Subject: Westlake University, Reinforce Learning, Lecture 2, Bellman Equation

Date: from December 20, 2024 to December 21, 2024

Contents

1 Lecture 2, Bellman Equation

Bilibili:Lecture 2, Bellman Equation

Outline

本节将重点介绍一个核心概念和一个重要工具，它们都是强化学习重要的基础内容.

A core concept: state value. 用于衡量一个 policy 的好坏，越好的 policy 对应的 state value 相对越大.

A fundamental tool: the **Bellman equation**. 用于分析 state value，描述所有的 state value 间的关系。解贝尔曼方程就可以获得这些 state values，这一过程也被称为 **policy evaluation**.

Contents:

1. ⋆ State value
2. ⋆ Bellman equation: Derivation → Matrix-vector form → Solve the state values
3. Action value

Recall: return

About return

在介绍 state value 前，我们首先回顾一个相似的概念：return. 因为 return 也可以用以衡量一个 policy 的好坏.

Q1: Why return is important?

A1: Return 可以评估一个 policy. 这是将我们对一个策略好坏的 直觉(intuition) 进行数学化的重要定量工具. 只有量化了一个 policy 我们才能不断改进策略.

Q2: How to calculate return?

A2:

1. By definition: 不断累和，即将此 policy 实施过程中所有 (discounted) reward 加和.
2. Bootstrapping[a]: 当前 state 的 return 依赖于其他 state 的 return，最后循环回到自身. 这样将所有的 state 组合起来就可以通过矩阵形式求解.(也就可以得到一般的 Bellman equation)

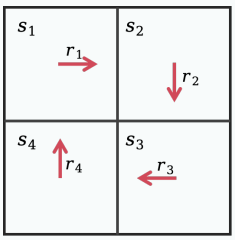

Figure 1: An illustrating example for computing reward

如图1所示，如果 by definition，就会得到：

$$\begin{aligned} v_1 &= r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots, \\ v_2 &= r_2 + \gamma r_3 + \gamma^2 r_4 + \cdots, \\ v_3 &= r_3 + \gamma r_4 + \gamma^2 r_1 + \cdots, \\ v_4 &= r_4 + \gamma r_1 + \gamma^2 r_2 + \cdots. \end{aligned} \tag{1}$$

其中 γ 为 discounted rate.

但是可以发现，实际上 v_1 会依赖于下一个 state 的 reward v_2，依此类推，就可以得到如下系统：

$$\begin{aligned} v_1 &= r_1 + \gamma\left(r_2 + \gamma r_3 + \ldots\right) = r_1 + \gamma v_2, \\ v_2 &= r_2 + \gamma\left(r_3 + \gamma r_4 + \ldots\right) = r_2 + \gamma v_3, \\ v_3 &= r_3 + \gamma\left(r_4 + \gamma r_1 + \ldots\right) = r_3 + \gamma v_4, \\ v_4 &= r_4 + \gamma\left(r_1 + \gamma r_2 + \ldots\right) = r_4 + \gamma v_1. \end{aligned} \tag{2}$$

虽然看似每个量间都相互关联构成循环，因此不可解，但是实际上如果写成如下线性的矩阵-向量方程(linear matrix-vector equation)就可以看出只需求逆矩阵即可：

$$\underbrace{\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix}}_{v} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix} + \begin{bmatrix} \gamma v_2 \\ \gamma v_3 \\ \gamma v_4 \\ \gamma v_1 \end{bmatrix} = \underbrace{\begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix}}_{r} + \gamma \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}}_{P} \underbrace{\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix}}_{v}, \tag{3a}$$

$$v = r + \gamma P v \Rightarrow v = (I - \gamma P)^{-1} r \tag{3b}$$

[a] Bootstrapping 来源于统计抽样，其思想是通过一些方法作用于一个系统就可以只利用该系统本身来获取其自身信息

State value

State Value

从上述关于 return 的回顾中可以看到 return 已经可以度量一个 policy 的好坏了，那么为什么还需要再提出 state value 的概念呢？– 原因在于 return 只能计算确定的 trajectory，无法融入随机性(stochastic)。例如下图2的例子，在初始点 s_1，此 policy 有 p_1 概率向右，有 p_2 概率向下($p_1 + p_2 = 1$)，因此一个简单的想法就是取均值（期望）：

$$\begin{aligned} v_1 &= p_1\left(r_{11} + \gamma r_2 + \gamma^2 r_3 + \gamma^3 r_3 + \cdots\right) + p_2\left(r_{12} + \gamma r_4 + \gamma^2 r_3 + \gamma^3 r_3 + \cdots\right) \\ &\xlongequal{p_1+p_2=1} p_1(r_{11} + \gamma r_2) + p_2(r_{22} + \gamma r_4) + r_3(\gamma^2 + \gamma^3 + \cdots) \end{aligned} \tag{4}$$

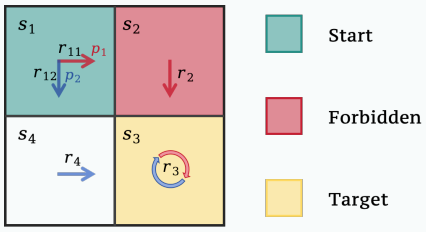

Figure 2: An illustrating example for computing state value

下面我们形式化地将随机性引入计算过程中，进而介绍 state value.

在 agent 每执行一次动作时，都会相应地到达下一个 state 并获得一个(只与当前 state 和 action 有关的) reward，也就得到了如下 Single-step process:

$$S_t \xrightarrow{A_t} R_{t+1}, S_{t+1} \tag{5}$$

其中 t 是为引入时序性，大写字母是表示随机变量(random variables)，引入随机性.

- A_t is governed by $\pi(A_t = a \mid S_t = s)$，即某 state 采取某 action 的概率
- $S_t \xrightarrow{A_t} R_{t+1}$ is governed by $p(R_{t+1} = r \mid S_t = s, A_t = a)$，表示第 t-state 采取某 action 后获得某 reward 的概率
- $S_t \xrightarrow{A_t} S_{t+1}$ is governed by $p(S_{t+1} = s' \mid S_t = s, A_t = a)$，表示第 t-state 采取某 action 后获得变成某 state 的概率

自然地就可以延伸至如下 Multi-step / state-action-reward (random) trajectory:

$$S_t \xrightarrow{A_t} R_{t+1}, S_{t+1} \xrightarrow{A_{t+1}} R_{t+2}, S_{t+2} \xrightarrow{A_{t+2}} R_{t+3}, \dots \tag{6}$$

如此可以计算 random discounted return:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots \tag{7}$$

⋆ **State value:** 定义为 G_t 的期望(expectation / expected alue / mean)，全称 state-value function:

$$v_\pi(s) \triangleq \mathbb{E}[G_t \mid S_t = s] \tag{8}$$

Note 1.
1. *state value* 是 *state* s 的函数，并且依赖于 *policy* π
2. 如果 $v_\pi(s)$ 更大，那么说明对于该 *state* 此 *policy* 较好
3. *state value* 不依赖于时间步 t，当 *policy* 给定后，各个 *state* 的 *state value* 也就确定了

State value vs. return

Q: What is the relationship between return and state value?

A:

1. state value 是所有情况下的 return 求和后 G_t 的期望；return 仅针对一个单独的 trajectory，而 state value 需要考虑全部可能的 trajectory.
2. 当没有随机性时，即只有一条确定性的 trajectory 时，return 的累和 G_t 就与 state value 相同.
3. 可以看出使用 state value 作为评判 policy 好坏较使用 return 是更好的.

Bellman equation

Bellman Equation – Elementwise form

下面开始介绍重要工具 Bellman equation. 总的来说， Bellman equation 是一组描述所有 state values 之间关系的方程。解出了 Bellman equation 也就可以得到相应 policy 的各 state values，进而可以评价该 policy.

对于一个 state-action-reward trajectory:

$$S_t \xrightarrow{A_t} R_{t+1}, S_{t+1} \xrightarrow{A_{t+1}} R_{t+2}, S_{t+2} \xrightarrow{A_{t+2}} R_{t+3}, \dots$$

Discounted return 可以写为:

$$\begin{aligned} G_t &= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3} + \cdots) \\ &= \underbrace{R_{t+1}}_{\text{immediate reward}} + \gamma \cdot \underbrace{G_{t+1}}_{\text{future reward}} \end{aligned} \tag{9}$$

进而根据定义，state s 的 state value 可以写为:

$$v_\pi(s) = \mathbb{E}[G_t|S_t = s] = \mathbb{E}[R_{t+1} + \gamma G_{t+1}|S_t = s] = \mathbb{E}[R_{t+1}|S_t = s] + \gamma\mathbb{E}[G_{t+1}|S_t = s] \tag{10}$$

其中第一项为 **mean of immediate rewards**，为:

$$\mathbb{E}[R_{t+1} \mid S_t = s] = \sum_{a\in\mathcal{A}(s)} \pi(a \mid s)\mathbb{E}[R_{t+1} \mid S_t = s, A_t = a] = \sum_{a\in\mathcal{A}(s)} \pi(a \mid s) \sum_{r\in\mathcal{R}(s,a)} p(r \mid s, a)\cdot r \tag{11}$$

其中第一个等号使用了双重期望定理(law of total expectation)，第二个等式是期望的定义.

第二项为 **mean of future reward**，为:

$$\begin{aligned} &\mathbb{E}[G_{t+1}|S_t = s] \xrightarrow[\text{expectation}]{\text{law of total}} \sum_{s'\in\mathcal{S}} \mathbb{E}[G_{t+1}|S_t = s, S_{t+1} = s']p(s'|s) \\ &\xrightarrow[\text{process property}]{\text{Markov decision}} \sum_{s'\in\mathcal{S}} \mathbb{E}[G_{t+1}|S_{t+1} = s']p(s'|s) \\ &= \sum_{s'\in\mathcal{S}} v_\pi(s')p(s'|s) \xrightarrow[\text{expectation}]{\text{law of total}} \sum_{s'\in\mathcal{S}} v_\pi(s') \sum_{a\in\mathcal{A}(s)} p(s'|s, a)\pi(a|s) \end{aligned} \tag{12}$$

将分解式(11)(12)代回式(10)中就可以得到如下的 **Bellman Equation**[a]:

$$\begin{aligned} v_\pi(s) &= \mathbb{E}[R_{t+1}|S_t = s] + \gamma\mathbb{E}[G_{t+1}|S_t = s] \\ &= \underbrace{\sum_{a\in\mathcal{A}(s)} \pi(a|s) \sum_{r\in\mathcal{R}(s,a)} p(r|s, a)r}_{\text{mean of immediate rewards}} + \underbrace{\gamma \sum_{a\in\mathcal{A}(s)} \pi(a|s) \sum_{s'\in\mathcal{S}} p(s'|s, a)v_\pi(s')}_{\text{mean of future rewards}} \\ &= \sum_{a\in\mathcal{A}(s)} \pi(a|s) \left[\sum_{r\in\mathcal{R}(s,a)} p(r|s, a)r + \gamma \sum_{s'\in\mathcal{S}} p(s'|s, a)v_\pi(s') \right], \quad \forall s \in \mathcal{S} \end{aligned} \tag{13}$$

[a] 这是 elementwise form 的 Bellman equation，下面会讲解 matrix-vector form 的 Bellman equation

Small summary

- Bellman equation 描述了不同 state 的 state-value function 间的关系
- Bellman equation 看似陷入循环、不可解，但其是一族方程，其包含了状态空间$\mathcal{S}$中全部 state，共 $|\mathcal{S}|$个方程，求解方法就是前面介绍的 Bootstrapping
- $\pi(a|s)$ 是一个给定的 policy，求解 Bellman equation 可以得到相应的 state value. 因此求解 Bellman equation 也被称为 policy evalution，即评价一个 policy 的好坏
- $p(r|s,a)$ 和 $p(s'|s,a)$ 表示系统的模型，称为 dynamic model / environment model，这个 model 有时已知有时未知，后续会介绍未知情形下如何进行 policy evaluation.
- 根据全概率公式(law of total probability)

$$p(s' \mid s,a) = \sum_{r\in\mathcal{R}} p(s', r \mid s, a),$$
$$p(r \mid s,a) = \sum_{s'\in\mathcal{S}} p(s', r \mid s, a).$$

  Bellman equation (13)也等价地可以写为：

$$v_\pi(s) = \sum_{a\in\mathcal{A}(s)} \pi(a|s) \sum_{s'\in\mathcal{S}} \sum_{r\in\mathcal{R}} p(s', r|s, a)[r + \gamma v_\pi(s')] \tag{14}$$

- 如果在某些问题中 reward 只与 s' 相关，那么上述 bellman equation 又可以进一步写为

$$v_\pi(s) = \sum_{a\in\mathcal{A}} \pi(a|s) \sum_{s'\in\mathcal{S}} p(s'|s, a) \left[r(s') + \gamma v_\pi(s')\right]$$

Bellman equation – Matrix-vector form

之前介绍的 Bellman equation (13) 实际上是其 elementwise form，共有 $|\mathcal{S}|$ 个公式，无法单独求解，但是将他们组合起来就可以得到一组线性方程，即求解一个线性方程组，也就是 matrix-vecotr form.

首先为书写方便，将式(13)改写为：

$$v_\pi(s) = r_\pi(s) + \gamma \sum_{s'} p_\pi(s'|s) v_\pi(s') \tag{15}$$

其中 $r_\pi(s)$ 为 average of immediate reward，$p_\pi(s'|s)$ 为在 policy π 下 state 由 s 转变为 s' 的概率：

$$r_\pi(s) \triangleq \sum_a \pi(a|s) \sum_r p(r|s,a) r$$
$$p_\pi(s'|s) \triangleq \sum_a \pi(a|s) p(s'|s,a)$$

对于n个状态的状态空间$\mathcal{S} = \{s_1, s_2, \cdots, s_n\}$，$n$ 个 Bellman equation 可写为：

$$v_\pi(s_i) = r_\pi(s_i) + \gamma \sum_{s_j} p_\pi(s_j \mid s_i) v_\pi(s_j), \quad i = 1, 2, \ldots, n \tag{16}$$

写成 matrix-vector form 就是：

$$\boldsymbol{v}_\pi = \boldsymbol{r}_\pi + \gamma \boldsymbol{P}_\pi \boldsymbol{v}_\pi \tag{17}$$

其中

- $\boldsymbol{v}_\pi = [v_\pi(s_1), \dots, v_\pi(s_n)]^T \in \mathbb{R}^n$
- $\boldsymbol{r}_\pi = [r_\pi(s_1), \dots, r_\pi(s_n)]^T \in \mathbb{R}^n$
- $\boldsymbol{P}_\pi \in \mathbb{R}^{n \times n}$，其中$[\boldsymbol{P}_\pi]_{ij} = p_\pi(s_j|s_i)$，为状态转移矩阵(state transition matrix)

写成矩阵形式就是

$$\underbrace{\begin{bmatrix} v_\pi(s_1) \\ v_\pi(s_2) \\ \vdots \\ v_\pi(s_n) \end{bmatrix}}_{\boldsymbol{v}_\pi} = \underbrace{\begin{bmatrix} r_\pi(s_1) \\ r_\pi(s_2) \\ \vdots \\ r_\pi(s_n) \end{bmatrix}}_{\boldsymbol{r}_\pi} + \gamma \underbrace{\begin{bmatrix} p_\pi(s_1|s_1) & \cdots & p_\pi(s_{n-1}|s_1) & p_\pi(s_4|s_1) \\ p_\pi(s_1|s_2) & \cdots & p_\pi(s_{n-1}|s_2) & p_\pi(s_4|s_2) \\ \vdots & \ddots & \vdots & \vdots \\ p_\pi(s_1|s_n) & \cdots & p_\pi(s_{n-1}|s_4) & p_\pi(s_n|s_n) \end{bmatrix}}_{\boldsymbol{P}_\pi} \underbrace{\begin{bmatrix} v_\pi(s_1) \\ v_\pi(s_2) \\ \vdots \\ v_\pi(s_n) \end{bmatrix}}_{\boldsymbol{v}_\pi}.$$

矩阵 $\boldsymbol{P}_\pi$ 有如下两个特点：

1. 所有元素非负. 因为显然概率 $0 < [\boldsymbol{P}_\pi]_{ij} = p_\pi(s_j|s_i) \leqslant 1$
2. 行和为 1，即 $\boldsymbol{P}_\pi \, \mathbf{1} = \mathbf{1}$，其中 $\mathbf{1} = [1, 1, \cdots, 1]^T$. 因为 $\sum_j p_\pi(s_j|s_i) = 1$

Bellman equation: Solve the state values

1 第一种求解方式就是直接求其闭式解(closed-form solution)，为

$$\boldsymbol{v}_\pi = (\boldsymbol{I} - \gamma \boldsymbol{P}_\pi)^{-1} \boldsymbol{r}_\pi \tag{18}$$

虽然此解数学形式直接，但是实际中大规模矩阵求逆矩阵很困难，仍然需要使用特殊的数值方法（且仍然难以求解）.

矩阵 $\boldsymbol{I} - \gamma \boldsymbol{P}_\pi$ 实际上有如下性质：

1. $\boldsymbol{I} - \gamma \boldsymbol{P}_\pi$ 可逆. 证明需要使用 Gershgorin circle theorem，详见附录A.
2. $(\boldsymbol{I} - \gamma \boldsymbol{P}_\pi)^{-1} \geqslant \boldsymbol{I}$. 因为 $(I - \gamma P_\pi)^{-1} = I + \gamma P_\pi + \gamma^2 P_\pi^2 + \cdots \geqslant I \geqslant 0$.
3. $(\boldsymbol{I} - \gamma \boldsymbol{P}_\pi)^{-1} \boldsymbol{r} \geqslant \boldsymbol{r} \geqslant 0, \forall \boldsymbol{r} \geqslant 0$. 因为直接利用性质 2 即可.

---

2 第二种方法是迭代法找其迭代解(iterative solution)：

$$\boldsymbol{v}_{k+1} = \boldsymbol{r}_\pi + \gamma \boldsymbol{P}_\pi \boldsymbol{v}_k, k = 0, 1, 2, \cdots \tag{19}$$

最终可以证明[a](证明过程详见附录A)

$$\boldsymbol{v}_k \xrightarrow{k \to \infty} \boldsymbol{v}_\pi = (\boldsymbol{I} - \gamma \boldsymbol{P}_\pi)^{-1} \boldsymbol{r}_\pi$$

Note 2. *1.* 计算 *state value* 可以评价一个 *policy* 好不好

2. 不同的 *policy* 也可以得到相同的 *state value*

[a]如果熟悉的话这其实就是一个不动点迭代的证明，使用Banach不动点定理

Action value

Action value

Action value 也是强化学习中的一个重要概念，放在这里才提出是因为其定义需要和 state value 相联系. Action value 是指在某 state 下采取相应的某 action 后能够获得的 average return，即

$$q_\pi(s,a) \triangleq \mathbb{E}[G_t|S_t=s,A_t=a] \tag{20}$$

- $q_\pi(s,a)$ 是 state-action pair (s,a) 的函数，而不仅依赖于 action
- $q_\pi(s,a)$ 依赖于 policy π

State value 与 Action value 辨析

- state value 是指一个 agent 从一个 state 出发能得到的 average return，定义为

$$v_\pi(s) = \mathbb{E}[G_t \mid S_t = s]$$

- action value 是指一个 agent 从一个 state 出发并且做出一个 action 后能得到的 average return，定义为

$$q_\pi(s,a) \triangleq \mathbb{E}[G_t|S_t=s,A_t=a]$$

由于

$$\underbrace{\mathbb{E}[G_t|S_t=s]}_{v_\pi(s)} = \sum_a \underbrace{\mathbb{E}[G_t|S_t=s,A_t=a]}_{q_\pi(s,a)}\pi(a|s)$$

$$= \sum_a \pi(a|s)\underbrace{\left[\sum_r p(r|s,a)r + \gamma\sum_{s'}p(s'|s,a)v_\pi(s')\right]}_{q_\pi(s,a)}$$

因此

$$v_\pi(s) = \sum_a \pi(a|s)q_\pi(s,a) \tag{21a}$$

$$q_\pi(s,a) = \sum_r p(r|s,a)r + \gamma\sum_{s'}p(s'|s,a)v_\pi(s') \tag{21b}$$

式(21a)和(21b)就像一个硬币的两面，式(21a)说明可以从 action value 获得 state value，式(21b)说明可以从 state value 获得 action value.

Note 3. 当一个确定性的 *policy* 中在某个 *state* 只有一个 *action* 时，其他的 *action* 产生的 *action value* 并不是*0*，而是也可以计算，此时的 *immediate reward* 一般为*0*，但是仍然有 *future reward*。这样就可以与这个确定的 *action* 比较看是否这个 *action* 是好的.

A Proof

Convergence of iteration solution of Bellman equation

$$\boldsymbol{v}_k \xrightarrow{k\to\infty} \boldsymbol{v}_\pi = (\boldsymbol{I} - \gamma \boldsymbol{P}_\pi)^{-1} \boldsymbol{r}_\pi$$

Proof. 首先定义残差 $\delta_k = v_k - v_\pi$，我们只需要证明 $\delta_k \to 0$ 即可.

将 $v_{k+1} = \delta_{k+1} + v_\pi$, $\delta_k = v_k - v_\pi$ 代入 $v_{k+1} = r_\pi + \gamma P_\pi v_k$ 中得到

$$\delta_{k+1} + v_\pi = r_\pi + \gamma P_\pi (\delta_k + v_\pi)$$

$$\begin{aligned}\delta_{k+1} &= -v_\pi + r_\pi + \gamma P_\pi \delta_k + \gamma P_\pi v_\pi \\ &= \gamma P_\pi \delta_k - v_\pi + (r_\pi + \gamma P_\pi v_\pi) \\ &\xrightarrow{v_\pi = r_\pi + \gamma P_\pi v_\pi} \gamma P_\pi \delta_k\end{aligned}$$

因此有

$$\delta_{k+1} = \gamma P_\pi \delta_k = \gamma^2 P_\pi^2 \delta_{k-1} = \cdots = \gamma^{k+1} P_\pi^{k+1} \delta_0.$$

已知 $0 < [P_\pi]_{ij} = p_\pi(s_j|s_i) \leqslant 1$，因此 $0 < [P_\pi^k]_{ij} \leqslant 1$，且 $\gamma < 1 \Rightarrow \gamma^k \to 0$，从而有

$$\delta_{k+1} = \gamma^{k+1} P_\pi^{k+1} \delta_0 \xrightarrow{k\to\infty} 0$$

□

Reversibility of closed-form solution

$$\boldsymbol{I} - \gamma \boldsymbol{P}_\pi \text{ 可逆}$$

Proof. 根据 Gershgorin circle theorem 1，矩阵 $I - \gamma P_\pi$ 的所有特征值至少落在一个 Gershgorin circle 中，其中第 i 个 Gershgorin circle 中心为 $[I - \gamma P_\pi]_{ii} = 1 - \gamma p_\pi(s_i|s_i)$，半径为 $R_i = \sum_{j\neq i} [I - \gamma P_\pi]_{ij} = -\sum_{j\neq i} \gamma p_\pi(s_j|s_i)$.

由于 $0 < [I - \gamma P_\pi]_{ii} = 1 - \gamma p_\pi(s_i|s_i) < 1$, $0 < \gamma < 1$，故

$$[I - \gamma P_\pi]_{ii} + R_i = 1 - \sum_j \gamma p_\pi(s_j|s_i) = 1 - \gamma > 0$$

因此所有的 Gershgorin circle 都落在零点左侧，自然地所有的特征值都严格大于 0，因此可逆. □

Theorem 1 (Gershgorin circle theorem)**.** 令 $A = [a_{ij}]$ 为 $n \times n$ 的复矩阵，R_i 为第 i 行除对角元外元素之和:

$$R_i \triangleq \sum_{j\neq i} |a_{ij}|$$

令 $D(a_{ii}, R_i) \subseteq \mathbb{C}$ 表示中心为 a_{ii}，半径为 R_i 的闭圆盘*(disc)*，称之为 *Gershgorin disc.*

那么， A 的所有特征值至少落在一个 *Gershgorin disc* $D(a_{ii}, R_i)$ 中.

First updated: December 19, 2024
Last updated: March 12, 2025

References