A reinforcement learning algorithm for sampling design in Markov random fields

Mathieu BONNEAU   Nathalie PEYRARD   Régis SABBADIN

INRA-MIA Toulouse
E-Mail: {mbonneau,peyrard,sabbadin}@toulouse.inra.fr

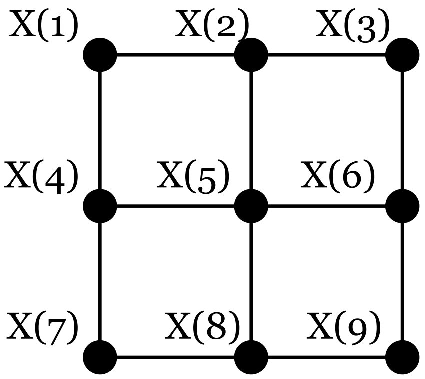
SSIAB, Avignon, 9 mai 2012

Plan

1. Problem statement.
2. General Approach.
3. Formulation using dynamic model.
4. Reinforcement learning solution.
5. Experiments.
6. Conclusions.

PROBLEM STATEMENT

Adaptive sampling in Markov random fields



- Adaptive selection of variables to observe for reconstruction of the random vector **X**=(X(1),….,X(n))

- c(A) -> Cost of observing variables X(A)

- B -> Initial budget

- Observations are reliable

PROBLEM STATEMENT

Adaptive sampling in Markov random fields

- Adaptive selection of variables to observe for reconstruction of the random vector **X**=(X(1),....,X(n))

- Observations are reliable

- c(A,x(A)) -> Cost of observing variables X(A) in state x(A)

- B -> Initial budget

<u>Problem</u>: Find strategy / sampling policy to adaptively select variables to observe in order to :

Optimize Quality of the reconstruction of X / Respect Initial Budget

PROBLEM STATEMENT

Adaptive sampling in Markov random fields

- Adaptive selection of variables to observe for reconstruction of the random vector **X**=(X(1),….,X(n))
- Observations are reliable

- c(A,x(A)) -> Cost of observing variables X(A) in state x(A)
- B -> Initial budget

Problem: Find strategies / sampling policy to adaptively select variables to observed in order to :

Optimize Quality of the reconstructed vector X / Respect Initial Budget

DEFINITION: Adaptive sampling policy

For any *sampling plans* $A^1,\ldots,A^t$ and observations $x(A^1),\ldots,x(A^t)$, an adaptive sampling policy δ is a function giving the next variable(s) to observe:

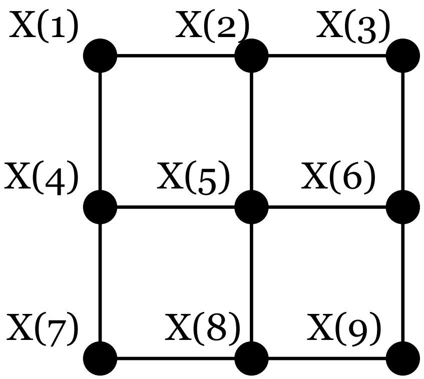
$$\delta\big((A^1,x(A^1)),\ldots,(A^t,x(A^t))\big)=A^{t+1}.$$

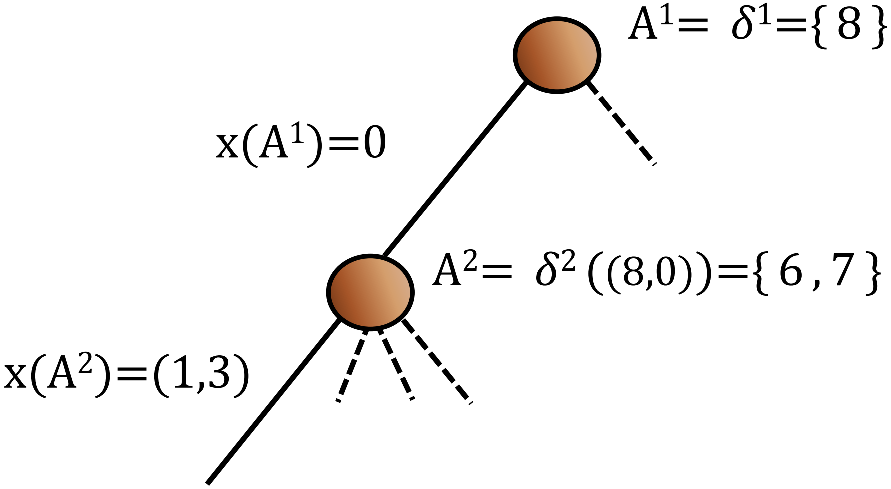
DEFINITION: Adaptive sampling policy

For any *sampling plans* $A^1,\dots,A^t$ and observations $x(A^1),\dots, x(A^t)$, an adaptive sampling policy δ is a function giving the next variable(s) to observe:

$$\delta((A^1,x(A^1)),\dots,(A^t,x(A^t)))=A^{t+1}.$$

-- Example --





$A^1= \delta^1=\{8\}$

$x(A^1)=0$

$A^2= \delta^2((8,0))=\{6,7\}$

$x(A^2)=(1,3)$

DEFINITIONS

For any *sampling plans* $A^1,\ldots,A^t$ and observations $x(A^1),\ldots,x(A^t)$, an adaptive sampling policy δ is a function giving the next variable(s) to observe:

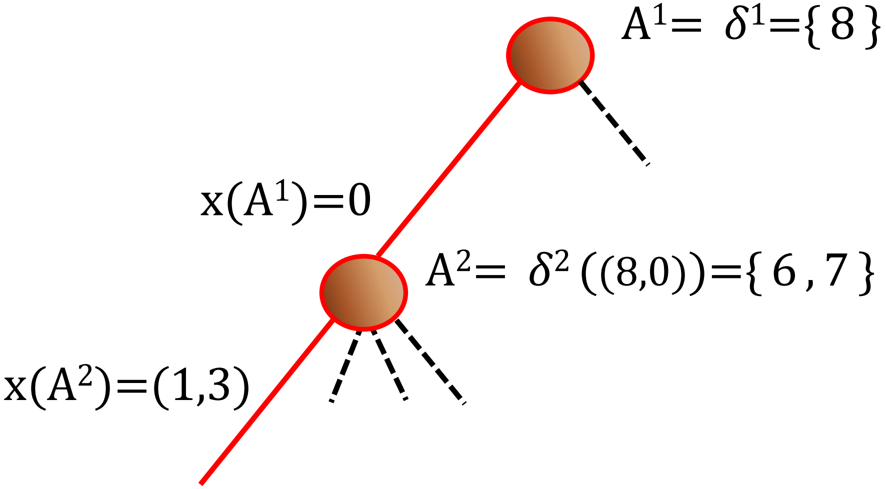
$$\delta\big((A^1,x(A^1)),\ldots,(A^t,x(A^t))\big)=A^{t+1}.$$

•<u>Vocabulary :</u>

•A *history* $\{(A^1,x(A^1)),\ldots,(A^H,x(A^H)\}$ is a trajectory followed when applying δ

• τ_δ : set of all reachable histories of δ

• $c(\delta) \leq B$ ⇔ cost of any history respects the initial budget



GENERAL APPROACH

GENERAL APPROACH

1. Find a distribution $\mathbb{P}$ that well describes the phenomenon under study.

2. Define the value of adaptive sampling policy:

$$V(\delta) = \sum_{(A,x(A))\in\tau_\delta} \mathbb{P}\big(x(A)\big)\, U\big(A, x(A)\big)$$

3. Define approximate resolution method for finding near optimal policy:

$$\delta^* = \arg\max_{\delta, c(\delta)\leq B} V(\delta)$$

STATE OF THE ART

1. Find a distribution $\mathbb{P}$ that well describes the phenomenon under study.

   - ➢ X continuous random vector
   - ➢ $\mathbb{P}$ multivariate Gaussian joint distribution

2. Define the value of adaptive sampling policy:

$$V(\delta) = \sum_{(A,x(A))\in\tau_\delta} \mathbb{P}(x(A))\, U(A, x(A))$$

   - ➢ Entropy based criterion
   - ➢ Kriging variance

3. Define approximate resolution method for finding near optimal policy:

$$\delta^* = \arg \max_{\delta, c(\delta)\leq B} V(\delta)$$

   - ➢ Greedy algorithm

OUR CONTRIBUTION

1. Find a distribution $\mathbb{P}$ that well desribed the phenomenon under study.

   ➢ X continuous random vector

   ➢ $\mathbb{P}$ multivariate Gaussian joint distribution

   ➢ X discrete random vector

   ➢ $\mathbb{P}$ Markov random field distribution

2. Define the value of adaptive sampling policy:

$$V(\delta) = \sum_{(A,x(A))\in\tau_\delta} \mathbb{P}(x(A))\,U(A, x(A))$$

   ➢ Entropy based criterions

   ➢ Kriging variance

   ➢ Maximum Posterior Marginals (MPM)

3. Define approximate resolution method for finding near optimal policy:

$$\delta^* = \arg\max_{\delta, c(\delta)\leq B} V(\delta)$$

   ➢ Greedy algorithm

   ➢ Reinforcement learning

OUR CONTRIBUTION

1. Find a distribution $\mathbb{P}$ that well desribed the phenomenon under study.

   - X continuous random vector
   - $\mathbb{P}$ multivariate Gaussian joint distribution

   - X discrete random vector
   - $\mathbb{P}$ Markov random field distribution

2. Define the value of adaptive sampling policy:

$$V(\delta) = \sum_{(A,x(A))\in\tau_\delta} \mathbb{P}(x(A))\, U(A, x(A))$$

   - Entropy based criterions
   - Kriging variance

   - Maximum Posterior Marginals (MPM)

3. Define approximate resolution method for finding near optimal policy:

$$\delta^* = \arg\max_{\delta, c(\delta)\leq B} V(\delta)$$

   - Greedy algorithm

   - Reinforcement learning

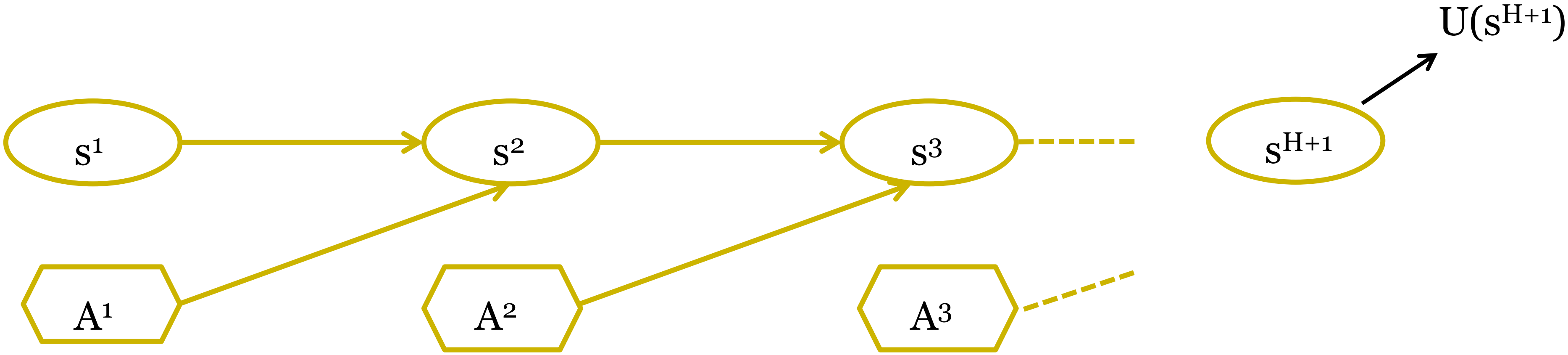
Formulation using dynamic model

An adapted framework for reinforcement learning

- ➢ Summarize knowledge on **X** in a random vector S of length n

•Observe variables → update our knowledge on **X** → Evolution of S

•<u>Example:</u> s = (-1, …….. , $\underset{i}{k}$, …….. , -1) ⟶ Variable X(i) was observed in state k

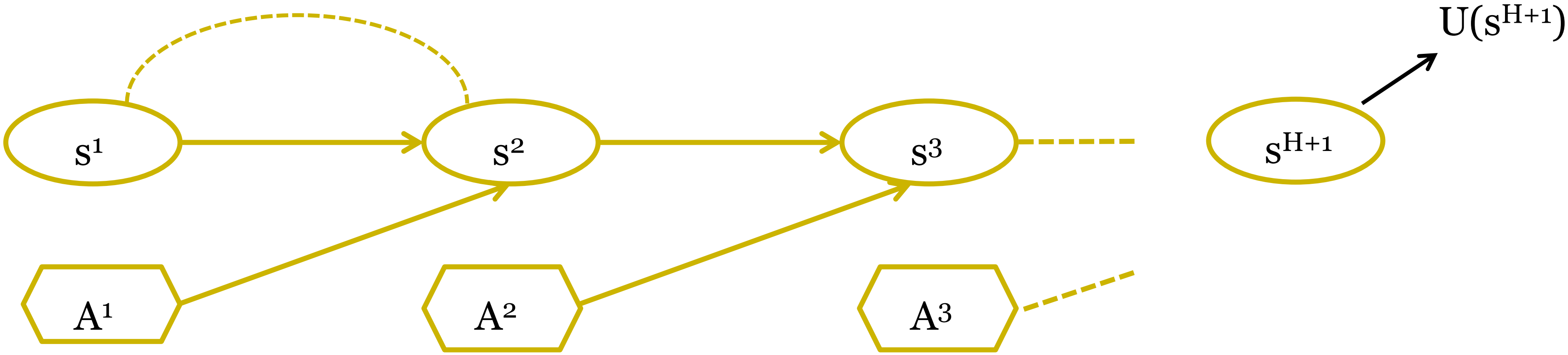


- ➢ Summarize knowledge on **X** in a random vector S of length n

- •Observe variables → update our knowledge on **X** → Evolution of S

- •<u>Example:</u> $s = (\underbrace{-1, \ldots\ldots}, \underset{i}{k}, \underbrace{\ldots\ldots, -1})$ ⟶ Variable X(i) was observed in state k

$$\mathbf{P}\big(s^{t+1} \mid s^t, A^t\big) = \mathbb{P}\big(x(A^t) \mid x(A^1), \ldots, x(A^{t-1})\big)$$



Reinforcement learning solution

Find optimal policy: The Q-function

$\forall t, \forall s^t, \forall A^t$ $Q^*(s^t, A^t)$ = « The expected value of the history when starting in s^t, observing variables $X(A^t)$and then following policy δ*»

$$= \sum_{s^{t+1}} \mathbf{P}(s^{t+1} \mid s^t, A^t) \max_{A^{t+1}} Q^*(s^{t+1}, A^{t+1})$$

- $\delta^*(s^t) = \arg\max_{A^t} Q^*(s^t, A^t)$

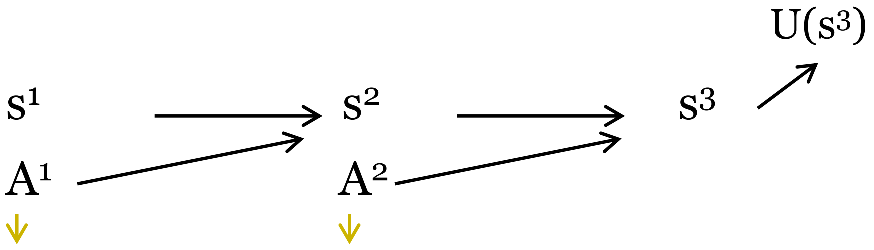
➢ Compute Q* ⇔ Compute δ *

Find optimal policy: The Q-function

- How to compute Q*: classical solution (Q-learning …)

1. Initialize Q

2. Simulate history

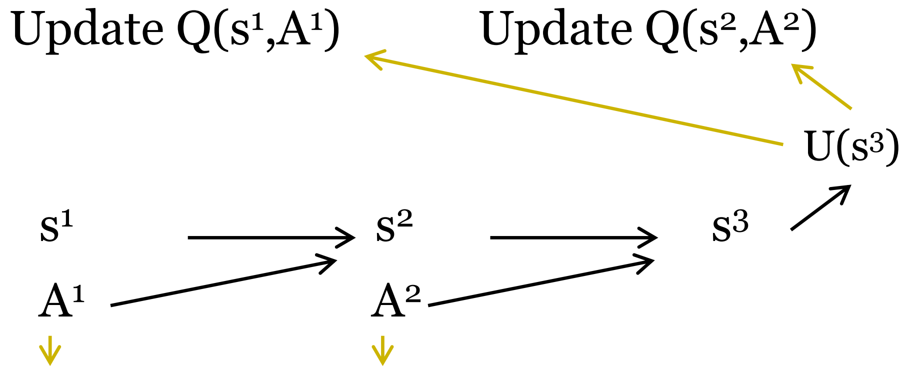


$$\begin{cases} \arg\max_{A^t} Q(s^t, A^t) & \text{with proba } 1\text{-}\varepsilon \\ \text{random} & \text{with proba } \varepsilon \end{cases}$$

Find optimal policy: The Q-function

- How to compute Q*: classical solution (Q-learning …)

1. Initialize Q



Update $Q(s^1,A^1)$

Update $Q(s^2,A^2)$

$U(s^3)$

2. Simulate history

s^1 s^2 s^3

many times!

A^1 A^2

$$\begin{cases} \arg\max_{A^t} Q(s^t, A^t) & \text{with proba } 1\text{-}\varepsilon \\ \text{random} & \text{with proba } \varepsilon \end{cases}$$

Alternative approach

- Linear approximation of Q-function:

$$\forall t = 1 \dots H \qquad \bullet\ \widetilde{Q}^*(s^t, A^t) = \sum_{i=1}^{p} w_i \phi_i(s^t, A^t)$$

$$\simeq \sum_{s^{t+1}} \mathbf{P}\big(s^{t+1} \mid s^t, A^t\big) \max_{A^{t+1}} Q^*(s^{t+1}, A^{t+1})$$

$$\bullet\ \widetilde{Q}^*(s^{H+1}) = Q^*(s^{H+1}) = U\big((A^1, x(A^1)), \dots, (A^H, x(A^H))\big)$$

- Choice of function Φ_i:

$$\forall i = 1 \dots n \qquad \phi_i(s^t, A^t) = \max_{x(i)} \mathbb{P}\big(x(i) \mid x(A^1), \dots, x(A^{t-1})\big)$$

$$= 1 \quad \text{si } i \in A^t$$

LSDP Algorithm

- ➢ Define weights for each decision step

- ➢Compute weights using “backward induction”

- Linear approximation of Q-function:

$$\forall t = 1 \ldots n \qquad \bullet\ \widetilde{Q}^*(s^t, A^t) = \sum_{i=1}^{n} w_i^t \phi_i(s^t, A^t)$$

$$\simeq \sum_{s^{t+1}} \mathbf{P}\big(s^{t+1} \mid s^t, A^t\big) \max_{A^{t+1}} Q^*(s^{t+1}, A^{t+1})$$

$$\bullet\ \widetilde{Q}^*(s^{H+1}) = Q^*(s^{H+1}) = U\big((A^1, x(A^1)), \ldots, (A^H, x(A^H))\big)$$

LSDP Algorithm: application to sampling

1. Computation of $\Phi_i(s^t, A^t)$:

$$\phi_i(s^t, A^t) = \max_{x(i)} \mathbb{P}\big(x(i) \mid x(A^1), \ldots, x(A^{t-1})\big)$$

2. Computation of $\mathbf{P}\big(s^{t+1} \mid s^t, A^t\big) = \mathbb{P}\big(x(A^t) \mid x(A^1), \ldots, x(A^{t-1})\big)$

3. Computation of $U(s^{H+1}) = U\big((A^1, x(A^1)), \ldots, (A^H, x(A^H))\big)$

$$= \sum_{i=1}^{n} \max_{x(i)} \mathbb{P}\big(x(i) \mid x(A^1), \ldots, x(A^H)\big)$$

LSDP Algorithm: application to sampling

1. Computation of $\Phi_i(s^t,A^t)$:

$$\phi_i(s^t, A^t) = \max_{x(i)} \mathbb{P}\big(x(i) \mid x(A^1), \ldots, x(A^{t-1})\big)$$

2. Computation of $\mathbf{P}\left(s^{t+1} \mid s^t, A^t\right) = \mathbb{P}\big(x(A^t) \mid x(A^1), \ldots, x(A^{t-1})\big)$

3. Computation of $U(s^{H+1}) = U\big((A^1, x(A^1)), \ldots, (A^H, x(A^H))\big)$

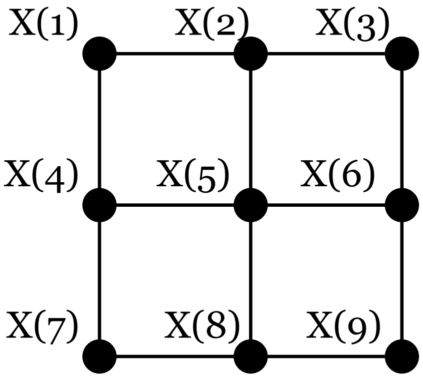
$$= \sum_{i=1}^{n} \max_{x(i)} \mathbb{P}\big(x(i) \mid x(A^1), \ldots, x(A^H)\big)$$

- We fix $|A^t|=1$ and use the approximation:

$$\mathbb{P}\big(x(i) \mid x(A^1), \ldots, x(A^t)\big) \simeq \mathbb{P}^{BP}\big(x(i)\big) + \left[\sum_{j=1}^{t} \mathbb{P}^{BP}\big(x(i) \mid x(A^j)\big) - \mathbb{P}^{BP}\big(x(i)\big)\right]$$

Experiments

Experiments



- R**egular grid** with first order neighbourhood.

•X(i) are **binary** variables.

•ℙ is a Potts model with β=0.5

$$\mathbb{P}\big(x(1),\ldots,x(n)\big) \propto \exp\left(\sum_{(i,j)\in E} \beta eq\big(x(i),x(j)\big)\right)$$

- Simple cost: observation of each variale cost 1

Experiments

- Comparison between:
  - ➢ Random policy
  - ➢BP-max heuristic: at each time step observed variable

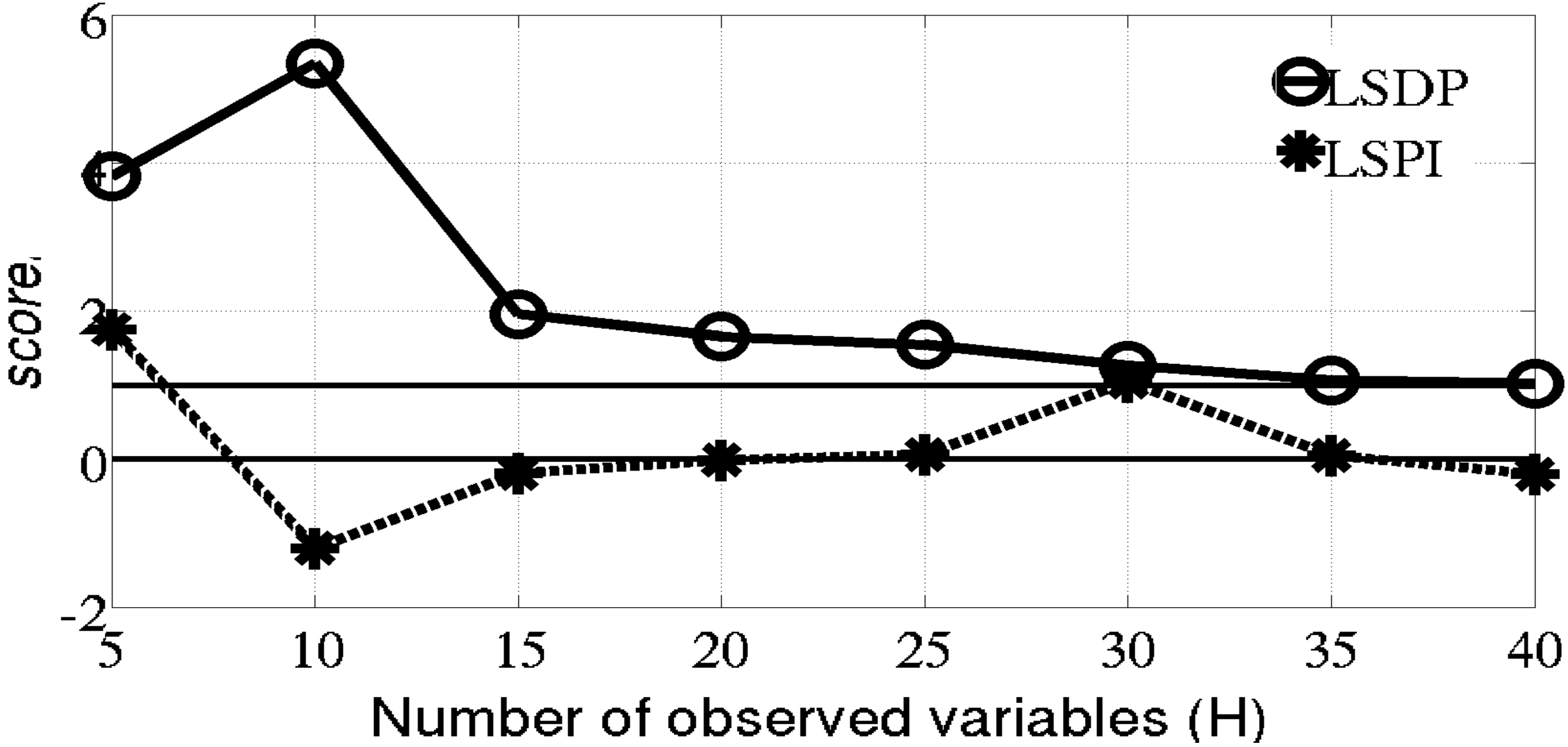
$$A^t = \operatorname*{argmin}_{i=1,\ldots,n} \left(\max_{x(i)} \mathbb{P}^{BP}(x(i) \mid x(A^1), \ldots, x(A^{t-1})) \right)$$

  - ➢LSPI policy   “ common reinforcement learning algorithm”
  - ➢LSDP policy

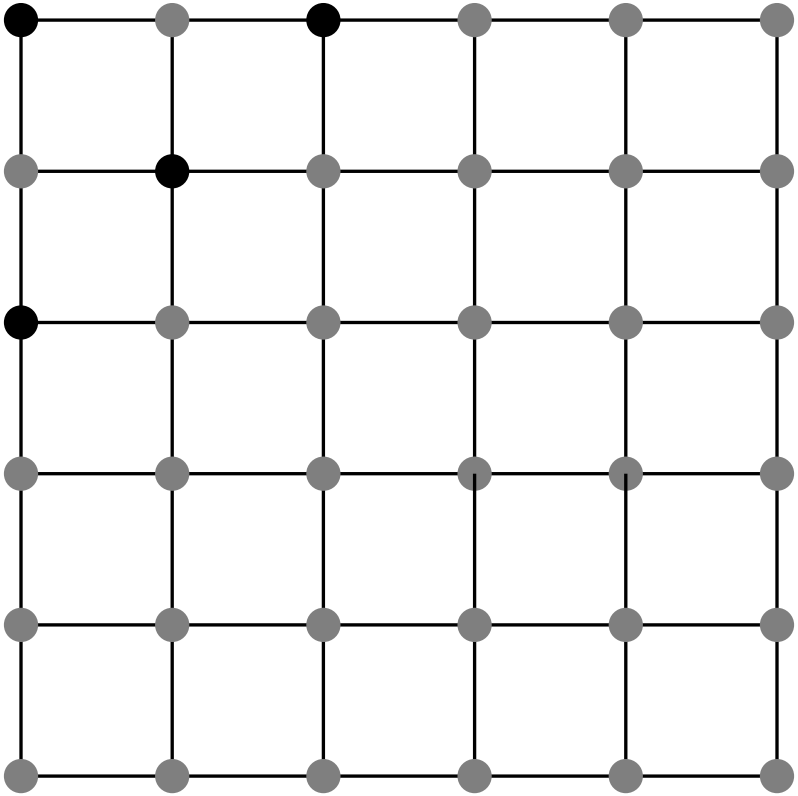
- using score:

$$score(\delta) = \frac{\widetilde{V}(\delta) - \widetilde{V}(\delta_R)}{|\widetilde{V}(\delta_{BP-max}) - \widetilde{V}(\delta_R)|}$$

Experiment: 100 variables (n=100)

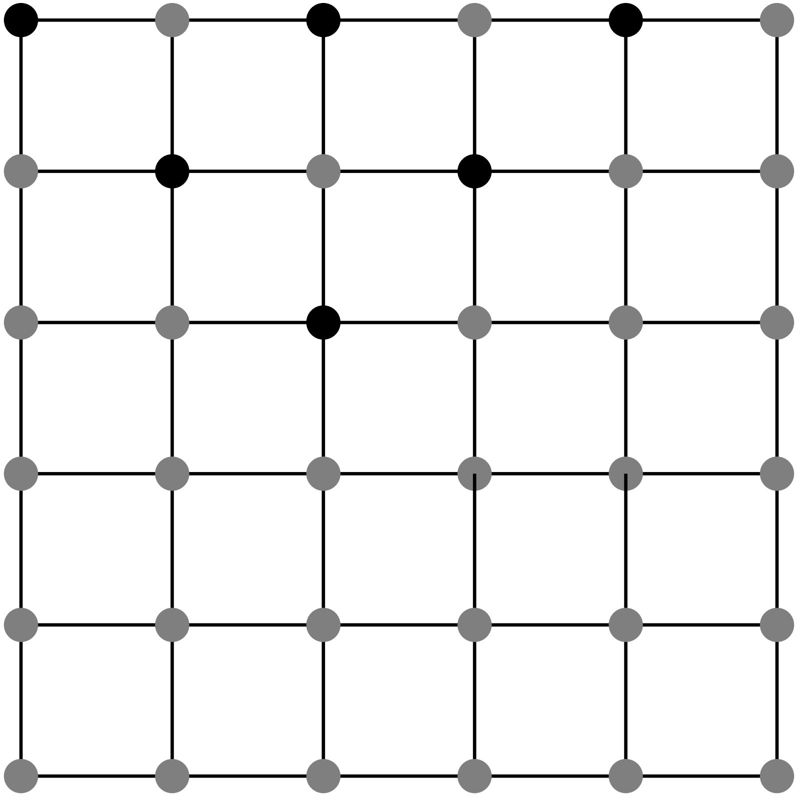


Experiment: 100 variables - constraint move



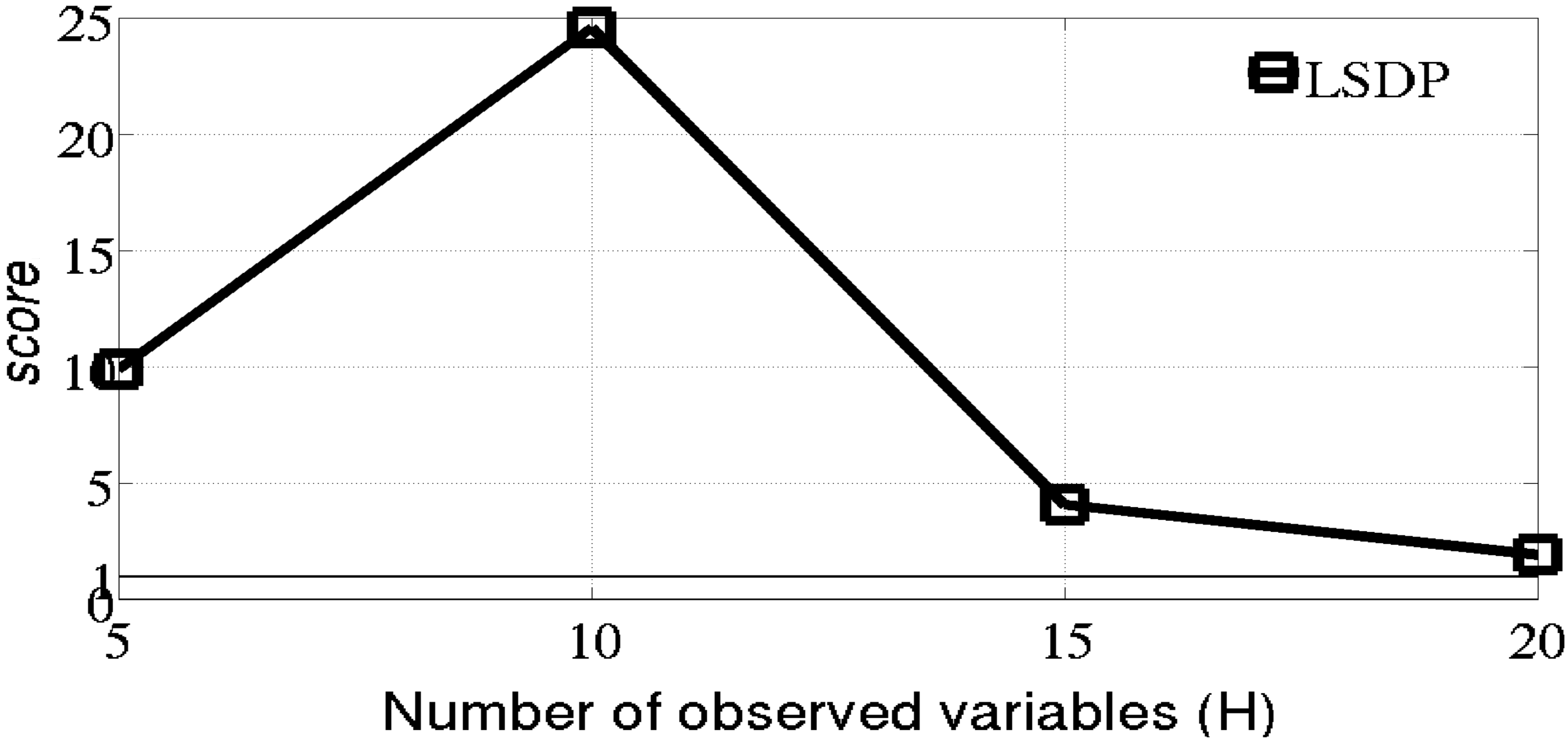
- Allowed to visit second ordre neighbourood only !

Experiment: 100 variables - constraint move



- Allowed to visit second ordre neighbourood only !

Experiment: 100 variables - constraint move



Conclusions

- An adapted framework for adaptive sampling in discrete random variables

- LSDP: a reinforcement learning approach for finding near optimal policy

  ➢ Adaptation of common reinforcement learning algorithm for solving adaptive sampling problem

  ➢ Computation of near optimal policy « off-line »

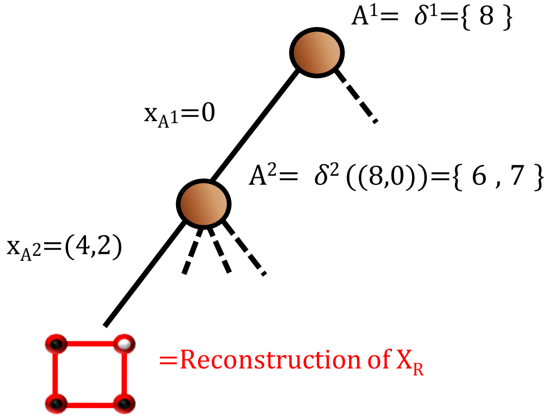
  ➢ Design of new policies that outperform simple heuristics and usual RL method

- Possible application?

  ➢ See next presentation !

THANK YOU!

Reconstruction of X(R) and trajectory value



$A^1 = \delta^1 = \{ 8 \}$

$x_{A^1} = 0$

$A^2 = \delta^2((8,0)) = \{ 6 , 7 \}$

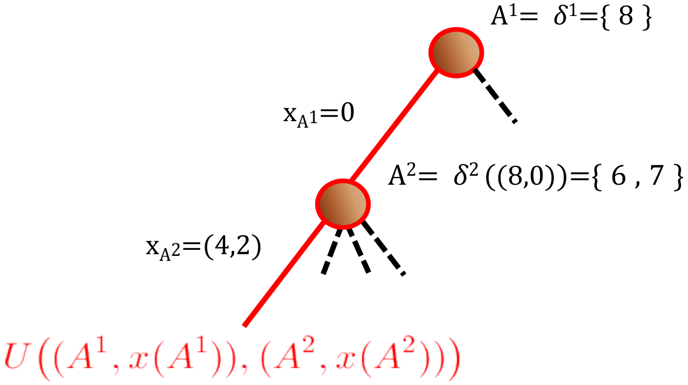
$x_{A^2} = (4,2)$

$=$Reconstruction of X_R

- Maximum Posterior Marginal for reconstruction:

$$\forall r \in R \qquad \widetilde{x}(r) = \arg\max_{x(r)} \mathbb{P}\big(x(r) \mid x(A^1), \ldots, x(A^H)\big)$$

Reconstruction of X(R) and trajectory value


$A^1 = \delta^1 = \{ 8 \}$
$x_{A^1}=0$
$A^2 = \delta^2((8,0)) = \{ 6 , 7 \}$
$x_{A^2}=(4,2)$
$U\big((A^1, x(A^1)), (A^2, x(A^2))\big)$

- Maximum Posterior Marginal for reconstruction:

$$\forall r \in R \qquad \widetilde{x}(r) = \arg\max_{x(r)} \mathbb{P}\big(x(r) \mid x(A^1), \ldots, x(A^H)\big)$$

- Quality of trajectory:

$$U\big((A^1, x(A^1)), \ldots, (A^H, x(A^H))\big) = \sum_{r \in R} \mathbb{P}\big(\widetilde{x}(r) \mid x(A^1), \ldots, x(A^H)\big)$$

$$= \mathbb{E}_{X^*(R)}\left[\ \sum_{r \in R} eq(x^*(r), \widetilde{x}(r)) \mid x(A^1), \ldots, x(A^H)\right]$$