PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

Story Recommender for the Fanfiction Community

Andrés Carvallo
Christian Meléndez

Índice

CONTEXTO DEL PROBLEMA

Desde los inicios de la web 2.0, han surgido comunidades en línea que han crecido rápidamente y se han convertido en una parte fundamental de la vida de los usuarios que forman parte de ella. En este estudio nos enfocaremos en la comunidad de `fanfiction.net`, en la cual los usuarios pueden leer y escribir historias que son adaptadas, recreadas y modificadas de obras originales de series de televisión, anime, películas, entre otros. Los miembros de fanfiction pueden seguir historias y a sus autores, creando así una red social.

Particularmente en este trabajo nos enfocaremos en recomendar historias favoritas a los usuarios de la comunidad de fanfiction ya que, actualmente, solo posee un menú con categorías de historias por las que el usuario puede navegar, pero no cuenta con una interfaz donde se entreguen sugerencias de lectura (personalizadas ni no-personalizadas). Además, dada la gran cantidad de usuarios e historias que están disponibles en la página, se complica la capacidad de encontrar historias que se adecúen al gusto del lector.

La dificultad para lograr estas recomendaciones está en que las historias no pueden recibir ratings, pero sí se reciben *reviews* y es posible que sean seguidas por usuarios registrados en fanfiction, por lo que existe una métrica similar al rating con la cual basar las sugerencias de historias a los usuarios.

PROBLEMA Y PROPUESTA DE SOLUCIÓN

La problemática a solucionar es recomendar historias a partir de información implícita existente en la comunidad de fanfiction, ya que no se cuenta con ratings. Se considerará, en primer lugar, si es que la historia es seguida por el usuario (*favorite story*) y se añadirán features que puedan mejorar la recomendación como: información de reviews que ha recibido la historia, métricas de sentimientos de los reviews, cantidad de reviews, cantidad de palabras del texto de la historia, entre otros.

Entonces surge la siguiente pregunta: cuál es el mejor recomendador en este caso? Para responderla, compararemos los resultados de tres algoritmos de recomendación que no necesiten información de ratings y escogeremos el que tenga mejor desempeño. También investigaremos cómo mejora la recomendación al incluir más features como input.

OBJETIVOS

El objetivo de este proyecto es encontrar el mejor algoritmo recomendador de acuerdo a las características del dataset. Además, queremos investigar si es que es posible mejorar la calidad de la recomendación de historias incorporando features como métricas de sentimiento de las reviews y del texto de las historias, o incluyendo datos como cantidad de palabras y cantidad de reviews.

DEFINICIÓN DE EXPERIMENTOS

Dataset

Los experimentos se harán sobre un dataset que consta de 15.195 registros de historias favoritas (seguidas por algún usuario) y otro dataset de 130.000 reviews de usuarios sobre estas historias.

Los datos de historias favoritas contienen el id del usuario, id de la historia favorita, nombre de la historia, cantidad de palabras, cantidad de reviews, resumen de la historia, autor, franquicia sobre la que fue escrita (i.e Starwars) y la fecha en que fue publicada. Cabe destacar que una historia no puede ser marcada como favorita por un usuario más de una vez.

Por otra parte, el dataset de reviews contiene el id de la historia que recibe el review, el texto del review y el usuario que hizo el review. Estos datos se utilizarán para obtener una métrica promedio de sentimiento de las reviews hechas sobre cada historia e incoporarla como un feature adicional del dataset principal.

Métricas de relevancia

Tenemos diferentes ideas para evaluar la relevancia de una historia para un usuario. La más simple y directa es considerar solo historias relevantes y no-relevantes dependiendo si el usuario ha marcado la hisoria como favorita. Sin embargo, existen otros factores que pueden dar atisbos de que una historia dentro de las favoritas (o fuera de estas) es más relevante para el usuario. Por ejemplo, si un usuario agrega una historia como favorita y, además, le entrega un comentario positivo, podría significar un mayor "rating" para esa historia que otra que marcó como favorita pero que no comentó. Este análisis de sentimientos de los reviews los añadimos a la base de datos ocupando la herramienta VADER [3] que tiene una implementación en `https://github.com/cjhutto/vaderSentiment`. Sin embargo, aún

está en etapa de evaluación el considerar los comentarios como parte de un "pesudo-rating" para las historias.

Comparación de algoritmos

Se compararán los resultados de los siguientes algoritmos:

1. **Implicit feedback Collaborative Filtering**: utilizaremos la metodología del paper de Hu, Koren & Volinsky [1] y consideraremos una variable binaria que toma un valor 1 si el usuario sigue a la historia (favorita) y 0 en otro caso. Así obtendremos un vector de usuarios y un vector de items que muestran las preferencias de estos. Ocuparemos la implementación del algoritmo del paper mencionado desde la librería *implicit* de Python que puede ser encontrada en `https://github.com/benfred/implicit`.

2. **Factor vector machines**: buscamos implementar este modelo, explicado en profundidad en [4], para poder agregar diversos atributos de los datos de las historias que podrían mejorar la recomendación. Específicamente, se incluirá en el modelo los usuarios con sus historias favoritas junto con franchise, cantidad de reviews, métricas de análisis de sentimiento de reviews promedio y texto del resumen de la historia. Para hacer experimentos con este sistema recomendador, ocuparemos la librería *libFMexe* implementada en R.

3. **Content-based collaborative filtering**: para realizar recomendaciones basadas en contenido, crearemos el perfil del usuario usando las historias que ha marcado como favoritas. Este perfil de usuario será creado mediante un clasificador naive Bayes, el cual ha sido ampliamente usado para recomendar ítems a partir de texto [2]. Específicamente, ocuparemos el clasificador multinomial naive Bayes de la librería *scikit-learn* de Python para crear los perfiles de usuarios, como también para evaluar las recomendaciones. Solo tenemos hasta el momento dos categorías: historia marcada como favorita (ítem relevante) o no (ítem no-relevante), por lo que los tiempos de ejecución de este algoritmo no deberían ser muy altos.

   Para el entrenamiento y test de este sistema, preprocesaremos los datos mediante Tokenization (para separar las palabras del texto), pasaremos las mayúsculas a minúsculas y también aplicaremos Stemming para obtener la raíz de la palabra. Se espera que así se puedan representar los resúmenes de las historias con solo las palabras importantes.

Los resúmenes de las historias serán representados como vectores (Vector Space Model), por lo tanto, al conteo de palabras se le aplicará la técnica de escalamiento *TF-IDF* para que así se tomen en consideración las palabras que aparecen en baja cantidad en un documento pero que podrían ser importante para describirlo.

Evaluación

Para evaluar los algoritmos, como no tenemos ratings explícitos, no ocuparemos métricas de predicción, sino que métricas de recomendación de varias historias: nDCG, P@n y MAP, las cuales sirven para medir la calidad de la lista de historias recomendadas. Hasta ahora, consideraremos como historias relevantes solo aquellas que el usuario ha marcado como favoritas y en base a esto se calcularán las métricas antes mencionadas.

Como se comentó en la sección "Métricas de relevancia", tenemos algunas ideas para mejorar la calidad de la relevancia que aún están en evaluación, como por ejemplo, incluir el análisis de sentimiento de algún comentario que haya hecho el usuario o el hecho de que un usuario siga a otro usuario por las historias que este último escribe.

REFERENCIAS

- [1] Yifan Hu, Yehuda Koren, Chris Volinsky (2008). Collaborative filtering for implicit feedback datasets. Data Mining, 2008. ICDM'08. Eighth IEEE International Conference (pp.263-272).ICDM

- [2] Lops, P., De Gemmis, M., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In Recommender systems handbook (pp. 73-105). Springer US.

- [3] Gilbert, E., & Hutto, C.J. (2014). VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. ICWSM.

- [4] Rendle, S. (2010, December). Factorization machines. In Data Mining (ICDM), 2010 IEEE 10th International Conference on (pp. 995-1000). IEEE.