ParallelFischerScoring_binomial

Manual Técnico

Tabla de contenido

1. Descripción

ParallelFischerScoring_binomial es un software que permite estimar los parámetros de un modelo de Regresión Logística Binaria a través del algoritmo de Fischer Scoring mediante procesamiento heterogéneo CPU/GPU.

2. Requisitos

Para el funcionamiento de este software, es necesario el siguiente hardware:

1. Tarjeta gráfica NVIDIA

También es necesario el siguiente software:

1. Python 2.7: https://www.python.org/download/releases/2.7/
2. Numpy 1.14.2: https://pypi.org/project/numpy/
3. Pycuda 2017.1.1: https://pypi.org/project/pycuda/
4. Skcuda 0.5.2: https://scikit-cuda.readthedocs.io/en/latest/install.html
6. Scikit-image 0.14.2: https://pypi.org/project/scikit-image/

3. Instalación

Para llevar a cabo la instalación, tenga en cuenta que es necesario tener instalado Python (mínimo versión 2.7), de lo contrario, no será posible llevar los siguientes pasos. Cuando se instala Python, tendrá instalado PyPl (Python Package Index) por defecto, que es el que permite instalar paquetes de Python, con el comando **pip**.

3.1. Instalar la tarjeta Grafica

Primero se debe verificar si el sistema operativo ya reconoce la tarjeta gráfica, para eso se digita el siguiente comando y se verifica que la tarjeta gráfica NVIDIA se encuentre en la lista:

```
lisci | grep -i nvidia
```

En caso de que no aparezca se digita el siguiente comando para actualizar los controladores conectados al computador

```
sudo update-pciids
```

Comprobando que la tarjeta gráfica ya se encuentra instalada se procede a realizar la instalación de los drivers por medio de los siguientes comandos:

```
sudo apt-add-repository ppa:xorg-edgers/ppa
```

```
sudo apt-add-repository ppa:ubuntu-x-swat/x-updates

sudo apt-get update

sudo apt-get install nvidia-current nvidia-settings
```

3.2. Instalar la CUDA y PyCUDA

Para correr la aplicación es necesario instalar CUDA y PyCUDA, el primero se instala por medio de los siguientes dos comandos:

```
sudo apt-get install cuda

sudo apt-get install nvidia-cuda-toolkit
```

Con esto instalado se procede a realizar la configuración de dos variables de entorno, la primera PATH y la segunda CUDA_ROOT de la siguiente forma:

```
export PATH=/usr/local/cuda/bin:$PATH

export CUDA_ROOT=/usr/local/cuda
```

Con estas dos variables de entorno se podrá ejecutar el siguiente comando para instalar PyCUDA:

```
Pip install pycuda
```

3.3. Clonar Repositorio

Como primer paso, clone el repositorio de GitHub en su equipo.

```
Git clone https://github.com/Parall-UD/ParallelFischerScoring_binomial
```

3.4. Instalar Librerías de Python

Con el comando pip debe instalar las librerías de Python, de la siguiente forma:

```
Pip install numpy

Pip install pycuda

Pip install scikit-image

Pip install scikit-cuda
```

4. Inicializar la aplicación

La ejecución de la aplicación presenta dos opciones de ejecución, la primera desde una versión de CPU, la segunda desde una versión de GPU. Para iniciar la aplicación, ubique la carpeta en la cual usted clono el repositorio a través de la consola.

Ahora, utilice el siguiente comando para realizar la ejecución de la aplicación en CPU:

```
python fisher_cpu.py 'parametro_de_entrada'
```

Y el siguiente comando para realizar la ejecución de la aplicación en GPU:

```
python fisher_gpu.py 'parametro_de_entrada'
```

A continuación, se describen los parámetros y sugerencias:

- **Parámetro_de_entrada:** String con la ruta absoluta del archivo .csv que contiene los datos, el archivo .csv debe estar compuesto de 4 columnas que representan:
    - **y:** Respuesta dicotómica (0-1)
    - **r:** Es la columna asociada al parámetro de intercepto (B0)
    - **x1:** Conjunto de datos aleatorios en una distribución normal
    - **x2:** Conjunto de datos aleatorios en una distribución normal
- **Salida:** Esta compuesta de 3 partes:
    - Numero de Iteraciones
    - Lista de Betas obtenidos
    - Tiempo en escala de segundos de cuanto se demoró la ejecución en la aplicación
- Si tiene más de una versión de Python se recomienda especificar la que se usará, de la siguiente manera:

```
Python2.7 fisher_gpu.py 'parametro_de_entrada'
```