



## CIENCIAS TÉCNICAS

### Premio Anual de la Academia de Ciencias de Cuba, 2020

# Métodos de procesamiento de datos para sistemas recomendadores de filtrado colaborativo

Raciel Yera Toledo <sup>1,5\*</sup> <https://orcid.org/0000-0001-9759-261X>

Yailé Caballero Mota <sup>2,4</sup> <https://orcid.org/0000-0002-6725-5812>

Jorge Castro Gallardo <sup>3</sup> <https://orcid.org/0000-0002-8390-8238>

Luis Martínez López <sup>3</sup> <https://orcid.org/0000-0003-4245-8813>

<sup>1</sup> Universidad de Ciego de Ávila. Ciego de Ávila, Cuba

<sup>2</sup> Universidad de Camagüey. Camagüey, Cuba

<sup>3</sup> Universidad de Jaén. Jaén, España

<sup>4</sup> Titular de la Academia de Ciencias de Cuba. La Habana, Cuba

<sup>5</sup> Joven Asociado de la Academia de Ciencias de Cuba. La Habana, Cuba

\*Autor para correspondencia: [ryera@unica.cu](mailto:ryera@unica.cu)

## RESUMEN

### Palabras clave

recomendación; inconsistencias; preferencias de los usuarios; ruido natural

**Introducción.** Los sistemas recomendadores ayudan a los usuarios a encontrar aquella información que mejor se corresponda con sus necesidades y preferencias, en un espacio de búsqueda sobrecargado de posibles opciones. Se han detectado muy pocos trabajos enfocados en preprocesar datos inconsistentes en sistemas recomendadores, con vistas a elevar su eficacia. **Métodos.** El objetivo de esta investigación es desarrollar nuevos métodos de preprocesamiento para la eliminación de inconsistencias de tipo ruido natural en sistemas recomendadores de filtrado colaborativo, que contribuyan a mejorar la eficacia de las recomendaciones generadas. Varios de los métodos propuestos como parte de la investigación, se apoyan en el uso de la lógica difusa. Como resultado se obtuvieron 4 nuevos métodos de preprocesamiento de datos para sistemas recomendadores de filtrado colaborativo, tanto para recomendadores individuales como grupales. **Resultados.** Los métodos propuestos fueron evaluados utilizando bases de datos internacionales concebidas con este fin, verificando su eficacia. En adición estos métodos se introdujeron en un escenario real de recomendación que es el de los jueces en línea de programación en Cuba, mostrándose una mejora en la eficacia de la recomendación en este escenario. Estos resultados fueron publicados en 8 artículos Grupo I MES, incluyendo revistas *Web of Science* de alto factor de impacto como *Knowledge-Based Systems* y *Decision Support Systems*. **Conclusiones.** Los métodos propuestos son, por tanto, integrables a cualquier sistema recomendador de filtrado colaborativo, siendo capaces de mejorar la eficacia de los mismos. Se evidencian además, aplicaciones de los mismos a escenarios como el de los jueces en línea de programación.



# Contributions to data preprocessing for collaborative filtering recommender systems

## ABSTRACT

### Keywords

recommendation; inconsistencies; user's preferences; natural noise

**Introduction.** Recommender systems are focused on helping users to find the information that best fits their preferences and needs, in a search space overloaded with possible options. Most of the research works in this area are focused on proposing new recommendation approaches that improve the accuracy of previous work. **Methods.** Based on this idea, the objective of this research is the development of new data preprocessing methods for removing natural noise in collaborative filtering recommender systems, in order to improve the accuracy of the generated recommendations. Several of the approaches proposed in this research are based on fuzzy logic for managing the uncertainty associated with the user's preferences as a relevant part of recommender systems. As a result, four new approaches for data preprocessing methods in collaborative recommender systems were obtained, both for individual and group recommendation. **Results.** The methods proposed were evaluated using international databases created for this purpose, thus verifying their accuracy. Furthermore, these methods were applied in a real recommendation situation associated with Programming Online Judges in Cuba, evidencing an improvement in the recommendation accuracy in such context. These results were published in 8 Group I papers, in Web of Science Journals such as Knowledge-Based Systems and Decision Support Systems. **Conclusions-**The methods proposed then can be integrated into any collaborative filtering recommendation system, in order to improve its accuracy. Such methods have been clearly applied to situations such as programming online judges.

## INTRODUCCIÓN

Los sistemas recomendadores (RSs) se han vuelto una tecnología de internet pertinente para superar el problema de carga excesiva de información. <sup>(1)</sup> Considerando que la información disponible está creciendo continuamente en la internet y otros escenarios similares, cada día se hace más difícil para los usuarios acceder a aquella información que mejor se corresponde con sus necesidades y preferencias. La naturaleza de los ítems de información es muy heterogénea: estos pueden ser tan diversos como los productos en una tienda electrónica, paquetes turísticos proporcionados por una agencia de viaje, música para escuchar en un sitio electrónico, materiales de aprendizaje electrónico, artículos científicos para leer, o tratamientos específicos para una persona enferma.

La mayoría de los sistemas recomendadores individuales y grupales asumen que las preferencias de los usuarios están libres de inconsistencias. Sin embargo, varios trabajos de investigación han apuntado que tales preferencias pueden contener "ruido", y que ese "ruido", denominado "ruido natural", puede afectar negativamente el desempeño del sistema de recomendación. En los últimos años, se ha determinado que los *ratings* de los usuarios pueden ser inconsistentes, ruidosos o

estar sesgados <sup>(2,3)</sup> y que tales inconsistencias son causadas por factores como condiciones personales, influencias sociales, contextos, o determinadas escalas de *ratings*. <sup>(4)</sup> Todo lo anterior, hace que el tratamiento de preferencias inconsistentes en sistemas recomendadores sea un aspecto clave para mejorar su desempeño. El objetivo de esta investigación es desarrollar nuevos métodos de preprocesamiento para la eliminación de inconsistencias de tipo "ruido natural" en sistemas recomendadores de filtrado colaborativo, que contribuyan a mejorar la eficacia de las recomendaciones generadas.

## MÉTODOS

Este trabajo presenta inicialmente 4 nuevos métodos para la corrección de preferencias inconsistentes en sistemas recomendadores de filtrado colaborativo. <sup>(5)</sup>

### Corrección de preferencias inconsistentes en recomendación individual

Inicialmente se presenta 1 nuevo método para corregir preferencias inconsistentes en recomendación individual. Estos resultados han sido previamente presentados en Yera *et al.* <sup>(3)</sup> en la revista *Knowledge-Based Systems*.

En este método, partiéndose de lo planteado por Amatriain *et al.* <sup>(6)</sup> sobre la presencia de *ratings* extremos (tanto bajos como altos) y de ratings medios, se propone clasificar cada rating en 3 clases diferentes acordes a su valor:

- Débil: si  $r(u,i) < \kappa$ .
- Medio: si  $\kappa \leq r(u,i) < \nu$ .
- Fuerte: si  $r(u,i) \geq \nu$ .

Siendo  $U$  e  $I$  el conjunto de usuarios e ítems, se propone agrupar las preferencias para cada usuario  $u$ , en los siguientes conjuntos  $W_u, A_u$  y  $S_u$

- Conjunto de **ratings** débiles provistos por el usuario  $u$ ,  $W_u$ :  
 $W_u = \{r(u,i) \mid \forall i \in I \text{ donde } r(u,i) < \kappa_u\}$
- Conjunto de **ratings** medios provistos por el usuario  $u$ ,  $A_u$ :  
 $A_u = \{r(u,i) \mid \forall i \in I \text{ donde } \kappa_u \leq r(u,i) < \nu_u\}$
- Conjunto de **ratings** fuertes provistos por el usuario  $u$ ,  $A_u$ :  
 $S_u = \{r(u,i) \mid \forall i \in I \text{ donde } r(u,i) \geq \nu_u\}$

Del otro lado, para cada ítem  $i$ , las preferencias son agrupadas en los conjuntos  $W_i, A_i$  y  $S_i$ :

- Conjunto de **ratings** débiles asignados al ítem  $i$ ,  $W_i$ :  
 $W_i = \{r(u,i) \mid \forall u \in U \text{ donde } r(u,i) < \kappa_i\}$
- Conjunto de **ratings** medios asignados al ítem  $i$ ,  $A_i$ :  
 $A_i = \{r(u,i) \mid \forall u \in U \text{ donde } \kappa_i \leq r(u,i) < \nu_i\}$
- Conjunto de **ratings** fuertes asignados al ítem  $i$ ,  $S_i$ :  
 $S_i = \{r(u,i) \mid \forall u \in U \text{ donde } r(u,i) \geq \nu_i\}$

Considerando estos conjuntos y las clases de los ratings, formalmente se definen nuevas clases asociadas a cada tendencia de usuario e ítem (tabla 1).

Una vez que cada *rating*, usuario e ítem ha sido clasificado, esta información se usa para encontrar *ratings* ruidosos analizando la presencia de contradicciones entre las clases. Con este propósito, se definen 3 grupos de clases análogas respectivamente asociadas a usuarios, ítems y *ratings* (tabla 1). En este sentido, se espera que si para un *rating* se

cumple que su usuario y su ítem corresponden al mismo grupo, entonces el *rating* también pertenezca a este grupo. De lo contrario, es posiblemente ruidoso.

Una vez que el método ha detectado los posibles *ratings* ruidosos, la siguiente fase manipula estos *ratings*. Se propone una estrategia que modifica cada posible *rating* ruidoso  $r(u,i)$ , a través del cálculo de un nuevo *rating*  $r^*(u,i)$  para el correspondiente usuario e ítem, a través de un algoritmo de filtrado colaborativo tradicional tomando como base los *ratings* disponibles. <sup>(7, 8)</sup> En caso de cumplirse la condición  $|r(u,i) - r^*(u,i)| > \delta$  (siendo  $\delta$  un umbral), entonces  $r(u,i)$  se reemplaza por  $r^*(u,i)$ . De lo contrario,  $r(u,i)$  mantiene su valor original.

### Corrección de preferencias inconsistentes en recomendación individual usando lógica difusa

Esta sección presenta un nuevo método para el tratamiento de ruido natural en sistemas recomendadores individuales, usando herramientas difusas <sup>(9)</sup> para lidiar con la incertidumbre asociada a este proceso (NNM-FT). Los resultados presentados en esta sección han sido previamente publicados en Yera *et al.* <sup>(5)</sup> y Yera *et al.* <sup>(10)</sup>

La propuesta se compone de las siguientes fases: a) perfilamiento difuso, b) detección de ruido, y c) corrección de ruido.

El perfilamiento difuso se enfoca en la modelación de las preferencias de los usuarios a través de una representación lingüística difusa que permite obtener perfiles difusos de usuarios, *items* y *ratings*. Estos perfiles son transformados a través del uso de computación con palabras, hacia perfiles modificados que resaltan sus tendencias a través de una manera flexible.

La fase de detección de "ruido" se enfoca en chequear si un *rating*  $r_{ui}$  dado es ruidoso, a través del análisis de la tendencia de evaluación por parte de los usuarios asociados  $u$  e  $i$ . Específicamente, los perfiles difusos son usados para identificar si el valor del rating coincide con las tendencias de su usuario y su ítem correspondiente, o si yace fuera de las

**Tabla 1.** Clases análogas

	Clases de usuario	Clases de ítem	Clases de <i>rating</i>
Grupo 1	Crítico	Débilmente preferido	Débil
Grupo 2	Promedio	Medianamente preferido	Medio
Grupo 3	Benevolente	Fuertemente preferido	Fuerte

tendencias detectadas. En este último caso, el *rating* es considerado como ruidoso.

La fase de corrección de ruido se centra en la corrección de los valores de los *ratings* identificados como ruidosos en la fase previa, basándose en el valor de su grado de “ruido”. Estos grados de “ruido” consideran cuán cerca está el perfil difuso del *rating* en relación con los perfiles difusos de usuario y de ítem. Una mayor brecha entre dicho perfil de *rating* y los perfiles de usuario y de ítem, implicaría un mayor grado de ruido por parte del *rating* asociado.

### Corrección de preferencias inconsistentes en recomendación grupal

Esta sección se enfoca en la presentación de métodos para el tratamiento de “ruido natural” en un sistema de recomendación grupal. Los métodos construidos toman como referencia el método de tratamiento de “ruido natural” para recomendación individual, presentado previamente. Estos métodos han sido publicados en Castro *et al.*,<sup>(11)</sup> en la revista *Decision Support Systems*.

Dos de los enfoques que se presentan, se centran en el preprocesamiento de datos a nivel local. Estos son el método de tratamiento de “ruido natural” a nivel local basado en información local del grupo (NNM-LL), y el método de tratamiento de ruido natural a nivel local utilizando información global de los usuarios. Un tercer enfoque propone realizar la corrección a nivel global de las preferencias de los usuarios, sin considerar el grupo al que pertenece (NNM-GG). Finalmente, se propone un cuarto enfoque que introduce una hibridación en cascada, donde primero se corrige a nivel global y posteriormente de manera local al nivel del grupo (NNM-H).

El algoritmo 1 formaliza la corrección en el enfoque NNM-LL. En este caso, para todos los *ratings*  $R_{G_a, \bullet}$  de cada grupo  $G_a$ , se realiza un proceso de preprocesamiento de datos utilizando el esquema de la subsección anterior, considerando sólo los *ratings* del propio grupo para caracterizar los usuarios y los ítems.

La corrección en el enfoque NNM-GG es esbozado en el algoritmo 2. En este caso, se realiza una corrección de “ruido natural” a nivel global, utilizando toda la información disponible para perfilar los correspondientes usuarios e ítems y detectar las inconsistencias. Este preprocesamiento se realiza previo a la construcción del modelo de recomendación, para realizar la recomendación grupal.

Finalmente, el algoritmo 3 presenta el método NNM-H utiliza los métodos de corrección presentados en los algoritmos 1 y 2. Específicamente, primero realiza una corrección global previa a la construcción del modelo de recomendación, y en un segundo momento realiza una corrección local para co-

regir alguna inconsistencia que no se haya corregido en el primer momento.

```

Data:  $G_a, R_{G_a, \bullet}$ 
Result:  $R_{G_a, \bullet}^*$ 
1 foreach  $r_{m_i, i_k}$  in  $R_{G_a, \bullet}$  do
2    $c_{r_{m_i, i_k}} = \text{Classify}(r_{m_i, i_k})$ 
3    $c_{m_i} = \text{Classify}(m_i, R_{m_i, \bullet})$ 
4    $c_{i_k} = \text{Classify}(i_k, R_{G_a, i_k})$ 
5   if  $(c_{m_i} = c_{i_k})$  and  $(c_{m_i} \neq c_{r_{m_i, i_k}})$  and  $(c_{m_i} \neq \text{variable})$  then
6      $r_{m_i, i_k}^* = \text{Predict}(R, m_i, i_k)$ 
7   else
8      $r_{m_i, i_k}^* = r_{m_i, i_k}$ 
9 return  $R_{G_a, \bullet}^*$ 

```

Algoritmo 1. Corrección de tipo NNM-LL

```

Data:  $R$ 
Result:  $R^*$ 
1 foreach  $r_{u_j, i_k}$  in  $R$  do
2    $c_{r_{u_j, i_k}} = \text{Classify}(r_{u_j, i_k})$ 
3    $c_{u_j} = \text{Classify}(u_j, R_{u_j, \bullet})$ 
4    $c_{i_k} = \text{Classify}(i_k, R_{\bullet, i_k})$ 
5   if  $(c_{u_j} = c_{i_k})$  and  $(c_{u_j} \neq c_{r_{u_j, i_k}})$  and  $(c_{u_j} \neq \text{variable})$  then
6      $r_{u_j, i_k}^* = \text{Predict}(R, u_j, i_k)$ 
7   else
8      $r_{u_j, i_k}^* = r_{u_j, i_k}$ 
9 return  $R^*$ 

```

Algoritmo 2. Corrección de tipo NNM-GG

```

Data:  $U, I, R, G$ 
1  $R^* = \text{NNMGG}(R)$ 
2  $\text{BuildRecommendationModel}(U, I, R^*)$ 
3 foreach  $G_a$  in  $G$  do
4    $R_{G_a, \bullet}^{**} = \text{NNMLG}(G_a, R_{G_a, \bullet}^*)$ 
5    $\text{recommendations}_{G_a} = \text{Recommend}(G_a, R_{G_a, \bullet}^{**})$ 

```

Algoritmo 3. Método NNM-H

### Corrección de preferencias inconsistentes en recomendación grupal usando lógica difusa

En este último resultado se obtuvo un método para el tratamiento de ruido natural en sistemas de recomendación grupales usando lógica difusa (NNMG-FT). Los resultados presentados en esta sección han sido previamente publicados en Castro *et al.*<sup>(12)</sup>

NNMG-FT analiza la base de datos de *ratings* para detectar *ratings* ruidosos y corregirlos para reducir su impacto en un sistema de recomendación grupal. NNMG-FT tiene 3 fases:

- Perfilamiento difuso: Genera una representación para los usuarios, ítems y *ratings* para caracterizarlos y facilitar su análisis en una fase siguiente. En adición a los perfiles generados en NNM-FT, también incluye un perfilamiento

del ítem basado en el grupo para modelar las preferencias del grupo considerando un ítem específico.

- Tratamiento global del “ruido”: Basado en los perfiles modelados en la fase previa, esta fase manipula el “ruido natural” de la base de datos del **rating** a un nivel global. Esta fase está conformada por las fases 2 y 3 del enfoque NNM-FT.
- Tratamiento local del “ruido”: Utiliza la información corregida que se obtuvo en el paso previo, para llevar a cabo un tratamiento de “ruido natural” enfocado en los **ratings** del grupo. Esta fase también está compuesta por las fases 2 y 3 del enfoque NNM-FT, pero tomando en consideración el nuevo perfil del ítem basado en grupo, en vez del perfil del ítem utilizado en NNM-FT.

Como resultado, se obtiene una base de datos de **ratings** sin ruido natural, la cual puede ser utilizada luego por un sistema de recomendación grupal.

### Un recomendador para el juez en línea de programación

Los jueces en línea de programación son herramientas informáticas que contienen una gran colección de ejercicios de programación a ser resueltos por sus usuarios (generalmente estudiantes). Su propósito fundamental es el de automatizar el proceso de compilación y evaluación a las soluciones dadas por los estudiantes a los problemas propuestos. Esta sección propone el desarrollo de un método de recomendación de problemas a resolver para jueces en línea de programación. Este resultado ha sido presentado en Yera y Martínez,<sup>(13)</sup> en la revista *Applied Intelligence*. En adición, una versión inicial del trabajo fue publicada en Yera y Caballero.<sup>(14)</sup>

La figura 1 presenta el esquema del sistema recomendador que se propone para los jueces en líneas de programación. Este recomendador está compuesto por 3 etapas:

-Construcción extendida de la matriz usuarios-problemas. Se propone la creación de una matriz binaria usuario-problema, donde  $M[u,p]=1$  si el usuario  $u$  resolvió el problema  $p$ , y 0

de lo contrario. Posteriormente esta matriz se enriquece con la información sobre los intentos fallidos de los usuarios de resolver cada uno de los problemas, acorde a las siguientes ecuaciones, en las que  $D(u,p).n$  es la cantidad de intentos fallidos del usuario  $u$  resolviendo el problema  $p$ . Los parámetros  $\lambda$  son determinados experimentalmente.

$$M[u,p] = 0 \rightarrow M^*[u,p] = \begin{cases} 0, \text{ if } D(u,p).n = 0 \text{ (no intentos de solución)} \\ 1, \text{ if } D(u,p).n \geq \lambda_{p1} \text{ (no resuelto, con muchos intentos de solución)} \\ 2, \text{ if } D(u,p).n > 0 \text{ and } D(u,p).n < \lambda_{p1} \text{ (no resuelto, con muchos intentos de solución)} \end{cases}$$

$$M[u,p] = 1 \rightarrow M^*[u,p] = \begin{cases} 3, \text{ if } D(u,p).n \geq \lambda_{p2} \text{ (resuelto, con muchos intentos de solución)} \\ 4, \text{ if } D(u,p).n < \lambda_{p2} \text{ (resuelto, con pocos intentos de solución)} \end{cases}$$

**Tratamiento de “ruido natural”.** Se aplica el enfoque presentado en la sección 2, realizando la corrección acorde a las siguientes ecuaciones:

$$M^*[u,p] = 3 \rightarrow M^{**}[u,p] = \begin{cases} 4, \text{ if } |C2_u| \geq 2|C1_u| \text{ and } |C2_p| \geq 2|C1_p| \\ 3, \text{ en otro caso} \end{cases}$$

$$M^*[u,p] = 4 \rightarrow M^{**}[u,p] = \begin{cases} 3, \text{ if } |C1_u| \geq 2|C2_u| \text{ and } |C1_p| \geq 2|C2_p| \\ 4, \text{ en otro caso} \end{cases}$$

Asumiendo que:  $C1_u = \{p \mid M^*[u,p] = 3\}$ ,  $C2_u = \{p \mid M^*[u,p] = 4\}$ ,  $C1_p = \{u \mid M^*[u,p] = 3\}$ , y  $C2_p = \{u \mid M^*[u,p] = 4\}$ .

**Recomendación de problemas.** Utilizando la matriz  $M^{**}$ , para recomendar problemas a resolver a un usuario, se buscan los  $k$  vecinos más cercanos tomando como base una versión modificada del coeficiente de coincidencia simple.<sup>(13)</sup> La similitud entre los usuarios así como el hecho de haberlo resuelto o no, se utiliza para ponderar cada problema a resolver, acorde a la siguiente ecuación:

$$w_p = \sum_{u \in \text{top}_k(x)} \text{sim}(x,u) * M[u,p]$$

El listado de recomendaciones finales se obtiene ordenando de forma descendente los problemas no resueltos

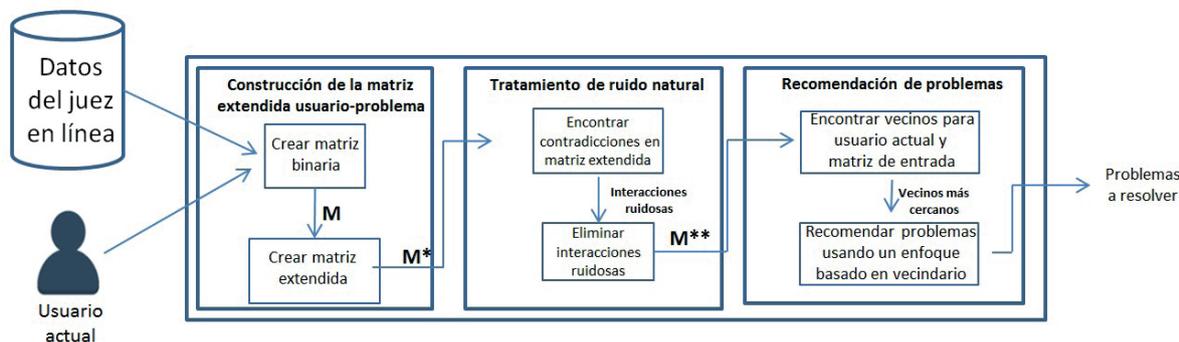


Fig. 1. Método para la recomendación de problemas a resolver en un juez en línea de programación.

acorde a su valor  $w_p$ . Una extensión de esta propuesta, integrando lógica difusa para una representación más precisa de la información, se desarrolló en Yera *et al.* <sup>(15)</sup>

## RESULTADOS Y DISCUSIÓN

La presente sección se centra en hacer una breve referencia a los resultados experimentales asociados a la evaluación de las presentes propuestas, realizándose de manera sintética por una cuestión de límite de espacio. Para más detalles, consultar las referencias asociadas a los métodos presentados. <sup>(3, 11, 13)</sup>

Se desarrollaron evaluaciones experimentales utilizando *datasets* bien conocidos en el área del conocimiento:

- **Movielens** 100K. Está compuesta por 100 000 *ratings* dados por 943 usuarios sobre 1682 filmes. Están en el rango [1; 5].
- **MovieTweeting**, <sup>(16)</sup> compuesta por alrededor de 140 000 *ratings* provistos por 21 018 usuarios sobre 12569 ítems, y en este caso los *ratings* están en el intervalo [0, 10].
- **Netflix Tiny**, compuesta por 4427 usuarios, 10 000 filmes, y 561 236 *ratings*. Es una versión pequeña del *dataset Netflix*.

Estas bases de datos son preparadas acorde al procedimiento propuesto por Gunawardana y Shani <sup>(17)</sup> para construir conjuntos de entrenamiento y prueba. En adición, el desempeño del método propuesto se evalúa utilizándose el protocolo definido previamente por Yera *et al.* <sup>(3)</sup> para sistemas recomendadores individuales.

Este protocolo se utiliza para comparar la propuesta inicial de preprocesamiento de datos presentados, contra trabajos de referencia previamente desarrollados. Se compara la propuesta de Yera *et al.* <sup>(3)</sup> contra O'Mahony *et al.* <sup>(18)</sup> y Li *et al.* <sup>(19)</sup> mostrándose el efecto posterior de la utilización de los datos corregidos, en modelos tradicionales de recomendación. En todos los casos la propuesta presentada, da lugar a un error medio absoluto menor en las recomendaciones.

Este protocolo se utiliza también para comparar el método NNM-FT con trabajos previamente relacionados. Específicamente, se compara contra los enfoques de O'Mahony <sup>(18)</sup> (DiffBased), Li *et al.* <sup>(19)</sup> (NNMU), y Yera *et al.* <sup>(3)</sup> (NN-Crisp), y un trabajo base que no considera ningún tratamiento de ruido natural. NNM-FT alcanza el mejor resultado en todos los casos, y prueba que se supera el desempeño de otros trabajos previos que no consideran el uso de la lógica difusa para el modelamiento de las preferencias de los usuarios.

Además, se realiza la evaluación de los métodos NNM-LL, NNM-LG, NNM-GG, NNM-H, presentados en Castro *et al.* <sup>(11)</sup> y del método NNMG-FT, <sup>(12)</sup> presentado previamente. Con este

fin, se utiliza el protocolo definido por De Pessemier *et al.* <sup>(20)</sup> específicamente para recomendación grupal.

Para medir el desempeño de cada método, se compara el desempeño (MAE) de varios recomendadores grupales usando los datos de preferencias después de la aplicación de estos métodos. Específicamente, se evalúan utilizando recomendadores grupales basados en agregación de recomendaciones, <sup>(20)</sup> e incorporan las estrategias de agregación promedio y "miseria mínima" por cada una de estas estrategias de agregación.

En la primera evaluación, el método NNM-H obtiene los mejores resultados para la mayoría de los escenarios. Sin embargo, es destacable que el método NNM-LG es también capaz de mejorar el método base, realizando únicamente la corrección a nivel de grupo y por tanto con mejor costo computacional.

En el caso del trabajo que considera el uso de técnicas difusas, se comparan 3 enfoques de tratamiento de "ruido natural" en recomendación grupal: a) NNMG-Crisp, centrado en tratamiento de ruido natural usando técnicas "duras", b) Base, que considera el uso del *dataset* original, y c) la presente propuesta NNMG-FT.

En el esquema de agregación de las recomendaciones usando la estrategia promedio, la mejora relativa de NNMG-FT en relación con el resto de los métodos fue uniforme para todos los tamaños de los grupos. En el caso de la agregación de recomendación con la estrategia de "miseria mínima", la mejora fue más grande que en el caso de la estrategia promedio debido a la mayor sensibilidad al "ruido" por parte de dicha estrategia. Específicamente, la mayor mejora fue obtenida para el caso de grupos grandes.

Para la evaluación del método para el juez en línea, se construyó una base de datos proveniente del Juez en Línea Caribeño, compuesta por 1910 usuarios, 584 problemas, y más de 14 8000 intentos de los usuarios de resolver problemas. Esta base de datos es preparada también acorde al procedimiento propuesto por Gunawardana y Shani <sup>(17)</sup> para construir conjuntos de entrenamiento y prueba. En este caso, como se trata de la tarea de recomendar un listado de ítems, se utilizan la métrica F1 para evaluar la eficacia de las recomendaciones.

La tabla 2 presenta la evaluación de la propuesta, así como su comparación con otros trabajos del estado del arte, <sup>(21)</sup> utilizando 130 vecinos para la generación de la recomendación. Se muestra que el método propuesto supera a otros trabajos del estado de arte por un amplio margen. Más detalles sobre este proceso de evaluación puede ser consultado en Yera y Martínez. <sup>(13)</sup>

**Tabla 2.** Evaluación del método de recomendación para jueces en línea de programación, utilizando la métrica F1

	5	10	15	20	25	30	35	40
Proposal	0,3875	0,3933	0,3784	0,3582	0,3397	0,3220	0,3055	0,2916
Binary	0,3614	0,3687	0,3540	0,3352	0,3189	0,3026	0,2895	0,2767
UCF-OJ	0,3833	0,3899	0,3736	0,3543	0,3367	0,3194	0,3035	0,2890
ICF-OJ	0,3602	0,3624	0,3494	0,3348	0,3191	0,3058	0,2932	0,2808

Tomado de Yera y Martínez. (13)

## Conclusiones

Esta memoria ha propuesto 4 métodos para la corrección de *ratings* inconsistentes en sistemas recomendadores, 2 de los cuales utilizan técnicas difusas para una representación más flexible de la información. Posteriormente, se construye un método de recomendación para un juez en línea de programación, tomando como base estos métodos. Los trabajos futuros están centrados en la incorporación de la dinámica temporal de las preferencias de los usuarios dentro de los modelos presentados.

## REFERENCIAS BIBLIOGRÁFICAS

- Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*. 2005;17(6):734-49.
- Amatriain X, Pujol JM, Oliver N, editors. I like it... i like it not: Evaluating user ratings noise in recommender systems. *Proceedings of the 17th International Conference on User Modeling, Adaptation and Personalization (UMAP)*; Trento, Italy; 2009.
- Yera R, Caballero Mota Y, Martínez L. Correcting noisy ratings in collaborative recommender systems. *Knowledge-Based Systems*. 2015;76:96-108.
- Pham HX, Jung JJ. Preference-based user rating correction process for interactive recommendation systems. *Multimedia Tools and Applications*. 2013;65(1):119-32.
- Yera R, Castro J, Martínez L. A fuzzy model for managing natural noise in recommender systems. *Applied Soft Computing*. 2016;40:187-98.
- Amatriain X, Pujol JM, Tintarev N, Oliver N, editors. Rate it again: Increasing recommendation accuracy by user re-rating. *Proceedings of 3rd ACM International Conference on Recommender Systems*; 2009; New York, USA.
- Ekstrand M, Riedl J, Konstan JA. Collaborative Filtering Recommender Systems. *Foundations and Trends in Human-Computer Interaction*. 2010;4(2):81-173.
- Ricci F, Rokach L, Shapira B, Kantor P. *Recommender Systems Handbook*. New York: Springer; 2011.
- Yera R, Martínez L. Fuzzy tools in recommender systems: A survey. *International Journal of Computational Intelligence Systems*. 2017;10(1):776-803.
- Yera R, Castro J, Martínez L, editors. An approach for natural noise management in recommender systems using fuzzy logic. *Uncertainty Modelling in Knowledge Engineering and Decision Making: Proceedings of the 12th International FLINS Conference*; World Scientific; 2016.
- Castro J, Yera R, Martínez L. An empirical study of natural noise management in group recommendation systems. *Decision Support Systems*. 2017;94:1-11.
- Castro J, Yera R, Martínez L. A fuzzy approach for natural noise management in group recommender systems. *Expert Systems with Applications*. 2018;94:237-49.
- Yera R, Martínez L. A recommendation approach for programming online judges supported by data preprocessing techniques. *Applied Intelligence*. 2017;47(2):277-90.
- Yera R, Caballero Y. An e-Learning Collaborative Filtering Approach to Suggest Problems to Solve in Programming Online Judges. *International Journal of Distance Education Technologies*. 2014;12(2):51-65.
- Yera R, Caballero Mota Y, Martínez L. A recommender system for programming online judges using fuzzy information modeling. *Informatics-Basel*. 2018;5:17.
- Dooms S, De Pessemier T, Martens L, editors. *MovieTweatings: a Movie Rating Dataset Collected From Twitter*. *Workshop on Crowdsourcing and Human Computation for Recommender Systems (CrowdRec at RecSys)*; Hong Kong. 2013.
- Gunawardana A, Shani G. A Survey of Accuracy Evaluation Metrics of Recommendation Tasks. *Journal of Machine Learning Research*. 2009;10:2935-62.
- O'Mahony MP, Hurley NJ, Silvestre GCM, editors. Detecting noise in recommender system databases. *Proceedings of the 11th international conference on Intelligent user interfaces*; 2006; Sydney, Australia: ACM.
- Li B, Chen L, Xingquan Z, Chengqi Z. Noisy but non-malicious user detection in social recommender systems. *World Wide Web*. 2013;16(5-6):677-99.
- De Pessemier T, Dooms S, Martens L. Comparison of group recommendation algorithms. *Multimedia tools and applications*. 2014;72(3):2497-541.
- Yu R, Cai Z, Du X, He M, Wang Z, Yang B, et al. The research of the recommendation algorithm in online learning. *International Journal of Multimedia and Ubiquitous Engineering*. 2015;10(4):71-80.

Recibido: 21/05/2021

Aprobado: 17/08/2021

### Conflicto de intereses

Los autores declaran que no hay conflictos de intereses con el trabajo presentado.

### Contribuciones de los autores

- Conceptualización: Raciél Yera Toledo, Yailé Caballero Mota, Jorge Castro Gallardo, Luis Martínez López
- Curación de datos: Raciél Yera Toledo
- Análisis formal: Raciél Yera Toledo, Yailé Caballero Mota, Jorge Castro Gallardo, Luis Martínez López
- Metodología: Raciél Yera Toledo, Yailé Caballero Mota, Jorge Castro Gallardo, Luis Martínez López
- Software: Raciél Yera Toledo, Jorge Castro Gallardo
- Supervisión: Yailé Caballero Mota, Luis Martínez López

- Validación: Raciél Yera Toledo, Jorge Castro Gallardo
- Redacción-borrador original: Raciél Yera Toledo
- Redacción-revisión y edición: Yailé Caballero Mota, Luis Martínez López

### Financiación

No se utilizó financiamiento externo, específico para este trabajo.

### Cómo citar este artículo

Yera Toledo R, Caballero Mota Y, Castro Gallardo J, Martínez López L *et al.* Métodos de procesamiento de datos para sistemas recomendadores de filtrado colaborativo. Anales de la Academia de Ciencias de Cuba [internet] 2022[citado en día, mes y año];12(1): e1056. Disponible en: <http://www.revistaccuba.cu/index.php/revacc/article/view/1056>

