

Fecha de publicación Mayo 8, 2020

Digital Object Identifier No disponible (No DOI Registration agency)

Sistema de Recomendación basado en Content-Based Filtering con Item-Based Analysis

C. MORALES-LARA¹

¹Universidad Rafael Landívar. Departamento de Informática y de Sistemas, Guatemala, Guatemala

Autor correspondiente: C. Morales Lara (correo electrónico: camoralesl@correo.url.edu.gt).

ABSTRACT Los sistemas de recomendación son herramientas de filtrado de información utilizada para aliviar el exceso de información para los usuarios. Los motores de recomendación son probablemente uno de los enfoques de ciencia de datos más aplicados en la actualidad. Existen dos técnicas principales para construir un sistema de recomendación: filtrado basado en contenido (Content-Based Filtering) y filtrado colaborativo (Collaborative Filtering). El Content-Based Filtering utiliza las propiedades de los elementos para buscar elementos con propiedades similares. Los algoritmos de Collaborative Filtering toman las calificaciones de los usuarios u otro comportamiento de los usuarios y hacen recomendaciones basadas en los comportamientos similares de los usuarios. Aquí proponemos un modelo de Content-Based Filtering simple en el que sigue su concepto base, el de aumentar la similaridad de dos objetos mediante búsquedas de propiedades similares. El programa tiene un arranque en frío el cual recomienda solo en base a los rankings de las películas de IMDB, y se trata de alimentar el motor de recomendación con las recomendaciones de los usuarios pasados tratando de aplicar el concepto base de Collaborative-Filtering. Los experimentos realizados presentan buenos resultados buscando el mejor rendimiento posible mejorando la precisión de la predicción en el data set de películas de IMDB.

INDEX TERMS Recommender System, Content-Based Filtering, Collaborative Filtering, Naive Bayes Classifier, Machine Learning, Java, Recommendation Engine.

I. INTRODUCTION

En el presente artículo de investigación se explicarán sobre los sistemas de recomendación (RS), su funcionalidad, el objetivo que cumplen y los distintos tipos de soluciones que nos brindan estos sistemas para la mejora de filtración de información para recomendar con más precisión según las acciones del usuario. Los sistemas de recomendación son una alternativa funcional para enfrentar los problemas de sobrecarga de información. Actualmente se maneja con grandes cantidades de data por lo que al abarcar mas data esto conlleva nuevos tipos de problemas que en los tiempos de antes no se presentaban. Uno de estos problemas son los de sobrecarga de información, sin embargo, gracias a la rama de las ciencias computacionales como Machine Learning se soluciona con los sistemas de recomendación. Se desarrollo una propuesta de solución a un problema de investigación el cual fue la elaboración de un motor de recomendación simple que tuviera su arranque en frío y su mejora con las interacciones del usuario, el software se implementó en un

lenguaje compatible con la máquina virtual de Java (Java). La interacción con la aplicación es por medio de GUI Desktop. En este artículo abordamos este problema creando un modelo probabilístico que los usuarios finales pueden interpretar. Nuestro modelo se basa en el enfoque BCF que combina enfoques basados en elementos. Por otro lado el arranque en frío es un CF basado en elementos, con la desventaja que para que sea preciso se necesita una gran cantidad de recomendaciones de los usuarios.

A. BASIC CONCEPTS

Los motores de recomendación tienen como objetivo mostrar los artículos de interés del usuario. Lo que los hace diferentes de los motores de búsqueda es que el contenido relevante generalmente aparece en un sitio web sin solicitarlo y los usuarios no tienen que generar consultas, ya que los motores de recomendación observan las acciones del usuario y crean consultas para los usuarios sin su conocimiento. [1] Los sistemas de recomendación tienen un nivel de eficiencia alto

ya que pueden asociar elementos de nuestros perfiles de consumo como el historial de compras, selección de contenidos e inclusive nuestras horas de actividad, cualquier acción de parte del usuario estos motores lo toman como dato importante para realizar las recomendaciones. El funcionamiento de los sistemas de recomendación ha evolucionado por el Machine Learning. Anteriormente los motores de búsqueda, plataformas de contenido y ventas de producto funcionaban con rankings o listas de popularidad de un objeto. Estos sistemas son funcionales sin embargo, no podían ser personalizados por la experiencia del usuario y mostraban elementos que no se correspondían a nuestros intereses. Los sistemas de recomendación filtra los datos de acuerdo con un análisis de las preferencias pasadas de los usuarios. Para este proceso, se pueden utilizar algunas técnicas, las más conocidas son el filtrado basado en contenido o Content-Based Filtering (CBF), el filtrado colaborativo o Collaborative Filtering (CF) y el filtrado híbrido (Hybrid Filtering) [2] . . . Los motores de recomendaciones requieren distintos tipos de entrada para realizar recomendaciones: [3]

- *Información del elemento descrita con atributos.*
- *Perfil de usuario como rango de edad, sexo, ubicación geográfica.*
- *Interacciones del usuario en forma de calificaciones, navegación, etiquetado, comparación, guardado, y correo electrónico.*

Contexto en el que se mostrarán los elementos; La categoría del elemento y la ubicación geográfica del elemento son ejemplos de esté. El motor de recomendación combina estas entradas para responder preguntas como:

- 1) Usuarios que compraron, vieron, marcaron u otra acción a este objeto o artículo.
- 2) Aspectos similares a este objeto u artículo.
- 3) Otros usuarios similares a uno.
- 4) Otros usuarios que tal vez uno conozca (Facebook friend recommender).

B. USER BASED AND ITEM BASED ANALYSIS

La creación de un motor de recomendación depende de si el motor busca artículos relacionados o usuarios cuando intenta recomendar un artículo en particular. [4]

- En el análisis basado en artículos (Item Based Analysis), el motor se enfoca en identificar artículos que son similares a un artículo en particular.¹
- En el análisis basado en el usuario (User Based Analysis), primero se determinan usuarios similares al usuario particular. Se recomiendan los mismos elementos a otros usuarios similares, para calcular una matriz de

¹Este análisis fue el que se utilizó para la creación del motor de recomendación, conclusión: Se utilizó la técnica de Content-Based Filtering con un análisis de Item-Based Analysis.

similitud, dependiendo de si estamos analizando los atributos del elemento o las acciones del usuario.

Existen tres enfoques fundamentales para calcular la similitud:

- 1) (Collaborative Filtering) Los algoritmos de filtrado colaborativo toman las calificaciones de los usuarios u otro comportamiento del usuario y hacen recomendaciones basadas en lo que les gustó o compró a los usuarios con un comportamiento similar.
- 2) (Content-Based Filtering) El algoritmo basado en contenido utiliza las propiedades de elementos para encontrar elementos con propiedades similares.
- 3) (Hybrid Filtering) Un enfoque híbrido que combina filtrado colaborativo y basado en contenido.

C. NORMALIZACIÓN

Para que funcionen mejor muchos algoritmos de Machine Learning usados en Data Science, hay que normalizar las variables de entrada al algoritmo. Normalizar significa comprimir o extender los valores de la variable para que estén en un rango definido.

$$X_{\text{normalized}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \quad (1)$$

El problema con este tipo de normalización, es que comprime los datos de entrada entre unos límites empíricos (el máximo y el mínimo de la variable). Esto quiere decir que si se presenta un ruido entre los datos entonces éste va a ser ampliado también.

Escalado estándar (Standard Scaler) Una alternativa al escalado de variables es usar otra técnica conocida como escalado estándar (a cada dato se le resta la media de la variable y se le divide por la desviación típica).

$$X_{\text{normalized}} = \frac{X - X_{\text{mean}}}{X_{\text{stddev}}} \quad (2)$$

II. BASED-CONTEND FILTERING

El filtrado basado en contenido (CBF), se basa en una descripción de elementos y un perfil de las preferencias del usuario combinadas de cierta manera. Primero, los ítems se describen con atributos, y para encontrar ítems similares, medimos la distancia entre los ítems utilizando dos distintos algoritmos, una se llama "Distancia de Cosenos" o "Similaridad Coseno" y la otra se llama "Coeficiente de Pearson". Dada la información sobre el tipo de elementos que le gustan al usuario, podemos introducir pesos que especifiquen la importancia de un atributo de elemento específico. Este enfoque inicialmente necesita muy poca información sobre la retroalimentación del usuario, por lo que efectivamente evita el problema del arranque en frío. CBF puede diseñarse para recomendar elementos similares a los que a un usuario predeterminado le gustaban en el pasado [5].

A. SIMILARIDAD COSENO

En minería de textos se aplica la similitud coseno con el objeto de establecer una métrica de semejanza entre textos. En

minería de datos se suele emplear como un indicador de cohesión de clústeres de textos. La similitud coseno no debe ser considerada como una métrica debido a que no cumple la desigualdad triangular. La similitud coseno es una medida de la similitud existente entre dos vectores en un espacio que posee un producto interior con el que se evalúa el valor del coseno del ángulo comprendido entre ellos. Esta función trigonométrica proporciona un valor igual a 1 si el ángulo comprendido es cero, es decir si ambos vectores apuntan a un mismo lugar. [8] Cualquier ángulo existente entre los vectores, el coseno arrojaría un valor inferior a uno. Si los vectores fuesen ortogonales el coseno se anularía, y si apuntasen en sentido contrario su valor sería -1. De esta forma, el valor de esta métrica se encuentra entre -1 y 1, es decir en el intervalo cerrado [-1,1]. Esta distancia se emplea frecuentemente en la búsqueda y recuperación de información representando las palabras (o documento) en un espacio vectorial.

$$\mathbf{A} \cdot \mathbf{B} = \|\mathbf{A}\| \|\mathbf{B}\| \cos \theta$$

Dados dos vectores de atributos, A y B, la similitud del coseno, se representa usando un producto de punto y magnitud como:

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}},$$

B. COEFICIENTE DE PEARSON

En estadística, el coeficiente de Pearson es una medida de dependencia lineal entre dos variables aleatorias cuantitativas. A diferencia de la covarianza, la correlación de Pearson es independiente de la escala de medida de las variables. Podemos definir el coeficiente de Pearson como un índice que puede utilizarse para medir el grado de relación de dos variables siempre y cuando ambas sean cuantitativas y continuas.

$$\rho_{X,Y} = \frac{\sigma_{XY}}{\sigma_X \sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y},$$

III. COLLABORATIVE FILTERING

El filtrado colaborativo (CF) se basa únicamente en las calificaciones de los usuarios u otro comportamiento/acción del usuario. Una ventaja clave del filtrado colaborativo es que no depende del contenido del elemento y, por lo tanto, es capaz de recomendar elementos complejos como películas, como en el caso del proyecto, sin comprender el elemento en sí. La suposición subyacente es que las personas que acordaron en el pasado estarán de acuerdo en el futuro, y que les gustarán tipos de artículos similares a los que les gustaban en el pasado. [6] El filtrado colaborativo es la implementación más popular de un sistema de recomendación. Estos sistemas de recomendación se basan en una matriz de calificación en

la que cada usuario proporciona información sobre cuánto le gustan o no le gustan algunos artículos u objetos. Los métodos del filtrado colaborativo actúan directamente sobre la matriz de calificación para calcular las predicciones y generar recomendaciones. Este enfoque se divide en dos tipos: CF basada en el usuario y CF basada en elementos.

- 1) El CF basado en el usuario analiza un grupo de usuarios que comparten experiencias/intereses similares con el usuario objetivo y recomienda los elementos que el grupo generalmente prefiere.
- 2) El CF basado en ítems recomienda ítems que tienen una mayor similitud con la lista de ítems que le gustaba a un usuario en el pasado [7]. Una desventaja seria de este enfoque es el llamado "Arranque en frío", que significa que si queremos construir un sistema de filtrado colaborativo preciso, el algoritmo a menudo necesita una gran cantidad de recomendaciones de los usuarios.

IV. HYBRID FILTERING

El filtrado híbrido combina varios algoritmos de filtrado para obtener un conjunto de artículos o productos que se ajustan a las preferencias de un usuario. El filtrado colaborativo puede aprender las preferencias del usuario de las acciones del usuario con respecto a una fuente de contenido y usarlas en otros tipos de contenido. El filtrado basado en contenido se limita a recomendar contenido del mismo tipo que el usuario ya está utilizando. Esto es útil si se pueden recomendar diferentes fuentes como libros y películas basadas en la exploración de noticias. Filtrado colaborativo y filtrado basado en contenido no son mutuamente excluyentes; se pueden combinar para ser más efectivos en algunos casos.

Ejemplo: Los patrones de búsqueda de Netflix utiliza filtrado Colaborativo (CF) mientras que para analizar patrones de búsqueda y visualización de usuarios similares utiliza el filtrado basado en contenido (CBF) para ofrecer películas que comparten características con películas que el usuario ha calificado altamente. Es por eso que la unión de estas se forma el filtrado híbrido (HF).

V. NAIVE BAYES CLASSIFIER

Naive Bayes se ha estudiado ampliamente en la comunidad de recuperación de texto y sigue siendo un método popular para la categorización de texto, el problema de juzgar documentos como pertenecientes a una categoría u otra (como spam o legítimo) con frecuencias de palabras como características. Con el preprocesamiento adecuado, es competitivo en este dominio con métodos más avanzados que incluyen máquinas de vectores de soporte. Los clasificadores Naive Bayes son altamente escalables y requieren una serie de parámetros lineales en el número de variables (características / predictores) en un problema de aprendizaje. En resumen, Bayes ingenuo es un modelo de probabilidad condicional: dada una instancia de problema para ser clasificada, representada por un vector $x = (x_1, \dots, x_n)$ representa algunas características n (variables independientes), asigna a esta

instancia las probabilidades

$$p(C_k | x_1, \dots, x_n)$$

para cada uno de los K posibles resultados o clases Ck. El problema con la formula anterior es que si el número de características n es grande o si una característica puede tomar una gran cantidad de valores. Por lo tanto, reformando el modelo para hacerlo más manejable. Usando el teorema de Bayes, la probabilidad condicional se puede descomponer como

$$p(C_k | \mathbf{x}) = \frac{p(C_k) p(\mathbf{x} | C_k)}{p(\mathbf{x})}$$

VI. CLASES AND METHODS DEFINITION

- JFrame.java (Main) Esta clase es el Main del proyecto y equivale al menú principal del motor de recomendación, contiene un JFrame en el cual tiene las dos opciones que tiene el motor de recomendación mas la carga del data set de películas de IMDB. Esta clase el método de lectura de archivo el cual se aplica para la lectura del data set ya sea el data set incluido en el proyecto o la lectura de otro data set propia del usuario. Además cuenta con un método llamado Normalizar el cual requiere de una lista de valores enteros que aplica normalización, cuenta con dos métodos extras, NumeroMinimo y NumeroMaximo ambos son utilizados por el método Normalizar para obtener el valor máximo y mínimo de la lista previamente utilizada y así calcular la respectiva normalización de esa lista.
- ArranqueFrio.java Esta clase es un JFrame, es donde se desarrolla el arranque en frio del motor de recomendación, tiene dos métodos uno llamado RecomendacionDefault que es cuando el motor de recomendación recomienda sin ninguna recomendación de usuarios previos, sin que el motor este alimentado por ninguna información por lo que recomienda solamente por las mejores películas por sus rankings de IMDB. El otro método se llama RecomendacionAlimentada el cual antes de recomendar una película primero lee todas las interacciones de los usuarios pasados en un archivo que se crea en el proyecto, siguiendo el principio del Collaborative Filtering basado en ítems, el cual consiste en recomendar ítems que tienen una mayor similitud con la lista de ítems que le gustaba a un usuario en el pasado. Esto tiene su desventaja ya que si no hay muchas interacciones de usuarios no será muy preciso o si hay muchas interacciones de usuarios pasados será demasiado precisos que puede que no encuentre ítems que den similitud.
- RecomendacionUsuario.java Esta es la clase mas importante del proyecto, pues conlleva la recomendación por interacción del usuario que luego esto servirá para alimentar el arranque en frio por CF basado en ítems. Esta recomendación se basa en el concepto básico de Based-Content Filtering basado en el enfoque de Item-Based analysis.² Esta clase es un JFrame y algunos de sus métodos son Similitud, GetListaRecomendacion, setPesoPorProbabilidad. El método Similitud es el método donde busca similitud entre los ítems por el usuario con los ítems del data set, el peso de cada ítem es dado previamente, se buscó determinar los pesos por la probabilidad de los mismos ítems, sin embargo, esto dio falsos y no tan precisos resultados comparado con definir pesos anteriormente. SetPesoPorProbabilidad es el método donde se pensaron definir los pesos por su probabilidad. GetListaRecomendacion es un método donde que ya con una lista de similitudes se busca los que tenga la similaridad mas alta y retorna en una nueva lista las películas que tengan más ítems en similitud. Finalmente, esta clase tiene un método llamado CrearArchivo en el cual crea un archivo que contendrá las interacciones de los usuarios para alimentar el motor de recomendación de CF para el arranque en frío.
- DataSetMovies.java Esta clase consiste en la manipulación del data set de películas de IMDB, se creo una lista por cada columna que contiene el data set, luego con el método LeerArchivo se lee el movie metadata.csv y adentro de este método se llama el método LlenarListas lo que llena todas las listas previamente definidas.
- Pelicula.java Esta es una clase en el cual se definen todos los atributos que tiene nuestro objetivo a analizar, que en este caso es una Película. además, se realizan sus respectivos Getters y Setters. Su constructor tiene como parámetro movie title ya que el nombre de la película se analizo como su posible ID único.
- ClasificadorBayes.java Esta es una clase donde se desarrolló el algoritmo clásico conocido como Naive Bayes Classifier.³ Esta clase dispone de varios métodos como SepararPalabras en cual separa cuales son “Positivas” y cuales son “Negativas” y tener un contador de cada uno para futuro análisis. El método Clasificar es el actual método donde se desarrolla el clasificador calculando todas las probabilidades necesarias para determinar si un futuro texto es positivo o negativo. El fin de esta clase fue recomendar películas por medio de las palabras clave que tuviera el data set en su columna llamada “plot keywords” y que el usuario escribiera sus propias palabras clave para que el clasificador hiciera su trabajo, sin embargo, esto dio resultados negativos e incluso recomendaciones muy imprecisas por lo que se dejo de utilizar.

²Estos conceptos ya se definieron previamente en el artículo .

³Se desarrollo SIN librerías externas.

VII. EXPERIMENTAL RESULTS

Los resultados dados por el modelo de un motor de recomendación basado en Content-Based Filtering con enfoque basado en ítems son positivos y eficientes. Son recomendaciones precisas. Las recomendaciones por Collaborative Filtering también basado en ítems sus resultados son directamente proporcional a la cantidad de información de recomendaciones previas de los usuarios. Entre más información de recomendación mas precisa es la recomendación pero entre menos información es menos precisa (problema del arranque en frío).

Los resultados con Content-Based Filtering pero alterando los pesos de los ítems respecto con su probabilidad dieron resultados negativos, dando recomendaciones imprecisas y con un margen de aleatoriedad.

Los resultados con el algoritmo de Naive Bayes Classifier fueron negativos también, dando resultados nada precisos, esto puede ser a que el banco de diccionario base (palabras claves del data set) no sea suficiente de información para brindar recomendaciones precisas.

VIII. CONCLUSION

En este artículo presentamos el diseño de un modelo de un motor de recomendación que proporciona recomendaciones eficientes y precisas. Se ha combinado enfoques de Filtrado de Colaboración y de Contenido basados en el espacio de elementos dentro de un solo modelo. Los resultados de los experimentos indican que el enfoque propuesto obtiene mejores resultados de precisión, especialmente cuando el número de recomendaciones es más alto que sin información como el arranque en frío. El enfoque propuesto ofrece mejores resultados que al haber utilizado Naive Bayes Classifier con palabras clave. Se presenta mejoras significativas en la precisión de la predicción de las dos distintas maneras de conjuntos de datos probados. En consecuencia, los resultados obtenidos con CBF con Item-Based Analysis se consideran aceptables porque el modelo permite estimar la recomendación que un elemento podría tener. El método propuesto se amplió por la integración de información de atributos de usuarios, lo que genero modificar los cálculos de probabilidad para recomendar una película.

REFERENCES

- [1] Kaluza, Bostjan, *Machiner Learning in Java*. Birmingham, UK : Packt Publishing. 2016, chapter 6, pp.101.
- [2] W.-K. Chen, *Linear Networks and Systems*. Belmont, CA, USA: Wadsworth, 1993, pp. 123–135.
- [3] Kaluza, Bostjan, *Machiner Learning in Java*. Birmingham, UK : Packt Publishing. 2016, chapter 6, pp.102.
- [4] Kaluza, Bostjan, *Machiner Learning in Java*. Birmingham, UK : Packt Publishing. 2016, chapter 6, pp.103.
- [5] J. U. Duncombe, “Infrared navigation—Part I: An assessment of feasibility,” *IEEE Trans. Electron Devices*, vol. ED-11, no. 1, pp. 34–39, Jan. 1959, 10.1109/TED.2016.2628402.
- [6] Kaluza, Bostjan, *Machiner Learning in Java*. Birmingham, UK : Packt Publishing. 2016, chapter 6, pp.104.
- [7] E. E. Reber, R. L. Michell, and C. J. Carter, “Oxygen absorption in the earth’s atmosphere,” Aerospace Corp., Los Angeles, CA, USA, Tech. Rep. TR-0200 (4230-46)-3, Nov. 1988.

- [8] *Algoritmo de similitud de coseno*, Graph Everywhere. [Online]. Available: <https://www.grapheverywhere.com/algoritmo-de-similitud-de-coseno/>



C. Morales-Lara Carlos Andrés Morales

Lara. Este autor actualmente es estudiante de la carrera de licenciatura en ingeniería en informática y de sistemas, cursando su cuarto año en el curso de Inteligencia Artificial. Nacido en la ciudad de Guatemala, el 20 de noviembre de 1997, actualmente con una edad de 22 años. Su educación superior esta actualmente en desarrollo en la universidad Rafael Landívar campus central, ciudad de Guatemala.

•••