Tarea 3 - ¿Qué película me recomiendas? Reconocimiento de Patrones en Minería de Datos

Profesor: Marcelo Mendoza - mmendoza@inf.utfsm.cl
Ayudante: Ignacio Espinoza - ignacio.espinoza@alumnos.inf.utfsm.cl

3 de junio de 2017

1. Introducción

En esta tarea tendrán que trabajar con el *framework* Apache Spark[1], que es un motor de procesamiento a gran escala, y con MLlib[2] que es la librería de Machine Learning de Spark. Los objetivos de la tarea serán:

- Implementar y analizar algoritmo Alternating Least Squares (ALS) de MLlib para la creación un recomendador de películas.
- Estudiar cómo trabaja Spark, su formato de datos y qué tal es su desempeño a gran escala.

1.1. Apache Spark

Como Spark posee APIs para trabajar en Java, Scala, Python y R, puede usar el lenguaje que más le acomode. Para descargar Spark, ingresar a `https://spark.apache.org/downloads.html` y descargar la última versión estable (2.1.1).

IMPORTANTE: Antes de empezar a programar revise bien la documentación. Spark posee estructuras de datos (RDD, Dataframes) y funciones especiales (transformaciones y acciones) para trabajar con datos de forma paralela/distribuida. Hay ejemplos para que usted revise y ejecute.

1.2. MovieLens 10M

Se trabajará con el dataset MovieLens 10M, el cual contiene información de 10 millones de ratings anónimos de aproximadamente 10.000 películas hechas por usuarios de MovieLens. Para descargar los datos ingresar a `https://grouplens.org/datasets/movielens/10m/`. Dentro del archivo viene: **movies.dat** (información de las películas), **users.dat** (información de usuarios), **ratings.dat** (ratings de las películas), **README** (descripción de los archivos anteriores).

[1] https://spark.apache.org/
[2] https://spark.apache.org/docs/latest/ml-guide.html

2. Trabajo a realizar

1. Descargue el dataset del link indicado. Lea como se estructura cada archivo para su manejo.
2. Descargar Spark y configúrelo para su trabajo. Se recomienda trabajar en un sistema operativo basado en GNU/Linux.
3. Cargue los datos en un RDD o Dataframe, según la implementación a utilizar.
4. Por medio del algoritmo ALS entrene un sistema recomendador. Pruebe diferentes configuraciones de parámetros (para λ y rank), detallando en el informe cada una de ellas (al menos 5). Use α por defecto. Evalúe cada modelo con *Root-mean-square error* (RMSE), identificando cuál configuración entrega un mejor resultado.
5. Usando la mejor configuración encontrada muestre, para los primeros 5 usuarios, las Top-10 películas recomendadas para cada uno.
6. Si se usa la configuración con mayor RMSE, ¿Cambia el resultado para las Top-10 películas recomendadas a los primeros 5 usuarios?

2.1. Prueba de rendimiento

1. Repita el paso 4 de la sección anterior cambiando la RAM asignada para su ejecución (por defecto se ocupa 1GB). ¿Hay cambios notorios en tiempo de ejecución? ¿Tiene algún efecto, positivo o negativo, en el resultado de la recomendación en términos de RMSE? Ejemplo de ejecución con 4GB asignados:

```
>>> bin/spark-submit --executorr-memory 4g spark-recomendations.py
```

2. Asigne una cantidad de threads igual a la cantidad de núcleos lógicos que tenga su computador (local[*]) [3] y repita el punto 4. ¿Hay cambios notorios en desempeño, como reducción de tiempo de ejecución? ¿Tiene algún efecto, positivo o negativo, en el resultado de la recomendación en términos de RMSE? Ejemplo de ejecución:

```
>>> bin/spark-submit --master local[*] spark-recomendations.py
```

2.2. Desarrollo

En el informe deberá incluir (brevemente):

- Descripción de Apache Spark y MLlib.
- ¿Qué son los *RDD* y *Dataframe*?
- Descripción del dataset utilizado.

3. Entregable y consideraciones:

- Los equipos de trabajo pueden ser de 2 a 3 personas.

[3] `http://spark.apache.org/docs/latest/submitting-applications.html#master-urls`

- Informe: subir archivo tar.gz con el informe y códigos utilizados. El informe debe contener los resultados obtenidos en su tarea, configuraciones de parámetros utilizadas y desarrollo del punto 2.2. Incluir Nombre y rol de los integrantes.
- **FECHA DE ENTREGA: Viernes 23 de Junio hasta las 23:55. Pasada la hora tendrá 20 puntos de descuento por día de atraso.**

4. Documentación

- MLlib: `https://spark.apache.org/docs/latest/ml-guide.html`
- Correr código en Spark: `https://spark.apache.org/docs/latest/submitting-applications.html`
- Quick Start Spark: `https://spark.apache.org/docs/latest/quick-start.html`