Quantized Q-Learning for Near-Optimal Quantizers at Queen's (Q^4)

Liam Cregg

June 23, 2022

1 Introduction

1.1 Zero-Delay Lossy Coding and Optimal Quantizers

We wish to encode a source symbol X_t from a finite alphabet $\mathbb{X}$ using a (smaller) finite alphabet $\mathcal{M} := \{1, 2, \ldots, M\}$ with zero delay (hence a block-coding approach is not viable). We assume the source $\{X_t\}_{t\geq 0}$ is a time-homogenous discrete-time Markov process with initial distribution π_0 and transition kernel $P(dx_{t+1}|x_t)$. The encoder is defined by a quantization policy $\Pi = \{\eta_t\}_{t\geq 0}$, where $\eta_t : \mathcal{M}^t \times \mathbb{X}^{t+1} \to \mathcal{M}$ is a Borel measurable function. That is, the encoder can use all past quantization outputs and all past and current source inputs to generate the current quantization output. This can be viewed as the quantization policy selecting a quantizer $Q_t : \mathbb{X} \to \mathcal{M}$ using past information, then quantizing X_t as $q_t = Q_t(X_t)$ **REF**.

Then, the decoder generates the reconstruction U_t without delay, using decoder policy $\gamma = \{\gamma_t\}_{t\geq 0}$, where $\gamma_t : \mathcal{M}^{t+1} \to \mathcal{U}$ is a measurable function with $\mathcal{U}$ being the (finite) reconstruction alphabet. Thus we have $U_t = \gamma_t(q_{[0,t]})$.

In general, for the zero-delay coding problem, the goal is to minimize the average cost/distortion. In the infinite horizon case with cost function/distortion measure $c_0 : \mathbb{X} \times \mathcal{U} \to \mathbb{R}$, this is given by:

$$J(\pi_0, \Pi, \gamma) := \limsup_{T\to\infty} \mathbf{E}_{\pi_0}^{\Pi,\gamma}\left[\frac{1}{T}\sum_{t=0}^{T-1} c_0(X_t, U_t)\right]$$

However, for the time being we will consider the discounted cost problem, as this problem is easier to tackle using Q-learning methods (to be discussed later). Thus, for some $\beta \in (0, 1)$, we wish to minimize:

$$J(\pi_0, \Pi, \gamma) := \lim_{T\to\infty} \mathbf{E}_{\pi_0}^{\Pi,\gamma}\left[\frac{1}{T}\sum_{t=0}^{T-1} \beta^t c_0(X_t, U_t)\right]$$

For the finite horizon problem, policies using only the conditional probability measure $\pi_t = P(dx_{t+1}|q_{0,t-1})$ and X_t to generate q_t have been shown to be optimal by Walrand and Varaiya **REF**. That is, for every admissable policy, there exists a policy of the form $Q_t = \eta_t(\pi_t)$ and $q_t = Q_t(X_t)$ that performs at least as well. Such policies are called Walrand-Varaiya type or Markov policies, and denoted by Π_W. If η_t does not depend on t, we call such policies stationary and denote the set of these policies by Π_{WS}. In **REF*, Walrand and Varaiya's result was also shown to apply to the infinite horizon discounted cost problem, and in fact that the optimal policy is stationary (that is, in Π_{WS}).

Importantly, it was shown in **REF** that π_{t+1} is conditionally independent of $(\pi_{[0,t-1]}, Q_{[0,t-1]})$ given π_t and Q_t, and hence $\{\pi_t\}$ is a controlled Markov process with control $\{Q_t\}$. Note that using conditional probability properties, we obtain the udpate equation for π_t:

$$\pi_{t+1}(dx_{t+1}) = \frac{1}{\pi_t(Q^{-1}(q_t))}\int_{Q^{-1}(q_t)} P(dx_{t+1}|x_t)\pi_t(dx_t) \tag{1}$$

Therefore, in theory one could use dynamic programming principles to run a policy- or value-iteration algorithm on this controlled Markov process in order to obtain the optimal policy $\pi \in \Pi_{WS}$. However, in practice this proves to be difficult given the setup of the problem and the above update equation. Should elaborate on this, based on Serdar/others' experience in trying to write an iteration algorithm. Hence, it is desirable to use learning techniques such as Q-learning in order to find the optimal quantization policy. To this end, we utilize the recent work of **REF** in the near-optimality of policies under quantization.

1.2 Near-Optimality of Quantized Policies

By viewing quantization as a measurement kernel and using recent results on convergence of Q-learning algorithms for POMDPs, **REF** showed that under mild conditions (namely, weak continuity of the transition kernel), quantized Q-learning (that is, Q-learning where the action and state spaces are quantized versions of the original MDP) leads to asymptotically optimal policies as the number of quantization bins increases. In particular, for any compact $K \subset \mathbb{X}$,

$$\sup_{x_0 \in K} |\hat{J}_\beta(x_0) - J^*_\beta(x_0)| \to 0$$

where $\hat{J}_\beta$ is the optimal value function for the finite model obtained by quantizing the state and action spaces. This value function (and the policy yielding this value function) can then be extended to the original MDP by making constant over the quantization bins **REF**.

We can obtain such a near-optimal policy by running the "standard" Q-learning algorithm but viewing our quantized state as the true state, i.e.

$$\begin{aligned} Q_{t+1}(q(x),u) = (1-&\alpha_t(q(x),u))Q_t(q(x),u)+ \\ &\alpha_t(q(x),u)(c(x,u) + \beta \min_{v \in \mathbb{U}} Q_t(q(X_{t+1}),v)) \end{aligned} \tag{2}$$

Note that the above Q and q are different from those used in Section 1.1, but are used here for consistency with **REF**. We need a few assumptions to hold for this to converge, in particular we need that the transition kernel is weakly continuous, and we need that the state process $\{X_t\}_{t \geq 0}$ is positive Harris recurrent.

We note that in **REF** it was shown that the transition kernel of the controlled Markov chain $\{\pi_t\}$ from Section 1.1 is weakly continuous, i.e. $P(\pi_{t+1}|\pi_t, Q_t)$ is weakly continuous in (π_t, Q_t).

1.2.1 Positive Harris Recurrence of π_t?

Need to show positive Harris recurrence of π_t for the quantized Q-learning algo to converge

Note that π_t does not hit all of $\mathcal{P}(\mathbb{X})$. For example, take $\mathbb{X} = \{0, 1\}$, with transition matrix $T = \begin{pmatrix} 0.75 & 0.25 \\ 0.75 & 0.25 \end{pmatrix}$

Say we have 2 possible quantizers, always mapping to 0 or 1, respectively (the optimal quantizer here will be the one mapping to 0, since X takes the value 0 more often). Note that since the quantizers map all of $\mathbb{X}$ to the same value, the update equation always gives us the stationary distribution, i.e.

$$\begin{aligned} \pi_{t+1}(dx_{t+1}) &= \sum_{i=0}^{1} \pi_t(i) P(dx_{t+1}|i) \\ &= \pi_0(dx_{t+1}) \end{aligned}$$

The Q-learning algorithm above still gives the correct optimal quantizer for π_0, but never hits any other state. So we have to be careful when defining the "state" space for π_t so that we still hit every state-action pair infinitely often (and so that our process $\{\pi_t\}$ is positive Harris recurrent).

If we can show positive Harris recurrence, then the above result on near-optimality of the quantized Q-learning algorithm is applicable.

In summary, we will quantize the state and actions of this controlled Markov chain (that is, we will quantize π_t and Q_t) and run Q-learning on this finite model approximation to obtain a near-optimal quantization policy $\hat{\Pi}$ for the original quantization problem.

2 Setup

2.1 Finite State and Action Spaces

First we consider the case where our quantization problem has a finite state space and we have a finite number of quantizers from which to choose, that is $\mathbb{X}$ and $\mathcal{Q}$ are both finite. Then, as in Section 1.1, we let $\pi_t = P(dx_{t+1}|q_{[0,t-1]})$, and consider the controlled Markov chain $\{\pi_t\}$, with control $\{Q_t\}$. We note that our action space $\mathcal{Q}$ is finite but our "state" is $\pi_t \in \mathcal{P}(\mathbb{X})$, which in this case is a simplex in $\mathbb{R}^{|\mathbb{X}|}$. We wish to quantize π_t, which we will do using the algorithm in **REF**. Essentially, it finds quantizes π_t to an information-theoretic "type" distribution, in a nearest-neighbour fashion. The overall algorithm follows.

2.1.1 Algorithm

- State space ($\mathbb{X}$) and state transition kernel ($P(x_{t+1}|x_t)$)
- Distribution of $\mathbb{X}_0$ (π_0)
- Granularity of types for quantization of π_t (n)
- Set of quantizers ($\mathcal{Q}$)

2.2 Pseudocode

1. Initialize arbitrary Q-table of size $|Z_n| \times |\mathcal{Q}|$ (see below for definition of Z_n)
2. Initialize state $\mathbb{X}_0$ according to π_0
3. Quantize π_0 according to *quantization of the belief-space algorithm*, call this $\hat{\pi}_0$
4. Select quantizer Q_0 according to arbitrary initial quantization policy $\Pi \in \mathbf{\Pi}_W^C$ and $\hat{\pi}_0$, i.e. $Q_0 = \Pi(\hat{\pi}_0)$
5. Quantize $\mathbb{X}_0$ according to $q_0 = Q_0(X_0)$
6. For $t = 0 \dots T-1$
   (a) Receive cost of quantization according to $(X_t - q_t)^2$
   (b) Receive X_{t+1} according to $P(x_{t+1}|x_t)$
   (c) Receive π_{t+1} according to filtering equation
   (d) Quantize π_{t+1} according to *quantization of the belief-space algorithm*, call this $\hat{\pi}_{t+1}$
   (e) Update Q-table T_t using standard update equation
   (f) Select quantizer Q_{t+1} according to quantization policy (arbitrary). $Q_{t+1} = \Pi(\hat{\pi}_{t+1})$
   (g) Quantize $\mathbb{X}_{t+1}$ according to $q_{t+1} = Q_{t+1}(X_{t+1})$

3 Examples