Cosine Similarity, Collaborative Filtering and Creed:

How Maths Can Read Your Mind

By Martha Smith

Why did the Spotify DJ make me listen to One Last Breath by Creed 8 times in one day? Is it a coincidence or are my mental breakdowns being tracked by Spotify's algorithm? The answer, disconcertingly may be yes, they are.

Spotify bases its song recommendations on a plethora of factors and algorithms, some simpler than others. You listened to this artist. You will probably like another song by the same artist. The more complex maths and algorithms take control for everything beyond that.

Spotify may not have diagnosed my mental state necessarily – but it did detect patterns in my listening behaviour and use mathematics to make some oddly specific recommendations that I was almost guaranteed to relate to.

One of the simplest ways content is recommended is using collaborative filtering, like almost every other social media platform. Thousands of users all listening to the same music, with this volume of data it's almost guaranteed to find similar users.

Figure 1:

Listened to all the way through	X
Favourited	Y
Skipped	Z

Song	1	2	3	4	5	6	7	8	9	10	11	12	13
User 1		X	Y		X	Z	Y	Y			Y	Z	
User 2	Y	X			Y	Z			Z	X	X		

For example, **Figure 1** shows users with a somewhat similar preference for songs. User 1 and User 2 both listened fully to Song 2 and skipped Song 6. Obviously, this is an extremely limited set of data, but it shows the broad concept.

As User 2 skipped Song 9, we could assume that User 1 would also not be a fan. The same goes for Song 12. We could also assume that User 1 would like User 2's favourited Song 1. Out of the 11 songs the users both interacted with, they behaved the same way on 2 of them, giving them a similarity score of 0.18 (2/11).

With more data about different users' preferences, it can tailor well to different users' taste profiles. Even though more data would make the predictions better, the

scalability of an algorithm like this is limited and even with the most efficient code and GPUs it would be expensive to run. The results of this data inspection may not even yield results; many users do not rate or interact with their music. Users sometimes listen to music in the background and therefore don't skip songs they don't like, these users can skew the data.

Furthermore, the latest music from smaller artists has essentially no data about users listening to this music and the algorithm would instead constantly recommend the same mainstream music. Niche artists and listeners would be deprived if this method of recommendation was solely implemented, the grey sheep who don't fit into any of the data clusters wouldn't be given accurate recommendations.

Overall, Collaborative Filtering is best suited to mainstream users who interact with their recommendations by skipping songs they don't like and liking and adding to playlists the ones that they do are best accommodated for with this method. There is more data available and therefore more accurate results.

Many of the features used in song vectors are:

1. Danceability
2. Energy
3. Tempo
4. Valence
5. Acousticness/Instrumentation

These factors are mapped to vectors in a multidimensional space; cosine similarity measures the cosine of the angle between two song vectors. The closer the value is to 1, the higher the similarity of the songs.

Cosine similarity focuses on the direction of the vector, sound profile, as opposed to the Euclidean distance, which focuses more on the magnitude and intensity. This makes it more effective for finding similar sounding music as opposed to similarly produced music, therefore creating a wider range of suggestions.

For example, if Song A has vector (8,6,1) and Song B has vector (7,9,3) for Tempo, Valence, and Energy, the cosine similarity could be calculated in the following equation:

$$\frac{8 \cdot 7 + 6 \cdot 9 + 1 \cdot 3}{\sqrt{8^2 + 6^2 + 1^2}\sqrt{7^2 + 9^2 + 3^2}} = 0.953696 \ldots$$

As this value is close to 1, this means they are like each other whereas if we compare Song A to Song C (2,1,7), which has an incredibly low cosine similarity which is not close to 1.

$$\frac{8 \cdot 2 + 6 \cdot 1 + 1 \cdot 7}{\sqrt{8^2 + 6^2 + 1^2}\sqrt{2^2 + 1^2 + 7^2}} = 0.3926814 \ldots$$

This shows that Song C points in a vastly different direction to Song A in the multidimensional plane. If the user enjoys Song A, Song B will be recommended.

These vectors serve more of a purpose than just finding songs that are like each other though. Further matrices are built based on user data, such as user signals, play counts, liked songs and skips, rather than an explicit rating system which Spotify lacks. These song vectors and user vectors are compared to each other, meaning not only can similar songs be linked but also users' taste profiles can be linked to songs.

With the millions of data points from every user, even when individuals lack data in a certain area, it's viable to use users' taste profiles to make accurate predictions. The cosine similarity can be used to make these predictions about how a user will rate items. Using the millions of other users to create a weighted average based on how similar the taste profile of the user is, in comparison to all the other users contributing to the prediction with their data.

Spotify also uses something called natural language processing – NLP. Although not understanding emotions, it considers the lyrics of a song by detecting patterns in the text. For example, when I placed One Last Breath by Creed into my playlist titled "#icantbreathe", this established a link for this song and if other users placed this same song into similarly titled playlists, the algorithm learns that this song has a certain context associated with it.

So, no – Spotify isn't tracking my breakdowns, but it is tracking patterns in not only my behaviour but also millions of others which can make its recommendation of a classic mental breakdown song disconcertingly accurate.