

第六部分：多样性

- 1、推荐系统的多样性
- 2、Maximal Marginal Relevance (MMR)
- 3、DPP (行列式点过程)

1、推荐系统的多样性

1.1 相似性的度量

1.1.1 基于物品属性标签

根据类目、品牌、关键词等

(是由cv, nlp算法推断出来的)

可以根据一级类目、二级类目、品牌计算相似度

例子：

物品i：美妆、彩妆、香奈儿

物品j：美妆、香水、香奈儿

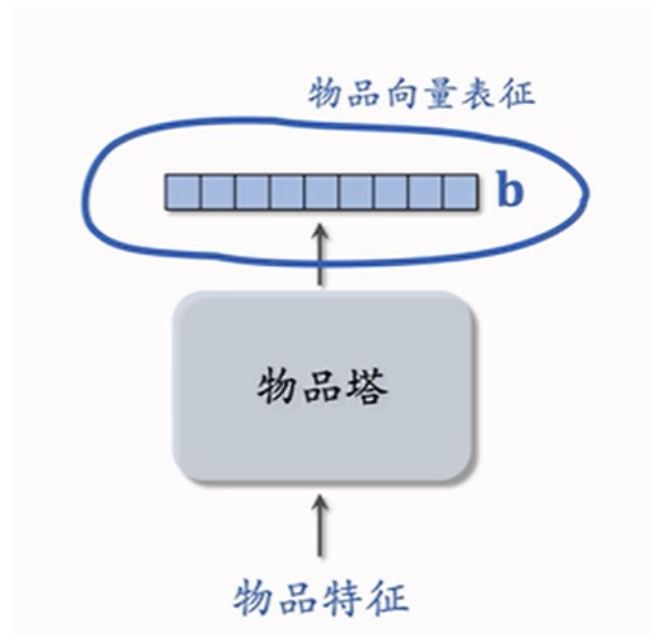
相似度： $sim_1(i, j) = 1$, $sim_2(i, j) = 0$, $sim_3(i, j) = 1$

再根据类目、品牌、关键词等设置的权重，加权计算总的相似度

1.1.2 基于物品向量表征

用召回的双塔模型学到的物品向量 (不好)

如采用内积，余弦相似度等方法

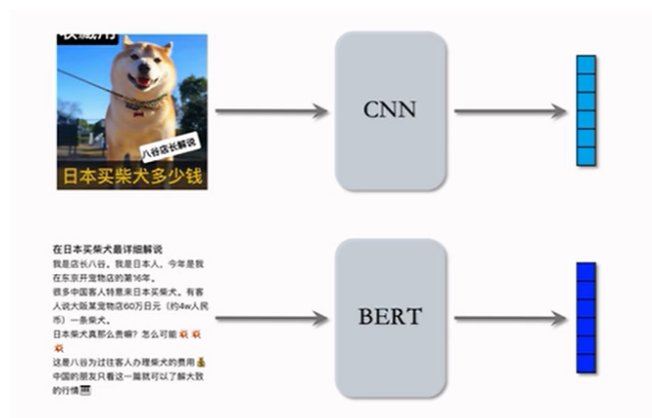


两个物品相似度高，则说明双塔模型得到的物品向量表征内积/余弦相似度高

但推荐系统头部现象很严重，新物品/长尾物品曝光和点击很少，双塔模型学不好这些物品的表征

基于内容的向量表征（好）

（根据cv，nlp模型提取出文字和图片的向量）



难点：如何训练CNN、BERT

CLIP：当前公认最有效的预训练方法

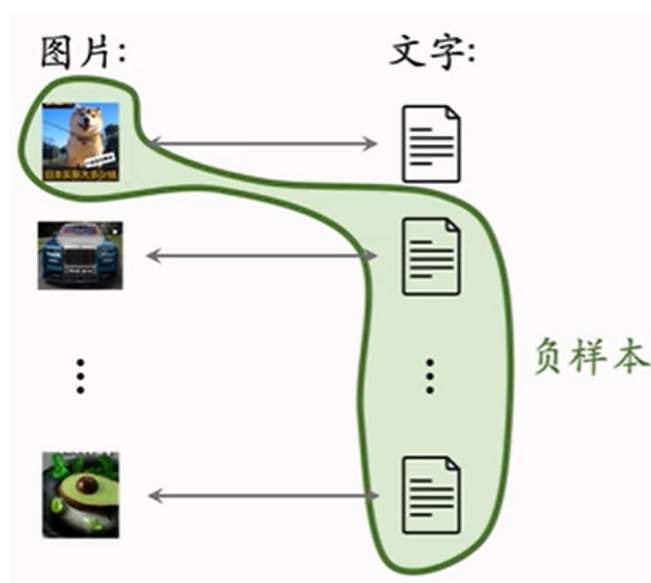
思想：对于图片-文本二元组，**预测图文是否匹配**

优势：无需人工标注，小红书的笔记天然包含图片+文字，大部分笔记图文相关

1.1.3 样本选取（和batch内负样本类似）：

正样本：图片直接对应的文字

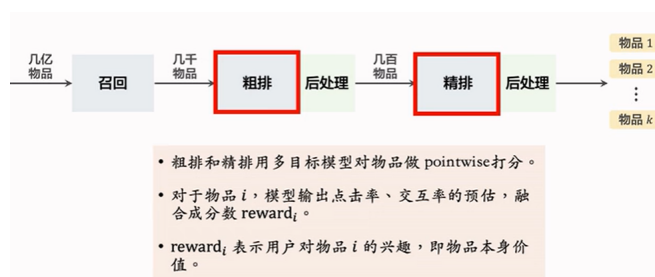
负样本：如下图



一个batch内有m对正样本，一张图片和m-1条文本组成负样本，那么这个batch内一共有 $m(m-1)$ 对负样本

1.2 提升多样性的方法

考虑推荐系统的链路：



粗排和精排的分数字用pointwise独立打分，不考虑物品之间的关联

然后对n个候选物品的打分进行排序，后处理就是从这n个候选物品（粗排n为几千，精排n为几百）选出k个，既要总分高，也需要它们有多样性

因此：

既要考虑n个候选物品的打分 $reward_1, reward_2, \dots, reward_n$

也要考虑第i个物品和第j个物品的相似度 $\text{sim}(i, j)$

精排的后处理：通常称为**重排**，它决定了**哪k个物品被曝光**，以及k个物品**曝光的顺序**

2、Maximal Marginal Relevance (MMR)

MMR先用在搜索排序，再用在推荐排序

MMR考虑的是物品间的相似度

2.1 MMR多样性算法

(1) 已选中的**物品S初始化为空集**，未选中的物品R初始化为全集 $\{1, \dots, n\}$

(2) 选择**精排分数 $reward_i$ 最高的物品**，从R移到S



当i和选中物品j**相似度很高**时，对物品i的选中**起到抑制作用**

θ 参数用来平衡物品价值和相似度， θ 越大，物品价值起作用越多，多样性起作用越少

最大MR：

• Maximal Marginal Relevance (MMR) :

$$\arg\max_{i \in R} MR_i.$$

选出最大的MR作为MMR，对应物品i从R中移动到S

(3) 对 (2) 操作**做k-1轮循环**，结束后**一共选出k个物品**。由于每次循环S和R的集合情况都发生了变化，因此**需要重新计算MR和MMR**

2.2 滑动窗口的引入

2.2.1 上面MMR计算的问题

已选中的物品越多（集合S越大），越不容易找出R中的物品i，使得i与S中的物品都不相似——即 $\max \text{sim}(i, j)$ 总是约等于1，导致MMR算法失效

2.2.2 解决方案

设一个滑动窗口W，比如最近选中的10个物品，用W代替MMR公式中的S



（没有必要让第30个物品和第一个物品也具有很强的不相似，因此滑动窗口的方式是合理的）

2.3 重排的规则

2.3.1 最多连续出现k篇某种笔记

小红书推荐系统的物品分为图文笔记，视频笔记（以下数据仅用于举例，非真实）

最多连续出现 $k=5$ 篇图文笔记，最多连续出现 $k=5$ 篇视频笔记

例如，排 i 到 $i+4$ 的全都是图文笔记，那么排在 $i+5$ 的必须是视频笔记

2.3.2 每k篇笔记最多出现1篇某种笔记

运营推广笔记的精排分数会乘一个大于1的系数（boost），帮助笔记获得更多曝光

例如：为了防止boost影响体验，限制每k=9篇笔记最多出现1篇运营推广笔记
即：如果排在第i位的是运营推广笔记，那么排i+1到i+8的不能是运营推广笔记

2.3.3 前t篇笔记最多出现k篇某种笔记

排名前t篇笔记最容易被看到，对用户体验最重要（小红书的top4为首屏）

小红书推荐系统有带电商卡片的笔记，过多可能会影响体验

例如：

前t=1篇笔记最多出现k=0篇带电商卡片的笔记

前t=4篇笔记最多出现k=1篇带电商卡片的笔记

2.4 MMR+重排规则

重排结合MMR与规则，在满足规则的前提下最大化MR

具体做法：每一轮先用规则排除掉R中的部分物品（防止违反规则），得到子集 R' ，按下面规则计算（公式不变），即可符合要求：

$$\operatorname{argmax}_{i \in \mathcal{R}} \left\{ \theta \cdot \text{reward}_i - (1 - \theta) \cdot \max_{j \in \mathcal{W}} \text{sim}(i, j) \right\}.$$

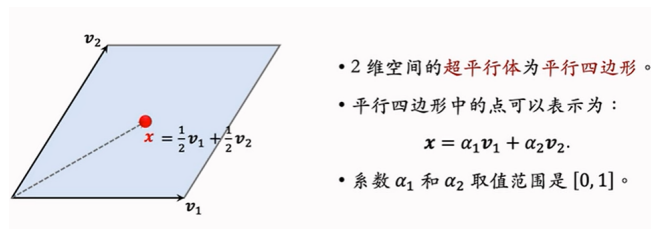
把 \mathcal{R} 替换成子集 \mathcal{R}'

3、DPP（行列式点过程）

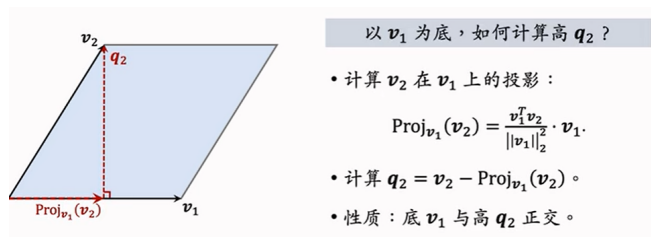
DPP的目标：从集合中选出多样的物品，它考虑选出的物品集合的多样性，是公认的最好的推荐系统多样性算法

3.1 DPP数学基础

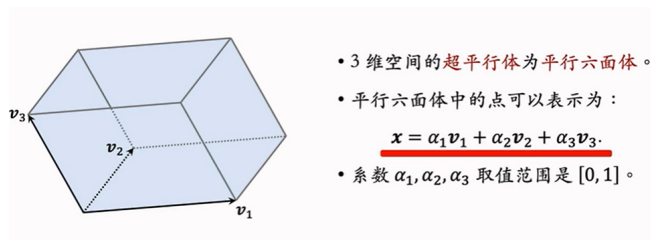
3.1.1 二维空间



平行四边形的面积计算：采用底×高的方式计算面积，即以 v_1 为底，计算高 q_2 ；以 v_2 为底，计算高 q_1 也可



3.1.2 三维空间



平行六面体的体积计算：采用底面积×高的方式计算面积，即计算平行四边形 $P(v_1, v_2)$ 的面积（它是平行六面体 $P(v_1, v_2, v_3)$ 的底），再乘以高 q_3 （垂直于地面 $P(v_1, v_2)$ ）

问题：体积何时最大化，最小化？

设 v_1, v_2, v_3 都是单位向量

当3个向量正交时，平行六面体为正方体，体积最大化， $\text{vol}=1$

当3个向量线性相关时，体积最小化， $\text{vol}=0$

3.1.3 多维空间

一组向量 $v_1, \dots, v_k \in R^d$ 可以确定一个 k 维超平行体：

$$P(v_1, \dots, v_k) = [\alpha_1 v_1 + \dots + \alpha_k v_k | 0 \leq \alpha_1, \dots, \alpha_k \leq 1]$$

要求 $k \leq d$ ，比如 $d=3$ 维空间中可以有 $k=2$ 维平行四边形，但不能有 $k=4$ 维平行四边形

如果 v_1, \dots, v_k 线性相关（即某个 v 可以用其它 v 表示），那么向量将落在一个平面上，则体积 $\text{vol}(P)=0$

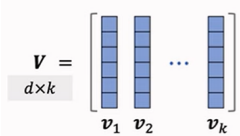
3.2 DPP算法衡量物品多样性

给定 k 个物品，把它们表征为单位向量 $v_1, \dots, v_k \in \mathbb{R}^d$ ，要求 $d \geq k$

用超平行体的体积衡量物品的多样性，体积介于0和1之间

如果 v_1, \dots, v_k 两两正交，此时体积将最大化， $\text{vol}=1$ ，说明物品之间多样性好

如果 v_1, \dots, v_k 线性相关，此时体积将最小化， $\text{vol}=0$ ，说明物品之间多样性差



$V = \begin{bmatrix} | & | & & | \\ v_1 & v_2 & \dots & v_k \\ | & | & & | \end{bmatrix}$

- 给定 k 个物品，把它们表征为单位向量 $v_1, \dots, v_k \in \mathbb{R}^d$ 。 ($d \geq k$)
- 把它们作为矩阵 $V \in \mathbb{R}^{d \times k}$ 的列。
- 设 $d \geq k$ ，行列式与体积满足：
$$\det(V^T V) = \text{vol}(\mathcal{P}(v_1, \dots, v_k))^2.$$
- 因此，可以用行列式 $\det(V^T V)$ 衡量向量 v_1, \dots, v_k 的多样性。

3.3 DPP多样性算法原理

精排给 n 个物品打分： $\text{reward}_1, \dots, \text{reward}_n$ ，其中 n 个物品的向量表征：

$$v_1, \dots, v_n \in \mathbb{R}^d$$

从 n 个物品中选出 k 个物品，组成集合 S ，需要 S ：

价值大： 分数之和 $\sum_{j \in S} \text{reward}_j$ 越大越好

多样性好： S 中 k 个向量组成的超平行体 $P(S)$ 的体积越大越好

$V_S = \begin{bmatrix} | & | & & | \\ \text{集合 } S \text{ 中物品的向量} & & & \end{bmatrix}$

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

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

因此可以用**行列式**衡量向量（即物品）之间的多样性

- DPP 是一种传统的统计机器学习方法：

$$\operatorname{argmax}_{S: |S|=k} \log \det(V_S^T V_S).$$
- Hulu 的论文[1] 将 DPP 应用在推荐系统：

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

*论文的贡献在于**快速求解这个公式**

将上面公式 $V_S^T V_S$ 设为 $A_S (k \times k)$

设A为n×n矩阵，它的(i, j)元素为 $a_{ij} = v_i^T v_j$

给定向量 $v_1, \dots, v_n \in \mathbb{R}^d$ ，需要 $O(n^2 d)$ 时间计算A

$A_S = V_S^T V_S$ 为A的一个 $k \times k$ 子矩阵，如果 $i, j \in S$ ，则 a_{ij} 是 A_S 的一个元素

由此：

- DPP 应用在推荐系统：

$$\operatorname{argmax}_{S: |S|=k} \theta \cdot (\sum_{j \in S} \text{reward}_j) + (1 - \theta) \cdot \log \det(A_S).$$
- 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 S 。
- 用 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：

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

$A_{S \cup i}$ 在矩阵 A_S 基础上多一行和一列，初始状态下S只有1个物品， A_S 是 1×1 矩阵，我们希望红色下划线的两项都尽可能大

3.4 DPP的求解

3.4.1 暴力算法

对于单个 i ，计算 $A_{S \cup i}$ 的行列式需要 $O(|S|^3)$ 时间

对 R 中所有的 i ，计算行列式需要 $O(|S|^3 \cdot |R|)$

需要重复上过程 k 次来选择 k 个物品，暴力计算行列式，总时间复杂度为 $O(|S|^3 \cdot |R| \cdot k) = O(nk^4)$

因此暴力算法总时间复杂度为 $O(n^2d + nk^4)$ ，第一项是计算矩阵 A 的时间，第二项是计算行列式的时间，这个算法太慢

3.4.2 Hulu快速算法

Hulu仅需 $O(n^2d + nk^2)$ 的时间从 n 个物品中选出 k 个物品，它利用Cholesky分解来计算所有的行列式，仅需 nk^2

Cholesky分解：

(1) 分解 $A_S = LL^T$ ，其中 L 是下三角矩阵（对角线以上元素全为0）

(2) Cholesky分解可用于计算 A_S 的行列式

***性质：**下三角矩阵 L 的行列式 $\det(L)$ 等于 L 对角线元素乘积

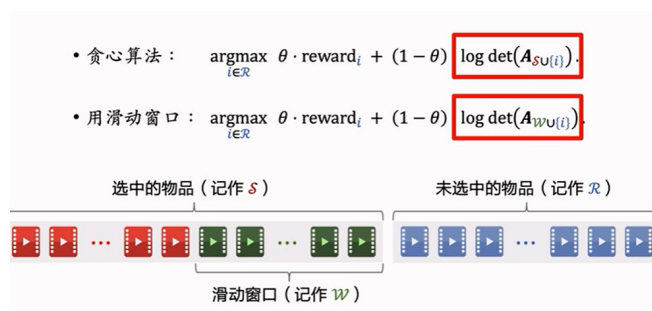
于是 $\det(A_S) = \det(L)^2 = \prod_i l_{ii}^2$

(3) 已知矩阵 $A_S = LL^T$ ，则可以快速求出所有 $A_{S \cup i}$ 的Cholesky分解（每轮循环增加一行和一系列，求Cholesky分解的情况变化不大），因此可以快速算出所有 $A_{S \cup i}$ 的行列式

3.5 DPP的扩展（改进）

用上述贪心算法的问题：随着集合 S 增大，其中相似物品越来越多，物品向量会趋近线性相关，导致行列式趋近为0，对数趋于负无穷，因而做出如下改进：

3.5.1 DPP结合滑动窗口



3.5.2 规则约束

实际推荐系统有很多规则约束，例如最多连续出5篇视频笔记（若已经连续出了5篇视频笔记，下一篇必须是图文笔记）

因此算法不能从 R 中选择任意物品，它必须按规则排除掉 R 中的部分物品，从而满足规则，得到求解公式：

• 用规则排除掉 R 中的部分物品，得到子集 R' ，然后求解：

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