

音乐标签方面的思考

笔记本： 冲突的修改

创建时间： 2014/9/4 17:47

更新时间： 2014/9/7 9:47

URL: file:///

说明：本文章主要介绍一种基于标签有向图的聚合推荐方法，主要针对基于标签的音乐推荐，在文章最后会提出如何将此模型应用到标签检索领域

文章内容包括：

1. 技术领域
2. 背景技术
3. 方案介绍
4. 详细描述
5. 总结

技术领域

此方法用于满足网络环境中为用户推荐音乐等个性化需求。基于标签的音乐推荐技术为网络音乐聚合服务带来了新的表现力，但是这些研究主要围绕平面分类标签。平面分类标签技术存在的问题是标签之间属于离散关系，无法体现出用户对音乐标注的分类偏好，无法反映用户在分类时的认知序：



神经兴奋剂



创建人：人在变

创建时间：2014-7-22

标签：夜店嗨歌 欢快 感性 叛逆 网络热歌

收听人数：32106

8.5

68评价



★ 收藏

➦ 分享

😊 顶

😞 踩



暖心治愈系女声



创建人：我是少根筋姐姐

创建时间：2014-8-30

标签：小清新 网络热歌 90后 流行 华语

收听人数：49216

7.6

38评价



★ 收藏

➦ 分享

😊 顶

😞 踩





谁让蔡琴一夜间家喻户晓



创建人：音乐百科

创建时间：2014-9-1

标签：K歌金曲 经典好歌 70后

收听人数：35945

9.2

46评价



91%



9%



收藏



分享



顶



踩



以上三个歌单来自于QQ音乐，从标签序列上可以认识到该歌单的标签从左到右对该歌单下歌曲的认知度是降低的，可以认为是该音乐人对于此歌单下歌曲的一个认知序列。为更准确地进行音乐推荐，改善平面分类这一问题，该方法引入认知顺序这一维度，更准确的模拟出用户的兴趣特征，以及歌曲的特征。

背景技术

标签推荐技术依据其推荐策略可分成基于内容的推荐技术、协同标签推荐技术、基于社会关系的标签推荐技术、基于图的标签推荐技术、基于加权的标签推荐技术等方面。

(1) 基于内容的标签推荐技术的主要思想是，标签是用户自发标注在音乐上，其反应了音乐的分类、内容和主题。因此，音乐在内容上的相似度表明所标注的标签也应该是相似的。基于内容的标签推荐技术一般应用在音乐可以进行特征描述的领域，如文档、产品等。这种方法采用分类和聚类的方法进行音乐分类，相关的音乐聚集成音乐类别，标签是用来标注音乐的，因此相关的音乐也会有类似的标签。**此技术考虑了标签与音乐内容之间的相互关系，为标签的语义指明了方向。然而，这种技术缺乏对用户主题的考虑。**

(2) 协同标签推荐技术是根据用户在音乐上的标注行为和标签规律确定相似的音乐群或用户群，即利用标签来确定音乐之间的相似度和用户之间的相似度，然后利用这些相似音乐群或用户群所使用的标签进行推荐。协同标签推荐技术不考虑音乐本身的描述特征，用于不易提取音乐特征的领域进行标签推荐，如图像、音频等非文档音乐。协同标签推荐系统应用最广泛的推荐策略就标签的共现思想。这种技术在基于内容的标签推荐技术基础上，将用户、标签与音乐统一考虑，给出了基于标签对用户之间、音乐之间相似度的一种度量策略。**但是此方法缺乏对用户兴趣识别、音乐特征识别及其用户兴趣与音乐特征相似度的度量方法。因此这种技术灵活性差、难以使用网络环境下多样化的需求**

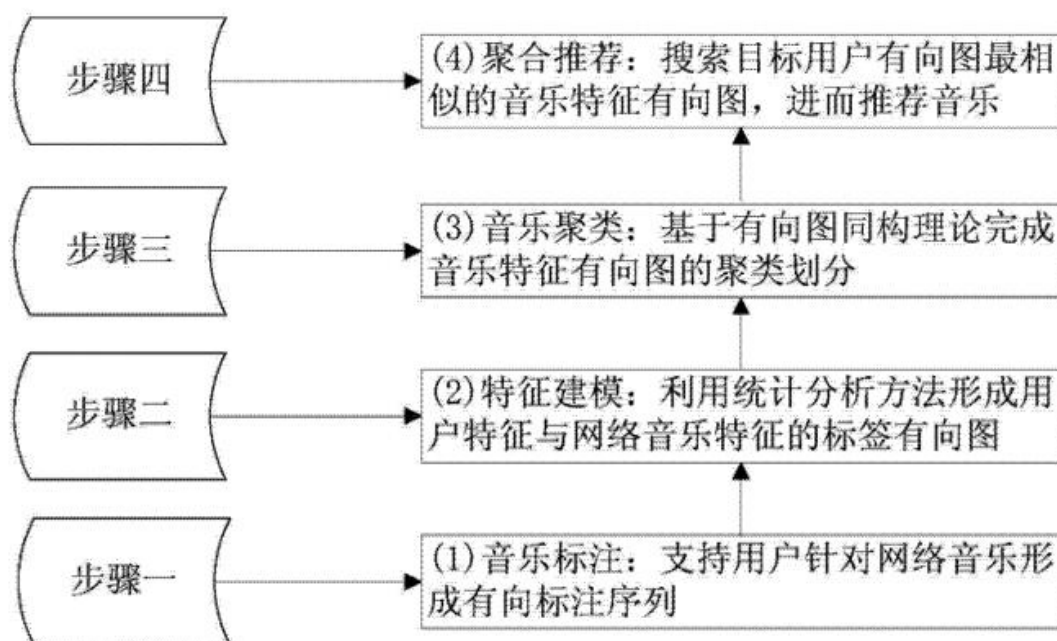
(3) 基于社会网络的网络音乐聚合技术，核心是通过用户社会关系进行个性化的协同标签推荐。首先，利用用户在音乐上所标注的标签建立用户与音乐之间的关系；然后，依据标签的共现和相似度建立音乐与音乐之间的关系；最后，利用用户在社会网络中的联系建立用于用户之间的关系，由此形成三维关系的图模型，利用随机游走算法在三维关系图模型上形成用户的标签群，利用标签群为用户推荐符合用户偏好的标签。在社会标签系统中的三维关系通过映射的方法分成用户与标签之间，用户与音乐之间的二维关系，采用经典的协同过滤方法寻找邻近的用户，由近邻的标签集为用户进行标签的推荐。这种技术充分利用了权威邻居用户对目标用户的影响作用，权威用户的标签可以辅助当前用户对同一音乐的标注。**严格来讲，此技术仍然是利用用户相似度为中心，给出了用户、标签两种元素下的推荐策略，缺乏对音乐特征和用户自身偏好的分析支持。**

(4) 基于图的标签推荐技术是将用户、音乐、标签三者的关系构成三维的图结构，利用图结构进行标签推荐。其主要思想是重要性传递理论，重要的用户采用了重要的标签所标注的音乐，意味着该音乐也是重要的，同理，标注在重要音乐上的标签也是重要的。基于此思想，标签推荐中的音乐、用户、标签三者的关系通过这种方式互相增强，形成了完全的三维关系图结构，采用基于图遍历的算法完成标签推荐，常见的推荐思想是采用随机游走算法在关系形成的图结构中为用户确定推荐的标签。**此技术仍然是以用户相似度为中心，给出了用户、标签两种元素下的推荐策略。缺乏对音乐特征和用户自身偏好的直接分析支持。**

(4) 基于标签的个性化推荐技术比较广泛的应用是利用标签形成的加权标签向量为用户偏好建模。现有的技术中，基本围绕用户在音乐上标注标签的频率、次数、出现的特点等信息，采用概率计算模型等方法统计这些信息的规律，然后把规律转换成可计算的量，用这些量来反应用户的偏好，这不仅具有基于曲目的推荐算法同样的准确性，还具有多样性和新颖性。**但是，此技术有两个明显不足之处：1. 缺乏对音乐特征的考虑 2. 用户偏好方法仅是给出单个标签偏好度量的策略，并未考虑用户多标签组合偏好的情况。**

综上，上述基于标签推荐的音乐推荐技术主要围绕用户、音乐与标签之间的关系展开。但是，基于标签的音乐推荐系统中，主要利用标签共现的思想，这些研究都是建立在标签采取平面分类技术的基础上。平面分类法存在的问题是标签之间属于离散关系，无法体现出用户对音乐标注归类时的分类认知序。接下来介绍的方法建立在一种四元立体关系上，采取有向标签反应用户对音乐分类标注时的认知顺序，形成用户、标签、音乐与认知序的四元关系。采取四元关系，定位于创新性的提高了音乐聚合推荐的精度。

方案介绍



1. 音乐标注：

用于采集用户在网络音乐访问时对音乐的标注数据，形成标签序列数据记录。从用户视角，可以记录用户对不同音乐的每一次标注过程，都产生一个标注序列；从音乐视角，可以记录当前音乐被不同用户的不同标注。此部分数据可以从歌单部分获取，每一个歌单记录了一个音乐人对一部分音乐的认识，标签数据反映了该用户对这批音乐的认知顺序。（例如上面技术领域部分qq音乐的歌单截图）

2. 实现特征建模：

从用户角度，在用户自己的标签序列集中挖掘、提取用户兴趣特征有向图；从音乐角度，在音乐的标签序列集中挖掘、提取音乐特征有向图。特征有向图用以抽象表达用户兴趣特征模型或音乐特征模型。

3. 实现音乐聚类：

为了高效的音乐推荐，本步骤的核心是利用有向图的同构技术，实现音乐特征有向图集合的聚类划分，形成聚类簇。

4. 实现音乐推荐：

当用户使用该系统时，为目标用户快速准确的实现音乐推荐，达到兴趣适应的目的。

详细描述

步骤一：音乐标注

这部分我们没有专门标注平台，但歌单数据完全可以弥补需求，同一个音乐人会标注不同的歌单，同一首歌曲也会出现在不同的歌单里。这样便可以采集到音乐人的标签序列，歌曲被多次标注的标签序列，而且，歌单中的标签排列是一定程度反映认知顺序的，最突出表达歌单主题的标签排在前面：

(1) 关联音乐

$$gmusic_j = \langle Music_j; k_j^1, \dots, k_j^z, \dots, k_j^x \rangle$$

[表达式1]

表达式1表示music_j(该歌曲的ID)在一个歌单里被标注的序列, k1表示该歌曲被标注的第一个标签, kx表示该歌曲被标注的第x个标签
music_j 会出现在不同的歌单里, 于是形成了标注序列的集合:

$$gSetToMusic_j = \{gmusic_j^1, \dots, gmusic_j^n\}$$

[表达式2]

表达式2表示music_j在n个歌单中出现过, 一共有n个标注序列
(2) 关联用户

根据上面的解释, 当获取用户收藏的不同音乐, 或该用户自己为音乐标注的不同标签序列时, 可以得到该用户的m个标注序列集合

$$gSetToUser_s = \{guser_s^1, \dots, guser_s^m\}$$

[表达式3]

步骤二：实现特征建模

首先定义以下变量及概念:

定义1. 标签词频度量

给定标签t, t为标签序列集合 $G = \{g1, \dots, gw\}$ ($w \geq 1$, w为标签序列的总数)中任意标签。设定属性fFreq代表t在标签序列集合G中出现的总次数, 代表标签t的使用频度, 因此可称为频度属性。

定义2. 有向边频度量

有向边代表两个节点之间连接边是有向的, 代表一种顺序关系。假设有向边e为 $ta \rightarrow tb$ (ta, tb 为标签序列中先后出来的两个标签, ta 是 tb 的前置标签)。设定eFreq代表有向边e在标签序列G中出现的次数, 代表使用频度

定义3. 标签有向图

gview代表一个标签有向图, $gview = \langle V, E, LV, LE \rangle$, 其中:

- V为gview中节点的集合;
- E为gview中有向边的集合;
- LV为gview中节点的频度值;
- LE为gview中有向边的频度值;

定义4. 同构标签子图

给定两个标签有向图 $gview = (V, E, LV, LE)$ 和 $gview' = (V', E', LV', LE')$, 其中假设存在一个标签有向图fview, 满足如下两个规则:

(1) 如果任意节点 $v \in fview$, 都有 $v \in gview$ 且 $v \in gview'$;

(2) 如果任意有向边 $e \in fview$, 都有 $e \in gview$ 且 $e \in gview'$;

则称有向图fview是gview和gview'的同构标签子图, gview和gview'是fview的父图

定义5. 子图同构度

假设: 有向图gview与其他有向图存在同构子图fview;

假设: V与E表示同构子图fview中的标签节点集合和有向边的集合; 从上述特征建模技术中频度视角, fview图中V和E在gview存在频度属性值, 定义 子图fview在父图gview上的同构度 $\Phi(fview, gview)$ 为:

$$\Phi(fview, gview) = \text{sum}(v, fview) \times \frac{\sum_{v \in fview \cap E \in gview} LV(v, gview)}{\sum_{v \in gview} LV(v, gview)} + \text{sum}(e, fview) \times \frac{\sum_{e \in fview \cap E \in gview} LE(e, gview)}{\sum_{e \in gview} LE(e, gview)}$$

[表达式4]

其中: (1) $\text{sum}(v, fview)$ 表示累加计算得出图fview中的总结点数

(2) $LV(v, gview)$ 表示获取有向图节点v在有向图gview上的频度属性值tFreq

(3) $\text{sum}(e, fview)$ 表示累加计算出子图fview中的有向边总数

(4) $LE(e, view)$ 表示获取有向边 e 在图 $gview$ 上的频度属性 $eFreq$

$\Phi(fview, gview)$ 综合考虑了子图 $fview$ 中节点和有向边的规模, 并考虑这两者的频度属性值对父图 $gview$ 所占的比例, 反应规律如下

(1) $fview$ 中节点越多, 与父图 $gview$ 同构度越高

(2) $fview$ 中有向边越多, 与父图 $gview$ 同构度越高

(3) $fview$ 中节点的频度属性和在父图 $gview$ 中的节点频度属性所占的百分比越高, 与父图 $gview$ 的同构度越高

(4) $fview$ 中有向边的频度属性和在父图 $gview$ 中的有向边频度属性所占的百分比越高, 与父图 $gview$ 的同构度越高

定义6. 有向图同构图

假设, 两个有向图 $gview$ 和 $gview'$, 存在同构子图 $fview$; 子图 $fview$ 与 $gview$, $gview'$ 的同构度分别为 $\Phi(fview, gview)$, $\Phi(fview, gview')$,

则有:

$$\mu(gview, gview') = \sqrt{[\Phi(fview, gview)]^2 + [\Phi(fview, gview')]^2}$$

[表达式5]

其中:

(1) $\Phi(fview, gview)$ 为子图 $fview$ 与父图 $gview$ 的同构度, 计算法则见表达式4;

(2) $\Phi(fview, gview')$ 为子图 $fview$ 与父图 $gview'$ 的同构度, 计算法则见表达式4;

(3) $\mu(gview, gview')$ 表示存在同构子图 $fview$ 的两个有向图之间的同构度 μ ;

其中: 子图 $fview$ 与父图 $gview, gview'$ 的同构图 $\Phi(fview, gview)$, $\Phi(fview, gview')$ 越高, 父图 $gview, gview'$ 之间的同构度 $\mu(gview, gview')$ 越高

假设在音乐库里, 音乐总量有 N 首, $N \geq 1$: $MusicSet = \{Music_1, Music_2, \dots, Music_j, \dots, Music_N\}$
根据上述所说的建模方法, 对 $MusicSet$ 进行特征建模, 获得音乐特征有向图集合:

$$GviewSetToMusic = \{gviewMusic_1, \dots, gviewMusic_j, \dots, gviewMusic_N\}$$

[表达式6]

其中: $gviewMusic_j$ 是指音乐库中第 j 首音乐通过特征建模方案所形成的音乐特征有向图 (包含特征请看定义3)

步骤三: 音乐聚类

通过步骤二的用户特征建模和音乐特征建模, 可以得到用户特征有向图集合, 以及所有音乐的特征有向图集合。当获取一个用户的特征有向图后, 不可能跟所有音乐的特征有向图逐个比较同构度(定义6), 这个计算量在线上应用中是不可承受的。本步骤是根据音乐特征有向图的分布和规律, 将已有音乐集合按照有向图的相似性规则(有向图的同构度)划分若干个音乐子集。具体步骤如下:

通过聚类方法将数据对象进行划分时, 经常产生聚类领域所谓冷启动和稀疏等现象。为了避免这两个问题, 更高效地完成音乐聚类的过程, 采取将歌曲按照同歌单的歌曲分配到同一初始组的方式, 分组总数定位歌单总数的 $1/4$ (暂定), 在介绍聚类过程前, 首先介绍一下聚类中心的概念:

[3.1] $GviewSetToMusic = \{C_1, \dots, C_i, \dots, C_r, \dots, C_d\}$ (d 代表分组数),
表示所有音乐有向图集合分成了 d 组;

[3.2] $C_i \cap C_r = \Phi$ (C_i, C_r 为不同的分组, 两者交集为空), 求解上述聚类分组的过程称为音乐特征有向图的聚类过程。

[3.3] $C_i = \{gviewMusic_i^1, gviewMusic_i^2, \dots, gviewMusic_i^h\}$ 称为音乐特征有向图集合 $GviewSetToMusic$ 的第 i 个聚类簇, 其中 h 为聚类划分后聚类簇 C_i 中的有向图总数。这些音乐有向图聚类簇满足一下要求:

(1) 每组至少包含一个音乐有向图;

(2) 每个音乐有向图必须属于且只属于某一音乐有向图聚类簇。

[3.4] CC_i 为对应聚类簇 C_i 的聚类中心, 对应聚类簇的代表特征。下面解释聚类中心的获取方法:

$C_i = \{gviewMusic_i^1, gviewMusic_i^2, \dots, gviewMusic_i^h\}$ ，从 C_i 中按顺序选择一个有向图 $gviewMusic_i^y$ 聚类中心 CC_i 初始定义为空的有向图，将 $gviewMusic_i^y$ 中每个节点 v 添加到聚类中心中：如果已经存在节点 v ，则不需要在 CC_i 再增加节点将聚类中心节点 v 的频度属性 $tFreq$ 累加，如果不存在节点 v ，将节点 v 添加到聚类中心节点中，并将节点 v 的频度属性赋值给聚类中心的节点 v 。同理，有向边的处理方式跟节点的处理方式一致。

循环上述步骤直至 C_i 中各有向图中节点和有向边元素合并至聚类中心 CC_i 中。
最后使用均值频度属性代表聚类中心 CC_i 中每个节点 v 和每个边 e 的频度属性，执行策略如下：

$$CC_i.v.tFreq = \left\lceil \frac{CC_i.v.tFreq}{h} \right\rceil$$

$$CC_i.e.eFreq = \left\lceil \frac{CC_i.e.eFreq}{h} \right\rceil \quad [\text{表达式7}]$$

$\lceil \rceil$ 代表向上取整数。

以上是聚类中心的表达方式，聚类过程如下：

Step1: 初始创建下述变量：

(1) 创建聚类簇的数目 d

(2) 创建聚类簇集合 $C = \{C_1, C_2, \dots, C_d\}$

(3) 创建聚类簇中心集合 $CC = \{CC_1, CC_2, \dots, CC_d\}$ ，其中 CC_1 代表聚类簇 C_1 的聚类中心

Step2: 设置上述变量初始所需的数值：

(1) 将有向图集集合 $GviewSetToMusic$ (表达式6)划分为 d 组，将划分结果赋值为

$C = \{C_1, C_2, \dots, C_d\}$ ，保证每个聚类簇中至少有一个音乐特征有向图；

(2) 利用方案[3. 4]计算每个聚类簇的聚类中心

Step3: 执行下述循环步骤：

(1) 从 C_i 中逐一选择每一个聚类簇，然后逐一计算该聚类簇中的音乐特征有向图与所有聚类中心的同构

度(表达式5)： $\mu(gviewMusic_i^a, CC_1), \dots, \mu(gviewMusic_i^a, CC_d)$

根据同构度 μ 大小，判断 $gviewMusic_i^a$ 与哪一个聚类中心的同构度最高，将 $gviewMusic_i^a$ 归并至最高同构度聚类中心对应的聚类簇中。

(2) 上述步骤完成后，重新计算每个聚类簇的聚类中心有向图。

Step4: 当循环直至前后两次形成的聚类簇完全相同时，退出循环步骤，返回聚类划分结果：

(1) 聚类簇集合表示为 $C = \{C_1, C_2, \dots, C_d\}$

(2) 聚类中心表示为 $CC = \{CC_1, CC_2, \dots, CC_d\}$

步骤四：音乐推荐与音乐搜索

与目标用户兴趣特征有向图相似的音乐特征有向图，大部分分布在与目标用户兴趣特征有向图同构度较高的若干个音乐特征有向图聚类簇中。因此，不需要在整个音乐特征有向集上查询与目标用户兴趣特征相似的有向图，只需要在与目标用户兴趣特征有向图同构度较高的若干音乐特征有向图聚类簇中就能查询到大部分适用的有向图。操作步骤如下：

(1) 创建邻居聚类簇集合 $C_{neighbor} = \{C_{neighbor_1}, \dots, C_{neighbor_b}, \dots\}$ ，用于存储于目标有向图 $gviewToUser$ 相邻近的聚类集合。

(2) 判断 $\mu(CC_i, gviewToUser)$ ($1 \leq i \leq d$) (CC_i 表示聚类过程获取的聚类中心) 是否大

于同构度的临界值，如果大于则将聚类簇 CC_i 加到聚类簇集合 $C_{neighbor}$ 中，如果小于则跳过。

(3) 在获取的邻居聚类中心 $C_{neighbor}$ 中，执行下述循环：

从 $C_{neighbor}$ 中逐一选择每一个聚类簇 $C_{neighbor_b}$ ，从该聚类簇中逐一选择每一个音乐有向图 $gviewMusic_b^f$ ，计算该音乐有向图与目标用户的同构度

$$\mu(gviewMusic_b^f, gviewToUser)$$

(4) 根据同构度 μ 的大小，按照自高到低的顺序将 $gviewMusic_b^f$ 添加到推荐歌曲列表中。

假设通过上述步骤取前 p 首歌曲添加到推荐歌曲列表 $gList$ 中，则对该用户推荐的精度可以表示为：

$$\overline{G}_{user} = \frac{\sum_{g \in gList} \mu(g, gviewToUser)}{p}$$

[表达式8]

由于聚类中心集合 CC 的规模远小于整个有向图空间集合 $GviewSetToMusic$ 的规模，因此分析聚类中心对有向图 $gviewToUser$ 同构度的代价相当小。在进行目标有向图 $gviewToUser$ 的最佳有向图查询时，只需要分别在对目标有向图相似度较高的邻居中进行查询，而不是在整个有向图空间中查询，因此能够高效的求解目标用户有向图的最适合的图集。

总结

如上所述，本方案实现了一种为用户进行音乐聚合推荐的新技术。此方案将网络音乐特征与用户兴趣特征用有向图描述，将音乐特征有向图集合分成若干有向图类，使得每个簇中的有向图之间最大程度的同构，而不同簇中的有向图之间最大程度的相异。进行相似度匹配时，不需要在整个音乐特征有向图集上查询与目标用户兴趣有向图相似的音乐特征有向图，只需要在于目标用户兴趣有向图相似度最高的若干音乐特征有向图聚类中就能查询到大部分适用的有向图。因此，能够达到快速、准确推荐的目的。

在音乐标签查询方面，处理方式可以是类似的。用户在输入希望查询的标签组合时，往往最先输入最喜欢的标签，于是便得到了该新用户的兴趣特征图，只不过该特征图结构比较简单，推荐精度可能降低一些，但仍可以高效、准确的获取与该新用户的兴趣特征有向图同构度较高的音乐有向图，达到搜索的目的。

我们可以做的事：

完成此方案需要大量的有认知顺序的歌曲标签，用于音乐特征有向图的建模。目前我们没有能力创建专门的平台来吸引用户来主动给音乐标注，或者组织同事进行大批量的音乐标签标注。最合适的资源应该是歌单，歌单中的标签简洁、明确，并且从观察上来看歌单中的标签排列是有认知顺序的，即最贴合歌单主题的标签被排在最前面。我们可以持续下载热门的歌单，获取比较准确的歌曲-标签信息，用于不断丰富我们的音乐特征有向图模型集合。在有了这个模型集合的基础上，其一，当有条件获取用户收藏的一部分歌曲时，就可以根据这一部分歌曲，建立该用户的兴趣有向图特征模型，从而达到推荐的目的。其二，在组合标签搜索上，系统接收到用户输入的标签组合，进而根据此次输入的标签序列建立该用户的兴趣有向图模型，输出推荐歌曲，以达到搜索的目的。

因此实现该方法需要给予以下两个假设：

假设1：歌单资源中，标签的排列是按照认知顺序排列的，即第一个标签最能反映该歌单的主题

假设2：在实现搜索功能时，用户输入标签的顺序也是按照认知顺序输入的，即用户最先输入的标签，是用户最喜欢的标签

之前的标签规整的工作可以用于歌单歌曲同意标签的合并，在搜索时也可以支持更多社会化标签的检索。

同时，该方法在模型建立时记录了节点的频度，以及边的频度，通过统计一定时期歌单数据里歌曲包含节点、边的总数，可以推理出比较热门的歌曲（节点数与边的总数越多，说明该歌曲在越多的歌单中

被引用，也就越热门），也可以推理出哪些标签比较热门，哪些标签组合比较热门，从而在我们自己的平台上自动的创建高质量的歌单。

以上是我在歌曲标签方面的一些思考，梳理了自己的想法，也更正了之前的一些错误想法，和接下来的主要研究方向，欢迎大家指正。