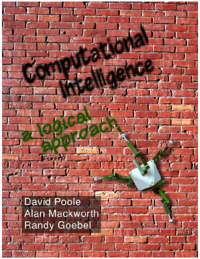
The Essence of Intelligence is Appropriate Action (not thinking, reasoning, learning or language) and other things every student of AI should know

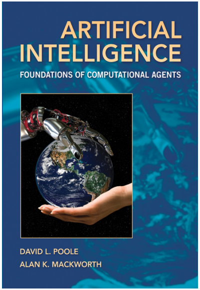
David Poole and Alan Mackworth

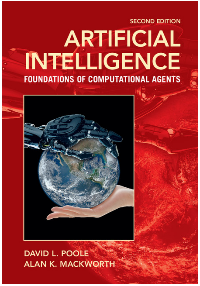
Department of Computer Science,
University of British Columbia

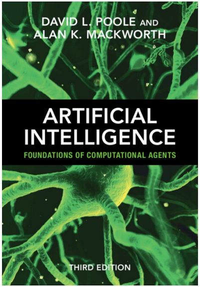
January 25, 2026

Books and online resources









`https://artint.info` – full text online
`https://www.aispace.org` – Java apps for learning AI
`http://aipython.org` – open source Python code

Audience Question

Humans evolved intelligence because

A it was selected for when finding a suitable mate

B it enabled behavior that helped humans to survive and flourish

C it enabled them to spread gossip and misinformation

D it enabled people to have deep inner thoughts

E it was just random

Outline

Agents

What should an agent believe?

What should an agent do?

Where do values/goals/preferences come from?

Outline

Agents

What should an agent believe?

What should an agent do?

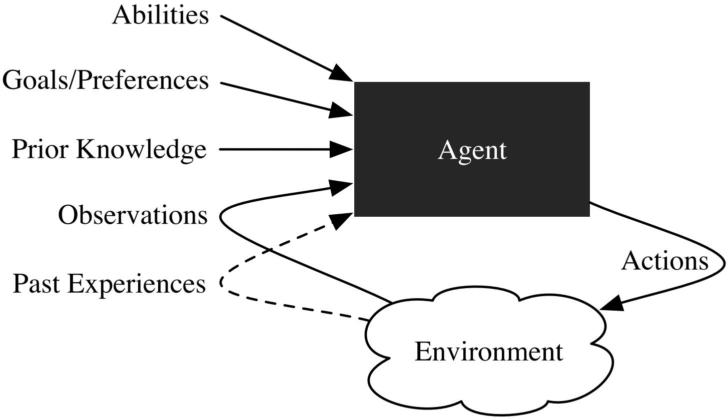
Where do values/goals/preferences come from?

Artificial Intelligence and Agents

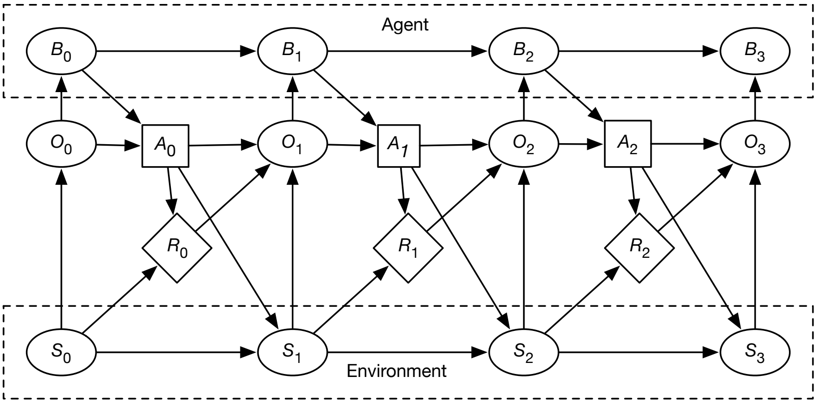
- Artificial intelligence is the synthesis and analysis of computational agents that act appropriately.

Artificial Intelligence and Agents

- Artificial intelligence is the synthesis and analysis of computational agents that act appropriately.

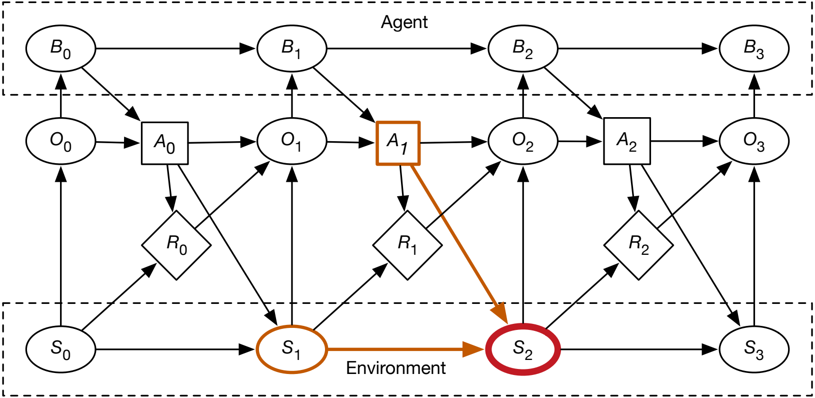


An agent situated in time



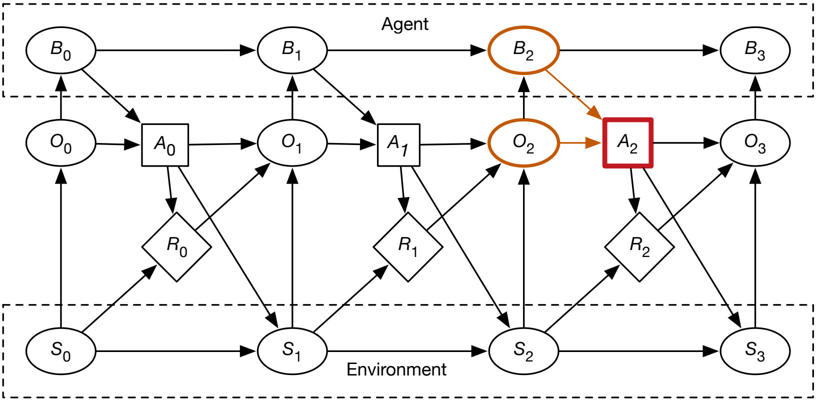
- S_t environment state at time t
- A_t agent's action
- O_t observation
- B_t agent's belief state
- R_t agent's reward

An agent situated in time



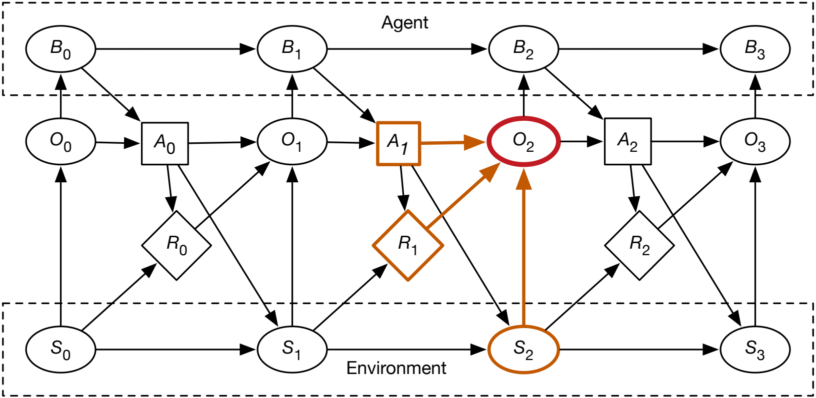
- S_t environment state at time t
- A_t agent's action
- O_t observation
- B_t agent's belief state
- R_t agent's reward

An agent situated in time



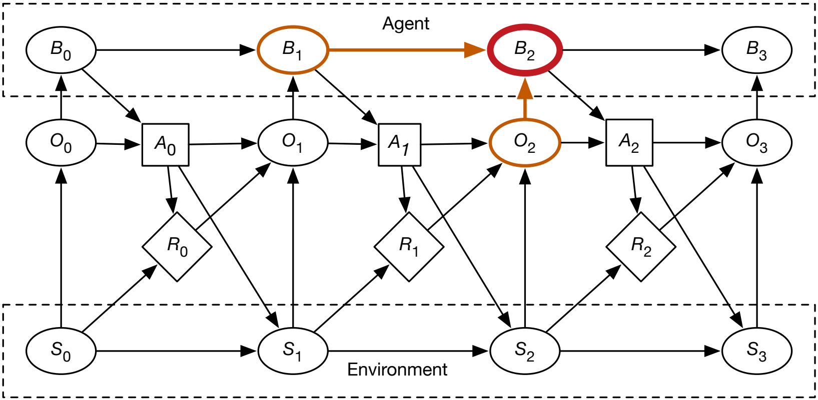
- S_t environment state at time t
- A_t agent's action
- O_t observation
- B_t agent's belief state
- R_t agent's reward

An agent situated in time



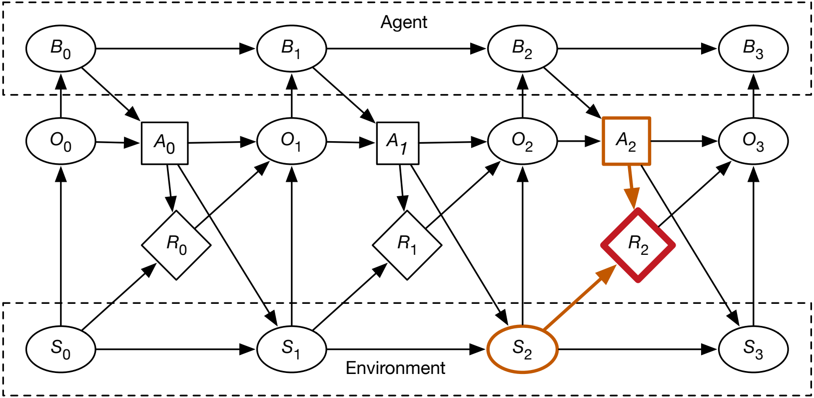
- S_t environment state at time t
- A_t agent's action
- O_t observation
- B_t agent's belief state
- R_t agent's reward

An agent situated in time



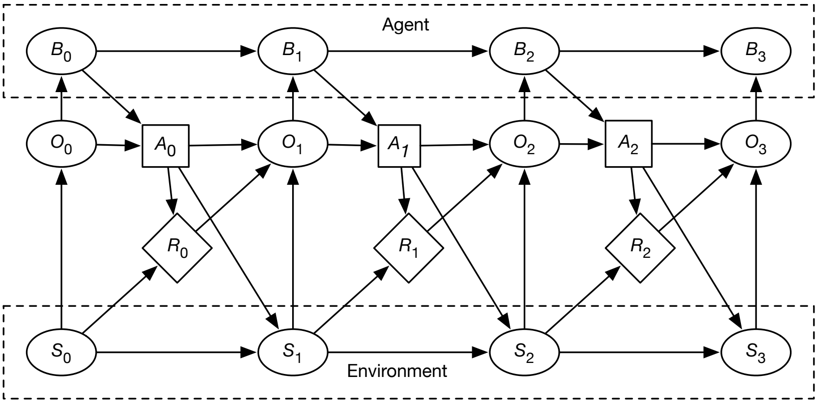
- S_t environment state at time t
- A_t agent's action
- O_t observation
- B_t agent's belief state
- R_t agent's reward

An agent situated in time



- S_t environment state at time t
- A_t agent's action
- O_t observation
- B_t agent's belief state
- R_t agent's reward

An agent situated in time



- S_t environment state at time t
- A_t agent's action
- O_t observation
- B_t agent's belief state
- R_t agent's reward

Outline

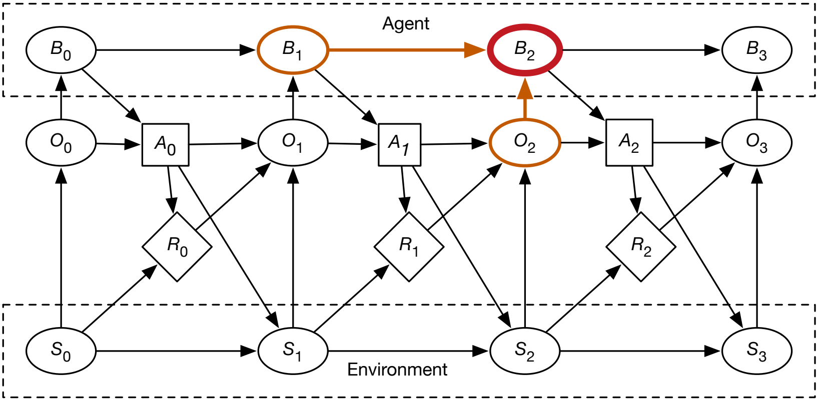
Agents

What should an agent believe?

What should an agent do?

Where do values/goals/preferences come from?

An agent situated in time



- S_t environment state at time t
- A_t agent's action
- O_t observation
- B_t agent's belief state
- R_t agent's reward

What if the environment is not fully-observable?

- When an agent does not know the outcome of its action, it is gambling. Probability is the calculus of gambling.

What if the environment is not fully-observable?

- When an agent does not know the outcome of its action, it is gambling. Probability is the calculus of gambling.
- Learning a conditional probability $P(y|x)$ from $\{(x_1, y_1), (x_2, y_2), \dots\}$ data is supervised learning.

What if the environment is not fully-observable?

- When an agent does not know the outcome of its action, it is gambling. Probability is the calculus of gambling.
- Learning a conditional probability $P(y|x)$ from $\{(x_1, y_1), (x_2, y_2), \ldots\}$ data is supervised learning.
- Common representations of conditional probabilities:
  - tables
  - decision trees
  - logistic regression
  - noisy-or
  - gradient boosted trees
  - neural networks (with sigmoid or softmax output)

What if the environment is not fully-observable?

- When an agent does not know the outcome of its action, it is gambling. Probability is the calculus of gambling.
- Learning a conditional probability $P(y|x)$ from $\{(x_1, y_1), (x_2, y_2), \ldots\}$ data is supervised learning.
- Common representations of conditional probabilities:
  - tables
  - decision trees
  - logistic regression
  - noisy-or
  - gradient boosted trees
  - neural networks (with sigmoid or softmax output)
- Common: end-to-end (monolithic) learning: learn everything at once.

Example: diagnostic agent for human diseases

- belief state is joint probability distribution of patient's diseases

Example: diagnostic agent for human diseases

- belief state is joint probability distribution of patient's diseases
- observations are the symptoms of patient, test results, electronic health records. . .

Example: diagnostic agent for human diseases

- belief state is joint probability distribution of patient's diseases
- observations are the symptoms of patient, test results, electronic health records. . .
- belief state update: $P(Diseases \mid Symptoms)$.
  Example: $P(hepatitis \mid chills, fever)$

Example: diagnostic agent for human diseases

- belief state is joint probability distribution of patient's diseases
- observations are the symptoms of patient, test results, electronic health records. . .
- belief state update: $P(Diseases \mid Symptoms)$.
  Example: $P(hepatitis \mid chills, fever)$
- Could use end-to-end learning to learn the conditional distribution

Example: diagnostic agent for human diseases

- belief state is joint probability distribution of patient's diseases
- observations are the symptoms of patient, test results, electronic health records. . .
- belief state update: $P(Diseases \mid Symptoms)$.
  Example: $P(hepatitis \mid chills, fever)$
- Could use end-to-end learning to learn the conditional distribution
- Probability depends on the location, eg Cambodia vs Canada, time of year, prevalence of flu, whether patient is vaccinated against flu and/or hepatitis, sickness of contacts and where they have travelled, . . .

Example: diagnostic agent for human diseases

- belief state is joint probability distribution of patient's diseases
- observations are the symptoms of patient, test results, electronic health records...
- belief state update: $P(Diseases \mid Symptoms)$.
  Example: $P(hepatitis \mid chills, fever)$
- Could use end-to-end learning to learn the conditional distribution
- Probability depends on the location, eg Cambodia vs Canada, time of year, prevalence of flu, whether patient is vaccinated against flu and/or hepatitis, sickness of contacts and where they have travelled, ...
- Want $P(Diseases \mid Symptoms, Location, \dots)$

Example: diagnostic agent for human diseases

- belief state is joint probability distribution of patient's diseases
- observations are the symptoms of patient, test results, electronic health records. . .
- belief state update: $P(Diseases \mid Symptoms)$.
  Example: $P(hepatitis \mid chills, fever)$
- Could use end-to-end learning to learn the conditional distribution
- Probability depends on the location, eg Cambodia vs Canada, time of year, prevalence of flu, whether patient is vaccinated against flu and/or hepatitis, sickness of contacts and where they have travelled, . . .
- Want $P(Diseases \mid Symptoms, Location, \ldots)$

$\boxed{\text{Do we just need to learn with more data?}}$

Can't we just get more data?

- Many domains have abundant homogenous data:
  sensor networks, images of cats, text, DNA sequences, ...
  Or generate arbitrary amount of data: games, programs,...
  Usually better to collect more data than provide knowledge.

Can't we just get more data?

- Many domains have abundant homogenous data:
  sensor networks, images of cats, text, DNA sequences, ...
  Or generate arbitrary amount of data: games, programs,...
  Usually better to collect more data than provide knowledge.
- SNOMED-CT is a healthcare terminology with over 360,000 concepts in multiple languages.

Can't we just get more data?

- Many domains have abundant homogenous data:
  sensor networks, images of cats, text, DNA sequences, ...
  Or generate arbitrary amount of data: games, programs,...
  Usually better to collect more data than provide knowledge.
- SNOMED-CT is a healthcare terminology with over 360,000 concepts in multiple languages.
- Over 100,000 of these are diseases and other ailments

Can't we just get more data?

- Many domains have abundant homogenous data:
  sensor networks, images of cats, text, DNA sequences, ...
  Or generate arbitrary amount of data: games, programs,...
  Usually better to collect more data than provide knowledge.
- SNOMED-CT is a healthcare terminology with over 360,000 concepts in multiple languages.
- Over 100,000 of these are diseases and other ailments
- Many people have multiple diseases.
- There are about the same number of pairs of diseases as there are people on Earth.

Can't we just get more data?

- Many domains have abundant homogenous data:
  sensor networks, images of cats, text, DNA sequences, ...
  Or generate arbitrary amount of data: games, programs,...
  Usually better to collect more data than provide knowledge.
- SNOMED-CT is a healthcare terminology with over 360,000 concepts in multiple languages.
- Over 100,000 of these are diseases and other ailments
- Many people have multiple diseases.
- There are about the same number of pairs of diseases as there are people on Earth.
- With 40 diseases, there are more combinations of diseases than people who have ever lived

Can't we just get more data?

- Many domains have abundant homogenous data:
  sensor networks, images of cats, text, DNA sequences, ...
  Or generate arbitrary amount of data: games, programs,...
  Usually better to collect more data than provide knowledge.
- SNOMED-CT is a healthcare terminology with over 360,000 concepts in multiple languages.
- Over 100,000 of these are diseases and other ailments
- Many people have multiple diseases.
- There are about the same number of pairs of diseases as there are people on Earth.
- With 40 diseases, there are more combinations of diseases than people who have ever lived
- There can't be enough data to learn disease interactions using only data-hungry approaches

Can't we just get more data?

- Many domains have abundant homogenous data:
  sensor networks, images of cats, text, DNA sequences, . . .
  Or generate arbitrary amount of data: games, programs,. . .
  Usually better to collect more data than provide knowledge.
- SNOMED-CT is a healthcare terminology with over 360,000 concepts in multiple languages.
- Over 100,000 of these are diseases and other ailments
- Many people have multiple diseases.
- There are about the same number of pairs of diseases as there are people on Earth.
- With 40 diseases, there are more combinations of diseases than people who have ever lived
- There can't be enough data to learn disease interactions using only data-hungry approaches
- Same argument goes for symptoms.

Modularity

Avoiding the combinatorics $P(Diseases \mid Symptoms, Location, \dots)$

- Computer science solution:

Modularity

Avoiding the combinatorics $P(\mathit{Diseases} \mid \mathit{Symptoms}, \mathit{Location}, \ldots)$

- Computer science solution: to solve a complex problem break it into simpler problems.

Modularity

Avoiding the combinatorics $P(Diseases \mid Symptoms, Location, \ldots)$

- Computer science solution: to solve a complex problem break it into simpler problems.
- Modular: components learned separately and combined.

Modularity

Avoiding the combinatorics $P(Diseases \mid Symptoms, Location, \ldots)$

- Computer science solution: to solve a complex problem break it into simpler problems.
- Modular: components learned separately and combined.
- Transportable: components are stable when changing locations/times/populations.

Modularity

Avoiding the combinatorics $P(Diseases \mid Symptoms, Location, \dots)$

- Computer science solution: to solve a complex problem break it into simpler problems.
- Modular: components learned separately and combined.
- Transportable: components are stable when changing locations/times/populations.
- Bayes' rule:

$$P(Diseases \mid Symptoms) = \frac{P(Symptoms \mid Diseases)P(Diseases)}{P(Symptoms)}$$

Modularity

Avoiding the combinatorics $P(Diseases \mid Symptoms, Location, \dots)$

- Computer science solution: to solve a complex problem break it into simpler problems.
- Modular: components learned separately and combined.
- Transportable: components are stable when changing locations/times/populations.
- Bayes' rule:

$$P(Diseases \mid Symptoms) = \frac{P(Symptoms \mid Diseases) P(Diseases)}{P(Symptoms)}$$

- $P(Symptoms \mid Diseases)$ is stable across time, populations, where people have travelled, friends, etc.

Modularity

Avoiding the combinatorics $P(Diseases \mid Symptoms, Location, \ldots)$

- Computer science solution: to solve a complex problem break it into simpler problems.
- Modular: components learned separately and combined.
- Transportable: components are stable when changing locations/times/populations.
- Bayes' rule:

$$P(Diseases \mid Symptoms) = \frac{P(Symptoms \mid Diseases) P(Diseases)}{P(Symptoms)}$$

- $P(Symptoms \mid Diseases)$ is stable across time, populations, where people have travelled, friends, etc.
- What happens when a new virus arrives?

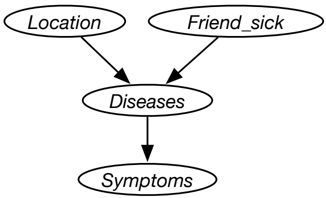
Context and Independence

$$
\begin{aligned}
P(Diseases \mid Symptoms, Location, Friend\_sick) \\
\propto P(Symptoms \mid Diseases) \\
* P(Diseases \mid Location, Friend\_sick)
\end{aligned}
$$

Context and Independence

$$P(Diseases \mid Symptoms, Location, Friend\_sick)$$
$$\propto P(Symptoms \mid Diseases)$$
$$* P(Diseases \mid Location, Friend\_sick)$$

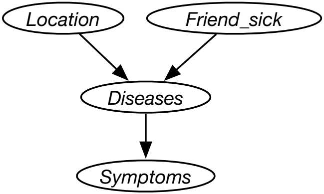
- belief network (Bayesian network, directed graphical model) represents conditional independence.
- arc means "depends on"
- independence assumption: $P(Symptoms \mid Diseases)$ does not depend on $Location$, $Friend\_sick$.



Context and Independence

$$\begin{aligned} P(Diseases \mid Symptoms, Location, Friend\_sick) \\ \propto P(Symptoms \mid Diseases) \\ * P(Diseases \mid Location, Friend\_sick) \end{aligned}$$

- belief network (Bayesian network, directed graphical model) represents conditional independence.
- arc means "depends on"
- independence assumption: $P(Symptoms \mid Diseases)$ does not depend on $Location$, $Friend\_sick$.

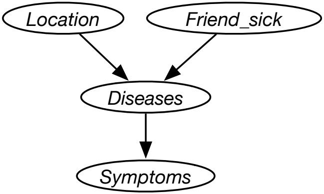


- modular, transportable

Context and Independence

$$P(Diseases \mid Symptoms, Location, Friend\_sick)$$
$$\propto P(Symptoms \mid Diseases)$$
$$* P(Diseases \mid Location, Friend\_sick)$$

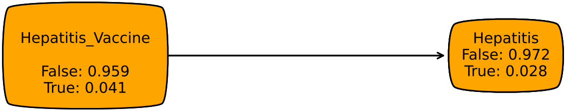
- belief network (Bayesian network, directed graphical model) represents conditional independence.
- arc means "depends on"
- independence assumption: $P(Symptoms \mid Diseases)$ does not depend on $Location$, $Friend\_sick$.



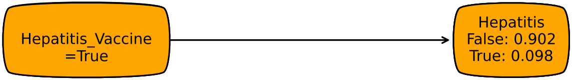
- modular, transportable
- can be learned from data

Determining the Effects of Acting

$P(Hepatitis)$, $P(Hepatitis\_Vaccine)$ – fictional probabilities

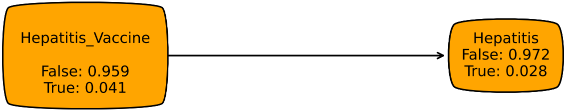


$P(Hepatitis \mid Hepatitis\_Vaccine)$

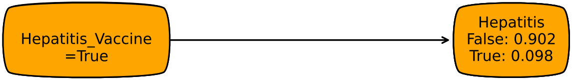


Determining the Effects of Acting

$P(Hepatitis)$, $P(Hepatitis\_Vaccine)$ – fictional probabilities



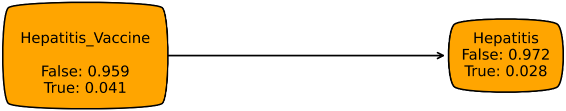
$P(Hepatitis \mid Hepatitis\_Vaccine)$



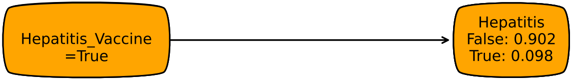
What if we give the vaccine?

Determining the Effects of Acting

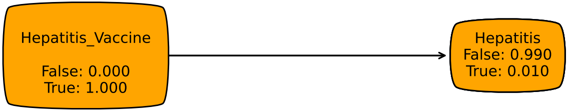
$P(Hepatitis)$, $P(Hepatitis\_Vaccine)$ – fictional probabilities



$P(Hepatitis \mid Hepatitis\_Vaccine)$

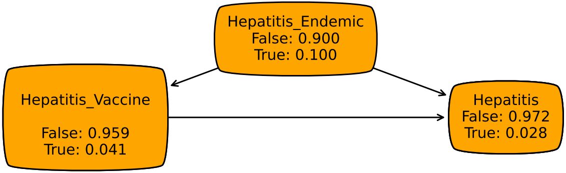


What if we give the vaccine?

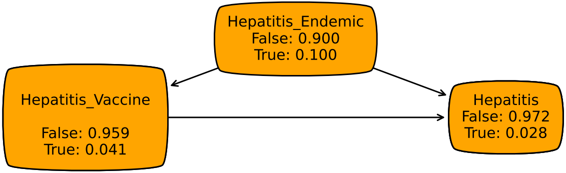


Pearl's do notation $P(Hepatitis \mid do(Hepatitis\_Vaccine))$

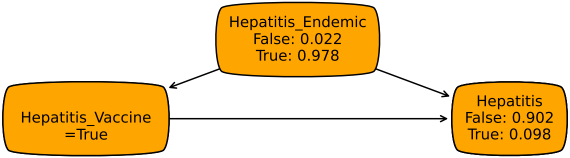
Determining the Effects of Acting



Determining the Effects of Acting



$P(Hepatitis \mid Hepatitis\_Vaccine)$.



A causal model predicts the effects of an action
Hepatitis_Endemic is a confounder.
Open source code at AIPython.org

Effects of acting cannot be learned from observational data

- Effects of acting cannot be learned from observational data!

Effects of acting cannot be learned from observational data

- Effects of acting cannot be learned from observational data!
- An agent has to interact with the world to determine the effects of actions.

Effects of acting cannot be learned from observational data

- Effects of acting cannot be learned from observational data!
- An agent has to interact with the world to determine the effects of actions.
- It needs a real body to carry out experiments in its world

Effects of acting cannot be learned from observational data

- Effects of acting cannot be learned from observational data!
- An agent has to interact with the world to determine the effects of actions.
- It needs a real body to carry out experiments in its world
- It could use all connected sensors and actuators, as long as it can control the actuators.

Missing Data

Missing Data

If missing data (dropped out) are ignored:

Missing Data

If missing data (dropped out) are ignored:

- higher proportion of "took drug" are well than placebo

Missing Data

If missing data (dropped out) are ignored:

- higher proportion of "took drug" are well than placebo
- fewer are sick

Missing Data

If missing data (dropped out) are ignored:

- higher proportion of "took drug" are well than placebo
- fewer are sick
- fewer are very sick

Missing Data

If missing data (dropped out) are ignored:

- higher proportion of "took drug" are well than placebo
- fewer are sick
- fewer are very sick
- the drug looks good!

Missing Data

If missing data (dropped out) are ignored:

- higher proportion of "took drug" are well than placebo
- fewer are sick
- fewer are very sick
- the drug looks good!

Model/find out why data is missing!

Audience Question

What is the real world made of?

A Features or random variables

B Words, pixels, phonemes ...

C Entities and events (e.g., plants, people, diseases, talks, conferences)

D Huh? There is a real world?

Relational Models

"The mind is a neural computer, fitted by natural selection with combinatorial algorithms for causal and probabilistic reasoning about plants, animals, objects, and people."

How the Mind Works, Steven Pinker, 1997

Relational Models

"The mind is a neural computer, fitted by natural selection with combinatorial algorithms for causal and probabilistic reasoning about plants, animals, objects, and people."

How the Mind Works, Steven Pinker, 1997

Relational Learning

Relational learning is learning models that make (probabilistic) predictions about entities (things, objects, including events):

- their properties
- relations among them
- existence
- identity

Relational Learning

Relational learning is learning models that make (probabilistic) predictions about entities (things, objects, including events):

- their properties
- relations among them
- existence
- identity

Also statistical relational AI,
relational probabilistic models,
logic learning

Relations and Knowledge Graphs

- Example relation:

Patient	Test	Technician	Result	DateTime
54326	353	99807	1	202601220945
54326	353	87601	0	202601250830
...				

Relations and Knowledge Graphs

- Example relation:

Patient	Test	Technician	Result	DateTime
54326	353	99807	1	202601220945
54326	353	87601	0	202601250830
...				

- Relational datasets typically contain lots of identifiers: typically arbitrary integers

Relations and Knowledge Graphs

- Example relation:

Patient	Test	Technician	Result	DateTime
54326	353	99807	1	202601220945
54326	353	87601	0	202601250830

...

- Relational datasets typically contain lots of identifiers: typically arbitrary integers
- (subject, verb, object) triples $\longrightarrow$ knowledge graphs.

Relations and Knowledge Graphs

- Example relation:

Patient	Test	Technician	Result	DateTime
54326	353	99807	1	202601220945
54326	353	87601	0	202601250830
...				

- Relational datasets typically contain lots of identifiers: typically arbitrary integers
- (subject, verb, object) triples $\longrightarrow$ knowledge graphs.
- Given a table: use (row, column, value) where
  row is either a primary key or
  create an identifier for the tuple – reify it

Relations and Knowledge Graphs

- Example relation:

Patient	Test	Technician	Result	DateTime
54326	353	99807	1	202601220945
54326	353	87601	0	202601250830
...				

- Relational datasets typically contain lots of identifiers: typically arbitrary integers
- (subject, verb, object) triples $\longrightarrow$ knowledge graphs.
- Given a table: use (row, column, value) where
  row is either a primary key or
  create an identifier for the tuple – reify it
- E.g, (54326353202601220945, patient, 54326)
  (54326353202601220945, result, 1)

Relations and Knowledge Graphs

- Example relation:

Patient	Test	Technician	Result	DateTime
54326	353	99807	1	202601220945
54326	353	87601	0	202601250830
...				

- Relational datasets typically contain lots of identifiers: typically arbitrary integers
- (subject, verb, object) triples $\longrightarrow$ knowledge graphs.
- Given a table: use (row, column, value) where row is either a primary key or create an identifier for the tuple – reify it
- E.g, (54326353202601220945, patient, 54326) (54326353202601220945, result, 1)
- Number of triples for reified tuple = number of columns

Relations and Knowledge Graphs

- Example relation:

Patient	Test	Technician	Result	DateTime
54326	353	99807	1	202601220945
54326	353	87601	0	202601250830
...				

- Relational datasets typically contain lots of identifiers: typically arbitrary integers
- (subject, verb, object) triples $\longrightarrow$ knowledge graphs.
- Given a table: use (row, column, value) where row is either a primary key or create an identifier for the tuple – reify it
- E.g, (54326353202601220945, patient, 54326) (54326353202601220945, result, 1)
- Number of triples for reified tuple = number of columns
- In Wikidata over 98% of entities are reified with 10 or fewer triples.

From Knowledge Graphs to Random Variables

Reconciling probabilistic models and knowledge graphs:

From Knowledge Graphs to Random Variables

Reconciling probabilistic models and knowledge graphs:

- functional verbs (properties): a random variable for each subject-verb pair.

From Knowledge Graphs to Random Variables

Reconciling probabilistic models and knowledge graphs:

- functional verbs (properties): a random variable for each subject-verb pair.
  e.g.: *birth year*,

From Knowledge Graphs to Random Variables

Reconciling probabilistic models and knowledge graphs:

- functional verbs (properties): a random variable for each subject-verb pair.
  e.g.: *birth year*, *birth mother*

From Knowledge Graphs to Random Variables

Reconciling probabilistic models and knowledge graphs:

- functional verbs (properties): a random variable for each subject-verb pair.
  e.g.: *birth year*, *birth mother*
- non-functional verbs: Boolean random variable for each subject–verb-object triple.

From Knowledge Graphs to Random Variables

Reconciling probabilistic models and knowledge graphs:

- functional verbs (properties): a random variable for each subject-verb pair.
  e.g.: *birth year*, *birth mother*
- non-functional verbs: Boolean random variable for each subject–verb-object triple.
  e.g.: *has-streamed*

From Knowledge Graphs to Random Variables

Reconciling probabilistic models and knowledge graphs:

- functional verbs (properties): a random variable for each subject-verb pair.
  e.g.: *birth year*, *birth mother*
- non-functional verbs: Boolean random variable for each subject–verb-object triple.
  e.g.: *has-streamed*
- $\longrightarrow$ relational probabilistic models represent how these random variables interrelate,
  independently of the actual entities: "logic variables", "universal quantification", "lifted", "exchangeable", "parameter sharing", "weight tying", "convolutional"

Determining Truth

Determining Truth

- How to determine truth should not be a mystery; people have been doing it for millennia.

Determining Truth

- How to determine truth should not be a mystery; people have been doing it for millennia.
- Determining truth is the subject of science and criminal law.

Determining Truth

- How to determine truth should not be a mystery; people have been doing it for millennia.
- Determining truth is the subject of science and criminal law.
  - Consider *all* relevant evidence
  - Build hypotheses (models) that explain the evidence. Hypotheses should be diverse.
  - Critically evaluate all hypotheses: be skeptical.

Determining Truth

- How to determine truth should not be a mystery; people have been doing it for millennia.
- Determining truth is the subject of science and criminal law.
    - Consider *all* relevant evidence
    - Build hypotheses (models) that explain the evidence. Hypotheses should be diverse.
    - Critically evaluate all hypotheses: be skeptical.
    - With enough independent evidence, the truth tends to pop out: truth fits the evidence, whereas other hypotheses won't quite.

Determining Truth

- How to determine truth should not be a mystery; people have been doing it for millennia.
- Determining truth is the subject of science and criminal law.
  - Consider *all* relevant evidence
  - Build hypotheses (models) that explain the evidence. Hypotheses should be diverse.
  - Critically evaluate all hypotheses: be skeptical.
  - With enough independent evidence, the truth tends to pop out: truth fits the evidence, whereas other hypotheses won't quite. Much is settled: relativity, molecules, evolution, vaccines...

Determining Truth

- How to determine truth should not be a mystery; people have been doing it for millennia.
- Determining truth is the subject of science and criminal law.
  - Consider *all* relevant evidence
  - Build hypotheses (models) that explain the evidence. Hypotheses should be diverse.
  - Critically evaluate all hypotheses: be skeptical.
  - With enough independent evidence, the truth tends to pop out: truth fits the evidence, whereas other hypotheses won't quite. Much is settled: relativity, molecules, evolution, vaccines. . .
- The law must make a binary guilty/not guilty decision.
- Scientific hypotheses are always open to revision.

Determining Truth

- How to determine truth should not be a mystery; people have been doing it for millennia.
- Determining truth is the subject of science and criminal law.
  - Consider *all* relevant evidence
  - Build hypotheses (models) that explain the evidence. Hypotheses should be diverse.
  - Critically evaluate all hypotheses: be skeptical.
  - With enough independent evidence, the truth tends to pop out: truth fits the evidence, whereas other hypotheses won't quite. Much is settled: relativity, molecules, evolution, vaccines...
- The law must make a binary guilty/not guilty decision.
- Scientific hypotheses are always open to revision. Probability is the logic of science (e.g. Jaynes, 1995)

Determining Truth

- How to determine truth should not be a mystery; people have been doing it for millennia.
- Determining truth is the subject of science and criminal law.
  - Consider *all* relevant evidence
  - Build hypotheses (models) that explain the evidence. Hypotheses should be diverse.
  - Critically evaluate all hypotheses: be skeptical.
  - With enough independent evidence, the truth tends to pop out: truth fits the evidence, whereas other hypotheses won't quite. Much is settled: relativity, molecules, evolution, vaccines...
- The law must make a binary guilty/not guilty decision.
- Scientific hypotheses are always open to revision. Probability is the logic of science (e.g. Jaynes, 1995)
- Science requires physical experiments – acting on world – to determine how the world works.

Determining Truth

- How to determine truth should not be a mystery; people have been doing it for millennia.
- Determining truth is the subject of science and criminal law.
  - Consider *all* relevant evidence
  - Build hypotheses (models) that explain the evidence. Hypotheses should be diverse.
  - Critically evaluate all hypotheses: be skeptical.
  - With enough independent evidence, the truth tends to pop out: truth fits the evidence, whereas other hypotheses won't quite. Much is settled: relativity, molecules, evolution, vaccines...
- The law must make a binary guilty/not guilty decision.
- Scientific hypotheses are always open to revision. Probability is the logic of science (e.g. Jaynes, 1995)
- Science requires physical experiments – acting on world – to determine how the world works.
- Keep track of the provenance of all data.

Outline

Agents

What should an agent believe?

What should an agent do?

Where do values/goals/preferences come from?

Preferences

Alice . . . went on "Would you please tell me, please, which way I ought to go from here?"
"That depends a good deal on where you want to get to," said the Cat.
"I don't much care where —" said Alice.
"Then it doesn't matter which way you go," said the Cat.

Alice's Adventures in Wonderland, Lewis Carroll, 1865

Utility [von Neumann and Morgenstern, 1944]

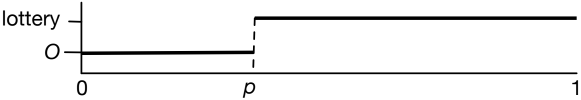
- Actions result in outcomes. Agents have preferences over outcomes.

Utility [von Neumann and Morgenstern, 1944]

- Actions result in outcomes. Agents have preferences over outcomes.
- Would agent prefer outcome O or lottery: best outcome with probability p and worst outcome with probability $1 - p$?

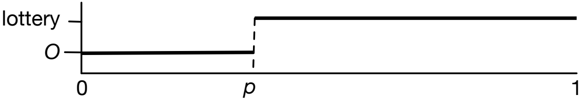
Utility [von Neumann and Morgenstern, 1944]

- Actions result in outcomes. Agents have preferences over outcomes.
- Would agent prefer outcome O or lottery: best outcome with probability p and worst outcome with probability $1 - p$?



Utility [von Neumann and Morgenstern, 1944]

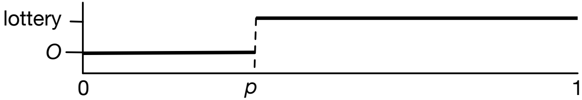
- Actions result in outcomes. Agents have preferences over outcomes.
- Would agent prefer outcome O or lottery: best outcome with probability p and worst outcome with probability $1 - p$?



- The p where preference flips is the utility of the outcome O.

Utility [von Neumann and Morgenstern, 1944]

- Actions result in outcomes. Agents have preferences over outcomes.
- Would agent prefer outcome O or lottery: best outcome with probability p and worst outcome with probability $1 - p$?



- The p where preference flips is the utility of the outcome O.
- A few intuitive(?) assumptions $\Rightarrow$ prefer outcome iff higher expected utility

Utility and time

- How would you compare the following sequences of rewards (per week):

  A: \$1,000,000, \$0, \$0, \$0, \$0, \$0,...
  B: \$1000, \$1000, \$1000, \$1000, \$1000,...

Rewards and Values

Suppose the agent receives a sequence of rewards $r_1, r_2, r_3, r_4, \ldots$ in time. What utility (return or value) should be assigned?

Rewards and Values

Suppose the agent receives a sequence of rewards $r_1, r_2, r_3, r_4, \ldots$ in time. What utility (return or value) should be assigned?

- total reward $V = \sum_{i=1}^{\infty} r_i$
- average reward $V = \lim_{n \to \infty} (r_1 + \cdots + r_n)/n$

Rewards and Values

Suppose the agent receives a sequence of rewards $r_1, r_2, r_3, r_4, \ldots$ in time. What utility (return or value) should be assigned?

- total reward $V = \sum_{i=1}^{\infty} r_i$
- average reward $V = \lim_{n\to\infty} (r_1 + \cdots + r_n)/n$
- discounted return, for discount γ:

$$V = r_1 + \gamma r_2 + \gamma^2 r_3 + \gamma^3 r_4 + \cdots$$

Rewards and Values

Suppose the agent receives a sequence of rewards $r_1, r_2, r_3, r_4, \ldots$ in time. What utility (return or value) should be assigned?

- total reward $V = \sum_{i=1}^{\infty} r_i$
- average reward $V = \lim_{n \to \infty} (r_1 + \cdots + r_n)/n$
- discounted return, for discount γ:

$$V = r_1 + \gamma r_2 + \gamma^2 r_3 + \gamma^3 r_4 + \cdots$$

- Let V_t be the discounted return from time t:

$$V_t = r_t + \gamma V_{t+1}$$

Rewards and Values

Suppose the agent receives a sequence of rewards $r_1, r_2, r_3, r_4, \ldots$ in time. What utility (return or value) should be assigned?

- total reward $V = \sum_{i=1}^{\infty} r_i$
- average reward $V = \lim_{n \to \infty} (r_1 + \cdots + r_n)/n$
- discounted return, for discount γ:

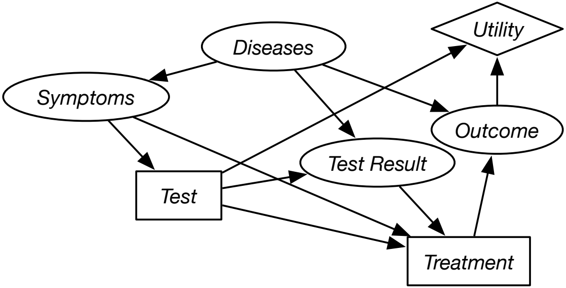
$$V = r_1 + \gamma r_2 + \gamma^2 r_3 + \gamma^3 r_4 + \cdots$$

- Let V_t be the discounted return from time t:

$$V_t = r_t + \gamma V_{t+1}$$

- Discount γ is not a hyperparameter to be optimized, but specifies how much future rewards are worth.

Decision Network



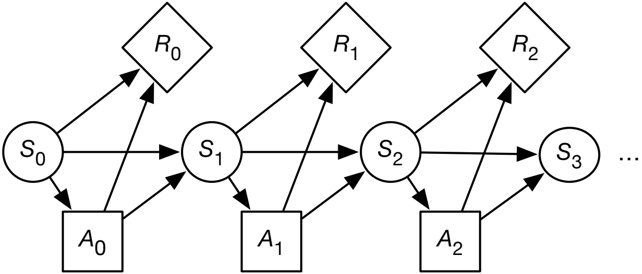
- Ellipses are random variables
- Rectangles are decisions
- Diamond is utility
- Arc means "can depend on"

Decision Processes

- For artificial domains, such as games and factory floors, a agent can observe state of environment: fully observable

Decision Processes

- For artificial domains, such as games and factory floors, a agent can observe state of environment: fully observable
- A fully-observable Markov decision process can go on indefinitely:



Decision Processes (cont.)

- What to do can be described in terms of:

Decision Processes (cont.)

- What to do can be described in terms of:
  - $Q(s, a)$ expected value of doing action a in state s
  - value: $V(s) = max_a Q(s, a)$
  - policy: $\pi(s) = \arg\max_a Q(s, a)$

Decision Processes (cont.)

- What to do can be described in terms of:
  - $Q(s, a)$ expected value of doing action a in state s
  - value: $V(s) = max_a Q(s, a)$
  - policy: $\pi(s) = \arg\max_a Q(s, a)$
- Can be computed using
  - learning with abundant experiential data or a model to generate data: "reinforcement learning"

Decision Processes (cont.)

- What to do can be described in terms of:
  - $Q(s, a)$ expected value of doing action a in state s
  - value: $V(s) = max_a Q(s, a)$
  - policy: $\pi(s) = \arg\max_a Q(s, a)$
- Can be computed using
  - learning with abundant experiential data or a model to generate data: "reinforcement learning"
  - planning – search or dynamic programming – requires a model: "decision-theoretic planning"

Decision Processes (cont.)

- What to do can be described in terms of:
  - $Q(s, a)$ expected value of doing action a in state s
  - value: $V(s) = max_a Q(s, a)$
  - policy: $\pi(s) = \arg\max_a Q(s, a)$
- Can be computed using
  - learning with abundant experiential data or a model to generate data: "reinforcement learning"
  - planning – search or dynamic programming – requires a model: "decision-theoretic planning"
  - or both – e.g.: alphaGo/alphaZero learn V, π and play multiple games for each position.

Decision Processes (cont.)

- What to do can be described in terms of:
  - $Q(s, a)$ expected value of doing action a in state s
  - value: $V(s) = max_a Q(s, a)$
  - policy: $\pi(s) = \arg\max_a Q(s, a)$
- Can be computed using
  - learning with abundant experiential data or a model to generate data: "reinforcement learning"
  - planning – search or dynamic programming – requires a model: "decision-theoretic planning"
  - or both – e.g.: alphaGo/alphaZero learn V, π and play multiple games for each position.
- Q, V, π works for partially observable domains:

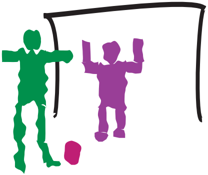
Decision Processes (cont.)

- What to do can be described in terms of:
  - $Q(s, a)$ expected value of doing action a in state s
  - value: $V(s) = max_a Q(s, a)$
  - policy: $\pi(s) = \arg\max_a Q(s, a)$
- Can be computed using
  - learning with abundant experiential data or a model to generate data: "reinforcement learning"
  - planning – search or dynamic programming – requires a model: "decision-theoretic planning"
  - or both – e.g.: alphaGo/alphaZero learn V, π and play multiple games for each position.
- Q, V, π works for partially observable domains: use history instead of state

Decision Processes (cont.)

- What to do can be described in terms of:
  - $Q(s,a)$ expected value of doing action a in state s
  - value: $V(s) = max_a Q(s,a)$
  - policy: $\pi(s) = \arg\max_a Q(s,a)$
- Can be computed using
  - learning with abundant experiential data or a model to generate data: "reinforcement learning"
  - planning – search or dynamic programming – requires a model: "decision-theoretic planning"
  - or both – e.g.: alphaGo/alphaZero learn V, π and play multiple games for each position.
- Q, V, π works for partially observable domains:
  use history instead of state
  if the agent observes and remembers relevant history
  (no-forgetting)

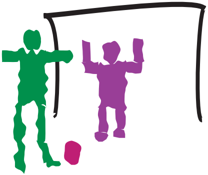
Multiple Agent Example: Football Penalty Kick



		goalkeeper left	goalkeeper right
kicker	left	0.6	0.2
	right	0.3	0.9

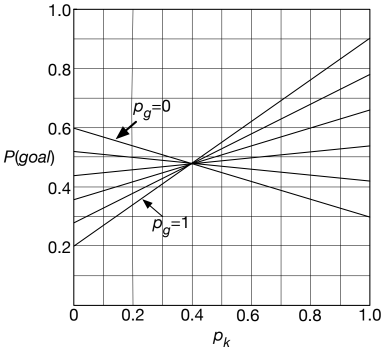
Probability of a goal.

Multiple Agent Example: Football Penalty Kick



		goalkeeper	
		left	right
kicker	left	0.6	0.2
	right	0.3	0.9

Probability of a goal.



p_k is $P(kicker = right)$
p_j is $P(goalkeeper = right)$

Multiagent Example: Tragedy of the Commons

Example:

- There are 100 agents.
- There is an common environment that is shared amongst all agents. Each agent has $1/100$ of the shared environment.

Multiagent Example: Tragedy of the Commons

Example:

- There are 100 agents.
- There is an common environment that is shared amongst all agents. Each agent has $1/100$ of the shared environment.
- Each agent can choose to do an action that has an individual payoff of $+10$ but has a shared -100 payoff on the environment
  or do nothing with a zero payoff

Multiagent Example: Tragedy of the Commons

Example:

- There are 100 agents.
- There is an common environment that is shared amongst all agents. Each agent has $1/100$ of the shared environment.
- Each agent can choose to do an action that has an individual payoff of $+10$ but has a shared -100 payoff on the environment
  or do nothing with a zero payoff
- For each agent, doing the action has a payoff of

Multiagent Example: Tragedy of the Commons

Example:

- There are 100 agents.
- There is an common environment that is shared amongst all agents. Each agent has $1/100$ of the shared environment.
- Each agent can choose to do an action that has an individual payoff of $+10$ but has a shared -100 payoff on the environment
  or do nothing with a zero payoff
- For each agent, doing the action has a payoff of $10 - 100/100 = 9$

Multiagent Example: Tragedy of the Commons

Example:

- There are 100 agents.
- There is an common environment that is shared amongst all agents. Each agent has $1/100$ of the shared environment.
- Each agent can choose to do an action that has an individual payoff of $+10$ but has a shared -100 payoff on the environment
  or do nothing with a zero payoff
- For each agent, doing the action has a payoff of $10 - 100/100 = 9$
- If every agent does the action the total payoff is

Multiagent Example: Tragedy of the Commons

Example:

- There are 100 agents.
- There is an common environment that is shared amongst all agents. Each agent has $1/100$ of the shared environment.
- Each agent can choose to do an action that has an individual payoff of $+10$ but has a shared -100 payoff on the environment
  or do nothing with a zero payoff
- For each agent, doing the action has a payoff of $10 - 100/100 = 9$
- If every agent does the action the total payoff is $1000 - 10000 = -9000$

Multiagent Example: Tragedy of the Commons

Example:

- There are 100 agents.
- There is an common environment that is shared amongst all agents. Each agent has $1/100$ of the shared environment.
- Each agent can choose to do an action that has an individual payoff of $+10$ but has a shared -100 payoff on the environment
  or do nothing with a zero payoff
- For each agent, doing the action has a payoff of $10 - 100/100 = 9$
- If every agent does the action the total payoff is $1000 - 10000 = -9000$

Mechanism design: design the rules. E.g., maximize common good.

Outline

Agents

What should an agent believe?

What should an agent do?

Where do values/goals/preferences come from?

Where do preferences come from?

Two cases:

- The only impacts of actions are on consenting participants. E.g., selecting meals, selecting treatments, playing a game

Where do preferences come from?

Two cases:

- The only impacts of actions are on consenting participants.
  E.g., selecting meals, selecting treatments, playing a game
- Non-consenting people and/or the environment may be impacted.
  E.g., driving, building sewage treatment centre

Determining user preferences

- Be suspicious of someone (or thing) that tells you what to do, without asking what you want.

Determining user preferences

- Be suspicious of someone (or thing) that tells you what to do, without asking what you want.
- Many apps have very limited alternatives to choose from.

Determining user preferences

- Be suspicious of someone (or thing) that tells you what to do, without asking what you want.
- Many apps have very limited alternatives to choose from.
  Example: Google maps lets you choose
  - the start and destination,
  - mode of transport: walk, car, transit, bicycle, fly
  - a few route options; e.g., for driving: avoid highways, tolls, and/or ferries.

Determining user preferences

- Be suspicious of someone (or thing) that tells you what to do, without asking what you want.
- Many apps have very limited alternatives to choose from.
  Example: Google maps lets you choose
  - the start and destination,
  - mode of transport: walk, car, transit, bicycle, fly
  - a few route options; e.g., for driving: avoid highways, tolls, and/or ferries.

  and returns a few options that are estimated to minimize time.

Determining user preferences

- Be suspicious of someone (or thing) that tells you what to do, without asking what you want.
- Many apps have very limited alternatives to choose from.
  Example: Google maps lets you choose
  - the start and destination,
  - mode of transport: walk, car, transit, bicycle, fly
  - a few route options; e.g., for driving: avoid highways, tolls, and/or ferries.

  and returns a few options that are estimated to minimize time.
- It doesn't let you
  - avoid going by your mother's home
  - go via the most scenic route
  - take the train with a 20-30 minute walk through somewhere interesting

Determining user preferences

- Be suspicious of someone (or thing) that tells you what to do, without asking what you want.
- Many apps have very limited alternatives to choose from.
  Example: Google maps lets you choose
  - the start and destination,
  - mode of transport: walk, car, transit, bicycle, fly
  - a few route options; e.g., for driving: avoid highways, tolls, and/or ferries.

  and returns a few options that are estimated to minimize time.
- It doesn't let you
  - avoid going by your mother's home
  - go via the most scenic route
  - take the train with a 20-30 minute walk through somewhere interesting
- LLM-based systems allow users to say anything, but users don't know what to say.

What happens when people are disproportionally impacted

- Some actions affect people disproportionally.
- Example: where to place a sewage treatment plant and how much to pay for smell reduction?

What happens when people are disproportionally impacted

- Some actions affect people disproportionally.
- Example: where to place a sewage treatment plant and how much to pay for smell reduction?
- People near the plant want the least smelly system.
- Taxpayers away from the plant want the cheapest option.

What happens when people are disproportionally impacted

- Some actions affect people disproportionally.
- Example: where to place a sewage treatment plant and how much to pay for smell reduction?
- People near the plant want the least smelly system.
- Taxpayers away from the plant want the cheapest option.
- The majority want the cheapest option.

What happens when people are disproportionally impacted

- Some actions affect people disproportionally.
- Example: where to place a sewage treatment plant and how much to pay for smell reduction?
- People near the plant want the least smelly system.
- Taxpayers away from the plant want the cheapest option.
- The majority want the cheapest option.
- Sometimes ignorance is bliss: better to choose the smell reduction without knowing the location. Everyone is gambling.

What happens when people are disproportionally impacted

- Some actions affect people disproportionally.
- Example: where to place a sewage treatment plant and how much to pay for smell reduction?
- People near the plant want the least smelly system.
- Taxpayers away from the plant want the cheapest option.
- The majority want the cheapest option.
- Sometimes ignorance is bliss: better to choose the smell reduction without knowing the location. Everyone is gambling. (But people at top of hill know it won't be near them.)

What happens when people are disproportionally impacted

- Some actions affect people disproportionally.
- Example: where to place a sewage treatment plant and how much to pay for smell reduction?
- People near the plant want the least smelly system.
- Taxpayers away from the plant want the cheapest option.
- The majority want the cheapest option.
- Sometimes ignorance is bliss: better to choose the smell reduction without knowing the location. Everyone is gambling. (But people at top of hill know it won't be near them.)
- One solution: small focus groups informed about the issues discuss options.

What happens when people are disproportionally impacted

- Some actions affect people disproportionally.
- Example: where to place a sewage treatment plant and how much to pay for smell reduction?
- People near the plant want the least smelly system.
- Taxpayers away from the plant want the cheapest option.
- The majority want the cheapest option.
- Sometimes ignorance is bliss: better to choose the smell reduction without knowing the location. Everyone is gambling. (But people at top of hill know it won't be near them.)
- One solution: small focus groups informed about the issues discuss options.
- How to make it fair when the people acting and people impacted are in different socioeconomic groups or different countries or not-yet-born?

Moral Decisions

". . . self-driving cars . . . what are the values that we're going to embed in the cars? There are gonna be a bunch of choices that you have to make, the classic problem being: If the car is driving, you can swerve to avoid hitting a pedestrian, but then you might hit a wall and kill yourself. It's a moral decision, and who's setting up those rules?"

Barack Obama, 2016

Conclusions

- Agents are judged by their actions.

Conclusions

- Agents are judged by their actions.
- Acting under uncertainty is gambling. Probability is the calculus of gambling.

Conclusions

- Agents are judged by their actions.
- Acting under uncertainty is gambling. Probability is the calculus of gambling.
- It is impossible to learn the effects of actions from observations alone: you need a causal model.
  An agent needs to act in the world with a real body.

Conclusions

- Agents are judged by their actions.
- Acting under uncertainty is gambling. Probability is the calculus of gambling.
- It is impossible to learn the effects of actions from observations alone: you need a causal model.
  An agent needs to act in the world with a real body.
- Determining truth is the realm of science. Use all available evidence. Have diverse hypotheses. Carry out experiments.
  Always be open to new evidence and hypotheses. Be skeptical.

Conclusions

- Agents are judged by their actions.
- Acting under uncertainty is gambling. Probability is the calculus of gambling.
- It is impossible to learn the effects of actions from observations alone: you need a causal model.
  An agent needs to act in the world with a real body.
- Determining truth is the realm of science. Use all available evidence. Have diverse hypotheses. Carry out experiments. Always be open to new evidence and hypotheses. Be skeptical.
- What an agent should do depends on its preferences and others affected by its actions.

Conclusions

- Agents are judged by their actions.
- Acting under uncertainty is gambling. Probability is the calculus of gambling.
- It is impossible to learn the effects of actions from observations alone: you need a causal model.
  An agent needs to act in the world with a real body.
- Determining truth is the realm of science. Use all available evidence. Have diverse hypotheses. Carry out experiments. Always be open to new evidence and hypotheses. Be skeptical.
- What an agent should do depends on its preferences and others affected by its actions.
- Determining which values to use, and how preferences can be established fairly, is a major non-technical challenge.

Conclusions

- Agents are judged by their actions.
- Acting under uncertainty is gambling. Probability is the calculus of gambling.
- It is impossible to learn the effects of actions from observations alone: you need a causal model. An agent needs to act in the world with a real body.
- Determining truth is the realm of science. Use all available evidence. Have diverse hypotheses. Carry out experiments. Always be open to new evidence and hypotheses. Be skeptical.
- What an agent should do depends on its preferences and others affected by its actions.
- Determining which values to use, and how preferences can be established fairly, is a major non-technical challenge.
- The essence of (artificial) intelligence is action.