

多样性问题

- 精排给 n 个物品打分： $\text{reward}_1, \dots, \text{reward}_n$ 。
- n 个物品的向量表征： $\boldsymbol{v}_1, \dots, \boldsymbol{v}_n \in \mathbb{R}^d$ 。
- 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 个向量组成的超平行体 $\mathcal{P}(\mathcal{S})$ 的体积越大越好。

多样性问题

$$\mathbf{V}_{\mathcal{S}} = \begin{bmatrix} \text{vector 1} & \text{vector 2} & \dots & \text{vector } k \end{bmatrix}$$

$d \times k$

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 $\mathbf{V}_{\mathcal{S}} \in \mathbb{R}^{d \times k}$ 。
- 以这 k 个向量作为边，组成超平行体 $\mathcal{P}(\mathcal{S})$ 。
- 体积 $\text{vol}(\mathcal{P}(\mathcal{S}))$ 可以衡量 \mathcal{S} 中物品的多样性。
- 设 $k \leq d$ ，行列式与体积满足：

$$\det(\mathbf{V}_{\mathcal{S}}^T \mathbf{V}_{\mathcal{S}}) = \text{vol}(\mathcal{P}(\mathcal{S}))^2.$$

行列式点过程 (DPP)

- DPP 是一种传统的统计机器学习方法：

$$\operatorname{argmax}_{\mathcal{S}: |\mathcal{S}|=k} \log \det(\mathbf{V}_{\mathcal{S}}^T \mathbf{V}_{\mathcal{S}}).$$

- Hulu 的论文[1] 将 DPP 应用在推荐系统：

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

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：

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

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

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

行列式点过程 (DPP)

- DPP 应用在推荐系统：

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

- DPP 是个组合优化问题，从集合 $\{1, \dots, n\}$ 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：

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

求解 DPP

暴力算法

- 贪心算法求解：

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

- 对于单个 i ，计算 $\mathbf{A}_{\mathcal{S} \cup \{i\}}$ 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：

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

- 暴力算法的总时间复杂度为

$$\underline{O(n^2 d + nk^4)}.$$

Hulu的快速算法

- Hulu 的论文设计了一种数值算法，仅需 $O(n^2d + nk^2)$ 的时间从 n 个物品中选出 k 个物品。
- 给定向量 $\boldsymbol{v}_1, \dots, \boldsymbol{v}_n \in \mathbb{R}^d$ ，需要 $O(n^2d)$ 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 $\mathbf{A}_S = \mathbf{L}\mathbf{L}^T$ ，其中 \mathbf{L} 是下三角矩阵（对角线以上的元素全零）。
- Cholesky 分解可供计算 \mathbf{A}_S 的行列式。
 - 下三角矩阵 \mathbf{L} 的行列式 $\det(\mathbf{L})$ 等于 \mathbf{L} 对角线元素乘积。
 - \mathbf{A}_S 的行列式为 $\det(\mathbf{A}_S) = \det(\mathbf{L})^2 = \prod_i l_{ii}^2$ 。
- 已知 $\mathbf{A}_S = \mathbf{L}\mathbf{L}^T$ ，则可以快速求出所有 $\mathbf{A}_{S \cup \{i\}}$ 的 Cholesky 分解，因此可以快速算出所有 $\mathbf{A}_{S \cup \{i\}}$ 的行列式。

Hulu的快速算法

- 贪心算法求解：


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

- 初始时 \mathcal{S} 中只有一个物品， $\mathbf{A}_{\mathcal{S}}$ 是 1×1 的矩阵，
- 每一轮循环，基于上一轮算出的 $\mathbf{A}_{\mathcal{S}} = \mathbf{L}\mathbf{L}^T$ ，快速求出 $\mathbf{A}_{\mathcal{S} \cup \{i\}}$ 的 Cholesky 分解（ $\forall i \in \mathcal{R}$ ），从而求出 $\log \det(\mathbf{A}_{\mathcal{S} \cup \{i\}})$ 。

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：

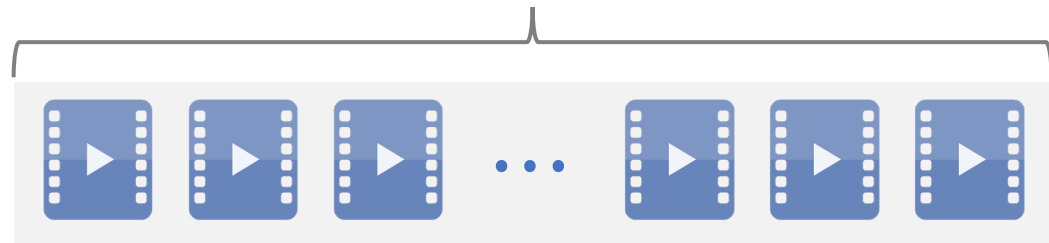
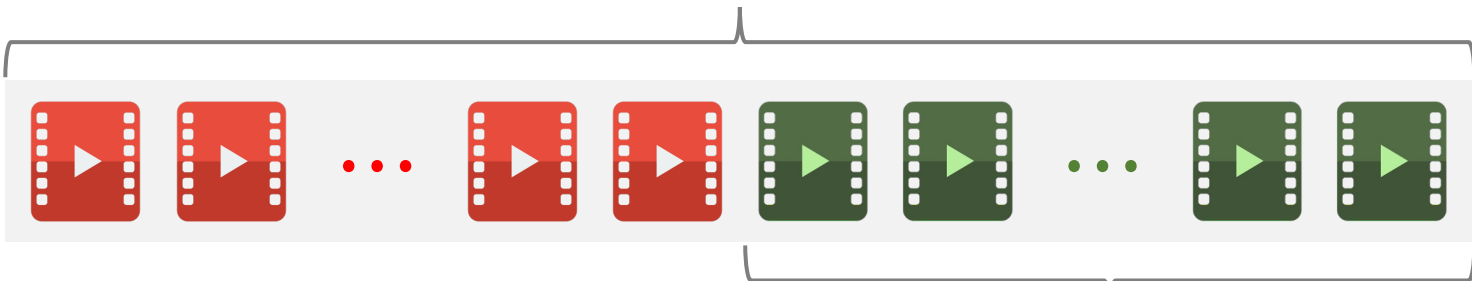
$$\operatorname{argmax}_{i \in \mathcal{R}} \theta \cdot \text{reward}_i + (1 - \theta) \cdot \log \det(\mathbf{A}_{\mathcal{S} \cup \{i\}}).$$


- 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 $\det(\mathbf{A}_{\mathcal{S}})$ 会坍塌到零，对数趋于负无穷。

滑动窗口

选中的物品（记作 \mathcal{S} ）

未选中的物品（记作 \mathcal{R} ）



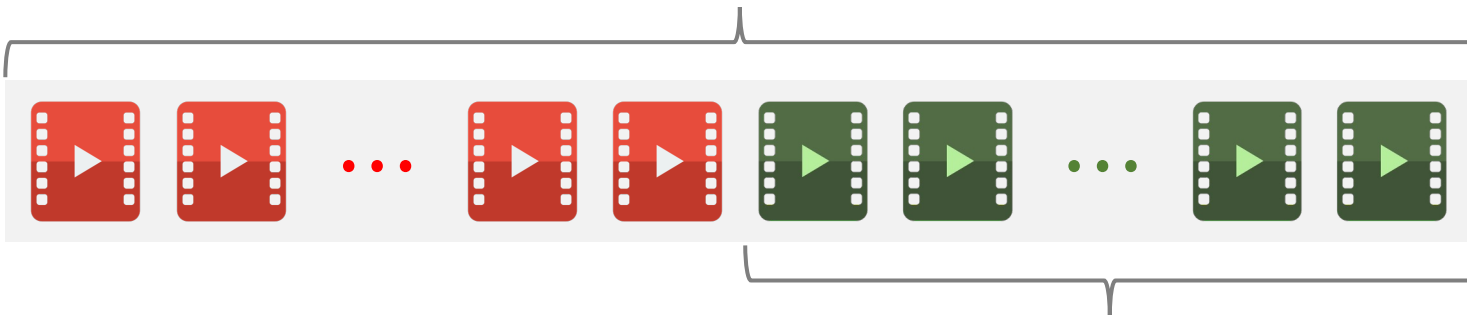
滑动窗口（记作 \mathcal{W} ）

滑动窗口

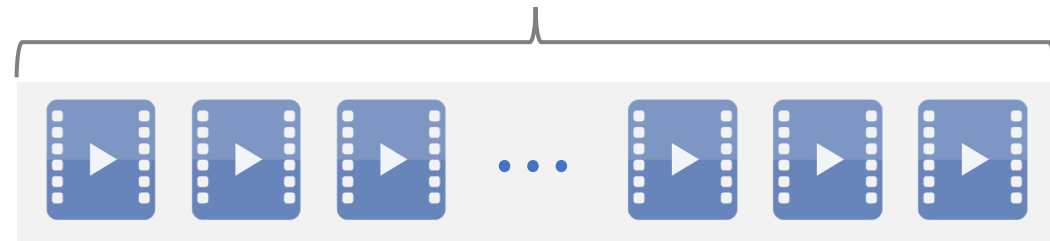
- 贪心算法：
$$\operatorname{argmax}_{i \in \mathcal{R}} \theta \cdot \text{reward}_i + (1 - \theta) \log \det(\mathbf{A}_{\mathcal{S} \cup \{i\}}).$$
- 用滑动窗口：
$$\operatorname{argmax}_{i \in \mathcal{R}} \theta \cdot \text{reward}_i + (1 - \theta) \log \det(\mathbf{A}_{\mathcal{W} \cup \{i\}}).$$

选中的物品（记作 \mathcal{S} ）

未选中的物品（记作 \mathcal{R} ）



滑动窗口（记作 \mathcal{W} ）



规则约束

- 贪心算法每轮从 \mathcal{R} 中选出一个物品：

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

- 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：

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

Thank You!

<http://wangshusen.github.io/>