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

IIC3633 Sistemas Recomendadores (2019-2)

Tarea 1

Indicaciones

- Fecha de entrega: **Viernes 6 de Septiembre de 2019, 20:00 horas.**
- La tarea debe realizarse **en grupos de a dos personas**. La copia será sancionada con una nota 1.1 en la tarea, además de las sanciones disciplinarias correspondientes.
- Entrega a través del repositorio personal en GitHub asociado a esta clase y tarea, por anunciar. Basta que lo entregue uno de los miembros del grupo, pero el README.md debe identificar claramente lo(a)s dos miembros del grupos.
- Cada hora o fracción de atraso descuenta 0.5 puntos de la nota obtenida, llegando a 1.0 en 6 horas. Se considera como entrega el último *commit* presente en el repositorio, es decir, la hora en la que este hace presente en GitHub (no su hora de creación). No se revisarán *commits* anteriores.

Objetivo

En esta tarea tendrán la oportunidad de poner en práctica sus conocimientos sobre Sistemas Recomendadores. En particular, exprimentarán con recomendación basada en feedback implícito y basada en contenido. Las librerías a utilizar serán **pyreclab**, desarrollada por Gabriel Sepúlveda, así como **implicit** para el algoritmo BPR. La parte de recomendación basada en contenido la harán principalmente con su propio código en python y scikit-learn.

Dataset

En esta tarea utilizarán datos de imágenes de *WikiMedia Commons* con las que usuarios han interactuado, indicando si aprueban o no ciertas imágenes que se suben al portal. El dataset consiste en:

- Dataset de train (*training.csv*): 96,534 registros que contienen id del usuario, id de la imágen con la que interactuó y el timestamp. Para probar sus modelos haga las particiones que estime necesarias sobre este set de datos para validar y testear su rendimiento. Descargar aquí.
- Dataset de test (*test.csv*): una lista de 1,077 IDs de usuarios. Para cada uno de estos usuarios debes generar una lista de 10 recomendaciones. Descargar aquí.

- Archivo con representación de imágenes en forma de vectores obtenidos con una red neuronal ResNet (neural embeddings). En cada fila del archivo en formato json, el key es el id de la imagen y los números a continuación son los valores del embedding de la imagen. El tamaño del embedding de cada imagen es 2.048. Descargar aquí. Considerar que este archivo pesa alrededor de 278MB.
- Si quieres analizar tus recomendaciones puedes ver las imágenes en una URL de cada imagen. Por ejemplo, para ver la imagen id **201312221** puedes buscarla aquí. Te pedimos no scrapear las imágenes para evitar generar mucho tráfico y porque queremos evitar tener problemas de copyright.

1 Exploración de datos (20%)

El siguiente análisis hágalo con el dataset de training.

- Grafique la distribución de usuarios con número de interacciones, identifique los 5 usuarios más activos en el dataset de *WikiMedia Commons*. Comente la forma de la distribución y qué porcentaje de las interacciones han sido hechas por estos 5 usuarios.

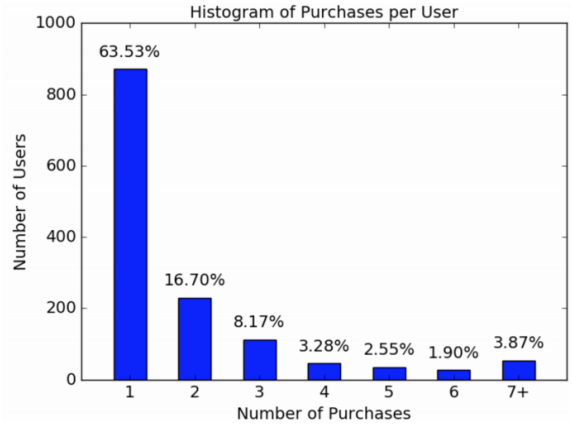

Figure 1: Ejemplo de gráfico de distribución, en este caso de compras por usuario. Haga algo similar para la cantidad de interacciones en el dataset de training de la tarea.

- Grafique la distribución de imágenes por número de interacciones, Identifique las 5 imágenes que han sido más vistas en el dataset de *WikiMedia Commons*. Comente la forma de la distribución y qué porcentaje de las interacciones han sido sobre estas 5 imágenes.
- Tabla con número de usuarios distintos, número de items distintos, promedio y desviación estándar de imágenes por usuario, promedio y desviación estándar de usuarios por imagen, y densidad del dataset (o *sparsity*) en cuanto a interacciones.

2 Recomendación basada en interacciones (40%)

2.1 Actividad 1: ALS con gradiente conjugado

- Muestre análisis de sensibilidad de resultados para métricas MAP@10 y nDCG@10 en función de tiempos, hiperparámetros, *learning rate* (0.001, 0.01), factores latentes (50, 100, 200) y regularización (0.01, 0.1). Grafique cada uno y comente.
- Reporte tiempos de entrenamiento y recomendación.

2.2 Actividad 2: Bayesian Personalized Ranking (BPR)

- Para esta actividad no es posible utilizar la librería pyreclab, por lo que se sugiere usar implicit u otra que tenga implementado el algoritmo. Se debe especificar la librería utilizada en su informe.
- Muestre análisis de sensibilidad de resultados para métricas MAP@10 y nDCG@10 en función de tiempos, hiperparámetros, *learning rate* (0.001, 0.01), factores latentes (50, 100, 200) y regularización (0.01, 0.1). Grafique cada uno y comente.
- Reporte tiempos de entrenamiento y recomendación.
- Explique su estrategia de *negative sampling*.

2.3 Comparación de resultados

- Haga una tabla comparativa de las métricas para los 2 métodos utilizados en el mejor de los casos luego de modificar hiperparámetros. Identifique claramente los valores de los hiperparámetros para cada método.
- Con el mejor de los métodos (y su mejor conjunto de hiperparámetros) genere 10 recomendaciones por cada usuario del dataset de testing. La lista de recomendaciones debe ser entregada en este formato:

```
{
"user_id1": [img_id1, img_id2, ...],
"user_id2": [img_id1, img_id2, ...],
"user_id3": [img_id1, img_id2, ...],
"user_id4": [img_id1, img_id2, ...],
...
}
```

3 Recomendación basada en contenido (40%)

3.1 Actividad 1: Recomendación basada en contenido

En esta actividad el objetivo es obtener una representación vectorial de imágenes con las que interactuó el usuario y recomendar basándose en métricas de similaridad de éstas con otras imágenes. (*tip:* para esta actividad no es obligatorio utilizar la librería *pyreclab*).

- Antes de recomendar, se sugiere reducir la dimensionalidad de los *embeddings* de imágenes. Normalice los vectores (estandarización z-score) y luego aplique PCA, generando *embeddings* de tamaño 20 y 50. Compare los resultados de MAP@10 y nDCG@10 para ambas dimensiones.

- Para recomendar utilice el algoritmo de recomendación de imágenes basada en contenido descrito en la ecuación (6) de este *paper*. Compare las 3 formas de *scoring* que se indican en el *paper*.
- Muestre ejemplos de recomendación con imágenes reales para algún usuario en particular, comparando las imágenes con las que este ha interactuado y las que el se le recomendaron. Comente.
- Con el mejor de los métodos (y su mejor conjunto de hiperparámetros) genere 10 recomendaciones por cada usuario del dataset de testing. La lista de recomendaciones debe ser entregada en este formato:

```
{
"user_id1": [img_id1, img_id2, ...],
"user_id2": [img_id1, img_id2, ...],
"user_id3": [img_id1, img_id2, ...],
"user_id4": [img_id1, img_id2, ...],
...
}
```

BONUS: Ensemble o Híbrido

Genere recomendaciones creando un método que combine las recomendaciones de los métodos basados en interacciones y de contenido. Recuerde generar 10 recomendaciones para cada usario del dataset de test, en el mismo formato solicitado antes:

```
{
"user_id1": [img_id1, img_id2, ...],
"user_id2": [img_id1, img_id2, ...],
"user_id3": [img_id1, img_id2, ...],
"user_id4": [img_id1, img_id2, ...],
...
}
```

Consideraciones y formato de entrega

La tarea deberá ser entregada a través del repositorio personal en GitHub asociado a esta clase y tarea, por anunciar. Se deberá desarrollar la tarea en un Jupyter Notebook con todas las celdas ejecutadas, es decir, no se debe borrar el resultado de las celdas antes de entregar. Si las celdas se encuentran vacías, se asumirá que la celda no fue ejecutada. Es importante que todas las actividades tengan respuestas explícitas, es decir, no basta con el *output* de una celda para responder.