Spark Cheatsheet

Contents

[Spark Cheatsheet 1](#_Toc440644472)

[Pair RDDs 1](#_Toc440644473)

[Count the occurrences of a word 1](#_Toc440644474)

[Count the number of unique words 2](#_Toc440644475)

[Get an RDD Containing Just the Values of a PairRDD 2](#_Toc440644476)

[groupByKey() 2](#_Toc440644477)

[Order a PairRDD Based on the Value 2](#_Toc440644478)

[reduceByKey() 2](#_Toc440644479)

[Resilient Distributed Datasets (RDDs) 4](#_Toc440644480)

[Distinct Elements of the Source Dataset (transformation) 4](#_Toc440644481)

[Extract the first n data points from an RDD 4](#_Toc440644482)

[Number of elements in the RDD 4](#_Toc440644483)

[Read a file into an RDD 5](#_Toc440644484)

# Pair RDDs

## Count the occurrences of a word

In:

wordsList = ['cat', 'elephant', 'rat', 'rat', 'cat']

wordsRDD = sc.parallelize(wordsList, 4)

Out:

['cat', 'elephant', 'rat', 'rat', 'cat']

In:

wordPairs = wordsRDD.map(lambda x : (x,1))

Out:

[('cat', 1), ('elephant', 1), ('rat', 1), ('rat', 1), ('cat', 1)]

In:

wordCounts = wordPairs.reduceByKey(lambda a,b : a+b)

Out:

[('rat', 2), ('elephant', 1), ('cat', 2)]

## Count the number of unique words

In:

uniqueWords = len(wordCounts.collect())

print uniqueWords

Out:

3

## Get an RDD Containing Just the Values of a PairRDD

wordCounts = wordPairs.reduceByKey(lambda a,b: a+b)

print wordCounts.collect()

values = wordCounts.values().collect()

print values

Out:

[('rat', 2), ('elephant', 1), ('cat', 2)]

[2, 1, 2]

## groupByKey()

print wordPairs.collect()

[('cat', 1), ('elephant', 1), ('rat', 1), ('rat', 1), ('cat', 1)]

wordsGrouped = wordPairs.groupByKey()

for key, value in wordsGrouped.collect():

print '{0}: {1}'.format(key, list(value))

rat: [1, 1]

elephant: [1]

cat: [1, 1]

## Order a PairRDD Based on the Value

top15WordsAndCounts = (wordCount(shakeWordsRDD).takeOrdered(15, lambda (word, count): -1 \* count))

print '\n'.join(map(lambda (w, c): '{0}: {1}'.format(w, c), top15WordsAndCounts))

## reduceByKey()

>>> rdd = sc.parallelize([(1,2), (3,4), (3,6)])

>>> rdd.reduceByKey(lambda a, b: a + b)

RDD: [(1,2), (3,4), (3,6)] → [(1,2), (3,10)]

# Resilient Distributed Datasets (RDDs)

## Distinct Elements of the Source Dataset (transformation)

uniqueWords = wordsRDD.distinct()

.collect()

print wordsRDD.collect()

print uniqueWords

output:

['cat', 'elephant', 'rat', 'rat', 'cat']

['rat', 'elephant', 'cat']

## Extract the first n data points from an RDD

samplePoints = rawData.take(n)

## Number of elements in the RDD

numPoints = rawData.count()

print numPoints

## Read a file into an RDD

from test\_helper import Test

import os.path

baseDir = os.path.join('data')

inputPath = os.path.join('cs190', 'millionsong.txt')

fileName = os.path.join(baseDir, inputPath)

numPartitions = 2

rawData = sc.textFile(fileName, numPartitions)