Trabajo Escrito

Introducción y Objetivo del Proyecto

En este proyecto se desarrollará un algoritmo de sistemas de recomendación basado en filtrado colaborativo que sea capaz de dar una acertada sugerencia dependiendo el sistema al que se aplique. Para poder generar este sistema de recomendación se utilizará el método de descomposición SVD (Singular Value Descomposition), debido a que es uno de los que tienen mejor respuesta y resultados tienen. Debido a que uno de los objetivos del proyecto es desarrollar este sistema de recomendación en una plataforma en paralelo, se utilizará CUDA, por lo que se desarrollará el algoritmo utilizando CuSolver a través del método Jacobi que ayuda a encontrar la descomposición SVD con vectores singulares.

La razón principal por la que se seleccionó este proyecto es debido a que en la actualidad los sistemas de recomendación forman parte fundamental de muchas de las aplicaciones y herramientas que son parte de nuestra vida diaria, estas ayudan a que nuestra experiencia de navegación y utilización sea de mayor calidad y más personalizada. Durante los últimos años se ha invertido mucho dinero y tiempo en investigación para mejorar estos sistemas de recomendación y que sean lo más precisos posibles. Es por esto que durante la investigación que realizamos, seleccionamos la descomposición SVD, la cual por ejemplo fue utilizada en el famoso concurso de Netflix, donde en conjunto con algunos otros algoritmos, formó parte de la solución ganadora. Una desventaja de este sistema es que puede llegar a ser un tanto lento para cantidades de información muy grandes, pero sus resultados son altamente confiables.

Introducción al Filtrado Colaborativo

Comenzaremos explicando a detalle en qué consiste el filtrado colaborativo, ya que forma parte de las bases de nuestro algoritmo y de los sistemas de recomendación. Los sistemas de recomendación y que específicamente utilizan filtrado colaborativo como su nombre lo sugiere, utilizan los datos o valoraciones (ya existentes) de cierto usuarios para predecir las valoraciones del resto del conjunto. Este concepto es sumamente importante debido a que en conjunto con la descomposición SVD funciona de manera muy eficiente.

Para poder realizar la clasificación de usuarios y poder adaptar el algoritmo al filtrado colaborativo, se debe seleccionar el sistema de clasificación para poder seleccionar los elementos que serán más parecidos a nuestro usuario y así poder realizar una clasificación clara. De los tres tipos de clasificación se encuentra el método basado en memoria, metodo basado en modelos y el método híbrido. De los anteriores se utilizará el método basado en modelos, ya que una vez que se construyan los vectores a partir de la descomposición, se podrá utilizar una matriz de votaciones para así poder seleccionar el conjunto de usuarios más similiar al que utilice nuestro sistema, y generar un modelo que nos proporcione la mayor similitud.

Introducción a SVD

SVD (Singular Value Descomposition), es un método utilizado frecuenemente en diversas aplicaciones de Ciencia de Datos. Ésta es una técnica en la que se descompone una matriz en tres distintas matrices que nos ayudará posteriormente a utilizar cada una de ellas en nuestro sistema de recomendación. Como se menciona en el paper de Ricardo Moya“SVD APLICADO A SISTEMAS DE RECOMENDACIÓN BASADOS EN FILTRADO COLABORATIVO” Respecto a las dimensiones de las matrices tenemos que la matriz A va a tener unas dimensiones de (n $\times$ m) es decir n filas y m columnas. La matriz U va a tener dimensión (n$\times$n), la matriz S tendrá dimensión (n$\times$m) y por último la matriz V tendrá dimensión (m $\times$ m). Al ser V una matriz cuadrada las dimensiones de la matriz traspuesta serán las mismas que la matriz original. Es importante notar que las matrices tienen que cumplir con ciertas propiedades para poder ser utilizadas en estos cálculos. Si se desea revisar a profundidad este tema, se pueden consultar las referencias anexas a este documento, donde se explica a detalle este procedimiento.

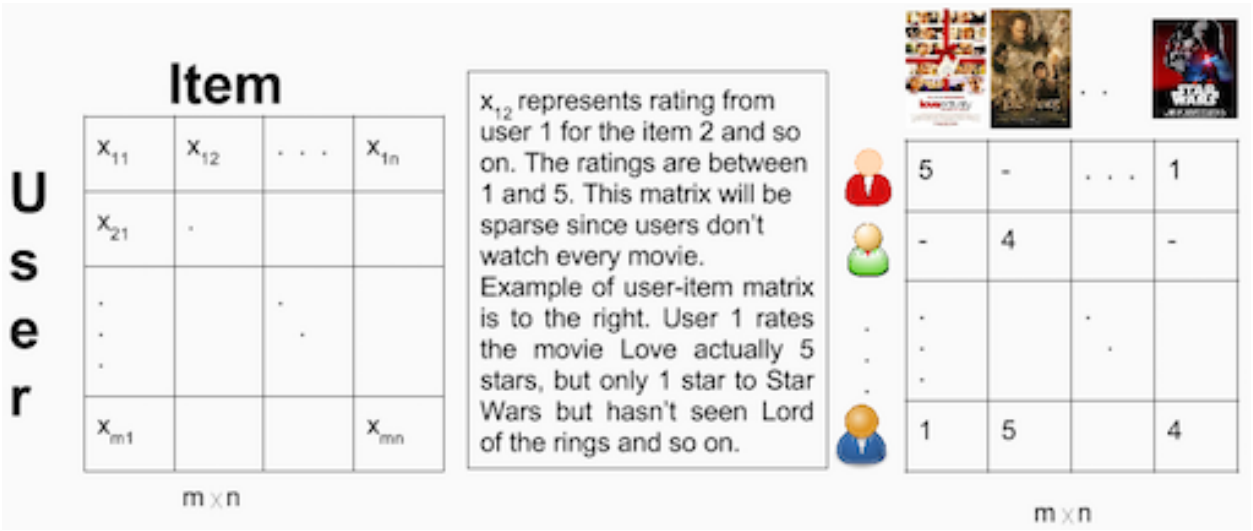

Figure 1:

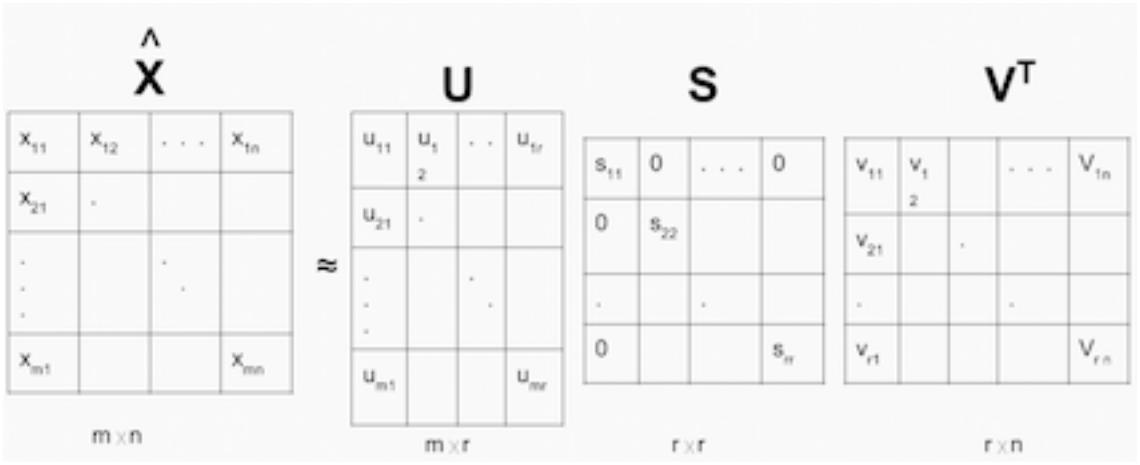

Figure 2:

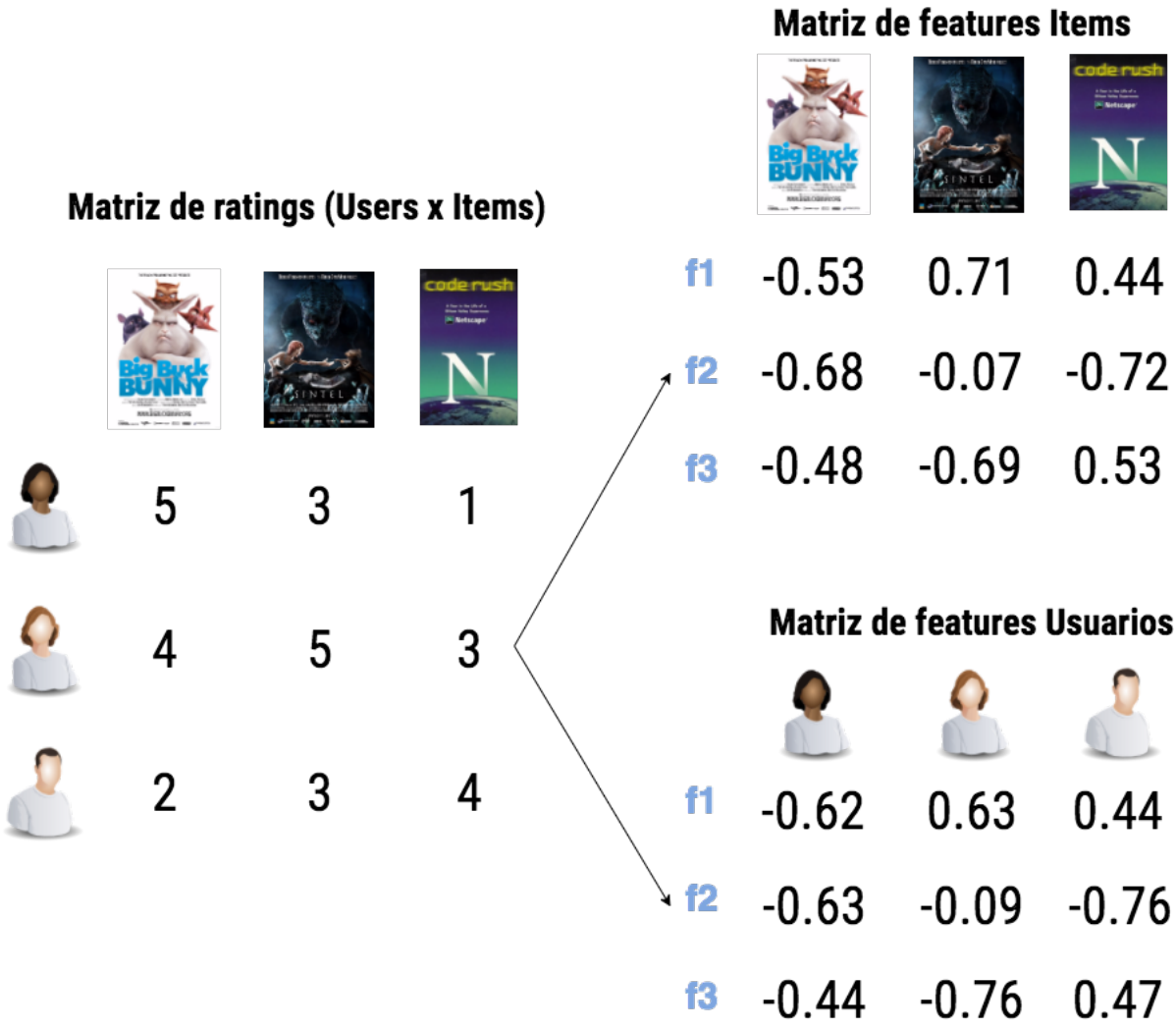

Figure 3:

SVD y el Filtrado Colaborativo

Como es de esperarse, el método de SVD juega un papel súmamente importante en el filtrado colaborativo, y tanto en la selección de vectores para los usuarios globales como en la selección de vectores para nuestro usuario y sus preferencias a predecir. El poder de este algoritmo es importante ya que nos ayuda a encontrar una relación entre item y usuario y así calcular las distancias que existe entre cada uno para así aplicar alguna métrica de similutd para el filtrado colaborativo. Esto nos ayudará en gran medida a entender correctamente el proceso que se debe realizar para poder aplicar la descomposición SVD a sistemas de recomendación y con esto construir el algoritmo necesario para la paralelización del proceso.

De manera breve explicaremos el papel que juega cada una de las matrices obtenidas de la descomposición SVD. Específicamente la matriz U contendrá a los usuarios en una dimensión de: usuarios x K). La matriz V contendrá a los items a recomendar en una dimensión de: items x K. La matriz S puede ser dividida en dos matrices y la mutliplicamos por cada una de las U y V, podremos obtener una matriz U y V con los factores caracterísiticos de cada usuario y item y así poder predecir los items faltantes para cada usuario.

Cálculo de las matrices U, S y V

Cálculo de la matriz U

1. Para poder obtener la matriz U debemos multiplicar la matriz A por su transpuesta para de esta maner obtener una matriz cuadrada como especificamos con anterioridad.

$$A_{n\times m} \times A^t_{m\times n} = C_{n\times n}$$

2. Como segundo paso se obtienen los autovalores
3. Se obtienen los autovectores una vez que tenemos los autovalores
4. Con los autovalores y autovectores obtenemos la matriz U

Cálculo de la matriz V

Mismos pasos que para obtener matriz U pero la primera multiplicación es al revés

$$A^t_{m\times n} \times A_{n\times m} = C_{m\times m}$$

1. Para poder obtener la matriz U debemos multiplicar la matriz A por su transpuesta para de esta maner obtener una matriz cuadrada como especificamos con anterioridad.
2. Como segundo paso se obtienen los autovalores
3. Se obtienen los autovectores una vez que tenemos los autovalores
4. Con los autovalores y autovectores obtenemos la matriz U

Cálculo de matriz S

1. Utilizar los autovalores y obtener la raiz cuadrada de estos.
2. Los autolvalores formarán la diagonal de la matriz S
3. El resto de los valores que no sean los ubicados en la diagonal serán ceros

A continuación podremos encontrar un ejemplo de la descomposición SVD con una matriz pequeña:

Ejemplo de SVD

Para el presente ejemplo de descomposición en valores singulares se utilizará la siguiente matriz A:

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}$$

1. Cálculo de Valores y Vectores Propios

Para obtener los valores singulares de una matriz A, se debe de encontrar el determinante de la matriz $A^TA - \lambda I$. Con la matriz ejemplo tenemos:

$$A^TA = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

Luego restamos λI:

$$A^TA - \lambda I = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} = \begin{bmatrix} 1-\lambda & 1 \\ 1 & 1-\lambda \end{bmatrix}$$

Y obteniendo el determinante tenemos que:

$$det(A^TA - \lambda I) = (1-\lambda)^2 - 1 = \lambda(\lambda - 2)$$

Y por lo tanto los valores propios son: $\lambda_1 = 0$ y $\lambda_2 = 2$

Por lo que los valores singulares son las correspondientes raices cuadradas de los valores propios: $\sigma_1 = 0$ y $\sigma_2 = \sqrt{2}$

Por otra parte, para obtener los vectores propios se sustituyen los valores propios en la matriz $A^TA - \lambda I$ y se multiplica por el vector que se desea obtener de tal manera que el resultado se un vector nulo. Por lo tanto:

$$(A^TA - \lambda_1 I)V_1 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} V_1$$

$$(A^TA - \lambda_2 I)V_2 = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} V_2$$

E igualando a cero cada ecuacion obtenemos que:

$$V_1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, V_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

2. Descomposición SVD

Un primer paso para obtener la descomposición de $A = U\Sigma V^T$ es normalizar los vectores propios. Por lo tanto:

$$V_1^* = \begin{bmatrix} -1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}, V_2^* = \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}$$

Por lo que V y V^T serían:

$$V = V^T = \begin{bmatrix} -1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$$

Por otra parte, Σ corresponde a la matriz diagonal con los valores singulares de A:

$$\Sigma = \begin{bmatrix} 0 & 0 \\ 0 & \sqrt{2} \end{bmatrix}$$

Y para U, se debe de sustituir los valores singulares y los vectores propios normalizados en la siguiente ecuación:

$$U_i = \frac{1}{\sigma_i} A V_i^*$$

Esto solamente para valores singulares no nulos, por lo que:

$$U_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$U_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Y finalmente sustituyendo tenemos que:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & \sqrt{2} \end{bmatrix} \begin{bmatrix} -1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$$

Método Jacobi

El método Jacobi es utilizado como uno de los métodos base para poder realizar el cómputo de la descomposición SVD y especialmente eficientes para el cómputo en paralelo. Dentro del método Jacobi, se encuentra el "one sided" y el "two sided". El más apropiado para realizar un cómputo en paralelo de SVD sería el utilizar el Hestnes "one sided".

En el paper de B. B. Zhou se puede apreciar un algoritmo para poder obtener la descomposición SVD con el método Jacobi pero asegurando que habrá una gran eficiencia en el momento de hacer el cómputo en paralelo haciendo posible que se realice el ordenamiento y acomodo de los indices en únicamente una pasada obteniendo la misma convergencia final en los resultados

Este algoritmo conste básicamente en los siguientes pasos que escencialmente son dos procedimientos; forward sweep y backward sweep:

1. Los índices se organizan en dos renglones
2. Los índices de cada columna son intercambiados como se muestra en la siguiente imagen. Las flechas verticales indican el intercambio de dos indices antes de intercambiar una columna:
3. Permutar las posiciones iniciales de los `n` índices y luego mostramos como cada paso se generan los mismos pares de índices
4. Ordenar elementos en sentido no-creciente en un sistema en paralelo de P PEs
5. Estableceer una correcta comunicación entre pasos

Función cuSolver

En el siguiente documento se pueden visualizar funciones de CUDA BLAS (CUBLAS) en la codificación del algoritmo, por lo que se profundizó en la búsqueda de las funciones y se encontró la librería cuSOLVER en la siguiente referencia A grandes rasgos la librería cuSOLVER es un paquete de tres librerías que realizan operaciones complejas de algebra lineal. Las librerias son las siguientes: A. cuSolverDN: Dense LAPACK. Entre algunas de sus funciones provee rutinas para bidiagonalización de matrices y para SVD.

Por otro lado, también se consultó la siguiente referencia donde se encuentran algoritmos para la descomposición SVD utilizando las librerías de cuSolver.

Implementación

A continuación se realizará una descripción de los datos que se utilizarán en el proyecto:

Los datos fueron recolectados por "GroupLens Resarch Project" en la Universidad de Minnesota y consisten de lo siguiente:

- 100,000 ratings con calificaciones del 1 al 5
- Calificaciones de 943 usuarios
- Calificaciones de 1682 películas
- Cada usuario calificó por lo menos 20 películas
- Información demográfica de cada usuario (edad, género, ocupación y código postal)

¿Cómo fue recolectada la información?

La información fue recolectada a través del sitio de intenrnet de la compañía MovieLens (movielens.umn.edu) en un periodo de 7 meses de septiembre 19 a abril 22 de 1988.

Los datos fueron limpiados de la siguiente manera:

- Usuarios con menos calificaciones de menos de 20 películas fueron removidos
- Usuarios sin información demográfica completa fueron removidos

Descripción de archivos

1. ml-data.tar.gz -> carpeta comprimida con todos los archivos
2. u.data -> dataset completo numerado del 1 al 100,000. Los datos están ordenados de manera aleatoria separado de la siguiente manera: user id | item id | rating | timestamp
3. u.info -> números de usuarios, items y ratings
4. u.item -> información sobre las películas con las siguientes columnas: movie id | movie title | release date | video release date | IMDb URL | unknown | Action | Adventure | Animation | Children's | Comedy | Crime | Documentary | Drama | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western |
5. u.genere -> lista de géneros de películas
6. u.user -> información demográfica de los usuarios con las siguientes columnas: user id | age | gender | occupation | zip code
7. u.occupation -> lista de ocupaciones
8. u1.base - u5.base y u1.test - u5.test -> esta es una división de 80%/20% donde se divide un set de entrenamiento y un set de prueba de todos los datos
9. ua.base, ua.test, ub.base, ub.test -> esta es una división de prueba y entrenamiento con exactamente 10 ratings por usuario
10. allbut.pl -> es un script que genera los sets de prueba y entrenamiento en porcentajes deseados
11. mku.sh -> un shell script para generar todos los datasets u de u.data

Referencias

1. Ricardo Moya
2. J Villena
3. B. B. Zhou
4. Chrzeszczyk
5. nVidia
6. F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets:History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19 (December 2015), 19 pages. DOI=http://dx.doi.org/10.1145/2827872