Optimization of sequential-decision problems in health using reinforcement learning

E. Le Pennec
R. Besson - S. Allassonnière

Telecom - 25/11/2021

Outline

Birth and Rare Diseases

A Few Numbers

- 780 000 births/year in France, 5 millions births/year in Europe
- 3 to 4% are affected by at least one congenital anomalies
- Rare diseases: 3 millions patients in France, 30 millions in Europe.

Medical Setting

Prenatal Ultrasound Diagnosis

- **France:** three compulsory ultrasound tests during pregnancy.
- Some classical measures (e.g. Down syndrome).
- **No strict examination protocol.**

Necker Hospital Obstetrician

- Rare disease expertise.
- Among world largest medical database.
- Will to **systematize** their knowledge.

Proposed Tool

Ultrasound as a Sequential Process

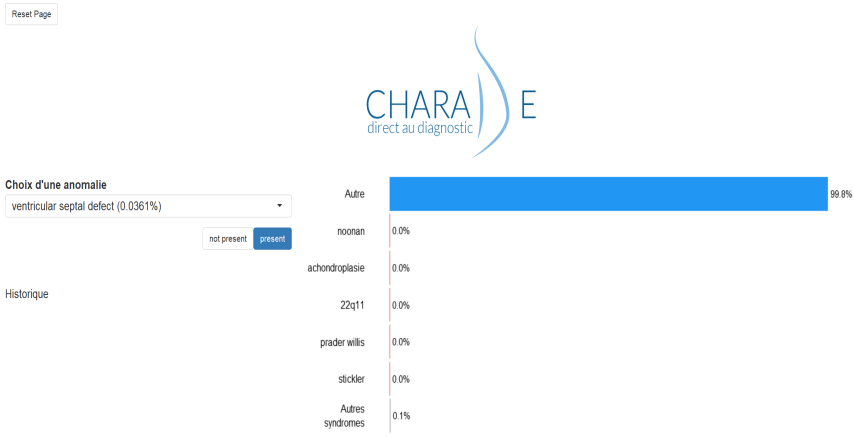
- Ultrasound exam seen as a **sequence of measures**.
- **Goals:**
  - **Reduce the time** required to obtain a diagnosis
  - **Avoid missing** a rare disease.

Diagnosis Assistance Tool

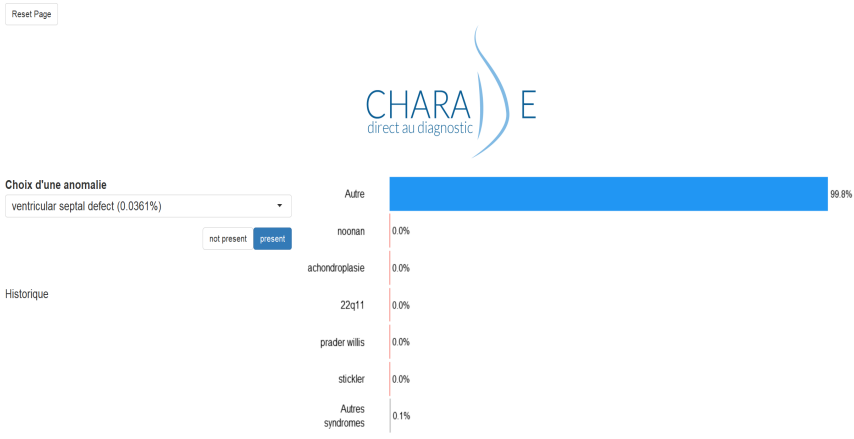
- **Propose** the next measure to make.
- **Show** the current most probable diseases.
- **Easy to use GUI** implemented in **R**!

What's inside this tool?

Charade


Reset Page
CHARADE
direct au diagnostic
Choix d'une anomalie
ventricular septal defect (0.0361%)
not present
present
Historique
Autre
99.8%
noonan
0.0%
achondroplasie
0.0%
22q11
0.0%
prader willis
0.0%
stickler
0.0%
Autres syndromes
0.1%

Charade


Reset Page
CHARADE
direct au diagnostic
Choix d'une anomalie
ventricular septal defect (0.0361%)
not present
present
Historique
Autre
99.8%
noonan
0.0%
achondroplasie
0.0%
22q11
0.0%
prader willis
0.0%
stickler
0.0%
Autres syndromes
0.1%

Outline

Data at Hands

id disease	id symptom	probability of symptom knowing the disease
16	29	0.39
16	136	0.67
16	149	0.50
16	176	0.16
16	181	0.50
16	231	0.75

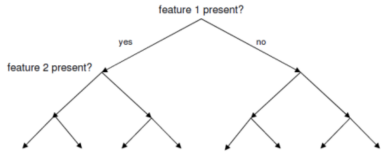
- Rare diseases: very few cases even in the world's largest DB!

Excel Type Dataset

- Expert database build from OrphaData (E. Spaggiari).
- 81 diseases, 202 symptoms (signs visible with ultrasound):
  - Disease probability: $P[D = d_j]$
  - Symptom probability given each disease: $P[S_i = k \mid D = d_j]$.

- Database will be enriched from the future exams.

Our Goals



Medical Goals

- Guide a (not rare-disease expert) sonographer to assess as fast as possible potential diseases.
- Propose her/him the next symptom to check.

Technical Goals

- Build a *good* decision tree (a *good* policy).
- Develop a GUI that can be easily used.

Markov Decision Process

State, Action and Policy

- State: $\mathbb{S} = \{P, A, U\}^{202}$ (presence, absence, not yet looked at) for each symptom.
- Action: $\mathbb{A} = \{1, \ldots, 202\}$ next symptom.
- Policy: $\pi : s \in \mathbb{S} \mapsto a \in \mathbb{A}$ next symptom given the state.

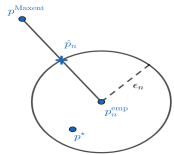
Probabilistic setting

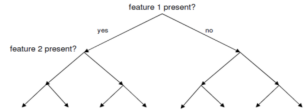
- Natural Markovian modeling: $\mathcal{S}_{t+1}$ depends only on $\mathcal{S}_t$ and a_t!

Markovian Decision Process

- Any strategy π defines a law on $(\mathcal{S}_t)$ starting from $\mathcal{S}_0$.
- Let T be the stopping time before a diagnosis can be posed.
- We need to find $\pi^\star$ such that $\pi^\star(\mathcal{S}_0) = \text{argmin}_\pi \, \mathbb{E}[T|\mathcal{S}_0]$!

Problems to be Solved





Environment Learning with Maximum Entropy Principle

- We have $P[S_i \mid D]$, but we need to know $P[S_{i_1}, ..., S_{i_K} \mid D]$.
- We need to take into account future exams.
- **Idea:** add some expert knowledge and maximize uncertainty, interpolate between the expert model and the data.

- Yields a simulator rather than the MDP transition proba...

Diagnostic Strategy Optimization by Reinforcement Learning

- Find a policy that allows to detect the disease while minimizing the average duration.
- **Idea:** recast the problem as a planning issue and find the optimal strategy.

Diagnostic Strategy Optimization

Diagnostic Strategy Optimization.

- Find a policy that allows to detect the disease while minimizing the average duration.

Measure of Performance

- Number of questions before being able to diagnose a disease.

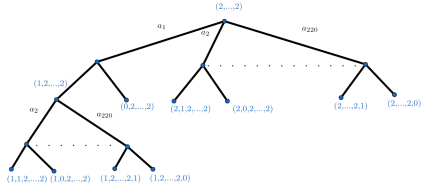
Alternative Formulations

- Trade-off: cost of misdiagnosis/cost of medical tests to perform.
- Reach the lowest uncertainty under fixed budget constraint (time, money).

Non Adversarial Game

- The disease and symptoms do not change during the exam.
- Strategy: given what has been seen, what is the next symptom to look at?

Stochastic Shortest Path



Stochastic Shortest Path

- T is a stooping time at some *final states*
- How to minimize the expectation of T?

Final States

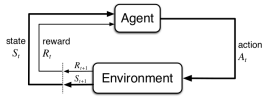
- Entropy based criterion: $H(D \mid \mathcal{S}) \leq \epsilon$

MDP

- Rewards: $\forall \mathcal{S}_t, a_t,\ r(\mathcal{S}_t, a_t) = -1$

Outline

Reinforcement Learning



Reinforcement Learning Setting

- **Env.:** provides a reward and a new state for any action.
- **Agent policy π:** choice of an action A_t from the state S_t.
- **Total reward:** (discounted) sum of the rewards.

Questions

- **Policy evaluation:** how to evaluate the expected reward of a policy knowing the environment?
- **Planning:** how to find the best policy knowing the environment?
- **Reinforcement Learning:** how to find the best policy without knowing the environment?

Outline

The Agent-Environment Interface

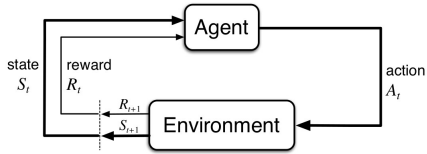


Figure 3.1: The agent–environment interaction in a Markov decision process.

MDP

- At time step $t \in \mathcal{N}$:
  - State $S_t \in \mathcal{S}$: representation of the environment
  - Action $A_t \in \mathcal{A}(S_t)$: action chosen
  - Reward $R_{t+1} \in \mathcal{R}$: instantaneous reward
  - New state S_{t+1}
- Dynamic entirely defined by

$$\mathbb{P}\left(S_t = s', R_r = r | S_{t-1} = s, A_{t-1} = a\right) = p(s', r|s, a)$$

- Finite MDP: $\mathcal{S}$, $\mathcal{A}$ and $\mathcal{R}$ are finite.

Returns and Episodes

Return

- (Discounted) Return:

$$G_t = \sum_{t'=t+1}^{T} \gamma^{t'} R_{t'}$$

- Recursive property

$$G_t = R_{t+1} + \gamma G_{t+1}$$

- Finiteness if $|R| \leq M$

$$|G_t| \leq \begin{cases} (T - t - +1)M & \text{if } T < \infty \\ M \frac{1}{1-\gamma} & \text{otherwise} \end{cases}$$

- Not well defined if $T = \infty$ and $\gamma = 1$.

Policies and Value Functions

Policy and Value Functions

- Policy: $\pi(a|s)$
- Value function:

$$v_\pi(s) = \mathbb{E}_\pi\left[G_t | S_t = s\right] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \middle| S_t = s\right]$$

- Action value function:

$$q_\pi(s, a) = \mathbb{E}_\pi\left[G_t | S_t = s, A_t = a\right]$$

Two natural problems

- Policy evaluation: compute v_π given π.
- Planning: find $\pi^\star$ such that $v_{\pi^\star}(s) \geq v_\pi(s)$ for all s and π.

- Those objects may not exist in general!
- Can be traced back to the 50's!

Outline

Policy Evaluation by Bellman Backup

Fixed Point Property

- Bellman Equation

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r|s, a) \left[r + \gamma v_\pi(s') \right] = \mathcal{T}_\pi(v_\pi)(s)$$

- Linear equation that can be solved.

Policy Evaluation by Dynamic Programming

- Fixed point iterative algorithm: $v_{k+1}(s) = \mathcal{T}_\pi(v_k)(s)$

- Converge if $T < \infty$ or $\gamma < 1$.

Planning by Policy Improvement

Policy Improvement Property

- If π' is such that $\forall s, q_\pi(s, \pi'(s)) \geq v_\pi(s)$ then $v_{\pi'} \geq v_\pi$.

- ϵ-greedy improvement among ϵ-policy: classical improvement degraded by picking uniformly the action with probability ϵ

Policy Iteration Algorithm

- Compute v_{π_k}
- Greedy update:

$$\begin{aligned}\pi_{k+1}(s) &= \operatorname*{argmax}_a q_{\pi_k}(s, a) \\ &= \operatorname*{argmax}_a \sum_{s', r} p(s', r|s, a) \left(r + \gamma v_{\pi_k}(s')\right)\end{aligned}$$

- If $\pi' = \pi$ after a greedy update $v_{\pi_{k+1}} = v_{\pi_k} = v_*$.

- Convergence in finite time in the finite setting.

Planning by Bellman Backup

Fixed Point Property

- Bellman Equation

$$v_*(s) = \max_a \sum_{s'} \sum_r p(s', r|s, a) \left[r + \gamma v_*(s') \right] = \mathcal{T}_*(v_*)(s)$$

- Linear programming problem that can be solved.

Policy Evaluation by Dynamic Programming

- Iterative algorithm: $v_{k+1}(s) = \mathcal{T}_*(v_k)(s)$

- Converge if $T < \infty$ or $\gamma < 1$.
- Amount to improve the policy after only one step of policy evaluation.

Planning by Bellman Backup

Q-value and enhancement

- Q-value:

$$q_\pi(s,a) = \sum_{s'} \sum_{r} p(s',r|s,a) \left[r + \gamma \sum_{a'} \pi(a'|s') q_\pi(s',a') \right]$$

- Easy policy enhancement: $\pi'(s) = \underset{a}{\operatorname{argmax}}\, q(s,a)$

Fixed Point Property

- Bellman Equation

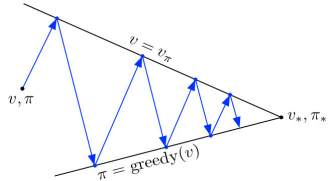
$$q_*(s,a) = \sum_{s'} \sum_{r} p(s',r|s,a) \left[r + \gamma \max_{a'} q_*(s',a') \right] = \mathcal{T}_*(q_*)(s,a)$$

- Linear programming problem that can be solved.

Policy Evaluation by Dynamic Programming

- Iterative algorithm: $q_{k+1}(s,a) = \mathcal{T}_*(q_k)(s,a)$

Generalized Policy Iteration



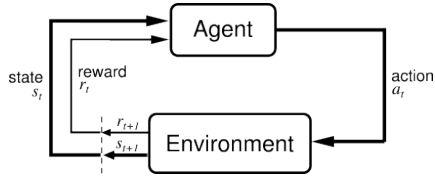
Generalized Policy Iteration

- Consists of two simultaneous interacting processes:
  - one making a value function consistent with the current policy (policy evaluation)
  - one making the policy greedy with respect to the current value function (policy improvement)

- Stabilizes only if one reaches the optimal value/policy pair.
- Asynchronous update are possible, provided every state(/action) is visited infinitely often.
- Very efficient but requires the knowledge of the transition probabilities.

Outline

Reinforcement Learning



Reinforcement Learning - Sutton (98)

- An agent takes actions sequentially, receives rewards from the environment and tries to maximize its long-term (cumulative) reward.

Reinforcement Learning

- MDP setting with cumulative reward.
- Planning problem.
- Environment known only through interaction, i.e. some sequences $\cdots S_t A_t R_{t+1} S_{t+1} A_{t+1} \cdots$.

Monte Carlo

MC Methods

- Back to $v_\pi(s) = \mathbb{E}_\pi [G_t | S_t = s]$.
- Monte Carlo:
  - Play several episodes using policy π.
  - Average the returns obtained after any state s.

- Good theoretical properties provided every state is visited asymptotically *infinitely often*.

Extensions

- Extension to off-policy setting (behavior policy $b \neq$ target policy π) with importance sampling.
- Extension to planning with policy improvement steps

- No theoretical results for the last case.
- Need to wait until the end of an episode to update anything...

Bootstrap and TD Prediction

Bootstrap and TD

- Rely on

$$\begin{aligned} v_\pi(s) &= \mathcal{T}_\pi v_\pi(s) \\ &= \mathbb{E}\left[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s\right] \end{aligned}$$

- Temporal Difference: stochastic approximation scheme

$$V(S_t) \leftarrow V(S_t) + \alpha\left(R_{t+1} + \gamma V(S_{t+1}) - V(S_t)\right)$$

- Update occurs at each time step.
- Can be proved to converge (under some assumption on α)!

- Combine the best of Dynamic Programming and MC.
- Can be written in terms of Q:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\left(R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)\right)$$

SARSA and Q Learning

- How to use this principle to obtain the best policy?

SARSA: Planning by Prediction and Improvement (online)

- Update Q following the current policy π

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left(R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t) \right)$$

- Update π by policy improvement.

- May not converge if one use a greedy policy update.

Q Learning: Planning by Bellman Backup (off-line)

- Update Q following the behavior policy b

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left(R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t) \right)$$

- No need to use importance sampling correction for depth 1 update.

- Proof of convergence in both cases.

Variations

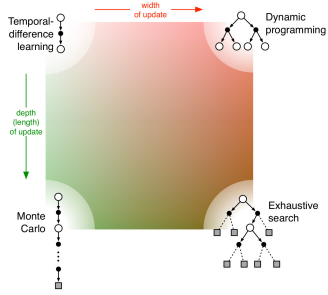


Figure 8.11: A slice through the space of reinforcement learning methods, highlighting the two of the most important dimensions explored in Part I of this book: the depth and width of the updates.

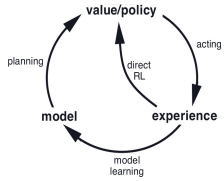
Depth

- Number of steps in the update. x

Width

- Number of states/actions considered at each step.

Planning and Learning



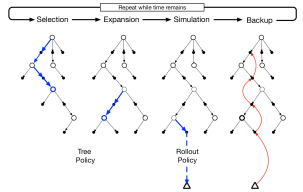


Figure 8.10: Monte Carlo Tree Search. When the environment changes to a new state, MCTS executes as many iterations as possible before an action needs to be selected, incrementally building a tree whose root node represents the current state. Each iteration consists of the four operations **Selection**, **Expansion** (though possibly skipped on some iterations), **Simulation**, and **Backup**, as explained in the text and illustrated by the bold arrows in the trees. Adapted from Chaslot, Bakkes, Szita, and Spronck (2008).

Planning and Models

- Planning can combine a model estimation (DP) and direct learning (RL).

Real Time Planning

- Planning can be made online starting from the current state.

- Curse of dimensionality: methods are hard to use when the cardinality of the states and the actions are large!

Outline

Value Function Approximation

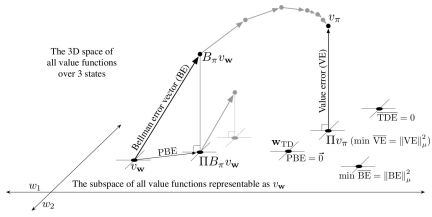
Value Function Approximation

- **Idea:** replace $v(s)$ by a parametric $\hat{v}(s, \boldsymbol{w})$.
- **Issues:**
  - Which approximation functions?
  - How to define the quality of the approximation?
  - How to estimate $\boldsymbol{w}$?

Approximation functions

- Any parametric (or kernel based) approximation could be used.
- Most classical choice:
  - Linear approximation.
  - Deep Neural Nets...

Approximation Quality



- How define when $\hat{v}(\cdot, \boldsymbol{w})$ is close to v_π (or v_*)

Prediction(/Control)

- Prediction objective:

$$\sum_s \mu(s)(v_\pi(s) - \hat{v}(s, \boldsymbol{w}))^2$$

- Bellman Residual:

$$\sum_s \mu(s)(\mathcal{T}_\pi \hat{v}(s, \boldsymbol{w}) - \hat{v}(s, \boldsymbol{w}))^2$$

or its projection...

Online Gradient and Semi-Gradient

Online Prediction

- SGD algorithm on $\boldsymbol{w}$:

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \alpha \left(v_\pi(S_t) - \hat{v}(S_t, \boldsymbol{w}) \right) \nabla \hat{v}(S_t, \boldsymbol{w})$$

- MC approximation (still SGD):

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \alpha \left(G_t - \hat{v}(S_t, \boldsymbol{w}) \right) \nabla \hat{v}(S_t, \boldsymbol{w})$$

- TD approximation (not SGD anymore):

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \alpha \left(R_{t+1} + \gamma \hat{v}(S_{t+1}, \boldsymbol{w}_t) - \hat{v}(S_t, \boldsymbol{w}) \right) \nabla \hat{v}(S_t, \boldsymbol{w})$$

- Deeper or wider scheme possible.

Online Control

- SARSA-like algorithm:
  - Prediction step as previously with the current policy

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \alpha \left(R_{t+1} + \gamma \hat{q}(S_{t+1}, A_{t+1}, \boldsymbol{w}) - \hat{q}(S_t, A_t, \boldsymbol{w}) \right) \nabla \hat{q}(S_t, A_t, \boldsymbol{w})$$

  - ϵ-greedy update of the current policy

Offline Control with Approximation

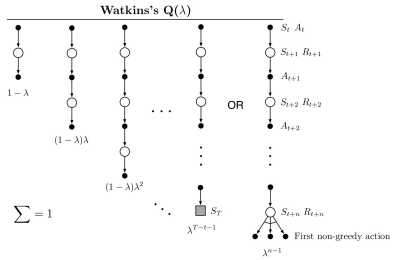


Figure 12.12: The backup diagram for Watkins's Q(λ). The series of component updates ends either with the end of the episode or with the first nongreedy action, whichever comes first.

Offline Control

- Q-Learning like algorithm:

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \alpha \left(R_{t+1} + \gamma \max_a \hat{q}(S_{t+1}, a, \boldsymbol{w}) - \hat{q}(S_t, A_t, \boldsymbol{w}) \right) \times \nabla \hat{q}(S_t, A_t, \boldsymbol{w})$$

  with an arbitrary policy b.

- Deeper formulation using importance sampling possible.

Deadly Triad

Sutton-Barto's Deadly Triad

- **Function Approximation**
- **Bootstrapping**
- **Off-policy training**

Stabilization Tricks

- (Back to policy iteration),
- Memory replay: sample from a set of episodes
- Frozen Q: use the previous weights in the max
- Clip/normalize rewards...

Actor-Critic

- Other approach with a **parametric policy**.

Actor-Critic

- Goal: minimize $J(\pi) = \mathbb{E}_\pi [v(S_t)]$ ($J = \mathbb{E}_\pi [v(S_0)]$ epis.)
- Simultaneous parameterization of the policy π by $\boldsymbol{\theta}$ (actor) and a value function v by $\boldsymbol{w}$ (critic)
- Update formula based on the policy gradient theorem:
$$\nabla_{\boldsymbol{\theta}} J(\pi) = \mathbb{E} [(Q^\pi(S_t, A_t) - C(S_t)) \nabla_{\boldsymbol{\theta}} \log \pi(A_t|S_t])]$$
- Approximate formula:
$$\nabla_{\boldsymbol{\theta}} J(\pi) \simeq \mathbb{E} [(Q(S_t, A_t, \boldsymbol{w}) - C(S_t, \boldsymbol{w})) \nabla_{\boldsymbol{\theta}} \log \pi(A_t|S_t])]$$
- REINFORCE: gradient descent for policy and MC estimate of Q function (with $C(S_t)$ the average return so far).
- AC: gradient descent for policy and TD estimate of Q (and $C = V$).

- Online formulation but can be adapted to offline.

Outline

Numerical Experiments

Naive approach: Breiman CART

- Greedy policy that optimizes the expectation of next step entropy.

Baseline: Actor-critic with REINFORCE

- Linearly parametrized policy using next step entropy expectation and other simple features

Deep Q-Learning

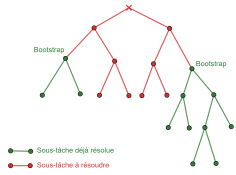
- Q-Learning with Neural Networks.

- Nothing specific for the first two approaches...

Very High Dimension Case!

Issues

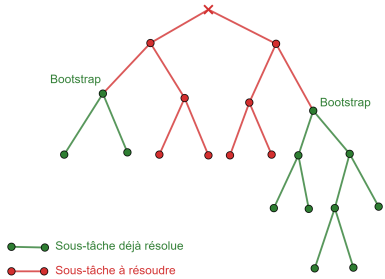
- DQN is unstable with TD in our setting (too slow to backpropagate the rewards?)
- Much better results using MC!
- Still hard to optimize everything from the beginning!



Dimension Reduction Trick

- State space partitioning to solve several smaller sub-problems.

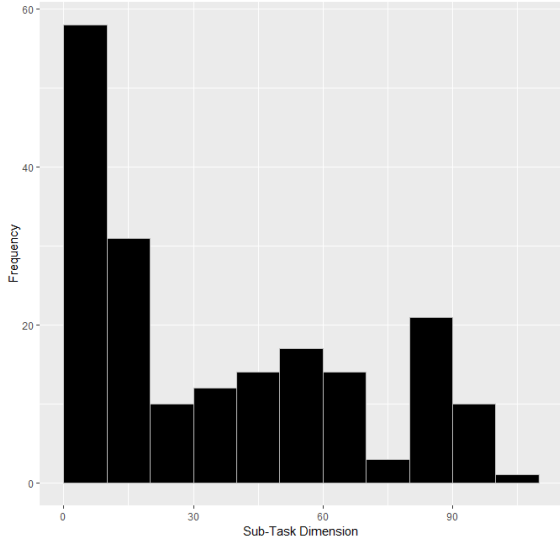
State Partition and MC



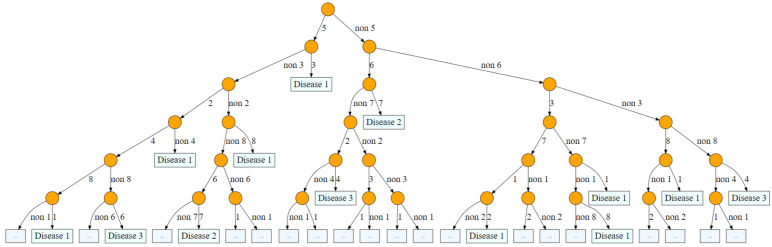
- Partition obtained by solving the problem starting from an anomaly and falling back to previously computed strategy as soon as one reach a common state.
- Similar to an n-step bootstrapping!
- Works well with MC as n is not too large.

Subtask Dimension

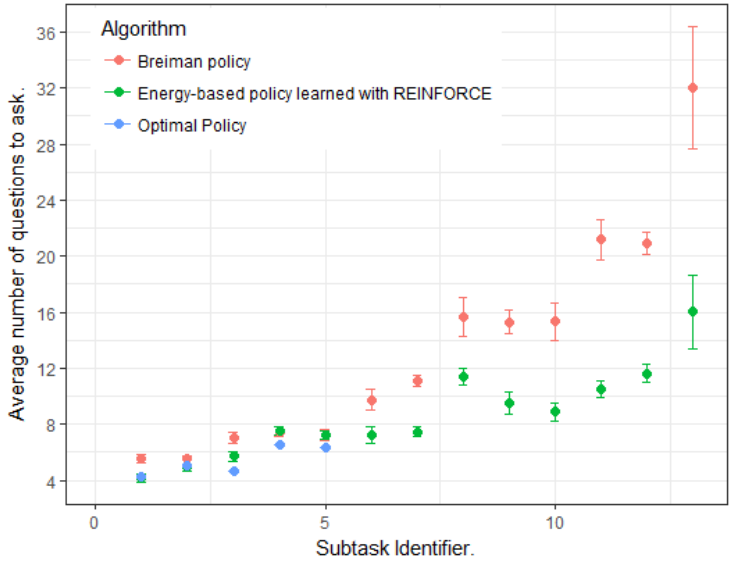
Back to Prenatal Ultrasound



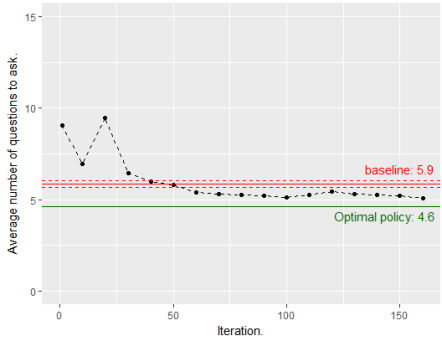
Optimal Decision Tree for a Small Subtask



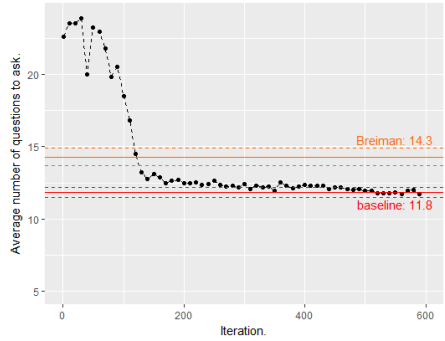
Optimal Policy for Small Subsets?



DQN vs REINFORCE vs Breiman

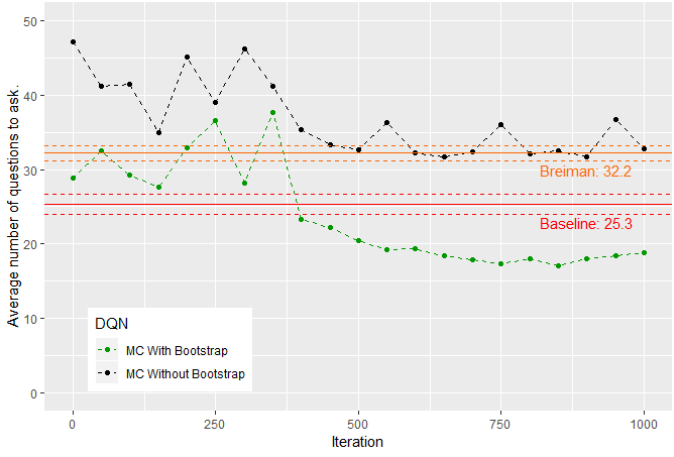


Task Dimension: 10



Task Dimension: 26

DQNs vs REINFORCE vs Breiman



Task Dimension: 70

Challenge: model the combination of abnormalities typical of a rare disease

Diagnostic algorithm:
input: absence / presence of malformations, contextual information
output: Probability of different diagnoses: isolated anomaly (or fortuitous association) vs a basal syndrome
Recommendation algorithm:
input: absence / presence of malformations, contextual information
output: interest score for the remaining anomalies to be consulted

- **Challenges:**
  - **Unstructured data**
  - **No automatic image analysis, therefore a need to list all the relevant variables and make them more reliable or set reasonable mathematical assumptions**
  - **Large dimension**
  - **300 diseases**
  - **600 anomalies**
  - **1000 ultrasound signs**

Creation of a database from Orphanet

Base Orphanet

9000 syndroms

Age of Onset
Antenatal, Neonatal, NA

Prevalence >1/1 million

Description contains “ultrasound” or “prenatal”

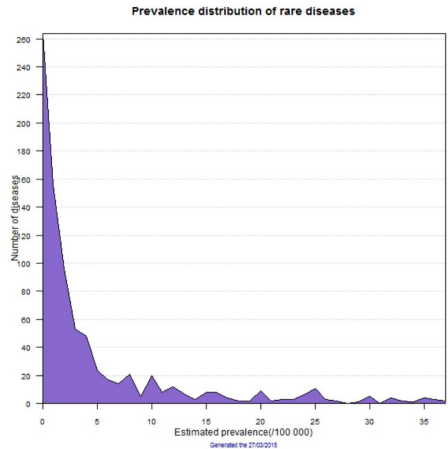
Cross-referencing with Phenotip, Smith and Foetus.net

700 syndroms

Human filter

> 99% prevalence

300 syndroms



Resumption of thesaurus and phenotypic annotation

Thesaurus

- International reference base: HPO (Human Phenotype Ontology)
- Removal of non-diagnosable prenatal abnormalities
- Merging of terms that are too close (example: Retrognathia vs Micrognathia)
- Addition of clean ultrasound signs

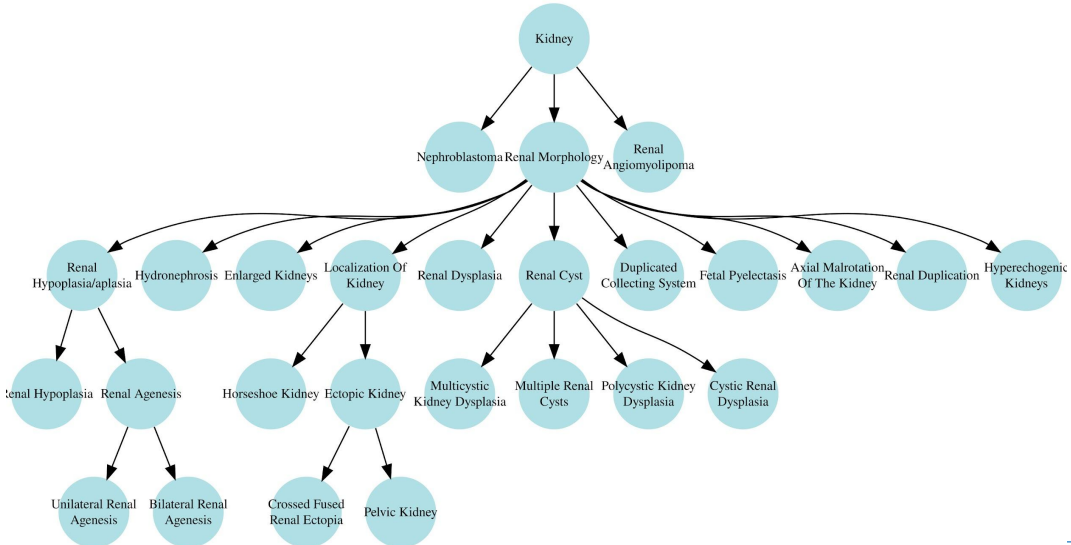
Expert literature review

- Lack of precision sometimes in Orphanet
- Search for ad hoc articles and review of phenotypes and probabilities

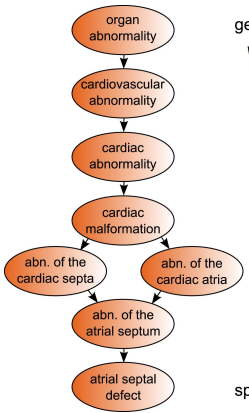
An extract from the database

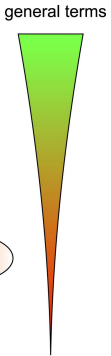
670	Kabuki syndrome	2322	Depressed nasal bridge	5280	Très Fréquent (99-80%)	80 %	T2/T3
671	Kabuki syndrome	2322	Abnormal facial shape	1999	Fréquent (79-30%)	NP	T2/T3
672	Kabuki syndrome	2322	Prominent nasal bridge	426	Occasionnel (29-5%)	21 %	T2/T3
673	Kabuki syndrome	2322	Macrotia	400	Fréquent (79-30%)	70 %	T2/T3
674	Kabuki syndrome	2322	Atrial septal defect	1631	Occasionnel (29-5%)	21 %	T2/T3
675	Kabuki syndrome	2322	Ventricular septal defect	1629	Occasionnel (29-5%)	21 %	T2/T3
676	Kabuki syndrome	2322	Clinodactyly of the 5th finger	4209	Fréquent (79-30%)	60 %	T2/T3
677	Kabuki syndrome	2322	Cleft palate	175	Fréquent (79-30%)	60 %	T2/T3
678	Silver-Russell syndrome	813	Trigonocephaly	243	Très Fréquent (99-80%)	94 %	T2/T3
679	Silver-Russell syndrome	813	Downturned corners of mouth	2714	Fréquent (79-30%)	50 %	T2/T3
680	Silver-Russell syndrome	813	Rocker bottom foot	1838	Fréquent (79-30%)	45 %	T2/T3
681	Silver-Russell syndrome	813	Abnormality of male external genitalia	32	Fréquent (79-30%)	40 %	T2/T3
682	Silver-Russell syndrome	813	Toe syndactyly	1770	Fréquent (79-30%)	30 %	T2/T3
683	Femur-fibula-ulna complex	2019	Aplasia/Hypoplasia of the ulna	6495	Fréquent (79-30%)	35 %	T2/T3
684	Femur-fibula-ulna complex	2019	Humeroradial synostosis	3041	Fréquent (79-30%)	35 %	T2/T3
685	Femur-fibula-ulna complex	2019	Abnormality of the humerus	3063	Occasionnel (29-5%)	15 %	T1/T2/T3

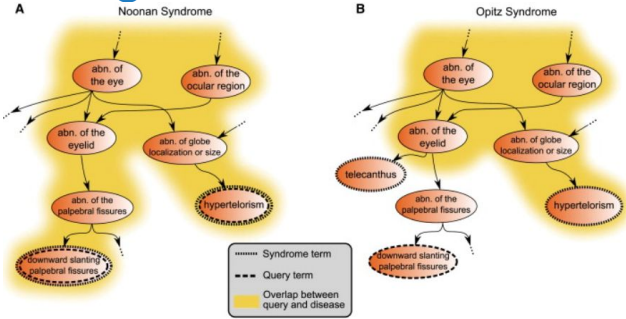
Ontology of anomalies

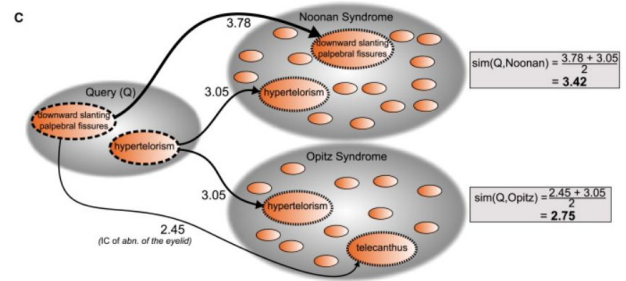


Similarity and ontologies







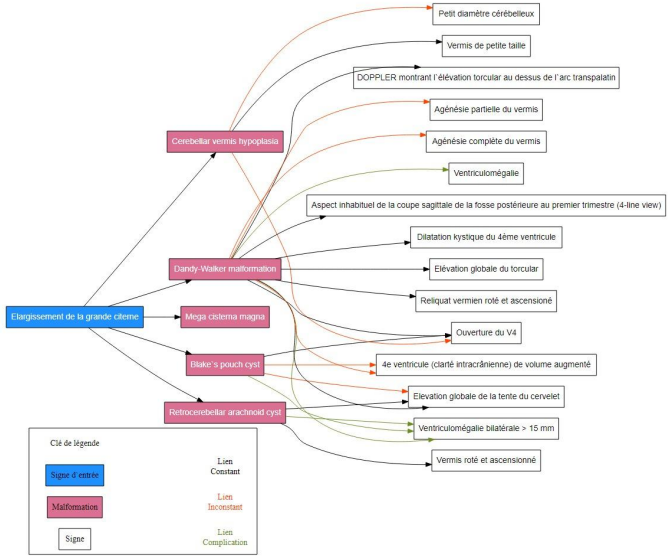


Clinical Diagnostics in Human Genetics with Semantic Similarity Searches in Ontologies, S. Köhler and al, AJHG 2009.

Contextual information

Temporality of variables	Baby's gender	Risk factors

Causal links between malformations



Outline

sonio

Securing prenatal diagnosis

A multidisciplinary and complementary team to make Sonio a standard of care in fetal diagnosis

Cécile Brosset
CEO

Rémi Besson
CSO

Deepak Prakash
CTO

Dagmar Nuber
Business developer

David Amouyal
Product Manager

Marketing / Business
1 Chief Marketing Officer
2 interns

Medical strategy
1 Chief Medical Officer
1 medical consultant
1 intern

Product / Tech
1 data scientist
1 fullstack developer
1 UX designer

General & Adm.
4 freelancers

Regulatory / QMS
2 freelancers

Founding Partners & Scientific Committee

Yves Ville (KOL prenatal diagnostic)
Julien Stirnemann (Clinical validation)
Emmanuel Spaggiari (Database annotation)
Stéphanie Allassonnière (Health AI / image)
Erwan Le Pennec (Datascience)

The first product is the symptom checker

Two patented algorithms

Tailored Decision Tree Optimised next step

Bayesian Network Real-time diagnosis

Curated expert database

HPO, Orphanet, Human expertise

User-friendly interface

Compliant web-architecture

Sonio's symptom checker guides the practitioner in real time during fetal ultrasound

1) In each anatomical area, the practitioner is provided with an exhaustive check-list of items according to guidelines

2) If there is an anomaly, the practitioner can click on the corresponding item

5) Live questions on risk factors or patient history are triggered based on the answers given during the exam

3) The practitioner selects the anomaly he/she has spotted and Sonio suggests other anomalies to check within the same anatomical area

4) Given the previous inputs, Sonio might suggest to check other anatomical areas (white dot)

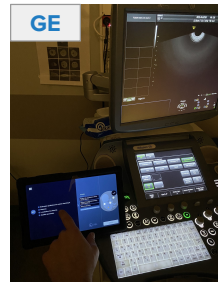
Sonio fits to the hardware ecosystem of the practitioner

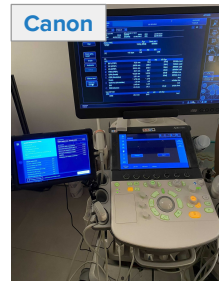
Access Sonio on any browser

or

Lease a specific ergonomic equipment composed of a tablet and an adjustable arm

All you need is a wi-fi or 4G connection





Outline

Next steps

Medical Wandering

- New PhD student (P. Clavier)
- Goal: reduce the time to reach a diagnosis for rare disease...
- but with a much smaller state space.

Actor-Critic Advances

- TRPO: Replace the global goal by a simple local goal

$$J_{\pi_{\text{old}}}(\pi) = \mathbb{E}_{\pi_{\text{old}}}\left[\frac{\pi(A_t|S_t)}{\pi_{\text{old}}(A_t|S_t)} Q_{\pi_{\text{old}}}(S_T, A_t)\right]$$

- PPO: further simplification by clipping.

Distributional RL

- Bellman operator for a given policy is a contraction for a return distribution estimate.
- Allow taking into account risk (or more) into the goal...

Take Away Message

Medical Goals

- Help obstetricians by improving/systematizing ultrasonic diagnostic **(MDP modeling)**
- Guide a (non rare-disease expert) sonographer to assess as fast as possible potential diseases **(first product at Sonio)**

Technical Goals

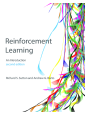
- Build an optimized decision tree:
  - Need to learn the environment **(MaxEnt and data assim.)**
  - Reinforcement learning **(Param. policy and MC vs Deep Q)**
- **Not yet** (theoretical) guarantees.

Take Away Message

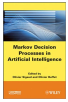
- Reinforcement learning (or MDP) is an interesting tool.
- Formalization requires a true dialog between the mathematicians and the practicians.
- Product available at Sonio.

Outline

References



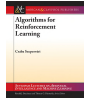
R. Sutton and A. Barto.
Reinforcement Learning, an Introduction (2nd ed.)
MIT Press, 2018



O. Sigaud and O. Buffet.
Markov Decision Processes in Artifical Intelligence.
Wiley, 2010

M. Puterman.
Markov Decision Processes. Discrete Stochastic Dynamic Programming.
Wiley, 2005

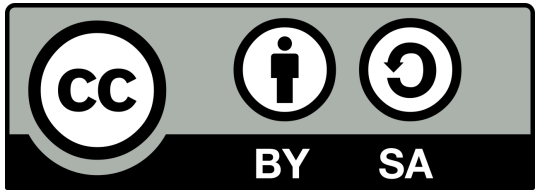
D. Bertsekas and J. Tsitsiklis.
Neuro-Dynamic Programming.
Athena Scientific, 1996



Cs. Szepesvári.
Algorithms for Reinforcement Learning.
Morgan & Claypool, 2010

R. Besson, E. Le Pennec, E. Spaggiari, A. Neuraz, J. Stirnemann, S. Allassonnière, "A Model-Based Reinforcement Learning Approach for a Rare Disease Diagnostic Task", ICAART 20

Licence and Contributors



Contributors

- Main contributor: E. Le Pennec
- Contributor: R. Besson