REINFORCEMENT LEARNING AND MATRIX COMPUTATION

Vivek Borkar

IIT, Mumbai

Feb. 7, 2014, ICDCIT 2014, Bhubaneshwar

Q-learning (Watkins)

Recall 'finite state finite action' *Markov decision process*:

- $\{X_n\}$ a random process taking values in a finite *state space* $S := \{1, 2, \cdots, s\}$,

- governed by a *control process* $\{Z_n\}$ taking values in a finite *action space* A,

- with transition mechanism:

$$P(X_{n+1} = j | X_m, Z_m, m \leq n) = P(X_{n+1} = j | X_n, Z_n)$$

$$= p(j | X_n, Z_n).$$

Applications in communications, control, operations research, finance, robotics, $\cdots$

Discounted cost:

$$J(\{Z_n\}) := E\left[\sum_{m=0}^{\infty} \beta^m k(X_m, Z_m) | X_0 = i\right]$$

where:

$k : S \times A \mapsto \mathcal{R}$ is the 'cost per stage' function, and,

$0 < \beta < 1$ is the discount factor

(e.g., $\beta = \frac{1}{1+r}$ where $r > 0$ is the interest rate).

Define the *value function* $V : S \mapsto \mathcal{R}$ as

$$V(i) := \min_{\{Z_n\}} J_i(\{Z_n\}).$$

This is the 'minimum cost to go' and satisfies the dynamic programming principle:

Min. cost to go = min(cost of current stage + min. cost to go from next stage on).

$\Longrightarrow$ the *Dynamic Programming* (DP) equation:

$$V(i) = \min_{u \in A} Q(i,u) := \min_{u \in A} \left[k(i,u) + \beta \sum_j p(j|i,u) V(j) \right].$$

Here $v(i) := \text{argmin}_A Q(i, \cdot)$ is the optimal *stationary Markov* policy: $Z_n := v(X_n)\ \forall n$ is optimal

'stationary' : no explicit dependence on time.

'Markov': a function of the current state alone, no need to remember the past.

Analogously, we have 'Q-DP' equation

$$Q(i,u) = k(i,u) + \beta \sum_j p(j|i,u) \min_{u \in A} Q(i,a).$$

Thus solution of the DP equation or the Q-DP equation $\Longleftrightarrow$ solution of the control problem.

This prompts the search for computational schemes to solve these.

Value iteration: recursive solution scheme given by

$$V^{n+1}(i) = \min_u \left[k(i,u) + \beta \sum_j p(j|i,u) V^n(j) \right].$$

Similarly, *Q-value iteration*

$$Q^{n+1}(i,u) = k(i,u) + \sum_j p(j|i,u) \min_a Q(j,a).$$

Disadvantage: bigger *curse of dimensionality*

Advantage: Averaging with respect to $p(\cdot|\cdot)$ is now *outside* of the nonlinearity (i.e., minimization)

$\Longrightarrow$ makes it amenable to *stochastic approximation*.

Stochastic Approximation (Robbins and Monro)

To solve $h(x) = 0$ given noisy observations $h(x)+$ noise, do:

$$x_{n+1} = x_n + a(n)\left[h(x_n) + M_{n+1}\right],\ n \geq 0,$$

where h is 'nice' and $\{M_n\}$ uncorrelated with past (i.e., $E[M_{n+1}|$ past till $n] = 0$).

Need: $\sum_n a(n) = \infty,\ \sum_n a(n)^2 < \infty$.

Usually, the original iteration is of the form

$$x_{n+1} = x_n + a(n) f(x_n, \zeta_{n+1}),\ n \geq 0,$$

where $\{\zeta_n\}$ are independent and identically distributed random variables. This can be put in the above form by defining

$$h(x) \ := \ E\left[f(x, \xi)\right],\ \xi \approx \zeta_n,$$

$$M_{n+1} \ := \ f(x_n, \zeta_{n+1}) - h(x_n),\ n \geq 0.$$

This will usually be the scenario in the problems we consider.

ODE approach (Derevitskii-Fradkov-Ljung) $\Longrightarrow$ this is a noisy discretization of the ODE (ordinary differential equation)

$$\dot{x}(t) = h(x(t))$$

Under suitable conditions, the stochastic approximation scheme has the same asymptotic behavior as the ODE with probability 1.

Thus ODE convergence to an equilibrium $x^* \Longrightarrow$

$x_n \to x^*$ w.p. 1.

Caveats:

- More generally, multiple equilibria or more general limit sets.
- Need stability guarantee: $\sup_n \|x_n\| < \infty$ w.p. 1.
- Problems of asynchrony.

Q-Learning: For $\xi^{iu}_{n+1} \approx p(\cdot|i,u)$,

$$Q^{n+1}(i,u) =$$
$$(1-a(n))Q^n(i,u) + a(n)\left[k(i,u) + \beta \min_a Q^n(\xi^{iu}_{n+1}, a)\right].$$

More common to use a single simulation run $\{X_n, Z_n\}$ with 'persistent excitation'* and do:

$$\begin{aligned} Q^{n+1}(i,u) \;=\; & Q^n(i,u) + a(n) I\{X_n = i, Z_n = u\} \times \\ & \left[k(i,u) + \beta \min_a Q^n(\xi^{iu}_{n+1}, a) - Q^n(i,u)\right]. \end{aligned}$$

*some randomization to ensure adequate exploration

Limiting ODE has the form

$$\dot{Q}(t) = F(Q(t)) - Q(t)$$

where $F : \mathcal{R}^{|S|\times|A|} \mapsto \mathcal{R}^{|S|\times|A|}$ is a 'contraction':

$\|F(x) - F(y)\|_\infty \leq \beta\|x - y\|_\infty$.

Then F has a unique 'fixed point' $Q^* : F(Q^*) = Q^*$, i.e., the desired solution.

Moreover, $Q(t) \to Q^*$, implying $Q^n \to Q^*$ w.p. 1.

Other costs:

1. finite horizon cost $E[\sum_{m=0}^{N} k(X_m, Z_m) + h(X_N)]$, with the DP equation

$$\begin{aligned} V(i,m) &= \min_{u \in A}(k(i,u) + \sum_j p(j|i,u)V(j,m+1)), m < N, \\ V(i,N) &= h(i), \quad i \in S. \end{aligned}$$

2. average cost $\limsup_{N\uparrow\infty} \frac{1}{N} \sum_{m=0}^{N-1} E[k(X_m, Z_m)]$, with the DP equation

$$V(i) = \min_{u \in A}(k(i,u) - \kappa + \sum_j p(j|i,u)), \ i \in S,$$

3. risk-sensitive cost $\limsup_{N\uparrow\infty} \frac{1}{N} \log E\left[e^{\sum_{m=0}^{N-1} k(X_m, Z_m)}\right]$,

   with the DP equation

$$V(i) = \min_{u \in A} \frac{e^{k(i,u)} \sum_j p(j|i,u) V(j)}{\lambda}, \quad i \in S.$$

   (a nonlinear eigenvalue problem).

In what follows, we extend this methodology to three other problems not arising from Markov decision processes.

Averaging

Gossip algorithm for averaging: 'DeGroot model'

$$x_{n+1} = (1-a)x_n + aPx_n,\ n \geq 0.$$

$P := [[p(j|i)]]$ a $d \times d$ irreducible stochastic matrix with stationary distribution π (i.e., $\pi P = \pi$) and $0 < a \leq 1$. Then

$$x_n \to \sum_i p(i)x_0(i).$$

Traditional concerns: design P (usually doubly stochastic so that π is uniform) so as to optimize the convergence rate (Boyd et al).

Stochastic version: At time n, node i polls a neighbor $\xi_n(i) = j$ with probability $p(j|i)$ and averages her opinion with that of the neighbor:

$$x_{n+1}(i) = (1 - a_n)x_n(i) + a_n x_{\xi_n(i)}(i).$$

Here $\{a_n\}$ as before, or $a_n \equiv a$.

Limiting ODE

$$\dot{x}(t) = (P - I)x(t)$$

is marginally stable (one eigenvalue zero), hence we do get consensus, but possibly to a wrong value due to random drift.

Alternative: Consider the 'discrete Poisson equation'

$$V(i) = x_0(i) - \kappa + \sum_j p(j|i)V(j), \ i \in S.$$

Here κ is unique, $= \sum_i \pi(i)x_0(i)$ and V unique up to an additive constant.

This arises in average cost problems and can be solved by the *Relative Value Iteration* (RVI)

$$V^{n+1} = x_0 - V(i_0)\mathbf{1} + PV^n.$$

Stochastic approximation version:

$$V^{n+1}(i) = V^n(i) + a(n)I\{X_n = i\} \times \left(x_0(i) - V^n(i_0) + V^n(X_{n+1})\right).$$

Limiting ODE

$$\dot{V}(t) = (P - I)V(t) + x_0 - V_{i_0}(t)$$

converges to the desired V with $V_{i_0} = \kappa$.

Drawback: The value of the i_0th component needs to be broadcast. Alternatively, can use arithmetic mean as offset, obtainable by another averaging scheme using a doubly stochastic matrix on a faster time scale.

Remark

This is a linear (i.e., uncontrolled) counterpart of Q-learning for average cost control.

J. Abounadi, D. P. Bertsekas and V. S. Borkar, "Learning algorithms for Markov decision processes with average cost", *SIAM J. Control and Opt.* 40(3) (2001), 681-692.

Ranking problems

These amount to computation of the *Perron-Frobenius eigenvector* of an irreducible non-negative matrix Q.

Usual approach: the power method:

$$q_{n+1} = \frac{Qq_n}{f(q_n)}, \ n \geq 0,$$

where f is suitably chosen,

e.g., $f(q) = q_{i_0}$ which makes it a multiplicative analog of the RVI. More traditional, $f(q) := \|q\|$.

Stochastic approximation version: Let $d(i) := \sum_j q(j), D := \text{diag}(d(1), \cdots, d(s)), P = [[p(j|i)]] := D^{-1}Q$. Then

$$q_{n+1}(i) = q_n(i) + a(n)\left(\frac{q_n(\xi_n(i))}{q_n(i_0)} - q_n(i)\right), \ n \geq 0.$$

Limiting ODE

$$\dot{q}(t) = \frac{Qq(t)}{q_{i_0}(t)} - q(t)$$

converges to the desired q with $q_{i_0} =$ the Perron-Frobenius eigenvalue. Thus $q_n \to$ this vector w.p. 1.

Even if the Perron-Frobenius eigenvalue is known, this is a more stable iteration because of the scaling properties of the first term on the right.

Remark

This is the linear (i.e., uncontrolled) counterpart of Q-learning for risk-sensitive control.

V. S. Borkar, "Q-learning for risk-sensitive control", *Math. Operations Research* 27(2) (2002), 291-311.

Special case: PageRank

Consider the *random web-surfer model*:

from web page i, with probability $\frac{c}{N(i)}$ go to one of the web pages to which i points, where $c :=$ a prescribed constant $\in (0, 1)$, and $N(i) :=$ the number of web pages to which i points.

With probability $\frac{1-c}{N}$, initiate a new search with a random initial web page chosen uniformly ($N :=$ the number of web pages).

This defines a stochastic matrix $Q = [[q(j|i)]]$,let π be the stationary distribution, i.e., $\pi Q = \pi$. Rank web pages according to decreasing values of π.

Note: $c < 1$ ensures irreducibility.

Equivalently, find the right Perron-Frobenius eigenvector $q := \pi^T$ of $G := Q^T$. Let $P = [[p(j|i)]]$ with $p(j|i) := \frac{1}{N(i)}$ if i points to j, zero otherwise. Then

$$x = \frac{1-c}{N}(I - cP^T)^{-1}\mathbf{1}.$$

Since scaling does not matter, we solve

$$x = \mathbf{1} + sP^T x.$$

Use *split sampling*:

Need the conditional distribution $p(\cdot|\cdot)$, the marginals are not so crucial. Hence instead of the Markov chain $\{X_n\}$, generate i.i.d. pairs (X_n, Y_n) so that $\{X_n\}$ are i.i.d. uniform on S and the conditional law of Y_n given X_n, conditionally independent of all else, is $p(\cdot|\cdot)$.

The algorithm is:

$$\begin{aligned} z_{n+1}(i) \;=\; & z_n + a(n)(I\{X_{n+1} = i\}(1 - z(n)) \\ & + c z_n(X_{n+1}) I\{Y_{n+1} = i\}). \end{aligned}$$

Limiting ODE

$$\dot{z} = I + cP^T z - z$$

is a stable linear system which converges to the desired solution.

$\longrightarrow$ desired convergence of the stochastic approximation scheme w.p. 1.

References

1. V. S. Borkar and A. S. Mathkar, "Reinforcement learning for matrix computations: PageRank as an example", *Proceedings of ICDCIT 2014* (Raja Natarajan, ed.)

2. V. S. Borkar; R. Makhijani and R. Sundaresan, "How to gossip if you must", *IEEE J. on Selected Topics in Signal Processing*, to appear, 2014.

General methodology:

1. Express a non-negative matrix as a diagonal matrix times a stochastic matrix.

2. Replace pre-multiplication by the stochastic matrix by an evaluation at a sample generated according to it.

3. Make an incremental correction according to stochastic approximation.

THANK YOU