University of Zurich[UZH]

Department of Informatics

University of Zurich
Department of Informatics
Binzmühlestr. 14
CH-8050 Zürich
Phone. +41 44 635 43 11
Fax +41 44 635 68 09
www.ifi.uzh.ch/dbtg

UZH, Dept. of Informatics, Binzmühlestr. 14, CH-8050 Zurich

Alessandro De Carli

Switzerland

Prof. Dr. Michael Böhlen
Professor
Phone +41 44 635 43 33
Fax +41 44 635 68 09
boehlen@ifi.uzh.ch

Zürich, 18. März 2014

Large Scale Centroid Decomposition under Spark

Master-Basismodul (3 ECTS):

Work overview:

The Centroid Decomposition (CD) [1] is a matrix decomposition technique that decomposes an $n \times m$ matrix $\mathbf{X} = [X_1|\ldots|X_m]$ into an $n \times m$ *loading* matrix $\mathbf{L} = [L_1|\ldots|L_m]$ and an $m \times m$ *relevance* matrix $\mathbf{R} = [R_1|\ldots|R_m]$ as follows:

$$CD(\mathbf{X}) = \mathbf{L}, \mathbf{R} \quad (1)$$

$$s.t. \quad \mathbf{X} = \mathbf{L} \times \mathbf{R}^T \quad (2)$$

$$= \sum_{i=1}^{d} L_i \times R_i^T$$

Where $d \leq m$ is the number of dimensions to compute.

Khayati et al. [2] proposed a scalable implementation of the Centroid Decomposition, termed SCD, that reduces its space complexity from quadratic to linear. However, the run time complexity remains quadratic. Due to this runtime complexity, SCD does not scale to large matrices of several millions of elements. As an example, it takes around 2 hours to compute the SCD of a matrix containing up to 20 millions elements (half million of rows and 4 columns).

The aim of this thesis is to investigate and implement a parallelizable version of SCD that scales up to large matrices of billions of values. The implementation of SCD under a classical distributed platform, e.g., Hadoop, will produce little benefit since SCD is an iterative process and Hadoop is not suitable for iterative algorithms. Therefore, the implementation should be performed on a platform that supports iterative algorithms. We propose to use Spark platform ([3], [4], [5]) since it is based on Resilient Distributed Datasets (RDD). The latter are partitioned collection of objects that efficiently handle iterative algorithms.

The implementation will be firstly executed on a single machine and then on a parallel environment e.g., Amazon EC2 [6]. The proposed implementation should scale to large scale matrices of millions of rows and millions of columns.

Work tasks:

1. Familiarize yourself with RDDs and Spark (see [3], [4] and [5]).
2. Understand and implement SCD algorithm under Spark.

Literature:

1. Chu, M.T., and Funderlic, R.E.: *The Centroid Decomposition: Relationships Between Discrete Variational Decompositions and SVDs*, in SIAM J. Matrix Analysis and Applications, 2002.
2. Khayati, M., Böhlen, M.H., and Gamper, J. *Memory-efficient Centroid Decomposition for Long Time Series*, in ICDE, 2014.
3. Zaharia., M. et al. *Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing*, in NSDI, 2012.
4. Zaharia., M. et al. *Spark: cluster computing with working sets*, in HotCloud, 2010.
5. `http://spark.incubator.apache.org/`.

Task assignment and supervisor:

- Mourad Khayati (mkhayati@ifi.uzh.ch)

Starting date of thesis: 24/03/2014

Ending date of thesis: 23/06/2014

University of Zurich
Department of Informatics

Prof. Dr. Michael Böhlen
Professor