Recomendador

Decisiones Tomadas

Gerard Madrid Miró
Guillem Gràcia Andreu
Ismael Quiñones Gama
Pol Ken Galceran Kimura

En este documento se encuentran las decisiones tomadas por el equipo de desarrollo en relación a los dos algoritmos principales del programa: Collaborative Filtering y Content Based Filtering. También las decisiones acerca de cómo hacer las queries y otras decisiones tomadas sobre otros aspectos.

COLLABORATIVE FILTERING

1: Se ha decidido que “encontrar el centro de un grupo” consiste en tomar todos los ítems valorados por gente de ese grupo y crear un usuario “falso” que ha valorado todos esos items. Las valoraciones para cada uno de esos ítems vendrá dada por la media de las valoraciones a ese ítem de los usuarios afines.

2: Para calcular las distancias entre usuario y centroide, necesarias para construir los clusters, se han ignorado todos aquellos ítems que no hubiesen sido valorados tanto por el usuario en cuestión como por el centroide; tal como hacía en profesor del grupo 21 en el PDF de ayuda que publicó.

CONTENT BASED FILTERING

1: Se ha decidido que, en ComputePonderations, donde se miran todos los atributos de todos los items, no era necesario tratar más de 1500 items, ya que no mejoraba demasiado el resultado y sí influye en el rendimiento. Por ese motivo se corta el tamaño a iterar en el máximo entre el número de ítems totales y 1500. (Línea 3 de la función ComputePonderations).

2: Hemos decidido también, tratar todos los atributos de la misma manera, tanto si eran de texto libre como si eran palabras o códigos. Esto ha facilitado al algoritmo aplicar el mismo caso para todos dado que el tiempo necesario para aplicar un nuevo algoritmo que agrupara el texto libre en palabras y las comparara no salía rentable en mejora de precisión en comparación con el Jaccard implementado.

3: Como está explicado en el documento de pseudocódigo, hemos decidido comparar los atributos uno a uno y parte por parte, eso es comparando por trozos los atributos que tenían varios valores, y los hemos sometido a Jaccard en caso de ser Strings o hemos comprobado su igualdad de forma totalmente binaria en caso de no serlo. Para saber si un atributo era “digno de ser tratado”, se aplica un algoritmo de ponderaciones que itera todos los atributos para definir si son relevantes o no. Posteriormente, a la hora de comparar items solo se comparan (de la misma forma: Jaccard los strings y Binario los demás) los que tienen aplicada una ponderación.

QUERIES

Se ha decidido mantener un archivo llamado Algorithm Tester, que se ejecutará por terminal una vez se cierre la interfaz gráfica. Esta clase ejecutará unas queries que se encuentran en `./FONTS/res/queries/`

Existen queries para los tres algoritmos principales, dónde cada una tiene sus peculiaridades. Las que podremos encontrar siguen las normas explicadas a continuación.

Queries y output de Collaborative Filtering

Las queries de este algoritmo se encuentran en el fichero "`InputQueriesCF_anime750.txt`".

Encontramos un primer valor indicando el número de usuarios que habrá en las queries, **llamémoslo N**.

A continuación, encontramos N bloques con el siguiente formato:

- Primero hay una tripleta formada por un userID, num_known y num_unknown, siendo cada uno:
    - **userID:** Identificador del usuario al que vamos a recomendar
    - **num_known:** la cantidad de items que tiene ese user en el fichero de KNOWN (para el dataset de 750 series)
    - **num_unknown:** la cantidad de items que tiene ese user en el fichero de UNKNOWN (para el dataset de 750 series).

  Este último valor siempre será 10, por como están conformados los archivos de la base de datos.

- Después, aparecerán num_known items con sus notas (Las que el usuario dió a esos ítems). Coinciden con las que hay en el fichero known de la base de datos.

- Por último, encontramos num_known items, que serán los items sobre los cuales queremos que se prediga (en lugar de utilizar todos los items del sistema como candidatos, para así poder computar el NDCG).

El output de las queries, estará compuesto por N bloques con el siguiente formato:

- 4 columnas indicando, respectivamente, userID, recommendedItemID (el item de unknown), predicted rating (predicho por el algoritmo), real rating (la nota que le acabó dando el usuario a ese item, presente en el fichero unknown) .

- El DCG, IDCG y NDCG para esa predicción individual (ese bloque)

Finalmente, después de los N bloques, aparecerá el dato del DCGmedio, IDCGmedio y NDCGmedio. Donde el NDCG indicará cómo de buena es la predicción global en un valor normalizado entre 0 y 1.

Queries y output de Content Based Filtering

Las queries de este otro algoritmo se encuentran en el fichero "`InputQueriesCBF_anime750.txt`".

Encontramos un primer valor indicando el número de usuarios que hay en las queries, **llamémoslo N**.

A continuación hay N filas. Por cada fila, tenemos lo siguiente:

- Un userID (presente en el fichero de unknown del dataset de 750 series)
- El valor de K para el algoritmo de content based filtering

El algoritmo cogerá cada uno de los usuarios y sus 10 ítems presentes en el fichero de unknown, y para cada uno de ellos dirá los K con más similitud (mirando entre todos los ítems del sistema). Por tanto el output será el siguiente:

Veremos 10*N bloques, donde cada bloque representa a un cierto user y a uno de los 10 ítems de unknown asociados al user. Para cada bloque, muestra los 3 items del sistema más similares con su porcentaje de similitud calculado.

Primero nos encontraremos los 10 bloques referentes al primer usuario de las queries, después los siguientes 10 se referirán al segundo y así sucesivamente hasta llegar a los N usuarios.

Queries y output de Hybrid Approach

Las queries son exactamente las mismas que el collaborative filtering.

La salida del algoritmo es también la misma salvo una excepción: No dice la nota predicha. Esto es porque para calcular el DCG, IDCG y NDCG no se tiene nunca en cuenta la nota predicha, y este algoritmo en sí no predice ninguna nota, sino que utiliza la nota que había predicho el collaborative; pero como el orden de recomendación no es el mismo que en el Collaborative en todos los casos, y el orden es lo que influye para el DCG, IDCG y NDCG, se ha decidido no dar la nota predicha porque no aporta nada y simplemente confunde.

INTERFAZ GRÁFICA

A la hora de diseñar la interfaz gráfica, se han tenido que tomar múltiples decisiones. Por un lado, tomábamos decisiones en función de cómo de bonita podía quedar una cosa u otra, y por otro tomábamos decisiones en función del coste computacional y humano que tendría programarlas.

Primero de todo, parece interesante revisar el fichero “FirstUI.jpg” que se encuentra en la misma carpeta que este documento, dónde se pueden observar las primeras decisiones de diseño de la interfaz gráfica, donde ese primer *sketch* no se aleja tanto de la futura imagen que ha tomado nuestro proyecto.

SIGN UP

En el apartado de SignUp, tomamos la decisión de seguir el estilo clásico en cuanto a qué pedir al usuario. Creemos que no era demasiada molestia pedir al usuario un nombre que le identificara, un mail para contactar con él, una contraseña y una respuesta a una pregunta genérica de seguridad.

Para la pregunta de seguridad, se barajaron varias opciones dado que queríamos ofrecer una sola pregunta que pudiera contestar todo el mundo sin sentirse ofendido o desplazado. Por este último motivo quedaron descartadas otras preguntas como “Cuál es el nombre de tu abuelo“ o “Cómo se llamaba tu primera mascota”. Finalmente nos decantamos por preguntar sobre el colegio en que estudió el usuario dado que la gran mayoría de personas han cursado algún tipo de estudios primarios.

LOG IN

Este apartado tuvo poca toma de decisiones fuera de la necesidad de añadir un nuevo apartado para prevenir que el usuario pudiera haber olvidado su contraseña y la adición de la imagen de una figura amigable que salude al usuario y le invite a entrar en su cuenta.

FORGOT PASSWORD

En cuanto a este apartado surgido de las decisiones anteriores, tuvimos que ceder, por falta de tiempo, ante una sencilla pregunta de seguridad como forma de recuperación de contraseña y dejar de lado la idea de mandar un correo electrónico al usuario con sus nuevas credenciales por falta de tiempo.

La decisión principal tomada en esta pantalla fué la de dar al usuario directamente una nueva contraseña, decidida tras barajar el permitirle entrar de nuevo a su cuenta y cambiar la contraseña o permitirle añadir ahí mismo una nueva contraseña. Esta última opción se descartó para evitar mandar al usuario de pantalla en pantalla dado que puede resultar molesto y confuso.

HOME

En la pantalla principal, decidimos desde un primer momento que debíamos mostrar al usuario los Ítems mejor valorados, como se puede comprobar en el sketch ya mencionado.

La decisión más importante en este apartado fué la de añadir a la barra de búsqueda la funcionalidad de autocompletado. Tomamos esta decisión tras ver la dificultad que podría tener el usuario en encontrar un item del que no conocía el nombre exacto. Mediante esta nueva herramienta, el usuario puede buscar parcialmente un Ítem y recibir las respuestas más acordes a su búsqueda.

A raíz de la decisión anterior, decidimos que tras darle a buscar el usuario fuera directamente a la información sobre el Ítem buscado, ya que gracias al autocompletado ya tendría el Ítem en concreto. Descartamos entonces la opción de mostrar los items con nombres similares tras la búsqueda, evitando un clic más al usuario.

EDIT PROFILE

En el apartado de edición de perfil se tomaron diversas decisiones para tratar de dar al usuario una experiencia amena. Primero de todo, se decidió que los valores actuales de todos los campos excepto el de la contraseña debían ser mostrados para poder tener al corriente de sus datos al usuario.

Decidimos también que el campo de la contraseña no debería cambiar ni siquiera en tamaño dado que reduciría la seguridad de usar nuestra aplicación. Sustituimos el feedback aportado por el texto con un mensaje de confirmación de cambio de contraseña.

Por otro lado, se decidió añadir aquí la eliminación de cuenta dado que es referente al perfil del usuario, pero dándole un color rojo para destacar y mostrando una alerta de confirmación para evitar el clic erróneo.

GET STARTED

El primer día que hicimos el sketch ya marcamos esta página como necesaria. Decidimos que era obligatorio tener información del usuario para poder darle recomendaciones afines, así que se obliga a completar esta sección antes de recibir ninguna recomendación.

Decidimos, antes de nada, dar la opción al usuario de añadir Ítems mediante la barra de búsqueda con autocompletado frente a la opción de mostrarle Ítems genéricos y pedirle una valoración dado que no tenía porqué conocerlos.

Decidimos más adelante que podíamos permitir al usuario navegar libremente y valorar por su cuenta 5 Ítems, librándose así de completar este apartado.

RECOMMEND ME

En una primera instancia, este apartado iba a mostrar al usuario cinco recomendaciones de Ítems, y la opción de guardar esas 5.

Más adelante, decidimos que era mejor mostrar más recomendaciones al usuario para evitar que tenga que salir y volver a entrar en el apartado. De esta nueva forma, el usuario podía pedir hasta 25 recomendaciones y, luego, valorar más ítems para recibir más recomendaciones. Decidimos que las recomendaciones se guardaran en este apartado hasta que el usuario añadiera nuevas valoraciones para evitar tener que volver a ejecutar el algoritmo.

Se puede consultar en el apartado 2 de "Otros Aspectos", por qué decidimos no añadir la funcionalidad de filtrado en las recomendaciones que se ofrecían al usuario.

ITEM INFO

Este apartado fue una idea nueva, que surgió a raíz de la intención de pulsar sobre un item para obtener más información. Decidimos pues que así debíamos hacerlo, mostrando al usuario el nombre, descripción, categorías e imagen (si se tiene) del Ítem en cuestión.

Decidimos más adelante que era conveniente añadir 3 Ítems similares al Ítem que se estaba visitando para permitir navegar mediante Ítems similares al usuario. Para ello usamos Content Based Filtering, ya que da los Ítems más cercanos de uno dado.

Finalmente, decidimos que se pudiera clicar en estos Ítems e ir a su propio Item Info, permitiendo después al usuario volver hacia atrás mediante el Stack de Items de SceneManager (explicado en Estructuras de Datos y Algoritmos).

LIKED Y RECOMMEND LIST

Finalmente, las últimas decisiones que tomamos en cuánto a la interfaz gráfica fueron enfocadas a la idea inicial que teníamos de lista para mostrar tanto los items que le habían gustado al usuario como las recomendaciones que se había guardado.

Una vez nos pusimos a diseñar y programar las dos clases, encontramos la posibilidad de gestionar todos los Ítems de forma matricial en una Grid. Este descubrimiento superó nuestras expectativas y decidimos implementar en ambas clases el mismo sistema, y aplicar eliminación de ítems en el caso de la lista de recomendaciones guardadas.

OTROS ASPECTOS

1: Se ha decidido dar IDs negativas a los nuevos usuarios creados a través de la interfaz gráfica de la aplicación (es decir, a aquellos que no venían en los datasets originales que se nos proporcionó en la asignatura). Se ha decidido así para evitar tener que controlar que un ID dado a un usuario nuevo no coincida con el ID de un usuario que existía en el dataset previo. Por tanto, los usuarios que creemos a través de la interfaz gráfica tendrán los ID -1, -2, -3, y sucesivos.

2: Hemos decidido eliminar el caso de uso optativo que teníamos de aplicar filtros a las recomendaciones. Por un lado, quedaba poco tiempo para la entrega y resultaba más complejo de lo que teníamos pensado dado que habría que extraer de la base de datos distintos atributos sobre los que filtrar sin saber si realmente el usuario es capaz de apreciar su valor. Aún haciendo ya esto para los títulos, imágenes, descripciones y categorías de los items, aplicar el filtro resultaba bastante más complejo que simplemente mostrar por pantalla dicho atributo.