DPP：多样性算法

王树森

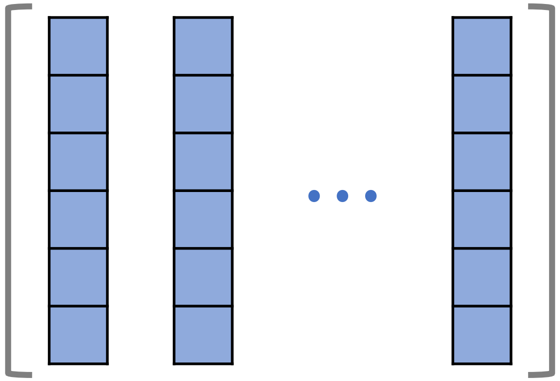
http://wangshusen.github.io/

多样性问题

- 精排给 n 个物品打分： $\text{reward}_1, \cdots, \text{reward}_n$ 。
- n 个物品的向量表征： $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_n \in \mathbb{R}^d$ 。
- 从 n 个物品中选出 k 个物品，组成集合 $\mathcal{S}$。
  - 价值大：分数之和 $\sum_{j \in \mathcal{S}} \text{reward}_j$ 越大越好。
  - 多样性好：$\mathcal{S}$ 中 k 个向量组成的超平形体 $\mathcal{P}(\mathcal{S})$ 的体积越大越好。

多样性问题

$\boldsymbol{V}_{\mathcal{S}} =$



$d \times k$

集合 $\mathcal{S}$ 中物品的向量

- 集合 $\mathcal{S}$ 中的 k 个物品的向量作为列，组成矩阵 $\boldsymbol{V}_{\mathcal{S}} \in \mathbb{R}^{d \times k}$ 。
- 以这 k 个向量作为边，组成超平形体 $\mathcal{P}(\mathcal{S})$ 。
- 体积 $\mathrm{vol}\big(\mathcal{P}(\mathcal{S})\big)$ 可以衡量 $\mathcal{S}$ 中物品的多样性。
- 设 $k \le d$，行列式与体积满足：

$$\det\left(\boldsymbol{V}_{\mathcal{S}}^T \boldsymbol{V}_{\mathcal{S}}\right) = \mathrm{vol}\big(\mathcal{P}(\mathcal{S})\big)^2 .$$

行列式点过程 (DPP)

- DPP 是一种传统的统计机器学习方法：

$$\underset{\mathcal{S}:\,|\mathcal{S}|=k}{\operatorname{argmax}} \ \log \det\left(\boldsymbol{V}_{\mathcal{S}}^{T}\, \boldsymbol{V}_{\mathcal{S}}\right).$$

- Hulu 的论文[1] 将 DPP 应用在推荐系统：

$$\underset{\mathcal{S}:\,|\mathcal{S}|=k}{\operatorname{argmax}} \ \theta \cdot \left(\textstyle\sum_{j\in\mathcal{S}} \text{reward}_j\right) + (1-\theta) \cdot \log \det\left(\boldsymbol{V}_{\mathcal{S}}^{T}\, \boldsymbol{V}_{\mathcal{S}}\right).$$

参考文献：

1. Chen et al. Fast greedy map inference for determinantal point process to improve recommendation diversity. In *NIPS*, 2018.

行列式点过程（DPP)

- DPP 应用在推荐系统：

$$\underset{\mathcal{S}:\,|\mathcal{S}|=k}{\operatorname{argmax}} \;\; \theta \cdot \left(\sum\nolimits_{j\in\mathcal{S}} \text{reward}_j\right) + (1-\theta)\cdot \log\det\left(\boldsymbol{V}_{\mathcal{S}}^T\,\boldsymbol{V}_{\mathcal{S}}\right).$$

$$= \boldsymbol{A}_{\mathcal{S}} \quad (k\times k)$$

- 设 $\boldsymbol{A}$ 为 $n\times n$ 的矩阵，它的 (i,j) 元素为 $a_{ij} = \boldsymbol{v}_i^T\boldsymbol{v}_j$。
- 给定向量 $\boldsymbol{v}_1,\cdots,\boldsymbol{v}_n \in \mathbb{R}^d$，需要 $O(n^2d)$ 时间计算 $\boldsymbol{A}$。
- $\boldsymbol{A}_{\mathcal{S}} = \boldsymbol{V}_{\mathcal{S}}^T\,\boldsymbol{V}_{\mathcal{S}}$ 为 $\boldsymbol{A}$ 的一个 $k\times k$ 子矩阵。如果 $i,j\in\mathcal{S}$，则 a_{ij} 是 $\boldsymbol{A}_{\mathcal{S}}$ 的一个元素。

行列式点过程（DPP）

- DPP 应用在推荐系统：

$$\underset{\mathcal{S}:\,|\mathcal{S}|=k}{\operatorname{argmax}}\ \ \theta\cdot\left(\sum_{j\in\mathcal{S}}\mathrm{reward}_j\right)+(1-\theta)\cdot\log\det(\boldsymbol{A}_{\mathcal{S}}).$$

- DPP 是个组合优化问题，从集合 $\{1,\cdots,n\}$ 中选出一个大小为 k 的子集 $\mathcal{S}$。
- 用 $\mathcal{S}$ 表示已选中的物品，用 $\mathcal{R}$ 表示未选中的物品，贪心算法求解：

$$\underset{i\in\mathcal{R}}{\operatorname{argmax}}\ \ \theta\cdot\mathrm{reward}_i\ +\ (1-\theta)\cdot\log\det\left(\boldsymbol{A}_{\mathcal{S}\cup\{i\}}\right).$$

求解 DPP

暴力算法

- 贪心算法求解：

$$\underset{i \in \mathcal{R}}{\operatorname{argmax}} \;\; \theta \cdot \text{reward}_i \; + \; (1-\theta) \; \boxed{\log \det\left(\boldsymbol{A}_{\mathcal{S} \cup \{i\}}\right).}$$

- 对于单个 i，计算 $\boldsymbol{A}_{\mathcal{S} \cup \{i\}}$ 的行列式需要 $O(|\mathcal{S}|^3)$ 时间。

- 对于所有的 $i \in \mathcal{R}$，计算行列式需要时间 $O(|\mathcal{S}|^3 \cdot |\mathcal{R}|)$。

- 需要求解上式 k 次才能选出 k 个物品。如果暴力计算行列式，那么总时间复杂度为

$$\underline{O(|\mathcal{S}|^3 \cdot |\mathcal{R}| \cdot k) = \; O(nk^4).}$$

暴力算法

- 贪心算法求解：

$$\underset{i \in \mathcal{R}}{\operatorname{argmax}} \;\; \theta \cdot \mathrm{reward}_i \; + \; (1-\theta) \cdot \log\det\left(\boldsymbol{A}_{\mathcal{S} \cup \{i\}}\right).$$

- 暴力算法的总时间复杂度为

$$O(n^2 d \;+\; nk^4).$$

Hulu的快速算法

- Hulu 的论文设计了一种数值算法，仅需 $O(n^2 d + nk^2)$ 的时间从 n 个物品中选出 k 个物品。
- 给定向量 $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_n \in \mathbb{R}^d$，需要 $O(n^2 d)$ 时间计算 $\boldsymbol{A}$。
- 用 $O(nk^2)$ 时间计算所有的行列式（利用 Cholesky 分解）。

Hulu的快速算法

- Cholesky 分解 $\boldsymbol{A}_S = \boldsymbol{L}\boldsymbol{L}^T$，其中 $\boldsymbol{L}$ 是下三角矩阵（对角线以上的元素全零）。
- Cholesky 分解可供计算 $\boldsymbol{A}_S$ 的行列式。
  - 下三角矩阵 $\boldsymbol{L}$ 的行列式 $\det(\boldsymbol{L})$ 等于 $\boldsymbol{L}$ 对角线元素乘积。
  - $\boldsymbol{A}_S$ 的行列式为 $\det(\boldsymbol{A}_S) = \det(\boldsymbol{L})^2 = \prod_i l_{ii}^2$.
- 已知 $\boldsymbol{A}_S = \boldsymbol{L}\boldsymbol{L}^T$，则可以快速求出所有 $\boldsymbol{A}_{S\cup\{i\}}$ 的 Cholesky 分解，因此可以快速算出所有 $\boldsymbol{A}_{S\cup\{i\}}$ 的行列式。

Hulu的快速算法

- 贪心算法求解：

$$\underset{i \in \mathcal{R}}{\operatorname{argmax}} \;\; \theta \cdot \mathrm{reward}_i \; + \; (1-\theta) \; \boxed{\log\det\left(\boldsymbol{A}_{\mathcal{S}\cup\{i\}}\right).}$$

- 初始时 $\mathcal{S}$ 中只有一个物品，$\boldsymbol{A}_{\mathcal{S}}$ 是 1×1 的矩阵，

- 每一轮循环，基于上一轮算出的 $\boldsymbol{A}_{\mathcal{S}} = \boldsymbol{L}\boldsymbol{L}^T$，快速求出 $\boldsymbol{A}_{\mathcal{S}\cup\{i\}}$ 的 Cholesky 分解（$\forall i \in \mathcal{R}$），从而求出 $\log\det\left(\boldsymbol{A}_{\mathcal{S}\cup\{i\}}\right)$。

DPP 的扩展

滑动窗口

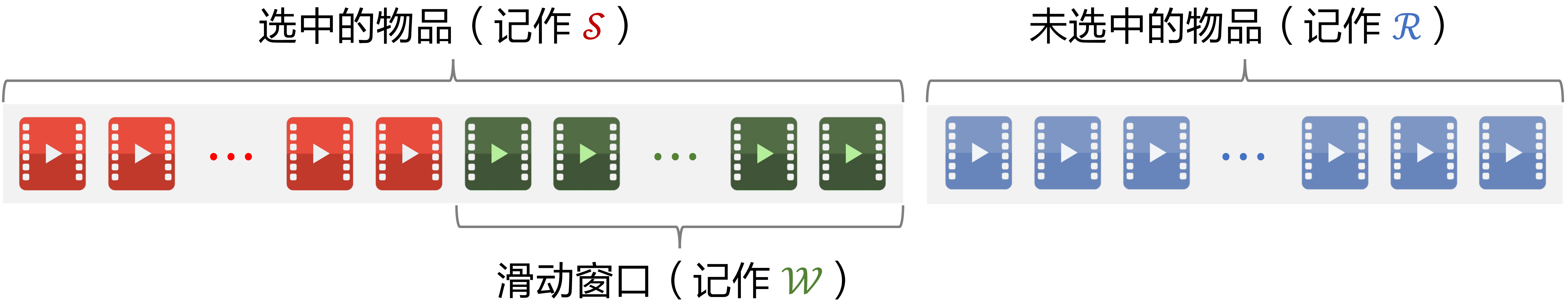
- 用 $\mathcal{S}$ 表示已选中的物品，用 $\mathcal{R}$ 表示未选中的物品，DPP 的贪心算法求解：

$$\underset{i \in \mathcal{R}}{\operatorname{argmax}} \ \theta \cdot \text{reward}_i \ + \ (1-\theta) \cdot \log \det\left(\boldsymbol{A}_{\mathcal{S} \cup \{i\}}\right).$$

- 随着集合 $\mathcal{S}$ 增大，其中相似物品越来越多，物品向量会趋近线性相关。

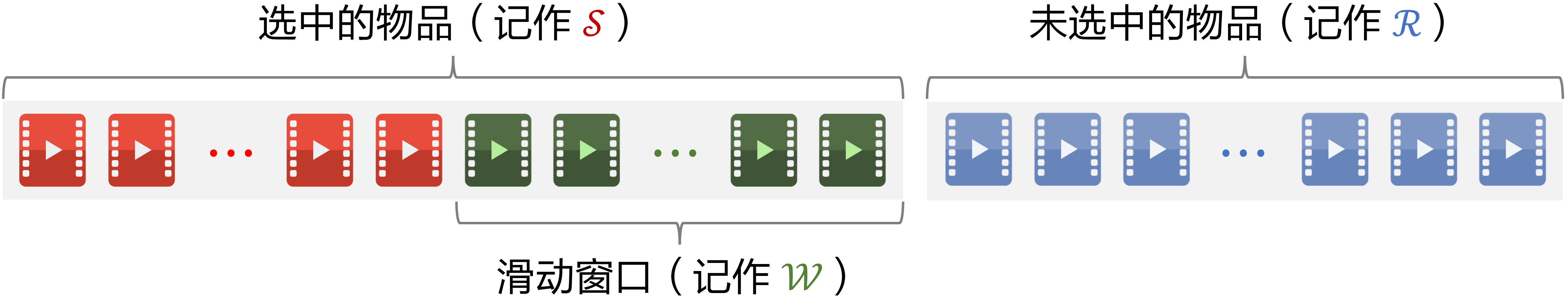
- 行列式 $\det(\boldsymbol{A}_{\mathcal{S}})$ 会坍缩到零，对数趋于负无穷。

滑动窗口



滑动窗口

- 贪心算法： $\underset{i\in\mathcal{R}}{\operatorname{argmax}} \; \theta \cdot \text{reward}_i + (1-\theta) \cdot \boxed{\log\det\left(\boldsymbol{A}_{\mathcal{S}\cup\{i\}}\right)}$.
- 用滑动窗口： $\underset{i\in\mathcal{R}}{\operatorname{argmax}} \; \theta \cdot \text{reward}_i + (1-\theta) \cdot \boxed{\log\det\left(\boldsymbol{A}_{\mathcal{W}\cup\{i\}}\right)}$.



规则约束

- 贪心算法每轮从 $\mathcal{R}$ 中选出一个物品：

$$\boxed{\underset{i\in\mathcal{R}}{\operatorname{argmax}}}\ \theta\cdot\text{reward}_i\ +\ (1-\theta)\cdot\log\det\left(\boldsymbol{A}_{\mathcal{W}\cup\{i\}}\right).$$

- 有很多规则约束，例如最多连续出 5 篇视频笔记（如果已经连续出了 5 篇视频笔记，下一篇必须是图文笔记）。

- 用规则排除掉 $\mathcal{R}$ 中的部分物品，得到子集 $\mathcal{R}'$，然后求解：

$$\boxed{\underset{i\in\mathcal{R}'}{\operatorname{argmax}}}\ \theta\cdot\text{reward}_i\ +\ (1-\theta)\cdot\log\det\left(\boldsymbol{A}_{\mathcal{W}\cup\{i\}}\right).$$

Thank You!