



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

IIC3633 Sistemas Recomendadores (2020-2)

Tarea 1

Indicaciones

- Fecha de entrega: **Miércoles 16 de Septiembre de 2020, 20:00 horas.**
 - La tarea debe realizarse **individualmente o en grupos de máximo tres 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 grupo.
 - 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.
 - Se sugiere hacer la tarea en Google colab o en jupyter notebooks para facilitar la revisión. Deberán entregar estos notebooks ejecutados como parte de su código.
-

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. Puede experimentar con otras bibliotecas en algunos casos, por ejemplo **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 *Pinterest* con las que usuarios han interactuado. El dataset consiste en:

- Dataset de train (*train.csv*): **78023** registros que contienen id del usuario, id de la imagen 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_users.txt*): una lista de **1276** IDs de usuarios. Para cada uno de estos usuarios debes generar una lista de 20 recomendaciones. Descargar [aquí](#). Recuerda que con tus predicciones este dataset compites por un bono si quedas entre los 5 mejores grupos del curso en términos de MAP@20 y nDCG@20.
- Embeddings de ResNet50 (*resnet50_embeddings.npy*) Archivo con representación de imágenes en forma de vectores obtenidos con una red neuronal, específicamente ResNet50 (neural embeddings). Es un archivo **.npy**, se carga utilizando la librería numpy. Cada fila del arreglo en el primer lugar está el *item_id* y en el segundo se encuentra el vector con el embedding. El tamaño del embedding de cada imagen es 2.048. Descargar [aquí](#). Considerar que este archivo pesa alrededor de 600MBs.
- Si quieres analizar tus recomendaciones puedes acceder a cada imagen poniendo el *ID de la imagen* a continuación de <http://niebla.ing.puc.cl/iic3633-2020/> o descargarlas todas [aquí](#).

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 *imágenes*. Comente la forma de la distribución y qué porcentaje de las interacciones han sido hechas por estos 5 usuarios.

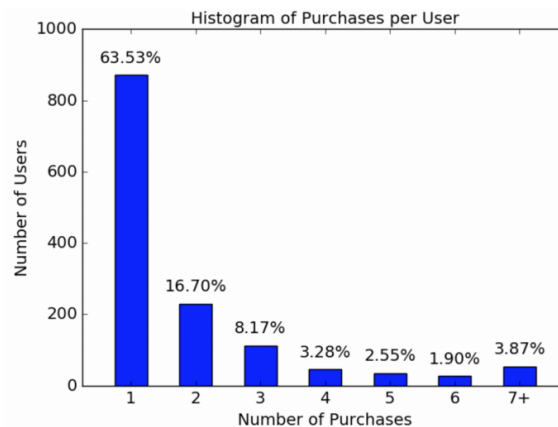


Figure 1: Ejemplo de gráfico de distribución, en este caso de compras por usuario. Haga algo similar para la cantidad de interacciones en el dataset de training de la tarea.

- Grafique la distribución de imágenes por número de interacciones, Identifique las 5 imágenes que han sido más vistas en el dataset de *imágenes*. 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@20 y nDCG@20 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 utilizar la librería **pyreclab** pero puede también usar la librería **implicit**.
- Muestre análisis de sensibilidad de resultados para métricas MAP@20 y nDCG@20 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:

```

1 {
2 "user_id1": [img_id1, img_id2, ...],
3 "user_id2": [img_id1, img_id2, ...],
4 "user_id3": [img_id1, img_id2, ...],
5 ...
6 }
```

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 similitud 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@20 y nDCG@20 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:

```

1 {
2 "user_id1": [img_id1, img_id2, ...],
3 "user_id2": [img_id1, img_id2, ...],
4 "user_id3": [img_id1, img_id2, ...],
5 ...
6 }

```

BONUS: VBPR

Genere recomendaciones utilizando **Visual Bayesian Personalized Ranking (VBPR)** que está diseñado especialmente para recomendación de imágenes considerando igualmente filtrado colaborativo. Recuerde generar 10 recomendaciones para cada usuario del set de test, en el mismo formato solicitado antes, y entregar en formato json para que el equipo docente haga un ranking de los mejores grupos:

```

1 {
2 "user_id1": [img_id1, img_id2, ...],
3 "user_id2": [img_id1, img_id2, ...],
4 "user_id3": [img_id1, img_id2, ...],
5 ...
6 }

```

Entregables

La tarea deberá ser entregada a través del repositorio de su grupo y tarea en **GitHub classroom**. Siga el siguiente link y las instrucciones para crear el repositorio de su grupo: [link a tarea 1](#).

Se deberá ENTREGAR un informe en formato PDF, así como código en uno o varios **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 toda la información solicitada de parámetros y análisis tengan una explicación, es decir, no basta con el *output* de una celda para responder una pregunta, se debe explicar qué se está respondiendo.

Informe. En el repositorio de su grupo deberá entregar un informe en PDF que contenga el análisis de datos solicitado en la Sección 1 (**Exploración de Datos**) y un resumen de los resultados y parámetros para entender los modelos y experimentos realizados sobre ALS, BPR, y recomendación basada en contenido. Si realiza el bonus, debe agregar una sección de forma análoga.

Código. Por cada uno de los métodos solicitados (ALS, BPR y recomendación basada en Contenido) debe entregar en su repositorio el código que permita replicar los resultados obtenidos. Se solicita entregar uno o varios jupyter notebooks que permitan replicar experimentos.

Es obligatorio agregar un archivo README.md al repositorio de su tarea que permita entender la estructura de archivos y detalles necesarios para replicar los experimentos realizados.