DATA 643: Discussion 2

Logan Thomson

6/21/2017

CLUSTER COMPUTING, LINEAR ALGEBRA, AND MUSIC RECOMMENDATIONS

For this discussion item, please watch the following talk and summarize what you found to be the most important or interesting points. The first half will cover some of the mathematical techniques covered in this unit's reading and the second half some of the data management challenges in an industrial-scale recommendation system.

SPOTIFY BACKGROUND

Figure 1:

Spotify is one of the major players in the streaming content arena, focusing solely on music, and not streaming video. Founded in 2005, Spotify grew quickly after coming out of Beta in 2008, releasing app versions of the service in 2009, and then finally launching in the United States in 2011.

From 2012 to 2017, Spotify grew its active user base nearly ten times the size, from 15 million users to 140 million. With this type of growth, and the success of their business model hinging on providing good recommendations to constantly shifting tastes and new content, creating a system that works and scales quickly is crucial.

DISTRIBUTED COMPUTING MEETS LINEAR ALGEBRA

At the time of Chris Johnson's talk at Spark Summit 2014, he reported Spotify had 40 million users, and the main theme of the presentation was reducing I/O overhead and processing times while working on a massive amount of data. This was done not just by changing the data used, the algorithm used to process the data, or the system used, but a combination of all three. For the data itself, rather than use explicit ratings (thumbs up/down, 1-5 scale), Spotify utilizes implicit ratings based on streams (1 = played, 0 = not streamed).

For the algorithm, the matrix factorization technique of Alternating Least Squares is used. Since matrix factorization takes the original matrix (R) and turns it into two vectors (U and P) that, when multiplied together approximate the original ($U \times P = R$). By fixing one of the vectors and solving (like linear regression) for the other, then alternating until convergence (little or no change in the two vectors).

Lastly, for the system, Spotify is using Hadoop MapReduce, but in Chris Johnson's talk, was demonstrating the difference between Hadoop, and the reduction in processing time using Spark (which runs on the Hadoop framework, it's not an entirely different system), which processes everything in memory (and can even use the

disk if everything doesn't fit into memory). However, at the the presentation, Spotify had not implemented Spark due to issues of using more than 10% of its user base in testing.

ALTOGETHER NOW

Putting all three points together, Spotify experimented with different ways of using the simplified matrix of users and streams, alternating least squares method of matrix factorization, and distributed computer processing together to optimize processing times. Of the three methods (broadcast everything, full-, and half-"gridify"), the half-gridify method showed an 85% reduction over using the regular method on Hadoop MadReduce. This half-gridigy method takes the rows (users) and colums (streams) of the matrix, partitions them into blocks (sub-matrices), and then broadcasts to each worker in Spark for solving.

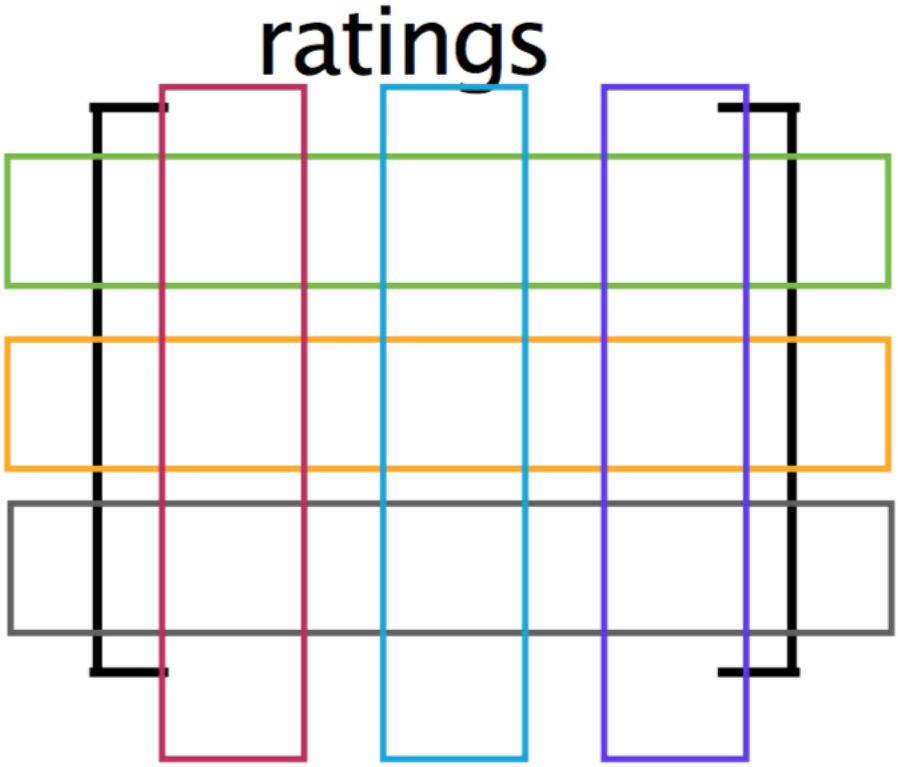

Figure 2: Partitioning the matrix with half-gridify

CONCLUSION

What I found to be the most fascinating about all of this is how the concepts of the separate but very-related fields of linear algebra and computer science play a role in optimizing speed and scalability of a system. Without the knowledge of how matrix factorization can be incorporated into distributed computing, the advancements that have been made in recommendation systems never would have happened. In other words, it's not just about the code or the speed of the machine. Also, its refreshing to get an inside look at how complex creating something like Spotify playlists is, and that even with advanced data and computer scientists, there's still quite a bit of experimentation and trial and error.