第四章 实验设计与评估指标

4.1 数据集与实验设计

4.1.1 数据集特性

MovieLens

出自F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19 (December 2015), 19 pages.

该数据集包含在线电影推荐服务MovieLens的 69878 位用户对 10681 部电影提出的 10000054 个评分和 95580 个标签。

用户是随机选择的。所有选定的用户至少评价过 20 部电影。与之前的 MovieLens 数据集不同，本数据集不包含人口统计信息。每个用户都由一个 ID 表示，不提供任何其他信息。

数据包含在三个文件中：`movies.dat`、 `ratings.dat`和`tags.dat`。

4.2 实验基本思路

实验采用控制变量法，分两阶段验证假设：

1.基线实验：

- 目标：验证传统协同过滤在千万级数据的可行性
- 参数设置： 相似度阈值：Pearson > 0.3 邻域规模：动态调整（10-50人）
- 计算流程：

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in N(u)} \text{sim}(u,v)(r_{vi} - \bar{r}_v)}{\sum_{v \in N(u)} |\text{sim}(u,v)|}$$

2.优化实验：

- 创新点：引入乘积量化（PQ）加速策略
- 量化参数：

参数	值	理论依据
子空间数（M）	8	平衡压缩率与重建误差
码本大小（K）	256	满足K=2 ^8的二进制存储优化

- 加速原理：

$$计算复杂度从 O(n^2) \rightarrow O(nlogn)$$

4.3 评估指标

4.3.1 推荐质量评估：前十电影与评分依据

1.算法原理支撑

- 基础协同过滤：
  - 邻域筛选：基于用户相似度动态选择Top-N邻居（N=50），权重计算：

$$w_{uv} = \frac{\text{sim}(u,v)}{\sum_{v \in N(u)} \text{sim}(u,v)}$$

  - 评分预测：

$$\hat{r}_{ui} = \bar{r}_u + \sum_{v \in N(u)} w_{uv} \cdot (r_{vi} - \bar{r}_v)$$

- PQ优化方法
  - 特征降维：用户特征从原始10,681维压缩至40维，通过：

$$特征 = [流派偏好, 标签 TF-IDF, 评分分布]$$

  - 量化近似：PQ编码将相似度计算简化为码本内积：

$$\text{sim}_{\text{PQ}}(u,v) = \sum_{m=1}^{M} \langle c_u^{(m)}, c_v^{(m)} \rangle$$

4.3.2 系统效率评估:耗时计算依据

1.时间消耗分解

阶段	基础方法	PQ优化
数据加载	逐行解析（O(n)）	内存映射+并行预处理（O(n/p)）
特征工程	无显式特征构建	40维稠密向量生成（O(dn)）
相似度计算	全量用户对计算（O(n²)）	PQ编码近似（O(n log n)）
排序	全量排序（O(m log m)）	分桶排序（O(m)）

- 关键瓶颈分析：
  - 基础方法中，用户相似度计算占时95%以上（277.6秒/总耗时485.5秒），源于双重循环：

```
for (auto& u1 : users) {          // O(n)
  for (auto& u2 : users) {        // O(n)
    compute_similarity(u1, u2);  // 耗时核心
  }
}
```

- PQ方法通过以下优化降低耗时：
  - **并行化**：OpenMP加速数据加载
  - **量化跳跃**：仅计算同码本簇内用户（减少计算量80%）

2.时间计算依据

- 实验测量法：

```
auto start = high_resolution_clock::now();
load_ratings(...); // 数据加载阶段
auto end = high_resolution_clock::now();
analysis.load_time = duration_cast<milliseconds>(end - start).count() / 1000.0;
```

- 理论验证：
  - 数据加载复杂度： 基础方法：69878用户×10681电影≈7.5亿次I/O操作 PQ方法：内存映射减少磁盘寻址次数（耗时↓77%）
  - 推荐生成复杂度： 基础方法：$69878^2 \approx 4.8\times10^9$次相似度计算 PQ方法：8子空间×256码本→计算量降低至1.2×10^7次

第五章 实现结果对比分析

5.1 推荐质量评估

5.1.1全局准确性对比

1.全局准确性对比指标示例

指标	基础方法	PQ优化	差异分析
RMSE	0.89	0.91	量化误差导致轻微上升（+2.2%）
MAE	0.71	0.73	绝对误差波动在可控范围（<3%）
HitRate@10	68.7%	66.5%	因近似计算下降3.2%

- 误差来源分析：
  - 量化信息损失：40维特征压缩损失高频细节（Top1电影评分误差达±0.5）
  - 邻域覆盖缩减：仅计算同码本簇用户，导致部分潜在高相似用户漏检

2.全局准确性对比指标说明

- RMSE（Root Mean Square Error，均方根误差）
  - 定义： 衡量预测评分与用户真实评分的平均偏差程度，计算所有预测误差的平方均值的平方根。

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(r_i - \hat{r}_i)^2}$$

- 实验意义：
  - 基础方法（0.89）：表示预测评分平均偏离真实评分约0.89分。
  - PQ优化方法（0.91）：量化误差导致误差轻微上升（+0.02），但仍在可接受范围。
  - 敏感度：对高误差（如预测5分但真实1分）敏感，因平方放大误差。

- MAE（Mean Absolute Error，平均绝对误差）
  - 定义： 预测评分与真实评分的绝对误差的平均值。

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|r_i - \hat{r}_i|$$

  - 实验意义：
    - 基础方法（0.71）：平均每个预测偏离真实评分0.71分。
    - PQ优化方法（0.73）：误差增长0.02，与RMSE趋势一致。
    - 特点：对异常值不敏感，直接反映误差绝对值。
- HitRate@10（命中率@10）
  - 定义：

    在推荐的前10个电影中，用户实际评分高于阈值（如4分）的比例。

$$\text{HitRate@10} = \frac{\sum_u \mathbb{I}(\exists i \in \text{Top10}(u),\ r_{ui} \geq 4)}{|U|}$$

    - 基础方法（68.7%）：约68.7%的用户在前10推荐中至少有一个高评分电影。
    - PQ优化方法（66.5%）：因近似计算导致覆盖率下降3.2%，但仍在合理范围内。
    - 实际价值：衡量推荐列表的实用性（是否能覆盖用户真实兴趣）。

5.2 系统效率优化验证

5.2.1耗时分解对比

阶段	基础方法	PQ优化	优化效果
数据加载	207.6秒	62.77秒	耗时降低 77.4%（内存映射+并行I/O）
推荐生成	277.9秒	167.2秒	耗时降低39.8%（PQ近似计算）
总执行时间	485.5秒	195.4秒	效率提升 2.5倍

5.2.2关键优化策略

1.数据加载优化：

- 内存映射技术：通过ifstream的二进制读取，减少磁盘I/O次数。
- 并行预处理：使用#pragma omp parallel for分割文件解析任务，提升吞吐量。

2.推荐生成优化：

- PQ编码加速：将用户特征从原始高维稀疏向量压缩为8子空间编码，计算复杂度从全量用户的O(n^2)降为码本内积的O(n log n)。
- 分桶排序：基于预测评分分桶筛选Top10，避免全量排序的O(m log m)开销。

5.3 工程实践启示

5.3.1优化方案选择建议

场景需求	推荐方案	理论依据
延迟敏感	PQ优化方法	总耗时195秒（满足分钟级响应）
精度敏感	基础协同过滤	RMSE=0.89（误差最低）
资源受限	PQ优化方法	内存占用3.2GB（适合边缘设备）

5.3.2代码可扩展性验证