デジタルプラクティス　Vol.10 No.2(Apr. 2019)

特集号招待論文

AWAにおける類似プレイリスト探索システムの構築

水上 裕貴[1]　福田 鉄也[1]　山下 剛史[1]　Connolly Juhani[1]　武内 慎[1]　數見 拓朗[1]　福田 一郎[1]

[1]（株）サイバーエージェント

本稿ではストリーミング型音楽配信サービスAWAにおけるコンテンツ推薦システムの構築，およびシステム稼働に至るまでの実践的なプロセスについて紹介する．AWAではユーザに対する適切なコンテンツの推薦，とりわけ類似コンテンツの推薦が，体験向上の大きな要因となる．本サービスにおいてユーザに推薦すべきコンテンツとしては，単一の「楽曲」の他に，楽曲の集まりである「プレイリスト」があるが，後者のプレイリストは多くの場合，ユーザが高頻度で内容を更新することが多い．このため従来の大規模集計による手法では更新の反映待ち時間と推薦元の網羅率に課題があった．そこで我々は適切な特徴量加工と近似最近傍探索を用いた類似プレイリスト探索システムを構築した．その結果として，更新の反映待ち時間については最大で24時間ほど要していたものをほぼリアルタイムに，網羅率に関しては65%から98%に改善した．

1. はじめに

近年では，レコード産業においても情報処理技術の利活用が活発になってきている．レコード産業とは音楽産業のうち，とりわけ録音・録画された楽曲や映像を流通させる産業を指すが，流通の形態は常に変化を続けている．レコード産業の誕生以降，音楽はレコードやカセットテープそしてCDなどの物理メディアを通じて流通していた．やがて情報通信技術が発展しスマートフォンなどが普及すると，インターネットを通じて楽曲や映像が流通するようになり，これらは総称してディジタルメディアと呼ばれている．

また，レコード産業では物理メディアからディジタルメディアへの急速な代替が起きている[1][2]．2001年時点でレコード産業の市場のほぼ全体を占めていた物理メディアの市場規模も2015年にはディジタルメディアの市場規模が逆転した．また，同時にレコード産業全体の規模もこの年以降増加傾向に転じている[1]．ディジタルメディアを通じた音楽体験の向上は，レコード産業において今後ますます重要な役割を果たすと考えられる．

消費者の立場からみれば，ディジタルメディアの普及により音楽の体験は大きく変化した．物理メディアにおいて発生していた生産や流通コストの大部分はディジタルメディアによって効率化され，インターネットと端末があればどこでも楽曲が入手出来るようになった．そして楽曲の流通の効率化に伴い，楽曲を選択するコストが顕在化し課題になった．この課題に対する技術的回答として，ユーザが指定したクエリに合致する情報を探索する「情報検索」などがあるが，近

年では行動ログなどのデータを用いて必要な情報を予測して，高い精度で提示する「情報推薦」も応用範囲を広げている．この情報推薦の技術は，利用者に対して必ずしも情報要求の言語化・クエリ化を課さず，音楽体験においても未知の楽曲との接触機会を増加させると考えられており，レコード産業からも注目を集めている．

AWAは5000万曲超の楽曲を再生時にインターネット経由でスマートフォンなどの端末に配信するストリーミング型音楽配信サービスである．筆者らは，利用者の音楽体験の向上に向けて日々ユーザインタフェースの洗練や配信コンテンツの拡充，データを活用した楽曲推薦の品質向上に取り組んでいる．本稿では，AWAの特徴的な機能であるプレイリストに関して，ニューラルネットワーク自然言語処理分野で実績のあるskip-gramモデルを応用して構築した推薦システムの事例について紹介する．

1.1　推薦システム

推薦システムとは，価値の高い商品やコンテンツを特定するのを助ける道具[3]である．マーケティングの言葉に言い換えれば顧客が購入する商品・アイテムを予測し，訴求する技術といえる．

アイテムの推薦は基本的には利用者の嗜好に沿うように行われる．例としては「近しい嗜好の利用者同士は近しいアイテムを好む」という仮説にもとづく協調フィルタリング（Collaborative Filtering）がある．この協調フィルタリングは多くの場合，利用者の行動履歴を蓄積・比較することで実現される．また似たような方針として，アイテム自体の情報にもとづく，内容ベースフィルタリング（Content-Based Filtering）がある．この内容ベースフィルタリングは，「興味を示したアイテムの類似アイテムにも興味を示す」という仮説のもとで，何らかの意味でアイテム間の類似度を定義することにより実現される．

また実際の推薦システムの推薦結果には往々にして，推薦者側の都合や意向が反映される．サービスの継続的な発展のために収益性を担保することや，知見獲得のために情報を収集することも推薦システムの重要な役割であり，その他にも総合的なユーザ体験の設計において情報供給者の思想を強く反映することもある．たとえば推薦されるアイテムとして，酒類などの成人向けの商品やコンテンツを取り扱う際，当然推薦者は未成年に対する影響を考慮しなければならないと考えるだろう．本稿では，このような推薦者側の意向にもとづく推薦方式をを総称して意向ベースフィルタリング（Intention-Based Filtering）と呼ぶこととする．

本稿では，上記で紹介した複数の根拠を加味したアイテム選定基準，そしてそれを利用者に届けるインターフェースおよびそのバックエンドを含むシステム全体を推薦システムと呼ぶこととする．

1.2　AWAとプレイリスト

多くの音楽再生ソフトウェアでは，あらかじめ順番が定められた楽曲の列は「プレイリスト」と呼ばれている．AWAにおけるプレイリストは，主に利用者によって作成され，公開される時点では最大で8曲から構成される．また大部分のプレイリストには，作成者によりタイトルと説明文が付与されており，利用者がプレイリストを選択する際の補助的な情報として機能している[**図1**]．以降，推薦対象となるAWAにおけるプレイリストを混同のない範囲で単にプレイリストと呼ぶこととする．AWAは利用者による，プレイリストの作成や公開が活発で，作成されたプレイリストの数は累計で1100万を超える．また多くの利用者は，プレイリストやその作成者に対してフ

ィードバックを送り合いながら新しい音楽と出会っていくが，このことはAWAにおける音楽体験の大きな特徴である．実際に，楽曲再生利用者数に対する再生動線毎の利用者数の比率をみれば，物理メディアとしての販売単位であるシングル・アルバムに匹敵する割合でユーザ作成プレイリストが利用されていることが確認できる[**図2**].

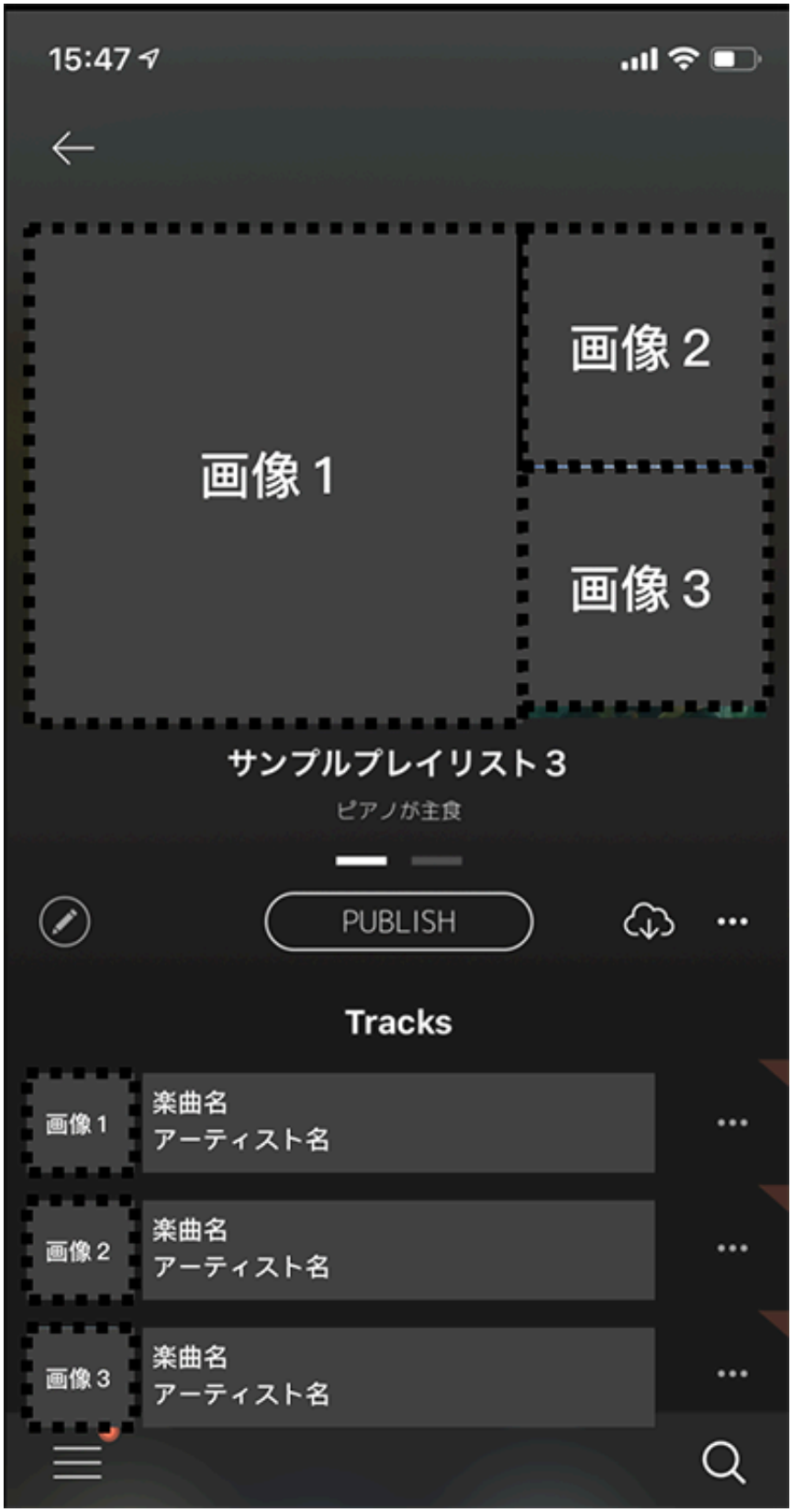

図1　ユーザ作成プレイリストのイメージ

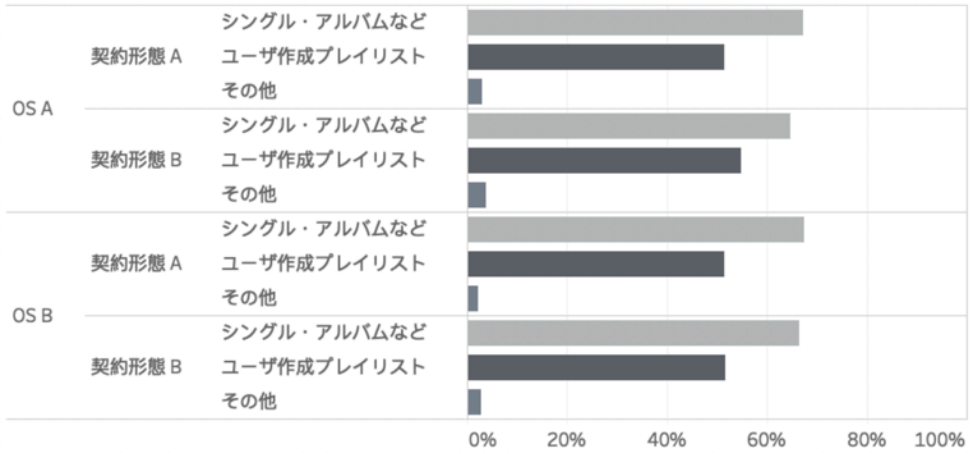

図2　再生ユーザに対する，再生動線利用率の比較

そこで，AWAではプレイリストに注目した推薦機能を実現することにした．利用者から見ると，プレイリストの再生画面の最下部には「Related Playlist」と呼ばれる関連プレイリスト推薦枠が表示されるが，この部分にシステムが推薦するプレイリストを表示するようにしている．この関連プレイリスト推薦によって，AWAの利用者に対して連続的な音楽体験を提供する機能を実装した[**図3**]．

図3　関連プレイリスト推薦枠の例

2. 類似プレイリスト探索における課題

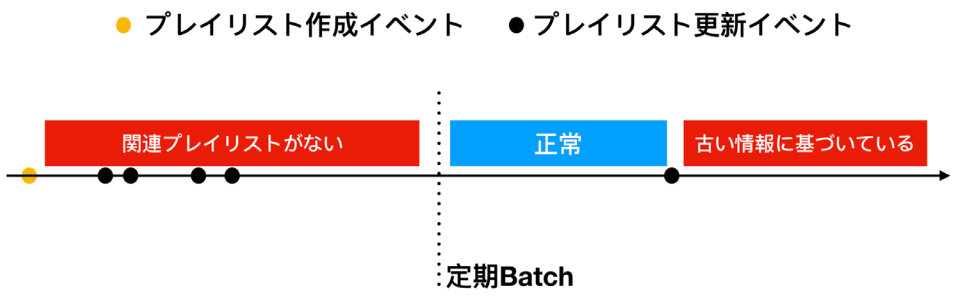

図4 関連プレイリストの作成フロー

AWAにおける関連プレイリスト推薦システムは，従来，プレイリストの類似度計算を，含まれる楽曲の楽曲情報の重複数を総あたりで計算する大規模なバッチ処理で行っていた（これを旧仕様と呼ぶ）．この旧仕様は構成楽曲の楽曲情報の一致を根拠にした内容ベース推薦の一種といえる．この章では旧仕様が抱える2点の課題について述べる．

2.1 網羅率の低さ

1点目は網羅率が低いことである．ここでは網羅率を，再生されるプレイリストのうち，関連プレイリストが1つ以上表示できるプレイリストの比率と定義する．網羅率が低い原因の1つは，計算資源の制約から，上述の大規模なバッチ処理において集計対象を作成日時や再生回数などいくつかの条件で限定していることである．もう1つの原因は，厳密な楽曲情報の一致を条件とするため，プレイリストに採用実績が無い楽曲のみからなるプレイリストは関連プレイリストが計算できないことである．

2.2 プレイリスト作成・更新に対する結果反映の遅延

2点目はプレイリストの作成・更新に対して，関連プレイリストへの結果反映が遅延することである．楽曲の内容情報は一度登録されると更新されることはとても希であるが，プレイリストでは利用者によって内容情報の作成や更新処理が頻繁に行われる．このため従来の手法ではプレイリストの作成や更新からバッチ処理までの時間に，関連プレイリストが表示されない，もしくは古い情報に基づいた関連プレイリストが表示されていた．

一方でアプリケーションの設計上プレイリストは作成・更新の直後が最も再生されるため，関連プレイリストの欠損は利用者の体験を低下させている．この適切な類似プレイリストが表示されるまでの待ち時間の内訳は以下の通りである．

- 作成待ち時間

プレイリストが作成・更新されてから次の処理が始まるまでの時間である．バッチ処理には計算機を効率的に稼働させられるなどの利点がある一方で，最大で実行間隔分の待ち時間が発生する．

- 特徴量の計算時間

次にプレイリストの類似度計算のもとになる特徴量を取得するための時間が必要になる．今回，内容ベースの推薦ならば，多くの場合，各楽曲に関連するアーティストIDなどの，参照が容易な情報が特徴量となるので，このコストは無視できるほど小さい．

一方，たとえば協調フィルタリングを行う場合は，再生ログなどのソフトクラスタリングによって得られたベクトルが特徴量となる[4]．しかしソフトクラスタリング自体に大きな計算を要するほか，ソフトクラスタリングに必要なログの収集にも時間が必要になる．高頻度で内容の作成・更新処理が発生する今回のケースに，そういった大規模な学習や集計を要する手法は適さない．

- 最近傍探索の時間

プレイリストに対して特徴量が得られたあとは，その特徴量を用いてプレイリスト間の類似度や距離を定義する．その上で最も類似すると思われる複数のプレイリストIDを得たい．最も単純なアイディアでn個のすべてのプレイリスト間で最近傍を求める場合，計算量オーダー$O(n^2)$もの

距離の評価が必要になる．また使用する類似度や距離にもよるが，特徴量の次元の大きさも近傍探索の計算時間に大きな影響を与える．プレイリストに対して含まれる楽曲IDを特徴量とするとき，この特徴量の次元は楽曲IDの8倍ほどの大きい次元になり，この場合，空間計算量が楽曲IDの数に比例する．

プレイリストのように高頻度で作成や内容の更新が行われるケースではより効率的な探索アルゴリズムや距離計算，そして低次元な特徴量が求められる．

- ネットワークの通信時間

上記の他にもクライアント-サーバ間の通信やサーバ同士の通信にも時間を要する．上記の他の要素に比べれば基本的には無視できるほど小さいが，プレイリストのように高頻度で作成・更新などのリクエストを受け取るシステムでは，各構成要素において，効率的なキャッシュの設計などの工夫がパフォーマンス向上においても重要である．

3. 要素技術

これまでよりリアルタイムな関連プレイリストの計算のために，より低次元な特徴量と，効率的な近傍探索が重要であることを述べた．この章では本稿で紹介する推薦システムで活用される先行研究を紹介する．

3.1 分散表現を用いた低次元特徴量の獲得

自然言語処理分野において文章を単語の系列とみなし，学習データとする単語のソフトクラスタリング手法の1つにskip-gramがある[5]．skip-gramは系列データのある要素から，その周辺の要素を推測する3層のニューラルネットワークで，中間層が入出力層と比べて小さい次元になっている．このモデルは「任意の単語の意味はその周辺語によって特徴づけられる」という仮説に基づき周辺語からその単語の低次元ベクトル表現を獲得する．実際に得られた単語のベクトルについて「Madrid」-「Spain」+「France」＝「Paris」のような意味上の演算が構成できるなど，単語の意味上の関係を低次元の空間上で幾何学的・代数的に再現するケースが確認されている．この変換は特徴埋め込み（Feature embedding），変換後のベクトルは分散表現（Distributed representation）と呼ばれている．

3.2 直積量子化にもとづく距離の近似計算

ユークリッド距離の近似計算には直積量子化（Product quantization）を用いることができる．いま，検索対象となる有限個の特徴量ベクトルの集合$Y \subset \mathbb{R}^d$から，クエリベクトル$x \in \mathbb{R}^d$とのユークリッド距離を最小にする特徴量ベクトル$y \in Y$を探索したい．先に述べたようにこの計算には膨大な計算が必要になるが，直積量子化により距離の計算を簡略化できる．

直積量子化のコアなアイディアは2つある．1点目は特徴量の空間をあらかじめk-平均法によりクラスタリングしておき，クエリベクトルとの距離をクラスタの重心との距離で近似し，計算量を減らすことが出来る点である．2点目は2つのベクトル間の距離の2乗が複数の区分に分割して計算出来るという点である．$\mathbb{R}^d$上のベクトルyに対して$y = \left(y_1, y_2, \ldots y_k\right) \in Y_1 \times Y_2 \times \cdots \times Y_k \subset \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \times \cdots \times \mathbb{R}^{d_k}$のようにk個の区分への分割を考える．このとき2つのベクトル$x, y \in \mathbb{R}^d$の距離の2乗 $\|x-y\|^2$は

$$||x-y||^2 = \sum_{j=1}^{k} ||x_j - y_j||^2$$

のように区分毎の距離の2乗の和と一致する.

この2つの事実を組み合わせて，分割した空間でクラスタリングした結果を元に，近似的に距離を計算するのが直積量子化による距離計算である．本システムでは[6]中で紹介されている非対称型の距離計算（Asymetric Distance Computation=ADC）を用いる.

4. 類似プレイリスト探索システムの構築

本章では類似プレイリスト推薦システムで利用する推薦モデルと，その実装例を紹介する.

大まかな方針としてプレイリストの低次元特徴量の幾何学的な探索により網羅率の向上を図り，また分散表現を用いたリアルタイムな特徴量計算と近似最近傍探索アルゴリズムを用いて，結果反映までの所要時間の短縮を図るものである.

4.1 プレイリストの特徴量モデル

AWAでは利用者による再生楽曲の系列データを保有している．この再生楽曲の系列データについても単語の系列データと同様の「楽曲の個性は前後に再生される楽曲で特徴づけられる」という仮説をとる．この仮説のもとで，各楽曲に対して，Skip-gramによる分散表現を獲得しておく．そして，プレイリストの特徴量としてはそのプレイリストに含まれる楽曲の分散表現の重心を用いる.

楽曲の分散表現については事前に学習することが可能である．したがってプレイリストの作成や更新イベントの度に簡単なベクトルの演算でプレイリストの低次元特徴量を得ることができる.

また分散表現を学習する際のオフライン評価は定量的な指標を作成することが難しい．無論，機械学習モデルである以上，最小化すべき損失関数が存在する．しかしながら学習した分散表現を用いて具体的なアプリケーションやPoCを作成してみると，その損失の優劣と，学習結果を利用したアプリケーションが利用者にとって魅力的かどうかとは，必ずしも一致しない事例が複数例確認された.

損失関数が小さいがアプリケーションとして魅力的でない例：

学習に際しては，特徴量として使う楽曲に対して再生数などを根拠に絞り込みを行っている．この絞り込みの条件を厳しくすると，学習データの次元が小さくなり，低い損失のモデルを作成することが出来る．一方で，モデル化されてない楽曲を多く含むプレイリストにおいては，そのプレイリストの特徴を十分に反映した推薦が行えず，アプリケーションとしては魅力が劣るものになってしまう.

そこで我々は機械学習自体のパラメタチューニングと並行して，関係者向けのデモを作成した[**図5**].

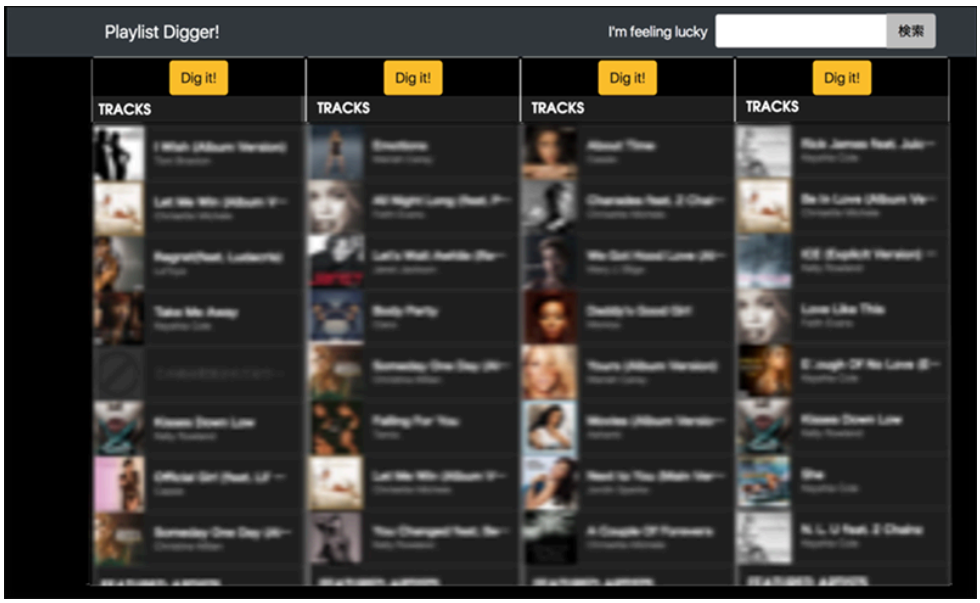

図5　関連プレイリストモデル確認用のデモ画面

このデモは1番左の列にクエリとなるプレイリストを表示し，右側3列には探索した類似プレイリストを表示している．上の黄色のボタンを押せば，そのプレイリストの類似プレイリストを表示することが可能で，アプリケーション上で期待される連続的な音楽体験をデモ上で擬似的に再現するものである．

複数のモデルにもとづくデモを作成し，関係者からのヒアリングした内容をもとに，実装するモデルを選択した．

また使用モデル決定後も，実際のアプリ上で全ユーザに対するリリースに先立って，開発関係者向けのリリースから始まり，全関係者向けのリリースまで，内部向けに段階的に機能を開放した．その間で，該当機能利用者からのヒアリングやフィードバックを元に次節で述べるようなモデルの微調整を行った．

また実際のサービス利用者に対しても5%の利用者への適用から始まり段階的に全利用者に適用範囲を広げた．これは利用者からのフィードバックを旧手法と比較するのと同時に，必要なシステム負荷を見積る，もしくは未知のバグを含んでいた場合の障害の範囲を制御するなどのカナリアリリース的な意味合いも大きい．

4.2　意向ベースフィルタリング

AWAでは1100万を超えるユーザ作成プレイリストのデータを保有している．距離の近似計算について直積量子化により計算量を削減しているが，距離の計算対象の絞り込みによっても計算量を削減することが期待される．そのため，推薦対象となるプレイリストについて事前に何らかの意味で絞り込みを行いたい．

また，上記で作成したデモを通じていくつか，アプリケーション設計上の意向を正しく反映していない例が見つかった．そういった課題に対しては，単純なルールベースの処理により推薦対象を選定した．この節では本推薦システム実装時点で考慮されたいくつかのルールを紹介する

- アーティストがプレイリスト内で一定以上重複しない．

本システムでは「新しい楽曲との出会い」を重要視している．そのため推薦するプレイリストは適度な多様性を持つことが望ましい．本システムではプレイリスト内の楽曲の多様性を担保する基準として，含まれる楽曲のアーティストが一定以上重複しないことをルールとして採用した．

- 8曲から構成される

もう1つのプレイリストの多様性の基準として，プレイリストを構成する楽曲数も考慮した．筆者らは，楽曲数が多いものほど多様性の高いプレイリストであると考え，推薦されるプレイリストを最大の8曲から構成されるものに限定している．

- 十分な長さのタイトルと説明文が付与されている

利用者が次に再生するプレイリストをより決定しやすくするために，そのプレイリストがどのようなプレイリストであるかを再生以前に利用者に伝えられることが望ましい．そのような観点から推薦するプレイリストには，説明文やプレイリストのタイトルの充実度に一定の基準を設けてフィルタリングを行っている．

- 先頭3曲を含むアルバムが重複しない

プレイリストの内容を事前に伝えるもう1つの手段として，AWAでは先頭3曲に対応するアートワークを組み合わせたアイキャッチ画像を表示している[**図6**]．このアイキャッチ画像により多くの情報を持たせるという観点で先頭3曲のアルバムが重複するものを推薦対象から除外している．

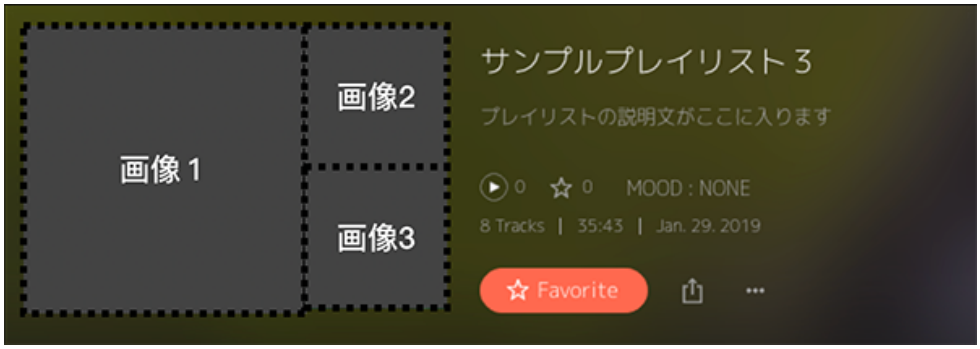

図6　アートワークを組み合わせたアイキャッチ画像

- できるだけ最近作成更新された

推薦対象となるプレイリストが増え続けると次第に，新しい楽曲を含むプレイリストの比率が減少する．一方で，筆者らはレコード業界の継続的な発展には，新しい作品の台頭が欠かせない要素と考えている．したがって，本システムではプレイリスト作成・更新日時にしきい値を定め，直近作成されたプレイリストのみに推薦対象を限定している．

このようなルールベースのフィルタにより推薦対象を約数十万にまで絞り込み，計算量の削減と品質の向上を実現している．

4.3　システム構成

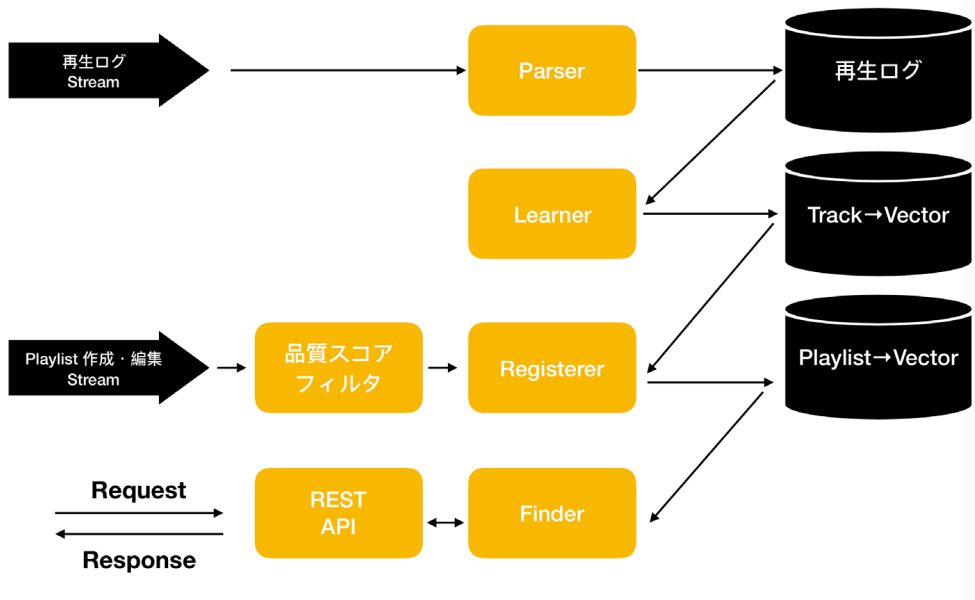

図7　関連プレイリスト探索システムのシステム構成

この章ではこれまで紹介した要素技術を組み合わせた関連プレイリスト探索システムの概要について説明する．本システムはユーザによる楽曲再生ログ，プレイリスト作成・更新ログの2種類のストリームデータを処理するParser, Learner, Filter, Registerer, Finderの5つのマイクロサービスからなる．それぞれの概要は以下の通りである．

- Parser：学習データの前処理

Parserでは利用者の楽曲再生ログを学習フレームワークが学習しやすい形式に処理してストレージに格納する役割を果たす．各データに施す処理は大まかに2つある．

1点目は必要な情報の選択で，利用者の楽曲再生ログにはクライアントの状態を表すさまざまな情報が格納されている．このログから利用者ID，楽曲ID，時刻のみを取り出す処理を行う．

2点目はIDを整数値に変換することである．IDは一般にランダムな文字列だが，学習させる際には1-of-k表現のための整数値への対応が必要になる．このIDと整数値の一意な対応を作成しながら変換することが2つ目の処理である[**図8**]．

```
'8973169291764 64ebc9ad085f31e7284' => 0
'b026324c6904b2a9cb4b88d6d61c81d1' => 1
'26ab0db90d72e28ad0ba1e22ee510510' => 2
```

図8　IDと整数の対応のイメージ

- Learner：skip-gramによる分散表現の学習

Learnerでは格納された再生系列データ[**図9**]を用いてskip-gramの学習を行う．学習が終了するとこれを再び楽曲IDをKey，分散表現をvalueとするKey-Value Store（以下KVS）のテーブルに格納する．

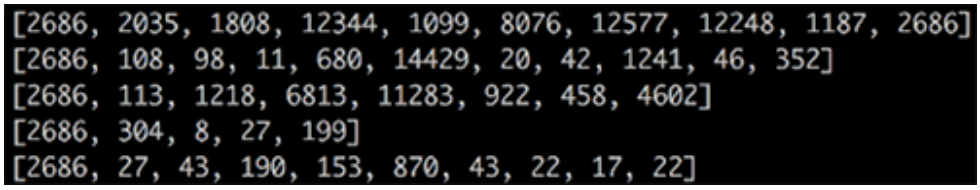

図9　学習用に処理された再生系列データの例

- Filter：意向ベースのフィルタ処理

Filterではプレイリスト作成・更新ログを受け取りそのログの内容を評価して，そのプレイリストが推薦対象となるかを判定してそのフラグを付与する．

- Registerer：プレイリスト特徴量の計算

RegistererではFilterが処理した推薦対象フラグ付きのプレイリスト作成更新ログから，プレイリストIDとそれを構成する楽曲IDの列と推薦対象フラグを取り出す．次に楽曲IDの列からそれぞれの分散表現を取得しその重心を計算する．この重心をプレイリストの特徴量としてプレイリストIDをKey，この特徴量と推薦対象フラグをValueとするKVSのテーブルに格納する．

- Finder：特徴量の近似最近傍探索

FinderはAPIからプレイリストIDを受け取り関連プレイリストの列を返すサービスである．起動時に直近一定期間に作成・更新されたプレイリストのうち検索対象となるプレイリストの特徴量を先のテーブルから読み込み，直積量子化に用いるためのクラスタリング処理を行い各クラスタの重心（＝辞書）をメモリ上に格納する．その後数十秒程度の一定間隔で直近の更新分をも量子化した上でメモリ上に格納する．

プレイリストIDを受け取るとその特徴量を先のプレイリスト特徴量テーブルから取り出し，あらかじめ量子化した推薦対象リストから近似的な近傍プレイリストを，要求された数だけ返却する．

5. 評価

提案法を5%の利用者に適用し1カ月のA/Bテストを行った（現在は全利用者に適用済み）．その結果以下に挙げる網羅率，反映までの時間差の2項目について大幅な改善がみられた．また実際の利用者への適用を通じて，関連プレイリストの利用率についても改善が見られた．

5.1 網羅率

検索元のプレイリストに対して類似プレイリストが存在する比率，すなわち網羅率を比較する．

従来では楽曲情報の厳密な紐付けを元に類似プレイリストを探索していたため，一部のプレイリストには類似プレイリストが割り当てられなかった．本システムでは分散表現を用いた幾何学的な近傍探索を行うことで， 楽曲の類似性から類似プレイリスト探索が行え，そのため一定数の再生実績があるプレイリストが1つでも含まれていれば類似プレイリストの探索が行える．この

ことにより従来およそ64%であった網羅率が現在ではおよそ98%に改善した．これは再生実績のない楽曲のみからなるなどの，例外的なプレイリストを除くおよそすべてのプレイリストを網羅していることに匹敵する．

5.2　反映待ち時間

ここでは，類似プレイリストが計算されるまでの最大の時間差を比較する．従来の反映待ち時間はバッチ間隔と集計の実行時間の和で，最大で24時間程度要していた．一方本システムではRegisterによる処理時間＋Finderによる近似最近傍探索の処理時間の和となり，負荷ピーク時においても1000ms程度のほぼリアルタイムな推薦を実現している．

5.3　関連プレイリスト利用率

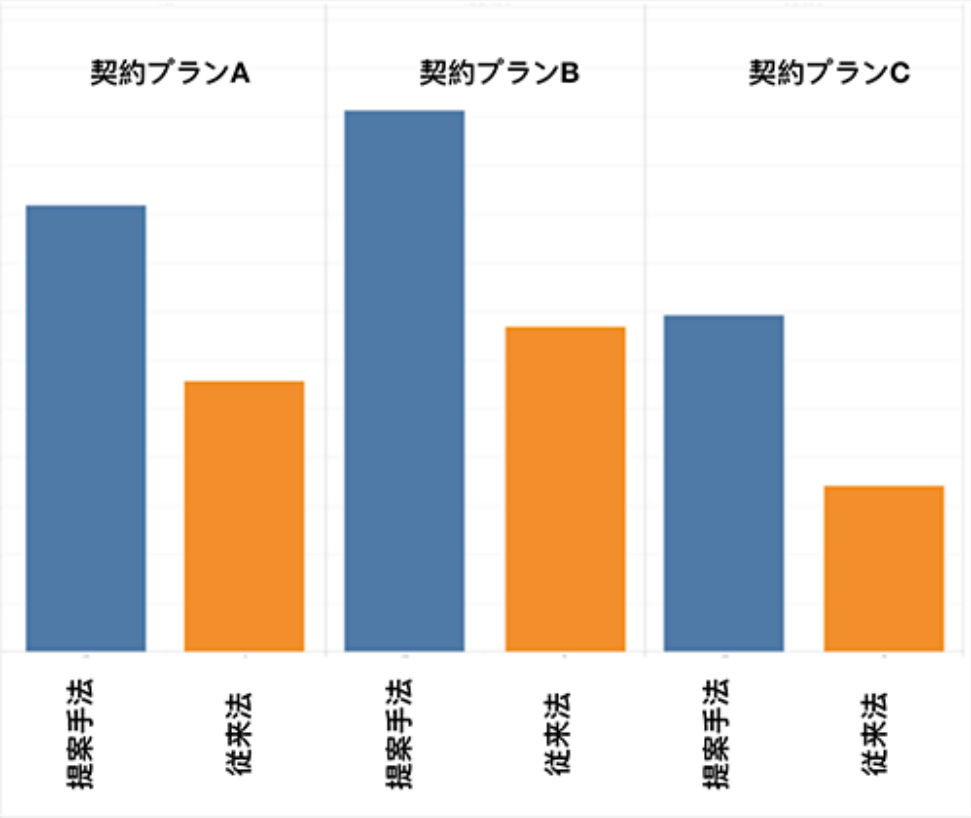

図10　利用者による関連プレイリスト経由の再生開始比率

AWAでは各機能を通じた新しい楽曲との出会いを重要視している．このことを測る尺度としてここでは対象機能を通じて楽曲を再生開始した比率を比較した．契約形態に応じて利用動向が大きく異なることがわかっているため契約形態毎にその比率を比較したところ，各料金プランで2倍程度の比率のユーザが対象機能経由で楽曲を再生したことが確認された．

6. まとめと今後の課題

本システムの実装と実験を通じて自然言語処理分野で実績のあるskip-gramモデルが，楽曲の再生系列データに対しても有効な特徴量を獲得できること，そしてプレイリスト，つまり楽曲の集合においてその分散表現の演算結果がプレイリストの特徴量として機能することを確認した．また従来法との比較を通じて，いずれの指標においてもユーザ体験を向上させていることを確認した．またシステム設計の観点で見れば，特定の学習モデルに依存しない汎用的な学習データの

前処理を行うParser，特定のアプリケーションに依存しない学習モデルであるLearnerおよびRegisterer，また汎用的なベクトルの近似最近傍探索を行うFinderと複数のサービスからなる再利用性を意識したアプリケーションの構築事例を紹介した．

また，今後の課題としては，モデルの継続的な更新やライフサイクル管理がある．総合的なアプリケーションとしての魅力に対して有力なオフライン評価がない今，学習データやモデルの更新には，利用者や管理者のフィードバックを取り入れる必要があると考えられる．本システムはその意味でモデル更新の完全なる自動化には至っていない．[7]や[8]などで人間のフィードバックを取り入れた継続的デリバリーの仕組みが提案・実装されはじめており，また学術的な観点では，AI応用システムにおいて旧来のソフトウェア工学を適用する際の課題に対する提言として[9]がある．今後，本システムを含む多くの機械学習を応用したアプリケーションおいても開発プロセスの改善やモデルの更新自動化・高性能化および品質向上を通した，利用者の音楽体験の向上に取り組んでいきたい．

参考文献

1）IPFI, Global Music Report 2018 : ANNUAL STATE OF THE INDUSTRY, https://www.ifpi.org/downloads/GMR2018.pdf
2）一般社団法人日本レコード協会：Statistics Trends – the Recording Industry in Japan, https://www.riaj.or.jp/f/pdf/issue/industry/RIAJ2018.pdf
3）Cosley, D. et al. : Is Seeing Believing? : How Recommender System Interfaces Affect Users' Opinions, Proceedings of The SIGCHI Conference on Human Factors in Computing Systems, ACM (2003).
4）Agarwal, D. k. and Chen, B-C. : Statistical Methods for Recommender Systems, Cambridge University Press (2016).
5）Mikolov, T. et al. : Distributed Representations of Words and Phrases and Their Compositionality, Advances in Neural Information Processing Systems (2013).
6）Jégou, H., Douze, M. and Schmid, C. : Product Quantization for Nearest Neighbor Search, IEEE Transactions on Pattern Analysis and Machine Intelligence 33.1, 117-128 (2011).
7）Baylor, D. et al. : Tfx : A Tensorflow-based Production-scale Machine Learning Platform, Proceedings of The 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM (2017).
8）Spinnaker. io : Spinnaker Continuous Delivery for Enterprise, Fast, Safe, Repeatable Deployments, https://www.spinnaker.io/
9）JST研究開発戦略センター システム・情報科学技術ユニット：AI応用システムの安全性・信頼性を確保する新世代ソフトウェア工学の確立, https://www.jst.go.jp/crds/pdf/2018/SP/CRDS-FY2018-SP-03.pdf

水上 裕貴（非会員）mizukami_hiroki@cyberagent.co.jp
サプライチェーンで物流事業部，店舗の販売管理を経て2014年より現職．秋葉原ラボにてAWAのデータ利活用，機械学習モデル，およびシステムの設計・開発・運用・管理に従事．情報処理学会ビッグデータ研究グループ運営委員．

福田 鉄也（正会員）fukuda_tetsuya@cyberagent.co.jp
2016年（株）サイバーエージェント中途入社．以後，秋葉原ラボで推薦システムの開発と運用に従事．

山下 剛史（非会員）yamashita_takeshi@cyberagent.co.jp

通信事業者, 小規模SIerを経て2011年より現職．主にメディア系新規サービス開発のサーバサイドを担当．2015年4月に株式会社AWAに参画.

Connolly, Juhani（非会員）juhani_connolly@cyberagent.co.jp
前職にてHbaseなど分散システムの開発に携わったのち，2013年サイバーエージェント入社．Apache Flumeのコミッター兼PMCメンバーになりその後推薦システムの開発に視野を広げる.

武内 慎（非会員）takeuchi_makoto@cyberagent.co.jp
前職の通信事業者にて携帯電話端末開発に携わったのち，2015年サイバーエージェント入社．現在，秋葉原ラボにてAWAのデータ分析業務に従事.

數見 拓朗（非会員）kazumi_takuro@cyberagent.co.jp
2013年，株式会社サイバーエージェント入社．2017年,大阪大学大学院経済学研究科博士後期課程修了．博士（経済学）．主にアメブロ，広告で利用する機械学習システムの構築と運用や分析業務などを担当.

福田 一郎（非会員）fukuda_ichiro@cyberagent.co.jp
2006年，東京大学工学部システム創成学科卒業．2008，年東京大学大学院工学系研究科精密機械工学専攻修士課程修了．同年より，（株）サイバーエージェント勤務．AmebaPiggの開発，大規模データ処理基盤Patriotの設計，開発，運用等に従事．現在，秋葉原ラボ研究室長.

採録決定：2018年12月18日
編集担当：福島 俊一（科学技術振興機構（JST））