An heuristic approach to Markov decision processes based on the Interior point method

Wang, J.

Jianfu Wang

An heuristic approach to Markov decision processes based on the Interior point method

Master thesis, defended on August 26, 2008

Thesis advisor: Prof. dr. Lodewijk Kallenberg

Mathematisch Instituut, Universiteit Leiden

Contents

Chapter 0 Introduction

0.1 Standard method of MDPs

There are three main methods for MDPs: Policy iteration, Linear programming and Value iteration. We will give a short introduction for these three methods first.

Policy iteration

In the method of policy iteration, we constructed a sequence of deterministic policies, which have increasing value vectors. As the space of deterministic policies is finite, this method will terminate with an optimal policy within a finite number of iterations. The optimal value vector will be generated as by-product.

Linear programming

This method transforms the MDP models into a linear programming problem. Furthermore, there is a correspondence between extreme feasible points of the linear programming problem and deterministic policies of the MDP model. Hence once we get the optimal solution of the linear programming problem, we get the optimal deterministic policy for the MDP model. In this thesis, we will only consider linear programming method for MDPs.

Value iteration

Converse to the policy iteration, the value iteration focuses on value vectors. In this method, the value vector is successively approximated, starting with some guess v^1, by a sequence $\{v^n\}_{n=1}^{\infty}$, which converges to the optimal value vector. This method is also called *successive approximation*. Finally, we will get a value vector, whose distance to the optimal value vector is smaller than a given accuracy parameter ε. A so-called ε-optimal policy is constructed as a by-product.

0.2 Heuristic approach to MDPs based on the IPM

IPM is an efficient method to solve linear programming problem. The general idea about using IPM to solve MDPs is: get an ε-optimal solution of the linear programming problem from IPM, and get a corresponding ε-optimal policy. However, in MDPs, nearly always we can get a better result: an optimal deterministic policy, and also quicker.

The idea is: once we get a feasible solution in the linear programming problem with IPM, we transform it into a stationary policy. Based on this policy, we make a new heuristic policy. Then, we can do several tests to check whether this heuristic policy is an optimal policy. If it is not, we just go some more steps in IPM, until the heuristic policy changes, and check again.

Because of some unique properties of MDPs, this heuristic method works very fast.

In this thesis, we start with the MDPs models and two important criteria: total expected discounted rewards and average expected rewards. In Chapter 2, we will introduce the Interior point method based on Self-concordant functions, which can be used for solving the Linear programming problem in Chapter 1. Chapter 3 will deal with how to make an heuristic approach in the IPM to solve the LP problem in Chapter 1. Appendix A contains some technical lemmas, and in Appendix B the codes are given. Some numerical results are reported in Appendix C.

Chapter 1 Introduction to Markov decision processes

In this chapter, we introduce the model of a Markov decision process (MDP) and we present several optimality criteria.

1.1 The MDP model

1. State space

At any time point at which a decision has to be made, the state of the system is observed by the decision maker. The set of possible states is called the state space. Although the state space could be finite, denumerable, compact or even more general, in this study we only consider the MDP model with finite state space. The state space will be denoted by $S = \{1,2,...,N\}$.

2. Action sets

When the decision maker observes that the state of the system is state i, he chooses an action from a certain action set, which may depend on the observed state: the action set in state i is denoted by $A(i)$. Similarly to the state space, we assume that the action sets are finite.

3. Decision time points

The time intervals between the decision points may be constant or random. In the first case the model is said to be a Markov decision process; when the times between consecutive decision points are random the problem is called a semi-Markov decision problem. In this thesis, we restrict ourselves to Markov decision processes.

4. Rewards

Given the state of the system and the chosen action, an immediate reward is earned. Such reward only depends on the decision time point, the observed state and the chosen action and not on the history of the process. The immediate reward at decision time point t for an action a in state i will be denoted by $r_i^t(a)$; if the reward is independent of the time t, we denote $r_i(a)$ instead of $r_i^t(a)$. In this study we consider only stationary rewards.

5. Transition probabilities

Given the state of the system and the chosen action, the state at the next decision time point is determined by a transition law. These transitions only depend on the decision time point, the observed state and the chosen action and not on the history of the process. This property is called the Markov property. If the transitions depend on the decision time point, the problem is said to be non-stationary, and by $p_{ij}^t(a)$ the probability denotes that the next state is state j, given that the state at time t is state i and that action a is chosen. If the transitions are independent of the time points, the problem is called stationary, and the transition probabilities are denoted by $p_{ij}(a)$.

In this thesis we restrict ourselves to stationary transitions.

6. Planning horizon

This process has a planning horizon. This horizon may be finite, infinite or with random length. In this study the planning horizon will be infinite.

7. Optimality criterion

The objective is to determine a policy, i.e. a decision rule for each decision time point and each history of the process, which optimizes the performance of the system. The performance is measured by a utility function. This function assigns to each policy, given the starting state of the process, a value. In this thesis, we consider criteria based on discounted and average rewards.

1.2 Policies and Optimality criteria

1.2.1 Policies

A policy R is a sequence of decision rules: $R = (\pi^1, \pi^2, ..., \pi^t, ...)$, where π^t is the decision rule at time point $t, t = 1,2,....$ the decision rule π^t may depend on all information of the system until time t, i.e. on the states at the time points $1,2,...,t$ and the actions at the time points $1,2,...,t-1$. The formal definition of a policy is as follows.

Let $S \times A = \{(i,a) \mid i \in S, a \in A(i)\}$ and let H_t denote the set of the possible histories of the system up to time point t, i.e.

$$H_t = \{(i_1, a_1, ..., i_{t-1}, a_{t-1}, i_t) \mid (i_k, a_k) \in S \times A, 1 \le k \le t-1; i_t \in S\}. \quad (1.1)$$

A decision rule π^t at time point t gives the probability, as a function of the history H_t to the action space, of choosing action a, i.e.

$$\pi^t_{h_t a_t} \ge 0 \text{ for every } a_t \in A(i_t) \text{ and } \sum_{a_t} \pi^t_{h_t a_t} = 1 \text{ for every } h_t \in H_t. \quad (1.2)$$

Let C denote the set of all policies. A policy is said to be Markov if the decision rule π^t is independent of $(i_1, a_1, ..., i_{t-1}, a_{t-1})$ for every $t \in N$. Hence, in a Markov policy the decision rule at time t only depends on the state i_t; therefore the notation $\pi^t_{i_t a_t}$ is used. Let $C(M)$ be the set of Markov policies. If a policy is a Markov policy and the decision rules are independent of the time point t, i.e. $\pi^1 = \pi^2 = ...$, then the policy is called stationary. Hence, a stationary policy is determined by a nonnegative function π on $S \times A$ such that $\sum_a \pi_{ia} = 1$ for every $i \in S$. The

stationary policy $R=(\pi,\pi,...)$ is denoted by π^{∞}, and the set of stationary policies by $C(S)$. If the decision rule π of a stationary policy π^{∞} is nonrandomized, i.e. for every $i\in S$, we have $\pi_{ia}=1$ for exactly one action a_i (consequently $\pi_{ia}=0$ for every $a\neq a_i$), then the policy is called deterministic. Therefore, a deterministic policy can be described by a function f on S, where $f(i)$ is the chosen action a_i, $i\in S$. A deterministic policy is denoted by f^{∞} and the set of deterministic policies by $C(D)$.

A matrix $P=(p_{ij})$ is a transition matrix if $p_{ij}\geq 0$ for all (i,j) and $\sum_j p_{ij}=1$ for all i. Markov policies, and consequently also stationary and deterministic policies, induce transition matrices.

Assumption 1.1
In the following chapters, we only consider stationary policies, that means the immediate rewards and the transition probabilities are stationary, and denoted by $r_i(a)$ and $p_{ij}(a)$, respectively, for all i,j and a.

For the stationary policy $R=(\pi,\pi,...)$ the transition matrix $P(\pi)$ and the reward vector $r(\pi)$ are defined by

$$P(\pi)_{ij}=\sum_a p_{ij}(a)\pi_{ia} \quad \text{for every } (i,j)\in S\times S; \tag{1.3}$$

$$r(\pi)_i=\sum_a r_i(a)\pi_{ia} \quad \text{for every } i\in S. \tag{1.4}$$

Let the random variables X_t and Y_t denote the state and action at time t, $t=1,2,...$. For any policy R and any initial distribution β, i.e. β_i is the probability that the system starts in state i, let $P_{\beta,R}\{X_t=j,Y_t=a\}$ be the notation for the probability that at time t the state is j and the action is a. If $\beta_i=1$ for some $i\in S$, then we write $P_{i,R}$ instead of $P_{\beta,R}$. The expectation operator with respect to the probability measure $P_{\beta,R}$ or $P_{i,R}$ is denoted by $E_{\beta,R}$ or $E_{i,R}$ repectively.

1.2.2 Optimality criteria

Total expected discounted rewards over an infinite horizon

An amount r earned at time point 1 can be deposited in a bank with interest rate ρ. Then this amount grows and becomes $(1+\rho)\cdot r$ at time point 2, $(1+\rho)^2\cdot r$ at time point 3, etc. Hence, an amount r at time point 1 is comparable with $(1+\rho)^{t-1}\cdot r$ at time point t, $t=1,2,\ldots$.

Let $\alpha=(1+\rho)^{-1}$, called the discount factor. Note that $\alpha\in(0,1)$. Then, conversely, an amount r received at time point t can be considered as equivalent to an amount $\alpha^{t-1}\cdot r$ at time point 1. The total expected α-discounted rewards, given initial state i and a policy R, is denoted by $v_i^{\alpha}(R)$ and defined by

$$v_i^{\alpha}(R)=\sum_{t=1}^{\infty}E_{i,R}\{\alpha^{t-1}\cdot r_{X_t}(Y_t)\}=\sum_{t=1}^{\infty}\alpha^{t-1}\sum_{j,a}P_{i,R}\{X_t=j,Y_t=a\}\cdot r_j(a).$$

For a stationary policy π^{∞}, we have:

$$v^{\alpha}(\pi^{\infty})=\sum_{t=1}^{\infty}\alpha^{t-1}P(\pi)^{t-1}r(\pi).$$

The value-vector v^{α} and the optimality of a policy R_* are defined by

$$v^{\alpha}:=\sup_R v^{\alpha}(R) \qquad \text{and} \qquad v^{\alpha}(R_*):=v^{\alpha}.$$

In the following section, it will be shown that there exists an optimal deterministic policy f_*^{∞} for this criterion and that the value vector v^{α} is the unique solution of the so-called optimality equation

$$x_i=\max_{a\in A(i)}\{r_i(a)+\alpha\sum_j p_{ij}(a)x_j\},\ i\in S.$$

Furthermore, it will be shown that f_*^{∞} is an optimal policy if

$$r_i(f_*)+\alpha\sum_j p_{ij}(f_*)v_j^{\alpha}\geq r_i(a)+\alpha\sum_j p_{ij}(a)v_j^{\alpha},\ a\in A(i),i\in S.$$

Average expected reward over an infinite horizon

In the criterion of average rewards the limiting behavior of $\frac{1}{T}\sum_{t=1}^{T} r_{X_t}(Y_t)$ is considered for $T \to \infty$. Since $\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} r_{X_t}(Y_t)$ may not exist and interchanging limit and expectation is not allowed, in general, there are four different evaluation measures which can be considered:

1. Lower limit of the average expected rewards:

$$\phi_i(R) = \liminf_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} E_{i,R}\{r_{X_t}(Y_t)\},\ i \in S \text{, with value vector } \phi = \sup_R \phi(R).$$

2. Upper limit of the average expected rewards:

$$\overline{\phi}_i(R) = \limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} E_{i,R}\{r_{X_t}(Y_t)\},\ i \in S \text{, with value vector } \overline{\phi} = \sup_R \overline{\phi}(R).$$

3. Expectation of the lower limit of the average rewards:

$$\psi_i(R) = E_{i,R}\{\liminf_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} r_{X_t}(Y_t)\},\ i \in S \text{, with value vector } \psi = \sup_R \psi(R).$$

4 Expectation of the upper limit of the average rewards:

$$\overline{\psi}_i(R) = E_{i,R}\{\liminf_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} r_{X_t}(Y_t)\},\ i \in S \text{, with value vector } \overline{\psi} = \sup_R \overline{\psi}(R).$$

Lemma 1.1

$\psi(R) \le \phi(R) \le \overline{\phi}(R) \le \overline{\psi}(R)$ for every policy R.

Proof

The second inequality is obvious. The first and the last inequality follow from Fatou's lemma (e.g. Bauer [1], p.126):

$$\psi_i(R) = E_{i,R}\{\liminf_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} r_{X_t}(Y_t)\} \le \liminf_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} E_{i,R}\{r_{X_t}(Y_t)\} = \phi_i(R)$$

and

$$\overline{\phi}_i(R) = \limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} E_{i,R}\{r_{X_t}(Y_t)\} \le E_{i,R}\{\liminf_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} r_{X_t}(Y_t)\} = \overline{\psi}_i(R).$$

For these 4 criteria the value vector and the concept of an optimal policy can be defined in the usual way. In Bierth [2] is shown that

$$\psi(\pi^\infty) = \phi(\pi^\infty) = \overline{\phi}(\pi^\infty) = \overline{\psi}(\pi^\infty) \quad \text{for every deterministic policy } \pi^\infty,$$

and that for all 4 criteria there exists a deterministic optimal policy. Hence, the 4 criteria are equivalent in the sense that an optimal deterministic policy for one criterion is also optimal for the

others.

1.3 Discounted Rewards

1.3.1 Introduction

This section deals with the total expected discounted reward over an infinite planning horizon. This criterion is quite natural when the planning horizon is rather large and returns at the present time are of more value than returns of the same value which are earned later in time. We recall that the total expected $\alpha-$discounted rewards, given initial state i and a stationary policy π^{∞}, is denoted by $v_i^{\alpha}(\pi^{\infty})$ and satisfies

$$v^{\alpha}(\pi^{\infty}) = \sum_{t=1}^{\infty} \alpha^{t-1} P(\pi)^{t-1} r(\pi) = \{I - \alpha P(\pi)\}^{-1} r(\pi).$$

The second equation follows from

$$\{I - \alpha P(\pi)\} \cdot \{I + \alpha P(\pi) + \cdots + \{\alpha P(\pi)\}^{t-1}\} = I - \{\alpha P(\pi)\}^{t}$$

and

$$\{\alpha P(\pi)\}^{t} \to 0 \quad \text{for} \quad t \to \infty.$$

In the next section, we first show some theorems of monotone contraction mappings in the context of MDPs without proof. For the proof we refer to Kallenberg [9]. Then, the optimality equation, bounds for the value vector and suboptimal actions are considered. Finally, the linear programming method is introduced.

1.3.2 Monotone contraction mappings

Let X be a Banach space with norm $\|\cdot\|$, and let $B: X \to X$. The operator B is called a contraction mapping if for some $\beta \in (0,1)$

$$\| Bx - By \| \leq \beta \| x - y \| \quad \text{for all} \quad x, y \in X. \tag{1.5}$$

The number β is called the contraction factor of B. An element $x \in X$ is said to be a fixed-point of B if $Bx^* = x^*$. The next theorem shows the existence of a unique fixed-point for a contraction mapping in a Banach space.

Theorem 1.1 (Fixed-point Theorem)

Let X be a Banach space and suppose $B: X \to X$ is a contraction mapping. Then,

(1) $x^* = \lim_{n\to\infty} B^n x$ exists for every $x \in X$, and x^* is a fixed-point of B.

(2) x^* is the unique fixed-point of B.

The next theorem gives bounds on the distance between the fixed-point x^* and iterations $B^n x$ for $n = 0,1,2,\ldots$.

Theorem 1.2

Let X be a Banach space and suppose $B: X \to X$ is a contraction mapping with contraction factor β and fixed-point x^*. Then,

(1) $\| x^* - B^n x \| \le \beta(1-\beta)^{-1} \| B^n x - B^{n-1} x \| \le \beta^n (1-\beta)^{-1} \| Bx - x \|, \ \forall x \in X, n \in N$;

(2) $\| x^* - x \| \le (1-\beta)^{-1} \| Bx - x \|, \ \forall x \in X$.

Remark:

The above theorem implies that the convergence rate of $B^n x$ to the fixed-point is at least linear. (cf. Stoer and Bulirsch [13], p.251). This kind of convergence is called geometric convergence.

Let X be a partially ordered set and $B: X \to X$. The mapping B is called monotone if $x \le y$ implies $Bx \le By$.

Theorem 1.3

Let X be a partially ordered Banach space. Suppose that $B: X \to X$ is a monotone contraction mapping with fixed-point x^*. Then

(1) $Bx \le x$ implies $x^* \le Bx \le x$;

(2) $Bx \ge x$ implies $x^* \ge Bx \ge x$.

Lemma 1.2

(1) Let $B: R^N \to R^N$ be a monotone contraction mapping with contraction factor β, and let d be a scalar. Then $x \le y + d \cdot e$ implies $Bx \le By + \beta \cdot | d | \cdot e$.

(2) Let $B: R^N \to R^N$ be a mapping with the property that $x \le y + d \cdot e$ implies $Bx \le By + \beta \cdot | d | \cdot e$ for some $0 \le \beta < 1$ and for all scalars d. Then B is a monotone

contraction, with respect to the supremum norm, with contraction factor β.

Lemma 1.3

Let $B: R^N \to R^N$ be a monotone contraction mapping, with respect to the supremum norm, with contraction factor β and fixed-point x^*. Suppose that there exist scalars a and b such that $a \cdot e \le Bx - x \le b \cdot e$ for some $x \in R^N$. Then,

$$x - (1-\beta)^{-1} |a| \cdot e \le Bx - \beta(1-\beta)^{-1} |a| \cdot e \le x^* \le Bx + \beta(1-\beta)^{-1} |b| \cdot e \le x + (1-\beta)^{-1} |b| \cdot e.$$

Corollary 1.1

Let B be a monotone contraction in R^N, with respect to the supermum norm $\|\cdot\|_\infty$, with contraction factor β and fixed-point x^*. Then

$$x - (1-\beta)^{-1} \cdot \| Bx - x \|_\infty \cdot e \le Bx - \beta(1-\beta)^{-1} \cdot \| Bx - x \|_\infty \cdot e \le x^*$$
$$\le Bx + \beta(1-\beta)^{-1} \cdot \| Bx - x \|_\infty \cdot e \le x + (1-\beta)^{-1} \cdot \| Bx - x \|_\infty \cdot e.$$

Lemma 1.4

Let $B: R^N \to R^N$ be a monotone contraction in R^N, with respect to the supremum norm, with contraction factor β, fixed-point x^* and with the property that $B(x + c \cdot e) = Bx + \beta c \cdot e$ for every $x \in R^N$ and scalar c.

Suppose that there exist scalars a and b such that $a \cdot e \le Bx - x \le b \cdot e$ for some $x \in R^N$. Then,

$$x + (1-\beta)^{-1} a \cdot e \le Bx + \beta(1-\beta)^{-1} a \cdot e \le x^* \le Bx + \beta(1-\beta)^{-1} b \cdot e \le x - (1-\beta)^{-1} b \cdot e.$$

1.3.3 The optimality equation

Suppose that at time point $t = 1$, when the system is in state i, action $a \in A(i)$ is chosen, and that from $t = 2$ on an optimal policy is followed. Then, the total expected α-discounted reward is equal to $r_i(a) + \alpha \sum_j p_{ij}(a) v_j^\alpha$. Since any optimal policy obtains at least this amount, we have

$$v_i^\alpha \ge \max_{a \in A(i)} \{ r_i(a) + \alpha \sum_j p_{ij}(a) v_j^\alpha \}, \; i \in S.$$

On the other hand, let a_i be the action chosen by an optimal policy in state i. Then,

$$v_i^\alpha = r_i(a_i) + \alpha \sum_j p_{ij}(a_i) v_j^\alpha \le \max_{a \in A(i)} \{ r_i(a) + \alpha \sum_j p_{ij}(a) v_j^\alpha \},\ i \in S .$$

Hence, v^α is a solution of

$$x_i^\alpha = \max_{a \in A(i)} \{ r_i(a) + \alpha \sum_j p_{ij}(a) x_j^\alpha \},\ i \in S . \tag{1.6}$$

According to the contraction mapping theory in section 1.3.2, v^α is a fixed-point of the mapping $U : R^N \to R^N$, defined by

$$(Ux)_i = \max_{a \in A(i)} \{ r_i(a) + \alpha \sum_j p_{ij}(a) x_j \},\ i \in S . \tag{1.7}$$

Besides the mapping U, defined above, we introduce for any randomized decision rule π a mapping $L_\pi : R^N \to R^N$, defined by

$$L_\pi x = r(\pi) + \alpha P(\pi) x . \tag{1.8}$$

Let $f_x(i)$ be such that

$$r_i(f_x(i)) + \alpha \sum_j p_{ij}(f_x(i)) x_j = \max_{a \in A(i)} \{ r_i(a) + \alpha \sum_j p_{ij}(a) x_j \},\ i \in S .$$

Then,

$$L_{f_x} x = Ux = \max_f L_f x ,$$

where the maximization is taken over all deterministic decision rules f.

Let $\| P(\pi) \|_\infty$ be the subordinate matrix norm (cf. Stoer and Bulirsch [13], p.178), then $\| P(\pi) \|_\infty$ satisfies

$$\| P(\pi) \|_\infty = \max_i \sum_j p_{ij}(\pi) = 1 .$$

Theorem 1.4

The mapping L_π and U are monotone contraction mappings with contraction factor α.

Proof

Suppose that $x \ge y$. Let π be any stationary decision rule. Because $P(\pi) \ge 0$,

$$L_\pi x = r(\pi) + \alpha P(\pi) x \ge r(\pi) + \alpha P(\pi) y = L_\pi y , \tag{1.9}$$

i.e. L_π is monotone. U is also monotone, since

$$Ux = \max_f L_f x \ge L_{f_y} x \ge L_{f_y} y = Uy .$$

Furthermore, we obtain

$$\| L_\pi x - L_\pi y \|_\infty = \| \alpha P(\pi)(x-y) \|_\infty \le \alpha \| P(\pi) \|_\infty \| (x-y) \|_\infty = \alpha \cdot \| (x-y) \|_\infty ,$$

i.e. L_π is a contraction mapping with contraction factor α. The derivation for operatior U is

$$Ux - Uy = L_{f_x} x - L_{f_y} y \le L_{f_x} x - L_{f_x} y = \alpha \cdot P(f_x)(x-y) \le \alpha \cdot \| x - y \|_\infty \cdot e . \quad (1.10)$$

Interchanging x and y yields

$$Uy - Ux \le \alpha \cdot \| y - x \|_\infty \cdot e . \quad (1.11)$$

From (1.10) and (1.11) in follows that $\| Ux - Uy \|_\infty \le \alpha \cdot \| x - y \|_\infty$, i.e. U is a contraction mapping with contraction factor α.

The next theorem shows that for any randomized decision rule π, the total expected $\alpha -$ discounted reward of the policy π^∞ is the fixed-point of the mapping L_π.

Theorem 1.5

$v^\alpha(\pi^\infty)$ is the unique solution of the functional equation $L_\pi x = x$.

Proof

Theorem 1.1 and Theorem 1.4 imply that it is sufficient to show that $L_\pi v^\alpha(\pi^\infty) = v^\alpha(\pi^\infty)$.

We have

$$\begin{aligned} L_\pi v^\alpha(\pi^\infty) - v^\alpha(\pi^\infty) &= r(\pi) - \{I - \alpha P(\pi)\} v^\alpha(\pi^\infty) \\ &= r(\pi) - \{I - \alpha P(\pi)\}\{I - \alpha P(\pi)\}^{-1} r(\pi) = 0. \end{aligned}$$

Corollary 1.2

$v^\alpha(\pi^\infty) = \lim_{n\to\infty} L_\pi^n x$ for any $x \in R^N$.

The next theorem shows that the value vector v^α is the fixed-point of the mapping U.

Theorem 1.6

v^α is the unique solution of the functional equation $Ux = x$.

Proof

It is sufficient to show that $Uv^\alpha = v^\alpha$. Let $R = (\pi^1, \pi^2, ...)$ be an arbitrary Markov policy. Then,

$$v^{\alpha}(R) = r(\pi^1) + \sum_{t=2}^{\infty} \alpha^{t-1} P(\pi^1)P(\pi^2)\cdots P(\pi^{t-1})r(\pi^t)$$
$$= r(\pi^1) + \alpha P(\pi^1)\sum_{s=1}^{\infty} \alpha^{s-1} P(\pi^2)P(\pi^3)\cdots P(\pi^s)r(\pi^{s+1})$$
$$= r(\pi^1) + \alpha P(\pi^1)v^{\alpha}(R_2) = L_{\pi^1} v^{\alpha}(R_2),$$

where $R_2 = (\pi^2, \pi^3, ...)$. From the monotonicity of L_{π^1} and the definition of U, we obtain

$$v^{\alpha}(R) = L_{\pi^1} v^{\alpha}(R_2) \le L_{\pi^1} v^{\alpha} \le U v^{\alpha},\ R \in C(M).$$

Hence, $v^{\alpha} = \sup_{R \in C(M)} v^{\alpha}(R) \le U v^{\alpha}$. Take any $\varepsilon > 0$. Since $v^{\alpha} = \sup_{R \in C(M)} v^{\alpha}(R)$, for any $j \in S$ there exists a Markov policy $R_j^{\varepsilon} = (\pi^1(j), \pi^2(j), ...)$ such that $v_j^{\alpha}(R_j^{\varepsilon}) \ge v_j^{\alpha} - \varepsilon$.

Let $a_i \in A(i)$ be such that $r_i(a_i) + \alpha \sum_j p_{ij}(a_i) v_j^{\alpha} = \max_a \{r_i(a) + \alpha \sum_j p_{ij}(a) v_j^{\alpha}\},\ i \in S$.

Consider the policy $R^* = (\pi^1, \pi^2, ...)$ defined by

$$\pi_{ia}^1 = \begin{cases} 1 & \text{if a = a}_\text{i} \\ 0 & \text{otherwise} \end{cases} \quad \text{and } \pi_{i_1 a_1 \ldots i_t a}^{\text{t}} = \pi_{i_t a}^{\text{t-1}}(i_2),\ a \in A(i_t),\ t \ge 2,$$

i.e. R^* is the policy that chooses a_i in state i at time point $t = 1$, and if the state at time $t = 2$ is i_2, then the policy follows $R_{i_2}^{\varepsilon}$ where the process is considered as originating in state i_2.

Therefore,

$$v_i^{\alpha} \ge v_i^{\alpha}(R^*) = r_i(a_i) + \alpha \sum_j p_{ij}(a_i) v_j^{\alpha}(R_j^{\varepsilon}) \ge r_i(a_i) + \alpha \sum_j p_{ij}(a_i)(v_j^{\alpha} - \varepsilon)$$
$$= \max_a \{r_i(a) + \alpha \sum_j p_{ij}(a) v_j^{\alpha}\} - \alpha\varepsilon = (U v^{\alpha})_i - \alpha\varepsilon,\ i \in S.$$

Since $\varepsilon > 0$ is arbitrarily chosen, $v^{\alpha} \ge U v^{\alpha}$.

Because $v^{\alpha} = U v^{\alpha} = L_{f_{v^{\alpha}}} v^{\alpha}$, it follows from Theorem 1.5 that $v^{\alpha} = v^{\alpha}(f_{v^{\alpha}}^{\infty})$, i.e. $f_{v^{\alpha}}^{\infty}$ is an optimal policy. If $f^{\infty} \in C(D)$ satisfies

$$r_i(a) + \alpha \sum_j p_{ij}(f) v_j^{\alpha} = \max_a \{r_i(a) + \alpha \sum_j p_{ij}(a) v_j^{\alpha}\},\ i \in S,$$

then f^{∞} is called a conserving policy. Conserving polices are optimal. Therefore, the equation $Ux = x$ is called the optimality equation.

Corollary 1.3

(1) There exists a deterministic α-discounted optimal policy.

(2) $v^{\alpha} = \lim_{n\to\infty} U^n x$ for any $x \in R^N$.

(3) Any conserving policy is α-discounted optimal.

As already mentioned, we derive some bounds for the value vector v^{α}. These bounds can be obtained from Lemma 1.4. Therefore, we note that the mappings L_{π} and U satisfy, for any $x \in R^N$ and scalar c, $L_f(x + c\cdot e) = L_f x + \alpha c \cdot e$ and $U(x + c\cdot e) = Ux + \alpha c \cdot e$.

Theorem 1.7

For any $x \in R^N$, we have

(1) $x - (1-\alpha)^{-1} \| Ux - x \|_{\infty} \cdot e \le Ux - \alpha(1-\alpha)^{-1} \| Ux - x \|_{\infty} \cdot e \le v^{\alpha}(f_x^{\infty}) \le v^{\alpha} \le$

$$Ux + \alpha(1-\alpha)^{-1} \| Ux - x \|_{\infty} \cdot e \le x + (1-\alpha)^{-1} \| Ux - x \|_{\infty} \cdot e .$$

(2) $\| v^{\alpha} - x \|_{\infty} \le (1-\alpha)^{-1} \| Ux - x \|_{\infty}$.

(3) $\| v^{\alpha} - v^{\alpha}(f_x^{\infty}) \|_{\infty} \le 2\alpha(1-\alpha)^{-1} \| Ux - x \|_{\infty}$.

Proof

Take any $x \in R^N$. By Lemma 1.4, for $a = -\| Ux - x \|_{\infty}$, $b = \| Ux - x \|_{\infty}$ and $B = L_{f_x}$, we obtain (notice that $Bx = L_{f_x} x = Ux$),

$$x - (1-\alpha)^{-1} \| Ux - x \|_{\infty} \cdot e \le Ux - \alpha(1-\alpha)^{-1} \| Ux - x \|_{\infty} \cdot e \le v^{\alpha}(f_x^{\infty}) \le v^{\alpha} .$$

Next, again applying Lemma 1.4, for $B = U$ the remaining part of (1) implies,

$$v^{\alpha} \le Ux + \alpha(1-\alpha)^{-1} \| Ux - x \|_{\infty} \cdot e \le x + (1-\alpha)^{-1} \| Ux - x \|_{\infty} \cdot e .$$

The part (2) and (3) follow directly from part (1).

Theorem 1.8

For any $x \in R^N$, we have

(1) $x - (1-\alpha)^{-1} \min_i (Ux - x)_i \cdot e \le Ux - \alpha(1-\alpha)^{-1} \min_i (Ux - x)_i \cdot e \le v^{\alpha}(f_x^{\infty}) \le v^{\alpha} \le$

$$Ux + \alpha(1-\alpha)^{-1} \max_i (Ux - x)_i \cdot e \le x + (1-\alpha)^{-1} \max_i (Ux - x)_i \cdot e .$$

(2) $\| v^{\alpha} - v^{\alpha}(f_x^{\infty}) \|_{\infty} \le 2\alpha(1-\alpha)^{-1} span(Ux - x)$ where $span(y) := \max_i y_i - \min_i y_i$.

Proof

Notice that $\min_i (Ux - x)_i \cdot e \le Ux - x \le \max_i (Ux - x)_i \cdot e$. It is easy to verify that for $a = \min_i (Ux - x)_i$ and $b = \max_i (Ux - x)_i$ the proof is similar to the proof of Theorem 1.7.

Remark

Since $-\min_i (Ux - x)_i \le \| Ux - x \|_\infty$ and $\max_i (Ux - x)_i \le \| Ux - x \|_\infty$, we have $span(Ux - x) \le 2 \| Ux - x \|_\infty$. Consequently, the bounds of Theorem 1.8 are stronger than the bounds of Theorem 1.7.

Next, we discuss the elimination of suboptimal actions. An action $a \in A(i)$ is called suboptimal if there doesn't exist an α-discounted optimal policy $f^\infty \in C(D)$ with $f(i) = a$. Because f^∞ is α-discounted optimal if and only if $v^\alpha(f^\infty) = v^\alpha$, and because $v^\alpha = Uv^\alpha$, an action $a \in A(i)$ is suboptimal if and only if

$$v_i^\alpha > r_i(a) + \alpha \sum_j p_{ij}(a) v_j^\alpha , \tag{1.12}$$

Suboptimal actions can be disregarded. Notice that formula (1.12) is unuseful, because v^α is unknown. However, by upper and lower bounds on v^α as given in Theorem 1.7 and 1.8, suboptimal tests can be derived, as illustrated in the following theorem.

Theorem 1.9

Suppose that $x \le v^\alpha \le y$. If $r_i(a) + \alpha \sum_j p_{ij}(a) y_j < (Ux)_i$, then action $a \in A(i)$ is suboptimal.

Proof,

$$v_i^\alpha = (Uv^\alpha)_i \ge (Ux)_i > r_i(a) + \alpha \sum_j p_{ij}(a) y_j \ge r_i(a) + \alpha \sum_j p_{ij}(a) v_j^\alpha .$$

The first inequality follows from the monotonicity of U.

Corollary 1.4

Suppose that for some scalars b and c, we have $x + b \cdot e \le v^\alpha \le x + c \cdot e$. If

$$r_i(a) + \alpha \sum_j p_{ij}(a) x_j < (Ux)_i - \alpha(c - b) , \tag{1.13}$$

then action $a \in A(i)$ is suboptimal.

Proof

$$r_i(a) + \alpha \sum_j p_{ij}(a)(x_j + c) = r_i(a) + \alpha \sum_j p_{ij}(a)x_j + \alpha c < (Ux)_i + \alpha b = \{U(x + b \cdot e)\}_i .$$

Applying corollary 1.4 on the bound of Theorem 1.8, gives the following test for the elimination of a suboptimal action $a \in A(i)$:

$$r_i(a) + \alpha \sum_j p_{ij}(a)x_j < (Ux)_i - \alpha(1-\alpha)^{-1} span(Ux - x) . \qquad (1.14)$$

1.3.4 Linear programming

The value-vector v^α is the unique solution of the optimality equation (1.6), i.e.

$$v_i^\alpha = \max_{a \in A(i)} \{r_i(a) + \alpha \sum_j p_{ij}(a)v_j^\alpha\}, \ i \in S .$$

Hence v^α satisfies

$$v_i^\alpha \geq r_i(a) + \alpha \sum_j p_{ij}(a)v_j^\alpha \quad \text{for all} \quad (i,a) \in S \times A . \qquad (1.15)$$

Intuitively it is clear that v^α is the smallest vector which satisfies (1.15). This property is the key property for the linear programming approach to compute the value-vector. It turns out that an optimal policy can be obtained from the dual linear program. We also show a one-to-one correspondence between the stationary policies and the feasible solutions of the dual program, such that the extreme points correspond to the deterministic policies. Furthermore, we show that the exclusion of suboptimal actions can be included in the linear programming method.

A vector $v \in R^N$ is said to be α-superharmonic if

$$v_i \geq r_i(a) + \alpha \sum_j p_{ij}(a)v_j \quad \text{for all} \quad (i,a) \in S \times A . \qquad (1.16)$$

Theorem 1.10

v^α is the smallest α-superharmonic vector.

Proof

Since $v_i^\alpha = \max_{a \in A(i)} \{r_i(a) + \alpha \sum_j p_{ij}(a)v_j^\alpha\} \geq r_i(a) + \alpha \sum_j p_{ij}(a)v_j^\alpha$ for all $(i,a) \in S \times A$,

v^α is α-superharmonic. Suppose that $v \in R^N$ is also α-superharmonic. Then

$$v \geq r(a) + \alpha P(f)v \quad \text{for every} \quad f^\infty \in C(D) ,$$

which implies $\{I-\alpha P(f)\}v \geq r(f)$. Since $\{I-\alpha P(f)\}^{-1} = \sum_{t=0}^{\infty} \alpha^t P^t(f) \geq 0$, we obtain

$$v \geq \{I-\alpha P(f)\}^{-1} r(f) = v^{\alpha}(f^{\infty}),\ f^{\infty} \in C(D).$$

Hence, $v_i^{\alpha} = \max_f v^{\alpha}(f^{\infty}) \leq v$, i.e. v^{α} is the smallest α-superharmonic vector.

Corollary 1.5

v^{α} is the unique optimal solution of the linear programming problem

$$\min\{\sum_j \beta_j v_j \mid \sum_j \{\delta_{ij} - \alpha p_{ij}(a)\} v_j \geq r_i(a),\ (i,a) \in S \times A\}, \tag{1.17}$$

where β_j is any strictly positive number for every $j \in S$.

Proof

From theorem 1.10 it follows that v^{α} is a feasible solution of (1.17) and that $v^{\alpha} \leq v$ for every feasible solution v of (1.17). Hence, v^{α} is the unique solution of (1.17).

By corollary 1.5, the value vector v^{α} can be found as optimal solution of the linear program (1.17). This program does not give an optimal policy. However, an optimal policy can be obtained from the solution of the dual program

$$\max\left\{\sum_{(i,a)} r_i(a) x_i(a) \;\middle|\; \begin{array}{r} \sum_{(i,a)} \{\delta_{ij} - \alpha p_{ij}(a)\} x_i(a) = \beta_j,\ j \in S \\ x_i(a) \geq 0, (i,a) \in S \times A \end{array}\right\}. \tag{1.18}$$

Theorem 1.11

(1) Any feasible solution x of (1.18) satisfies $\sum_a x_j(a) > 0,\ j \in S$

(2) The dual program (1.18) has a finite optimal solution, say x^*.

(3) Any $f_*^{\infty} \in C(D)$ with $x_i^*(f_*(i)) > 0$ for every $i \in S$ is an α-discounted optimal policy.

Proof

(1) Let x be a feasible solution of (1.18). From the constraints of (1.18) it follows that

$$\sum_a x_j(a) = \beta_j + \alpha \sum_{(i,a)} p_{ij}(a) x_i(a) \geq \beta_j > 0,\ j \in S.$$

(2) Since the primal program (1.17) has a finite optimal solution, namely the value-vector v^{α}, it follows from the theory of linear programming that the dual program (1.18) also has a finite optimal

solution.

(3) Take any $f_*^\infty \in C(D)$ with $x_i^*(f_*(i)) > 0$ for every $i \in S$ (such policy exists by part (1)).

Because $x_i^*(f_*(i)) > 0$, $i \in S$, the complementary slackness property of linear programming implies

$$\sum_j \{\delta_{ij} - \alpha p_{ij}(f_*)\} v_j^\alpha = r_i(f_*),\ i \in S.$$

Hence, in vector notation,

$$\{I - \alpha P(f_*)\} v^\alpha = r(f_*) \text{ which implies } v^\alpha = \{I - \alpha P(f_*)\}^{-1} r(f_*) = v^\alpha(f_*^\infty),$$

i.e. f_*^∞ is an α-discounted optimal policy.

If the simplex method is used, then the programs (1.17) and (1.18) are solved simultaneously. Hence by the simplex method both the value vector v^α and an optimal policy are computed.

Next, we show the one-to-one correspondence between the feasible solution of (1.18) and the set $C(S)$ of stationary policies. For $\pi^\infty \in C(S)$ the vector $x(\pi)$ with component $x_i^\pi(a), (i,a) \in S \times A$, is defined by

$$x_i^\pi(a) = \{\beta^T \{I - \alpha P(\pi)\}^{-1}\}_i \cdot \pi_{ia},\ (i,a) \in S \times A. \tag{1.19}$$

Define for any $t \in N$ and $(i,a) \in S \times A$ a random variable $n_{ia}^{(t)}$ by

$$n_{ia}^{(t)} = \begin{cases} 1 & \text{if } (X_t, Y_t) = (i,a); \\ 0 & \text{otherwise.} \end{cases}$$

Then, the total discounted number of times that $(X_t, Y_t) = (i,a)$ equals $\sum_{t=1}^\infty \alpha^{t-1} n_{ia}^{(t)}$.

Lemma 1.5

Given initial distribution β, i.e. $P\{X_1 = j\} = \beta_j$ for all $j \in S$, and a stationary policy π^∞, $x_i^\pi(a)$ satisfies $x_i^\pi(a) = E_{\beta,\pi}\{\sum_{t=1}^\infty \alpha^{t-1} n_{ia}^{(t)}\},\ (i,a) \in S \times A.$

Proof

Since $\{I - \alpha P(\pi)\}^{-1} = \sum_{t=1}^\infty \alpha^{t-1} P^{t-1}(\pi)$, we have

$$x_i^{\pi}(a) = \sum_j \beta_j \cdot \{\sum_{t=1}^{\infty} \alpha^{t-1} P^{t-1}(\pi)\}_{ji} \cdot \pi_{ia} = \sum_{t=1}^{\infty} \alpha^{t-1} \{\sum_j \beta_j P_{\pi}\{X_t = i \mid X_1 = j\}\} \cdot \pi_{ia}$$
$$= \sum_{t=1}^{\infty} \alpha^{t-1} \{\sum_j \beta_j P_{\pi}\{X_t = i, Y_t = a \mid X_1 = j\}\} = \sum_{t=1}^{\infty} \alpha^{t-1} \cdot E_{\beta,\pi}\{n_{ia}^{(t)}\}$$
$$= E_{\beta,\pi}\{\sum_{t=1}^{\infty} \alpha^{t-1} n_{ia}^{(t)}\}.$$

Conversely, for a feasible solution x of (1.18), define $\pi(x)$ with elements π_{ia}^{x} by

$$\pi_{ia}^{x} = \frac{x_i(a)}{\sum_a x_i(a)}, \ (i,a) \in S \times A. \tag{1.20}$$

Theorem 1.12

The mapping (1.19) is a one-to-one mapping of the set of stationary policies onto the set of feasible solution of the dual program (1.18) with (1.20) as the inverse mapping; furthermore, the set of extreme feasible solution of (1.18) corresponds to the set $C(D)$ of deterministic policies.

Proof

First, we show that x^{π} is a feasible solution of (1.18).

$$\sum_{(i,a)} \{\delta_{ij} - \alpha p_{ij}(a)\} x_i^{\pi}(a) = \sum_{(i,a)} \{\delta_{ij} - \alpha p_{ij}(a)\} \{\beta^T \{I - \alpha P(\pi)\}^{-1}\}_i \cdot \pi_{ia}$$
$$= \sum_i \{\beta^T \{I - \alpha P(\pi)\}^{-1}\}_i \cdot \sum_a \{\delta_{ij} - \alpha p_{ij}(a)\} \cdot \pi_{ia}$$
$$= \sum_i \{\beta^T \{I - \alpha P(\pi)\}^{-1}\}_i \cdot \{I - \alpha P(\pi)\}_{ij}$$
$$= \{\beta^T \{I - \alpha P(\pi)\}^{-1} \cdot \{I - \alpha P(\pi)\}\}_j = \beta_j, \ j \in S.$$

Since $\{I - \alpha P(\pi)\}^{-1} = \sum_{t=0}^{\infty} \{\alpha P(\pi)\}^t \geq 0, \ x_i^{\pi}(a) \geq 0$ for every $(i,a) \in S \times A$.

Next, we prove the one-to-one correspondence. Let x be a feasible solution of (1.18).

Then, (1.20) yields $x_i(a) = \pi_{ia}^{x} \cdot x_i$, where $x_i = \sum_a x_i(a), \ i \in S$. Therefore, we can write

$$\beta_j = \sum_{(i,a)} \{\delta_{ij} - \alpha p_{ij}(a)\} x_i(a) = \sum_{(i,a)} \{\delta_{ij} - \alpha p_{ij}(a)\} \cdot \pi_{ia}^{x} \cdot x_i$$
$$= \sum_i \{\delta_{ij} - \alpha p_{ij}(\pi(x))\} x_i, \ j \in S.$$

Hence, in vector notation,

$$\beta^T = x^T \{I - \alpha P(\pi(x))\}, \ \textit{i.e.} \ x^T = \beta^T \{I - \alpha P(\pi(x))\}^{-1} = \{x(\pi(x))\}^T.$$

Conversely,

$$\pi_{ia}^{x(\pi)} = \frac{x_i^{\pi}(a)}{\sum_a x_i^{\pi}(a)} = \pi_{ia}^{x^{\pi}}, \ (i,a) \in S \times A. \tag{1.21}$$

Therefore, we have shown the one-to-one correspondence and that (1.20) is the inverse of (1.19).

Finally, we show the correspondence between the extreme points of (1.18) and the set $C(D)$.

Let $f^{\infty} \in C(D)$. Then, for every $i \in S$,

$$x_i^f(a) = \begin{cases} \{\beta^T\{I - \alpha P(\pi)\}^{-1}\}_i, a = f(i); \\ 0 \qquad\qquad\qquad\quad , a \neq f(i). \end{cases}$$

Suppose x^f is not an extreme feasible solution. Then, there exist feasible solutions x^1 and x^2 of (1.18) and a real number $\lambda \in (0,1)$ such that $x^1 \neq x^2$ and $x^f = \lambda x^1 + (1-\lambda)x^2$.

Since $x_i^f(a) = 0, a \neq f(i),\ i \in S$, we have $x_i^1(a) = x_i^2(a) = 0,\ a \neq f(i),\ i \in S$.

Hence, the N-vectors $x^1 = x_i^1(f(i))$ and $x^2 = x_i^2(f(i))$ are solutions of the linear system $x^T\{I - \alpha P(f)\} = \beta^T$. However, this linear system has a unique solution $x^T = \beta^T\{I - \alpha P(f)\}^{-1}$. This implies $x^1 = x^2 = \beta^T\{I - \alpha P(f)\}^{-1}$, which contradicts $x^1 \neq x^2$. Hence, we have shown that x^f is an extreme solution.

Conversely, let x be an extreme feasible solution of program (1.18). Since (1.18) has N constraints, x has at most N positive components. On the other hand, Theorem 1.11, part (1), implies that in each state there is at least one positive component. Consequently, x has in each state i exactly one positive component, i.e. the sorresponding stationary policy is deterministic.

Algorithm 1.1 Linear programming algorithm

1. Take any vector β, where $\beta_j > 0,\ j \in S$.
2. Use a linear programming algorithm to compute optimal solutions v^* and x^* of the dual pair of linear programs:

   $$\min\{\sum_j \beta_j v_j \mid \sum_j \{\delta_{ij} - \alpha p_{ij}(a)\}v_j \geq r_i(a),\ (i,a) \in S \times A\}$$

   and

   $$\max\left\{\sum_{(i,a)} r_i(a)x_i(a) \;\middle|\; \begin{array}{r} \sum_{(i,a)} \{\delta_{ij} - \alpha p_{ij}(a)\}x_i(a) = \beta_j,\ j \in S \\ x_i(a) \geq 0, (i,a) \in S \times A \end{array}\right\}.$$

3. Take $f_*^\infty \in C(D)$ such that $x_i^*(f_*(i)) > 0$ for every $i \in S$.

v^* is the value-vector v^a and f_*^∞ is an α-discounted optimal policy **(STOP).**

Next, we discuss the elimination of suboptimal actions with test (1.14).

Let $y_i^f(a)$ be the dual slack variable. i.e.

$$y_i^f(a) = \sum_j \{\delta_{ij} - \alpha p_{ij}(a)\}v_j^\alpha(f) - r_i(a).$$

Since

$$(Ux - x)_i = \max_a \{r_i\ (a) + \alpha \sum_j p_{ij}(a) v_j^\alpha\} - v_j^\alpha = -\min_a y_i^f(a),\ i \in S$$

and

$$span(Ux - x) = \min_i \min_a y_i^f(a) - \max_i \min_a y_i^f(a),$$

then the test (1.14) becomes

$$y_i^f(a_i) > \min_a y_i^f(a) - \alpha(1-\alpha)^{-1}\{\min_i \min_a y_i^f(a) - \max_i \min_a y_i^f(a)\},$$

which results in the following theorem.

Theorem 1.13

If $y_i^f(a_i) > \min_a y_i^f(a) - \alpha(1-\alpha)^{-1}\{\min_i \min_a y_i^f(a) - \max_i \min_a y_i^f(a)\}$, then action $a_i \in A(i)$ is suboptimal.

1.4 Average Rewards

1.4.1 Introduction

When decisions are made frequently, so that the discount rate is very close to 1, or when performance criterion cannot easily be described in economic terms with discount factors, the decision maker may prefer to compare policies on the basis of their average expected rewards instead of their expected total discounted rewards. Consequently, the average rewards criterion occupies a cornerstone of queueing control theory especially when applied to control computer systems and communication networks. In such systems, the controller makes frequent decisions and usually assesses system performance on the basis of throughput rate or the average time a job remains in the system. This optimality criterion may also be appropriate for inventory systems with frequent restocking decisions.

In this section we start with theorems about the stationary matrix, the fundamental matrix and the deviation matrix of a Markov chain, without proof. For the proof we refer to Kallenberg [9]. These matrices play an important role in the average reward criterion and also in more sensitive criteria. The most sensitive criterion is Blackwell optimality. The existence of a deterministic Blackwell optimal policy is shown in a separate section. Laurent series expansion relates the average reward to the total discounted reward. This is the subject of section 1.4.4. The optimality equation for average rewards is the subject of section 1.4.5 and section 1.4.6 deals with linear programming.

1.4.2 The stationary, fundamental and deviation matrices

The stationary matrix

Consider a policy $f^{\infty} \in C(D)$. In average reward MDPs, the limiting behavior of $\{P(f)\}^n$ as n tends to infinity plays an important role. In general, $\lim_{n\to\infty}\{P(f)\}^n$ does not exist. Therefore, we consider other types of convergence.

Let $\{b_n\}_{n=0}^{\infty}$ be a sequence. This sequence is called Cesaro convergent with Cesaro limit b if

$\lim_{n\to\infty}\frac{1}{n}\sum_{k=0}^{n-1} b_k$ exists and is equal to b.

We denote this convergence by $\lim_{n\to\infty} b_n =_c b$ or $b_n \to_c b$. The sequence is said to be Abel convergent with Abel limit b if

$\lim_{\alpha\uparrow 1}(1-\alpha)\sum_{n=0}^{\infty}\alpha^n b_n$ exists and is equal to b.

This convergence is denoted by $\lim_{n\to\infty} b_n =_a b$ or $b_n \to_a b$. Ordinary convergence implies both Cesaro and Abel convergence, but the converse statement is not true. The next result is well known in the theory of the summability of series (e.g. Powell and Shah [11], p.9).

Theorem 1.14

If the sequence $\{b_n\}_{n=0}^{\infty}$ is Cesaro convergent to b, then $\{b_n\}_{n=0}^{\infty}$ is also Abel convergent to b.

Remark

The converse statement of Theorem 1.14 is not true.

Theorem 1.15

Let P be any stochastic matrix, i.e. the matrix of a Markov chain. Then,

(1) $P^* := \lim_{n\to\infty}\frac{1}{n}\sum_{k=0}^{n-1} P^k$ exists, i.e. $P^n \to_c P^*$.

(2) $P^*P = PP^* = P^*P^* = P^*$.

The matrix P^* is called the stationary matrix of the stochastic matrix P.

Corollary 1.6

$\lim_{\alpha\uparrow 1}\sum_{n=0}^{\infty}\alpha^n(P^n - P^*) = 0$.

Let P^n be any stochastic matrix with ergodic classes $E_1, E_2, ..., E_m$ and transient states T. By renumbering of the states the matrix can be written in the following so-called standard form:

$$P=\begin{bmatrix} P_1 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & P_2 & 0 & \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & & & \cdot & 0 \\ \cdot & & \cdot & \cdot & \cdot & & \cdot & 0 \\ \cdot & & & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & \cdot & \cdot & \cdot & 0 & P_m & 0 \\ A_1 & A_2 & \cdot & \cdot & \cdot & \cdot & A_m & Q \end{bmatrix}, \tag{1.22}$$

where the matrix P_k corresponds to the ergodic class $E_k, 1 \le k \le m$, and the matrix Q to the transient states. It is well known (e.g. Doob[4] p. 180), that $Q^n \to 0$ for $n \to \infty$. Since

$$(I-Q)(I+Q+\cdots Q^{n-1}) = I - Q^n, \tag{1.23}$$

the right hand side of (1.23) tends to I, i.e. $I-Q$ is nonsingular and $(I-Q)^{-1} = \sum_{n=0}^{\infty} Q^n$. From the theory of Markov chain it is also well know (e.g. Chung[3] p.33) that the stationary matrix of an ergodic class has strictly positive, identical rows, say π^k for P_k, and that π^k is the unique solution of the following system of linear equations

$$\begin{cases} \sum_{i \in E_k} (\delta_{ij} - p_{ij}) x_i = 0, \ \ j \in E_k; \\ \sum_{i \in E_k} x_i = 1. \end{cases} . \tag{1.24}$$

Since (1.24) is a system of $|E_k|+1$ equations and $|E_k|$ variables, the first equation can be deleted for the computation of π^k.

The following results are also well known (e.g. Feller[5]).

Lemma 1.7

Let a_i^k be the probability that, starting from state $i \in T$, the Markov chain will be absorbed in ergodic class E_k, $1 \le k \le m$. Then a_i^k, $i \in T$, is the unique solution of the linear system $(I-Q)x = b^k$, where $b^k = A_k e$.

Theorem 1.16

Let P be any stochastic matrix written in the standard form (1.22). Then,

$$P^* = \begin{bmatrix} P_1^* & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & P_2^* & 0 & \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & & & \cdot & 0 \\ \cdot & & \cdot & \cdot & \cdot & & \cdot & 0 \\ \cdot & & & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & \cdot & \cdot & \cdot & 0 & P_m^* & 0 \\ A_1^* & A_2^* & \cdot & \cdot & \cdot & \cdot & A_m^* & 0 \end{bmatrix}, \qquad (1.25)$$

where P_k^* has identical rows π^k, which are the unique solution of (1.24) and $A_k^* = \{I - Q\}^{-1}\{A_k e\}\{\pi^k\}^T$, $1 \le k \le m$.

Algorithm 1.2 Determination of the stationary matrix P^*

1. Determine the ergodic classes $E_1, E_2, ..., E_m$ and the transient states T and write P in standard form (1.22).
2. Determine for $k = 1,2,...,m$:

a. the unique solution π_j^k, $j \in E_k$, of the linear system $\begin{cases} \sum_{i \in E_k} (\delta_{ij} - p_{ij}) x_i = 0, & j = 2,3,... \\ \sum_{i \in E_k} x_i = 1 \end{cases}$.

b. the unique solution a_i^k, $i \in T$ of the linear system $\sum_{j \in T} (\delta_{ij} - p_{ij}) x_j = \sum_{l \in E_k} p_{il}$, $i \in T$.

3. $p_{ij}^* = \begin{cases} x_j^k & \text{i} \in \text{E}_\text{k}, j \in E_k, k = 1,2,...,m \\ a_i^k x_j^k & \text{i} \in \text{T}, \text{j} \in \text{E}_\text{k}, k = 1,2,...,m \\ 0 & \text{else} \end{cases}$.

The fundamental matrix and the deviation matrix

Theorem 1.17

Let P be any stochastic matrix. Then $I - P + P^*$ is nonsingular and $Z := (I - P + P^*)^{-1}$ satisfies $Z = \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} \sum_{k=0}^{i-1} (P - P^*)^k$.

The matrix $Z := (I - P + P^*)^{-1}$ is called the fundamental matrix of P.

The deviation matrix D is defined by $D := Z - P^* = \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} \sum_{k=0}^{i-1} (P - P^*)^k - P^*$.

Theorem 1.18

The deviation matrix D satisfies

(1) $D = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \sum_{k=0}^{i-1} (P^k - P^*)$.

(2) $P^* D = D P^* = (I - P) D + P^* - I = D(I - P) + P^* - I = 0$.

The fundamental and the deviation matrix can be computed as follows. From (1.22) and (1.25) if follows that

$$I - P + P^* = \begin{bmatrix} C_1 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & C_2 & 0 & \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & & & \cdot & 0 \\ \cdot & & \cdot & \cdot & \cdot & & \cdot & 0 \\ \cdot & & & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & \cdot & \cdot & \cdot & 0 & C_m & 0 \\ D_1 & D_2 & \cdot & \cdot & \cdot & \cdot & D_m & I - Q \end{bmatrix},$$

where $C_k = I - P_k + P_k^*$ and $D_k = -A_k + A_k^*, 1 \le k \le m$. Hence,

$$Z = (I - P + P^*)^{-1} = \begin{bmatrix} C_1^{-1} & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & C_2^{-1} & 0 & \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & & & \cdot & 0 \\ \cdot & & \cdot & \cdot & \cdot & & \cdot & 0 \\ \cdot & & & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & \cdot & \cdot & \cdot & 0 & C_m^{-1} & 0 \\ S_1 & S_2 & \cdot & \cdot & \cdot & \cdot & S_m & (I - Q)^{-1} \end{bmatrix},$$

Where $S_k = -(I - Q)^{-1} D_k C_k^{-1}$, $1 \le k \le m$. Then, the deviation matrix is simply $Z - P^*$.

Theorem 1.19

(1) $Z = \lim_{\alpha \uparrow 1} \sum_{n=0}^{\infty} \alpha^n (P - P^*)^n$.

(2) $D = \lim_{\alpha \uparrow 1} \sum_{n=0}^{\infty} \alpha^n (P^n - P^*)$.

The following theorem gives the relation between average rewards, discounted rewards over an infinite horizon and total rewards over a finite horizon.

Theorem 1.20

Let f^∞ be a deterministic policy. Then,

(1) $\phi(f^\infty) = P^*(f)r(f)$.

(2) $\phi(f^\infty) = \lim_{\alpha \uparrow 1}(1-\alpha)v^\alpha(f^\infty)$.

(3) $v^T(f^\infty) = T\phi(f^\infty) + D(f)r(f) - P^T(f)D(f)r(f)$.

The regular case

A Markov chain P is called a regular Markov chain if the chain is irreducible and aperodic. In that case it can be shown that $P^* = \lim_{n\to\infty} P^n$. Since $(P-P^*)^n = P^n - P^*$ for $n = 1,2,...$ we have $(P-P^*)^n \to 0$ if $n \to \infty$. Therefore,

$$Z := (I - P + P^*)^{-1} = \sum_{n=0}^{\infty}(P-P^*)^n .$$

Because $D = Z - P^*$ and $Z = I + \sum_{n=1}^{\infty}(P-P^*)^n = I + \sum_{n=1}^{\infty}(P^n - P^*)$, we obtain

$$D = \sum_{n=0}^{\infty}(P^n - P^*),$$

i.e. D represents the total deviation with respect to the stationary matrix. This explains the name deviation matrix.

1.4.3 Blackwell optimality

In this section we prove the existence of a deterministic policy f_0^∞ such that $v^\alpha(f_0^\infty) = v^\alpha$ for all $\alpha \in [\alpha_0, 1)$ for some $0 \le \alpha_0 < 1$. Such a policy is called a Blackwell optimal policy. The next theorem shows that the interval $[0,1)$ can be partitioned in a finite number of subintervals such that in each subinterval there exists a deterministic policy which is optimal over the whole subinterval.

Theorem 1.21

There are numbers $\alpha_m, \alpha_{m-1}, ..., \alpha_0, \alpha_{-1}$ and deterministic policies $f_m^\infty, f_{m-1}^\infty, ..., f_0^\infty$ such that

(1) $0 = \alpha_m < \alpha_{m-1} < ... < \alpha_0 < \alpha_{-1} = 1$;

(2) $v^\alpha(f_j^\infty) = v^\alpha$ for all $\alpha \in [\alpha_j, \alpha_{j-1}),\ j = m, m-1, ..., 0$

Proof

For any deterministic policy f^∞, $v^\alpha(f^\infty)$ is the unique solution of the linear system

$$\{I - \alpha P(f)\}x = r(f).$$

By Cramer's rule* $v_i^\alpha(f^\infty)$ is a rational function in α for each component i.

Suppose that a deterministic Blackwell optimal policy does not exist. For any fixed α a deterministic α-discounted optimal policy exists. This implies a series $\{\alpha_k,\ k = 1,2,...\}$ and a series $\{f_k,\ k = 1,2,...\}$ such that

$\alpha_1 \le \alpha_2 \le ...$ with $\lim_{k\to\infty} \alpha_k = 1$ and $v^\alpha = v^\alpha(f_k^\infty) > v^\alpha(f_{k-1}^\infty)$ for $\alpha = \alpha_k,\ k = 2,3,...$

Since there are only a finite number of deterministic policies, there must be a couple of policies, say f^∞ and g^∞, such that for some nondecreasing subsequence α_{k_n}, $n = 1,2,...$ with $\lim_{n\to\infty} \alpha_{k_n} = 1$

$$\begin{cases} v^\alpha(f^\infty) > v^\alpha(g^\infty) \text{ for } \alpha = \alpha_{k_1}, \alpha_{k_3}, ... \\ v^\alpha(f^\infty) < v^\alpha(g^\infty) \text{ for } \alpha = \alpha_{k_2}, \alpha_{k_4}, ... \end{cases} \tag{1.26}$$

Let $h(\alpha) = v^\alpha(f^\infty) - v^\alpha(g^\infty)$, then $h_i(\alpha)$ is a continuous rational function in α on $[0,1)$ for each $i \in S$. From (1.26) it follows that $h_i(\alpha)$ has an infinite number of zeros, which is in contradiction with the rationality of $h_i(\alpha)$. Hence, there exists a deterministic Blackwell optimal policy, i.e. a policy f_0^∞ such that $v^\alpha(f_0^\infty) = v^\alpha$ for all $\alpha \in [\alpha_0, 1)$ for some $0 \le \alpha_0 < 1$.

With similar arguments it can be shown that for each fixed $\alpha \in [0,1)$ there is a lower bound $L(\alpha) < \alpha$ and a deterministic policy $f_{L(\alpha)}^\infty$ such that $v^\alpha(f_{L(\alpha)}^\infty) = v^\alpha$ for all $\alpha \in (L(\alpha), \alpha)$. Similarly, for each fixed $\alpha \in [0,1)$ there is an upper bound $U(\alpha) > \alpha$ and a deterministic policy $f_{U(\alpha)}^\infty$ such that $v^\alpha(f_{U(\alpha)}^\infty) = v^\alpha$ for all $\alpha \in (\alpha, U(\alpha))$.

The open intervals $(-1, U(0))$, $\{(L(\alpha), U(\alpha)) \mid \alpha \in (0,1)\}$ and $(L(1), 2)$ are a covering of the compact set $[0,1]$. By the Heine-Borel-Lebesque covering theorem, the interval $[0,1]$ is covered by a finite number of intervals, say $(-1, U(0)), \{(L(\alpha_j), U(\alpha_j)),\ j = m-1, m-2, ..., 1\}$ and

* see e.g. J.B. Fraleigh and R.A. Beauregard: Linear Algebra, Addison Wesley, 1987, p. 214.

$(L(1),2)$. We may assume that

$$\alpha_m := 0 < \alpha_{m-1} < ... < \alpha_0 < \alpha_{-1} = 1,\ \mathrm{L}(\alpha_{\mathrm{m\text{-}1}}) < U(0), L(1) < U(\alpha_1)$$

and

$$L(\alpha_j) < L(\alpha_{j-1}) < U(\alpha_j) < U(\alpha_{j-1}),\ \ j = m-1, m-2, ..., 2\,.$$

Since the rational function $v^\alpha(f^\infty_{L(\alpha_{j-1})}) = v^\alpha(f^\infty_{U(\alpha_j)}) = v^\alpha$ for all $\alpha \in (L(\alpha_{j-1}), U(\alpha_j))$ we have

$$v^\alpha(f^\infty_{L(\alpha_{j-1})}) = v^\alpha(f^\infty_{U(\alpha_j)}),\ \ j = 0,1,..., m\,.$$

Let $f_j = f_{U(\alpha_j)},\ \ j = 0,1,..., m$. Then,

$$v^\alpha(f_j^\infty) = v^\alpha \quad \text{for all} \quad \alpha \in (\alpha_j, \alpha_{j-1}),\ \ j = 0,1,..., m\,.$$

Since $v^\alpha(f^\infty)$ is continuous in α, also

$$v^\alpha(f_j^\infty) = v^\alpha \quad \text{for} \quad \alpha = \alpha_j,\ \ j = 0,1,..., m\,.$$

1.4.4 The Laurent series expansion

Theorem 1.20 part (2) shows a relation between discounted and average rewards when the discount factor tends to 1. This relation is based on the Laurent expansion of $v^\alpha(f^\infty)$ close to $\alpha = 1$ as expressed in the next theorem.

Theorem 1.22

Let $u^k(f),\ k = -1,0,...$ be defined by $u^{-1}(f) = P^*(f)r(f)$, $u^0(f) = D(f)r(f)$ and $u^{k+1}(f) = -D(f)u^k(f),\ k \geq 0$. Then, $\alpha v^\alpha(f^\infty) = \sum_{k=-1}^{\infty} \rho^k u^k(f)$ for $\alpha_0(f) < \alpha < 1$, where $\rho = \dfrac{1-\alpha}{\alpha}$ and $\alpha_0(f) = \dfrac{\| D(f) \|}{1 + \| D(f) \|}$.

Proof

Let $x(f) = \dfrac{1}{\alpha}\sum_{k=-1}^{\infty} \rho^k u^k(f) = \dfrac{\phi(f^\infty)}{1-\alpha} + \dfrac{1}{\alpha}\sum_{k=0}^{\infty} \rho^k u^k(f)$.

Since $u^k(f) = D(f)\{-D(f)\}^k r(f)$ for $k \geq 0$, the series $\sum_{k=0}^{\infty} \rho^k u^k(f)$ is well defined if

$\| \rho D(f) \| < 1$, i.e. $\alpha \geq \dfrac{\| D(f) \|}{1 + \| D(f) \|}$.

Since $v^{\alpha}(f^{\infty})$ is the unique solution of the linear system $\{I - \alpha P(f)\}x = r(f)$, it is sufficient to show that $\{I - \alpha P(f)\}x(f) = r(f)$, i.e. $y(f) := r(f) - \{I - \alpha P(f)\}x(f) = 0$.

$$y(f) = r(f) - \{I - \alpha P(f)\}\frac{P^{*}(f)r(f)}{1-\alpha} - \{I - \alpha P(f)\}\frac{D(f)}{\alpha}\sum_{k=0}^{\infty}\{-\rho D(f)\}^{k} r(f)$$

$$= r(f) - P^{*}(f)r(f) - \{\alpha(I - P(f)) + (1-\alpha)I\}\frac{D(f)}{\alpha}\sum_{k=0}^{\infty}\{-\rho D(f)\}^{k} r(f)$$

$$= \{I - P^{*}(f)\}r(f) - \{I - P(f)\}D(f)\sum_{k=0}^{\infty}\{-\rho D(f)\}^{k} r(f)$$

$$- \frac{1-\alpha}{\alpha} D(f)\sum_{k=0}^{\infty}\{-\rho D(f)\}^{k} r(f)$$

$$= \{I - P^{*}(f)\}r(f) - \{I - P^{*}(f)\}\sum_{k=0}^{\infty}\{-\rho D(f)\}^{k} r(f) + \sum_{k=0}^{\infty}\{-\rho D(f)\}^{k+1} r(f)$$

$$= \{I - P^{*}(f)\}r(f) - \sum_{k=0}^{\infty}\{-\rho D(f)\}^{k} r(f) + P^{*}(f)r(f) + \sum_{k=1}^{\infty}\{-\rho D(f)\}^{k} r(f)$$

$$= \{I - P^{*}(f)\}r(f) - r(f) - \sum_{k=1}^{\infty}\{-\rho D(f)\}^{k} r(f) + P^{*}(f)r(f) + \sum_{k=1}^{\infty}\{-\rho D(f)\}^{k} r(f)$$

$$= 0.$$

Corollary 1.7

$v^{\alpha}(f^{\infty}) = \dfrac{\phi(f^{\infty})}{1-\alpha} + u^{0}(f) + \varepsilon(\alpha)$, where $\varepsilon(\alpha)$ satisfies $\lim_{\alpha \uparrow 1} \varepsilon(\alpha) = 0$.

Proof

From Theorem 1.22 if follows that $v^{\alpha}(f^{\infty}) = \dfrac{\phi(f)}{1-\alpha} + \dfrac{u^{0}(f)}{\alpha} + \sum_{k=1}^{\infty}\dfrac{(1-\alpha)^{k}}{\alpha^{k+1}} u^{k}(f)$.

Since $\dfrac{1}{\alpha} = \dfrac{1}{1-(1-\alpha)} = 1 + (1-\alpha) + (1-\alpha)^{2} + \cdots$, we may write

$$v^{\alpha}(f^{\infty}) = \frac{\phi(f^{\infty})}{1-\alpha} + u^{0}(f) + \varepsilon(\alpha),$$

where $\lim_{\alpha \uparrow 1} \varepsilon(\alpha) = 0$.

1.4.5 The optimality equation

In the discounted case, the value vector is the unique solution of an optimality equation. For the average reward criterion a similar result holds, but the equation is more complicated.

Theorem 1.23
Consider the system

$$\begin{cases} x_i = \max_{a\in A(i)} \sum_j p_{ij}(a)x_j, i \in S \\ x_i + y_i = \max_{a\in A(i,x)}\{r_i(a) + \sum_j p_{ij}(a)y_j\}, i \in S \end{cases} \tag{1.27}$$

where $A(i,x) = \{a \in A(i) \mid x_i = \sum_j p_{ij}(a)x_j\},\ i \in S$.

This system has the following properties:

(1) $x = u^{-1}(f_0),\ y = u^0(f_0)$, where f_0^∞ is a Blackwell optimal policy, satisfies (1.27).

(2) If (x, y) is a solution of (1.27), then $x = \phi$, the value vector.

Proof

Since f_0^∞ is a Blackwell optimal policy, for α sufficiently close to 1, say $\alpha \in [\alpha_0, 1)$, one can write

$$v_i^\alpha(f_0^\infty) = v_i^\alpha = \max_{a\in A(i)}\{r_i(a) + \alpha\sum_j p_{ij}(a)v_j^\alpha\} \geq r_i(a) + \alpha\sum_j p_{ij}(a)v_j^\alpha,\ (i,a) \in S \times A.$$

Combining this result with Corollary 1.7 gives for all $\alpha \in [\alpha_0, 1)$:

$$\frac{\phi_i(f_0^\infty)}{1-\alpha} + u_i^0(f_0) + \varepsilon_i(\alpha) \geq r_i\ (a) + \{1-(1-\alpha)\}\sum_j p_{ij}(a)\{\frac{\phi_j(f_0^\infty)}{1-\alpha} + u_j^0(f_0) + \varepsilon_j(\alpha)\},\ (i,a) \in S \times A$$

$$= r_i\ (a) + \sum_j p_{ij}(a)\{\frac{\phi_j(f_0^\infty)}{1-\alpha} + u_j^0(f_0) + \varepsilon_j(\alpha)\} +$$

$$(1-\alpha)\sum_j p_{ij}(a)\{\frac{\phi_j(f_0^\infty)}{1-\alpha} + u_j^0(f_0) + \varepsilon_j(\alpha)\},\ (i,a) \in S \times A,$$

i.e.

$$\frac{1}{1-\alpha}\{\phi_i(f_0^\infty) - \sum_j p_{ij}(a)\phi_j(f_0^\infty)\} + \{u_i^0(f_0) - r_i\ (a) - \sum_j p_{ij}(a)u_j^0(f_0) - \sum_j p_{ij}(a)\phi_j(f_0^\infty)\} + \varepsilon(\alpha) \geq 0.$$

Since this result holds for all $\alpha \in [\alpha_0, 1)$, the term multiplied by $\frac{1}{1-\alpha}$ has to be nonnegative, i.e.

$$\phi_i(f_0^\infty) \geq \sum_j p_{ij}(a)\phi_j(f_0^\infty) \text{ for all } i \in S \text{ and } a \in A(i). \tag{1.28}$$

Furthermore, when $\phi_i(f_0^\infty) = \sum_j p_{ij}(a)\phi_j(f_0^\infty)$, the next term has to be nonnegative, i.e.

$$u_i^0(f_0) \geq r_i\ (a) - \sum_j p_{ij}(a)u_j^0(f_0) - \sum_j p_{ij}(a)\phi_j(f_0^\infty) = r_i\ (a) - \sum_j p_{ij}(a)u_j^0(f_0) - \phi_i(f_0^\infty). \tag{1.29}$$

For $a = f_0(i),\ i \in S$, the inequalities in (1.28) and (1.29) are equalities, because:

$$\phi_i(f_0^\infty) = P^*(f_0)r(f_0) = P(f_0)P^*(f_0)r(f_0) = P(f_0)\phi(f_0^\infty)$$

and

$$u^0(f_0) = D(f_0)r(f_0) = \{I - P^*(f_0) + P(f_0)D(f_0)\}r(f_0) = r(f_0) - \phi(f_0^\infty) + P(f_0)u^0(f_0).$$

By these results, part (1) is shown. For part (2), let (x, y) be a solution of (1.27). Then, for any $f^\infty \in C(D), x \geq P(f)x$, implying that $x \geq P^n(f)x$ for all $n \in N$, and consequently, $x \geq P^*(f)x$.

Furthermore, since $0 = P^*(f)\{x - P(f)\}$ and all elements of $P^*(f)$ and $x - P(f)$ and nonnegative, $p_{ij}^*(f)\{x - P(f)\}_j = 0$ for all $i, j \in S$, implying that $p_{ii}^*(f)\{x - P(f)\}_i = 0$ for all $i \in S$.

For an ergodic state $i, p_{ii}^*(f) > 0$, and consequently $x_i - \sum_j p_{ij}(a)x_j = 0$, i.e. $f(i) \in A(i, x)$, and therefore, by (1.27) $x_i + y_i \geq r_i(f) + \sum_j p_{ij}(f)y_j$.

The columns of $P^*(f)$ corresponding to the transient states are zero, implying that

$$P^*(f)(x + y) \geq P^*(f)\{r(f) + P(f)y\} = \phi(f^\infty) + P^*(f)y,$$

i.e.

$$\phi(f^\infty) \leq P^*(f)x \leq x. \qquad (1.30)$$

On the other hand, any solution of system (1.27) gives a policy g^∞ which satisfies $x = P(g)x$ and $x + y = r(g) + P(g)y$. Hence, $x = P^*(g)x$ and therefore,

$$\phi(g^\infty) = P^*(g)f(g) = P^*(g)\{x + y - P(g)y\} = x + P^*(g)\{y - P(g)y\} = x. \quad (1.31)$$

From (1.30) and (1.31) if follows that $x_i = \max_{a \in A(i)} \sum_j p_{ij}(a)x_j = \phi_i,\ i \in S$.

Remarks

1. Since the x-vector in (1.27) is unique, namely $x = \phi$, the set $A(i, x)$ is also unique for all $i \in S$.
2. If policy f^∞ satisfies $\phi = P(f)\phi$ and $\phi + y = r(f) + P(f)y$ for some vector y, then

the policy is average optimal, namely

$$\phi = P^*(f)\phi = P^*(f)\{r(f) + P(f)y - y\} = \phi(f^\infty).$$

3. The proof suggests that a Blackwell optimal policy f_0^∞ is also average optimal, i.e. $\phi(f_0^\infty) \geq \phi(R)$ for every policy R. This result is shown below (Corollary 1.8).

4. If ϕ has identical components (e.g. if there is a unichain average optimal policy), then the first equation of (1.27) is superfluous and (1.27) can be replaced by the single optimality equation

$$x + y_i = \max_{a \in A(i)}\{r_i(a) + \sum_j p_{ij}(a)y_j\},\ i \in S. \qquad (1.32)$$

Theorem 1.24

$\lim_{\alpha \uparrow 1}(1-\alpha)v^\alpha(R) \geq \phi(R)$ for all policies R.

Proof

For $f^\infty \in C(D)$ we have shown in Theorem 1.20 part (2) that

$$\phi(f^\infty) = \lim_{\alpha \uparrow 1}(1-\alpha)v^\alpha(f^\infty).$$

For an arbitrary policy R the deviation is as follows.

Let $i \in S$ be any starting state and let $x_t = \sum_{(j,a)} P_{i,R}\{X_t = j, Y_t = a\} \cdot r_j(a),\ t = 1,2,...$

Since the sequence $\{x_t \mid t = 1,2,..\}.$ is bounded, we may write

$$(1-\alpha)^{-1}v_i^\alpha(R) = \{\sum_{t=1}^\infty \alpha^{t-1}\} \cdot \{\sum_{t=1}^\infty \alpha^{t-1}x_t\} = \sum_{t=1}^\infty \{\sum_{s=1}^t x_s\} \cdot \alpha^{t-1},$$

$(1-\alpha)^{-2} = \sum_{t=1}^\infty t\alpha^{t-1}$ for $\alpha \in (0,1)$, and therefore, $\phi_i(R) = \{\sum_{t=1}^\infty t\alpha^{t-1}\} \cdot (1-\alpha)^2 \cdot \phi_i(R)$

Hence, $(1-\alpha)v_i^\alpha(R) - \phi_i(R) = (1-\alpha)^2 \cdot \sum_{t=1}^\infty \{\frac{1}{t}\sum_{s=1}^t x_s - \phi_i(R)\} \cdot t\alpha^{t-1}$.

Choose any arbitrary $\varepsilon > 0$. Since $\phi_i(R) = \liminf_{T \to \infty} \frac{1}{T}\sum_{t=1}^T x_t$, there exists a T_ε such that $\phi_i(R) < \frac{1}{T}\sum_{t=1}^T x_t + \varepsilon$ for all $T > T_\varepsilon$. This gives

$$(1-\alpha)^2 \sum_{t > T_\varepsilon} \{\frac{1}{t}\sum_{s=1}^t x_s - \phi_i(R)\} t\alpha^{t-1} > -\varepsilon(1-\alpha)^2 \sum_{t > T_\varepsilon} t\alpha^{t-1} \geq -\varepsilon(1-\alpha)^2 \sum_{t=1}^\infty t\alpha^{t-1} = -\varepsilon.$$

We also have

$$(1-\alpha)^2 \sum_{t \leq T_\varepsilon} \{\frac{1}{t}\sum_{s=1}^t x_s - \phi_i(R)\} t\alpha^{t-1} \geq (1-\alpha)^2 \min_{1 \leq t \leq T_\varepsilon} \{\frac{1}{t}\sum_{s=1}^t x_x - \phi_i(R)\} \sum_{t \leq T_\varepsilon} t\alpha^{t-1} > -\varepsilon$$

for α sufficiently close to 1. Hence, $(1-\alpha)v_i^\alpha(R) - \phi_i(R) \geq -2\varepsilon$ for α sufficiently close to 1, i.e. $\lim_{\alpha \uparrow 1}(1-\alpha)v^\alpha(R) \geq \phi(R)$.

Corollary 1.8

A Blackwell optimal policy f_0^∞ is also average optimal and consequently there exists a deterministic optimal policy.

Proof

Let f_0^∞ be a Blackwell optimal policy and R an arbitrary policy. Then,

$$\phi(f_0^\infty) = \lim_{\alpha\uparrow 1}(1-\alpha)v^\alpha(f_0^\infty) = \lim_{\alpha\uparrow 1}(1-\alpha)v^\alpha \geq \lim_{\alpha\uparrow 1}(1-\alpha)v^\alpha(R) \geq \phi(R).$$

1.4.6 Linear programming

To apply linear programming in order to obtain the value vector and an average optimal policy we need a property for which the value vector is an extreme element. Such property, called superharmonicity, can be derived from the optimality equation. A vector $v \in R^N$ is average-superharmonic if there exists a vector $u \in R^N$ such that the pair (u,v) satisfies the following sustem of inequalities

$$\begin{cases} v_i \geq \sum_j p_{ij}(a)v_j & \text{for every } (i,a) \in S \times A \\ v_i + u_i \geq r_i(a) + \sum_j p_{ij}(a)u_j & \text{for every } (i,a) \in S \times A \end{cases}. \tag{1.33}$$

Theorem 1.25

The value vector ϕ is the smallest average-superharmonic vector.

Proof

Let f_0^∞ be a Blackwell optimal policy. From Theorem 1.23 it follows that

$$\begin{cases} \phi_i \geq \sum_j p_{ij}(a)\phi_j & \text{for every } i \in S, a \in A(i) \\ \phi_i + u_i^0(f_0) \geq r_i(a) + \sum_j p_{ij}(a)u_j^0(f_0) & \text{for every } i \in S, a \in A(i,\phi) \end{cases}. \tag{1.34}$$

where $A(i,\phi) = (a \in A(i) \mid \phi_i = \sum_j p_{ij}(a)\phi_j\},\ i \in S$.

Let $A^*(i) = (a \in A(i) \mid \phi_i + u_i^0(f_0) < r_i(a) + \sum_j p_{ij}(a)u_j^0(f_0)\},\ i \in S$.

Define

$$s_i(a) = \phi_i - \sum_j p_{ij}(a)\phi_j,\ \ t_i(a) = \phi_i + u_i^0(f_0) - r_i(a) - \sum_j p_{ij}(a)u_j^0(f_0),\ (i,a) \in S \times A,$$

$$M = \begin{cases} \min\{\dfrac{s_i(a)}{t_i(a)} \mid a \in A^*(i), i \in S\} & \text{if } \bigcup_{i\in S} A^*(i) \neq \phi \\ 0 & \text{if } \bigcup_{i\in S} A^*(i) = \phi \end{cases} \quad \text{and} \quad u = u^0(f_0) - M \cdot \phi .$$

For $a \in A(i,\phi)$, we have

$$\phi_i = \sum_j p_{ij}(a)\phi_j$$

and

$$\phi_i + u_i = \phi_i + u_i^0(f_0) - M \cdot \phi_i \geq r_i(a) + \sum_j p_{ij}(a)\{u_j^0(f_0) - M \cdot \phi_j\} = r_i(a) + \sum_j p_{ij}(a)u_j .$$

For $a \in A^*(i)$, we have

$$\phi_i > \sum_j p_{ij}(a)\phi_j$$

and

$$\begin{aligned} \phi_i + u_i &= \phi_i + u_i^0(f_0) - M \cdot \{s_i(a) + \sum_j p_{ij}(a)\phi_j\} \\ &= \mathrm{t}_i(a) + r_i(a) + \sum_j p_{ij}(a)u_j^0(f_0) - M \cdot s_i(a) \geq r_i(a) + \sum_j p_{ij}(a)u_j . \end{aligned}$$

For $a \notin \{a \in A(i,\phi) \cup A^*(i)\}$, we have

$$\phi_i > \sum_j p_{ij}(a)\phi_j$$

and

$$\begin{aligned} \phi_i + u_i &= \phi_i + u_i^0(f_0) - M \cdot \phi_i \geq \mathrm{t}_i(a) + r_i(a) + \sum_j p_{ij}(a)\{u_j^0(f_0) - M \cdot \phi_j\} \\ &= t_i(a) + r_i(a) + \sum_j p_{ij}(a)u_j \geq r_i(a) + \sum_j p_{ij}(a)u_j . \end{aligned}$$

Hence, the value vector ϕ is average-superharmonic.

Suppose that y is also average-superharmonic with corresponding vector x. Then, $y \geq P(f_0)y$, implying that

$$y \geq P^*(f_0)y \geq P^*(f_0)\{r(f_0) + (P(f_0) - I)x\} = P^*(f_0)r(f_0) = \phi(f_0^\infty) = \phi ,$$

i.e. ϕ is the smallest average-superharmonic vector.

Corollary 1.9

From the proof of Theorem 1.25 it follows that there exists a solution of the modified optimality equation

$$\begin{cases} x_i = \max_{a\in A(i)} \sum_j p_{ij}(a)x_j, \ i \in S \\ x_i + y_i = \max_{a\in A(i)} \{r_i(a) + \sum_j p_{ij}(a)y_j\}, \ i \in S \end{cases} \tag{1.35}$$

with $x = \phi$ as unique x-vector in this solution.

Corollary 1.10

The value vector ϕ is the unique v-part of an optimal solution (u, v) of the linear program

$$\min\left\{ \sum_j \beta_j v_j \left| \begin{array}{ll} \sum_j \{\delta_{ij} - p_{ij}(a)\}v_j \geq 0 & \text{for every } (i,a) \in S \times A \\ v_i + \sum_j \{\delta_{ij} - p_{ij}(a)\}u_j \geq \mathrm{r_i}\ (a) & \text{for every } (i,a) \in S \times A \end{array} \right. \right\}, \tag{1.36}$$

where $\beta_j > 0, j \in S$, is arbitrarily chosen.

The dual linear program of (1.36) is

$$\max\left\{ \sum_{(i,a)} r_i(a)x_i(a) \left| \begin{array}{ll} \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}x_i(a) = 0 & j \in \mathrm{S} \\ \sum_a x_j(a) + \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}y_j(a) = \beta_j & j \in \mathrm{S} \\ x_i\ (a), y_i\ (a) \geq 0 & (\mathrm{i,a}) \in \mathrm{S} \times \mathrm{A} \end{array} \right. \right\}. \tag{1.37}$$

Theorem 1.26

Let (x, y) be an extreme optimal solution of (1.37). Then, any $f_0^\infty \in C(D)$, where $x_i\ (f(i)) > 0$ if $\sum_a x_i\ (a) > 0$ and $y_i(f(i)) > 0$ if $\sum_a x_i\ (a) = 0$ is an average optimal policy.

Proof

First, notice that f^∞ is well defined, because for every $j \in S$,

$$\sum_a x_j(a) + \sum_a y_j(a) = \sum_{(i,a)} p_{ij}(a)y_i\ (a) + \beta_j > 0, \ j \in S,$$

Let $S_x = \{i \in S \mid \sum_a x_i\ (a) > 0\}$. Since $x_i\ (f(i)) > 0, \ i \in S_x$ and $y_i(f(i)) > 0, \ i \notin S$, it follows from the complementary slackness property of linear programming that

$$\phi_i + \sum_j \{\delta_{ij} - p_{ij}(f(i))\}u_j = r_i\ (f(i)), \ i \in S_x \tag{1.38}$$

and

$$\sum_j \{\delta_{ij} - p_{ij}(f(i))\}\phi_j = 0, \ i \notin S_x. \tag{1.39}$$

The primal program (1.36) implies $\sum_j \{\delta_{ij} - p_{ij}(a)\}\phi_j \geq 0, \ (i,a) \in S \times A$. Suppose that $\sum_j \{\delta_{kj} - p_{kj}(f(k))\}\phi_j > 0$ for some $k \in S_x$. Since $x_k(f(k)) > 0$, this implies that

$$\sum_j \{\delta_{kj} - p_{kj}(f(k))\}\phi_j \cdot x_k(f(k)) > 0.$$

Furthermore, $\sum_j \{\delta_{ij} - p_{ij}(a)\}\phi_j \cdot x_i(a) \geq 0,\ (i,a) \in S \times A$.

Hence, $\sum_{(i,a)} \sum_j \{\delta_{ij} - p_{ij}(a)\}\phi_j \cdot x_i(a) > 0$.

On the other hand, this result is contradictory to the constraints of the dual program (1.37) from which follows that

$$\sum_{(i,a)} \sum_j \{\delta_{ij} - p_{ij}(a)\}\phi_j \cdot x_i(a) = \sum_{(i,a)} \{\sum_j (\delta_{ij} - p_{ij}(a)) x_i(a)\} \cdot \phi_j = 0.$$

This contradiction implies that

$$\sum_j \{\delta_{ij} - p_{ij}(f(i))\}\phi_j = 0,\ i \in S_x. \tag{1.40}$$

From (1.39) and (1.40) it follows that

$$\sum_j \{\delta_{ij} - p_{ij}(f(i))\}\phi_j = 0. \tag{1.41}$$

We now show that S_x is closed under $P(f)$, i.e. $p_{ij}(f(i)) = 0,\ i \in S_x, j \notin S_x$ Suppose that $p_{kl}(f(k)) > 0$ for some $k \in S_x,\ l \notin S_x$. From the constraints of dual program (1.37) it follows that

$$0 = \sum_a x_l(a) = \sum_{(i,a)} p_{il}(a) x_i(a) \geq p_{kl}(f(k)) x_k(f(k)) > 0, \tag{1.42}$$

implying a contradiction.

Next, we show that the states of $S \setminus S_x$ are transient in the Markov chain induced by $P(f)$. Suppose that $S \setminus S_x$ has an ergodic state. Since S_x is closed, the set $S \setminus S_x$ contains an ergodic class, say $J = \{j_1, j_2, ..., j_m\}$. Since (x, y) is an extreme solution and $y_j(f(j)) > 0,\ j \in J$, the corresponding columns in (1.37) are linearly independent. Because these columns have zeroes in the first N rows, the second parts of these vectors are also independent vectors. Since for components $\delta_{j_i k} - p_{j_i k}(f(j_i)),\ k \in J$, are also linear independent.

However,

$$\sum_{k=1}^m b_k^i = \sum_{k=1}^m \{\delta_{j_i j_k} - p_{j_i j_k}(f(j_i))\} = 1 - 1 = 0,\ i = 1,2,...,m$$

which contradicts the independency of $b^1, b^2, ..., b^m$.

We finish the proof as follows. From (1.40) it follows that $\phi = P(f)\phi$, and consequently we have $\phi = P^*(f)\phi$. Since that states of $S \setminus S_x$ are transient in the Markov chain induced by $P(f)$, the columns of $P^*(f)$ corresponding to $S \setminus S_x$ are zero-vectors. Hence, by (1.38),

$$\phi(f^\infty) = P^*(f) r(f) = P^*(f)\{\phi + \{I - P(f)\}u\} = P^*(f)\phi = \phi,$$

i.e. f^{∞} is an average optimal policy.

Algorithm 1.3 Linear programming algorithm

1. Take any vector β, where $\beta_j > 0,\ j \in S$.

2. Use linear programming algorithm to compute solution (u, v) and (x, y) of the dual pair of linear programs:

$$\min\left\{\sum_j \beta_j v_j \middle| \begin{array}{ll} \sum_j \{\delta_{ij} - p_{ij}(a)\} v_j \geq 0 & \text{for every } (i,a) \in S \times A \\ v_i + \sum_j \{\delta_{ij} - p_{ij}(a)\} u_j \geq \mathrm{r_i}\ (a) & \text{for every } (i,a) \in S \times A \end{array}\right\}$$

and

$$\max\left\{\sum_{(i,a)} r_i(a) x_i(a) \middle| \begin{array}{ll} \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\} x_i(a) = 0 & j \in \mathrm{S} \\ \sum_a x_j(a) + \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\} y_j(a) = \beta_j & j \in \mathrm{S} \\ x_i(a), y_{ij} \geq 0 & (\mathrm{i,a}) \in \mathrm{S} \times \mathrm{A} \end{array}\right\}.$$

4. Take $f^{\infty} \in C(D)$ such that $x_i(f(i)) > 0$ if $\sum_a x_i(a) > 0$ and $y_i(f(i)) > 0$ if $\sum_a x_i(a) = 0$. Then, f^{∞} is an average optimal policy and ϕ is the value vector.

In the average reward case there is in general no one-to-one correspondence between the feasible solution of the dual program (1.37) and the set of stationary policies. The natural formula for mapping feasible solution (x, y) to the set of stationary policies is:

$$\pi_{ia}^{x,y} = \begin{cases} \dfrac{x_i(a)}{\sum_a x_i(a)},\ a \in A(i), i \in S_x \\ \dfrac{y_i(a)}{\sum_a y_i(a)},\ a \in A(i), i \in S \setminus S_x \end{cases}. \tag{1.43}$$

Conversely, for a stationary policy π^{∞}, we define a feasible solution (x^{π}, y^{π}) of the dual program by

$$\begin{cases} x_i^{\pi}(a) = \{\sum_j \beta_j \{P^*(\pi)\}_{ji}\} \cdot \pi_{ia} \\ y_i^{\pi}(a) = \{\sum_j \beta_j \{D(\pi)\}_{ji} + \sum_j \gamma_j \{P^*(\pi)\}_{ji}\} \cdot \pi_{ia} \end{cases}, \tag{1.44}$$

where γ_j is 0 on a transient class and constant on a recurrent class.

If $f^{\infty} \in C(D)$, then the corresponding solution $(x(f), y(f))$ is an extreme solution; the reverse statement is not true.

Chapter 2 Interior point method

2.1 Self-concordant functions

2.1.1 Introduction

In this section, we introduce the notation of a self-concordant function and we derive some properties of such functions. We consider a strictly convex function $\phi: D \to R$, where the domain D is an open convex subset of R^n. Our first aim is to find the minimal value ϕ on its domain D (if it exists).

The classical convergence analysis of Newton's method for minimizing ϕ has some major shortcomings. The first shortcoming is that the analysis uses quantities that are not a priori known, for example uniform lower and upper bounds for the eigenvalues of the Hessian matrix of ϕ on D. The second shortcoming is that while Newton's method is affine invariant, these quantities are not affine invariant. As a result, if we change coordinates by an affine transformation (i.e. replace x by $ax + b, a \neq 0$) this has in essence no effect on the behavior of Newton's method but these quantities all change, and as a result also the iteration bound changes.

A simple and elegant way to avoid these shortcomings was proposed by Nesterov and Nemirovski [10]. They posed an affine invariant condition on the function ϕ, named *self-concordance*. The well known logarithmic barrier functions, that play an important role in interior-point methods for linear and convex optimization, are self-concordant (abbreviated below as SC). The analysis of Newton's method for SC functions does not depend on any unknown constants. As a consequence, the iteration bound resulting from the analysis is invariant under (affine) changes of coordinates.

The aim of this section to provide a brief introduction to the notion of self-concordance, and to recall some results on the behavior of Newton's method when minimizing a SC function.

Having dealt with this we will consider the problem of minimizing a linear function over the closure of D, while assuming that a self-concordant function on D is given.

2.1.2 Epigraphs and closed convex function

In this section and further on, ϕ always denotes a function whose domain D is an open subset

of R^n.

Definition 2.1

The *epigraph* of ϕ is the set $api\,\phi := \{(x,t) : x \in D, \phi(x) \le t\}$.

Definition 2.2

A function is called closed if its epigraph is closed. If, moreover, ϕ is convex then ϕ is called a closed convex function.

Lemma 2.1

Let $\phi : D \to R$ be closed convex function and let $\bar{x}$ belong to the boundary of D. If a sequence $\{x_k\}_{k=0}^{\infty}$ in the domain converges to $\bar{x}$ then $\phi(x_k) \to \infty$.

Proof

Consider the sequence $\{\phi(x_k)\}_{k=0}^{\infty}$. Assume that it is bounded above. Then it has a limit point $\bar{\phi}$. Of course, we can think that this is the unique limit point of the sequence. Therefore,

$$z_k := (x_k, \phi(x_k)) \to (\bar{x}, \bar{\phi}).$$

Note that z_k belongs to the epigraph of ϕ. Since ϕ is a closed function, then also $(\bar{x}, \bar{\phi})$ belongs to the epigraph. But this is a contradiction since $\bar{x}$ does not belong to the domain of ϕ.

We conclude that if the function ϕ is closed convex, then it has the property that $\phi(x)$ approaches infinity when x approaches the boundary of the domain D. This is also expressed by saying that ϕ is a *barrier function* on D.

2.1.3 Definition of the self-concordance property

We want to minimize $\phi : D \to R$ by using Newton's method. Recall that Newton's method is exact if ϕ is a quadratic function. As we will see the self-concordance property guarantees good behavior of Newton's method.

To start with, we consider the case where ϕ is a univariate function. So we assume for the moment that $n = 1$, and that the domain D of the function $\phi : D \to R$ is just an open interval

in R. The third order Taylor polynomial of ϕ around $x \in D$ is given by

$$P_3(\alpha) = \phi(x) + \alpha\phi'(x) + \frac{1}{2}\alpha^2\phi''(x) + \frac{1}{6}\alpha^3\phi'''(x).$$

The self-concordance property bounds the third order term in terms of the second order term, by requiring that

$$\frac{(\phi'''(x)\alpha^3)^2}{(\phi''(x)\alpha^2)^3} = \frac{(\phi'''(x))^2}{(\phi''(x))^3}, x \in D,$$

is bounded above by some uniform constant.

Definition 2.3

Let $\kappa \geq 0$. The univariate function ϕ is called κ-self-concordant if

$$|\phi'''(x)| \leq 2\kappa(\phi''(x))^{\frac{3}{2}}, \forall x \in D. \quad (2.1)$$

Note that this definition assume that $\phi''(x)$ is nonnegative, whence ϕ is convex, and moreover that ϕ is three times differentiable.

It is easy to verify that the property (2.1) is affine invariant. Because, let ϕ be κ-self-concordant and let $\bar{\phi}$ be defined by $\bar{\phi}(y) = \phi(ay + b)$, where $a \neq 0$. Then one has

$$\bar{\phi}'(y) = a\phi'(x), \ \bar{\phi}''(y) = a^2\phi''(x), \ \bar{\phi}'''(y) = a^3\phi'''(x),$$

where $x = ay + b$, hence if follows, due to the exponent $\frac{3}{2}$ in the definition, that $\bar{\phi}$ is κ-self-concordant as well.

Now suppose that $n > 1$, so ϕ is a multivariate function. Then ϕ is called a κ-self-concordant function if its restriction to an arbitrary line in its domain is κ-self-concordant. In other words, we have the following definition.

Definition 2.4

Let $\kappa \geq 0$. The function ϕ is called κ-self-concordant if and only if $\varphi(\alpha) := \phi(x + \alpha h)$ is κ-self-concordant at $\alpha = 0$ for all $x \in D$ and for all $h \in R^n$,

i.e. $$|\varphi'''(0)| \leq 2\kappa\varphi''(0)^{\frac{3}{2}}, \forall x \in D, \forall h \in R^n. \quad (2.2)$$

Here the domain of $\varphi(\alpha)$ is defined in the natural way: given x and h it consists of all α such that $x+\alpha h \in D$. Note that since D is an open convex subset of R^n, the domain of $\varphi(\alpha)$ is an open interval in R.

2.1.4 Equivalent formulations of the self-concordance property

We assume that $\phi : D \to R$, where D is an open convex subset of R^n. To verify if ϕ is SC we need to compute the derivatives of $\varphi(\alpha) = \phi(x+\alpha h)$ at $\alpha = 0$. We have

$$\varphi'(0) = \sum_{i=1}^{n} h_i \frac{\partial \phi(x)}{\partial x_i}$$

$$\varphi''(0) = \sum_{i=1}^{n}\sum_{j=1}^{n} h_i h_j \frac{\partial^2 \phi(x)}{\partial x_i \partial x_j}$$

$$\varphi'''(0) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} h_i h_j h_k \frac{\partial^3 \phi(x)}{\partial x_i \partial x_j \partial x_k}.$$

It will be convenient to use sort-hand notations for the above right-hand side expressions. We denote these expressions respectively as $\nabla\phi(x)[h], \nabla^2\phi(x)[h,h]$ and $\nabla^3\phi(x)[h,h,h]$ respectively. Thus we may write

$$\varphi'(0) = \nabla\phi(x)[h] \qquad = h^T \nabla\phi(x)$$

$$\varphi''(0) = \nabla^2\phi(x)[h,h] \quad = h^T \nabla^2\phi(x) h$$

$$\varphi'''(0) = \nabla^3\phi(x)[h,h,h] = h^T \nabla^3\phi(x)[h] h.$$

As consequence, we have the following lemma, which is immediate from Definition 2.4.

Lemma 2.2

Let ϕ be three times continuously differentiable and $\kappa \geq 0$. Then ϕ is κ-self-concordant if and only if

$$|\nabla^3\phi(x)[h,h,h]| \leq 2\kappa(\nabla^2\phi(x)[h,h])^{\frac{3}{2}}, \forall x \in D. \qquad (2.3)$$

Let ϕ be any three times differentiable convex function with open domain. We will say that ϕ

is self-concordant, without specifying κ, if ϕ is κ-self-concordant for some $\kappa \geq 0$. Obviously, this will be the case if and only if the quotient

$$\frac{(\nabla^3 \phi(x)[h,h,h])^2}{(\nabla^2 \phi(x)[h,h])^3} \tag{2.4}$$

is bounded above by $4\kappa^2$ when x runs through the domain of ϕ and h through all vectors in R^n. Note that the condition for κ-self-concordance is homogeneous in h: if it holds for some h then it holds for any λh, with $\lambda \in R$.
The κ-self-concordance condition bounds the third order term in terms of the second order term in the Taylor expansion. Hence, if it is satisfied, it makes that the second order Taylor expansion locally provides a good quadratic approximation of $\phi(x)$. The latter property makes that Newton's method behaves well on self-concordant functions. This will be shown later on.

In the sequel we use the following notations:

$$g(x) := \nabla \phi(x), \forall x \in D$$

and

$$H(x) := \nabla^2 \phi(x), \forall x \in D.$$

As we will see in the next section, under a very weak assumption the matrix $H(x)$ is always positive definite. As a consequence it defines a norm, according to

$$\| v \| := \sqrt{v^T H(x) v}, v \in R^n.$$

Of course, this norm depends on $x \in D$. We call it the *local Hessian norm* of v at $x \in D$, and it will be denoted as $\| v \|_{H(x)}$, or simply as $\| v \|_x$. Using this notation, the inequality (2.3) can be written as

$$| \nabla^3 \phi(x)[h,h,h] | \leq 2\kappa \| h \|_x^3.$$

We conclude this section with the following characterization of the self-concordance property.

Lemma 2.3

A three time differentiable closed convex function ϕ with open domain D is κ-self-concordance if and only if

$$| \nabla^3 \phi(x)[h_1, h_2, h_3] | \leq 2\kappa \| h_1 \|_x \| h_2 \|_x \| h_3 \|_x$$

holds for any $x \in D$ and all $h_1, h_2, h_3 \in R^n$.

Proof

This statement is nothing but a general property of three-linear forms. For the proof we refer to Lemma A.2 in the appendix.

2.1.5 Positive definiteness of the Hessian matrix

In this section we deal with an interesting, and important, consequence of Lemma 2.1. Before dealing with it, we introduce a useful function. Let $x \in D$ and $0 \neq d \in R^n$ be such that $x + d \in D$. Fixing v, we define for $0 \leq \alpha \leq 1$,

$$q(\alpha) := v^T H(x + \alpha d)v = \| v \|^2_{x+\alpha d} . \tag{2.5}$$

The $q(\alpha)$ is nonnegative and continuous differentiable. The derivative to α is given by

$$q'(\alpha) := v^T(\nabla^3\phi(x+\alpha d)[d])v = \nabla^3\phi(x+\alpha d)[d,v,v] .$$

Using Lemma 2.3 we obtain

$$|q'(\alpha)| = |\nabla^3\phi(x+\alpha d)[d,v,v]| \leq 2\kappa \| d \|_{x+\alpha d} \| v \|^2_{x+\alpha d} = 2\kappa \| d \|_{x+\alpha d}\ q(\alpha) .$$

If $q(\alpha) > 0$ this implies

$$\left| \frac{d \log q(\alpha)}{d\alpha} \right| = \left| \frac{q'(\alpha)}{q(\alpha)} \right| = \frac{|q'(\alpha)|}{q(\alpha)} \leq 2\kappa \| d \|_{x+\alpha d} . \tag{2.6}$$

In the special case where $v = d$ we have $\| d \|_{x+\alpha d} = q(\alpha)^{\frac{1}{2}}$, and hence we then have

$$|q'(\alpha)| \leq 2\alpha q(\alpha)^{\frac{3}{2}} . \tag{2.7}$$

If $q(\alpha) > 0$ this implies

$$\left| \frac{d}{d\alpha} \frac{1}{\sqrt{q(\alpha)}} \right| = \left| \frac{q'(\alpha)}{2q(\alpha)^{\frac{3}{2}}} \right| \leq \kappa . \tag{2.8}$$

Theorem 2.1

Let the closed convex function ϕ with open domain D be κ-self-concordant. If D does not contain a straight line then the Henssian $\nabla^2\phi(x)$ is positive definite at any $x \in D$.

Proof

Suppose that $H(x)$ is not positive definite for some $x \in D$. Then there exists a nonzero vector

$d \in R^n$ such that $d^T H(x) d = 0$ or, equivalent, $\| d \|_x = 0$. Let $q(\alpha) := \| d \|^2_{x+\alpha d}$, just as in (2.5) with $v = d$. Then $q(0) = 0$ and $q(\alpha)$ is nonnegative and continuously differentiable. Now (2.7) gives $q'(\alpha) \le 2\kappa q(\alpha)^{\frac{3}{2}}$. We claim that this implies $q(\alpha) = 0$ for every $\alpha \ge 0$ such that $x + \alpha d \in D$. This is a consequence of the following claim.

Claim

Let $I = [0, a)$ for some $a > 0$ and $q : I \to R_+$. If $q(0) = 0$ and $q'(\alpha) \le 2\kappa q(\alpha)^{\frac{3}{2}}$ for every $\alpha \in I$ then $q(\alpha) = 0$ for every $\alpha \in I$.

Proof

Assume $q(\alpha_1) > 0$ for some $\alpha_1 \in I$, Let

$$\alpha_0 := \min\{\xi : q(\alpha) > 0, \alpha \in (\xi, \alpha_1]\}.$$

Since q is continuous and $q(0) = 0$, we have $0 \le \alpha_0 < \alpha_1$ and $q(\alpha_0) = 0$. Now define

$$h(t) := \frac{1}{\sqrt{q(\alpha_1 - t)}}, t \in [0, \alpha_1 - \alpha_0).$$

Then, since $\alpha_1 - t \in (\alpha_0, \alpha_1]$, the definition of α_0 implies that $h(t)$ is well defined and positive. Note that $h(t)$ goes to ∞ if t approaches $\alpha_1 - \alpha_0$. On the other hand we have

$$h'(t) = \frac{1}{2} \frac{q'(\alpha_1 - t)}{q(\alpha_1 - t)^{\frac{3}{2}}} \le \frac{1}{2} \frac{2\kappa q(\alpha_1 - t)^{\frac{3}{2}}}{q(\alpha_1 - t)^{\frac{3}{2}}} = \kappa$$

and hence $h(t) \le h(0) + \kappa t$ for all $t \in [0, \alpha_1 - \alpha_0)$. Since $h(0) + \kappa t$ remains bounded when t approaches $\alpha_1 - \alpha_0$ we have a contradiction. Hence the claim is proved.

Thus we have shown that $q(\alpha) = 0$ for every $\alpha \ge 0$ such that $x + \alpha d \in D$. This implies that $\phi(x + \alpha d)$ is linear in α, because we have for some $\beta, 0 \le \beta \le \alpha$,

$$\phi(x + \alpha d) = \phi(x) + \alpha d^T g(x) + \frac{1}{2} \alpha^2 q(\beta) = \phi(x) + \alpha d^T g(x).$$

Since D does not contain a straight line there exists an $\overline{\alpha}$ such that $x + \overline{\alpha} d$ belongs to the boundary of D. We may assume that $\overline{\alpha} \ge 0$ (else replace d by $-d$). Since $\lim_{\alpha \uparrow \overline{\alpha}} \phi(x + \alpha d) = \phi(x) + \overline{\alpha} d^T g(x)$, which is finite, this gives conflict with the barrier

property of ϕ on D. Thus the proof is completed.

Corollary 2.1

If ϕ is closed and self-concordant, and D does not contain a line, the $\phi(x)$ has a unique minimizer.

From now on it will be assumed that the hypothesis of Theorem 2.1 if satisfied. So the domain D does not contain a straight line. As a consequence we have

$$\forall x \in D, \forall h \in R^n : \| h \|_x = 0 \Leftrightarrow h = 0 .$$

2.1.6 Some basic inequalities

From now on, we assume that ϕ is strictly convex. By Theorem 2.1 this is the case if ϕ is closed and self-concordant, and D does not contain a line. The Newton step at x is given by

$$\Delta x = -H(x)^{-1} g(x) . \tag{2.9}$$

Suppose that x^* is a minimizer of $\phi(x)$ on D. A basic equation is how we can measure the 'distance' from x to x^*? One obvious measure for the distance in the Euclidean norm $\| x - x^* \|$. But x^* is unknown! So this measure cannot be computed without knowing the minimizer. Therefore we might use the Euclidean norm of $\Delta x, i.e. \| \Delta x \|$, which vanishes only if $x = x^*$. However, instead of the Euclidean norm we use the local Hessian norm and measure the 'distance' from x to x^* by the quantity

$$\lambda(x) := \| \Delta x \|_x = \sqrt{\Delta x^T H(x) \Delta x} = \sqrt{g(x)^T H(x)^{-1} g(x)} . \tag{2.10}$$

Lemma 2.4

Let $x \in D$ and $\alpha \in R_+$ and $d \in R^n$ such that $x + \alpha d \in D$. Then

$$\frac{\| d \|_x}{1 + \alpha\kappa \| d \|_x} \le \| d \|_{x+\alpha d} \le \frac{\| d \|_x}{1 - \alpha\kappa \| d \|_x} ;$$

The left inequality holds for all α such that $1 + \alpha\kappa \| d \|_x > 0$ and the right for all α such that $1 - \alpha\kappa \| d \|_x > 0$.

Proof

Let $q(\alpha) := \| d \|^2_{x+\alpha d}$ just as in (2.5) with $v = d$. Then, from (2.8).

$$\left| \frac{dq(\alpha)^{-\frac{1}{2}}}{d\alpha} \right| = \left| \frac{q'(\alpha)}{2q(\alpha)^{\frac{3}{2}}} \right| \leq \kappa .$$

Consequently, if $x + \alpha d \in D$ then

$$q(0)^{-\frac{1}{2}} - \alpha\kappa \leq q(\alpha)^{-\frac{1}{2}} \leq q(0)^{-\frac{1}{2}} + \alpha\kappa .$$

Since $q(0)^{\frac{1}{2}} = \| d \|_x$ and $q(\alpha)^{\frac{1}{2}} = \| d \|_{x+\alpha d}$, this gives

$$\frac{1}{\| d \|_x} - \alpha\kappa \leq \frac{1}{\| d \|_{x+\alpha d}} \leq \frac{1}{\| d \|_x} + \alpha\kappa ,$$

or equivalently,

$$\frac{1 - \alpha\kappa \| d \|_x}{\| d \|_x} \leq \frac{1}{\| d \|_{x+\alpha d}} \leq \frac{1 + \alpha\kappa \| d \|_x}{\| d \|_x} .$$

Hence, if $1 + \alpha\kappa \| d \|_x > 0$ we obtain

$$\frac{\| d \|_x}{1 + \alpha\kappa \| d \|_x} \leq \| d \|_{x+\alpha d}$$

and if $1 - \alpha\kappa \| d \|_x > 0$ we obtain

$$\| d \|_{x+\alpha d} \leq \frac{\| d \|_x}{1 - \alpha\kappa \| d \|_x} ,$$

proving the lemma.

Lemma 2.5

Let x and d be such that $x \in D, x + d \in D$ and $\kappa \| d \|_x < 1$. Then we have, for any nonzero $v \in R^n$,

$$(1 - \kappa \| d \|_x) \| v \|_x \leq \| v \|_{x+d} \leq \frac{\| v \|_x}{1 - \kappa \| d \|_x} . \qquad (2.11)$$

Proof

Let $q(\alpha) := \| d \|^2_{x+\alpha d}$, just as in (2.5). Then $q(0) = \| v \|^2_x$ and $q(1) = \| v \|^2_{x+d}$. Hence we may write

$$\log \frac{\| v \|_{x+d}}{\| v \|_x} = \frac{1}{2} \log \frac{q(1)}{q(0)} = \frac{1}{2} (\log q(1) - \log q(0)) = \frac{1}{2} \int_0^1 \left(\frac{d \log q(\alpha)}{d\alpha} \right) d\alpha .$$

By (2.6) we have $\left|\frac{d\log q(\alpha)}{d\alpha}\right| \le 2\kappa \| d \|_{x+\alpha d}$. Also using Lemma 2.4 this implies

$$\log\frac{\| v \|_{x+d}}{\| v \|_x} \le \int_0^1 \frac{\kappa \| d \|_x}{1-\alpha\kappa \| d \|_x} d\alpha = -\log(1-\alpha\kappa \| d \|_x)\,|_{\alpha=0}^1 = \log(\frac{1}{1-\kappa \| d \|_x})$$

and

$$\log\frac{\| v \|_{x+d}}{\| v \|_x} \ge -\int_0^1 \frac{\kappa \| d \|_x}{1-\alpha\kappa \| d \|_x} d\alpha = \log(1-\kappa \| d \|_x).$$

Since the log function is monotonically increasing, we obtain from the above inequalities that

$$1-\kappa \| d \|_x \le \frac{\| v \|_{x+d}}{\| v \|_x} \le \frac{1}{1-\kappa \| d \|_x}.$$

This proves the lemma.

Lemma 2.6

Let $x \in D$ and $d \in R^m$. If $\| d \|_x < \frac{1}{\kappa}$ then $x + d \in D$.

Proof

Since $\| d \|_x < \frac{1}{\kappa}$, we have from Lemma 2.5 that $H(x+\alpha d)$ is bounded for all $0 \le \alpha \le 1$, and thus $\phi(x+\alpha d)$ is bounded. On the other hand, ϕ takes infinite values on the boundary of the feasible set, by Lemma 2.1. As a consequence we must have $x + d \in D$.

2.1.7 Quadratic convergence of Newton's method

Let $x^+ := x + \Delta x$ denote the iterate after the Newton step at x. Recall that the Newton step at x is given by

$$\Delta x = -H(x)^{-1} g(x)$$

where $H(x)$ and $g(x)$ are the Hessian matrix and the gradient of $\phi(x)$, respectively.

Recall from (2.10) that we measure the distance from x to the minimizer x^* of $\phi(x)$ by the quantity

$$\lambda(x) = \| \Delta x \|_x = \sqrt{g(x)^T H(x)^{-1} g(x)}.$$

Note that if $x = x^*$ then $g(x) = 0$ and hence $\lambda(x) = 0$; whereas in all other cases $\lambda(x)$ will be positive.

After the Newton step we have

$$\lambda(x^+) = \| \Delta x^+ \|_{x^+} = \| H(x^+)^{-1} g(x^+) \|_{x^+} = \sqrt{g(x^+)^T H(x^+)^{-1} g(x^+)} .$$

We are now ready to prove our first main result on Newton's behavior on self-concordant functions.

Theorem 2.2

If $\lambda(x) \le \frac{1}{\kappa}$ then x^+ is feasible. Moreover, if $\lambda(x) < \frac{1}{\kappa}$ then

$$\lambda(x^+) \le \kappa \left(\frac{\lambda(x)}{1 - \kappa \lambda(x)} \right)^2 .$$

Proof

The feasibility of x^+ follows from Lemma 2.6, since $\| \Delta x \|_x = \lambda(x) \le \frac{1}{\kappa}$.

To prove the second statement in the theorem we denote the Newton step at x^+ shortly as v. So

$$v := H(x^+)^{-1} g(x^+) .$$

For $0 \le \alpha \le 1$ we consider the function

$$k(\alpha) := v^T g(x + \alpha \Delta x) - (1 - \alpha) v^T g(x) .$$

Note that $k(0) = 0$ and

$$k(1) := g(x^+)^T H(x^+)^{-1} g(x^+) = \lambda(x^+)^2 .$$

Taking the derivative of k to α we get, also using $H(x)\Delta x = -g(x)$,

$$k'(\alpha) := v^T H(x + \alpha \Delta x) \Delta x + v^T g(x) = v^T (H(x + \alpha \Delta x) - H(x)) \Delta x .$$

By substituting $d = \alpha \Delta x$ in (2.11) and the definition of local Hessian norm, we can derive

$$H(x + \alpha \Delta x) - H(x) \preceq \left(\frac{1}{(1 - \alpha \kappa \| \Delta x \|_x)^2} - 1 \right) H(x) .$$

Now applying the generalized Cauchy inequality in the Appendix (Lemma A.1) we get

$$v^T (H(x + \alpha \Delta x) - H(x)) \Delta x \le \left(\frac{1}{(1 - \alpha \kappa \| \Delta x \|_x)^2} - 1 \right) \| v \|_x \| \Delta x \|_x .$$

Hence, combining the above results, and using $\| \Delta x \|_x = \lambda(x)$, we may write

$$k'(\alpha) \le \left(\frac{1}{(1 - \alpha \kappa \lambda(x))^2} - 1 \right) \| v \|_x \ \lambda(x) .$$

Therefore, since $k(0) = 0$

$$k(1) \le \lambda(x) \| v \|_x \int_0^1 \left(\frac{1}{(1-\alpha\kappa\lambda(x))^2} - 1 \right) d\alpha = \| v \|_x \frac{\kappa\lambda(x)^2}{1-\kappa\lambda(x)}.$$

Since $v = H(x^+)^{-1} g(x^+)$, we have, by Lemma 2.5,

$$\| v \|_x \le \frac{\| v \|_{x^+}}{1-\kappa \| \Delta x \|_x} = \frac{\lambda(x^+)}{1-\kappa\lambda(x)}.$$

Since $k(1) = \lambda(x^+)^2$, it follows by substitution,

$$\lambda(x^+)^2 = k(1) \le \frac{\lambda(x^+)}{1-\kappa\lambda(x)} \frac{\kappa\lambda(x)^2}{1-\kappa\lambda(x)}.$$

Dividing both sides by $\lambda(x^+)$ the lemma follows.

Corollary 2.2

If $\kappa\lambda(x) \le \frac{1}{2}(3-\sqrt{5}) \approx 0.3820$ then x^+ is feasible and $\lambda(x^+) \le \lambda(x)$.

Corollary 2.3

If $\lambda(x) \le \frac{1}{3\kappa}$ then x^+ is feasible and $\lambda(x^+) \le \kappa(\frac{3}{2}\lambda(x))^2 = (\frac{3}{2}\lambda(x)\sqrt{\kappa})^2$.

2.1.8 Algorithm with full Newton steps

Assuming that we have a point $x \in D$ with $\lambda(x) \le \frac{1}{3\kappa}$ we can easily obtain a point $x \in D$ such that $\lambda(x) \le \varepsilon$, for prescribed $\varepsilon > 0$, with the algorithm 2.1. We assume that ϕ is not linear or quadratic. Then $\kappa > 0$. Actually, from the Definition 2.3, we can easily prove if λ is some positive constant then $\lambda\phi$ is $\left(\frac{\kappa}{\sqrt{\lambda}} \right)$-self-concordant. So we may always assume that $\kappa \ge \frac{4}{9}$. We will assume this from now on.

Algorithm 2.1 (Algorithm with full Newton steps)

Input

An accuracy parameter $\varepsilon \in (0,1)$;

$x \in D$ such that $\lambda(x) \le \frac{1}{3\kappa}$.

while $\lambda(x) \geq \varepsilon$ **do**

$x := x + \Delta x$

endwhile

The following theorem gives an upper bound for the number of iterations required by the algorithm,

Theorem 2.3

Let $x \in D$ and $\lambda(x) \leq \frac{1}{3\kappa}$. Then the algorithm with full Newton steps requires at most

$$\left\lceil {}^{2}\log\left((3.4761\log\frac{1}{\varepsilon}\right)\right\rceil$$

iterations. The output is a point $x \in D$ such that $\lambda(x) \leq \varepsilon$.

Proof

Let $x^0 \in D$ be such that $\lambda(x^0) \leq \frac{1}{3\kappa}$. Starting at x^0 we repeatedly apply full Newton steps until the k-iterate, denoted as x^k, satisfies $\lambda(x^k) \leq \varepsilon$, where $\varepsilon > 0$ is the prescribed accuracy parameter. We can estimate the required number of Newton steps by using Corollary 2.3. To simplify notation we define for the moment $\lambda^0 = \lambda(x^0)$ and $\gamma = \frac{3}{2}\sqrt{\kappa}$. Note that $\gamma \geq 1$. It then follows that

$$\lambda(x^k) \leq (\gamma\lambda(x^{k-1}))^2 \leq (\gamma(\gamma\lambda(x^{k-2}))^2)^2 \leq \cdots \leq \gamma^{2+4+\cdots 2^k}(\lambda^0)^{2^k}.$$

This gives

$$\lambda(x^k) \leq \gamma^{2^{k+1}-2}(\lambda^0)^{2^k} = \gamma^{-2}(\gamma^2\lambda^0)^{2^k} \leq (\gamma^2\lambda^0)^{2^k}.$$

Using the definition of γ and $\lambda(x^0) \leq \frac{1}{3\kappa}$ we obtain

$$\gamma^2\lambda^0 \leq \left(\frac{3}{2}\sqrt{\kappa}\right)^2 \frac{1}{3\kappa} = \frac{3}{4}.$$

Hence, we certainly have $\lambda(x^k) \leq \varepsilon$ if $\left(\frac{3}{4}\right)^{2^k} \leq \varepsilon$. Taking logarithm at both sides this reduces to $2^k \log\frac{3}{4} \leq \log\varepsilon$.

Dividing by $\log\frac{3}{4}$, we get $2^k \geq \frac{\log\varepsilon}{\log\frac{3}{4}}$, or, equivalently, $k \geq \log_2 \frac{\log\varepsilon}{\log\frac{3}{4}}$. Thus we find that after no more than

$$\log_2\left(\frac{\log\varepsilon}{\log\frac{3}{4}}\right)=\log_2(-3.4761\log\varepsilon)=\log_2\left(3.4761\log\frac{1}{\varepsilon}\right)$$

iterations the process will stop and the output will be an $x\in D$ such the $\lambda(x)\le\varepsilon$.

2.1.9 Linear convergence of the damped Newton method

In this section, we consider the case where $x\in D$ lies outside the region where the Newton process is quadratically convergent. More precisely, we assume that $\lambda(x)>\frac{1}{3\kappa}$. In that case we perform a *damped Newton step*, with *damping factor* α, and the new iterate is given by

$$x^+=x+\alpha\Delta x.$$

In the Algorithm 6.2 below, we use $\alpha=\frac{1}{1+\kappa\lambda(x)}$ as a default step size.

Algorithm 2.2

Input:

$x\in D$ such that $\lambda(x)>\frac{1}{3\kappa}$

while $\lambda(x)>\frac{1}{3\kappa}$ **do**

$$\alpha=\frac{1}{1+\kappa\lambda(x)}$$

$$x^+=x+\alpha\Delta x$$

endwhile

In the next theorem we use the function

$$\omega(t):=t-\log(1+t),t>-1. \tag{2.12}$$

Note that this is a strictly convex nonnegative function, which is minimal at $t=0$, and $\omega(0)=0$. The next theorem shows that with an appropriate choice of α we can guarantee a fixed decrease in ϕ after the step.

Theorem 2.4

Let $x\in D$ and $\lambda:=\lambda(x)$. If $\alpha:=\frac{1}{1+\kappa\lambda}$ then

$$\phi(x) - \phi(x + \alpha\Delta x) \geq \frac{\omega(\kappa\lambda)}{\kappa^2}.$$

Proof

Define $\Delta(\alpha) := \phi(x) - \phi(x + \alpha\Delta x)$.

Then $\Delta'(\alpha) := -g(x + \alpha\Delta x)^T \Delta x$

$$\Delta''(\alpha) := -\Delta x^T H(x + \alpha\Delta x)\Delta x = -\nabla^2\phi(x + \alpha\Delta x)[\Delta x, \Delta x]$$

$$\Delta'''(\alpha) := -\nabla^3\phi(x + \alpha\Delta x)[\Delta x, \Delta x, \Delta x].$$

Now using that ϕ is κ-self-concordant, we deduce from the last expression that

$$\Delta'''(\alpha) \geq -2\kappa \| \Delta x \|^3_{x+\alpha\Delta x}.$$

Hence, also using Lemma 2.4

$$\Delta'''(\alpha) \geq -2\kappa \frac{\| \Delta x \|^3_x}{(1 - \alpha\kappa \| \Delta x \|_x)^3} = \frac{-2\kappa\lambda^3}{(1 - \alpha\kappa\lambda)^3}.$$

This information on the third derivative of $\Delta(\alpha)$ is used to prove the theorem, by integrating three times. By integrating once we obtain

$$\Delta''(\alpha) - \Delta''(0) \geq \int_0^\alpha \frac{-2\kappa\lambda^3}{(1 - \beta\kappa\lambda)^3} d\beta = \frac{-\lambda^2}{(1 - \beta\kappa\lambda)^2}\Big|_{\beta=0}^{\alpha} = \frac{-\lambda^2}{(1 - \alpha\kappa\lambda)^2} + \lambda^2.$$

Since $\Delta''(0) = -\nabla^2\phi(x)[\Delta x, \Delta x] = -\lambda^2$, we obtain

$$\Delta''(\alpha) \geq \frac{-\lambda^2}{(1 - \alpha\kappa\lambda)^2}.$$

By integrating once more we derive an estimate for $\Delta'(\alpha)$:

$$\Delta'(\alpha) - \Delta'(0) \geq \int_0^\alpha \frac{-\lambda^2}{(1 - \beta\kappa\lambda)^2} d\beta = \frac{-\lambda}{\kappa(1 - \beta\kappa\lambda)}\Big|_{\beta=0}^{\alpha} = \frac{-\lambda}{\kappa(1 - \alpha\kappa\lambda)} + \frac{\lambda}{\kappa}.$$

Since $\Delta'(0) = -g(x)^T \Delta x = \Delta x^T H(x)\Delta x = \lambda^2$, we obtain

$$\Delta'(\alpha) \geq \frac{-\lambda}{\kappa(1 - \alpha\kappa\lambda)} + \frac{\lambda}{\kappa} + \lambda^2.$$

Finally, in the same way we derive an estimate for $\Delta(\alpha)$. Using that $\Delta(0) = 0$ we have

$$\Delta(\alpha) \geq \int_0^\alpha \left(\frac{-\lambda}{\kappa(1 - \beta\kappa\lambda)} + \frac{\lambda}{\kappa} + \lambda^2 \right) d\beta = \frac{1}{\kappa^2}(\log(1 - \alpha\kappa\lambda) + \alpha\kappa\lambda + \alpha\kappa^2\lambda^2).$$

One may easily verify that the last expression is maximal for $\overline{\alpha} = \frac{1}{1+\kappa\lambda}$. Substitution of this value yields

$$\Delta(\overline{\alpha}) \geq \frac{1}{\kappa^2}\left(\log\left(1-\frac{\kappa\lambda}{1+\kappa\lambda}\right)+\kappa\lambda\right) = \frac{1}{\kappa^2}(\kappa\lambda - \log(1+\kappa\lambda)) = \frac{1}{\kappa^2}\omega(\kappa\lambda),$$

which is the desired inequality.

Since $\omega(t)$ is monotonically increasing for positive t, and $\lambda > \frac{1}{3\kappa}$, the following result is an immediate consequence of Theorem 2.4.

Corollary 2.4

If $\lambda(x) > \frac{1}{3\kappa}$ then x^+ is feasible and

$$\Delta(\alpha) \geq \frac{1}{\kappa^2}\omega(\frac{1}{3}) = \frac{0.0457}{\kappa^2} > \frac{1}{22\kappa^2}.$$

The next result is an obvious consequence of this corollary.

Theorem 2.5

Let $x \in D$ such that $\lambda(x) > \frac{1}{3\kappa}$. If x^* denotes the minimizer of $\phi(x)$, then the algorithm with damped Newton steps requires at most

$$22\kappa^2(\phi(x^0) - \phi(x^*))$$

iterations. The output is a point $x \in D$ such that $\lambda(x) \leq \frac{1}{3\kappa}$.

In order to obtain a solution such that $\lambda(x) \leq \varepsilon$, after the algorithm with damped Newton steps we can proceed with full Newton steps. Due to Theorem 2.3 and Theorem 2.4 we can obtain such a solution after a total of at most

$$\left\lceil 22\kappa^2(\phi(x^0) - \phi(x^*)) \right\rceil + \left\lceil {}^2\log\left((3.4761\log\frac{1}{\varepsilon}\right) \right\rceil \qquad (2.13)$$

iterations. Note the drawback of the above iteration bound: usually we have no prior knowledge of $\phi(x^*)$ and the bound cannot be calculated at the start of the algorithm. However, in many cases we can derive a good estimate for $\phi(x^0) - \phi(x^*)$ and we obtain an upper bound for the number of iterations before starting the optimization process.

2.1.10 Further estimates

In the above analysis, we found an upper bound for the number of iterations that the algorithm needs to yield a feasible point x such that $\lambda(x) \le \varepsilon$. But we can provide more information about $\phi(x) - \phi(x^*)$ and $x - x^*$.

We start with the following lemma.

Lemma 2.7

Let $x \in D$ and $d \in R^n$ such that $x + d \in D$. Then

$$\frac{\|d\|_x^2}{1+\kappa\|d\|_x} \le d^T(g(x+d) - g(x)) \le \frac{\|d\|_x^2}{1-\kappa\|d\|_x}; \tag{2.14}$$

$$\frac{\omega(\kappa\|d\|_x)}{\kappa^2} \le \phi(x+d) - \phi(x) - d^T g(x) \le \frac{\omega(-\kappa\|d\|_x)}{\kappa^2}. \tag{2.15}$$

In the right-hand side inequalities it is assumed that $\kappa\|d\|_x < 1$.

Proof

We have

$$d^T(g(x+d) - g(x)) = \int_0^1 d^T H(x+\alpha d) d \, d\alpha = \int_0^1 \|d\|_{x+\alpha d}^2 \, d\alpha .$$

Using Lemma 2.4 we may write

$$\frac{\|d\|_x^2}{1+\kappa\|d\|_x} = \int_0^1 \frac{\|d\|_x^2}{(1+\alpha\kappa\|d\|_x)^2} d\alpha \le \int_0^1 \|d\|_{x+\alpha d}^2 \, d\alpha$$

$$\le \int_0^1 \frac{\|d\|_x^2}{(1-\alpha\kappa\|d\|_x)^2} d\alpha = \frac{\|d\|_x^2}{1-\kappa\|d\|_x}.$$

From this the inequalities in (2.14) immediately follow. To obtain the inequalities in (2.15) we write

$$\phi(x+d) - \phi(x) - d^T g(x) = \int_0^1 d^T(g(x+\alpha d) - g(x)) d\alpha .$$

Now using the inequalities in (2.14) we obtain

$$\int_0^1 d^T(g(x+\alpha d) - g(x)) d\alpha \le \int_0^1 \frac{\alpha\|d\|_x^2}{1-\alpha\kappa\|d\|_x} d\alpha = \frac{-\kappa\|d\|_x - \log(1-\kappa\|d\|_x)}{\kappa^2}$$

$$= \frac{\omega(-\kappa\|d\|_x)}{\kappa^2}$$

and

$$\int_0^1 d^T(g(x+\alpha d)-g(x))d\alpha \geq \int_0^1 \frac{\alpha\|d\|_x^2}{1+\alpha\kappa\|d\|_x}d\alpha = \frac{\kappa\|d\|_x - \log(1+\kappa\|d\|_x)}{\kappa^2}$$

$$= \frac{\omega(\kappa\|d\|_x)}{\kappa^2}.$$

This completes the proof.

As usual, for each $x \in D$, $\lambda(x) = \|\Delta x\|_x$, with Δx denoting the Newton step at x. We now prove that if $\lambda(x) < \frac{1}{\kappa}$ for some $x \in D$ then ϕ must have a minimizer. Note that this surprising result expresses that some local condition on ϕ provides us with a global property, namely the existence of a minimizer.

Theorem 2.6

Let $\lambda(x) < \frac{1}{\kappa}$ for some $x \in D$. Then ϕ has a unique minimizer x^* in D.

Proof

The proof is based on the observation that the level set

$$L := \{y \in D : \phi(y) \leq \phi(x)\},$$

with x as given in the theorem, is compact. This can be seen as follows. Let $y \in D$. Writing $y = x + d$, with $d \in R^n$, Lemma 2.7 implies the inequality

$$\phi(y) - \phi(x) \geq d^T g(x) + \frac{\omega(\kappa\|d\|_x)}{\kappa^2} = -d^T H(x)\Delta x + \frac{\omega(\kappa\|d\|_x)}{\kappa^2},$$

where we used that, by definition, the Newton step Δx at x satisfies $H(x)\Delta x = -g(x)$. Since

$$d^T H(x)\Delta x \leq \|d\|_x \|\Delta x\|_x = \|d\|_x \lambda(x),$$

we thus have

$$\phi(y) - \phi(x) \geq -\|d\|_x \lambda(x) + \frac{\omega(\kappa\|d\|_x)}{\kappa^2}.$$

Now let $y = x + d \in L$. Then $\phi(y) \leq \phi(x)$, whence we obtain

$$-\|d\|_x \lambda(x) + \frac{\omega(\kappa\|d\|_x)}{\kappa^2} \leq 0,$$

which implies

$$\frac{\omega(\kappa\|d\|_x)}{\kappa\|d\|_x} \le \kappa\lambda(x) < 1. \qquad (2.16)$$

Putting $\xi := \kappa\|d\|_x$ one may easily verify that $\omega(\xi)/\xi$ is monotonically increasing for $\xi > 0$ and goes to 1 if $\xi \to \infty$. Therefore, since $\kappa\lambda(x) < 1$, we may conclude from (2.16) that $\kappa\|d\|_x$ cannot be arbitrary large. In other words, $\kappa\|d\|_x$ is bounded above. This means that the set of vectors d such that $x + d \in L$ is bounded. This implies that the level set L itself is bounded. Since this set is also closed, the set L is compact. Hence ϕ has a minimal value in L, and this value is attained at some $x^* \in L$. Since ϕ is convex, x^* is a global minimizer of ϕ, and by Corollary 2.1, this minimizer is unique.

Lemma 2.8

For $s < 1$ one has

$$\omega(-s) = \sup_{t>-1}\{st - \omega(t)\},$$

whence

$$\omega(-s) + \omega(t) \ge st, \quad s < 1, \quad t > -1.$$

Proof

Let $F(s,t) = \omega(-s) + \omega(t) - st$. Hence

$$\begin{aligned} F(s,t) &= -s - \log(1-s) + t - \log(1+t) - st \\ &= -s - \log(1-s) + t - \log(1+t) - st \\ &= -s + t - st - \log(1 - s + t - st). \end{aligned}$$

Let $x = -s + t - st$, then

$$F(s,t) = \omega(x), \quad x > -1.$$

It is easy to see that $\omega(x) \ge 0$, so $F(s,t) \ge 0$.

Hence we get

$$\omega(-s) + \omega(t) \ge st.$$

Theorem 2.7

Let $x \in D$ be such that $\lambda(x) < \frac{1}{\kappa}$ and let x^* denote the unique minimizer of ϕ. Then, with

$\lambda := \lambda(x)$,

$$\frac{\omega(\kappa\lambda)}{\kappa^2} \le \phi(x) - \phi(x^*) \le \frac{\omega(-\kappa\lambda)}{\kappa^2} \tag{2.17}$$

$$\frac{\omega'(\kappa\lambda)}{\kappa} = \frac{\lambda}{1+\kappa\lambda} \le \|x - x^*\|_x \le \frac{\lambda}{1-\kappa\lambda} = -\frac{\omega'(-\kappa\lambda)}{\kappa}. \tag{2.18}$$

Proof

The left inequality in (2.17) follows from Theorem 2.4, because ϕ is minimal at x^*. Furthermore from (2.15) in Lemma 2.7, with $d = x^* - x$, we get the right inequality in (2.17):

$$\begin{aligned}
\phi(x^*) - \phi(x) &\ge d^T g(x) + \frac{\omega(\kappa\|d\|_x)}{\kappa^2} \\
&\ge -\|d\|_x \lambda + \frac{\omega(\kappa\|d\|_x)}{\kappa^2} \\
&= \frac{1}{\kappa^2}\left(-\kappa\|d\|_x \kappa\lambda + \omega(\kappa\|d\|_x)\right) \\
&\ge -\frac{\omega(-\kappa\lambda)}{\kappa^2},
\end{aligned}$$

where the second inequality holds since

$$\left|d^T g(x)\right| = \left|-d^T H(x)\Delta x\right| \le \|d\|_x \|\Delta x\|_x = \|d\|_x \lambda(x) = \|d\|_x \lambda \tag{2.19}$$

and the fourth inequality follows from Lemma 2.8.

For the proof of (2.18) we first derive from (2.19) and the (2.14) in Lemma 2.7 that

$$\frac{\|d\|_x^2}{1+\kappa\|d\|_x} \le d^T(g(x^*) - g(x)) = -d^T g(x) \le \|d\|_x \lambda,$$

where we used that $g(x^*) = 0$. Dividing by $\|d\|_x$ we get

$$\frac{\|d\|_x}{1+\kappa\|d\|_x} \le \lambda,$$

which gives rise to the right inequality in (2.18), since it follows now that

$$\|d\|_x \le \frac{\lambda}{1-\kappa\lambda} = -\frac{\omega'(-\kappa\lambda)}{\kappa}.$$

Note that the left inequality in (2.18) is trivial if $\kappa\|d\|_x \ge 1$, because then $\|d\|_x \ge 1/\kappa$, whereas $\frac{\lambda}{1+\kappa\lambda} < \frac{1}{\kappa}$. Thus we may assume that $1 - \kappa\|d\|_x > 0$. For $0 \le \alpha \le 1$, consider

$$k(\alpha) := g(x^* - \alpha d)^T H(x)^{-1} g(x).$$

One has $k(0) = 0$ and $k(1) = \lambda(x)^2 = \lambda^2$. From (2.11) and the Cauchy inequality we get

$$k'(\alpha) = -d^T H(x^* - \alpha d) H(x)^{-1} g(x) = d^T H(x^* - \alpha d)\Delta x \le \frac{\|d\|_x \lambda(x)}{(1-\kappa\|d\|_x)^2}.$$

Hence we have

$$\lambda^2 = k(1) \le \int_0^1 \frac{\|d\|_x \lambda}{(1-\kappa\|d\|_x)^2} d\alpha = \frac{\|d\|_x \lambda}{1-\kappa\|d\|_x}.$$

After dividing both sides by λ this implies

$$\|d\|_x \ge \frac{\lambda}{1+\kappa\lambda}.$$

Thus the proof is complete.

2.2 Minimization of a linear function over a closed convex domain

2.2.1 Introduction

In this section, we consider the problem of minimizing a linear function over a closed convex domain $\overline{D}$:

$$(P) \qquad \min\{c^T x : x \in \overline{D}\}.$$

We assume that we have a self-concordant function $\phi : D \to R$, where $D = \operatorname{int} \overline{D}$, and also that $H(x) = \nabla^2 \phi(x)$ is positive definite for every $x \in D$.

For each $\mu > 0$ we define

$$\phi_\mu(x) := \frac{c^T x}{\mu} + \phi(x), \quad x \in D$$

and we consider the problem

$$(P_\mu) \qquad \inf\{\phi_\mu(x) : x \in D\}.$$

We denote the gradient and Hessian matrix of $\phi_\mu(x)$ as $g_\mu(x)$ and $H_\mu(x)$, respectively. Then we may write

$$g_\mu(x) := \nabla \phi_\mu(x) = \frac{c}{\mu} + \nabla \phi(x) = \frac{c}{\mu} + g(x) \tag{2.20}$$

and

$$H_\mu(x) := \nabla^2 \phi_\mu(x) = \nabla^2 \phi(x) = H(x). \tag{2.21}$$

An immediate consequence of (2.21) is $\nabla^3 \phi_\mu(x) = \nabla^3 \phi(x)$.

So it becomes clear that the second and third derivative of $\phi_\mu(x)$ coincide with the second and third derivatives of $\phi(x)$, and do not depend on μ. Assuming that $\phi(x)$ is κ-self-concordant, if follows that $\phi_\mu(x)$ is κ-self-concordant as well.

The minimizer of $\phi_\mu(x)$, if it exists, is denoted as $x(\mu)$. When μ runs through all positive numbers then $x(\mu)$ runs through the so-called *central path* of (P). We expect that $x(\mu)$ converges to an optimal solution of (P) when μ approaches 0, since then the linear term in the objective function of (P_μ) dominates the remaining part. Therefore, our aim is to use the central path as a guideline to the optimal solution of (P). This approach is likely to be feasible, because since $\phi_\mu(x)$ is self-concordant its minimizer can be computed efficiently.

The Newton step at $x \in D$ with respect to $\phi_\mu(x)$ is given by $\Delta x = -H(x)^{-1} g_\mu(x)$.

Just as in the previous section we measure the distance of $x \in D$ to the μ-center $x(\mu)$ by the local norm of Δx. So for this purpose we use the quantity $\lambda_\mu(x)$ defined by

$$\lambda_\mu(x) = \| \Delta x \|_x = \sqrt{\Delta x^T H(x) \Delta x} = \sqrt{g_\mu(x)^T H(x)^{-1} g_\mu(x)} = \| g_\mu(x) \|_{H^{-1}}.$$

Before presenting the algorithm we need to deal with two issues. First, when is μ small enough? We want to have the guarantee that the algorithm generates a feasible point whose objective value deviates no more than ε from the optimal value, where $\varepsilon > 0$ is some prescribed accuracy parameter. Second, we need to know what the effect is of an update of μ on our proximity measure $\lambda_\mu(x)$. We start with the second issue.

2.2.2 Effect of μ-update

Let $\lambda := \lambda_\mu(x)$ and $\mu^+ = (1-\theta)\mu$. Our aim is to estimate $\lambda_{\mu^+}(x)$. We have

$$\begin{aligned} g_{\mu^+}(x) &= \frac{c}{\mu^+} + \nabla\phi(x) = \frac{c}{(1-\theta)\mu} + \nabla\phi(x) = \frac{c}{(1-\theta)\mu} + g(x) \\ &= \frac{1}{1-\theta}\left(\frac{c}{\mu} + g(x) - \theta g(x) \right) = \frac{1}{1-\theta}\left(g_\mu(x) - \theta g(x) \right). \end{aligned}$$

Hence, denoting $H(x)$ shortly as H, we may write

$$\lambda_{\mu^+}(x) = \frac{1}{1-\theta}\left\|g_\mu(x) - \theta g(x)\right\|_{H^{-1}} \le \frac{1}{1-\theta}\left(\underbrace{\left\|g_\mu(x)\right\|_{H^{-1}}}_{\lambda_\mu(x)} + \theta\underbrace{\left\|g(x)\right\|_{H^{-1}}}_{\lambda(x)}\right)$$

$$= \frac{1}{1-\theta}(\lambda_\mu(x) + \theta\lambda(x)). \qquad (2.22)$$

At present we have no means to obtain an upper bound for the quantity $\lambda(x)$. Therefore, we use the following definition.

Definition 2.5

Let $\nu \ge 0$. The self-concordant function ϕ is called a ν-barrier if

$$\lambda(x)^2 = \left\|g(x)\right\|^2_{H^{-1}} \le \nu, \qquad \forall x \in D. \qquad (2.23)$$

An immediate consequence of this definition and (2.22) is the following lemma, which requires no further proof.

Lemma 2.9

If ϕ is a self-concordant ν-barrier then

$$\lambda_{\mu^+}(x) \le \frac{\lambda_\mu(x) + \theta\sqrt{\nu}}{1-\theta}.$$

In the sequel we shall say that ϕ is a ν-barrier function if it satisfies (2.23). If ϕ is also κ-self-concordant then we say that ϕ is a (κ,ν)-barrier function.

Here we present an obvious fact which is important for the MDP model:

Corollary 2.5

$\phi(x) = -\sum_{i=1}^n \log x_i$ is a 1-self-concordant n-barrier function for $R^n_+ = \{x \in R^n : x \ge 0)$.

Proof

With e denoting the all-one vector, for $\forall h \in R^n$,

$$g(x) = \nabla\phi(x) = \frac{-e}{x};$$

$$H(x) = \nabla^2\phi(x) = diag(\frac{e}{x^2});$$

$$\nabla H(x) = \nabla^3\phi(x)[h] = diag(\frac{-2h}{x^3}).$$

Hence, we have for any $\forall h \in R^n$

$$\left|\nabla^3\phi(x)[h,h,h]\right| = \left|\sum_{i=1}^{n}\frac{-2h_i^3}{x_i^3}\right|$$

and

$$\nabla^2\phi(x)[h,h] = h^T diag\left(\frac{e}{x^2}\right)h = \sum_{i=1}^{n}\frac{h_i^2}{x_i^2}.$$

For any $\xi \in R^n$ one has

$$\left|\sum_{i=1}^{n}\xi_i^3\right| \le \sum_{i=1}^{n}\left|\xi_i\right|^3 \le \left(\sum_{i=1}^{n}\left|\xi_i^2\right|\right)^{\frac{3}{2}}.$$

Hence, taking $\xi_i = \frac{h_i}{x_i}$ we get

$$\left|\nabla^3\phi(x)[h,h,h]\right| \le 2\left(\nabla^2\phi(x)[h,h]\right)^{\frac{3}{2}}$$

proving that $\phi(x)$ is 1-self-concordant.

Since $H(x) = \nabla^2\phi(x) = diag(\frac{e}{x^2})$, we have

$$H(x)^{-1} = diag(x^2).$$

Then

$$\| g(x) \|_{H^{-1}}^2 = g(x)^T H(x)^{-1} g(x) = n.$$

So, we can conclude $\phi(x)$ is a 1-self-concordant n-barrier for R_+^n.

Before proceeding to the next section, we introduce the so-called *Dikin-ellipsoid* at x, and using this we give a new characterization of our proximity measure $\lambda(x)$.

Definition 2.6

For any $x \in D$ the *Dikin-ellipsoid* at x is defined by

$$\varepsilon_x := \{d \in R^n : \|d\|_x \le 1\}.$$

Lemma 2.10

For any $x \in D$ one has

$$\max\left\{\left|d^T g(x)\right| : d \in \varepsilon_x\right\} = \lambda(x).$$

Proof

Due to Definition 2.6 the maximization problem in the lemma can be reformulated as

$$\max\left\{\left|d^T g(x)\right| : d^T H(x) d \le 1\right\}.$$

If $g(x) = 0$ then the lemma is obviously true, because then $\lambda(x) = 0$. So we may assume that $g(x) \neq 0$ and $\lambda(x) \neq 0$. In that case any optimal solution d will certainly satisfy $d^T H(x) d = 1$. Hence, if d is optimal then

$$g(x) = \alpha H(x) d, \ \alpha \in R$$

where α is a Lagrange multiplier. This implies $\alpha d = H(x)^{-1} g(x) = -\Delta x$, where Δx denotes the Newton step at x with respect to ϕ. Now $d^T H(x) d = 1$ implies $\Delta x^T H(x) \Delta x = \alpha^2$. Since we also have $\Delta x^T H(x) \Delta x = \lambda(x)^2$, it follows that $\alpha = \pm\lambda(x)$. So we get

$$d = \pm \frac{\Delta x}{\lambda(x)},$$

whence, using $H(x)\Delta x = -g(x)$,

$$\left|d^T g(x)\right| = \frac{\left|g(x)^T \Delta x\right|}{\lambda(x)} = \frac{\Delta x^T H(x) \Delta x}{\lambda(x)} = \frac{\lambda(x)^2}{\lambda(x)} = \lambda(x)$$

proving the lemma.

For future use we also state the following result.

Lemma 2.11

If ϕ is a self-concordant ν-barrier then we have

$$\left(d^T g(x)\right)^2 \le \nu d^T H(x) d, \ \ \forall \mathrm{d} \in \mathrm{R}^{\mathrm{n}}, \ \forall x \in D.$$

Proof

The inequality in the lemma is homogeneous in d. Hence we may assume that $d^T H(x) d = 1$. Now Lemma 2.10 implies that $\left|d^T g(x)\right| \le \lambda(x)$. Hence we obtain $\left(d^T g(x)\right)^2 \le \lambda(x)^2$. By

Definition 2.5 this implies the lemma.

Assuming that (P) has x^* as optimal solution, we proceed with estimating the objective value $c^T x$ in terms of μ and $\lambda_\mu(x)$. This is the subject in the next section.

2.2.3 Estimate of $c^T x - c^T x^*$

For the analysis of our algorithm we will need some more lemmas.

Lemma 2.12

Let ϕ be a self-concordant ν-barrier function and $x \in D$ and $x + d \in \overline{D}$. Then

$$d^T g(x) \le \nu .$$

Proof
Consider the function

$$q(\alpha) = d^T g(x + \alpha d), \quad \alpha \in [0,1) .$$

Observe that $q(0) = d^T g(x)$. So we need to show that $q(0) \le \nu$. If $q(0) \le 0$ there is nothing to prove. Therefore, assume that $q(0) > 0$. Since $\phi(x)$ is a ν-barrier, we have by Lemma 2.11, for any $\alpha \in [0,1)$,

$$q'(\alpha) = d^T H(x + \alpha d) d \ge \frac{1}{\nu}\left(d^T g(x + \alpha d)\right)^2 = \frac{1}{\nu}\left(q(\alpha)\right)^2 .$$

Therefore, $q(\alpha)$ is increasing and hence positive for $\alpha \in [0,1]$. Therefore, we may write

$$\frac{1}{\nu} \le \int_0^1 \frac{q'(\alpha)}{\left(q(\alpha)\right)^2} d\alpha = -\frac{1}{q(\alpha)}\Big|_0^1 = \frac{1}{q(0)} - \frac{1}{q(1)} < \frac{1}{q(0)} .$$

This imples $q(0) < \nu$, completing the proof of the lemma.

Before proceeding we recall the definition of a dual norm.

Definition 2.7

Given any norm $\|\cdot\|$ in R^n, the corresponding dual norm $\|\cdot\|^*$ is defined by

$$\|s\|^* = \max\{s^T x : \|x\| \le 1\}.$$

For any $x \in D$ we denote the dual norm of the local norm $\|\cdot\|_x$ as $\|\cdot\|_x^*$. Apparently, $\|\cdot\|_x^*$ is the local norm determined by $H(x)^{-1}$. So,

$$\|d\|_x^* = \sqrt{d^T H(x)^{-1} d}, \quad \mathrm{d} \in \mathrm{R}^{\mathrm{n}}.$$

Lemma 2.13

Let $\lambda := \lambda_u(x) < \frac{1}{\kappa}$ and let x^* be an optimal solution of (P). Then

$$c^T x \le c^T x^* + \mu\left(\nu + \frac{\lambda(\lambda + \sqrt{\nu})}{1 - \kappa\lambda}\right).$$

Proof

First we consider the case there $x = x(\mu)$. Since then $g_\mu(x) = 0$, we derive from (2.20) that $c = -\mu g(x)$. Since $x \in D$ and $x^* \in \overline{D}$, using Lemma 2.12 with $d = x^* - x$, we get

$$c^T x(\mu) - c^T x^* = c^T\left(x(\mu) - x^*\right) = \mu g(x)^T d \le \mu\nu.$$

Now let us turn to the general case. Then, using (2.20) once more and also the inequality:

$$a^T b \le \|a\|_x^* \|b\|_x, a, b \in R^n,$$

we may write

$$\begin{aligned} c^T x - c^T x(\mu) &= c^T\left(x - x(\mu)\right) = \mu\left(g_\mu(x) - g(x)\right)^T (x - x(\mu)) \\ &\le \mu\left\|g_\mu(x) - g(x)\right\|_x^* \|x - x(\mu)\|_x. \end{aligned}$$

where $\|\cdot\|_x^*$ denotes the local norm determined by $H(x)^{-1}$. Since $\left\|g_\mu(x)\right\|_x^* = \lambda_\mu(x) = \lambda$ and $\left\|g(x)\right\|_x^* = \lambda(x) \le \sqrt{\nu}$ we have

$$\left\|g_\mu(x) - g(x)\right\|_x^* \le \left\|g_\mu(x)\right\|_x^* + \left\|g(x)\right\|_x^* \le \lambda + \sqrt{\nu}.$$

Moreover, by Theorem 2.7,

$$\|x - x(\mu)\|_x \le \frac{\lambda}{1 - \kappa\lambda}.$$

Substitution gives

$$c^T x - c^T x(\mu) \le \mu \frac{\lambda(\lambda + \sqrt{\nu})}{1 - \kappa\lambda}.$$

Hence we may write

$$c^T x - c^T x^* = c^T (x(\mu) - x^*) + c^T (x - x(\mu)) \leq \mu\left(\nu + \frac{\lambda(\lambda + \sqrt{\nu})}{1 - \kappa\lambda} \right),$$

proving the lemma.

2.2.4 Algorithm with full Newton steps

We assume that we know a point $x^0 \in D$ and $\mu^0 > 0$ such that $\lambda_{\mu^0}(x^0) \leq \tau = \frac{1}{4\kappa}$. Then we decrease $\mu = \mu^0$ with a factor $1 - \theta$, where the barrier update parameter θ is a suitable number in the interval $(0,1)$, and perform a full Newton step. This process is repeated until μ is small enough, i.e. until $\nu\mu \leq \varepsilon$ for some small number ε. The algorithm is described below. The number of iterations is completely determined by ν, μ^0 and θ, according to the lemma stated after the algorithm.

Algorithm 2.3 Algorithm with full Newton steps

Input

an accuracy parameter $\varepsilon > 0$;

a proximity parameter $\tau \in (0, \frac{1}{\kappa})$;

an update parameter $\theta, 0 < \theta < 1$;

$x^0 \in D$ and $\mu^0 > 0$ such that $\lambda_{\mu^0}(x^0) \leq \tau$;

begin

$x := x^0; \mu := \mu^0$;

while $\nu\mu > \varepsilon$ **do**

$\mu := (1 - \theta)\mu;$

$x := x + \Delta x;$

endwhile

end

Lemma 2.14

The number of iterations of the algorithm does not exceed the number

$$\frac{1}{\theta} \log \frac{\nu\mu^0}{\varepsilon}.$$

Proof

The algorithm stops when $\nu\mu \le \varepsilon$. After the k-th iteration we have $\mu = (1-\theta)^k \mu^0$, where μ^0 denotes the initial value of μ. Hence the algorithm will stop if

$$(1-\theta)^k \mu^0 \nu \le \varepsilon .$$

Taking logarithms at both sides this gives

$$k \log(1-\theta) \le \log \frac{\varepsilon}{\mu^0 \nu} .$$

Since $-\log(1-\theta) \ge \theta$, this certainly holds if

$$k\theta \ge \log \frac{\nu\mu^0}{\varepsilon},$$

which implies the lemma.

Theorem 2.8

If $\tau = \frac{1}{9\kappa}$ and $\theta = \frac{5}{9 + 36\kappa\sqrt{\nu}}$, then the algorithm with full Newton steps is well-defined and requires not more than

$$\left\lceil \frac{9 + 36\kappa\sqrt{\nu}}{5} \log \frac{\nu\mu^0}{\varepsilon} \right\rceil$$

iterations. The output is a point $x \in D$ such that

$$c^T x \le c^T x^* + \varepsilon\left(1 + \frac{1 + 9\kappa\sqrt{\nu}}{72\kappa^2\nu}\right),$$

where x^* denotes an optimal solution of (P).

Proof

We need to find values of τ and θ that makes the algorithm well-defined. At the start of the first iteration we have $x = x^0 \in D$ and $\mu = \mu^0$ such that $\lambda_\mu(x) \le \tau$. When the barrier parameter is updated to $\mu^+ = (1-\theta)\mu$, Lemma 2.9 gives

$$\lambda_{\mu^+}(x) \le \frac{\lambda_\mu(x) + \theta\sqrt{\nu}}{1-\theta} \le \frac{\tau + \theta\sqrt{\nu}}{1-\theta} . \qquad (2.24)$$

Then after the Newton step, the new iteration is $x^+ = x + \Delta x$ and

$$\lambda_{\mu^+}(x^+) \le \kappa\left(\frac{\lambda_{\mu^+}(x)}{1-\kappa\lambda_{\mu^+}(x)}\right)^2. \qquad (2.25)$$

The algorithm is well defined if we choose τ and θ such that $\lambda_{\mu^+}(x^+) \le \tau$. To get the lowest iteration bound, we need at the same time to maximize θ. From (2.25) we deduce that $\lambda_{\mu^+}(x^+) \le \tau$ certainly holds if

$$\frac{\lambda_{\mu^+}(x)}{1-\kappa\lambda_{\mu^+}(x)} \le \frac{\sqrt{\tau}}{\sqrt{\kappa}},$$

which is equivalent to

$$\lambda_{\mu^+}(x) \le \frac{\sqrt{\tau}}{\kappa\sqrt{\tau}+\sqrt{\kappa}}. \qquad (2.26)$$

According to (2.24) –and hence $\lambda_{\mu^+}(x^+) \le \tau$ – this will hold if

$$\frac{\tau+\theta\sqrt{\nu}}{1-\theta} \le \frac{\sqrt{\tau}}{\kappa\sqrt{\tau}+\sqrt{\kappa}}.$$

This leads us to the following condition on θ:

$$\theta \le \sqrt{\tau}\,\frac{1-\kappa\tau-\sqrt{\kappa\tau}}{\sqrt{\tau}+\sqrt{\nu\kappa}(1+\sqrt{\nu\kappa})}.$$

Substitution of $\tau = \frac{1}{9\kappa}$ in the right-hand side expression yields the value $\frac{5}{9+36\kappa\sqrt{\nu}}$. Thus we have justified the choice of the value of τ and θ in the theorem

Now that θ is given, the iteration bound is immediate from Lemma 2.14. The last statement in the theorem is implied by Lemma 2.13. because at termination of the algorithm we have $9\kappa\lambda_\mu(x) < 1$ and $\nu\mu \le \varepsilon$. Hence, denoting $\lambda = \lambda_\mu(x)$, Lemma 2.13 implies that

$$\begin{aligned} c^T x &\le c^T x^* + \nu\mu\left(1+\frac{\lambda(\lambda+\sqrt{\nu})}{\nu(1-\kappa\lambda)}\right) \\ &\le c^T x^* + \varepsilon\left(1+\frac{\frac{1}{9\kappa}(\frac{1}{9\kappa}+\sqrt{\nu})}{\nu(\frac{8}{9})}\right) \\ &\le c^T x^* + \varepsilon\left(1+\frac{1+9\kappa\sqrt{\nu}}{72\kappa^2\nu}\right). \end{aligned}$$

This completes the proof.

2.2.5 Algorithm with damped Newton steps

The method that we considered in the previous sections is in practice rather slow. This is due to the fact that the barrier update parameter θ is rather small. For example, in the case of linear optimization the set $\overline{D}$ is the intersection of R^n and the affine space $\{x : Ax = b\}$, for some A and b. From Corollary 2.5, we know that the logarithmic barrier function $\phi(x) = -\sum_{i=1}^{n} \log x_i$ is a 1-self-concordant n-barrier function for R_+^n. In that case we have $\kappa = 1$ and $\nu = n$, and hence the value of θ is given by $\theta = \frac{5}{9+36\sqrt{n}}$. Assuming $\mu^0 = 1$ in Theorem 2.8, this leads to the iteration bound

$$2(1+4\sqrt{n})\log\frac{n}{\varepsilon} = \mathrm{O}\left(\sqrt{n}\log\frac{n}{\varepsilon}\right),$$

which is up till now the best know bound for linear optimization.

In practice one is tempted to accelerate the algorithm by taking larger values of θ. But this is not justified by the theory, and in fact may cause the algorithm to fail because the full Newton step may yield an infeasible point. However, by *damping* the Newton step we can keep these iterates feasible. In this section we investigate the resulting method, which is in practice much faster than the full-Newton step method. So we consider in this section the case where θ is some small (but fixed) constant in the interval $(0,1)$, for example $\theta = 0.5$ or $\theta = 0.99$, and where the new iterate is obtained from

$$x^+ = x + \alpha\Delta x,$$

where Δx is the Newton step at x and where α is the so-called *damping factor*, which is also taken from the interval $(0,1)$, but which has to be carefully chosen.

The algorithm is described below. We refer to the first **while**-loop in the algorithm as the *outer loop* and to the second **while**-loop as the *inner loop*. Each execution of the outer loop is called an *outer iteration* and each execution of the inner loop an *inner iteration*. The main task in the analysis of the algorithm is to derive an upper bound for the number of iterations in the inner loop, because the number of outer iterations follows from Lemma 2.14.

Algorithm 2.4 Algorithm with damped Newton steps

Input

an accuracy parameter $\varepsilon > 0$;

a proximity parameter $\tau = \frac{1}{3\kappa}$;

an update parameter $\theta, 0 < \theta < 1$;

$x^0 \in D$ and $\mu^0 > 0$ such that $\lambda_{\mu^0}(x^0) \le \tau$

begin

$x := x^0; \mu := \mu^0;$

while $\nu\mu > \varepsilon$ **do**

$\mu := (1-\theta)\mu;$

While $\lambda_\mu(x) > \tau$ **do**

$$\alpha = \frac{1}{1+\kappa\lambda_\mu(x)};$$

$x = x + \alpha\Delta x;$

endwhile

endwhile

end

As we will see, in the analysis of the algorithm many results can be used that we already obtained in the analysis of the algorithm for minimizing a self-concordant function with damped Newton steps, in section 2.1.9

Due to the choice of the damping factor α in the algorithm, Theorem 2.4 implies that in each inner iteration the decrease in the value of ϕ_μ satisfies

$$\phi_\mu(x) - \phi_\mu(x + \alpha\Delta x) \ge \frac{\omega(\kappa\lambda_\mu(x))}{\kappa^2}.$$

Since during each inner iteration $\lambda_\mu(x) \ge \tau$ and $\tau > \frac{1}{3\kappa}$, we obtain

$$\phi_\mu(x) - \phi_\mu(x + \alpha\Delta x) \ge \frac{\omega(\kappa\tau)}{\kappa^2} > \frac{1}{\kappa^2}\omega(\frac{1}{3}) = \frac{0.0457}{\kappa^2} > \frac{1}{22\kappa^2}.$$

Thus we see that each inner iteration decreases the value of ϕ_μ with at least $\frac{1}{22\kappa^2}$.

This implies that we can easily find an upper bound for the number of inner iterations during one outer iteration if we know the difference between the values of ϕ_μ at the start and at the end of one outer iteration. Since $\phi_{\mu^+}(x)$ is minimal at $x = x(\mu^+)$, this difference is not larger than

$$\phi_{\mu^+}(x) - \phi_{\mu^+}(x(\mu^+)),$$

where x denotes the iterate at the start of an outer iteration and $\mu^+ = (1-\theta)\mu$ the value of the barrier parameter after the μ-update.

The proofs of the next two lemmas follow similar argument as used in proof of Theorem 2.2 in

Hertog[7]

Lemma 2.15

Let $0 < \mu$. Then we have

$$\frac{d\phi_\mu(x(\mu))}{d\mu} = -\frac{c^T x(\mu)}{\mu^2} = \frac{g(x(\mu))^T x(\mu)}{\mu}.$$

Proof

Denoting the derivative of $x(\mu)$ with respect to μ as $x'(\mu)$, we may write

$$\frac{d\phi_\mu(x(\mu))}{d\mu} = \frac{d}{d\mu}\left(\frac{c^T x(\mu)}{\mu} + \phi(x(\mu))\right) = -\frac{c^T x(\mu)}{\mu^2} + \frac{c^T x'(\mu)}{\mu} + g(x(\mu))^T x'(\mu).$$

The definition of $x(\mu)$, as minimizer of $\phi_\mu(x)$, implies

$$g(x(\mu)) = -\frac{c}{\mu}.$$

Hence we write

$$\frac{c^T x'(\mu)}{\mu} + g(x(\mu))^T x'(\mu) = 0$$

whence

$$\frac{d\phi_\mu(x(\mu))}{d\mu} = -\frac{c^T x(\mu)}{\mu^2},$$

which implies the lemma.

Lemma 2.16

Let $x \in D, \lambda_\mu(x) \le \tau = \frac{1}{3\kappa}$ and $\mu^+ = (1-\theta)\mu$. Then we have

$$\phi_{\mu^+}(x) - \phi_{\mu^+}(x(\mu^+)) \le \frac{1}{13\kappa^2} + \frac{\theta\nu}{1-\theta}.$$

Proof

Fixing $x \in D$, we define

$$\varphi(\mu) = \phi_\mu(x) - \phi_\mu(x(\mu)).$$

Then we need to find an upper bound for $\varphi(\mu^+)$. According to the Mean Value Theorem there exists a $\hat{\mu} \in (\mu^+, \mu)$ such that

$$\varphi(\mu^+) = \varphi(\mu) + \varphi'(\hat{\mu})(\mu^+ - \mu). \qquad (2.27)$$

Let us consider first $\varphi'(\mu)$. We have

$$\varphi'(\mu) = \frac{d\phi_\mu(x)}{d\mu} - \frac{d\phi_\mu(x(\mu))}{d\mu} = \frac{-c^T x}{\mu^2} - \frac{d\phi_\mu(x(\mu))}{d\mu}. \tag{2.28}$$

Using Lemma 2.15 we get

$$\varphi'(\mu) = \frac{-c^T x}{\mu^2} + \frac{c^T x(\mu)}{\mu^2} = \frac{c^T(x(\mu) - x)}{\mu^2} = \frac{g(x(\mu))^T(x - x(\mu))}{\mu}.$$

Now applying Lemma 2.12 twice, with $d = x - x(\mu)$ and $d = x(\mu) - x$ respectively, we obtain

$$|\varphi'(\mu)| \leq \frac{\nu}{\mu}.$$

Hence, since $\hat{\mu} \in (\mu^+, \mu)$, we get

$$|\varphi'(\hat{\mu})| \leq \frac{\nu}{\mu^+}.$$

Substitution into (2.27) yields

$$\varphi(\mu^+) \leq \varphi(\mu) + \frac{\nu}{\mu^+}(\mu - \mu^+) = \varphi(\mu) + \frac{\theta\nu}{1-\theta}.$$

In other words,

$$\varphi_{\mu^+}(x) - \varphi_{\mu^+}(x(\mu^+)) \leq \phi_\mu(x) - \phi_\mu(x(\mu)) + \frac{\theta\nu}{1-\theta}.$$

Since $\lambda_\mu(x) \leq \tau = \frac{1}{3\kappa}$, we derive from Theorem 2.7 that

$$\phi_\mu(x) - \phi_\mu(x(\mu)) \leq \frac{1}{\kappa^2}\omega(-\frac{1}{3}) = \frac{0.0721318}{\kappa^2} < \frac{1}{13\kappa^2}.$$

Hence the lemma follows.

Theorem 2.9

The algorithm with damped Newton steps requires not more than

$$\left\lceil \frac{22\kappa^2}{\theta}\left(\frac{1}{13\kappa^2} + \frac{\theta\nu}{1-\theta}\right)\log\frac{\nu\mu^0}{\varepsilon} \right\rceil.$$

iterations. The output is a point $x \in D$ such that

$$c^T x \leq c^T x^* + \varepsilon\left(1 + \frac{1 + 3\kappa\sqrt{\nu}}{6\kappa^2\nu}\right),$$

where x^* denotes an optimal solution of (P).

Proof

Since each inner iteration decreases the value of ϕ_μ with at least $\frac{1}{22\kappa^2}$, an immediate consequence of Lemma 2.16 that the number of inner iteration between two successive μ-updates does not exceed the number

$$22\kappa^2\left(\frac{1}{13\kappa^2}+\frac{\theta\nu}{1-\theta}\right).$$

Using Lemma 2.14, the iteration bound in the theorem follows.
The last statement in the theorem follows from Lemma 2.13. At termination of the algorithm we have $3\kappa\lambda_\mu(x)<1$ and $\nu\mu\le\varepsilon$. Hence, denoting $\lambda=\lambda_\mu(x)$, Lemma 2.13 implies

$$\begin{aligned} c^Tx &\le c^Tx^*+\nu\mu\left(1+\frac{\lambda(\lambda+\sqrt{\nu})}{\nu(1-\kappa\lambda)}\right)\\ &\le c^Tx^*+\varepsilon\left(1+\frac{\frac{1}{3\kappa}(\frac{1}{3\kappa}+\sqrt{\nu})}{\nu(\frac{2}{3})}\right)\\ &\le c^Tx^*+\varepsilon\left(1+\frac{1+3\kappa\sqrt{\nu}}{6\kappa^2\nu}\right). \end{aligned}$$

This completes the proof.

It is interesting to compare the iteration bounds that we obtained for full-Newton and damped-Newton steps. When initialized with the same $x^0\in D$ and $\mu^0>0$ these bounds are given by

$$\left\lceil\frac{9+36\kappa\sqrt{\nu}}{5}\log\frac{\nu\mu^0}{\varepsilon}\right\rceil$$

and

$$\left\lceil\frac{22\kappa^2}{\theta}\left(\frac{1}{13\kappa^2}+\frac{\theta\nu}{1-\theta}\right)\log\frac{\nu\mu^0}{\varepsilon}\right\rceil=\left\lceil\left(\frac{22}{13\theta}+\frac{22\kappa^2\nu}{1-\theta}\right)\log\frac{\nu\mu^0}{\varepsilon}\right\rceil,$$

respectively. Neglecting the factor $\log\frac{\nu\mu^0}{\varepsilon}$, we see that the first bound is $O(\kappa\sqrt{\nu})$. On the other hand, when assuming $\theta=\Theta(1)$, the second bound is $O((\kappa\sqrt{\nu})^2)$.

This shows that from a theoretical point of view the full-Newton step method is more efficient than the damped-Newton step method. In practice, however, the converse holds. This phenomenon has become known as the *irony of interior-point methods* (e.g. Renegar[12],page 51).

Also note that in both cases the quantity $\kappa\sqrt{\nu}$ is solely responsible for the iteration bound, or

complexity of the algorithm. Following Glineur[6] we call this the *complexity value*.

2.2.6 Adding equality constraints

In many cases the vector x of variables in (P) not only has to belong to $\overline{D}$ but has also to satisfy a system of equality constraints. The problem then becomes

$$(P) \qquad \min\left\{c^T x : Ax = b, x \in \overline{D}\right\}.$$

We assume that A is a $m \times n$ matrix and $rank(A) = m$. This problem can be solved without much extra effort. The search direction has to be designed such that feasibility is maintained. Given a feasible x we take as search direction Δx the direction that minimizes the second order Taylor polynomial at x subject to the condition $A\Delta x = 0$. Thus we consider the problem

$$\min\left\{\phi_\mu(x) + \Delta x^T g_\mu(x) + \frac{1}{2}\Delta x^T H(x)\Delta x : A\Delta x = 0\right\}.$$

This gives rise to the system

$$H(x)\Delta x + g_\mu(x) = A^T y, \quad A\Delta x = 0,$$

whence, denoting $H(x)$ as H,

$$\Delta x = H^{-1}A^T(AH^{-1}A^T)^{-1}AH^{-1}g_\mu(x) - H^{-1}g_\mu(x)$$

or, equivalently,

$$H^{\frac{1}{2}}\Delta x = -\left(I - H^{-\frac{1}{2}}A^T(AH^{-1}A^T)^{-1}AH^{-\frac{1}{2}}\right)H^{-\frac{1}{2}}g_\mu(x) = -P_{AH^{-\frac{1}{2}}}H^{-\frac{1}{2}}g_\mu(x),$$

where $P_{AH^{-\frac{1}{2}}}$ denotes the orthogonal projection onto the null space of $AH^{-\frac{1}{2}}$. Note that if the system $Ax = b$ is void, i.e. $A = 0$ and $b = 0$, then Δx is just the 'old' direction.

Denoting the feasible region of (P) as $\overline{P}$ and its interior as P, one easily understands that the restriction ϕ_P of ϕ to P is a κ-self-concordant ν-barrier for P. Moreover, Δx as above, is precisely the Newton direction for ϕ_P at $x \in P$. Hence, essentially the same full-Newton step method and damped-Newton step method as before can be used to solve the above problem in polynomial time.

Chpater 3 Heuristic approach to MDPs based on the IPM

3.1 Introduction

Now the model and the algorithm which can be used to solve the MDPs have already been described. In this chapter we present how to get an optimal policy of the MDP model with the IPM. The main idea is to use Algorithm 2.4 with Δx described in section 2.2.6 to solve linear programming problems under both discounted rewards and average rewards, and to get a series stationary policies which converge to an optimal deterministic policy. Next we will consider some tests which may accelerate this process.

In this chapter, we will use the following example for a better description.

Example 3.1

$\alpha = \frac{1}{2}$; $S = \{1,2,3\}$, $A(1) = A(2) = A(3) = \{1,2,3\}$; $r_1(1) = 1, r_1(2) = 2, r_1(3) = 3$

$r_2(1) = 6, r_2(2) = 4, r_2(3) = 5$; $r_3(1) = 8, r_3(2) = 9, r_3(3) = 7$.

$p_{11}(1) = 1, p_{12}(1) = p_{13}(1) = 0$; $p_{11}(2) = 0, p_{12}(2) = 1, p_{13}(2) = 0$;

$p_{11}(3) = p_{12}(3) = 0, p_{13}(3) = 1$; $p_{21}(1) = 1, p_{22}(1) = p_{23}(1) = 0$;

$p_{21}(2) = 0, p_{22}(2) = 1, p_{23}(2) = 0$; $p_{21}(3) = p_{22}(3) = 0, p_{23}(3) = 1$;

$p_{31}(1) = 1, p_{32}(1) = p_{33}(1) = 0$; $p_{31}(2) = 0, p_{32}(2) = 1, p_{33}(2) = 0$;

$p_{31}(3) = p_{32}(3) = 0, p_{33}(3) = 1$.

3.2 Discounted rewards

In this section we consider linear programming for MDP with discounted rewards, which is basically to compute optimal solutions v^* and x^* of the dual pair of linear programs:

$$\min\{\sum_j \beta_j v_j \mid \sum_j \{\delta_{ij} - \alpha p_{ij}(a)\} v_j \geq r_i(a), \quad (i,a) \in S \times A\} \tag{1.17}$$

and

$$\max\left\{\sum_{(i,a)} r_i(a) x_i(a) \;\middle|\; \begin{array}{l} \sum_{(i,a)} \{\delta_{ij} - \alpha p_{ij}(a)\} x_i(a) = \beta_j, \quad j \in S \\ x_i(a) \geq 0, \quad (i,a) \in S \times A \end{array}\right\}. \tag{1.18}$$

We will use Algorithm 2.4 to get a dual optimal solution x^*; the primal solution v^* is

generated as by-product.

We notice from the linear constraints in (1.18) that for a fixed $j \in S$

$$\sum_a x_j(a) = \alpha \sum_{(i,a)} p_{ij}(a) x_i(a) + \beta_j > 0 .$$

We know there are only $|S|$ linear constraints in (1.18). That means in the extreme optimal solution of (1.18), for every state $i \in S$ there must be one $a^* \in A(i)$ s.t. $x_i(a^*) > 0$ and all other $a \in A(i) \setminus a^*$ satisfy $x_i(a) = 0$. Hence, using IPM, we will get a series of interior points convergent to an extreme optimal solution* of (1.18) which has the form described above.

3.2.1 Initial point

In order to start the Algorithm 2.4 we need an initial interior point which satisfies $\lambda_{\mu^0}(x^0) \le \tau$, $x^0 \in D$, $\mu^0 > 0$, $\tau = \frac{1}{3\kappa}$. The first thing we should notice is that we can use the inner loop of Algorithm 2.4 to get an interior point which satisfies $\lambda_{\mu^0}(x^0) \le \tau$ starting from any interior feasible point. Then finding the initial interior point is reduced to finding an interior feasible point $x^0 \in D$, in which

$$D := \left\{ x^0 \;\middle|\; \begin{array}{r} \sum_{(i,a)} \{\delta_{ij} - \alpha p_{ij}(a)\} x_i^0(a) = \beta_j, \quad j \in S \\ x_i^0(a) > 0, (i,a) \in S \times A \end{array} \right\}. \tag{3.1}$$

In general case, this can be a complicated problem. However, in MDP with discounted rewards, there is a property we can use to get an interior feasible point.

According to Theorem 1.12, the mapping (1.19) is a one-to-one mapping from the set of stationary policies onto the set of feasible solution of the dual program (1.18) with (1.20) as the inverse mapping. Hence we can get the interior feasible point x^0 with (1.19) using a special policy, which brings us the next theorem.

Theorem 3.1

Let $\beta > 0$ and π^∞ the stationary policy with

* Or the middle point of extreme optimal solutions, if there are several extreme optimal solutions with the same optimal value. We will describe this later.

$$\pi_{ia} = \frac{1}{|A(i)|}, \quad a \in A(i), \quad i \in S. \tag{3.2}$$

Then

$$x_i^{\pi}(a) = \{\beta^T \{I - \alpha P(\pi)\}^{-1}\}_i \cdot \pi_{ia}, (i,a) \in S \times A \tag{3.3}$$

is an interior feasible point in the feasible set of(1.18).

Proof

Theorem 1.12 proved

$$x_i^{\pi}(a) = \{\beta^T \{I - \alpha P(\pi)\}^{-1}\}_i \cdot \pi_{ia}, \ (i,a) \in S \times A$$

is a feasible point of (1.18). Then we only need to prove

$$x_i^{\pi}(a) > 0, \quad (i,a) \in S \times A.$$

In section 1.3.1, we proved

$$\{I - \alpha P(\pi)\}^{-1} = \sum_{t=1}^{\infty} \alpha^{t-1} P(\pi)^{t-1} \geq I.$$

Hence

$$x_i^{\pi}(a) \geq \beta^T \pi_{ia} > 0, \quad (\mathrm{i}, \mathrm{a}) \in \mathrm{S} \times \mathrm{A},$$

proving the theorem.

3.2.2 Computational performance

As we have mentioned in section 2.2.2:

$\phi(x) = -\sum_{i=1}^{n} \log x_i$ is a 1-self-concordant n-barrier function for $R_+^n = \{x \in R^n : x \geq 0)$;

Furthermore, we can also notice from Corollary 2.5 that $\phi(x)$ has neat second and third term derivatives. So we will use $\phi(x) = -\sum_{i=1}^{n} \log x_i$ as a barrier function in the IPM to solve the MDP with discounted rewards.

The next result is a theorem about the complexity.

Theorem 3.2

Given $\phi(x) = -\sum_{i=1}^{n} \log x_i$, the algorithm with damped Newton steps requires not more than

$$\left\lceil \frac{22}{\theta}\left(\frac{1}{13} + \frac{\theta n}{1-\theta}\right) \log \frac{n\mu^0}{\varepsilon} \right\rceil \tag{3.4}$$

iterations. The output is a point $x \in D$ such that

$$c^T x \le c^T x^* + \varepsilon\left(1 + \frac{1+3\sqrt{n}}{6n}\right),$$

where x^* denotes an optimal solution of (1.18).

Proof

Directly from Theorem 2.9

Remark

1. We can minimize the iteration bound

$$\left\lceil \frac{22}{\theta}\left(\frac{1}{13} + \frac{\theta n}{1-\theta}\right)\log\frac{n\mu^0}{\varepsilon} \right\rceil$$

by letting

$$\theta = \frac{-22 + \sqrt{286n}}{13n - 22},$$

but it's just a theoretical minimal bound, not very useful in practice. Although the damped Newton steps can make sure every step is feasible even if we use a very big θ, we should not let θ be too close to 1, because the inner loop will take more iterations to get a x such that $\lambda_\mu(x) \le \tau$. So in our code, we choose $\theta = 0.9$.

Summing up every row of the linear constraints in (1.18), we can get:

$$\sum_{(i,a)} x_i(a) = \frac{\sum_{j=1}^{|S|}\beta_j}{1-\alpha}.$$

As we know, $\sum_{j=1}^{|S|}\beta_j$ and $1-\alpha$ are both fixed, so $\sum_{(i,a)} x_i(a)$ is a fixed number.

To make parameters simple, we choose

$$\beta_i = \frac{1}{|S|}, \quad \mathrm{i} \in \mathrm{S}.$$

After the above preparation, we can start to solve MDP with discounted rewards (the MATLAB code is in Appendix I). We will try to solve Example 3.1.

First we calculate the initial interior point using Theorem 3.1 with $\beta_i = \frac{1}{|S|}, \quad \mathrm{i} \in \mathrm{S}$:

$x^0 = (0.2222 \quad 0.2222 \quad 0.2222 \quad 0.2222 \quad 0.2222 \quad 0.2222 \quad 0.2222 \quad 0.2222 \quad 0.2222)$

Starting from this point, Algorithm 2.4 brings us the following result:

k	($x_1(1)$	$x_1(2)$	$x_1(3)$	$x_2(1)$	$x_2(2)$	$x_2(3)$	$x_3(1)$	$x_3(2)$	$x_3(3)$)
0	0.2222	0.2222	0.2222	0.2222	0.2222	0.2222	0.2222	0.2222	0.2222
1	0.1499	0.2067	0.2636	0.2171	0.1964	0.2532	0.2067	0.2636	0.2429
2(ε =2)	0.1333	0.1828	0.2853	0.2097	0.1778	0.2735	0.1933	0.2947	0.2495
3	0.0622	0.1227	0.3495	0.1880	0.1245	0.3409	0.1521	0.3928	0.2673
4	0.0420	0.0739	0.3675	0.1524	0.0830	0.4281	0.1058	0.5034	0.2438
5	0.0313	0.0540	0.3566	0.1118	0.0597	0.5199	0.0741	0.6024	0.1902
6	0.0254	0.0440	0.3473	0.0841	0.0471	0.5813	0.0570	0.6674	0.1464
7(ε =1)	0.0226	0.0393	0.3439	0.0727	0.0415	0.6075	0.0498	0.6958	0.1269
8	0.0134	0.0243	0.3396	0.0448	0.0251	0.6715	0.0298	0.7668	0.0847
9	0.0084	0.0153	0.3368	0.0277	0.0156	0.7116	0.0184	0.8122	0.0539
10	0.0054	0.0100	0.3354	0.0178	0.0101	0.7352	0.0118	0.8393	0.0349
11	0.0037	0.0069	0.3347	0.0122	0.0069	0.7486	0.0081	0.8549	0.0239
12	0.0028	0.0052	0.3344	0.0091	0.0052	0.7558	0.0061	0.8633	0.0180
13(ε =0.1)	0.0024	0.0044	0.3342	0.0078	0.0045	0.7590	0.0052	0.8670	0.0154

We can transform these into stationary policies. Because of the one-to-one correspondence between the set of stationary policies and the set of feasible solutions of the dual program (1.18), we can use (1.20) to transfer this series of feasible solutions into policies:

Table 3.1

k	(π_{11}	π_{12}	π_{13}	π_{21}	π_{22}	π_{23}	π_{31}	π_{32}	π_{33})
0	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333
1	0.2417	0.3333	0.4250	0.3256	0.2946	0.3799	0.2898	0.3696	0.3406
2(ε =2)	0.2217	0.3040	0.4744	0.3172	0.2690	0.4138	0.2621	0.3996	0.3383
3	0.1165	0.2296	0.6539	0.2877	0.1906	0.5217	0.1872	0.4837	0.3291
4	0.0869	0.1529	0.7603	0.2297	0.1252	0.6452	0.1240	0.5901	0.2858
5	0.0708	0.1221	0.8071	0.1616	0.0864	0.7520	0.0855	0.6951	0.2195
6	0.0610	0.1055	0.8335	0.1181	0.0661	0.8158	0.0655	0.7664	0.1681
7(ε =1)	0.0558	0.0969	0.8473	0.1007	0.0575	0.8418	0.0570	0.7975	0.1455
8	0.0356	0.0643	0.9001	0.0604	0.0338	0.9058	0.0338	0.8701	0.0961
9	0.0232	0.0425	0.9343	0.0366	0.0207	0.9427	0.0208	0.9182	0.0610
10	0.0155	0.0284	0.9561	0.0233	0.0132	0.9635	0.0134	0.9472	0.0394
11	0.0108	0.0199	0.9693	0.0158	0.0090	0.9752	0.0091	0.9639	0.0270
12	0.0082	0.0151	0.9767	0.0119	0.0068	0.9814	0.0069	0.9728	0.0203
13(ε =0.1)	0.0070	0.0130	0.9800	0.0101	0.0058	0.9841	0.0059	0.9767	0.0174

It looks like we have a problem here. Because we only solve the dual linear program, there is no estimate about the value vector in the primal linear program, so we cannot give a statement as in the value iteration approach, saying "this policy is δ-optimal policy". Of course, we have another way to compute the value vector using $v^{\alpha}(\pi^{\infty}) = \{I - \alpha P(\pi)\}^{-1} r(\pi)$ with the policy

we get in the dual linear program. However, we will see from section 3.2.4 that we don't really need this value vector.

We can see from the 13th iteration π_{13}, π_{23}, π_{32} are so close to 1. We can even drop all other actions and guess $f(1)=3, f(2)=3, f(3)=2$ is the optimal deterministic policy. Then, anther question comes up: how to choose ε? Because, in the 7th iteration, π_{13}, π_{23}, π_{32} are already close to 1. it seems not necessary to go to ε=0.1. So, we need a new test to identify an optimal deterministic policy. Fortunately, Theorem 1.13 brings us an efficient test which we will show below in the next section.

3.2.3 Suboptimality test

The suboptimality test is described as:
If

$$y_i^f(a_i) > \min_a y_i^f(a) - \alpha(1-\alpha)^{-1}\{\min_i \min_a y_i^f(a) - \max_i \min_a y_i^f(a)\}, \qquad (3.5)$$

then action $a_i \in A(i)$ is suboptimal, where $y_i^f(a)$ is the dual slack variable.

Given an arbitrary stationary policy, we do the suboptimality test trying to find suboptimal actions. If only one action $a \in A(i)$ for each state is not suboptimal, then we can drop all other actions, and get the optimal deterministic policy.

The following table shows the result of the Algorithm 2.4 on example 3.1 with suboptimality test. Signal "1" means this action is suboptimal; "0" means this action is not suboptimal.

Example 3.1(continuous)

k	($x_1(1)$	$x_1(2)$	$x_1(3)$	$x_2(1)$	$x_2(2)$	$x_2(3)$	$x_3(1)$	$x_3(2)$	$x_3(3)$)
0	1	1	0	1	1	0	1	0	0
1	1	1	0	1	1	0	1	0	0
2(ε =2)	1	1	0	1	1	0	1	0	0
3	1	1	0	1	1	0	1	0	1

We get the optimal deterministic policy after 3 iterations. If we compare this result with Table 3.1, we get the following policy at the third iteration:

π_{ia} :	(π_{11}	π_{12}	π_{13}	π_{21}	π_{22}	π_{23}	π_{31}	π_{32}	π_{33})
	0.1165	0.2296	0.6539	0.2877	0.1906	0.5217	0.1872	0.4837	0.3291

Obviously, we cannot get any conclusion about the optimal deterministic policy here without

suboptimality test.

The performance of suboptimality test depends on the value of α. We can see this from the test (3.5). A bigger α will make fewer actions to be excluded with this inequality. But, when α is small, this test works really well.

The following is the result of the same example with $\alpha = 0.1$

k	(π_{11}	π_{12}	π_{13}	π_{21}	π_{22}	π_{23}	π_{31}	π_{32}	π_{33})
0	1	1	0	0	1	1	1	0	1

We even get the optimal deterministic policy at the initial point. It works really good in this problem. However, if we try $\alpha = 0.9$:

k	(π_{11}	π_{12}	π_{13}	π_{21}	π_{22}	π_{23}	π_{31}	π_{32}	π_{33})
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	1	0	0	0	0	0	0	0	0
10	1	0	0	0	0	0	0	0	0
11	1	0	0	0	0	0	0	0	0
12	1	1	0	1	1	0	1	0	0
13	1	1	0	1	1	0	1	0	0
14	1	1	0	1	1	0	1	0	0
15	1	1	0	1	1	0	1	0	0
16	1	1	0	1	1	0	1	0	0
17(ε =0.1)	1	1	0	1	1	0	1	0	0

The suboptimality test cannot bring us a optimal deterministic policy even when $\varepsilon = 0.1$. If we look at the approximate policy we get from Algorithm 2.4:

π_{ia}:	(π_{11}	π_{12}	π_{13}	π_{21}	π_{22}	π_{23}	π_{31}	π_{32}	π_{33})
	0.0054	0.0106	0.9840	0.0008	0.0008	0.9984	0.0008	0.9793	0.0199

It is very likely that $f(1) = 3,\ f(2) = 3,\ f(3) = 2$ is the optimal deterministic policy.

What's more, there is another situation which can not be solved by suboptimality test: multiple optimal solutions (MOS). Another way to express this is: there exist several optimal deterministic policies with the same value vector. In this situation, suboptimality test can never end up with an

optimal deterministic policy, because there are two actions $a_1, a_2 \in A(i)$ for some $i \in S$ and both generate an optimal policy.

So we need another test which works well for all α, also in the multiple optimal solutions situation.

3.2.4 Optimality equation test

We first take a look at the behavior of Algorithm 2.4 in MDP with discounted rewards. According to the statement at the beginning of section 3.2, IPM has different behavior in MOS case and non-MOS cases. In non-MOS case, there is only one optimal deterministic policy. Therefore, for every state i, we have one $\pi_{ia^*}, a^* \in A(i)$ which is very close to 1 and all other $\pi_{ia}, a \in A(i) \setminus a^*$ are close to 0. On the other hand, in MOS case, there are several optimal deterministic policies with the same optimal value vector. In this case, there are several $\pi_{ia}, a \in A(i)$ which are close to $\frac{1}{n_i}$, where n_i is the number of optimal actions in state i, and all other π_{ia} go to 0. Hence this gives us a new idea: once we get a policy from Algorithm 2.4, we make a new policy:

$$\pi_{ia}^* = \begin{cases} 1 & \pi_{ia} = \max\limits_a \{\pi_{ia}\}, \quad i \in S \\ 0 & \text{otherwise} \end{cases} \tag{3.6}$$

and check whether it is an optimal policy. We can do this by checking whether the value vector of this policy fulfills the optimality equation:

$$v_i^\alpha(\pi^*) = \max_{a \in A(i)} \{r_i(a) + \alpha \sum_j p_{ij}(a) v_j^\alpha(\pi^*)\}, \ i \in S \, . \tag{3.7}$$

If the answer is no, we will go several steps further in IPM, until the heuristic policy changes. Then we do the optimality equation test again.

Remark

Of course, we can make the new policy in another way: set up a threshold $d \in (0, \frac{1}{|A|})$. Then, for every state $i \in S$, we let every $\pi_{ia} : \pi_{ia} < d, \ a \in A(i)$ to be zero, and randomly pickup an action a^* from $\{a : \pi_{ia} \geq d\}$. Then we choose $f(i) = a^*$ in the new policy.

Here we treat the Example 3.1 with $\alpha = 0.9$. It is non-MOS.

k	opt?	(π_{11}	π_{12}	π_{13}	π_{21}	π_{22}	π_{23}	π_{31}	π_{32}	π_{33})
0	No	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
1	Yes	0	0	1	0	0	1	0	1	0

It's much better than the suboptimality test in this example.
(For the initial point, the policy is not in a form of a deterministic policy, because we start from a policy which gives every possible action equal possibilities for every state. Then they are all maximum in the initial point. However, this situation will be changed in following iterations.)

On the other hand, in a MOS case, we cannot get every possible optimal deterministic policy, but one optimal deterministic policy is enough in general.

We consider the next example which was obtained by modifying Example 3.1

Example 3.2

$\alpha = \frac{1}{2}$; $S = \{1,2,3\}$, $A(1) = A(2) = A(3) = \{1,2,3\}$; $r_1(1) = 1, r_1(2) = 2, r_1(3) = 3$

$r_2(1) = 6, r_2(2) = 4, r_2(3) = 9$; $r_3(1) = 9, r_3(2) = 9, r_3(3) = 9$.

$p_{11}(1) = 1, p_{12}(1) = p_{13}(1) = 0$; $p_{11}(2) = 0, p_{12}(2) = 1, p_{13}(2) = 0$;

$p_{11}(3) = p_{12}(3) = 0, p_{13}(3) = 1$; $p_{21}(1) = 1, p_{22}(1) = p_{23}(1) = 0$;

$p_{21}(2) = 0, p_{22}(2) = 1, p_{23}(2) = 0$; $p_{21}(3) = p_{22}(3) = 0, p_{23}(3) = 1$;

$p_{31}(1) = 1, p_{32}(1) = p_{33}(1) = 0$; $p_{31}(2) = 0, p_{32}(2) = 1, p_{33}(2) = 0$;

$p_{31}(3) = 0, p_{32}(3) = 1, p_{33}(3) = 0$.

It is not hard to find out

$$\pi^1 : f(1) = 3, f(2) = 3, f(3) = 2 \quad \text{and} \quad \pi^2 : f(1) = 3, f(2) = 3, f(3) = 3$$

are both optimal policies. The following table is the result we get.

k	Opt?	(π_{11}	π_{12}	π_{13}	π_{21}	π_{22}	π_{23}	π_{31}	π_{32}	π_{33})
0	No	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
1	Yes	0	0	1	0	0	1	0	0	1

We only get one of the optimal deterministic policies.

In Appendix C, we list the performance measure for this heuristic method in bigger MDP models, which have more than 10 states and 4 actions.

Because we start from a policy which gives every possible action equal possibility for every state, this heuristic approach can reveal the optimal set in the first several steps. As we can see from

Table 3.1, π_{13}, π_{23}, π_{32} start to increase in the first iteration and other π_{ia} start to decrease at the same time. So, at that time we can already see the moving trend of the IPM. Hence, Algorithm 2.4 with optimality equation test has advantage against suboptimality test when α is close to 1, and also in the MOS case which we showed above.

3.3 Average rewards

As we can see from the section 1.4.6, the linear programming approach to MDP with average rewards is to compute the optimal solution (v^*, u^*) and (x^*, y^*) of the dual pair of linear programs:

$$\min\left\{\sum_j \beta_j v_j \;\middle|\; \begin{array}{ll} \sum_j \{\delta_{ij} - p_{ij}(a)\} v_j \geq 0 & \text{for every } (i,a) \in S \times A \\ v_i + \sum_j \{\delta_{ij} - p_{ij}(a)\} u_j \geq \mathrm{r_i}\ (\mathrm{a}) & \text{for every } (i,a) \in S \times A \end{array}\right\} \quad (1.36)$$

and

$$\max\left\{\sum_{(i,a)} r_i(a) x_i(a) \;\middle|\; \begin{array}{ll} \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\} x_i(a) = 0 & j \in \mathrm{S} \\ \sum_a x_j(a) + \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\} y_j(a) = \beta_j & j \in \mathrm{S} \\ x_i(a), y_i(a) \geq 0 & (\mathrm{i,a}) \in \mathrm{S} \times \mathrm{A} \end{array}\right\}. \quad (1.37)$$

The same as in section 3.2, we are going to use Algorithm 2.4 to get the dual optimal solution x^*; the primal solution v^* is generated as by-product.

However, this linear programming problem is much more complicated than the one we treated in section 3.2. But we have a certain way to solve it, and we will describe the solution in the following two sections.

3.3.1 Initial point

We now discuss how to start Algorithm 2.4 in linear programming approach for MDP with average rewards. The original idea is to generate an initial point with (1.44) in the same way we did in section 3.2.1. However it doesn't work in general.

If there exists a state $i \in S$ which is transient under any policy $f^\infty \in C$, then the corresponding part of x will always be zero. That means there is no strictly feasible point in the feasible set. Hence we cannot apply IPM in this case.

The following example shows this phenomenon:

Example 3.3

$S = \{1,2,3\}$, $A(1) = A(2) = A(3) = \{1,2\}$; $r_1(1) = 1, r_1(2) = 2$;

$r_2(1) = 6, r_2(2) = 4$; $r_3(1) = 8, r_3(2) = 9$.

$p_{11}(1) = 1, p_{12}(1) = p_{13}(1) = 0$; $p_{11}(2) = 0, p_{12}(2) = 1, p_{13}(2) = 0$;

$p_{21}(1) = 1, p_{22}(1) = p_{23}(1) = 0$; $p_{21}(2) = 0, p_{22}(2) = 1, p_{23}(2) = 0$;

$p_{31}(1) = 1, p_{32}(1) = p_{33}(1) = 0$; $p_{31}(2) = 0, p_{32}(2) = \frac{1}{2}, p_{33}(2) = \frac{1}{2}$.

We can see state 3 is transient under any policy. The first part of the dual linear programming problem is:

$$\begin{bmatrix} 0 & 1 & -1 & 0 & -1 & 0 \\ 0 & -1 & 1 & 0 & 0 & -\frac{1}{2} \\ 0 & 0 & 0 & 0 & 1 & \frac{1}{2} \end{bmatrix} x = 0.$$

Because $x \geq 0$, any feasible solution x of the above equation must have $x_3(1) = x_3(2) = 0$. So we don't have a strictly feasible interior point here, but this problem is solvable.

The problems (1.36) and (1.37) are of the following form

(P) $\min\{b^T y : Ay \geq c\}$,

and its dual:

(D) $\max\{c^T x : A^T x = b,\ x \geq 0\}$.

Assuming that (P) and (D) are both feasible, the optimal sets of (P) and (D) are denoted by P^* and D^*. We define the index sets B and N by

$$B := \{i : A_i y > c_i,\ y \in P^*\}.$$

$$N := \{i : x_i > 0,\ x \in D^*\}.$$

From the strong duality theorem, B and N form a partition of the full index set and the optimal values for both of these linear problems are the same. We denote the *optimal-value function* as $z(b,c)$.

Then we start to investigate the effect of changes in b and c on the optimal value function $z(b,c)$. We consider one-dimensional parameter perturbations of b and c. We assume that b and c are such that (P) and (D) are feasible. Then $z(b,c)$ is well defined and finite. It is convenient to introduce the following notations:

$$f(\lambda) := z(b + \lambda\Delta b, c), \qquad g(\mu) = z(b, c + \mu\Delta c).$$

It can be proved that the domains of f and g are closed intervals on the real line.

Theorem 3.4

$f(\lambda)$ is continuous, concave and piecewise linear.

Proof

By definition,

$$f(\lambda) = \min\{(b + \lambda\Delta b)^T y : y \in P\}.$$

For each λ the minimum value is attained at a central solution y^* of (P). Now y^* is uniquely determined by the optimal partition of (P) and $(b + \lambda\Delta b)^T y^*$ is constant for all optimal y^*. Associating one particular y^*, we obtain that

$$f(\lambda) = \min\{(b + \lambda\Delta b)^T y : y \in S\},$$

where S is a finite subset of P, For each $y \in S$, we have

$$(b + \lambda\Delta b)^T y = b^T y + \lambda\Delta b^T y,$$

which is a linear function of λ. This makes clear that $f(\lambda)$ is the minimum of a finite set of linear functions. It can be proved that the minimum of a finite set of linear functions is continuous, concave and piecewise linear.

Therefore, $f(\lambda)$ is continuous, concave and piecewise linear, proving the theorem.

In the same way, we get:

Theorem 3.5

$g(\mu)$ is continuous, convex and piecewise linear.

For any λ in the domain of f we denote the optimal set of (P_λ) by P_λ^* and the optimal set of (D_λ) by D_λ^*.

Theorem 3.6

If $f(\lambda)$ is linear on the interval $[\lambda_1, \lambda_2]$, where $\lambda_1 < \lambda_2$, then the primal optimal set P_λ^* is constant (i.e. invariant) for $\lambda \in (\lambda_1, \lambda_2)$.

Proof

Let $\bar{\lambda} \in (\lambda_1, \lambda_2)$ be arbitrary and let $\bar{y} \in P_{\bar{\lambda}}^*$ be arbitrary as well. Since $\bar{y}$ is optimal for $(P_{\bar{\lambda}})$ we have

$$f(\bar{\lambda}) = b(\bar{\lambda})^T \bar{y} = b^T \bar{y} + \bar{\lambda} \Delta b^T \bar{y},$$

and, since $\bar{y}$ is feasible for all λ,

$$b(\lambda_1)^T \bar{y} = b^T \bar{y} + \lambda_1 \Delta b^T \bar{y} \le f(\lambda_1), \qquad b(\lambda_2)^T \bar{y} = b^T \bar{y} + \lambda_2 \Delta b^T \bar{y} \le f(\lambda_2).$$

Hence we find

$$f(\lambda_1) - f(\bar{\lambda}) \ge (\lambda_1 - \bar{\lambda}) \Delta b^T \bar{y}, \qquad f(\lambda_2) - f(\bar{\lambda}) \ge (\lambda_2 - \bar{\lambda}) \Delta b^T \bar{y}.$$

The linearity of f on $[\lambda_1, \lambda_2]$ implies

$$\frac{f(\bar{\lambda}) - f(\lambda_1)}{\bar{\lambda} - \lambda_1} = \frac{f(\lambda_2) - f(\bar{\lambda})}{\lambda_2 - \bar{\lambda}}.$$

Now using that $\lambda_2 - \bar{\lambda} > 0$ and $\bar{\lambda} - \lambda_1 > 0$ we obtain

$$\Delta b^T \bar{y} \le \frac{f(\lambda_2) - f(\bar{\lambda})}{\lambda_2 - \bar{\lambda}} = \frac{f(\bar{\lambda}) - f(\lambda_1)}{\bar{\lambda} - \lambda_1} \le \Delta b^T \bar{y}.$$

Hence, the last two inequalities are equalities, and the slope of f on the closed interval $[\lambda_1, \lambda_2]$ is just $\Delta b^T \bar{y}$. This means that the derivative of f with respect to λ on the open interval (λ_1, λ_2) satisfies

$$f'(\bar{\lambda}) = \Delta b^T \bar{y}, \ \forall \lambda \in (\lambda_1, \lambda_2),$$

or equivalently,

$$f(\lambda) = b^T \bar{y} + \lambda \Delta b^T \bar{y} = b(\lambda)^T \bar{y}, \ \forall \lambda \in (\lambda_1, \lambda_2).$$

We conclude that $\bar{y}$ is optimal for any (P_λ) with $\lambda \in (\lambda_1, \lambda_2)$. Since $\bar{y}$ was arbitrary in P_λ^*, it follows that

$$P_{\bar{\lambda}}^* \subseteq P_\lambda^*, \ \forall \lambda \in (\lambda_1, \lambda_2).$$

Since $\bar{\lambda}$ was arbitrary in the open interval (λ_1, λ_2), the above argument applies to any $\tilde{\lambda} \in (\lambda_1, \lambda_2)$; so we also have

$$P^*_{\tilde{\lambda}} \subseteq P^*_{\lambda},\ \forall \lambda \in (\lambda_1, \lambda_2).$$

We may conclude that $P^*_{\bar{\lambda}} \subseteq P^*_{\tilde{\lambda}}$ and $P^*_{\tilde{\lambda}} \subseteq P^*_{\bar{\lambda}}$, which gives $P^*_{\bar{\lambda}} = P^*_{\tilde{\lambda}}$. The theorem follows,

In the same way we can also prove the following theorem.

Theorem 3.7

If $g(\mu)$ is linear on the interval $[\mu_1, \mu_2]$, where $\mu_1 < \mu_2$, then the dual optimal set D^*_{μ} is constant (i.e. invariant) for $\mu \in (\mu_1, \mu_2)$.

Theorem 3.6 gives us an idea to deal with the case that we don't have strictly feasible interior point. We start from the same policy we used in section 3.2.1, and put this policy in (1.44):

$$\begin{cases} x_i^{\pi}(a) = \{\sum_j \beta_j \{P^*(\pi)\}_{ji}\} \cdot \pi_{ia} \\ y_i^{\pi}(a) = \{\sum_j \beta_j \{D(\pi)\}_{ji} + \sum_j \gamma_j \{P^*(\pi)\}_{ji}\} \cdot \pi_{ia} \end{cases}$$

to get a feasible point of (1.37). Of course, this point may not be an interior point of the feasible set. Then we modify the original problem by adding

$$\Delta b = A^T \Delta x \qquad \text{in which} \quad \mathrm{x} + \Delta \mathrm{x} > 0$$

to b, and make sure Δx is small enough compared to x, so that the primal optimal set will not be changed. Hence, from the modified problem we can get an optimal policy which is also optimal for the original problem.

Remark

1) About the choice of Δx. If we sum every row of the linear constraints of (1.37), we can see

$$\sum_{(i,a)} x_i(a) = \sum_j \beta_j .$$

Hence, normally, we can take a Δx related to $\sum_j \beta_j$. In our code, we just choose

$$\Delta x_i(a) = \begin{cases} 0 & x_i(a) \neq 0 \\ \dfrac{\sum_j \beta_j}{|S \times A|^3} & x_i(a) = 0 \end{cases}. \tag{3.8}$$

2) About the y part of the initial point. For i belongs to a communicating set, we can always choose γ to make sure $y_i(a) > 0$. On the other hand, if i belongs to a transient set, $\sum_j \gamma_j \{P^*(\pi)\}_{ji}\} \cdot \pi_{ia}$ is always zero. However, the corresponding part of the transient set in $D(\pi)$ is $(I - Q)^{-1}$, and it is bigger than I. Hence we can conclude that the y part of the initial point is strictly positive.

3.3.2 Computational performance

Basically, we have the same approach as in section 3.2.2. We try to use Algorithm 2.4 to solve the LP problem (1.37) and to get an optimal solution, but the problem is not that easy.

We can see from (1.37) that there is no y in the objective function $\sum_{(i,a)} r_i(a)x_i(a)$. That means if we have an optimal solution (x^*, y^*) for (1.37) and a Δy, s.t.

$$\sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}\Delta y_j(a) = 0,$$

then we can get unbounded optimal solutions in the feasible set with the form

$$(x^*, y^* + M\Delta y)$$

in which $M \in R,\ y^* + M\Delta y \geq 0$.

This may not be a problem in the simplex method, because the simplex method moves from one extreme feasible point to another, but it may cause the IPM to fail. Even if IPM can end up with an optimal solution, it can be an interior point in the middle of the feasible set, not close to any extreme optimal solution. Then we cannot apply Theorem 1.26 to get an optimal policy.

Fortunately, Theorem 3.7 offers us a good way to overcome this disadvantage. What we do is adding a proper penalty on y to the objective function and transforms it to:

$$\sum_{(i,a)} r_i(a)x_i(a) - \delta \sum_{(i,a)} y_i(a). \tag{3.9}$$

Here the "proper" means δ is small enough to make sure the new LP problem has the same optimal set as the original problem, but not too small that the penalty doesn't really work. Because if δ is almost zero, y can still be very big and the optimal solution we get is not close to the extreme optimal solutions enough. In our code, we just let $\delta = 1$, and it works fine.

Now, we are fully prepared, and we can start to solve MDP problem with average rewards. The following is the result of Example 3.1 (because of the limit of space, we just list only a few iterations here). Here we choose $\delta = 1$.

	$(x_1(1)$	$x_1(2)$	$x_1(3)$	$x_2(1)$	$x_2(2)$	$x_2(3)$	$x_3(1)$	$x_3(2)$	$x_3(3)$
	$y_1(1)$	$y_1(2)$	$y_1(3)$	$y_2(1)$	$y_2(2)$	$y_2(3)$	$y_3(1)$	$y_3(2)$	$y_3(3))$
Initial	(0.1111	0.1111	0.1111	0.1111	0.1111	0.1111	0.1111	0.1111	0.1111
	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333)
	⋮								
iteration	14								
$\varepsilon = 2$	(0.0176	0.0250	0.0778	0.0527	0.0325	0.2544	0.0500	0.2822	0.2079
	0.1048	0.1446	0.2186	0.0819	0.1048	0.1401	0.0684	0.0837	0.1048)

$\vdots$

iteration 23

$\varepsilon = 1$ (0.0020 0.0028 0.0129 0.0079 0.0035 0.3994 0.0077 0.4046 0.1592
0.0105 0.0847 0.2418 0.0056 0.0105 0.0113 0.0053 0.0097 0.0105)

$\vdots$

iteration 32

$\varepsilon = 0.1$ (0.0002 0.0003 0.0014 0.0008 0.0004 0.4172 0.0008 0.4178 0.1611
0.0011 0.0856 0.2469 0.0005 0.0011 0.0011 0.0005 0.0010 0.0011)

$\vdots$

We can see in 32nd iterations, we are very close the extreme solution:

(0 0 0 0 0 $\frac{5}{12}$ 0 $\frac{5}{12}$ $\frac{1}{6}$

0 $\frac{1}{12}$ $\frac{1}{4}$ 0 0 0 0 0 0).

From this, we can get the corresponding deterministic policy using (1.43):

$f(1) = 3$ (or $f(1) = 2$), $f(2) = 3$, $f(3) = 2$ (or $f(3) = 3$).

It is obviously that every combination of the above is an optimal deterministic policy..

Also we try to solve Example 3.3 which has no strictly feasible interior point. We start from point

(0.2500 0.2500 0.2500 0.2500 0 0
0.6667 0.6667 0.6111 0.6111 0.2222 0.2222).

Then we add $\Delta x = [0\ 0\ 0\ 0\ 0.0046\ 0.0046\ 0\ 0\ 0\ 0\ 0\ 0]$ to it, so the initial point will be

(0.2500 0.2500 0.2500 0.2500 0.0046 0.0046
0.6667 0.6667 0.6111 0.6111 0.2222 0.2222).

The linear constraints will be

$$\begin{bmatrix} 0 & 1 & -1 & 0 & -1 & 0 & & & & & & \\ 0 & -1 & 1 & 0 & 0 & -\frac{1}{2} & & & & & & \\ 0 & 0 & 0 & 0 & 1 & \frac{1}{2} & & & & & & \\ 1 & 1 & & & & & 0 & 1 & -1 & 0 & -1 & 0 \\ & & 1 & 1 & & & 0 & -1 & 1 & 0 & 0 & -\frac{1}{2} \\ & & & & 1 & 1 & 0 & 0 & 0 & 0 & 1 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -0.0046 \\ -0.0023 \\ 0.0069 \\ 0.3333 \\ 0.3333 \\ 0.3426 \end{bmatrix}.$$

We also put a penalty in the objective function, and it becomes

$$(1\ \ 2\ \ 6\ \ 4\ \ 8\ \ 9\ \ -1\ \ -1\ \ -1\ \ -1\ \ -1\ \ -1) \cdot \begin{pmatrix} x \\ y \end{pmatrix}.$$

Then we solve the modified linear programming problem:

($x_1(1)$ $x_1(2)$ $x_2(1)$ $x_2(2)$ $x_3(1)$ $x_3(2)$

$y_1(1)$ $y_1(2)$ $y_2(1)$ $y_2(2)$ $y_3(1)$ $y_3(2)$)

Initial	(0.2500	0.2500	0.2500	0.2500	0.0046	0.0046
	0.6667	0.6667	0.6111	0.6111	0.2222	0.2222)
	⋮					
iteration 7						
$\varepsilon = 2$	(0.0353	0.3769	0.3784	0.2078	0.0031	0.0077
	0.1060	0.2133	0.0703	0.1060	0.2220	0.2197)
	⋮					
iteration 13						
$\varepsilon = 1$	(0.0041	0.4832	0.4868	0.0222	0.0010	0.0119
	0.0105	0.1060	0.0055	0.0105	0.2544	0.1505)
	⋮					
iteration 19						
$\varepsilon = 0.1$	(0.0004	0.4942	0.4987	0.0021	0.0001	0.0137
	0.0011	0.0973	0.0005	0.0011	0.2581	0.1415)
	⋮					

The same as last example, in 19th iterations, we are very close to an extreme solution:

(0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0
0	$\frac{1}{12}$	0	0	$\frac{1}{4}$	$\frac{1}{6}$)

Then, we can get two corresponding deterministic policies using (1.43):

$$f(1)=2,\ \ f(2)=1,\ \ f(3)=1 \ \text{ or } \ f(1)=2,\ \ f(2)=1,\ \ f(3)=2.$$

Both of them are optimal policies.

3.3.3 Optimality equation test

Here we follow the same idea in section 3.2.4: based on the policy we get from the IPM, we make a new deterministic policy and check whether it is optimal. If it is not, we go several steps further in the IPM until the heuristic policy changes. However, in MDP with average rewards, the situation is more complicated.

There is no property like: for every $i \in S$, there exists an action $a \in A(i)$ such that in the extreme optimal solution $x_i(a) > 0$. Therefore we cannot use the same trick in this section. In (1.43) we have to find the set $S_x = \{i \in S \mid \sum_a x_i(a) > 0\}$ first, but in the IPM, we move inside the feasible set. That means every point we get from IPM is strictly bigger than zero. Hence, the first thing we shall do is to set up a threshold d, and set any $x_i(a)$ which are lower than this threshold to zero.

$$x_i^*(a)=\begin{cases} x_i(a) & x_i(a)\geq d \\ 0 & x_i(a)<d \end{cases},\quad (i,a)\in S\times A. \tag{3.10}$$

Now, it is possible for us to use (1.43) to get a policy.

Remark

About the choice of d. Because of the same reason as Δx, normally, we can take a d related to $\sum_j \beta_j$. We should also take Δx into account. The amount $\frac{\sum_j \beta_j}{|S\times A|^3}$ should always be smaller than d so that the optimal set of the original linear programming problem stay the same. In our code, we just choose

$$d=\frac{\sum_j \beta_j}{|S\times A|^2}. \tag{3.11}$$

Then, we face another problem often: there are much more possible optimal deterministic policies in average rewards case than in the discounted rewards case. We can see this from the stationary matrix $P^*(f)$ of an optimal policy f. Different policies can lead to the same stationary matrix $P^*(f)$, so they all have the same value vector. That means they are all optimal policies. We only consider deterministic policies here. The simplest way to get a deterministic policy is:

$$\pi_{ia}=\begin{cases} 1 & \text{if } i\in\{j\mid \sum_a x_j(a)>0\},\ x_i^*(a)=\max\limits_a x_i^*(a) \\ 1 & \text{if } i\in\{j\mid \sum_a x_j(a)=0\},\ y_i(a)=\max\limits_a y_i(a), \\ 0 & \text{otherwise} \end{cases}\quad (i,a)\in S\times A. \tag{3.12}$$

In this heuristic way, we need a test to check whether it is an optimal policy. The next theorem introduces us a test.

For every $i\in S$ and $f^\infty\in C(D)$, the action set $B(i,f)$ is defined by

$$B(i,f)=\left\{ a\in A(i)\left| \begin{array}{c} \sum_j p_{ij}(a)\phi_j(f^\infty)>\phi_i(f^\infty) \text{ or} \\ \sum_j p_{ij}(a)\phi_j(f^\infty)=\phi_i(f^\infty) \text{ and } \mathrm{r}_i(a)+\sum_j p_{ij}(a)u_j^0(f^\infty)>\phi_i(f^\infty)+u_i^0(f) \end{array} \right. \right\}. \tag{3.13}$$

Theorem 3.8

If $B(i,f)=\phi$ for every $i\in S$, then f^∞ is an average optimal policy.

Proof

Since $B(i,f)=\phi$ for every $i\in S$, for any $h^\infty\in C(D)$, we have

$$\sum_j p_{ij}(h)\phi_j(f^\infty)\leq\phi_i(f^\infty)$$

and

$$r_i(a) + \sum_j p_{ij}(a) u_j^0(f^\infty) \le \phi_i(f^\infty) + u_j^0(f^\infty) \quad \text{if} \quad \sum_j p_{ij}(h)\phi_j(f^\infty) = \phi_i(f^\infty) \ .$$

Let $R = (h, f, f, ...)$. Then, $v^\alpha(R) = r(h) + \alpha P(h) v^\alpha(f^\infty)$ and, by Theorem 1.22,

$$\alpha v^\alpha(f^\infty) = \frac{\alpha}{1-\alpha}\phi(f^\infty) + u^0(f) + \varepsilon_1(\alpha) = \{1-(1-\alpha)\}\frac{\phi(f^\infty)}{1-\alpha} + u^0(f) + \varepsilon_1(\alpha)\cdot e$$
$$= \frac{\phi(f^\infty)}{1-\alpha} + u^0(f) - \phi(f^\infty) + \varepsilon_1(\alpha)\cdot e.$$

(in this proof $\varepsilon_k(\alpha)$ satisfies $\lim_{\alpha\uparrow 1}\varepsilon_k(\alpha) = 0$) implying

$$v^\alpha(R) = r(h) + P(h)\{\frac{\phi(f^\infty)}{1-\alpha} + u^0(f) - \phi(f^\infty) + \varepsilon_1(\alpha)\cdot e\}$$
$$= \frac{P(h)\phi(f^\infty)}{1-\alpha} + r(h) + P(h)u^0(f) - P(h)\phi(f^\infty) + \varepsilon_1(\alpha)\cdot e.$$

Since $v^\alpha(f^\infty) = \frac{\phi(f^\infty)}{1-\alpha} + u^0(f) + \varepsilon_2(\alpha)\cdot e$, we have

$$v^\alpha(f^\infty) - v^\alpha(R) = \frac{1}{1-\alpha}\{\phi(f^\infty) - P(h)\phi(f^\infty)\}$$
$$+ \{u^0(f) - r(h) - P(h)u^0(f) + P(h)\phi(f^\infty)\} + \varepsilon_3(\alpha)\cdot e.$$

Since $\phi(f^\infty) - P(h)\phi(f^\infty) \ge 0$ and, if $\{\phi(f^\infty) - P(h)\phi(f^\infty)\}_i = 0$,

$$\{u^0(f) - r(h) - P(h)u^0(f) + P(h)\phi(f^\infty)\}_i = \{u^0(f) - r(h) - P(h)u^0(f) + \phi(f^\infty)\}_i \ge 0\ ,$$

we obtain

$v^\alpha(f^\infty) - v^\alpha(R) \ge \varepsilon_3(\alpha)\cdot e$ for α sufficiently close to 1, i.e.

$$v^\alpha(f^\infty) \ge v^\alpha(R) + \varepsilon_3(\alpha)\cdot e = r(h) + \alpha P(h) v^\alpha(f^\infty) + \varepsilon_3(\alpha)\cdot e\ .$$

Hence,

$$\{I - \alpha P(h)\} v^\alpha(f^\infty) \ge r(h) + \alpha P(h) v^\alpha(f^\infty) + \varepsilon_3(\alpha)\cdot e\ .$$

Therefore,

$$v^\alpha(f^\infty) \ge \{I - \alpha P(h)\}^{-1}\{r(h) + \varepsilon_3(\alpha)\cdot e\} = v^\alpha(h^\infty) + \frac{\varepsilon_3(\alpha)}{1-\alpha}\cdot e\ .$$

From the Laurent expansion follows $\phi(f^\infty) \ge \phi(h^\infty)$, i.e. f^∞ is an average optimal policy.

From the above theorem, we get a way to judge whether a deterministic policy is average optimal.

And we try to use this test in Algorithm 2.4 to solve Example 3.1:

k	opt?	$(\pi_{11}$	π_{12}	π_{13}	π_{21}	π_{22}	π_{23}	π_{31}	π_{32}	$\pi_{33})$
0	no	(0.333	0.333	0.333	0.333	0.333	0.333	0.333	0.333	0.333)
1	yes	(0	0	1	0	0	1	0	1	0)

This test turned out to be extremely good in this example. We also try Example 3.3

k	opt?	$(\pi_{11}$	π_{12}	π_{21}	π_{22}	π_{31}	$\pi_{32})$
0	no	0.5	0.5	0.5	0.5	0.5	0.5
1	yes	0	1	1	0	0	1

It seems as the optimality equation test is unbelievably efficient, but it is reasonable. By the choice of initial point, we get a point which fulfills

$$x_i(a_1) = x_i(a_2) \quad \text{and} \quad y_i(a_1) = y_i(a_2)$$

for $\forall a_1, a_2 \in A(i),\ i \in S$. One the other hand, the optimal solution of (1.37) must have

one $x_i(a) > 0$ or $y_i(a) > 0$ for every state $i \in S$.

Take into account

$$\sum_{(i,a)} x_i(a) = \sum_j \beta_j$$

which is a constant, we can see why the first move of the IPM can show the clue of the optimal policy.

For performance measure of this heuristic method in big MDP models with averages rewards, we refer to Appendix C.

3.3.4 Blackwell optimal policy

There is another algorithm for the MDP with average rewards. As we see from the section 1.4.3, if α is close enough to 1, the optimal policy for discounted rewards is also optimal for average rewards.

We can compare the optimal policy for MDP with discounted rewards in the case $\alpha = 0.9$ with the optimal policy for MDP with average rewards, we can see they are actually the same.

Another question comes up: when is α is close enough to 1?
This is a parametric analysis problem of linear programming problem

$$\max\{c^T x : A^T x = b,\ x \geq 0\}$$

like we did at the beginning of section 3.3.1. Here, we don't consider how the optimal set changes with the change of b and c, but under the change of matrix A, which is a much harder problem.

However, in practice, if we choose $\alpha = 0.99$, we will nearly always get an optimal policy for MDP with average rewards from solving the MDP problem with discounted rewards.

In Appendix C, we list the performance measure for this heuristic method in discounted rewards with $\alpha = 0.99$. We can compare the result with average rewards.

Conclusion

Because of the special way to choose the starting point and construct the heuristic policy, in nearly all cases, we don't need to go very close to the optimal solution of the linear programming problem to get the optimal deterministic policy. As we can see from Appendix C, this heuristic approach to MDPs based on the IPM is very efficient. Even for a Linear programming problem with 160 variables (20 state, 8 actions), we are able to get the optimal deterministic policy for discounted rewards case in less than 30 iterations (on average out of 1000 random MDPs). Hence, in MDPs, this method apparently has an advantage against simplex method.

There is still something we need to do to complete our research in this method. We don't have a theoretical complexity bound for this method. It's not that easy to get complexity bound. However, simplex method doesn't have an exact complexity bound neither, and it's still a well accepted method.

What's more, we can also use this heuristic approach in value iteration. In value iteration, we need to calculate:

$$v^{n+1} = r_i(f_n) + \alpha \sum_j p_{ij}(f_n) v^n = \max_a \{r_i(a) + \alpha \sum_j p_{ij}(a) v^n\}.$$

We also have a guess of the policy f_n for the optimal policy.

Appendix A

Some technical lemmas

We start with a slightly generalized version of the well-known Cauchy-Schwarz inequality. The classical Cauchy-Schwarz inequality follows by taking $A = M = I$ in the next lemma (where I is the identity matrix).

Lemma A.1 (generalized Cauchy-Schwarz inequality).

If A, M are symmetric matrices with $|x^T Mx| \le x^T Ax$, $\forall x \in R^n$, then

$$(a^T Mb)^2 \le (a^T Aa)(b^T Ab), \quad \forall a, b \in R^n.$$

Proof

Note that $x^T Ax \ge 0$, $\forall x \in R^n$, so A is positive semi-definite. Without loss of generality, we assume that A is positive definite. Otherwise $A + \varepsilon I$ is positive definite for all $\varepsilon > 0$, and we take the limit as $\varepsilon \to 0$, with a and b are nonzero. It follows from

$$a^T Mb = \frac{1}{4}\left((a+b)^T M(a+b) - (a-b)^T M(a-b)\right)$$

that

$$\begin{aligned}
(a^T Mb)^2 &= \frac{1}{16}\left((a+b)^T M(a+b) - (a-b)^T M(a-b)\right)^2 \\
&\le \frac{1}{16}\left(\left|(a+b)^T M(a+b)\right| + \left|(a-b)^T M(a-b)\right|\right)^2 \\
&\le \frac{1}{16}\left((a+b)^T A(a+b) + (a-b)^T A(a-b)\right)^2 \\
&= \frac{1}{16}\left(2a^T Aa + 2b^T Ab\right)^2 \\
&= \frac{1}{4}\left(a^T Aa + b^T Ab\right)^2.
\end{aligned}$$

Let $\mu := \sqrt[4]{\dfrac{a^T Aa}{b^T Ab}}$.

When replacing a by $\dfrac{a}{\mu}$ and b by μb this implies

$$(a^T Mb)^2 = \left(\left(\frac{a}{\mu}\right)^T M(\mu b)\right)^2 \le \frac{1}{4}\left(\frac{1}{\mu^2} a^T Aa + \mu^2 b^T Ab\right)^2 = \left(a^T Aa\right)\left(b^T Ab\right),$$

which was to be shown.

The following lemma gives an estimate for the spectral radius of a symmetric homogeneous trilinear form. The proof is due to Jarre [8].

Lemma A.2 (Spectral Radius for Symmetric Trilinear Forms).

Let a symmetric homogeneous trilinear form $M : \mathrm{R}^n \times \mathrm{R}^n \times \mathrm{R}^n \to R$ be given by its coefficient matrix $M \in \mathrm{R}^{n \times n \times n}$. Let $A : \mathrm{R}^n \times \mathrm{R}^n \to R$ be a symmetric bilinear form, with matrix $A \in \mathrm{R}^{n \times n}$, and $\mu > 0$ a scalar such that

$$M[x,x,x]^2 \le \mu A[x,x]^3 = \mu \|x\|_A^6, \ \forall x \in R^n .$$

Then

$$|M[x,y,z]| \le \mu \|x\|_A \|y\|_A \|z\|_A, \ \forall x,y,z \in R^n .$$

Proof

Without loss of generality we assume that $\mu = 1$. Otherwise we replace A by $\sqrt[3]{\mu} A$. As in the proof of Lemma A.1 we assume that A is positive definite. Then, using the substitution

$$M[x,y,z] := M[A^{-\frac{1}{2}}x, A^{-\frac{1}{2}}y, A^{-\frac{1}{2}}z]$$

we can further assume that $A = I$ is the identity matrix and we need to show that

$$|M[x,y,z]| \le \mu \|x\|_2 \|y\|_2 \|z\|_2, \ \forall x,y,z \in R^n .$$

under the hypothesis

$$|M[x,x,x]| \le \mu \|x\|_2^3, \ \forall x \in R^n .$$

For $x \in R^n$ denote by M_x the (symmetric) matrix defined by

$$y^T M_x z := M_x[y,z] := M[x,y,z], \ \forall y,z \in R^n .$$

It is sufficient to show that

$$|M[x,y,y]| \le \mu \|x\|_2 \|y\|_2^2, \ \forall x,y \in R^n ,$$

because the remaining part follows by applying Lemma A.1, with $M = M_x$, for fixed x.

Define

$$\sigma := \max\{M[x,y,y] : \|x\|_2 = \|y\|_2 = 1\}$$

and let $\bar{x}$ and $\bar{y}$ represent a solution of this maximization problem. The necessary optimality condition for $\bar{x}$ and $\bar{y}$ imply that

$$\begin{pmatrix} M_{\bar{y}}\bar{y} \\ 2M_{\bar{y}}\bar{x} \end{pmatrix} = \alpha \begin{pmatrix} 2\bar{x} \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 2\bar{y} \end{pmatrix},$$

where α and β are the Lagrange multipliers. From this we deduce that $\alpha = \frac{\sigma}{2}$ and $\beta = \sigma$, by multiplying from the left with $\left(\bar{x}^T, 0\right)$ and $\left(0, \bar{y}^T\right)$, and thus we find

$$M_{\bar{y}}\bar{y} = \sigma\bar{x}, \quad 2M_{\bar{y}}\bar{x} = \sigma\bar{y},$$

which implies that $M_{\bar{y}}^2\bar{y} = \sigma^2\bar{y}$. Since $M_{\bar{y}}$ is symmetric, it follows that $\bar{y}$ is an eigenvector of $M_{\bar{y}}$ with the eigenvalue $\pm\sigma$, which gives that

$$\sigma = \left|\bar{y}^T M_{\bar{y}}\bar{y}\right| = M\left[\bar{y}, \bar{y}, \bar{y}\right].$$

This completes the proof.

Appendix B

Code I

This code is for MDPs with discounted rewards.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%% main %%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Damped_Newton_steps

[A, P, Q, alpha, beta, r, c, x, n, m, v, taw, epsilon, thet, mu] = initiation;
s=0;

[deltx, lambd] = calculate_lambd(A, x, c, v, mu);

[A,x,c]=analyse_x(A, P, Q, alpha, beta, r, c, x, n, m, v, taw, epsilon, thet, mu);

while (lambd > taw)
    s=s+1;
    x = x+deltx/(1+lambd);
    [A,x,c]=analyse_x(A, P, Q, alpha, beta, r, c, x, n, m, v, taw, epsilon, thet, mu);
    [deltx, lambd] = calculate_lambd(A, x, c, v, mu);
end;

disp('---------------------------------');

while (v*mu > epsilon)
    mu = (1-thet)*mu;

    [deltx, lambd] = calculate_lambd(A, x, c, v, mu);

    while (lambd>taw)
        s=s+1;
        x = x+deltx/(1+lambd);
        [A,x,c]=analyse_x(A, P, Q, alpha, beta, r, c, x, n, m, v, taw, epsilon, thet, mu);
        [deltx, lambd] = calculate_lambd(A, x, c, v, mu);
    end;
disp('---------------------------------');
end;
```

```
disp(sprintf('total number of iterations: %.6f', s));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%% subfunctions %%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%% initiation %%%%%%%%%%%%%%%%%%%%%%
function [A, P, Q, alpha, beta, r, c, x, n, m, v, taw, epsilon, thet, mu] = initiation;

alpha=0.5;
r=[1; 2; 3; 6; 4; 5; 8; 9; 7];
c=(-1)*r;
n=3;                                    % # of states %
m=3;                                     % # of actions %
v=n*m;
beta=ones(n,1)/n;

taw=1/3;
epsilon=0.1;
thet=0.9;
mu=1;

policy=[];                              % initial policy
for i=1:n
    temp=r((i-1)*m+1:i*m)>0;
    policy=[policy; temp/sum(temp)];
end;

P{1} = [1, 0, 0; 1, 0, 0; 1, 0, 0]; % Pij(a=1) %
P{2} = [0, 1, 0; 0, 1, 0; 0, 1, 0];
P{3} = [0, 0, 1; 0, 0, 1; 0, 0, 1];


for i=1:n                               % Pij(a=1,2,3)=>Qi=1,2,3 j(a)
    Q{i}=[];
    for j=1:m
        Q{i}=[Q{i};P{j}(i,:)];
    end;
end;

Ppolicy=[];
for i=1:n
    Ppolicy=[Ppolicy; (policy((i-1)*m+1:i*m))'*Q{i}];
```

```
end;

for i=1:m                                    % matrix A
    temp=eye(n,n)-alpha*P{i};
    for j=1:n
        A(:,i+(j-1)*m)=temp(j,:)';
    end;
end;

temp=beta'*(eye(n,n)-alpha*Ppolicy)^(-1);          % initial point %
x=[];
for i=1:n
    x=[x;temp(i)*policy((i-1)*m+1:i*m)];
end;




%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%% calculate deltx & lambd %%%%%%%%%%%%%%%%%%%%
function [deltx, lambd] = calculate_lambd(A, x, c, v, mu);

y = x;
B = A;
d = c;
position = (x~=0);
u = sum(position);

for i=v:-1:1
    if (position(i)~=1)
        y(i)=[];
        d(i)=[];
        B(:,i)=[];
    end;
end;

h = diag(y);
H = diag( y.*y);
delty = (H*B'*(B*H*B')^(-1)*B-eye(u)) * (H*(d/mu) - y);
hdelty = -(eye(u)-h*B'*(B*H*B')^(-1)*B*h)*(h*(d/mu) - ones(u,1));
lambd = (hdelty'*hdelty)^(0.5);

deltx=[];
for i=1:v
    if position(i)==1
```

```
        deltx = [deltx; delty(1)]; delty(1)=[];
    else
        deltx = [deltx; 0];
    end;
end;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%% analyse x %%%%%%%%%%%%%%%%%%%%%%%%
function [A,x,c]=analyse_x(A, P, Q, alpha, beta, r, c, x, n, m, v, taw, epsilon, thet, mu);

%%%%%%%%%%%% suboptimal test %%%%%%%%%%%%%%
policy=[];
for i=1:n
    temp=x(1+(i-1)*m:i*m);
    policy=[policy; temp/sum(temp)];
end;

Ppolicy=[];                              % P(pi)
for i=1:n
    Ppolicy=[Ppolicy; (policy((i-1)*m+1:i*m))'*Q{i}];
end;

rpolicy=[];                              % r(pi)
for i=1:n
    rpolicy=[rpolicy; r((i-1)*m+1:i*m)'*policy((i-1)*m+1:i*m)];
end;

vpolicy=(eye(n)-alpha*Ppolicy)^(-1)*rpolicy;                    % v(pi)

spolicy=[];                              % s(pi)
for i=1:m
    spolicy(i:m:v)=r(i:m:v)+alpha*P{i}*vpolicy-vpolicy;
end;
spolicy=spolicy';

Ux_x=[];                                 % Ux-x
for i=1:n
    Ux_x=[Ux_x; max(spolicy((i-1)*m+1:i*m))];
end;

subopt=[];
for i=1:n
    subopt=[subopt; spolicy((i-1)*m+1:i*m)<(Ux_x(i)-alpha/(1-alpha)*range(Ux_x))];
```

```
end;


%%%%%%%%%%% guess a deterministic policy %%%%%%%%%%%%%%
Dpolicy=[];                              % guess the deterministic policy: the action with the maxim
probability
for i=1:n
    temp=policy((i-1)*m+1:i*m)==max(policy((i-1)*m+1:i*m));
    Dpolicy=[Dpolicy; temp/sum(temp)];
end;

PDpolicy=[];
for i=1:n
    PDpolicy=[PDpolicy; (Dpolicy((i-1)*m+1:i*m))'*Q{i}];
end;

rDpolicy=[];                             % r(pi)
for i=1:n
    rDpolicy=[rDpolicy; r((i-1)*m+1:i*m)'*Dpolicy((i-1)*m+1:i*m)];
end;

vDpolicy=(eye(n)-alpha*PDpolicy)^(-1)*rDpolicy;

Dtest=[];
for i=1:n
    temp=r((i-1)*m+1:i*m)+alpha*Q{i}*vDpolicy;
    Dtest=[Dtest; max(temp)];
end;

disp(x');
disp(policy');
disp(subopt');
disp(Dpolicy');
disp(sprintf('------ %.6f', sum(vDpolicy==Dtest)==n));
```

Code II

The code for MDPs with average rewards is:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% main %%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Damped_Newton_steps

[A, P, Q, beta, r, c, x, n, m, w, taw, epsilon, thet, mu] = initiation;
s=0;

[deltx, lambd] = calculate_lambd(A, x, c, w, mu);

analyse_x(A, P, Q, beta, r, c, x, n, m, w, taw, epsilon, thet, mu);


while (lambd > taw)
    s=s+1; disp(s);
    x = x+deltx/(1+lambd);
    analyse_x(A, P, Q, beta, r, c, x, n, m, w, taw, epsilon, thet, mu);

    [deltx, lambd] = calculate_lambd(A, x, c, w, mu);
end;

disp('---------------------------------');

while (w*mu > epsilon)
    mu = (1-thet)*mu;

    [deltx, lambd] = calculate_lambd(A, x, c, w, mu);

    while (lambd>taw)
        s=s+1; disp(s);
        x = x+deltx/(1+lambd);
        analyse_x(A, P, Q, beta, r, c, x, n, m, w, taw, epsilon, thet, mu);

        [deltx, lambd] = calculate_lambd(A, x, c, w, mu);
    end;
disp('---------------------------------');
end;
```

```
disp(sprintf('total number of iterations: %.6f', s));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%% subfunctions %%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%% initiation %%%%%%%%%%%%%%%%%%%%%%%
function [A, P, Q, beta, r, c, x, n, m, v, taw, epsilon, thet, mu] = initiation;

r1=[1;2;3;6;4;5;8;9;7];
r=[r1;-1*ones(size(r1))];
c=(-1)*r;
n=3;                    % # of states %
m=3;                    % # of actions %
v=2*n*m;                % # of variables in linear programming
d=1/m;
beta=[zeros(n,1);ones(n,1)/n];

taw=1/3;
epsilon=1;
thet=0.9;
mu=1;

policy=[];              % initial policy
for i=1:n
    temp=r((i-1)*m+1:i*m)>0;
    policy=[policy; temp/sum(temp)];
end;

P{1} = [1 0 0; 1 0 0; 1 0 0]; % Pij(a=1) %
P{2} = [0 1 0; 0 1 0; 0 1 0];
P{3} = [0 0 1; 0 0 1; 0 0 1];

for i=1:n               % Pij(a=1,2,3)=>Qi=1,2,3 j(a)
    Q{i}=[];
    for j=1:m
        Q{i}=[Q{i};P{j}(i,:)'];
    end;
end;

Ppolicy=[];             % P(pi)
for i=1:n
    Ppolicy=[Ppolicy; (policy((i-1)*m+1:i*m))'*Q{i}];
end;
```

```
for i=1:m                                   % matrix A
    temp1=eye(n,n)-P{i};
    temp2=eye(n,n);
    for j=1:n
        tempA(:,i+(j-1)*m)=temp1(j,:)';
        tempB(:,i+(j-1)*m)=temp2(j,:)';
    end;
end;
A=[tempA, zeros(size(tempA));tempB, tempA];

sum=0;                                      % the stationary matrix of P(pi)
for i=1:10000
    sum=sum+Ppolicy^i;
end;
Pstar=sum/10000;

Z=(eye(n,n)-Ppolicy+Pstar)^(-1);        % the fundamental matrix

D=Z-Pstar;                                  % the deviation matrix

temp=beta(n+1:2*n)'*Pstar;              % initial point %
t1=[];
for i=1:n
    t1=[t1;temp(i)*policy((i-1)*m+1:i*m)];
end;
temp=beta(n+1:2*n)'*D+ones(1,n)*Pstar;
while sum(temp>0)<n
    temp=temp+ones(1,n)*Pstar;
end;
t2=[];
for i=1:n
    t2=[t2;temp(i)*policy((i-1)*m+1:i*m)];
end;
x=[t1;t2];

x=x+0.01*(x==0);
beta=A*x;

A(1:(2*n-rank(A)),:)=[];                % make sure matrix A is full rank
beta(1:(2*n-rank(A)))=[];
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%% calculate deltx & lambd %%%%%%%%%%%%%%%%%
function [deltx, lambd] = calculate_lambd(A, x, c, v, mu);

y = x;
B = A;
d = c;
position = (x~=0);
u = sum(position);

for i=v:-1:1
    if (position(i)~=1)
        y(i)=[];
        d(i)=[];
        B(:,i)=[];
    end;
end;

h = diag(y);
H = diag( y.*y);
delty = (H*B'*(B*H*B')^(-1)*B-eye(u)) * (H*(d/mu) - y);
hdelty = -(eye(u)-h*B'*(B*H*B')^(-1)*B*h)*(h*(d/mu) - ones(u,1));
lambd = (hdelty'*hdelty)^(0.5);

deltx=[];
for i=1:v
    if position(i)==1
        deltx = [deltx; delty(1)]; delty(1)=[];
    else
        deltx = [deltx; 0];
    end;
end;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%% analyse x %%%%%%%%%%%%%%%%%%%%%%%
function [A,x,c]=analyse_x(A, P, Q, beta, r, c, x, n, m, v, taw, epsilon, thet, mu);


%%%%%%%%%%%% guess a deterministic policy %%%%%%%%%%%%%
policy=[];                              % guess the deterministic policy
for i=1:n
    temp1=x((i-1)*m+1:i*m)>=(sum(beta)/(m*n)/10);
    if sum(temp1)~=0
```

```
            policy=[policy; x((i-1)*m+1:i*m)==max(x((i-1)*m+1:i*m))];
        else
            policy=[policy;
x((m*n+(i-1)*m+1):(m*n+i*m))==max(x((m*n+(i-1)*m+1):(m*n+i*m)))];
        end;
end;

Ppolicy=[];
for i=1:n
        Ppolicy=[Ppolicy; (policy((i-1)*m+1:i*m))'*Q{i}];
end;

rpolicy=[];                                 % r(pi)
for i=1:n
        rpolicy=[rpolicy; r((i-1)*m+1:i*m)'*policy((i-1)*m+1:i*m)];
end;

temp=eye(n,n);                                          % the stationary matrix of P(pi)
for i=1:10000-1
        temp=temp+Ppolicy^i;
end;
PpolicyStar=temp/10000;

Z=(eye(n,n)-Ppolicy+PpolicyStar)^(-1);        % the fundamental matrix

D=Z-PpolicyStar;                                        % the deviation matrix

v=PpolicyStar*rpolicy;
u=D*rpolicy;

vtest=[]; utest=[];
for i=1:n
        temp1=Q{i}*v;
        vtest=[vtest; max(temp1)];
        temp2=(r((i-1)*m+1:i*m)+Q{i}*u);
        utest=[utest; max(temp2)];
end;

disp(x');
disp(policy');
disp(sprintf('------ %.6f', sum([abs(v-vtest);abs(v+u-utest)]<1/10^2)==2*n));
```

Appendix C

In this section, we report our numerical results based on 1000 random MDPs, and list the average performance in discounted rewards (DR) with $\alpha = 0.5$ and $\alpha = 0.99$ (mostly the Blackwell policy), and also in average rewards (AR).

We generate MDPs in the following way:

1) Fix the size of the state space and the action space.

   Let $n = |S|$ and $m = |A|$.

2) Let every item of reward r be a random integer from $[1,100]$.

3) For every $a \in A$, we randomly choose k percent items from every row of the transition matrix and put a random number from $[0,1]$ in these positions. Then normalize every row of the transition matrix to make it a stochastic matrix.

The following table is the average number of iterations for the heuristic approach to get an optimal policy.

$k = 20$

n	m	DR with $\alpha = 0.5$	DR with $\alpha = 0.99$	AR
10	2	2.483	7.915	17.587
10	4	5.360	16.770	36.648
20	4	8.829	16.939	45.589
20	8	16.242	29.048	94.180

$k = 40$

n	m	DR with $\alpha = 0.5$	DR with $\alpha = 0.99$	AR
10	2	1.944	3.565	6.894
10	4	3.894	7.370	18.446
20	4	6.017	9.660	31.294
20	8	12.340	18.245	72.021

$k = 60$

n	m	DR with $\alpha = 0.5$	DR with $\alpha = 0.99$	AR
10	2	1.577	2.450	3.930
10	4	3.258	4.930	11.965
20	4	5.202	7.459	24.133
20	8	9.516	13.743	59.490

As we can see, to get a Blackwell optimal policy from letting $\alpha = 0.99$ in the discounted rewards case, costs much less time than to get an average optimal policy directly. In practice, if we want to get an average optimal policy, we just let $\alpha = 0.99$ in the discounted rewards case. However, there is no theory to guarantee what we get from this way is an average optimal policy. What's more, standard techniques of Policy iteration and Value iteration have numerical problem for $\alpha \approx 1$, but this approach works very well.

Bibliography

[1] Bauer, H.: *Probability theory and elements of measure theory*, Second English Edition, Academic Press, London, 1981.

[2] Bierth, K-J.: *An expected average reward criterion*, Stochastic Processes and Applications 26 (1987) 133-140.

[3] Chung, K.L.: *Markov chains with stationary transition probabilities*, Springer, 1960.

[4] Doob, J.L.: *Stochastic processes*, Wiley, 1953.

[5] Feller, W.: *An introduction to probability theory and its applications*, Volume I, third edition, Wiley, 1970.

[6] Glineur, F.: *Topics in Convex Optimization: Interior-Point Methods, Conic Duality and Approximations*. PhD thesis, Faculte Polytechnique de Mons, Mons, Belgium, 2001.

[7] Hertog, D.D.: *Interior Point Approach to Linear, Quadratic and Convex Programming*, Volume 277 of *Mathematics and its Applications*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994.

[8] Jarre, F.: *Interior-point methods via self-concordance or relative Lipschitz condition*. Fakultat fur Mathematik, Bayerischen Jelius-Maximilians-Universitat, Wurzburg, Deutschland, 1994. Habilitationsschrift.

[9] Kallenberg L.C.M: *Markov decision processes*, lecture note, University of Leiden, The Netherlands, Fall 2007.

[10] Nesterov, Y.E. and A.S. Nemirovskii. *Interior Point Polynomial Methods in Convex Programming: Theory and Algorithms*. SIAM Publications. SIAM, Philadelphia, USA, 1993.

[11] Powell, R.E. and S.M. Shah: *Summability theory and applications*, Van Nostrand Reinhold, London (1972).

[12] Renegar James. *A mathematical view of interior-point methods in convex optimization. MPS/SIAM Series on Optimization.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2001.

[13] Stoer, J. and R. Bulirsch: *Introduction to numerical analysis*, Springer, 1980.