Dual Representations for Dynamic Programming and Reinforcement Learning

Tao Wang Michael Bowling Dale Schuurmans
Department of Computing Science
University of Alberta
Edmonton, Canada
Email: {trysi,bowling,dale}@cs.ualberta.ca

【强化学习 73】Dual Representation

张楚珩
清华大学 交叉信息院博士在读

强化学习问题的对偶形式。

原文传送门

Wang, Tao, Michael Bowling, and Dale Schuurmans. "Dual representations for dynamic programming and reinforcement learning." 2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning. IEEE, 2007.

特色

考虑找到MDP上的最优策略使得累积奖励最大，如果已知 dynamics（P）和 reward（R）那么该问题是一个动态规划问题（DP），又称作规划问题（planning problem）；如果这两者未知只能通过和环境交互得到，那么该问题是一个强化学习问题（RL），又称作学习问题（learning problem）。相应的动态规划问题可以表示为一个线性规划问题（LP，可以参见我导师的讲义），相应地，该问题就其对偶形式。前面讲到的 successor representation （SR）就实际上是在解相应的对偶问题。本文给出了 DP 和 RL 问题的相应对偶形式，具体地，给出了对偶形式的 policy evaluation、policy iteration、TD evaluation、SARSA、Q-learning。

过程

这篇文章都是矩阵-向量表示形式，需要注意符号定义，特别是维度。

1. 线性规划及其对偶问题

考虑一个规划问题，它可以写为 LP 的形式

$$\begin{aligned} &\min_{\mathbf{v}} (1-\gamma)\boldsymbol{\mu}^{\top}\mathbf{v} \quad \text{subject to} \\ &\mathbf{v}_{(s)} \geq \mathbf{r}_{(sa)} + \gamma P_{(sa,:)}\mathbf{v} \qquad \forall s, a \end{aligned} \tag{1}$$

其中 $\mu \in \mathbb{R}^{|S|}$ 是初始状态分布，$v \in \mathbb{R}^{|S|}$，$r \in \mathbb{R}^{|S||A|}$，$P \in \mathbb{R}^{|S||A| \times |S|}$。要证明它的解是规划问题的解，只需要证明相应的 v^* 满足 Bellman equation $\mathcal{T}v = v$。对于任意一个状态 s'，只要 $\mathcal{T}v(s') > v(s')$，我们都可以只减小 $v(s')$ 的值并且不与其他的约束条件冲突，考虑到 $\mu > 0$，这样的操作一定使得目标函数更优。反复这样操作可以得到 $\mathcal{T}v^* = v^*$。

相应的最优策略可以写为

$$
\begin{aligned}
a^*(s) &= \arg\max_a \mathbf{r}_{(sa)} + \gamma P_{(sa,:)}\mathbf{v}^* \\
\boldsymbol{\pi}^*_{(sa)} &= \begin{cases} 1 & \text{if } a = a^*(s) \\ 0 & \text{if } a \neq a^*(s) \end{cases}
\end{aligned} \tag{2}
$$

简要地来说，转化为对偶问题的步骤是，先写出 LP 的拉格朗日函数 L，然后令 L 对原变量求导为零，把拉格朗日乘子作为新的变量。相应的对偶关系可以说明对偶问题的最优解和原问题最优解之间的关系。下面就按照这些步骤来做。

其拉格朗日函数可以写为

$$
L(\mathbf{v}, \mathbf{d}) = (1-\gamma)\boldsymbol{\mu}^\top \mathbf{v} + \mathbf{d}^\top(\mathbf{r} + \gamma P\mathbf{v} - \Xi^\top \mathbf{v}), \;\; \mathbf{d} \geq 0
$$

其中 $d \in \mathbb{R}^{|S||A|}$ 为拉格朗日乘子，$\Xi \in \mathbb{R}^{|S| \times |S||A|}$ 只含有0和1，只是把原问题里面的约束条件表示为矩阵形式。

拉格朗日函数对 v 求导并且令其为零，可以得到

$$
\Xi\mathbf{d} = (1-\gamma)\boldsymbol{\mu} + \gamma P^\top \mathbf{d}
$$

将其代回拉格朗日函数，可以得到相应的对偶问题

$$
\begin{aligned}
&\max_{\mathbf{d}} \mathbf{d}^\top \mathbf{r} \quad \text{subject to} \\
&\mathbf{d} \geq 0, \;\; \Xi\mathbf{d} = (1-\gamma)\boldsymbol{\mu} + \gamma P^\top \mathbf{d}
\end{aligned} \tag{3}
$$

对偶问题里面的约束条件要求 d 是一个在 state action 上的概率分布，即

Lemma 1: If $\mathbf{d}$ satisfies the constraint in (3) then $\mathbf{1}^\top \mathbf{d} = 1$.

Proof: First note that $\mathbf{1}^\top \mathbf{d} = \mathbf{1}^\top \Xi \mathbf{d}$, where the first $\mathbf{1}$ is $|S||A| \times 1$ and the second is $|S| \times 1$. Then one can determine that $\mathbf{1}^\top \Xi \mathbf{d} = (1-\gamma)\mathbf{1}^\top \boldsymbol{\mu} + \gamma \mathbf{1}^\top P^\top \mathbf{d} = (1-\gamma)1 + \gamma 1 = 1$.

根据对偶关系，最优策略和对偶问题的最优解有如下关系

$$\boldsymbol{\pi}^*_{(sa)} = \frac{\mathbf{d}^*_{(sa)}}{\sum_a \mathbf{d}^*_{(sa)}} \tag{4}$$

对偶问题的最优解其实就是从规定初始状态开始、遵循最优策略，相应得到的 discounted state-action visits。

2. Policy evaluation （state value function / V function）

给定一个策略，把它表示为一个 $|\mathcal{S}| \times |\mathcal{S}||\mathcal{A}|$ 的矩阵

$$\Pi_{(s,s'a)} = \begin{cases} \boldsymbol{\pi}_{(sa)} & \text{if } s' = s \\ 0 & \text{if } s' \neq s \end{cases}$$

考虑到 P 是一个 $|\mathcal{S}||\mathcal{A}| \times |\mathcal{S}|$ 的矩阵，由此 ΠP 表示在相应策略下的一步状态转移矩阵， $P\Pi$ 表示在相应策略下的一步 state-action 转移矩阵。

V函数可以表示为

$$\mathbf{v} = \sum_{i=0}^{\infty} \gamma^i (\Pi P)^i \Pi \mathbf{r} \tag{5}$$

也可以表示为 $v = (I - \gamma \Pi P)^{-1} \Pi r$ 。

如果令（即前面所说的 successor representation，不过相差了一个常系数 $1-\gamma$ ）

$$M = (1-\gamma) \sum_{i=0}^{\infty} \gamma^i (\Pi P)^i \tag{9}$$

那么V函数就可以由它表示出来： $v = (1-\gamma)^{-1} M \Pi r$ 。

3. Policy evaluation （state action value function / Q function）

Q函数可以表示为

$$\mathbf{q} = \sum_{i=0}^{\infty} \gamma^i (P\Pi)^i \mathbf{r} \tag{10}$$

它满足如下递归关系 $q = r + \gamma P \Pi q$ ，和相应V函数之间满足关系 $v = \Pi q$ 。

令（state-action形式下的 successor representation）

$$H = (1-\gamma) \sum_{i=0}^{\infty} \gamma^i (P\Pi)^i \tag{14}$$

那么Q函数就可以由它表示出来： $q = (1-\gamma)^{-1} H r$ 。同时，它和SR有如下关系 $M\Pi = \Pi H$ 。

4. Policy iteration

Policy iteration 包含 policy evaluation 和 policy improvement，前者在前面两个部分已经说明了，policy improvement 需要每次采取使当前状态下Q函数最大的行动，或者采取一个行动使得其到达的下一个状态有更大的V函数值。即

$$\begin{aligned} a^*(s) &= \arg\max_a \mathbf{q}_{(sa)} \\ &= \arg\max_a \mathbf{r}_{(sa)} + \gamma P_{(sa,:)} \mathbf{v} \end{aligned} \tag{15}$$

写成对偶形式有

$$\begin{aligned} a^*(s) &= \arg\max_a H_{(sa,:)} \mathbf{r} \\ &= \arg\max_a (1-\gamma) \mathbf{r}_{(sa)} + \gamma P_{(sa,:)} M \Pi \mathbf{r} \end{aligned} \tag{17}$$

最后可以得到对偶形式下的policy iteration。

1. Initialization
   $M \leftarrow$ a matrix with rows that are probability distributions
   $\Pi \leftarrow$ arbitrary policy π
2. Policy Evaluation
   Solve for M in $M = (1-\gamma) I + \gamma M \Pi P$
3. Policy Improvement
   policy-stable $\leftarrow$ true
   For each $s \in \mathcal{S}$
   best-action $\leftarrow a$ s.t. $\pi_{(sa)} = 1$
   $a^*(s) = \arg\max_a (1-\gamma) \mathbf{r}_{(sa)} + \gamma P_{(sa,:)} M \Pi \mathbf{r}$
   $$\pi_{(sa)} = \begin{cases} 1 & \text{if } a = a^*(s) \\ 0 & \text{if } a \neq a^*(s) \end{cases}$$
   If *best-action* $\neq a^*(s)$, then *policy-stable* $\leftarrow$ false
   If *policy-stable*, then stop; else go to 2

Algorithm 2: The dual policy iteration algorithm

5. TD evaluation / SARSA / Q-learning

前面都是考虑 planning 的情形，现在考虑 learning 的情形。不难得到，SR 可以通过 TD-learning 的方式来学习。

$$M_{(s,:)} \leftarrow (1-\alpha)M_{(s,:)} + \alpha\left[(1-\gamma)\mathbf{1}_s^\top + \gamma M_{(s',:)}\right] \quad (23)$$

SARSA 分为 policy evaluation 和 policy improvement，其中 policy evaluation 步骤更新估计 on-policy 的 SR，可以写为如下形式

$$H_{(sa,:)} \leftarrow (1-\alpha)H_{(sa,:)} + \alpha\left[(1-\gamma)\mathbf{1}_{sa}^\top + \gamma H_{(s'a',:)}\right]$$

where $\mathbf{1}_{sa}$ is the vector of all zeros except for a 1 in the sa^{th} position.

Initialize H with rows that are probability distributions

Repeat (for each episode):
  Initialize s arbitrarily
  Choose a from s using policy derived from $H_{(sa,:)}$
  (e.g. ϵ-greedy where $a_{greedy} = \arg\max_a H_{(sa,:)}\mathbf{r}$)
  Repeat (for each step of episode):
    Take action a and observe the reward r and next state s'
    Choose a' from s' using policy derived from $H_{(s'a',:)}$
    (e.g. ϵ-greedy)
    $H_{(sa,:)} \leftarrow (1-\alpha)H_{(sa,:)} + \alpha\left[(1-\gamma)\mathbf{1}_{sa}^\top + \gamma H_{(s'a',:)}\right]$
    $s \leftarrow s'$ $a \leftarrow a'$
  until s is terminal

Algorithm 6: The dual Sarsa algorithm

Q-learning 在更新的过程中隐含了 policy improvement 步骤，可以写为

$$H_{(sa,:)} \leftarrow (1-\alpha)H_{(sa,:)} + \alpha\left[(1-\gamma)\mathbf{1}_{sa}^{\top} + \gamma H_{(s'a'^{*},:)}\right]$$
where $a'^{*} = \arg\max_{a'} H_{(s'a',:)}\mathbf{r}$.

Initialize H with rows that are probability distributions

Repeat (for each episode):
  Initialize s arbitrarily
  Repeat (for each step of episode):
    Choose a from s using policy derived from $H_{(sa,:)}$
    (e.g. ϵ-greedy where $a_{greedy} = \arg\max_a H_{(sa,:)}\mathbf{r}$)
    Take action a and observe the reward r and next state s'
    $H_{(sa,:)} \leftarrow (1-\alpha)H_{(sa,:)} + \alpha\left[(1-\gamma)\mathbf{1}_{sa}^{\top} + \gamma H_{(s'a'^{*},:)}\right]$
    where $a'^{*} = \arg\max_{a'} H_{(s'a',:)}\mathbf{r}$
    $s \leftarrow s'$
  until s is terminal

Algorithm 8: The dual Q-learning algorithm

发布于 2019-06-18

强化学习 (Reinforcement Learning)

文章被以下专栏收录

强化学习前沿
读呀读paper

进入专栏