

WWWからの楽曲情報と評判情報の獲得に基づく楽曲データベースの作成

Create the Music Database about Track Information and Reputation Information from World Wide Web

中島 洋樹*¹
Hiroyuki NAKASHIMA

岡田 将吾*¹
Shogo OKADA

新田 克己*¹
Katsumi NITTA

*¹ 東京工業大学大学院 総合理工学研究科

Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology

In this research, we build a database for recommendation system from text data related to music by World Wide Web (WWW). Our proposed database is composed of reputation information extracted from last.fm, and track information extracted from Wikipedia and has many kind of features which each music has. Reputation information has tag which annotation from last.fm user, and tag's weight which represent how many used in the track. Track information has producer name, album category and artist category. To evaluate whether our database is effective for recommendation system, we perform preliminary experiment with Contents based filtering for each track feature. Experimental results shows that increase

1. はじめに

ウェブの普及によって、高度情報化と情報発信が低コスト化し、大量のデータが常に生成されているため、情報検索では、情報過多問題が大きな課題となっている。この問題を解決する手段として情報推薦技術が研究されている。情報推薦技術には主に協調フィルタリングと、内容に基づくフィルタリングがある。協調フィルタリングは推薦要求者(以降、ターゲットユーザ)のプロファイルや、アイテムへの評価値を他のユーザと比較することによって嗜好が似ているユーザ(以降、ピアユーザ)を決定し、ピアユーザが高く評価するアイテムのなかでターゲットユーザがまだ評価していないものを推薦する。協調フィルタリングを利用している代表的なシステムにAmazon.comがある。Amazon.comではターゲットユーザの購入履歴からピアユーザを決定しターゲットユーザがまだ購入していない商品を推薦する。内容に基づくフィルタリングは推薦対象(以降、アイテム)同士の要素を比較することによって類似度を測定し、類似度の高いものを推薦する。楽曲推薦でも同様で、協調フィルタリング・内容に基づくフィルタリングによって行われている。楽曲推薦では主に音響情報をもとに、楽曲のテンポや音階を解析した情報をもとに楽曲の比較を行いアイテム間の類似度を測定する。

しかし、音響情報を収集するには音源の入手に、人手と費用が掛かる。また、廃盤などによって、入手そのものが困難なものがある。本研究では楽曲の内容には音響情報以外にも、作成された年代や、アーティストの特徴・楽曲を作成したプロデューサーの情報など楽曲を特徴付ける要素があると考え、ウェブ上に存在するテキスト情報から、楽曲情報を抽出し、情報推薦用のデータベースを作成する方法を提案する。本研究では年代などの楽曲情報を wikipedia から、ユーザの楽曲に対する印象を評判情報とし last.fm から、それぞれ抽出し楽曲のデータベースを作成し、このデータベースを用いた場合に推薦されるアイテムが従来手法により推薦されるアイテムを比較分析し、提案手法の評価を行った。

2. 関連研究

2.1 代表的な楽曲推薦システム

代表的な楽曲推薦システムには iTunes や last.fm, HORST[Stephan '10]が存在する。iTunesでは主に協調フィルタリングと専門家によるタグ付けによって推薦を行っている。last.fmでは楽曲にアノテーションされたタグを用いて同じタグが付けられているアーティストや楽曲を推薦する。HORSTはFreebaseを用いることによってユーザの好みに合った音楽アーティストの推薦を目的としている。斬新な音楽アーティストの推薦を目的としている。Freebaseは2029のジャンルで組織された階層構造を持つデータベースであり、HORSTはこのデータベースを利用することでアーティストの共演やジャンルなどを考慮してユーザの好みのアーティストと類似するアーティストの選択をして推薦する。

2.2 関連研究の問題点

iTunesで行われている専門家によるタグ付けは正確であるが、タグ付け自体に手間や費用がかかる。last.fmではユーザが自由にタグ付けできるため、楽曲に対するユーザの多様な評価データが手に入る一方で、同様の評価を表すタグに表記ゆれが存在する場合があるため、この表記ゆれを改善する必要がある。また、音楽自体の印象、ジャンル、音楽の周辺情報などタグの属性も複数存在するため、タグをドメインごとに分類する必要もある。単にタグの類似度だけで推薦を行った場合、例えば楽曲が映画やゲームに使われたことを表すタグの sound track が存在し、このタグをトリガとした場合、現行のシステムでは sound track のタグが付いた楽曲が推薦されてしまう。これは異なる映画に使われていた場合には、両者に関係性は薄いことと、同じ映画に使われている場合には同アルバムに収録されているためユーザがすでに知っている可能性が極めて高いため有意義な推薦にはならない。HORSTは、ジャンルと年代を統合した行列をアーティストの類似度を用いていることと、楽曲を推薦する際にこのアーティストの類似度を用いるため、提案される楽曲が同じアーティストのものばかりになる問題がある。

3. 獲得要素

関連研究の問題点をふまえて本研究では、last.fm から得られる各ユーザにより付与された多様性を持つタグ情報と wikipedia に書かれている客観性の高い楽曲に関する情報を抽出することにより、多様な観点での楽曲に関する属性を備えた、楽曲推薦用のデータベースを作成する。

本研究では獲得できる楽曲情報と評判情報が多いことが望ましいため、英語の曲を選んだ。これは、評判情報を獲得する last.fm のタグが英語で与えられているためである。今回、近代のメジャーなジャンルの一つであるラップミュージックを対象としてデータベースを作成した。提案するデータベースの作成方法は wikipedia, last.fm から情報が抽出出来るジャンルであれば適用可能であり、ラップミュージックだけに限らない。

3.1 楽曲情報

Wikipedia から得られる楽曲情報として、楽曲名、アーティスト、アーティストのカテゴリ、プロデューサ、収録されているアルバムのカテゴリ、収録年、そして年代ごとの分類を獲得した。

(1) アーティストのカテゴリ

アーティストのカテゴリはそのアーティストに関する要約であり、出身地や、誕生年、所属していたレコード会社やグループなどの情報が与えられている。

(2) プロデューサ

プロデューサは楽曲の作成過程全般に関わる人物で、流行などを考慮して、その楽曲の方向性を決めていいると考えられる。

(3) 年代ごとの分類

今回扱うラップミュージックには大まかに 3 つに分類されており、1970-1986 年までを old school hip hop, 1987-1999 年までを golden age hip hop, それ以降を new school hip hop とする。new school に分類された楽曲は特に、作成された地域の特徴、オーケストラの特徴などに多様性があり、ジャンルが細分化されているという特徴がある。

3.2 評判情報

楽曲の印象を得るために、last.fm からタグとその重みを獲得した。タグには、ラップミュージックのさらに詳細な deeply south などのジャンルの他に、catchy や angry などの音響の特徴を表す単語などがある。そして、そのタグを使ったユーザの度合いを示す重みも与えられている。タグの重みは 0 から 100 の値をとる。last.fm では API を用いて登録ユーザの様々な情報を獲得することができる。本研究では、ユーザの好みを判断するために、ユーザの視聴頻度の高いアーティストと楽曲を獲得した。

4. 作成したデータベース

獲得した楽曲情報と評判情報を用いてデータベースを作成した。

4.1 楽曲情報と評判情報のデータベース

Wikipedia と last.fm から得た属性と[Stephan ‘10]で抽出された属性との関係を図 1 に示した。図 1 に示す属性を 50 名のアーティストに関して 2230 曲集めた。本研究ではジャンルがラップミュージックであるため年代とその分類を HORST と同様の要素と考えた。

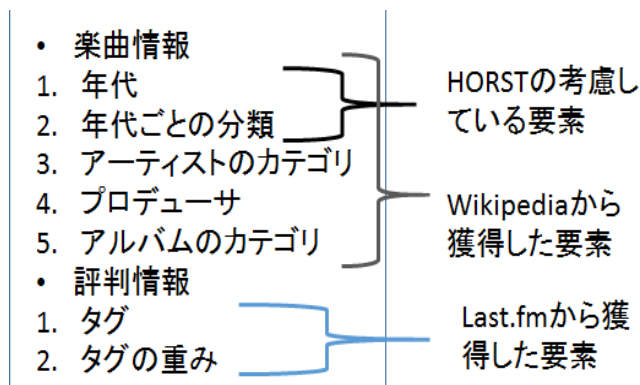


図 1: データベースの要素

4.2 ユーザの評価値行列

(1) ユーザとアーティストの評価値行列

獲得した 1398 名のユーザの視聴頻度が高いアーティストの評価値行列を作成した。評価値は得られたアーティスト数(今回は 50)を最大値とした値に正規化されている。表 1 に例を示した。

表 1: ユーザ-アーティストの評価値行列(各ユーザに評価されていないアーティストの項目には 0 を入力する)

	2Pac	Dr. Dre	50 Cent	Eminem
User1	19	0	0	0
User2	50	0	1	4
User3	45	1	43	0
User4	38	0	0	2
User5	22	0	0	0

(2) ユーザと楽曲の評価値行列

ユーザの視聴頻度が高い楽曲と、お気に入りに登録した楽曲の評価値行列を作成した。このとき、お気に入りに登録した楽曲同士の優先順位は得られないため、評価値を 0 と 1 の 2 値で与えた。

表 2: ユーザ-楽曲の 2 値の評価行列

	Track1	Track2	Track3	Track4
User1	1	0	0	0
User2	0	1	0	1
User3	1	0	0	1
User4	0	1	1	1
User5	1	1	0	0

5. 予備実験

本研究で作成したデータベースの属性の多様性を評価するため、[Stephan ‘10]で用いられた属性との比較分析を行った。本実験では楽曲の名前とアーティスト名が入力された際に、類似する楽曲をデータベース上でトランザクションを検索し、ランキングすることで曲を推薦する、コンテンツベースの推薦システムである。楽曲の検索手順を図 2 に示す。

図 3:2001 年に収録された Fabolous の Can't Deny It の類似度の高い上位 20 曲

年代とその分類の上位20曲				楽曲情報の類似度の上位20曲				評判情報の類似度の上位20曲				楽曲情報と評判情報の類似度の上位20曲			
artists	title	age	similarity	artist	title	age	similarity	artist	title	age	similarity	artist	title	age	similarity
Timbaland	Intro	2001	0.4	Fabulous	Not Give a Fuck	2003	1.81538	Fabulous	Keepin' It Gangsta	2001	0.75362	Fabulous	Not Give a Fuck	2003	2.43857
Timbaland	Drop (Timbaland & Magoo song)	2001	0.4	Snoop_Dogg	Candy (Drippin' Like Water)	2006	1.4924	Fabulous	Throw Back	2003	0.74917	Xzibit	Symphony In X Major	2002	2.1001
Timbaland	All Ya'll	2001	0.4	Xzibit	Symphony In X Major	2002	1.49156	Fabulous	Baby (Fabolous song)	2004	0.74107	Xzibit	Concentrate	2006	1.94168
Timbaland	It's Your Night	2001	0.4	Xzibit	Break Yourself	2002	1.49156	Fabulous	Breathe (Fabolous song)	2004	0.74047	Xzibit	Break Yourself	2002	1.9349
Timbaland	Indian Carpet	2001	0.4	Xzibit	Missin' U	2002	1.49156	Fabulous	Damn	2003	0.73505	Fabulous	Throw Back	2003	1.56455
Timbaland	Party People	2001	0.4	Xzibit	D.N.A.	2000	1.47918	Fabulous	Young'n (Holla Back)	2001	0.71949	Fabulous	Damn	2003	1.55044
Timbaland	People Like Myself	2001	0.4	Xzibit	Concentrate	2006	1.37391	Fabulous	Get Smart	2001	0.70979	Jadakiss	We Gonna Make It	2001	1.54876
Timbaland	Voice Mail	2001	0.4	Jadakiss	Intro	2001	1.09281	Fabulous	One Day	2001	0.70857	Jadakiss	Knock Yourself Out	2001	1.51671
Timbaland	Serious	2001	0.4	Jadakiss	Jada's Got a Gun	2001	1.09281	Fabulous	Trade It All	2001	0.69633	Fabulous	Trade It All (Part 2)	2003	1.51171
Timbaland	Roll Out	2001	0.4	Jadakiss	Show Discipline	2001	1.09281	Fabulous	Trade It All (Part 2)	2003	0.69633	Fabulous	This Is My Party	2003	1.49635
Timbaland	Love Me	2001	0.4	Jadakiss	Knock Yourself Out	2001	1.09281	Fabulous	Everything, Everyday, Everywhere	2009	0.68477	Snoop_Dogg	Candy (Drippin' Like Water)	2006	1.4924
Timbaland	Baby Bubba	2001	0.4	Jadakiss	We Gonna Make It	2001	1.09281	The_Notorious_B.I.G	I Love the Dough	1997	0.68412	Xzibit	Missin' U	2002	1.49156
Timbaland	In Time	2001	0.4	Jadakiss	None of Y'all Better	2001	1.09281	Fabulous	This Is My Party	2003	0.68097	Fabulous	Respect	2003	1.4871
Timbaland	Mr. Richards	2001	0.4	Jadakiss	Stick Yourself (skit)	2001	1.09281	Fabulous	Respect	2003	0.67172	Fabulous	Bad Bitch	2003	1.48364
Timbaland	Considerate Brotha	2001	0.4	Jadakiss	I'm a Gangsta	2001	1.09281	Fabulous	Gangsta	2004	0.67172	Xzibit	D.N.A.	2000	1.47918
Timbaland	Beat Club	2001	0.4	Jadakiss	Nasty Girl	2001	1.09281	Fabulous	Girls	2004	0.67172	Fabulous	Never Duplicated	2003	1.46351
Timbaland	I Am Music	2001	0.4	Jadakiss	Put Your Hands Up	2001	1.09281	Cypress_Hill	Kronologik	2001	0.67027	Busta_Rhymes	Genesis	2001	1.45648
Fabulous	Click & Spark	2001	0.4	Jadakiss	Jay Jerkin' (skit)	2001	1.09281	50_Cent	Intro	2005	0.66933	Busta_Rhymes	As I Come Back	2001	1.42925
Fabulous	Keepin' It Gangsta	2001	0.4	Jadakiss	On My Way	2001	1.09281	Fabulous	Bad Bitch	2003	0.66826	Busta_Rhymes	Break Ya Neck	2001	1.4251
Fabulous	Young'n (Holla Back)	2001	0.4	Jadakiss	Cruisin'	2001	1.09281	The_Notorious_B.I.G	Nasty Boy	1997	0.66042	Busta_Rhymes	Bad Dreams	2001	1.40599

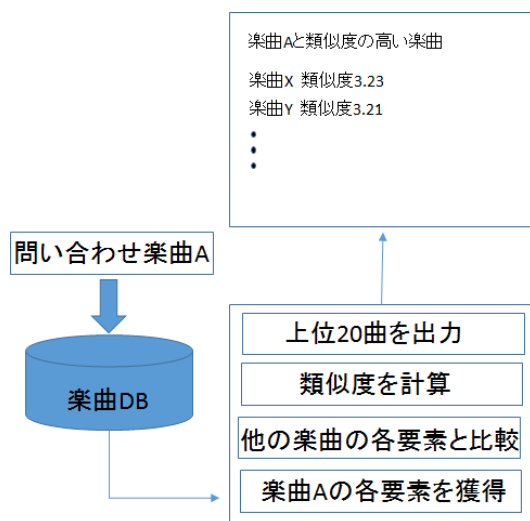


図 2:楽曲データベースを用いた推薦

楽曲データベースを用いて、入力アイテムとその他のアイテムの要素を比較して類似度の高い上位 20 件を表示した。以下に各手法で用いた属性の概要を述べる。

(属性 1). 年代と年代の分類のみを考慮 ([Stephan '10])

[Stephan '10]で用いられていた年代と年代の分類のみを属性とした。

(属性 2). Wikipedia から抽出した楽曲情報のみを考慮

プロデューサーと、アーティストとアルバムのカテゴリの類似度にはダイス係数を用いた。ダイス係数は楽曲 A と楽曲 B の要素の集合を X, Y としたときに、類似度は以下の式で与えられる。

$$sim(A, B) = \frac{2 \times |X \cap Y|}{|X| + |Y|} \quad (1)$$

(属性 3). lastm.fm から抽出したタグ情報のみを考慮

タグとその重みを比較要素として用いた。このとき、類似度はタグの重みを Min/Max 値を用いてコサイン距離を測定した。楽曲 A, B のタグの重みの集合を、 $A = \{W_A | w_{Ai} (i = 0, 1, 2, \dots, n)\}$, $B = \{W_B | w_{Bi} (i = 0, 1, 2, \dots, n)\}$ とすると、

$$sim(A, B) = \frac{\sum_{i \in n} (w_{Ai}^* \times w_{Bi}^*)}{\sqrt{\sum_{i \in n} (w_{Ai}^*)^2} \sqrt{\sum_{i \in n} (w_{Bi}^*)^2}} \quad (2)$$

$$w_{Ai}^* = \frac{w_{Ai} - Min(W_A)}{Max(W_A) - Min(W_A)}$$

となる。

(属性 4) 楽曲情報と評判情報を考慮((属性 2)+(属性 3))

楽曲情報の類似度に評判情報の類似度を加えて類似度を測定した。

5.1 結果と考察

ここでは 2001 年に収録された Fabolous の Can't Deny It という楽曲をクエリとして類似度計算を行い(属性 1)~(属性 4)の属性を用いた場合の検索結果を比較した。表 3 に(属性 1)~(属性 4)の属性を用いた場合の上位 20 位のアーティスト(artist)・曲のタイトル(title)・年代(age)・類似度(similarity)を示す。

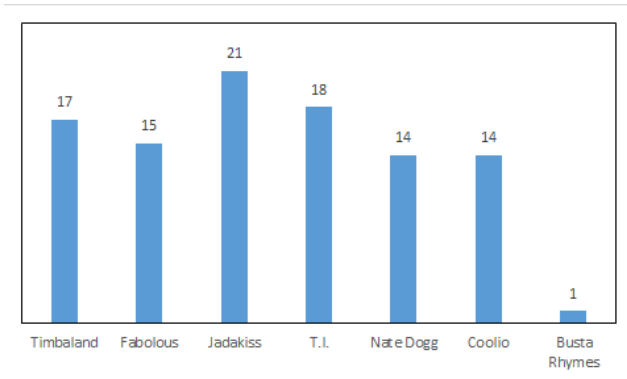


図 3: (属性 1)を用いた場合の上位 100 曲のアーティストと出現回数

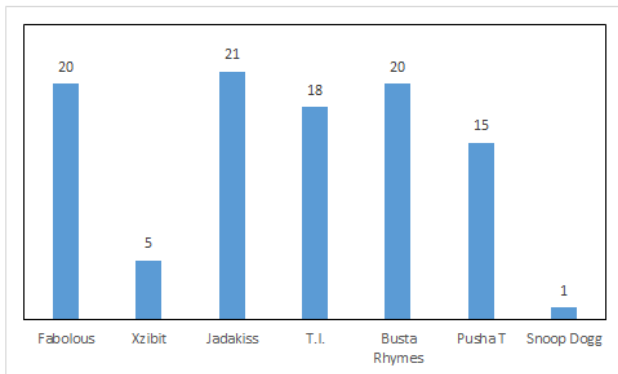


図 4: (属性 2)を用いた場合の上位 100 曲のアーティストと出現回数

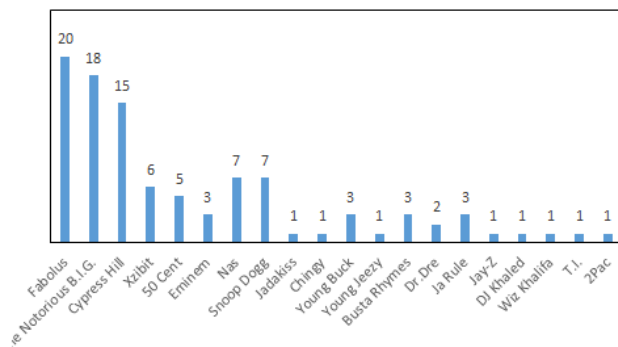


図 5: (属性 3)を用いた場合の上位 100 曲のアーティストと出現回数

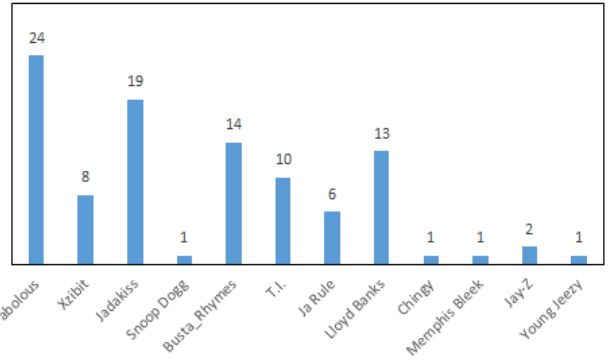


図 6: (属性 4)を用いた場合の上位 100 曲のアーティストと出現回数

図 3 に(属性 1)を用いた場合の結果を示す。年代と年代の分類のみの属性のため、年代の近いアーティストのみが検索上位にランクインされる。図 4 に(属性 2)を用いた場合の上位 100 曲のアーティストと出現回数を示す。楽曲情報全体を用いた推薦では上位 20 曲に異なる年代のものが推薦されている。これはプロデューサーやアルバムの一致によるものと考えられる。しかし、上位 1100 曲でのアーティスト数は(属性 1)と等しかった。これはアルバムのカテゴリによるものだと考えられる。実験(属性 1)と同様でユーザが楽曲を購入する際にアルバムで購入するため推薦される楽曲に既に視聴したことがある楽曲があることが予想される。

図 5 に(属性 3)を用いた場合の上位 100 曲のアーティストとその出現回数を示す。(属性 3)は楽曲のタグのみを用いたため、上位 100 曲のアーティスト数は 20 人と最も多い結果となった。アーティストの要素は考慮されていないため、知らないアーティストの楽曲が推薦されることが期待できる。図 6 に(属性 4)を用いた場合の上位 100 曲とその出現回数を示す。(属性 4)の結果は(属性 3)のそれと比べるとアーティスト数は減ってしまったがアーティストの大部分が Fabolous と出身地が一緒であったり、クエリの曲と曲調が類似するような楽曲アーティストがランキングされており、多様な尺度で類似するアーティストが検索できていることが分かる。

6. まとめ

本研究では年代などの楽曲情報を wikipedia から、ユーザの楽曲にたいする印象の評判情報を last.fm から、それぞれ抽出し楽曲のデータベースを作成し、その評価を検索実験により評価した。今回、検索結果の満足度などは評価していない。また、ユーザと楽曲の評価値行列が疎なものになってしまったため、予備実験にユーザ情報を用いなかった。今後はユーザ情報を収集し、提案するデータベースを備えた推薦システムを構築することを検討している。

参考文献

[Stephan 2010] Stephan Baumann: Towards a Storytelling Approach for Novel Artist Recommendations, Lecture Notes in Computer Science, 2010.
 [McVicar 2011] Matt McVicar: The Correlation between Lyrical and Audio Features and the Emergence of Mood, ISMIR, 2011.
 [田中 2012] 田中 克己, 情報推薦システム入門, 共立出版, 2012