Soft-Subspace Clustering on a High-Dimensional Musical Dataset

Emil Juzovitski

August 27, 2019

Master Thesis Presentation

Examiner: Pawel Herman
Supervisor: Johan Gustavsson

Introduction

Background

Problem Statement

Method

Results

Discussion

What is Clustering analysis?

Finding clusters (groups) in a set of data objects

What is Clustering analysis?

Finding clusters (groups) in a set of data objects

- feature similar data objects partitioned into the same cluster
- feature dissimilar data objects partitioned into different cluster.

What is Clustering analysis?

Finding clusters (groups) in a set of data objects , based on similarity

Example Tasks:

- Fitting products into different aisles in a grocery store
- Grouping distributors based on the products they sell

What is Clustering analysis?

Finding clusters (groups) in a set of data objects , based on similarity

- Point/Object - vector of features describing the object
- How do we measure similarity?

$$D(X_1, X_2) = \sum_j^m d(x_{1j}, x_{2j})$$

Where j is a feature.

Task

Categorizing songs based on anonymous numerical audio features

-
-

Our Dataset and Task

Task

Categorizing songs based on anonymous numerical audio features

- Not specifically looking to categorize data based on genre
- Can we find usable themes, composers can not?

k-means

Partition-based clustering measuring similarity through the Euclidean distance

-

k-means

Partition-based clustering measuring similarity through the Euclidean distance

- Find k clusters by iteratively optimizing cluster- centers and members

k-means

Iteratively optimize one variable at a time for cost function $P(U, C)$ where U is the partition matrix, and C the cluster center.

P1. Assign each point to the most similar cluster.

Fix C = $\hat{C}$, optimize $P(U, \hat{C})$

P2. Update the mean for each cluster.

Fix U = $\hat{U}$ optimize $P(\hat{U}, C)$

k-means

Iteratively optimize one variable at a time for cost function $P(U, C)$ where U is the partition matrix, and C the cluster center.

P1. Assign each point to the most similar cluster.

Fix C = $\hat{C}$, optimize $P(U, \hat{C})$

$$u_{il} = \begin{cases} 1 & d(X_i, C_l) \leq d(X_i, C_t) \quad \text{for} \quad 1 \leq t \leq k \\ 0 & \text{for} \quad t \neq l \end{cases}$$

k-means

Iteratively optimize one variable at a time for cost function $P(U, C)$ where U is the partition matrix, and C the cluster center.

P2. Update the mean for each cluster.

Fix U = $\hat{U}$ optimize $P(\hat{U}, C)$

$$c_{lj} = \frac{\sum_{i=1}^{n} u_{il} x_{ij}}{\sum_{i=1}^{n} u_{il}}$$

k-Means Iteration Steps

1. Choose initial cluster representatives C^0
2. Fix $C^t = \hat{C}$, optimize $P(U^t, \hat{C})$, Obtain $P(U^{t+1}, \hat{C})$

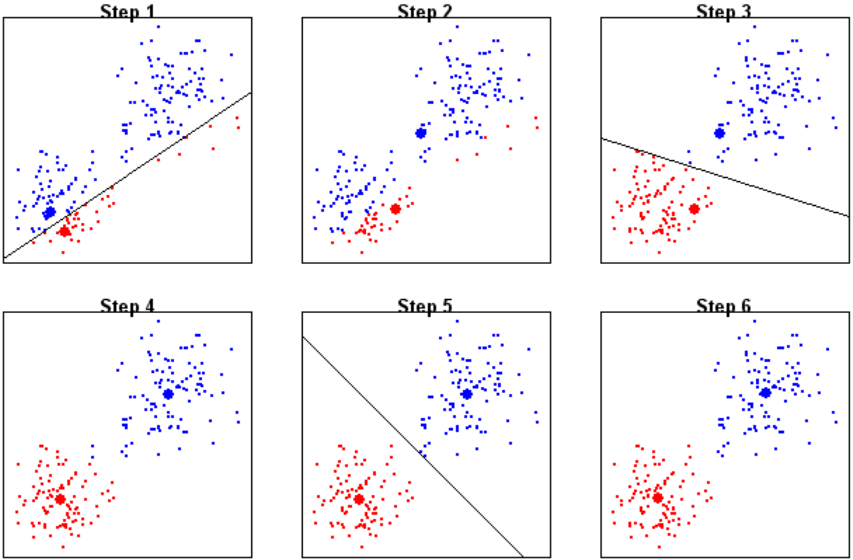
   if $P(U^t, \hat{C}) = P(U^{t+1}, \hat{C})$

   return $P(U^t, \hat{C})$ (Convergence)
3. Fix $U^{t+1} = \hat{U}$, optimize $P(\hat{U}, C^t)$ Obtain $P(\hat{U}, C^{t+1})$

   if $P(\hat{U}, C^t) = P(\hat{U}, C^{t+1})$

   return $P(\hat{U}, C^t)$ (Convergence)
4. Repeat steps 2. and 3.

k-Means Iteration Steps



Problems:

- Using the wrong features is destructive.
- Knowing what features to use is hard and time-intensive, sometimes impossible.
- Rhythm, Vocals? for theme$_2$?

Soft-subspace clustering

An extension of K-means with an additional variable to optimize W — allowing cluster unique feature weights.

- Automatic
- Embedded
- Cluster unique feature weights (Subspaces)
- Linear!

SSC Iteration Steps

1. **Choose initial cluster representatives** C^0
2. **Fix** $C^t = \hat{C}$, $W^t = \hat{W}$, **optimize** $P(U^t, \hat{C}, \hat{W})$. **Obtain** $P(U^{t+1}, \hat{C}, \hat{W})$

   if $P(U^t, \hat{C}, \hat{W}) = P(U^{t+1}, \hat{C}, \hat{W})$

   return $P(U^t, \hat{C}, \hat{W})$ **(Convergence)**
3. **Fix** $U^{t+1} = \hat{U}$, $W^t = \hat{W}$, **optimize** $P(\hat{U}, C^t, \hat{W})$. **Obtain** $P(\hat{U}, C^{t+1}, \hat{W})$

   if $P(\hat{U}, C^t, \hat{W}) = P(\hat{U}, C^{t+1}, \hat{W})$

   return $P(\hat{U}, C^t, \hat{W})$ **(Convergence)**
4. **Fix** $\hat{U} = U^{t+1}$, $\hat{C} = C^{t+1}$, **optimize** $P(\hat{U}^t, \hat{C}^t, W^t)$. **Obtain** $P(\hat{U}, \hat{C}, W^{t+1})$

   if $P(\hat{U}, \hat{C}, W^t) = P(\hat{U}, \hat{C}, W^{t+1})$.

   return $P(\hat{U}, \hat{C}, W^t)$ **(Convergence)**

SSC Iteration Steps

4. Fix $\hat{U} = U^{t+1}$, $\hat{C} = C^{t+1}$, optimize $P(\hat{U}^t, \hat{C}^t, W^t)$. Obtain $P(\hat{U}, \hat{C}, W^{t+1})$

   if $P(\hat{U}, \hat{C}, W^t) = P(\hat{U}, \hat{C}, W^{t+1})$.

   return $P(\hat{U}, \hat{C}, W^t)$ (Convergence)

1. FSC

$$P(U, C, W) = \sum_{l=1}^{k} \left[\sum_{i=1}^{n} \sum_{j=1}^{m} u_{il} w_{lj}^{\beta} d(x_{ij}, c_{lj}) + \epsilon \sum_{j=1}^{m} w_{lj}^{\beta} \right] \tag{1}$$

$$w_{lj} = \frac{1}{\sum_{t=1}^{m} \left[\frac{D_{lj}+\epsilon}{D_{lt}+\epsilon} \right]^{\frac{1}{\beta-1}}} \tag{2}$$

$$D_{lj} = \sum_{X_i \in C_l} d(x_{ij}, c_{lj}) \tag{3}$$

2. EWKM
3. LEKM

1. FSC
2. EWKM

$$P(U,C,W) = \sum_{l=1}^{k}\left[\sum_{i=1}^{n}\sum_{j=1}^{m} u_{il} w_{lj} d(x_{ij}, c_{lj}) + \gamma \sum_{j=1}^{m} w_{lj} log(w_{lj})\right] \tag{1}$$

$$c_{lj} = \frac{\sum_{X_i \in C_l} x_{ij}}{\mathrm{Count}_{X_i \in C_l}} \tag{2}$$

$$w_{lj} = \frac{\exp(\frac{-D_{lj}}{\gamma})}{\sum_{t=1}^{m} \exp\left(\frac{-D_{lt}}{\gamma}\right)} \tag{3}$$

$$\tag{4}$$

3. LEKM

1. FSC
2. EWKM
3. LEKM

$$P(U,C,W)=\sum_{l=1}^{k}\sum_{i=1}^{n}u_{il}\left[\sum_{j=1}^{m}w_{lj}\ln\left[1+d(x_{ij},c_{lj})\right]+\gamma\sum_{j=1}^{m}w_{lj}\ln(w_{lj})\right] \tag{1}$$

$$c_{lj}=\frac{\sum_{X_i\in C_l}\left[1+d(x_{ij},c_{lj})\right]^{-1}x_{ij}}{\sum_{X_i\in C_l}\left[1+d(x_{ij},c_{lj})\right]^{-1}} \tag{2}$$

$$w_{lj}=\frac{\exp(\frac{-D_{lj}}{\gamma})}{\sum_{t=1}^{m}\exp\left(\frac{-D_{lt}}{\gamma}\right)} \tag{3}$$

$$D_{lj}=\frac{\sum_{X_i\in C_l}\ln\left[1+d(x_{ij},c_{lj})\right]}{\text{Count}_{X_i\in C_l}} \tag{4}$$

Introduction

Background

Problem Statement

Method

Results

Discussion

Problem Statement

1. *What is a suitable soft-subspace algorithm for the given dataset?*
2. *How does the performance of the chosen soft-subspace clustering algorithm compare to K-means from the perspective of novelty and general quality?*

Method PS 1. Tuning parameters on an external criteria — genre purity

Selecting the best algorithm.

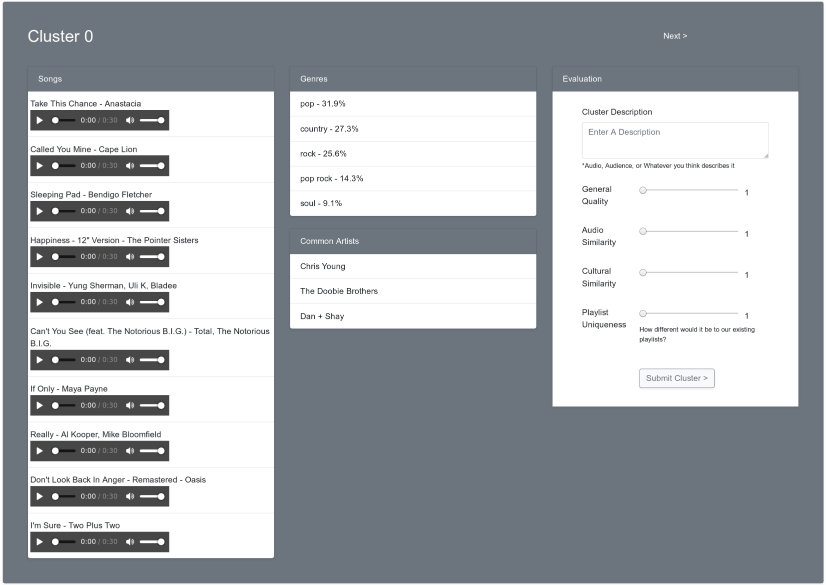
Method PS 2. Asked a panel of music composers to rate clusters.

Method PS 1.

Hyperparameter	Description	Algorithm
k	Number of clusters	All
β	Weighting factor for fuzzy-weighting algorithms	FSC
γ	Entropy regularization factor	EWKM, LEKM

Method PS 2.

Criteria	Description
Audio Similarity	How well do songs in the cluster share audio patterns?
Cultural Similarity	How well do songs in the cluster share an audience?
Playlist Uniqueness	How easy would it be for a composer come up with a similar playlist?
General Quality	How useful is the cluster for the composers?


Cluster 0
Next >
Songs
Take This Chance - Anastacia
0:00 / 0:30
Called You Mine - Cape Lion
0:00 / 0:30
Sleeping Pad - Bendigo Fletcher
0:00 / 0:30
Happiness - 12" Version - The Pointer Sisters
0:00 / 0:30
Invisible - Yung Sherman, Uli K, Bladee
0:00 / 0:30
Can't You See (feat. The Notorious B.I.G.) - Total, The Notorious B.I.G.
0:00 / 0:30
If Only - Maya Payne
0:00 / 0:30
Really - Al Kooper, Mike Bloomfield
0:00 / 0:30
Don't Look Back In Anger - Remastered - Oasis
0:00 / 0:30
I'm Sure - Two Plus Two
0:00 / 0:30
Genres
pop - 31.9%
country - 27.3%
rock - 25.6%
pop rock - 14.3%
soul - 9.1%
Common Artists
Chris Young
The Doobie Brothers
Dan + Shay
Evaluation
Cluster Description
Enter A Description
*Audio, Audience, or Whatever you think describes it
General Quality
1
Audio Similarity
1
Cultural Similarity
1
Playlist Uniqueness
1
How different would it be to our existing playlists?
Submit Cluster >

Introduction

Background

Problem Statement

Method

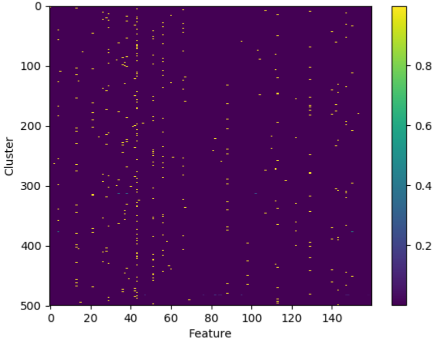
Results

Discussion

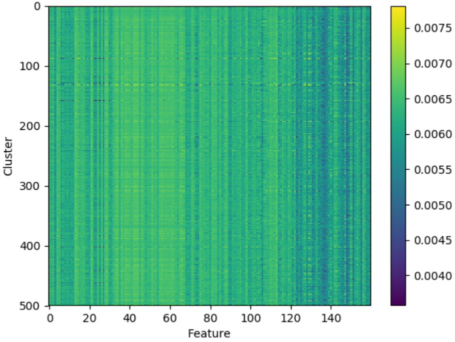
	K-means	EWKM		LEKM		FSC	
k	Purity	γ	Purity	γ	Purity	β	Purity
50	39.0%	0.005	28.2%	0.0	38.8%	30.0	38.0
100	40.8%	0.005	28.9%	2.1	40.6%	15	39.6
500	45.4%	0.001	28.6%	1.4	**45.5%**	30.0	43.8%

Best parameter- and purity-score (γ, β) given k, where scores represent the mean purity of three runs.

Results Part I.



(a) EWKM



(b) LEKM

Results Part 2.

	Mean And Standard Deviation			
Source	General Quality	Audio Similarity	Cultural Similarity	Playlist Uniqueness
Playlist	7.7 (1.2)	7.8 (1.5)	7.7 (0.6)	4.5 (3.5)
K-means	6.6 (2.5)	7.0 (2.3)	6.2 (2.9)	5.4 (3.2)
LEKM	6.4 (3.4)	6.4 (2.4)	6.0 (2.5)	3.8 (2.6)

	One-Way ANOVA			
Index	General Quality	Audio Similarity	Cultural Similarity	Playlist Uniqueness
$F_{0.05}$	0.21	0.45	0.47	0.37
F_{crit}	4.10	3.98	4.10	3.98
P_{-value}	0.81	0.65	0.64	0.70

Findings

1. Cannot differ k-means and LEKM in terms of purity
2. Cannot reject H_0, $\mu_{playlist} = \mu_{kmeans} = \mu_{LEKM}$!

Introduction

Background

Problem Statement

Method

Results

Discussion

Discussion

- Limitations in testing purity, composer evaluation
- Underlying dataset problems, another limitation
- Dataset properties enjoy uniform weights?

Future Work

1. Better Initialization
2. Mixed Data SSC.
3. Inter-cluster distances
4. Automize k-selection.

1. **Opponent Questions.**
2. **Other Questions.**