源码使用说明

1 目录结构

本文实验主要包括离线的 LDA 建模和在线实验两部分，涉及的目录和文件如下所示：

GitRepo/eclipse_workspace/mallet/mallet-2.0.7/ - 使用 Mallet 进行离线 LDA 建模。

data/ - 需要建模的文档集合。

version.txt - 需要建模的文档集合参数:名称、LOC、文档数、主题数。

version-major.txt - 需要建模的文档集合名称。

data/LDA/ - 建模的结果。

script/ - 使用 mallet 进行 LDA 建模的 shell 脚本。

import-data.sh - 导入文档集合。

calculate-perplexity.sh - 计算 perplexity 以确定最优的主题数目。

runLDA.sh - 执行 LDA 建模，结果保存在../data/LDA/中。

GitRepo/Python/experiment/ - 在线实验以及实验结果。

img/ - 所有实验结果的示意图。

log/ - 所有实验的中间 log。

sys/ - 系统实现时歌曲数据集的分析(实验室可忽略)

txt/ - 所有实验的中间结果或需要保留的最终结果，以避免重复计算。

src/ - 所有实验的源码及文档(Python2.7)。

documentation/ - HTML 形式的源文件文档。

const.py - 基本常量的定义。

DBInfo.py - 数据库统计信息的获取。

arimaTrendTest.py - 验证“随着序列的增长，MTSA 的推荐效果并不会 一直提升而是会达到一个平衡”并以此作为选取最大序列长度的基础。

DocGenerate.py - 从数据库中读取用户列表和描述歌曲的显著标签，进而生成歌曲对应的文档。

lastfm.py - 从 Lastfm 上抓取实验相关的数据并存储到数据库中，

本文数据集即由该模块抓取。

main.py - 推荐的实验执行入口，当然也可以在各模块中独立执行。

model.py - 定义歌曲 Song 和列表 Playlist 两个类。

persist.py - 读取和保存中间结果到../txt 中。

predict.py - 实验的核心模块，描述了 MTSA、Local、Global、UserKNN、Markov、PatternMining 等推荐引擎以及混合推荐框架的工作流程。

PrefixSpan.py - 使用 PrefixSpam 挖掘序列数据库中的频繁项集并预测当前序列的下一个项。

test.py - 配置各种推荐引擎的输入参数并得到实验结果。

test_session.py - 验证“MTSA 在会话刚开始时或者跨会话时效果不佳”，进而为混合框架的提出提供基础。

textAnylize.py - 验证“LDA 与 VSM、TF-IDF 等模型相比能够得到更优的推荐效果”。

util.py - 实验中一些常用工具的定义，如相似度和距离的计算、推荐引擎名称和评测指标名称的获取、用户-歌曲矩阵的构建以及转移矩阵的构建等。