Proyecto Final
Recomendador de Películas

Pedro Flores, Miguel Figueira, David Padrino

Escuela de Computacin
Facultad de Ciencias
Universidad Central de Venezuela (UCV)
Av. Paseo Los Ilustres, Los Chaguaramos, Caracas, Venezuela

Resumen

En este artículo se dan a conocer los conceptos básicos requeridos para empezar a comprender los sistemas de recomendación, específicamente bajo el uso de filtrado colaborativo.
La plataforma Kaggle permite utilizar datasets para la resolución de determinados problemas de minería de datos. El problema en cuestión es la realización de un sistema que sirva de recomendador de películas de acuerdo a ciertas votaciones realizadas por un grupo de usuarios a un grupo de películas de diferentes géneros, duraciones y clasificaciones.
Al trabajar con una cantidad tan grande de elementos hubo problemas al momento de realizar análisis porque el proceso se hacia un tanto tedioso y la lenta respuesta de los computadores hizo que las pruebas fueran la parte del proyecto que más tiempo demandara..
Los resultados de las predicciones fueron acertados hasta cierto punto, debido a que la cantidad de votaciones, a pesar de ser en gran número, no fue suficiente para realizar una predicción más acertada y precisa. Sin embargo se lograron predecir todos los valores para los usuarios en cuestión, más adelante se hablará al respecto con mayor detenimiento

1. Introducción

El grupo de desarrollo GroupLens busca desarrollar un recomendador de películas basado en filtrado colaborativo llamado *Movielens*, el cual se espera que funcione de acuerdo a 1,000,209 valoraciones o ratings de 6040 usuarios de MovieLens hacia 3900 películas. Este conjunto de datos es del año 2000 aproximadamente.
Para la realización de este proyecto fue desarrollada una aplicación en la herramienta *RShiny*, que permite interactuar con el algoritmo, hacer selecciones de datos y así de esta manera mostrar los resultados de forma estructurada y en una sola ventana. La mencionada herramienta fue utilizada para algunos gráficos que se muestran en el presente artículo.

2. Análisis Exploratorio de Datos

Se proveen tres archivos en base a los cuales se tomarán los datos (users.dat, ratings_for_kaggle_comp.csv y movies.dat) y un archivo en base al que se realizarán las lecturas (sample_submission.csv) en donde se puede observar cuál usuario valoró a cuál película.
Se realizó una lectura de los archivos y para poder utilizarlos dentro del procesamiento fue necesario cambiar su formato.
En la siguiente estructura se muestra la data cruda, luego de cambiado su formato quedo estructurado en columnas ”limpias”.

- users.dat: Esta en un rango entre 1 y 6040
  El formato de las columnas es el siguiente:

  UserID::Gender::Age::Occupation::Zip-code

  Toda la información demográfica fue dada voluntariamente por los usuarios, los cuales fueron los únicos que fueron agregados al conjunto.
  El género es denotaro por ”M”para hombres y ”F”para mujeres.
  Las edades fueron seleccionadas de acuerdo a los siguiente rangos:

  * 1: ”menor de 18”
  * 18: ”18-24”
  * 25: ”25-34”
  * 35: ”35-44”
  * 45: ”45-49”
  * 50: ”50-55”
  * 56: ”56+”

  La ocupación es elegida de acuerdo a los siguientes elementos:

  * 0: .otheror not specified
  * 1: .academic/educator”
  * 2: .artist”
  * 3: çlerical/admin”
  * 4: çollege/grad student”
  * 5: çustomer service”
  * 6: ”doctor/health care”
  * 7: .executive/managerial”
  * 8: ”farmer”
  * 9: ”homemaker”
  * 10: ”K-12 student”
  * 11: ”lawyer”
  * 12: ”programmer”
  * 13: retired”
  * 14: ”sales/marketing”
  * 15: ”scientist”
  * 16: ”self-employed”
  * 17: ”technician/engineer”
  * 18: ”tradesman/craftsman”
  * 19: ünemployed”
  * 20: ”writer”

- ratings_for_kaggle_comp.csv:
  Todos los elementos contenidos en este archivo siguen el siguiente formato:

  UserID::MovieID::Rating::ID

  - UserIDs: Rango entre 1 y 6040.
  - MovieIDs: Rango entre 1 y 3952.
  - Ratings: son hechos en base a 5 estrellas .
  - ID: Una mezcla de los UserIDs con MovieIDs separados por *underscore (_)*.

- movies.dat: La información dentro de este archivo está de la siguiente forma:

  MovieID::Title::Genres

  Los títulos son idénticos a los provistos por IMDB, incluyendo año de lanzamiento.
  Los géneros estan separados por *pipe* y se toman de la siguiente lista:

  * Action
  * Adventure
  * Animation
  * Children’s
  * Comedy
  * Crime
  * Documentary
  * Drama
  * Fantasy
  * Film-Noir
  * Horror
  * Musical
  * Mystery
  * Romance
  * Sci-Fi
  * Thriller
  * War
  * Western

- sample_submission.csv: Como se explicó anteriormente es el dataset central con el cual se va a realizar el trabajo, permite ver cuáles usuarios han realizado valoraciones a cuáles películas.
  Se muestra en el siguiente formato:

  UserID::Rating::ID

  ID: Representa un híbrido de la forma UserID_MovieID.
  El resto de los elementos son intuitivos de entender.

Además de los cambios de formato a los archivos, se realizó el borrado de aquellos usuarios que no hayan votado a ninguna película y aquellas películas que no fueron votadas, de manera que no afectara el resultado del procesamiento.

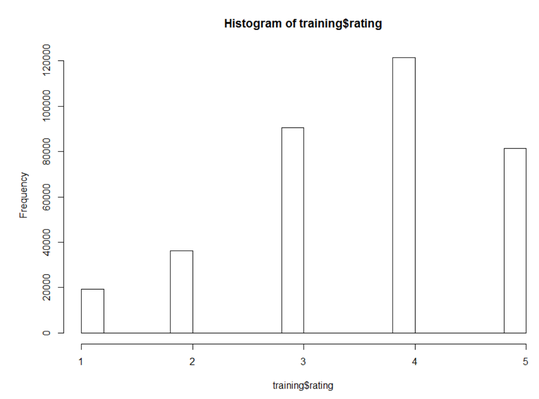

Figura 1: *distribución de las rating de las películas*

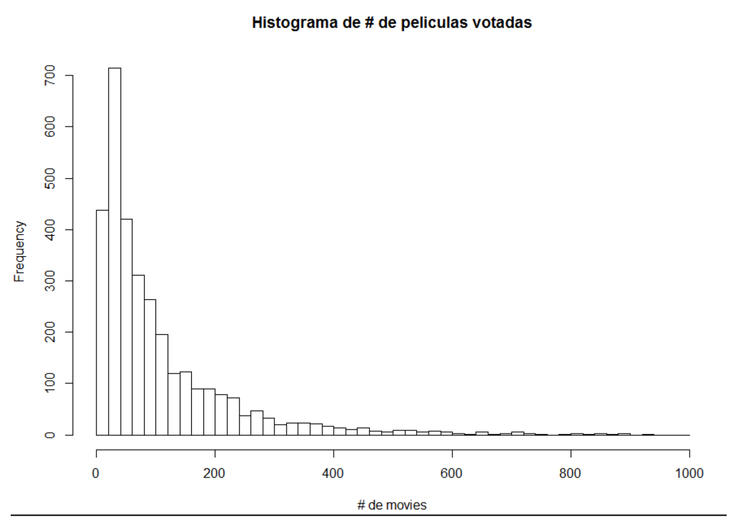

Figura 2: *distribución por número de películas votadas por usuario , la mayoría de los usuarios votó entre 20 y 40 películas*

3. Preprocesamiento—Crear Matriz Dispersa

3.1 Crear data de entrenamiento y prueba

Ya que la data que provee Kaggle solo ofrece data de entrenamiento, se procede a crear una data personalizada de entrenamiento y otra de prueba a partir de esa data inicial. Para ello se usa una técnica de muestreo estratificado tomando el 70 % de las votaciones realizadas por los usuario como entrenamiento y el 30 % restante como prueba, ya que si se toma el 70 % es posible que algún usuario no tenga ninguna película dentro de la data de entrenamiento.

3.2 Crear matriz dispersa

Viendo el formato de la data en la figura N, es necesario transformarla para tener una matriz dispersa donde los usuarios sean las instancias (o filas), las películas sean las columnas y los elementos de la matriz son los ratings o valoraciones de los usuarios. Para lograr esto se hace uso de la función *acast* de la biblioteca *reshape*.
Es bueno mencionar que la función *acast* retorna elementos de clase *vector*, *matrix*, o *array*

4. Filtrado Colaborativo

El algoritmo de filtrado colaborativo es ampliamente conocido por usarse en sistemas de recomendación, en nuestro caso usamos el algoritmo basado en usuario-usuario y se basa en la premisa de que si un usuario A se parece a un usuario B en cuánto a las películas que ha votado , entonces es posible que el usuario A tenga la misma valoración sobre otras películas que ha evaluado el usuario B. Además cabe destacar que para este usar este algoritmo solo usamos las valoraciones dejando de lado , la descripción de usuario como edad y ocupación , o información de las películas como género o año
Para lograr esto hicimos uso de una biblioteca en R llamada *Recommenderlab* usandola de la siguiente manera:
Luego de todo el preprocesamiento se procedió a convertir la matriz de votaciones en un tipo especial de matriz heredado de la clase Rating Matrix (específicamente *RealRatingMatrix*), con la cual permite hacer el proceso de recomendación—predicción de acuerdo a lo establecido en la biblioteca.

Una vez con la matrix de la clase *RatingMAtrix* se inicia el proceso de recomendación , en el cual se utilizará el comando *Recommender* basado en usuario (UBCF), con una

lista de parámetros adicionales como: Z-Score , que es el método de normalización, Jaccard como medida de similaridad (en algunas publicaciones es sugerida esta medida para estos casos de recomendación), nn representa el número de usuarios en base a los que se realizarán las recomendaciones y por último minRating, que representa el menor valor a partir del cual se realizarán las predicciones.

```
rec=Recommender(train.RatingMatrix[1:nrow(train.RatingMatrix)],
                method="UBCF",param=list(normalize = "Z-score",
                method="Jaccard",nn=5, minRating=1))
```

Figura 3: *genera rec que es el modelo necesario para calcular posteriormente las predicciones*

Para las predicciones se usará el comando *predict*, pasando como parámetros el modelo obtenido anteriormente, junto a la RatingMatrix creada anteriormente.

```
recom <- predict(rec, train.RatingMatrix[1:nrow(train.RatingMatrix)],
                type="ratings")
```

Figura 4: *código para generar el modelo para luego generar las recomendaciones con predict*

Luego de ordenar y limpiar los datos se pueden apreciar los siguientes resultados:
Predicciones de ratings que asignará un usuario sobre un determinado número de películas:

	Toy Story (1995)	Jumanji (1995)	Grumpier Old Men (1995)	Waiting to Exhale (1995)	Father of the Bride Part II (1995)	Heat (1995)	Sabrina (1995)	Tom and Huck (1995)	Sudden Death (1995)	GoldenEye (1995)	American President, The (1995)	Dracula: Dead and Loving It (1995)
1	3.94	3.94	3.94	3.94	3.94	3.94	3.94	3.94	3.94	3.94	3.94	3.94
2	3.20	3.20	3.20	3.20	3.20	3.20	3.20	3.20	3.20	3.20	3.20	3.20
3	3.30	3.18		3.13	3.13	3.18	3.19	3.18	3.18	3.18	3.17	3.18
4	4.03	4.03	4.03	4.03	4.03	4.11	4.03	4.03	4.03	4.03	4.03	4.03
5		3.97	3.97	3.97	3.97	3.97	3.97	3.97	3.97	3.97	3.97	3.97
6		4.48	4.48	4.48	4.48	4.48	4.48	4.35	4.48	4.48	4.48	4.48
7	3.88	3.74	3.69	3.69	3.69	3.77	3.69	3.69	3.69	3.61	3.69	3.69
8	4.26	4.26	4.26	4.26	4.26	4.26	4.26	4.26	4.26	4.26	4.26	4.26
9	4.09	4.09	4.09	4.09	4.09	4.09	4.09	4.09	4.09	4.09	4.09	4.09
10		3.06	3.06	3.06	3.06	3.08		3.06	3.06	3.05	3.06	3.06

Figura 5: *matriz resultado de las predicciones, los espacios en "blanco.es porque ya habían sido puntuados por el usuario*

	X	user	movie	rating	real
1	1	2783	589	3.83	5
2	2	2783	1127	4.05	4
3	3	2783	610	3.94	3
4	4	2783	1301	3.89	5
5	5	2783	1356	3.91	2
6	6	2783	3175	3.76	5
7	7	2783	1580	3.91	2
8	8	2783	1584	3.74	5
9	9	2783	2553	3.91	5
10	10	2783	1019	3.76	5

Figura 6: *Comparación del resultado de las puntuaciones predichas con las puntuaciones reales:*

	peli1	peli2	peli3	peli4	peli5	peli6	peli7	peli8	peli9	peli10
2783	Mad Max 2 (a.k.a. The Road Warrior) (1981)	Bottle Rocket (1996)	Life Is Beautiful (La Vita è bella) (1997)	Terminator, The (1984)	Mad Max (1979)	Independence Day (ID4) (1996)	Sneakers (1992)	Starship Troopers (1997)	Dark Crystal, The (1982)	Soylent Green (1973)

Figura 7: *Lista de 10 películas recomendadas a un usuario específico:*

5. Evaluación de Modelo

Matriz de Confusion

Nota: columna son los valores reales

```
##
##        1     2     3     4     5
##  1    35     0     1     0     0
##  2   821   452   363   208    91
##  3  5021  9170 18127 14358  5829
##  4  2310  6081 20821 37502 28109
##  5    19    26   132   556  1392
```

Figura 8: *Comparación del resultado de las puntuaciones predichas con las puntuaciones reales, las filas son el resultado del algoritmo*

5.1 Observaciones de la Matriz de Confusión

- Casi todos los que predijo como 1 eran realmente 1
- Los 4 y los 3 los predijo bastante bien, entendiendo que la diferencia entre un 4 y un 3 no es mucha.
- Los que eran realmente 1 los predijo casi todos como 2,3 o 4

6. Resultados — Conclusión

Los sistemas de recomendación son herramientas muy útiles al momento de mejorar la experiencia del usuario con una aplicación o sitio web debido a que brindan la sensación de que el sistema conociera los gustos del usuario e hiciera recomendaciones en base a ellos, evidentemente todo es una abstracción de lo que realmente es.
Con el presente proyecto y sus resultados se pudo observar que es necesario tener un cluster para poder generar recomendaciones en tiempo real debido a la gran cantidad de datos y procesamiento que se necesitan para poder realizar predicciones y recomendaciones acertadas en un tiempo prudencial.

Analizando la figura 5 se puede observar que los resultados no son los mejores, es decir, repite muchos valores de recomendación para un mismo usuario, lo cual posiblemente es una consecuencia de no contar con suficientes datos, además de que muchos usuarios tienen muy pocas recomendaciones como se puede observar en la figura 2 donde pocos usuarios tienen más de 60 películas votadas ,e incluso una cantidad significativa de personas tiene menos de 20 películas calificadas.
Además de que las valoraciones del rating '1' las predice bastante mal como se menciona en la sección Evaluación de Modelos, debido a que existe una diferencia significativa entre la cantidad de películas calificadas como 3,4,5 con películas calificadas con 1, como se puede apreciar en la figura 1

Referencias

[1] Michael Hahsler recommenderlab: A Framework for Developing and Testing Recommendation Algorithms. 2011.

[2] Linux Uncle. Using R package, recommenderlab, for predicting ratings for MovieLens data. 2014. https://ashokharnal.wordpress.com/2014/12/18/using-recommenderlab-for-predicting-ratings-for-movielens-data/

[3] inside-R.org. Similarity and Dissimilarity Functions, http://www.inside-r.org/packages/cran/sets/docs/similarity

[4] wikipedia.org, Jaccard index, https://en.wikipedia.org/wiki/Jaccard