Ludwig-Maximilians-Universität München
Institut für Informatik
Dr. Eirini Ntoutsi
PD Dr. Matthias Schubert

Knowledge Discovery in Databases II

WS 2015/2016

Übungsblatt 5: Parallel Data Mining

Aufgabe 5-1 A brief Introduction to Spark

The examples for parallel data mining in this tutorial are based on Apache Spark offering easy access to parallel algorithmic patterns like map and reduce.In this exercise, we have a look on the basic concepts and write a simple method for counting the letters in a string.

Note that it is not required to really have a spark cluster available to practise spark programming. For the exercise, we will rely on a local spark context which can be run on any machine where the pyspark library is in the python path. To use spark in python, download spark and add the python subfolder to the *python_path* system variable. In the template *SparkIntro_template.py*, you see how to create a local spark context.

(a) The most important data structure in Spark is the so-called RDD (resilient distributed dataset). A RDD represents a large data collection which might be spread on a distributed file system like the HDFS (Hadoop file system). RDDs are generated using various methods of the spark context. Generate a new RDD based on the method *parallelize()* which transforms local input data into an RDD.

(b) The RDD object provides two types of methods, transformations and actions. A transformation like map, reduceByKey, filter, join etc. transforms the values within the RDD. Actions collect data from the RDD and transfer the results to the master. Note transformations are not processed until an action is called. Thus, the computation of transformations is delayed and the transformations are stored in a graph structure until an action is performed. Implement a map and a reduceByKey method for generating an RDD containing pairs of the form (letter, counter).

(c) After defining the transformations, we need to perform actions to trigger the transformation and receive an result on the master. Implement actions to collect all (letter, count) pairs, receive 5 arbitrary pairs and draw a 50% sample of the of the pairs.

Aufgabe 5-2 Parallel K-Meas using Spark

In this excercise, we want to implement a simple parallel version of the k-means algorithm.

(a) Download the template *Para_KMeans_template.py* and study the given code.

(b) Write a method *init_cluster_centers(k,D)* which draws a sample having k elements from the RDD D. Convert the result to a $k \times d$ numpy matrix.

(c) Write a method *assign_cluster* receiving a pair of the format (cluster_id(counter, vector)) which replaces cluster_id with the row index of the nearest centroid in the centroid matrix C.

(d) Complete the k-means code by calling a mapper for the assign_cluster method. Furthermore, implement reduceByKey methods which compute the new cluster centers based on the assignment of the mapper.

(e) Run the complete method for the data set *birch2.csv*.