**IE 3186: Approximate Dynamic Programming** **Fall 2018**

# Lecture 8: Policy Evaluation and TD Learning

*Lecturer: Daniel Jiang* *Scribes: Tarik Bilgic, Shaoning Han*

References:

D. P. Bertsekas. *Dynamic Programming and Optimal Control: Approximate Dynamic Programming*, Vol. 2, 4th ed., Athena Scientific, Belmont MA, 2012. (§6.3)

R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*, 2nd ed., The MIT Press, Cambridge, MA, 2018. (Ch. 6)

W. B. Powell and H. Topaloglu. *Approximate dynamic programming for large-scale resource allocation problems*, Tutorials in Operations Research, 2012.

## 8.1 Policy Evaluation using Projected Bellman Equation

Evaluation of the value of fixed policy $\mu$

$$J_\mu(i) = \mathbf{E}\left[\sum_{k=0}^{\infty} \gamma^k g(i_k, \mu(i_k), w_{k+1}) \,\Big|\, i_0 = i\right]$$

can be done by solving the fixed point equation $J = T_\mu J$ or by iterating $T_\mu$ from any starting $J_0$.

For today, since $\mu$ is fixed, we can ignore it and focus on the evolution of states under $\mu$, $i_k, i_{k+1}, i_{k+2}, \ldots$, which is a Markov chain. Instead of $f(i_k, u_k, w_{k+1}) = i_{k+1}$, let $p_{ij}$be the probability of moving from state $i$ to $j$, following $\mu(i)$. The states form a Markov Chain under $\mu$ that has a unique steady state distribution with positive components:

$$\xi_j = \lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} \mathbf{P}(i_k = j \,|\, i_0 = i) \ > 0.$$

Recall that we are looking at

$$J_{k+1} = \Pi T_\mu J_k$$

Assume that $\Phi$ has rank $m$ (where $m$ is the dimension of the parameter vector). This means that every vector in $S = \{\Phi r : r \in \mathbb{R}^m\}$ is associated with a unique parameter vector $r$.

Recall the definition of the weighted Euclidean norm $\|J\|_w = \sqrt{\sum_i w_i J(i)^2}$, where $w = (w_1, w_2, \ldots, w_n)$ is a vector of weights.

**Proposition 8.1** (Projections are non-expansive)**.**

$$\|\Pi J - \Pi J'\|_w \leq \|J - J'\|_w.$$

*Proof.* See previous lecture. □

If $\Pi$ were non-expansive in the same norm under which $T_\mu$ is a contraction, then we could show that $\Pi T_\mu$ is a contraction. So far, we only have that $T_\mu$ is a contraction in the max-norm, so we have a *norm mismatch problem.*

**Lemma 8.2** (Non-expansiveness of transition matrix)**.** *Let $P = (p_{ij})$ be the transition matrix of a Markov chain that has an invariant distribution $\xi = (\xi_1, \ldots, \xi_n)$, meaning that $\xi' = \xi' P$. Then,*

$$\|Pz\|_\xi \leq \|z\|_\xi \quad \textit{for any } z \in \mathbb{R}^n.$$

*Proof.* Note that:

$$\begin{aligned}
\|Pz\|_\xi^2 &= \sum_{i=1}^n \xi_i \left( \sum_{j=1}^n p_{ij} z_i \right)^2 \\
&\leq \sum_{i=1}^n \xi_i \sum_{j=1}^n p_{ij}\, z_j^2 && \textit{(by Jensen's Inequality)} \\
&= \sum_{j=1}^n \left( \sum_{i=1}^n \xi_j p_{ij} \right) z_j^2 \\
&= \sum_{j=1}^n \xi_j z_j^2 = \|z\|_\xi^2,
\end{aligned}$$

which completes the proof. □

We take advantage of this to get a contraction property in $T_\mu$ for policy evaluation.

**Proposition 8.3.** *The combined operator $\Pi T_\mu$, where $\Pi$ is the projection operator with respect to $\| \cdot \|_\infty$ is a $\gamma$-contraction in $\| \cdot \|_\xi$.*

*Proof.* Let $T_\mu J = g + \gamma P J$, where $g(i) = \mathbf{E}[(g(i, \mu(i), w)]$. Then, we have

$$\begin{aligned}
\|\Pi T_\mu J - \Pi T_\mu J'\|_\xi &\leq \|T_\mu J - T_\mu J'\|_\xi \\
&= \|\gamma P J - \gamma P J'\|_\xi \\
&= \gamma \|P(J - J')\|_\xi \\
&\leq \gamma \|J - J'\|_\xi,
\end{aligned}$$

which completes the proof. □

This means that $(\Pi T_\mu)^k J_0 \to \Phi r^*$, where $\Phi r^*$ is the unique fixed point of $\Pi T_\mu$. Also, $r^*$ is the unique solution (due to the rank assumption) of $\Phi r = (\Pi T_\mu)(\Phi r)$.

**Proposition 8.4** (Policy evaluation error bound)**.**

$$\|J_\mu - \Phi r^*\|_\xi \leq \frac{1}{\sqrt{1-\gamma^2}} \|J_\mu - \Pi J_\mu\|_\xi.$$

*Proof.*

$$\begin{aligned}
\|J_\mu - \Phi r^*\|_\xi^2 &\leq \|J_\mu - \Pi J_\mu\|_\xi^2 + \|\Pi J_\mu - \Phi r^*\|^2 \\
&= \|J_\mu - \Pi J_\mu\|_\xi^2 + \|\Pi T_\mu J_\mu - \Pi T_\mu \Phi r^*\|_\xi^2 \\
&\leq \|J_\mu - \Pi J_\mu\|_\xi^2 + \gamma^2 \|J_\mu - \Phi r^*\|_\xi^2,
\end{aligned}$$

so we conclude by rearranging the terms. □

We have shown that projected VI works reliability if the norm is chosen correctly.

**Remark 8.5.** *There are more direct methods to solve*

$$\Phi r = \Pi T_\mu \Phi r,$$

*involving matrix inversion. See the textbook.*

Next, can we improve upon the coefficient of $1/\sqrt{1-\gamma^2}$?

## 8.2 Conceptual Intro to Temporal Difference Learning

Our current VI approaches are "bootstrapped" approaches, where we are using the old value function approximation to update the new value function approximation (also true for Q-learning). This introduces bias. Alternatively, we could evaluate the policy simply by running it for a long time and get an unbiased observation:

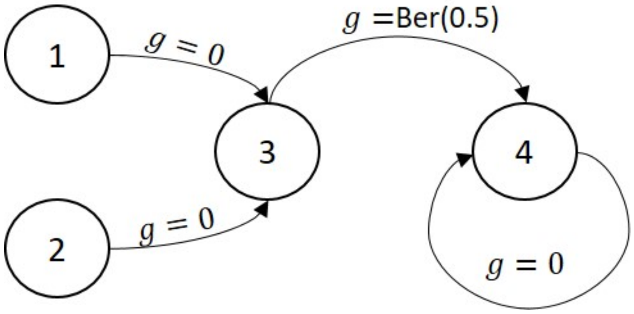
$$\text{“new observation} = \text{cost}_0 + \gamma \text{cost}_1 + \gamma^2 \text{cost}_2 + \ldots\text{”}$$

This is called the Monte Carlo approach.

**Example 8.6.** *Set* $\gamma = 1$ *and let* $\mathbf{P}(\textcircled{1}) = 0.9, \mathbf{P}(\textcircled{2}) = 0.1$ *and consider the transition dynamics and rewards given in the diagram below.*

*Consider stochastic approximation (Q-learning-like approaches) for evaluating this Markov Chain. Case 1:*

$$\begin{aligned}
J(i_k) &\leftarrow (1-\alpha)\, J(i_k) + \alpha_k \big[g(i_k, i_{k+1}) + J(i_{k+1})\big] \\
&\leftarrow J(i_k) + \alpha_k \big[g(i_k, i_{k+1}) + J(i_{k+1}) - J(i_{k+1})\big].
\end{aligned} \tag{8.1}$$



*The term $g(i_k, i_{k+1}) + J(i_{k+1})$ is called the "target" or the new value that we are trying to mimic. Case 2 is given below; the target is now replaced with a Monte Carlo estimate instead of the bootstrapped observation.*

$$J(i_k) \leftarrow J(i_k) + \alpha_k \left[\sum_{k'=k}^{\infty} g(i_k, i_{k+1}) - J(i_k)\right]. \tag{8.2}$$

*Suppose $\alpha_k = \frac{1}{k}$ (which corresponds to simple averaging).*
*Case (1): At $n$th visit of ②, ① is visited approximately $9n$ times, ③ is visited approximately $10n$ times.*

$$J_n(3) \approx \frac{1}{10n} \sum_{i=1}^{10n} \text{Ber}(0.5)$$
$$\implies J_n(2) \approx \frac{1}{n} \sum_{i=1}^{k} (0 + J_i(3))$$

*Note that $J_i(3)$ quickly converges to $0.5$ and variance is decreasing. Thus, $J_n(2)$ will quickly start to average nearly deterministic values of $J_i(3)$.*

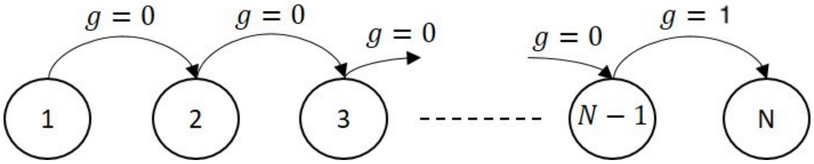
*Case (2): At $n$th visit of ②,*

$$\implies J_n(2) = \frac{1}{n} \sum_{i=1}^{n} \text{Ber}(0.5),$$

*since we are simply running the Markov chain until it reaches the terminal state. The variance of each observation is fixed. In this example, Case (1) is better.*

**Example 8.7.** *Now, consider a different Markov chain, a long sequence of states and the same two algorithms as in the previous example.*

*Case (1): Consider ①. Target is $0 + J(2)$, but $J(2)$ depends on $J(3)\ldots$ and $J(3)$ depends on $J(4)$, etc. $J(N-1)$ might be slowly averaging observations of $0$ from its initial value. Due to the multiple levels of dependence, this could take a long time.*



*Case (2): All states see the correct target of 1 when the Markov chain is simulated until the terminal state. Therefore, in this example, Case (2) is better.*

We conclude that both TD($\lambda$) and MC have merits. Why not unify them?

## 8.3 TD($\lambda$)

- Define
$$G_k^{(n)} = g(i_k, i_{k+1}) + \gamma g(i_{k+1}, i_{k+2}) + \cdots + \gamma^n gJ(i_{k+n}).$$
- The Monte Carlo (MC) algorithm uses $G_k^{(\infty)}$ as a target. The bootstrapped version, which we'll call TD(0), uses $G_k^{(1)}$.
- Since we don't know ahead of time whether MC or TD(0) should be performed, one approach would be to combine them:
$$G_k^{avg} = \frac{1}{2}G_k^{(1)} + \frac{1}{2}G_k^{(\infty)}$$
$$or$$
$$G_k^{avg} = \frac{1}{2}G_k^{(1)} + \frac{1}{4}G_k^{(2)} + \frac{1}{4}G_k^{(3)}.$$
- TD($\lambda$) is a specific way, parameterized by $\lambda$, to do the combining.
- $\lambda \in [0, 1]$ interpolates from TD(0) to MC, where TD(1) is MC.
- Given some $0 < \lambda < 1$, the target of TD($\lambda$) update is a mixture of multi-step returns.

Let the weight of $G_k^{(i)}$ be $\mathbf{P}(X = i)$ where $X$ is a Geometric random variable with parameter $1 - \lambda$. The $\lambda$-return is

$$G^k = (1 - \lambda)\sum_{l=0}^{\infty} \lambda^l G_k^{l+1}$$

Operator notation for TD($\lambda$) replace $T_\mu$ with

$$T_\mu^{(\lambda)} = (1-\lambda)\sum_{l=0}^{\infty} \lambda^l T_\mu^{l+1} \qquad \text{for} \quad 0 < \lambda < 1.$$

The fixed point problem is now $J = T_\mu^{(\lambda)} J$. Is it better than $T_\mu$?

**Proposition 8.8.**

$$T_\mu^{(\lambda)} \textit{ is a } \frac{\gamma(1-\lambda)}{1-\gamma\lambda} \textit{ contraction in } \|\cdot\|_\xi.$$

*Proof.* First, note that

$$\begin{aligned}
T_\mu^2 J &= g + \gamma P T_\mu J = g + \gamma P(g + \gamma P J) \\
&= (I + \gamma P) g + \gamma^2 P^2 J \\
T_\mu^3 J &= g + \gamma P (T_\mu^2 J) = g + \gamma P((I + \gamma P)g + \gamma^2 P^2 J) \\
&= (I + \gamma P + \gamma^2 P^2) g + \gamma^2 P^2 J \\
&\vdots \\
T_\mu^n J &= \left(\sum_{k=0}^{n-1} \gamma^k P^k\right) g + \gamma^n P^n J.
\end{aligned}$$

Next, using the above,

$$\begin{aligned}
T_\mu^{(\lambda)} J &= (1-\lambda)\sum_{l=0}^{\infty} \lambda^l (T_\mu^{l+1} J) \\
&= (1-\lambda)\sum_{l=0}^{\infty} \lambda^l \left[\left(\sum_{k=0}^{l} \gamma^k P^k\right) g + \gamma^{l+1} P^{l+1} J\right] \\
&= g^{(\lambda)} + \gamma P^{(\lambda)} J, \text{ where } P^{(\lambda)} = (1-\lambda)\sum_{l=0}^{\infty} \lambda^l \gamma^l P^{l+1}.
\end{aligned}$$

For any $J$ and $J'$,

$$\begin{aligned}
\|T_\mu^{(\lambda)} J - T_\mu^{(\lambda)} J'\|_\xi &= \|\gamma P^{(\lambda)} J - \gamma P^{(\lambda)} J'\|_\xi \\
&= \gamma \|P^{(\lambda)}(J - J')\|_\xi \\
&= \gamma \|(1-\lambda)\sum_{l=0}^{\infty} \gamma^l \lambda^l P^{l+1}(J - J')\|_\xi \\
&\le \frac{\gamma(1-\lambda)}{1-\gamma\lambda} \||J - J'\|_\xi
\end{aligned}$$

□

We can rewrite the contraction coefficient to see that it is smaller than $\gamma$:

$$\begin{aligned}\gamma^{(\lambda)} = \frac{\gamma(1-\lambda)}{1-\gamma\lambda} &= \frac{\gamma(1-\gamma\lambda+\gamma\lambda-\lambda)}{1-\gamma\lambda} \\ &= \gamma - \frac{\gamma(\lambda-\gamma\lambda)}{1-\gamma\lambda} < \gamma,\end{aligned}$$

since the second term is positive. Also, $\gamma^{(\lambda)} \to 0$ as $\lambda \to 1$, so we can make this arbitrarily small.

**Theorem 8.9.** *Let $\Phi r^*_\lambda$ be fixed point of $\Pi T^{(\lambda)}_\mu$. Then*

$$\|J_\mu - \Phi r^*_\lambda\|_\xi \leq \frac{1}{\sqrt{1-(\gamma^{(\lambda)})^2}} \|J_\mu - \Pi J_\mu\|_\xi$$

**Remark 8.10.** *$T^{(\lambda)}_\mu$ can be a contraction in any norm if $\lambda$ large enough by norm equivalence. As $\lambda \to 1$, error bound says that $\Phi r^*$ converges to the best approximation of $J_\mu$ in $S$.*

## 8.4 Paper Discussion: Large-Scale Resource Allocation

### 8.4.1 Model and Notations

- $R_t = (R_{ta})_{a\in\mathcal{A}}$: $R_{ta}$ is the number of resources of type $a$ at $t$
- $D_t = (D_{tb})_{b\in\mathcal{B}}$: $D_{tb}$ is the number of demands of type $b$ at $t$
- $S_t = (R_t, D_t)$: state vector
- $W_t = (\hat{R}_t, \hat{D}_t)$: exogenous information
- $x_t = (x_{tad})_{a\in\mathcal{A}, d\in\mathcal{D}}$: decision vector, where $x_{tad}$ is the number of resources of type $a$ modified by using decision $d$ at time $t$
- $C_t(x_t)$: cost(profit) linear function of $x_t$
- $X^\pi_t(\cdot)$: function that maps $S_t$ to a feasible decision $x_t \in \mathcal{X}(S_t)$
- $\mathcal{X}(S_t)$: set of feasible decisions for $S_t$ at time $t$
- Transition function:
$$S_{t+1} = S^M(S_t, x_t, W_{t+1}) \tag{8.3}$$
- Objective function:
$$\max_{\pi\in\Pi} \mathbf{E}\{\sum_{t\in\mathcal{T}} C_t(X^\pi_t(S_t))\} \tag{8.4}$$

- Bellman equation:

$$V_t(S_t) = \max_{x_t \in \mathcal{X}(S_t)} C_t(x_t) + \mathbf{E}\{V_{t+1}(S^M(S_t, x_t, W_{t+1}))|S_t\} \tag{8.5}$$

### 8.4.2 An Approximation Strategy using Postdecision State

#### Postdecision State Vector

- $R_t^x = S^{M,x}(S_t, x_t)$: the number of resources immediately after we make the decisions at $t$
- $S_t^x = R_t^x$: postdecision vector, assuming any unserved demands are lost

$$(S_0, x_0, S_0^x, W_1, S_1, x_1, S_1^x, \ldots, W_t, S_t, x_t, S_t^x, \ldots, W_T, S_T, x_T, S_T^x)$$

Rewrite the Bellman equation in terms of $R_t^x$:

$$V_{t-1}^x(R_{t-1}^x) = \mathbf{E}\left[\max_{x_t \in \mathcal{X}(R_{t-1}^x, \hat{R}_t, \hat{D}_t)} C_t(x_t) + V_t^x(S^{M,x}(S_t, x_t)) \mid R_{t-1}^x\right] \tag{8.6}$$

By doing so, we interchange $\mathbf{E}$ and the operator max.

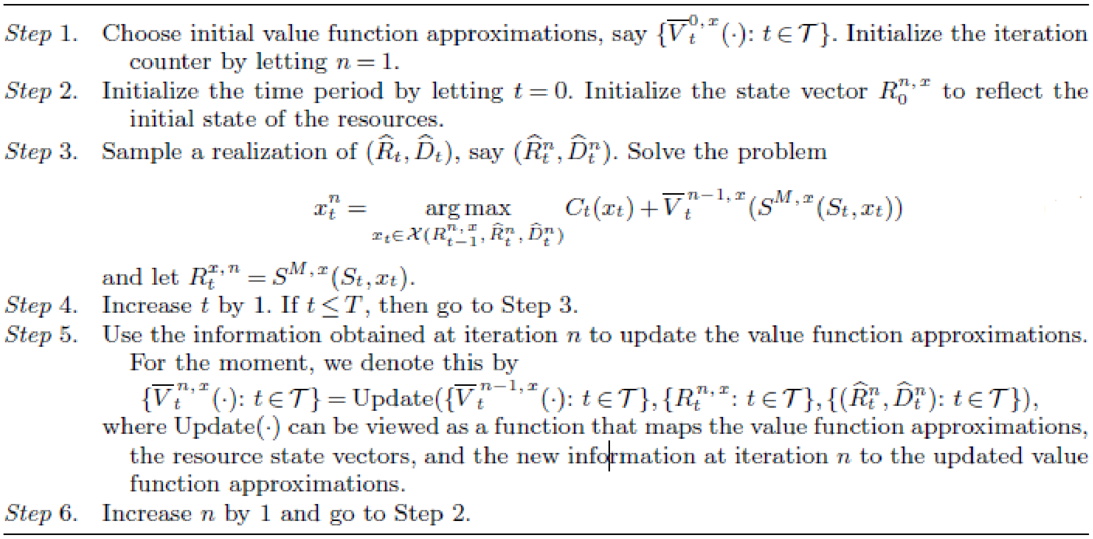
Use a sample realization $W_t(\omega) = (\hat{R}_t, \hat{D}_t)$ to drop $\mathbf{E}$ and get the approximation:

$$\tilde{V}_{t-1}^x(R_{t-1}^x) = \max_{x_t \in \mathcal{X}(R_{t-1}^x, \hat{R}_t, \hat{D}_t)} C_t(x_t) + V_t^x(S^{M,x}(R_{t-1}^x, W_t(\omega), x_t)) \tag{8.7}$$

Use $\bar{V}_t^x(\cdot)$ to approximate $V_t^x(\cdot)$.

## An algorithmic framework for ADP

TABLE 1. An algorithmic framework for approximate dynamic programming.



*Step* 1. Choose initial value function approximations, say $\{\overline{V}_t^{0,x}(\cdot)\colon t\in\mathcal{T}\}$. Initialize the iteration counter by letting $n=1$.

*Step* 2. Initialize the time period by letting $t=0$. Initialize the state vector $R_0^{n,x}$ to reflect the initial state of the resources.

*Step* 3. Sample a realization of $(\widehat{R}_t,\widehat{D}_t)$, say $(\widehat{R}_t^n,\widehat{D}_t^n)$. Solve the problem

$$x_t^n = \operatorname*{arg\,max}_{x_t\in\mathcal{X}(R_{t-1}^{n,x},\widehat{R}_t^n,\widehat{D}_t^n)} C_t(x_t)+\overline{V}_t^{n-1,x}(S^{M,x}(S_t,x_t))$$

and let $R_t^{x,n}=S^{M,x}(S_t,x_t)$.

*Step* 4. Increase $t$ by 1. If $t\le T$, then go to Step 3.

*Step* 5. Use the information obtained at iteration $n$ to update the value function approximations. For the moment, we denote this by

$$\{\overline{V}_t^{n,x}(\cdot)\colon t\in\mathcal{T}\}=\text{Update}(\{\overline{V}_t^{n-1,x}(\cdot)\colon t\in\mathcal{T}\},\{R_t^{n,x}\colon t\in\mathcal{T}\},\{(\widehat{R}_t^n,\widehat{D}_t^n)\colon t\in\mathcal{T}\}),$$

where Update$(\cdot)$ can be viewed as a function that maps the value function approximations, the resource state vectors, and the new information at iteration $n$ to the updated value function approximations.

*Step* 6. Increase $n$ by 1 and go to Step 2.

## Structure for The Value Function Approximation

A generic structure

$$\bar{V}_t^x(R_t^x)=\sum_{f\in\mathcal{F}}\theta_{tf}\phi_f(R_t^x), \tag{8.8}$$

where the function $\phi_f$ characterize the structure of state vectors and $\theta_{tf}$'s are the parameters tuned in every iteration. The characteristics are the type and the number of resources. Or, we could consider a separable approximation:

$$\bar{V}_t^x(R_t^x)=\sum_{a\in\mathcal{A}}\bar{V}_{ta}^x(R_{ta}^x). \tag{8.9}$$

## Form of $\bar{V}_{ta}^x(\cdot)$

Linear value function approximation

$$\bar{V}_{ta}^x(R_{ta}^x)=\bar{v}_{ta}R_{ta}^x \tag{8.10}$$

Piecewise-linear value function approximation

- $\bar{V}_{ta}^x(\cdot)$ is piecewise-linear. Assume the number of resources are integers and have an upper bound $Q$. $\bar{V}_{ta}(\cdot)$ can be characterized by a sequence of slopes $\{\bar{v}_{ta}(q):q=1,2,\ldots,Q\}$.

$$\bar{v}_{ta}(q)=\bar{V}_{ta}^x(q)-\bar{V}_{ta}^x(q-1) \tag{8.11}$$

- $\bar{V}^x_{ta}(\cdot)$ is concave
$$\bar{v}_{ta}(1) \geq \bar{v}_{ta}(2) \geq \cdots \geq \bar{v}_{ta}(Q) \tag{8.12}$$

**Updating Linear Value Function Approximations**

We can perturb one attribute and use an estimate of $V^x_t(R^{n,x}_t + e_a) - V^x_t(R^{n,x}_t)$ to update the slopes.

$$\vartheta^n_{ta} = \tilde{V}^{n,x}_t(R^{n,x}_t + e_a, \hat{R}^n_t, \hat{D}^n_t) - \tilde{V}^{n,x}_t(R^{n,x}_t, \hat{R}^n_t, \hat{D}^n_t) \tag{8.13}$$

Combine $\vartheta^n_{ta}$ and $\bar{v}^{n-1}_{ta}$,

$$\bar{v}^n_{ta} = [1 - \alpha_{n-1}]\bar{v}^{n-1}_{ta} + \alpha_{n-1}\vartheta^n_{ta} \tag{8.14}$$

Linear approximations can be unstable and do not perform as well as piecewise-linear combinations.

**Updating Piecewise-Linear Value Function Approximations**

Similarly, let

$$\theta^n_{ta}(q) = \begin{cases} [1 - \alpha_{n-1}]\bar{v}^{n-1}_{ta}(q) + \alpha_{n-1}\vartheta^n_{ta} & \text{if } q = R^{n,x}_{ta} + 1 \\ \bar{v}^{n-1}_{ta}(q) & \text{o.w.} \end{cases} \tag{8.15}$$

$\theta^n_{ta}(1) \geq \cdots \geq \theta^n_{ta}(Q)$ may not hold. Project to set of concave functions:

$$\bar{v}^n_{ta} = \text{argmin}_{z \in \mathcal{W}} \|z - \theta^n_{ta}\|_2, \tag{8.16}$$

where $\mathcal{W} = \{z \in \mathbb{R}^Q : z_1 \geq z_2 \geq \cdots \geq z_Q\}$ is a convex cone. It's easy to solve problem (8.16) using `KKT` conditions.