グループ全体の好みを考慮した楽曲リスト自動生成ツール

Fixstars Amplify Hackathon

早稲田大学　武笠 陽介

こんなことありませんか？

みんな楽しそうだけど
実はこの曲知らない…

ドライブにて

本当は○○がいいけど
マイナーすぎるから
ここは無難に△△で…

いつも無難な△△で
飽きてきたなあ…

カラオケにて

みんなが知ってる曲で楽しみたい！！

だけど参加者の好みを確認して
人力で決めるのは現実的ではない。。。

みんなが知ってる曲で楽しみたい！！

参加者の好みを反映したデータがあれば
最適化問題として解けるのでは？

問題設定

前提

- 参加者各自の好みを反映した楽曲リストがある
  （ここでは楽曲リストの構成曲を候補曲と呼ぶ）

目的

- グループの好みを考慮した楽曲リストを作成する

ただし...

- グループの好みの合計を最大化したい
- グループの好みのばらつきを最小化したい
- 楽曲リストの時間を指定した時間と同程度にしたい

定式化

- 参加者数：M
- 参加者 j の候補曲集合：F_j
- グループの候補曲集合：$F = \bigcup F_j = \{f_1, f_2, \ldots, f_N\}$
- 候補曲数：$N\ (\leq \sum |F_j|)$
- 候補曲 f_i の時間： t_i
- 指定時間：T
- 参加者 j の候補曲 f_i に対する好み：$p_{i,j} \in [0,\ 3]$
- 候補曲 f_i を選択するかどうか：$x_i \in \{0,\ 1\}$

定式化

参加者 j の好みの総和：$P_j = \sum_{i=1}^{N} p_{i,j} x_i$

グループの好みの平均：$\overline{P} = \frac{1}{M} \sum_{j=1}^{M} P_j$

- グループ全体の好みの総和を最大化

$$H_{\text{sum}} = -\sum_{j=1}^{M} P_j$$

- グループ全体の好みの分散を最小化

$$H_{\text{var}} = \frac{1}{M} \sum_{j=1}^{M} (\overline{P} - P_j)^2$$

- 指定時間と同程度にする制約

$$H_{\text{time}} = \left(\sum_{i=1}^{N} t_i x_i - T \right)^2$$

- ハミルトニアン　$H = H_{\text{sum}} + H_{\text{var}} + \lambda H_{\text{time}}$

Amplify AEで実行

- 200曲分の楽曲データ（Spotifyから取得）を用意。
- 5人の参加者を仮定して、それぞれ200曲のうち100曲をランダムに選択し、選択した曲の好み(1-3)をランダムに設定。選択されていない曲は好みを0に設定。
- 生成したい楽曲リストの時間は30分に設定。

サンプルデータ

	ID	アーティスト	曲名	時間	ユーザー1	ユーザー2	ユーザー3	ユーザー4	ユーザー5
0	JPTF00809111	Mr.Children	花の匂い	05:10	⭐⭐⭐			⭐⭐	
1	JPTF09205801	Mr.Children	虹の彼方へ	03:35	⭐		⭐		⭐⭐⭐
2	JPTF09809702	Mr.Children	Prism	04:40	⭐⭐⭐			⭐	
3	JPU901803007	King Gnu	Prayer X (Acoustic)	01:29	⭐⭐	⭐⭐⭐			
4	JPTF09501701	Mr.Children	【es】~Theme of es~	05:49	⭐	⭐	⭐		⭐
...	...	...	...	...	...	...	...	...	...
189	JPU901702278	King Gnu	サマーレイン・ダイバー	04:17				⭐⭐	
190	JPU901702271	King Gnu	Tokyo Rendez-Vous	04:01				⭐	
191	JPU901702276	King Gnu	ロウラヴ	03:32					⭐
192	JPB600057201	浜崎あゆみ	M	04:26					⭐⭐⭐
193	JPWP02000424	美波	アメヲマツ、	04:54					⭐⭐

Amplify AEで実行

結果

設定した時間どおりで
好みのばらつきも小さく
意図した結果が得られた

合計
時間 0:30:05

	ユーザー1	ユーザー2	ユーザー3	ユーザー4	ユーザー5	合計	平均	分散
好み	18.0	18.0	19.0	17.0	20.0	92.0	18.4	1.04

	image	アーティスト	曲名	時間	ユーザー1	ユーザー2	ユーザー3	ユーザー4	ユーザー5
0		King Gnu	McDonald Romance	02:45	★★★		★		★★★
1		Rin音, RhymeTube	snow jam	03:05	★★★		★★★	★	★
2		LiSA	She	03:56	★★★	★★★	★	★★★	★
3		King Gnu	飾りじゃないのよ 涙は	03:08	★★	★★★	★★	★★	★
4		King Gnu	Overflow	03:21	★★★		★★★		★★
5		Mrs. GREEN APPLE	インフェルノ	03:31	★★	★★★	★★★	★★★	★★★
6		King Gnu	Prayer X (Acoustic)	01:29		★★★	★★★	★★	
7		米津玄師	Flamingo	03:16		★★★	★★	★★★	★★★
8		King Gnu	It's a small world	03:13		★★★	★	★★★	★★★

手法の妥当性

ナイーブな2種類のアルゴリズムと比較

1. グループの好みの最大化を狙う

(i) 時間あたりの参加者の好みの合計が高い順にソートする。

(ii) 設定時間の範囲内で上から順に選択する。

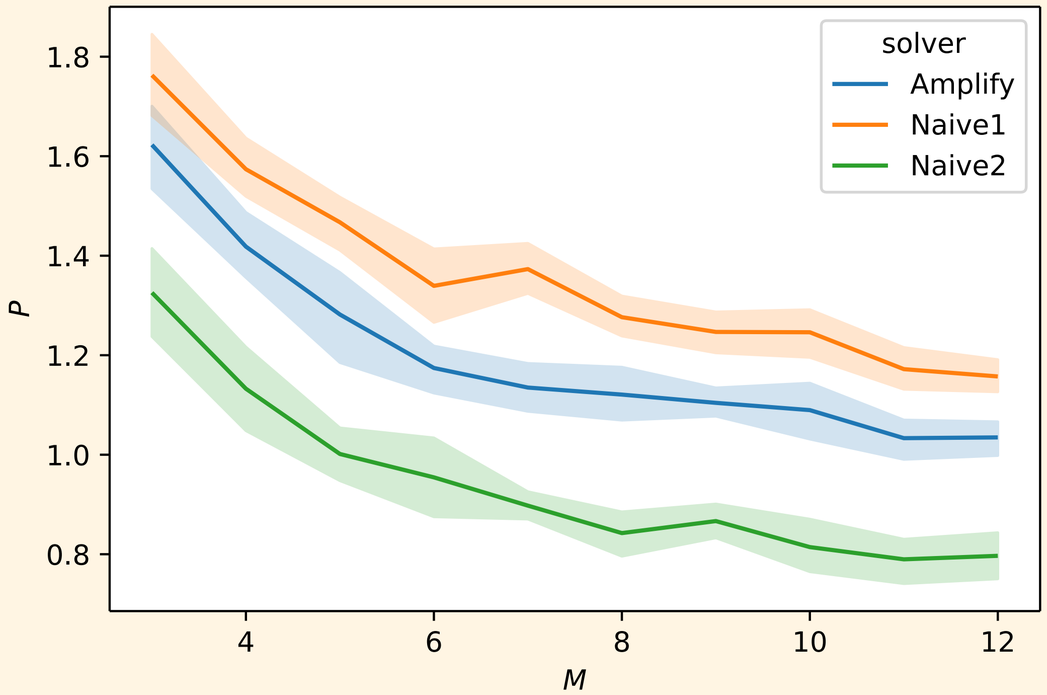
2. 分散を抑えつつ個人の好みの最大化を狙う

(i) 時間あたりの参加者 i の好みが最も高いものを選択する。

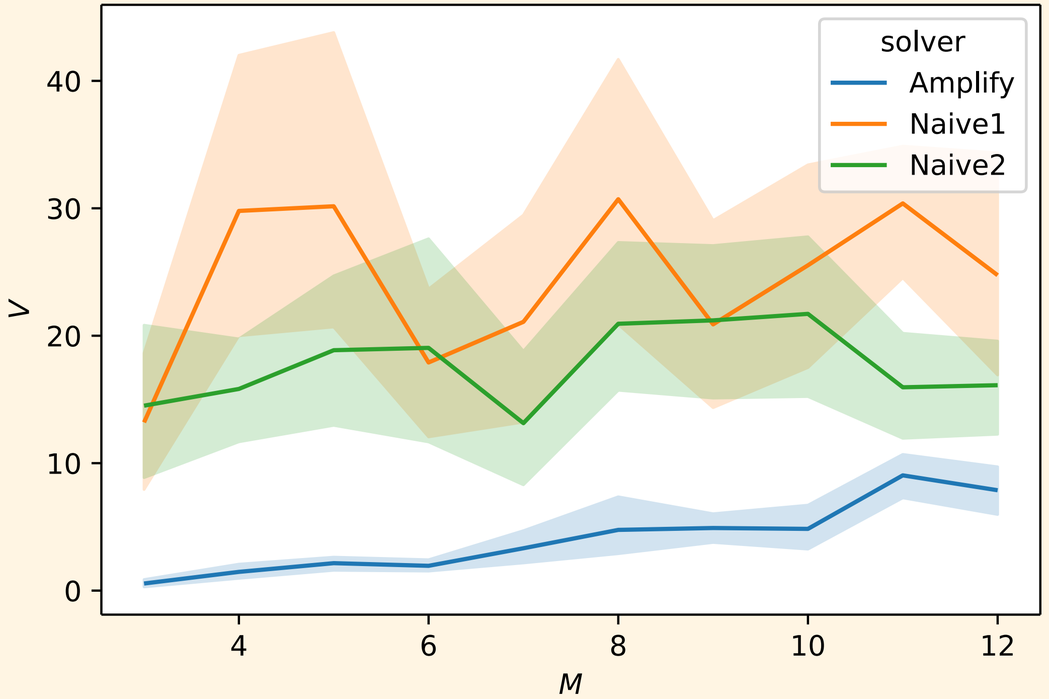
(ii) 設定時間の範囲内で各参加者 i について手順(i)を繰り返す。

手法の妥当性

- M人が300曲からランダムに選択した50曲に対して好みをランダムに設定したデータセット
- $M = \{3,4,\ldots,12\}$ について各100サンプルを取得し、1曲あたりの好みと分散を比較



参加者数 M – 1曲あたりの好み P



参加者数 M – 好みの分散 V

サンプルデータだとよくわからない 🤷‍♂️

実際のデータでどうなるか気になる 🤔

Webサイト上で試せるようにしました

おまかセトリ（β版）

https://omakasetli.com

おまかセトリ（β版）

https://omakasetli.com

システム構成

Webインターフェース
(HTML, CSS, React.js)

Google Cloud Functions

Cloud Firestore

Web API サーバー
(Python)

AMPLIFY

アニーリング

おまかセトリ（β版）

https://omakasetli.com

1. ユーザー登録

Googleアカウントでログイン可能で、
初回ログイン時にユーザー登録をします。

以下の項目を設定可能です。

・ユーザーID
・ユーザー名
・プロフィール画像

おまかセトリ（β版）

https://omakasetli.com

2. サービス連携

右上アイコン > アカウント設定 > サービス連携

自分のプレイリストを作成するために、
音楽配信サービスと連携することができます。
現時点では以下の2種類に対応しています。

- Apple Music
- Spotify

おまかセトリ（β版）

https://omakasetli.com

3. プレイリストに楽曲インポート

右上アイコン > プレイリスト > ＋ボタン

連携したサービスの自分のプレイリストから
楽曲をインポートすることができます。

プレイリストを選択すると1曲ずつ好みを設定
する画面となるので入力して保存します。

おまかセトリ（β版）

https://omakasetli.com

4. ルーム作成 & 入室

右上アイコン > ルーム

作成
ルーム名を入力して作成可能です。

入室
ルームNo.で検索して入室することができます。

おまかセトリ（β版）

https://omakasetli.com

5. セットリスト作成

[※ルーム作成者のみ可能]

ルームの参加者が揃ったら、
予定時間を入力して作成ボタンを押します。

各参加者のプレイリストに基づいた
セットリストがAmplifyを用いて作成されます。

おまかセトリ（β版）

https://omakasetli.com

6. セットリスト確認

作成されたセットリストについて、

・選択された楽曲（構成曲）
・参加者の好みを反映した指標（満足度）

を確認することができます。

まとめ

- Amplifyを用いてグループの楽曲の好みの分散を最小化・総和を最大化するソルバーを開発
- ソルバーを実際に利用するWebアプリを開発して実データでも動作確認を実施
- 個人的に調べた限りでは類似するアプリケーションはない・・・新規性
- 音楽配信サービスのAPIと連携したデータを活用・・・進歩性
- Amplifyを利用した独自のWeb API & UI を実装・・・技術力
- 現実的なユースケースに対してAmplify(約6万ビット=曲数)で十分に対応可能・・・実用性