

Survival Analysis with Apache Spark and Apache SystemML on Stackexchange

Mateo Álvarez Calvo

May 14, 2017

Contents

1	Introduction & main goals	3
1.1	Main technologies	3
1.1.1	Apache Spark	3
1.1.2	Apache SystemML	4
1.2	The Stackexchange data	4
1.3	Main Objectives	4
2	Technologies	5
2.1	Apache Spark	5
2.1.1	Apache Spark Structure	5
2.2	Apache SystemML	5
2.2.1	SystemML Structure	5
2.3	Reproducible research with Python, Scala and Jupyter	5
2.3.1	Environment setup	5
3	Infrastructure and resources	6
3.1	Architecture scheme	6
3.2	Configuration	6
3.3	Workflow	6
4	Results	7
4.1	Used data	7
4.2	Kaplan-Meier model	7
4.3	Cox Proportional Hazards Model	7
5	Conclusions	7
5.1	Most important results	7
5.2	Lessons learnt	7

1 Introduction & main goals

Many studies have been done over the Stackexchange community [such as], one of the biggest Q&A sites in the world. The present is yet another study over the data of the famous site, but in this case, the study has two particularities, the use of Apache Spark with the library of Apache SystemML for the processing in a parallel environment, and the use of Survival Analysis to analyze the impact of the variables in the time an answer is accepted for each question, the "survival of each question" in the community.

1.1 Main technologies

As one of the biggest Q&A communities, Stackexchange has large amount of data of each interaction. Stackexchange is separated in several communities, regarding different topics. These communities can be small, as [] or really big, as Stackoverflow, the developers community. This particularity makes necessary the use of technologies prepared to process large amounts of data, in the later case.

The purpose of the present study is to analyze this big community, so a distributed processing technology has to be used. For this purpose, Apache Spark, the latest distributed open-source processing technology, has been chosen to parallelize the operations on the data.

Spark ML is the machine learning library of spark, which contains lots of algorithms. It also includes some Survival Analysis algorithms, but just for parametric modeling. This gives an excuse to use the recently adopted by the Apache Foundation SystemML, a machine learning library developed by IBM, which has non, semi and parametric algorithms for survival analysis.

Regarding the development environment, Jupyter Notebook provides a simple and flexible interface for this analysis, and can also be integrated with Spark, allowing the complete development in just one environment.

1.1.1 Apache Spark

Apache Spark is a distributed processing technology developed in Scala by Databricks that represents the next step of Apache Hadoop, including the best parts of it, such as the Hadoop File System, but under a complete new paradigm that allows operations different from the famous map-reduce, using RAM as storage for results rather than writing to disk, lazy and optimized execution of tasks, and special focus on machine learning and SQL-like language, to mention some of the main features.

cosa	cosa
------	------

Table 1

1.1.2 Apache SystemML

Recently included in the Apache Foundation Incubating program, Apache SystemML is a machine learning library that works over distributed frameworks, Spark or Hadoop, written in Java.

This library provides many distributed implementations of important algorithms as well as a sintaxis to create new algorithms in a distributed mode. It provides a high-level declarative machine learning language, which has two variations, the R-like sintaxis, DML, and the Python-like sintaxis, Py-DML. All the

1.2 The Stackexchange data

Stackexchange facilitates raw partially processed? data from the communities database every once in a while or every three months? for data scientists and people in general to download and analyze. The data is available in a torrent file and each package has about 35 - 40 GB of compressed information.

This compressed file has data from different communities for a certain period of time. In this case, the analysis is done over Scyfy community, which is a median size community for science-fiction Q&A.

Selecting just the data from Scyfy community, it weights around 110 MB in a 7z compressed format. The uncompressed data is divided in 8 files, regarding different information:

1.3 Main Objectives

For this study several objectives have been proposed:

- Use Spark to make the data cleaning to create a script for further research on the Stackexchange site.
- Verify SystemML integration with Spark for further research and scalability.
- Use SystemML survival analysis algorithms to analyze Stackexchange's data and obtain conclusions on the main variables affecting the time taken by the community to answer a question.

2 Technologies

The downloaded data from Stackexchange for the analysis [weights?] 40 GB, which is enough amount [size] to consider distributed processing.

2.1 Apache Spark

The latest distributed processing technology for data processing is Apache Spark, born in [2015] in [Ucla?], which is the predecessor of the well known processing framework Apache Hadoop, with all its ecosystem.

Comparing to Hadoop, Spark is able to speedup calculations using RAM instead of disk, with a higher variety of operations, not limited to Map-Reduce, and [].

2.1.1 Apache Spark Structure

2.2 Apache SystemML

Developed by IBM, Apache SystemML is the [precursor] of Spark ML (MLLib), the machine learning libraries of Spark. SystemML is coded in a self-made programming language, Distributed Machine Learning (DML) and it can be used from Spark or Hadoop in a [submit-like execution or in a interactive execution, the one it has been used in this study]

2.2.1 SystemML Structure

2.3 Reproducible research with Python, Scala and Jupyter

Going down to the selection of the development environment, it was important to use a standardized one so that the analysis could be reproduced by anyone. Jupyter notebook is one of the most commonly used, specially in the education and investigation institutions, it is easy to use and configure and it can be easily integrated with Spark.

To configure the same environment as the one used in this study, some steps have to be followed:

2.3.1 Environment setup

The first step is to setup Jupyter Notebook, either running it in a Docker container or installing it directly on the machine. The docker image can be obtained entering the fol-

Technology	Version
Jupyter Notebook	4.3.1
Python	3.5
Scala	2.11
Toree kernel	0.2.0.dev1
Spark	2.1
SystemML	0.12.0

Table 2: Technologies and versions used

lowing command in a shell: *docker pull jupyter/notebook*. Regarding the other option, installing it, the instructions can be found in the following link: <http://jupyter.readthedocs.io/en/latest/install>.

Once Jupyter Notebook running, the kernels have to be configured.

For the python kernel, the SystemML library has to be installed, to do so,

3 Infrastructure and resources

As commented before, selecting an environment that

3.1 Architecture scheme

3.2 Configuration

3.3 Workflow

The first step is to clean the data and make the feature selection

4 Results

4.1 Used data

4.2 Kaplan-Meier model

4.3 Cox Proportional Hazards Model

5 Conclusions

5.1 Most important results

5.2 Lessons learnt