CS4225/CS5425 Big Data Systems for Data Science

Assignment 2: Spark

Chengxi Xue, Bingsheng He
School of Computing
National University of Singapore
xuechengxi@u.nus.edu

Coding Assignment Guideline for CS4225&5425: Assignment 2

- Task1: CommonWords using Spark
- Task2: K-means
- Information about assignment 2:
  - Submission requirement
  - What is this coding assignment about (you need to implement them into Hadoop)

Task1: Commonwords in Spark

- You need to implement the **Commonwords** using Spark.
- The application description is same as Task1 in assignment1, except that we require you to write the program in Scala.
  - Why Scala? Because it is native in Spark.
- Learn Scala here: https://docs.scala-lang.org/learn.html

Task 1: Report

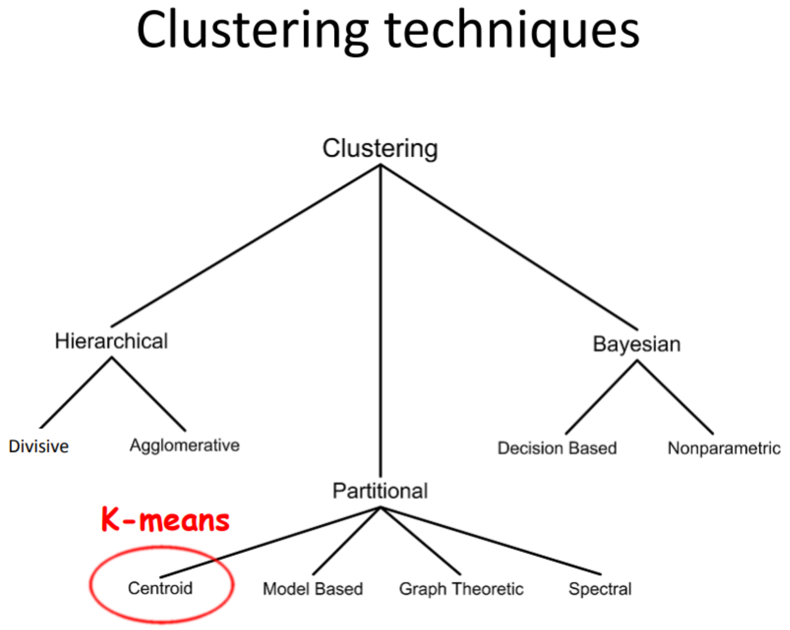
- You need to summarize Task1 and write a report **(up to 2 pages)**, including at least the following two aspects.
- Comparisons on programming with Hadoop and Spark
  - The difference between your implementation with two programming platforms.
  - Pros and Cons among Hadoop and Spark
  - …
- Comparisons on runtime execution with Hadoop and Spark
  - Program performance (in comparison with Hadoop)
  - Pros and Cons between Hadoop and Spark
  - …

Task2

- 1.The goal of Task2 is to implement a k-means algorithm using Scala which clusters some posts according to their score and domains. Moreover, this clustering should be executed in parallel for different domains.
- **2.Do not use some libraries like Mllib directly, you need to implement k-means step by step.**

Task Overview

- Motivation
  - Clustering is an unsupervised learning problem whereby we aim to group subsets of entities with one another based on some notion of similarity.
- Different kinds of clustering techniques:
  - Hierarchical algorithms
  - Partitional algorithms
  - Bayesian algorithms.



k-means

- k-means is one of the most commonly used clustering algorithms that clusters the data points into a predefined number of clusters.
- Given k, the k-means algorithm works as follows:
  - 1. Choose k (random) data points (seeds) to be the initial centroids, cluster centers
  - 2. Assign each data point to the closest centroid
  - 3. Re-compute the centroids using the current cluster memberships
  - 4. If a convergence criterion is not met, repeat steps 2 and 3

Background

- Now we have some popular question-answer platforms like Quora, yahoo answers, StackOverflow…
- There are a lot of posts there, we can cluster some posts according to their score and domains(tags) to find some interesting results.
- For example:
  some questions about "Machine-Learning", "Compute-Science", "Algorithm", "Big-Data", "Security“… and the score is used to evaluate the quality of the answer.

Data format

- Input data

  Task2_data/QA_data.zip

- The data value: (1,100, ,9,Big Data)
- The meaning: PostingType, ID, ParentID, Score, Domains.
  - PostingType:
    - PostingType=1: this post is a question.
    - PostingType=2: this post is an answer.
  - ID: the id for the post
  - ParentId: which question it belongs to.
  - Score: the score of the answer.
  - Domains: which domain does this question belong to.

Requirements

- You need to follow a structure like this.
  - 1.group the questions and answers together
  - 2.computing the highest score
  - 3.design the vectors for clustering from the data
  - 4.clustering
  - 5.some additional parts

- You can see the code framework in Task2_code, you need to fill this code and test it using Spark.
  - Note: keep the function name but you can modify the parameters for this function

Hints

- You should use pair RDDs. It’s something like a map data structure, and it is similar to a format of key-value paris. Comparing to regular RDDs you get a set of powerful functions which you can apply to exactly to pair RDDs. They give a more easier way to operate with data. For example when you want to group or aggregate a data based on some of its properties. For this purposes, Spark has a special set of functions such as: **groupByKey**, **aggregateByKey**, **reduceByKey** etc.
- Small example. If you have an RDD of goods and You want to group these goods by their price. With Spark, you need
  1.create a pair RDD, where a role of key will play a price field, a role of value will play an appropriate good.
  2.apply **groupByKey** function to the pair RDD.

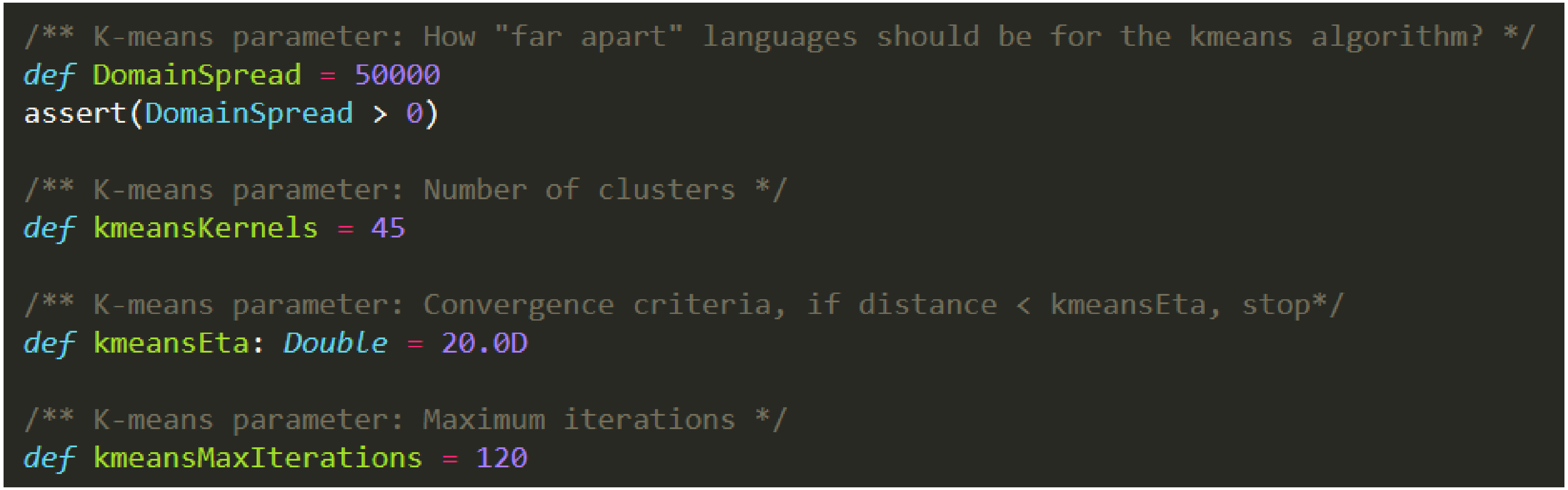
Hints

There are many ways to create a vector from a post for clustering.
In this task, we choose this methods:

For a question from domain A, Score s is the highest score from all its answers. And the index of domain A in the domain list (provided) is x, a predefined parameter DomainSpread is d, the vector for this question is $(d*x, s)$.

Parameters in k-means

- You can see these key parameters in k-means are predefined in the provided code framework.



```
/** K-means parameter: How "far apart" languages should be for the kmeans algorithm? */
def DomainSpread = 50000
assert(DomainSpread > 0)

/** K-means parameter: Number of clusters */
def kmeansKernels = 45

/** K-means parameter: Convergence criteria, if distance < kmeansEta, stop*/
def kmeansEta: Double = 20.0D

/** K-means parameter: Maximum iterations */
def kmeansMaxIterations = 120
```

Clustering Result

- **Include the below output into your lab report:**
  - The cluster centroid for every cluster (the domain).
  - The percentage of the centroid’s domain in its cluster.
  - The size of every cluster.
  - The median score of every cluster.
  - The average score of every cluster.

Report

- In Task2, you need to submit a report (2-3 pages), including:
  - Analyse your result
    - The insight that you can get from the result of clustering for QA_data.
  - Analysis of the parameters (in Slide 13) in k-means
    - how do different parameters impact the performance and clustering results of k-means?
  - Further discussion on the system performance
    - How to improve the efficiency?
    - How to speed up the processing?

Submission requirement

- Deadline: Apr 5, 2019 11:59pm
- Submit the following:
  - Your whole project code without the data (with documentation within the code)
  - Task1: Top-15 output of the result using the data files listed above.
  - Task1 report.(1-2 pages)
  - Task2: the clustering results with predefined parameters. (shown in previous slides)
  - Task2 report.(2-3 pages)

Submission requirement

- Files should be compressed in a zip file to IVLE, with the name [Your Student ID]-Assignment2.zip

Marking Schemes

- Total: 12% of final mark.
  - Task1 Code & Report: 4%
  - Task2 Code & Report: 6%
  - Writing assessment: 2%
    - The written assessment’s questions depend on your submission. You need to understand your code. For example, please explain some specific lines of your code.
    - The written assessment will be conducted in tutorial session.
    - Time: Tutorial Week 12 (“9 (Stream Processing)”).

Notice1

- Please don't consider this homework as the same as ACM-ICPC programming contest (check by exact input-output pairs), we use this to enhance your understanding about the programming using Spark.
- Don't need to worry about whether your result "exactly matches" final result.

Feedbacks are Welcome

- Email me: xuechengxi@u.nus.edu
- Or, post your questions in the IVLE forum (preferred).