Vol.2018-NL-236 No.3
2018/7/9

ArtistVector: Web文書分散表現によるアーティスト特徴量獲得

篠井 暖[1,a)]

概要：
音楽ファンにとって，自分の好みに合う新たな楽曲に出会えることは大きな喜びである．近年の定額制音楽配信サービスの普及で膨大な量の楽曲を聴取可能になった一方で，数百万〜数千万曲という規模の楽曲の中から好みに合う楽曲をひとつひとつ試聴しながら探すのはもはや不可能になっており，リスナーの好みに合う楽曲を簡単に検索可能な仕組みが必要になっている．本稿では，楽曲を探す際の有力な手がかりとしてアーティスト情報に着目し，アーティストの特徴抽出手法について検討する．アーティストに関する情報を記述した文書の潜在表現を学習することによりアーティストのベクトル表現 (ArtistVector) を獲得し，クエリアーティストと類似するアーティストを検索可能にする手法を提案する．アーティストを特徴づける文書として，(1) アーティスト自身の説明を記述した文書 と (2) リスナーからのアーティストの評価を記述した文書 が重要になると考え，両者に対応する文書として Wikipedia 記事と Web レビュー記事を学習データに利用して ArtistVector を獲得した．得られた ArtistVector に対しジャンル分類タスクによる評価を行い，データセットおよび手法の有効性を検証した．また ArtistVector を UMAP により 2 次元平面上に可視化し，コンテキストに基づく関係性を反映した類似アーティストが得られていることを確認した．

ArtistVector: Artist Feature Extraction with Web Document Embeddings

Dan Sasai[1,a)]

1. はじめに

音楽ファンにとって，自分の好みに合う新たな音楽に出会えることは大きな喜びである．近年の定額制音楽配信サービスの普及 [1][2][3] により膨大な量の楽曲を聴取可能になった一方で，数百万〜数千万曲といった規模の楽曲から自分の好みに合う楽曲をひとつひとつ試聴しながら探すのはもはや不可能になっている．また，音楽配信サービス事業者側の立場としては，大量にある楽曲コンテンツのうちランキング上位などのごく一部の曲しか聴かれないような状況では楽曲コンテンツの価値をユーザに十分訴求しきれておらず，ユーザの好みに合う様々な楽曲を提示したいという要求がある．ユーザの好みに合う楽曲を推薦するシステムを実現することでリスナーと配信事業者双方の課題を解決することが可能と考えられる．

従来の音楽推薦システムに関する研究では，楽曲単位で推薦を行う手法が数多く提案されてきた．[4][5] 一方，音楽作品はアーティスト単位での作品としてパッケージされているものが大多数である（特にポピュラー音楽）．リスナーが新しい音楽を探す際にも、アーティストの名称で検索するなどアーティスト情報を手がかりにするケースが多い．このような状況を鑑み，本研究では楽曲単位ではなくアーティスト単位で推薦を行う手法を考案する．

推薦システムは協調フィルタリング（CF）による手法 [6] と内容ベースの手法 [4]，そして両者を併用したハイブリッド手法 [7][8] の 3 種類に大別される．CF ではアイテムとユーザ評価の行列を用い，アイテムもしくはユーザ評価の類似度により推薦を行う．行列分解による手法が state-of-

[1] ヤマハ株式会社
Yamaha Corporation
[a)] dan.sasai@music.yamaha.com

the-art として知られている [6]．CF の欠点として，コールドスタート問題と知名度の低いアーティストが推薦されない問題が挙げられる．

コールドスタート問題　コールドスタート問題は 2 種類に分類可能である．1 つは，新しく推薦システムを利用し始めたユーザに対し適切な推薦を行うことが難しいという問題であり，もう 1 つは，システムに新しく追加されたコンテンツを推薦することが難しいという問題である．前者について，新しくシステムを使用し始めたユーザは十分な評価情報を持っておらず他のユーザとの類似性を判定できないため信頼度の高い推薦を行うことが難しい．後者について，発売されたばかりの新曲などでまだユーザの評価が蓄積されていない楽曲コンテンツを適切に推薦することは難しい．特に楽曲配信サービスへの応用を考えた時に後者の問題は無視できないと考える．というのも，配信事業者や音楽レーベルの立場としては新曲をいち早く届けたいと考えるのは自然で，ユーザの評価が蓄積されるまで待っていると販売の機会損失に繋がる可能性があるからである．

知名度の低いアーティストが推薦されない問題　CF において，少数派の嗜好パターンをもつユーザはごく少数の類似するユーザしか見つけられないので，適切な推薦を行えない可能性がある．たとえば，ポップスのようなリスナーの多いジャンルの推薦は効果的に行えるが，マニアックなジャンルの推薦が効果的に行えないといった問題が起こりうる．

一方，内容ベースの手法では楽曲などのコンテンツがあれば推薦を行うことが可能で，前述のコールドスタート問題および知名度の低いアーティストが推薦されない問題への対処が可能である．従来は MFCC（Mel Frequency Cepstral Coefficients）に代表される楽曲データの音響特徴量を用いて類似度を計算し，その類似度に基づき推薦を行う手法 [4] が主流であったが，近年ではソーシャルタグやメタデータを併用した研究 [9] も行われている．これらの手法の問題点として「アーティスト間のコンテキストに基づく関係性を反映した結果を得るのが困難」という点が挙げられる．音響特徴量に基づく手法は楽曲からの特徴抽出により得られた音色，コード，リズムといった音楽的情報に基づき推薦を行うため，音楽的な内容が類似するアーティストを推薦することには長けているが，例えば「アーティスト A とアーティスト B はよくフェスで共演しており親交がある」などのコンテキストに基づく情報は反映することができない．一方，メタデータにはジャンルや年代といった情報は記述されているが，アーティストごとの関係性に相当する情報は記述されていない．CF ではこれらのコンテキストに基づく関係性を間接的に反映することが可能と考えられる．しかし，前述のコールドスタート問題および知名度の低いアーティストが推薦されない問題により，特に知名度の低いアーティストに対してはこれらの情報を適切に反映することができない．そこで，本研究では，(1) CF におけるユーザ評価が蓄積されていないと推薦が行えない問題と (2) 内容ベース手法におけるコンテキスト情報が捉えられない問題の両方を解決するために，コンテキスト情報を捉えることが可能な内容ベースの手法を提案する．具体的には，アーティストのコンテキストに基づく関係性を捉えるためのリソースとしてアーティストに関する Web 上の記事に着目する．例えば Wikipedia のアーティストの記事にはバイオグラフィに相当する情報が記載されており，コンテキストに基づく関係性を捉えるのに有力なデータセットとなると期待される．アーティスト記事のデータセットを元に自然言語処理のアプローチによってアーティストの特徴量を抽出することで，従来の音響特徴量やメタデータでは捉えられなかった関係性を捉えられる特徴量を獲得可能になると考える．

上記の仮説に基づき，本研究では Web 上のアーティストに関する記事をもとにアーティストの特徴量を抽出する手法を提案する．提案手法は以下の 2 段階にて構成される．

(1) アーティストに関する記事を Web から収集しデータセット（アーティスト記事データセット）構築
(2) アーティスト記事データセットより各アーティストを表現する特徴量の獲得

を行うことでアーティストの特徴量の獲得を行う．提案手法にて獲得されるアーティストの特徴量を本研究では ArtistVector と呼称する．

本稿の残りの構成は以下である．2 章にてアーティスト記事データセットの構築について説明し，3 章にて ArtistVector の獲得法について説明する．4 章にて評価実験について説明し，5 章にてまとめを行う．

2. アーティスト記事データセットの構築

アーティストを特徴づける情報にはジャンル，年代などのメタデータや協調フィルタリングで用いられるユーザのアーティスト評価情報，あるいは楽曲データから抽出した音響特徴量など様々なものが考えられるが，本研究では

(1) アーティスト自身を説明するテキスト記事
(2) リスナーのアーティストに対する評価を記述したテキスト記事

の 2 種類の文書データを用いてアーティストの特徴をモデル化する．アーティスト自身の説明を記述した文書（Biography）とリスナーからのアーティストに対する評価を記述した文書（Review）の双方を利用することにより，アーティストの特徴を捉えられると考える．具体的には，(1) には各アーティストの Wikipedia 記事 [10] を用い，(2) には Google 検索 [11] により検索したアーティストの感想記事を用いる．

アーティストの Wikipedia 記事　Wikipedia よりアーティスト名の項目を抽出し，その本文をアーティストを説明する記事として用いた．

アーティストの感想記事　Google 検索で「アーティスト名+(感想 or レビュー)」というクエリで検索を行い，上位 30 位の検索結果の記事をアーティストの感想記事として用いた．なお，Amazon，楽天などの商品情報ページといった感想記事とは考えにくいサイトの記事は除外してある．

1000 アーティスト分の上記テキストデータを収集し，アーティスト記事データセットを構築した．

3. ArtistVector(アーティストごとの記事の潜在表現) の学習

Biography と Review それぞれに対し，文書単位での潜在表現を学習し，Biography の潜在表現（BiographyVector）と Review の潜在表現（ReviewVector）を得る．さらに，両者を結合することにより ArtistVector を獲得する．ArtistVector の獲得フローを図 1 に示す．

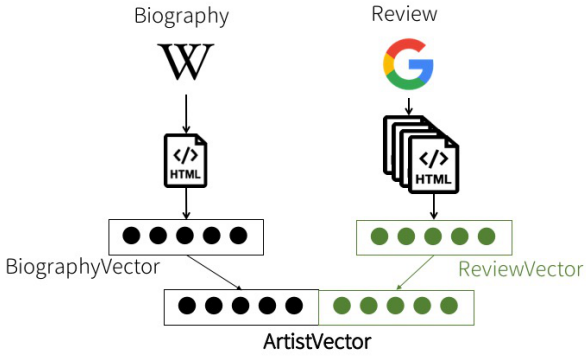

図 1　ArtistVector 獲得フロー

文書の潜在表現獲得には Paragraph Vector[13] を用いる．ここでは，Paragraph Vector とその前提となっている word2vec[14] について説明する．

3.1 word2vec

word2vec は，「同じ文脈で出現する単語は同じ意味を持つ」という分布仮説 [15] に基づき，文中の周辺単語を用いて単語の意味を表現する固定サイズの潜在表現ベクトルを獲得するニューラルネットワークで，Skip-gram と Continuous Bag-of-words（CBOW）の 2 種類のモデルが存在する．Skip-gram モデルのアーキテクチャを図 2 に示す．

Skip-gram は現在の単語 $w(t)$ から周辺単語 $w(t-c),...,w(t-1),w(t+1),...,w(t+c)$ を予測するモデルである．

次に，CBOW モデルのアーキテクチャを図 3 に示す．

CBOW は周辺単語 $w(t-c),...,w(t-1),w(t+1),...,w(t+c)$ から現在の単語 $w(t)$ を予測するモデルである．

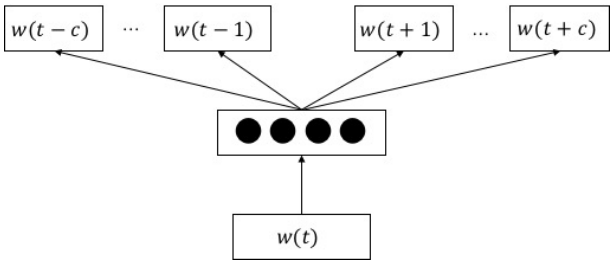

図 2　Skip-gram

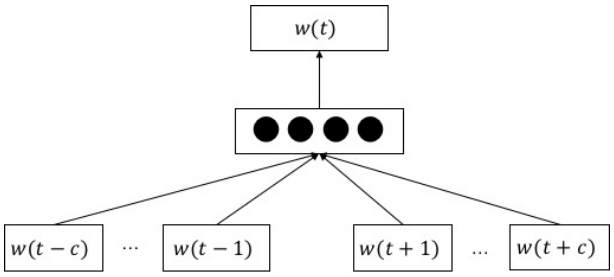

図 3　CBOW

3.2 Paragraph Vector

Paragraph Vector は word2vec を文章単位に拡張したもので，文章の ID から計算されるユニット D を word2vec のモデルに導入することで文章の潜在表現ベクトルを獲得する．word2vec と同様 2 種類のモデルが存在し，文章中の単語の語順を考慮する Distributed memory model（PV-DM）と語順を考慮しない Distributed Bag-of-Words（PV-DBOW）が存在する．PV-DM モデルのアーキテクチャを図 4 に示す．

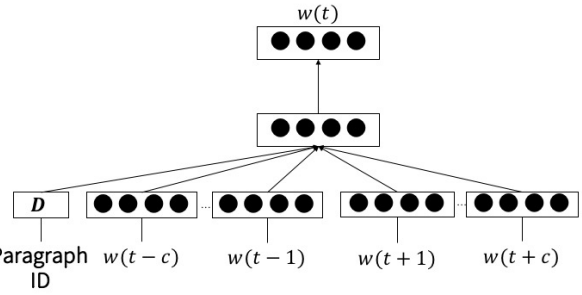

図 4　PV-DM

PV-DM は周辺単語に加え D を入力として現在の単語 $w(t)$ を予測するモデルで，CBOW の拡張となっている．

次に，PV-DBOW モデルのアーキテクチャを図 5 に示す．

PV-DBOW は D を入力として文章に含まれる単語集合を予測するモデルで，Skip-gram の拡張となっている．

Le らによると多くのタスクで PV-DM のみでも品質の良い分散表現が得られるが，PV-DBOW と併用することでさらに頑健な分散表現が得られることが報告されてい

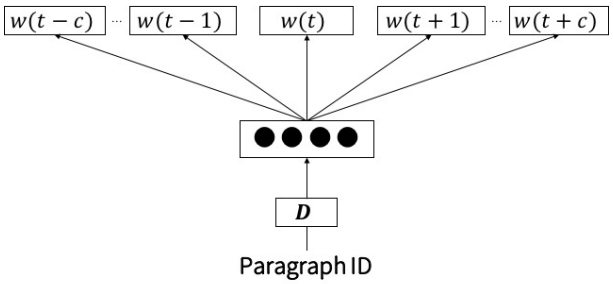

図 5　PV-DBOW

る [13].

本研究では，1 文書単位の Paragraph Vector を学習することで文書単位の潜在表現ベクトルを得る．

3.3 BiographyVector

本節では BiographyVector 獲得方法について述べる．日本語版 Wikipedia 全文を学習データとして Paragraph Vector の学習を行い，その後，各アーティストの Wikipedia 文書の潜在表現ベクトルを抽出した．ParagraphVector の次元数は 400，文脈の窓長は 8 とした．辞書は mecab-ipadic-neologd[16] を使用した．

3.4 ReviewVector

本節では ReviewVector 獲得方法について述べる．日本語版 Wikipedia 全文に収集した感想記事データを加えたデータセットを学習データとして Paragraph Vector の学習を行い，その後，各アーティストのそれぞれの感想記事ごとの潜在表現ベクトルを抽出した．さらに，得られた感想記事の潜在表現ベクトルをアーティストごとに平均し，その結果をアーティストの感想記事の潜在表現ベクトルとした．ParagraphVector の次元数，文脈の窓長，辞書は BiographyVector と同様である．

3.5 Wikipedia に項目が存在しないアーティストの BiographyVector 獲得方法

デビューして間もないアーティストや知名度の低いアーティストの場合，Wikipedia にそのアーティストの項目が存在しない場合がある．その場合は，以下の手順により BiographyVector を近似的に得る．

(1) ReviewVector の類似度を総当たり計算する
(2) アーティスト A の ReviewVector に最も類似度の高い ReviewVector を持つアーティスト B を抽出する
(3) (2) で得たアーティスト B の BiographyVector をアーティスト A の BiographyVector として用いる

4. 評価実験

本章では評価実験について述べる．ArtistVector のアーティスト特徴量としての品質を評価するため，評価実験は以下の 2 種類の評価・分析を行った．

(1) アーティストジャンル分類タスクによる ArtistVector のアーティスト特徴量としての品質評価
(2) ArtistVector 可視化による類似アーティスト分析

4.1 データセット

アーティスト記事データセットは，ライブファンズ株式会社 [17] より提供いただいた 2016 年 7 月〜2017 年 7 月のライブ人気アーティスト Top1000 のアーティスト名およびジャンル名のリストをもとに Wikipedia 記事および感想記事をクロールすることで構築した．構築したアーティスト記事データセットの書誌情報を表 1 に示す．

表 1　アーティスト記事データセット

項目	数
アーティスト	1000
Wikipedia 記事数	916
レビュー記事数	29430

また，ジャンルごとのアーティスト数を表 2 に示す．

表 2　ジャンルごとのアーティスト数

ジャンル	アーティスト
ロック	382
ポップス	350
オルタナティブ/パンク	129
アニメ/ゲーム/声優	77
アイドル	75
ヴィジュアル系	59
エレクトロニカ/ダンス	34
R&B/ソウル	28
K-POP	28
ヒップホップ/ラップ	25
ハードロック/メタル	25
フォーク/ニューミュージック	24
no genre	17
ジャズ/フュージョン	15
レゲエ	6
歌謡曲	4
イージーリスニング	4
ブルース	3
その他	2
日本伝統音楽/民謡	2
クラシック	1

なお，今回使用したジャンルラベルは 1 アーティストに複数ジャンルが付与されることを許容しているラベルなので，表 2 におけるアーティスト数の合計と対象アーティスト数（1000）は必ずしも一致しないことに注意されたい．

4.2 アーティストジャンル分類タスクによる ArtistVector のアーティスト特徴量としての品質評価

アーティストの特徴量からジャンルを識別するアーティストジャンル分類タスクにより ArtistVector の性能を評価する．以下の手順でアーティストの特徴量を入力としたジャンル識別器を構築し，ジャンル識別の正解率を評価した．

(1) アーティスト記事データセットより ArtistVector を獲得
(2) ArtistVector を特徴量，ジャンルラベルを正解ラベルとする Support Vector Machine（SVM）を学習
(3) 10-fold cross validation によりジャンル正解率を評価

ArtistVector 獲得において，使用するデータの種類 と 文書分散表現獲得手法 それぞれについて複数の手法を試し比較を行った．使用するデータの種類は（1）Biography のみ（Bio）（2）Review のみ（Rev）（3）Biography および Review（BioRev）の 3 種類を比較した．文書分散表現獲得手法は（1）word2vec の文書全体での平均（W2V）（2）Paragraph Vector の PV-DBOW モデル（PV-DBOW）、(3) Paragraph Vector の PV-DM モデル（PV-DM）(4) PV-DBOW と PV-DM を連結したベクトル（PV-BOTH）の 4 種類を比較した．word2vec の次元数は 100，窓長は 8 とした．Paragraph Vector のパラメータは 3.3 節，3.4 節に記載したものと同一である．

評価結果を表 3 に示す．

表 3　アーティストジャンル分類 評価結果

	W2V	PV-DBOW	PV-DM	PV-BOTH
Bio	0.3271	0.6252	0.5946	**0.6525**
Rev	0.1784	0.6083	0.5372	**0.6218**
BioRev	0.4123	**0.6853**	0.6374	0.6370

まず使用するデータの種類については，Bio の方が Rev よりも分類精度が高い．さらに，両者を併用する BioRev を使用した場合に大きく分類精度が向上した．次に文書分散表現獲得手法については，PV-DBOW の方が PV-DM よりも性能が高い結果となった．PV-DM の性能が低くなった理由としては，今回用いたデータセットのアーティスト数が 916 と少ないため，語順を考慮する PV-DM では ReviewVector をうまく学習できていない可能性がある点や，Wikipedia は表や箇条書きになっている部分も多いので語順を考慮することの利点が薄いと考えられる点が挙げられる．PV-BOTH は Bio あるいは Rev 単体の場合は最も性能が高いが，BioRev で Bio 単体よりも性能が下がる現象が起きている．また，文書分散表現のベースライン手法として用いた W2V は Paragraph Vector に比べ分類精度が著しく低い．これは，単純に文書に含まれる全単語の単語分散表現を平均しているため，機能語などの影響で文書の意味ベクトルが適切に得られていないためと考えられる．

総合的には，BioRev を使用した場合の PV-DBOW が最も性能が高い．この結果から，Biography と Review 両者を併用したアーティスト特徴量のモデル化は有効であることが示された．

4.3 ArtistVector 可視化による類似アーティスト分析

ArtistVector を 2 次元空間上に可視化することにより，アーティストの類似性を分析・評価する．次元圧縮には UMAP（Uniform Manifold Approximation and Projection）[18] を使用し，2 次元に圧縮した特徴量を散布図上にプロットすることにより可視化を行った．可視化結果の全体図を図 6 に示す．

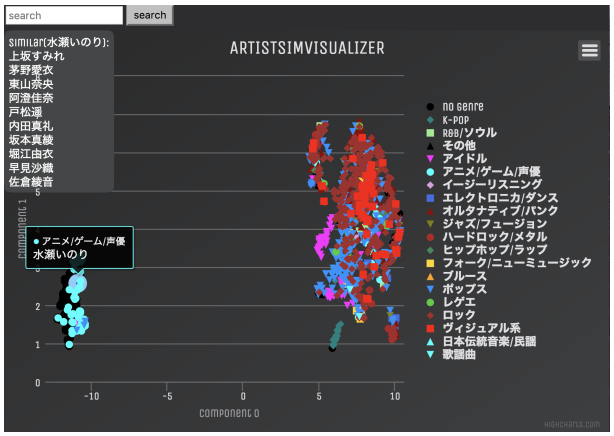

図 6　ArtistVector 可視化結果

UMAP は特徴空間上の類似度を確率分布として扱うことで次元圧縮を行う t-SNE[19] とほぼ同等の結果をより少ない計算量で，かつハイパーパラメータに依存せずに計算する手法である．UMAP で得られた低次元空間では特徴量が類似するものが近くに配置され，類似しないものが遠くに配置されるため，低次元空間上での距離を測ることにより類似アーティストの検索が可能である．

図 6 において各点はアーティストを示し，点の色はジャンルラベルを示す．また，UI 上の左上部のリストは選択したアーティストと類似するアーティストの Top10 を示す．ここで，類似度は低次元空間上のユークリッド距離の逆数で計算している．

まず全体を概観すると，ある程度ジャンルごとにクラスタが形成されていることがわかる．さらに，より細かく見ていくといくつか興味深いクラスタが形成されている．可視化結果のうち，アイドルが多く配置されている領域を図 7,8 に示す．

マゼンタ色の点はジャンル：アイドルのアーティストを示しているが，アイドルの中でも男性アイドルと女性アイドルそれぞれで別個のクラスタが形成されている．

また，ロック系アーティストが多く配置されている領域を図 9 に示す．

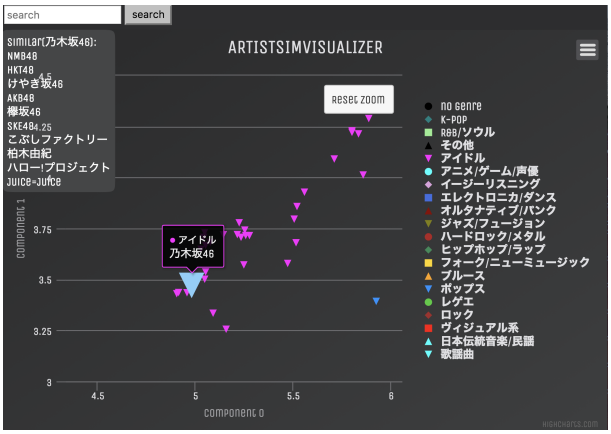

図 7　ArtistVector 可視化結果（女性アイドル）

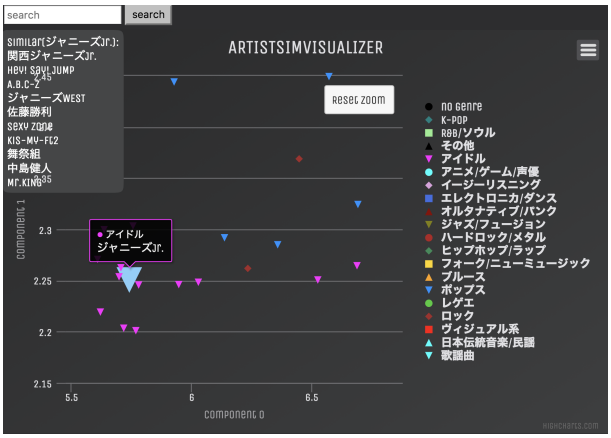

図 8　ArtistVector 可視化結果（男性アイドル）

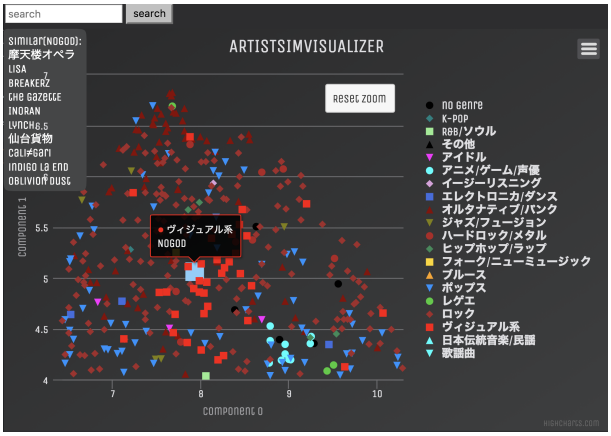

図 9　ArtistVector 可視化結果（ロック）

同じロックの中でもヴィジュアル系寄りのバンドとパンク/オルタナティブ寄りのバンドで別個のクラスタが形成されるなど，細かな音楽性の違いを反映したクラスタが形成されている．また，もう 1 つの重要な特徴としてレーベルが同じ，あるいはメンバーが共通しているなどのコンテキスト上の関係性を捉えた類似マップになっていることが挙げられる．その例として，X JAPAN の近傍の図を図 10 に示す．

X JAPAN のメンバーである YOSHIKI やほぼ同年代デ

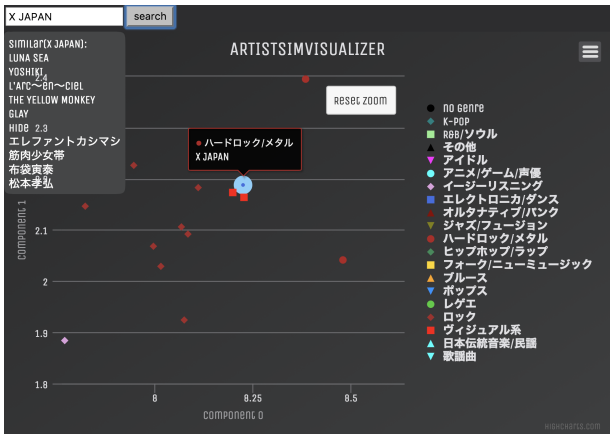

図 10　ArtistVector 可視化結果（X JAPAN 近傍）

ビュー，同じレーベル出身でギタリストが共通している LUNA SEA などが類似アーティストに出現している．このようなコンテキスト上の関係性は従来の音響特徴量などの内容ベースの手法では得ることが困難なもので，Web 上のテキストデータに記述されている知識を利用することの価値を示している．

最後に，BiographyVector と ReviewVector がそれぞれアーティストのどのような特徴を捉えているのかについて考察する．

BiographyVector のみで可視化を行った結果を図 11 に，ReviewVector のみで可視化を行った結果を図 12 に示す．

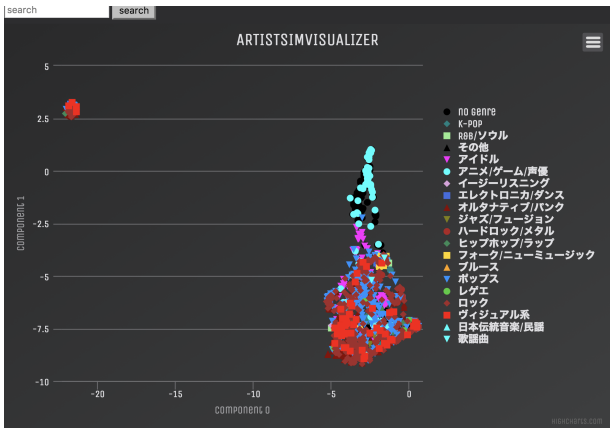

図 11　BiographyVector 可視化結果

図 11 より，BiographyVector はジャンルごとのクラスタはあまり明確に捉えられていないが，細かく見ていくと隣接するアーティスト間で先に述べたようなコンテキストによる関係性が顕れている．一方，図 12 より，ReviewVector はジャンルごとのクラスタを形成しており，主にアーティストのジャンルごとの大まかな特徴を獲得するのに寄与していると考えられる．この結果は，両者がそれぞれアーティストにまつわる異なる特徴を捉えており，両者を併用することでジャンル分類精度が向上することの裏付けになっている．

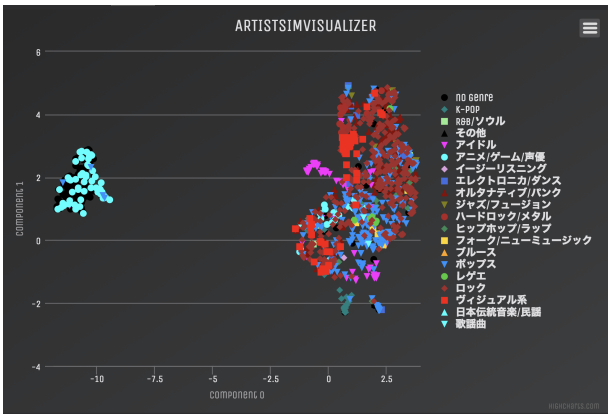

図 12　ReviewVector 可視化結果

5. まとめ

本稿ではコンテキストに基づく関係性を反映したアーティストの特徴量を獲得する手法 ArtistVector を提案した．Wikipedia 記事と Web の感想記事を収集することによりアーティスト記事データセットの構築を行い，そのデータセットに対し Paragraph Vector により BiographyVector と ReviewVector を学習し両者を連結することで ArtistVector を得た．アーティストジャンル分類タスクによる評価を実施し，BiographyVector と ReviewVector 両者を併用することで分類精度が向上することを確認した．また，UMAP によるアーティスト特徴量の可視化を行い，ArtistVector によりコンテキストに基づく関係性が捉えられていることを確認した．

今後の課題としては，主観評価実験の実施により人間の主観と合う類似アーティストが得られているかの定量的な評価の実施，およびアーティスト記事データセットの品質検証（アーティストに関連する感想記事が収集できているか，など）や使用するデータの追加（例えば感想記事として Twitter，Facebook など SNS 書き込みの利用など）による ArtistVector の品質向上などが挙げられる．

参考文献

[1] Spotify: 入手先 ⟨https://www.spotify.com/⟩.
[2] AWA: 入手先 ⟨https://awa.fm/⟩.
[3] mysound: 入手先 ⟨https://mysound.jp/⟩.
[4] B. Logan, “*Music Recommendation From Song Sets,*” In Proc. Intl. Society for Music Information Retrieval (ISMIR), 2004.
[5] A. Oord, S. Dieleman, and B. Schrauwen, “*Deep content-based music recommendation,*” In Advances in Neural Information Processing Systems 26, 2013.
[6] Y. Koren, R. Bell, and C. Volinsky, “*Matrix Factorization Techniques for Recommender Systems*,” Computer 42, 8 (2009), 4249. hps://doi.org/10.1109/ MC.2009.263 arXiv:ISSN 0018-9162.
[7] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, “*Hybrid Collaborative and Content-based Music Recommendation Using Probabilistic Model with Latent User Preferences,*” In Proc. Intl. Society for Music Information Retrieval (ISMIR), 2006.
[8] S. Oramas, O. Nieto, M. Sordo, and X. Serra, “*A Deep Multimodal Approach for Cold-start Music Recommendation,*” In Proceedings of DLRS 2017, Como, Italy, August 27, 2017, 6 pages.
[9] J. Bu, S. Tan, C. Chen, C. Wang, H. Wu, L. Zhang, and X. He, “*Music recommendation by unified hypergraph: combining social media information and music content,*” In MM ’10 Proceedings of the 18th ACM international conference on Multimedia, pages 391-400 (2010).
[10] Wikipedia: 入手先 ⟨https://ja.wikipedia.org/⟩.
[11] Google: 入手先 ⟨https://www.google.co.jp/⟩.
[12] C. Xu, N. C. Maddage, X. Shao, F. Cao, and Q. Tian, “*Musical Genre Classification Using Support Vector Machines,*” In Proc. ICASSP, Hong Kong, China, April 2003.
[13] Q. Le, and T. Mikolov, “*Distributed Representations of Sentences and Documents,*” Proceedings of the 31st International Conference on Machine Learning (2014).
[14] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “*Distributed Representations ofWords and Phrases and their Compositionality,*” In Advances in Neural Information Processing Systems, 2013.
[15] Z. Harris, “*Distributional structure,*” Word, 10(23): 146-162. （1954）.
[16] T. Sato, T. Hashimoto, and M. Okumura, “*Implementation of a word segmentation dictionary called mecab-ipadic-NEologd and study on how to use it effectively for information retrieval (in Japanese),*” Proceedings of the Twenty-three Annual Meeting of the Association for Natural Language Processing (2017).
[17] LiveFans: 入手先 ⟨http://www.livefans.jp/⟩.
[18] L. Mclnnes, and J. Healy, “*UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction,*” arXiv preprint arXiv:1802.03426.
[19] L. Maaten, G. Hinton, “*Visualizing Data using t-SNE,*” Journal of Machine Learning Research 9 (2008) 2579-2605.