Spotify's Road to Personalizing User Experience

DATA612 - Recommender Systems

William Outcault

18 June 2020

Overview

In 2014 Christopher Johnson, now Director of Data Science at Indeed.com, briefed his audience on Spotify's approach towards building an industrial scale recommendation system. He begins by touching upon the main motifs for their recommendation system which include features; Discover, Radio, Related Artists and Now Playing. Systems from companies such as Songza, Dre Beats, Pandora, echonest and last.fm are brought up to help generalize the different approaches for music recommendation systems. He mentions Spotify's focus on collaborative filters and text analysis before delving into the ongoing problem of wrestling very large datasets.

Minimizing Cost

He does not go into their approach for text analysis, however considering the hoops they jumped through for collaborative filtering, the text analysis would cause headaches to say the least. Their collaborative filtering method, unlike Netflix, is based off implicit matrix factorization (Netflix uses explicit matrix factorization). They use binary labels to rank each user/item as either listened to or never streamed. They use Alternating Least Squares to minimize cost, this is a function of parameters such as; user bias, item bias, regularization parameter, user latent factor, item latent factor and lastly user/item label value.

Building to Scale

At Spotify the dataset is estimated to be 9x the size of Netflix's dataset. This puts into perspective the size of the challenge they have to face when conducting their own recommendation methods. They took three different approaches towards handling this cumbersome dataset. They begin by using Hadoop, then 'Full Gridify', then 'Half Gridify'. Chris mentions that he coins this phrase 'Full Gridify', which in short, partitions data into user/items blocks such that only required user ratings are taken into acount. They run into this problem where they find themselves shuffling significants amount of data around when grouping by user. To combat this they took a 'Half Gridify' approach. They use the MLlib package within Spark to utilize this method; they partition ratings matrix into a subset of users and all ratings from a given user. By their third attempt they had reduced the total run time from 10 hours, to 3.5 hours to 1.5 hours.

Conclusion

Keep in mind this was back in 2014. Just like Christopher Johnson finding a new job, I am sure Spotify has found a new model. Due to Moore's Law their processing time has almost certainly reduced by a significant amount.

P.S. To use as reference, Christopher Johnson also worked at Amazon between his time at Spotify and his current role with Indeed. Considering all this time that has passed, how much different is their current model?