# 47|向量空间：如何实现一个简单的音乐推荐系统？

很多人都喜爱听歌，以前我们用MP3听歌，现在直接通过音乐App在线就能听歌。而且，各种音乐App的功能越来越强大，不仅可以自己选歌听，还可以根据你听歌的口味偏好，给你推荐可能会喜爱的音乐，而且有时候，推荐的音乐还非常适合你的口味，甚至会惊艳到你！如此智能的一个功能，你知道它是怎么实现的吗？

## 算法解析

实际上，要解决这个问题，并不需要特别高深的理论。解决思路的核心思想非常简单、直白，用两句话就能总结出来。

- 找到跟你口味偏好相似的用户，把他们爱听的歌曲推荐给你；
- 找出跟你喜爱的歌曲特征相似的歌曲，把这些歌曲推荐给你。

接下来，我就分别讲解一下这两种思路的具体实现方法。

### 1.基于相似用户做推荐

如何找到跟你口味偏好相似的用户呢？或者说如何定义口味偏好相似呢？实际上，思路也很简单，我们把跟你听类似歌曲的人，看做口味相似的用户。你可以看我下面画的这个图。我用“1”表示“喜爱”，用“0”笼统地表示“不发表意见”。从图中我们可以看出，你跟小明共同喜爱的歌曲最多，有5首。于是，我们就可以说，小明跟你的口味非常相似。

| | 安静 | 晴天 | 十年 | 后来 | 春天里 | 遇见 | 平凡之路 | 勇气 | 存在 | 征服 |
|---|---|---|---|---|---|---|---|---|---|---|
| 你 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 小明 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 小王 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 小红 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 小白 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |

我们只需要遍历所有的用户，对比每个用户跟你共同喜爱的歌曲个数，并且设置一个阈值，如果你和某个用户共同喜爱的歌曲个数超过这个阈值，我们就把这个用户看作跟你口味相似的用户，把这个用户喜爱但你还没听过的歌曲，推荐给你。

不过，刚刚的这个解决方案中有一个问题，我们如何知道用户喜爱哪首歌曲呢？也就是说，如何定义用户对某首歌曲的喜爱程度呢？

实际上，我们可以通过用户的行为，来定义这个喜爱程度。我们给每个行为定义一个得分，得分越高表示喜爱程度越高。

| 单曲循环 | 分享 | 收藏 | 搜索 | 听完 | 没听过 | 跳过 |
| --- | --- | --- | --- | --- | --- | --- |
| 5 | 4 | 3 | 2 | 1 | 0 | -1 |

还是刚刚那个例子，我们如果把每个人对每首歌曲的喜爱程度表示出来，就是下面这个样子。图中，某个人对某首歌曲是否喜爱，我们不再用“1”或者“0”来表示，而是对应一个具体的分值。

| | 安静 | 晴天 | 十年 | 后来 | 春天里 | 遇见 | 平凡之路 | 勇气 | 存在 | 征服 |
|---|---|---|---|---|---|---|---|---|---|---|
| 你 | 5 | 3 | 3 | 0 | -1 | 2 | 5 | 4 | 1 | -1 |
| 小明 | 4 | 5 | 2 | 1 | 0 | 3 | 2 | 0 | 1 | 1 |
| 小王 | 1 | 0 | 5 | 5 | -1 | 5 | 0 | 0 | 0 | 2 |
| 小红 | 3 | 0 | 0 | 3 | 0 | 2 | 0 | 4 | -1 | -1 |
| 小白 | 0 | 0 | 0 | -1 | 5 | -1 | 5 | 0 | 4 | 1 |

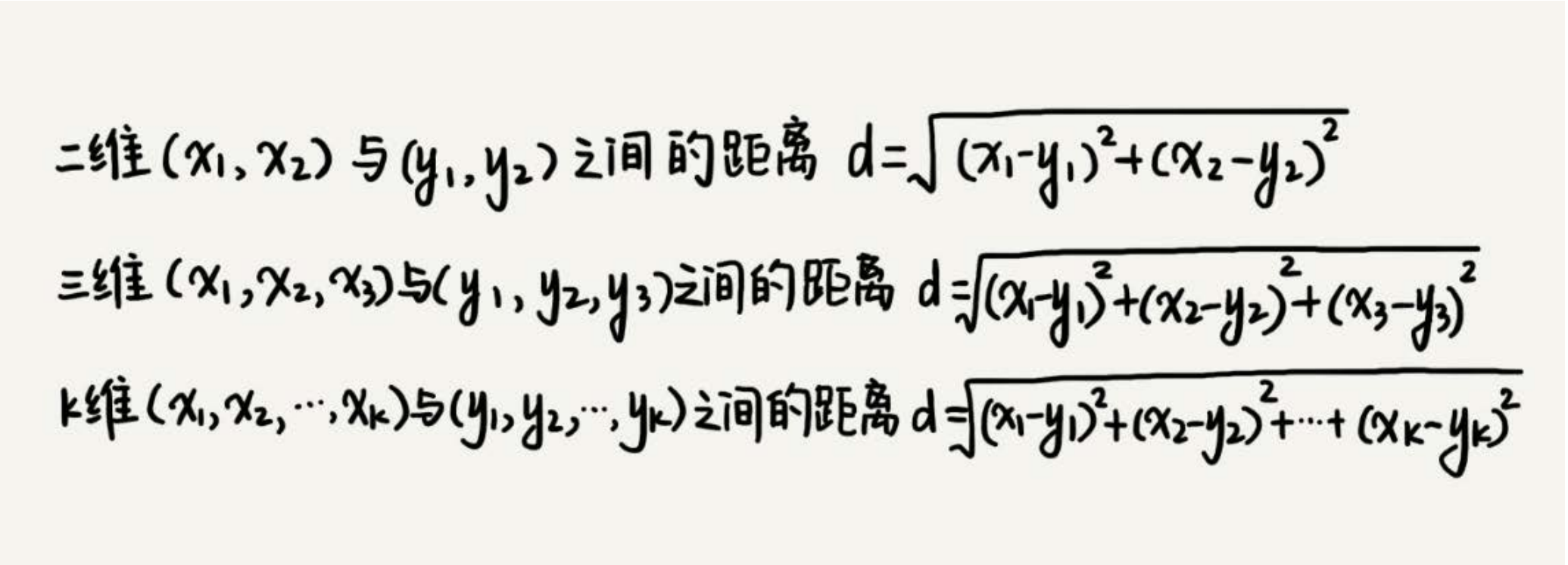
有了这样一个用户对歌曲的喜爱程度的对应表之后，如何来判断两个用户是否口味相似呢？

显然，我们不能再像之前那样，采用简单的计数来统计两个用户之间的相似度。还记得我们之前讲字符串相似度度量时，提到的编辑距离吗？这里的相似度度量，我们可以使用另外一个距离，那就是欧几里得距离（Euclidean distance）。欧几里得距离是用来计算两个向量之间的距离的。这个概念中有两个关键词，向量和距离，我来给你解释一下。

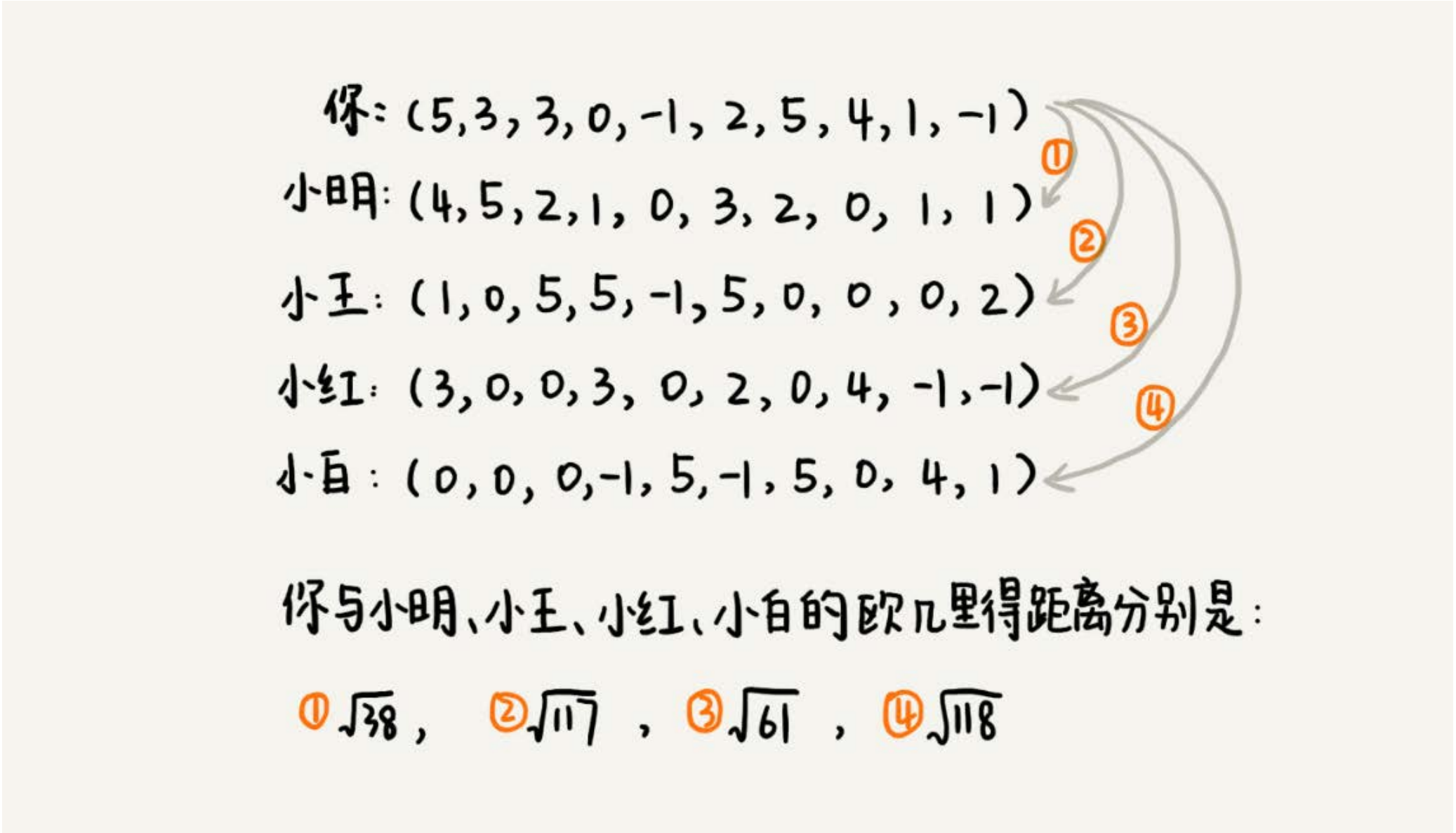
一维空间是一条线，我们用1，2，3……这样单个的数，来表示一维空间中的某个位置；二维空间是一个面，我们用（1，3）（4，2）（2，2）……这样的两个数，来表示二维空间中的某个位置；三维空间是一个立体空间，我们用（1，3，5）（3，1，7）（2，4，3）……这样的三个数，来表示三维空间中的某个位置。一维、二维、三维应该都不难理解，那更高维中的某个位置该如何表示呢？

类比一维、二维、三维的表示方法，K维空间中的某个位置，我们可以写作（$X_{1}$，$X_{2}$，$X_{3}$，…，$X_{K}$）。这种表示方法就是向量（vector）。我们知道，二维、三维空间中，两个位置之间有距离的概念，类比到高纬空间，同样也有距离的概念，这就是我们说的两个向量之间的距离。

那如何计算两个向量之间的距离呢？我们还是可以类比到二维、三维空间中距离的计算方法。通过类比，我们就可以得到两个向量之间距离的计算公式。这个计算公式就是欧几里得距离的计算公式：




我们把每个用户对所有歌曲的喜爱程度，都用一个向量表示。我们计算出两个向量之间的欧几里得距离，作为两个用户的口味相似程度的度量。从图中的计算可以看出，小明与你的欧几里得距离距离最小，也就是说，你俩在高维空间中靠得最近，所以，我们就断定，小明跟你的口味最相似。




## 2.基于相似歌曲做推荐

刚刚我们讲了基于相似用户的歌曲推荐方法，但是，如果用户是一个新用户，我们还没有收集到足够多的行为数据，这个时候该如何推荐呢？我们现在再来看另外一种推荐方法，基于相似歌曲的推荐方法，也就是说，如果某首歌曲跟你喜爱的歌曲相似，我们就把它推荐给你。

如何判断两首歌曲是否相似呢？对于人来说，这个事情可能会比较简单，但是对于计算机来说，判断两首歌曲是否相似，那就需要通过量化的数据来表示了。我们应该通过什么数据来量化两个歌曲之间的相似程度呢？

最容易想到的是，我们对歌曲定义一些特征项，比如是伤感的还是愉快的，是摇滚还是民谣，是柔和的还是高亢的等等。类似基于相似用户的推荐方法，我们给每个歌曲的每个特征项打一个分数，这样每个歌曲就都对应一个特征项向量。我们可以基于这个特征项向量，来计算两个歌曲之间的欧几里得距离。欧几里得距离越小，表示两个歌曲的相似程度越大。

但是，要实现这个方案，需要有一个前提，那就是我们能够找到足够多，并且能够全面代表歌曲特点的特征项，除此之外，我们还要人工给每首歌标注每个特征项的得分。对于收录了海量歌曲的音乐App来说，这显然是一个非常大的工程。此外，人工标注有很大的主观性，也会影响到推荐的准确性。

既然基于歌曲特征项计算相似度不可行，那我们就换一种思路。对于两首歌，如果喜欢听的人群都是差不多的，那侧面就可以反映出，这两首歌比较相似。如图所示，每个用户对歌曲有不同的喜爱程度，我们依旧通过上一个解决方案中定义得分的标准，来定义喜爱程度。

| | 用户A | 用户B | 用户C | 用户D | 用户E | 用户F | 用户G |
|---|---|---|---|---|---|---|---|
| 安静 | 5 | 3 | 0 | 1 | 0 | 0 | 5 |
| 晴天 | 5 | 2 | -1 | 1 | 1 | 1 | 4 |
| 十年 | 3 | 1 | 0 | 0 | 0 | 5 | 0 |
| 遇见 | -1 | 2 | 2 | 2 | 3 | 4 | 0 |
| 勇气 | 0 | 2 | 3 | 1 | 5 | 5 | 0 |

你有没有发现，这个图跟基于相似用户推荐中的图几乎一样。只不过这里把歌曲和用户主次颠倒了一下。基于相似用户的推荐方法中，针对每个用户，我们将对各个歌曲的喜爱程度作为向量。基于相似歌曲的推荐思路中，针对每个歌曲，我们将每个用户的打分作为向量。

有了每个歌曲的向量表示，我们通过计算向量之间的欧几里得距离，来表示歌曲之间的相似度。欧几里得距离越小，表示两个歌曲越相似。然后，我们就在用户已经听过的歌曲中，找出他喜爱程度较高的歌曲。然后，我们找出跟这些歌曲相似度很高的其他歌曲，推荐给他。

## 总结引申

实际上，这个问题是推荐系统（Recommendation System）里最典型的一类问题。之所以讲这部分内容，主要还是想给你展示，算法的强大之处，利用简单的向量空间的欧几里得距离，就能解决如此复杂的问题。不过，今天，我只给你讲解了基本的理论，实践中遇到的问题还有很多，比如冷启动问题，产品初期积累的数据不多，不足以做推荐等等。这些更加深奥的内容，你可以之后自己在实践中慢慢探索。

## 课后思考

关于今天讲的推荐算法，你还能想到其他应用场景吗？

欢迎留言和我分享，也欢迎点击“请朋友读”，把今天的内容分享给你的好友，和他一起讨论、学习。


## 精选留言：

- 許敲敲 2019-01-14 00:32:10
  具体的基于用户相似来推荐的话，如果每个用户喜欢的歌曲数量很大，或者说用户数也很多的情况下，也就是考虑到老师画的表 行列都很多，是不是相当于矩阵的维数很大，这样找到两个向量的距离是有什么trick嘛？或者该用什么算法计算比较好？ [5赞]

作者回复2019-01-14 15:48:48
你指出的这点很好，我会晚点再写一下，补充到文章里。

- Kudo 2019-01-14 11:52:33
  推荐系统（Recommender System）是典型的机器学习应用场景。其核心就是通过算法得到用户偏好向量以及内容向量，两个向量的内积即为用户对内容的的评分预测（即用户对某内容的喜好程度）。推荐学习算法本质上就是学习这两个向量的过程。
  通常有两种方法：
  1. 已知内容向量，学习用户偏好向量的方法就是基于内容的推荐算法（content-based）；
  2. 用户偏好向量和内容向量都未知，则适合使用联合过滤算法（collaborative filtering）同时学习两个向量。 [2赞]

- alic 2019-01-14 09:45:41
  其实就和nlp中计算两个句子之间的相似度类似。 [2赞]

- 李皮皮皮皮皮 2019-01-14 18:44:15
  抱歉老师，我之前可能理解有点偏差，判断两首歌曲是否是同一类型，向量是横向构造的。 [1赞]

- yongxiang 2019-01-14 08:40:44
  还可以用来推荐喜欢的商品 [1赞]

- 睡痴儿 2019-01-26 10:29:25
  包括现在最火的抖音短视频系列的，腾讯看点，淘宝。

- sjz 2019-01-25 17:30:56
  相识图片识别技术也可以使用多维向量相似算法来分析

- orcababyface 2019-01-18 15:58:40
  2.基于歌曲做推荐
  问题：老师的方案的逻辑是：人们对一首歌喜爱程度越一致，那么两首歌越相似。这不是很好吧？难道现在一般音乐app基于歌曲推荐都是这么做的？

- yohann 2019-01-15 16:13:22
  越到后面越难理解，非科班的孩子好忧桑。一步一步来吧！

- Z7k 2019-01-14 20:28:34
  打卡，想问一下现在学习机器学习一般都是找推荐系统的工作吗？一般都是做什么，处理数据还是模型优化？

- cw 2019-01-14 13:26:32
  推荐系统挺复杂的。知乎上有个回答特别好，讲的是小明和自己的狗逛淘宝的故事。
  然后距离也有很多种，欧式距离，曼哈顿距离，余旋距离等。这种矩阵分解式的推荐系统在sparkML库中有专门的api

- 敲键盘的人 2019-01-14 10:59:02
  特征的向量维度不一样怎么处理呢？

- 中午要吃鱼摆摆 2019-01-14 10:41:05
  之前有看过基于奇异值分解SVD的做推荐系统，嗯……现在都忘了，冒泡打卡，勉勉强强跟上了！

- 卡罗 2019-01-14 09:16:12
  基于相似用户做推荐，这一栏里。如果只有欧几里得距离作比较，应该不准确吧，用户，分享和收藏的是不同的歌曲，但是欧几里得距离相近。

- 纯洁的憎恶 2019-01-14 09:02:52
  买书。我买过算法导论，就会推荐算法艺术什么的。

- 纯洁的憎恶 2019-01-14 09:02:01
  把特征抽象为向量，把相似度抽象为距离，就可以使用数学方法量化相近程度。

- 莫弹弹 2019-01-14 09:01:21
  高级篇的人越来越少了……
  我觉得，推荐可以看成一种选优，所以思维上可以跳出“推荐”两个字，进而扩展“相似”“热门”等等这类场景
  例如搜索引擎关键词拼写错误的推荐词，导航app的推荐路径，电商的热门商品等，都可以用上推荐算法

- 伟忠 2019-01-14 08:41:44
  这个算法就是我研究生期间做的化合物相似性，化学反应的相似性的原理。
  原理很简单，但如何在计算机中表示化合物，反应，如何拆分化合物成一个个片段是比较难的。

- 林 2019-01-14 08:35:46
  王前辈，能不能给讲一下原生的协同过滤算法

  作者回复2019-01-14 15:52:38
  跟我讲的这个差不多吧。

- 失火的夏天 2019-01-14 07:58:18
  今天头条的新闻推送，淘宝的猜你喜欢应该都是这种推荐算法