Spotify's Collaborative Filtering

By: Soren Nelson

Introduction

There is a reason that Spotify is still around after three of the biggest companies in the world: Apple, Google, and Amazon; have all stepped into the music streaming industry. They are able to recommend better music to me than any of the other three. At first, I thought I was biased in believing this, as I am a loyal Spotify user. However, in doing research, I have come to find that Spotify does, in fact, have a much better system than the other three.

Spotify uses three styles of models to recommend music. The first is Natural Language Processing models. These compare songs based on words used to describe the songs via things such as articles on the web. The next is used to recommend songs that aren't as popular. These Content Based models look at the actual audio and use similarities to recommend similar songs to you. The last style is known as Collaborative Filtering. Essentially what it is doing is creating a user vector for every user and a song vector for every song. It then compares them to recommend music that is similar to each other and music that similar users listen to. Throughout this paper, I will discuss Collaborative Filtering as it is by far the most popular and is at the heart of Spotify's most popular music recommendation method: Discover Weekly. My focus will be on this algorithm and more specifically how linear algebra is used to power it.

It must be noted that a similar approach to Spotify's model is used by other companies such as Last.fm and another has been popularized by the Netflix Prize. I will solely be exploring Spotify's Collaborative Filtering method which differs from Netflix in that it uses implicit feedback as opposed to Netflix's more explicit feedback approach using ratings and other explicit user data. It must also be said that this is my interpretation of Spotify's Collaborative Filtering based on my research and is most likely not exact in its description.

Collaborative Filtering

Discover Weekly is a playlist made by Spotify for every one of their 140 million users on a weekly basis. For every user, they sift through over forty million songs to find the songs most likely to be liked by you that you don't already have in your music library. Collaborative Filtering tries to emulate your friend telling you about a certain song x because they found out that you like song y. The way this is done is through a latent factor model, which is a machine learning algorithm used to turn raw data into latent (not directly observable) features. The latent factors are computed using an alternating-least-squares process to minimize a cost function. Alternating-least-squares is the process, in our case, of switching back and forth between computing the user-factors and the song-factors, this process continually converges the cost function.

A cost function is used to judge how far off a given estimation is from what the actual value is. You plug a certain set of features into the equation. You then tweak these features via our alternating-least-squares process until the cost function converges. When it does, in our case, it will leave us with a preferences-confidence pair for the users which shows a preference and how sure we are of that preference.

Cost Function

The cost function Spotify uses is below along with its explanation.

$$min(x,y)\sum_{u,i}^{k} c_{ui} (p_{ui} - x_{u}^{T}y_{i})^{2} + \lambda \left(\sum_{u}^{n} ||x_{u}||^{2} + \sum_{i}^{m} ||y_{i}||^{2}\right)$$

To start we have two matrices comprised of user and item (song) vectors. ${}^{x}{}_{u}$ is a user vector and ${}^{y}{}_{i}$ is a song vector. To explain the preference ${}^{p}{}_{ui}$ and confidence ${}^{c}{}_{ui}$ variables, we must first introduce our raw data variable ${}^{r}{}_{ui}$. This is the number of times a user u has listened to a song i. ${}^{p}{}_{ui}$ is our preference. The preference is a binary variable where:

$$p_{ui} = \begin{cases} 1 & r_{ui} > 0\\ 0 & r_{ui} = 0 \end{cases}$$

In other words, if a song has been listened to then it has a preference of one, otherwise, it has no preference.

The preference is not enough. There are many reasons why a user may not have listened to a particular song other than that person not liking the song and many reasons why they may have listened to a song other than liking it. This is why we introduce our confidence variable. Once again, there are multiple ways to approach how confident you are of a preference. We will choose to use:

$$c_{ui} = 1 + \alpha r_{ui}$$

With this, we can have a greater weight given to songs that have been listened to more than once. The constant lpha is the rate of increase. I am not entirely sure of what constant Spotify

uses here, but in my research, I have found around 40 to be reliable. The ladder half of the cost function:

$$\lambda \left(\sum_{u}^{n} \| x_{u} \|^{2} + \sum_{i}^{m} \| y_{i} \|^{2} \right)$$

is used to regularize it and in doing so keep it from overfitting. This means that we will use it to make sure that the cost function converges.

Computing Features

Now that we understand the cost function, we need to figure out how to use the alternating-least-squares process to minimize it. We start by initializing x_u and y_i . We then begin computing the user vector by:

$$x_{u} = (Y^{T}C^{u}Y + \lambda I)^{-1}Y^{T}C^{u}p(u)$$

This is followed by computing the item vector with:

$$y_i = (X^T C^i X + \lambda I)^{-1} X^T C^i p(i)$$

We alternate plugging each of these new values into the cost function until it converges. The eigenvalue is data dependent.

Once we have minimized the cost function we can use the preference-confidence pair to rank the songs. Once again, there are multiple ways to do this. One way would be to focus on the more popular songs, as there is a reason they are popular. Another would be to focus on ones in a certain genre so the whole playlist is more similar. Lastly, you could merely order the songs by rank and add to the playlist the top thirty. You could then recommend different songs weekly by remembering which songs have been previously recommended. This was something that didn't come up much in my research as I imagine Spotify keeps it under wraps.

Issues and Optimizations

There are issues with every recommendation system. The main issue with Collaborative Filtering is what is known as a cold start problem. If an unknown artist uploads a song, you might like the song if you heard it but since it hasn't been listened to by many people, it is most likely not going to be recommended to you. Another issue is that with implicit data there is no way to know if a user dislikes a track, only if they do like one. This can pose issues where you are recommended a song you dislike that other users like because the model can't differentiate between a song you merely haven't listened to and a song that you dislike.

I will not discuss optimizations of Collaborative Filtering in depth as I feel it is out of the scope of this paper. I will note that there are optimizations to make in terms of speed and quality. I explain a couple here. Instead of the raw data being the number of times a user has listened to a full song, Spotify reduces the length of time to a shorter time. This is because you may skip the end of a song that you still like. They also use a logarithmic scale on the initial preference matrix to try to reduce the cold start problem as much as possible.

Conclusion

To overview what was discussed in this paper. I believe Discover Weekly is the best recommendation system that any music streaming service has to offer. It is created by taking matrices of all user-song pairs. The playlist uses Collaborative Filtering to turn the pairs into preference-confidence pairs to rank the songs for each user according to factors such as what other similar users listen to and what other songs are similar to the ones they listen to.

In this paper I discussed how linear algebra is used to power a machine learning cost function and how implementing an alternating-least-squares model, you can minimize the cost function to find the perfect playlist. The only way such a playlist could be made for 140 million users on a service with 40 million songs is through Linear Algebra.

Bibliography

Hu, Yifan, et al. "Collaborative Filtering for Implicit Feedback Datasets."

Madathil, Mithun. Music Recommendation System Spotify - Collaborative Filtering.