

Gestión de Información en la Web

Master en Ingeniería Informática

Práctica 4: Desarrollo de un Sistema de Recomendación basado en Filtrado Colaborativo

Luis Alberto Segura Delgado
DNI: 45922174-Y
`segura2010@correo.ugr.es`

Jueves 9 de Junio de 2016

Índice

| | | |
|----------|--------------------------------------|----------|
| 1 | Introducción | 3 |
| 2 | Trabajo Realizado | 3 |
| 2.1 | Detalles de Implementación | 3 |
| 2.2 | Manual de Usuario | 3 |

1 Introducción

El objetivo de esta práctica es comprender el proceso de recomendación basado en filtrado colaborativo, y una vez hecho esto, llevarlo a la práctica y desarrollar nuestro propio sistema de recomendación basado en filtrado colaborativo. Para ello, se ha utilizado la colección de **MovieLens**.

2 Trabajo Realizado

Se ha desarrollado un sistema de recomendación basado en filtrado colaborativo sencillo. Se ha desarrollado en C++ de forma que sea lo más eficiente posible a la hora de trabajar con un conjunto de datos relativamente grande.

2.1 Detalles de Implementación

En el fichero *main.cpp*, podemos encontrar la funcionalidad básica de nuestro sistema. En el se puede ver como creamos los objetos de las clases **Movies** y **Ratings**. Que nos sirven para hacer los cálculos principales y las lecturas de los datos, como veremos más adelante. Este fichero también contiene la parte de "interfaz de usuario", que básicamente consiste en una interfaz a través de la consola que solicita al usuario que evalúe 20 películas escogidas al azar con una calificación entre 0 y 5. Una vez evaluadas, llamará a las funciones que correspondan para obtener los usuarios más parecidos al usuario actual y las películas recomendadas a partir de estos usuarios.

Para la clase **Movies** podemos ver su implementación en el fichero *Movies.h*. Como podemos ver, esta clase simplemente nos sirve para leer el fichero que contiene la información de las películas y almacenar, para cada película su identificador y su nombre (que son los datos que nos hacen falta). Para almacenar las películas, podemos ver que utilizamos un "Map", lo que nos permite acceder de forma rápida al nombre de una película a partir de su identificador. Además, de esta forma evitamos hacer suposiciones de que tenemos identificadores "seguidos", que tendríamos que hacer si simplemente utilizásemos un vector y el índice del mismo como identificador de la película.

Para la clase **Ratings** podemos ver su implementación en el fichero *Ratings.h*. Esta clase nos sirve para leer el fichero con las puntuaciones de cada usuario a cada película, y para almacenar dicha información. Además, para simplificar el uso, también se encarga de calcular la similitud entre los usuarios y un usuario dado (ver función *getPearsonNearestUsers*). Para hacer el cálculo de la similitud, se utiliza el algoritmo del vecino más cercano visto en clase y el coeficiente de Pearson como función de similitud. Los usuarios se van almacenando en una cola con prioridad, de forma que permanecen ordenados y después solo hace falta ir sacando de la cola los usuario y devolver los K más cercanos.

Una vez que tenemos los usuario más cercanos, podemos utilizarlos con la función *getUserRecommendations* para obtener las recomendaciones más apropiadas para el usuario actual. Al igual que en el cálculo de la similitud, en este caso se realiza el cálculo de la valoración que daría el usuario a las película en base a la valoración que han realizado los usuarios más parecidos. Cada película y la valoración predicha, se almacenan en una cola con prioridad, de forma que quedan ordenadas en función de la valoración predicha.

2.2 Manual de Usuario

La utilización del software desarrollado es sencilla. Simplemente se debe compilar el software haciendo uso del Makefile proporcionado. El ejecutable se generará en la carpeta **bin**. Para ejecutarlo, será necesario pasar como parámetros los ficheros en los que se encuentran las valoraciones de MovieLens y la información de las películas. Un ejemplo de uso sería el siguiente:

```
./p4 ./data/ua.base ./data/u.item
```

Siendo el primer parámetro (`./data/ua.base`) el fichero en el que se encuentran las valoraciones que todos los usuarios han dado a todas las películas, y el segundo parámetro (`./data/u.item`) el fichero donde se encuentra la información de las películas.

Una vez ejecutado, el software preguntará que calificación da el usuario a 20 películas al azar. Y una vez valoradas esas 20 películas, el sistema realizará todos los cálculos y proporcionará una lista de 10 películas recomendadas junto con la valoración predicha para dicha película. Si se desea que aparezcan más películas recomendadas, se puede modificar el valor de la variable **maxRecommendations** en fichero *main.cpp*.