Bandits, Global Optimization, Active Learning, and Bayesian RL – understanding the common ground

Marc Toussaint

Machine Learning & Robotics Lab – University of Stuttgart
marc.toussaint@informatik.uni-stuttgart.de

Machine Learning Summer School, Tübingen, Sep 2013
Bosch Expertentag Kognitive Systeme, Oct 2013

- Instead of focussing on a single topic, I'll try to emphasize the common underlying problem in sereral topics.

- The perspective taken in this tutorial is simple. All of these problems are eventually Markovian processes in belief space.

  Disclaimer: Whenever I say "optimal" I mean "Bayes optimal" (we always assume having priors $P(\theta)$)

Outline

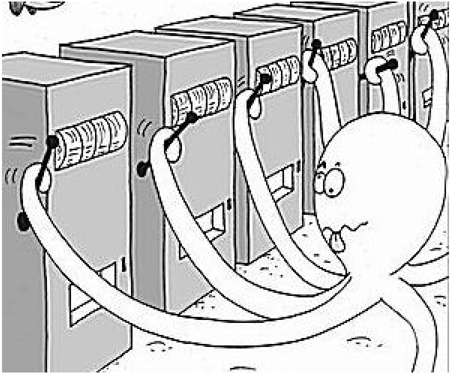
- Problems covered:
  - Bandits
  - Global optimization
  - Active learning
  - Bayesian RL

    (POMDPs)

- Methods covered (interweaved with the above):
  - Belief planning
  - Upper Confidence Bound (UCB)
  - Expected Improvement, probability of improvement
  - Predictive Entropy, Uncertainty Sampling, Shannon Information
  - Bayesian exploration bonus, Rmax
  - "greedy heuristics vs. belief planning"

Bandits

Bandits



- There are n machines.
- Each machine i returns a reward $y \sim P(y; \theta_i)$
  The machine's parameter θ_i is unknown

Bandits

- Let $a_t \in \{1, .., n\}$ be the choice of machine at time t
  Let $y_t \in \mathbb{R}$ be the outcome with mean $\langle y_{a_t} \rangle$

- A policy or strategy maps all the history to a new choice:

$$\pi : \ [(a_1, y_1), (a_2, y_2), ..., (a_{t\text{-}1}, y_{t\text{-}1})] \mapsto a_t$$

- Problem: Find a policy π that

$$\max \left\langle \sum_{t=1}^T y_t \right\rangle$$

  or

$$\max \langle y_T \rangle$$

  or other objectives like discounted infinite horizon $\max \left\langle \sum_{t=1}^\infty \gamma^t y_t \right\rangle$

Exploration, Exploitation

- "Two effects" of choosing a machine:
  - You collect more data about the machine $\rightarrow$ knowledge
  - You collect reward

- Exploration: Choose the next action a_t to $\min \langle H(b_t) \rangle$
- Exploitation: Choose the next action a_t to $\max \langle y_t \rangle$

The Belief State

- "Knowledge" can be represented in two ways:
  - as the full history

$$h_t \; = \; [(a_1, y_1), (a_2, y_2), ..., (a_{t\text{-}1}, y_{t\text{-}1})]$$

  - as the **belief**

$$b_t(\theta) = P(\theta | h_t)$$

    where θ are the unknown parameters $\theta = (\theta_1, .., \theta_n)$ of all machines

- In the bandit case:
  - The belief factorizes $b_t(\theta) = P(\theta | h_t) = \prod_i b_t(\theta_i | h_t)$
    e.g. for Gaussian bandits with constant noise, $\theta_i = \mu_i$

$$b_t(\mu_i | h_t) = \mathcal{N}(\mu_i | \hat{y}_i, \hat{s}_i)$$

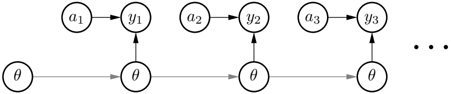
    e.g. for binary bandits, $\theta_i = p_i$, with prior $\text{Beta}(p_i | \alpha, \beta)$:

$$b_t(p_i | h_t) = \text{Beta}(p_i | \alpha + a_{i,t}, \beta + b_{i,t})$$

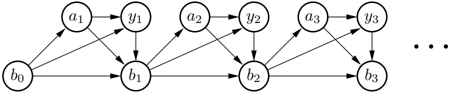
$$a_{i,t} = \sum_{s=1}^{t-1} [a_s \!=\! i][y_s \!=\! 0] \;, \quad b_{i,t} = \sum_{s=1}^{t-1} [a_s \!=\! i][y_s \!=\! 1]$$

The Belief MDP

- The process can be modelled as



or as Belief MDP



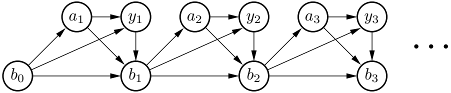
$$P(b'|y,a,b) = \begin{cases} 1 & \text{if } b' = b[a,y] \\ 0 & \text{otherwise} \end{cases} \quad , \quad P(y|a,b) = \int_{\theta_a} b(\theta_a)\ P(y|\theta_a)$$

- The Belief MDP describes a *different* process: the interaction between the information available to the agent (b_t or h_t) and its actions, where *the agent uses his current belief to anticipate observations,* $P(y|a,b)$*.*
- The belief (or history h_t) is all the information the agent has avaiable; $P(y|a,b)$ the "best" possible anticipation of observations. If it acts optimally in the Belief MDP, it acts optimally in the original problem.

Optimality in the Belief MDP $\Rightarrow$ *optimality in the original problem*

Optimal policies via Belief Planning

- The Belief MDP:



$$P(b'|y,a,b) = \begin{cases} 1 & \text{if } b' = b[a,y] \\ 0 & \text{otherwise} \end{cases} \quad , \quad P(y|a,b) = \int_{\theta_a} b(\theta_a)\ P(y|\theta_a)$$

- Belief Planning: Dynamic Programming on the value function

$$\begin{aligned} V_{t\text{-}1}(b_{t\text{-}1}) &= \max_\pi \left\langle \sum_{t=t}^T y_t \right\rangle \\ &= \max_{a_t} \int_{y_t} P(y_t|a_t, b_{t\text{-}1})\ \Big[y_t + V_t(b_{t\text{-}1}[a_t, y_t])\Big] \end{aligned}$$

Optimal policies

- The value function assigns a value (maximal achievable return) to a state of knowledge
- The optimal policy is greedy w.r.t. the value function (in the sense of the $\max_{a_t}$ above)
- Computationally heavy: b_t is a probability distribution, V_t a function over probability distributions

- The term $\int_{y_t} P(y_t|a_t, b_{t\text{-}1}) \left[y_t + V_t(b_{t\text{-}1}[a_t, y_t])\right]$ is related to the *Gittins Index*: it can be computed for each bandit separately.

Example exercise

- Consider 3 binary bandits for $T = 10$.
  - The belief is 3 Beta distributions $\text{Beta}(p_i|\alpha + a_i, \beta + b_i) \quad \rightarrow \quad$ 6 integers
  - $T = 10 \quad \rightarrow \quad$ each integer ≤ 10
  - $V_t(b_t)$ is a function over $\{0, .., 10\}^6$

- Given a prior $\alpha = \beta = 1$,
  a) compute the optimal value function and policy for the final reward and the average reward problems,
  b) compare with the UCB policy.

Greedy heuristic: Upper Confidence Bound (UCB)

1: Initializaiton: Play each machine once
2: **repeat**
3: Play the machine i that maximizes $\hat{y}_i + \sqrt{\frac{2 \ln n}{n_i}}$
4: **until**

$\hat{y}_i$ is the average reward of machine i so far
n_i is how often machine i has been played so far
$n = \sum_i n_i$ is the number of rounds so far

See *Finite-time analysis of the multiarmed bandit problem*, Auer, Cesa-Bianchi & Fischer, Machine learning, 2002.

UCB algorithms

- UCB algorithms determine a **confidence interval** such that

$$\hat{y}_i - \sigma_i < \langle y_i \rangle < \hat{y}_i + \sigma_i$$

  with high probability.
  UCB chooses the upper bound of this confidence interval

- *Optimism in the face of uncertainty*

- Strong bounds on the regret (sub-optimality) of UCB (e.g. Auer et al.)

Further reading

- ICML 2011 Tutorial *Introduction to Bandits: Algorithms and Theory*, Jean-Yves Audibert, Rémi Munos
- *Finite-time analysis of the multiarmed bandit problem*, Auer, Cesa-Bianchi & Fischer, Machine learning, 2002.
- *On the Gittins Index for Multiarmed Bandits*, Richard Weber, Annals of Applied Probability, 1992.

  Optimal Value function is submodular.

Conclusions

- The bandit problem is an archetype for
  - Sequential decision making
  - Decisions that influence knowledge as well as rewards/states
  - Exploration/exploitation

- The same aspects are inherent also in global optimization, active learning & RL

- Belief Planning in principle gives the optimal solution

- Greedy Heuristics (UCB) are computationally much more efficient and guarantee bounded regret

Global Optimization

Global Optimization

- Let $x \in \mathbb{R}^n$, $f : \mathbb{R}^n \to \mathbb{R}$, find

$$\min_x \; f(x)$$

(I neglect constraints $g(x) \leq 0$ and $h(x) = 0$ here – but could be included.)

- Blackbox optimization: find optimium by sampling values $y_t = f(x_t)$
  No access to ∇f or $\nabla^2 f$
  Observations may be noisy $y \sim \mathcal{N}(y \mid f(x_t), \sigma)$

Global Optimization $=$ infinite bandits

- In global optimization $f(x)$ defines a reward for every $x \in \mathbb{R}^n$
  – Instead of a finite number of actions a_t we now have x_t

- Optimal Optimization could be defined as: find $\pi : h_t \mapsto x_t$ that

$$\min \left\langle \sum_{t=1}^{T} f(x_t) \right\rangle$$

  or

$$\min \langle f(x_T) \rangle$$

Gaussian Processes as belief

- The unknown "world property" is the function $\theta = f$
- Given a Gaussian Process prior $GP(f|\mu, C)$ over f and a history

$$D_t \;=\; [(x_1, y_1), (x_2, y_2), ..., (x_{t\text{-}1}, y_{t\text{-}1})]$$

the belief is

$$b_t(f) = P(f \mid D_t) = \mathsf{GP}(f|D_t, \mu, C)$$

$$\mathsf{Mean}(f(x)) = \hat f(x) = \boldsymbol{\kappa}(x)(\boldsymbol{K} + \sigma^2 \mathbf{I})^{\text{-}1} \boldsymbol{y} \qquad \textit{response surface}$$

$$\mathsf{Var}(f(x)) = \hat\sigma(x) = k(x,x) - \boldsymbol{\kappa}(x)(\boldsymbol{K} + \sigma^2 \mathbf{I}_n)^{\text{-}1} \boldsymbol{\kappa}(x) \qquad \textit{confidence interval}$$

- Side notes:
  - Don't forget that $\mathsf{Var}(y^*|x^*, D) = \sigma^2 + \mathsf{Var}(f(x^*)|D)$
  - We can also handle discrete-valued functions f using GP classification

Optimal optimization via belief planning

- As for bandits it holds

$$
\begin{aligned}
V_{t\text{-}1}(b_{t\text{-}1}) &= \max_\pi \Big\langle \textstyle\sum_{t=t}^T y_t \Big\rangle \\
&= \max_{x_t} \int_{y_t} P(y_t|x_t, b_{t\text{-}1}) \; \Big[y_t + V_t(b_{t\text{-}1}[x_t, y_t]) \Big]
\end{aligned}
$$

  $V_{t\text{-}1}(b_{t\text{-}1})$ is a function over the GP-belief!
  If we could compute $V_{t\text{-}1}(b_{t\text{-}1})$ we "optimally optimize"

- I don't know of a minimalistic case where this might be feasible

Greedy 1-step heuristics

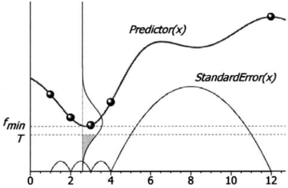


Figure 14. Using kriging, we can estimate the probability that sampling at a given point will 'improve' our solution, in the sense of yielding a value that is equal or better than some target T.

from Jones (2001)

- Maximize Probability of Improvement (MPI)

$$x_t = \underset{x}{\operatorname{argmax}} \int_{-\infty}^{y^*} \mathcal{N}(y|\hat{f}(x), \hat{\sigma}(x))$$

- Maximize Expected Improvement (EI)

$$x_t = \underset{x}{\operatorname{argmax}} \int_{-\infty}^{y^*} \mathcal{N}(y|\hat{f}(x), \hat{\sigma}(x))\ (y^* - y)$$

- Maximize UCB

$$x_t = \underset{x}{\operatorname{argmax}}\ \hat{f}(x) + \beta_t \hat{\sigma}(x)$$

(Often, $\beta_t = 1$ is chosen. UCB theory allows for better choices. See Srinivas et al. citation below.)

From Srinivas et al., 2012:

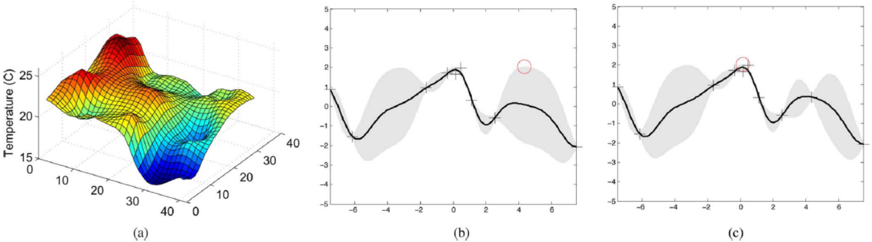


Fig. 2. (a) Example of temperature data collected by a network of 46 sensors at Intel Research Berkeley. (b) and (c) Two iterations of the GP-UCB algorithm. The dark curve indicates the current posterior mean, while the gray bands represent the upper and lower confidence bounds which contain the function with high probability. The "+" mark indicates points that have been sampled before, while the "o" mark shows the point chosen by the GP-UCB algorithm to sample next. It samples points that are either (b) uncertain or have (c) high posterior mean.

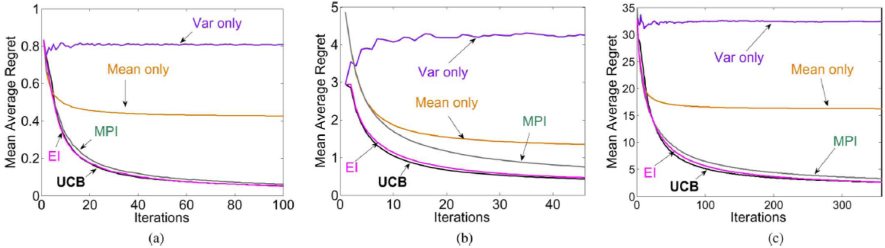


Fig. 6. Mean average regret: GP-UCB and various heuristics on (a) synthetic and (b, c) sensor network data.

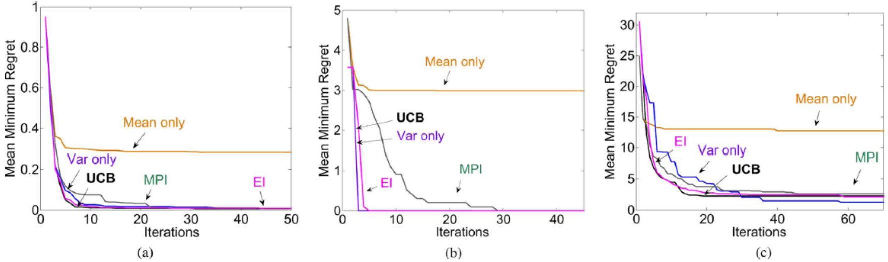


Fig. 7. Mean minimum regret: GP-UCB and various heuristics on (a) synthetic, and (b, c) sensor network data.

Further reading

- Classically, such methods are known as *Kriging*

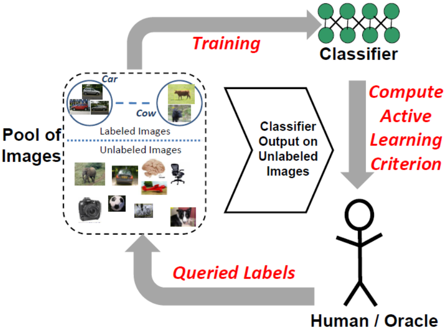
- *Information-theoretic regret bounds for gaussian process optimization in the bandit setting* Srinivas, Krause, Kakade & Seeger, Information Theory, 2012.

- *Efficient global optimization of expensive black-box functions.* Jones, Schonlau, & Welch, Journal of Global Optimization, 1998.
- *A taxonomy of global optimization methods based on response surfaces* Jones, Journal of Global Optimization, 2001.
- *Explicit local models: Towards optimal optimization algorithms*, Poland, Technical Report No. IDSIA-09-04, 2004.

Active Learning

Example

Active learning with gaussian processes for object categorization.
Kapoor, Grauman, Urtasun & Darrell, ICCV 2007.



Active Learning

- In standard ML, a data set $D_t = \{(x_s, y_s)\}_{s=1}^{t\text{-}1}$ is given.
  In active learning, the learning agent sequencially decides on each x_t
  – where to collect data

- Generally, the aim of the learner should be to learn as fast as possible, e.g. minimize predictive error

- Finite horizon T predictive error problem:
  Find a policy $\pi : D_t \mapsto x_t$ that

$$\min \left\langle -\log P(y^*|x^*, D_T) \right\rangle_{y^*, x^*, D_T; \pi}$$

  This also can be expressed as *predictive entropy*:

$$\begin{aligned} \left\langle -\log P(y^*|x^*, D_T) \right\rangle_{y^*, x^*} &= \left\langle -\int_{y^*} P(y^*|x^*, D_T) \ \log P(y^*|x^*, D_T) \right\rangle_{x^*} \\ &= \left\langle H(y^*|x^*, D_T) \right\rangle_{x^*} =: H(f|D_T) \end{aligned}$$

- Find a policy that $\min \left\langle H(f|D_T) \right\rangle_{D_T; \pi}$

Gaussian Processes as belief

- Again, the unknown "world property" is the function $\theta = f$
- We can use a Gaussian Process to represent the belief

$$b_t(f) = P(f \mid D_t) = \mathsf{GP}(f|D_t, \mu, C)$$

Optimal Active Learning via belief planning

- The only difference to global optimization is the reward.
  In active learning it is the predictive entropy: $-H(f|D_T)$

- Dynamic Programming:

$$V_T(b_T) = -H(b_T)\ , \quad H(b) := \langle H(y^*|x^*, b)\rangle_{x^*}$$
$$V_{t\text{-}1}(b_{t\text{-}1}) = \max_{x_t} \int_{y_t} P(y_t|x_t, b_{t\text{-}1})\ V_t(b_{t\text{-}1}[x_t, y_t])$$

- Computationally intractable

Greedy 1-step heuristic

- The simplest greedy policy is 1-step Dynamic Programming:
  Directly maximize immediate expected reward, i.e., minimizes $H(b_{t+1})$.

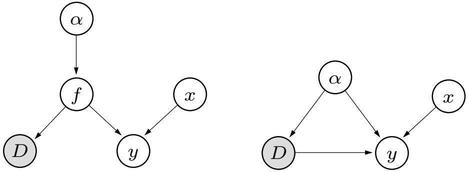
$$\pi : \ b_t(f) \mapsto \operatorname*{argmin}_{x_t} \int_{y_t} P(y_t|x_t, b_t)\ H(b_t[x_t, y_t])$$

- For GPs, you reduce the entropy most if you choose x_t where the current predictive variance is highest:

$$\text{Var}(f(x)) = k(x, x) - \boldsymbol{\kappa}(x)(\boldsymbol{K} + \sigma^2 \mathbf{I}_n)^{\text{-1}} \boldsymbol{\kappa}(x)$$

  This is referred to as *uncertainty sampling*
- Note, if we fix hyperparameters:
  - This variance is *independent* of the observations y_t, only the set D_t matters!
  - The order of data points also does not matter
  - You can pre-optimize a set of “grid-points” for the kernel – and play them in any order

Greedy Active Learning with hyperparameters



- Change the reward function: Minimize expected entropy over α:

$$\underset{x}{\operatorname{argmin}} \int_y P(y|x,D)\ H[p(\alpha|D,x,y)]$$

- This can be rewritten in many ways (adding a constant $H[p(\alpha)]$)

$$-\int_y p(y|x)\ H[p(\alpha|y,x)] - H[p(\alpha)] \tag{1}$$
$$= \int_{y,\alpha} p(y,\alpha|x)\ \log p(\alpha|y,x) - H[p(\alpha)] \tag{2}$$
$$= \int_y p(y|x)\ D\big(p(\alpha|y,x) \,\|\, p(\alpha)\big)\ , \tag{3}$$

- Eq. (2) maximizes the gain in *Shannon Information*
- Eq. (3) maximizes the expected KLD
  (see Chaloner et al.)

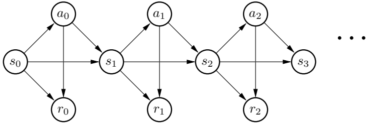
Further reading

- *Active learning literature survey.* Settles, Computer Sciences Technical Report 1648, University of Wisconsin-Madison, 2009.
- *Bayesian experimental design: A review.* Chaloner & Verdinelli, Statistical Science, 1995.
- *Active learning with statistical models.* Cohn, Ghahramani & Jordan, JAIR 1996.
- ICML 2009 Tutorial on *Active Learning*, Sanjoy Dasgupta and John Langford `http://hunch.net/~active_learning/`

Bayesian Reinforcement Learning

Markov Decision Process

- Other than the previous cases, actions now influence a world state



  – initial state distribution $P(s_0)$
  – transition probabilities $P(s'|s,a)$
  – reward probabilities $P(r|s,a)$
  – agent's policy $P(a|s;\pi)$

- Planning in MDPs: Given knowledge of $P(s'|s,a)$, $P(r|s,a)$ and $P(y|s,a)$, find a policy $\pi:\ s_t \mapsto a_t$ that maximizes the discounted infinite horizon return $\langle \sum_{t=0}^\infty \gamma^t r_t \rangle$:

$$V(s) = \max_a \Big[\mathsf{E}(r|s,a) + \gamma \sum_{s'} P(s' \mid s,a)\, V(s') \Big]$$

Model-based Reinforcement Learning

- In *Reinforcement Learning* we do not know the world
  Unknown MDP parameters $\theta = (\theta_s, \theta_{s'sa}, \theta_{rsa})$
  (for $P(s_0), P(s'|s,a), P(r|s,a)$)

- In *model-free* RL, there is no attempt to learn/estimate θ
  - Instead: directly estimate $V(s)$ or $Q(s,a)$
  - *TD, Q-learning*
  - Policy gradients, blackbox policy search, etc

- Basic *model-based* RL computes estimates $\hat\theta$:
  - Exploit: Dynamic Programming with current $\hat\theta$ to decide on next action
  - Explore: e.g., sometimes choose random actions (more on this later)

Bayesian RL: The belief state

- "Knowledge" can be represented in two ways:
  - as the full history

$$h_t \;=\; [(s_0, a_0, r_0), ..., (s_{t\text{-}1}, a_{t\text{-}1}, r_{t\text{-}1}), (s_t)]$$

  - as the belief

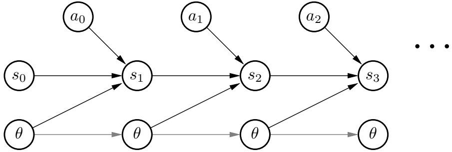
$$b_t(\theta) = P(\theta | h_t)$$

    where θ are the unknown parameters $\theta = (\theta_1, .., \theta_n)$ of all machines

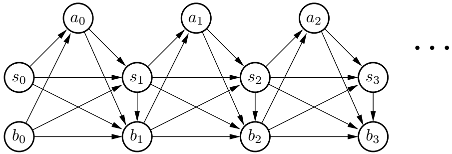
- In the case of discrete MDPs
  - θ are CPTs (conditional probability tables)
  - Assuming Dirichlet priors over CPTs, the exact posterior is a Dirichlet
  - Amounts to counting transitions

Optimal policies

- The process can be modelled as (omitting rewards)



or as Belief MDP



$$P(b'|s',s,a,b) = \begin{cases} 1 & \text{if } b' = b[s',s,a] \\ 0 & \text{otherwise} \end{cases} \quad , \quad P(s'|s,a,b) = \int_\theta b(\theta)\ P(s'|s,a,\theta)$$

$$V(b,s) = \max_a \Big[\mathsf{E}(r|s,a,b) + \textstyle\sum_{s'} P(s'|a,s,b)\ V(s',b') \Big]$$

- Dynamic programming can be approximated (Poupart et al.)

Heuristics

- As with UCB, choose estimators for R^*, P^* that are optimistic/over-confident

$$V_t(s) = \max_a \Big[R^* + \sum_{s'} P^*(s'|s,a)\; V_{t+1}(s')\Big]$$

- Rmax:
  - Choose $R^*(s,a) = \begin{cases} R_{\text{max}} & \text{if } \#_{s,a} < n \\ \hat\theta_{rsa} & \text{otherwise} \end{cases}$
  - Choose $P^*(s'|s,a) = \begin{cases} \delta_{s's^*} & \text{if } \#_{s,a} < n \\ \hat\theta_{s'sa} & \text{otherwise} \end{cases}$
  - Guarantees over-estimation of values, polynomial PAC results!
  - Read about “KWIK-Rmax”! (Li, Littman, Walsh, Strehl, 2011)
- Bayesian Exploration Bonus (BEB), Kolter & Ng (ICML 2009)
  - Choose $P^*(s'|s,a) = P(s'|s,a,b)$ integrating over the current belief $b(\theta)$ (non-over-confident)
  - But choose $R^*(s,a) = \hat\theta_{rsa} + \frac{\beta}{1+\alpha_0(s,a)}$ with a hyperparameter $\alpha_0(s,a)$, over-estimating return

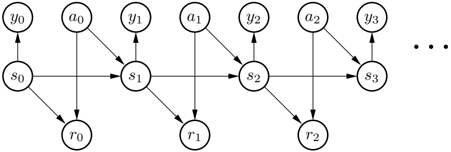
Further reading

- ICML-07 Tutorial on Bayesian Methods for Reinforcement Learning
  `https://cs.uwaterloo.ca/~ppoupart/ICML-07-tutorial-Bayes-RL.html`
  Esp. part 3: Model-based Bayesian RL (Pascal Poupart); and the methods cited on slide 22
- *Optimal learning: Computational procedures for Bayes-adaptive Markov decision processes.* Duff, Doctoral dissertation, University of Massassachusetts Amherst, 2002.
- *An analytic solution to discrete Bayesian reinforcement learning.* Poupart, Vlassis, Hoey, & Regan (ICML 2006)
- KWIK-Rmax: *Knows what it knows: a framework for self-aware learning.* Li, Littman, Walsh & Strehl, Machine learning, 2011.
- Bayesian Exploration Bonus: *Near-Bayesian exploration in polynomial time.* Kolter & Ng, ICML 2009.
- The “interval exploration method” described in *Reinforcement learning: A survey*. Kaelbling, Littman & Moore, arXiv preprint cs/9605103, 1996.

POMDPs

POMDPs

- A belief MDP is a special case of a POMDP



  – initial state distribution $P(s_0)$
  – transition probabilities $P(s'|s,a)$
  – observation probabilities $P(y|s)$
  – reward probabilities $P(r|s,a)$

- Embedding a Belief MDP in a POMDP:

  $s_{\text{POMDP}} \leftarrow (\theta, s)_{\text{BeliefMDP}}$

  $y_{\text{POMDP}} \leftarrow s_{\text{BeliefMDP}}$

Optimal policies

- Again, the value function is a function over the belief

$$V(b) = \max_a \Big[R(b,s) + \gamma \textstyle\sum_{b'} P(b'|a,b)\, V(b') \Big]$$

- Sondik 1971: V is piece-wise linear and convex: Can be described by m vectors $(\alpha_1, .., \alpha_m)$, each $\alpha_i = \alpha_i(s)$ is a function over discrete s

$$V(b) = \max_i \textstyle\sum_s \alpha_i(s) b(s)$$

  Exact dynamic programming possible, see Pineau et al., 2003

Approximations & Heuristics

- Point-based Value Iteration (Pineal et al., 2003)
  – Compute $V(b)$ only for a finite set of belief points

- Discard the idea of using belief to “aggregate” history
  – Policy directly maps history (window) to actions
  – Optimize finite state controllers (Meuleau et al. 1999, Toussaint et al. 2008)

Further reading

- *Point-based value iteration: An anytime algorithm for POMDPs.* Pineau, Gordon & Thrun, IJCAI 2003.
- The standard references on the "POMDP page" `http://www.cassandra.org/pomdp/`
- *Bounded finite state controllers.* Poupart & Boutilier, NIPS 2003.
- *Hierarchical POMDP Controller Optimization by Likelihood Maximization.* Toussaint, Charlin & Poupart, UAI 2008.

Discussion

3 points to make

Point 1: Common ground

What bandits, global optimization, active learning, Bayesian RL & POMDPs share

- Sequential decisions
- Markovian w.r.t. belief
- Decisions influence the knowledge as well as rewards/states
- Sometimes described as “exploration/exploitation problems”

Point 2: Optimality

- In all cases, belief planning would yield optimal solutions
  $\rightarrow$ Optimal Optimization, Optimal Active Learning, etc...
- Even if it may be computationally infeasible, it is important to know conceptually
- Optimal policies “navigate through belief space”
  - This automatically implies/combines “exploration” and “exploitation”
  - There is no need to explicitly address “exploration vs. exploitation” or decide for one against the other. Policies that maximize the single objective of future returns will automatically do this.

Point 3: Greedy (1-step) heuristics

- Also the optimal policy is greedy – w.r.t. the value function!
- “Greedy heuristics” replace the value function by something simpler and more direct to compute, typically 1-step criteria
  - UCB
  - Probability of Improvement, Expected Improvement
  - Expected immediate reward, expected predictive entropy
- Typically they reflect *optimism in the face of uncertainty*
- Regret bounds for UCB on bandits and optimization (Auer et al.; Srinivas et al.)
- Theory on submodularity very stongly motivates greedy heuristics

- In RL: Optimism w.r.t. θ, but planning w.r.t. s
  - Bayesian Exploration Bonus (BEB), Rmax, interval exploration method

Thanks

for your attention!