

Simulation d'évènements rares: application à l'optimisation d'une politique de contrôle dans un cadre partiellement observé

Frédéric Dambreville

general@FredericDambreville.com

Délégation Générale pour l'Armement, DGA/CEP/GIP/SRO
16 Bis, Avenue Prieur de la Côte d'Or
F94114 Arcueil, France

Introduction

Activité de l'équipe GIP. Expertise et recherche

- Traitement de l'image
- Traitement de la parole
- Fusion de donnée; pistage, théorie de l'évidence, Bayésien, méthode ensembliste, logique
- Planification de capteurs, contrôle; apprentissage, simulation d'évènements rares, théorie des jeux
- Robotique

Introduction; simulation d'évènements rares

Evaluation d'évènements à faible ou très faible probabilité

Ex. Probabilité de panne d'un système

Grace à la simulation, comment paramétrer un système pour réduire sa probabilité de panne

Optimisation de fonction

Principe : l'ensemble des paramètres optimaux d'une fonction constitue un évènement rare

C'est cet aspect qui nous intéresse

Introduction; méthode de simulation

Méthode introduite par Reuven Rubinstein: méthode de simulation par cross-entropie

principe: utiliser une famille de loi instrumentale pour approcher récursivement la loi de l'évènement rare par échantillonnage préférentiel (Importance Sampling)

La méthode par cross-entropie permet de converger vers un bon paramétrage des lois

N.B. D'autres méthodes de simulation existent

Introduction; planification

Objectif: optimiser le contrôle d'un mobile dans le but de maximiser un gain

→ environnement bruité

Contexte non dynamique. Optimiser une trajectoire dans le temps et l'espace

Contexte dynamique. Le contexte observé doit être pris en compte. Il s'agit d'optimiser (approximer) un *arbre de décision*

Intérêt de la CE. Le choix de la famille de loi correspond au choix d'une sémantique du déplacement ou du contrôle



Simulation d'évènements rares

Hypothèse

- Variable $x \in X$
- Loi $p \in \mathcal{P}(X)$ caractérisant le comportement probabiliste de x
- Fonction $f : X \rightarrow \mathbb{R}$ caractérisant une performance sur x ; à seuiller

Typiquement, f peut mesurer un écart par rapport à un fonctionnement nominal

- Seuil $\gamma \in \mathbb{R}$ définissant l'évènement $E_\gamma = \{x \in X / f(x) > \gamma\}$

Typiquement, $\gamma =$ seuil délimitant le fonctionnement nominal et la situation de panne

Question: comment évaluer $p(E_\gamma)$

Monte-Carlo

Effectuer N tirages $x_n \in X$

Calculer $p(f(x) > \gamma) \simeq \frac{1}{N} \#\{n / f(x_n) > \gamma\} = \hat{P}_\gamma$

Variance: $\text{var}(\hat{P}_\gamma) = \mathbf{E}_p(\hat{P}_\gamma - P_\gamma)^2 = \frac{1}{N} P_\gamma(1 - P_\gamma)$

Variance relative: $\frac{\sqrt{\text{var}(\hat{P}_\gamma)}}{\hat{P}_\gamma} = \sqrt{\frac{1 - P_\gamma}{N P_\gamma}}$

Convergence relative très mauvaise. Inapplicable

Utiliser une loi instrumentale q pour approximer la probabilité de l'évènement rare:

$$\hat{P}_\gamma = \frac{1}{N} \sum_{n=1}^N \delta[f(x_n) \geq \gamma] \frac{p(x_n)}{q(x_n)} .$$

Le choix de q n'est pas aisé

La densité optimale $q^*(x) = \delta[f(x) \geq \gamma] p(x) P_\gamma^{-1}$

n'est pas accessible car dépend de P_γ

Simulation par cross-entropie

Principe: échantillonnage à partir d'une famille de lois

Est donnée une famille de lois $\pi(\cdot; \lambda) | \lambda \in \Lambda$

Minimiser la distance de Kullback-Leiber

$\mathcal{D}(q^*, \pi(\cdot; \lambda))$ se traduit par la maximisation:

$$\lambda_* \in \arg \max_{\lambda \in \Lambda} \mathbf{E}_p (\delta[f(x) \geq \gamma] \ln \pi(x; \lambda)) .$$

Espérance associé à un évènement rare. Nécessité un échantillonnage préférentiel:

$$\lambda_* \in \arg \max_{\lambda \in \Lambda} \mathbf{E}_q \left(\delta[f(x) \geq \gamma] \frac{p(x)}{q(x)} \ln \pi(x; \lambda) \right)$$

Approximé par échantillonnage x_1, \dots, x_N selon q

$$\hat{\lambda}_* \in \arg \max_{\lambda \in \Lambda} \sum_{n=1}^N \left(\delta[f(x_n) \geq \gamma] \frac{p(x_n)}{q(x_n)} \ln \pi(x_n; \lambda) \right)$$

Induit un point fixe; résolu itérativement par CE

Algorithme de CE

Paramètre de vitesse $\rho \in]0, 1[$

1. choisir λ_0 tel que $\pi(\cdot, \lambda_0) \simeq p$, poser $t = 0$
2. Tirer x_1, \dots, x_N selon $\pi(\cdot, \lambda_t)$, et les ordonner de manière à avoir $f(x_n) \leq f(x_{n+1})$
3. Définir $\gamma_t = f(x_{\lfloor \rho N \rfloor})$ le $(1 - \rho)$ quantile
4. Calculer: $\lambda_{t+1} \in \arg \max_{\lambda \in \Lambda} \sum_{n=1}^N \left(\delta [f(x_n) \geq \gamma_t] \frac{p(x_n)}{\pi(x_n, \lambda_t)} \ln \pi(x_n, \lambda) \right)$
5. Si $\gamma_t < \gamma$, poser $t = t + 1$ et aller en 2
Sinon, terminer par:

$$\hat{P}_\gamma = \frac{1}{N} \sum_{n=1}^N \delta [f(x_n) \geq \gamma] \frac{p(x_n)}{\pi(x_n, \lambda_{t+1})}$$

CE et optimisation

Plus de critère d'arrêt γ , plus de probabilité p de référence

1. choisir λ_0 tel que $\pi(\cdot, \lambda_0)$ soit uniforme, poser $t = 0$
2. Tirer x_1, \dots, x_N selon $\pi(\cdot, \lambda_t)$, et les ordonner de manière à avoir $f(x_n) \leq f(x_{n+1})$
3. Définir $\gamma_t = f(x_{\lfloor (1-\rho)N \rfloor})$ le $(1 - \rho)$ quantile
4. Calculer:
$$\lambda_{t+1} \in \arg \max_{\lambda \in \Lambda} \sum_{n=1}^N (\delta [f(x_n) \geq \gamma_t] \ln \pi(x_n, \lambda))$$
5. Poser $t = t + 1$ et aller en 2 si la convergence n'est pas suffisante

CE et optimisation

Plus de critère d'arrêt γ , plus de probabilité p de référence

1. choisir λ_0 tel que $\pi(\cdot, \lambda_0)$ soit uniforme, poser $t = 0$
2. Tirer x_1, \dots, x_N selon $\pi(\cdot, \lambda_t)$, et sélectionner les $\lfloor \rho N \rfloor$ meilleurs échantillons
3. Calculer: $\lambda_{t+1} \in \arg \max_{\lambda \in \Lambda} \sum_{\text{selection}} \ln \pi(x_n, \lambda)$
4. Poser $t = t + 1$ et aller en 2 si la convergence n'est pas suffisante

Planification de trajectoire (non dynamique)

Problem of interest

- a mobile equipped with sensors,



[Francis Celeste]

Problem of interest

- a mobile equipped with sensors,
- a known map describing its environment
- one mission : *join a goal point (or set) q_f from a starting point q_i*

Plan trajectories which will guarantee map-based localization performance during execution

[Francis Celeste]

The reference map model

- A *metric* map composed of N_f geometric features :
 - * points.
 - * lines.
 - * 3D objects.

Each feature m_i with $i \in \{1, \dots, N_f\}$ is defined by some parameters p_i .

e.g. for point landmarks $p_i = [r_i^x \ r_i^y]^T$.

The dynamic system model

- mobile state transition or motion equation

$$x_0 \sim \pi_0 \triangleq \mathcal{N}(\bar{x}_0, P_0)$$

$$x_{k+1} = f(x_k, a_k) + w_k \quad w_k \sim \mathcal{N}(0, Q_k)$$

with $x_0, x_k, x_{k+1} \in \mathbb{R}^{n_x}$, $a_k \in \mathcal{A}(x_k)$.

e.g.

$$x_k = [r_k^x \ r_k^y \ \theta_k]^T$$

$$\mathcal{A}(x_k) = \{-d, 0, +d\} \times \{-d, 0, +d\}, \quad d \in \mathbb{R}^+$$

The dynamic system model

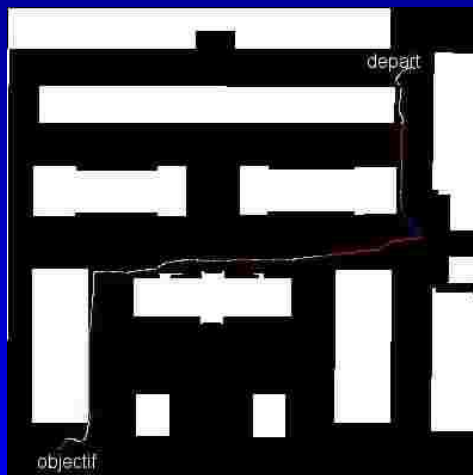
- measurement or observation model

We are mainly interested in *range* and *bearing* information :

for object m_j assumed “visible” :

$$z^k(j) \triangleq \begin{cases} z_r^k(j) = d(x_k, p_j) + \gamma_r^k(j) \\ z_\beta^k(j) = \alpha(x_k, p_j) + \gamma_\beta^k(j) \end{cases}$$

$\gamma_r^k(j)$; $\gamma_\beta^k(j)$ white gaussian noise processes.

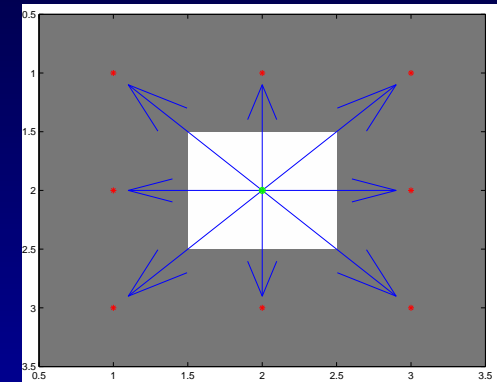


$$d(x_k, p_j) = \sqrt{(r_k^x - r_j^x)^2 + (r_k^y - r_j^y)^2}$$

$$\alpha(x_k, p_j) = \arctan_2(r_j^y - r_k^y, r_j^x - r_k^x) - \theta_k$$

A discrete approach

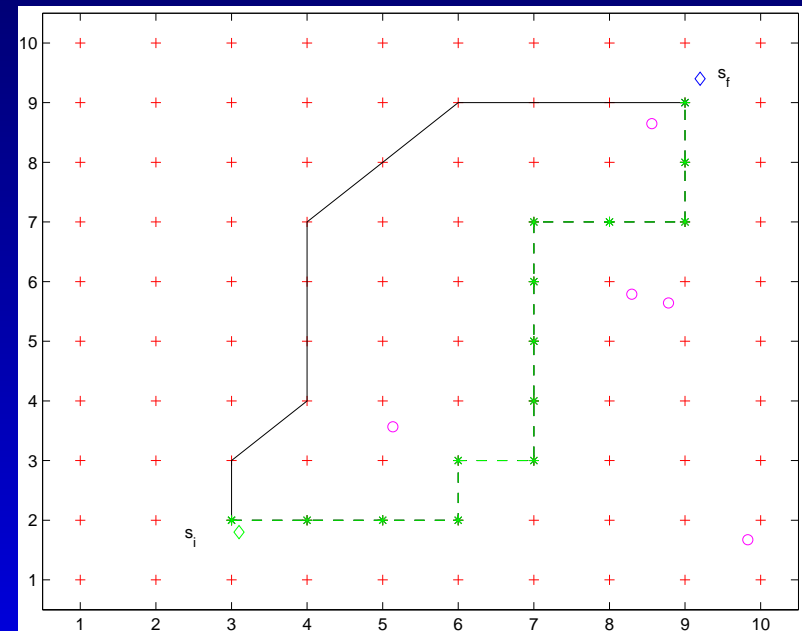
- The map is discretized in $N_s = N_x \times N_y$ locations (states s)
- One action a (or decision) is a move between two neighbor locations
 - State space $S = \{1, \dots, N_s\}$
 - Action space $A(s) = \{1, \dots, N_a\}$
($N_a = 8$ if all neighbors are reachable)
- One trajectory is a sequential state or decision vector
- One admissible trajectory for the planning task must satisfied under *motion constraints*:
 - first action is taken in q_i
 - last action allows the mobile to reach q_f



A discrete approach

- motion constraints
 - Choose a_{k+1} with respect to the decision a_k taken at time k
 - define $\delta(a_k, a_{k+1})$

$$\delta = \begin{pmatrix} & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array} & \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \end{array} \end{pmatrix}$$



constrained/unconstrained

Map-based localization

- Estimate the state x_k at time k from the measurements $Z_{1:k}$ up to time $k' = k$ (filtering problem).
 - unbiased estimate
 - random measurements
- ⇒ Posterior Cramer Rao Bound: measure of performance.
- If $V_a^k = \{a_1, \dots, a_T\}$, is a policy which generates an admissible trajectory $x_{0:T}$, let \hat{x}_k ($k \geq 1$) be the estimate of x_k from $Z_{1:k}$ then :

$$E\{(\hat{x}_k - x_k) (\hat{x}_k - x_k)^T | V_a^k\} \succeq J_k^{-1}(V_a^k)$$

where $J_k^{-1}(V_a^k)$ is the Fisher information submatrix which can be recursively computed [Tichavsky and al.]:

$$J_{k+1} = D_k^{22} - D_k^{21} (J_k + D_k^{11})^{-1} D_k^{12}$$

PCRB computation

$$D_k^{11} = E\{-\Delta_{x_k}^{x_k} \log(p(x_{k+1}|x_k))\},$$

$$D_k^{12} = E\{-\Delta_{x_k}^{x_{k+1}} \log(p(x_{k+1}|x_k))\},$$

$$D_k^{21} = [D_k^{12}]^T,$$

$$D_k^{22} = E\{-\Delta_{x_{k+1}}^{x_{k+1}} \log(p(x_{k+1}|x_k))\} + \\ E\{-\Delta_{x_{k+1}}^{x_{k+1}} \log(p(Z_{k+1}|x_{k+1}))\}.$$

In our case: $J_0 = P_0^{-1}$, $D_k^{11} = Q_k^{-1}$, $D_k^{12} = -Q_k^{-1}$, $D_k^{21} = -[Q_k^{-1}]^T$, $D_k^{22} = Q_k^{-1} + J_{k+1}(Z)$

- depends on visible features
- no explicit expression (nonlinear observation)
- approximated from Monte Carlo simulation

Cost functionals

- Given T_{max} , Find $V_a^* = \{a_1^*, \dots, a_T^*\}$, $T \leq T_{max}$ which maximizes a functional of the PCRB along the trajectory.

Let $J_{0:T} = \{J_0, \dots, J_T\}$

$$\phi_1(J_{0:T}) = - \sum_{k=0}^K w_k \det(J_k^{-1})$$

$$\phi_2(J_{0:T}) = -\det(J_T^{-1})$$

- Dynamic Programming is not relevant because the “*Matrix Dynamic Programming Property*” is not satisfied for the determinant operator [Lecadre and Tremois SIAM 97].
- A learning based approach : cross entropy method.

Generating trajectories

- Generate admissible trajectories using probability matrix $\mathbf{P}_{sa} = (p_{sa})$, $s \in \{1, \dots, N_s\}$ and $a \in \{1, \dots, N_a\}$ (in our case $N_a = 8$):

$$P_s(a = i) = p_{si}, i = 1, \dots, 8 \text{ with } \sum_{i=1}^8 p_{si} = 1$$

=> optimize $N_s \times N_a$ parameters (p_{sa}) via the C.E algorithm.

Sampling with rejection of the invalid paths

Updating step

- Let $x(j) = (q_i, a_0^j, s_1^j, a_1^j, \dots, s_{k-1}^j, a_{k-1}^j, q_f)$ be one admissible path :

$$\ln \pi(x(j), \mathbf{P}_{sa}) = \sum_{i=0}^{k-1} I[\{x(j) \in \chi_{sa}\}] \ln p_{sa}$$

- Minimize CE using Lagrange multipliers

$$p_{sa} = \frac{\sum_{j=1}^N I[\{\phi_k(x(j)) \geq \gamma_n\}] I[\{x(j) \in \chi_{sa}\}]}{\sum_{j=1}^N I[\{\phi_k(x(j)) \geq \gamma_n\}] I[\{x(j) \in \chi_s\}]}$$

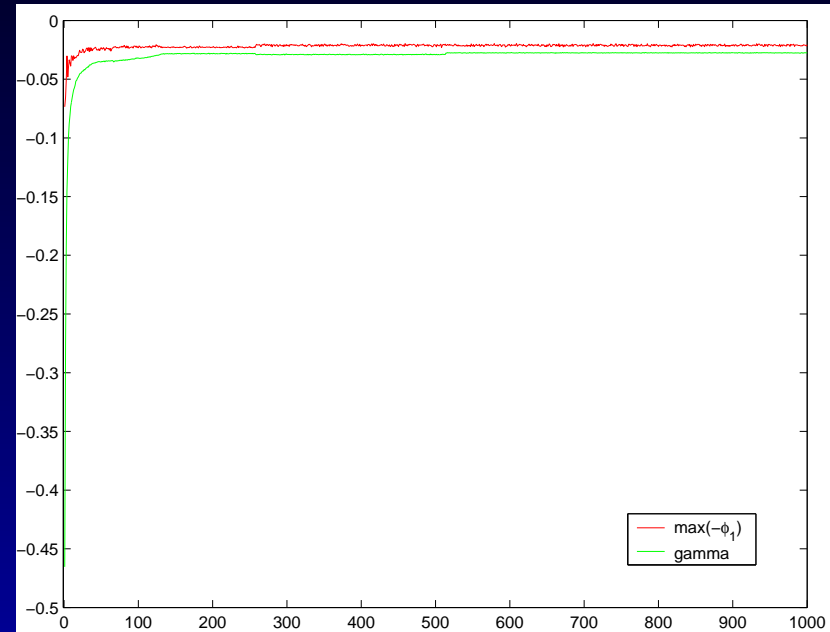
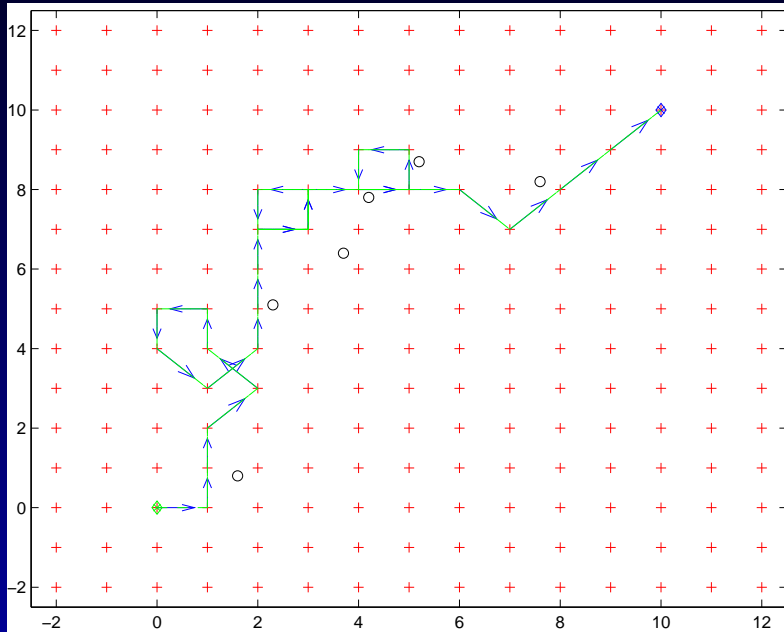
$I[\{x(j) \in \chi_{sa}\}]$: $x(j)$ contains a visit in s where a is taken.

$I[\{x(j) \in \chi_s\}]$: $x(j)$ contains a visit in s .

Experiment

- motion constraint
 - $T_{max} = 30$.
 - matrix δ : only $[-\frac{\pi}{2}; \frac{\pi}{2}]$ headings controls.
- PCRB computation
 - $N_c = 1000$ for $J_k(Z)$ estimation
- C.E algorithm
 - $N_{iter} = 1000$.
 - $N = 5000$ admissible trajectories.
 - $\rho = 0.1 \implies 500$ trajectories for the updating step.
 - performance function ϕ_1 .

Results(1)



- The C.E converges rapidly.
- All the allowed length for the path is used.
- The mobile operates to keep the landmarks visible while the maneuvers (matrix δ) allow it.

Evolution future

- Filtrage basé sur des méthodes ensemblistes
- Prise en compte dans la planification d'une fonction de recalage sur la trajectoire
- Prise en compte des observations dans un cadre dynamique



Planification de trajectoire (dynamique)

Problématique

Optimal plan and partial observations. Optimize a mission for a mobile agent endowed with sensors:

- Action should enhance the observation
- Observation may enhance the action
- Observation cost has to be compatible with the mission

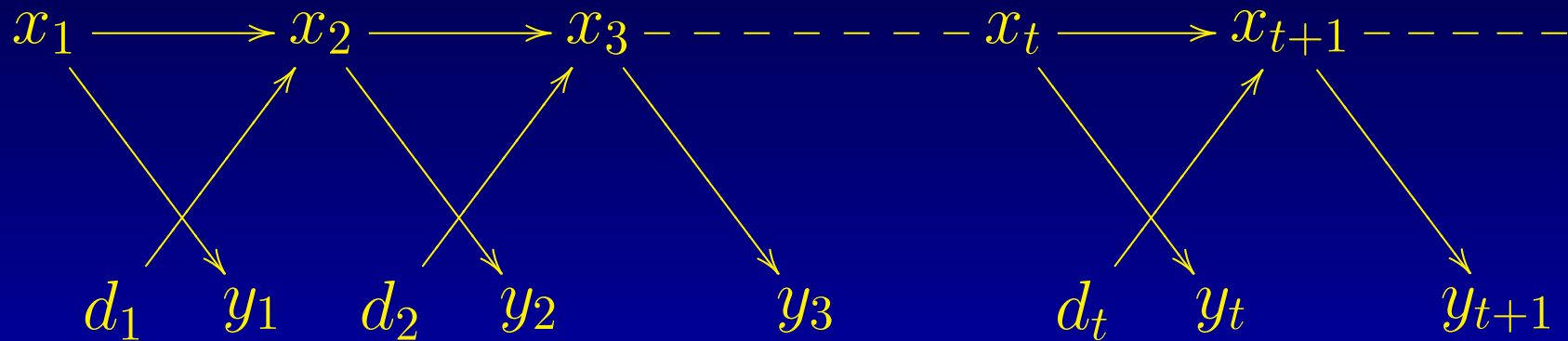
+ noisy context

Implemented method.

- Agent strategy = Dynamic Bayesian Network
→ Learn the optimal parameters of this strategy
- Control discret/continuous in x (discret in t)
- Use of semi-continuous DBN

Partially Observable Markov Decision Process

1. A controlled and partially observed universe.
→ Hidden Markov Model with control



x = hidden states; y = observation; d = action

$$P(x, y|d) = \prod_{t=1}^T p(y_t|x_t)p(x_t|d_{t-1}, x_{t-1})$$

T is the mission length

Example; hidden states

In a space of 20×20 cells:

- a target C , coordinates i_C, j_C
- a patrol P , coordinates i_P, j_P , direction Δ_P
- a patrol Q , coordinates i_Q, j_Q , direction Δ_Q

$$x_t \equiv i_{C,t}, j_{C,t}, i_{P,t}, j_{P,t}, \Delta_{P,t}, i_{Q,t}, j_{Q,t}, \Delta_{Q,t}$$

States changes.

- Patrols are controlled by actions (*determinist*)
- Target move is stochastic, favorizing escapes:

$$\left\{ \begin{array}{l} P(C_{t+1}|C_t) = 0 \text{ if } |i_{C,t+1} - i_{C,t}| > 1 \text{ or } |j_{C,t+1} - j_{C,t}| > 1 \\ P(C_{t+1}|C_t) \propto (i_{C,t+1} - i_{P,t})^2 + (j_{C,t+1} - j_{P,t})^2 + \\ \quad + (i_{C,t+1} - i_{Q,t})^2 + (j_{C,t+1} - j_{Q,t})^2 \quad \text{else} \end{array} \right.$$

Unexpected: a distant patrol hides a nearby patrol!

Example; control & observations

Control. d_t contains the actions:

- P_t (resp. Q_t) does nothing
- P_t (resp. Q_t) goes forward
- P_t (resp. Q_t) turns right
- P_t (resp. Q_t) turns left

→ $16 = 4 \times 4$ possible actions (4 per agent)

Observation. y_t contains the informations:

- P_t near C_t : $d(P_t, C_t) < 3$? ($d \equiv \max | \cdot - \cdot |$)
- Q_t near C_t : $d(Q_t, C_t) < 3$?
- C_t is forward of P_t ?
- C_t is forward of Q_t ?

→ $16 = 2^4$ possible observations (4 per agent)

POMDP – Reward

Evaluates the whole trajectories. $V(x, y, d)$

Quick computation requirement.

→ EG. a recursive definition:

$$\begin{cases} V(x, y, d) = V_T(x, y, d) \\ V_t(x, y, d) = v_t(V_{t-1}(x, y, d), x_t, y_t, d_t) \\ V_0(x, y, d) = 0 \end{cases}$$

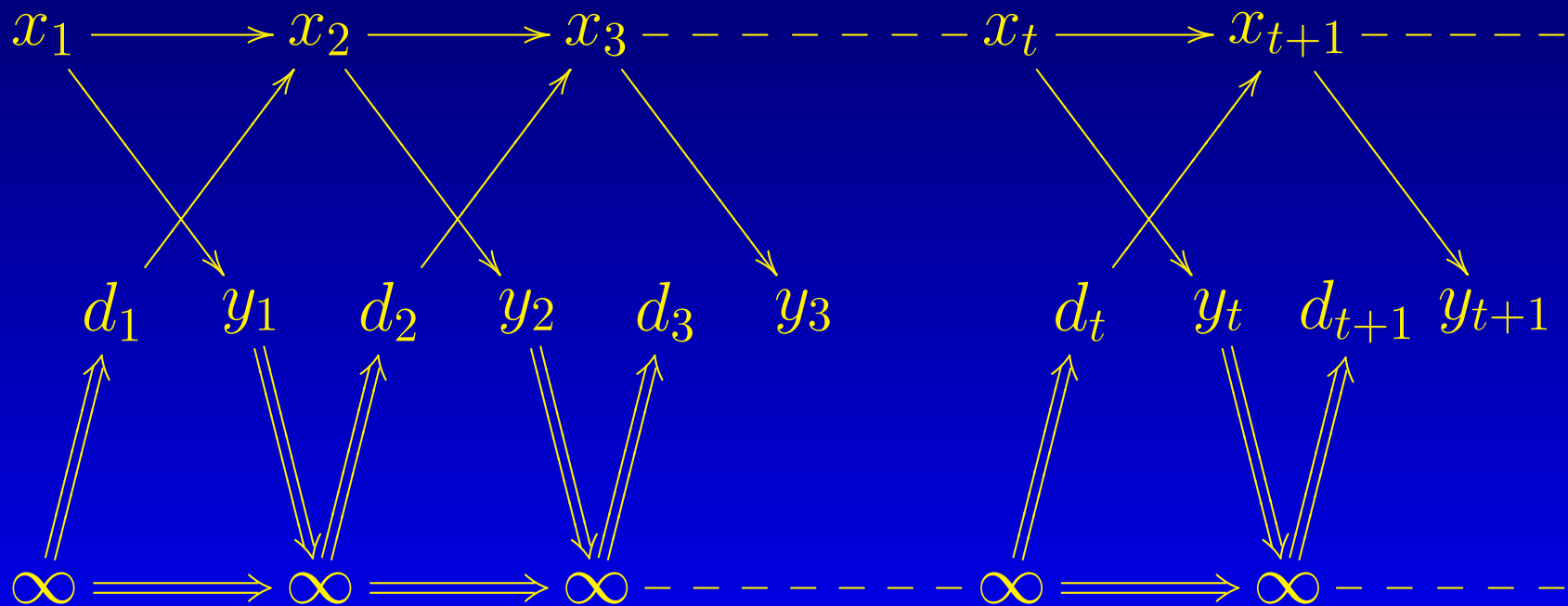
Example. Reward incremented each time a patrol (at least one) is in the vicinity of the target:

- $V_t = V_{t-1} + 1$ if $d(P_t, C_t) \leq 3$ **or** $d(Q_t, C_t) \leq 3$
- $V_t = V_{t-1}$ else

The optimal decision

Issue. Optimize the decision tree $(d_t(y_{1:t-1}))_{1:T}$ and maximize the expected reward:

$$\sum_{x,y} P(x, y | d_t(y_{1:t-1})_{1 \leq t \leq T}) V(x, y, d_t(y_{1:t-1})_{1 \leq t \leq T})$$



∞ = very high capacity memory

Approximating the decision tree

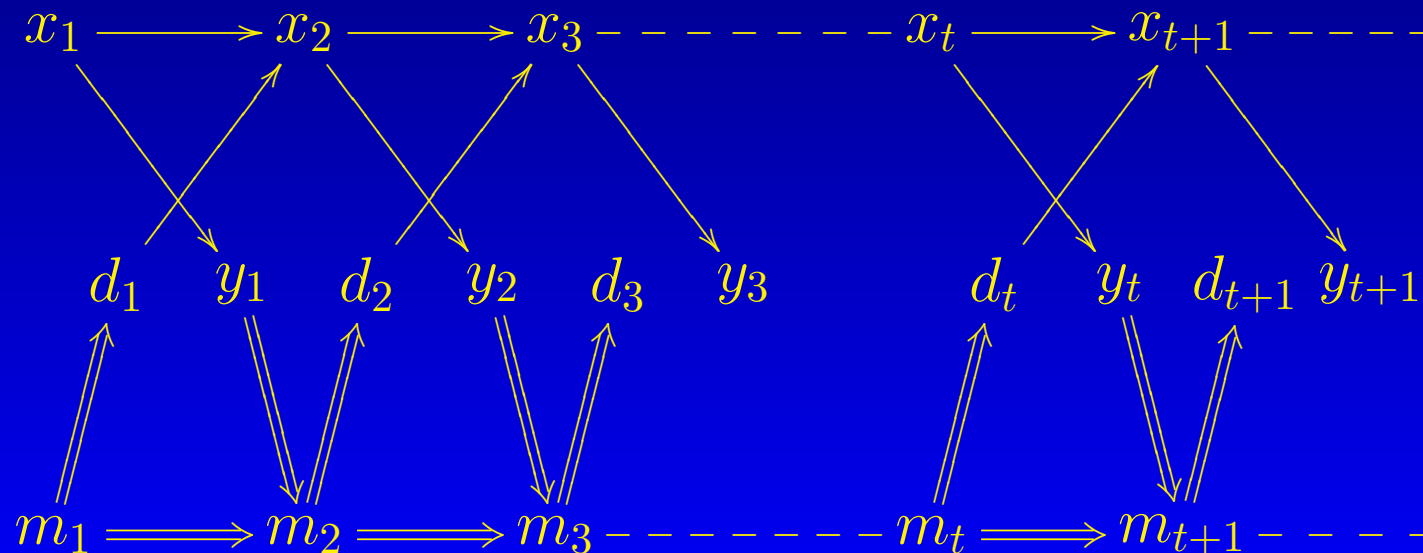
Step 1. Probabilize the problem (equivalent):

Find $\pi_O(d|y) \in \arg \max_{\pi(d|y)} V[\pi]$, where:

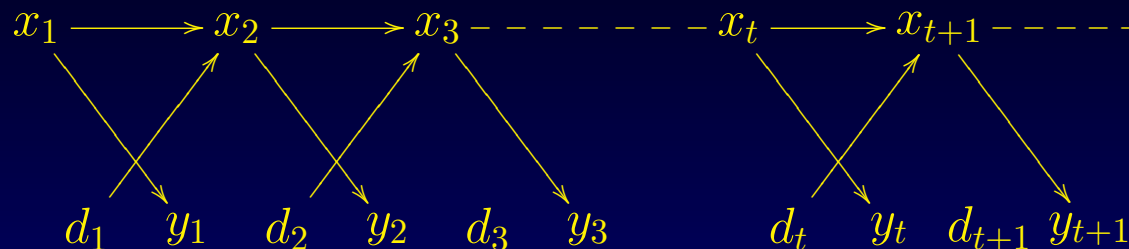
$$V[\pi] = \sum_{x,y,d} \prod_{t=1}^T \pi(d_t | d_{1:t-1}, y_{1:t-1}) \times P(x, y | d) V(x, y, d)$$

Step 2. Restrict to a parameterized family of probabilistic laws (approximation). In particular:

FINITE MEMORY \Rightarrow Family = HMM



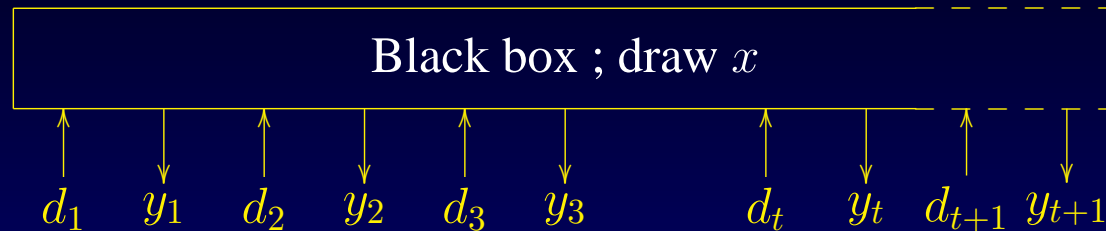
Method and generalization



- HMM + control : $P(x, y|d) = \prod_t p(y_t|x_t)p(x_t|d_{t-1}, x_{t-1})$
- $x/y/d$: hidden state / observation / action
- Each trajectory (d, x, y) evaluated by $V(d, x, y)$
Constraint: **quick computation** ; additivity is **not needed**

$\max E(V) \Rightarrow$ Optimize the decision law $\pi(d_t|d_{1:t-1}, y_{1:t-1})|_{1:T}$

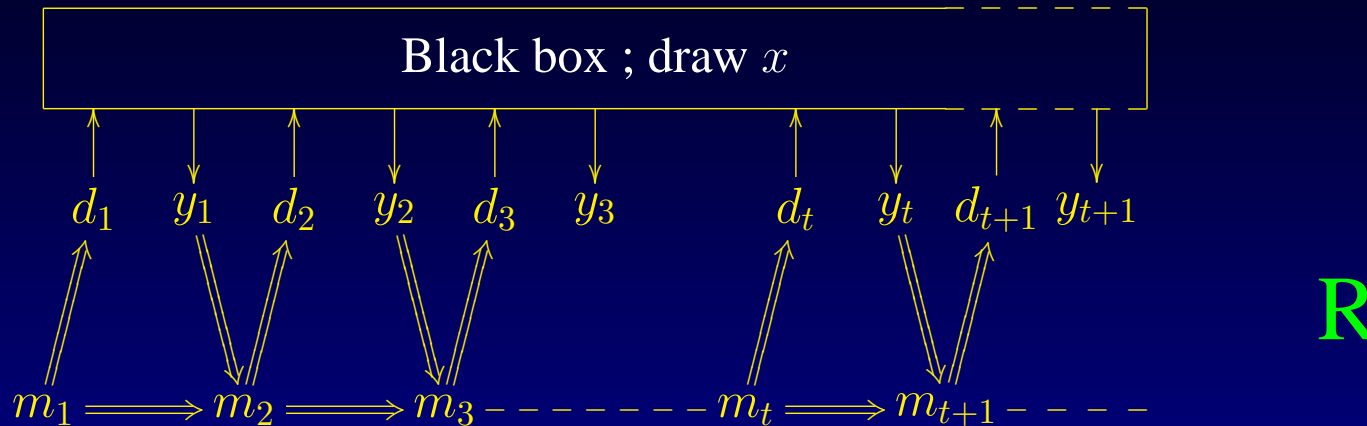
Method and generalization



- Black box : $P(x, y|d)$ with **quick draw**
- $x/y/d$: hidden state / observation / action
- Each trajectory (d, x, y) evaluated by $V(d, x, y)$
Constraint: **quick computation** ; additivity is **not needed**

$\max E(V) \Rightarrow$ Optimize the decision law $\pi(d_t | d_{1:t-1}, y_{1:t-1}) \Big|_{1:T}$

Method and generalization



- Black box : $P(x, y|d)$ with **quick draw**
- $x/y/d$: hidden state / observation / action
- Each trajectory (d, x, y) evaluated by $V(d, x, y)$
 Constraint: **quick computation** ; additivity is **not needed**

$\max E(V) \Rightarrow$ Optimize the decision law $\pi(d_t | d_{1:t-1}, y_{1:t-1}) |_{1:T}$

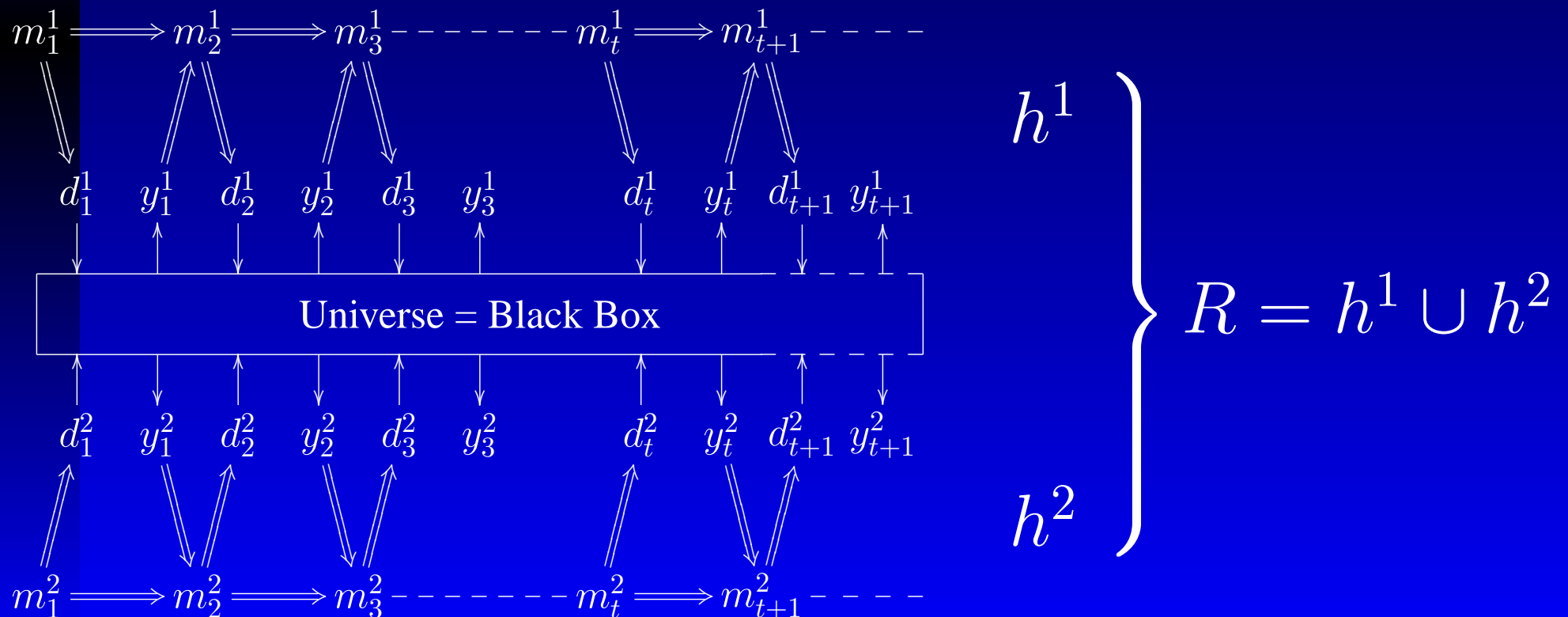
- Approximate π by a Dynamic Bayesian Networks **R**
- **Method**: Optimize **R** by the Cross-Entropy algorithm
- **Constraint**: Quick draw and evaluation of the samples

Examples of DBN

Jointly controlled agents. A unique HMM inputting all informations and outputting all decisions

Previous work: Hierarchical HMMs \Rightarrow less memory

Independent agents. Disjoints DBN interacting with the universe (\supset information exchange)



Optimizing the parameters

Define (for A agents $\alpha = 1$ to n)

$$P[h](x, y, z, m) = P(y, z|x) \times \prod_{\alpha=1}^A \prod_{t=1}^T h^{\alpha}(x_t^{\alpha}|m_t^{\alpha}) h^{\alpha}(m_t^{\alpha}|y_{t-1}^{\alpha}, m_{t-1}^{\alpha})$$

Issue. Find $h_O \in \arg \max_h \sum_{x,y,z,m} P[h](x, y, z, m) V(x, y, z)$

\Rightarrow **Cross Entropy method.** Selection Rate $\rho \in]0, 1[$

1. **Initialize** h . For example a flat h
2. Make N **samples** $\theta^n = (x^n, y^n, z^n, m^n)$ **according to** $P[h]$
3. Let S be the set of the ρN **best samples** according to $V(x, y, z)$
4. Update h as the **minimizer** of the **cross-entropy** relatively to S :

$$h \in \arg \max \sum_{n \in S} \ln P[h](\theta^n) \quad (\text{easy to compute!})$$

5. **Reiterate** from step 2 until convergence

Applying to the example

About five hours needed for convergence (2GHz PC)

Settings. Total number of turns $T = 100$

Initially ($t = 0$), the patrols are directed down and located down-left $0, 19$ and down-right $19, 19$

R moves stochastically. Initially located randomly (uniform) within the 20×10 upper lattice

Jointly controlled Case.

Test 1. The **observations are removed** ($y \rightarrow m$ is removed). The optimal mean evaluation is **32**

Test 2. **No observation memory.** ($m \rightarrow m$ is removed): The optimal mean evaluation is **54**

Example (continued)

Test 3. Full HMM.

$ \{m\} $	16	32	64	256	256, “∞”run
$E(V)$	65	66	67	67	69

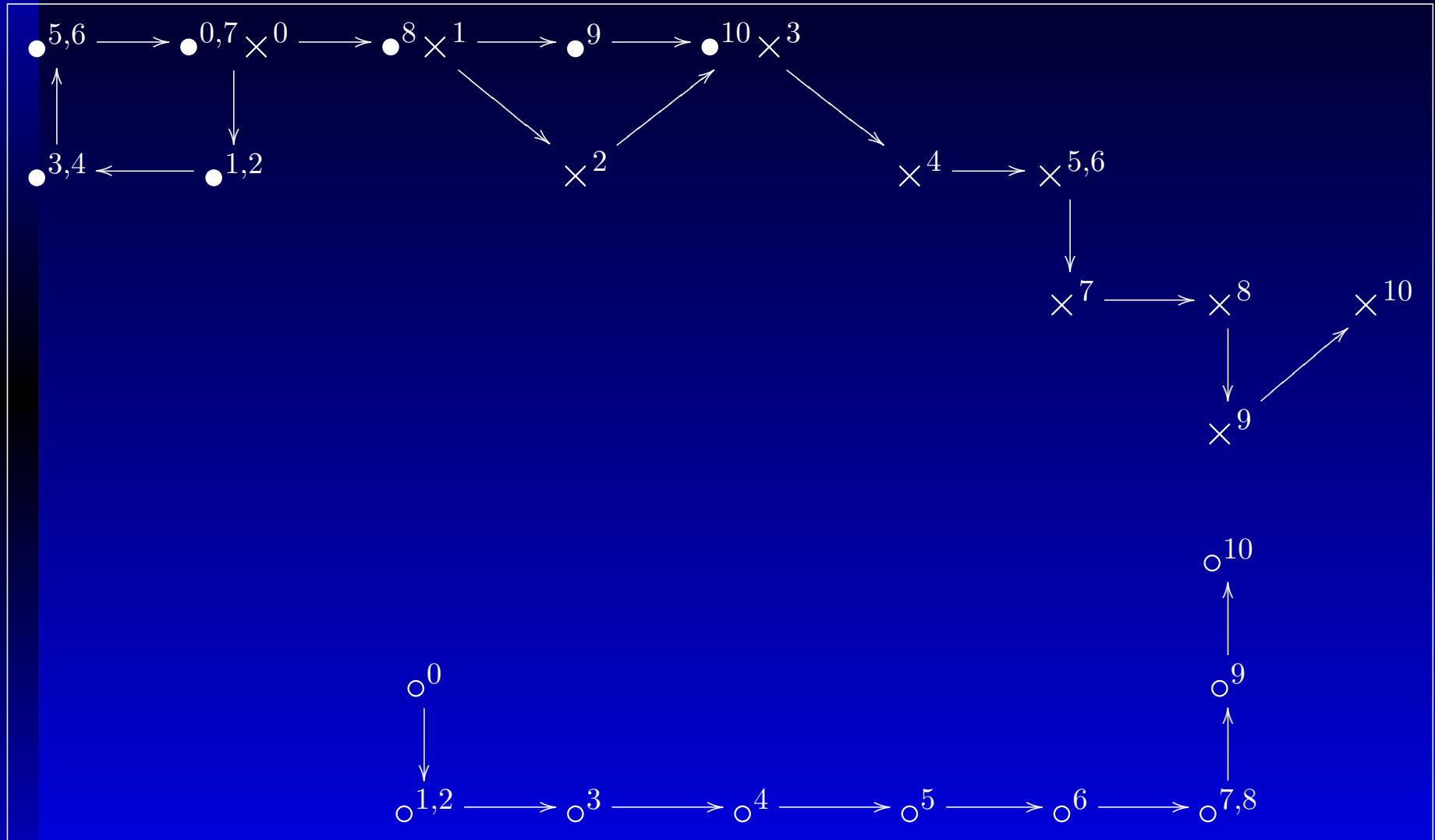
Independent patrols. The patrols exchange their observations and actions with noise (bit inversion)

Tests made on HMMs with $|\{m\}| = 128$. First cases, patrols are running simultaneously. Last case, *in turn*

$P(\text{bit inv})$	0.5	0.25	0	$P, Q, P \dots$
$E(V)$	61	62	64	65 (should be 67?)

Convergence criterion: 1000 unsuccessful tries (strong)
 The convergence is more difficult than in the joint case;
 More memory is needed; the effect of the noise is weak

Example: an escape/tracking sequence



A continuous example

Setting

Hidden states. In the continuous space $[-20, 20] \times [-20, 20]$:

- a target T , coordinates xT, yT
- a patrol P , coordinates xP, yP
- a patrol Q , coordinates xQ, yQ

$$x_t = (xT_t, yT_t, xP_t, yP_t, xQ_t, yQ_t)$$

States changes.

- Patrols move according to the decision
- Target tries to escape:
 - 5 random locations for T are chosen within $[xT - vT_{max}, xT + vT_{max}] \times [yT - vT_{max}, yT + vT_{max}]$
 - Is kept the location xT', yT' maximizing the distance $d(T', \{P, Q\})$ with the patrols

Example; control, observation, reward

Control. $d_t = (\Delta x_P, \Delta y_P, \Delta x_Q, \Delta y_Q)$ is continuous: $\|\Delta P\| \leq vP_{max}$ and $\|\Delta Q\| \leq vQ_{max}$

Observation. Each patrol observes the relative angular position of the target with a noise, *i.e.*

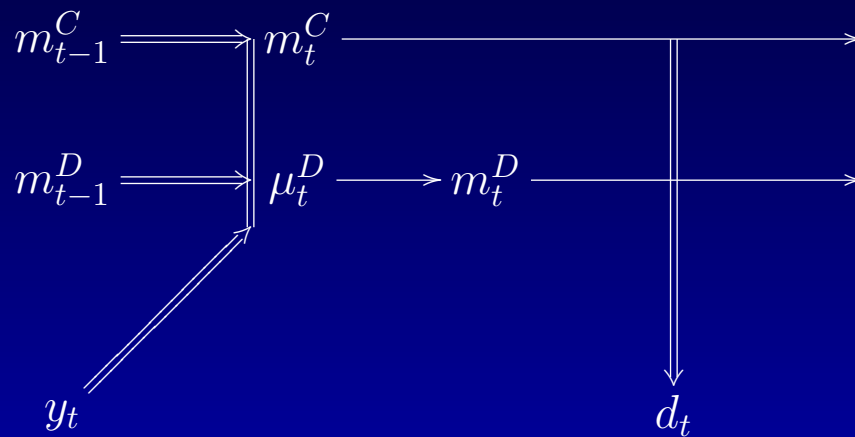
$$\theta_P = (x'x, T - P) + n_P \text{ and } \theta_Q = (x'x, T - Q) + n_Q$$

- The noises n_P and n_Q are increasing with the distance up to $+\pi/2$
- The noisy observation is discretized by a grid of 8 angular sectors: 64 observations for both P and Q

Reward. It is given by the minimum distance to the target, *i.e.* $V(x, d, y) = \min_t d(T_t, \{P_t, Q_t\})$

Choice of a strategic model

First attempt: choice of a semi-continuous transition for $h \rightarrow$ no discretization of the observation



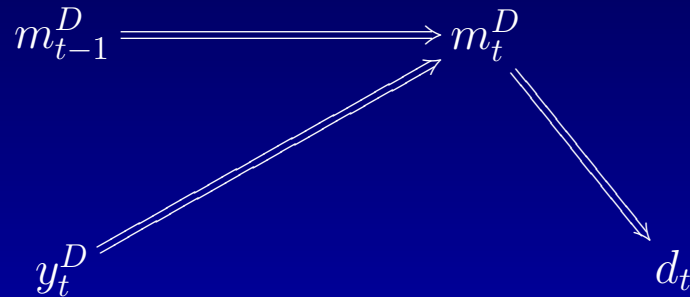
The variables m^C , y_t , μ^D , d_t are continuous, and the transitions are Gaussian (to be learned)

The variable m^D is discrete, and obtained from μ^D by a discretization grid (pre-defined)

At this time, the transition learning does not work properly; The decision $m^C, m^D \rightarrow d$ seems OK

Alternative model

Second attempt: transition+observation are strictly discrete, but decisions are continuous



d_t is the only continuous variable

The transitions for d is Gaussian, while the other transition are general discrete law

All the transitions have to be learned

Typical updating

Obtained by the CE optimization

Discrete transition $h(B|A) = \frac{\#n \in S / B_n = B \& A_n = A}{\#n \in S / A_n = A}$

Gaussian transition $\mu(B|A) = \frac{\sum_{n \in S / A_n = A} B^n}{\#n \in S / A_n = A}$

$$\Sigma(B|A) = \frac{\sum_{n \in S / A_n = A} B^{nT} B^n}{\#n \in S / A_n = A} - \mu(B|A)^T \mu(B|A)$$

Applying to the example

Size of the model: m takes 256 possible values // [5pt]
About 30mn needed for convergence (2GHz PC)

Settings. Total number of turns $T = 100$

Initially ($t = 0$), the patrols are located at $(-20, 0)$
 T is initially located randomly (uniform) within the
space $[0, 20] \times [-20, 20]$

Maximum speeds are 2 for target and patrols

0 observation, 0 transition. Evaluation is about 10

Continuous model. Evaluation is about 10

Alternative model. Evaluation is 1.04

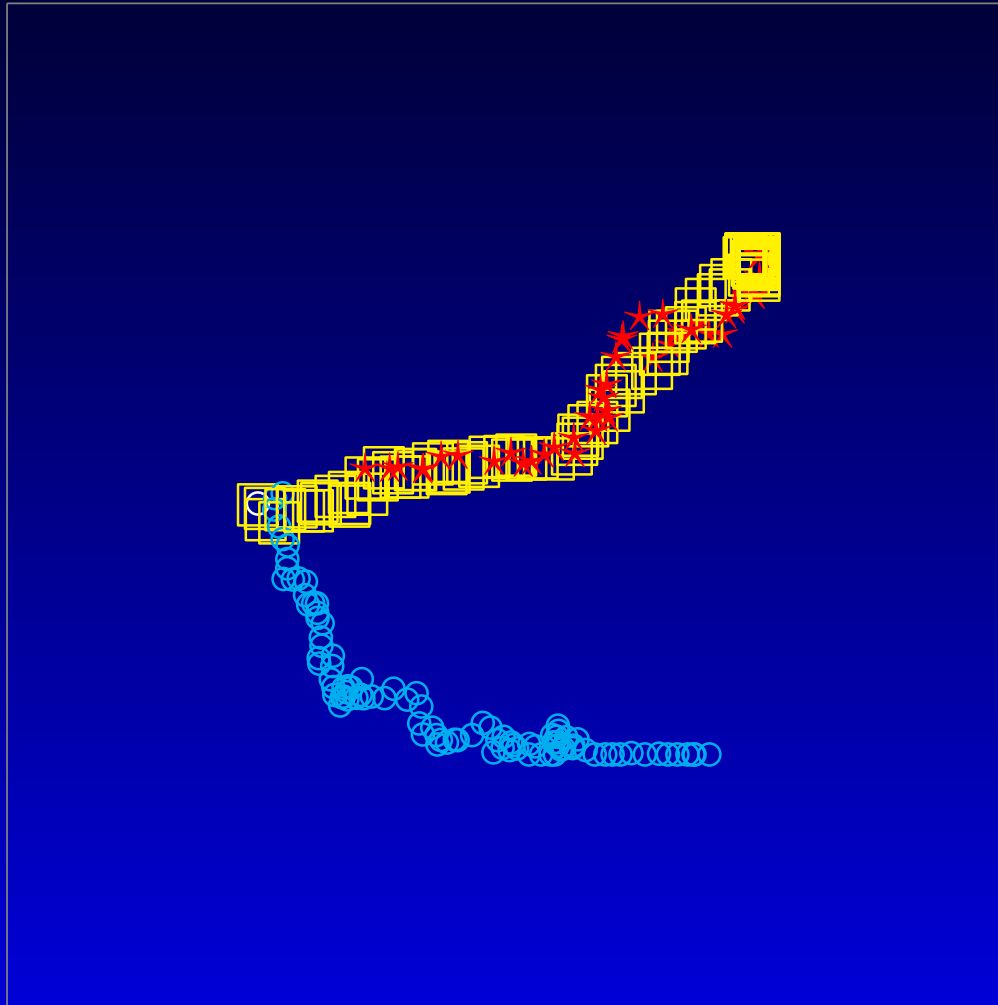
Example continued

Convergence. Start from 30.2 down to 1.04

Influence of the Noise. Comparison of different noise increase

	<i>No</i>	<i>low</i>	<i>middle</i>	<i>high</i>
<i>Eval</i>	1.04	1.07	1.97	2.65

Example of trajectories



★: target ; □: patrol P ; ○: patrol Q

Evolution future

Fonctionnement online. Typiquement, modifier l'algorithme CE afin de paralléliser le processus de simulation et le processus d'action/observation *réel*

Le processus réel fournira des échantillons plus lentement que le processus de simulation

Cf. apprentissage selectif en robotique

Planification et hiérarchie. La grande affinité de la CE avec les modèles bayésiens permet de prendre en compte les modèles hiérarchiques les plus classiques

Evolution du schéma de sélection de la CE. Sur des problèmes stochastiques, il semble que la sélection par quantile ne soit pas la plus adaptée

Conclusion

Flexibilit de la CE. Optimisation de fonctionnelles non aditives. Exemple pour une planification de trajectoire à partir d'une carte d'amers

Optimisation de rseau bayisien. Intérêt pour l'approximation d'arbres de décision. Forte compatibilité avec la CE Application à des problèmes de décision avec observation partielle

Futurs objectifs. Mise en oeuvre dans un contexte online. Accélération de la convergence. Exploitation des capacités de modélisation des réseaux bayésiens