UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

TFG del Grado en Ingeniería Informática

Sistema de Recomendación basado en Aprendizaje Profundo

Presentado por Raúl Negro Carpintero
en Universidad de Burgos — 29 de junio de 2019
Tutor: Bruno Baruque Zanón

UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

D. Bruno Baruque Zanón, profesor del departamento de nombre departamento, área de nombre área.

Expone:

Que el alumno D. Raúl Negro Carpintero, con DNI 71305764Z, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado título de TFG.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 29 de junio de 2019

Vº. Bº. del Tutor:	Vº. Bº. del co-tutor:
D. nombre tutor	D. nombre co-tutor

Resumen

El tema principal son los sistemas de recomendación. El objetivo de este proyecto es comparar el rendimiento entre los modelos clásicos y los modelos basados en aprendizaje profundo sobre diferentes conjuntos de datos.

Descriptores

Sistemas de recomendación, aprendizaje profundo, redes neuronales.

Abstract

The main topic are the recommendation systems. The goal of this project is to compare the performance between classic models and those based on deep learning on different datasets.

Keywords

Recommendation systems, deep learning, neural networks.

Índice general

Índice de figuras

Índice de tablas

Introducción

Descripción del contenido del trabajo y del estrucutra de la memoria y del resto de materiales entregados.

Objetivos del proyecto

2.1. Objetivos técnicos

- Comprender el funcionamiento de un sistema de recomendación clásico.
- Recoger y evaluar los resultados obtenidos por el modelo clásico.
- Comprender el funcionamiento de un sistema de recomendación basado en aprendizaje profundo.
- Recoger y evaluar los resultados obtenidos por el modelo basado en aprendizaje profundo.
- Comparar los resultados obtenidos por ambos modelos sobre una serie de conjuntos de datos.
- Desplegar el programa Python en un servidor y hacer posible la interacción con nuevos usuarios.

2.2. Objetivos personales

- Estudiar los sistemas de recomendación debido a su gran importancia.
- Poner en uso los conocimientos adquiridos durante la carrera.
- Ahondar más en el campo del aprendizaje profundo.

Conceptos teóricos

Los conceptos teóricos más importantes del proyecto son los relacionados con los sistemas de recomendación y las redes neuronales.

3.1. Sistemas de recomendación

Los sistemas de recomendación son herramientas que generan recomendaciones sobre un determinado objeto de estudio, a partir de las preferencias y opiniones dadas por los usuarios. [5]

En un sistema de recomendación hay dos clases de entidades: usuarios e ítems. Los datos están representados en una matriz de utilidad de tal manera que los usuarios son filas y los ítems columnas. Para cada par hay un valor que representa el grado de preferencia de un usuario para un ítem. Se asume que la mayoría de los valores se desconocen. Es por ello, que el objetivo de un sistema de recomendación es rellenar esos espacios en blanco.

Tipos de sistemas de recomendación

Existen tres tipos de sistemas (o modelos) de recomendación:

- Modelos basados en contenido
- Modelos colaborativos
- Modelos híbridos

Modelos basados en contenido

Este tipo de sistemas tienen en cuenta los gustos del usuario y las características de los ítems.

Para cada ítem hay que construir un perfil compuesto por las propiedades del mismo. Por ejemplo, si los ítem son películas, algunas de las propiedades serían género, director, fecha de estreno, reparto, etc. Una vez hecho esto, se recomiendan ítems cuyas propiedades sean parecidas a los ítems que el usuario ha valorado positivamente.

Modelos colaborativos

Recomiendan ítems basándose únicamente en los gustos de usuarios similares.

En este caso, en lugar de usar el vector ítem-perfil de un ítem, usamos las filas de la matriz de utilidad. Los usuarios son parecidos si sus filas se acercan de acuerdo a alguna distancia.

Modelos híbridos

Este tipo de sistemas utilizan las dos técnicas anteriores para ofrecer mejores recomendaciones.

Así pues, se recomendarán ítems que sean parecidos a los ítems valorados positivamente por usuarios parecidos.

3.2. Medidas de calidad

Podemos hacer uso de diferentes métricas para conocer cómo de bueno es nuestro sistema de recomendación. Dado que las dos librerías que se han utilizado en el proyecto las ha creado la misma persona, se utilizarán las métricas ofrecidas en ellas por su gran similitud. Las métricas son:

- Precisión k
- Recall k
- Score AUC
- Ranking recíproco
- *Root Mean Square Error*

Precisión *k*

La precisión k [8] mide el número de elementos relevantes conocidos que se encuentran en las primeras k posiciones del ranking de predicciones, es

decir, el porcentaje de coincidencias entre los elementos relevantes conocidos y los elementos devueltos por el modelo. Su fórmula es:

$$Precision\ at\ k = \frac{ERC \subset kRP}{k} \tag{3.1}$$

siendo ERC los elementos relevantes conocidos y kRP las primeras k posiciones del ranking de predicciones.

Recall *k*

El recall k [9] la división del número de elementos relevantes que hay en las primeras k posiciones del ranking de predicciones entre el número de elementos relevantes conocidos en el conjunto de test. Su fórmula es:

$$Recall\ at\ k = \frac{ERk}{ERT} \tag{3.2}$$

siendo ERk los elementos relevantes en las primeras k posiciones del ranking de resultados y ERT los elementos relevantes del conjunto de test.

AUC Score

El AUC (Area Under the Curve, Área Bajo la Curva) score [7] es la probabilidad de que un elemento relevante escogido aleatoriamente tenga un score mayor que un elemento no relevante escogido aleatoriamente.

Ranking recíproco

El ranking recíproco [10] es el inverso del score del elemento más relevante devuelto en el ranking de resultados. Su fórmula es:

$$Reciprocal\ rank = \frac{1}{SER} \tag{3.3}$$

siendo SER el score del elemento más relevante devuelto en el ranking de resultados.

En el modelo de *Spotlight* aparece como *MMR*, o *Mean Reciprocal Rank*.

Root Mean Square Error

El *RMSE* se define como el error que hay entre dos conjuntos de datos. Aplicado a nuestro caso, el *RMSE* compara un valor conocido con un valor

observado. Su fórmula sería la siguiente:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(C_i - O_i)^2}{n}} \tag{3.4}$$

siendo C una recomendación conocida y O una recomendación aportada por el sistema.

3.3. Tratamiento de los datos

Lo más común en los sistemas de recomendación es tener una gran cantidad de usuarios y de ítems. En cambio, la cantidad de valoraciones de ítems por cada usuario que se tiene es muy pequeña. Esto hace que la matriz con la que representamos las valoraciones sea muy grande pero con muchos valores a 0 (o celdas vacías, dependiendo de la representación escogida).

Esto es lo que llamamos *matrices dispersas* [16]. Trabajar con estas matrices supone un gran coste de rendimiento, por lo que necesitamos convertirlas a otras estructuras con las que poder trabajar mejor.

En el caso de *LightFM*, las estructuras utilizadas son:

- Matrices *COO*
- Matrices *CSR*

Matrices COO

Las *matrices COO* [13], o matrices coordenadas, dividen la matriz dispersa en tres vectores:

- Vector con los valores no nulos
- Vector con el índice de las filas
- Vector con el índice de las columnas

Esto lo podemos ver más claro con el siguiente ejemplo. Dada la siguiente matriz dispersa

$$\begin{bmatrix} 27 & 0 & 8 & 14 & 0 & 0 \\ 0 & 7 & 0 & 0 & 0 & 0 \\ 3 & 0 & 7 & 0 & 0 & 9 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 4 & 0 & 77 & 0 & 0 & 0 \end{bmatrix}$$

obtendríamos los siguientes vectores:

- Vector de valores:

$$\begin{bmatrix}27 & 8 & 14 & 7 & 3 & 7 & 9 & 1 & 4 & 77\end{bmatrix}$$

- Vector de filas:

$$\begin{bmatrix}1 & 1 & 1 & 2 & 3 & 3 & 3 & 4 & 5 & 5\end{bmatrix}$$

- Vector de columnas:

$$\begin{bmatrix}1 & 3 & 4 & 2 & 1 & 3 & 6 & 6 & 1 & 3\end{bmatrix}$$

Este tipo de estructura es utilizada por *LightFM* para representar la matriz de interacciones entre usuarios e ítems.

Matrices CSR

Otras estructuras ampliamente utilizadas, por ser más compactas que las matrices *COO*, son las matrices *CSR* [14]. Las matrices *CSR* (*Compressed Sparse Row*) no almacenan los índices de las filas, si no que almacenan punteros a los inicios de fila.

Utilizando como ejemplo la matriz dispersa del caso anterior, ahora obtendríamos:

- Vector de valores:

$$\begin{bmatrix}27 & 8 & 14 & 7 & 3 & 7 & 9 & 1 & 4 & 77\end{bmatrix}$$

- Vector de columnas:

$$\begin{bmatrix}1 & 3 & 4 & 2 & 1 & 3 & 6 & 6 & 1 & 3\end{bmatrix}$$

- Vector de filas:

$$\begin{bmatrix}1 & 4 & 5 & 9 & 10 & 11\end{bmatrix}$$

Este tipo de estructura es utilizada por *LightFM* para representar las matrices de features de usuarios e ítems.

Spotlight

Spotlight utiliza directamente los arrays de *numpy*, aunque también ofrece métodos para transformar estos arrays a matrices *COO* y *CSR*.

3.4. Imágenes

Se pueden incluir imágenes con los comandos standard de LaTeX, pero esta plantilla dispone de comandos propios como por ejemplo el siguiente:

Figura 3.1: Autómata para una expresión vacía

3.5. Tablas

Igualmente se pueden usar los comandos específicos de LaTeXo bien usar alguno de los comandos de la plantilla.

Herramientas	App AngularJS	API REST	BD	Memoria
HTML5	X			
CSS3	X			
BOOTSTRAP	X			
JavaScript	X			
AngularJS	X			
Bower	X			
PHP		X		
Karma + Jasmine	X			
Slim framework		X		
Idiorm		X		
Composer		X		
JSON	X	X		
PhpStorm	X	X		
MySQL			X	
PhpMyAdmin			X	
Git + BitBucket	X	X	X	X
MikTEX				X
TEXMaker				X
Astah				X
Balsamiq Mockups	X			
VersionOne	X	X	X	X

Tabla 3.1: Herramientas y tecnologías utilizadas en cada parte del proyecto

Técnicas y herramientas

4.1. Metodologías

Scrum

Scrum es una técnica de desarrollo ágil que se caracteriza por: [15]

- Adoptar una estrategia de desarrollo incremental, en lugar de la planificación y ejecución completa del producto.
- Basar la calidad del resultado más en el conocimiento tácito de las personas en equipos auto organizados, que en la calidad de los procesos completados.
- Solapamiento de las diferentes fases del desarrollo, en lugar de realizar una tras otra en un ciclo secuencia o en cascada.
- Realizar a diario una reunión con el objetivo de obtener realimentación sobre las tareas del equipo y los obstáculos que se presentan.

Al ser un proyecto educativo, las reuniones no han sido diarias, se han realizado cada dos semanas aproximadamente.

4.2. Control de versiones

Git

Git es un software de control de versiones distribuido diseñado por el creador del kernel Linux Linus Torvalds. Su objetivo es mantener un registro de los cambios que se producen en los archivos de proyectos que normalmente están compartidos.

4.3. Repositorio

- Herramientas consideradas: GitHub, GitLab y Bitbucket
- Herramienta escogida: GitHub

Se escoge *GitHub* por estar familiarizado con la plataforma al haber sido usada durante toda la carrera. Junto con *GitHub* se ha trabajado con *GitHub Desktop* para la realización de los commits.

4.4. Gestión del proyecto

- Herramientas consideradas: ZenHub y Trello y
- Herramienta escogida: ZenHub

La integración de *ZenHub* con *GitHub* es muy superior a la de *Trello*. No solo es integra directamente en la página del repositorio en *GitHub* sino que además ofrece herramientas muy útiles, como los reportes, especialmente el de *burndown*. Además, gracias a la extensión de *ZenHub* para navegador podemos ver el estado del repositorio de una manera muy sencilla y rápida.

4.5. Entorno de Desarrollo Integrado (IDE)

Python

- Herramientas consideradas: Jupyter, Spyder, PyCharm y Sublime Text
- Herramientas escogidas: Jupyter, Spyder y Sublime Text

Durante el inicio del proyecto se utilizó *Jupyter* para obtener en sus *notebooks* modelos de recomendación iniciales sin prestar mucho detalle a la organización del proyecto y sus paquetes.

Una vez se tuvo una idea clara de cómo iba a estar estructurado el proyecto se pasó a *Spyder*, con el que podíamos tener el código y la consola en el mismo sitio.

Finalmente, se pasó a *Sublime Text* para realizar los últimos cambios en el código de los modelos y para obtener la aplicación *Flask*.

Documentación

- Herramientas consideradas: Texmaker y Apache OpenOffice
- Herramienta escogida: Texmaker

Se elige *Texmaker* por encima de *OpenOffice* debido a la novedad que supone utilizar LaTeX para crear documentos.

Además, es una herramienta muy potente gracias a los comandos y funciones que tiene, que en ocasiones hacen que sea mucho más fácil realizar una tarea con *Texmaker* que con *OpenOffice*.

4.6. Librerías

NumPy

NumPy es una librería muy utilizada en el ámbito de la computación científica gracias a su potente manejo de arrays y a las funciones de álgebra lineal que contiene. Forma parte del ecosistema de *SciPy*.

En el proyecto se emplea *NumPy* para transformar listas de Python en arrays que puedan ser utilizados por los sistemas.

pandas

pandas, al igual que *NumPy*, es una librería muy utilizada en el campo de la computación científica debido a la facilidad que ofrece para manejar datos en tablas y series. También forma parte del ecosistema de *SciPy*.

En el proyecto se emplea *pandas* para el manejo inicial de los conjuntos de datos, pasando los *.csv* a *dataframes*.

LightFM

LightFM es una implementación para Python de populares algoritmos clásicos de recomendación [6]. Está creado por la misma gente que creó *Spotlight*.

Se decidió emplear *LightFM* debido a su aparente sencillez y por ser una librería bastante completa.

Spotlight

Spotlight es una librería creada por la misma gente que *LightFM* que utiliza *PyTorch* para construir modelos de recomendación [12] basados en aprendizaje profundo.

Se decidió emplear *Spotlight* por su aparente sencillez y por haber sido creada por la misma gente que la librería clásica.

Flask

Flask es un framework para Python con el que se obtiene una interfaz web en la que poder interactuar con el proyecto.

tkinter

Chardet

Chardet es un detector de encoding universal [**?**].

Se puede emplear en el proyecto para obtener el encoding de los archivos *.csv*. Con ello, se logra que el usuario no necesite saber qué encoding emplea los archivos de datos que quiere utilizar.

pickle

pickle es un módulo utilizado para la serialización de objetos en Python [**?**].

Se utiliza en el proyecto para guardar las matrices y los modelos obtenidos por los sistemas.

4.7. Datasets

Anime

Este conjunto de datos [3] contiene información relativa a las preferencias de 73.516 usuarios sobre 12.294 películas o series de anime. Cada usuario ha valorado unos 106 animes.

El conjunto de datos se encuentra disponible en: Anime Dataset.

Book-Crossing

Este conjunto de datos [17] contiene 1.149.780 valoraciones de 278.858 usuarios sobre 271.379 libros. Hay alrededor de 11 valoraciones por usuario.

El conjunto de datos se encuentra disponible en Book-Crossing Dataset.

MovieLens

Este conjunto de datos [4] contiene 100.000 valoraciones de 943 usuarios sobre 1.682 películas. Hay unas 106 valoraciones por usuario.

El conjunto de datos se encuentra disponible en los propios archivos descargados por la librería *LightFM* [11].

Last.FM

Este conjunto de datos [2] contiene las veces que 1.892 usuarios han escuchado a 17.632 artistas. Cada usuario ha escuchado alrededor de unos 49 artistas.

El conjunto de datos se encuentra disponible en Last.FM.

Dating Agency

Este conjunto de datos [1] contiene 17.359.346 valoraciones de 135.359 usuarios sobre 168.791 perfiles de otros usuarios. Hay unas 128 valoraciones por usuario.

El conjunto de datos se encuentra disponible en Dating Agency.

Aspectos relevantes del desarrollo del proyecto

Este apartado pretende recoger los aspectos más interesantes del desarrollo del proyecto.

5.1. Metodologías

Para la gestión del proyecto se ha adaptado la metodología ágil Scrum. Las principales características han sido:

- La duración de los sprints fue de dos semanas aproximadamente.
- Los sprints finalizaban con una reunión en la que se revisaba el trabajo hecho y se planteaban las tareas del siguiente sprint.
- Se utilizó ZenHub para llevar el seguimiento de las tareas.

5.2. Formación

El proyecto ha requerido obtener una serie de conocimientos que no se tenían inicialmente. Se ha estudiado el campo de los sistemas de recomendación y la utilización de aprendizaje profundo en Python. Para la formación en los sistemas de recomendación se usó el libro:

- *Mining of Massive Datasets* (Jure Leskovec, Anand Rajaraman y Jeffrey D. Ullman).

Para la formación en aprendizaje profundo en Python se realizó el siguiente curso:

- *Practical Deep Learning For Coders, Part 1* (Jeremy Howard).

5.3. Desarrollo del código

El proyecto se ha centrado en dos aspectos fundamentales: obtener un modelo clásico y obtener un modelo basado en aprendizaje profundo. Para la obtención del modelo clásico se estudió, en primer lugar, hacer uso de la librería Crab. Esta opción se descartó debido a la cantidad de fallos que saltaron durante la instalación (no se pudo llegar a instalar) y a la falta de actividad por parte de sus autores en el repositorio (no se ha tocado desde hace 7 años). Por todo esto, se decidió utilizar la librería LightFM.

5.4. Documentación

Las opciones que se plantearon para realizar la documentación fueron Apache OpenOffice y Texmaker. Ya que el Trabajo Fin de Grado lo veo como una forma de aprender conceptos nuevos que se desmarcan un poco de lo visto durante la carrera, opté por Texmaker debido a la novedad y al querer aprender y usar LaTeX(es la primera vez que lo utilizo).

Trabajos relacionados

Este apartado sería parecido a un estado del arte de una tesis o tesina. En un trabajo final grado no parece obligada su presencia, aunque se puede dejar a juicio del tutor el incluir un pequeño resumen comentado de los trabajos y proyectos ya realizados en el campo del proyecto en curso.

Conclusiones y Líneas de trabajo futuras

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.

Bibliografía

[1] Lukas Brozovsky and Vaclav Petricek. Recommender system for online dating service. In *Proceedings of Conference Znalosti 2007*, Ostrava, 2007. VSB.

[2] Iván Cantador, Peter Brusilovsky, and Tsvi Kuflik. 2nd workshop on information heterogeneity and fusion in recommender systems (hetrec 2011). In *Proceedings of the 5th ACM conference on Recommender systems*, RecSys 2011, New York, NY, USA, 2011. ACM.

[3] CopperUnion. Anime recommendations database. Technical report, Kaggle, 2017.

[4] GroupLens. Movielens. In *MovieLens*, 1998.

[5] Enrique Herrera-Viedma & Carlos Porcel & Lorenzo Hidalgo. Sistemas de recomendaciones: herramientas para el filtrado de información en internet. *Hipertext.net*, 2004.

[6] Maciej Kula. Lightfm. https://github.com/lyst/lightfm.

[7] Maciej Kula. *Model evaluation - auc score.*

[8] Maciej Kula. *Model evaluation - precision at k.*

[9] Maciej Kula. *Model evaluation - recall at k.*

[10] Maciej Kula. *Model evaluation - reciprocal rank.*

[11] Maciej Kula. Metadata embeddings for user and item cold-start recommendations. In Toine Bogers and Marijn Koolen, editors, *Proceedings of the 2nd Workshop on New Trends on Content-Based Recommender*

Systems co-located with 9th ACM Conference on Recommender Systems (RecSys 2015), Vienna, Austria, September 16-20, 2015., volume 1448 of *CEUR Workshop Proceedings*, pages 14–21. CEUR-WS.org, 2015.

[12] Maciej Kula. Spotlight. https://github.com/maciejkula/spotlight, 2017.

[13] Python Software Foundation. *Chardet: The Universal Character Encoding Detector.*

[14] Python Software Foundation. *pickle - Python object serialization.*

[15] SciPy.org. $scipy.sparse.coo_matrix$.

[16] SciPy.org. $scipy.sparse.csr_matrix$.

[17] Wikipedia. Scrum (desarrollo de software) — Wikipedia, the free encyclopedia. http://es.wikipedia.org/w/index.php?title=Scrum%20(desarrollo%20de%20software)&oldid=115368106, 2019. [Online; accessed 25-February-2019].

[18] Wikipedia. Sparse matrix — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Sparse%20matrix&oldid=893179870, 2019. [Online; accessed 12-April-2019].

[19] Cai-Nicolas Ziegler. Book-crossing dataset. Technical report, IIF, 2004.