Un estudio de robustez en recomendación basada en vecinos cercanos

Carmen Bonilla[1] and Alejandro Bellogín[1][0000−0001−6368−2510]

Universidad Autónoma de Madrid, Madrid, Spain
carmen.bonillau@estudiante.uam.es, alejandro.bellogin@uam.es

Abstract. Los sistemas de recomendación se han convertido en una herramienta fundamental para guiar a los usuarios en entornos digitales caracterizados por catálogos de productos amplios y heterogéneos. Entre los distintos enfoques existentes, los algoritmos basados en vecinos más cercanos destacan por su simplicidad conceptual y su amplia adopción en aplicaciones reales. No obstante, cuando se aplican a escenarios con grandes volúmenes de datos y matrices de valoraciones altamente dispersas, surgen desafíos relacionados con la fiabilidad y el comportamiento de las recomendaciones generadas, lo que pone de manifiesto la importancia de analizar la robustez de estos sistemas. El presente trabajo aborda el estudio de la robustez en sistemas basados en vecinos y se propone un marco de análisis que permite evaluar el comportamiento de estos algoritmos bajo diferentes condiciones de los datos y configuraciones del modelo. Se consideran varios conjuntos de datos con características estructurales diversas analizando cómo estas características influyen en el comportamiento del algoritmo desde distintas perspectivas de robustez.

Keywords: Sistemas de recomendación · Vecinos cercanos · Robustez

1 Introducción

Los sistemas de recomendación comprenden una serie de técnicas destinadas a generar sugerencias personalizadas de ítems para los usuarios, facilitando la búsqueda de información relevante en entornos con un elevado volumen de opciones [21, 1]. En la actualidad, el crecimiento exponencial de los catálogos digitales ha incrementado la necesidad de este tipo de sistemas, que desempeñan un papel fundamental en múltiples aplicaciones al mejorar la experiencia del usuario mediante el filtrado eficiente de contenidos.

Entre los distintos enfoques existentes, los algoritmos basados en vecinos más cercanos (k-Nearest Neighbors, k-NN) destacan por su simplicidad y su amplia adopción en entornos reales [16]. Estos métodos generan recomendaciones a partir de la similitud entre usuarios o ítems, constituyendo una aproximación de referencia en filtrado colaborativo. Sin embargo, en escenarios caracterizados por grandes volúmenes de datos y matrices altamente dispersas, el comportamiento de estos algoritmos puede deteriorarse, produciendo recomendaciones inestables o poco consistentes. En este contexto, la robustez emerge como una propiedad clave para su evaluación.

No obstante, el concepto de robustez es heterogéneo y sin una definición unificada en la literatura, lo que ha dado lugar a múltiples interpretaciones que abarcan desde la

estabilidad frente a perturbaciones en los datos hasta la resistencia a ruido o ataques adversarios. El presente trabajo realiza tres contribuciones principales. En primer lugar, propone una revisión del concepto de robustez en sistemas de recomendación, organizando diversas definiciones presentes en la literatura bajo un conjunto común de dimensiones que permiten comparar sus solapamientos y diferencias. En segundo lugar, se introduce una métrica operativa de robustez basada en la reproducibilidad de las predicciones, que cuantifica el grado de estabilidad de los algoritmos k-NN ante la incorporación de componentes de aleatoriedad en el proceso de entrenamiento y en los datos. Finalmente, se lleva a cabo un análisis empírico sistemático sobre distintos conjuntos de datos con diferentes niveles de sparsity, evaluando el impacto de la configuración del modelo —en particular, el tamaño del vecindario y la métrica de similitud— sobre la robustez y el error predictivo, permitiendo caracterizar el comportamiento de estos algoritmos bajo diversas condiciones experimentales.

2 Definiciones de robustez en recomendación

El término de robustez puede ser interpretado de maneras muy diversas, no sólo si se tienen en cuenta algoritmos de distintas naturalezas, sino también dentro de un mismo ámbito. Dada la heterogeneidad del concepto, se han analizado sus diversas definiciones tanto en el contexto de los sistemas de recomendación como en otras disciplinas, con el fin de identificar la más adecuada para este trabajo. Asimismo, se ha procurado adaptar cada una de ellas al contexto específico de los k-NN. A continuación, se presentan las diez definiciones más relevantes encontradas en la literatura, sin desestimar la importancia de las restantes.

La selección de definiciones se realizó a partir de una revisión exploratoria en bases de datos académicas, utilizando términos relacionados con *robustness* y *stability*, en el contexto de sistemas de recomendación. También se revisó material potencialmente valioso aplicado a otros dominios de aprendizaje automático. En total, se preseleccionaron 53 artículos con potencial relevancia para el estudio del concepto, de los cuales se extrajeron diez definiciones representativas para el objetivo del presente estudio.

2.1 Pairwise Robustness

En [10] se habla de dos términos de robustez en el ámbito de la clasificación, diferenciándose entre la '*Pointwise Robustness*' y la '*Pairwise Robustness*'. El primer concepto hace referencia a la robustez de los modelos clásicos y se define como la mínima perturbación en los datos necesaria para alterar la etiqueta de clasificación. Sin embargo, se hace hincapié en el segundo, el cual es útil cuando los datos de entrada se corresponden con pares de instancias en vez de con instancias individuales.

2.2 Root Mean Squared Shift

En [2] y [1] encontramos un concepto asimilable a robustez pero con distinto nombre: estabilidad. Se define como la medida en la que las predicciones del sistema permanecen iguales o parecidas cuando se introducen nuevas calificaciones para ciertos ítems, coincidiendo estas calificaciones reales con las ya predichas por el sistema.

2.3 Robustez de las redes sociales

El artículo [5] se centra en las OSN (*Online Social Networks*) para medir la robustez. Se hace a través del valor llamado *coreness*, este se obtiene de la red de interacciones dirigida. Una OSN se considera robusta si el impacto provocado por eliminar un nodo no provoca cascadas de desconexiones entre nodos de dicha red.

2.4 Robustez para Aprendizaje métrico

En [3], los autores tratan de adaptar la noción de robustez de los algoritmos al aprendizaje métrico, tomando una definición estadística aplicada al ámbito de la clasificación. Con este fin se toma el espacio $Z = X \times Y$, donde X denota al conjunto de instancias e Y al conjunto de etiquetas. Además los algoritmos a analizar cuentan con una determinada función de distancia o similitud, siendo esta $f : X \times X \longrightarrow \mathbb{R}$. A cada par de instancias $z_1, z_2 \in Z$ y cada función f se les asocia una función de pérdida $l(f, z_1, z_2)$. Usando esta notación, un algoritmo se considera robusto si, cuando dos instancias del conjunto de *test* son cercanas a otras dos del conjunto de *train*, sus funciones de pérdida resultan igualmente próximas.

2.5 Robustness Gym for RecSys (RGRecSys)

Otra posibilidad es considerar que la robustez no es un término definido por si solo, sino que un algoritmo es robusto cuando engloba una serie de propiedades. En [18] se hace uso de una definición holística considerando que se conforma por diversas dimensiones, creando una librería (RGRecSys) que evalúa las mismas.

La primera dimensión está relacionada con las subpoblaciones, considerando que un recomendador es robusto si su desempeño no empeora al tomar tan solo un grupo de ítems o usuarios en lugar de la población en su totalidad. En segundo lugar, muchos algoritmos consideran que los datos de entrenamiento y testeo son independientes e idénticamente distribuidos, por lo que se evalúa la robustez alterando esta asunción. Concretamente, se realizan pruebas en las que la distribución por género de los usuarios difiere entre los conjuntos de *train* y *test*. También se examina la robustez frente a transformaciones o corrupción de datos, midiendo la eficacia del sistema cuando parte de la información disponible es errónea. Para evaluar este aspecto, se altera la información sobre la edad de los usuarios y se cuantifica la variación del error del modelo.

Otro aspecto a analizar es la robustez del sistema frente a ataques, existen ataques producidos de diversas formas con la intención de alterar los rankings para determinados ítems. La última de las dimensiones es sobre la escasez de datos: se mide el desempeño del sistema cuando falta una determinada fracción de interacciones. Concretamente, se simula esta situación suprimiendo el 25% de las interacciones de cada usuario.

2.6 Simplicity Bias

Otra propuesta para la medición de la robustez es a través del análisis del '*Simplicity Bias*' [19]. El trabajo se basa en tomar un par de instancias x y $\bar{x}$, siendo la segunda de ellas más compleja (esta definición depende del dominio [19]). Un clasificador

sufrirá de este fenómeno si, fijando x y alterando aleatoriamente $\bar{x}$, la predicción del clasificador no cambia. Por el contrario, fijando $\bar{x}$ y alterando x la precisión del sistema disminuye hasta la precisión aleatoria base.

2.7 Hit Ratio

En [9] la robustez se define como la estabilidad de las recomendaciones ante la presencia de información falsa o manipulada, habitualmente introducida a través de ataques al sistema. Posteriormente, en [4] se propone evaluar esta robustez mediante la diferencia media de predicción o, en el caso de sistemas basados en ranking, utilizando el *'Hit Ratio'*.

2.8 Ranking List Sensitivity

Prosiguiendo con las medidas de robustez aplicadas a rankings, se tiene '*Ranking List Sensitivity*' (RSL) [17]. Esta mide la variación de un ránking en la fase de *test* como resultado de una perturbación en los datos de entrenamiento. A menor variación, se considera que el sistema cuenta con mayor robustez.

Para medir dicha variación el RLS hace uso de dos mecanismos distintos. En primer lugar, '*Rank Biased Overlap*' (RBO) mide la similitud en el orden de todos los elementos de los ránkings, mientras que '*Jaccard Score*' compara la superposición de los top-N elementos de los ránkings, sin tener en cuenta el orden. Además se proponen dos planteamientos distintos para la perturbación de los datos de entrenamiento: la primera de ellas consiste en perturbaciones aleatorias; la segunda, denominada '*CASPER*' (*Cascade Based)*, se trata de un sistema que busca la vulnerabilidad del modelo por atacantes para manipular las potenciales recomendaciones.

2.9 Robustez del entrenamiento

En [21] se realiza una revisión sobre la robustez de los sistemas de recomendación, resultando especialmente interesantes las subclasificaciones que se proponen. En primer lugar, los autores identifican tres fases en las que la robustez desempeña un papel relevante: entrenamiento, inferencia y evaluación del modelo. Posteriormente, centran su atención en la fase de entrenamiento, definiendo la robustez como la capacidad del modelo para preservar su desempeño ante la presencia de datos corruptos. En este contexto, se distinguen dos tipos de robustez: la robustez ante ataques maliciosos y la robustez frente al 'ruido natural' que existe en los datos.

2.10 Reproducibilidad aleatoria

Otro término a relacionar con la robustez es la reproducibilidad, siendo un algoritmo reproducible si es resistente a variaciones en sus muestras. En [11] dicha resistencia se mide analizando si el *output* obtenido en la ejecución de dos muestras de la misma población es el mismo con alta probabilidad. Es decir, se analiza la probabilidad de que un algoritmo devuelva la misma respuesta cuando se ejecuta dos veces utilizando diferentes muestras aleatorias pero con la misma aleatoriedad interna.

3 Propuesta de robustez para vecinos cercanos

Dada la heterogeneidad del término de robustez en la literatura y con el fin de clarificar las distintas perspectivas abordadas, se presenta una tabla comparativa que sintetiza las definiciones más relevantes del repaso bibliográfico. En particular, la Tabla 1 clasifica los diez enfoques presentados según las siguientes dimensiones:

1. Entrada: Perturbaciones o cambios en los datos de entrada (usuarios, ítems o valoraciones).
2. Multidimensional: Considera múltiples factores simultáneamente, como subpoblaciones, distribución *train-test* o escasez de datos.
3. Ataques: Robustez frente a manipulaciones maliciosas o corrupción intencionada de datos.
4. Sensibilidad: Dependencia de la estructura de usuarios o ítems, por ejemplo, usuarios con pocas valoraciones o ítems poco conectados.
5. Reproducibilidad: Capacidad del sistema para producir resultados consistentes ante diferentes muestras o ejecuciones.

La clasificación propuesta pone de manifiesto que no existe una definición única de robustez que abarque todas las dimensiones consideradas. Esta diversidad justifica la necesidad de organizar las distintas propuestas bajo criterios comunes, con el objetivo de identificar solapamientos, complementariedades y carencias. De esta forma, se ofrece una visión comparativa que permite clarificar qué aspectos de la robustez aborda cada definición y cuáles quedan fuera de su alcance.

Dado el carácter abstracto de las definiciones analizadas, se observan solapamientos conceptuales entre varias de las dimensiones consideradas. En particular, la robustez frente a perturbaciones en la entrada y la reproducibilidad pueden interpretarse como manifestaciones de una misma propiedad subyacente: la estabilidad del modelo ante variaciones en los datos. Asimismo, la robustez frente a ataques puede entenderse como un caso particular de perturbaciones en la entrada, caracterizado por una intencionalidad adversaria. Como consecuencia, distintas métricas empleadas para evaluar la robustez pueden resultar equivalentes en la práctica, aun cuando su formulación y motivación teórica difieran.

Esta falta de fronteras nítidas entre dimensiones refuerza la idea de que la robustez es un concepto multifacético. Precisamente, esta observación motiva la adopción de una definición específica en el marco de este trabajo, que busca capturar la robustez

Table 1. Comparativa de definiciones de robustez.

Sección	Entrada	Multidimensional	Ataques	Sensibilidad	Reproducibilidad
2.1	✓			✓	
2.2					✓
2.3				✓	
2.4				✓	
2.5	✓	✓	✓		
2.6	✓			✓	
2.7		✓	✓		
2.8	✓		✓		
2.9	✓		✓		
2.10					✓

desde un enfoque estadístico y reproducible, alineado con el comportamiento observable de los algoritmos k-NN (aunque se podría generalizar a otras familias de algoritmos) bajo distintos escenarios experimentales.

Tras el análisis de las distintas nociones de robustez aplicables a los sistemas de recomendación basados en k-NN, se considera que la definición presentada en la Sección 2.10 es la más general desde un punto de vista estadístico, por lo que se ajusta bien a los objetivos de este trabajo. En consecuencia, dicha definición se adopta como base conceptual del estudio.

Bajo este enfoque, un algoritmo k-NN se considera robusto si, al introducir una determinada componente de aleatoriedad en su funcionamiento, las predicciones resultantes se mantienen similares a las obtenidas por el mismo algoritmo en ausencia de dicha aleatoriedad. Esta concepción de la robustez pone el foco en la reproducibilidad del modelo, entendida como la capacidad de producir resultados consistentes frente a variaciones controladas en los datos o en el proceso de entrenamiento pero también, como se ha discutido, reconoce la importancia de los datos de entrada y de la sensibilidad del modelo.

De manera general, para evaluar el grado de robustez del algoritmo, se ejecutan dos versiones del modelo: una versión base, sin aleatoriedad, y una versión aleatorizada. A continuación, se mide el grado de coincidencia entre las predicciones generadas por ambas versiones. Concretamente, se estima la probabilidad de que la valoración predicha por el algoritmo base, denotada como r_{base}, coincida con la valoración predicha por el algoritmo aleatorizado, $r_{aleatorio}$, lo que se expresa formalmente como:

$$prob(r_{base} = r_{aleatorio}) \tag{1}$$

Para el cálculo de esta probabilidad, se adopta una formulación binaria de coincidencia exacta. Sea R el conjunto de valoraciones predichas y sea $r_{base}(u_n,i_m)$ y $r_{aleatorio}(u_n,i_m)$ la valoración predicha para el usuario u_n sobre el ítem i_m por el modelo base y el modelo aleatorizado, respectivamente. Se define entonces la probabilidad introducida en la Ecuación 1 como la media sobre el conjunto total de predicciones, usando la función indicatriz $\mathbb{1}$:

$$prob(r_{base} = r_{aleatorio}) = \frac{1}{|R|} \sum_{n,m} \mathbb{1}(r_{base}(u_n,i_m) = r_{aleatorio}(u_n,i_m)) \tag{2}$$

De este modo, la probabilidad se expresa en escala porcentual y representa el grado de coincidencia entre las predicciones del modelo base y del modelo aleatorizado. En consecuencia, un valor elevado de esta probabilidad se interpreta como un mayor grado de robustez, ya que indica que el algoritmo es capaz de mantener un comportamiento favorable frente a la introducción de aleatoriedad en su funcionamiento.

4 Experimentos

Con el objetivo de analizar de manera sistemática el comportamiento de los algoritmos k-NN desde la perspectiva de la robustez, nuestros experimentos se estructuran en torno a tres preguntas de investigación. Primero, se estudia el impacto del tamaño del

vecindario (k) sobre la estabilidad del modelo (**RQ1**). Segundo, se evalúa la influencia de la métrica de similitud empleada en el comportamiento robusto del sistema (**RQ2**). Finalmente, se analiza cómo el nivel de dispersión (sparsity) de los datos condiciona la robustez de las recomendaciones generadas (**RQ3**). Estas preguntas guían el diseño experimental y permiten interpretar los resultados obtenidos de forma estructurada.

4.1 Datos y métricas de evaluación

Se han utilizado las siguientes colecciones sobre diferentes dominios:

- *MovieLens 100K* [12]: Este conjunto de datos consta de 100.000 valoraciones (del 1 al 5) de 943 usuarios sobre 1.682 películas, donde cada usuario ha valorado al menos 20 películas.
- *Gift Cards* (2018) [15]: Este conjunto consta de 147.194 valoraciones (del 1 al 5) de 128.877 usuarios para 1.548 tarjetas de regalo vendidas por Amazon.
- *Musical Instruments* (2014) [13]: Este conjunto consta de 500.176 valoraciones (del 1 al 5) de 83.046 usuarios para 339.231 instrumentos musicales.

MovieLens presenta el menor nivel de *sparsity*, con un valor del 93,70%, seguido de *Gift Cards*, con un 99,93%. Por último, *Musical Instruments* alcanza un nivel de *sparsity* próximo al 100% (99,998%).

Los experimentos se evalúan usando las métricas de error clásicas MAE y RMSE, a pesar de que se sabe que no siempre constituyen el criterio más adecuado para evaluar la calidad de un sistema de recomendación [14]. No obstante, en el contexto de este trabajo, MAE y RMSE (no incluido por falta de espacio) resultan apropiadas por varias razones. En primer lugar, estas métricas permiten cuantificar de forma directa la magnitud de las desviaciones en las predicciones numéricas cuando el modelo es sometido a una componente de aleatoriedad. Adicionalmente, el análisis conjunto de la métrica de robustez con MAE y RMSE permite estudiar si las variaciones introducidas, ya sea por la componente de aleatoriedad o por cambios en los hiperparámetros del modelo, afectan únicamente a la reproducibilidad de las predicciones o también a su error. En particular, resulta de interés identificar escenarios en los que modificaciones en parámetros como el número de vecinos o la métrica de distancia no solo alteran el grado de robustez del algoritmo, sino que también se traducen en un empeoramiento (o mejora) del error predictivo.

Por otro lado, las métricas de ranking o de predicción de uso no resultan tan adecuadas bajo este escenario. Ambos tipos de métricas están orientados a evaluar el rendimiento final del sistema de recomendación [9], mientras que el objetivo principal de este trabajo es analizar la estabilidad interna de las predicciones numéricas de los algoritmos k-NN bajo distintas fuentes de aleatoriedad.

4.2 Configuración

Para la ejecución de las pruebas experimentales, se lleva a cabo un proceso de diseño de la componente de aleatorización a introducir en el sistema k-NN. Adaptando la definición vista en la Sección 2.10, y de acuerdo a la explicación aportada en la Sección 3, los

experimentos se implementan ejecutando el algoritmo k-NN un número de iteraciones determinadas, agregando en cada iteración una componente de aleatoriedad. Se contemplan dos enfoques distintos para introducir este factor de aleatoriedad en el modelo:

1. Aleatoriedad sobre el sistema de recomendación.
   El modelo base se entrena mediante un algoritmo k-NN utilizando una muestra que representa el 90% del conjunto de datos completo. Esta muestra de entrenamiento no se selecciona de forma aleatoria, sino que se construye excluyendo las valoraciones más recientes del sistema, identificadas a través de la variable *Timestamp*, independientemente del usuario o ítem al que correspondan.
   Por su parte, el modelo aleatorio se entrena utilizando una submuestra aleatoria del conjunto de entrenamiento del modelo base, que abarca aproximadamente el 80% del total del conjunto de datos.
   En cada iteración, el modelo aleatorio se entrena con una submuestra de entrenamiento diferente a través de la variación de la semilla de aleatoriedad. Ambos modelos, base y aleatorio, se evalúan sobre una muestra de *test* fija, compuesta por las valoraciones más recientes del conjunto de datos; es decir, el conjunto complementario al de entrenamiento del modelo base.
2. Aleatoriedad sobre el conjunto de datos.
   Con el propósito de analizar la sensibilidad del modelo frente a la aleatoriedad presente en los datos de entrenamiento, se plantea generar el modelo base y el modelo aleatorio conforme a los siguientes criterios. El modelo base se entrena mediante un algoritmo k-NN utilizando el conjunto de entrenamiento original, sin aplicar ningún tipo de alteración sobre las valoraciones.
   Por su parte, el modelo aleatorio se entrena sobre una versión corrompida del mismo conjunto de entrenamiento, obtenida mediante la aplicación de la técnica de *flip aleatorio*. Esta técnica consiste en modificar aleatoriamente el valor de un subconjunto de las puntuaciones, sustituyendo el 10% de los ratings por nuevos valores seleccionados aleatoriamente dentro del rango posible (1–5).

Como hiperparámetros del modelo se consideran el número de vecinos y la métrica de similitud. Estas métricas se seleccionan a través de *distance_metrics* [20]: *cityblock*, *cosine*, *euclidean*, *haversine*, *l2*, *l1*, *manhattan*, *nan_euclidean* y *minkowski*. Sin embargo, como las métricas *euclidean*, *l2* y *minkowski* presentan resultados prácticamente idénticos, en los experimentos se emplea únicamente la distancia *euclidean*; igualmente, las distancias *cityblock*, *l1* y *manhattan* obtienen resultados equivalentes, por lo que se selecciona la métrica *cityblock*.

4.3 Resultados

En esta sección se presentan los experimentos donde se analiza el impacto que tiene la variación de la métrica de distancia y del tamaño del vecindario sobre la robustez.

Robustez frente a la aleatorización del modelo Los resultados obtenidos (Figura 1) permiten extraer varias conclusiones generales sobre el comportamiento del modelo. En primer lugar, se observa un incremento progresivo de la robustez conforme aumenta el número de vecinos (RQ1). Este patrón se mantiene de forma consistente

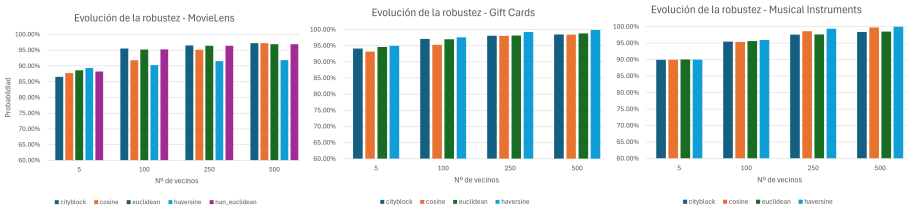


Fig. 1. Robustez frente a la aleatorización del sistema en MovieLens (izquierda), Gift Cards (medio) y Musical Instruments (derecha).

en todas las métricas evaluadas (RQ2) y conjuntos de datos (RQ3), posiblemente relacionado con el efecto natural que se obtiene al agregar un número mayor de predicciones, estabilizándolas en el proceso. Además, a partir de aproximadamente 500 vecinos, las diferencias entre métricas se reducen de manera significativa, alcanzándose un comportamiento estable y prácticamente convergente, lo que sugiere que, en vecindarios amplios, la elección de la métrica pierde relevancia en términos de robustez.

En cuanto al análisis de comparación entre métricas (RQ2), *cityblock*, *euclidean* y *nan_ euclidean* muestran un comportamiento muy similar en todas las configuraciones evaluadas (esta última solo se muestra en *MovieLens* por su redundancia con *euclidean*). La métrica *cosine* presenta una ligera penalización en los vecindarios más reducidos, sin embargo, a partir de 250 vecinos, converge hacia niveles de robustez equiparables a los del resto de métricas. Por el contrario, *haversine* mantiene prácticamente sus niveles de robustez con el aumento del vecindario en MovieLens, lo que la sitúa como la métrica menos estable en vecindarios extensos frente a las perturbaciones aleatorias del sistema con esos datos. En los otros dominios, su robustez es más comparable al del resto de métricas e, incluso, llega a ser la mejor con muchos vecinos.

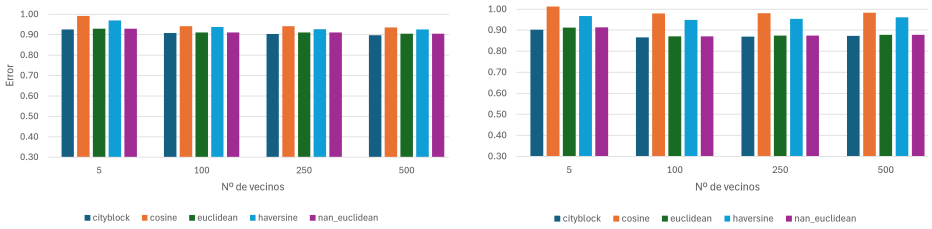


Fig. 2. Error medido por MAE en MovieLens para el experimento de aleatorización del sistema (izquierda) o de datos (derecha).

En línea con los resultados obtenidos para la robustez, el análisis de las métricas de error (Figura 2, solo MovieLens por cuestiones de espacio) muestra un comportamiento coherente con las conclusiones previamente expuestas. En general, a medida que aumenta el número de vecinos, los errores tienden a disminuir ligeramente y a estabilizarse. De forma consistente, las métricas *cityblock*, *euclidean* y *nan_ euclidean* mantienen valores muy próximos entre sí en todos los tramos, confirmando su comportamiento equivalente. La métrica *cosine* registra un error algo mayor en

vecindarios reducidos, lo que coincide con la penalización observada anteriormente en términos de robustez; sin embargo, aunque los niveles de error disminuyan, no llegan a converger a los valores observados para el resto de métricas. Por último, *haversine* mantiene errores elevados, aunque para todos los vecindarios la métrica *cosine* presenta peores valores de error. Esto se puede deber a que *cosine* es más apropiada para métricas de ranking [7], mientras que la semántica de la entrada esperada por haversine (latitudes y longitudes) no es coherente con los datos de valoraciones.

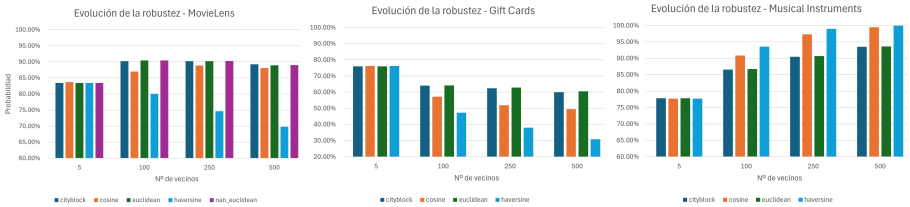


Fig. 3. Robustez frente a la aleatorización de los datos en MovieLens (izquierda), Gift Cards (medio) y Musical Instruments (derecha).

Robustez frente a la aleatorización de los datos En este caso, los resultados obtenidos (Figura 3) muestran un incremento progresivo de la robustez conforme se incrementa el número de vecinos (RQ1) excepto en GiftCards, alcanzando un comportamiento óptimo alrededor de los 250 vecinos para MovieLens y MusicalInstruments. En MovieLens, sin embargo, a partir de los 500 vecinos, se observa una ligera reducción de la robustez para todas las métricas excepto *cosine*. En este caso, si bien no se produce una disminución significativa, se alcanza un comportamiento estable y prácticamente convergente, lo que sugiere que la similitud coseno es menos sensible al aumento del tamaño del vecindario en rangos elevados, como se puede observar en los tres dominios. Excepcionalmente, tanto en MovieLens como en GiftCards, la métrica *haversine* evidencia un comportamiento notablemente decreciente de la robustez en relación con el aumento del vecindario, posiblemente ligado a los niveles de dispersión de los datos (RQ3).

La Figura 2 muestra que MAE reproduce el patrón observado en la robustez para las métricas *cityblock*, *euclidean* y *nan_ euclidean* en MovieLens: el error disminuye conforme aumenta el número de vecinos hasta aproximadamente 250, momento a partir del cual el comportamiento se estabiliza con un empeoramiento prácticamente imperceptible. Estas métricas mantienen perfiles muy similares y coherentes. La métrica *cosine* presenta los valores de error más altos, sin embargo, no es la métrica de peor robustez. En cuanto al descenso continuo de sus valores, está en línea con el patrón esperado de los resultados de robustez. Por su parte, *haversine* muestra errores elevados en todos los tramos, en concordancia con su menor nivel de robustez. No obstante, en este caso el comportamiento del error difiere respecto al observado en probabilidad: para los valores de MAE no se aprecia un deterioro progresivo, sino una mejora inicial seguida de una fase de estabilidad.

5 Conclusiones y trabajo futuro

En este trabajo se ha abordado el concepto de robustez aplicado a los sistemas de recomendación, centrándose especialmente en los algoritmos k-NN. Del repaso bibliográfico se concluye que la robustez es un concepto multidimensional que puede abordarse desde distintas perspectivas, incluyendo la estabilidad de las predicciones ante cambios en los datos, la sensibilidad frente a la configuración del sistema y la capacidad de mantener un desempeño consistente en distintos escenarios. Los experimentos de este trabajo muestran que la aleatorización del sistema conduce a comportamientos más robustos que la aleatorización de los datos en las distintas pruebas realizadas. Este patrón se mantiene estable al analizar diferentes métricas y tamaños de vecindario, con lo que se concluye que las perturbaciones introducidas a nivel del sistema permiten preservar mejor la estabilidad de las recomendaciones. Asimismo, se observa que la relación de robustez entre las métricas de distancia analizadas es notablemente consistente: aunque no en todos los casos las métricas más y menos robustas coinciden exactamente, las variaciones detectadas no son abruptas y, cuando se producen, se explican por la reducida diferencia del nivel de robustez existente entre ellas.

De hecho, los resultados de este trabajo muestran que el error no se comporta de manera análoga a la robustez, lo que confirma que ambos conceptos capturan dimensiones distintas del rendimiento del sistema y deben ser analizados de forma independiente. En este sentido, la robustez emerge como una propiedad estructural del sistema, más que como un subproducto directo del rendimiento medido mediante el error, lo que justifica su análisis específico. No obstante, resultaría de interés ampliar el análisis a distintos tipos de algoritmos k-NN, además de los enfoques clásicos basados en usuarios o ítems, podrían considerarse variantes más avanzadas como enfoques híbridos. La inclusión de este tipo de arquitecturas permitiría estudiar si las nociones de robustez propuestas se mantienen en configuraciones más complejas. Además, podrían explorarse nuevas métricas de distancia, así como distintas estrategias de aleatorización que permitan capturar otros tipos de perturbaciones realistas. Otra línea prometedora consiste en estudiar la relación entre robustez y otros factores relevantes en sistemas de recomendación, como la equidad, la diversidad o la explicabilidad, entre otros [8, 6].

Acknowledgments. This work was supported by Grants PID2022-139131NB-I00 and PID2025-167968NB-I00 funded by MCIN/AEI/10.13039/501100011033 and by "ERDF, a way of making Europe."

References

1. Adomavicius, G., Zhang, J.: Stability of recommendation algorithms. ACM Trans. Inf. Syst. **30**(4), 23:1–23:31 (2012)
2. Adomavicius, G., Zhang, J.: Improving stability of recommender systems: A meta-algorithmic approach. IEEE Trans. Knowl. Data Eng. **27**(6), 1573–1587 (2015)
3. Bellet, A., Habrard, A.: Robustness and generalization for metric learning. Neurocomputing **151**, 259–267 (2015)
4. Burke, R., O'Mahony, M.P., Hurley, N.J.: Robust collaborative recommendation. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 961–995. Springer (2015)

5. Casiraghi, G., Schweitzer, F.: Improving the robustness of online social networks: A simulation approach of network interventions. Frontiers Robotics AI **7**, 57 (2020)
6. Castells, P., Hurley, N., Vargas, S.: Novelty and diversity in recommender systems. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 603–646. Springer US (2022)
7. Cremonesi, P., Koren, Y., Turrin, R.: Performance of recommender algorithms on top-n recommendation tasks. In: Amatriain, X., Torrens, M., Resnick, P., Zanker, M. (eds.) Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30, 2010. pp. 39–46. ACM (2010)
8. Deldjoo, Y., Jannach, D., Bellogín, A., Difonzo, A., Zanzonelli, D.: Fairness in recommender systems: research landscape and future directions. User Model. User Adapt. Interact. **34**(1), 59–108 (2024)
9. Gunawardana, A., Shani, G., Yogev, S.: Evaluating recommender systems. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 547–601. Springer US (2022)
10. Huai, M., Zheng, T., Miao, C., Yao, L., Zhang, A.: On the robustness of metric learning: An adversarial perspective. ACM Trans. Knowl. Discov. Data **16**(5), 95:1–95:25 (2022)
11. Impagliazzo, R., Lei, R., Pitassi, T., Sorrell, J.: Reproducibility in learning. CoRR **abs/2201.08430** (2022)
12. Kaggle Dataset: Prajit Datta: Movielens 100k dataset (2016), https://www.kaggle.com/datasets/prajitdatta/movielens-100k-dataset
13. McAuley, J.: Amazon product data (2014), https://cseweb.ucsd.edu/ jmcauley/datasets/amazon/links.html
14. McNee, S.M., Riedl, J., Konstan, J.A.: Being accurate is not enough: how accuracy metrics have hurt recommender systems. In: Olson, G.M., Jeffries, R. (eds.) Extended Abstracts Proceedings of the 2006 Conference on Human Factors in Computing Systems, CHI 2006, Montréal, Québec, Canada, April 22-27, 2006. pp. 1097–1101. ACM (2006)
15. Ni, J.: Amazon review data (2018) (2018), https://cseweb.ucsd.edu/ jmcauley/datasets/amazon_v2/
16. Nikolakopoulos, A.N., Ning, X., Desrosiers, C., Karypis, G.: Trust your neighbors: A comprehensive survey of neighborhood-based methods for recommender systems. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 39–89. Springer US (2022)
17. Oh, S., Ustun, B., McAuley, J.J., Kumar, S.: Rank list sensitivity of recommender systems to interaction perturbations. In: Hasan, M.A., Xiong, L. (eds.) Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022. pp. 1584–1594. ACM (2022)
18. Ovaisi, Z., Heinecke, S., Li, J., Zhang, Y., Zheleva, E., Xiong, C.: Rgrecsys: A toolkit for robustness evaluation of recommender systems. In: Candan, K.S., Liu, H., Akoglu, L., Dong, X.L., Tang, J. (eds.) WSDM '22: The Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event / Tempe, AZ, USA, February 21 - 25, 2022. pp. 1597–1600. ACM (2022)
19. Pinto, F., Torr, P.H.S., Dokania, P.K.: An impartial take to the CNN vs transformer robustness contest. CoRR **abs/2207.11347** (2022)
20. Scikit-Learn: distance_metrics (2007), https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.distance_metrics.html
21. Zhang, K., Cao, Q., Sun, F., Wu, Y., Tao, S., Shen, H., Cheng, X.: Robust recommender system: A survey and future directions. CoRR **abs/2309.02057** (2023)