# Introduction to Graphical Models
## lecture 11 - planning by inference

Marc Toussaint
TU Berlin

# Motivation

- a humble question: *how does thinking work?*

# Motivation

- a humble question: *how does thinking work?*

- psychology/philosophy:
  Rick Grush (Behavioral and Brain Sciences, 2004):
  *The emulation theory of representation: motor control, imagery, and perception.*
  (20 pages + 46 pages commentary & response!)

  keywords: *Kalman filters, overt & covert actions, imagery*

# Motivation

- cognitive sciences:
  G. Hesslow (Trends in Cog Sciences, 2002):
  *Conscious thought as simulation of behaviour and perception.*

> A 'simulation' theory of cognitive function can be based on three assumptions about brain function.
> (1) First, behaviour can be simulated by activating motor structures, as during an overt action but suppressing its execution.
> (2) Second, perception can be simulated by internal activation of sensory cortex, as during normal perception of external stimuli.
> (3) Third, both overt and covert actions can elicit perceptual simulation of their normal consequences.
> A large body of evidence supports these assumptions. It is argued that the simulation approach can explain the relations between motor, sensory and cognitive functions and the appearance of an inner world.

# Motivation

- neuroscience:

  Johnson & Redish (J o Neuroscience, 2007): *Neural ensembles in CA3 transiently encode paths forward of the animal at a decision point*

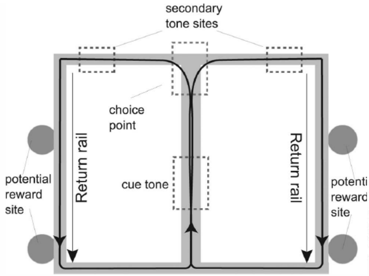
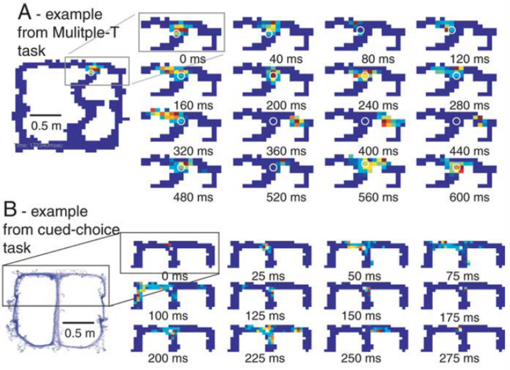



**Figure 2.** The cued-choice maze. The task consists of a single T turn with food rewa
available at two sites on each return rail. On each lap, a cue tone on the center stem (red dash
box) signaled whether the left or right arm would be rewarded. If and only if the rat made t
correct turn, a matching tone would play at the T arm (blue dashed boxes).



## Motivation

- bottom line (as I see it...):

  – people have this idea of cognition/thinking/planning as internal simulation

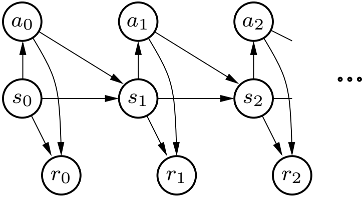
  – we need a proper math for this!

## Outline

- Markov Decision Processes as (simplest) formal framework
- Expectation Maximization to learn optimal behavior parameters (=policy)

# Markov Decision Process I

- Markov process on the random variables of states $x_t$, actions $a_t$, and rewards $r_t$, defined by graphical model




$$P(s_{0:T}, a_{0:T}, r_{0:T}; \pi) =$$
$$P(s_0) P(a_0|s_0;\pi) P(r_0|a_0,s_0) \ \prod_{t=1}^T P(s_t|a_{t\text{-}1}, s_{t\text{-}1}) P(a_t|s_t;\pi) P(r_t|a_t,s_t)$$

- the *world* defines: (stationarity: no explicit dependency on time)

| | |
|---|---|
| $P(s_0)$ | initial state distribution |
| $P(s_{t+1} \mid a_t, s_t)$ | transition probabilties |
| $P(r_t \mid a_t, s_t)$ | reward probabilities |

- the *agent* defines: ($\pi$ is a *parameter* of the model)

$P(a_t = a \mid s_t = x; \pi) \equiv \pi_{ax}$ action probabilities (policy)

## Markov Decision Process II

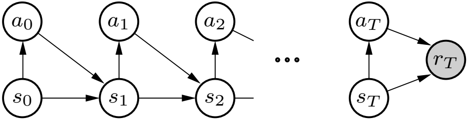
- what's the objective?
  – collect as much reward as possible    (in *expectation!*)
- expected discounted future return of a policy $\boldsymbol{\pi}$

$$V^{\boldsymbol{\pi}} = \mathrm{E}\{\sum_{t=0}^{\infty} \gamma^t \; r_t; \pi\}$$

with discount factor  $\gamma \in [0, 1]$

# EM for planning I

- address a simplified case:
  - we care only for the reward $r_T$ at finite time $T$
  - we assume binary rewards: dom$(r_T) = \{0, 1\}$




$\rightarrow$ optimize the model parameters $\pi$ to maximize the likelihood of observing reward $r_T = 1$!!:

- note: much more latent than observed:
  observed variables ("data"): $r_T = 1$
  latent (unobserved) variables: $s_{0:T}, a_{0:T}$

# EM for planning II

- "observed data likelihood" (last lecture)

$$\exp \hat{L}(\pi) = P(r_T = 1; \pi)$$
$$= \sum_{a_{0:T}, s_{0:T}} P(r_T = 1, a_{0:T}, s_{0:T}; \pi) = \mathrm{E}\{r_T; \pi\}$$

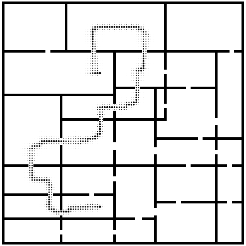
- doing the summation exactly is intractable $\rightarrow$
- EM algorithm
  - E-step: *compute posterior over* $a_{0:T}, s_{0:T}$ *conditioned on "data"* $r_T = 1$

$$P(a_{0:T}, s_{0:T} \mid r_T = 1; \pi^{\text{old}})$$

  - M-step: assign policy to new optimum of expected data log-likelihood

$$\pi^{\text{new}} = \underset{\pi}{\operatorname{argmax}}\, Q(\pi, \pi^{\text{old}})$$

# Example: MDP maze



- states: all locations
- actions: up, down, left, right
- transition probabilities:
  - $s \in$ wall: completely stuck
  - $s \notin$ wall: 90% make correct step, 10% make random step

  $\rightarrow$ keep away from walls!

## Interpretation of the E-step

- the E-step is really interesting:
  *compute the posterior over states and actions conditioned on "data"*
  $r_T = 1$

- we do as if $r_T = 1$ was data – although we (the agent) hasn't observed this data (yet)
  but we *imagine* we will observe it in the future

- internal simulation & mental imagery: we imagine to observe the event $r_T = 1$, and we "internally simulate" (compute the posterior over) the trajectory $a_{0:T}, s_{0:T}$ to get there

## More general case

- what if the true rewards are not binary?
  – can rescale true rewards $r_t$ such that $\mathrm{E}\{r_t \,|\, a_t, s_t\} \propto P(\hat{r}_t = 1 \,|\, a_t, s_t)$

- what if we care about all rewards $V^{\boldsymbol{\pi}} = \mathrm{E}\{\sum_{t=0}^{\infty} \gamma^t \, r_t; \pi\}$
  – we can introduce a "mixture model"

# Mixture models (interlude)

- mixture models are a special case of graphical models
- assume we have random variables $X_{1:N}$
  - the distribution over $X_{1:N}$ depends on another random variable $Y$

$$P(X_{1:N}|Y) = \begin{cases} p_1(X_{1:N}) & Y = 1 \\ p_2(X_{1:N}) & Y = 2 \\ \vdots & \end{cases}$$

  - consequently, the marginal over $X_{1:N}$ is a "mixture" of $p_y$'s:

$$P(X_{1:N}) = \sum_Y P(Y)\, P(X_{1:N}|Y) = \sum_y P(y)\, p_y(X_{1:N})$$

# Mixture of finite-time MDPs

- so far we assumed fix final time $T$ and addressed the distribution

  $P(r_T\!=\!1, a_{0:T}, s_{0:T}; \pi)$

  – now assume we are uncertain what $T$, we only have a prior $P(T)$

  $P(r_T\!=\!1, a_{0:T}, s_{0:T}, T; \pi) = P(T)\ P(r_T\!=\!1, a_{0:T}, s_{0:T} | T; \pi)$

- if we choose $P(T)$ geometric, $P(T) = (1-\gamma)\ \gamma^T$,

$$\begin{aligned}\exp \hat L(\pi) &= P(r_T\!=\!1; \pi)\\ &= \sum_{a_{0:T}, s_{0:T}, T} P(T)\ P(r_T\!=\!1, a_{0:T}, s_{0:T} | T; \pi)\\ &= \sum_T P(T)\ \mathrm{E}\{r_T; \pi\} = (1-\gamma) \sum_T \gamma^T\ \mathrm{E}\{r_T; \pi\} = (1-\gamma) V^{\boldsymbol{\pi}}\end{aligned}$$

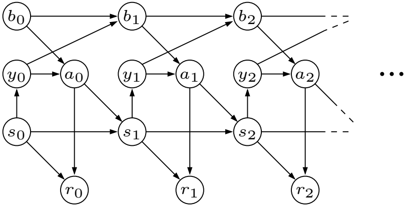
$\Rightarrow$ maximization of likelihood $r_T\!=\!1$

$\iff$ maximization of expected discounted future return

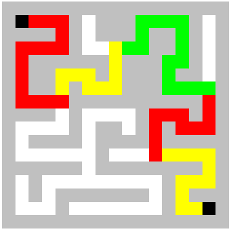
$\Rightarrow$ we can compute optimal (in the traditional definition) policies using Expectation Maximization in $P(r_T\!=\!1, a_{0:T}, s_{0:T}, T; \pi)$

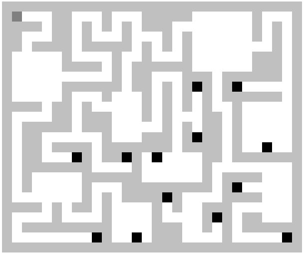
# Example: POMDP maze

- POMDP = Partially Observable Markov Decision Process
  – in POMDPs the agent needs some kind of memory



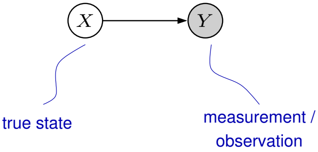

- mazes: T-junctions, halls & corridors (379 locations, 1516 states)





# discussion: inference for sensor processing

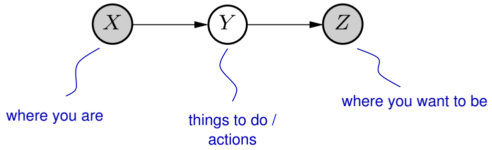
$$P(X,Y) = P(Y|X)\ P(X)$$




- inference $:=$ compute $P(X|Y)$
- examples:
  - HMMs (speech, discrete processes, ...)
  - image processing (denoising, super-resolution, segmentation, ...)
  - Kalman filters
  - etc

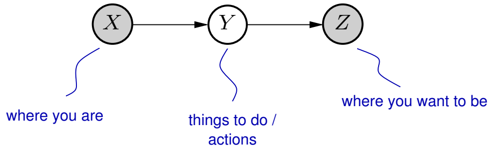
# discussion: inference for action planning

probabilistic inference for planning / control / decision making



# discussion: inference for action planning

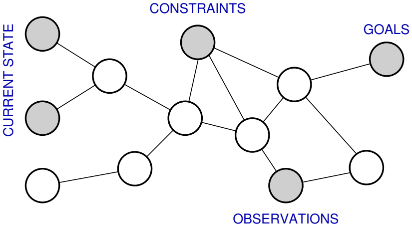
probabilistic inference for planning / control / decision making




- *rephrase problem of planning as problem of inference!*
- infer actions to reach a goal
- link to “internal simulation” (cog. science, Matt Botvinick)

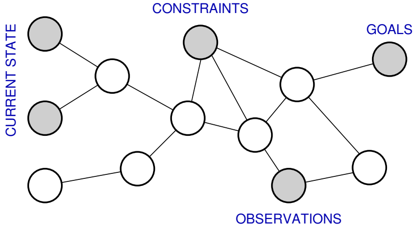
# discussion: inference for action planning

- works for arbitrary *networks* of goals, constraints, observations, etc.



# discussion: inference for action planning

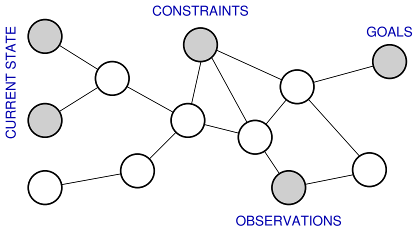
- works for arbitrary *networks* of goals, constraints, observations, etc.




→ planning on distributed representations!
  – on mixed discrete/continuous representations
  – contrasts classical notion of *state* as one big variable
    (value functions, spreading activation, RRTs, configuration space)
  – we know how to exploit structure with inference!   (ML methods)

# discussion: inference for action planning

- works for arbitrary *networks* of goals, constraints, observations, etc.




→ planning on distributed representations!
  - on mixed discrete/continuous representations
  - contrasts classical notion of *state* as one big variable
    (value functions, spreading activation, RRTs, configuration space)
  - we know how to exploit structure with inference!   (ML methods)

→ no distinction between sensor and motor, perception and action!

- next time:
  - summary & open problems