Non-Negative Matrix Factorization for Clustering Spotify Songs

Irsa Ashraf **Dharini Ramaswamy** **Kelley Sarussi**

1 Introduction

We apply non-negative matrix factorization to Spotify song data for cluster analysis. We are interested in this topic because content-based recommendation systems rely on a basis of songs that can describe a user's taste profile; for example, Spotify's Discover Weekly feature creates a curated playlist of songs that are similar to the genres and preferences of a specific user.

We used a Spotify "tracks" dataset composed of approximately 600,000 Spotify songs and 19 corresponding audio features, as well as artist name. The audio features include id_songs, name_song, popularity_songs, duration_ms, explicit, id_artists, release_date, danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo, time_signature, followers, genres, name_artists, popularity_artists. The "tracks" dataset is used as our "X" matrix. X is a 12 x 89,609 matrix of audio features and songs, reduced from 19 x 600,000 after pre-processing, audio feature selection, and filtering some songs out. Each row n represents an audio feature and each column p represents a song.

We utilize an additional "artists" dataset for the purpose of assigning meaning to the values in our W and H matrices later on. Since we do not have a 1:1 match between genre and song in the tracks dataset, we use the "artists" dataset to map song to artist to genre, since the artists dataset includes a "genre"field. We want to make sure that each song can only have one genre from the "artists" dataset. Many artists had multiple genres assigned to them, for example, Taylor Swift is classified as both 'country' and 'pop'. Filtering our data to songs with one genre per artist made interpreting and comparing our results to Spotify's labeling of songs easier. Once we filtered the data to only songs with an artist with one genre label, we ended up with 89,609 songs in our X matrix.

2 Literature Review

NMF is commonly used for analyzing and extracting meaningful features from a non-negative dataset. Lee and Seung (2000) show how to "factorize a data matrix subject to different constraints." Other work has been done to apply non-negative matrix factorization to facial images, by learning "a part-based representation of faces" to find an approximate factorization (Lee and Seung 1999). In Valdez et al, NMF is used to factorize a user-matrix in order to find commonalities between users and items in the health field and to improve efficiency of calculating recommendations. Another example of an application of NMF includes applying it as a clustering approach to environmental research in public health, as shown in Fogel et al. Our application of NMF to Spotify song data incorporates a mix of the use of NMF in recommendation systems and clustering in order to group similar songs into common genres that can potentially be used as part of a recommendation system.

3 Parameters

The parameters of our model consisted of the value of k, number of songs to include (since not all songs are necessary to represent a basis for each genre and our X matrix is unlikely to be full rank), and which audio features to include.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

K represents the number of clusters to group songs into, or the number of genres we think are necessary to group the songs into best. The best k can be determined a few different ways: through estimation using the SVD, use of "experts insight", and trial and error (Gillis 2014). Through trial and error of testing multiple values of K less than P, we ended up choosing k = 10 for our final model, meaning that we would be grouping our songs into ten clusters.

We included audio features that had continuous values (duration, valence, tempo, etc.) and omitted features with large whole numbers (artist popularity, number of followers). Features with larger whole numbers skewed our clustering results. We also considered the context of genre classification; for example, the number of followers or popularity of an artist would not necessarily impact the type of genre that a song is classified as. Since loudness included negative values, we took the absolute value of it before including it in the model. The final set of features we chose is reflected in Table 1. Our final X matrix had dimensions n x p: 12 x 89,609. Table 1 shows the first 10 columns and all rows of matrix X below:

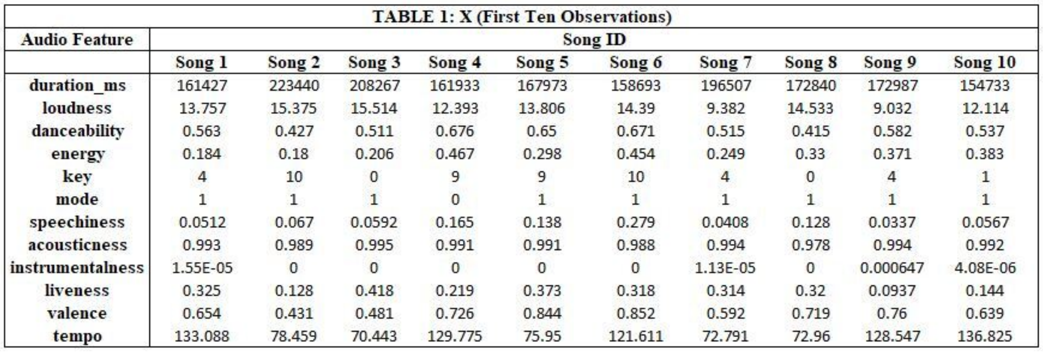


TABLE 1: X (First Ten Observations)

Audio Feature	Song ID									
	Song 1	Song 2	Song 3	Song 4	Song 5	Song 6	Song 7	Song 8	Song 9	Song 10
duration_ms	161427	223440	208267	161933	167973	158693	196507	172840	172987	154733
loudness	13.757	15.375	15.514	12.393	13.806	14.39	9.382	14.533	9.032	12.114
danceability	0.563	0.427	0.511	0.676	0.65	0.671	0.515	0.415	0.582	0.537
energy	0.184	0.18	0.206	0.467	0.298	0.454	0.249	0.33	0.371	0.383
key	4	10	0	9	9	10	4	0	4	1
mode	1	1	1	0	1	1	1	1	1	1
speechiness	0.0512	0.067	0.0592	0.165	0.138	0.279	0.0408	0.128	0.0337	0.0567
acousticness	0.993	0.989	0.995	0.991	0.991	0.988	0.994	0.978	0.994	0.992
instrumentalness	1.55E-05	0	0	0	0	0	1.13E-05	0	0.000647	4.08E-06
liveness	0.325	0.128	0.418	0.219	0.373	0.318	0.314	0.32	0.0937	0.144
valence	0.654	0.431	0.481	0.726	0.844	0.852	0.592	0.719	0.76	0.639
tempo	133.088	78.459	70.443	129.775	75.95	121.611	72.791	72.96	128.547	136.825

Figure 1: Non-negative matrix factorization of our original data to represent songs clustered into genres.

4 Model and Error

Our objective with using NMF on this dataset is to decompose it into two matrices, W and H. Our code implementation of NMF consisted of two auxiliary functions that updated the H and W matrices, and a main function that ran the NMF algorithm. The function begins by randomly initializing the W and H matrices using the shape of the input matrix. In each iteration, we updated the W and H matrices by calling their respective auxiliary functions. They implement the multiplicative update rule described by Lee and Seung, where each update utilizes multiplication by a factor (Lee and Seung, 1999). After updating the H and W matrices, we calculated the loss by using Numpy's linear algebra package's Frobenius norm equation:

$$min_{W,H}||X - WH||_F^2 \tag{1}$$

. Reconstructed X Matrix Equation:

$$X \approx WH$$

For our outputted H matrix, we extracted the indices of the row that contains the maximum value in each column to create our clusters. For our outputted W matrices, we extracted the indices of the maximum values in each row to associate the indices with a genre.

5 Results

NMF allows us to represent X as a linear combination of $\mathbf{W}^{n \times k}$ and $\mathbf{H}^{k \times p}$. Each cluster "k" represents a genre. When we decompose X into the W matrix, each row represents an audio feature and each column represents a cluster. The values of w are the weights assigned to each audio feature for a song. The higher the weight, the more a particular cluster is determined by that audio feature. To determine the most important audio features for each cluster, we take the maximum value in each row; these maximum values are shown as gray in Table 2. The maximum value in (highlighted in gray in Table 2) a row indicates which audio feature is most important to that particular cluster. Together, the gray values within a column make up the basis for that cluster.

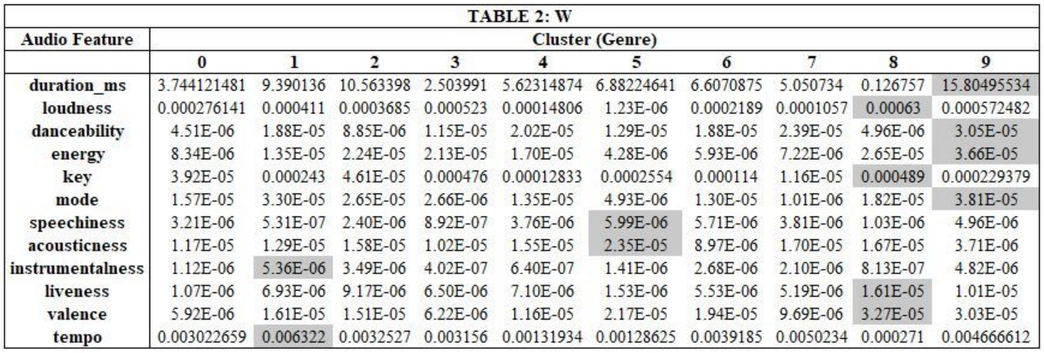


TABLE 2: W

Audio Feature	Cluster (Genre)									
	0	1	2	3	4	5	6	7	8	9
duration_ms	3.744121481	9.390136	10.563398	2.503991	5.62314874	6.88224641	6.6070875	5.050734	0.126757	15.80495534
loudness	0.000276141	0.000411	0.0003685	0.000523	0.00014806	1.23E-06	0.0002189	0.0001057	0.00063	0.000572482
danceability	4.51E-06	1.88E-05	8.85E-06	1.15E-05	2.02E-05	1.29E-05	1.88E-05	2.39E-05	4.96E-06	3.05E-05
energy	8.34E-06	1.35E-05	2.24E-05	2.13E-05	1.70E-05	4.28E-06	5.93E-06	7.22E-06	2.65E-05	3.66E-05
key	3.92E-05	0.000243	4.61E-05	0.000476	0.00012833	0.0002554	0.000114	1.16E-05	0.000489	0.000229379
mode	1.57E-05	3.30E-05	2.65E-05	2.66E-06	1.35E-05	4.93E-06	1.30E-05	1.01E-06	1.82E-05	3.81E-05
speechiness	3.21E-06	5.31E-07	2.40E-06	8.92E-07	3.76E-06	5.99E-06	5.71E-06	3.81E-06	1.03E-06	4.96E-06
acousticness	1.17E-05	1.29E-05	1.58E-05	1.02E-05	1.55E-05	2.35E-05	8.97E-06	1.70E-05	1.67E-05	3.71E-06
instrumentalness	1.12E-06	5.36E-06	3.49E-06	4.02E-07	6.40E-07	1.41E-06	2.68E-06	2.10E-06	8.13E-07	4.82E-06
liveness	1.07E-06	6.93E-06	9.17E-06	6.50E-06	7.10E-06	1.53E-06	5.53E-06	5.19E-06	1.61E-05	1.01E-05
valence	5.92E-06	1.61E-05	1.51E-05	6.22E-06	1.16E-05	2.17E-05	1.94E-05	9.69E-06	3.27E-05	3.03E-05
tempo	0.003022659	0.006322	0.0032527	0.003156	0.00131934	0.00128625	0.0039185	0.0050234	0.000271	0.004666612

Similarly, for $\mathbf{H}^{k \times p}$, each row represents a genre and each column represents a song. The maximum value within each column, based on the row that that maximum value falls in, indicates what cluster (genre) the song belongs to. The table shows a subset of our H matrix. The highlighted value in each column is the maximum value for that song. Thus, the row that this maximum value falls in for each song corresponds to its cluster.

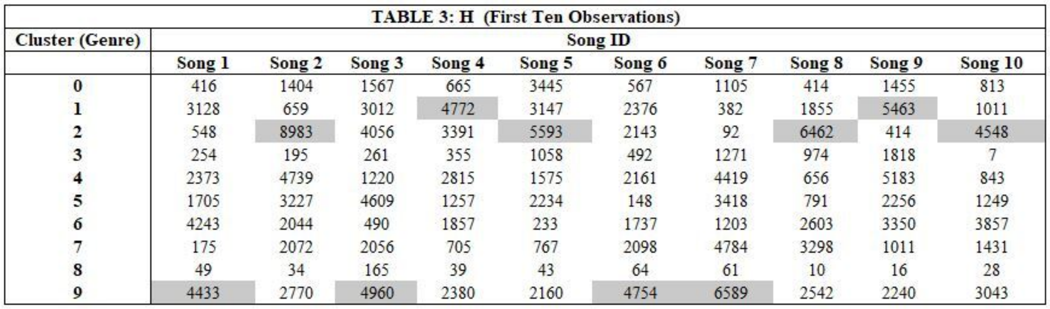
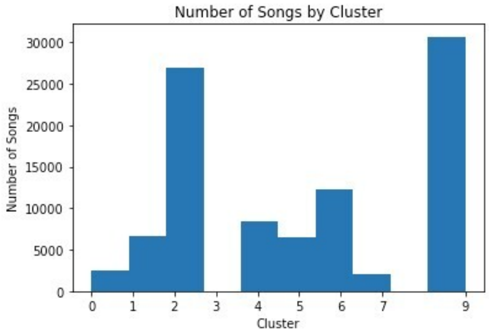


TABLE 3: H (First Ten Observations)

Cluster (Genre)	Song ID									
	Song 1	Song 2	Song 3	Song 4	Song 5	Song 6	Song 7	Song 8	Song 9	Song 10
0	416	1404	1567	665	3445	567	1105	414	1455	813
1	3128	659	3012	4772	3147	2376	382	1855	5463	1011
2	548	8983	4056	3391	5593	2143	92	6462	414	4548
3	254	195	261	355	1058	492	1271	974	1818	7
4	2373	4739	1220	2815	1575	2161	4419	656	5183	843
5	1705	3227	4609	1257	2234	148	3418	791	2256	1249
6	4243	2044	490	1857	233	1737	1203	2603	3350	3857
7	175	2072	2056	705	767	2098	4784	3298	1011	1431
8	49	34	165	39	43	64	61	10	16	28
9	4433	2770	4960	2380	2160	4754	6589	2542	2240	3043

For example, the values shaded in grey under Cluster 9 in the W matrix indicate that songs with larger values for audio features include duration_ms, danceability, energy, and mode fall under the genre defined by cluster 9. In the H matrix, we can see that songs 1, 3, 6, and 7 are classified as the genre described by cluster 9 because they have the maximum value in their respective song columns. This implies that songs 1, 3, 6, and 7 are more likely to have a longer duration and a higher measure of danceability, energy, and mode.

In our manual matching process to see if the clustering made sense, we first took the cluster assigned to each song from the H matrix. Since our X matrix did not contain any genre labels or intrinsic values, we tried to match these clustered genres with the genres in the artists dataset. For example, we extracted all the songs whose genre contained the word “pop” and then looked at the most occurring cluster within that group. The 4,694 songs labeled as “pop” were classified as cluster 9, according to our H matrix. Another genre we examined was “new age,” and we found that forty songs within this genre matched up to cluster 2 from our H matrix. If our algorithm continued to work as expected, we could repeat this process for other genres and each genre would have a match to one of the nine clusters in the H matrix. However, there are a variety of reasons as to why this did not turn out to be the case.



6 Limitations and Further Questions

Overall, it ended up being difficult to distinguish and label obvious genres to each cluster found in W and H. Going forward, we would look into whether different features would have been more helpful in the clustering of songs. The Spotify data included songs from all over the world and included 2,690 genres; therefore, we might try and filter the data to one country or culture and see if this helps improve the interpretability of our H matrix by reducing the total number of possible genres that are needed to describe so many songs from different cultures.

Better acoustic features may exist for grouping songs together, which could help us improve our decomposition and interpretation of W and H. Since the majority of our songs were classified to cluster 9, additional acoustic features such as mel spectogram, timbre, or beats per minute, could help us differentiate between songs more effectively. In addition, the songs included in the dataset release dates range from the 1950s through the 2020s. We could also consider limiting our data to a smaller period of time since genres may have variability over time.

Although we used trial and error to select the correct “k” for our data, it is also possible that a completely different number of clusters is more appropriate for this data. The “artist” dataset included many types of genres we had never heard of before, often from countries around the world, so it might be possible to more robustly come up with the proper k by using the SVD or other methods suggested by Gillis. Even though we often think of music being classified into only a few big genres, it’s likely that there are many more genres than we initially thought that best represent all Spotify songs.

It is likely that we did not need all 89,609 songs in order to determine the basis of genres that would apply to most songs. Many recommendation systems and user preferences can be implemented using a small number of users, in this case, we may have had better results by using even a smaller number of songs. We also made a decision to limit the songs in our dataset to be those which only had one genre assigned to its artist. This means that the remaining 89,609 songs in our X matrix might not reflect the true reality of what is going on in the data, due to the possible peculiarity of songs that only are assigned one genre. Going forward, we could include songs with multiple genres to more accurately classify our clusters.

7 Conclusion

Our goal for this project was to implement NMF to Spotify song data in order to classify songs into genres. We were able to group songs into nine distinct clusters and used the “artists” dataset in order to interpret our W and H matrices. However, most of our songs fell into the ninth cluster, indicating that the audio features we currently have may not divide up the songs into as many distinct genres as we had initially thought they would. In the future, we would address the limitations of this experiment in ways such as adding more audio features and having a robust method of tuning our hyperparameters. In doing so, we would hope to have a more accurate and distinct set of clusters that classify genres of songs.

References

[1] Fogel, P., Gaston-Mathé, Y., Hawkins, D., Fogel, F., Luta, G., & Young, S. S. (2016). Applications of a novel clustering approach using non-negative matrix factorization to environmental research in public health. *International journal of environmental research and public health*, 13(5), 509.

[2] Gillis, N. (2014). it The why and how of nonnegative matrix factorization. Connections, 12(2).

[3] Lee, D. D., & Seung, H. S. (1999). *Learning the parts of objects by non-negative matrix factorization.* Nature, 401(6755), 788-791.

[4] Lee, D., & Seung, H. S. (2000). *Algorithms for non-negative matrix factorization. Advances in neural information processing systems, 13.*

[5] Calero Valdez, A., Ziefle, M., Verbert, K., Felfernig, A., & Holzinger, A. (2016). *Recommender systems for health informatics: state-of-the-art and future perspectives. In Machine learning for health informatics (pp. 391-414).* Springer, Cham.

[6] https://medium.com/logicai/non-negative-matrix-factorization-for-recommendation-systems-985ca8d5c16c

[7] https://iksinc.online/2016/03/21/what-is-nmf-and-what-can-you-do-with-it/

[8] https://towardsdatascience.com/non-negative-matrix-factorization-for-image-compression-and-clustering-89bb0f9fa8ee

[9] https://towardsdatascience.com/using-nmf-to-classify-companies-a77e176f276f

[10] https://www.kaggle.com/datasets/yamaerenay/spotify-dataset-19212020-600k-tracks?resource=download