推荐算法调研

推荐模块主要包括三个环节：召回、排序和调整。如图 1 所示

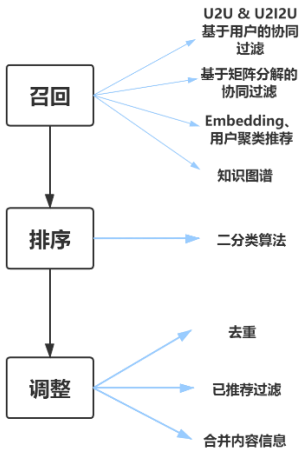

图 1 推荐模块的主要环节

1. 召回

数据集中数据是非常大的，召回是从海量数据中选取跟用户直接相关或间接相关的，比较粗略的内容，召回是我们重点研究的一个点。

我们日常接触到的大部分推荐都是对“物品”进行推荐，比如电影、视频。**而我们项目是基于用户去推荐另一个用户，是用户之间的双向推荐，所以我们采用的召回模型主要是 U2U（User to User）和 U2I2U(User to Item to User)。**

U2U：即给用户推荐用户。如图 2 示，当学生一和学生二都具有很多相似点，比如都喜欢或者不喜欢某个课程，都喜欢去自习等等，我们就认为学生一和学生二是值得推荐的学习伙伴。

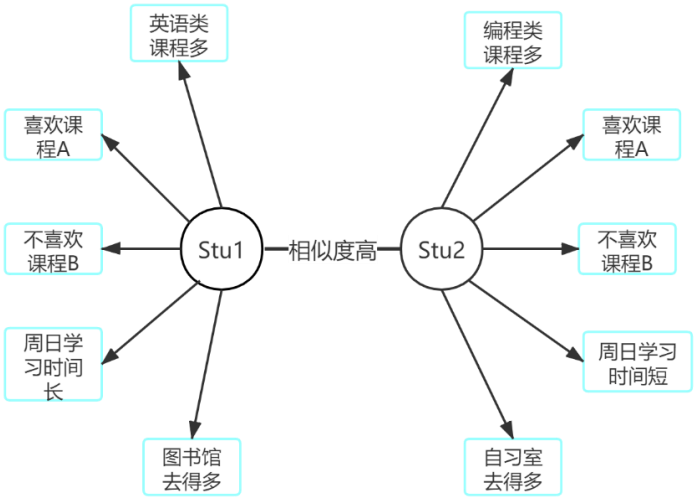

图 2 2U 模型图

（1）基于用户的协同过滤

我们将每个学生的属性特征、行为数据进行数据挖掘后构成特征向量。然后计算每个学生之间的相似度，基于相似度，选取 TOP-N 学生对用户进行推荐。

	CourseA	CourseB	CourseC	Library	Classroom	Sport
StuA	4			5	1	
StuB	5	5	4			
StuC				2	4	5
StuD		3				3

图 3 数据示例表

图 3 是一数据集示例，列属性是每个特征项，然后单元格中的数字是计算得来的每个学生对每项的评分值，空格是缺失值。根据数据，我们可以大致确认 StuA 与 StuB 的相似度大于 StuA 与 StuC 的相似度。下面我们尝试计算每个用户的相似度：

经过资料查找，我们发现了三种相似度的计算方法：

① Jaccard 相似度

给定两个集合 A,B，Jaccard 系数定义为 A 与 B 交集的大小与 A 与 B 并集的大小的比值，即：

$$\mathrm{sim}(A,B)=\frac{|A\cap B|}{|A\cup B|}$$

由此我们可以计算 StuA、StuB 和 StuC 之间的相似度：

$$\mathrm{sim}(StuA,StuB)=\frac{1}{5};\mathrm{sim}(StuA,StuC)=\frac{2}{4}$$

所以计算结果说明 StuA 和 StuC 更相似，但显然结果是不准确的。

分析： Jaccard 相似度实现比较简单，但是这种算法没有利用用户对每一项的评分值，所以是不够准确的

②**余弦相似度**：通过计算两个向量的夹角余弦值来评估他们的相似度。即：

$$\cos\theta=\frac{\sum_{i=1}^{n}(A_i\times B_i)}{\sqrt{\sum_{i=1}^{n}(A_i)^2}\times\sqrt{\sum_{i=1}^{n}(B_i)^2}}$$

这里将缺失值补为 0，计算 StuA、StuB 和 StuC 之间的相似度可得：

$$\mathrm{sim}(StuA,StuB)=0.38;\ \mathrm{sim}(StuA,StuC)=0.32$$

这里之间将缺失值补为 0，显然会影响推荐的准确性，在我们最后的数据测试中也发现，这种方式的召回率准确率不是最好的。

② **皮尔逊相关系数**：度量用户 A 和用户 B 之间的线性关系

$$r=\frac{\sum_{i=1}^{n}(X_i-\overline{X})(Y_i-\overline{Y})}{\sqrt{\sum_{i=1}^{n}(X_i-\overline{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i-\overline{Y})^2}}.$$

这里我们对缺失值的处理是：将缺失值都补为平均值。然后代入用户向量进行计算可得：

$$sim(StuA,StuB)\ =\ 0.09\ ;\ sim(StuA,StuB)\ =\ -0.56$$

这个结果反应了用户之间的线性相关度，负数代表负相关也就是相反，越接近 1，相似度越高，越值得推荐。

在不断测试以及查资料后，**我们最终选取了皮尔逊相关系数法对用户进行了协同过滤。**这种协同过滤的推荐对于我们项目一个最大的优势就是，它只需要行为数据，不需要过多的用户和物品信息，方法直接容易实现，并且可以提供解释。但问题在于如果数据稀疏，对于缺失值的处理也不是很恰当，会导致效果受到影响；另外一个问题是数据量稍大，会导致计算速度变慢，很长时间都出不来结果。

针对这些问题，我们将研究方向转为**基于矩阵分解的协同过滤**。

（2）基于矩阵分解的协同过滤

矩阵分解是将一个矩阵分解成两个或者多个矩阵的乘积，意义层面的解释为通过隐含特征将用户兴趣与特征联系起来；如下图所示，我们通过将输入矩阵进行分解得到两个低维矩阵，再将两个低维矩阵相乘即得到近似于原矩阵的输出矩阵。可以看出，输出矩阵填补了缺失值。

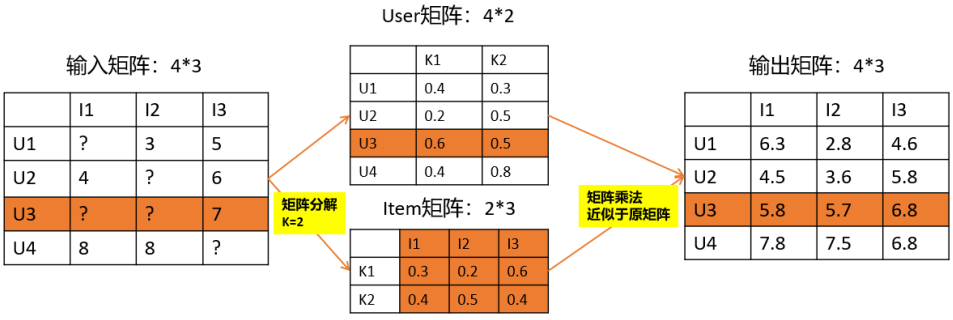

图 4 矩阵分解演示图

通过上面的模型训练，我们对用户向量分解为用户向量和物品向量，这个过程实现了对用户的降维。我们这时对用户向量相乘进行余弦相似度计算，即可得到用户之间的相似度，最后 TopN 排序，选取合适的学习伙伴进行推荐。

（3）聚类推荐

除了基于协同过滤的推荐外，我们也将尝试用户的聚类召回。聚类是一种数据点分组的机器学习技术。给定一组数据点，可以用聚类算法将每个数据点分到特定的组中。主要思想是：找共同点，将用户聚类，充分发挥集体智慧，在每个聚类中互相推荐学习伙伴。

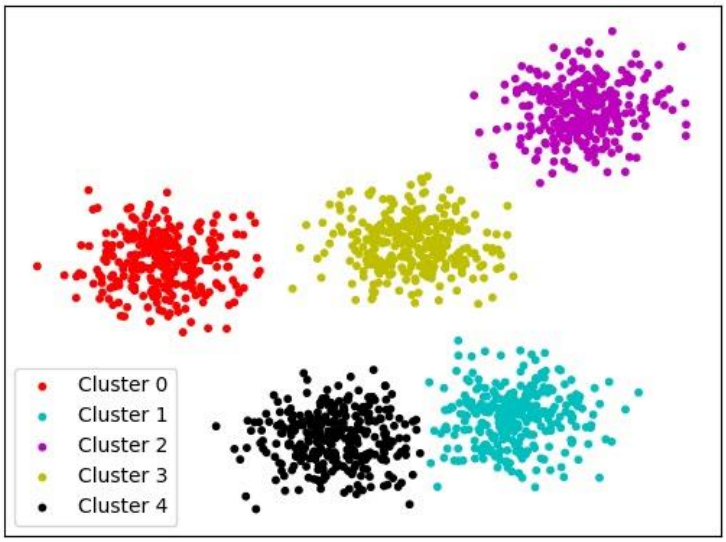

图 21 用户聚类效果图

2 排序

对数据进行召回后，我们会从海量数据中得到少量的内容。然后对结果进行排序，这部分用到主要是机器学习和二分类算法，而针对现有的模型，如上面的代码所示，我们采用用户相似度的二分类算法排序，相似度越高排在越前。

3 调整

前面经过排序得到的结果还需要进行进一步的优化调整，比如去重，已推荐过滤。