

楽曲推薦システムの効率性とスケーラビリティの改善のための 確率的推薦モデルのインクリメンタル学習法

吉井 和佳[†] 後藤 真孝[‡] 駒谷 和範[†] 尾形 哲也[†] 奥乃 博[‡]

[†] 京都大学大学院 情報学研究科 知能情報学専攻 [‡] 産業技術総合研究所
yoshii@kuis.kyoto-u.ac.jp m.goto@aist.go.jp
{komatani, ogata, okuno}@i.kyoto-u.ac.jp

本稿では、ハイブリッド型楽曲推薦システムの効率性とスケーラビリティを改善する手法について述べる。我々は以前、確率的推薦モデルを用いて「ユーザの楽曲評価」と「楽曲の音響的特徴」とを同時に考慮するハイブリッド型推薦システムを開発し、高精度な推薦ができることを実証したが、効率性とスケーラビリティに関する問題が残されていた。すなわち、新規評価や新規ユーザ、新規楽曲が追加されるたびにモデル全体を再学習する必要があった。また、モデル学習にかかる計算コストがユーザ数と楽曲数に比例して増加するため、現実的なサイズのデータを扱うことができなかった。そこで本稿では、効率性を改善するため、モデルをわずかなコストで部分的に更新できるインクリメンタル学習法を提案する。さらに、スケーラビリティを改善するため、まず少数の代表的な仮想ユーザと仮想楽曲に対して「コアモデル」を構築し、次にインクリメンタル学習法によって実際のユーザと楽曲をコアモデルに追加登録していく手法を提案する。実験の結果、上記二つの改良を施した推薦システムでは、推薦精度が向上することが分かった。

Incremental Training of Probabilistic Recommendation Model for Improving Efficiency and Scalability of Music Recommender System

KAZUYOSHI YOSHII[†], MASATAKA GOTO[‡], KAZUNORI KOMATANI[†], TETSUYA OGATA[†]
and HIROSHI G. OKUNO[‡]

[†] Dept. of Intelligence Science and Technology, Graduate School of Informatics, Kyoto University
[‡] National Institute of Advanced Industrial Science and Technology (AIST)

We aimed at improving the efficiency and scalability of a hybrid music recommender system. Although this system was proved to make accurate recommendations by using a probabilistic model that integrates rating scores provided by users acoustic features of musical pieces, it lacks efficiency and scalability. That is, the entire model needs to be re-trained from scratch whenever a new score, user, or piece is added. Furthermore, the system cannot deal with practical numbers of users and pieces. To improve efficiency, we propose an incremental method that partially updates the model at low computational cost. To enhance scalability, we propose a method that first constructs a small “core” model over fewer virtual representatives created from real users and pieces, and then adds the real users and pieces to the core model by using the incremental method. The experimental results revealed that the proposed system was not only efficient and scalable but also outperformed the original system in terms of accuracy.

1. はじめに

現在、インターネットを介した音楽配信サービスは数百万曲オーダの大規模データベースを運用している¹⁾。ユーザがこの中から好みに合う楽曲を探し出すことは容易ではないため、そのような楽曲を自動的に提示する楽曲推薦システムが果たす役割は大きい。例えば、多くの商用サービス (last.fm や Amazon.com²⁾ など) で

は、他のユーザがどのように楽曲を評価したのかを参考にして推薦を行う協調フィルタリングという手法を利用している。この手法は目覚ましい成果をあげてきたが、「新アイテム問題 (new-item problem)」を抱えている。すなわち、未評価の楽曲は推薦できないという欠点がある。また、多くのユーザは少数の人気アーティストにばかり高評価を与える傾向があるので、推薦されるアーティストのバラエティは乏しくなりやすい。

そのため、ロングテール効果は限定的であると言える。ロングテール効果とは、人気がない楽曲（テール）の需要は一曲ごとではわずかであるが、そのような楽曲が大量に存在することで、売り上げの和が少数のヒット曲の売り上げの和に匹敵する現象を指す³⁾。

近年、このような問題を解決するため、音楽的内容の面でユーザの好みに合いそうな楽曲を推薦する手法が多く研究者の注目を集めている。しかし、音響信号から内容情報を自動的に抽出するアプローチ^{4),5)}は、ユーザの支持を十分に獲得するには至っていない。一方、Pandoraという音楽配信サービスでは、専門家が人手で商業楽曲に内容情報を付与しており、好評を博している。このことから、適切な推薦を行うためには、音響信号から抽出することができない要素（文化的背景や市場での人気など）を考慮する必要があることが分かる。しかし、このような手動アノテーションに基づくアプローチは移植性を欠いている。例えば、一般ユーザが作曲し、ソーシャルネットワークサービス (Muzie や MySpace.com など) 上で公開した楽曲すべてに人手で内容情報を付与することは現実的ではない。

利用可能な情報の種類によらずに適切な推薦を行うためには、柔軟な情報統合の枠組みが必要である。このようなハイブリッド型のアプローチを利用することで、現実世界における不適切な情報（悪意ある評価、自動アノテーションの誤り、人手アノテーションの一貫性の欠如など）に対する頑健性が向上する。

このような考えのもと、我々は確率モデルを用いてユーザの楽曲評価スコアと楽曲の音響的特徴とを理論的に統合するハイブリッド型推薦システムを開発した。さらに、本システムは前述した従来の推薦手法よりも高精度な推薦ができることを実験で確かめた。しかし、数百万オーダーの大量のユーザや楽曲を扱う上で、効率性とスケーラビリティに関する深刻な問題があった。本システムは、あるユーザが評価を変えた場合、それに即座に適應することができないだけでなく、新規ユーザや新規楽曲を逐次的に追加登録していくこともできなかった。なぜなら、データが変化するたびに確率モデル全体を再学習する必要があり、その学習時間はユーザ数と楽曲数に比例するからである。

本稿では、効率性を改善するため、確率モデル中でデータの変化に関係するパラメータのみを更新するオンライン手法を提案する。これにより、データ変化を逐次的にモデルに反映させることができる。また、スケーラビリティを改善するため、大量のユーザと楽曲から一定数の代表的なユーザと楽曲を仮想的に生成し、それらに対してコンパクトな「コアモデル」を構築する手法を提案する。コアモデルは、もとのユーザと楽曲を逐次的に登録しながら更新される。

以下、まず2章でハイブリッド型楽曲推薦システムについて説明する。3章で提案する効率性向上手法とスケーラビリティ向上手法について述べる。4章で評価実験について報告し、5章でまとめとする。

2. ハイブリッド型楽曲推薦システム

本章ではまず、楽曲推薦タスクを定義する。次に、以前提案したハイブリッド型楽曲推薦システムにおける確率モデルの学習法について述べる。

2.1 タスク定義

楽曲推薦とは、あるユーザが与えられたとき、そのユーザが評価を与えていない楽曲に対してユーザが好みそうな順にランク付けを行うタスクを指す。いま、ユーザ群を $U \equiv \{u_1, \dots, u_{|U|}\}$ とし、楽曲群を $M \equiv \{m_1, \dots, m_{|M|}\}$ とする。ここで、 $|U|$ と $|M|$ は、ユーザ数と楽曲数を表し、 U および M はあらかじめデータベースに登録されているものとする。本研究の楽曲推薦システムは確率モデルに基づいており、 $u \in U$ と $m \in M$ を確率変数として扱う。その他の付加的な情報（タイトルやアーティスト名、ジャンルなど）などは楽曲推薦に利用しない。

ユーザの楽曲評価スコアはあらかじめデータベースに登録されているが、変化したり増加したりすることがある。本稿では、楽曲評価スコアは0（最低スコア：大嫌い）から4（最高スコア：大好き）の5段階とし、 $r_{u,m}$ ($0 \leq r_{u,m} \leq 4$) をユーザ u が楽曲 m に対して与えたスコアと定義する。ただし、ユーザ u が楽曲 m に評価を与えていない場合、 $r_{u,m}$ には「空スコア」として便宜上 ϕ を代入しておく。全ユーザの評価スコアを集めることで、評価行列 R が得られる。

$$R = \{r_{u,m} | u \in U, m \in M\} \quad (1)$$

ここで、ユーザは M 中のごく一部の楽曲にしか評価スコアを付与しないため、 R 中のほとんどの要素が ϕ であることに注意する。

一方、内容に基づく情報として、楽曲の音楽的な内容を一つのベクトルで表現する必要がある。このベクトルは各次元が意味的に等価であり、和は1に正規化されているものとする。例えば、楽曲全体の平均的な周波数帯域ごとのパワー比、あるいは小さな時間フレームから抽出した音響的特徴群を混合ガウス分布でフィッティングした時の重みなどが考えられる。本研究では、楽曲群 M に対応する音響信号が利用可能であると仮定している。いま、 $T \equiv \{t_1, \dots, t_{|T|}\}$ を特徴量とする。 $|T|$ は特徴量の種類数（特徴量ベクトルの次元）を表す。 $c_{m,t}$ を楽曲 m から抽出した特徴 t の重みとする。ここで、 $\sum_t c_{m,t} = 1$ である。全楽曲の特徴量ベクトルを集めることで、内容行列 C が得られる。

$$C = \{c_{m,t} | m \in M, t \in T\} \quad (2)$$

本稿では、我々が以前提案した Bag-Of-Timbres 法を用いて多重音色分布行列 C を生成した。

2.2 基本的な推薦方法

楽曲評価スコアと音楽的内容とを統合するため、我々は Popescu らによって提案された三方向アスペクトモデル⁷⁾ と呼ばれるベイジアンネットワークを利用する。このモデルには隠れ変数が導入されており、両者の情報が発生する過程を確率的に表現できる。

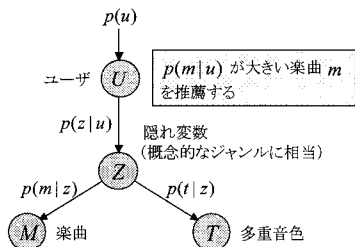


図1 音楽的コンテンツとして多重音色を利用した三方向アスペクトモデルの非対称表現

三方向アスペクトモデルを図1に示す。評価スコアや楽曲内容などの観測データは隠れ変数 $Z = \{z_1, \dots, z_n\}$ と関連付けられる。ここで、 $|Z|$ は隠れ変数の数である。隠れ変数はジャンルに対応していると考えると理解しやすいが、あくまで便宜的かつ概念的な考え方であり、一般的な意味でのジャンルではないことに注意する。このモデルでは、観測データには直接現れないユーザーの音楽的嗜好を U と Z の確率的関係として表現する。すなわち、あるユーザー u の嗜好は、各ジャンル z の出現確率 $p(z|u)$ によって定まる。このモデルでは、ユーザー u は自分の嗜好 $p(z|u)$ に従ってジャンル z を選択し、そのジャンル z が確率的に楽曲 m と多重音色 t を生起させる過程が自然に表現できる。

アスペクトモデルでは、ユーザー U 、楽曲 M 、多重音色 T が、隠れ変数 Z を介して条件付き独立であることを仮定している。また、1人のユーザーが複数のジャンルへの嗜好を持つことが許されている。これは、多くのクラスタリング手法が、各ユーザーをいずれかのジャンルクラスに割り当ててしまうのと異なる点である。

2.2.1 三方向アスペクトモデルの定式化

本節では、三方向アスペクトモデルの定式化を行う。いま、三つ組 (u, m, t) でユーザー u 、楽曲 m 、音色 t の共起を表す。これは、「ユーザー u が楽曲 m 中の多重音色 t を聴く」という観測可能な事象に対応する。さらに、 $n(u, m, t)$ を上記事象の生起回数とし、以下で示す通り二つの観測データの積に分解できると仮定する。

$$n(u, m, t) = r_{u,m} \times c_{m,t} \quad (3)$$

ここで、 $r_{u,m}$ と $c_{m,t}$ は以下の通り定義されている。

- $r_{u,m}$ は、ユーザー u の楽曲 m に対する評価スコアである。未評価の場合はデフォルトスコアとして2.5が代入されている。
- $c_{m,t}$ は、楽曲 m 中の多重音色 t の重みである。

事象 (u, m, t) は、ユーザー u が楽曲 m を好む度合いが大きいほど、あるいは楽曲 m 中の多重音色 t の重みが大きいほど生起しやすいと考えられる。

同様に、四つ組 (u, m, t, z) でユーザー u 、楽曲 m 、音色 t 、ジャンル z の共起を表す。これは、「ユーザー u が多重音色 z を選択し、 z が楽曲 m と多重音色 t を生起している」事象に対応するが、観測は不可能である。上記事象の生起確率 $p(u, m, t, z)$ は以下で与えられる。

$$p(u, m, t, z) = p(u)p(z|u)p(m|z)p(t|z) \quad (4)$$

ここで、 $p(u)$ はユーザー u の事前確率、 $p(z|u)$ はユーザー u が与えられた場合にジャンル z が生起する確率、 $p(m|z)$ および $p(t|z)$ はジャンル z が与えられた場合に楽曲 m および音色 t が生起する確率である。式(4)を z で周辺化することで、事象 (u, m, t) の生起確率 $p(u, m, t)$ は以下の通り求まる。

$$p(u, m, t) = \sum_z p(u)p(z|u)p(m|z)p(t|z) \quad (5)$$

本モデルにおける未知パラメータは、 $\{p(z|u)|z \in Z, u \in U\}$ 、 $\{p(m|z)|m \in M, z \in Z\}$ および $\{p(t|z)|t \in T, z \in Z\}$ である。これらは、観測データである評価行列 R と内容行列 C を用いて推定する。推薦を行う際には、各ユーザー u に対して、 $p(m|u) \propto \sum_z p(u, m, t)$ に従ってユーザー u の未評価楽曲に対してランク付けを行う。

2.2.2 基本的なモデル学習法

本節では、未知パラメータの推定方法について説明する。まず、事象 (u, m, t) が独立に生起すると仮定すると、学習用データ（評価行列 R と内容行列 C ）に対するパラメータの尤度 L は以下で与えられる。

$$L = \prod_{u,m,t} p(u, m, t)^{n(u,m,t)} \quad (6)$$

$$= \prod_{u,m,t} \left(\sum_z p(u)p(z|u)p(m|z)p(t|z) \right)^{n(u,m,t)} \quad (7)$$

求めるべきパラメータは尤度 L を最大化するものである。このように隠れ変数を含む尤度を最大化する問題を解くため、本研究ではDAEMアルゴリズム⁸⁾を利用する。DAEMアルゴリズムは通常のEMアルゴリズムに比べて局所解に陥りにくいという利点を持つ。

3. 効率性とスケーラビリティの改善

推薦システムに用いられるアルゴリズムには主に、メモリベースの手法とモデルベースの手法の二つがある。メモリベースの手法は、推薦を行う際に常に観測データ全体（評価行列 R と内容行列 C ）を参照する。一方、モデルベースの手法は、観測データを用いてユーザーの音楽的嗜好を推定するためのモデルを構築したのち、モデルに基づいて推薦を行う。

これら二つの手法には一長一短がある。メモリベースの手法は事前準備を必要としないが、推薦には時間がかかる。一方、モデルベースの手法は事前のモデル学習に時間がかかるが、推薦自体は迅速に行える。我々の推薦手法はモデルベースであるので、モデル学習コストを抑えることが課題である。

3.1 問題点とアプローチ

DAEMアルゴリズムによるアスペクトモデルの学習コストは、 $|T|$ と $|Z|$ は一定である（本稿ではそれぞれ64と10）ので、 $O(|U||M||T||Z|) \approx O(|U||M|)$ となる。これは二つの深刻な問題を引き起こす。一つめは効率性に関する問題である。観測データが変化する度にコストの大きなモデル学習をやり直す必要がある。二つめはスケーラビリティに関する問題である。時間的かつ

空間的なコスト（学習時間と必要メモリ）は、 $O(|U||M|)$ のオーダーで急激に増加する。効率性とスケーラビリティは実用上極めて重要な要素であるにもかかわらず、これまでほとんど考慮されてこなかった。

本稿ではまず、効率性を改善するため、三方向ベクトルモデルに対するインクリメンタル学習法を提案する。Zhang ら⁹⁾は、協調フィルタリングのための基本的なアスペクトモデルに対するインクリメンタル学習法を提案しており、我々は内容情報を考慮できるように拡張した。さらに、スケーラビリティを改善するため、ユーザ群と楽曲群それぞれに対するクラスタリングをインクリメンタル学手法と組み合わせる手法を提案する。通常、クラスタリングのような近似的な操作を行うと推薦精度は低下するが（精度と効率のトレードオフ¹⁰⁾、本手法は逆に改善に寄与する。

3.2 インクリメンタル学習法

本節では、三方向アスペクトモデルのパラメータを逐次的に更新する手法について述べる。以降、DAEM アルゴリズムを用いて得られたモデルを基本モデル、インクリメンタル学習法を用いて更新されたモデルを更新モデルと呼ぶことにする。更新モデルを生成する際には、以下の三つのケースを別々に扱う。

- (1) 新たに評価スコアを与えた既存ユーザ $u' \in U$ にモデルを適応する場合。
- (2) いくつかの評価スコアを持つ新規ユーザ $u' \notin U$ をモデルに追加登録する場合。
- (3) 評価スコアを持たない新規楽曲 $m' \notin M$ をモデルに追加登録する場合。

最初のケースではモデルサイズは増加しないが、残りの二つのケースでは新規ユーザや新規楽曲の追加により、モデルサイズが増加する。

3.2.1 既存ユーザのプロファイル更新

あるユーザ u の音楽的嗜好を表す条件付き確率分布 $\{p(z|u)|z \in Z\}$ をユーザプロファイルと呼ぶことにする。本研究の確率モデルは、全てのユーザのプロファイルが独立であることを仮定している。したがって、あるユーザが評価スコアを変化させた場合、尤度が最大化された状態が維持されるように該当ユーザのプロファイルのみを更新すればよい。

いま、尤度 L が最大化されているところに、既存ユーザ $u' \in U$ が評価スコアを変化させたために、プロファイル $\{p(z|u')|z \in Z\}$ を更新する問題を考える。ユーザ u' を含まないパラメータは一定であると仮定すると、評価スコア変更後の尤度 L の最大化は、 L 中でユーザ u' を含む項の和の最大化と等価になる。 $L_{u'}$ をユーザ u' に関する尤度とすると、以下で求まる。

$$L_{u'} = \sum_{m,t} n(u', m, t) \log p(m, t|u') \quad (8)$$

$$= \sum_{\langle m,t|u' \rangle} \log p(m, t|u') \quad (9)$$

ここで、ある変数 X に対して $\sum_{m,t} n(u', m, t)X$ のように作用するオペレータ $\sum_{\langle m,t|u' \rangle}$ を便宜的に導入した。

式 (9) は、Jensen の不等式を用いて以下のように変形できる。

$$L_{u'} = \sum_{\langle m,t|u' \rangle} \log \sum_z p(m|z)p(t|z)p(z|u') \quad (10)$$

$$= \sum_{\langle m,t|u' \rangle} \log \sum_z \frac{p(m|z)p(t|z)}{\delta_{m,t}} p(z|u') \delta_{m,t} \quad (11)$$

$$\geq \sum_{\langle m,t|u' \rangle} \sum_z \frac{p(m|z)p(t|z)}{\delta_{m,t}} \log p(z|u') + \sum_{\langle m,t|u' \rangle} \log \delta_{m,t} \quad (12)$$

ここで、以下の補助関数 $\delta_{m,t}$ を導入した。

$$\delta_{m,t} = \sum_z p(m|z)p(t|z) \quad (13)$$

$p(m|z)$ および $p(t|z)$ は一定であることを考慮すると、 $L_{u'}$ の最大化は、式 (12) の第一項の最大化と等価である。すなわち、以下の制約付き最大化問題を得る。

$$\text{Maximize } \bar{L}_{u'} = \sum_{\langle m,t|u' \rangle} \sum_z \frac{p(m|z)p(t|z)}{\delta_{m,t}} \log p(z|u') \quad (14)$$

$$\text{s.t. } \sum_z p(z|u') = 1 \quad (15)$$

この問題はラグランジュの未定乗数法を用いて解くことができる。いま、ラグランジュの未定乗数を λ として、新たな関数 $L_{u'}^*$ を以下で定義する。

$$L_{u'}^* = L_{u'} - \lambda \left(\sum_z p(z|u') - 1 \right) \quad (16)$$

$L_{u'}^*$ を $p(z|u')$ に関して偏微分すると以下が得られる。

$$\frac{\partial L_{u'}^*}{\partial p(z|u')} = \frac{\sum_{\langle m,t|u' \rangle} \frac{p(m|z)p(t|z)}{\delta_{m,t}}}{p(z|u')} - \lambda \quad (17)$$

目的関数 $L_{u'}$ が最大となるには、式 (17) がゼロに等しい必要がある。したがって、以下の等式を得る。

$$\lambda p(z|u') = \sum_{\langle m,t|u' \rangle} \frac{p(m|z)p(t|z)}{\delta_{m,t}} \quad (18)$$

次に、式 (18) を式 (15) に代入することで λ を求める。

$$\sum_z p(z|u') = 1 \Leftrightarrow \lambda = \sum_z \sum_{\langle m,t|u' \rangle} \frac{p(m|z)p(t|z)}{\delta_{m,t}} \quad (19)$$

$$\Leftrightarrow \lambda = \sum_{\langle m,t|u' \rangle} \frac{\sum_z p(m|z)p(t|z)}{\delta_{m,t}} \quad (20)$$

$$\Leftrightarrow \lambda = \sum_{m,t} n(u', m, t) \quad (21)$$

最終的に、以下のユーザプロファイル更新式を得る。

$$p(z|u') = \frac{\sum_{m,t} n(u', m, t) \frac{p(m|z)p(t|z)}{\sum_{z'} p(m|z')p(t|z')}}{\sum_{m,t} n(u', m, t)} \quad (22)$$

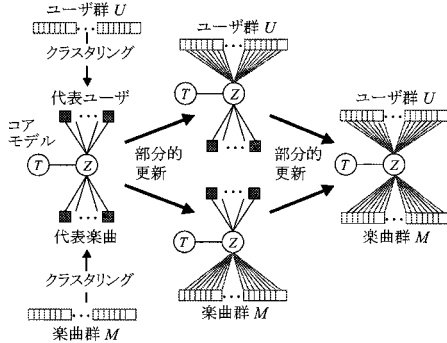


図2 スケーラビリティ改善法

3.2.2 新規ユーザのプロファイル更新

新規ユーザ $u' \in U$ のプロファイル $\{p(z|u') | z \in Z\}$ を新たに生成する問題を考える。ユーザ u' は既存楽曲に対して評価スコア $\{r_{u',m} | m \in M\}$ を付与している。そのため、この場合でも式 (22) を適用することが可能である。このとき、 $p(m|z)$ および $p(t|z)$ は既存ユーザの評価スコアを用いて学習されたものを用いる。

ここで、 $\Delta|M|$ を新規ユーザ u' が付与した評価スコアの数 (u' が既存ユーザの場合は追加・変更した評価スコアの数) だとすれば、更新にかかる計算コストは $O(\Delta|M|)$ である。式 (22) 中で、それらの楽曲に関する項のみ再計算すればよい。

3.2.3 新規楽曲の追加登録

新規楽曲 $m' \notin M$ が与えられたとき、条件付き確率 $\{p(m'|z) | z \in Z\}$ を推定する問題を考える。このケースにもこれまでと同様の議論が適用でき、条件付き確率分布 $\{p(z|m') | z \in Z\}$ を求めたあと $p(m'|z) \propto p(z|m')/p(z)$ とすればよい。このとき、新規楽曲 m' は評価スコアを持っておらず、内容ベクトル $\{c_{m',t} | t \in T\}$ しか利用できないことに注意する。したがって、新規楽曲追加のための更新式は以下で与えられる。

$$p(z|m') = \frac{\sum_t c_{m',t} \frac{p(t|z)}{\sum_{z'} p(t|z')}}{\sum_t c_{m',t}} \quad (23)$$

この更新にかかる計算コストは定数オーダーである。

3.3 スケーラビリティ改善法

本節では、大規模なユーザ数と楽曲数を扱えるようにするためのスケーラビリティ改善法について述べる。図2に概要を示す。まず、DAEMアルゴリズムによる通常の学習法を用いて、少数の仮想的な代表ユーザと代表楽曲に対するコンパクトなコアモデルを構築する。次に、実際のユーザ群 U と楽曲群 M を新規ユーザおよび新規楽曲とみなし、インクリメンタル学習法を用いてコアモデルに追加登録していく。追加登録の順序には二通りあるが、楽曲 M を先に追加登録する方式 (図2中で下のパスに対応) の方が推薦精度が高くなることを4章で報告する実験で確かめた。

		グループ					代表楽曲			
		1	2	3	4	5	1	2	3	
ユーザ	楽曲									
グループ1	1	1	0	φ	3	4	1	0.33	φ	3.50
	2	0	φ	φ	4	3				
グループ2	3	φ	4	4	φ	2	2	3.50	4.00	1.33
	4	φ	3	φ	1	1				

図3 代表ユーザと代表楽曲のための評価行列の生成

代表ユーザと代表楽曲を生成するには、K-means法のようなクラスタリング手法を利用する。まず、ユーザ群 U をユーザの評価スコアベクトルの類似性によってグループ化する。距離尺度として、協調フィルタリングでよく利用されるピアソン相関係数を用いた。一方、楽曲群 M は特徴量ベクトル間のユークリッド距離に従ってグループ化する。代表ユーザと代表楽曲は、各グループの平均的なものとなる。

ここで問題は、コアモデル学習に用いる評価行列 R' と内容行列 C' の生成である。図3に評価行列 R' の計算例を示す。ある代表ユーザがある代表楽曲に付与したスコアは、対応するグループ内のユーザが対応するグループ内の楽曲に付与した実際のスコアの平均値である。新しい内容行列 C' は、グループ内の楽曲の特徴量ベクトルを平均することで得られる。

4. 評価実験

インクリメンタル学習法を利用することで、推薦システムの効率性とスケーラビリティが大幅に向上することは理論的に明らかである。しかし、推薦精度への影響は実験的に検証する必要がある、これが本章で報告する実験の目的である。

4.1 実験条件

信頼性の高い実験を行うには、各ユーザがある程度の数の楽曲に評価を与えている楽曲評価データを用いることが望ましい。この問題に対処するため、評価スコアをWEBサイトから収集することにした。本稿で利用する楽曲評価データは我々の以前の研究報告⁶⁾で利用したものと同一である。

使用した楽曲は、2000年4月から2005年12月までに、日本のシングルCD売り上げランキングで上位20位にランクインしたものである。それらに与えられた評価スコアはAmazon.co.jpから収集した。評価スコアにはユーザIDが付与されているため、あるユーザが複数の楽曲に評価を行っている場合、それらの評価は同一ユーザのものであると判定できる。その後、評価数が4に満たないユーザと楽曲を削減することを繰り返し、評価数が4以上のユーザと楽曲のみを抽出した。最終的に、ユーザ数 $|U| = 316$ 、楽曲数 $|M| = 358$ となった。評価行列 R における全評価スコア数中の4から0の各スコアの割合は、それぞれ57.9%, 19.1%, 8.6%, 4.9%, 9.5%であり、半分以上がスコア4の「大好き」という評価で占められていた。

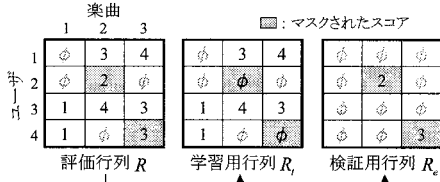


図4 10クロスバリデーションのためのデータ準備

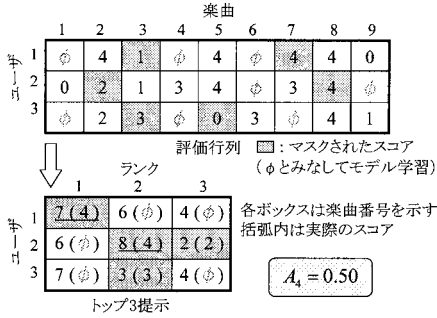


図5 推薦精度の計算例

4.2 評価方法

実験は10クロスバリデーションで行う。まず、図4に示すように、評価行列 R 中の評価要素の10%をランダムにマスクし、学習用行列 R_l と検証用行列 R_e を生成する。推薦精度は、各ユーザごとに x ($x = 1, 3, 10$) 曲を推薦する場合、全ユーザで合計 $x \times N_U$ 曲に着目して評価する。このとき、推薦精度が高いとは、推薦楽曲には「実際には」高い評価スコア(3や4)が与えられていた楽曲の割合が高く、低い評価スコア(0や1)の楽曲がほとんどない状態をさす。しかし、 R_e 内にはわずかな評価要素しかない(R 内の評価要素数の10%なので R_e の密度は0.219%)ので、推薦される $x \times N_U$ 曲全てについて、実際に与えられていた評価スコアを参照することはできない。そこで、 $x \times N_U$ 曲の中から実際に評価スコアが与えられていた楽曲のみを抽出し、そのうち高い評価スコアが与えられている楽曲の割合を算出することにした。ここで、抽出された楽曲数を N ($N < x \times N_U$) と定義しておく。

以下に具体的な算出方法について述べる。いま、 N_r を R_e 中で実際はスコア r が与えられていたが、 R_l 中でマスクされた要素数とする。ここで、 $N = \sum_r N_r$ とし、 N に対する N_r の割合を $A_r = N_r/N$ で定義する。 $x = 3$ とした場合の算出例を図5に示す。ここで注意すべきは、ランダムに楽曲を推薦した場合でも、 A_4, \dots, A_0 は20%にはならないことである。 A_4, \dots, A_0 のチャンスレートは、 R 中の評価要素数に対する評価スコアごとの割合に等しく、それぞれ57.9%, 19.1%, 8.6%, 4.9%, 9.5%である。以降、スコア4の楽曲が推薦される割合 A_4 に着目して議論を進める。 A_4 の値が大きいかほど、精度の高い推薦ができたことを示す。

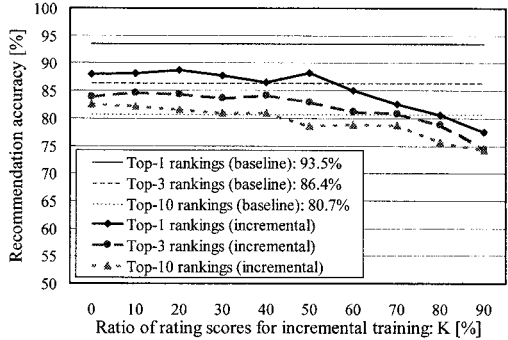


図6 インクリメンタル学習に利用する評価スコアの増加に伴う推薦精度 A_4 の低下

4.3 インクリメンタル学習法の評価

インクリメンタル学習法を評価するため、3.2節で述べた三つのケースに分けて実験を行ったので報告する。

4.3.1 既存ユーザに対する推薦

本実験では、基本モデルを構築するのに利用する評価スコアの割合が低下するに伴い、推薦精度がどのように低下するかを調査した。また、基本モデルと更新モデルとの推薦精度の差を調査した。

まず、実験手順を説明する。学習用行列 R_l を用いて、1個の基本モデルと10個の更新モデルを構築した。基本モデルは R_l をそのまま学習用データとし、DAEMアルゴリズムによる通常の学習法を適用することで得られた。更新モデルは以下の以下の手順で得られた。

- (1) 学習用行列 R_l 中の評価スコア*のうち $K\%$ ($K = 0, 10, 20, \dots, 90$)をランダムにマスクすることで、一時的な学習用行列 R'_l を生成する。
- (2) R'_l を学習用データとし、DAEMアルゴリズムによる通常の学習法を適用することで、一時的な基本モデルを得る。
- (3) マスクされた $K\%$ の評価スコアをインクリメンタルに追加し、すなわち学習用行列 R_l を用いて、更新モデルを得る。

このようにして計11個の異なるモデルが得られるが、モデル構築に利用した評価スコアの数是最終的に等しくなる。ただし、基本モデル学習分とインクリメンタル学習分の割合が異なる。各モデルの推薦精度を評価するには、検証用行列 R_e を利用した。我々はこれまで述べた実験手順を10回繰り返した。4.2節で述べた10クロスバリデーションにおいて、学習用行列 R_l は10通りできるからである。

図6に実験結果を示す。インクリメンタル学習法により、評価スコアが追加されてもユーザの好みに合うように適切に推薦結果を適応させることができた。基本モデルに逐次的に追加される評価スコアを、基本モ

*以降特に注釈がないかぎり、単に「評価スコア」とあった場合、 ϕ 以外の実際にユーザが付与したスコア(0から4)のことを指すものとする。

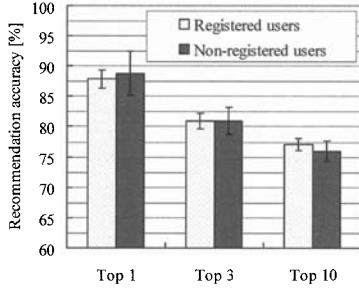


図7 既存ユーザと新規ユーザに対する推薦精度 (A_4) の比較

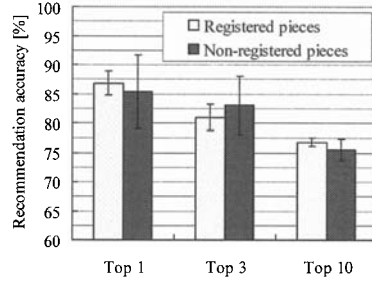


図8 既存楽曲と新規楽曲の推薦精度 (A_4) の比較

デルを学習するために利用する評価スコアと同数まで増やしても (K を 50 まで増加), 推薦精度はほとんど低下しないことが分かった. 各ユーザにランク一位の楽曲のみを推薦する場合 ($x=1$) を見ると, 基本モデルと更新モデルとの推薦精度の差は 5% 程度であり, 十分に高い推薦精度を保つことができた.

4.3.2 新規ユーザに対する推薦

本実験では, 既存ユーザに対する推薦精度と新規ユーザに対する推薦精度を比較した. この差が小さいほど, 優れた推薦ができたことを示す.

まず, 以下に実験手順を説明する.

- (1) ユーザ群 U から 10% の数のユーザをランダムに抽出して U_{new} とし, 新規ユーザとみなす. 一方, 残りのユーザを U_{reg} とし, 既存ユーザとみなす.
- (2) 学習用行列 R_i から U_{new} に対応する行を削除することで, 縮退した評価行列 R'_i を生成する. すなわち, R'_i のサイズは R_i の 90% である.
- (3) R'_i を学習データとして一時的な基本モデルを構築する. この基本モデルには U_{new} のユーザプロフィールは含まれていない.
- (4) まず, 既存ユーザ U_{reg} に対する推薦精度を以下の手順で求める.
 - (a) R'_i を式 (22) に入力して (R'_i を再度利用することになる) U_{reg} のユーザプロフィールを更新し, 更新モデルを得る.
 - (b) 更新モデルによる推薦精度を検証用行列 R_e 中の U_{reg} に対応する行を用いて算出する. 次に, 新規ユーザ U_{new} に対する推薦精度を以下の手順で求める.
 - (a) 手順 (2) で削除された評価データを式 (22) に入力して U_{new} のユーザプロフィールを生成し, 更新モデルを得る.
 - (b) 更新モデルによる推薦精度を検証用行列 R_e 中の U_{new} に対応する行を用いて算出する.

4.3.1 節と同様に 10 クロスバリデーションでは学習用行列 R_i は 10 通り生成されるので, 上記の (1) から (4) の手順を R_i を切り替えながら 10 回行った. 推薦精度の平均と分散を求めるために, このような 10 クロスバリデーションを 10 回行った.

図 7 に実験結果を示す. インクリメンタル学習法により, 既存ユーザと同様に新規ユーザに対しても精度よく推薦ができた. 有意水準 5% で F 検定を行った結果, 両者の分散には違いがあることが分かった. しかし, Welch の t 検定を行ったところ, 両者の推薦精度の平均には差がないことが分かった.

4.3.3 新規楽曲の推薦

本実験では, 既存楽曲の推薦精度と新規楽曲の推薦精度を比較した. この差が小さいほど, 優れた推薦ができたことを示す.

まず, 以下に実験手順を説明する.

- (1) 楽曲群 M から 10% の数の楽曲をランダムに抽出して M_{new} とし, 新規楽曲とみなす. 一方, 残りの楽曲を M_{reg} とし, 既存楽曲とみなす.
- (2) 学習用行列 R_i から M_{new} に対応する列を削除することで, 縮退した評価行列 R'_i を生成する. すなわち, R'_i のサイズは R_i の 90% である.
- (3) R'_i を学習データとして一時的な基本モデルを構築する. この基本モデルには M_{new} と Z との確率的な関係は含まれていない.
- (4) まず, 既存楽曲 M_{reg} の推薦精度を以下の手順で求める.
 - (a) R'_i を式 (23) に入力して (R'_i を再度利用することになる) M_{reg} の確率的な関係を再計算し, 更新モデルを得る.
 - (b) 更新モデルによる推薦精度を検証用行列 R_e 中の M_{reg} に対応する列を用いて算出する. 次に, 新規楽曲 M_{new} に対する推薦精度を以下の手順で求める.
 - (a) 手順 (2) で削除された評価データを式 (23) に入力して M_{new} と Z の確率的な関係を推定し, 更新モデルを得る.
 - (b) 更新モデルによる推薦精度を検証用行列 R_e 中の M_{new} に対応する列を用いて算出する.

4.3.1 節と同様に 10 クロスバリデーションでは学習用行列 R_i は 10 通り生成されるので, 上記の (1) から (4) の手順を R_i を切り替えながら 10 回行った. 推薦精度の平均と分散を求めるために, このような 10 クロスバリデーションを 10 回行った.

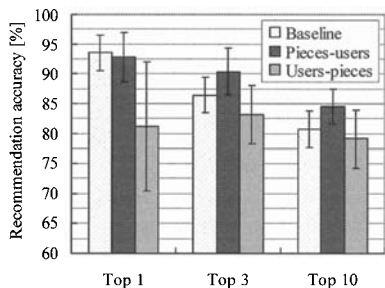


図9 スケーラビリティ改善法による推薦精度 (A_4) の変化

図8に実験結果を示す。インクリメンタル学習法により、既存楽曲と同様に新規楽曲も精度よく推薦ができた。有意水準5%でF検定とt検定を行った結果、この実験においても両者の推薦精度の平均に差は見られなかった。

4.4 スケーラビリティ改善法の評価

本実験では、2章で述べたベースラインシステムと二つのスケラブルなシステムの推薦精度を比較した。二つのスケラブルなシステムは仮想的なユーザ群と楽曲群に対して同じコアモデルを構築するが、実際のユーザ群と楽曲群を追加登録するときの順序が異なる。ここで、最初に楽曲群 M を追加し、次にユーザ群 U を追加するのを *pieces-users* システム (図2中の下のパス)、逆の順序で追加するのを *users-pieces* システム (図2中の上のパス) と呼ぶことにする。

図9に実験結果を示す。*pieces-users* システムが他の二つのシステムより高い推薦精度を示した。各ユーザにランク一位の楽曲のみを推薦する場合 ($x=1$) では、*pieces-users* システムはベースラインシステムと同等の推薦精度を示し、各ユーザにランク上位の3曲あるいは10曲を推薦する場合 ($x=3, 10$) では、推薦精度が有意に高かった。これは、ユーザ群 U と楽曲群 M をクラスタリングすることで、評価行列のサイズおよびスパースネスが減少し (図3参照)、DAEM アルゴリズムにより推定されるモデルパラメータの品質が向上したからである。通常のEMアルゴリズムより局所解に陥りにくいDAEMアルゴリズムを利用して、もとの評価行列 R は極めてスパースで局所解に陥りやすいため、スパースネスを減少させるスケラビリティ改善法は効果的であった。

対照的に *users-pieces* システムでは、ベースラインシステムより推薦精度は低下した。実際のユーザ群 U のユーザプロフィールを生成するための式(22)は楽曲群 M に関するサメーションを含んでいるため、仮想的な楽曲群の内容行列ではなく、実際の楽曲群 M の内容行列を利用するほうが望ましい。一方、式(23)はユーザ群 U に関するサメーションを含んでいない。そのため、実際のユーザを追加登録する前に、実際の楽曲を追加登録しておく必要がある。

5. おわりに

本稿では、ハイブリッド型楽曲推薦システムの効率性とスケラビリティを改善する方法について述べた。具体的には、ユーザの楽曲評価と楽曲の音響的特徴を統合するための確率的推薦モデルに対するインクリメンタル学習法を提案した。本手法は、ユーザや楽曲を効率的にシステムへ追加登録できるように、確率モデルのパラメータを部分的に更新することを可能にした。さらに、インクリメンタル学習法をクラスタリング手法と組み合わせることで、スケラビリティ改善法を提案した。本手法は確率モデルの学習を劇的に高速化するだけでなく、推薦精度の改善にも寄与することが分かった。すなわち、推薦精度と効率性・スケラビリティとの間のトレードオフを解決した。

今後、本システムを大規模楽曲データベースに適用し、試験運用を行う予定である。このとき、いつまでモデルをインクリメンタルに更新していけるのか、すなわち、いつ推薦精度の低下が許容できなくなるのかを調査する必要がある。また、さまざまなジャンルの楽曲を含むデータベースに対してモデルを学習し、隠れ変数の音楽的な意味を調査することは興味深い。

謝辞 本研究を進める上で有益な議論をして頂いた麻生英樹氏 (産業技術総合研究所) に感謝する。本研究は日本学術振興会特別研究員 (DC1) 科研費の補助を受けた。

参考文献

- 1) A. Uitdenbogerd and R. van Schyndel, "A Review of Factors Affecting Music Recommender Success," *ISMIR*, 2002, pp. 204–208.
- 2) G. Linden and B. Smith, "Amazon.com Recommendations: Item-to-Item Collaborative Filtering," *IEEE Internet Computing*, pp. 76–80, Vol. 7, No. 1, pp. 76–80, 2003.
- 3) C. Anderson, "The Long Tail: Why the Future of Business Is Selling Less of More," Hyperion, 2006.
- 4) K. Hoashi, K. Matsumoto, and N. Inoue, "Personalization of User Profiles for Content-based Music Retrieval based on Relevance Feedback," *ACM Multimedia*, 2003, pp. 110–119.
- 5) B. Logan, "Music Recommendation from Song Sets," *ISMIR*, 2004, pp. 425–428.
- 6) 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," *ISMIR*, 2006, pp. 296–301.
- 7) A. Popescu, L. Ungar, D. Pennock, and S. Lawrence, "Probabilistic Models for Unified Collaborative and Content-based Recommendation in Sparse-data Environments," *UAI*, 2001, pp. 437–444.
- 8) N. Ueda and R. Nakano, "Deterministic Annealing EM Algorithm," *Neural Net.*, Vol. 11, No. 2, pp. 271–282, 1998.
- 9) L. Zhang, C. Li, Y. Xu, and B. Shi, "An Efficient Solution to Factor Drifting Problem in the pLSA Model," *CIT*, 2005, pp. 175–181.
- 10) L. Ungar and D. Foster, "Clustering Methods for Collaborative Filtering," *Workshop on Recommendation Systems*, 1998.