Escuela Superior
de Ingeniería y Tecnología
Universidad de La Laguna

Desarrollo filtrado colaborativo

Gestión del conocimiento para las organizaciones

Alejando Martín Linares
alu0101476476@ull.edu.es
Guillermo Emmanuel González Méndez
alu0101466941@ull.edu.es
Joel Saavedra Páez
alu0101437415@ull.edu.es

ÍNDICE

Implementación y desarrollo

Hemos desarrollado la aplicación utilizando el lenguaje C++ debido a que los integrantes del grupo tenemos un conocimiento amplio en dicho lenguaje, además de aprovechar su rápida ejecución.

El programa sigue un estilo POSIX para las entradas y proporciona la salida de los datos mediante consola. **Los parámetros de entrada se especifican en el README del repositorio**.

Ficheros y contenidos

El programa cuenta con 3 archivos principales de código (*main.cc, recommender-system.cc y tools.cc)* y 2 archivos headers (*recommender-system.h y tools.h)*.

main.cc

Contiene la función principal. Toma los parámetros por línea de comando, llama al constructor de la clase que representará el sistema de recomendación y al método para predecir las valoraciones.

recommender-system.cc

Implementación de los métodos de la clase ***RecommenderSystem***, un sistema de recomendación basado en usuarios. Su función es leer una matriz de valoraciones entre usuarios e ítems desde un archivo, calcular las similitudes entre usuarios utilizando métricas como correlación de Pearson, coseno o distancia euclídea, seleccionar los vecinos más parecidos para cada usuario y predecir las valoraciones desconocidas en la matriz mediante predicción simple o distancia con respecto a la media.

El sistema muestra por pantalla todas las predicciones realizadas, la matriz original con las incógnitas, las similitudes de cada usuario con el resto, los vecinos seleccionados y la matriz completa con las predicciones generadas.

Optamos por calcular la media de cada usuario, las similitudes y la elección de vecinos en el constructor de la clase, es decir, antes de realizar las predicciones, de esta manera obtenemos un mejor rendimiento que si se re calculase después de cada predicción. Además, se está calculando con datos reales, y no basándose en predicciones que pueden ser erróneas, ocasionando una propagación de errores.

En la elección de vecinos evitamos escoger vecinos con similitud negativa, específicamente para cuando esta se calcula con el Coeficiente de Pearson. Por lo que se puede dar el caso en el que un usuario tenga menos vecinos de los que se introdujeron por línea de comandos debido a esta restricción.

Se optó por resolver las predicciones empezando por los usuarios que tengan menos incógnitas, y para cada uno de estos se empieza por los ítems que tengan también menos incógnitas. Esta información se precalcula y ordena en el constructor de la clase. El motivo de esta decisión es que las predicciones sobre los usuarios e ítems que menos incógnitas tienen, van a ser, por lo general, más precisas. De esta forma contaremos con más información para los usuarios que más incógnitas tienen.

tools.cc

Se implementan funciones de comprobación y uso, además de una función para mostrar el funcionamiento del programa. Se verifica que los argumentos pasados al ejecutar el programa sean correctos y estén completos (archivo de matriz, métrica de similitud, número de vecinos y tipo de predicción), mostrando mensajes de error o ayuda cuando sea necesario. Si los parámetros son válidos, los guarda en una estructura *CommandLineArgs* para su posterior uso. Además, ofrece una opción *--help* o *-h* que muestra una descripción detallada del programa, sus opciones, uso correcto y ejemplos de ejecución.

tools.h

Se declaran las funciones de comprobación, uso e información del programa, además de declararse la estructura de los argumentos recibidos por línea de comando para que sea más fácil trabajar con ellos.

recommender-system.h

Se declara la clase ***RecommenderSystem***, que representa un sistema de recomendación colaborativo basado en usuarios. En él se declaran los atributos y métodos necesarios para leer una matriz de valoraciones entre usuarios e ítems, calcular similitudes entre usuarios, seleccionar los vecinos más parecidos y predecir valoraciones faltantes. Incluye métodos privados para realizar los cálculos internos (como medias, similitudes, elección de vecinos y predicciones) y métodos públicos para acceder a los atributos y generar recomendaciones. También se declara la sobrecarga del operador ‘<<’ para mostrar de forma formateada los resultados del sistema (matrices, similitudes y vecinos) por consola.

1. Análisis de los resultados matriz 5x10-1

Ejemplo 1 - Pearson/Simple

```
None
╔═══════════════════════════════════════════╗
║               PREDICTIONS                 ║
╚═══════════════════════════════════════════╝
Predicted score for user 1 and item 9: 2.68444
Predicted score for user 0 and item 8: 4.085
Predicted score for user 0 and item 3: 2.1225
Predicted score for user 2 and item 5: 0.883
Predicted score for user 2 and item 3: 2.1225
Predicted score for user 2 and item 9: 3.143
Predicted score for user 3 and item 1: 4.887
Predicted score for user 3 and item 2: 1.217
Predicted score for user 3 and item 9: 0.437


╔═══════════════════════════════════════════╗
║               INPUT DATA                  ║
╚═══════════════════════════════════════════╝
Metric: Pearson Correlation
Number of Neighbors: 5
Prediction Type: Simple Prediction


╔═══════════════════════════════════════════╗
║             ORIGINAL MATRIX               ║
╚═══════════════════════════════════════════╝
User 0: 3.142 2.648 1.649 - 1.116 0.883 0.423 3.976 - 3.143
User 1: 3.412 0.314 3.796 4.233 2.159 4.513 2.392 0.868 2.473 -
User 2: 4.408 4.495 2.052 - 0.051 - 3.355 3.739 4.085 -
User 3: 1.731 - - 1.511 4.866 2.217 3.003 2.901 2.113 -
User 4: 0.555 4.887 1.217 0.803 3.799 4.877 2.831 0.991 4.493 0.437


╔═══════════════════════════════════════════╗
║           SIMILARITIES MATRIX             ║
╚═══════════════════════════════════════════╝
User 0:   1.0000  -0.4606   0.5589  -0.3562  -0.5803
User 1:  -0.4606   1.0000  -0.2388  -0.5469  -0.2425
User 2:   0.5589  -0.2388   1.0000  -0.9400  -0.0485
User 3:  -0.3562  -0.5469  -0.9400   1.0000   0.3401
User 4:  -0.5803  -0.2425  -0.0485   0.3401   1.0000

╔═══════════════════════════════════════════
```

```
╔═══════════════════════════════════════════╗
║        CHOSEN NEIGHBORS PER USER          ║
╚═══════════════════════════════════════════╝
User 0: [ 2 ]
User 1: [  ]
User 2: [ 0 ]
User 3: [ 4 ]
User 4: [ 3 ]

╔═══════════════════════════════════════════╗
║      MATRIX WITH PREDICTED RATINGS        ║
╚═══════════════════════════════════════════╝
User 0: 3.142 2.648 1.649 2.122500 1.116 0.883 0.423 3.976 4.085000 3.143
User 1: 3.412 0.314 3.796 4.233 2.159 4.513 2.392 0.868 2.473 2.684444
User 2: 4.408 4.495 2.052 2.122500 0.051 0.883000 3.355 3.739 4.085 3.143000
User 3: 1.731 4.887000 1.217000 1.511 4.866 2.217 3.003 2.901 2.113 0.437000
User 4: 0.555 4.887 1.217 0.803 3.799 4.877 2.831 0.991 4.493 0.437
```

Datos de Entrada

- Matriz: 5 usuarios × 10 ítems
- Métrica: Coeficiente de Pearson
- Número de vecinos: 5
- Tipo de predicción: Predicción simple
- Rango de puntuaciones: 0 a 5

Observaciones

- Valores perdidos: 9 valores no puntuados (representados con "-")

Correlaciones más significativas

- Usuario 2 y Usuario 3: Correlación muy negativa (-0.9400), indicando gustos completamente opuestos
- Usuario 0 y Usuario 2: Correlación positiva moderada (0.5589), gustos similares
- En general, correlaciones negativas.

Selección de Vecinos

El algoritmo seleccionó solo 1 vecino por usuario, excepto el usuario 1, que no tiene vecinos elegibles. Esto se debe a que se estableció que no se pueden elegir vecinos como similitud negativa.

Análisis de las Predicciones

Predicciones destacables:

- Para los usuarios que solo tienen un vecino, sus predicciones serán la calificación de ese vecino al ítem.

- En el caso del usuario 1, que no tiene vecinos elegibles, se usa la media para la predicción de su ítem.

Observaciones Importantes

- Estas predicciones no son confiables, ya que al ser una matriz pequeña y con poca similitud entre usuarios, no se puede elegir un número considerable de vecinos por usuario.

Ejemplo 2 - Coseno/Simple

```
None
╔═══════════════════════════════════════╗
║              PREDICTIONS              ║
╚═══════════════════════════════════════╝
Predicted score for user 1 and item 9: 1.71259
Predicted score for user 0 and item 8: 3.30677
Predicted score for user 0 and item 3: 2.21139
Predicted score for user 2 and item 5: 3.03768
Predicted score for user 2 and item 3: 2.19946
Predicted score for user 2 and item 9: 1.84481
Predicted score for user 3 and item 1: 3.09507
Predicted score for user 3 and item 2: 2.17605
Predicted score for user 3 and item 9: 1.71289


╔═══════════════════════════════════════╗
║              INPUT DATA               ║
╚═══════════════════════════════════════╝
Metric: Cosine Distance
Number of Neighbors: 5
Prediction Type: Simple Prediction


╔═══════════════════════════════════════╗
║            ORIGINAL MATRIX            ║
╚═══════════════════════════════════════╝
User 0: 3.142 2.648 1.649 - 1.116 0.883 0.423 3.976 - 3.143
User 1: 3.412 0.314 3.796 4.233 2.159 4.513 2.392 0.868 2.473 -
User 2: 4.408 4.495 2.052 - 0.051 - 3.355 3.739 4.085 -
User 3: 1.731 - - 1.511 4.866 2.217 3.003 2.901 2.113 -
User 4: 0.555 4.887 1.217 0.803 3.799 4.877 2.831 0.991 4.493 0.437


╔═══════════════════════════════════════╗
║          SIMILARITIES MATRIX          ║
╚═══════════════════════════════════════╝
```

```
User 0:   1.0000   0.6156   0.9015   0.6932   0.5389
User 1:   0.6156   1.0000   0.7494   0.7799   0.6903
User 2:   0.9015   0.7494   1.0000   0.6857   0.7609
User 3:   0.6932   0.7799   0.6857   1.0000   0.8477
User 4:   0.5389   0.6903   0.7609   0.8477   1.0000

╔═══════════════════════════════════════╗
║        CHOSEN NEIGHBORS PER USER      ║
╚═══════════════════════════════════════╝
User 0: [ 1, 2, 3, 4 ]
User 1: [ 0, 2, 3, 4 ]
User 2: [ 0, 1, 3, 4 ]
User 3: [ 0, 1, 2, 4 ]
User 4: [ 0, 1, 2, 3 ]

╔═══════════════════════════════════════╗
║      MATRIX WITH PREDICTED RATINGS    ║
╚═══════════════════════════════════════╝
User 0: 3.142 2.648 1.649 2.211395 1.116 0.883 0.423 3.976 3.306770 3.143
User 1: 3.412 0.314 3.796 4.233 2.159 4.513 2.392 0.868 2.473 1.712591
User 2: 4.408 4.495 2.052 2.199464 0.051 3.037682 3.355 3.739 4.085 1.844812
User 3: 1.731 3.095071 2.176047 1.511 4.866 2.217 3.003 2.901 2.113 1.712894
User 4: 0.555 4.887 1.217 0.803 3.799 4.877 2.831 0.991 4.493 0.437
```

Datos de Entrada

- Matriz: 5 usuarios × 10 ítems
- Métrica: Distancia coseno
- Número de vecinos: 5
- Tipo de predicción: Predicción simple
- Rango de puntuaciones: 0 a 5

Observaciones

- Valores perdidos: 9 valores no puntuados (representados con "-")

Correlaciones más significativas

- Usuario 0 y Usuario 4: Correlación más baja, esta también era una correlación negativa al calcularla con Pearson(-0.5803).
- Usuario 0 y Usuario 2: Mayor correlación, cuadra con el ejemplo de Pearson, donde también son la mayor correlación.

Selección de Vecinos

El algoritmo seleccionó 4 vecinos por usuario(los máximos posibles), ya que no se especificó ningún valor mínimo de similitud.

Análisis de las Predicciones

- Vemos diferencias con Pearson, ya que en este caso no se puso ninguna restricción a la similitud mínima.

2. Análisis de los resultados matriz 10x25-1

Ejemplo 1 - Pearson/DistConMedia

[Enlace matrix-10-25-1.txt]

Datos de Entrada

- Matriz: 10 usuarios × 25 ítems
- Métrica: Coeficiente de Pearson
- Número de vecinos: 5
- Tipo de predicción: Diferencia con la media
- Rango de puntuaciones: 0 a 5

Observaciones

- Valores perdidos: 21 valores no puntuados (representados con "-")

Correlaciones más significativas

- Usuario 0 y Usuario 6: Mayor correlación (0.5945), indicando gustos similares.
- Usuario 7 y Usuario 9: Menor correlación (-0.4482), gustos diferentes.
- En general, correlaciones bajas, excepto algunos grupos como el formado por el usuario 0, 5 y 6.

Selección de Vecinos

Los usuarios 0, 1, y 5 tienen 5 vecinos. Los usuarios 2, 3, 6, 7, 9 tienen 4 vecinos, el usuario 4 tiene 4 y el usuario 8 tiene 2. Esto se debe a que se estableció que no se pueden elegir vecinos como similitud negativa.

Análisis de las Predicciones

Predicciones destacables:

- En general son predicciones medias-bajas, excepto el usuario 2 con el ítem 22 que se predice un 4.53749.

Ejemplo 2 - Euclídea/DistConMedia

[Enlace matrix-10-25-1.txt]

Datos de Entrada

- Matriz: 10 usuarios × 25 ítems
- Métrica: Distancia euclidiana
- Número de vecinos: 5
- Tipo de predicción: Diferencia con la media
- Rango de puntuaciones: 0 a 5

Observaciones

- Valores perdidos: 21 valores no puntuados (representados con "-")

Correlaciones más significativas

- Usuario 0 y Usuario 6: Mayor correlación, coincidiendo con el ejemplo anterior calculado con Pearson.
- Usuario 7 y Usuario 9: Menor correlación (-0.4482), coincidiendo con el ejemplo anterior calculado con Pearson.
- En general, correlaciones bajas.

Selección de Vecinos

Cada usuario tiene 5 vecinos, ya que no se especificó ningún valor mínimo de similitud.

Análisis de las Predicciones

Predicciones destacables:

- En general son predicciones medias-bajas, al usuario 2 con el ítem 22 ahora se le predijo un 3.06009, esto se puede deber a que el número de vecinos es diferente que con el ejemplo anterior. En general, las predicciones son parecidas a las del ejemplo anterior.

3. Análisis de los resultados matriz 25x100-1

Ejemplo 1 - Coseno/DistConMedia

[Enlace matrix-25-100-1.txt]

Datos de Entrada

- Matriz: 25 usuarios × 100 ítems
- Métrica: Distancia coseno
- Número de vecinos: 5
- Tipo de predicción: Diferencia con la media
- Rango de puntuaciones: 0 a 5

Observaciones

- Valores perdidos: 65 valores no puntuados (representados con "-")

Correlaciones más significativas

- Usuario 16 y Usuario 17: Mayor correlación (0.8391), indicando gustos similares.
- Usuario 0 y Usuario 6: Menor correlación (0.6834), gustos diferentes.
- En general, correlaciones medias.

Selección de Vecinos

Todos los usuarios tienen los 5 vecinos.

Análisis de las Predicciones

Predicciones destacables:

- Predicción más alta: 4.00859 al usuario 11 en el ítem 4.
- Predicción más baja: 1.032 al usuario 5 en el ítem 36.

Observaciones Importantes

- Al ser una matriz con mayor número de usuarios los cálculos son más realistas y efectivos.

Ejemplo 2 - Euclidean/Simple

[Enlace matrix-25-100-1.txt]

Datos de Entrada

- Matriz: 25 usuarios × 100 ítems
- Métrica: Distancia euclidiana
- Número de vecinos: 5
- Tipo de predicción: Predicción simple
- Rango de puntuaciones: 0 a 5

Observaciones

- Valores perdidos: 65 valores no puntuados (representados con "-")

Correlaciones más significativas

- Usuario 16 y Usuario 17: Mayor distancia, coincide con el ejemplo anterior.

Selección de Vecinos

Todos los usuarios tienen los 5 vecinos. Estos varían del ejemplo anterior.

Análisis de las Predicciones

Predicciones destacables:

- Predicción más alta: 4.2066 al usuario 11 en el ítem 4. El usuario e ítem coincide con el ejemplo anterior.
- Predicción más baja: 0.816714 al usuario 5 en el ítem 36. El usuario e ítem coincide con el ejemplo anterior.

Observaciones Importantes

- Al ser una matriz con mayor número de usuarios los cálculos son más realistas y efectivos.
- Al usar la inversa de la distancia euclidiana y tener tantos valores la similitud aparece muy pequeña.

4.Análisis de los resultados matriz 50x250-1

Ejemplo 1 - Euclidiana/Simple

Enlace matrix-50-250-1.txt

Datos de Entrada

- Matriz: 50 usuarios × 250 ítems
- Métrica: Distancia euclidiana
- Número de vecinos: 5
- Tipo de predicción: Predicción simple
- Rango de puntuaciones: 0 a 5

Observaciones

- Valores perdidos: 119 valores no puntuados (representados con "-")

Correlaciones más significativas

- Usuario 5 y Usuario 18: Mayor correlación (0.0352).
- Usuario 7 y Usuario 41: Menor correlación (0.0285).

Selección de Vecinos

Todos los usuarios tienen los 5 vecinos.

Análisis de las Predicciones

Predicciones destacables:

- Predicción más alta: 4.48192 al usuario 34 en el ítem 96.
- Predicción más baja: 1.16766 al usuario 29 en el ítem 11.

Observaciones Importantes

- Al ser una matriz con mayor número de usuarios los cálculos son más realistas y efectivos.
- Al usar la inversa de la distancia euclidiana y tener tantos valores la similitud aparece muy pequeña.

Ejemplo 2 - Euclidiana/Media

[Enlace matrix-50-250-1.txt]

Datos de Entrada
- Matriz: 50 usuarios × 250 ítems
- Métrica: Distancia euclidiana
- Número de vecinos: 5
- Tipo de predicción: Distancia con la media
- Rango de puntuaciones: 0 a 5

Observaciones
- Valores perdidos: 119 valores no puntuados (representados con "-")

Correlaciones más significativas(IGUAL QUE EL ANTERIOR)
- Usuario 5 y Usuario 18: Mayor correlación (0.0352).
- Usuario 7 y Usuario 41: Menor correlación (0.0285).

Selección de Vecinos
Todos los usuarios tienen los 5 vecinos.

Análisis de las Predicciones
Predicciones destacables:
- Predicción más alta: 4.55707 al usuario 34 en el ítem 96. El usuario e ítem coincide con la predicción simple del ejemplo anterior.
- Predicción más baja: 1.15842 al usuario 29 en el ítem 11. El usuario e ítem coincide con la predicción simple del ejemplo anterior.

Observaciones Importantes
- Al ser una matriz con mayor número de usuarios los cálculos son más realistas y efectivos.
- Al usar la inversa de la distancia euclidiana y tener tantos valores la similitud aparece muy pequeña.
- La similitud no cambia con respecto al ejemplo anterior, ya que también se usa la distancia euclidiana.
- Lo que nos interesa es ver la diferencia entre los 2 tipos de predicciones. Como podemos observar las predicciones en ambos ejemplos son muy parecidas.

5. Análisis de los resultados matriz 100x1000-1

Pearson/Simple

[Enlace matrix-100-1000-1.txt]

Datos de Entrada

- Matriz: 50 usuarios × 250 ítems
- Métrica: Coeficiente de Pearson
- Número de vecinos: 5
- Tipo de predicción: Predicción simple
- Rango de puntuaciones: 0 a 5

Observaciones

- Valores perdidos: 260 valores no puntuados (representados con "-")

Selección de Vecinos

Todos los usuarios tienen los 5 vecinos.

Análisis de las Predicciones

Predicciones destacables:

- Ninguna predicción muy alta.
- Predicciones medias-bajas.

Observaciones Importantes

- Al ser una matriz con muchos items, las similitudes son muy bajas.

Análisis general de resultados

Matriz	Usuarios	Items	Calificaciones Totales Posibles	Predicciones	Porcentaje Predicciones /Total
5×10	5	10	50	9	18.0%
10×25	10	25	250	21	8.4%
25×100	25	100	2,500	65	2.6%
50×250	50	250	12,500	119	0.952%
100×1000	100	1000	100,000	260	0.26%