Quantized Q-Learning for Near-Optimal Quantizers

Liam Cregg

June 28, 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 $q_t = Q_t(X_t)$ [2].

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 [4]. 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 [5], 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 [5] 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 [1] 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, [1] 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.

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 [1].

We note that in [2] 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). Then $\{\pi_t\}$ is weak Feller on a compact space, so it admits an invariant probability measure. Therefore, 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.

1.3 Some Notes on the State Space of π_t

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. In general, the possible states of π_t are more difficult to characterize, as the sum is not over the entire state space, but only over the preimage of the quantization output $(Q_t^{-1}(q_t))$.

This is not a significant barrier to convergence of the algorithm, as we can simply redefine the domain of the obtained policy to only include possible values of π_t (i.e. the ones that are visited during training). However, it is of practical interest to only include states which are attainable, in order to shrink the Q-table (and thus memory requirements) during training. Determining which values of π_t are attainable before running the algorithm is a direction of future research.

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 [3]. 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 Pseudocode

Algorithm 1: Finite State and Action

Set Parameters
  State space ($\mathbb{X}$) and state transition kernel ($P(x_{t+1}|x_t)$)
  Distribution of X_0 (π_0)
  Granularity of types for quantization of π_t (n)
  Set of quantizers ($\mathcal{Q}$)
Initialize arbitrary Q-table of size $|Q_n| \times |\mathcal{Q}|$ (see [3] for definition of Q_n)
Initialize state X_0 according to π_0
Quantize π_0 according to *Algorithm 1* in [3], call this $\hat{\pi}_0$
Select quantizer Q_0 according to arbitrary quantization policy $\Pi \in \mathbf{\Pi}_W^C$ and $\hat{\pi}_0$, i.e. $Q_0 = \Pi(\hat{\pi}_0)$
Quantize X_0 according to $q_0 = Q_0(X_0)$
for $t = 0 \dots T-1$
  Receive cost of quantization according to $(X_t - q_t)^2$
  Receive X_{t+1} according to $P(x_{t+1}|x_t)$
  Receive π_{t+1} according to (1)
  Quantize π_{t+1} according to *Algorithm 1* in [3], call this $\hat{\pi}_{t+1}$
  Update Q-table using (2)
  Select quantizer Q_{t+1} according to quantization policy (arbitrary). $Q_{t+1} = \Pi(\hat{\pi}_{t+1})$
  Quantize X_{t+1} according to $q_{t+1} = Q_{t+1}(X_{t+1})$
end

2.1.2 Proof of Convergence

The proof that this algorithm converges to the optimal policy as n grows follows directly from the quantized Q-learning result in [1]. We need the following lemmas:

Lemma 2.1. *[2, Lemma 11]. The transition kernel $P(d\pi_{t+1}|\pi_t, Q_t)$ is weakly continuous in (π_t, Q_t). That is,*

$$\int_{\mathcal{P}(\mathbb{X})\times\mathcal{Q}} f(\pi^{'}) P(d\pi^{'}|\pi, Q)$$

is continuous on $\mathcal{P}(\mathbb{X}) \times \mathcal{Q}$ for all continuous and bounded f.

Proof. The proof can be found in its entirety in [2], but essentially follows by writing the integral as a sum over the message set $\mathcal{M}$, and using the additional lemma that $\pi_n Q_n \to \pi Q$ in total variation. □

Lemma 2.2. *The Markov process $\{\pi_t\}_{t\geq 0}$ admits an invariant probability measure.*

Proof. The proof follows from Lemma 2.1 and the fact that every weak-Feller Markov chain on a compact space admits an invariant probability measure (and indeed $\mathcal{P}(\mathbb{X})$ is compact). Is is necessarily true that the subset of $\mathcal{P}(\mathbb{X})$ where π_t lives is compact? □

Lemma 2.3. *[3, Proposition 2] The maximum radius of the quantization regions for π_t under the L_∞ norm is given by*

$$b_\infty = \frac{1}{n}(1 - \frac{1}{m})$$

Then, by restricting the space of quantized beliefs $\hat{\pi}_t$ to only those which are visited infinitely often this feels hand-wavey,. Even if I can't come up with an explicit description of the possible values of π_t, I'd like to describe it more formally, and under a randomized (and independent) exploration policy Π^* (see [1]), we have the following theorem for convergence. I will probably switch some of the Qs in this equation for other letters

Theorem 2.4. *The algorithm in 2.1.1 converges to*

$$Q^*(\hat{\pi}_i, Q) = C^*(\hat{\pi}_i, Q) + \beta \sum_{\hat{\pi}_j \in Q_n} P^*(\hat{\pi}_j | \hat{\pi}_i, Q) \min_{v \in Q} Q^*(\hat{\pi}_j, v)$$

Here, P^* and C^* are defined by

$$C^*(\hat{\pi}_i, Q) = \int_{B_i} c(\pi, Q)\phi_i(d\pi)$$

and

$$P^*(\hat{\pi}_j | \hat{\pi}_i, Q) = \int_{B_i} P(B_j | \pi, Q)\phi_i(dx)$$

where

$$\phi_i(A) := \frac{\phi_{\Pi^*}(A)}{\phi_{\Pi^*}(B_i)}, \quad \forall A \subset B_i, \quad \forall i \in \{1, \dots, |Q_n|\}$$

and ϕ_{Π^*} is the invariant measure of $\{\pi_t\}_{t \geq 0}$ under the exploration policy Π^*. Also, $B_i \subset \mathcal{P}(\mathbb{X})$ is the bin under the quantization algorithm in [3] (i.e. $\{B_i\}_{i=1}^{|Q_n|}$ is a partition of $\mathcal{P}(\mathbb{X})$).

Furthermore, the policy $\bar{\Pi}(\hat{\pi}) = \operatorname{argmin}_{Q \in \mathcal{Q}} Q^*(\hat{\pi}, Q)$ satisfies

$$\sup_{\pi \in \mathcal{P}(\mathbb{X})} |\hat{J}_\beta(\pi, \bar{\Pi}) - J^*_\beta(x_0)| \to 0$$

as $n \to \infty$.

Proof. This is essentially due to [1, Theorem 3.2 and Corollary 3.3]. From Lemmas 2.1 and 2.2, we have that all of the necessary assumptions are met, and due to Lemma 2.3, we have that the maximum radius of the quantization bins $b_\infty \to 0$ as $n \to \infty$. The result follows. □

3 Examples

3.1 Trivial Example

We start with the simplest case, where $\mathbb{X} = \{1, 2\}$. Let the transition matrix of $\{X_t\}_{t\geq 0}$ be $T = \begin{pmatrix} 0.75 & 0.25 \\ 0.75 & 0.25 \end{pmatrix}$. The only useful quantizers to pick in this situation are constant ones, that is $Q_i(x) = i \quad \forall x \in \mathbb{X}, i = 1, 2$.

Now suppose we wish to quantize π_t very coarsely, say n=2. Then we have $Z_n = \{(1, 0), (\frac{1}{2}, \frac{1}{2}), (0, 1)\}$, so our Q-table is of size 3×2.

Running the above algorithm for $T = 10^5$ iterations gives the following Q-table (the Q-table was initialized to 10).

	Quantizer	
State	4.77	5.27
	10	10
	10	10

Note that in this case, it doesn't matter what n is, since π_t is constant (as described in Section 1.3). Nevertheless, the algorithm converges to the correct optimal quantizer (it should always quantize to 1 since the value 1 appears more frequently).

3.2 More Interesting Example

Now we have $\mathbb{X} = \{1, \ldots, 8\}$, using a randomly generated transition matrix.

Say we have 2-cell quantizers with cells given by $B_i^1 = \{1, \ldots, i\}$ and $B_i^2 = \{i + 1, \ldots, 8\}$. For simplicity, we will let the codepoints be the middle of the cell, rounded down (e.g. the codepoint of $\{1, 2, 3\}$ would be 2). There are 7 such quantizers. As before, let's start with a coarse quantization of π_t, say $n = 2$, and double n each time. We should get lower costs (i.e. better quantization policies) as the granularity increases. Indeed, averaging the costs of the resultant policies gives the following table.

n	Average Cost
2	2.6585
3	2.0278
4	1.9910
5	1.7190
6	1.5220
8	1.6072

Unfortunately, the size of Q_n (i.e. the number of possible quantized beliefs) grows quickly with n, and so further increasing the granularity becomes very computationally demanding. However, the number of actual visited states tends to be much lower than the total size of Q_n, and so it may be possible to increase the efficiency of this algorithm by restricting the state space of π_t.

References

[1] Ali Devran Kara, Naci Saldi, and Serdar Yüksel. *Q-Learning for MDPs with General Spaces: Convergence and Near Optimality via Quantization under Weak Continuity*. 2021. DOI: 10.48550/ARXIV.2111.06781. URL: https://arxiv.org/abs/2111.06781.

[2] T. Linder and S. Yüksel. "On optimal zero-delay coding of vector Markov sources". In: *IEEE Trans. Inf. Theory* 60.10 (Oct. 2014), pp. 5975–5991.

[3] Y.A. Reznik. "An algorithm for quantization of dicrete probability distributions". In: *DCC 2011* (Mar. 2011), pp. 333–342.

[4] J. Walrand and P. Varaiya. "Optimal causal coding-decoding problems". In: *IEEE Trans. Inf. Theory* IT-29.6 (Nov. 1983), pp. 814–820.

[5] R. Wood, T. Linder, and S. Yüksel. *Optimal Zero Delay Coding of Markov Sources: Stationary and Finite Memory Codes*. 2016. DOI: 10.48550/ARXIV.1606.09135. URL: https://arxiv.org/abs/1606.09135.