PySpark and Matrix Completion (Netflix Challenge) with MLlib

Vittorio Bisin

Python in Data Science

1. High level syntax and easy to learn
2. Interfaces easily with C/C++ allowing for high performance libraries
3. Extensive library of packages
    A. numpy/scipy/matplotlib
        - Typical MATLAB functionality: fast array operations, scientific functions, plotting
    B. pandas
        - similar to R's data.frame
    C. scikit-learn/statsmodels
        - Implementation of machine learning algorithms
    D. nltk
        - natural language processing

PySpark Architecture

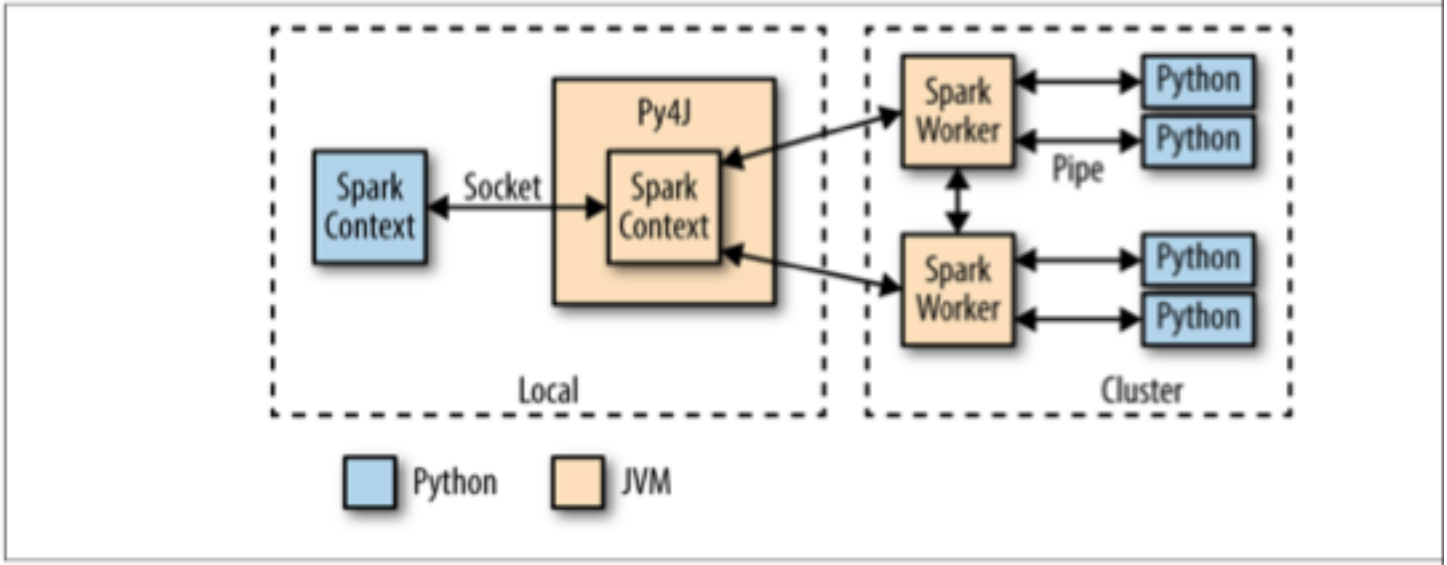


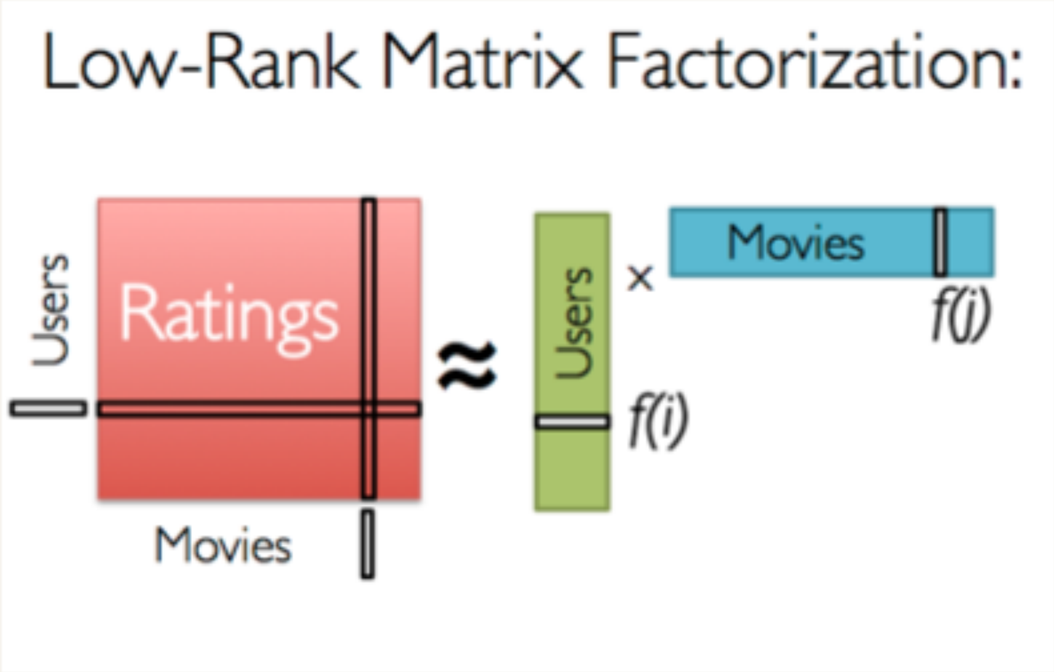
Figure 11-1. PySpark internal architecture

Figure and notes taken from Advanced Analytics with Spark: chapter 11

1. When PySpark's Python interpreter starts it also starts a JVM, and communicates with it using a socket.
2. The JVM is used as the Spark driver, loading a JavaSparkContext that communicates with the Spark executors across the cluster.
3. The Python API calls to the SparkContext object are translated into Java API calls to the JavaSparkContext
   A. A Python RDD in the local PySpark client corresponds to a Python RDD object in the local JVM

Low Rank Matrix Factorization

1. Approximate high rank matrix with a low dimensional one (e.g. SVD)
   A. Becomes necessary with training sets composed of 10 billion entries (see Netflix challenge on the next slide)
2. Normally used as a recommender system - where a user has a sparse representation and we want to predict its unknown entries



The Netflix Challenge

1. Collaborative filtering and matrix completion with user rating data.
   A. $1 million prize
2. Very sparse matrix since each user has not rated every possible movie.
   B. 1% of matrix entries nonzero
   C. Adapts well to low rank matrix factorization

	Dirty Dancing	Meet the Parents	Top Gun	The Sixth Sense	Catch Me If You Can	The Royal Tenenbaums	Con Air	Big Fish	The Matrix	A Few Good Men
Customer 1	·	·	·	·	4	·	·	·	·	·
Customer 2	·	·	3	·	·	·	3	·	·	3
Customer 3	·	2	·	4	·	·	·	·	2	·
Customer 4	3	·	·	·	·	·	·	·	·	·
Customer 5	5	5	·	·	4	·	·	·	·	·
Customer 6	·	·	·	·	·	2	4	·	·	·
Customer 7	·	·	5	·	·	·	·	3	·	·
Customer 8	·	·	·	·	·	2	·	·	·	3
Customer 9	3	·	·	·	5	·	·	5	·	·
Customer 10	·	·	·	·	·	·	·	·	·	·

Methods 1

1. Original Matrix is NxM (N users and M movies)
2. Fix a K<<N,M
   A. K is the rank of the low dimensional approximation and the number of "concepts"
3. Let each user u be summarized as a k dimensional vector, each movie i by a k dimensional vector
4. Objective function is derived from minimizing the least squares error of the rating (LHS) and regularization (RHS)
   B. Related to Tikhonov regularization/Ridge Regression

$$X = \begin{bmatrix} | & & | \\ x_1 & \cdots & x_n \\ | & & | \end{bmatrix}, Y = \begin{bmatrix} | & & | \\ y_1 & \cdots & y_m \\ | & & | \end{bmatrix}$$

$$\min_{X,Y} \sum_{r_{ui}\ observed} (r_{ui} - x_u^\intercal y_i)^2 + \lambda\left(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2\right)$$

Equations taken from "Distributed Algorithms and Optimization" Stanford Spring 2015 lecture slides

$x_{1,1}$ =how much user 1 likes topic 1 (e.g. action movies)
$y_{1,1}$ =how much movie 1 is about topic 1 (e.g. how much action it contains)

Methods 2: Alternating Least Squares

"Coordinate Descent for matrices"

$$\min_{X,Y} \sum_{r_{ui}\ observed} (r_{ui} - x_u^\intercal y_i)^2 + \lambda(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2)$$

1. Unfortunately the objective function is not convex
   A. The LHS is the problem
2. However it is convex w.r.t. X and Y separately
3. ALS -> minimize the objective function w.r.t X and Y separately, and repeat until convergence
4. Easily parallelizable (great with Spark)

My Demo

1. Used the MovieLens dataset
2. The best low dimensional matrix approximation was rank 4
3. Using the test data, the root mean squared error was 0.97
4. I used the model to predict my own movie recommendations

My Ratings:
Toy Story (1995) : 5
Casino (1995): 6
Ace Ventura: When Nature Calls (1995) 1
Babe (1995): 1
Il Postino (1994): 9
Taxi Driver (1976): 10
Show Girls (1995): 6
Clerks (1994): 8
Start Wars (1997): 6
Four Weddings and a Funeral (1994): 7
The Flinstones (1994): 3
Timecop (1994): 2
Pulp Fiction (1994): 10
The Godfather (1972): 10
The Usual Suspects (1995): 7

Results

1. Algorithm predicts my ratings for all movies in the dataset
2. Then recommends 25 movies with the highest predicted rating
   A. Only selecting from movies that have been rated by other users a minimum of 25 times

Recommendations:

Sex: 9.6
Three Colors: Red (1994): 9.5
Brokeback Mountain (2005): 9.4
Manhattan (1979): 9.4
Network (1976): 9.3
Annie Hall (1977): 9.2
Chinatown (1974): 9.2
Deer Hunter (1978): 9.2
Adaption (2002): 9.1
Citizen Kane (1941): 9.1
Vertigo (1958): 9.1
2001: A Space Odyssey (1968): 9
Bound (1996): 9
Apocalypse Now (1979): 9
Midnight Cowboy (1969): 9
Harold and Maude (1971): 8.9
Rosemary's Baby (1968): 8.8
Dr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb (1964): 8.8
Psycho (1960): 8.8
All About Eve (1950): 8.8
Third Man: 8.8
Graduate: (8.8)
Fargo (1996): 8.8
Three Colors: Blue (1993): 8.7
Moonstruck (1987): 8.7