DATA 643: Recommender Systems

Discussion 2: Spark & Spotify

Walt Wells, Summer 2017

Overview

Christopher Johnson talks about Spotify's recommendation methods at the 2014 Spark Summit.

Of Interest

Explicit Matrix Factorization - helpful to see an example and formula for managing a collaborative filtering recommender based on rank by creating smaller, lower rank matrices.

Implicit Matrix Factorization - even more helpful to think of these as a binary that uses ALS for factorization.

Early Hadoop I recognize those 2009 Super Micro servers ;) IO issues from duplicative read / write effort. Overburdening workers.

Spark Their half-gridify attempt is hardware intensive. Probably can't use commodity hardware since disk intensive on each worker.

Questions

1) If a user has been streaming from Spotify for a long period of time, and possibly isn't even actively engaging with Spotify but is still streaming, why count a song played as a 'listen'. Is there some time duration cutoff for streaming services? Eg - if a user goes > 30 minutes without actively interacting with the system (swipes, button presses, etc), stop counting or at least minimize the weight of any user signal.
2) If they gridify their utility matrix and distribute sections to the workers, do they then make sure that user signal goes to the same workers?

References

- https://www.youtube.com/watch?v=3LBgiFch4_g