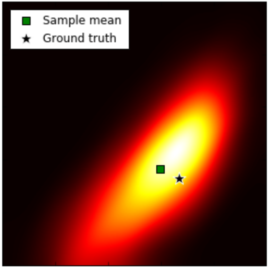
- Task was to infer reward weights for state features: $R(s) = \phi(s)^T \theta$
- We only knew the start and end locations of the agent and the length of the trajectory: $\xi_\sigma = (s_0, s_T, T)$

Miniature example:
"What kind of terrain might the agent prefer, given that moving from A to B took it T steps?"



Inferred distributions (example)

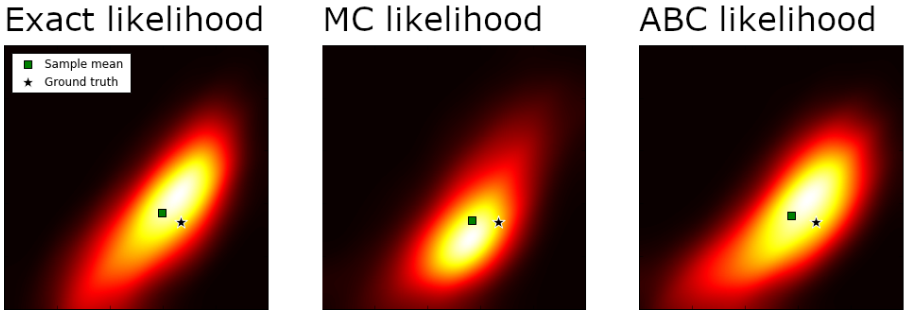
Exact likelihood



Takeaways

The parameter values can be inferred based on summary observations

Inferred distributions (example)

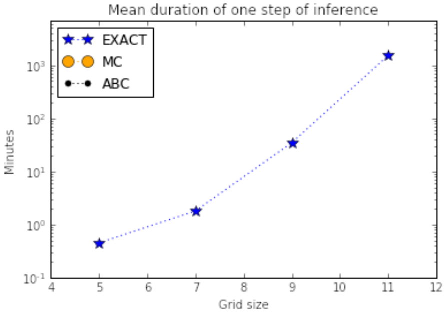


Takeaways

The parameter values can be inferred based on summary observations

The approximate distributions are similar to the true distribution

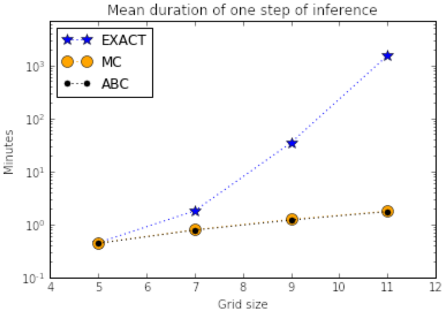
Efficiency



Takeaways

Summing over all plausible trajectories is expensive with larger MDPs

Efficiency

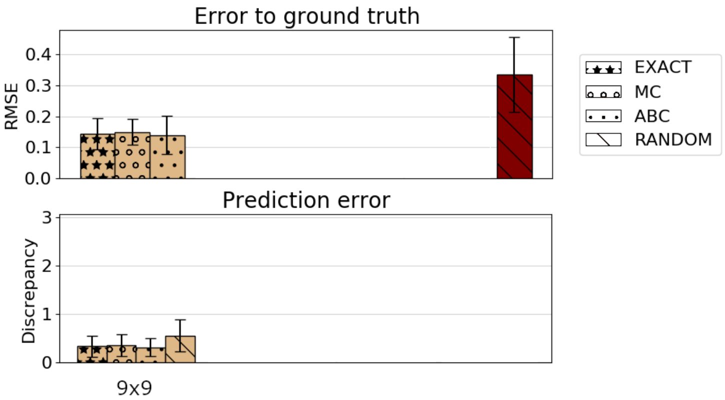


Takeaways

Summing over all plausible trajectories is expensive with larger MDPs

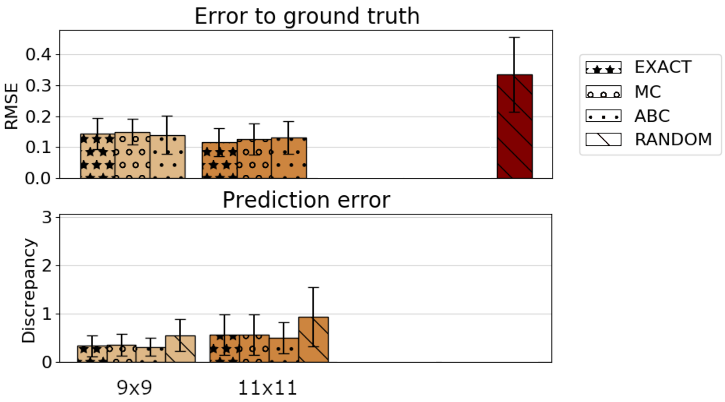
The approximate methods scale significantly better

Accuracy and model fit



Takeaways

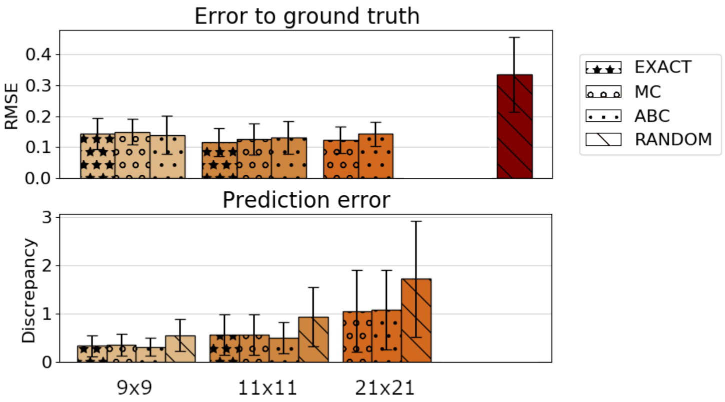
Accuracy and model fit



Takeaways

Good approximation performance while outperforming a random baseline

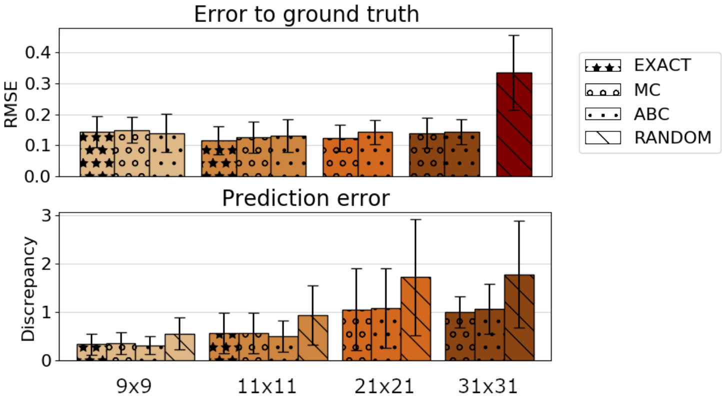
Accuracy and model fit



Takeaways

Good approximation performance while outperforming a random baseline

Accuracy and model fit



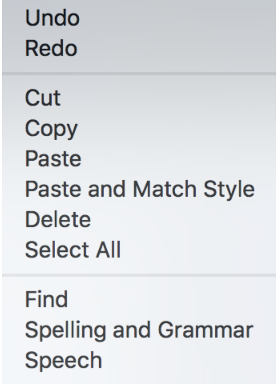
Takeaways

Good approximation performance while outperforming a random baseline

Approximate methods continue performing well even with larger MDPs

Realistic Experiment

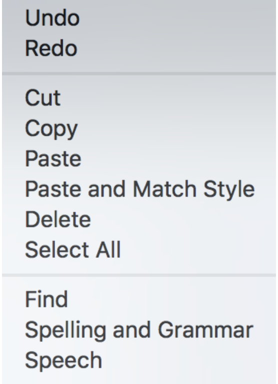
We performed experiments using an RL model from cognitive science



- User searched repeatedly for target items from drop-down menus

Realistic Experiment

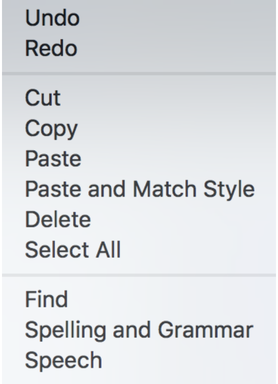
We performed experiments using an RL model from cognitive science



- User searched repeatedly for target items from drop-down menus
- The MDP contained a simple model of human vision and short-term memory

Realistic Experiment

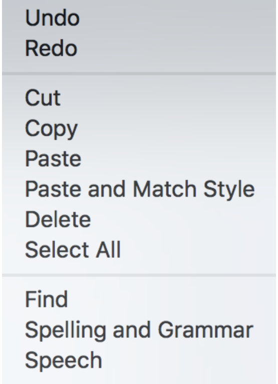
We performed experiments using an RL model from cognitive science



- User searched repeatedly for target items from drop-down menus
- The MDP contained a simple model of human vision and short-term memory
- Goal: infer values of three model parameters based on observing task completion times (TCT) and whether the target item was present in the menu:
  $\xi_\sigma = (target\_present?, TCT)$

Realistic Experiment

We performed experiments using an RL model from cognitive science



- User searched repeatedly for target items from drop-down menus
- The MDP contained a simple model of human vision and short-term memory
- Goal: infer values of three model parameters based on observing task completion times (TCT) and whether the target item was present in the menu:
  $\xi_\sigma = (target\_present?, TCT)$
  - visual fixation duration f_{dur}
  - item selection duration d_{sel}
  - menu layout recall probability p_{rec}

Model fit

	ABC	Hold-out data
Task Completion Time (abs)	430 ms	470 ms
Task Completion Time (pre)	980 ms	970 ms

abs = target absent from menu, pre = target present in menu

Takeaways

Predictions with parameters inferred by ABC match to hold-out observation data, indicating good model fit

Model fit

	ABC	Hold-out data
Task Completion Time (abs)	430 ms	470 ms
Task Completion Time (pre)	980 ms	970 ms
Number of Saccades (abs)	1.4	1.9
Number of Saccades (pre)	3.1	2.2

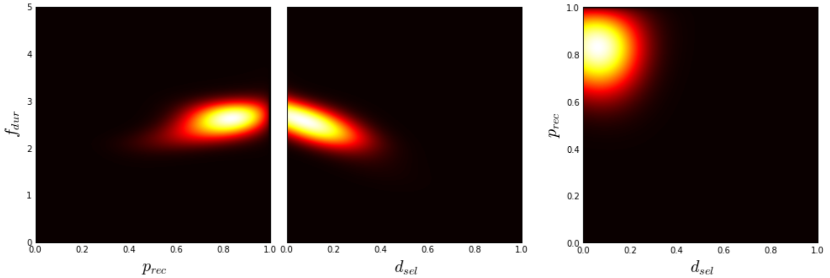
abs = target absent from menu, pre = target present in menu

Takeaways

Predictions with parameters inferred by ABC match to hold-out observation data, indicating good model fit

Also unobserved features match approximately to predictions

Approximate posterior



Takeaway

Posterior indicates good identification of model parameter values

Remaining parameter uncertainty is easy to visualize

Conclusions

We proposed two approximate methods (MC, ABC) for solving the problem of trajectory-level observation noise in IRL

- More scalable than exact likelihood
- Good approximation quality
- Full posterior inference, which is important due to noisy observations

Conclusions

We proposed two approximate methods (MC, ABC) for solving the problem of trajectory-level observation noise in IRL

- More scalable than exact likelihood
- Good approximation quality
- Full posterior inference, which is important due to noisy observations

We demonstrated applicability for a realistic cognitive science model based on real observation data

Conclusions

We proposed two approximate methods (MC, ABC) for solving the problem of trajectory-level observation noise in IRL

- More scalable than exact likelihood
- Good approximation quality
- Full posterior inference, which is important due to noisy observations

We demonstrated applicability for a realistic cognitive science model based on real observation data

Next steps: improve scalability

- Still requires solving RL problems in the inner loop
- Scalability of GP and BO to high dimensions

Conclusions

We proposed two approximate methods (MC, ABC) for solving the problem of trajectory-level observation noise in IRL

- More scalable than exact likelihood
- Good approximation quality
- Full posterior inference, which is important due to noisy observations

We demonstrated applicability for a realistic cognitive science model based on real observation data

Next steps: improve scalability

- Still requires solving RL problems in the inner loop
- Scalability of GP and BO to high dimensions

More details at the poster tomorrow