修士論文 2004 年度 (平成 16 年度)

探索的お薦め導出手法を用いた 作品の発見に関する研究

慶應義塾大学 政策・メディア研究科

氏名:鈴木貴晶

主査 村井 純

副査

中村 修

宮川 祥子

平成17年2月15日

修士論文要旨 2004年度 (平成16年度)

探索的お薦め導出手法を用いた作品の発見に関する研究

本研究は、ユーザが満足する作品を発見する可能性の向上を目的とする。目的を実現するモデルとして「探索的お薦め導出手法」を提案した。

近年、世界中には多くのアーティストが存在し、次々と新しい作品が生み出されている (対象の多種多様性)。そして、これらの作品の評価は、各自の嗜好によって異なる(主観 による選択)。この様な状況において多くの作品の中から、各々のユーザが、各自満足す る作品を発見することは難しい。

そこで本研究では「探索的お薦め導出手法」を提案した。探索的お薦め導出手法とは、(1) 各自が自己の嗜好を満足させるべく選択している行動の痕跡を収集し他のユーザに伝達する、(2) ユーザは他のユーザから伝達された他ユーザ行動の痕跡から、自分と似た嗜好のユーザを発見する、(3) 自分と似た嗜好のユーザからの行動の痕跡を優先的に取り入れ、お薦めを導き出す手法である。本モデルは、既存の問題点である、発見的でないこと・探索空間の狭さ・網羅性の乏しさを解決する。

本研究は、提案したモデルを適用するシーンとして、対象の多種多様性かつ主観による 選択であるという特徴を満たす、音楽を取り上げた。そして、モデルを検証するために、 既存手法を用いたモデルとの比較実験を行った。その結果によって、本モデルは、ユーザ が満足する作品を発見する可能性を、既存手段よりも向上させることを実証した。

さらに、音楽再生ソフトウェアの iTunes とインスタントメッセージングのクライアントソフトウェアの wija を用い、前述のモデルを wija プラグイン「Tunes プラグイン」として実装した。そして、実装を用いた実験を行い、実装が本モデルの要求を満たしていることを実証した。また、実装の運用を通して得られた利用状況やユーザの声より、今後の課題を明らかにした。

最後に、2つの実験から本研究が達成したことを整理し、本研究の今後の課題について 検討を行った。

キーワード

1. リコメンデーション (お薦め), 2. 嗜好, 3. 知人関係,

4. インスタントメッセージング (IM), 5. ソーシャルネットワーキングサービス (SNS)

慶應義塾大学 政策・メディア研究科 鈴木 貴品

Abstract of Master's Thesis Academic Year 2004

Discovering Agreeable Creative Works by Exploratory Recommendation Elicitation

The goal of this research project is to enhance possibilities for users to discover existence of create works agreeable to their taste. In order to achieve the goal, it offers a new model called "Exploratory Recommendation Elicitation".

There are many artists all over the world, continuously producing many new creative works (plentifulness of the objects). The audience appreciate those works differently according to their taste (preferences by the subjects). Such conditions make it difficult for users to discover creative works that would meet their taste.

This paper introduces "Exploratory Recommendation Elicitation" to solve the problem. This method derives recommendations by (1) collecting the traces of user's activities to meet their preferences, and forwarding the information to other users, (2) finding those other users who have similar preferences to them, and (3) selecting the traces of activities by those users with the most similar preferences. This solves the problems of existing methods: their lack of heuristic, small search space and narrow coverage.

In this research, the model is applied to music, which satisfies both *plentifulness of the objects* and *preferences by the subjects*. A comparison experiment with an existing model demonstrated that "Exploratory Recommendation Elicitation" offers higher possibility for users to discover songs that might meet their taste.

The model is implemented as a plug-in called "Tunes" for an instant messaging client software "wija". The plug-in works with a music player software "iTunes".

Experiments with the software showed that this implementation satisfies the requirements from the model. It also revealed several problems as future works through actual behaviors of the users and their comments.

The achievements of this research are summarized and the future works are clarified at the end of this paper.

Keywords:

1.recommendation, 2.preference, 3.Social Network, 4.instant messaging(IM), 5.social networking service(SNS)

Takaaki Suzuki Graduate School of Media and Governance Keio University

目 次

第1章	序論		1
1.1	本研究	の目的	1
1.2	本研究	の概要	1
1.3	本研究	の背景	2
	1.3.1	作品との出会い	2
	1.3.2	知人関係とコミュニケーション	3
1.4	本論文	の構成	3
第2章	既存手	段とその問題点	5
2.1		R による未知の作品の発見	
2.1		検索サイトを利用しキーワードによって作品を検索する・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	
		知人にお薦めを尋ねる	
		ショッピングモールなどのリコメンドサービスを利用する	
2.2		段の問題点	
2.2		既存手段による未知の作品の発見の限界・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	8
		既存手段における問題	10
笙 3 音	探索的	お薼め道出手法モデル	12
第 3 章 31			12 12
第 3章 3.1	要件定	義	12
	要件定 3.1.1	義	12 12
	要件定 3.1.1 3.1.2	義	12 12 13
	要件定 3.1.1 3.1.2 3.1.3	義	12 12 13 13
	要件定 3.1.1 3.1.2 3.1.3 3.1.4	義 ユーザが満足した作品の伝達 嗜好が近い人の発見とお薦めの導出 お薦めの絞り込み 網羅性の向上 網羅性の向上	12 12 13 13
3.1	要件定 3.1.1 3.1.2 3.1.3 3.1.4 3.1.5	義	12 13 13 13 16
	要件定 3.1.1 3.1.2 3.1.3 3.1.4 3.1.5 探索的	義	12 13 13 13 16 16
3.1	要件定 3.1.1 3.1.2 3.1.3 3.1.4 3.1.5 探索的 3.2.1	義	12 13 13 13 16 16
3.1	要件定 3.1.1 3.1.2 3.1.3 3.1.4 3.1.5 探索的 3.2.1 3.2.2	義	12 13 13 13 16 16 16
3.1	要件定 3.1.1 3.1.2 3.1.3 3.1.4 3.1.5 探索的 3.2.1 3.2.2 3.2.3	義	12 13 13 13 16 16 16 16 18
3.1	要件定 3.1.1 3.1.2 3.1.3 3.1.4 3.1.5 探索的 3.2.1 3.2.2 3.2.3 3.2.4	義	12 12 13 13 13 16 16 16 16 18 18
3.1	要件定 3.1.1 3.1.2 3.1.3 3.1.4 3.1.5 探索的 3.2.1 3.2.2 3.2.3 3.2.4 モデル	義	12 13 13 13 16 16 16 16 18 18
3.1	要件定 3.1.1 3.1.2 3.1.3 3.1.4 3.1.5 探索的 3.2.1 3.2.2 3.2.3 3.2.4 モデル 3.3.1	義	12 12 13 13 13 16 16 16 16 18 18

3.4	実現可能性の検討・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	20
	3.4.1 嗜好の近さの測り方 2	20
	3.4.2 お薦めの生成方法	23
第4章	モデルの比較検証実験と評価 2	24
4.1	検証のための本モデルの適用シーン	24
4.2	実験概要	24
4.3	シュミレーション環境の実装 2	26
	4.3.1 プログラミング言語 2	26
	4.3.2 ソフトウェア	26
	4.3.3 データベーススキーマ	26
	4.3.4 本モデルによるお薦めの提示システム 2	27
	4.3.5 既存のリコメンドサービスの手法によるお薦めの提示システム 2	28
4.4	要件を満たしているかの評価 2	28
	4.4.1 探索空間の拡張	
	4.4.2 網羅性の向上 2	29
4.5	既存手段に対する本モデルの優位性の評価 3	30
	4.5.1 アンケートの方法と仕様	30
		31
4.6	実験結果のまとめ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	35
第5章	設計 3	37
5.1	システムの全体構成	37
5.2	各部の機能 : : : : : : : : : : : : : :	38
		38
		38
		38
		39
	5.2.5 お薦め生成提示部	
5.3	処理の流れ	
第6章		12
6.1	実装環境	
0.1	6.1.1 プログラミング言語 4	
	6.1.2 ソフトウェア	
6.2	実装の構成・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	
0.2		43
		43 44
	·	$\frac{44}{45}$
6.3	6.2.5 Tunes フラグイン	
0.5		
	- 6.3.1 - 用語解説と定義	110

	6.3.2	プラットフォーム抽象化レイヤ4
	6.3.3	サンプル生成部 4
	6.3.4	サンプル送信部
	6.3.5	サンプル受信部
	6.3.6	ユーザランキング生成部 4
	6.3.7	お薦め生成提示部
	6.3.8	曲情報取得支援部4
	6.3.9	サンプル表示部
	6.3.10	環境設定部 5
	6.3.11	運用実験のための機能
第7章	実装を	: Hいた評価 55
7.1	仮想ユ	- ーザによる実験 5
	7.1.1	実験の目的 55
	7.1.2	ユーザが満足する作品の伝達 5.
	7.1.3	嗜好が近い人の発見とお薦めの導出及びお薦めの絞り込み 5
	7.1.4	網羅性の向上 5
	7.1.5	探索空間の拡張
7.2	運用実	験
	7.2.1	実験の目的 5
	7.2.2	実験概要5
	7.2.3	アンケート 5
第8章	関連研	· F究
8.1	知人関	係を利用した情報共有・情報検索6
8.2	知人関	係を利用した情報検索 6
	8.2.1	Referral Web
	8.2.2	Cite Seer
	8.2.3	本研究との関連 6
8.3	知人関	係を利用した人材マッチング
	8.3.1	ANS
	8.3.2	HUMANNET
	8.3.3	本研究との関連 6
8.4	嗜好の	近さを測るサービス
	8.4.1	booklog
	8.4.2	recoya
	8.4.3	本研究との関連
8.5	知人関	係を利用した作品の情報の伝達6
	8.5.1	Recommuni
	959	大研究との関連 6

第9章	结論		66
•			00
9.1	本研究	アが実現したこと	 66
9.2	今後の	D課題	 67
	9.2.1	嗜好の近さを求める際のサンプル数の限定	 67
	9.2.2	サンプルのプライバシー	 67
	9.2.3	試聴環境の整備・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	 68
謝辞			69
参考文献	就		71

図目次

2.1	Amazon 商品の評価	6
2.2	Amazon おすすめランキング	7
3.1	販売数と露出度の関係	14
3.2	3つのタイプのユーザの傾向と露出度の関係................	15
3.3	ビー玉のみせっこ....................................	17
3.4	優先順位付け	18
3.5	サンプルの推移	18
3.6	既存のリコメンドサービスのモデルとの比較	19
3.7	各ユーザの好きな物集合の重なり	21
3.8	各ユーザの好きな物集合の変化	22
4.1	iTunes ライブラリ、知人関係登録ページ	25
4.2	hop 数によるサンプルを届ける人数の総和の変化	29
5.1	システム構成図	37
5.2	利用の流れ	40
6.1	実装の構成	43
6.2	wija	45
6.3	お薦めリスト	49
6.4	サンプルの表示	50
6.5	Tunes プラグイン設定画面	51
8.1	Acquaintance Network System	61
8.2	HUMANNET	
8.3	booklog	63
8.4	棚の一致度ランキング	64

表目次

2.1 2.2	検索に用いたキーワードとヒット件数	
3.1	鑑賞の傾向とその状況	
3.2	各ユーザの要素数、特性一覧	21
3.3	ユーザ p1 から見た各ユーザの近さ	22
4.1	実装環境	26
4.2	Table "user_info"	26
4.3	Table "social"	26
4.4	Table "song"	27
4.5	Table "holding"	27
4.6	Table "point"	27
4.7	Table "song_relation"	27
4.8	各手法の延べ所持人数	30
4.9	問1の集計結果	31
	問2の集計結果	
4.11	未知の作品数と既知の作品数の比較・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	32
4.12	未知の作品の満足度の比較	32
4.13	タイプ別の未知の作品の満足度の比較	33
4.14	被験者からの感想	34
5.1	サンプルのデータ構造	38
7.1	5人の仮想ユーザ	54
7.2	モデルに基づいたお薦めリストの内訳	54
7.3	実装におけるお薦めリストの内訳	55
7.4	3人の仮想ユーザとライブラリの曲	55
7.5	お薦めリストの内訳 網羅性の向上	
7.6	3人の仮想ユーザ	56
7.7	お薦めリストの内訳 探索空間の拡大	
7.8	統計情報の集計結果・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	58
7.9	ユーザからのコメント	58

第1章 序論

1.1 本研究の目的

本研究は、音楽・映画等の様に次々と新しい作品が発表され(対象の多種多様性)、その評価が各自の嗜好によって異なる(主観による選択)シーンにおいて、ユーザが満足する作品を発見する可能性の向上を目的とする。

1.2 本研究の概要

現在、世界中には多くのアーティストが存在し、次々と新しい作品が生み出されている。そして、これらの作品の評価は、各自の嗜好によって異なる。この様な状況において、多くの作品の中からユーザが満足する作品を発見することは難しい。

ユーザが満足する作品を発見するための方法として、既存の手段が複数存在するが、それらは、

- 発見的でない
- 探索空間が狭い
- 網羅性に乏しい

といった問題を抱えている(第2章)。

本研究は、音楽・映画等の様に対象の多種多様性かつ主観による選択を対象として、ユーザの満足する作品を発見する可能性の向上を目的とする。

そのアプローチとして、後述するインスタントメッセージングサービス(以下 IM)やソーシャルネットワーキングサービス(以下 SNS)で用いられている知人関係に基づくコミュニケーションモデルを用いる。この上で、「曲を聴く」「本を読む」といった各自が嗜好を満足させるべく選択している行動の痕跡を送り合うことによって、作品に関する情報の伝達を促進させ、満足する作品の発見の可能性を高める。直接の知人だけでなく知人の知人に・・・と各ユーザが満足を覚えた作品を伝達していくことで、自分と嗜好が近い人を発見することができ、自分一人ではない、より多くの「目」を持って満足する作品を探していくことができる。そして、嗜好が近い人の行動の痕跡から、ユーザへの作品のお薦めを導き出すことを可能にする。

この方法を実現するため、既存手段の問題点を整理し、諸問題を解決するモデル化を行い、実際の利用シーンを絞り、ユーザが利用可能なシステムとして実装する。

1.3 本研究の背景

1.3.1 作品との出会い

今日、世界には音楽・映画・絵画・彫刻など様々な分野に渡って多くのアーティストが存在し、日夜次々と新しい作品が生み出されている(対象の多種多様性)。一方、それらを鑑賞する側にとって、多種多様な存在の中から自分が満足する作品を発見することは難しい。また、これらの作品は鑑賞する側の嗜好によって評価が異なる(主観による選択)ことも、作品の発見を難しくしている。

多くの作品の中で、マスメディア等によって大量に宣伝が行われているものほど、私たちは容易に出会うことができる。音楽を例に挙げると、新しい楽曲が登場するたびにマスメディアによって大量の宣伝が行われる著名なアーティストもいる一方、ローカルに活動しており新曲を出した場合であってもごく一部のファンの間でしかその新曲の情報が伝わらないといったアーティストも存在する。また、メジャーと呼ばれるレコード会社に所属していたとしても、大量に宣伝される他のアーティストの陰に隠れ、その活動を容易には知ることができないアーティストも多い。

しかし、純粋に一人一人の嗜好の視点から考えれば、大量に宣伝されている作品もよく知られていない作品も、ユーザを満足させる作品である可能性を持っている。現状において、作品の良悪を判断して大量の宣伝を行うか、を判断をするのは発掘を専門に行っているプロのプロモータである。アーティストは、その様な人たちの目に触れる機会が得られなければ、活動を多くの人に広く知ってもらうことが出来ない。しかし、どのアーティストについても、アーティストの日常の活動を目にしている一般の鑑賞者や支援者(以下、アマチュア)は確実に存在している。素晴らしいアーティストであれば、その活動を目にしたアマチュアは、周りの友人にその存在を教えるはずである。

つまり、素晴らしいアーティストの活動を目にした人からの口コミの広がりを支援することができれば、プロのプロモータによるボトルネックを解消し、より多くの人に作品を知ってもらう可能性が高まる。これにより、発掘を専門に行うプロモータだけではなく、アマチュアー人一人が発掘を行う人になることができる。

この際、アマチュアの口コミを単に支援するだけでは、必ずしも多くの人に満足されないお薦めを大量に生み出してしまう懸念がある。従って、お薦めの精度の向上が必要である。なお、先に述べたように嗜好物におけるお薦めの精度は、人それぞれ異なる。そこで、人それぞれ自分が満足を覚えた作品を周りの人に伝えあう仕組みを作る。このことにより、人はそれぞれ自分と嗜好が似た他者を発見することができ、嗜好の似た人同士で作品に関する情報の共有を可能とすることで、人それぞれに異なった精度の高いお薦めを実現する。

以上をまとめると本研究は、

- アマチュアによる口コミの広がりを支援することにより、ボトルネックを解消する
- 自分と似た嗜好の人を発見することにより、お薦めの精度を向上する

ことによって、ユーザが満足する作品を発見する可能性の向上を実現する。

1.3.2 知人関係とコミュニケーション

近年、インターネットにおいて、友人同士のコミュニケーションを促進するアプリケーションが増加し、多くの人々に利用されるようになった。例えば、インスタントメッセージングサービスと呼ばれるサービスが広く利用されるようになった。IM は、同じプロトコルで通信するソフトウェアを利用している仲間がオンラインかどうかが分かるプレゼンス共有や、仲間間でチャットやファイル転送などを行なう機能をもつ。最近ではビジネスシーンにおいても利用されるようになり、「音楽の共有」「ファイル転送」「リモートデスクトップ」「ライブ・音声チャット」などの機能拡張が行われている。IM クライアントとしては、AOL 社の AOL Messenger、Microsoft 社の MSN Messenger が有名である。

また、2003 年頃から、ソーシャルネットワーキングサービスと呼ばれるコミュニティ型の Web サイトが相次いで誕生した [1]。これらは、ユーザ間で互いに友人を紹介しあうことで新たな友人関係の構築 [2] や、既存の知人とのより深いコミュニケーションを目的 [3] とするものである。これらの中には、Amazon[4] 等のショッピングサイトと連携し、自分のお気に入りの商品をお薦めとして、友人等に紹介する機能が存在するものもある。音楽など特定の分野の推薦を目的とした SNS も存在する。[5]

これらは、消費対象の選定における価値判断を知人に委譲する、あるいは、委譲とまではいかなくとも、価値判断の材料として知人からの情報を重要視する、といった私たちの日常の行動をモデル化したものであるといえる。[6]。これらの行動は、知人間に存在する一種のソーシャルキャピタルを元にして生まれるものである。

本研究は、1.2に述べたように、

- アマチュアによる口コミの広がりを支援することにより、ボトルネックを解消する
- 自分と似た嗜好の人を発見することにより、お薦めの精度を向上する

といったことを通して、目的を実現する。前述したモデルは、これらのことを実現する上で応用すべきものと考える。

1.4 本論文の構成

本論文では、全9章で構成される。構成は次のようになっている。

- 第2章では、既存手段による作品発見の問題点について述べる。
- 第3章では、本研究の提案するお薦め導出手法を用いたモデルについて説明する。
- 第4章では、本モデルの有効性を評価するため、既存のリコメンドサービスの手法 を用いたモデルとの比較検証実験を行った。

- 第5章では、検証実験を踏まえ、第3章で述べたモデルを音楽再生ソフトウェアの iTunes と IM クライアントの wija を用いて実現化するための設計について述べる。
- 第6章では、その実装について述べる。
- 第7章では、実装の運用を通し、実装がモデルを満たしているかの評価を行った。
- 第8章では、本研究の関連研究について述べる。
- 第9章では、本論文での結論と今後の課題について論じる。

第2章 既存手段とその問題点

本章では、未知の作品を発見するために、現在、人々が採りうる手段のうち、1.2で述べた手段に関連するものを取り上げ、その問題点について考察する。

2.1 既存手段による未知の作品の発見

ここでは、1.2で述べた手段に関連する既存の手段として、

- 検索サイトを利用しキーワードによって作品を検索する
- 知人にお薦めを尋ねる
- ショッピングモールなどのリコメンドサービスを利用する

を取り上げる。以下、それぞれの手段について考察していく。

2.1.1 検索サイトを利用しキーワードによって作品を検索する

Google[7] 等の検索サイトを利用して未知の作品を発見する手段がある。この手段には発見する作品に関連するキーワードが必要になるが、有効なキーワードをユーザが選定することは難しい。

2.1.2 知人にお薦めを尋ねる

嗜好が似ている人や、その人の嗜好に信頼をおいている人に、お薦めを尋ねるという手段がある。この方法は、基本的には検索範囲が知人に限られる。しかし、自分の知人の知人の中に、直接の知人と比較してより嗜好が近い人がいる可能性がある。その場合、一般的には、その様な人の存在を知ることは困難である上、その人と直接コミュニケーションをとる手段はないと考えられる。自分とその人の間にいる共通の知人を経由して、そのお薦めを受けられる可能性はあるが、低い。またそのような場合であっても、共通の知人との嗜好のずれから、そのお薦めが届かない可能性がある。さらに、そのようなお薦めの伝播には時間がかかる。また、共通の知人はある程度のコストを負担することになると考えられる[6]。

2.1.3 ショッピングモールなどのリコメンドサービスを利用する

Amazonの様な大規模ショッピングサイトでは、利用者の購買履歴を分析することによって、利用者毎にお薦めを提示するリコメンドサービスを行っている。この際、協調フィルタリングという技術が基本的に用いられる¹。

協調フィルタリングとは、ユーザの嗜好を過去の行動という形で記録し、そのユーザと似たような行動を取っているユーザの嗜好情報を元に、ユーザの嗜好を推測するシステムである。これにより、ユーザの嗜好に近い作品を薦めることが可能になる。協調フィルタリングではユーザごとの嗜好情報が多く、ユーザの数自体も多い方がより有用な推測が可能となる。

Amazonでは、買った物の組み合わせを元に商品を薦めるサービス(以下、Amazon Aサービス)が存在する。「この商品を買った人はこんな商品も買っています」という形でユーザにお薦めする商品が提示されるこのサービスは、協調フィルタリングを基礎としている。また、よりよいお薦めを実現するために Amazon Aサービスを拡張し、ユーザの所持する商品を元にお薦めを作成するサービス(以下、Amazon Bサービス)も存在する。さらに、持っている商品に評価をつけることで、お薦めを絞り込むことができる(図2.1)。



図 2.1: Amazon 商品の評価

¹協調フィルタリングに関しては、人工知能学会の解説ページが詳しい。http://www.ai-gakkai.or.jp/jsai/whatsai/AItopics2.html

お薦め商品は、強く薦められる順にランキングされ、ユーザは上位100位を確認することができる(図2.2)。さらに、お薦めとして提示された商品に対して、「興味がない」という判断をユーザに行わせることで、お薦め商品の一覧の内容を変化させる機能を持つ。

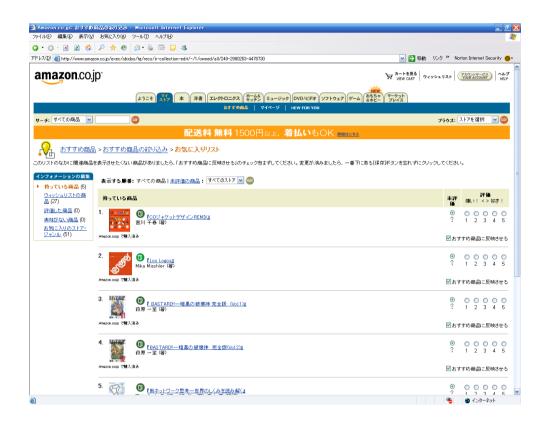


図 2.2: Amazon おすすめランキング

以上の様に、Amazonは複数の方法によって、ユーザにお薦めを提示している。しかし、これらいずれの方法も、多く購入された商品が優先的に扱われるため、多く購入されていない商品は、薦められない。以下、多く購入された商品が優先的に扱われる仕組みについて述べる。

まず、Amazon A サービスに見られる、商品同士の結びつきを用いたお薦めの仕組みについて述べる。

ユーザaが持っている商品の集合を、

$$I_a = \{i_1, \cdots, i_{|I_a|}\}$$

とする。また、商品 x と商品 y を共に持っている購買者の数を、NoP(x,y) とする。ユーザ a にとってある商品 j のスコア Saj は、

$$S_{aj} = \sum_{i \in I_a} (NoP(i,j))$$

となる。システムは、ユーザaにとってスコアが高い商品から推薦する。つまり、多く買

われている商品ほどスコアが高くなるため、優先的に推薦される。

次に、購買者の商品への評価情報を元にした Amazon B サービスにおけるお薦めの基本的な枠組みについて述べる。この様なサービスの基本的手法としては、相関係数法が挙げられる。

相関係数法では、まず購買者同士の商品への評価をベクトルと見なして、ベクトルの相関係数を求め、購買者同士が似ているかどうかの指標とする。両者が「似ている」場合、この相関係数が1に近づき、「中立」の場合0に、「大半対」の場合-1に近づく。

ある商品 x が購買者 a に薦めるか判断する際、相関係数法による協調フィルタリングでは、a が商品 x に対する評価を計算によって予測する。その評価は、x に対する評価をつけた購買者達との相関係数をあらかじめ求め、購買者それぞれの x への評価から、その購買者の全商品についての平均評価を引き、その値に a との相関係数を掛けたものの総和となる。

すなわち、a に対してお薦めを提示する際は、相関係数をかけたものを総和が大きい順にお薦めとして提示するため、より多く評価が行われる商品、つまり多く買われた商品が有利である。

2.2 既存手段の問題点

2.2.1 既存手段による未知の作品の発見の限界

先に述べた既存手段の限界を明らかにするために、実際のサービスを利用して作品を発見できるか調査を行った。その際、以下の2つの作品を利用した。

- 著名なアーティスト"スーパーカー"(以下 A)
- 最近登場した新人アーティスト"scarlet" (レコードショップや雑誌などで A の影響を強く受けたという意見が見られたアーティスト)(以下 B)

検索サイトにおけるキーワードによる作品の検索

まず、検索サイトにおけるキーワードによる作品の検索について調査を行った。先に挙げたアーティスト A、B をキーワードとして検索を行い作品が発見できるか検索サイト Google を利用し実験した。結果は表 2.1の通りになった。

キーワード	ヒット件数
A AND B	72 件
A AND B AND お勧め	2件
A AND B AND おすすめ	0件
A AND B AND お薦め	1件
A	5万件以上
A AND おすすめ	8990 件
A AND 最近	1780 件

表 2.1: 検索に用いたキーワードとヒット件数

結果を考察していく。まず、アーティストAとアーティストBの関連を主張するページが存在するかを確認するため、"A AND B"というキーワードで検索を行った。結果、72件のページがヒットし、それぞれ内容を見ていったところ、内B 45件のページがアーティストB 60関連を主張する内容を含んでいた。B 2これにより、レコードショップや雑誌以外においてもB 8 0音楽的関連を指摘するページが存在することを確認できた。さらに、"おすすめ"、"お薦め"、"お勧め"というキーワードを合わせて検索を行い、ヒットしたページの内容を見たところ、B 6気に入っている人に対してB 6をお薦めとしてあげているページが 3 件存在した。

表の後半3つの検索は、アーティストAをキーワードにしてアーティストBを発見しようという試みである。"A"というキーワードによる検索では、5万件以上のページがヒットした。このうち上から300件のページを実際に内容を見ていったが、Bの存在を発見することは出来なかった。次に、"おすすめ"というキーワードを合わせることで絞り込むと、8990件のページがヒットした。先と同様、上から300件のページの中にBの存在は発見出来なかった。また、Bは最近登場したアーティストなので、"最近"というキーワードを合わせて検索を行った。結果、1780件のページがヒットしたが、上から300件の中にBの存在は発見出来なかった。アーティストAをキーワードにした検索では、アーティストBを発見することは困難であるといえる。

以上の実験結果より、有効なキーワードをユーザが選定することの難しさが示唆された。

ショッピングモールなどのリコメンドサービスを利用する

次に先に述べた Amazonのリコメンドサービスを利用して、実験を行った。なお、Amazonにおいては、アーティスト A の作品は 2 9 作品、アーティスト B の作品は 2 作品購入することが可能である。

まず、B の作品を全て所持しているものとして登録し、お薦めに A の作品が登場するかを実験した。その結果、A の作品は、提示されたお薦め 1 0 0 位以内に、 8 作品登場した。よって、アーティスト A とアーティスト B の関連性が確認された。

²なお、それ以外のページでは、キーワードが同音異義語として扱われていた。

次に、Aの作品を全て所持しているものとして登録し、お薦めにBの作品が登場するかを実験した。その結果、提示されたお薦めの中に、Bの作品は登場しなかった。

最後に、 $A \ \ \, \ge B \ \ \, \ge B$ ともに自分が興味を持つアーティストである筆者が、現在所有している $CD\ 309$ 点を登録し、提示されるお薦めに対して、150 点の「興味ない」判断を登録したが、B がお薦めとして提示されることはなかった。

この実験結果より、2.1.3にて示された点、すなわち、多く買われていない作品はお薦め として提示される可能性が低い点を再確認した。

2.2.2 既存手段における問題

前節の各既存手段の考察より、既存手段の問題点として以下が挙げられる。

発見的でない

- キーワードを用いた検索では、検索する作品についてすでに何らかの情報を得ていなければならない。何の情報もなく、未知の作品を発見的に検索することは困難である。

● 探索空間が狭い

- 知人関係を利用する場合、自分の直接の知人よりも、さらにその知人の中に自分と嗜好がより近い人がいる可能性があるが、その人の存在を知ることは出来ない。さらに、その人から有用なお薦めが共通の知人に対して届いたとしても、そのお薦めが必ずしも自分まで届くとは限らない。よって、直接の知人までという狭い探索空間でしかお薦めが行われない。

● 網羅性に乏しい

既存の大規模ショッピングサイトで行われるリコメンドサービスは、ユーザに対してお薦めを提示する際、多く買われた商品同士の存在を優先的に表示する。よって、少数しか買われていない商品の存在は、その陰に隠れて見えないため、網羅性に乏しい。

以上の問題点と、取り上げた3つの既存手段の関係を表2.2に表す。

化 2.2. WIT 1 7 C 问题				
	発見的	探索空間	網羅性	
キーワードによって検索する	×			
知人に尋ねる		×		
リコメンドサービスを利用する			×	

表 2.2: 既存手段と問題点の関係

本研究が提案するモデルでは、上記問題点を解決し、満足する作品を発見する可能性を 高める。解決へのアプローチについては、3章で詳しく述べていく。

第3章 探索的お薦め導出手法モデル

3.1 要件定義

1章において、ユーザが満足する作品を発見する可能性を向上させるための本研究のア プローチが、

- アマチュアによる口コミの広がりを支援することにより、ボトルネックを解消する
- 自分と似た嗜好の人を発見することにより、お薦めの精度を向上する

であることを述べた。また、2章においては、解決すべき問題点として、

- 発見的でない
- 探査空間が狭い
- 網羅性に乏しい

について述べた。

これらを実現・解決するため、本モデルは、次の要件を満たす必要がある。

- 満足した作品の伝達
- 嗜好が近い人の発見とお薦めの導出
- お薦めの絞り込み
- 網羅性の向上
- 探索空間の拡張

以下、各要件の詳細について述べる。

3.1.1 ユーザが満足した作品の伝達

各自が満足した作品の情報を、各自の知人に対して伝達する機能が必要である。

3.1.2 嗜好が近い人の発見とお薦めの導出

各自がそれぞれ満足している作品の情報を伝達し合う中、各自は自分の嗜好に近い人を探し、似た人から伝達される作品の情報を優先的に受け取ることで、新しい作品の発見を実現する(1章)。これを実現するためには、各自が他者の満足する作品の情報を蓄積し、その中から自分の嗜好に近い人を発見し、その人から各自へのお薦めを導き出す機能が必要である。

3.1.3 お薦めの絞り込み

ユーザの情報処理能力には限界がある。つまり、ユーザにあまりにも大量のお薦めが送られても、ユーザはその全てを処理することができない。そこで、本モデルは、より自分に嗜好が近い人の作品の情報を優先して受け取り、その優先度によってユーザに対してお薦めの量を制御する機能を実現することで、ユーザの情報処理能力の限界に対応する。

3.1.4 網羅性の向上

本研究の目的は、ユーザが満足する作品を発見する可能性を向上させることだが、ユーザの満足とは、文字通りユーザが作品そのものの質のみの評価であり、その作品が多く買われている、多くメディアに取り上げられているといった二次的な評価は含まない。しかし、実情は、多く買われている作品ほど作品の露出度が多い。また、作品の露出度が高くなるほど、結果的に多く買われる、といったスパイラルが起きている。「作品の販売数」と「作品の露出度」は図3.1の様な関係にある。

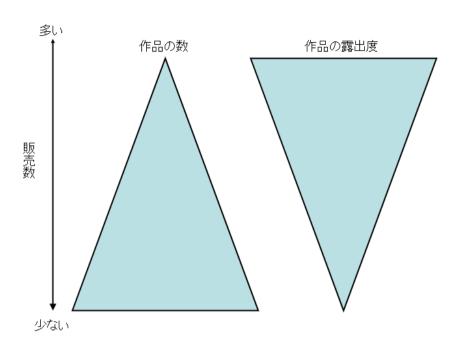


図 3.1: 販売数と露出度の関係

つまり、多く買われている作品ほど、ユーザの目に届きやすい。そこで、ユーザの傾向を「販売数の多い作品(メジャー)のみ鑑賞する人」「販売数の少ない作品(マイナー)のみ鑑賞する人」「メジャー/マイナーを意識しない人」と大きく3つのタイプにわけ、それを図上に示すと図3.2の様になる。

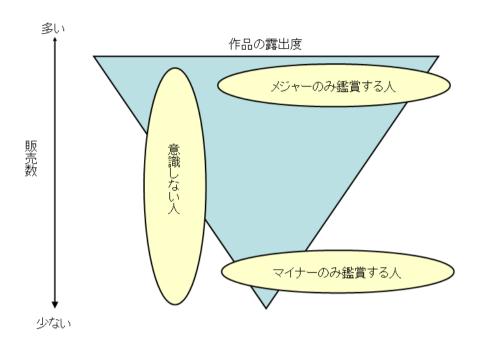


図 3.2: 3 つのタイプのユーザの傾向と露出度の関係

この時、各傾向における状況は以下の表にまとめられる。

表 3.1: 鑑賞の傾向とその状況

ユーザの傾向	状況
メジャーのみ鑑賞する人	現状では、メジャーな作品の情報は多いので満足
	する作品に出会う可能性が高い。
マイナーのみ鑑賞する人	満足する作品の情報を入手することが難しい。
メジャー/マイナーを意識しない人	メジャーな作品の情報しか手に入れられず、満足
	する可能性のあるマイナーな作品の情報を入手す
	ることが難しい。

本は、メジャーな作品だけに限らず、メジャー/マイナーを網羅する中から、ユーザが満足する可能性の高い作品をお薦めとして提示する必要がある。

3.1.5 探索空間の拡張

本研究の目的達成のアプローチは、各自が満足する作品を他者に伝えることであると述べた。より多くの人の中から、自分に嗜好が近い人を発見していくためには、より多くの人に各自が満足する作品を伝えなければならない。よって、直接の知人以外にも、各自が満足する作品を伝える機能を、本モデルは備える必要がある。

3.2 探索的お薦め導出手法

本節では、3.1で述べた要件定義を具体化する方法を述べる。

3.2.1 サンプルの伝達

ユーザは、日常、自分の好きな曲を聴く、好きな本を読むといった、自分の嗜好を満足させるための行動を何かしらとっている。本モデルは、この行動の痕跡、すなわちユーザが満足した作品の情報を、伝達・収集することで嗜好が近い人を発見する。本モデルでは、この行動の痕跡をサンプルと呼ぶ。そして、直接サンプルをやりとり可能であるユーザを知人と定義し、サンプルの収集に知人関係を用いる。

3.2.2 サンプルからの共通性による導出

新しい作品を発見する際、嗜好が似た知人から新しい作品を教えてもらうということがある。また、まったく嗜好の共通性のない人に教えてもらう作品より、共通性の高い人に教えてもらう作品の方が満足する可能性が高い。つまり、嗜好の共通性が高い人程、その新しい行動に共感を覚える可能性が高いといえる。

図3.3は、嗜好の共通性が高い人程、その新しい行動に共感を覚える可能性が高いことを、ビー玉のみせっこを用いて表したものである。

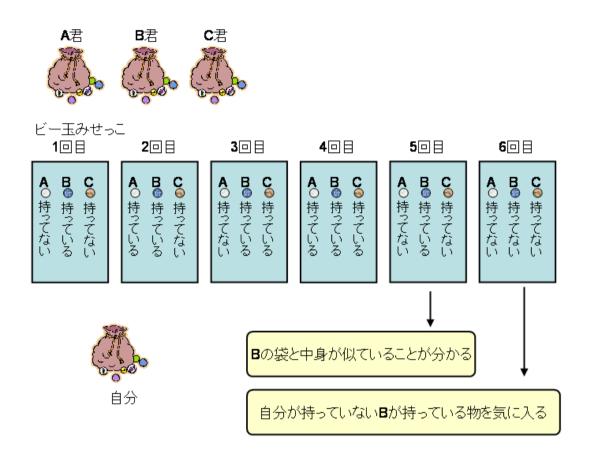


図 3.3: ビー玉のみせっこ

図3.3では、相手が持っているビー玉を袋から1つ1つ取り出して見せ、その度に自分の袋の中にそのビー玉があるかどうかを確認している。これを繰り返していく中で、どうも相手の中の一人の袋が自分と似ていることが分かっていく。そして、ある時自分が持っていないビー玉を袋が似ている相手に見せられたとき、それが自分も欲しいものかもしれない、と判断している。

この考え方は、検索サイトや SPAM フィルターなどで用いられる [8],[9] ベイズ理論 [10] に似ている。ベイズ理論とは、未来の出来事の確率はその事象の過去の発生頻度によって求められるという考えである。本モデルに適合すれば、未来の出来事の確率を求めるために用いる過去の発生頻度に該当するものは、嗜好の共通性となる。嗜好の共通性とは、届いたサンプルの蓄積によって推察される他ユーザの嗜好と、自ユーザの嗜好の重なりとなる。この重なりが、他ユーザとの嗜好の近さを表している。この重なりが大きい、すなわち嗜好の共通性が高い他ユーザほど、届いたサンプルを自ユーザが知らなかったとき、そのサンプルは、自ユーザが満足する作品である可能性が高い。

本モデルは、嗜好の共通性を元に他ユーザとの嗜好の近さを測り、嗜好が近い人のサンプルの蓄積の中で自ユーザが知らない作品を、お薦めとして導き出す。

3.2.3 お薦めの絞り込み

本モデルは、先の方法によりユーザ毎に嗜好の近さを測り、嗜好が近い順に他ユーザを順位付けする。そして、図 3.4の様に、この順位付けと蓄積されたサンプルを元に、お薦めを導出する際の絞り込みを行い、お薦めリストとしてユーザに提示する。この方法により、ユーザは自分が満足する可能性の高い順に、未知の作品を知ることが可能になる。

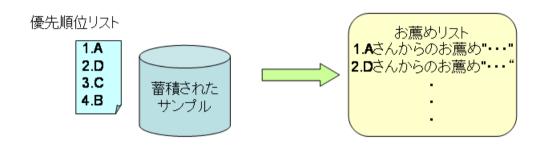
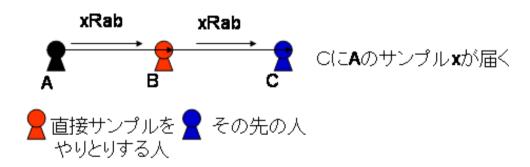


図 3.4: 優先順位付け

3.2.4 探索空間の拡張

本モデルでは、直接サンプルをやりとり可能であるユーザを知人と定義し、サンプルの 収集に知人関係を用いると述べた。知人関係を用いることで、直接の知人以外に自分と嗜 好が近い人を発見した際に、知人に紹介をしてもらうという行動も期待できる。

サンプルが人から人へ直接伝わる距離を 1 ホップと定義する。本モデルは、サンプルを知人だけでなく数ホップ先まで推移させることで、より多くの人の中からユーザと嗜好が近い人を探すことを可能にし、探索空間を拡張する。図 3.5はサンプルの推移の流れを示したものである。



xRab:AがBに送った作品xのサンプル

図 3.5: サンプルの推移

3.3 モデルの優位性

3.3.1 発見的である

2章において既存手段の問題点として発見的でないことを挙げたが、本モデルは、相手の行動の痕跡を蓄積し、自分の行動と照らし合わせることによって、新しい満足する可能性の高い作品を発見する。つまり、新しい作品を発見する際、なにかそのものを示すようなキーワードを必要とすることない。ユーザは、自分の嗜好を満足させる行動のみで新しく作品を発見することができる。

3.3.2 網羅性の向上

本モデルでは、2章で述べた既存のリコメンドサービスで用いる手法ではなく、探索的お薦め導出手法を用いて、嗜好が近い人の行動からお薦めを作り出す。図3.6は、各モデルを簡略に図示したものである。

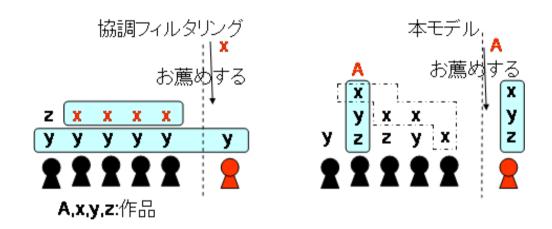


図 3.6: 既存のリコメンドサービスのモデルとの比較

図3.6では、左側が既存のリコメンドサービスのモデルであり、右側が本研究が提案するモデルである。

既存のリコメンドサービスの手法を用いたお薦めでは、他ユーザの中で、作品 y と共に作品 x を持っているユーザが多いため、作品 y を持つ自ユーザに対して作品 x がお薦めとして提示されるという仕組みである。一方、本モデルを用いたお薦めでは、所持している作品が自分と一番共通している他ユーザが持っている作品中から、自ユーザがまだ持っていない作品がお薦めとして提示される。

つまり、既存のリコメンドサービスを用いた仕組みでは、図 3.6の左側において一人しか持っていない作品 z は、お薦めとして提示されることはないが、本モデルでは、図 3.6の右側において一人しか持っていない作品 A もお薦めとして提示することが可能である。

この仕組みにより、本モデルは、2.2.2で述べた網羅性に乏しいという問題を解決する。

3.3.3 探索空間の拡張

本モデルでは、サンプルをユーザの知人関係の上で推移させることで、より多くの人に サンプルを届けると述べた。推移により、より多くのユーザから自分と嗜好が近い人を探 すことを可能にする。この機能によって、2章で述べた探索空間が狭いという問題を解決 する。

3.4 実現可能性の検討

3.4.1 嗜好の近さの測り方

本モデルにおける嗜好の近さとは、ユーザ同士の嗜好の重なりであると述べた。つまり、ユーザ p1 から見たユーザ p2 の嗜好の近さは、p1 が好きな物の集合 s1、p1 にサンプルとして届いた p2 の好きな物の集合の一部 s2 とすると、p1 から見た p2 との嗜好の近さ proximity(p1,p2) は、

$$proximity(p1, p2) = \frac{|s1 \cap s2|}{|s1 \cup s2|}$$

となる。

しかしこの場合では、p2 からサンプルが届くたびに s2 の要素数が変わる。本モデルを忠実に再現するために、集合が変化するたびにユーザ毎に重なりを計算するのは計算量が大きくなってしまう。そこで、実装的な単純さを目的として、積集合の要素数のみを用いる「ポイント制モデル」を検討した。つまり、proximity(p1,p2) は、

 $proximity(p1, p2) = |s1 \cap s2|$

となる。この場合、他ユーザから届いたサンプルが自ユーザの集合に入っていたら proximity の値が 1 増える。他ユーザとの嗜好の近さは、このポイントの大きさとするという方法である。しかし、この動作のみでは、同じ行動をとり続けるユーザのポイントが行動の都度増加してしまう。本モデルの嗜好の近さの測り方に近づけるためには、この繰り返しによるポイントの増加は防ぐ必要がある。よって、他ユーザから届いたサンプルがすでにそのユーザから届いたサンプルの蓄積の中にあるかどうかを確認した後に、ポイントの増加処理を行わなければならない。

そこで、「ポイント制モデル」の妥当性を検証するために、本モデルを忠実に再現したモデル(以下、「忠実モデル」)と「ポイント制モデル」の間に生じる差を実験で明らかにした。それぞれ特性をもったユーザp1, p2, p3, p4, p5において、「忠実モデル」と「ポイント制モデル」がどのように働くかを検討した。下図は、各ユーザの好きな物の集合s1, s2, s3, s4, s5の重なりを表したものである。

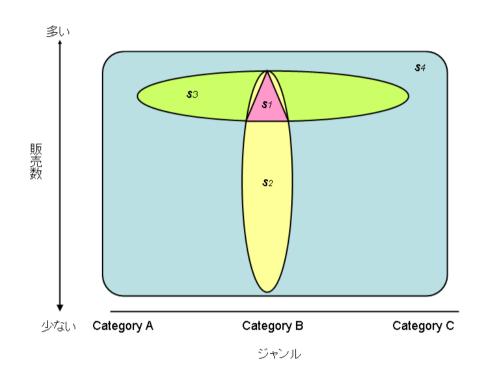


図 3.7: 各ユーザの好きな物集合の重なり

各ユーザの要素数、特性をまとめたものが表3.2である。

表 3.2: 各ユーザの要素数、特性一覧

	集合の要素数	s1 との積集合の要素数	特性
p1	100	-	特定ジャンルの販売数の
			多いものを聴く
p2	300	100	特定ジャンルを販売数の
			多少に関わらず聴く
р3	300	100	複数ジャンルの販売数の
			多い物を聴く
p4	1000	100	複数ジャンルを販売数の
			多少に関わらず大量に聴
			<

この時、ユーザp1から見た他のユーザの嗜好の近さを2つの方法それぞれで求めると表3.3になる。

表	3.3:	ユーザ	p1 から	見た各ユ	ーザの近さ
---	------	-----	-------	------	-------

方法	p1 か ら見た近さ
和集合/積集合	p2 = p3 > p4
ポイント制	p2 = p3 = p4

ポイント制モデルの場合、集合が大きいp4 は忠実モデルの近さでは、p2, p3 よりも遠いはずだが、ポイント制を用いたために同じ近さになってしまっている。そこで、近さを求めるために用いるサンプルの数を限定することを考える。ある一定の期間において、人が鑑賞できる作品の数というものには限界があり、その数はばらつきはあるが、値は大きくかけ離れることはないためである。よって、近さを求めるために用いるサンプルの数を、最新のサンプルから、狭い幅に存在する値xとする。他ユーザの集合の大きさがほぼ同じになり、先の図3.7は図3.8の様になる。

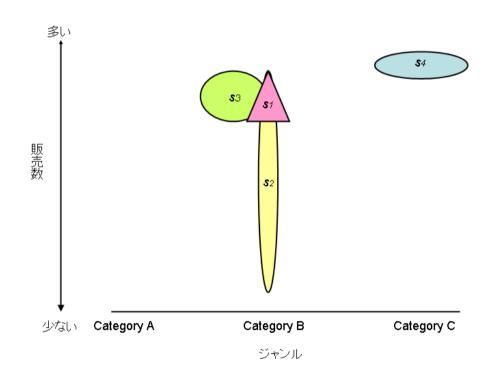


図 3.8: 各ユーザの好きな物集合の変化

図 3.8における集合 s2, s3, s4 の要素数はほぼ同じである。この時、前者の式

 $proximity(p1,p2) = \frac{|s1 \cap s2|}{|s1 \cup s2|}$

は、\$2の要素数が各ユーザともほぼ同じになるため、積集合の要素数によって嗜好の近さ

を表すことができる。よって、ポイント制によっても嗜好の近さを求めることが可能である。なお、当然この図 3.8は、各ユーザの最新の一定数の行動における特性を表したものであり、必ずしもこの図になるとは限らない。

本研究では、後者のポイント制を実装において用い、一定の数xの最適値は、今後の検討課題とした。

3.4.2 お薦めの生成方法

次に、他ユーザとの嗜好の近さを測った上で、他ユーザのサンプルの蓄積から、いかにしてお薦めリストを生成するかについての検討を行った。3.1.3で述べたように、嗜好が近い相手のサンプルは、満足する可能性が高い作品といえるため、より多く提示されるべきである。蓄積からサンプルを抽出する方法として以下の方法が考えられる。

- サンプルが届いた順に表示
- 同じ作品のサンプルが届いた回数順に表示
- ランダムに抽出して表示

1つ目は、短い期間でみた場合、最近の作品のサンプルのみからお薦めを生成した場合、お薦めの変化が乏しくなってしまい、作品との出会いの可能性を低くしてしまう。

2 つ目は、新しく追加されたばかりでサンプルが生成された回数が少ない作品の優先順位が下がるため、やはり作品との出会いの可能性を低くしてしまう。

よって、作品との出会いの可能性を高くする上では、嗜好が近い人の蓄積されたサンプルの中からランダム抽出を行う3つ目の方法が最良の方法であるといえる。

また、嗜好が近い人のサンプルをより多く提示する方法として、嗜好の近さによって 1 つのリストにおける数の枠を分配する方法をとる。お薦めとして提示するリストの作品数を NoR、ユーザa にサンプルを送ったユーザの集合を、 $P_a=\{p_1,\cdots,p_{|p_a|}\}$ 、ユーザpi のユーザa との近さを proximity(a,pi) としたとき、ユーザpi のリストにおける枠 NoRpi は

$$NoRpi = \frac{proximity(a,pi)}{\sum_{x=1}^{|p_a|} proximity(a,px)} NoR$$

となる。

第4章 モデルの比較検証実験と評価

4.1 検証のための本モデルの適用シーン

本研究では、本モデルが要件を満たしているか検証するため、適用するシーンを定め、システムとして実装し、検証実験を行った。本モデルが有効に働くのは、1章で述べたように、次々と新しい作品が登場し、嗜好によって作品の評価が異なるシーンである。本研究では、音楽を具体的な適用シーンとした。

検証実験では、ユーザ同士がサンプルをやり取りするシュミレーション環境を作り上げ、既存のリコメンドサービスによる手法と本モデルの手法の比較を行った。既存のリコメンドサービスの手法には、2章で述べた「この商品を買った人はこんな商品も買っています」という買った物の組み合わせを元に商品を薦める手法を用いた。

シーンが音楽であるため、本モデルが定義するサンプルは、ユーザが聴いている曲目と定義する。シュミレーション環境では、ユーザの音楽の再生環境として Apple 社が提供する音楽再生ソフトウェアである iTunes[11] を用いた。またサンプルをやり取りする環境として、より被験者を集めるために、IM の中では大きなシェア [12] を持つ Microsoft 社の IM クライアントソフトウェア MSN Messenger[13] を選択した。

4.2 実験概要

実験は、日常 IM を利用する、かつ、iTunes を用いて音楽を聴いているという条件に当てはまる 35 人を対象に行った。35 人は、17 人ずつの 2 つのコミュニティと筆者で形成される。両コミュニティとも、コミュニティ内での知人関係のみ存在し、コミュニティをまたいだ知人関係は存在しない。筆者のみが 2 つのコミュニティをまたいで知人関係を持つ。

被験者には、

- 自身の iTunes の全曲リスト (以下ライブラリ)
- MSN Messenger を使ってコンタクトを取れ、iTunes を使って PC で音楽を日常的に 聴いている人のリスト

を提出してもらった。その際、被験者にiTunesのライブラリと知人関係を提供してもらう方法として、webページからアップロード・登録を行うサイト(図4.1)を作成した。

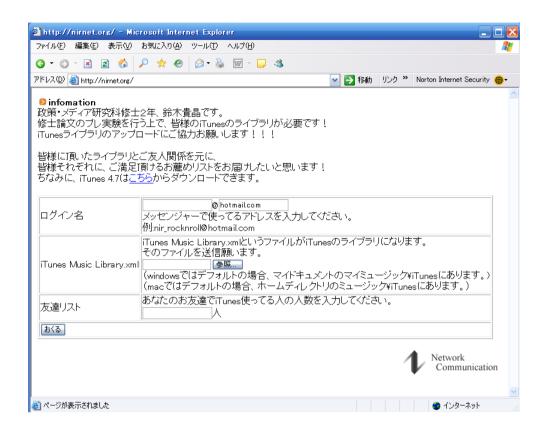


図 4.1: iTunes ライブラリ、知人関係登録ページ

このサイトより、

- iTunes のライブラリである iTunes Music Library.xml というファイル
- 先の対象者の MSN Messenger で用いているメールアドレス

を登録してもらった。

このサイトから登録されたデータを処理し、データベースに登録することで、擬似的に被験者各自のライブラリの全曲目がサンプルとして、他ユーザに行き渡り、他ユーザとの嗜好の近さを測ることが可能な環境を構築した。

この環境下において、

- 本モデルの手法を用いたお薦め提示システム
- 2章で述べた既存のリコメンドサービスの手法を用いたお薦め提示システム を実装した。そして、実装を用いた検証及び2つの実装の比較検証を行った。 これにより、
 - 本モデルが要件を満たしているか
- 既存手段より本モデルの方が満足する作品を発見する可能性が高いかの 2 点について評価を行った。

4.3 シュミレーション環境の実装

4.3.1 プログラミング言語

本実装は、PHP 4.3.10 を用いて実装した。

4.3.2 ソフトウェア

本実装は、以下の環境で実装した。

表 4.1: 実装環境

OS	Vine Linux 3.1
RDBMS	PostgreSQL 7.4.6
WWW Server	Apache 2.0.52

4.3.3 データベーススキーマ

本実装のデータベースのスキーマを、表4.2~4.7に示す。

表 4.2: Table "user_info"

Column	Type	Modifiers	内容
uid	character(50)	primary key	ユーザ識別子 (MSN Messenger のアドレス)

表 4.3: Table "social"

Column	Type	Modifiers	内容
X	character(50)	not null	ユーザ識別子
У	character(50)	not null	ユーザ識別子

表 4.4: Table "song"

Column	Type	Modifiers	内容
song_id	character(32)	primary key	曲識別子
name	character(255)	not null	曲名
artist	character(255)		アーティスト名
library_count	integer	not null	ライブラリに持っているユーザの数

表 4.5: Table "holding"

Column	Type	Modifiers	内容
song_id	character(32)	not null	曲識別子
uid	character(50)	not null	ユーザ識別子

表 4.6: Table "point"

Column	Type	Modifiers	内容
X	character(50)	not null	ユーザ識別子
У	character(50)	not null	ユーザ識別子
point	integer	not null	xとyのライブラリが共通する曲数

表 4.7: Table "song_relation"

Column	Type	Modifiers	内容
X	character(50)	not null	曲識別子
У	character(50)	not null	曲識別子
count	integer	not null	xとyが同時にライブラリに存在しているユーザ
			の数

4.3.4 本モデルによるお薦めの提示システム

本モデルによるお薦めの提示システムの実装について述べる。 まず、各被験者毎に他の被験者とライブラリで共通する曲の数を求め、表 4.6のテーブ ルに格納した。各被験者にお薦めを提示する際は、その被験者とのポイントの高い被験者順に、3章で述べた、

$$NoRpi = \frac{\textit{proximity}(\textit{a,pi})}{\sum_{x=1}^{|\textit{pa}|}\textit{proximity}(\textit{a,px})} NoR$$

の式によって、リスト内のライブラリから抽出する曲数を求める。そして、該当被験者の ライブラリから、求めた曲数分だけ、共通していない曲をランダムに抽出する。これを、 1つのリストの曲数を超えるまでポイントの高い被験者順に行い、お薦めリストとして被 験者に提示する。

4.3.5 既存のリコメンドサービスの手法によるお薦めの提示システム

既存のリコメンドサービスの手法によるお薦めの提示システムの実装について述べる。 まず、全被験者のライブラリから、曲の組み合わせを抽出し、表 4.7のテーブルに納め た。各被験者にお薦めを提示する際は、その被験者のライブラリの全ての曲毎に、 2 章で 述べた、

$$S_{aj} = \sum_{i \in I_a} (NoP(i,j))$$

の式によって、各作品の被験者に対するスコアを求める。そして、スコアの高い作品から順に、お薦めリストとして被験者に提示する。

4.4 要件を満たしているかの評価

4.4.1 探索空間の拡張

本モデルの要件として、探索空間の拡張を述べた。本モデルでは、サンプルの推移によって、探索空間の拡張を狙う。実験では、推移が実際に探索空間を拡張させるか検証した。1hop、2hop、3hopと推移させることで、どの程度探索空間が広がるかについて分析を行った。各被験者ごとの1hop、2hop、3hopのサンプルを届ける人数を求め、その総和をhop数ごとの変化のグラフにしたところ、以下の図4.2になった。

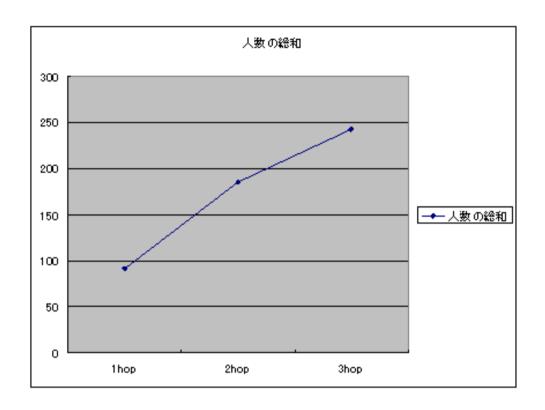


図 4.2: hop 数によるサンプルを届ける人数の総和の変化

本実験では、推移を行うことで探索空間が拡張している。なお、一般に知人関係のネットワークはスケールフリーネットワークであることが明らかになっている [14]。また、1hopから 2hop に比べ 2hop から 3hopへの増加は少ない。また、2hop の場合、自分の直接の知人以外に自分と嗜好が近い人が見つかった場合、間の共通の知人に紹介してもらうことが期待できる [6]。

4.4.2 網羅性の向上

本モデルの要件として、網羅性の向上を述べた。本モデルでは、嗜好が近い人のサンプルからお薦めを導き出すことで、既存のリコメンドサービスよりも網羅性を向上させることを狙う。ここでいう網羅性とは、多くの人のライブラリに存在しない曲であっても、お薦めとして提示されることである。実験では、本モデルの手法を用いたお薦め提示システムが既存のリコメンドサービスの手法を用いたお薦め提示システムよりも網羅性が高いことを検証するため、以下の計算を行った。

被験者一人あたりに1つのお薦めリストとして提示する曲数を10曲とする。曲毎の、その曲を持っている被験者の数を算出し、10曲分の総和を求めた。この総和を、全被験者分足した。この計算を、本モデルの手法によるお薦め提示システムと既存のリコメンドサービスの手法によるお薦めシステムの両方で行った。本モデルの手法によるお薦めリス

トは、提示毎にリストが変化するため、算出した値も変化してしまう。そのため計算を 10 回行い、その平均値を用いた。各システムの結果は、表 4.8となった。

表 4.8: 各手法の延べ所持人数

	延べ所持人数
本モデル	547
既存手法	1278

結果より、本モデルが既存のリコメンドサービスよりも、お薦めとして提示した作品の延べ所持人数が少ないことが、明らかとなった。延べの所持人数が少ないということは、多くの人のライブラリに存在しない曲が、既存手法に比べお薦めとして提示されたことを示している。よって、本モデルは、網羅性の向上の要件を満たしていることが実証できた。

4.5 既存手段に対する本モデルの優位性の評価

4.5.1 アンケートの方法と仕様

本モデルが、既存のリコメンドサービスの手法よりも有用であることを検証するため、 本モデルに基づいたお薦めと既存のリコメンドサービスに基づいたお薦めを共に提示し、 満足度を調査するアンケートを行った。

本アンケートは、iTunes のライブラリ及び MSN Messenger の知人関係を提供してくれた被験者に対して行い、全て Web 上にて回答してもらった。提供者 35 人中、29 人よりアンケートの回答を得ることが出来た。

被験者には、3つの質問に答えてもらった。 まず、問1は、自分が、

- メジャーのみ聴く
- マイナーのみ聴く
- ◆ メジャー/マイナーを意識しない人

の3つの中でどれに当てはまると思うかについて答えてもらい、ユーザのタイプを分類した。

次に、問2では、商品の結びつきによるお薦めと本モデルによるお薦めをそれぞれ10曲ずつ提示した。そして、未知の曲が提示されていた場合は、その曲について試聴やアーティストの情報収集などを、各自に行ってもらった。その後、曲毎に、

• 未知の曲で、試聴が出来た

- 未知の曲で、検索サイトによってのみ調べた
- 既知の曲で、既に持っている
- 既知の曲で、持ってはいない

の4つに分類し、その曲に「満足した/興味を持った」か「満足しなかった/興味がない」 を二者択一で答えてもらった。

最後に、問3では、本モデルによるお薦めリストの更新を何度か行ってもらい、それを踏まえ、このリストに対しての感想を答えてもらった。

4.5.2 アンケートの収集結果と考察

アンケートの集計結果を分析し、考察していく。 問1の集計結果は、表4.9となった。

表 4.9: 問1の集計結果

メジャー	5
マイナー	2
意識しない	22

問1の結果を用いて、ユーザのタイプ別に問2の結果を集計すると、表4.10となった。 表中、 は好評価だったこと、×は好評価でなかったことを指している。

表 4.10: 問 2 の集計結果

タイプ	実装	未知	ロ・試聴	未知	・検索	既知	印・所有	既知	・未所有
917	夫衣		×		×		×		×
7 = 3 + _	本モデル	4	3	19	5	4	0	11	4
メジャー	既存手法	0	0	1	4	14	0	10	21
マイナー	本モデル	1	0	9	1	2	0	7	0
\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	既存手法	0	0	0	0	2	0	5	13
意識しない	本モデル	14	3	106	36	32	0	24	5
	既存手法	2	2	8	27	67	5	22	87

この結果を用いて、本モデルの優位性について評価を行っていく。

まず、本モデルが未知の作品をユーザに提示出来たかを比較する。ユーザにお薦めとして 提示した作品を未知と既知で分け、延べの作品数を本モデルと既存手法を比較すると、表 4.11となった。

表 4.11: 未知の作品数と既知の作品数の比較

	未知	既知
本モデル	201	89
既存手法	44	246

表 4.11より、本モデルが既存手法よりも、未知の作品をユーザに提示したことが明らかになった。

次に、ユーザに提示した未知の作品の満足度を比較する。未知の作品を好評価だったものとそうでなかったものと分け、延べの作品数を比較すると、表 4.12となった。

表 4.12: 未知の作品の満足度の比較

	未知の作品の好評価
本モデル	159
既存手法	14

表より、本モデルが既存手法よりも、ユーザに未知の満足する作品をお薦めとして提示 したことが明らかになった。

3章において、本モデルは、

- メジャーのみ鑑賞する人
- マイナーのみ鑑賞する人
- メジャー/マイナーを意識しない人

のどのタイプのユーザにおいても、ユーザが満足する作品を発見する可能性を向上させることを目的とすると述べた。そこで、本モデルが既存手法に比べ、どのタイプにおいても未知の満足する作品をユーザにお薦めとして提示できたかを既存手法と比較した。タイプ別に好評価の未知の作品の一人あたりの平均作品数を比較した。なお、既知の好評価の作品数の一人あたりの平均も合わせて表記した。結果は、表 4.13となった。

衣 4.15: タイノ別の木和の作品の両足反の比較					
		好評価の未知の作品数の平均	好評価の既知の作品数の平均		
~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~	本モデル	4.6	3.0		
メジャー	既存手法	0.2	4.8		
マイナー	本モデル	5.0	4.5		
	既存手法	0	3.5		
意識しない	本モデル	5.5	2.5		
	既存手法	0.5	4.0		

表 4.13: タイプ別の未知の作品の満足度の比較

表 4.13より、本モデルはどのタイプのユーザにおいても、既存手法より未知の満足する 作品をお薦めとして提示したことが明らかになった。また、既知の作品においては、両手 法に明らかな差異はなかった。

以上の比較により、本モデルが既存手法よりも、ユーザが満足する作品をお薦めとして 提示することが可能であることを実証した。

最後に、問3の質問では、被験者から表4.5.2の様な感想が得られた。感想は、原文の まま載せている。

表 4.14: 被験者からの感想

全く知らない人からいくつかお薦めが表示された。自分が知ってる好きな曲も入っていた。知らない曲もたくさんあって更新するのが楽しかった。(同意見6件)

前から趣味が似ていることを知っていた友人から、まだ自分が手に入れていなかった曲がたくさんお薦めされた。欲しいと思ったけど、そのまま忘れていたものがあって、すごいと思った。ちゃんと似てる人がわかるんですね。(同意見7件)

自分の持っている曲リストやアーティストと重複が少ないので,新しく開拓するときには非常に有意義だと思う.(同意見2件)

既に持っている曲がリストアップされていた場合があった(坂本龍一とRyuichi Sakamoto, TMN と TM NETWORK)。(同意見 5 件)

邦楽/洋楽,ジャンルなどが選べる・表示できるといいかと思います。(同意見5件) 曲を聴く(借りる、購入する)という行為は非常にコストが高く、お薦めされる曲数が 多いと一曲に対する試聴モチベーションが分散してしまう。曲ごとのレーティングが数 値として表現されていると、飛びぬけている1.2 曲を試聴する気が増すと思います。

お薦め一覧に、様々なアーティストがバランス良く提示されている。既に知っているアーティストと、新規に教えられるアーティストの割合が 7:3 くらいなので、提示されたリストを聴くモチベーションも高くなる。また、推薦者の表示は興味深い。ただ、現在の結果を見ると、お互い多くのプレイリストを持っている人同士は、お互いに「近い存在」となってしまう(なりすぎてしまう)感じも受ける。

知っている人しか出てこないけれど、知り合いの持つ知らなかった趣味を垣間見られるのは楽しいことでした。また、知人なだけに聞かせて欲しいとかリクエスト出しやすいと思う。(同意見2件)

被験者の感想を考察していく。

- 網羅性の向上及び探索空間の拡張
 - 一 被験者の感想から、本モデルの優位性である網羅性の向上及び探索空間の拡張 についての言及が得られた。
- 嗜好の近さの測定
 - 本モデルによる嗜好の近さの測定が、有効に働いていることについての言及が得られた。
- 本モデルによるお薦めの有用性
 - 自分が好きな曲を探していく上で、本モデルが有用であるという感想が得られた。
- 曲の同定

- 自分のライブラリに含まれている曲が、お薦めとして提示されたという感想が複数あった。これは、同じ曲を別の曲と認識してしまったためである。曲名・アーティスト名などの情報は、iTunesの場合、CDDB[15]より取得する、もしくはユーザが手入力するという方法で定められる。しかし、CDDBでは、同じ曲であっても時期によって表記が違うことが稀にある。また、ユーザが手入力した場合には、表記の仕方がまちまちであるため、曲の同定が難しい。しかし、同定に失敗するのはこの様な特殊な時であり、ほとんどの場合、自分のライブラリの曲がお薦めされることはない。

• 被験者、サンプル数の不足

- ジャンルの選択、多曲との数値による視覚的差別化を求める声が存在した。しかし、本来、本モデルはユーザ自身の嗜好に近いユーザのサンプルからお薦めを導出するため、ユーザが偏ったジャンルの嗜好を持っていた場合は、同じような偏りを持ったユーザからのサンプルの蓄積よりお薦めが導出されるはずである。にもかかわらず、この感想を持った被験者は、同じような偏りを持つ被験者を発見できなかったということである。これは、被験者の数及びライブラリの曲数の不足が原因で、モデルの特徴が上手く活かされなかったことを表している。

● 一人あたりのライブラリの曲数のばらつきによる弊害

- この問題は、3章の実現方法の検討で述べた問題である。今回の実験では、一人あたりのライブラリの曲数のばらつきが多かったため、ライブラリの曲数が極端に多い被験者同士の嗜好の近さの値が大きくなった。3章でも述べたが、一定期間における人間の聴取できる曲数は限られているため、嗜好の近さを測る上でのサンプルの鮮度によって、ライブラリの曲数のばらつきを均すことができる。有効な一定期間の長さの検討は、今後の課題である。

● 知人からの推薦の利点

本モデルは、知人関係の上でサンプルを伝達・蓄積し、蓄積からお薦めを導出するため、お薦めが直接の知人のサンプルから導き出される可能性がある。直接の知人からの場合、実際にその曲を試聴する機会に恵まれやすい。また、普段は知ることがなかったが、実は嗜好が近かった知人の存在を知ることもできる。

以上より、被験者の感想からも、本モデルが既存手法よりユーザが満足する作品を発見 する可能性を向上させていることを実証した。

4.6 実験結果のまとめ

本実験では、

- 本モデルが要件を満たしているか
- 既存のリコメンドサービスの手法に対する本モデルの優位性

の2点について評価を行った。

その結果、本モデルは3章で述べた要件を確かに満たし、既存のリコメンドサービスの手法と比較し、ユーザが満足する作品を発見する可能性を高めたことを実証した。

第5章 設計

本章では、3章で提案したお薦め導出手法を用いたモデルのシステム設計について述べる。

5.1 システムの全体構成

本システムの構成を図5.1に示す。

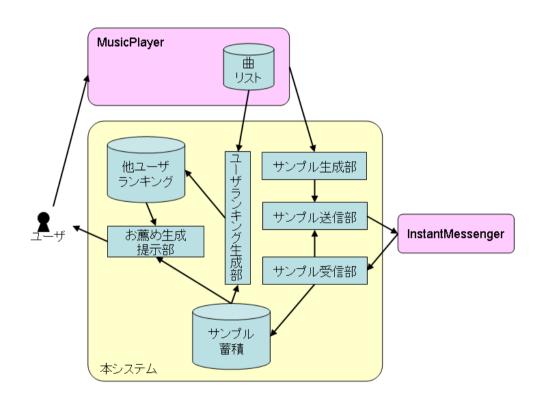


図 5.1: システム構成図

本システムは、次の機能により構成される。

- サンプル生成部
- サンプル送信部

- サンプル受信部
- ユーザランキング生成部
- お薦め生成提示部

以下、各部の機能を説明していく。

5.2 各部の機能

5.2.1 サンプル生成部

サンプル生成部は、ユーザが音楽生成ソフトウェア (以下 MusicPlayer) で聴いている曲の情報を取得し、必要な情報を付加してサンプル送信部へ渡す機能をもつ。サンプルのデータ構造は、表 5.1のようになる。

フィールド名	内容
UserID	サンプルを生成したユーザの InstantMessenger におけるの識別子
Song	聴いている曲の名前
Artist	聴いている曲のアーティスト名
Album	聴いている曲のアルバム名
TrackNo	聴いている曲のトラックナンバー
TrackTime	聴いている曲の長さ
hop count	サンプルが推移した回数を表す。1回知人を経由する度に値が1減少す
	る。hop count が0になったサンプルはその時点で廃棄される。

表 5.1: サンプルのデータ構造

5.2.2 サンプル送信部

サンプル送信部は、サンプル生成部及びサンプル受信部から渡されたサンプルを InstantMessenger に渡し、登録されている知人への送信を依頼する。

5.2.3 サンプル受信部

InstantMessenger から渡されたサンプルを解析し、処理を行う。まず、届いたサンプルの hop count から 1 を引く。その結果が 0 でない場合、hop count から 1 を引いた状態のサンプルをサンプル送信部へと渡す。次に、サンプルから hop count を除いたものを蓄積

する。その際、同一ユーザから同一の曲のサンプルが届いた場合は、蓄積内にある該当サンプルの TimeStamp のみ更新する。

5.2.4 ユーザランキング生成部

ユーザランキング生成部は、ユーザの MusicPlayer の曲のリスト(ライブラリ)と他 ユーザからのサンプルの蓄積を用いて、他ユーザのランキングを生成する。他ユーザ毎 に、ユーザとの嗜好の近さを、モデルで定義した方法で求める。求めた値を元に、他ユーザをユーザの嗜好が近い順に並びかえ、ランキングを生成する。ランキングの生成は、お 薦め生成時に行われる。

5.2.5 お薦め生成提示部

お薦め生成提示部は、サンプルの蓄積と他ユーザのランキングから、ユーザへのお薦めリストを生成し、提示する。リストに表示するお薦め曲数はユーザが指定可能である。ユーザから指定された曲数を全体とし、他ユーザのランキングを元に他ユーザ毎に全体を振り分け、振り分けられた曲数分、当該ユーザからのお薦めを表示する。お薦めはサンプルの蓄積からランダムに抽出する。なお、ユーザの MusicPlayer の曲のリストに既に含まれている曲は、サンプルから抽出されない。

5.3 処理の流れ

ユーザが本システムを利用する際の一連の流れを図5.2に示す。

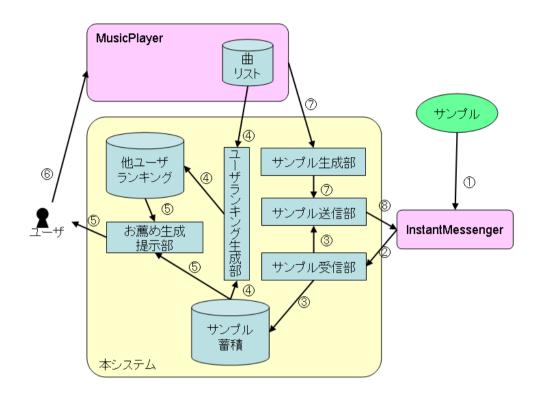


図 5.2: 利用の流れ

以下、流れを説明していく。

- 1. 他ユーザからのサンプルが InstantMessenger に届く。
- 2. InstantMessenger が、届いたサンプルをサンプル受信部へと渡す。
- 3. サンプル受信部がサンプルの蓄積を行う。また、hop count を調べ、他のユーザへ さらに転送が必要な場合は、サンプルをサンプル送信部に渡す。
- 4. ユーザランキング生成部が、ユーザの MusicPlayer の曲のリストとサンプルの蓄積 から、他ユーザのランキングを生成する。
- 5. お薦め生成部が、他ユーザのランキングとサンプルの蓄積からお薦めリストをユーザに提示する。
- 6. ユーザが、提示されたお薦めリストの曲を吟味し、満足したものが MusicPlayer の曲リストに追加される。
- 7. サンプル生成部が、ユーザが聴いている曲からサンプルを生成し、サンプル送信部へと渡す。

8. サンプル送信部は、渡されたサンプルをInstantMessengerへ渡し、他のユーザへの送信を依頼する。

この1~8の流れは常に繰り返される。この繰り返しの中で、ユーザは、届いたサンプルの曲を MusicPlayer の曲のリストに追加することもある。つまり、お薦めされた曲を気に入って、その曲を購入・取得することは起こりうる。届いたサンプルの曲を、MusicPlayerの曲のリストに追加した場合、本モデルのアルゴリズム上、そのユーザは、そのサンプルを生成元のユーザとの嗜好の近さが近づくことになる。つまり、ユーザの行動によって、ユーザのランキングは常に変化していく。

この変化は、ユーザの行動の結果として、よりユーザに近い他ユーザのサンプルが提示されるようになることを意味する。

第6章 実装

本研究では、前章で設計したシステムを、IM クライアントソフトウェアである wija[16] のプラグイン"Tunes プラグイン"として実装した。本章では、実装環境及び各モジュールの動作について述べる。

6.1 実装環境

6.1.1 プログラミング言語

本実装は、Java 言語 (Java 2 Standard Edition) を用いて実装した。

6.1.2 ソフトウェア

本実装は、IM クライアントソフトウェアとして wija のプラグインであり、音楽再生ソフトウェア"iTunes"と協調して動作する。

6.2 実装の構成

本実装の構成を、図6.1に示す。

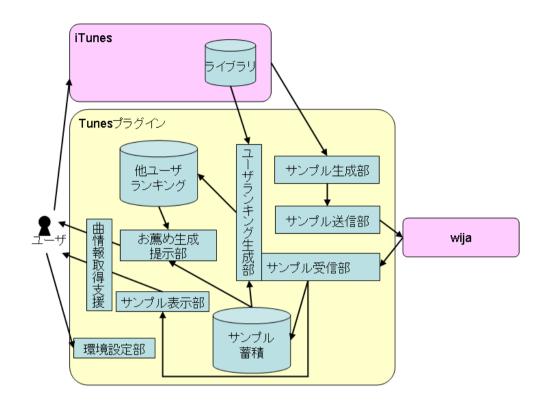


図 6.1: 実装の構成

本実装は、次の要素によって構成される。

- iTunes
- wija
- Tunes プラグイン

以下、各構成要素について述べる。

6.2.1 iTunes

iTunes とは、Apple 社の音楽プレーヤーソフトであり、MacOS 版及び Windows 版が 提供されている。音楽配信サービスの「iTunes Music Store」や携帯音楽プレーヤーの「iPod」と協調動作させることが可能で、音楽のダウンロード購入や CD からの読み込み、再生、CD への書き込みや携帯音楽プレーヤー iPod への送信まで、一つのソフトで統合的に管理することができるという特徴がある。また、CD から曲を読む込む際、CDDB を用いて曲名、アーティスト名などを取得し管理することができる。

本実装では、CDDBが提供する曲名、アーティスト名の情報を用いて、曲を一意に識別している。同社はiTunes 用の開発キットも提供しており、開発者はこれを用いることで容易にiTunes と他のプログラムを連動させることができる。

6.2.2 wija

wija とは、Media Art Online¹にて、Java 言語により開発された XMPP (eXtensible Messaging and Presence Protocol)[17][18] クライアントである²。XMPP とは、インスタント・メッセージングおよびプレゼンスのための、XML ベースのオープン・プロトコルであり、多数の XMPP クライアントが開発されている。wija の実行に必要な環境は、Windows 環境では、J2SE 5.0 が、MacOS X 環境では、Java Update 1.4.2 Update 1 以上が適用されている必要がある。wija の特徴として、下記が挙げられる。

- ◆ オープンで拡張可能なプロトコル XMPP に基づいたインスタントメッセージング とプレゼンス共有
- PGP (GnuPG) による暗号化と署名
- プラグインにより拡張可能
- 同梱されたiワットプラグイン による補完通貨機能
- GNU GPL により、自由に利用、改変、再配布が可能

次に述べる Tunes プラグインは、上記の第 2 の特徴「プラグインにより拡張可能」に基づき、wija のプラグインとして実装されている。図 6.2は、wija のメインウィンドウである。

 $^{^1}$ インターネットを用いたメディアアートの制作や配布をめぐる諸問題を解決していくことを目的としたコミュニティ 1 http://www.media-art-online.org

²現在もなお活発に開発が続けられている。

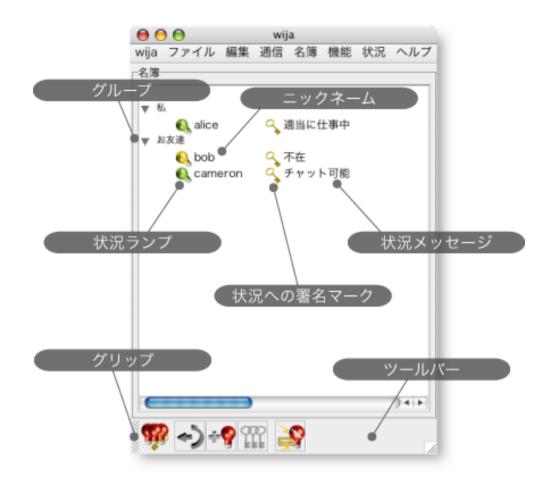


図 6.2: wija

6.2.3 Tunes プラグイン

wija のプラグインである。本研究で提案するモデルの大部分は Tunes プラグインによって実現されている。 Tunes プラグインは、次のモジュールによって構成される。

- プラットフォーム抽象化レイヤ
- サンプル生成部
- サンプル送信部
- サンプル受信部
- ユーザランキング作成部
- お薦め生成提示部

- 曲情報取得支援部
- サンプル表示部
- 環境設定部
- 運用実験のための機能

以下、Tunesプラグインの機能概要について説明していく。

6.3 各モジュールの動作

各モジュールの動作を説明に用いる用語の解説と定義を行う。

6.3.1 用語解説と定義

- JID(JabberID)
 - リソース (ユーザ、サーバを含む) を一意に識別する ID。
 jid = [node "@"] domain ["/" resource] の形を取る。
- 名簿
 - 通信を行う相手のリスト。知人の一覧。

6.3.2 プラットフォーム抽象化レイヤ

プラットフォーム抽象化レイヤは、サンプル生成部やランキング作成部が iTunes からライブラリや聴いている曲の情報を取得する際の、抽象化を行う。これにより、サンプル生成部やランキング作成部は、iTunes がどのプラットフォーム版であっても、それに関わらず iTunes から情報を取得することができる。

6.3.3 サンプル生成部

サンプル生成部は、プラットフォーム抽象化レイヤを通してiTunes から取得した、ユーザが現在聴いている曲の情報を取得する³。そして、取得した情報を処理し、XMPP の拡張プロトコルの 1 つである JEP-0118 User Tune[19] で定められている形にそってサンプルを生成する。そしてそのサンプルを、サンプル送信部へと渡す。サンプルには、次の情報が含まれる。

● 曲を聴いたユーザの JID

³デフォルトでは1分毎に取得する

- 曲名
- アーティスト名
- アルバム名
- トラック番号
- 曲の長さ
- hop count⁴

6.3.4 サンプル送信部

サンプル送信部は、サンプル生成部から渡されたサンプルを、名簿のオンラインのユーザに対して送信する。

6.3.5 サンプル受信部

Tunes プラグインは、他ユーザから受け取った曲の情報を、送信元のユーザ毎に蓄積していく。 1 曲毎に蓄積される内容は、下記の通りである。

- サンプルを生成したユーザの JID
- 曲名
- アーティスト名
- アルバム名
- 受信日時
- ・ ポイントアップに影響があったかのフラグ

「ポイントに影響にあったかのフラグ」は、自ユーザが他ユーザから届いた曲を後に自 ライブラリに追加した際、次に同じ曲が他ユーザから届いた際にポイントアップを行うた めに用いられる。

また、サンプルが届いた際、hop count が1 でない場合、値を-1 してサンプル送信部に届いたサンプルを渡す。

⁴本実装では、hop count の初期値は 2 とする。

6.3.6 ユーザランキング生成部

以下に、他ユーザより曲の情報を受信した際の処理を、条件別に述べる。

- 自ライブラリの中にサンプルの曲が存在する場合
 - サンプルを生成したユーザのサンプルの蓄積の中に、サンプルの曲が存在する 場合
 - * ポイントアップに影響があったかのフラグが true の場合
 - . 受信日時を更新
 - * ポイントアップに影響があったかのフラグが false の場合
 - ・サンプルを生成したユーザのポイントに 1 を加え、フラグを true にして、受信日時を更新
 - サンプルを生成したユーザのサンプルの蓄積の中に、サンプルの曲が存在しない場合
 - * サンプルを生成したユーザのポイントに 1 を加え、フラグを true にして、 新たに蓄積
- 自ライブラリの中にサンプルの曲が存在しない場合
 - サンプルを生成したユーザのサンプルの蓄積の中に、サンプルの曲が存在する 場合
 - * 受信日時を更新
 - サンプルを生成したユーザのサンプルの蓄積の中に、サンプルの曲が存在しない場合
 - * フラグを false にして、新たに蓄積

6.3.7 お薦め生成提示部

お薦め生成提示部は、他ユーザのポイントと蓄積された曲の情報を用いて、ユーザに対してお薦めの曲リストを作成し、提示する。1リストに表示する曲数は、最大20曲に設定されている。リストの作成方法は、ポイント上位の人より、まず下記の式によって、リストに表示させるそのユーザの曲数を求める。

曲数 = その人のポイント / ポイント総和 * 1 リスト曲数 (小数点は切り捨て、ただし0 になる場合は1 にする)

もとめた曲数分、そのユーザの蓄積からランダム抽出する。この計算を、ユーザによって 設定された1リストの表示曲数に達するまで繰り返す。この結果、生成されたお薦めの曲 リストをユーザに対して提示する。図 6.3は、その1 例である。

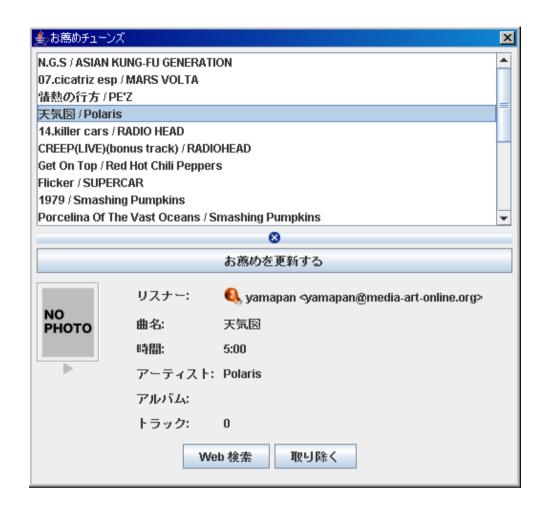


図 6.3: お薦めリスト

お薦めのリストは、起動時及びユーザがお薦めの表示機能を呼び出したタイミングにて、リストを作成し提示する。また、リストを作成する際に、ユーザランキング生成部にユーザのランキングの生成を依頼する。

6.3.8 曲情報取得支援部

曲情報取得支援部は、ユーザがお薦めとして提示された曲に関する情報を取得することを支援する。実際にお薦めした曲データ自身をユーザに提供することも検討したが、法的問題を考慮し、提供しないことにした。代替手段として、Web でその曲に関する情報を容易に検索できる支援機能を提供している。

6.3.9 サンプル表示部

サンプル表示部は、他ユーザから送られてきたサンプルを都度表示する機能をもつ。図 6.4は、サンプル表示部が表示するウィンドウの 1 例である。



図 6.4: サンプルの表示

曲情報取得支援部によって提供される Web 検索支援機能は、本ウィンドウに対しても 提供される。

6.3.10 環境設定部

環境設定部は、Tunes プラグインに関する設定を行う。図 6.5は、Tunes プラグインの 設定画面である。

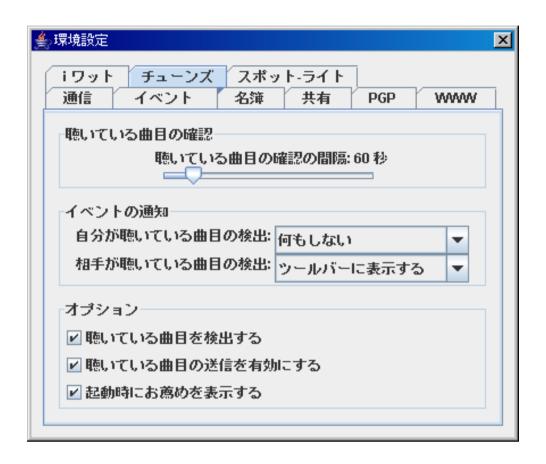


図 6.5: Tunes プラグイン設定画面

設定内容の詳細は、次の通りである。

- iTunes から現在聴いている曲の情報を確認・取得する間隔(デフォルト 60 秒)
- サンプルを生成し送信した際、及び、サンプルを受信した際のユーザへの通知方法 の選択
- 曲目を検出するか否かの選択
- サンプルを送信するか否かの選択
- 起動時にお薦めリストを表示するか否かの選択

6.3.11 運用実験のための機能

本実装の Tunes プラグインでは、運用実験のための機能として、お薦めリストから曲を取り除く機能を持つ。取り除く際には、

● ライブラリに加えた / 加える予定

- すでに持っている/知っている
- 興味がない/嫌い

の3つから、理由を選択してもらう。

第7章 実装を用いた評価

本章では、実装を用いて行った評価実験に関して述べる。この実験は以下の2点を目的として、仮想ユーザによるシミュレーション実験、及び、実際のユーザによる運用実験を行った。

- 実装が本モデルの要件を満たしているかの検証
- 実際の利用による改良点、今後の課題の収集

7.1 仮想ユーザによる実験

7.1.1 実験の目的

仮想ユーザによる実験では、実装が本モデルの要件を満たしているかを検証するため、モデルの特徴ごとに、特徴がより明らかになる形で仮想環境を作り上げ実験を行った。仮想環境では、それぞれ検証のためのiTunesのライブラリを持った仮想のユーザを複数用意し実験を行った。

この実験では、3.1 で述べたモデルの要件を、本実装が満たしているかを検証項目とした。

- ユーザが満足する作品の伝達
- 嗜好が近い人の発見とお薦めの導出
- お薦めの絞り込み
- 網羅性の向上
- 探索空間の拡張

7.1.2 ユーザが満足する作品の伝達

本実装では、iTunes で音楽を聴いているだけで、サンプルのやりとりが行われる。よって、本実装が本要件を満たしていることは自明である。

7.1.3 嗜好が近い人の発見とお薦めの導出及びお薦めの絞り込み

本要件は、本実装によるお薦めリストにおいて、他ユーザの嗜好の近さによって、1つのリストの枠が分配されていれば、達成されているといえる。そこで、表 7.1の内容のライブラリを持った 5 人の仮想ユーザ p1,p2,p3,p4,p5 を用意した。

投 7.1. 5人の収息ユータ						
ユーザ	ライブラリの総曲数	p1 のライブラリと重なっている曲数				
p1	10 曲	-				
p2	10 曲	4曲				
р3	10 曲	3 曲				
p4	10 曲	2曲				
p5	10 曲	1曲				

表 7 1:5 人の仮想ユーザ

また、ユーザの環境は以下の通りである。

- p1 は、p2,p3,p4,p5 が wija の名簿に追加されている
- p1 以外のユーザのおいて、p1 と重なっている曲以外では曲の重なりはない。
- 1つのお薦めリストの曲数は、20曲である

この状況下で、各ユーザにライブラリの曲を iTunes で一巡再生し、全ての曲のサンプルが p1 に届いた状態を作った。

この時、本モデルに基づけば、ユーザ p2 のお薦めリスト内の件数は、

$$NoRp2 = \frac{proximity(p1,p2)}{\sum_{x=2}^{5} proximity(p,px)} NoR$$

となる。proximity(p1,px) は、px のライブラリと p1 のライブラリの重なった曲数であるため、

$$NoRp2 = \frac{4}{4+3+2+1}20 = 8$$

となる。以上の計算を他ユーザにおいても行うと、表 7.2の様になった。

ユーザ	モデルに基づいた計算によるお薦めリスト内の曲数				
p2	8曲				
р3	6曲				
p4	4 曲				
p5	2曲				

表 7.2: モデルに基づいたお薦めリストの内訳

そこで、実際に実装において、p1のお薦めリストを表示したところ、20件の内訳は表7.3の通りとなった。

	表 7.3: 実装におけるお廌めリストの内訳				
	ユーザ	実装におけるお薦めリスト内の曲数			
ſ	p2	8曲			
ſ	p3	6曲			
	p4	4 曲			
	p5	2曲			

表 7.3: 実装におけるお薦めリストの内訳

よって、本実装においても本モデルと同じ結果が得られたため、本実装は、本要件を確かに備えていることが検証された。

7.1.4 網羅性の向上

本要件は、4章で用いた既存のリコメンドサービスの手法によるお薦めでは登場しないような曲がユーザのライブラリに存在する環境下において、本実装のお薦めリストにその曲が登場すれば、達成しているといえる。そこで、ライブラリに登録する曲として、 $a\sim u,x,y$ の 23 曲を用意し、表 7.4の様にそれらをライブラリに持つ 3 人の仮想ユーザ p1,p2,p3 を用意した。

表 7.4: 3	表 7.4: 3人の仮想ユーザとライブラリの曲					
ユーザ	ライブラリに登録されている曲					
p1	a					
p2	a ~ u,x					
р3	a ~ u,y					

また、ユーザの環境は以下の通りである。

- p1 は、p2,p3 が wija の名簿に追加されている。
- 1つのお薦めリストの曲数は、20曲である

この環境下において 4 章で用いた商品同士の結びつきによる 20 曲のお薦めリストを生成すると、 $a \sim u$ に比べ、一人のユーザのライブラリにしかないx とy は、優先度が低くなる。 $b \sim u$ で 20 曲あるため、x とy はお薦めリストに登場することができない。この時、本実装のお薦めリストにx 及びy がお薦めとして登場すれば、網羅性の向上の特徴を備え

ているといえる。なお、p2,p3 ともに p1 のライブラリと重なる曲は 1 曲であるため、お薦めリストに表示する 20 曲は、10 曲ずつの割り当てとなる。

そこで、本実装によるお薦めリストを表示したところ、表7.5の様になった。

表 7.5: お薦めリストの内訳 網羅性の向上

お薦め元になったユーザ	お薦めリストに登場した曲
p2	j,e,q,b,x,m,o,p,h,k
р3	t,y,f,d,i,l,r,c,u,h

表 7.5より、x と y がお薦めリストに登場したことがわかる。なお、お薦めリストを何度か更新すると x と y が登場しない場合もあったが、そのことは蓄積されたサンプルからランダムに抽出する本モデルの仕様からおきる現象である。したがって、このことは、本要件を満たしていないことにはならない。

7.1.5 探索空間の拡張

本要件は、自分の wija の名簿に登録されていないユーザからのお薦めが得られれば達成されているといえる。そこで、表 7.6の内容のライブラリを持った 3 人の仮想ユーザ p1,p2,p3 を用意した。

表 7.6: 3人の仮想ユーザ

ユーザ	ライブラリの総曲数	p1 のライブラリと重なっている曲数
p1	20 曲	-
p2	20 曲	1曲
р3	20 曲	4曲

また、ユーザの環境は以下の通りに設定した。

- p1 は、p2 が wija の名簿に追加されている。
- p2 は、p1,p3 が、p3 は p2 が wija の名簿に追加されている。
- p2 と p3 では、p1 と重なっている曲以外の曲の重なりはない。
- 1つのお薦めリストの曲数は、20曲である。

この状況下で、各ユーザにライブラリの曲を iTunes で一巡再生し、全ての曲のサンプルが p1 に届いた状態をつくった。

この時、本要件を備えてnれば、p3からのサンプルがp1のお薦めリストに登場するはずである。

そこで、p1のお薦めリストを表示したところ、20件の内訳は表7.7の通りとなった。

表 7.7: お薦めリストの内訳 探索空間の拡大

ユーザ	お薦めリスト内の曲数
p2	4 曲
р3	16 曲

確かに、p3 からのお薦めが届いているため、本実装は、本要件を満たしていることが 実証された。

7.2 運用実験

7.2.1 実験の目的

運用実験では、普段 IM 及び iTunes を利用している人を対象に、実際に本実装を日常的に利用してもらうことにより行った。目的は、今後の実装の改良点や今後の課題の検討に用いるフィードバック情報を得ることである。

7.2.2 実験概要

本実装の Tunes プラグインを備えた wija 及び iTunes を、2004/12/21 より 14 日間、普段 IM と iTunes を利用している人 10 人を対象に利用してもらった。期間中、インターネットに接続した状態で iTunes を使用している間、本実装も常に起動し利用してもらった。また、お薦めリストで気になった曲は、可能な限りの手段をつくして、直接その曲自体を聴いてもらうようにお願いした。さらに、6.3.11で述べた統計機能も併せて利用してもらった。

14日間の利用後、ユーザに統計機能によって得られた情報を提出してもらうと同時に、 利用する中で気づいた点、改良して欲しい点などを自由にコメントしてもらった。

7.2.3 アンケート

全ユーザの統計情報の集計結果を、表7.8に表す。

表 7.8: 統計情報の集計結果

ユーザー人あたりの蓄積サンプル数の平均	1554
ユーザー人あたりのポイント数の平均	138
「ライブラリに加えた/加える予定」の総和	112
「すでに持っている/知っている」の総和	0
「興味がない/嫌い」の総和	39

次に、ユーザからのコメントでは、表 7.9に示したコメントが得られた。コメントは、 原文のまま載せている。

表 7.9: ユーザからのコメント

お薦めから曲を取り除く際、1曲単位、アルバム単位を選べるようにして欲しい。(同意見3件)

自分に趣味が近い人順に友達を並び替えれるようにして欲しい。

アルバム単位のお薦めリストが欲しい。(同意見2件)

聴いてる曲を送るのはいいが、恥ずかしいので自分が聴いた曲であることを相手に教 えなくする機能が欲しい。(同意見3件)

簡単に試聴できると嬉しい。(同意見7件)

CD として持っているけど PC に取り込んでいなかった曲が薦められて、ちゃんと趣味が近いという人と同じ CD 持ってたりするんだなと思った。

とても良いのに今まで全然知らなかったバンドを見つけた!(同意見2件)

普段から趣味が似ているとは思ったが、実際似ていることが分かって面白かった。(同意見2件)

自分に趣味が近い人が最近よく聴いている曲を見てみたい。

同じアーティストの曲は、1回のお薦めリストの中で1つにして欲しい。(同意見2件)

コメントから明らかになったことを以下にまとめる。

- 未知の満足する作品の発見が実現した。
- 容易な曲の試聴を望む声が多い。
- 実験以前より嗜好が近いと認識していた人が、本実装上でも近いと判別された。
- 嗜好が近い人を知りたいという意見が複数あった。
- 取り除く機能によってお薦めをさらに絞り込もうとする行動が見られた。

● 自分が聴いている曲が他の人に知られるのを恥ずかしいと思う意見が複数あった。

また、運用実験による統計情報の結果及びユーザのコメントから、今後の課題として以下の点が挙げられる。

- ユーザのプライバシー
- 試聴環境の整備

これらの検討課題については、9章において詳しく述べる。

第8章 関連研究

8.1 知人関係を利用した情報共有・情報検索

本研究は、知人関係を利用するサービスを用いることで、目的を達成する。一方で、知人関係を用いたコミュニケーションや情報共有の研究は、数年前から数多く行われている。また、最近では、「音楽を推薦しあう」といった特定の目的をもった SNS や、自分の持っている本やレコードを登録していくことで、自分と似たような作品を持っている人がわかるというサービスが登場してきた。本章では、本研究のアプローチに関連する研究やサービスを取り上げ、その詳細について述べる。

8.2 知人関係を利用した情報検索

8.2.1 Referral Web

1997年に提案された Referral Web[20] は、キーワードによる論文検索のシステムであり、検索の際に論文に関係する人々の知人関係を利用するのが特徴である。その際利用する知人関係は、論文の著名者に加え、ニュースグループへの投稿などを分析することで、抽出している。実際に論文を検索する Web アプレットも公開されている。

8.2.2 Cite Seer

Cite Seer[21] は、主にコンピュータサイエンスの論文を集積しているデジタルライブラリである。検索に関する研究の一環として公開されており、自由に利用することができる。論文内での他論文の参照や、共著関係などの知人関係を利用した検索が特徴である。

8.2.3 本研究との関連

以上の2つのシステムは、共に知人関係を用いて、情報を検索するシステムである。両 システムとも、検索のきっかけとなるキーワードは、ユーザが入力する必要がある。キー ワード検索によって1つの検索ターゲットを特定したのち、そのターゲットを元に知人関 係を利用することで他の検索ターゲットを参照する。

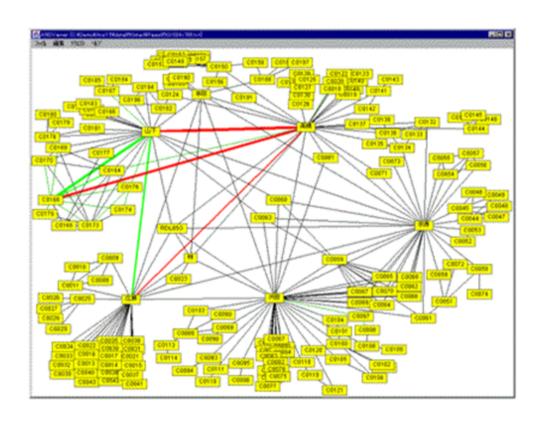
1つのターゲットから他のターゲットを示すという点は、探索的お薦め導出手法における「嗜好が近い人を発見し、その人のサンプルからお薦めを導出する」という点に似てい

る。しかし、本研究では、他ユーザからのサンプルを元に自分と嗜好が近い人を発見する ため、キーワードを必要としない。

8.3 知人関係を利用した人材マッチング

8.3.1 ANS

1998年に提案された ANS(Acquaintance Network System)[22] は、個人が持つ情報内容に応じて人と人を引き合わせることで、組織やコミュニティにおける情報共有を実現することを狙いとしたシステムである。ANS では知人の関係を、意味と重みを持ったパイプと呼ぶデータ構造で表現し、複数のパイプの合成処理により新しい関係に対応するパイプを生成する。図 8.1は、パイプによって結ばれた知人関係のネットワーク図である。これにより、情報の開示範囲を知人の範囲に限定しているにもかかわらず、広範囲の情報提供者と情報消費者の引き合わせを実現している。



■ 8.1: Acquaintance Network System

8.3.2 HUMANNET

HUMANNET[23] は、2004年に開かれた第 18 回人工知能学会全国大会にて利用された人間関係ネットワーク支援システムである。このシステムは、Web 上に存在する、論文リストや論文 DB、研究室のページや研究会のページなどの様々な情報を統合的に処理し、研究者の協働関係に関するネットワークを抽出している。図 8.2は、抽出されたネットワーク図である。このネットワークを用いて、知り合いの知り合いや、自分と他の人をつなぐ関係のパスを参加者に提供すると同時に、人工知能学会のグループの構造や中心的な研究者を俯瞰することも実現した。

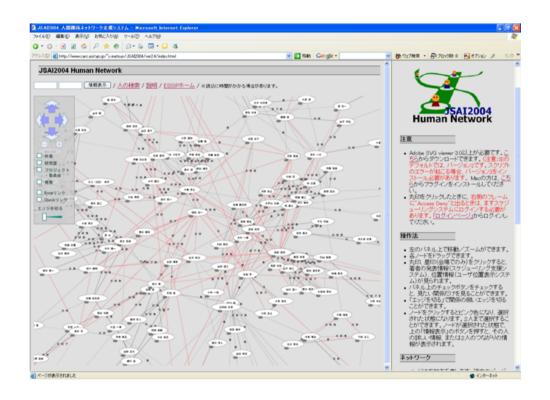


図 8.2: HUMANNET

8.3.3 本研究との関連

以上の2つの研究は、各自が持つ属性情報を元に、ユーザ同士の新たな繋がりを実現するモデルである。「各自の属性情報を元に、ユーザ同士を結びつける」点は、本研究において、「各ユーザから届くサンプルを元に、自分と嗜好が近いユーザを発見する」点に似ている。

しかし、これらの研究は、モデル自身の複雑さや、モデルを実現するために必要な事前 準備が大きい。このため、ユーザが気軽に利用することが難しい。一方、本モデルにおい ては、ユーザは自分の嗜好を満足させる行動を取ることだけで、お薦めを得ることがで きる。

8.4 嗜好の近さを測るサービス

8.4.1 booklog

2004 年に公開された booklog[24] は、自分の書棚を Web 上に作成できるサービスである。Amazon と連携することで、Amazon で扱われている商品を自分の書棚に登録でき、ユーザは互いの書棚を閲覧することができる(図 8.3)。また、自分の書棚と似ているユーザの書棚も提示してくれる。

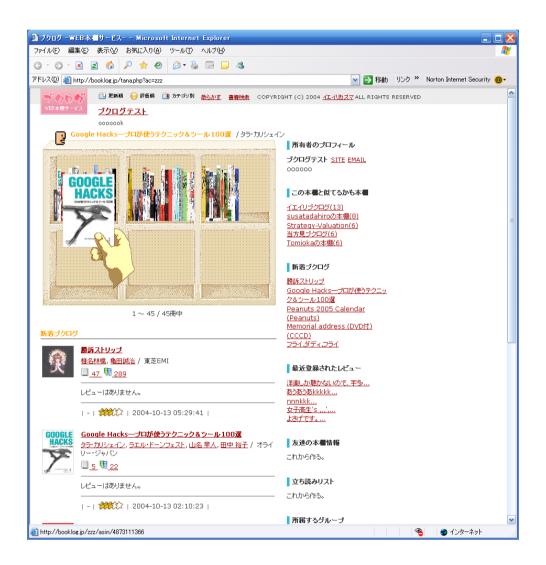


図 8.3: booklog

8.4.2 recoya

2004 年に公開された recoya[25] は、Web 上に自分の CD 棚を作成できるサービスである。ユーザは、サイトにあらかじめ登録されている CD や DVD の情報から、自分が所有

している物を検索し自分の棚に登録できる。ユーザは互いの棚を閲覧することができる。 さらに、自分の棚に入っている CD を売ったり、誰かが売っている CD を買ったりすることが可能である。

また、自分の棚と他のユーザの棚の一致度のランキングを見ることができ、ランキングは1日毎に更新される(図8.4)。



図 8.4: 棚の一致度ランキング

8.4.3 本研究との関連

以上の2つのサービスは、ユーザが登録した情報を元に、他のユーザとの嗜好の近さを求め、ユーザに嗜好が近い人を提示するサービスである。「自ユーザの嗜好と他ユーザの嗜好から近さを求める」点において、本研究に似ている。

しかし、これらのサービスは、ユーザが情報を登録する手間が大きい。また、単に嗜好が近い人が所持しているものを閲覧できるだけである。

一方、本研究では、8.2.3 で述べたように、何かを登録する必要はない。また、本研究は、各ユーザとの嗜好の近さを元に、お薦めの優先順位付けを行い、ユーザに提示する点でも異なる。

8.5 知人関係を利用した作品の情報の伝達

8.5.1 Recommuni

2004年10月に公開された Recommuni[5] は、友人同士で音楽の推薦をしあう SNS である。ユーザは自分が他人にお薦めしたい曲データをアップロードする。アップロードされた曲は、友人や友人の友人へとお薦めとして伝わる。曲データのアップロードは、権利者やレコード会社の許諾が得られている曲のみ可能となっている。お薦めを受け取ったユーザは、対価を支払うことでその曲データをダウンロードすることが可能である。お薦めを行うインセンティブとしてポイント制が導入されている。ユーザは、アップロードやお薦めを行うことでポイントが付与される。このポイントは曲データのダウンロードの際の料金の支払いに使うことができる。

8.5.2 本研究との関連

Recommuni は、知人関係を用いて作品のお薦め情報を伝達する点が、本研究と似ている。さらに、その情報が知人の知人へと伝わっていく点も同じである。

しかし、Recommuniでは、ユーザがお薦めとして作品を意識的に取り上げなければ他ユーザへの情報の伝達が行われない。一方、本実装では、ユーザは、他のユーザの誰かに意識的に薦めるという行動を取る必要はない。ユーザはただ普段の行動(音楽再生ソフトウェアで音楽を聴く)を行っているだけでよい。お薦めを受け取るユーザは、他のユーザのサンプルの蓄積を元に自らお薦めを導き出す。

第9章 結論

9.1 本研究が実現したこと

本研究は、ユーザが満足する作品を発見する可能性を高めることを目的とした。既存手段の問題点として、

- 発見的でない
- 探索空間が狭い
- 網羅性に乏しい

を挙げ、その解決モデルとして、「探索的お薦め導出手法」を提案した。

そして、本モデルのターゲットを音楽に絞り、既存のリコメンドサービスの手法と探索的お薦め導出手法の比較検証実験を行った。この比較実験では、共に大きなシェアを持つ音楽再生ソフトウェアであるiTunes とインスタントメッセージングのクライアントソフトウェアであるMSN Messenger の利用者を被験者とし、iTunes で聴いている曲のライブラリと MSN Messenger で登録している友達の中でiTunes を利用している人を提出してもらった。集めたデータを分析すると同時に、探索的お薦め導出手法を用いたお薦めリストと既存のリコメンドサービスの手法を用いたお薦めリストを作成し、ユーザにその結果をフィードバックした。ユーザには、フィードバックされたリストの曲1つ1つを実際に調べてもらい、満足する作品を発見できたかを、アンケートによって回答してもらった。実験の結果から、本モデルが、

- 既存手段の問題点を解決した
- その結果、ユーザが満足する作品を発見する可能性を高めた

ということが、実証できた。

さらに、音楽再生ソフトウェアの iTunes とインスタントメッセージングのクライアントソフトウェアの wija を用い、本モデルを wija プラグイン「Tunes プラグイン」として実装した。

そして、実装を用いた実験の結果から、本実装が本モデルの要件を満たしていることが確認された。また、満足する作品を発見する可能性を高めるアプリケーションを、ユーザに提供することができた。

9.2 今後の課題

検証実験及び運用実験を通して、以下の様な今後の課題点が明らかになった。

- 嗜好の近さを求める際のサンプル数の限定
- サンプルのプライバシー
- 試聴環境の整備

以下、各詳細について述べる。

9.2.1 嗜好の近さを求める際のサンプル数の限定

3.4.1 において、本モデルでは、他ユーザとの嗜好の近さを求める際、ポイントモデルを用いると述べた。ポイントモデルでは、既に多くのサンプルを発信したユーザと、少ないサンプルのみしか発信していないユーザとの嗜好の近さを測る場合、本来の集合論を元にした嗜好の近さと比較して、近さの精度が低下してしまう(度合いが離れてしまう)問題点を挙げた。比較実験における被験者のアンケートからも、この問題点を指摘する感想が得られた。

本研究では、この問題の解決策として、嗜好の近さを求める際のサンプルの数をある一定の狭い時間軸内の値にすることを提案した。これは、一定期間内に各ユーザが鑑賞できる作品の数には、限界があるためである。そして、嗜好の近さを測る際の最適な値を定めることを、今後の検討課題として挙げた。本論文中では、この検証を行うことができなかったが、今後、実装の長期運用の中で、嗜好の近さを測る際のサンプル数を変化させ、ユーザの満足度の調査を行うことで、最適な値を求めていく必要がある。

9.2.2 サンプルのプライバシー

運用実験後の意見調査の中で、「自分が何を聴いているのかを周りに知られてしまうのは恥ずかしい」という声がいくつか見られた。その理由として、「自分の嗜好と関係のない人に個人的嗜好を知られたくない」という理由が多かった。一方、「誰からのお薦めかがわかった方が良いか」の問には、多くのユーザが yes と答えた。

つまり、嗜好が似ている人に自分が何を聴いているのか知られることには抵抗が少なく、似ている人のお薦めは誰の行動かを知りたいというニーズがあることがわかった。ユーザが自分の行動を周りに知らせることに躊躇を感じないためにも、このニーズを実現する仕組みについて検討する必要がある。

個々の情報発信の匿名性を実現する研究として、Freenet[26] 等が挙げられる。これらの技術を適用することで、ユーザのプライバシーを考慮したサンプルの送信を実現する必要がある。

9.2.3 試聴環境の整備

同じく運用実験後の意見調査の中で、簡単に試聴できることを要望する声が多く見られた。音楽の様な嗜好物は、実際に試聴しないことには、作品を評価することが難しい。

本実装においても、直接の知人へは、試聴を実現する機能を検討した。検討の中で、wija の開発メーリングリストでのディスカッションも行った。ディスカッションでは、直接の知人という特定された人に音楽のデータを送信することが、著作権法の公衆送信に該当するかが争点となった。過去の事例からは、どの人数から特定多数と特定少数に分けられるかは事例毎に異なり、本研究が独自に判断を下すことはできないという結論に至った。そのため、本実装においては、試聴の機能は搭載せず、検索サイトによる検索を容易にする機能のみにとどめた。

しかし、実際の運用の結果からは、やはり試聴を望む声が多かった。今後の課題として、著作者の利益を損害しない形を実現することで、著作権にふれない試聴環境を整備する必要がある。

謝辞

本研究を進めるにあたり、ご指導頂いた主査である慶應義塾大学環境情報学部教授村井 純博士、副査である慶應義塾大学環境情報学部助教授中村修博士と慶應義塾大学看護医療 学部専任講師の宮川祥子博士に感謝致します。

また、最後の最後まで付きっきりで御指導を頂きました、慶應義塾大学政策・メディア研究科助手の斉藤賢爾氏と慶應義塾大学政策・メディア研究科博士課程の須子 義彦氏に深く感謝致します。また、何度も心配してお声をかけてくださった慶應義塾大学環境情報学部専任講師の重近範行博士、慶應義塾政策・メディア研究科博士課程の石田剛朗氏、海崎良氏に感謝致します。

そして、サポートしてくださったモービル広域ネットワーク (MAUI) プロジェクト及び 徳田・村井・中村・楠本・南研究室の皆様に感謝致します。中でも、土壇場まで作業を手 伝ってくれた後輩達には、とても助けられました。

また、長く辛く苦しい残留生活に笑いを提供してくれた入野 仁志氏、小椋康平氏、 だめな社員を暖かく見守ってくれている久松慎一氏、

「もう無理だー」と夜な夜な言い合った小柴晋氏、朝方おかしなテンションで漫才を繰り 広げた岡田耕司氏、

同じ修論生なのに、細やかサポートを決めてくれた渡里雅史氏、

卒論に続いて英語のサポートをしてくださった源中由紀氏、

修論の間の仕事を押しつけられてくれた山崎淳氏、

そしてなによりも、本研究の動機になった、素晴らしいアーティスト達に、 感謝致します。

最後に、大変な時期に支え続けてくれた家族に、感謝致します。

以上を持って謝辞と致します。

参考文献

- [1] Orkut Buyukkokten. Orkut. http://www.orkut.com/, December 2004.
- [2] 株式会社イー・マーキュリー. コミュニティ・エンターテイメント mixi. http://mixi.jp/, December 2004.
- [3] グリー株式会社. グリー (gree). http://gree.jp/, December 2004.
- [4] Amazon.com, Inc. Amazon. http://www.amazon.com/, December 2004.
- [5] 株式会社レコミュニ. 参加型音楽配信コミュニティ recommuni. http://recommuni.jp/, December 2004.
- [6] 須子 善彦. 知人関係を用いたプライバシ保護型マッチングシステムの研究. Master's thesis, 慶應義塾大学大学院, 政策・メディア研究科, January 2004.
- [7] Google, Inc. Google. http://www.google.com/, January 2005.
- [8] CNET Japan. グーグル、インテル、ms が注目するベイズ理論. http://japan.cnet.com/special/story/0,2000050158,20052855,00.htm, December 2004.
- [9] INTERNET Watch. ニフティ、ベイズ理論を応用した迷惑メール対策フィルターを提供開始. http://internet.watch.impress.co.jp/cda/news/2004/01/14/1743.html, December 2004.
- [10] The Regents of the University of Minnesota. History of statistics & probability, thomas bayes. http://www.mrs.umn.edu/sungurea/introstat/history/w98/Bayes.html, December 2004.
- [11] Apple Computer, Inc. itunes. http://www.apple.com/jp/ilife/itunes/index.html, December 2004.
- [12] ネットレイティングス株式会社. インスタントメッセンジャーの利用実態調査. http://www.netratings.co.jp/press_releases/0909_Release_J_final.pdf, December 2004.
- [13] Microsoft Corporation. Msn messenger. http://messenger.msn.com/, December 2004.

- [14] Albert-László Barabási. *Linked: The New Science of Networks*. Perseus Publishing, 2002.
- [15] グレースノート株式会社. CDDB. http://www.gracenote.com/gn_japan/cddb.html, December 2004.
- [16] Media Art Online. wija. http://www.media-art-online.org/wija/, December 2004.
- [17] P. Saint-Andre, Ed. Extensible Messaging and Presence Protocol (XMPP): Core, October 2004. RFC 3920.
- [18] P. Saint-Andre, Ed. Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence, October 2004. RFC 3921.
- [19] The Jabber Software Foundation (JSF). Jep-0118: User tune. http://www.jabber.org/jeps/jep-0118.html, December 2004.
- [20] H. Kautz, B. Selman and M. Shah. Referralweb: Combining social networks and collaborative filtering. *Communications of the ACM*, 40(3):63–65, 1997.
- [21] Kurt D. Bollacker, Steve Lawrence, C. Lee Giles. Citeseer: An autonomous web agent for automatic retrieval and identification of interesting publications. 2nd International ACM Conference on Autonomous Agents, pages 116–123, May 1998.
- [22] 高橋範泰, 山下剛史. 知人のネットワークの概念に基づいた情報共有機構. Technical Report 12, 富士ゼロックス, 1998.
- [23] 松尾 豊, 友部 博教, 橋田 浩一, 石塚 満. イベント空間支援における人間関係ネット ワーク抽出技術の活用. 人工知能学会全国大会, January 2004.
- [24] イエイリカズマ. ブクログ web 本棚サービス. http://booklog.jp/, December 2004.
- [25] 株式会社タフ. recoya. http://www.recoya.jp/, December 2004.
- [26] I. Clarke. A distributed decentralised information storage and retrieval system. Technical report, Division of Informatics, University of Edinburgh, 1999.