CIMAT

Centro de Investigación en Matemáticas, A.C.
Área de Ciencias de la Computación

Nombre de la tesis

Factorización de Matrices para Sistemas de Recomendación

Que para obtener el grado de:
Maestro en Ciencias con Especialidad en
Computación y Matemáticas Industriales

P R E S E N T A

Aristeo Gutiérrez Hernández

DIRECTOR DE TESIS:

Dr. Johan Jozef Lode Van Horebeek

Guanajuato, Gto. a 15 de Diciembre del 2011

Integrantes del jurado:

Presidente: Dr. Mariano José Juan Rivera Meraz

Secretario: Dr. Rogelio Ramos Quiroga

Vocal: Dr. Johan Jozef Lode Van Horebeek

Dr. Johan Jozef Lode Van Horebeek
Asesor

Aristeo Gutiérrez Hernández
Estudiante

Agradecimientos

Hago extenso mi agradecimiento al Consejo Nacional de Ciencia y Tecnología (CONACYT) por haberme permitido realizar mis estudios de maestría mediante el otorgamiento de una beca. Agradezco también a la comunidad CIMAT-DEMAT por haberme acogido y brindado su apoyo siempre que lo necesité.

Un agradecimiento especial a los doctores que me dieron clases durante la maestría, particularmente a aquellos de quienes aprendí muchas cosas aparte de las cuestiones académicas, a los Drs. Johan Van Horebeek, Jean-Bernard Hayet, Mariano J.J. Rivera-Meraz, con gusto puedo presumir de su gran calidad humana. Un agradecimiento también a las personas que me facilitaron material y apoyo para la realización de este trabajo, a Jaime Arellano Roig y Ricardo Martínez Bravo, gracias por el tiempo que dedicaron para proporcionarme los datos que necesité.

Mención especial a mis compañeros de maestría, también amigos, con quienes compartí muchos momentos agradables durante mi estancia en CIMAT. Y naturalmente, muchas gracias a mi familia, mi madre y hermanas, porque siempre me han hecho sentir su apoyo incondicional.

Índice general

1. Sistemas de Recomendación. **1**
1.1. Introducción. 1
1.2. Clasificación. 2
1.2.1. Métodos basados en contenidos. 3
1.2.2. Métodos basados en colaboración. 3

2. Factorización de Matrices para Sistemas de Recomendación. **5**
2.1. Introducción. 5
2.2. Modelo básico de factorización de matrices. 6

3. Factorización No-Negativa de Matrices. **9**
3.1. Introducción. 9
3.1.1. Métodos con reglas de actualización multiplicativas. 10
3.1.2. Mínimos Cuadrados Alternados No-Negativos. 10

4. Factorización Probabilística Bayesiana. **13**
4.1. Introducción. 13
4.2. Factorización Probabilística de Matrices. 13
4.3. Factorización Probabilística Bayesiana. 15

5. Factorización de Matrices con Agrupamiento. **19**
5.1. Introducción. 19
5.2. Optimización. 20
5.3. Pruebas sobre los datos de MovieLens. 21
5.4. Pruebas sobre los datos de la biblioteca del CIMAT. 28

6. Conclusiones y trabajos a futuro. **33**

A. Análisis de los datos de la biblioteca del CIMAT. **35**
A.1. Generalidades. 36
A.2. Categorización sobre la LCC. 36
A.3. Análisis de archivos con base en las categorías propuestas. 40
A.3.1. Fichas 40

A.3.2. Ejemplares 40
A.3.3. Devoluciones 41
A.4. Animaciones 42
A.5. Evolución de las categorías en el archivo de Devoluciones. 43

B. Métodos de Proyección de gradiente. 47
B.1. Alternando Mínimos Cuadrados No-Negativos usando Métodos de Proyección de Gradiente. 48
B.2. Aplicación directa del método de proyección de gradiente a la NMF. 50

C. Inferencia en el modelo BPMF. 53

Capítulo 1

Sistemas de Recomendación.

Hoy en día, la gran cantidad de opciones de productos y de servicios hace difícil la tarea de elegir al consumidor. Es por ello que cada vez más tiendas electrónicas como Amazon y proveedores de contenido como Netflix y Lastfm, han comenzado a interesarse por los sistemas de recomendación, para brindarle al usuario recomendaciones acertadas con base en sus gustos personales.

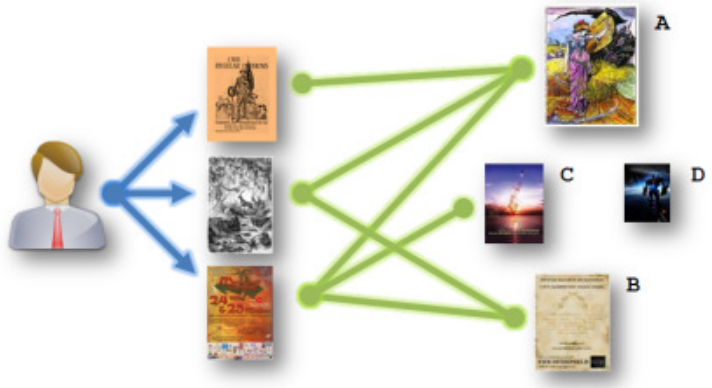


Figura 1.1: Los sistemas de recomendación analizan patrones de interés de los usuarios sobre los productos para realizar las recomendaciones.

1.1. Introducción.

Los sistemas de recomendación tienen sus orígenes en los trabajos de ciencias cognitivas, teoría de aproximación, recuperación de información, teoría de previsión, y en la modelación de preferencias del consumidor en el mercado (Gediminas Adomavicius, Alexander Tuzhilin [6]). Sin embargo, es a mediados de los 90s cuando surgen como un área de investigación

aparte. El problema central de los sistemas de recomendación se reduce a *estimar calificaciones sobre los productos que no han sido valorados por un usuario*, considerando para ello, las calificaciones que el usuario ha dado a otros productos y las calificaciones de usuarios con gustos similares. Sobre las calificaciones estimadas (que el usuario no ha valorado), se recomienda el producto que tenga la calificación más alta.

El problema puede ser formulado como sigue. Sea $\mathcal{C}$ el conjunto de todos los usuarios, y $\mathcal{S}$ el conjunto de todos los productos que pueden ser recomendados. Si definimos u como la función que estima la utilidad del producto s para el usuario c, entonces;

$$u : \mathcal{C} \times \mathcal{S} \to R,$$

donde R es un conjunto totalmente ordenado. Así, para cada $c \in \mathcal{C}$, se desea elegir $s' \in \mathcal{S}$ que maximize la utilidad del usuario c, es decir;

$$\forall c \in \mathcal{C}, \text{ se desea hallar } s'_c = \arg\max_{s \in \mathcal{S}} u(c, s). \tag{1.1.1}$$

Generalmente, la utilidad u está dada explícitamente por el usuario para algunos productos, aunque en algunos casos, la aplicación misma calcula la utilidad con base en las frecuencias, como por ejemplo, la cantidad de préstamos de un libro en una biblioteca, o el número de veces que un usuario visita determinada página web.

Cada usuario puede ser representado con un *perfil* que incluya varias de sus características, tales como edad, género, ingreso, estado civil, etc., aunque en ausencia de tal información puede llevar únicamente un ID (identificador). De manera similar, cada producto se puede representar con un conjunto de características. Por ejemplo, para una aplicación de recomendación de películas, cada película puede ser representado por su título, género, director, actores principales, etc.

El problema principal de los sistemas de recomendación es que la utilidad u no está definida en todo el espacio $\mathcal{C} \times \mathcal{S}$. Por tanto, u necesita ser *extrapolado* al espacio total. Inicialmente las calificaciones están definidas sólo en los productos que han sido calificados previamente por los usuarios. Luego, el motor de recomendación estima las calificaciones de las combinaciones no valoradas usuario/producto y genera recomendaciones apropiadas basadas en estas predicciones.

Se pueden especificar *heurísticas* que definan la función de utilidad para la extrapolación de las calificaciones desconocidas o se puede *estimar* la función de utilidad, optimizando sobre ciertos criterios de desempeño, como el error cuadrático medio.

1.2. Clasificación.

La división principal que se hace sobre los sistemas de recomendación se basa en la información que utiliza para hacer las recomendaciones:

i) **Recomendaciones basadas en contenidos:** Se recomendará al usuario productos que son similares a aquellos que el usuario ha preferido en el pasado.

ii) **Recomendaciones basadas en colaboración:** Se recomendará al usuario los productos que las personas con gustos y preferencias similares hayan preferido en el pasado.

iii) **Aproximaciones híbridas:** Combinación de las dos anteriores.

1.2.1. Métodos basados en contenidos.

En los métodos de recomendación basados en contenidos, la utilidad $u(c, s)$ de un producto s para un usuario c es estimado con base en las utilidades $u(c, s_i)$ asignados por el usuario c a los productos $s_i \in \mathcal{S}$ que son "similares" al producto s.

Entre los problemas principales que se presentan bajo este enfoque están las *limitaciones de contenido*, que refieren al hecho de que estos métodos están limitados por las *características* explícitamente asociadas con los productos. Otro problema que se tiene es la *sobre-especialización*, que ocurre cuando el usuario está limitado a recomendaciones de productos que son similares a aquellos que el usuario ha calificado anteriormente. Por último, se tiene el problema de los *nuevos usuarios*, ya que para que el sistema pueda aprender los gustos del usuario y realizar una buena recomendación, el usuario tiene que haber valorado un número considerable de productos con anterioridad.

1.2.2. Métodos basados en colaboración.

Los sistemas *colaborativos* (o *sistemas de filtrado colaborativo*) realizan la recomendación de un producto a un usuario con base en las calificaciones que otros usuarios han asignado a dicho producto. Es decir, la utilidad $u(c, s)$ del producto s para el usuario c es estimado con base en las utilidades $u(c_j, s)$ asignados al producto s por aquellos usuarios $c_j \in \mathcal{C}$ que son "similares" al usuario c.

Entre los sistemas más conocidos que utilizan este enfoque están el sistema de recomendación de libros de Amazon, el sistema PHOAKS que ayuda a las personas a encontrar información relevante en WWW, y el sistema Jersey que recomienda chistes.

Bajo este enfoque se tiene también el problema de los *nuevos usuarios*, pues para hacer una buena recomendación, el sistema debe primero aprender las preferencias del usuario con base en las calificaciones dadas por él mismo. También se tiene un problema con los *nuevos productos*, ya que hasta que el nuevo producto sea calificado por un número considerable de usuarios, el sistema de recomendación no será capaz de recomendarlo.

El éxito de los sistemas de recomendación colaborativos depende de la disponibilidad de una gran cantidad de usuarios. Una ventaja importante del filtrado colaborativo es que puede capturar aspectos de los datos que generalmente están ocultos o son difíciles de caracterizar usando métodos basados en contenidos.

Capítulo 2

Factorización de Matrices para Sistemas de Recomendación.

2.1. Introducción.

En su forma básica, la factorización de matrices caracteriza tanto a los productos como a los usuarios mediante vectores de factores, los cuales son inferidos de los patrones de calificación sobre los productos. Estos métodos ofrecen una buena combinación de escabilidad con precisión en la predicción, permitiendo la incorporación de información adicional, tales como información implícita, efectos temporales y niveles de confianza. Además, ofrecen mucha flexibilidad para modelar situaciones diversas. Los modelos de factorización de matrices son superiores a las técnicas clásicas del vecino más cercano para producir recomendaciones de productos.

Los datos de entrada más convenientes para un sistema de recomendación es información de alta calidad, que incluya entradas explícitas de usuarios con respecto a su interés en productos. Generalmente la información explícita implica una matriz rala, puesto que es muy probable que un usuario particular haya calificado sólo un porcentaje pequeño de los posibles productos.

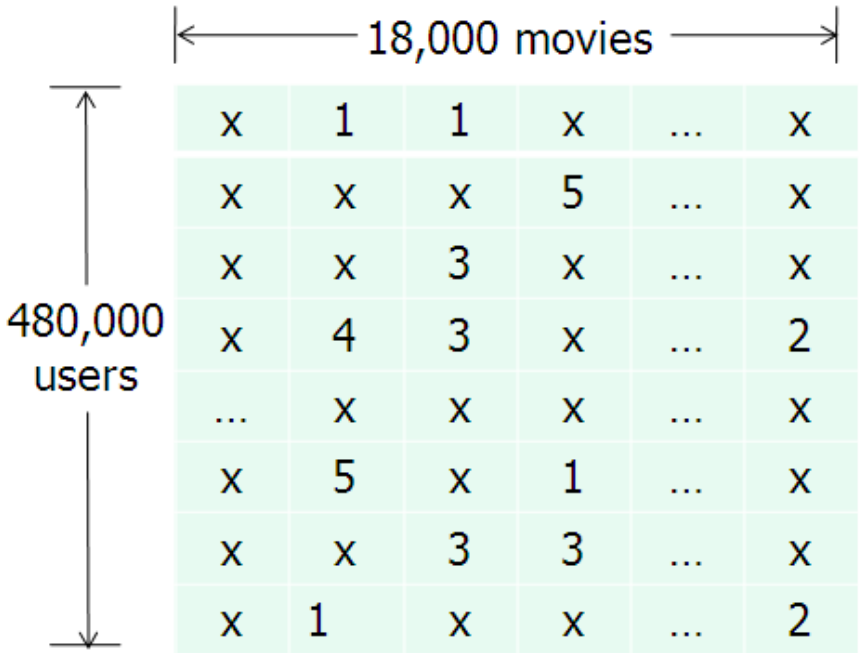


Figura 2.1: El punto de partida de un sistema de recomendación es una matriz rala $R \in \mathbb{R}^{N \times M}$ donde $\mathbf{X}$ indica las calificaciones faltantes.

Cuando no hay disponibilidad de información explícita, los sistemas de recomendación pueden inferir las preferencias de los usuarios usando *información implícita*, el cual refleja indirectamente la opinión del usuario mediante la observación de su comportamiento, incluyendo historial de compras, historial de búsquedas, patrones de búsqueda, etc.

2.2. Modelo básico de factorización de matrices.

Los modelos de factorización de matrices mapean usuarios y productos a un espacio común de factores latentes de dimensión r, modelando la interacción entre usuarios y productos como productos internos en dicho espacio (Koren, Bell and Volinsky [5]). En consecuencia, cada producto j está asociado con un vector $V_j \in \mathbb{R}^r$, y cada usuario i está asociado con un vector $U_i \in \mathbb{R}^r$. Se supone que el producto punto resultante, $U_i^T V_j$, captura la interacción entre el usuario i y el producto j, es decir,

$$R_{ij} = U_i^T V_j. \tag{2.2.1}$$

En general, dada una matriz de datos $R \in \mathbb{R}^{N \times M}$ y un entero pre-especificado $r \ll \text{mín}(N, M)$, el modelo de factorización de matrices busca dos matrices $U \in \mathbb{R}^{r \times N}$ y $V \in \mathbb{R}^{r \times M}$ tales que:

$$R \approx U^T V.$$

Como cada columna de R representa un producto, el modelo de factorización de matrices los aproxima mediante combinaciones lineales de r columnas *base* de U^T.

Una vez estimados U y V, se utiliza la Ec. (2.2.1) para estimar las entradas (calificaciones) faltantes. Si todas las entradas de la matriz R fueran observadas, se podría resolver el problema de factorización usando la descomposición en valores singulares *(SVD)*. En trabajos recientes usualmente se convierte lo anterior en un problema de optimización, donde se trata de minimizar una función de error sobre las entradas observadas:

$$f(U,V) := \sum_{(i,j)\in\mathcal{K}} \left[\left(R_{ij} - U_i^T V_j\right)^2 + \lambda \left(\| U_i \|^2 + \| V_j \|^2 \right) \right] \tag{2.2.2}$$

donde $\mathcal{K}$ es el conjunto de pares (i,j) para los cuales R_{ij} es conocido. La constante λ se introduce para evitar sobreajuste de los datos y para penalizar el crecimiento de las normas de las matrices factores.

Notar que:

$$\sum_{i=1}^{N} \sum_{j=1}^{M} \left(R_{ij} - U_i^T V_j\right)^2 = \| R - U^T V \|_F^2,$$

donde $\| \cdot \|_F$ es la norma de Frobenius.

Dos métodos populares para minimizar la Ec. (2.2.2) son *Descenso Estocástico de Gradiente* (DEG) y *Mínimos Cuadrados Alternados* (ALS, por sus siglas en inglés).

Descenso Estocástico de Gradiente (DEG).

El método de *descenso estocástico de gradiente* es un método de optimización que minimiza una función objetivo f que puede ser escrito como una suma de funciones diferenciables:

$$f(w) := \sum_i f_i(w), \tag{2.2.3}$$

Para minimizar la función (2.2.3), el método del gradiente realiza la siguiente actualización:

$$w = w - \gamma \nabla f(w) = w - \gamma \sum_i \nabla f_i(w),$$

donde γ es un tamaño de paso (factor de aprendizaje).

En algunos casos, la evaluación de todos los términos en la suma de gradientes puede resultar computacionalmente costoso, sobre todo cuando el conjunto de datos es muy grande y la forma del gradiente no es simple. Para economizar sobre el costo computacional, el método DEG muestrea un subconjunto de las funciones presentes en la suma (Ec. (2.2.3)) en cada iteración. Luego, para cada elemento i de la muestra se realiza la siguiente actualización:

$$w = w - \gamma \nabla f_i(w).$$

A continuación el algoritmo general.

Algorithm 1 Algoritmo DEG.

```
Inicializar el vector de parámetros w y el tamaño de paso γ.
repeat
    Escoger aleatoriamente el conjunto de entrenamiento 𝒯.
    for i ∈ 𝒯 do
        w = w − γ∇f_i(w)
    end for
until obtener un mínimo aproximado.
```

En nuestro caso la función objetivo f (Ec. (2.2.2)) se descompone en sumas de funciones diferenciables $f_{ij}(U, V)$, con

$$f_{ij}(U,V) = \begin{cases} (R_{ij} - U_i^T V_j)^2 + \lambda \left(\| U_i \|^2 + \| V_j \|^2 \right), & \text{si } R_{ij} > 0 \\ 0 & \text{en otro caso.} \end{cases}$$

con $1 \leq i \leq N$ y $1 \leq j \leq M$. El método DEG genera las siguientes reglas de actualización:

$$U_i = U_i + \gamma(\mathbf{e}_{ij} V_j - \lambda U_i)$$
$$V_j = V_j + \gamma(\mathbf{e}_{ij} U_i - \lambda V_j)$$

con $\mathbf{e}_{ij} = R_{ij} - U_i^T V_j$ y γ el radio de aprendizaje.

Mínimos Cuadrados Alternados (ALS).

Dado que tanto $\{U_i\}_{i=1}^N$ como $\{V_j\}_{j=1}^M$ son desconocidos, la Ec. (2.2.2) es no-convexa. Sin embargo, si fijamos alguno de los factores, el problema de optimización se vuelve cuadrático y puede ser resuelto eficientemente. Por lo anterior, el método ALS minimiza la Ec. (2.2.2) fijando alternadamente $\{U_i\}_{i=1}^N$ y $\{V_j\}_{j=1}^M$.

Algorithm 2 Mínimos Cuadrados Alternados.

Inicializar U_{ia}^1 y V_{bj}^1 aleatoriamente, para $1 \leq i, b \leq N$ y $1 \leq a, j \leq M$.

for $k = 1, 2, \ldots$ **do**

$$U^{k+1} = \arg\min_U f(U, V^k), \tag{2.2.4}$$
$$V^{k+1} = \arg\min_V f(U^{k+1}, V). \tag{2.2.5}$$

end for

Capítulo 3

Factorización No-Negativa de Matrices.

3.1. Introducción.

La factorización no-negativa de matrices (NMF) consiste en buscar una factorización de R en función de las matrices U y V, bajo la restricción de que los elementos de U y V sean no-negativos:

$$\begin{aligned} \min_{U,V} \quad & f(U,V) \equiv \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}\left(R_{ij} - U_i^T V_j\right)^2 \\ \text{sujeto a:} \quad & U_{ik} \geq 0, V_{kj} \geq 0, \quad \forall i,k,j. \end{aligned} \tag{3.1.1}$$

Así se tiene un problema estándar de optimización con restricciones de frontera. A continuación se presentan algunos métodos y se mencionan algunas observaciones importantes del problema de NMF (3.1.1).

El gradiente de la función $f(U,V)$ consiste de dos partes:

$$\nabla_U f(U,V) = V(V^T U - R^T) \qquad \text{y} \qquad \nabla_V f(U,V) = U(U^T V - R), \tag{3.1.2}$$

los cuales son, respectivamente, las derivadas parciales de los elementos en U y V. De las condiciones de optimalidad de Karush-Kuhn-Tucker (KKT), (U,V) es un punto estacionario de (3.1.1) si y sólo si:

$$\begin{gathered} U_{ik} \geq 0, V_{kj} \geq 0, \\ \nabla_U f(U,V)_{ik} \geq 0, \nabla_V f(U,V)_{kj} \geq 0, \\ U_{ik} \cdot \nabla_U f(U,V)_{ik} = 0, V_{kj} \cdot \nabla_V f(U,V)_{kj} = 0, \quad \forall i,k,j. \end{gathered} \tag{3.1.3}$$

Los métodos de optimización para la NMF producen una secuencia $\{U^k, V^k\}_{k=1}^{\infty}$ de iteraciones. El problema (3.1.1) es no-convexo y puede tener varios mínimos locales. Existen varios métodos para la NMF, sin embargo, las restricciones de frontera no son manejadas rigurosamente.

3.1.1. Métodos con reglas de actualización multiplicativas.

La aproximación más usada para minimizar (3.1.1) es un método simple con reglas de actualización multiplicativas propuesto por Lee y Seung (2001) [1].

Algorithm 3 Actualización Multiplicativa

1: Inicializar $U^1_{ia} > 0, V^1_{bj} > 0, \quad \forall i, a, b, j.$
2: **for** $k = 1, 2, \ldots$ **do**
3:

$$U^{k+1}_{ia} = U^k_{ia} \frac{(V^k R^T)_{ia}}{(V^k (V^k)^T U^k)_{ia}} \quad \forall i, a \tag{3.1.4}$$

$$V^{k+1}_{bj} = V^k_{bj} \frac{(U^{k+1} R)_{bj}}{(U^{k+1} (U^{k+1})^T V^k)_{bj}} \quad \forall b, j \tag{3.1.5}$$

4: **end for**

Este algoritmo es un tipo de método de punto fijo. Si $(V^k (V^k)^T U^k)_{ia} \neq 0$ y U^k_{ia} no cambia en la siguiente iteración, es decir, $U^{k+1}_{ia} = U^k_{ia} > 0$, entonces $(V^k R^T)_{ia} = (V^k (V^k)^T U^k)_{ia}$ implica que $\nabla_U f(U^k, V^k)_{ia} = 0$, el cual es parte de las condiciones KKT (3.1.3).

Lee y Seung [1] han mostrado que el valor de la función objetivo es no-creciente después de cada actualización. Para que el Algoritmo (3) esté bien definido, se debe asegurar que los denominadores en (3.1.4) y (3.1.5) son estrictamente positivos. Por otro lado, si $U^k_{ia} = 0$ en la k-ésima iteración, entonces $U_{ia} = 0$ en todas las iteraciones subsecuentes. Por tanto, se debería cumplir que $U^k_{ia} > 0$ y $V^k_{bj} > 0, \ \forall k$. El Teorema 3.1.1 (Chih-Jen Lin [2]) explica cuando esta propiedad se cumple:

Teorema 3.1.1. *Si R no tiene renglones ni columnas cero, y $U^1_{ia} > 0$ y $V^1_{bj} > 0, \forall i, a, b, j$, entonces:*

$$U^k_{ia} > 0 \text{ y } V^k_{bj} > 0, \forall i, a, b, j, \forall k \geq 1.$$

3.1.2. Mínimos Cuadrados Alternados No-Negativos.

El Algoritmo (3) es un caso especial de un esquema más general, el cual alternadamente fija una matriz y mejora la otra:

- Encuentra U^{k+1} tal que $f(U^{k+1}, V^k) \leq f(U^k, V^k)$, y
- Encuentra V^{k+1} tal que $f(U^{k+1}, V^{k+1}) \leq f(U^{k+1}, V^k)$.

Esta aproximación es conocida como el método de *descenso coordenado por bloques* en optimización con restricciones de frontera, donde escencialmente un bloque de variables es minimizado bajo las restricciones correspondientes y los bloques restantes se mantienen fijos. Para NMF, se tienen sólo dos bloques de variables U y V. Las ecuaciones (3.1.6) y (3.1.7)

Algorithm 4 Mínimos Cuadrados Alternados No-Negativos

1: Inicializar $U^1_{ia} \geq 0, V^1_{bj} \geq 0, \quad \forall i, a, b, j.$
2: **for** $k = 1, 2, \ldots$ **do**
3:

$$U^{k+1} = \arg\min_{U \geq 0} f(U, V^k), \tag{3.1.6}$$

$$V^{k+1} = \arg\min_{V \geq 0} f(U^{k+1}, V). \tag{3.1.7}$$

4: **end for**

serán referidas como subproblemas del Algoritmo (4). Cuando un bloque de variables permanece fijo, se puede ver que un subproblema es la colección de varios problemas de mínimos cuadrados no-negativos. Por ejemplo, de la Ec. (3.1.7), V^{k+1}_j, la j-ésima columna de la matriz V^{k+1} está dada por:

$$V^{k+1}_j = \min_{v \geq 0} \| R_j - (U^{k+1})^T v \|^2, \tag{3.1.8}$$

donde v es un vector variable de dimensión r. Se propone usar métodos de proyección de gradiente como método eficiente para resolver los subproblemas del el algoritmo anterior. Aunque los subproblemas no necesariamente tienen solución única (por no ser estrictamente convexos), se obtiene el siguiente resultado (Chih-Jen Lin [2]):

Teorema 3.1.2. *Cualquier punto límite de la secuencia $\{U^k, V^k\}$ generado por el Algoritmo (4) es un punto estacionario de (3.1.1).*

Resta asegurar que la secuencia $\{U^k, V^k\}$ tiene al menos un punto límite. Esta propiedad se sigue siempre que la región factible esté acotada, y aunque la región definida por las restricciones $U_{ia} \geq 0$ y $V_{bj} \geq 0$ no está acotada por arriba, se pueden agregar cotas superiores a todas las variables en (3.1.1), sin que ello afecte los resultados. Con esta modificación, el problema original sigue siendo un problema con restricciones de frontera. Por tanto, se puede aplicar el Algoritmo (4) y el Teorema 3.1.2 se sigue cumpliendo.

En resumen, contrario al Algoritmo (3), que carece de resultados de convergencia, el Algoritmo (4) tiene buenas propiedades de optimización. En el Apéndice B se puede ver con más detalle el método de proyección de gradiente, así como su aplicación a la factorización no-negativa de matrices.

Capítulo 4

Factorización Probabilística Bayesiana.

4.1. Introducción.

El entrenamiento del modelo de factorización de matrices, $R \approx U^T V$, consiste en hallar la mejor aproximación de rango r de la matriz objetivo observada $R \in \mathbb{R}^{N \times M}$ bajo una función de pérdida dada.

En este capítulo presentamos en primer lugar una breve descripción del enfoque probabilístico de la factorización de matrices, y posteriormente se aborda el enfoque probabilístico Bayesiano, el cual utiliza métodos de Cadenas de Markov Monte Carlo (MCMC) para realizar la inferencia.

Generalmente no se recomienda usar métodos de MCMC en problemas de gran escala por su lentitud. Sin embargo, MCMC arroja buenos resultados sobre conjuntos de datos muy grandes, esparcidos e imbalanceados, como los ofrecidos por Netflix.

4.2. Factorización Probabilística de Matrices.

La Factorización Probabilística de Matrices (PMF) (Salakhutdinov-Mnih [3]) es un modelo lineal probabilístico con ruido Gaussiano en las observaciones. Supóngase que tenemos N usuarios y M productos. Sean

- R_{ij} la calificación del usuario i sobre el producto j,
- U_i vector columna de características del usuario i,
- V_j vector columna de características del producto j,

con U_i y V_j vectores de dimensión r.

La distribución condicional sobre las calificaciones observadas $R \in \mathbb{R}^{N\times M}$ (el término de verosimilitud) y la distribución a priori sobre $U \in \mathbb{R}^{r\times N}$ y $V \in \mathbb{R}^{r\times M}$ están dadas por:

$$p(R|U,V,\alpha) = \prod_{i=1}^{N}\prod_{j=1}^{M}\left[\mathcal{N}(R_{ij}|U_i^T V_j, \alpha^{-1})\right]^{I_{ij}}, \tag{4.2.1}$$

$$p(U|\alpha_U) = \prod_{i=1}^{N}\mathcal{N}(U_i|0, \alpha_U^{-1} I), \tag{4.2.2}$$

$$p(V|\alpha_V) = \prod_{j=1}^{M}\mathcal{N}(V_j|0, \alpha_V^{-1} I), \tag{4.2.3}$$

donde $\mathcal{N}(x|\mu,\alpha)$ denota la distribución Gaussiana con media μ y precisión α, e

$$I_{ij} = \begin{cases} 1, & \text{si el usuario } i \text{ valoró el producto } j, \\ 0, & \text{en otro caso.} \end{cases}$$

El entrenamiento en este modelo se realiza mediante la maximización de la distribución *log-posterior* sobre las características del producto y del usuario, con los hiperparámetros fijos (i.e. la varianza del ruido observado y las varianzas apriori):

$$\log p\left(U,V|R,\alpha,\alpha_V,\alpha_U\right) = \log p\left(R|U,V,\alpha\right) + \log p\left(U|\alpha_U\right) + \log p\left(V|\alpha_V\right) + C,$$

donde C es una constante que no depende de los parámetros. Dado que:

$$\begin{aligned} p(R|U,V,\alpha) &= \prod_{i=1}^{N}\prod_{j=1}^{M}\left[\mathcal{N}(R_{ij}|U_i^T V_j, \alpha^{-1})\right]^{I_{ij}} \\ &= \prod_{i=1}^{N}\prod_{j=1}^{M}\left[\exp\left(-\frac{\alpha}{2}\left(R_{ij} - U_i^T V_j\right)^2\right)\right]^{I_{ij}} \\ &= \exp\left(-\frac{\alpha}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}(R_{ij} - U_i^T V_j)^2\right), \end{aligned}$$

$$\begin{aligned} p(U|\alpha_U) &= \prod_{i=1}^{N}\mathcal{N}(U_i|0, \alpha_U^{-1} I) \\ &= \prod_{i=1}^{N}\exp\left(-\frac{\alpha_U}{2}U_i^T I U_i\right) \\ &= \prod_{i=1}^{N}\exp\left(-\frac{\alpha_U}{2}\| U_i \|^2\right) \\ &= \exp\left(-\frac{\alpha_U}{2}\sum_{i=1}^{N}\| U_i \|^2\right) \end{aligned}$$

y

$$
\begin{aligned}
p(V|\alpha_V) &= \prod_{j=1}^{M} \mathcal{N}(V_j|0, \alpha_V^{-1} I) \\
&= \exp\left(-\frac{\alpha_V}{2} \sum_{j=1}^{M} \| V_j \|^2\right)
\end{aligned}
$$

se puede ver que la maximización de esta distribución posterior con respecto a U y V equivale a minimizar la función de error de la suma de cuadrados con términos cuadráticos de regularización:

$$
\begin{aligned}
E &= \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{M} I_{ij} (R_{ij} - U_i^T V_j)^2 \\
&+ \frac{\lambda_U}{2} \sum_{i=1}^{N} \| U_i \|_{Fro}^2 + \frac{\lambda_V}{2} \sum_{j=1}^{M} \| V_j \|_{Fro}^2, \qquad (4.2.4)
\end{aligned}
$$

donde $\lambda_U = \frac{\alpha_U}{\alpha}$, $\lambda_V = \frac{\alpha_V}{\alpha}$, y $\| \cdot \|_{Fro}$ denota la norma de Frobenius.

4.3. Factorización Probabilística Bayesiana.

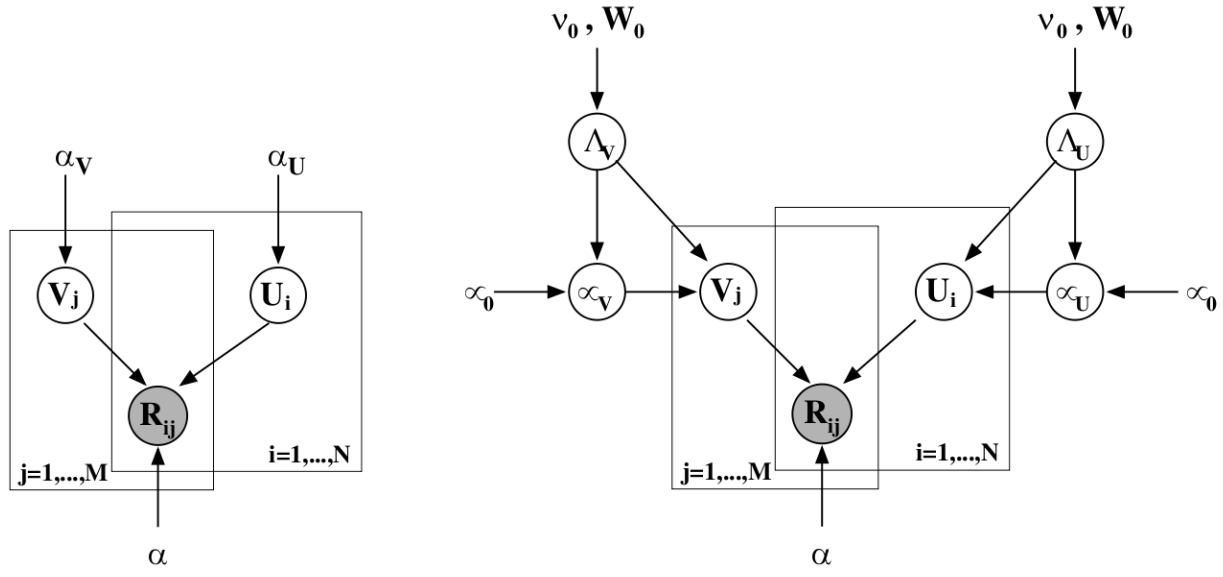


Figura 4.1: Modelos gráficos para PMF y BPMF, respectivamente.

El modelo PFM presentado en la sección 4.2 se resume en la parte izquierda de la Fig. (4.1). El modelo que se presenta en esta sección; Factorización Probabilística Bayesiana de Matrices (BPFM) (Salakhutdinov-Mnih [3]) se muestra en la parte derecha de la misma. El término de verosimilitud de las calificaciones observadas sigue siendo la Ec. (4.2.1), sin embargo, las distribuciónes apriori sobre los vectores de características del usuario y del producto tienen las siguientes formas:

$$p(U|\alpha_U, \Lambda_U) = \prod_{i=1}^{N} \mathcal{N}(U_i|\alpha_U, \Lambda_U^{-1}), \tag{4.3.1}$$

$$p(V|\alpha_V, \Lambda_V) = \prod_{i=1}^{M} \mathcal{N}(V_i|\alpha_V, \Lambda_V^{-1}). \tag{4.3.2}$$

donde se han introducido distribuciones apriori Gaussian-Wishart sobre los hiperparámetros de los usuarios y de los productos $\Theta_U = \{\alpha_U, \Lambda_U\}$ y $\Theta_V = \{\alpha_V, \Lambda_V\}$:

$$\begin{aligned} p(\Theta_U|\Theta_0) &= p(\alpha_U|\Lambda_U)p(\Lambda_U) \\ &= \mathcal{N}(\alpha_U|\alpha_0, (\beta_0\Lambda_U)^{-1})\mathcal{W}(\Lambda_U|W_0, \nu_0), \end{aligned} \tag{4.3.3}$$

$$\begin{aligned} p(\Theta_V|\Theta_0) &= p(\alpha_V|\Lambda_V)p(\Lambda_V) \\ &= \mathcal{N}(\alpha_V|\alpha_0, (\beta_0\Lambda_V)^{-1})\mathcal{W}(\Lambda_V|W_0, \nu_0). \end{aligned} \tag{4.3.4}$$

Aquí $\mathcal{W}$ es la distribución de Wishart con ν_0 grados de libertad y una matriz de escala W_0 de tamaño $D \times D$:

$$\mathcal{W}(\Lambda|W_0, \nu_0) = \frac{1}{C}|\Lambda|^{(\nu_0 - D - 1)/2} \exp\left(-\frac{1}{2}Tr(W_0^{-1}\Lambda)\right),$$

donde C es una constante de normalización. Por conveniencia, se define también $\Theta_0 = \{\alpha_0, \nu_0, W_0\}$.

Predicciones

La distribución de predicción de las calificaciones R_{ij}^* para el usuario i con respecto a un producto j (no valorado por el usuario), es obtenido mediante la marginalización sobre los parámetros e hiperparámetros:

$$\begin{aligned} p(R_{ij}^*|R, \Theta_0) = &\int\int p(R_{ij}^*|U_i, V_j)p(U, V|R, \Theta_U, \Theta_V) \\ &p(\Theta_U, \Theta_V|\Theta_0)d\{U, V\}d\{\Theta_U, \Theta_V\}. \end{aligned} \tag{4.3.5}$$

sin embargo, la evaluación exacta de ésta distribución de predicción es analíticamente intratable debido a la complejidad de la distribución posterior, por tanto, hay que considerar aproximaciones para la inferencia.

Los métodos basados en MCMC usan la aproximación de Monte Carlo para la distribución predictiva de la Ec. (4.3.5) dada por:

$$p(R_{ij}^*|R, \Theta_0) \approx \frac{1}{K}\sum_{k=1}^{K} p(R_{ij}^*|U_i^{(k)}, V_j^{(k)}), \tag{4.3.6}$$

Las muestras $\{U_i^{(k)}, V_j^{(k)}\}$ son generadas mediante corridas de una cadena de Markov cuya distribución estacionaria es la distribución posterior sobre los parámetros e hiperparámetros $\{U, V, \Theta_U, \Theta_V\}$ del modelo. En el Apéndice C se muestran los detalles de las distribuciones sobre los que se aplica el muestreo de Gibbs, presentamos a continuación el esquema general.

Algorithm 5 Muestreo de Gibbs para BPMF

1: Inicializar los parámetros del modelo $\{U^1, V^1\}$
2: **for** $t = 1, \ldots, T$ **do**
3: Muestrar los hiperparámetros Ec. (C.0.4):

$$\begin{aligned} \Theta_U^t &\sim p(\Theta_U | U^t, \Theta_0) \\ \Theta_V^t &\sim p(\Theta_V | V^t, \Theta_0) \end{aligned}$$

4: Para cada $i = 1, \ldots, N$ muestrar las características de los usuarios Ec. (C.0.1):

$$U_i^{t+1} \sim p(U_i | R, V^t, \Theta_U^t)$$

5: Para cada $i = 1, \ldots, M$ muestrar las características de los productos:

$$V_i^{t+1} \sim p(V_i | R, U^{t+1}, \Theta_V^t)$$

6: **end for**

Capítulo 5

Factorización de Matrices con Agrupamiento.

5.1. Introducción.

En un modelo básico de factorización de matrices, se desea aproximar la matriz de calificaciones R mediante el producto de dos matrices de rango bajo $U \in \mathbb{R}^{r\times N}$ y $V \in \mathbb{R}^{r\times M}$, es decir,

$$R \approx U^T V,$$

mientras se minimiza la función de error:

$$f(U,V) := \sum_{(i,j)\in\mathcal{K}} \left(R_{ij} - U_i^T V_j\right)^2 + \lambda \left(\| U_i \|^2 + \| V_j \|^2\right)$$

La penalización sobre las normas de las matrices surge de manera natural bajo el enfoque probabilístico (sección 4.2), sin embargo, es común agregar estos términos en la función objetivo para regular el crecimiento de valores de las matrices. En este trabajo proponemos una extensión del modelo básico de factorización de matrices (Ec. 2.2.2). Nuestra propuesta consiste en incluír más información, agregando términos de agrupamiento sobre usuarios y productos. Para ello vamos a suponer que:

1) los usuarios pertenecen a grupos (desconocidos) y,

2) dentro de cada grupo, las preferencias son similares.

3) los productos pertenecen a grupos (desconocidos) y,

4) dentro de cada grupo, los productos son similares.

Con los términos de agrupamiento, el nuevo modelo es el siguiente:

$$\begin{aligned} f(U,V,\mu^U,\mu^V,\Phi,\Psi) \;:=\; & \sum_{R_{ij}>0} (R_{ij} - U_i^T V_j)^2 + \lambda_1 \left(\| U_i \|^2 + \| V_j \|^2\right) + \\ & \lambda_2 \| U_i - \mu^U_{\Phi(i)} \|^2 + \lambda_3 \| V_j - \mu^V_{\Psi(j)} \|^2, \end{aligned} \tag{5.1.1}$$

donde μ^U y μ^V son matrices que contienen los centroides de los grupos (clusters), y Φ y Ψ son funciones auxiliares que mapean usuarios y productos a los índices de los grupos a los que pertenencen, respectivamente.

Con los términos de agrupamiento se agregan dos parámetros de penalización (λ_2 y λ_3) al modelo básico de factorización de matrices. También se deben considerar los parámetros implícitos correspondientes al número de grupos de usuarios y productos (k_U y k_V, respectivamente), así como el rango r de las matrices factores U y V.

5.2. Optimización.

Cualquier método de optimización que se utilize para minimizar la Ec. (5.1.1) necesita definir los grupos de usuarios y de productos (Φ y Ψ), así como los centroides de los grupos (μ^U y μ^V) previamente. Así, la resolución del problema de optimización de la Ec. (5.1.1) sobre las viariables U,V,μ^U,μ^V,Φ y Ψ se realiza en dos pasos.

1) En el primer paso, se mantienen fijos U y V, y se optimiza sobre μ^U, μ^V, Φ y Ψ. Para ello se aplica el algoritmo de agrupamiento k-medias sobre usuarios y sobre productos;

   - Se calculan los grupos de usuarios (Φ) fijando los centroides (μ^U). Y luego se actualizan los centroides μ^U con base en los grupos obtenidos.
   - Se calculan los grupos de productos (Ψ) fijando μ^V. Y luego se actualizan los centroides μ^V con base en los grupos obtenidos.

2) Una vez que se tienen definidos los grupos y los centroides, se procede a aplicar el método de optimización DEG (sección 2.2) sobre U, V, μ^U y μ^V, solamente para las entradas observadas ($R_{ij} > 0$) en la matriz de calificaciones R:

$$
\begin{aligned}
U_i &= U_i + \gamma(V_j \mathbf{e}_{ij} - \lambda_1 U_i - \lambda_2(U_i - \mu^U_{\Phi(i)})) \\
V_j &= V_j + \gamma(U_i \mathbf{e}_{ij} - \lambda_1 V_j - \lambda_3(V_j - \mu^V_{\Psi(j)})) \\
\mu^U_{\Phi(i)} &= \mu^U_{\Phi(i)} + \gamma\lambda_2(U_i - \mu^U_{\Phi(i)})) \\
\mu^V_{\Psi(j)} &= \mu^V_{\Psi(j)} + \gamma\lambda_3(V_j - \mu^V_{\Psi(j)}))
\end{aligned}
$$

Se repite este procedimiento un número de iteraciones.

Dado que el método de optimización (DEG) para resolver el problema de la Ec. (5.1.1) se combina con un método de agrupamiento (k-medias), se ha bautizado el método de solución como **DEGA**. En el Algoritmo (6) se puede ver el pseudocódigo.

Algorithm 6 Algoritmo DEGA.

1: Especificar el número de grupos de usuarios k_U y de productos k_V, el rango r de las matrices U y V, el factor de aprendizaje γ, y los parámetros de penalización λ_1, λ_2 y λ_3.

2: Inicializar las matrices factores U y V en el intervalo $(0, 1)$.
3: Inicializar las centros de los grupos μ^U y μ^V con columnas de U y V, aleatoriamente.
4: **for** $i = 1, 2, \ldots,$iterDEGA **do**
5: Correr el algoritmo k-medias sobre los usuarios (columnas de U) un número IterKmeans de iteraciones.
6: Correr el algoritmo k-medias sobre los productos (columnas de V) un número IterKmeans de iteraciones.
7: Para cada calificación ($R_{ij} > 0$) aplicar las siguientes actualizaciones:

$$\begin{aligned} U_i &= U_i + \gamma(V_j \mathbf{e}_{ij} - \lambda_1 U_i - \lambda_2(U_i - \mu^U_{\Phi(i)})) \\ V_j &= V_j + \gamma(U_i \mathbf{e}_{ij} - \lambda_1 V_j - \lambda_3(V_j - \mu^V_{\Psi(j)})) \\ \mu^U_{\Phi(i)} &= \mu^U_{\Phi(i)} + \gamma\lambda_2(U_i - \mu^U_{\Phi(i)})) \\ \mu^V_{\Psi(j)} &= \mu^V_{\Psi(j)} + \gamma\lambda_3(V_j - \mu^V_{\Psi(j)})) \end{aligned}$$

donde $\mathbf{e}_{ij} = R_{ij} - U_i^T V_j$.
8: **end for**

El costo computacional de agregar los términos de agrupamiento en la función objetivo original para el método DEG, es el tiempo de ejecución del algoritmo de agrupamiento k-medias, así como las actualizaciones de los centros de los grupos.

5.3. Pruebas sobre los datos de MovieLens.

Descripción de los datos de MovieLens.

El conjunto de datos MovieLens [1] fue recolectado por un grupo de colaboradores de la Universidad de Minnesota a través de su website [2], durante el periodo Septiembre de 2007 - Abril de 2008. Este conjunto de datos contiene $100,000$ calificaciones (1-5) de 943 usuarios y 1682 películas. Cada usuario ha valorado por lo menos 20 películas. Tanto los usuarios como las películas son numerados consecutivamente empezando de 1. Los datos están aleatoriamente colocados y vienen en el siguiente formato:

id usuario	**id película**	**calificación**	**marca de tiempo**

donde las marcas de tiempo están en unidades de segundo a partir de la fecha 01/01/1970 UTC, sin embargo, no se utiliza en nuestros algoritmos.

[1] `http://www.grouplens.org/node/12`
[2] `www.movielens.umn.edu`

En la Fig. 5.1 presentamos una gráfica para ilustrar la densidad de calificaciones de los usuarios. Los puntos negros representan calificaciones (1-5) que ciertos usuarios han hecho sobre ciertas películas y el fondo blanco representa la ausencia de calificaciones.

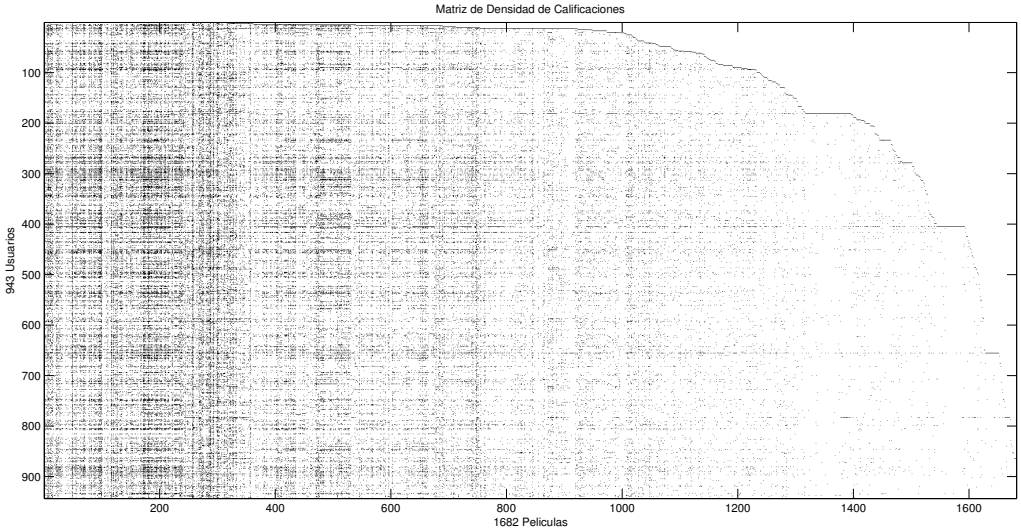


Figura 5.1: Densidad de la matriz de calificaciones.

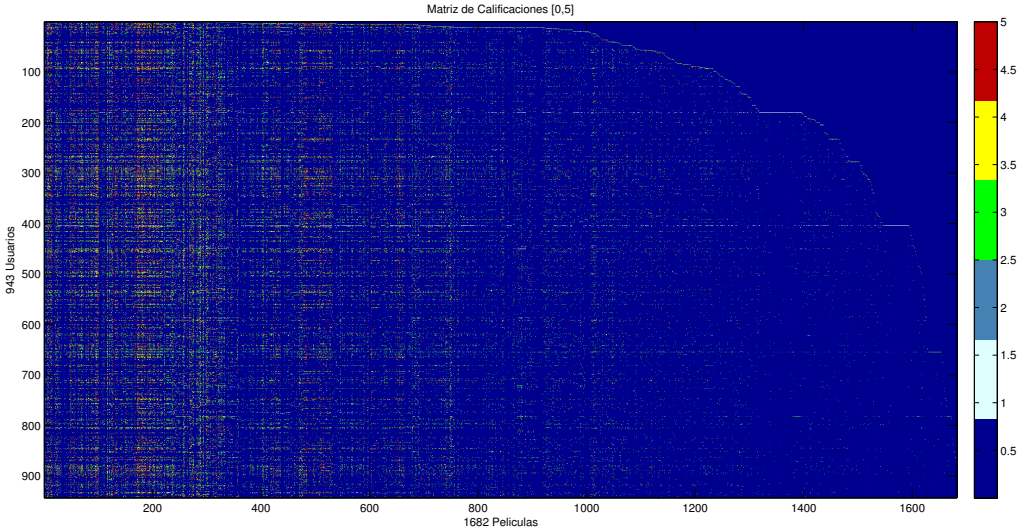


Figura 5.2: El fondo azul fuerte (0) corresponde a las calificaciones faltantes.

Se puede apreciar que hay mayor densidad de calificaciones en las primeras columnas, y que ésta va desapareciendo conforme se avanza hacia las últimas columnas. Se advierte también una especie de línea divisoria en las últimas columnas, conforme se avanza con los usuarios. Posiblemente se debe al tiempo de lanzamiento de las películas, puesto que los primeros usuarios no pueden valorar películas que aún no han salido. En la Fig. (5.2) presentamos también una matriz de colores, donde a cada calificación (1-5) se asigna un color diferente y el fondo azul corresponde a la ausencia de calificaciones (0).

Se particionan los datos en dos conjuntos; un conjunto de entrenamiento y un conjunto de prueba (0.8/0.2 respectivamente). Se repite este procedimiento 30 veces, para generar varios conjuntos de entrenamiento y prueba complementarios. Se asegura que los usuarios cuenten con un mínimo de 10 calificaciones en el conjunto de entrenamiento, sin embargo, no todas las películas cuentan con una calificación.

Los parámetros que se pueden variar en el método DEGA son los siguientes:

- k_U: Número de grupos sobre los usuarios U.
- k_V: Número de grupos sobre los productos V.
- r: Rango de las matrices factores U y V.
- γ: Factor de aprendizaje.
- λ_1: Penalización sobre las normas de las matrices factores U y V.
- λ_2: Penalización sobre el término de agrupamiento de usuarios.
- λ_3: Penalización sobre el término de agrupamiento de productos.
- IterDEGA: Número máximo de iteraciones del método DEGA.
- IterKmeans: Número máximo de iteraciones del método K-medias.

Los valores de los parámetros dependen de la aplicación, del número de usuarios y del número de productos. Por otro lado, se desea que r sea mucho menor que el número de usuarios y el número de productos. El factor de aprendizaje γ generalmente oscila en el intervalo $(0, 1)$. Igualmente los parámetros de penalización λ_1, λ_2 y λ_3.

Elección de valores de los parámetros r, γ, λ_1 para el método DEG.

En la Fig. (5.3) se muestra una tabla con los valores que se propusieron para los parámetros que afectan al método DEG. Primero se varía el rango r de las matrices factores (líneas 1-14), posteriormente se varían los valores del factor de aprendizaje γ (líneas 15-28), y los valores del parámetro de penalización λ_1 (líneas 29-48). Y por último se combinan las mejores asignaciones del rango r y el factor de aprendizaje γ y se prueban más valores para el parámetro de penalización λ_1 (líneas 49-68).

N Par	r	γ	λ_1
1	3	0.0020	0.1000
2	5	0.0020	0.1000
3	8	0.0020	0.1000
4	10	0.0020	0.1000
5	12	0.0020	0.1000
6	14	0.0020	0.1000
7	17	0.0020	0.1000
8	20	0.0020	0.1000
9	23	0.0020	0.1000
10	25	0.0020	0.1000
11	28	0.0020	0.1000
12	32	0.0020	0.1000
13	35	0.0020	0.1000
14	40	0.0020	0.1000
15	10	0.0400	0.1000
16	10	0.0200	0.1000
17	10	0.0090	0.1000
18	10	0.0070	0.1000
19	10	0.0050	0.1000
20	10	0.0030	0.1000
21	10	0.0010	0.1000
22	15	0.0400	0.1000
23	15	0.0200	0.1000
24	15	0.0090	0.1000
25	15	0.0070	0.1000
26	15	0.0050	0.1000
27	15	0.0030	0.1000
28	15	0.0010	0.1000
29	10	0.0020	0.9000
30	10	0.0020	0.5000
31	10	0.0020	0.1000
32	10	0.0020	0.0900
33	10	0.0020	0.0700
34	10	0.0020	0.0300

N Par	r	γ	λ_1
35	10	0.0020	0.0100
36	10	0.0020	0.0090
37	10	0.0020	0.0050
38	10	0.0020	0.0020
39	15	0.0020	0.9000
40	15	0.0020	0.5000
41	15	0.0020	0.1000
42	15	0.0020	0.0900
43	15	0.0020	0.0700
44	15	0.0020	0.0300
45	15	0.0020	0.0100
46	15	0.0020	0.0090
47	15	0.0020	0.0050
48	15	0.0020	0.0020
49	15	0.0070	0.7000
50	15	0.0070	0.5000
51	15	0.0070	0.3000
52	15	0.0070	0.2000
53	15	0.0070	0.1000
54	15	0.0070	0.0900
55	15	0.0070	0.0800
56	15	0.0070	0.0700
57	15	0.0070	0.0600
58	15	0.0070	0.0500
59	10	0.0070	0.7000
60	10	0.0070	0.5000
61	10	0.0070	0.3000
62	10	0.0070	0.2000
63	10	0.0070	0.1000
64	10	0.0070	0.0900
65	10	0.0070	0.0800
66	10	0.0070	0.0700
67	10	0.0070	0.0600
68	10	0.0070	0.0500

Figura 5.3: Parámetros con los que se probó el método DEG.

Se realizaron 50 repeticiones del método DEG sobre cada elección de parámetros de la Fig. (5.3), considerando un total de 50 iteraciones para el ciclo principal del método DEG. Se muestra una gráfica con los errores promediados para cada elección de parámetros en la Fig. (5.4). Se puede observar que para las variaciones de los tres parámetros, los errores van disminuyendo conforme se avanza en las asignaciones, hasta que vuelven a aumentar, lo que nos da una idea de los valores que hay que tomar.

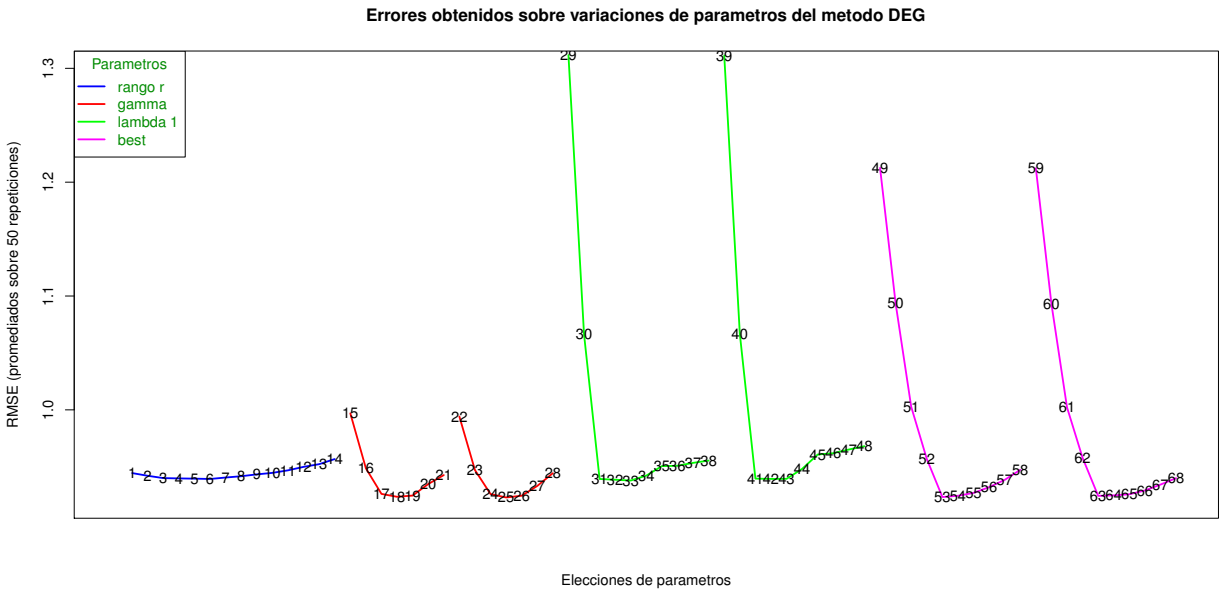


Figura 5.4: Errores obtenidos con el método DEG para cada asignación de valores de los parámetros de la tabla 5.4.

Los valores de los parámetros que corresponden a las líneas 6,18,25,33 y 43 de la tabla que se muestra en la Fig. 5.3 son los que arrojan errores más bajos para cada variación de parámetros. Por esta razón se combinan estos valores, para ver si es posible reducir el error cuadrático medio (RMSE), generando así las asignaciones 49-69 de la Fig. 5.3.

N Par	r	γ	λ_1
6	14	0.0020	0.1000
18	10	0.0070	0.1000
25	15	0.0070	0.1000
33	10	0.0020	0.0700
43	15	0.0020	0.0700

Figura 5.5: Valores que arrojaron los errores más bajos en la variación de cada parámetro.

Con los resultados anteriores se puede concluír que los valores de los parámetros que ofrecen los errores más bajos para el método DEG sobre los datos de MovieLens son los siguientes:

$$r = 15, \gamma = 0.0070, \lambda_1 = 0.1000.$$

Elección de valores de los parámetros k_U, k_V, λ_2, λ_3 para el método DEGA.

Sobre los mejores valores encontrados para los parámetros del método DEG se variaron los parámetros que afectan al método DEGA. En la tabla de la Fig. (5.6) se muestran los valores asignados. Se corrió el algoritmo DEGA sobre 30 conjuntos de datos, con los parámetros que

N Par	k_U	k_V	r	γ	λ_1	λ_2	λ_3
1	3	3	15	0.0070	0.1000	0.0100	0.0100
2	5	5	15	0.0070	0.1000	0.0100	0.0100
3	8	8	15	0.0070	0.1000	0.0100	0.0100
4	10	10	15	0.0070	0.1000	0.0100	0.0100
5	13	13	15	0.0070	0.1000	0.0100	0.0100
6	15	15	15	0.0070	0.1000	0.0100	0.0100
7	18	18	15	0.0070	0.1000	0.0100	0.0100
8	21	21	15	0.0070	0.1000	0.0100	0.0100
9	25	25	15	0.0070	0.1000	0.0100	0.0100
10	30	30	15	0.0070	0.1000	0.0100	0.0100
11	40	40	15	0.0070	0.1000	0.0100	0.0100
12	60	60	15	0.0070	0.1000	0.0100	0.0100
13	75	75	15	0.0070	0.1000	0.0100	0.0100
14	100	100	15	0.0070	0.1000	0.0100	0.0100

Figura 5.6: Parámetros con los que se probó el método DEGA.

se muestran en la tabla de la Fig. (5.6). Para cada asignación de parámetros y para cada conjunto de datos se realizaron 50 repeticiones. En la Fig. (5.7) se muestra en color verde los errores promediados que se obtuvieron con el método DEGA. También se corrió el método DEG sobre los 30 conjuntos de datos con los mejores valores de los parámetros

$$r = 15, \gamma = 0.0070, \lambda_1 = 0.1000.$$

En la Fig. (5.6) se muestra en color azul los errores obtenidos con el método DEG.

Se puede distinguir que para todas las asignaciones de los parámetros correspondientes al método DEGA, los errores arrojados por el método DEGA son menores a los arrojados por el método DEG.

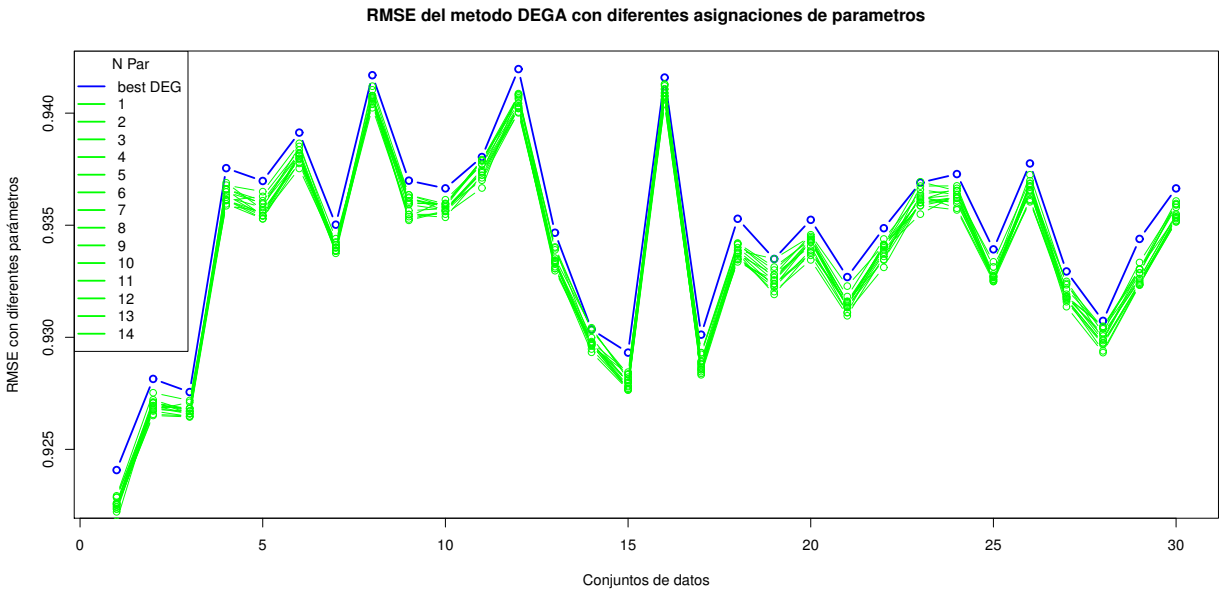


Figura 5.7: 50 repeticiones de los métodos DEG y DEGA sobre 30 conjuntos de datos para cada elección de los parámetros de prueba de la tabla de la Fig. (5.6).

Resultados de las pruebas.

La Fig. (5.8) muestra la matriz de calificaciones con los renglones y columnas reordenadas con base en las agrupaciones arrojadas por el método DEGA tanto para usuarios como para películas, se escogió la salida correspondiente a 10 grupos de usuarios y 10 grupos de películas.

Se incluye un ejemplo de salida del método DEGA con las recomendaciones y los valores de las calificaciones de las predicciones.

```
User: 930
{

PELICULAS RECOMENDADAS:

  4.26400 Angel Baby (1995)
  4.21500 Some Mother's Son (1996)
  4.11100 Faust (1994)
  4.10700 Saint of Fort Washington, The (1993)
  4.06700 Butcher Boy, The (1998)

PELICULAS MAS GUSTADAS POR EL USUARIO:
```

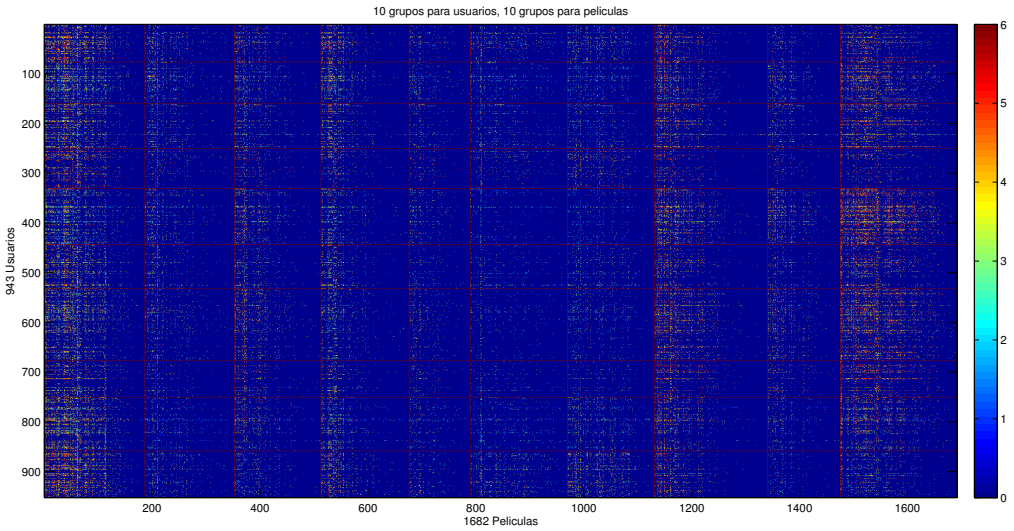


Figura 5.8: Se reordenaron los renglones y las columnas de la matriz de calificaciones con base en los grupos (clusters) arrojados por el método DEGA.

```
  5.00000 Jean de Florette (1986)
  5.00000 Cold Comfort Farm (1995)
  5.00000 Horseman on the Roof, The (Hussard sur le toit, Le) (1995)
  4.00000 Strictly Ballroom (1992)
  4.00000 Addicted to Love (1997)
}
```

5.4. Pruebas sobre los datos de la biblioteca del CIMAT.

Descripción de los datos.

En el Apéndice A se hace una descripción completa de los datos con que cuenta la biblioteca del CIMAT. Incluímos aquí una breve descripción. Un *libro* refiere a una publicación de la cual se pueden tener varios *ejemplares* o copias. Una *ficha* contiene la información total y única de cada libro. Los registros que se tienen sobre las *Fichas* y los *Ejemplares* comienzan a partir del año 2003, contando un total de **24035 ejemplares**, y un total de **18903 fichas** hasta el presente año 2011. Los registros que se tienen de las *Devoluciones* de libros comienzan en el año 1998, sin embargo, los registros sobre los *Préstamos* comienzan en el año 2005.

Solamente se consideran aquellas fichas que se sacaron a préstamo más de 5 veces entre todos los usuarios, durante el periodo 1998-2011. Sobre estas fichas se consideran a los usuarios que hayan realizado más de 20 devoluciones durante el periodo 2007-2010. Después de este

filtrado resultaron un total de **331 usuarios** y **3847 fichas**. Para cada usuario y para cada ficha, se contabiliza la frecuencia con que dicho usuario sacó a préstamo la ficha en cuestión y se asigna ésta como calificación. Así que las calificaciones están dadas en el formato:

id usuario | id ficha | frecuencia de préstamo

La frecuencia de préstamo varía en el intervalo $[0, 20]$. Con la finalidad de reducir el margen de diferencias entre las calificaciones se proyectan las frecuencias al intervalo $[0, 5]$, mediante la siguiente transformación:

$$f(x) = 5(1 - \exp(-x)).$$

Sobre estas calificaciones transformadas se corren los métodos DEG y DEGA.

Con el filtrado se contabilizan un total de 10924 calificaciones (frecuencia de préstamo > 0), sobre este total de datos se realiza un particionado en dos conjuntos; conjunto de entrenamiento (8739) y conjunto de prueba (2185), cuidando que en el conjunto de entrenamiento cada usuario tenga por lo menos 10 préstamos (aunque sean del mismo libro). Se muestra a continuación la densidad de préstamos con los usuarios y las fichas filtradas.

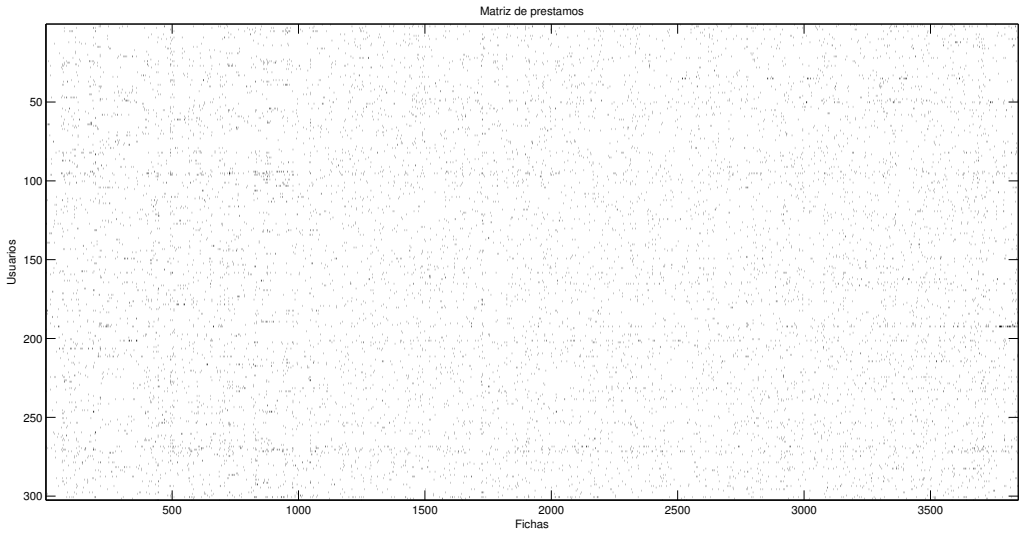


Figura 5.9: Matriz de préstamos (2007-2011). El fondo blanco corresponde a la ausencia de calificaciones.

Resultados de las pruebas.

Se realizó un total de 50 repeticiones sobre los datos de la biblioteca, para cada elección de los parámetros que se muestran en la tabla de la Fig. (5.10) y se ejecutó el método DEGA, presentamos los diagramas de caja de los errores arrojados para el conjunto de prueba.

N Par	k_U	k_V	r	γ	λ_1	λ_2	λ_3
1	10	10	10	0.0080	0.1000	0.5000	0.5000
2	10	15	10	0.0020	0.1000	1.0000	1.0000
3	10	15	15	0.0060	0.1000	0.1000	0.1000
4	15	10	5	0.0090	0.1000	0.0050	0.0050
5	20	20	10	0.0020	0.1000	0.5000	0.5000
6	30	30	15	0.0020	0.1000	0.5000	0.5000
7	60	60	20	0.0020	0.1000	0.5000	0.5000
8	100	100	20	0.0020	0.1000	0.0050	0.0050

Figura 5.10: Parámetros con los que se corrió el método DEGA sobre los datos de la biblioteca.

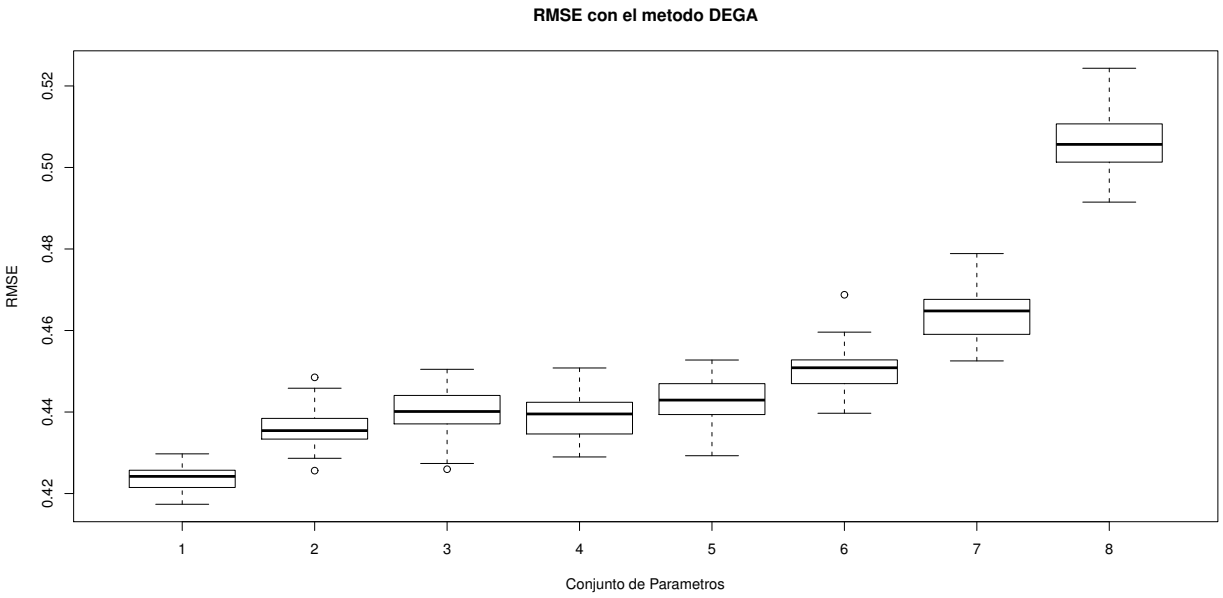


Figura 5.11: Los errores están por debajo de 0.50 para casi todas las asignaciones de parámetros.

Basándonos en los grupos arrojados por el método DEGA, re-agrupamos los renglones y columnas de la matriz de predicción para distinguir los grupos. Mostramos las gráficas donde los grupos de usuarios y fichas están limitados por las líneas rojas.

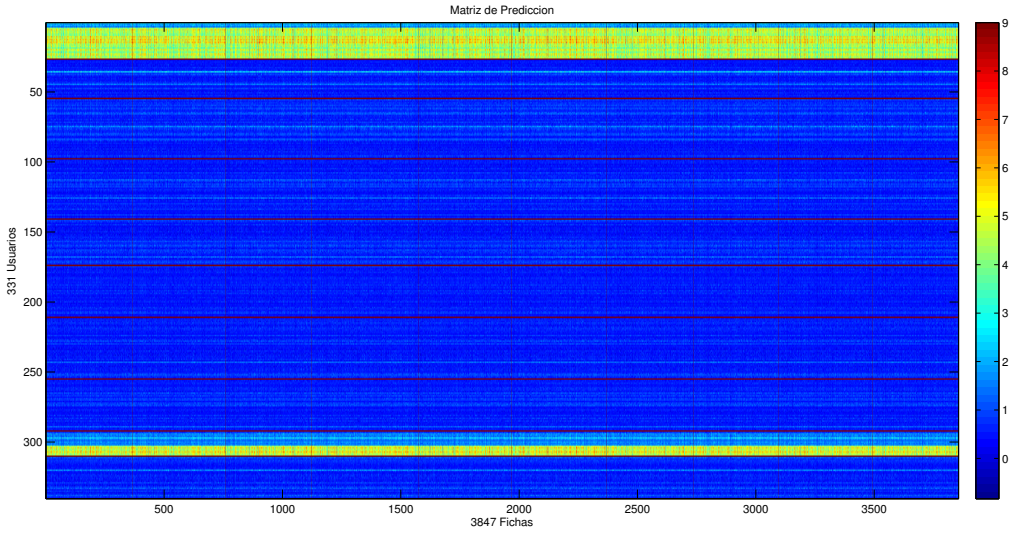


Figura 5.12: 10 grupos para usuarios, 10 grupos para fichas.

Presentamos un ejemplo de recomendación obtenida por el método DEGA:

```
User: 5 {
LIBROS MAS SACADOS:

LOGIC FOR MATHEMATICS AND COMPUTER SCIENCE         BURRIS, STANLEY N.
EXPERT SYSTEMS AND PROBABILISTIC NETWORK MODELS    CASTILLO, ENRIQUE
INTRODUCTION TO MATHEMATICAL STATISTICS            HOGG, ROBERT V.
ADVANCES IN PROBABILITY DISTRIBUTIONS WITH GIVEN M DALL'AGLIO, GIORGIO
GRAPHICAL MODELS IN APPLIED MULTIVARIATE STATISTIC WHITTAKER, JOE

LIBROS RECOMENDADOS:

CLASSIFICATION AND REGRESSION TREES                BREIMAN, LEO
THE ELEMENTS OF STATISTICAL LEARNING : DATA MINING HASTIE, TREVOR
DIRECT METHODS OF SOLVING MULTIDIMENSIONAL INVERSE KABANIKHIN, S. I.
GENERAL STATISTICS                                 CHASE, WARREN
APPLIED MULTIVARIATE DATA ANALYSIS                 JOBSON, J. D.
}
```

Se puede ver que la calidad de recomendación es coherente con los gustos del usuario.

Capítulo 6

Conclusiones y trabajos a futuro.

Los sistemas de recomendación surgen como una necesidad de ofrecer recomendaciones personalizadas de productos o servicios al usuario, principalmente por las tiendas electrónicas y los proveedores de contenido que manejan productos a gran escala. En este trabajo hemos mostrado un bosquejo de los *Sistemas de Recomendación*, de la amplia variedad de aplicaciones que ofrece, así como de los diferentes enfoques desde los cuales puede ser atacado.

Mostramos el modelo de factorización de matrices para sistemas de recomendación, el cual mapea usuarios y productos a un espacio común de factores latentes de dimensión r, donde cada producto j se asocia con un vector $V_j \in \mathbb{R}^r$, y cada usuario i se asocia con un vector $U_i \in \mathbb{R}^r$, tal que la interacción R_{ij} entre el usuario i y el producto j es modelado como un producto interno en dicho espacio: $R_{ij} = U_i^T V_j$.

Mostramos algunos de los enfoques principales del modelo de factorización de matrices, tales como: factorización no-negativa, factorización probabilística y factorización con margen maximal. En este trabajo se propuso una extensión del modelo básico de factorización de matrices, mediante la introducción de términos de agrupamiento sobre los usuarios y sobre los productos. Mostramos cómo con esta información adicional se puede mejorar la calidad de las recomendaciones, utilizando como método de aprendizaje el Descenso Estocástico de Gradiente. Realizamos pruebas sobre datos de películas proporcionados por MovieLens y sobre los datos de la biblioteca de un centro de investigación.
Entre los trabajos a futuro se tienen los siguientes:

i) Realizar las comparaciones de los resultados con los enfoques principales de factorización de matrices, mencionados en el párrafo anterior.

ii) Jugar con los términos de penalización y agrupamiento. Por ejemplo, se podría considerar únicamente los términos de agrupamiento y omitir los términos de penalización sobre las normas de las matrices factores U y V.

iii) Explorar otras técnicas de estimación de calificaciones para los sistemas en los que no se dan calificaciones explícitas, tales como bibliotecas, en los que generalmente se estiman las calificaciones con base en frecuencias.

Apéndice A

Análisis de los datos de la biblioteca del CIMAT.

Para realizar el análisis sobre los libros existentes en la biblioteca del CIMAT, se dispone de 4 archivos, a saber; **Fichas**, **Ejemplares**, **Devoluciones** y **Prestamos**. Sobre cada uno de estos archivos se hace un filtrado, ya que se tiene registro de material que no es parte del análisis del presente trabajo, como lo son: videos, tesis, ejemplares de otras unidades, etc.

A continuación mostramos las colecciones que no se consideraron:

Colección	Descripción
TE	Tesis
AGS	Unidad Aguascalientes
ZAC	Unidad Zacatecas
IN	Inegi
VFAMAT	Videos FAMAT
EM	Escuela de Matemáticas (FAMAT)
V	Video
C	Consulta

Figura A.1: Colecciones que no se toman en cuenta.

Antes de describir cada uno de los archivos cabe hacer una distinción entre libros y ejemplares. Un libro refiere a una publicación de la cual se pueden tener varios ejemplares o copias.

El archivo de **Fichas** contiene la información total y única de cada libro. En la ficha viene información importante como: *Número de ficha, Fecha de ingreso, Autor, Título y Clasificación*. El archivo de **Ejemplares** hace referencia a las adquisiciones que se hacen en la biblioteca. Contiene datos importantes tales como: *Número de ejemplar, Número de ficha, Número de adquisición y Fecha de ingreso.*

El archivo de **Devoluciones** contiene el historial de los libros devueltos (los que originalmente se pidieron prestados), así como el número de renovaciones que se hicieron durante

el préstamo. Los campos útiles en este archivo son: *Fecha de devolución, Número de renovaciones, Número de adquisición y Fecha de préstamo*. Cabe señalar que no todos los registros cuentan con este último dato, pues las fechas de préstamo de los que se tiene registro comienzan en el año 2005. El archivo de **Prestamos** no ofrece mucha información, puesto que no contiene los datos históricos de préstamos, sólo los vigentes, por tanto, no es de mucha utilidad y no se utiliza para el análisis actual.

Hay una sincronización entre el archivo de *Prestamos* y el archivo de *Devoluciones*, pues una vez que un ejemplar es devuelto, se borra del archivo de *Prestamos* y se agrega al de *Devoluciones*. Por otro lado, el archivo de ejemplares liga el *Número de adquisición* del archivo de *Devoluciones* con el *Número de ficha* del archivo de *Fichas*.

A.1. Generalidades.

A continuación, describimos un panorama general de los datos. Los registros que se tienen sobre las fichas y los ejemplares comienzan a partir del año 2003, sin embargo, esto no significa que no se tuvieran libros o registros de años anteriores, al parecer se debe a un cambio de software que se hizo en ese año, provocando que los registros con fechas anteriores a 2003 quedaran registrados con este año. Por este motivo se observa una diferencia considerable en la cantidad de datos en el año 2003 en comparación con años posteriores, como lo podemos apreciar en la Fig. (A.2). Por tanto, excluyendo el año 2003, en la Fig. (A.3) se puede apreciar mejor la comparación entre el archivo de *Fichas* y el de *Ejemplares* durante el periodo 2004-2010.

En la gráfica de la Fig. (A.2) se considera el campo *Fecha de Entrega* del archivo de *Devoluciones*, ya que como se mencionó al comienzo, el archivo de *Prestamos* no contiene el historial, sino sólo los préstamos vigentes.

En la Fig. (A.4) se puede apreciar la gráfica acumulativa de las fichas y ejemplares hasta el año 2010. En esta gráfica se aprecia también la evolución del archivo de *Devoluciones* (no acumulativa), para ver la relación entre la adquisición de ejemplares y los préstamos. Hasta el año 2010 se tienen registrados un total de **24035 ejemplares**, y un total de **18903 fichas**.

En el archivo de *Devoluciones* se tienen registros a partir del año 1998, esto para el campo correspondiente a la *Fecha de entrega*, pero para el campo de *Fecha de préstamo* se tienen registros a partir del año 2005.

A.2. Categorización sobre la LCC.

La biblioteca del CIMAT se rige por la clasificación de la biblioteca del congreso[1] (LCC, Library of Congress Classification). Con base en esta clasificación se hace una primera categorización tratando de incluír áreas comunes en una misma categoría con el fin de identificar las áreas que cuentan con más fichas. La categorización más general es la que abarca todas las áreas, como: Filosofía, Psicología, Historia, Geografía, Matemáticas, Ciencias Políticas,

[1] `http://www.loc.gov/catdir/cpso/lcco/`

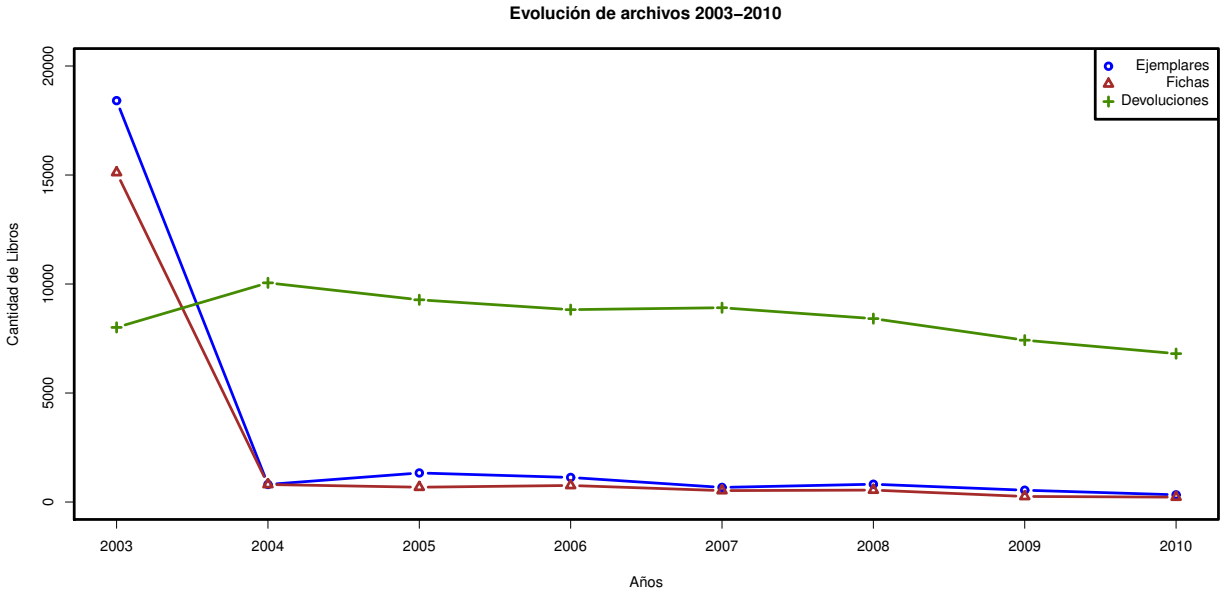


Figura A.2: El año 2003 contiene los registros de años anteriores de los archivos de Fichas y de Ejemplares.

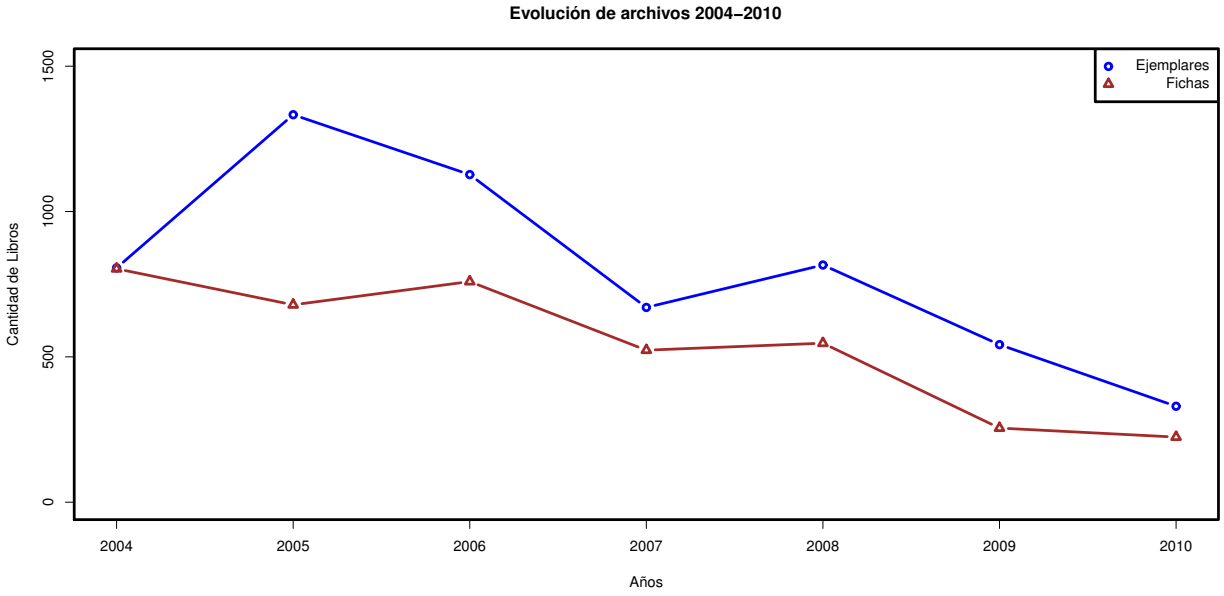


Figura A.3: Omitimos el año 2003 y el archivo de Devoluciones para una mejor comparación.

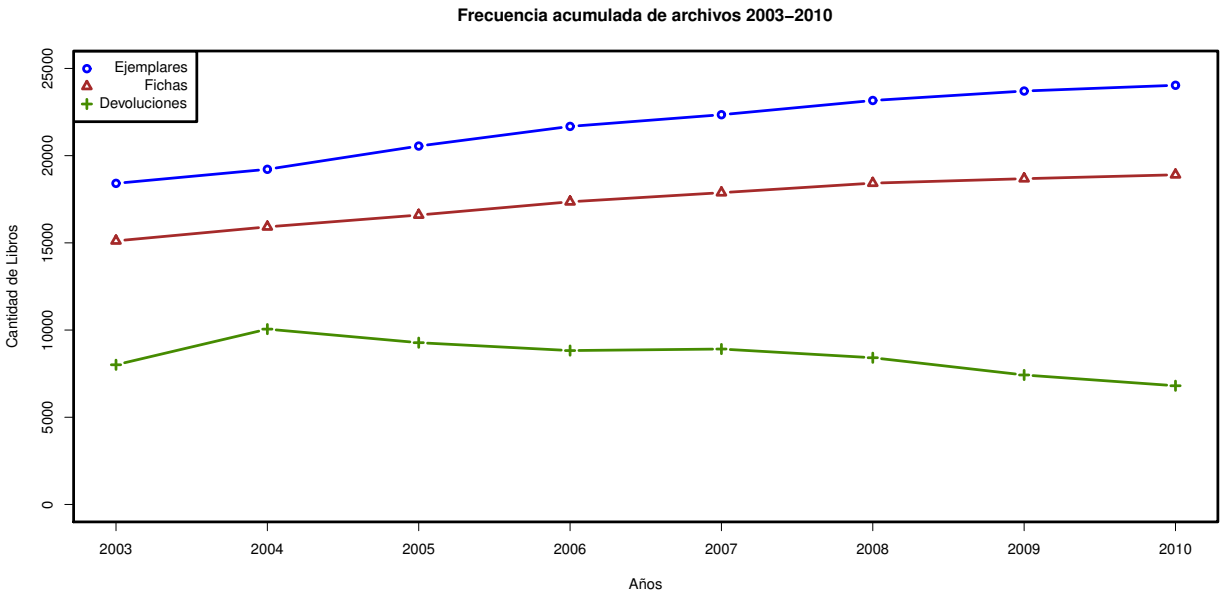


Figura A.4: Frecuencia acumulativa sobre las Fichas y los Ejemplares.

Leyes, Educación, Medicina, Tecnología, etc., las cuales están distribuidas a lo largo del abecedario.

En la Fig. (A.5) podemos observar el histograma correspondiente al número de fichas en la categorización más general. Se puede observar que la clasificación **Q** contiene la mayoría de las fichas, esto se debe a que la clasificación **Q** corresponde al área de Ciencias, que incluye particularmente las áreas de Matemáticas, Astronomía, Física, Química, entre otros. Las clasificaciones **T** y **H** corresponden a las áreas de Tecnología y Ciencias Sociales, respectivamente. En el área de Tecnología sobresalen las áreas de Ingeniería Civil, Ingeniería Eléctrica, Electrónica, Ingeniería Nuclear y Manufacturas. En el área de Ciencias Sociales sobresalen las áreas de Estadística, Teoría Económica, Industria, Comercio, Finanzas, entre otros.

La subclasificación **QA** tiene mayor presencia sobre las demás clasificaciones y subclasificaciones, puesto que corresponde al área de Matemáticas y Computación. Por tanto, la categorización final que se propone se hace considerando principalmente las fichas que pertenecen a dicha clasificación, y a ella se agregan el resto de las subclasificaciones, procurando afinidad. A continuación se presentan las categorías finales. En negrita las áreas representativas, así como las clasificaciones que pertenecen a cada categoría.

C1.- **Algebra. Geometry. ED. Analithyc Mechanics.** Combinatorics, Graph Theory, Linear Algegra, Approximation Theory, Topology.
"QA150-183","QA184-224","QA370-387","QA440-699"

C2.- **Applied Mathematics.** Theory of numbers, Set Theory, Mathematical Optimization,

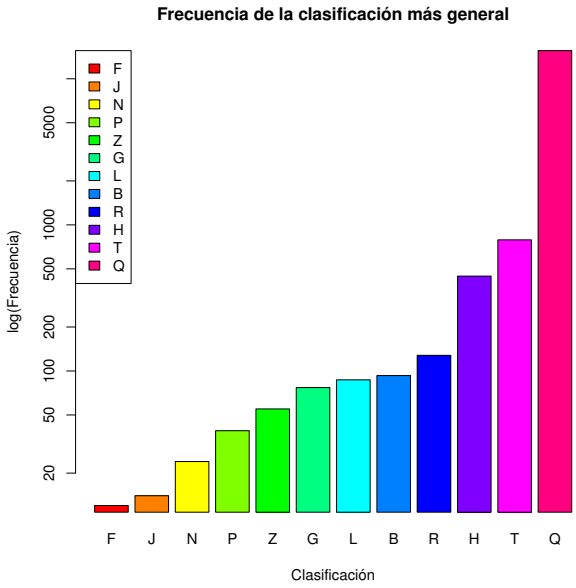


Figura A.5: Número de fichas (log-scale) por categoría.

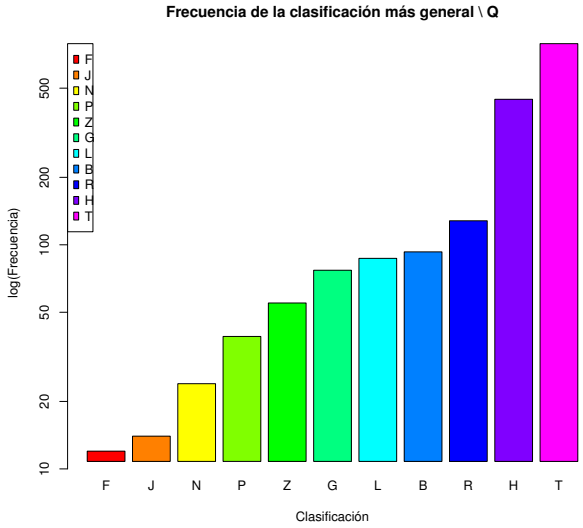


Figura A.6: Número de fichas (log-scale) por categoría excluyendo Q.

Fourier Analysis, Numerical Analysis.
"QA241-266.9","QA297-299.4","QA401-433","Others"

C3.- **Computer Science.** Machine theory, Coding theory, Programming languages, Computer Software, Technology, Computer Vision, Artificial Intelligence.
"QA71-90","QA267-272","TA-Z"

C4.- **Mathematical Analysis.** Theory of functions, Functions of real and complex variables, Functional Analysis, Operator theory, Topological Algebras.
"QA299.6-302","QA303-316","QA319-330","QA331-360"

C5.- **Mathematics (General issues). Study and Teaching.** Analityc Mechanics, Statics, Kinematics, Dynamics, General Works.
"QA1-7.9","QA8-149","QA800-939"

C6.- **Other Sciences. Astronomy. Chemistry. Economy. Physics.** Social Sciences.
"HA-Z","QA-Z","Q1-390","QC1-999"

C7.- **Probability and Statistics.**
"QA272.5-275","QA276-295"

Presentamos un barplot con la cantidad de fichas por año divididos en las categorías finales.

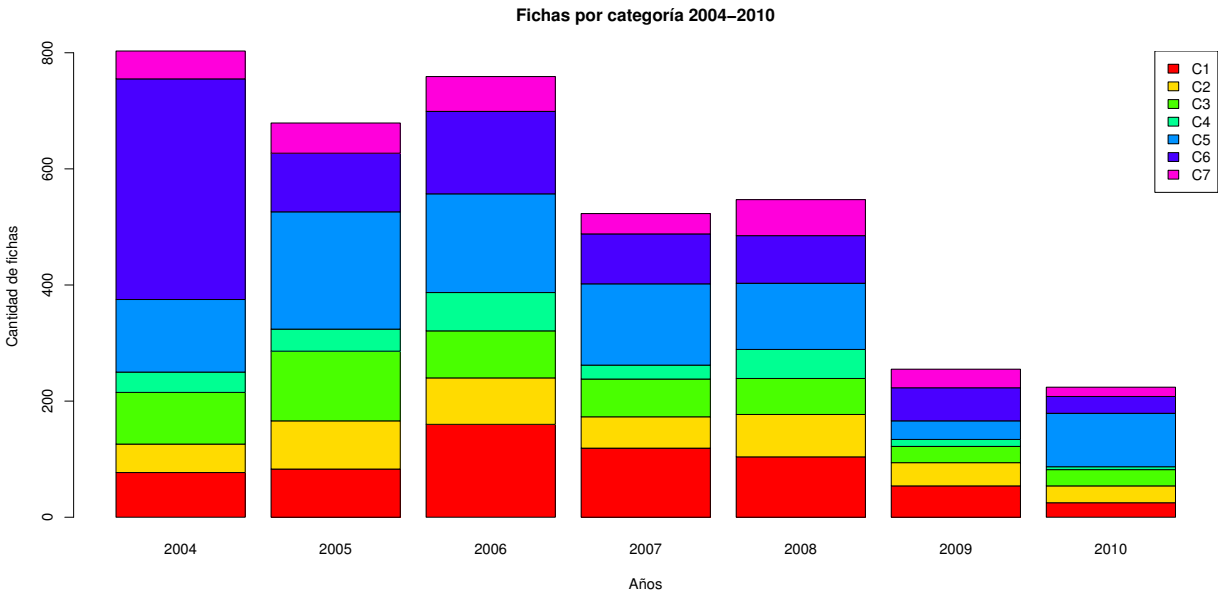


Figura A.7: Fichas por año con la categoría final.

A.3. Análisis de archivos con base en las categorías propuestas.

A continuación se muestra la evolución de las categorías en cada uno de los archivos a lo largo de sus respectivos periodos.

A.3.1. Fichas

En la gráfica que corresponde a la Fig. (A.7) se observa que en el año 2004 hay mayor cantidad de fichas en la categoría **C6** correspondiente a ciencias como: Física, Química, Astronomía, Economía, etc. Sin embargo, en los años restantes, la categoría **C5**, que corresponde a los tópicos generales de Matemáticas, predomina y se mantiene constante. También la categoría **C1** tiene un tamaño considerable a lo largo del periodo en cuestión, esta categoría incluye tópicos de Álgebra, Geometría, Ecuaciones Diferenciales, entre otras. Presentamos también la gráfica correspondiente a la frecuencia acumulada a través de los años, para comparar la adquisición de fichas entre las distintas categorías.

A.3.2. Ejemplares

Como su nombre lo indica, en el archivo de *Ejemplares* se tiene información del número de libros que se adquieren por año, por tanto, es posible que se adquieran varios ejemplares

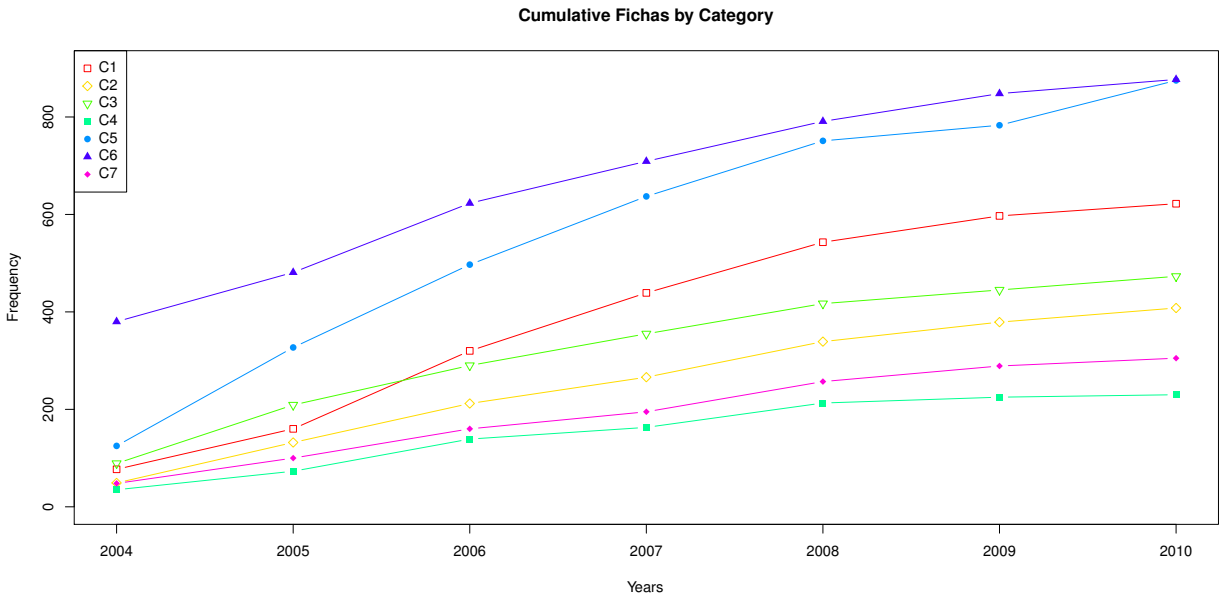


Figura A.8: Frecuencia acumulada de Fichas por categoría (2004-2010).

de una misma ficha, cada uno se diferencia del otro en el *Número de adquisición*, el cual se asigna consecutivamente conforme al orden de adquisición de los ejemplares.

En la gráfica se observa que al igual que para las fichas, las categorías que predominan en los ejemplares corresponden a: **C6, C5** y **C1**, lo cual no es difícil de inferir, puesto que al menos se debe contar con un ejemplar por cada ficha adquirida en cada año. De la misma forma que para las fichas, las categorías con menos ejemplares por año corresponden a: **C4** y **C5**. Estas categorías incluyen las áreas de Análisis Matemático, así como Probabilidad y Estadística.

Incluímos también la gráfica correspondiente a la frecuencia acumulada por categoría del archivo de *Ejemplares* en la Fig. (A.9).

A.3.3. Devoluciones

Dado que en el archivo de *Devoluciones* varios datos carecen de la *Fecha de préstamo*, tenemos como opción considerar la *Fecha de entrega* para estimar la frecuencia de uso de los libros, de manera particular, para ver la frecuencia de consulta de las categorías propuestas en la sección anterior.

Contrario a las fichas y ejemplares, se advierte aquí un mayor uso sobre los libros de las categorías **C4** y **C5**, que como se mencionó anteriormente corresponden a las áreas de Análisis Matemático y Probabilidad y Estadística. Aunque los libros de la categoría **C1: Algebra. Geometry. ED. Analithyc Mechanics.** se utilizan mayormente. Observar también que la categoría **C6** se consulta con escaza frecuencia en comparación con las demás categorías, a

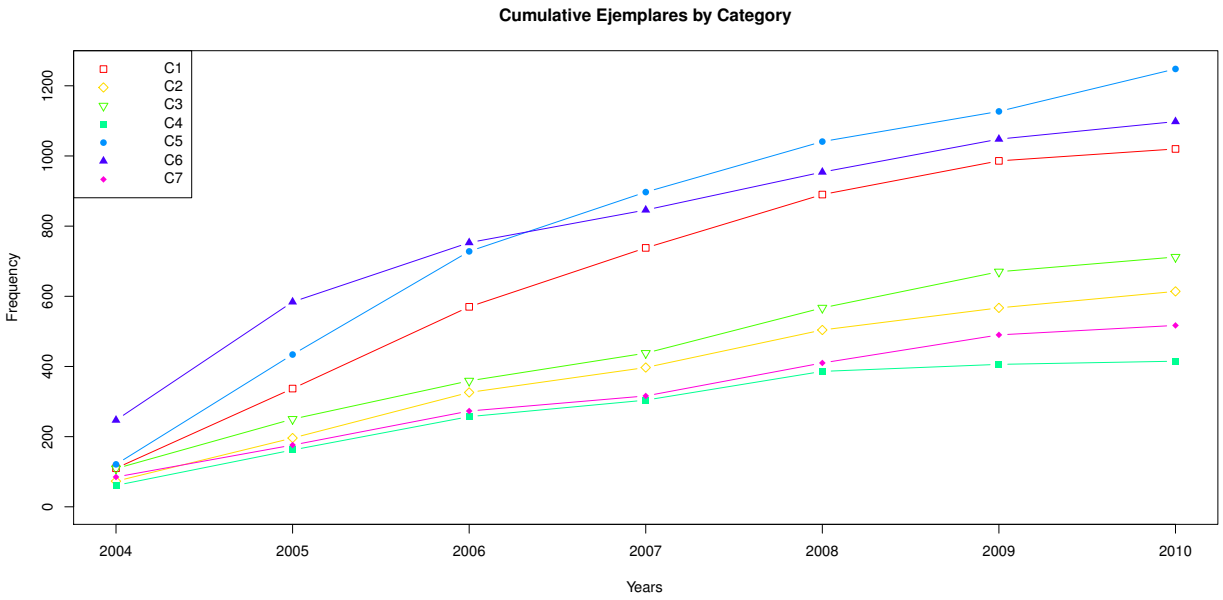


Figura A.9: Frecuencia acumulada de Ejemplares por categoría (2004-2010).

pesar de que tanto en las fichas como en los ejemplares tiene una presencia considerable.

Hasta el momento no se ha considerado el número de renovaciones en el análisis de datos, principalmente en cuanto a préstamos se refiere. Una opción podría ser contabilizar cada renovación como un nuevo préstamo y ver cómo repercute esto en las estadísticas, sin embargo, por el momento contabilizamos una vez cada préstamo, sin considerar el número de renovaciones.

A.4. Animaciones

A continuación se incluyen algunas ligas para ver la evolución de las categorías en los diferentes archivos mediante una animación de Google (Motion Chart). En la página se muestran al menos dos pestañas, la primera corresponde a la tabla de datos sobre la que se elabora la animación, la segunda pestaña corresponde a la animación.

Devoluciones:
https://spreadsheets.google.com/ccc?key=0AqctnY7OpxU3dGFCQjQ0eWgtYkNMWTZTc1otYngteHc&hl=en&authkey=CMefk-YJ#gid=1

Fichas:
https://spreadsheets.google.com/ccc?key=0AqctnY7OpxU3dElJelNmQ20wYzRVUnJuOTJieWhRekE&hl=en&authkey=COm38-cB#gid=7

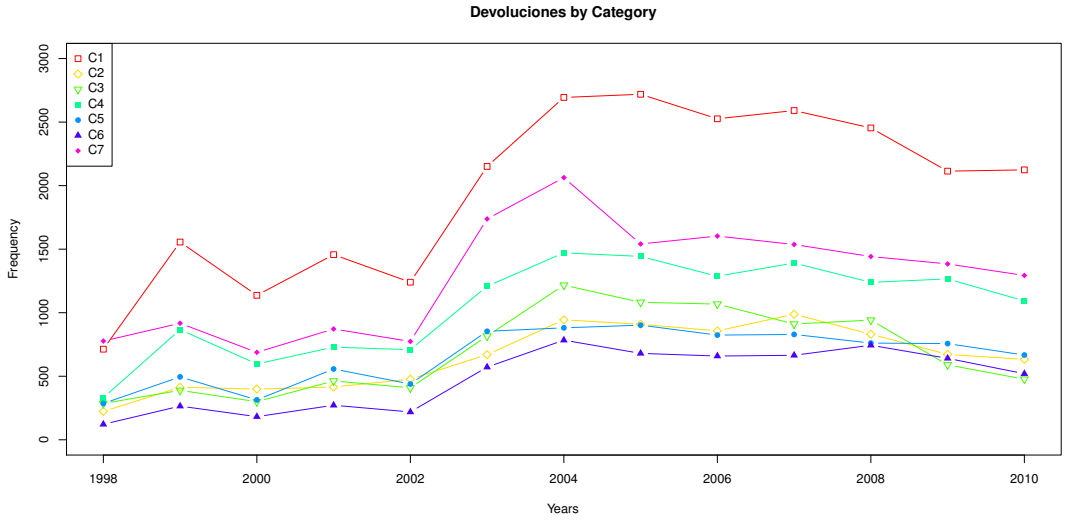


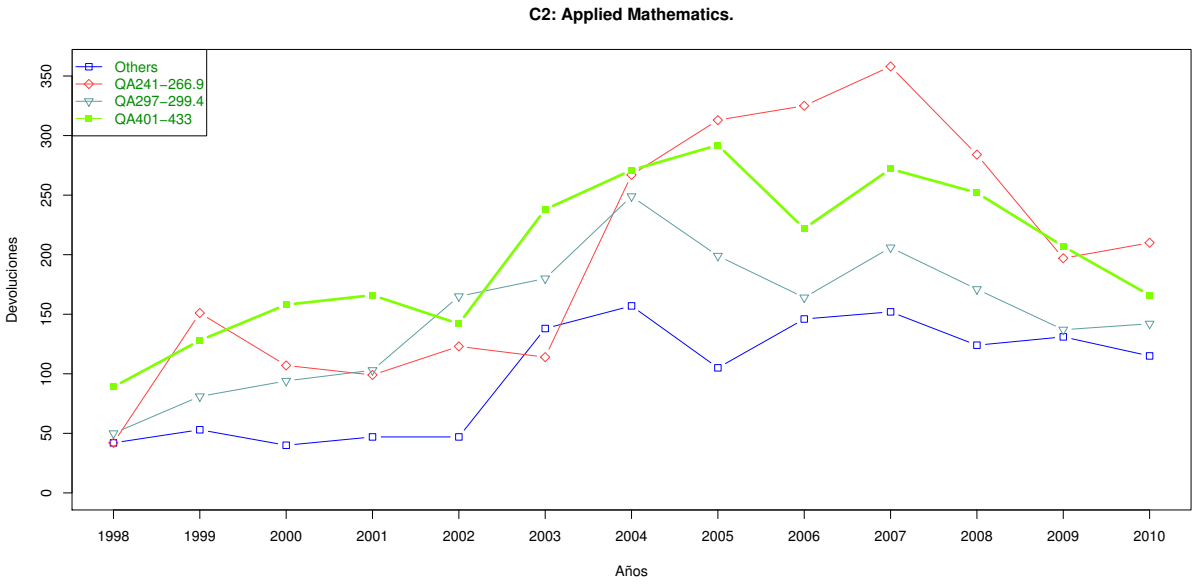
Figura A.10: Archivo de Devoluciones, periodo 1998-2010.

Ejemplares:
https://spreadsheets.google.com/ccc?key=0AqctnY7OpxU3dC0yS21qMXJIZW1oY3pHam5XT282dUE&hl=en&authkey=CImozO8K#gid=1

A.5. Evolución de las categorías en el archivo de Devoluciones.

A continuación mostramos la evolución de cada una de las categorías en el archivo de *Devoluciones*, divididas en clasificaciones, con el objeto de identificar las áreas que tienen mayor demanda en cuanto a préstamos se refiere. En cada categoría es visible la clasificación que tiene mayor demanda, excepto en la correspondiente a **Applied Mathematics** donde se van alternando diferentes clasificaciones, principalmente las áreas que corresponden a *Control Theory* y *Set Theory*.

En las siguientes gráficas se resalta la clasificación que ha obtenido más préstamos a lo largo de los años sobre cada una de las categorías y abajo se mencionan las áreas sobresalientes en dicha clasificación.


C2: Applied Mathematics.
Others
QA241−266.9
QA297−299.4
QA401−433
Devoluciones
0
50
100
150
200
250
300
350
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
Años

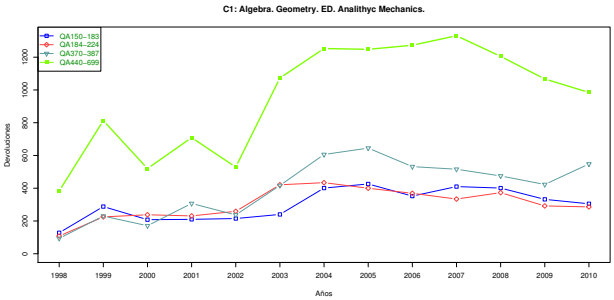


Figura A.11: Geometry. Trigonometry. Topology.

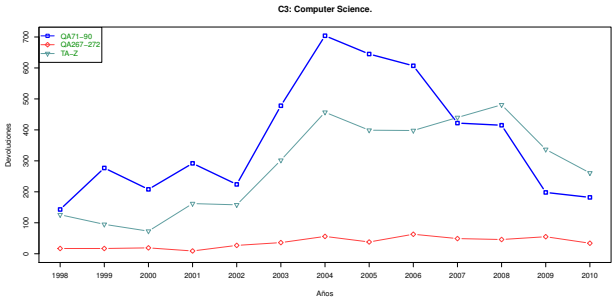


Figura A.12: Calculating Machines. Programming languages.

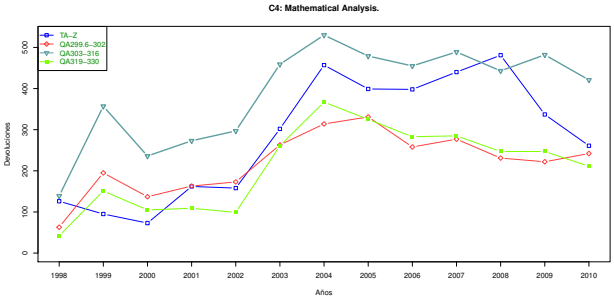


Figura A.13: Calculus of variations. Theory of measure and integration.

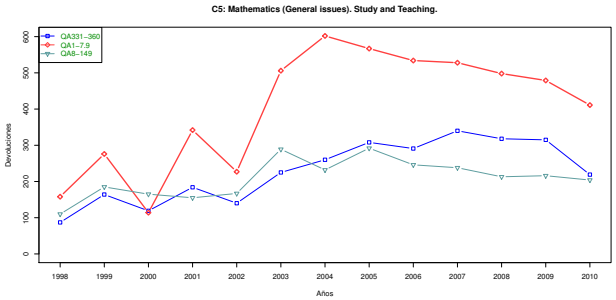


Figura A.14: Periodicals. Societies. Congresses. Serial collections.

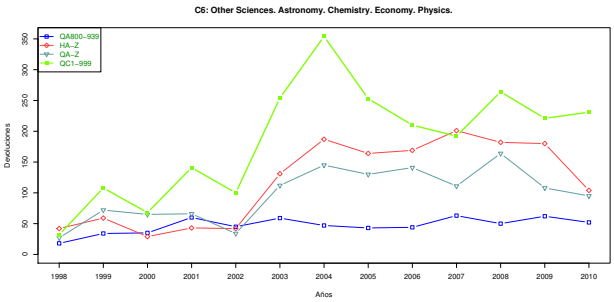


Figura A.15: Physics.

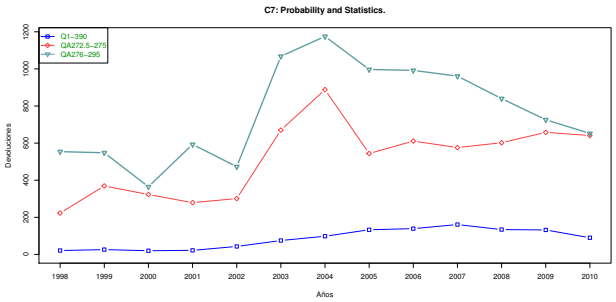


Figura A.16: Mathematical Statistics.

Apéndice B

Métodos de Proyección de gradiente.

Consideraremos la siguiente forma estándar del problema de optimización con restricciones de frontera:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{sujeto a} \quad & l_i \leq x_i \leq u_i, \quad i = 1, \ldots, n, \end{aligned} \tag{B.0.1}$$

donde $f(x) : \mathbb{R}^n \to \mathbb{R}$ es una función continuamente diferenciable, y l y u son cotas inferiores y superiores, respectivamente. Asumiendo k como el índice de las iteraciones. Los métodos de proyección de gradiente actualizan la solución actual x^k a x^{k+1} mediante la siguiente regla:

$$x^{k+1} = P[x^k - \alpha_k \nabla f(x^k)],$$

donde

$$P[x_i] = \begin{cases} x_i & \text{si } l_i < x_i < u_i, \\ u_i & \text{si } x_i \geq u_i, \\ l_i & \text{si } x_i \leq l_i, \end{cases}$$

mapea un punto de regreso a la región factible acotada. Las variantes de los métodos de proyección de gradiente difieren en la forma de seleccionar el tamaño de paso α^k. Consideraremos una forma simple y efectiva llamada "Regla de Armijo a lo largo del arco de proyección". El procedimiento es ilustrado en el Algoritmo (7).

La condición (B.0.2), usado en la mayoría de las pruebas de los métodos de proyección de gradiente, aseguran suficiente descenso sobre el valor de la función por cada iteración. Probando los tamaños de paso $1, \beta, \beta^2, \ldots,$ se ha probado que existe $\alpha_k > 0$ que satisface (B.0.2) y que cada punto límite de $\{x^k\}_{k=1}^{\infty}$ es un punto estacionario de (B.0.1). Una elección común de σ es 0.01, y para este trabajo consideraremos $\beta = \frac{1}{10}$.

La búsqueda de α_k es la operación que más tiempo consume en el Algoritmo (7), por tanto, se deberían checar pocos tamaños de paso como sea posible. Dado que α_{k-1} y α_k pueden ser similares, una sugerencia es usar α_{k-1} como un valor inicial y luego ir aumentando o disminuyendo dicho valor con el fin de encontrar β^{t_k} que satisfaga (B.0.2). Mas aún, si t_k es no-negativo, el Algoritmo (8) puede ser muy conservativo al restringir $\alpha_k \leq 1$. A veces, un

Algorithm 7 Un método de proyección de gradiente para optimización con restricciones de frontera.

Require: Matriz de adyacencia W.
Ensure: Matriz de costo mÃnimo $D^{(n)}$.

1: Dado $0 < \beta < 1, 0 < \sigma < 1$. Inicializar cualquier punto factible x^1.
2: **for** $k = 1, 2, \ldots$ **do**
3:

$$x^{k+1} = P[x^k - \alpha_k \nabla f(x^k)],$$

donde $x_k = \beta^{t_k}$, y t_k es el primer entero no-negativo t para el cual

$$f(x^{k+1}) - f(x^k) \leq \sigma \nabla f(x^k)^T (x^{k+1} - x^k). \tag{B.0.2}$$

4: **end for**
5: **return** $D^{(n)}$

tamaño de paso más grande proyecta las variables más efectivamente en cada iteración. El Algoritmo (8) mejora la selección del valor inicial α en cada iteración y permite que α sea mayor que uno.

A continuación se presenta la aplicación del método de proyección de gradiente al problema de NMF.

B.1. Alternando Mínimos Cuadrados No-Negativos usando Métodos de Proyección de Gradiente.

La sección 3.1.2 indica que el Algoritmo (2) depende de la eficiencia de resolver los subproblemas (3.1.6) y (3.1.7), donde cada uno es un problema de restricciones de frontera. El subproblema (3.1.7) consiste de m problemas independientes de mínimos cuadrados no-negativos de la forma (3.1.8).

Para una descripción simple del método, nos centraremos en (3.1.7) y lo reescribiremos como:

$$\begin{aligned} \min_{H} \quad & \bar{f}(H) \equiv \frac{1}{2} \| V - WH \|_F^2 \\ \text{sujeto a} \quad & H_{bj} \geq 0, \ \ \forall b, j. \end{aligned} \tag{B.1.1}$$

Tanto V como W son matrices constantes en (B.1.1). Si concatenamos las columnas de H

Algorithm 8 Un método de proyección de gradiente para optimización con restricciones de frontera.

Require: Matriz de adyacencia W.
Ensure: Matriz de costo mÃnimo $D^{(n)}$.

1: Dado $0 < \beta < 1, 0 < \sigma < 1$. Inicializar cualquier punto factible x^1. Fijar $\alpha_0 = 1$.
2: **for** $k = 1, 2, \ldots$ **do**
3: (a) Hacer $\alpha_k \leftarrow \alpha_{k-1}$

(b) Si α_k satisface (B.0.2), incrementar repetidamente mediante

$$\alpha_k \leftarrow \alpha_k/\beta$$

hasta que α_k viole (B.0.2) o $x(\alpha_k/\beta) = x(\alpha_k)$.
De lo contrario, decrementar repetidamente α_k mediante

$$\alpha_k \leftarrow \alpha_k \cdot \beta$$

hasta que α_k satisfaga (B.0.2).

(c) Fijar $x^{k+1} = P[x^k - \alpha_k \nabla f(x^k)]$.
4: **end for**
5: **return** $D^{(n)}$

en un vector vec(H), entonces

$$\begin{aligned} \bar{f}(H) &= \frac{1}{2}\| V - WH \|_F^2 \\ &= \frac{1}{2}\mathrm{vec}(H)^T \begin{bmatrix} W^TW & & \\ & \ddots & \\ & & W^TW \end{bmatrix} \mathrm{vec}(H)^T \\ &\quad + \text{términos lineales en } H. \end{aligned}$$

La matriz Hessiana (es decir, la segunda derivada) de $\bar{f}(H)$ es un bloque diagonal, y cada bloque W^TW es una matriz semidefinida positiva de tamaño $r \times r$. Como $W \in R^{n\times r}$ y $r \ll n, W^TW$ y toda la matriz Hessiana tienden a ser bien-condicionadas, una propiedad deseable para algoritmos de optimización. Por tanto, los métodos basados en gradientes pueden converger suficientemente rápido.

El alto costo computacional de resolver los subproblemas (3.1.6) y (3.1.7) en cada iteración es una preocupación. Por tanto es esencial analizar la complejidad del tiempo y encontrar implementaciones eficientes. Cada subproblema requiere un procedimiento de iteración, cuyas iteraciones son referidas como subiteraciones. Cuando usamos el Algoritmo (8) para resolver (B.1.1), hay que mantener el gradiente $\nabla\bar{f}(H) = W^T(WH - V)$ en cada subiteración. Continuando con la discusión acerca de la Ec. (**??**), ésta se debería calcular mediante $(W^TW)H - W^TV$. Las matrices constantes W^TW y W^TV pueden ser calculadas

respectivamente en $O(nr^2)$ y $O(nmr)$ operaciones antes de correr las subiteraciones.

La tarea principal por subiteración es encontrar el tamaño de paso α tal que la condición de suficiente descenso (B.0.2) sea satisfecha. Asumiendo $\bar{H}$ como solución actual, para checar si $\tilde{H} \equiv P[\bar{H} - \alpha\nabla\bar{f}(\bar{H})]$, satisface (B.0.2), calcular $\bar{f}(\tilde{H})$ toma $O(nmr)$ operaciones. Si hacen t pruebas de $\tilde{H}$'s, el costo computacional $O(tnmr)$ es elevado. Proponemos la siguiente estrategia para reducir el costo. Para una función cuadrática $f(x)$ y cualquier vector d,

$$f(x+d) = f(x) + \nabla f(x)^T d + \frac{1}{2} d^T \nabla^2 f(x) d. \tag{B.1.2}$$

Así, para dos iteraciones consecutivas $\bar{x}$ y $\tilde{x}$, (B.0.2) puede ser escrito como:

$$(1-\sigma)\nabla f(\bar{x})^T(\tilde{x}-\bar{x}) + \frac{1}{2}(\tilde{x}-\bar{x})^T\nabla^2 f(\bar{x})(\tilde{x}-\bar{x}) \leq 0.$$

Luego $\bar{f}(H)$ definida en (B.1.1) es cuadrática, por tanto (B.0.2) se convierte en

$$(1-\sigma)\langle\nabla f(\bar{H}), \tilde{H}-\bar{H}\rangle + \frac{1}{2}\langle\tilde{H}-\bar{H}, (W^TW)(\tilde{H}-\bar{H})\rangle \leq 0, \tag{B.1.3}$$

donde $\langle\cdot,\cdot\rangle$ es la suma del producto componente a componente de las dos matrices. La operación más grande en (B.1.3) es el producto de las matrices $(W^TW)\cdot(\tilde{H}-\bar{H})$, el cual toma $O(mr^2)$. Por tanto, el costo $O(tnmr)$ de checar (B.0.2) se reduce significativamente a $O(tmr^2)$. Con el costo $O(nmr)$ para calcular W^TV al inicio, la complejidad de usar el Algoritmo (8) para resolver el subproblema (B.1.1) es $O(nmr) = \#\text{subiteraciones } \times O(tmr^2)$, donde t es el promedio de checar (B.0.2) en cada subiteración. Se puede usar el mismo procedimiento para obtener W^{k+1} reescribiendo (3.1.7) como (B.1.1):

$$\bar{f}(W) \equiv \frac{1}{2}\| V^T - H^TW^T \|_F^2,$$

donde V^T y H^T son matrices constantes. El costo total para resolver (3.1.1) es

$$\#\text{iteraciones} \times (O(nmr) + \#\text{subiteraciones} \times O(tmr^2 + tnr^2)) \tag{B.1.4}$$

En cada iteración, se hacen dos $O(nmr)$ operaciones: $V(H^k)^T$ y $(W^{k+1})^TV$, , los mismos que se hacen en el método multiplicativo. Si tanto t como #iteraciones son pequeños, este método es eficiente.

B.2. Aplicación directa del método de proyección de gradiente a la NMF.

Podemos aplicar directamente el Algoritmo (8) para minimizar (3.1.1). Similar a resolver los problemas de mínimos cuadrados no-negativos en la Sección B.1, la operación más cara es

checar la condición de suficiente descenso (B.0.2). De la solución actual $(\tilde{W}, \tilde{H})$, simultáneamente se actualizan ambas matrices $(\tilde{W}, \tilde{H})$:

$$(\tilde{W}, \tilde{H}) \equiv P[(\bar{W}, \bar{H}) - \alpha(\nabla_W f(\bar{W}, \bar{H}), \nabla_H f(\bar{W}, \bar{H}))].$$

Como $f(W, H)$ no es una función cuadrática, (B.1.2) no se cumple. Así que la sugerencia (B.1.3) no puede ser aplicada para ahorrar el costo computacional. Luego, el cálculo $f(\tilde{W}, \tilde{H}) = \frac{1}{2}\| V - \tilde{W}\tilde{H} \|_F^2$ toma $O(nmr)$ operaciones. El costo computacional total es

$$\#\text{iteraciones} \times O(tnmr),$$

donde t es el número promedio de veces que se checa la condición (B.0.2) por iteración. Dado un punto inicial (W^1, H^1), si $\| V - W^1 H^1 \|_F^2 > \| V \|_F^2$, generalmente pasa que después de esta iteración $(W^1, H^1) = (0, 0)$ causa que el algoritmo se detenga. La solución $(0, 0)$ es inútilmente un punto estacionario de (3.1.1). Un remedio simple es encontrar un punto inicial nuevo $(W^1, \bar{H}^1)$ tal que $f(W^1, \bar{H}^1) < f(0, 0)$. Resolviendo $\bar{H}^1 = \arg\min_{H \geq 0} f(W^1, H)$ mediante el procedimiento descrito en la Sección B.1, se tiene que

$$\| V - W^1 \bar{H}^1 \|_F^2 \leq \| V - W^1 \cdot 0 \|_F^2 = \| V \|_F^2$$

Generalmente la desigualdad estricta se cumple, por tanto $f(W^1, \bar{H}^1) < f(0, 0)$.

Apéndice C

Inferencia en el modelo BPMF.

Uno de los algoritmos más simples de MCMC es el algoritmo de muestreo de Gibbs, el cual itera a través de las variables latentes, muestreando cada uno de sus distribuciones condicionales sobre los valores actuales de las variables restantes. El muestreo de Gibbs es típicamente empleado cuando estas distribuciones condicionales pueden ser muestreados fácilmente.

La distribución condicional sobre el vector de características del usuario U_i, condicionada sobre las características del producto, la matriz de calificaciones observadas R, y los valores de los hiperparámetros, es Gaussiano:

$$p(U_i|R, V, \Theta_U, \alpha) = \mathcal{N}(U_i|\mu_i^*, [\Lambda_i^*]^{-1}) \tag{C.0.1}$$

$$\sim \prod_{j=1}^{M} \left[\mathcal{N}(R_{ij}|U_i^T V_j, \alpha^{-1})\right]^{I_{ij}} p(U_i|\alpha_U, \Lambda_U),$$

donde

$$\Lambda_i^* = \Lambda_U + \alpha \sum_{j=1}^{M} [V_j V_j^T]^{I_{ij}} \tag{C.0.2}$$

$$\mu_i^* = [\Lambda_i^*]^{-1} \left(\alpha \sum_{j=1}^{M} [V_j R_{ij}]^{I_{ij}} + \Lambda_U \alpha_U \right). \tag{C.0.3}$$

Notar que la distribución condicional sobre la matriz de características latentes del usuario U está factorizada en productos de distribuciones condicionales sobre los vectores de características de cada usuario.

$$p(U|R, V, \Theta_U) = \prod_{i=1}^{N} p(U_i|R, V, \Theta_U).$$

La distribución condicional sobre los hiperparámetros del usuario condicionado sobre la matriz de características de los usuarios U está dada por la distribución Gaussian-Wishart:

$$p(\alpha_U, \Lambda_U|U, \Theta_0) = \mathcal{N}(\alpha_U|\alpha_0^*, (\beta_0^* \Lambda_U)^{-1}) \mathcal{W}(\Lambda_U|W_0^*, \nu_0^*), \tag{C.0.4}$$

donde

$$\alpha_0^* = \frac{\beta_0\alpha_0 + N\bar{U}}{\beta_0 + N}, \quad \beta_0^* = \beta_0 + m, \quad \nu_0^* = \nu_0 + m,$$

$$[W_0^*]^{-1} = W_0^{-1} + m\bar{S} + \frac{\beta_0 m}{\beta_0 + m}(\alpha_0 - \bar{U})(\alpha_0 - \bar{U})^T$$

$$\bar{U} = \frac{1}{N}\sum_{i=1}^{N} U_i, \qquad \bar{S} = \frac{1}{N}\sum_{i=1}^{N}(U_i - \bar{U})(U_i - \bar{U})^T.$$

Las distribuciones condicionales sobre los vectores de características de los productos y de los hiperparámetros de los productos tiene exactamente la misma forma.

Bibliografía

[1] Daniel Lee, Sebastian Seung. Algorithms for non-negative matrix factorization. *Advances in Neural Information Processing Systems* 13: MIT Press., 556-562, 2001.

[2] Chih-Jen Lin. Projected Gradient Methods for Non-negative Matrix Factorization. *Departmen of Computer Science*, National Taiwan University, Taipei 106, Taiwan, 1990.

[3] Ruslan Salakhutdinov, Andriy Mnih. Bayesian Probabilistic Matrix Factorization using Markov Chain Monte Carlo. In ICML '08: *Proceedings of the 25th International Conference on Machine Learning*, Finland, 2008.

[4] Natan Srebro, Jason Rennie and Tommi Jaakkola. Maximum Margin Matrix Factorization. *Advances in Neural Information Processing Systems* (NIPS) 17, 2005.

[5] Yehuda Koren, Robert Bell and Chris Volinsky. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer*, Volume 42, Issue 8, p.30-37, 2009.

[6] Gediminas Adomavicius, Alexander Tuzhilin. Toward the next generations of recommender systems: A survey of the state-of-art and possible extensions. *IEEE transactions on knowledge and data engineering*, Volume 17, Issue 6, 2005.