Movie Recomendation

Movie recomendation engine

Recomending movies using collaborative filtering.

The dataset is taken from Movie Lense (http://grouplens.org/datasets/movielens/).

```
import java.nio.file.{Paths, Files}
import java.lang.Math
import org.apache.spark.rdd.RDD
import org.apache.spark.mllib.recommendation.ALS
```

```
val numPartitions = 4 // equals # of cores for local processing
val seed = 5
val baseDir = "data"
val dataDir = "ml-latest-small" // change to the path to ml-latest to perfomr

val ratingsFilename = Paths.get(baseDir, dataDir, "ratings.csv")
val moviesFilename = Paths.get(baseDir, dataDir, "movies.csv")

// two data sources
val rawRatings = sc.textFile(ratingsFilename.toString).repartition(numPartitio
val rawMovies = sc.textFile(moviesFilename.toString).repartition(numPartitions
```

```
MapPartitionsRDD[68] at repartition at <console>:81
```

Each line in `rating.csv` is formatted as:

```
userId,movieId,rating,timestamp
```

Each line in `movies.csv` is formatted as:

```
movieId,"title",Genre1|Genre2|...
```

```
import au.com.bytecode.opencsv.CSVParser
import org.apache.spark.mllib.recommendation.Rating

val parser = new CSVParser(',')

val ratingsRDD = rawRatings.filter(x => !x.startsWith("userId"))
  .map(parser.parseLine)
  .map {case Array(userId, movieId, rating, _) => Rating(userId.toInt, movieId
```

MapPartitionsRDD[71] at map at <console>:77

```
type Movie = (Int, String)
val moviesRDD = rawMovies.filter(x => !x.startsWith("movieId"))
  .map(parser.parseLine)
  .map {case Array(movieId, title, _) => (movieId.toInt, title)}
```

MapPartitionsRDD[74] at map at <console>:75

Number of Ratings and Average Ratings for a Movie

List movies by average rating and number of ratings

```
// (userId, movieId, rating) => (movieId, rating)
val movieIdsWithRatingsRDD = ratingsRDD
  .map {case Rating(userId, movieId, rating) => (movieId, rating)}
  .groupByKey()

// calculate average rating and count of ratings
// (movieId, ratings) => (movieId, (numberOfRatings, averageRating))
val movieIdsWithAvgRatingsRDD = movieIdsWithRatingsRDD
  .map {case (movieId, ratings) => (movieId, (ratings.size, ratings.sum / rati
  .cache()
```

MapPartitionsRDD[77] at map at <console>:81

```
// (movieId, (title, (numberOfRatings, averageRating))) => (title, avearageRat
val movieNameWithAvgRatingsRDD = moviesRDD
  .join(movieIdsWithAvgRatingsRDD)
  .map {case (movieId, (title, (numberOfRatings, averageRating))) => (title, a
```

MapPartitionsRDD[81] at map at <console>:81

Movies with Highest Average Ratings and more than 150 reviews

```
val movieLimitedAndSortedByRatingRDD = movieNameWithAvgRatingsRDD
  .filter {case (title, mean, counts) => counts > 150}
  .sortBy( {case (title, mean, counts) => mean + title}, false)

movieLimitedAndSortedByRatingRDD.take(20)
```

20 items

_1	_2	_3
Shawshank Redemption, The (1994)	4.4564516129032254	310
Godfather, The (1972)	4.451030927835052	194
Usual Suspects, The (1995)	4.41703056768559	229
Schindler's List (1993)	4.298283261802575	233
Raiders of the Lost Ark (Indiana Jones and the Raiders of the Lost Ark) (1981)	4.288372093023256	215
Silence of the Lambs, The (1991)	4.250836120401337	299
Matrix, The (1999)	4.238866396761134	247
Fight Club (1999)	4.217171717171717	198
Pulp Fiction (1994)	4.2042042042042045	333
Lord of the Rings: The Return of the King, The (2003)	4.191011235955056	178
Monty Python and the Holy Grail (1975)	4.185185185185185	162
Princess Bride, The (1987)	4.173295454545454	176
Saving Private Ryan (1998)	4.162721893491124	169
Sixth Sense, The (1999)	4.152849740932642	193
Alien (1979)	4.149068322981367	161
Seven (a.k.a. Se7en) (1995)	4.145631067961165	206
Fargo (1996)	4.14390243902439	205
Star Wars: Episode IV - A New Hope (1977)	4.1415094339622645	265
Star Wars: Episode V - The Empire Strikes Back (1980)	4.1307339449541285	218
Lord of the Rings: The Fellowship of the Ring, The (2001)	4.116915422885572	201

Collaborative filtering

```
val Array(trainingRDD, validationRDD) = ratingsRDD.randomSplit(Array(7.0, 3.0)
```

MapPartitionsRDD[89] at randomSplit at <console>:75

```
// Use Root Mean Square Error to assess quality of prediction

def computeRMSE(predictedRDD: RDD[Rating], actualRDD: RDD[Rating]): Double = {
  def sqr(x: Double) = x * x

  val predictedKVRDD = predictedRDD.map(x =>((x.user, x.product), x.rating))
  val actualKVRDD = actualRDD.map(x =>((x.user, x.product), x.rating))

  val squaredErrorsRDD = predictedKVRDD.join(actualKVRDD).map(x => sqr(x._2._1
  val totalError = squaredErrorsRDD.sum()
  val numRatings = squaredErrorsRDD.count()
  Math.sqrt(totalError / numRatings)
}
```

Use ALS to build a model

```
import org.apache.spark.mllib.recommendation.ALS
import org.apache.spark.mllib.recommendation.Rating
val validationForPredictRDD = validationRDD.map(x => (x.user, x.product))
val ranks = List(4, 8, 12)
val iterations = 10
val regularizationParameter = 0.1
val models =  ranks.map(rank => ALS.train(trainingRDD, rank, iterations, regul
```

3 items

rank	userFeatures	productFeatures	formatVersion
4	users MapPartitionsRDD[296] at mapValues at ALS.scala:255	products MapPartitionsRDD[297] at mapValues at ALS.scala:259	1.0
8	users MapPartitionsRDD[503] at mapValues at ALS.scala:255	products MapPartitionsRDD[504] at mapValues at ALS.scala:259	1.0
12	users MapPartitionsRDD[710] at mapValues at ALS.scala:255	products MapPartitionsRDD[711] at mapValues at ALS.scala:259	1.0

```
val errors = models.map(model => computeRMSE(model.predict(validationForPredic
val bestModelIndex = errors.zipWithIndex.minBy(_._1)._2
```

1

Build custom ratings set

```
// Filter used to search for movies by title
moviesRDD.filter {case (movieId, title) => title.toLowerCase.matches(".*godfat
```

```
val myRatedMovies = List(
  Rating(0, 1186, 3.5), // Sex, Lies, and Videotape (1989)
  Rating(0, 27005, 4), //Interview, The (1998)
  Rating(0, 4369, 1), // Fast and the Furious, The (2001)
  Rating(0, 1610, 4.3), //Hunt for Red October, The (1990)
  Rating(0, 44555, 5), // Lives of Others, The (Das leben der Anderen) (2006)
  Rating(0, 59725, 2), // Sex and the City (2008)
  Rating(0, 5504, 1),// Spy Kids 2: The Island of Lost Dreams
  Rating(0, 3868, 2.5), // Naked Gun: From the Files of Police Squad!, The (19
  Rating(0, 1385, 1), // Under Siege (1992)
  Rating(0, 2383, 1.5), //Police Academy 6: City Under Siege (1989)
  Rating(0, 86320, 4), // Melancholia (2011)
  Rating(0, 7371, 4.7), // Dogville
  Rating(0, 125916, 2), // Fifty Shades of Grey (2015)
  Rating(0, 1101, 3.4), // Top Gun
  Rating(0, 5065, 4.8), // Mothman Prophecies, The (2002)
  Rating(0, 94469, 3), // Red Dog (2011)
  Rating(0, 6539, 3), // Pirates of the Caribbean: The Curse of the Black Pear
  Rating(0, 593, 5), // Silence of the Lambs, The (1991)
  Rating(0, 6711, 3.8),// Lost in Translation (2003)
  Rating(0, 1287, 2.5), // Ben-Hur (1959)
  Rating(0, 2022, 4), //Last Temptation of Christ, The (1988)
  Rating(0, 1979, 1.5), //Friday the 13th Part VI: Jason Lives (1986)
  Rating(0, 307, 4), // Three Colors: Blue (Trois couleurs: Bleu)
  Rating(0, 64614, 4.2), // Gran Torino (2008)
  Rating(0, 2700, 4.7), //South Park: Bigger, Longer and Uncut (1999)
  Rating(0, 112552, 5), // Whiplash
  Rating(0, 112183, 3.5), // The Birdman
  Rating(0, 109374, 3.7), // Grand Budapest Hotel, The (2014)
  Rating(0, 34437, 4.1), // Broken Flowers (2005)
  Rating(0, 48997 , 3.9), //Perfume: The Story of a Murderer (2006)
  Rating(0, 2467, 4.9), // Name of the Rose, The (Name der Rose, Der) (1986)
  Rating(0, 1214, 3.8), // Alien
  Rating(0, 924, 4.2), // 2001: A Space Odyssey (1968)
  Rating(9, 858, 4.5) // Godfather, The (1972)
  )
```

34 items (Out of 34 items, only the 25 first items are shown)

user	product	rating
0	1186	3.5
0	27005	4
0	4369	1
0	1610	4.3
0	44555	5
0	59725	2
0	5504	1
0	3868	2.5

0	1385	1
0	2383	1.5
0	86320	4
0	7371	4.7
0	125916	2
0	1101	3.4
0	5065	4.8
0	94469	3
0	6539	3
0	593	5
0	6711	3.8
0	1287	2.5
0	2022	4
0	1979	1.5
0	307	4
0	64614	4.2
0	2700	4.7

```
val myRatingsRDD = sc.parallelize(myRatedMovies)
val trainingWithMyRatingsRDD = trainingRDD.union(myRatingsRDD)
val myRatingsModel = ALS.train(trainingWithMyRatingsRDD, ranks(bestModelIndex)
```

org.apache.spark.mllib.recommendation.MatrixFactorizationModel@7d47c7cb

```
val myUserId = 0
val myRatedMoviesIds = myRatedMovies.map {case Rating(_, movieId, _) => movieI

// remove all movies reated by me
val myUnratedMoviesRDD = moviesRDD
                         .filter {case (movieId, _) => !myRatedMoviesIds.cont
                         .map {case (movieId, _) => (myUserId, movieId)}
```

MapPartitionsRDD[965] at map at <console>:82

Predicting my movies

```
val predictedRatingsRDD = myRatingsModel.predict(myUnratedMoviesRDD)

val movieCountsRDD = movieIdsWithAvgRatingsRDD.map {case (movieId, (num, avg))


val predictedRDD = predictedRatingsRDD.map(r => (r.product, r.rating))
val predictedWithCountsRDD = predictedRDD.join(movieCountsRDD)
val ratingsWithNamesRDD = predictedWithCountsRDD
     .filter {case (movieId, (predictedRating, ratingNum)) => ratingNum > 75 }
     .join(moviesRDD)
     .map {case (movieId, ((predictedRating, ratingNum), movieName)) => (predi
```

MapPartitionsRDD[983] at map at <console>:108

Highest rated recommended movies

```
ratingsWithNamesRDD.takeOrdered(20)(Ordering[Double].reverse.on(x => x._1))
```

20 items

_1	_2	_3
4.745542240921167	Fight Club (1999)	198
4.675104880292487	Apocalypse Now (1979)	110
4.5555228697982475	Reservoir Dogs (1992)	134
4.514047834630157	Pulp Fiction (1994)	333
4.4778325640119965	Trainspotting (1996)	122
4.420754617635426	L.A. Confidential (1997)	134
4.398769243745667	Annie Hall (1977)	87
4.366807741571392	Usual Suspects, The (1995)	229
4.342163370057376	Twelve Monkeys (a.k.a. 12 Monkeys) (1995)	220
4.3089818746054735	Gattaca (1997)	77
4.30737046794528	Memento (2000)	146
4.24903144867841	English Patient, The (1996)	82
4.243669497468986	Amelie (Fabuleux destin d'Amélie Poulain, Le) (2001)	116
4.236140394460174	Snatch (2000)	83
4.224360339171227	Clockwork Orange, A (1971)	129
4.205553407369345	Léon: The Professional (a.k.a. The Professional) (Léon) (1994)	117
4.205011348511712	Lord of the Rings: The Return of the King, The (2003)	178
4.201071892098194	Matrix, The (1999)	247
4.1997853367927815	Citizen Kane (1941)	88
4.198305073726201	Seven (a.k.a. Se7en) (1995)	206

Lowes rated recommended movies

```
val predictedLowestRatedMovies = ratingsWithNamesRDD.takeOrdered(20)(Ordering[
```

20 items

_1	_2	_3
2.0368344497190054	Mission: Impossible II (2000)	77
2.2853475075017524	Armageddon (1998)	112
2.2991693682040184	American Pie (1999)	123
2.3006806557940385	Congo (1995)	76
2.307865095407452	Batman Forever (1995)	164
2.33551840574202	Charlie's Angels (2000)	82
2.3356079533200775	Nutty Professor, The (1996)	96
2.3464275387861324	Eraser (1996)	84
2.3599282837178217	Mummy, The (1999)	96
2.384096065046156	Clueless (1995)	135
2.413329336817654	Honey, I Shrunk the Kids (1989)	79
2.422342692348537	Ace Ventura: Pet Detective (1994)	194
2.424114875365447	Dumb & Dumber (Dumb and Dumber) (1994)	151
2.4552872322492703	Cliffhanger (1993)	121
2.5045809683265086	Net, The (1995)	102
2.5739891764760783	Home Alone (1990)	148
2.6135406062446624	Ace Ventura: When Nature Calls (1995)	108
2.6156849632623773	Mrs. Doubtfire (1993)	162
2.6179638758871313	Mask, The (1994)	159
2.6252503215956478	Pretty Woman (1990)	175

Incomplete (hint: check the parenthesis)