Recommendation Algorithm Module

Detailed design specification

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Reviser | Instructions |
| V1.0 | 2020-9-12 | 郭思源 | Draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. **Introduction**
   1. **Purpose**

The purpose of this manual is to explain the technical scheme of the overall design of the system to users and relevant personnel of software development. From the point of view of system design, this manual explains the overall architecture, processing flow, module division, function allocation, interface design, operating environment, data structure design and error handling design of the system. To provide an overall description of how the logic and data functions of the program system are implemented throughout the design period, thus serving as a basis for detailed program design and coding. This document will be the core document for the design phase. The applicable readers of this summary design specification are: customers, system architects, system developers, testers.

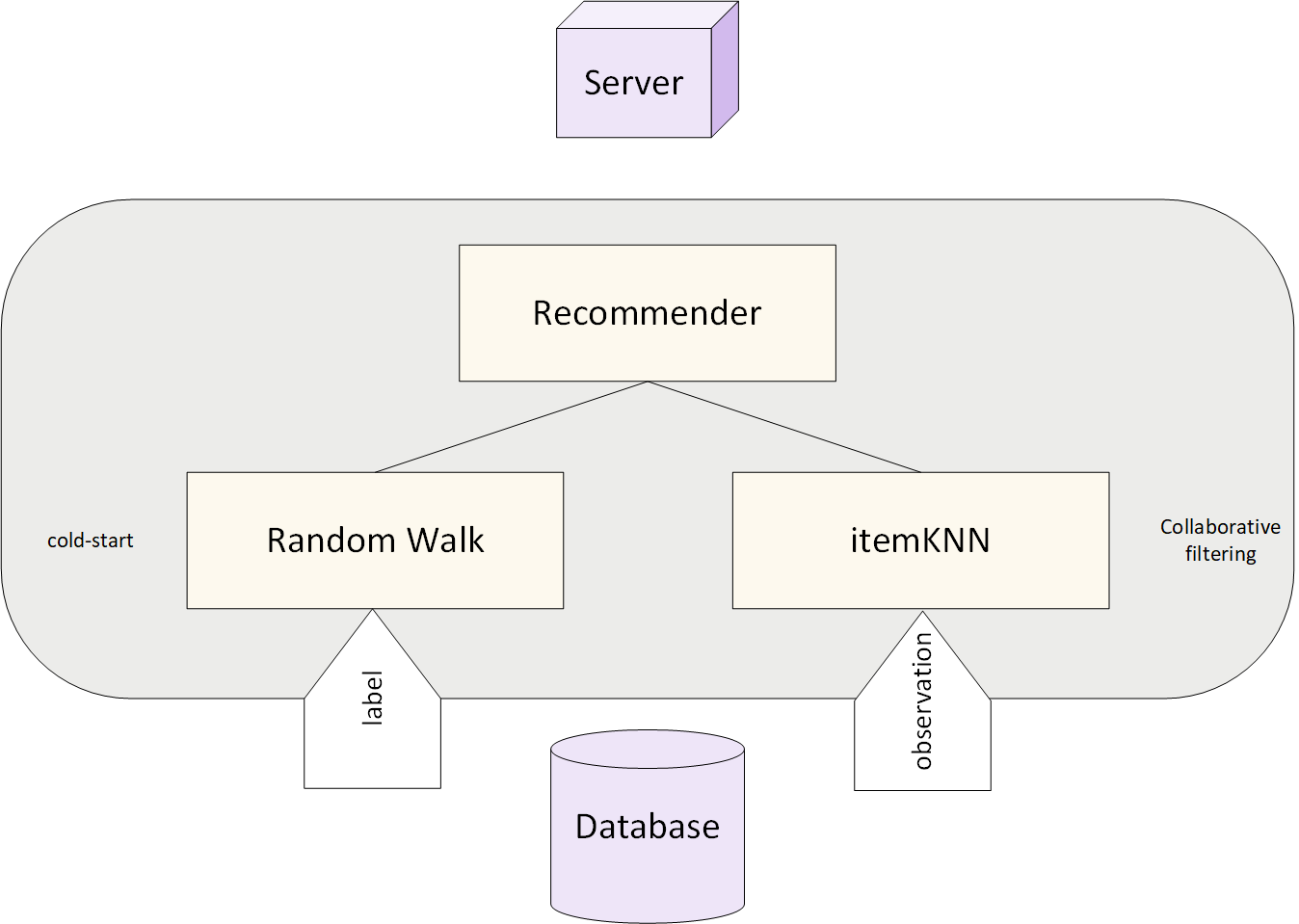
* 1. **References**

1. Sarwar, Badrul, et al. "Item-based collaborative filtering recommendation algorithms." Proceedings of the 10th international conference on World Wide Web. 2001.
2. He, Xiangnan, et al. "Neural collaborative filtering." Proceedings of the 26th international conference on world wide web. 2017.

[3] Jawaheer, Gawesh, Martin Szomszor, and Patty Kostkova. "Comparison of implicit and explicit feedback from an online music recommendation service." proceedings of the 1st international workshop on information heterogeneity and fusion in recommender systems. 2010.

1. **Over Design**

The main function of the module is to make personalized music recommendation for the users, which can be mainly divided into two sub-modules as: (1) Initially, recommend music according to the user-selected labels; (2) Afterwards, recommend music using Collaborative Filtering(CF) methods. The overall framework design can be illustrated by the following figure.



* 1. **Cold-start: recommend with user-selected labels**

**Input:**

user\_name

num (default is 5)

**Output:**

recommendList

**Algorithm:**

·Get label-songID dictionary from the *Database Module*, where the label is chosen by the user\_name

·Random walk to generate the recommendList. Notice that the music in the list is unique.

* 1. **CF: recommend using itemKNN[1]**

**Input:**

user\_name

num (default is 5)

**Output:**

recommendList

**Algorithm:**

·Get observation matrix and the user observation vector from the *Database Module*, where the former is the global data and the latter is the user-specific data.

·Calculate the distance between all the songs and the specific user with the following equation. Note that the distance is corrected by the L1-regularization to make the distance fair.



·Select the num-smallest distance, together with the corresponding song ID to generate the recommendList.

1. **Environment Requirements**

numpy=1.16.5

heapq