解　説

推薦システムのオフライン評価手法

Offline Evaluation for Recommender Systems

土方 嘉徳
Yoshinori Hijikata
大阪大学大学院基礎工学研究科
Graduate School of Engineering Science, Osaka University
hijikata@sys.es.osaka-u.ac.jp, http://soc-research.org

keywords: recommender system, offline experiment, evaluation metric, accuracy, novelty, serendipity, diversity

Summary

「ショートノート」は 200 ワード，それ以外は 200〜500 ワード以内の英文で summary を記す

1. は　じ　め　に

インターネット上のサービスや家電製品において，推薦機能が提供されるようになって久しい．オンラインショッピングの Web サイトでは，サイトを訪れるたびに，新しい商品を推薦してくれたり，人気の商品を教えてくれたりする．テレビやハードディスクレコーダでは，お薦めのテレビ番組を教えてくれる．このようなサービスや機能は，推薦システム (recommender system) と呼ばれる機構（日本では情報推薦 (information recommendation) と呼ばれることも多い）により自動で提供されることが多い．

推薦システムの起源は，1980 年代後半にまでさかのぼる．当時研究者の間で普及し始めた電子メールやネットニュースにおいて，ユーザの必要なメールや記事だけ表示することを目的としていた．当時は，それまでにあった情報検索技術を援用することが多かった [Loeb 92]．1990 年代半ば以降，データセットの整備と共に多くの機械学習アルゴリズムが適用されるようになり，急速に研究分野が大きくなったと言える [Resnick 97, Riecken 00]．このあたりの歴史については [土方 07] が詳しい．

これまで推薦サービスを提供できるのは，一部の大手企業や先進的なネット企業に限られてきた．大手企業は，多くの商品やコンテンツを電子化する体力があったことと，すでに膨大な数の顧客を抱えていたからである．また，先進的なネット企業は，設立当初から商品やコンテンツを電子化していたことと，サービスの試用公開期間を設けることにより多くのユーザを獲得してから，本サービスを立ち上げるという柔軟な運用が採れたからである．

しかし，多くのユーザがパソコンやスマートフォンで，インターネット上の情報にアクセスできるようになった今，中小企業や伝統的な企業も，商品やコンテンツ，顧客データを電子化し，インターネット上でのサービスを展開し始めている．それらの企業は，今まさにこれから，様々な推薦アルゴリズムを試し，推薦サービスを提供しようとしているところであろう．

推薦機能を実現するには，どのようなデータ（ユーザに関するデータとアイテム (item) [*1]に関するデータ）を使うかと，そこからどのような特徴量を抽出するか，そしてどのような推薦アルゴリズムを適用するかを決めなくてはならない（推薦アルゴリズムの詳細については文献 [神嶌 07, 神嶌 08a, 神嶌 08b] を参照 [*2]）．これらの選び方によって，その推薦の質は大きく変わってくる．そのため，これらを変動させて推薦結果の評価を行わなければならない．しかし，その評価手法には様々なものが存在し，どれを用いるかは簡単に決められるものではない．

推薦システムを評価する上で最も基本となる観点は，ユーザの興味や嗜好に適合する推薦が行えるかどうかという正確性 (accuracy または correctness) に関するものである．しかし，一口に正確性に関する評価と言っても，データセット中のアイテム全体に対する評価値（ユーザがアイテムに付与する興味や嗜好に関する評価値）の予測能力を測るべきだという考え方もあれば，ユーザに示す予定の上位 5 アイテムのみ正確に予測できれば良いとする考え方もある．また，ユーザがアイテムに与える興味・嗜好の評価値が，バイナリ（好き／嫌い）で与えられるか，N 段階の離散値（N は 3 以上）で与えられるかによっても，評価に用いることができる手法は変わってくる．また，近年では単にユーザの興味や嗜好に合うアイテムを推薦できれば良いとするだけでなく，ユーザが知らないものであったか（新規性）や，ユーザに驚きや発見をもたらすものであったか（意外性）などの，新しい観点も考慮すべきだという主張もある [Herlocker 04]．これらの評価は，正確性の評価よりも困難なものとなる

∗1　推薦システムの研究分野では，推薦対象の商品やコンテンツのことをアイテムと呼ぶことが多い
∗2　最新版は http://www.kamishima.net 参照

であろう．評価手法には様々なものが提案されているが，評価対象の推薦システムの目的や，用いるアルゴリズムの長短所を考慮して，適切な評価手法を用いなければならない．

本稿の目的は，推薦システムの研究分野において，伝統的に用いられている様々な評価手法を網羅的かつ体系的に説明することにある．また，各手法をなるべくワンストップで（他の教科書や情報源を参照することなく）学ぶことができるようにすることにある．これまで，推薦システムの研究分野において，このような目的で書かれた文献には [Herlocker 04] がある．この文献には，従来の嗜好の正確性に関する評価指標 (evaluation metric) が体系的にまとめられているだけでなく，アイテムの新規性や意外性を考慮した新しい評価指標の考え方についても述べられている（本稿では，これらに関する指標をまとめて発見性と呼ぶことにする）．しかし，この文献では，情報推薦や情報検索でよく用いられる評価指標の全てが紹介されているわけではない．また，この文献の発刊以降，新規性や意外性を考慮した具体的な評価指標もいくつか出現している．また，示されている評価指標を算出する具体的な手順について書かれていないものもあり，ワンストップで学習することができない．本稿には，上記の評価指標も含めることとし，評価指標を算出する具体的な式も明示する．読者が本稿を読むだけで各評価手法を実践できるようにする．

なお，推薦システムの評価手法は大きく分けると，オンライン評価（オンライン実験とも言う．英語では，online experiment または live user experiment）とオフライン評価（オフライン実験とも言う．英語では，offline experiment または offline analysis）に分けられる（2 章で詳しく説明する）．本稿では，伝統的に用いられているオフライン評価に絞って説明する．オンライン評価は自由度が高く比較的新しい評価手法であるため，その評価方法の体系化は，また別の機会に行いたいと思う．なお，説明する評価指標の一部はオンライン評価にも用いることができるため，オンライン評価を行う予定の読者も参考にすると良い．

本稿の構成は，以下のようである．2 章では，推薦システムの目的（達成するタスク）の分類について説明する．また，推薦システムの評価方法の分類の一つであるオフライン評価とオンライン評価の違いと，その長短所について述べる．さらに，考慮すべきデータセットの特徴と，データセットの分割方法についても述べる．3 章では，推薦システムの推薦の正確性を評価する指標を紹介する．適合率や再現率と言った基本的な評価指標から，推薦リストを閲覧するユーザ行動まで考慮した最新の指標まで紹介する．4 章では，推薦の発見性を評価する指標を紹介する．発見性を新規性，意外性，多様性の 3 つに分け，それぞれを評価する具体的な指標を紹介する．最後に，5 章でまとめを述べる．

2. 評価手法の分類

2·1 推薦システムのタスク

本稿では，推薦システムの評価手法について紹介していくが，推薦システムの目的によって用いるべき評価手法が異なってくる．Herlocker らは推薦システムがユーザに推薦を提供する目的をメジャーなもの 2 つ（以下の 1 と 2）と，それよりややマイナーなもの 4 つ（以下の 3〜6）に分類している [Herlocker 04]．まずは，これらについて簡単に説明する．

(1) フィルタリング (Annotation in context)
これは，ニュース記事の推薦や電子メールのフィルタリングでよく見られるが，ユーザがある限られたアイテムの中から，どのアイテムを消費すべきかを提示するものである．このタスクでは，もともとのアイテムの提示順序は維持したまま，読むべきかどうかをアノテーションで示す．また，消費する可能性の極めて低いアイテムは除去してしまう．

(2) ランキングリスト提示 (Find good items)
これは，膨大なアイテムの中から，ユーザに適したアイテムを順位付きのリストで提示するものである（図 1 に Amazon.com での提示例を示す）．通常，リストはある決まった長さ（5 個や 10 個など）で区切られて提示されることが多い．

(3) 全適合アイテム列挙 (Find all good items)
これは，ある限られたアイテムの中から，ユーザの要求に適合するアイテムを漏れなく列挙するものである．多くの推薦システムは，膨大なアイテムの中からユーザに適合するものを選ぶものが多いため，あまりこの目的を持つシステムは見当たらないが，例えば関連する科学論文や特許を提示するタスクは，これに当てはまる．

(4) 提示順推薦 (Recommend sequence)
これは，ユーザに 1 アイテムずつ順に，決まった順番で提示するものである．例えば，音楽のプレイリスト生成がこれに当たる．

(5) 閲覧支援 (Just browsing)
これは，購入や消費目的で推薦リストを閲覧するのではなく，単に楽しみで推薦リストを閲覧しているだけのユーザを支援するものである．新しいテクノロジーが好きな人の中には，自分の興味や嗜好をどうモデル化してくれるのだろうかや，それに基づきどんなアイテムを推薦してくれるのだろうかという興味本位で使う人もいるだろう．このような目的で使っているユーザを満足させるタスクがこれに当たる．

(6) 信頼性提示 (Find credible recommender)
これは，推薦の性能を確かめたがっているユーザを支援するものである．ユーザの中には，最初からは推薦メカニズムを信じていない者もいる．そのようなユーザは最初に推薦システムの評価を行おうとするが，その評価に必要な情報を提示するような支援を行うのがこれに当たる．

タスクの 5 と 6 は，ユーザの興味や嗜好に合うアイテムを推薦することだけを目標としているわけではない．そのため，これらを支援する仕組みを評価するには，本稿で説明する評価指標だけでは不十分である．例えば，システムがどのような機構によりアイテムを推薦しているのかを確認できるような透明性 (transparency) や，アイテムの推薦された根拠を説明した場合の妥当性 (validity) なども評価する必要があると思われる．これらの評価には，2·2 節で説明するオンライン評価の適用も考える必要がある．

図 1　An example of recommendation list (Find good items)

2·2　オンライン評価とオフライン評価

推薦システムの評価方法を大きく二つに分けると，オンライン評価とオフライン評価に分けられる [Herlocker 04, Gunawardana 09]．オンライン評価は，ヒューマン・コンピュータ・インタラクションの研究者にはなじみ深いであろう．実装したシステムを実際にユーザに使ってもらい，タスクの実施効率を調べたり，その後のアンケートによりユーザの主観的な評価を聞き出したりする方法である（図 2 の上部参照）．一方オフライン評価は，データ工学や自然言語処理，情報検索の研究者になじみ深いであろう．あらかじめユーザにいくつかのアイテムを評価させてそれを正解データとして保存しておき，提案するアルゴリズムで推薦を行った時に，推薦されたアイテムがこの正解データに適合するかどうかを確認する方法である（図 2 の下部参照）．この方法では，統一した評価基準で評価することが重視されるため，いくつかのデータセットが公開・共有されている（以下，本稿ではオープンデータセットと呼ぶ）．既存のオープンデータセットは文献 [奥 13] にまとめられているので，参考にすると良い．

オンライン評価とオフライン評価には，それぞれ利点と欠点がある．筆者は，オンライン評価とオフライン評価を，9 個の観点から特徴づけた（表 1 参照）．以下，順に説明する．

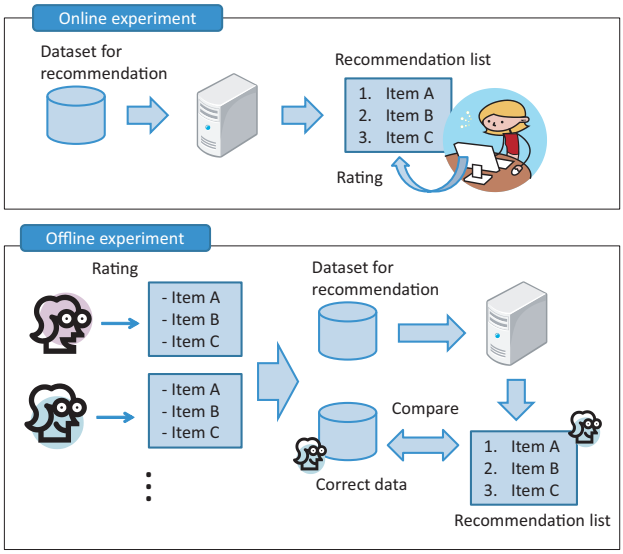

図 2　Evaluation process of online experiment and offline experiment

(1) 総合評価 (Overall evaluation)

オンライン評価とオフライン評価の最も大きな違いを挙げるとすると，システムそのものをユーザの直感から，直接的に評価することができるかどうかである．オフライン評価は，システムの評価が必ず間接的になる．なぜなら，オフライン評価はシステムの総合評価に影響を与える要素の一つについて（例えば正確性）正解データからその妥当性を検証することはできるが，システムの総合評価を直接ユーザから引き出してはいないからである．

一方オンライン評価では，実際にユーザに推薦システムを利用してもらいユーザに質問ができるため，システム全体に対する総合評価を直接聞くことができる．ユーザのシステムに対する総合評価は，必ずしも推薦の正確性だけから付けられるものではなく，実際にはインタフェースの見た目が洗練されたものかどうかや，システムを利用する前にかけた労力なども影響する．このような観点は，オフライン評価では難しいと言える．

(2) 再現性 (Reproducibility)

評価実験は他者による追試が可能である方が好ましい．この追試が容易であるかどうかを示すのが再現性である．オフライン評価で，特にオープンデータセットを使う場合は，誰もが同じ条件で実験を行うことができ，実験の再現が行いやすい．一方，オンライン評価では，実験環境を設定し，性別や年齢などを考慮して被験者を集め，被験者にタスクの指示を与えて実験しなければならない．また，リアルタイムにユーザの行動を測定したり，実験後にアンケートを取ることもある．他者によりこれらの設定方法や測定方法の詳細を再現し尽くすことは難しい．

表 1 The difference between online evaluation and offline evaluation

	Overall evaluation	Reproducibility	Measurement consistency	Preparation cost	Extensibility	Time sensitivity	Further analysis	Stabililty	Scalability
Online	Good	Bad	Bad	Bad	Good	Good	Good	Bad	Bad
Offline	Bad	Good	Good	Good	Bad	Bad	Bad	Good	Good

(3) 一貫性 (Measurement consistency)
評価実験は誰もが共通の認識を持つ評価指標により実施されることが好ましい．例えば，実行時間 (秒) や占有されるメモリ容量 (byte) などの評価指標は，誰にとっても同じ定義を持っていると言える．オープンデータセット中のデータは，ユーザがアイテムに与えた評価値という決まった形式で与えられることが多い．そのため，このデータセットを用いてオフライン評価を行う場合，計測する評価指標は比較的，多くの研究者にとって共通の概念として認識されていることが多い．

一方オンライン評価では，推薦システムを利用したユーザに，様々な観点から評価を尋ねることができる．そのため，同じ名前の評価指標でも，その意味が研究者により異なる可能性がある．例えばシステムが推薦したアイテムへの正確性をユーザに問う時に，ある研究者は推薦されたアイテムをその場で消費したいと思ったかどうかで判定させるかもしれない．一方，他の研究者は推薦されたアイテムが自身の長期の興味に合っているかどうかで判定するように指示するかもしれない．この微妙な指示の違いによって，得られる評価指標の意味が異なってくる可能性がある．

また，このような評価指標の解釈の違いは，実験を行う研究者間においてのみ見られるわけではない．実験対象のユーザ側にも見られる．ユーザに質問を行う際は，ユーザがその質問の意図を全く同じように解釈することが望ましい．オフライン評価では，タスクがシンプルであるため（通常，提示されたアイテムに対して N 段階の評価値を付与する．安定性の項目で詳細に説明する），質問への回答基準がユーザ間で一致する可能性が高い．しかし，オンライン評価では推薦システムを利用した後に（あるいは利用中に）回答を入力するため，その回答にはシステムの利用状況やそこでの経験が影響してくる．そのため，ユーザ間で質問に対する解釈の違いが発生する可能性が高くなる．オンライン評価では，質問への回答基準をいかに明確化するかが重要である．

(4) 実験前コスト (Preparation cost)
本実験を始める前に，どれだけ準備に人手や時間がかかるかも考慮する必要がある．オフライン評価で，オープンデータセットを利用する場合は，事前の準備にかかるコスト（実験前コスト）が低くなる（表 1 では，この場合を想定している）．しかし，テストするアルゴリズムにおいて，既存のデータセットが利用できず，しかもそれが協調フィルタリング [Resnick 94, Sarwar 01] の技術を用いていた場合は，大量の評価値データ（ユーザのアイテムに対する評価値）が必要となり，実験前コストが非常に高くなる．

一方，オンライン評価でも，上記と同様のことが言える．しかし，評価値の入力コストだけではなく，リアルタイムに何を測定するのか，ユーザにどのような質問を行うのか，ユーザにいつ質問を行うのかなど，事前に決めないといけないことが多い．そして，これらは最終的にはトライ＆エラーで調整するしかない．そのため，通常オンライン評価での実験前コストは高くなりがちである．

オンライン実験を行う場合は，なるべく先人の知見を利用するのが良い．Knijnenburg らは，オンライン実験により多くの質問を試し，どれが有効であったかを明らかにしている [Knijnenburg 12]．特にこの文献中の付録に，有効だった質問項目とそうでなかった質問項目の一覧があるので参考にされたい．

(5) 拡張性 (Extensibility)
評価実験を行う際に，新しい評価指標を加えられるかどうかは重要である．例えば，オフライン評価において，用いるデータセットにはアイテムに対する評価値がバイナリで付与されていたとする．この場合，アイテムがユーザの嗜好に適合していたか否かのみを問う評価指標は用いることができるが，予測した嗜好の程度の正確さを評価する場合には，情報が少なすぎる．まして，新規性や意外性に関する評価指標を用いることはできない．一方，オンライン評価であれば，どのような質問をどのような粒度でユーザに尋ねるかについては自由度があるので，新たな評価指標を導入しやすい．

(6) 適時性 (Time-sensitivity)
ユーザが推薦システムを使い始めて，それを使いこなしていく過程において，ユーザへの推薦の正確性やユーザのシステムへの総合評価がどのように変化するのかを分析したいこともあると思われる．つまり，推薦システムを時間の経過とともに評価を行いたい場合である．その場合は，リアルタイムにユーザに推薦を行えるオンライン評価の方が有利である．推薦過程の任意のタイミングにおいて，直接にユーザに質問を行うことができる．

一方，オフライン評価でオープンデータセットを用いる場合は，必ずしも時間経過を伴った分析ができるとは限らない．オープンデータセットには，ユーザがあるアイテムを評価付けした時間（タイムスタンプ）が記録されていないことがあるからである．また，たとえタイムスタンプがあったとしても，任意の時間において特定の

アイテムがどの位好きであったかを得ることは不可能である．

(7) 分析可能性 (Further analysis)

システムに与えた入力，システムの実行中の状態，システムの出力結果，それに対するユーザの反応や内部状態などから，実験結果に対してより多様で深い分析ができる方が望ましい．しかし，オフライン評価では限られた種類の情報（多くの場合ユーザのアイテムに対する評価値のみ）しか用いることができない．もし，推薦の正確性が向上したという結果が得られたとしても，その理由まで分析することは困難である．

一方オンライン評価では，ユーザに何を入力してもらい，何のアンケートを取るかは自由に決めることができる．特に上記のようなユーザの内部状態に関するアンケートを取っておくと，システムの出力とユーザの内部状態，ユーザの意思決定（や総合評価）といった因果関係の有無を検証することもできる（文献 [Bollen 10, Ekstrand 14] が解析例）．また，自前でシステムを実装するのであれば，ユーザのシステム上での入力の時間経過や，より詳細なマウス操作ログなども記録できる．これらも分析に利用することができるであろう．

(8) 安定性 (Stabililty)

推薦システムの評価は，ユーザがアイテムやシステムに与えた評価値を基にして行われることが多い．ユーザが同じアイテムやシステムに対して評価値を付ける場合には，常にその値が一定している方が良い（値の揺らぎが少ない方が良い）．すなわち，あるユーザに同じアイテムに対して何度評価付けさせても，毎回同じ評価値を返すことができる方が，評価値の信頼性が高くなる．しかし，ユーザはもともとアイテムの評価において，付与する評価値は常に一定しているわけではなく（入力する値に揺らぎがあり）[Hill 95, Cosley 03, Amatriain 09]，これが推薦システムの評価を難しくしている．Hill らや Cosley らは，あるユーザにあるアイテムに対して評価付けさせ，一定期間経過してから同じアイテムに評価付けさせると，一回目と異なる値を入力することを確かめている [Hill 95, Cosley 03]．また，ニュートラルに近い評価値ほど，値の不一致が多くなることも明らかにされている [Amatriain 09]．

オープンデータセットの各データは，シンプルなインタフェースで，ユーザにアイテムに対する評価値を入力させて，取得していることが多い（図 3 参照．これは MovieLens[*3]の例）．タスクがシンプルであり，それぞれのアイテムにおいて評価に至るまでのインタラクションが大きく変わることがない．そのため，一度あるアイテムに対して入力させた評価値を，一定期間明けて再度入力させた際，同じ値（または近い値）を入力する可能性が比較的高いと思われる．

しかし，オンライン評価では，様々なアンケートを取ることができる代わりに，推薦システムを利用する際のタスクが複雑なものになったり，そのアンケートに至るまでのインタラクションが多様になったりする可能性がある．その場合，アンケートで入力する値には大きなぶれが生じる可能性がある．

図 3 Interface for inputting ratings to items in MovieLens

なお，この評価のぶれは，評価付けのスケールを詳細にするほど軽減すると言われている [Garner 60]．しかし，詳細なスケールはユーザの評価付けにかかる時間を増大させるため [Sparling 11]，商用システムではあえて粗いスケールを採用していることもある．

(9) スケーラビリティ (Scalability)

推薦システムの評価においては，データ規模が重要である．ユーザ数やユーザが評価したアイテム数が少ないと実験結果の信頼性が上がらないからである．オフライン評価のためにデータセットを作成する際には，ユーザにアイテムに対する評価値をつけてもらうことが多い．このタスクはシンプルなものであり，この入力に使用するインタフェースもシンプルなものになる．そのため，比較的多くの入力データが集まりやすいと言える．また，実験者が監視したり，タスクの指導を行わなくても，すべて Web 上で実施することもできるため，不特定多数のユーザに実験に参加してもらえる．

一方オンライン評価では，評価対象のシステムをユーザに使ってもらう必要がある．ここではタスクがより複雑になりがちで，その場合はタスクの指導や，場合によっては実験中の監視が必要となる．そうすると，多くの被験者を一度に実験することは困難である．そのため，データの量を確保できるかというスケーラビリティの点では，オフライン評価に劣る．

このようにオフライン評価とオンライン評価には，それぞれ利点と欠点がある．より深い分析を行い，より詳細な考察を行いたい場合は，オンライン評価が有利である．アンケートでユーザの心理状態について問うておけば，より分野横断的な研究も可能となるであろう．しかし，いくつかのアルゴリズムや，アルゴリズム中の異なるパラメータ設定を，試行錯誤的に試しながら評価する場合は，オンライン評価ではコストがかかりすぎる．研究開発の進行状況や，評価したい項目に応じて，これらを使い分ける必要がある．なお，本研究では推薦システムの評価の基本となるオフライン評価に焦点を当て，そ

∗3 MovieLens: http://movielens.org/

の評価指標について説明する．

2·3 データセットの特徴

オフライン評価では，オープンデータセットや，独自にあらかじめ収集しておいたデータセット（企業ではアクセスログや購買ログであることも多いであろう）を用いて実験を行う．具体的な評価方法を考える前に，扱うデータセットがどういう特性を持っているかを理解しておく必要がある．Herlocker らは，データセットにおいて考慮しておくべき観点を整理している [Herlocker 04]. ここでは，そのいくつかを取り上げた上で，著者により新しく考慮すべき観点も加えて，詳しく説明する．

(1) 明示的入力か暗黙的入力か (Explicit or implicit)

データセットの核となるのは，ユーザのアイテムに対する評価値である．オープンデータセットでは，ユーザにアイテムに対する評価値を明示的 (explicit) に入力させていることが多い．しかし，企業などが独自のサービスにおいてデータを収集する際には，上記の情報を暗黙的 (implicit) に獲得している場合もある．最も代表的なデータはアクセスログである．ユーザがどのアイテムのページを閲覧したかや，どのアイテムを購入したかというデータである．この場合，分かるのはユーザが閲覧したか，購入したかであり，ユーザが気に入ったかどうかやその程度までは分からない．暗黙的に獲得したデータは，その評価値の信頼性が低くなるため，そこから算出される評価指標の値についても，高い信頼を期待することはできない．どのように興味や嗜好のデータを獲得するかと言うユーザプロファイリングについては，文献 [土方 04] が詳しいので参照されたい．

(2) スケーリング (Scaling)

ユーザに明示的に評価値を入力させる場合，その評価値のスケーリングは重要である.「好き」か「嫌い」かのバイナリにするか，さらにそれに「ニュートラル」を加えるか，それともももっと粒度を上げて，5 段階や 7 段階にするかという問題である．また，バイナリ値の場合は，複数アイテムをユーザに提示し，好きなものにチェックを入れるという聞き方もあり得る．その場合,「好き」と「それ以外」に分けられる（これを “unary” とも言う [Herlocker 04]）．この場合，ユーザの入力の負担は小さくなるが，得られる情報は少なくなる.「ニュートラル」を加えた 5 段階や 7 段階（時に 9 段階）での入力は，リッカート尺度 (Likert scale) とも言われる．また，ときに「ニュートラル」を設けずに，ユーザにとって馴染みのある数である 10 段階を用いることもある [*4]．段階の粒度をどの程度に設定すれば良いかは，ドメインの特性，そのドメインにおけるユーザ群の判定能力と，ユーザにどの程度の負担を強いることが可能かに依存する．

(3) 評価値の偏り (Rating bias)

付与される評価値の偏りも考慮する必要がある．例えば映画では，多くのユーザは自分が観たい作品を観て評価を付けている．また，新規ユーザによるサインアップ時の評価においても，提示されたアイテムのうち観たことがある（評価付けできる）ものは過去に観たいと思って観たものが多い．また，推薦結果においても，比較的多くのユーザに人気のあるアイテムが含まれやすい．これらのことから，ユーザの入力する評価値は，一般的にポジティブなスコアに偏りやすい [神嶌 07]．この偏りがどの程度あるかも，知っておく必要がある．

(4) 評価項目の次元 (Multi-criteria ratings)

多くのオープンデータセットは，ユーザのアイテムに対する評価値の次元は 1 次元である．しかし，アイテムへの評価値を複数の次元で付与する試みもある．レストランやホテル，家電製品の口コミサイトにおいては，アイテムへのレビューをいくつかの観点（例えばレストランなら，味，値段，雰囲気，定員の対応など）から行っている．Adomavicius らは，このような多次元の評価値を利用した推薦技術を提案している [Adomavicius 07]．また，Burke や Adomavicius らは，それぞれのサーベイ論文の中で，多次元の観点に対する評価値を，どのようにユーザプロファイルに反映させるべきかについて議論している [Burke 02, Adomavicius 05]．このような複数の次元で評価値を得ていれば，よりユーザのニーズに応じた推薦を行うことも可能である．

(5) 時間情報の有無 (Timestamp)

オープンデータセットの中には，タイムスタンプが記録されているものもある．このように評価値に時間情報が記録されていれば，時間経過に伴う推薦結果を評価することも可能である．

(6) データ規模 (Data size)

オープンデータセットの多くは，アイテムとユーザの行列で表現され，その要素にユーザの評価値が入る（2·4 節で詳細を説明する）．アイテムの数とユーザの数が大きくなれば，この行列の大きさも大きくなる．しかし，メモリベース方式の協調フィルタリングアルゴリズム [土方 07] を用いていた場合，その計算に時間がかかり，リアルタイムで利用できなくなる可能性がある（オフライン処理により軽減することは可能である）．また，データ規模が大きい場合，すべてのデータに対する厳密な評価（例えば，1 位から最下位まで並べた時の並びの妥当性）は意味をなさないこともある．データ規模によって，用いるアルゴリズムと評価方法を選択する必要がある．

(7) 記録密度 (Density)

上記行列の要素の全てに評価値が入力されていることは少ない．なぜなら，ユーザは全てのアイテムを消費して評価値を入力することはできないからである．行列の記録密度 (density) が低いことを sparsity と呼んでいる [Sarwar 00].

---

*4 書籍の評価サイトである The BookCrossing では，10 段階で評価付けさせている．http://www.bookcrossing.com

一般に，記録密度 (density または sparsity level) S_{level} は，

$$S_{level} = 1 - \frac{nonzero\_entries}{total\_entries}$$

で計算される [Sarwar 00]．$total\_entries$ は行列中の要素で，$nonzero\_entries$ はそのうち評価値が入力されている要素である．記録密度が低ければ，次元圧縮を行わなければ，良い推薦精度が得られないことがある．また，協調フィルタリングを適用する際，ユーザベース手法 [Resnick 94] よりもアイテムベース手法 [Sarwar 01] の方が良い結果をもたらすと言われている [Sarwar 01].

(8) データの増加 (Data increment)

一般に，アイテムとユーザの数は，時間の経過とともに変化する．この増加が頻繁に起こるかどうか，どの程度の頻度で増加分を推薦に反映させるかを考慮しておく必要がある．また，ユーザがどの程度，アイテムへの評価を増やしていくかも考慮しておく必要がある．特に，協調フィルタリングでオフライン処理を行う場合に検討が必要となる．協調フィルタリングでは，アイテム間またはユーザ間の類似度を計算する必要があるが，これをオフラインで計算しておくことで，推薦時の実行速度を上げることができる．実際の商用サービスでは，アイテムに比べるとユーザの方が大きく変動する可能性が高いため（新しいユーザが入ってくる可能性が高いうえ，新しいユーザが評価するアイテムが増える可能性も高い），アイテム間の類似度を事前に計算しておくと処理の高速化につながる [Sarwar 01]．データの増加に伴って，これを定期的に行っておく必要がある．

このようなデータ特性の差は，異なるアルゴリズムの実行結果に大きな影響を及ぼす．まず評価の前に，用いているアルゴリズムが適切かどうかを考える必要がある．また，時に研究者はアルゴリズムが有効であったか否かを，評価指標の絶対値から判断しようとすることがある．特に，正確性の評価指標の一つである適合率は，そのような判断に用いられやすい．しかし，データ特性によっては高い適合率を出すことが困難な場合がある．そのような場合には，例えば適合率 0.95 を有効性の判断の基準とすることは無理がある．絶対値は参考になるが，特定の値が有効性の判断の閾値として独り歩きするようなことは避けなければならない．評価の際には，データ特性とアプリケーションドメインを考慮して，評価指標の値を冷静に分析しなければならない．

2·4 データセット

§1 一般的なデータセット

オープンデータセットの多くは，ユーザのアイテムに対する評価値 (rating) の集合となっている．これは図 4 のように行列で表現できる．この図では 7 段階 (1-7) のリッカート尺度で入力されていることを想定している．0 は未評価であることを示している．データセットによっては，各評価値にそれが付与された日時（タイムスタンプ）とそのアイテムに対するタグ群が付与されている．また，アイテムにそのコンテンツに関する特徴量が付与されている場合もある（図 5 参照）．コンテンツの特徴量の表現形式は多様であるが，一般的には特徴に対して実数値（図では 0-1 に正規化されている）で表されることが多い．コンテンツに基づくフィルタリング [Adomavicius 05, Lops 11] を用いる場合には，この特徴量も用いる．オフライン評価では，図 4 の評価値行列のみを用いて評価を行う．

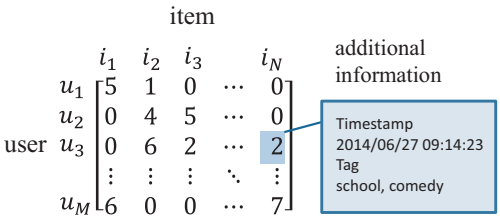

図 4 Standard data set

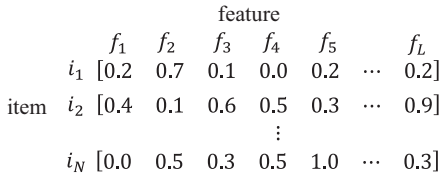

図 5 Feature data of contents

§2 交差検定

推薦アルゴリズムの多くは機械学習アルゴリズムを用いているものが多いため，評価の際にはデータセットを学習データ (training data または training set) とテストデータ (test data または test set) に分ける．学習データは，機械学習アルゴリズムにて判別用のモデルを構築するためのものであり，テストデータはモデルの性能を検証するためのものである．また，分けたテストデータの偶然の偏りを減少させるために，交差検定 (cross-validation) [Stone 74] が行われることが多い．一般には K-分割交差検定 (K-fold cross-validation) が用いられる．これはデータセット中の rating 群を K 個に分割し，そのうちの 1 つをテストデータとし，残る $K-1$ 個を学習データとする手法である．そして，テストデータに使う分割を順次入れ替えていく（図 6 参照）．そうやって得られた K 回の結果を平均して評価値を得る．

用いるアルゴリズムによっては，アルゴリズム内のハイパーパラメータを設定する必要がある．例えば，二値分類問題を解くときによく用いられるアルゴリズムとして SVM (Support Vector Machine) があるが，用いるカーネルによっては，いくつかのハイパーパラメータを設定しなければならない．その設定によって分類精度が変わっ

てくるわけであるが，そのパラメータの設定を行うには，一度テストデータで分類してみなくてはならない．分類精度の最も高かった時のパラメータを用いたいからである．しかし，テストデータをパラメータの設定に用いると，設定したパラメータを含めたアルゴリズムの性能を検証するためのデータセットがなくなってしまう．そこで，通常は K 個に分けた分割の一つを，このパラメータ検証用に用いる．これを確認用データ (validation data または development data または tuning data) と呼ぶ．

なお，タイムスタンプのついた評価値データに関して，学習データとテストデータの分割方法には 3 つの方法がある [Gunawardana 09]．一つは，ある時刻を設定し，その時刻より前にある評価値データ全て（評価値行列中のデータセット全体に適用）を学習データに，その時刻より後ろにある評価値データ全てをテストデータにする方法である．これは，推薦システムがその時間において推薦に用いる判別用のモデルを構築したことに相当する．もう一つは，ユーザごとに，学習データとテストデータを切り分ける時刻を設定する方法である．ここでの時刻設定には，ユーザの使用開始から N 個の評価値を入力するまでを学習データ，それ以降をテストデータとする方法や，ユーザの評価値のうちタイムスタンプの若い方から X% を学習データ，それ以降をテストデータとする方法などが考えられる．最後は，タイムスタンプを無視する方法である．これは，評価値行列中の評価値データをランダムに選択することができる．紹介した 3 つの方法のうち，最初の 2 つは時間と言う制約があるために，K-分割交差検定を行うことが難しくなる．

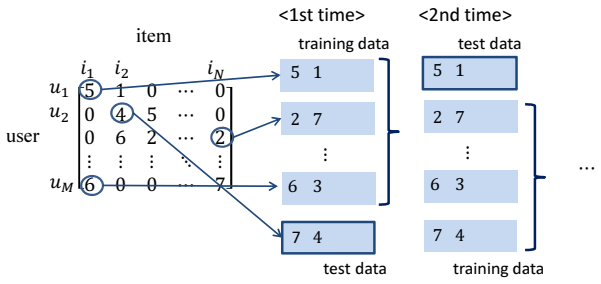

図 6 Cross validation for rating data

3. 正確性に関する評価指標

本稿で最初に紹介する評価指標 (evaluation metric) は，推薦の正確性 (accuracy) に関するものである．これは，システムがユーザに推薦を提示した時に，その推薦内容がユーザの興味や嗜好にどれだけ適合していたかを測るものである．ユーザへの有用性を考えた時に，ユーザが好むアイテムを提示することは必須と言える．推薦結果に対するユーザの総合評価（ユーザ満足度とも呼ばれる）は，推薦の正確性だけから決定されるわけではないが，良い推薦を行うためには高い正確性を持つことが前提であると言われている [Sinha 01]．また，これはユーザの推薦システムへの関与の深さとユーザ満足度との関係を調査した筆者の研究においても，この前提が確かめられている [Hijikata 12]．

ところで，正確性 (accuracy) と言う言葉は広義の意味で用いられる場合と，狭義の意味で用いられる場合がある．広義には推薦システムがどれだけユーザの興味や嗜好に適合したアイテムを推薦できるかや，推薦システムがユーザのアイテムに対する評価値をどれだけ正確に推定できるかを表す [Herlocker 04]．しかし，文献 [Olmo 08] にあるように，狭義にはシステムの判定（正判定も負判定も含む）のうち，どれだけが正解であったかを指すこともある（3·3·4 節で紹介）．この場合，対応する日本語としては「正解率」が適していると思われる．推薦システムの評価では，狭義の意味での accuracy を評価指標として用いることは少ないため，本稿では広義の意味で正確性 (accuracy) という言葉を用いる．

正確性を表す評価指標は数多く存在する．評価指標を選択するうえで考慮しなければならないのは，推薦アルゴリズムの予測能力を評価したいのか，ユーザに提示するアイテムの順序の妥当性を評価したいのか，ユーザに提示する上位スコアのアイテムの正確性を評価したいのかである．これらは，2·1 節で示したシステムのタスクとも深く関わってくる．また，過去の文献で用いていた手法と同じものかどうかや，その評価指標が評価したい性能の差を検出するのに十分な解像度を持っているかも考慮する必要がある．

筆者は，推薦の正確性を表す評価指標を，(1) 予測評価値の正確性 (accuracy of estimated rating)，(2) 予測順序の正確性 (accuracy of estimated ranking)，(3) 推薦リスト内の分類の正確性 (accuracy of list relevance)，(4) 推薦順位に基づく正確性 (accuracy based on ranking position) に分ける （図 7 参照）．以下，これらを順に説明する．

3·1 予測評価値の正確性 (Accuracy of estimated rating)

システムの予測評価値の正確性を評価する指標について説明する．図 8 に示すように，これはテストデータ中の正解データ (ground truth data または gold standard) となる評価値と，システムが予測した評価値との差を計測する指標である．

§1 MAE

予測評価値の正確性を測る最も代表的な指標は，平均絶対誤差 (MAE (Mean Absolute Error)) である．MAE は，推薦システムの正確性に関する評価における，最も基本的な指標となっている．テストデータ中のアイテム集合を B，その中から選択した 1 つのアイテムを b，システムのそのアイテムへの予測評価値を $p(b)$，テストデータ中の正解データであるユーザのアイテムへの評価値を

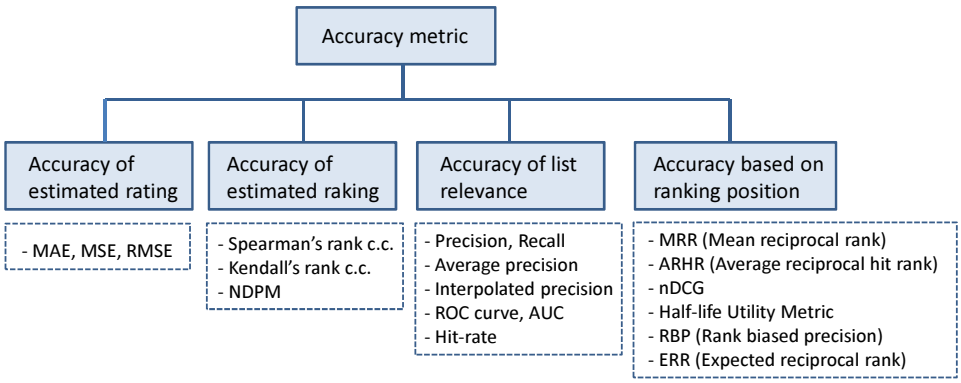

図 7 Categorization of accuracy metrics

$r(b)$ とすると，MAE は以下の式で計算される．

$$MAE = \frac{\sum_{b \in B} |r(b) - p(b)|}{|B|}$$

上記の式は評価値の入力のスケールによって値が大きく変わってくる．そこで，入力されうる評価値の最小値 r_{min} と最大値 r_{max} を用いて，下記のように正規化したものが用いられることもある．

$$normalizedMAE = \frac{MAE}{r_{max} - r_{min}}$$

MAE は，テストデータ中の全てのアイテムへの予測評価値の正確性を測定しており，推薦システムの全体的な予測能力を評価するのに適している．しかし，システムのタスクがランキングリスト提示である場合，ユーザに提示する上位 N 個以外のアイテムへの評価がほとんどを占める MAE は，必ずしも適切な評価指標とは言えない．また，システムのタスクがフィルタリングである場合も，ユーザが気にするのは，提示されたアイテムを消費すべきか無視すべきかの二択となる．この場合も，予測評価値と正解の評価値の差を厳密に計算する MAE は，オーバースペックであると言える．また，MAE では，例えば 5 段階評価の時に，ユーザの評価が 1 の時に 2 と評価してしまう場合と，ユーザの評価が 2 の時に 3 と評価してしまう場合の差は考慮しない．後者の場合，ニュートラルな予測なので，もしかするとユーザはそのアイテムを選択してしまう可能性がある．

§2 MSE と RMSE

MAE から派生した評価指標に，平均二乗誤差 (MSE (Mean Square Error)) と二乗平均平方根誤差 (RMSE (Root Mean Square Error)) がある．MSE と RMSE は下記の式で計算される．

$$MSE = \frac{\sum_{b \in B} |r(b) - p(b)|^2}{|B|}$$

$$RMSE = \sqrt{\frac{\sum_{b \in B} |r(b) - p(b)|^2}{|B|}}$$

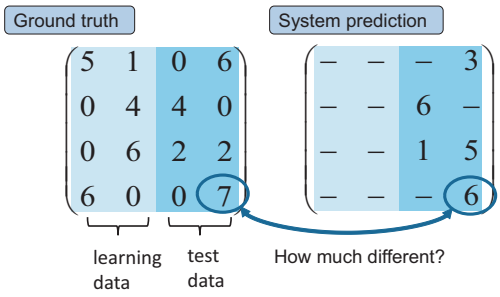

図 8 Evaluation on accuracy of estimated rating value

MAE 同様，上記の式は評価値の入力のスケールによって値が大きく変わってくる．そこで，入力されうる評価値の最小値 r_{min} と最大値 r_{max} を用いて，下記のように正規化したものが用いられることもある．

$$normalizedMSE = \frac{MSE}{r_{max} - r_{min}}$$

$$normalizedRMSE = \frac{RMSE}{r_{max} - r_{min}}$$

ユーザはアイテムへの予測評価値を見た時に，自分の直感的な評価値との間に微小な差があったとしても，気にしないかもしれない．2·2 節で示したように，もともとユーザがアイテムに付与する評価値は，安定したものではない．そのため，そのような微小な差には気づかないと思われる．しかし，自分の評価値と大きくかけ離れた予測評価値が付与されていた場合，推薦システムへの信頼は大きく下がるかもしれない．そこで，より大きな誤差を重要視し，小さな誤差は重要視しない評価指標として用いられるのが MSE と RMSE である．両指標とも，差の 2 乗の総和を取る．最終的に出力される値が，ユーザの与えた評価値と同じ単位になる RMSE の方が，値の解釈が容易になるため良く使われている．

この節では，予測評価値の正確性の評価指標を紹介してきた．これらの評価指標は推薦システムが持つ基本的な予測能力を評価するのには適しており，事実最も良く用いられている．しかし，ユーザがその微小な予測能力の差を重視するかと言えば，それはそうとは言えない．あ

る一つの（重要な）アイテムが提示されることに比べれば，アイテム全体に対する予測能力は，ユーザにとってはそれほど意味がないとも言われている [Lam 06].

3·2 予測順序の正確性 (Accuracy of estimated ranking)

推薦システムはアイテムに対して予測評価値を付与し，予測評価値が高い順序でユーザにアイテムを提示する．この提示においては，最もユーザが好むものが上位で提示され，最もユーザが好まないものが下位で提示されて欲しい．そこで，この予測順序の正確性を評価するための指標を紹介する．

§1 スピアマンの順位相関係数

スピアマンの順位相関係数 (Spearman's rank correlation coefficient) は，あるアイテムの集合に対して別々に得られたランキングの順位の間に，相関があるかどうかを測る指標である．2 つの変数（実数）の間に相関があるかどうかを測る指標にはピアソン相関（ピアソンの積率相関係数 (Pearson product-moment correlation coefficient)）がある．これは，偏差の正規分布を仮定するパラメトリック (parametric) な方法で，変数の線形関係を計測している．一方，スピアマンの順位相関係数は，このような仮定をおかないノンパラメトリック (non parametric) な方法である．

例えば，10 個のアイテムに対するユーザの実際の評価値が，$1.0, 2.0, 3.0, \cdots, 9.0, 10.0$ と均等な幅で付けられていたとする．ピアソン相関を 1 にするためには，推薦システムの予測評価値も，上記と全く同じようにしなければならない．これは，ピアソン相関が変数間の線形関係を前提としているからである．しかし，システムの予測評価値が，$1.0, 1.5, 2.0, 2.5, 3.0, 8.0, 8.5, 9.0, 9.5, 10.0$ であったとしても，両者の順位は変わらない．すなわち順位の相関は，上記のような値の線形関係を前提としていないのである．

スピアマンの相関係数 $r_{Spearman}$（一般には ρ が使われることが多い）の算出式は，下記のピアソン相関 $r_{Pearson}$ の算出式において（n はデータ数，x と y はデータの観測値），

$$r_{Pearson} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

x と y に観測値ではなく，順位を入れることによって求められる．x と y を順位とすることで，$\sum x_i = n(n+1)/2$, $\sum x_i^2 = n(n+1)(2n+1)/6$, $\bar{x} = (n+1)/2$ という置き換えができるため，これを用いて式を整理すると，スピアマンの相関係数の算出式は，

$$r_{Spearman} = 1 - \frac{6}{n(n^2-1)}\sum_{i=1}^{n}(x_i - y_i)^2$$

となる．推薦システムの評価においては，x が推薦システムが出力した各アイテムの順位，y がユーザの評価値に基づく各アイテムの順位，n がテストデータセット（または推薦リスト）中のアイテムの数となる．

§2 ケンドールの順位相関係数

ケンドールの順位相関係数 (Kendall's rank correlation coefficient) も，あるアイテムの集合 B ($|B| = n$) に対して別々に得られたランキングの順位の間に，相関があるかどうかを測る指標である．ケンドールの順位相関係数では，B 中のアイテムの全ての組み合わせ（アイテムのペアで $n(n-1)/2$ 個）を取り出して計算する．ペアの片方をアイテム a，もう片方を b としたとき，これに対してユーザ s，t の二人がどのような順位を付けたかを得る．ユーザ i がアイテム j に付与した順位を返す関数を $Rank_i(j)$ としたとき，

IF $Rank_s(a) < Rank_s(b)$ AND $Rank_t(a) < Rank_t(b)$ THEN $P \leftarrow P + 1$
IF $Rank_s(a) > Rank_s(b)$ AND $Rank_t(a) > Rank_t(b)$ THEN $P \leftarrow P + 1$
OTHERWISE $Q \leftarrow Q + 1$

とする．このとき，ケンドールの順位相関係数 $r_{Kendall}$（一般には τ が使われることが多い）は，

$$r_{Kendall} = \frac{P - Q}{\frac{1}{2}n(n-1)} = \frac{2P}{\frac{1}{2}n(n-1)} - 1$$

となる．このように，ケンドールの順位相関係数では，二人のユーザの，2 つのアイテムに対するランクの大小関係の一致度を見る．推薦システムの評価においては，s が推薦システム，t が対象ユーザとなる．推薦システムと対象ユーザの好みの順序が全く同じ場合は $Q = 0$ となり $r_{Kendall} = 1$，全く逆順なら $P = 0$ となり $r_{Kendall} = -1$ となる．

§3 NDPM

NDPM (Normalized Distance-based Performance Measure) は 2 つのランキングの間で，どれだけアイテムの順位に矛盾があるかを測る指標である [Yao 95]．テストデータ中の任意の 2 つのアイテムを取り出したとき，システムの予測した順序とユーザの評価順序とが異なれば，C^- を 1 増やす．また，システムの予測した順序がどちらかのアイテムを高い順位にしているが，ユーザの評価では同じ順位にしていた場合には C^u を 1 増やす．このようにして，全てのアイテムペアを取り出したのち，以下の式で $NDPM$ を計算する．ただし，C^i は，アイテムセット中でユーザが評価付けしたアイテムで，評価値に差があるアイテムペアの総数を表す．

$$NDPM = \frac{2C^- + C^u}{2C^i}$$

考え方は，ケンドールの順位相関係数に近い．ケンドールの順位相関係数は，システムとユーザで順序が一致した場合にスコアに加算し，そうでない場合に減算するが，NDPM は間違った場合のみ加算する違いがある．また，ユーザの評価順位が同じものを，システムが差をつけて

予測していた場合は，加算の重みを少なくしている特徴がある．このような場合を考慮しているのは，ユーザはそれほど高い解像度でアイテムの評価を行うことができないからである．そのため，同一順位となるアイテムが多く出現するからである．NDPM は，推薦システムにおけるユーザの評価行動をよく考慮した指標であると言える．しかし，NDPM は（ケンドールの順位相関係数も同様であるが），ランキングの上位での矛盾も，ランキングの下位での矛盾も等価に扱ってしまう欠点がある．

§4 順位相関係数の利用

この節では，予測順序の正確性に関する指標をいくつか提案してきた．ここではさらに，これらの指標の利用について考察する．推薦システムのタスクがランキングリスト提示では，推薦結果は順位付きのリストで提示される．そのため，これらの評価指標はよりユーザの利便性を考慮した指標であると言える．しかし，これらの順位相関係数は非常に基本的な統計的手法にもかかわらず，推薦システムの評価にはあまり用いられていない．筆者はこの理由を次のように考えている．この評価指標は，推薦リスト側とユーザ側の，完全なランキングリストを必要とする．しかし，ユーザにとって，テストデータ全体に対して，このような完全なランキングを付与することは極めて難しいと言える．アイテムへの評価値を用いる手はあるが，5 段階や 7 段階程度では細かいランキングまで表現することはできない．逆に，100 段階やより詳細な連続値で入力させるとしても，ユーザにそこまで評価の解像度があるとは限らない．そのため，評価への適用が難しいのだと思われる．

なお，順位相関係数は，テストデータのアイテム全体に対する予測順序の評価と推薦リスト内の予測順序の評価の両方に用いることができる．ただし，いずれの場合においても，オフライン評価では前もって全てのアイテムに対するユーザの評価順位を得ておく必要がある．上記で説明したように，これは難しいことである．順位相関係数は，どちらかと言えばオンライン評価（少数の提示されたアイテムに対してユーザに順序付けを行わせる実験）の方が適していると言える．

3·3 推薦リスト内の分類の正確性 (Accuracy of list relevance)

推薦システムでは通常，予測評価値が計算されたアイテムを，その大きさの順でユーザに提示する．これは推薦システムに限らず，情報検索においても行われている．これは，情報検索の分野における Probability Rank Principle (PRP) [Robertson 97] という考えに基づいている．文献 [Robertson 97] では,「クエリに対するシステムの応答を，クエリに対する適合性が低くなる順で提示するようにすれば，システムの有用性を最大化することができる」と述べている．既存の推薦システムと情報検索システムのほとんどがこの考え方に従っている．

しかし，推薦システムがユーザに推薦を行う時には，システムに登録されているアイテムの全てを順序付きで提示することはしない．ユーザが確認できるアイテムの数はたかが知れているので，上位 N 件を提示することが多い．その N 件のアイテムのリストを推薦リストと呼ぶ．推薦リストは，N を固定の値として獲得することもあれば，推薦アルゴリズム内で計算した予測評価値に基づき，それが θ 以上のアイテム全てを得ることで獲得することもある．この場合，推薦リストは可変長となる．本節以降では，その推薦リストを評価する指標を紹介する．特に，本節では推薦リスト内に含まれるアイテムが，ユーザの興味に一致しているのか否かという分類に関する評価について述べる．

§1 適合率と再現率 (Precision and Recall)

推薦リストを評価するために最も用いられる手法は，適合率（または精度）(precision) と再現率 (recall) である．これらは，情報検索の分野や機械学習の分野で古くから用いられているもので，基本的にはあるオブジェクトを true か false かに分ける分類問題を対象としている．そのため，推薦システムにおいてオープンデータセットを使用する場合で，なおかつユーザのアイテムに対する評価値が N 段階（例えば 7 段階）で付与されている場合には，適合率と再現率を計算する際に図 9 のような工夫を行う必要がある．この例では，ユーザはアイテムに 7 段階で評価しているものとする．0 は未評価を表す．この時，評価値において 1-4 を「好きでない (Dislike)」，5-7 を「好き (Like)」のようにバイナリ値に変換する．(図ではそれぞれ “D”，“L” という記号を付けている)．このようにすることで，評価値の予測問題から分類問題に変換することができる．

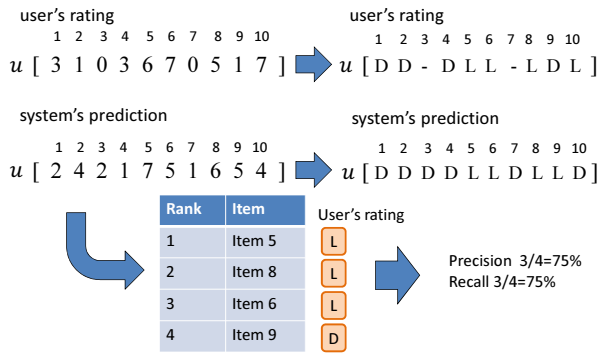

図 9 Calculation method of precision and recall

ユーザ a_i に提示された推薦リスト L_i を評価する．テストデータ中のユーザ a_i の好きなアイテム集合を T_i，$\Im$ を推薦リストから推薦リスト中のアイテム集合への写像と定義した時，具体的な，適合率 $Precision$ と再現率

$Recall$ の計算は以下のようになる．

$$Precision = \frac{|T_i \cap \Im L_i|}{|\Im L_i|}$$

$$Recall = \frac{|T_i \cap \Im L_i|}{|T_i|}$$

図 9 の例では，システムの予測評価値が 5 以上のものを推薦するとした時に，アイテム 5, 6, 8, 9 が予測評価値の順に，推薦リスト L_i に入れられる．ユーザ a_i の好きなアイテムは，アイテム 5, 6, 8, 10 であるため，適合率が 75.0%，再現率が 75.0%となっている．

先の例では，システムがユーザに提示するアイテムを，予測評価値が 5 以上のものとしたが，ここで 4 以上のものと変更してみる．そうすると，推薦リスト L_i に入るアイテムは，アイテム 2, 5, 6, 8, 9, 10 である．この時，適合率は 66.7%，再現率が 100.0%となる．このように，推薦リストの長さを変えると，適合率と再現率は変化する．

一般に，推薦リストの長さを長くすると適合率は下がり，再現率は上がる．適合率と再現率はトレードオフの関係がある．そのため，評価においては推薦リストの長さをいくつか変化させて，その適合率と再現率の変化を見ることが多い（図 10 に例を示す）．これを適合率-再現率曲線 (precision-recall curve) という．このグラフは，必要な再現率を達成したい時にどの程度の適合率が得られるのかを調べたい時や，適合率が急激に悪化する直前で閾値を設定したい時などに便利である．また，ユーザに提示する推薦リストの長さが決まっている場合や変化させる長さの単位が決まっている場合は，5 位まで表示した時の適合率や，10 位まで表示した時の適合率などを，$Prec@5$，$Prec@10$ のように記載し，その値を調べることも多い．

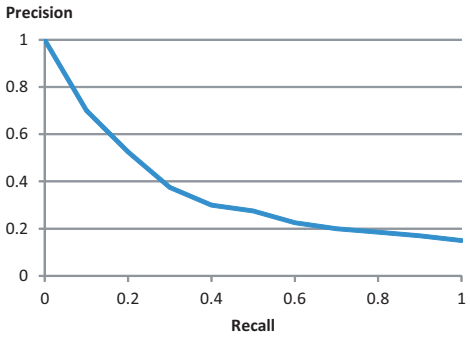

図 10 Precision - recall curve

また，このように評価する指標が 2 つ存在すると，アプリケーションやドメインによっては，そのどちらを重視するのか決めかねることもある．そこで，その両方の値を考慮した指標である F 値 (F-measure) が用いられることも多い．これは，適合率と再現率の調和平均で，下記のように計算される．

$$F\text{-}measure = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

適合率は，オンライン評価でも用いることができるので非常に有用な評価指標である．リアルタイムでユーザに推薦リストを提示し，そのすべてのアイテムに評価付けしてもらえれば，その場で推薦リストの評価ができる．一方，再現率は推薦システムの評価に用いることは難しい．オープンデータセットには，実際のユーザのアイテムへの評価値がついているが，それらのユーザは，データセット中の全てのアイテムを消費（閲覧）して評価値を付けたわけではないからである．したがって，再現率を求める時には，非常に限られたデータセットを用いることになる．また，この情報検索の分野では，同様の問題に対してプーリング法 (pooling method) [Büttcher 07, Baeza-Yates 11] により解決しているが，適合基準が主観的である推薦システムの分野においては，対象ユーザにプールされた多くのアイテムの評価を行わせることは現実的ではない．

§2 平均適合率 (Average Precision)

前節で示したように，提示する推薦リストの長さにより適合率は変化してしまう．そこで，非常に短い推薦リストから，最大でテストデータ中全てのアイテムを含む推薦リストまで変化させた時に，得られる適合率の平均を求め，それを推薦の正確性と捉える方法がよく用いられる．

その代表が平均適合率 (Average Precision) AP である [Buckley 05]．これは，推薦リストを作成した時に，最も低い順位で適合したアイテムまでを下記の式で評価するものである．

$$AP = \frac{1}{M} \sum_{1 \leq k \leq N} rel(k) \cdot Prec@k$$

で表される．ここで，推薦リストの長さを N，推薦リスト中の適合アイテム数を M，k 位に出現したアイテムがユーザに適合するかどうか (0 または 1) を返す関数を $rel(k)$，k 位までの精度を $Prec@k$ としている．すなわち，適合アイテムが出てきたときのみ，その順位までの精度を加算し，適合アイテム数で割った値となっている．

なお，情報検索の分野では，課題セット中の課題（クエリ）ごとに平均適合率を求め，その平均を求めた MAP (Mean Average Precision) を用いることが多い．MAP は下記の式で算出される（n は課題セット中の課題数である）．

$$MAP = \frac{1}{n} \sum_{n} AP_n$$

推薦システムの分野では，長期の興味や嗜好を表すユーザプロファイルに基づいて推薦されるため，検索課題に相当する概念がない．そのため，複数のユーザに対して AP を求め，その平均を求めることになる．

§3 補間適合率 (Interpolated precision)

推薦リストの長さを変化させてその適合率を見ていくと，特にリストが短い時には，次のランクが適合するかどうかによって，適合率の値が大きく変わってしまう．図 11 に例を示すが，この推薦リストでは，1 位は適合しているが，2 位は不適合である．この場合，適合率は 1.0 から 0.5 に落ちてしまう（この現象は，適合率-再現率曲線を描くときに顕著になることが多い）．一方，再現率は推薦リストが長くなればなるほど，単調増加する．そこで，このような適合率の落ち込みを無視し，同じ再現率を得た時に最も適合率が良くなる値のみを取り出したものを補間適合率 (interpolated precision) と言う [Manning 08]．これを用いれば，再現率 0.25 までの補完適合率は 1.0，再現率 0.5 まで補完適合率は 0.67 となる．

Rank	1	2	3	4	5	6	7	8	9	10
Fit?	Y	N	Y	N	Y	Y	N	N	N	N
Prec.	1.0	0.5	0.67	0.5	0.6	0.67	0.57	0.50	0.44	0.40
Recall	0.25	0.25	0.5	0.5	0.75	1.0	1.0	1.0	1.0	1.0

Recall	- 0.25	-0.5	-0.75	-1.0
Interpolated Precision	1.0	0.67	0.6	0.67

図 11 An example of calculating interpolated precision

図 11 の例は，非常に小規模なデータセットに適用した例であるが，もっと大きなデータセットに適用すれば，細かい再現率で，その時の補間適合率を算出することができる．これを全ての区間の再現率を考慮して一つの指標として表したのが，n 点補間適合率 (n-points interpolated average precision) である．n 点補間適合率 $n\text{-}IntPrec$ は，n 点の再現率を取り，その時の補完適合率 $IntPrec_i$ を計算し，その平均を求めたものである．具体的には，以下の式で計算される．

$$n\text{-}IntPrec = \frac{1}{n} \sum_{1 \leq i \leq n} IntPrec\_i$$

一般には，n には 11 を用いる（再現率 0.0, 0.1, ..., 1.0）．

§4 ROC 曲線 (ROC curve)

機械学習の分野で部類性能を示す評価技法として，ROC 曲線 (ROC curve) が使われている．これは，機械学習アルゴリズムが算出した予測スコアに基づいて事例を順序付けで並べた時に，その予測スコアの閾値を変化させて，正判定と負判定を分けた時に，どの程度の分類性能が得られるかをグラフで示すものである．推薦システムも，アイテムへの予測スコアを基にアイテムを順序付けてユーザに提示するため，これを利用可能である．

ROC 曲線の描画方法を図 12 に示す例を使って説明する．図の左にある表はテストセットのデータである．列の意味は左から順に，推薦リストにしたときのアイテムの順位，予測スコア，ユーザが付けた「好き (Like) 」か「好きでない (Dislike) 」の正解データである．システムは，予測スコアをある閾値で切って，推薦するアイテムを決定する．この閾値を，高いものから順にずらしていき，その時の分類の正確性を測る．

ROC 曲線を使うに当たり重要な概念がある．それは，TP (True Positive) , FP (Flase Positive) , FN (False Negative) , TN (True Negative) の 4 つである．これは図 12 の右上の表で定義される．行は，システムが Positive（「好き」と予測）と判定したか，Negative（「好きでない」と予測）と判定したかを表す．列は，ユーザが Like（「好き」）と評価したか，Dislike（「好きでない」）と評価したかを表す．TP は，システムが Positive（「好き」）と判定して，実際も Like（「好き」）であることを示す．FN は，システムが Negative（「好きでない」）と判定して，実際は Like（「好き」）であることを示す．後ろのアルファベットがシステムの判定で，前のアルファベットがそれが正しかったかどうかを表す．

機械学習の分野では，システムの正判定だけでなく，負判定も含めた正確性を評価することがある．これを正解率 (accuracy または Rand Accuracy または Rand Index) と呼ぶ（狭義の意味での accuracy である）．正解率 $accuracy$ は，上記の定義を用いると以下の式で計算できる．

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

システムの全ての判定のうち，正しく正/負を判定できていたものの割合を意味する．

また，機械学習及び情報検索の分野では，実際には Dislike であるアイテムのうち，システムが Positive と判定したものの割合を求めることもある．これを Fall-out と呼び，以下の式で算出される．

$$Fall\text{-}out = \frac{FP}{FP + TN}$$

Test set

Rank	Score	L/D
1	0.95	L
2	0.85	L
3	0.80	D
4	0.75	L
5	0.70	D
6	0.65	L
7	0.55	D
8	0.45	L
9	0.30	D
10	0.30	D

System \ User	Like	Dislike
Positive	TP	FP
Negative	FN	TN

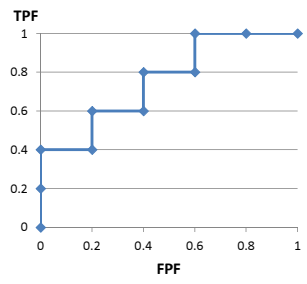

図 12 An example of drawing ROC curve

ROC 曲線では，真陽性率 TPR (True Positive Rate) と偽

陽性率 FPR (False Positive Rate) を下記の式で計算する．

$$TPR = \frac{TP}{TP+FN}$$

$$FPR = \frac{FP}{FP+TN}$$

要するに，TPR はユーザの「好き」の評価のうちどれだけシステムが「好き」と判定したかを示し，FPR はユーザの「嫌い」の評価のうちどれだけシステムが「好き」と判定してしまったかを示している．ROC 曲線は，推薦スコアの閾値を変化させながら，その時の TPR と FPR を計算し，TPR を縦軸に FPR を横軸にとったグラフである．図の例にあるテストセットからは，図右下のグラフが描ける．このグラフにおいて，最良の場合は座標 (0, 1) を通る場合である．複数のアルゴリズムの結果を比較する際には，左上に膨らんだグラフほど優れていることになる．これを一つの数値として表すため，ROC 曲線の下側にある領域の面積を求めることもある．これを，AUC (Area Under Curve) と呼ぶ．

ROC 曲線の利点としては，縦軸がユーザの好きなアイテムを推薦できた割合，横軸がユーザの嫌いなアイテムを推薦してしまった割合を表しており，ユーザは再現率重視で結果を確認できる点にある．特に，ユーザの嫌いなアイテムを推薦してしまうことが問題となるドメインでは有効と言える．

最後に，複数ユーザのテストデータを利用して ROC curve を作成する方法について説明する．Schein らは，テストデータ中の全てのユーザ-アイテムのペアに対して予測評価値を付与し，それらを予測評価値の大きい順に 1 つのリストにして，ROC curve を描く方法が試している．彼らは，これを globalROC (GROC) curve と呼んでいる [Schein 02]．また，Sarwar らはユーザごとに N の長さを持つ推薦リストを作成し，ユーザごとに TPR と FPR を計算し，その平均を取ってから ROC curve を描いている [Sarwar 00]．これを Schein らは，Customer ROC (CROC) curve と呼んでいる [Schein 02]．なお，これらの手法は，適合度-再現率曲線を描くときにも利用できる．

§5 ヒット率 (Hit-rate)

適合率と再現率は，対象ユーザの推薦リストに対する正確性を評価する指標であったが，推薦システムを利用する全てのユーザに対して，どれほど適合するアイテムがあったのかを測る指標として，ヒット率 (hit-rate) がある [Deshpande 04]．これは購買履歴や閲覧履歴のように，ユーザが購入（閲覧）したか否かを，評価値行列として持つデータセットを対象としている（すなわち評価値が ‘unary’ であるもの）．評価値行列において，テストデータ中の全ての非零要素 (non-zero entry) を検査する．各非零要素には対応するユーザとアイテムがあるが，そのアイテムがそのユーザの推薦リストに含まれたかどうかをチェックする（図 13 参照）．このチェックを全ての非零要素に対して行った時に，推薦リストに含まれた数を num_{hits} とする．n を全ユーザ数とすると，ヒット率 $hit\text{-}rate$ は，以下の式で算出される．

$$hit\text{-}rate = \frac{num_{hits}}{n}$$

直感的には，テストデータ中の購買（閲覧）アイテムのユーザ一人当たりの適合数と考えることができる．この指標は購買データの他にテレビ番組の視聴データを用いた推薦の評価にも用いられている [O’Sullivan 04].

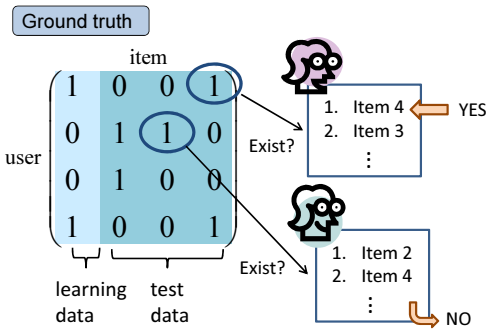

図 13 Calculation method of hit-rate

3·4 推薦順位に基づく評価指標 (Accuracy based on ranking position)

この節では，推薦リストにおいてリスト中の順位により重みをつけて正確性を評価する指標を紹介する．すなわち順位が高いほど，正確性の評価スコアに高い重みをつける方法である．このようなモデルを一般に，順位依存モデル (position-based model) と呼ぶ [Craswell 08, Richardson 07]．このモデルは，以下の数式で表される [Chapelle 09].

$$P(C=1|i,l) = atr_i \cdot p_l$$

i はアイテム，l は順位，C はそのアイテムを閲覧することを表す事象（アイテムを閲覧した場合は 1，閲覧しなかった場合は 0），atr_i はアイテム i がユーザにとってどれほど魅力的かを表す値，p_l は l 位までユーザが推薦リストを閲覧する確率である．すなわち，このモデルは推薦リスト中のある順位にあるアイテムをユーザが閲覧する確率を表している．このモデルに当てはまる指標には，MRR, ARHR，DCG，nDCG，HLU (Half-life Utility Metric)，RBP がある．また，順位依存モデルから派生したものとして，ユーザが適合するアイテムを獲得した時点で推薦リストの閲覧を止めることを考慮したカスケード依存モデル (cascade-based model) もある．このモデルに当てはまる指標には，ERR がある．上記の指標について，それぞれ順に説明する．

§1 平均逆順位 (MRR)

推薦結果の正確性の評価において，順位を考慮する最も簡単な手法は，平均逆順位 (MRR: Mean Reciprocal Rank) である．これは，正解（ユーザの興味に適合するアイテ

ム）が現れた順位の逆数 (reciprocal) を求め，その平均を取ったものである．具体的には，以下の式で計算される．

$$MRR = \frac{1}{|D_{rel}|}\sum_{i=1}^{N}\frac{rel_i}{i}$$

ここで，D_{rel} は，推薦リスト中のアイテムでユーザが好むものの集合，N は推薦リストの長さ，i は推薦リスト中の順位，rel_i は順位 i のアイテムをユーザが好むか否かを表すバイナリ値である．順位 i で割ることにより，順位による重みを反映させている．

§2 ARHR (Average Reciprocal Hit-Rank)

3·3·5 節で示したヒット率は，推薦リスト中の順位を考慮していなかった．これを MRR のように順位の逆数を用いて計算したものに ARHR (Average Reciprocal Hit-Rank) [Deshpande 04] がある．ヒット率と同様，対象となるデータセットは，評価値が ‘unary’ の評価値行列である．評価値行列において，テストデータの非零要素を対象として，以下の式により ARHR を求める．

$$ARHR = \frac{1}{n}\sum_{i\in Z_L}\frac{1}{pos(i)}$$

ここで，テストデータ中の非零要素の集合を Z，そのうち対応するユーザの推薦リストに含まれたものの集合を Z_L とする．また，$pos(i)$ を指定された非零要素 i に対応するアイテムの，その非零要素 i に対応するユーザの推薦リスト中での位置を表す関数とする．n は全ユーザ数である．ヒット率と同様，直感的には，テストデータ中の購買（閲覧）アイテムのユーザ一人当たりの適合数と考えることができるが，適合数が順位の逆数の総和になっている点が異なる．

§3 nDCG (n-Discounted Cumulative Gain)

順位を考慮して提示リストを評価する最も代表的な評価指標として nDCG (Normalized Discounted Cumulative Gain) がある [Järvelin 02]．情報検索の分野でよく利用される評価指標である．順位付きリストを出力する研究分野では任意の分野で利用できるため，推薦システムの評価にも利用できる．最も用いられている評価指標は nDCG であるが，その説明にはその基本形である CG (Cumulative Gain) と DCG (Discounted Cumulative Gain) を説明する必要があるので，まずはこれらの説明を行う．

CG は，推薦リストが得られた時に，リスト中のアイテムについて，ユーザの評価値の総和をとったものである．各アイテムの評価値が高いほど値が高くなる．リスト長を N，順位 i のアイテムに対する実際のユーザの評価値（N 段階 ($N \geq 3$) の評価値）を rel_i とすると，下記の式で計算される．

$$CG = \sum_{i=1}^{N} rel_i$$

しかし，CG では同じ評価値を持つアイテムなら，順位が高いものに対しても，順位が低いものに対しても，同じだけのスコアを CG に与えてしまう．高い評価値を持つアイテムが，低い順位にランク付けされれば，ペナルティを与えたくなるであろう．そのような順位による重みづけを考慮した指標が DCG (Discounted Cumulative Gain) である．これは以下の式のいずれかで計算される．

$$DCG = rel_1 + \sum_{i=2}^{N}\frac{rel_i}{log_2(i)}$$

$$DCG = \sum_{i=1}^{N}\frac{2^{rel_i}-1}{log_2(i+1)}$$

DCG は，順位による重みは反映されているものの，推薦リストが長くなるほど値が大きくなる．そこで，推薦リストの長さによらず，アルゴリズム間の比較ができるようにする評価指標が nDCG である．これは，DCG を基本とするが，アイテムを理想の順番で並べた時の DCG も計算する．これを IDCG (Ideal DCG) と呼ぶ．理想の状態とは，アイテムがユーザが付与した評価値の高い順に推薦リストに含まれていることを指す．これとの比を求めることで，どれだけ理想の状態からかけ離れているかを表す指標である．具体的には，以下の式で計算される．

$$nDCG = \frac{DCG}{IDCG}$$

§4 Half-life Utility Metric

nDCG によく似た指標で，推薦システムの分野において提案されたものに，Half-life Utility Metric (HLU) がある [Breese 98, Shani 08]．HLU も DCG も，良いアイテムが推薦リストの下位に提示されても意味がないと考える指標である．nDCG との違いは，ユーザのデフォルトの評価値（ニュートラルの評価値や少し負によった評価値）を考慮する点と，半減速度が異なる点である．$rel_{u,i}$ をユーザ u のアイテム i に対する評価値，d をデフォルトの評価値，N をリスト長，α を半減係数とすると，ユーザ u の HLU である HLU_u は下記の式で算出される．

$$HLU_u = \sum_{i}^{N}\frac{max(rel_{u,i}-d,0)}{2^{(i-1)/(\alpha-1)}}$$

また，ユーザ u の評価値の順にアイテムを並べた理想のリストを作成した時の HLU を $IHLU_u$ とすると，HLU 比 $rHLU_u$ は，

$$rHLU_u = \frac{HLU_u}{IHLU_u}$$

となる．rHLU が nDCG に相当する指標と言える．

§5 RBP (Rank Biased Precision)

DCG や Half-life Utility Metric によく似た指標に RBP (Rank Biased Precision) がある [Moffat 08]．DCG と HLU においては，推薦リストの下位におけるアイテムの正確

性を，指標算出の際に重みを小さくしていた．しかし，この重みの決定を指数関数や対数関数など，既存の数学関数に依存していた．推薦リストの下位の重みを下げるという考えは，ユーザは下位に行くほど，そのアイテムの適合性に疑問を感じるようになる可能性があることと，そもそも疲れて推薦リストの閲覧を止めてしまう可能性があるからである．これらの仮定に基づき，ユーザの閲覧行動をモデル化したものが RBP である（図 14 参照）．RBP は以下の式で算出される．

$$RBP = (1-p)\sum_{i=1}^{N} rel_i \cdot p^{i-1}$$

ここで，N は推薦リスト長，rel_i は順位 i で提示されたアイテムに対するユーザの評価値，p はユーザが推薦リストを閲覧中にどれだけ固執して見続けるかを表す値である．RBP の特徴は，辛抱強いユーザ（例えば $p = 0.95$）や辛抱強くないユーザ（例えば $p = 0.5$）などを仮定して，評価できる点である．ユーザは，推薦リストの上位から閲覧していく．確率 p で次の順位のアイテムを確認する．n 位まで閲覧しそこで閲覧を止めた時に，その推薦リストの正確性がどれだけあったかを表している．

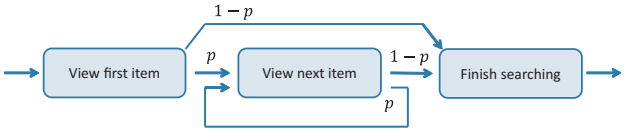

図 14 User browsing model in RBP

なお，実際にユーザがリスト（多くの研究では検索エンジンの検索結果）において，ランキングのどこまでチェックするかについては，多くの調査がなされている [Joachims 05, Järvelin 02, Hosanagar 05]．例えば，検索結果に対するユーザの閲覧行動を視線計測で観察した Joachims らの研究では，ユーザは検索結果の上位 3 件までしか見ないことが多いことを報告している [Joachims 05]．

§6 ERR (Expected Reciprocal Rank)

DCG，nDCG，half-life utility，RBP では，ユーザは推薦リストの下位になればなるほど，推薦結果を見なくなることをモデル化したものである．しかし，このモデルには一つ欠点がある．ユーザは推薦リストを閲覧中に，一度自分の好みに合ったアイテムを見つけてしまうと，それより下位に推薦されたアイテムを見なくなってしまうことがある．これらのモデルでは，ユーザが下位のアイテムまでチェックするかどうかを，単調減衰関数で表現しているが，上記のような場合には非連続的に 0 にしなければならない．このようなユーザの閲覧中止の行動を組み込んだモデルを，順位依存モデルの中でも，カスケード依存モデル (cascade-based model) またはカスケードユーザモデル (cascade user model) と呼んでいる [Chapelle 09]．カスケード依存モデル $P_{stop}(r)$（順位 r で推薦リストの閲覧を中止する確率）は，以下の式で表される．

$$P_{stop}(r) = \prod_{i=1}^{r-1}(1-R_i)R_r$$

R_i は，ユーザが推薦アイテムに満足し，それ以降の推薦リストの閲覧を止めてしまう確率である．このように，順位 r で打ち切る可能性を，順位 $1 \sim r-1$ までで打ち切らず，順位 r で初めて打ち切ることをモデル化することで表現している．

ERR (Expected Reciprocal Rank) は，ユーザの推薦リスト閲覧の打ち切りという行動を組み込んだ評価指標である．ユーザが推薦リストのより上位で閲覧を止めれば，より良い推薦結果であったというポリシーに基づいている．ユーザが順位 r まで推薦リストを閲覧するかどうかを，カスケード依存モデルで表す．そして，その順位まで閲覧していれば，順位に反比例する値 $\phi(r)$ を指標のスコアとして与える．つまり，下位の順位まで見てしまうと，その加算が小さくなるという考え方に基づいている．ERR は下記の式で計算される．

$$ERR = \sum_{r=1}^{n} \phi(r) P_{stop}(r)$$

ここで，$\phi(r)$ は，その順位で閲覧を打ち切った時の価値を表す関数を用いる．すなわち，順位が大きくなればなるほど減少する関数である．これには単純に順位の逆数を使うか，その減衰を和らげるために対数を取った関数を用いる．

$$\phi(r) = \frac{1}{r}$$

または，

$$\phi(r) = \frac{1}{log_2(r+1)}$$

となる．関数 $P(stop_r)$ は，上記のカスケードモデルを表す関数を用いる．これにより ERR は下記の式になる．

$$ERR = \sum_{r=1}^{n} \left\{ \phi(r) \prod_{i=1}^{r-1}(1-R_i)R_r \right\}$$

ここで，R_i は以下の式で与えられる．

$$R_i = \frac{2^g - 1}{2^{g_{max}}}$$

g はユーザの評価値，g_{max} は最大のユーザの評価値（1-5 段階で表した場合は 5）を表す．

ERR は，ユーザのリスト閲覧行動をよく表したモデルではあるが，このモデルの着想を得た元々のアプリケーションは，情報検索（実験では検索エンジンのクリックスルーデータ）である．情報検索の場合，ユーザの目的がはっきりしているので，検索クエリに適合する結果を見つければ，それ以降の閲覧を止める可能性が高い．しかし，推薦システムでは，ユーザは自分の興味や嗜好に適合するアイテムを見つけたとしても，その推薦リスト

を目的なく閲覧している場合や，他のアイテムと比較してから購入するどうかを意思決定したりする場合があるため，必ずしも適したモデルであるとは言い難い．ただし，ユーザが推薦してもらうアイテムのジャンルを指定したり，現在置かれているコンテキストを明示したりするなど，何らかの入力を行う場合には，情報検索と同様の行動を取る可能性がある．

最後に，この節で示してきた，順位依存モデルは，Carteretteによってより一般的なフレームワークで表現できることを示されている [Carterette 11]．ユーザが推薦結果の各アイテムから得た利得とその集積によって，4種類のモデルに集約されるとしている．興味のある方は参考にされたい．

3·5 正確性評価の限界

最後に，本稿で扱ってきた正確性評価の限界について述べておく．ユーザが推薦の価値を判定する時には，推薦リストの中にどのようなアイテムが含まれているかが最も重要になると思われる．従来の多くの推薦システムの研究においても，何 (what) が推薦されたかについて評価してきた．しかし，文献 [Olmo 08] にあるように，正確性の評価においても何 (what) が推薦されたかだけでなく，いつ (when)，どのように (how) 推薦されたのかも考慮して評価すべきだという主張もある．また，推薦システムの利用の機会が増えているため，さらにどこで (where) 推薦されたのかも含めて評価する必要があるかもしれない．しかし，本稿ではこれらを考慮した評価方法や評価指標の紹介までは行わないことにした．これらの評価は，オフライン評価では難しい点と（データセットに上記のようなコンテキストに関するデータが含まれていない上，ユーザは推薦された時のコンテキストの適切さを回答する必要があるため），未だこれらの評価方法は確立されていないからである．今後の評価研究の発展に期待するとともに，これらの紹介は別の機会に委ねることとする．

4. 発見性に関する評価指標

これまで多くの研究の評価指標は正確性に関するものであった．しかし，近年の研究の中には，推薦結果がユーザにとって本当に利益 (utility) をもたらすものであるかどうかを考慮するものも増えてきた．このような観点を利便性 (usefulness) と呼ぶ [Herlocker 04]．ここまで議論してきた推薦の正確性も，利便性を構成する概念の一つである．しかし，商用システムでは，推薦システムがより多くの商品を扱えないといけない．また，新規ユーザに対して早くに一定水準以上の推薦を行えるようにならないといけない．このように，利便性の評価には実運用を考慮した指標が必要となる．

また，特に利便性として注目されるのは，推薦されたアイテムがユーザにとって目新しいものであったかどうかと言う観点である．推薦サービスは一種の情報提供サービスと考えることができる．そのため，いつも同じ情報ばかりを提供していたのでは，ユーザはすぐにサービスに飽きてしまう．また，明らかにユーザが必要とするアイテムを推薦することも，ユーザの利益につながらないことが指摘されている [Yang 01]．本稿ではこのような観点を発見性 (discovery) と呼ぶ．発見性は，推薦システムの分野では serendipity（意外性）や novelty（新規性）という言い方で表現され [Herlocker 04]，これらの向上を目指した研究が多く行われている．また，発見性に関連する考え方に diversity（多様性）という概念もあり，これに関連する指標も盛んに研究されている．

ここで，発見性の説明を，発見性に欠ける推薦の例を挙げて行う．例えば，いつもオンラインのミュージックストアで音楽を購入しているユーザがいたとする．このユーザは，あるアイドルグループが好きで，過去にそのグループのアルバム数枚をそのストアで購入していたとする．ここで，このユーザにそのアイドルグループの他のアルバムを推薦したとする．このアルバムを好むかどうかという観点では，この推薦は正確であると言える．恐らくそのユーザは，そのアイドルグループの作品なら，何でも気に入ると思われるからである（少なくとも平均よりは上であると思われる）．

しかし，そのユーザがそのストアでまだ購入していないということは，他の販売ルートですでに手に入れているか，以前から買わないと決めている（すでに購入しているアルバムより好きな曲が入っていないなどの理由で）のかもしれない．いずれにしても，ユーザがすでに知っている作品は，そのユーザが過去にその作品に関する視聴や購入の意思決定をしている可能性が高い．仮に将来購入しようと予定をしていたとしても，逆にそのアイテムは推薦しなくともいずれ購入されると思われ，推薦のメリットは少ない．このような推薦結果は，協調フィルタリング，とりわけアイテムベースの協調フィルタリングで起こりやすい [McNee 06b]．あるアイドルグループの作品は，多くのユーザがまとめ買いをしていたり，全て揃えていたりするためである．

この章では，まず発見性以外の利便性に関連する指標を紹介する．次に，意外性，新規性，多様性の意味を説明する．そして，多様性，新規性，意外性の順で，具体的に算出可能な式として提案されている評価指標を紹介する．

4·1 発見性以外の利便性に関する指標

発見性以外で利便性に関連する指標を紹介する．これらが低いとアイテムと接する機会を失ってしまったり，推薦システムを信頼することができなくなったりしてしまう可能性があるため，重要である．

§1 被覆率 (Coverage)

正確性以外の評価指標で，特に商用システムにおいて重視される指標が被覆率 (coverage) である [Sarwar 98]. これは，システムで扱っているアイテムのうち，実際に推薦可能なアイテムがどれほど存在するかを示す指標である．推薦アルゴリズムによっては，推薦対象のアイテムに関する必要なデータがそろっていなければ推薦できないことがある．例えば，協調フィルタリングであれば，誰からも評価されていないアイテムは，誰にも推薦されることはない．コンテンツに基づくフィルタリングであれば，ある特徴量が欠損している場合，アルゴリズムによっては推薦できないことがある．そのようなアイテムは少なければ少ないほど良い．被覆率には大きく分けると，prediction coverage と catalogue coverage の 2 種類がある [Herlocker 04]．以下，順に説明する．

prediction coverage は，システムに登録されているアイテムのうち，どれだけが推薦アルゴリズムにより推薦対象とできるかを示したものである．prediction coverage Cov_{pre} の算出式は，以下のようになる．

$$Cov_{pre} = \frac{|B_{able}|}{|B|}$$

ここで B は，システムに登録されているアイテムの集合，B_{able} はそのうち推薦システムにより推薦可能なアイテムの集合である．B_{able} の定義は，評価対象の推薦システムにより異なる．このため，統一的な基準での適用が難しい指標とも言える．協調フィルタリングのアルゴリズムによっては，アイテムに対してある閾値以上の数の評価値を持つもののみ推薦対象にしている場合がある．そのような場合，B_{able} は閾値以上の数の評価値を集めるアイテムの集合となる [Ge 10].

catalogue coverage[Ge 10] は，ある時点（期間）での推薦結果を用いて被覆率を計算するもので，実際にユーザに提示する推薦リスト内に含まれたアイテムのみ，推薦可能なものとして考えるものである．j 回目の推薦であるユーザに提示される推薦リストを L_j とする．K を測定期間内に発生した推薦の回数（複数のユーザに対する推薦回数），B をデータセット中のアイテム集合とする．すると，catalogue coverage Cov_{cat} は以下のように計算される．

$$Cov_{cat} = \frac{|\cup_{j=1\ldots K} \Im L_j|}{|B|}$$

prediction coverage は，推薦アルゴリズムがどのようなアイテムを推薦対象とできないかを知っていないと計算できない．すなわち評価者のアルゴリズムに対する深い理解を必要とする．一方，catalogue coverage は，推薦を複数回実行した結果を用いて計算すればよいので，アルゴリズムに対する理解を必要としない点が利点である．

最後に，被覆率の概念をアイテムでなくユーザに適用した指標である user coverage[Kawamae 10] を紹介する．user coverage Cov_{usr} は以下の式で計算される．

$$Cov_{usr} = \frac{|U_{able}|}{|U|}$$

ここで U は，システムに登録されているユーザの集合，U_{able} はそのうち推薦システムにより 1 つでもアイテムを推薦可能なユーザの集合である．評価付けを行うユーザの数が少ない場合や，各ユーザの評価付けを行ったアイテムの数が少ない場合には，アイテムを 1 つも推薦できないことが考えられる．このような状況では，user coverage は便利である．

§2 学習率 (Learning rate)

機械学習のアルゴリズムを用いた推薦システムの多くにおいて，cold-start 問題 (cold-start problem) [Schein 02] は避けて通れない問題である．cold-start 問題とは，新しいユーザが使い始めた時には評価値の入力が少なく正確性の高い推薦が行えないこと，または新しいアイテムが登録された時にユーザからの評価値の入力が少なく，そのようなアイテムが推薦されないことを言う（first-rater 問題 (first-rater problem) または early-rater 問題 (early-rater problem) とも言う [Sarwar 98]）.

学習率 (learning rate) は，システムがどれだけ早く，新しいユーザの嗜好を学習し，そのユーザに適切な推薦ができるようになるかを示す指標である．また，既存のユーザであっても，その興味や嗜好が変化した時に，どれだけ早くその変化を推薦に反映させられるかを表す．ユーザが推薦システムに入力を与えてから（初期の評価値付けを終えてから），システムがすぐに推薦結果を返さないと，ユーザの満足度が低下することが実験で確かめられている [Jones 07]．このことからも学習率は重要な評価指標であることが分かる．

具体的な算出方法としては，既存のユーザの嗜好が変化してから（あるいは新しいユーザが利用し始めてから），システムがその変化を適切に反映した推薦結果を返せるようになるまでの時間を測定するというものがある．また，既存のユーザの嗜好が変化してから（あるいは新しいユーザが利用し始めてから）決まった時間が経過した後の，推薦の正確性で表現することもある [Koychev 00]. また，推薦システムが利用され始めてからの，ユーザの嗜好の学習の程度や推薦の正確さの向上を調査した研究もある [Rashid 02, Drenner 08]．例えば Drenner の研究では，MovieLens データを用い，オフライン評価にてユーザが推薦システムを利用し始めてからの MAE の推移を調査している [Drenner 08].

しかし，興味や嗜好の変化を，ユーザ自身が明示することは難しい．また，その変化の程度や量を統一することも難しい．そのため，既存ユーザの興味や嗜好の変化を捉えて評価することはかなり困難だと言える．また，新規ユーザも，サインアップ時に評価付けするアイテムの数が異なったり，その後の使用頻度にもばらつきがあるため，統一的な評価指標を定めるのが難しいと言える．

§3 確信度 (Confidence)

ユーザに，7段階評価 (1-7) でアイテムに評価値を入力してもらう場合で，推薦システムがあるアイテムのユーザの評価値を7と予測したとする．この場合，システムはこのアイテムを強く推す（すなわち推薦リストの上位で出力する）こととなる．しかし，予測評価値を7として推薦することは，推薦の強さ (strength) を表しているが，推薦の確かさ (confidence) を表すものではない．概してこのような高い評価値は，ごく限られた数のアイテムから予測されていることが多い．逆に，多くのアイテム（やユーザ）の評価値から予測された評価値の信頼性は高い．

確信度 (confidence) とは，システムがこの推薦をどれだけ確かと思っているかを表した指標である [Sinha 01, Herlocker 04]．予測評価値の算出に用いたアイテム数やユーザ数から計算することができる．また，メモリベースの協調フィルタリングでは，推薦に用いた近傍ユーザの類似度を用いることもできる [Bell 07]．これらは，システム側で事前に計算可能である．ただし，確信度の計算に，推薦に用いたアイテム数やユーザ数をそのまま用いてしまうと，もともとのデータセットの大きさやスパース性によって，値が大きく異なってしまう．また，総アイテム数やユーザ数で割って正規化を行うと，データセットが大きくなればなるほど，値が小さくなってしまう．これも統一した基準での算出が難しい指標と言える．

§4 信頼度 (Trustworthiness)

ユーザに推薦システムを利用してもらうためには，ユーザのシステムに対する信頼 (trust) を獲得しなければならない．ユーザの推薦システムに対する信頼の度合いを信頼度 (trustworthiness) と言う．信頼を得るためには，推薦システムがユーザの興味や嗜好を理解し，それらに適合したアイテムを提示する能力があることを示さなければならない．ユーザがすでに良く知っているアイテムを提示しなければ，ユーザからの信頼を獲得することができないと言われている [Sinha 01]．また，逆に信頼がなければ，システムを使い続ける意思がなくなることも確かめられている [Cramer 08]．また，推薦結果に対する説明付けにより，ユーザが推薦システムを使い続け，信頼を構築することができるとも言われている [Sinha 02, Tintarev 07].

一般的な信頼度を評価する方法は，オンライン評価にてユーザに直接に推薦結果の妥当性や推薦システムへの信頼度を尋ねることである [Bonhard 07, Cramer 08]．オフライン評価で，ユーザの信頼度を計測することは困難であるが，一つの評価方法としては，ユーザのシステムの利用頻度を測るものがある [O'Donovan 05]．これは，文献 [Cramer 08] で確かめられたことから逆に，利用頻度が高ければ，ユーザはシステムを信頼してくれているのであろうという仮定に沿った指標である．

4·2 Serendipity · Novelty · Diversity の意味

ユーザにとっての推薦アイテムの目新しさを指す言葉として，serendipity（意外性）と novelty（新規性）が使われている．意外性のある推薦とは，ユーザは自分では見つけられなかったような，意外にも興味を持てるアイテムを示してくれるものである．一方，新規性のある推薦とは，ユーザが知らなくてかつ興味を持つアイテムを示してくれるものである．

例で，この違いを説明する．あるユーザはある映画を好きであるとする．システムは，その映画の監督の別の作品を推薦し，ユーザはその作品を知らなかったとする．この場合，ユーザは新規性のある推薦を受け取ったと言える．しかし，好きな映画があれば，その映画監督で他の映画を探すという行為は多くのユーザが取るものであるので，これはそのユーザにとっては簡単に見つけられるアイテムと言える．そのため，この推薦は意外性のあるものではない．

また，違う例を挙げると，あるユーザはロマンスの映画が好きでそればかり見ていたとする．しかし推薦システムにより，見たことはないが，タイトル名は聞いたことがある，ある有名な喜劇の映画が推薦されたとする．この喜劇の映画は，主人公がヒロインに一目ぼれして，猛アタックするがうまくいかないという設定で，それでも最後には二人は結ばれるというシナリオだったとする．そしてユーザは，この映画を見て満足したとする．この場合，ユーザにとって既知のアイテムが推薦されているので，新規性のある推薦ではない（ここでは，既知／不既知の定義は，タイトル名を知っているかどうかとしている）．しかし，喜劇の映画として有名だと思っていた映画が，意外にも恋愛の要素を含んでいたため，この推薦はユーザにとって驚きを伴うものであった．そのため，これは意外性のある推薦であると言える．

これらの例からまとめると，推薦結果の意外性を測るには，ユーザが推薦結果に含まれるアイテムを好んだかどうかと，それらが驚きをもたらしてくれたかどうかと両方を測る必要がある．一方，推薦結果の新規性を測るには，ユーザが推薦結果に含まれるアイテムを好んだかどうかと，それらを知らなかったかどうかと両方測る必要がある．知らないアイテムを知らされることは，意外であることが多いが，これらは必ずしも一致しないので注意が必要である．

最後に，diversity（多様性）の意味を説明する．多様性は，推薦されたアイテム群の偏りを見る指標である．推薦システムは通常多くのユーザが利用しているが，システムがそれらのユーザに限られたアイテムのみ推薦していると，おのずと各ユーザが受け取るアイテムも限定的なものになってしまう．また，各ユーザの推薦リスト中のアイテムが，偏ったジャンルやアーティスト（監督や製造元メーカーなども含む）で構成されていれば（図1の例では，このユーザへの推薦リストは特定の漫画のシ

リーズのみが含まれている），ユーザはその推薦にすぐに飽きてしまう可能性がある．多様性はこれらの偏りを計測する指標である．

多様性は非常に実用的な評価指標である．意外性や新規性を，定量的な指標として計算しようとすると，本当にそのアイテムがユーザにとって驚きをもたらしたのか，あるいは知らないものであったのかを，ユーザに尋ねなければならない．それは，実運用上でも実験を行う上でも，非常に困難になる [McNee 06a]．そのため，その代替として，推薦リストの多様性を評価することが多い．しかし代替とは言え，多様なアイテムを推薦しないと，意外性や新規性のあるアイテムを推薦することは難しくなるため，重要な概念である．

多様性に関する評価指標は数多く提案されている．それらは大きく分けると，システム全体でユーザに多様なアイテムを推薦できているかを評価するものと，ユーザごとの推薦リストにおいて多様なアイテムが含まれているかを評価するものの 2 つに分けられる．前者の評価指標としては，4·3 節で説明する凝集多様性 (aggregate diversity)，ユーザ間相違度 (inter-user diversity)，ジニ係数 (Gini coefficient)，時間的多様性 (temporal diversity) がある．後者の評価指標としては，同じく 4·3 節で説明するリスト内類似度 (intra-list similarity)，subtopic retrieval 指標，MMR (Maximal Marginal Relevance)，α-nDCG がある．

ところで，情報検索や推薦システムの研究分野では，意外性，新規性，多様性に関連する指標が盛んに研究されているが，これらの解釈が研究者によって少しずつ異なるのが実情である．特に，新規性と多様性については，その境界が曖昧である．実際の新規性を計測することが困難であるため，新規性の評価を多様性により代用しているケースが多いためである．本稿では，できる限りこの節で説明してきた定義に基づいて説明するが，指標の名前については，元の論文で用いられている名称をそのまま残している．一方，指標が意外性，新規性，多様性のどれを対象としたものかについては，各指標が実質的に何を評価しているかを考慮して，筆者が分類している．元の論文への参照を容易にしつつ，各指標の目的や本質の理解を容易にするためであり，ご理解いただきたい．

以降の節で説明する発見性に関する具体的な評価指標を表 2 にまとめた．ここでは，各指標が，意外性，新規性，多様性のいずれの概念を対象にしたものかに注目している．また，各評価指標の評価対象が，リスト，ユーザ集合，アイテム，アイテム集合，アイテムペアのいずれであるかも記載した．評価指標により，評価対象が異なることがあるので注意されたい．また，発見性に関する評価指標の中には，アイテムへの評価値行列以外の情報を用いるものも多い．筆者は，(1) 評価値行列のみで算出できるもの，(2) オントロジーを利用するもの，(3) 複数の推薦システムを利用するもの，(4) 他のデータセットを利用するもの，の 4 つに分けて，これらのいずれに分類されるかも表に示した．これらは，手持ちにあるデータセットや評価対象の推薦システムで評価可能かどうかに係わってくるため，合わせて確認していただきたい．

4·3 多様性 (Diversity) に関する指標

ここでは，多様性（diversity）に関連する指標を紹介する．前半（4·3·1 節〜4·3·4 節）でシステム全体への多様性に関する指標を，後半（4·3·5 節〜4·3·8 節）で推薦リストへの多様性に関する指標を紹介する．

§1 凝集多様性 (Aggregate diversity)

推薦システムがどれだけ多様なアイテムを推薦できるかを測定するための最も簡単な指標に凝集多様性 (aggregate diversity) がある [Adomavicius 12]．これはユーザに推薦されたアイテムの種類数の総和をとったものである．U をユーザ集合，ユーザへの推薦リストを L_u, とすると，凝集多様性 Agg_{div} は以下の式で計算される．

$$Agg_{div} = |\cup_{u \in U} \Im L_u|$$

システム全体として，ユーザにより多くの種類のアイテムを推薦できていれば，それはすなわち各ユーザに個別化された推薦リストを提供できていることになる．評価値行列のみで算出可能であり，多様性を測定するのに最初に用いるのに適していると言える．

§2 ユーザ間相違度 (Inter-user diversity)

推薦システムがどれだけ個々のユーザに特化した推薦を行えているかどうかを測定することを個別化度合い (degree of personalization) と呼ぶ．Zhou らは，この個別化度合いを測定する方法を提案している [Zhou 10]．特に二人のユーザに注目した時に，そのユーザの推薦リストに含まれるアイテムがどれほど違っているかを測定する指標であるユーザ間相違度 (inter-user diversity) を提案している．

任意のユーザの組（ユーザ $u,v \in U$）を考える．U はユーザ集合である．二人のユーザへの推薦リストをそれぞれ L_u, L_v とする ($|\Im L_u| = |\Im L_v| = N$)．この二人のユーザのリストの違い（距離）は，以下のように計算できる（他の距離関数（集合の類似度を用いた距離関数）でも可）．

$$d_{u,v} = 1 - \frac{|\Im L_u \cap \Im L_v|}{N}$$

これを用いて，ユーザ間相違度 IUD は，以下のように計算される．

$$IUD = \frac{1}{|U|C_2} \sum_{u,v \in U} d_{u,v}$$

また，特定のユーザの推薦リストに焦点を当てた時，その推薦リストがどれほど他のユーザの推薦リストと異なっているかを表す指標も提案している．これをリストの個別化度合い (list personalization metric) と呼ぶ．ア

表 2　List of evaluation metrics regarding to discovery

Metrics	Type of Discovery	Target	Other Info.	Inventor
Aggregate diversity	Diversity	User set	None	Adomavicius, IEEE 2012
Inter-user diversity	Diversity	User set	None	Zhou, NAS 2010
List personalization metric	Diversity	List	None	Zhou, NAS 2010
Gini coefficient	Diversity	Item set	None	Fleder, EC 2007
Temporal diversity	Diversity	List pair	None	Lathia, SIGIR 2010
Intra-list similarity	Diversity	List	Ontology	Ziegler, WWW 2005
Subtopic retrieval	Diversity	List	Ontology	Zhai, SIGIR 2003
MMR	Diversity	List	Ontology	Carbonell, SIGIR 1998
α-nDCG	Diversity	List	Ontology	Clarke, SIGIR 2010
Discovery ratio	Novelty	List	Acquaintance rating	Hijikata, IUI 2009
Precision of novelty	Novelty	List	Acquaintance rating	Hijikata, IUI 2009
Item novelty	Novelty	Item	Ontology	Zhang, RecSys 2008
Temporal novelty	Novelty	List	None	Lathia, SIGIR 2010
Novelty based on HL utility	Novelty	List	None	Shani, RecSys 2008
Long tail metric	Novelty	List	None	Celma, RecSys 2008
Generalized novelty model	Novelty	List	None/Ontology	Vargas, RecSys 2011
Unexpectedness	Serendipity	List	Other system	Murakami, LNCS 2008
Entropy-based diversity	Serendipity	List	Other systems	Bellogin, HetRec 2010
Unserendipity	Serendipity	List	Ontology	Zhang, WSDM 2012
HL utility of serendipity	Serendipity	List	Serendipity rating	Murakami, JSAI 2009

イテム b が与えられた時，このアイテムがユーザにより選択される（高く評価される）確率 p_b は，下記のように計算できる．

$$p_b = \frac{|U_b|}{|U|}$$

ここで，U_b はユーザ集合全体 U のうちで，アイテム b を選択したユーザの集合を表す．この確率を用いれば，アイテムの選択情報量 (self-information)I_b は，

$$I_b = log_2\left(\frac{|U|}{|U_b|}\right)$$

で表される．これを用いると，ユーザ a_i の推薦リスト L_i ($|L_i| = N$) の個別化度合い $Per(L_i)$ は以下の式で計算される．

$$Per(L_i) = \frac{\sum_{b_j \in \Im L_i} log_2 \frac{|U|}{|U_{b_j}|}}{N}$$

ユーザ間相違度もリストの個別化度合いも，評価値行列のみで算出可能である．個別化された推薦リストが提供できているかを確認するのに用いると良いと思われる．

§3　ジニ係数 (Gini coefficient)

ユーザ間相違度は，どれほど多様なアイテムがユーザ集合全体に推薦されているかを考えたが，逆にユーザ集合全体に，どれほど偏ってアイテムを推薦しているかを見る指標にジニ係数 (Gini coefficient) がある．これは，経済学において，注目する値の不均衡について述べる時に使われる．よく引用される例としては，その国の所得格差を表現する場合である．本稿でも，まずは人口（世帯階級）とその階層の所得金額の関係を例に説明する．ジニ係数の算出では，一般に横軸に世帯階級による人口の累積比 u (cumulative share of people) を，縦軸に世帯階級による所得金額の累積比 $L(u)$ (cumulative share of income) を取る．ここで横軸は，所得の低い順に並べる．階級をいくつかに区切って計算すると図 15 のようなグラフがかける．このグラフ $L(u)$ はローレンツ曲線 (Lorenz curve) と呼ばれる．

ここで，一般にはジニ係数 G は以下の式で算出できる．

$$G = 1 - 2\int_0^1 L(u)du$$

図 15 には，(0,0)-(1,1) に 45 度の直線 (line of equality) を書いているが，この直線とローレンツ曲線の間の領域の面積を A，ローレンツ曲線の下の領域の面積を B とすると，ジニ係数 G は以下の式で算出できる．

$$G = A/(A+B)$$

実際には，グラフのように離散値でローレンツ曲線を取るため，この面積は三角形と台形の面積を求める公式を使い簡単に求めることができる．

Fleder と Hosanagar は，商用サイトで推薦システムがある時とない時で，アイテムの売り上げを縦軸にとりジニ係数を用いた分析をしている [Fleder 10]．これは，どのアイテムが多く売り上げ（金額）を上げたかを示してい

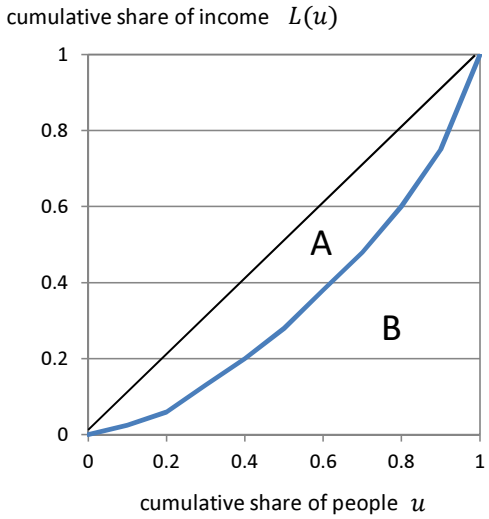

図 15 An example of calculation of Gini coefficient and drawing of Lorenz curve

る．彼らは音楽 CD のドメインで実験を行っている．音楽 CD の単価はアイテムによりそれほど格差はないと思われるので，これはすなわち売上本数と見て差し支えない．また，Kawamae も音楽と映画の推薦システムの多様性の評価指標の一つとしてジニ係数を用いている [Kawamae 10].

推薦システムにより推薦されたアイテムの多様性の評価に当てはめて考えると，縦軸を全ユーザを対象にしたアイテムの推薦リストへの登場回数（の累積比）とすれば良い．横軸を登場回数の少ないアイテム順に並べれば（アイテムの累積比），推薦システムの推薦アイテムの偏りの度合いが計算できる．ジニ係数は，評価値行列のみで算出できる．経済学の分野でもよく用いられるため，他分野の人に説明するのに適していると思われる．

§4 時間的多様性 (Temporal diversity)

推薦システムが，時間の経過とともに，以前と違う推薦結果を返すことができるかどうかを測る指標に時間的多様性 (temporal diversity) がある [Lathia 10]．時間の経過とともに，ユーザはアイテムへの評価値を増やすであろうし，システム全体でも新しいユーザと新しいアイテムが入ってくる．このような変化に対応するため，既存の推薦システム（特に，アイテムベースの協調フィルタリングを採用しているシステム）では，定期的に上記の新しいデータを推薦機構に反映させる．したがって，時間の経過とともにシステムが，異なる推薦結果を返していることを計測する必要がある．

同じ対象ユーザに，異なる時間で，同じシステムを用いて作成した 2 つの推薦リスト L_1 と L_2 ($|\Im L_1| = |\Im L_2| = N$) を取り上げる．この 2 つの推薦リストの時間的多様性 $Div_{tmp}(L_1, L_2, N)$ は以下の式で計算される．

$$Div_{tmp}(L_1, L_2, N) = \frac{|\Im L_2 \backslash \Im L_1|}{N}$$

ここで，$\backslash$ は，

$$\Im L_2 \backslash \Im L_1 = \{x | x \in \Im L_2 \wedge x \notin \Im L_1\}$$

を表す．つまり，新しい推薦リスト L_2 に，以前の推薦リスト L_1 に入っていなかったアイテムが多いほど，時間的多様性は大きくなることを意味する．

時間的多様性は，推薦時刻の異なる 2 つのリストに含まれるアイテムが，完全に一致していれば 0 で，全く異なれば 1 である．時間の経過とともに，どれだけ異なるアイテムが含まれているかを確認するのに便利である．過去にユーザに提示された推薦リストについて，全ての組の時間的多様性を計算し，さらに全ユーザでその平均を計算すれば，システムに対する時間的多様性を計算できる．なお，リスト中のアイテムの順番は考慮していないことに注意する必要がある．また，評価値行列のみで算出可能であるため，実用的である．

§5 リスト内類似度 (Intra-list similarity)

推薦リスト中のアイテムに目新しさを感じない場合と言うのは，概してそれらが同じような内容のものであったり，あるジャンルに偏っていたりすることが多い．本節以降では，多様性を評価する指標においても，ジャンルやカテゴリに関する情報であるオントロジーを必要とするものについて説明する．オントロジーと書いたが，ここで使われるオントロジーは，概念間の関係の種類まで定義されたような本格的なものではない．ジャンルやカテゴリに関する階層関係（図 16 に例を示す）を利用する程度のシンプルなものである．

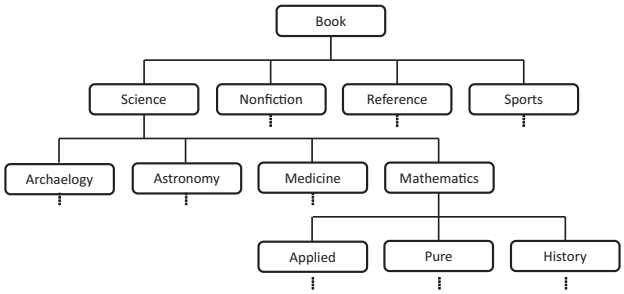

図 16 Ontology (category) used for measuring diversity (Reprinted from the figure in [Ziegler 05])

このようなリスト内のアイテムの内容やジャンルに関する多様性 $Div(\Im L_i)$ を評価する指標は，下記のように一般化して表現できる [Ziegler 05].

$$Div(\Im L_i) = \sum_{l,m \in \Im L_i} \frac{1}{sim(l,m)}$$

ここで，L_i が評価対象のリスト，$sim(l,m)$ は 2 つのアイテム l,m の類似度を返す関数である．

リスト内のアイテムがどれだけ似通っているかを測定する最も基本的な指標はリスト内類似度 (intra-list similarity) [Ziegler 05] である．これは，推薦リスト内の任意

の 2 つのアイテムを取り出し，そのアイテム間の類似度の総和を取ったものである．

L_i を推薦リスト ($|\Im L_i| = N$)，アイテム集合を B，推薦リスト中のアイテムを $b \in B$，二つのアイテム間の類似度を測定する関数を $sim(b_k, b_e) : B \times B \longrightarrow [-1, +1]$ とすると，リスト内類似度 ILS は以下のように計算される（ただし，文献 [Ziegler 05] の定義に基づき，正規化を行っている）．

$$ILS(L_i) = \frac{\sum_{b_k \in \Im L_i} \sum_{b_e \in \Im L_i, b_k \neq b_e} sim(b_k, b_e)}{{}_N C_2}$$

類似度を算出する関数は，任意の関数を利用できる．もし，アイテムにコンテンツに基づく特徴量が付与されていれば，それを用いて標準的な類似度の指標が利用できる．例えば，コサイン類似度やピアソン相関などである．ただし，一般的な商用サイトでは，詳細なコンテンツに基づく特徴量が付与されているとは限らない．その代り，多くの商用サイトでは，ユーザの商品へのナビゲーションのために，カテゴリ（ディレクトリ構造）が用意されている．Ziegler らは，このカテゴリと木構造での上位・下位関係も利用して類似度を計算している [Ziegler 04].

上記で示したリストの多様性は，Zhang と Hurley によって一般化されており [Zhang 08]，リスト内多様性 (intra-list diversity) とも呼ばれている [Adamopoulos 13]．以下に一般化の詳細を示す．アイテムの集合を B ($|B| = M$)，推薦リストを L_i ($|\Im L_i| = N$) とする．ここで，B 中のアイテムが L_i に含まれるかどうかを表すベクトル $\mathbf{y}$ ($y(i) = 1$ if $b_i \in \Im L_i$ and $y(i) = 0$ otherwise) を用いる．また，行列 D を $M \times M$ でアイテム間の距離関数 ($d : B \times B \longrightarrow \mathbb{R}$) の演算結果を格納しているとする．行列 D の一要素は，$d(b_i, b_j)$ と表される．すると，推薦リスト L_i の多様性 $Div(\Im L_i)$ は，以下のように表される．

$$Div(\Im L_i) = \frac{1}{2 \cdot {}_N C_2} \mathbf{y}^T D \mathbf{y}$$

ここで示した推薦リスト内の多様性を上げるためには，上位で提示したアイテムと類似度の低いアイテムを，推薦リストに入れる必要がある．しかし，上位で提示したアイテムはユーザの興味や嗜好に適合している可能性が高いため，それらとの類似度の低いアイテムを入れることは，ユーザの興味や嗜好から外れたアイテムを入れることにつながる可能性が高い．そのため，正確性と多様性にはトレードオフの関係があると言える．多様性を上げるためには，どの程度正確性を犠牲にしても良いかを判断しなければならない．正確性については，同じようにトレードオフの関係がある適合率と再現率について，判定の閾値を変えながら適合率-再現率曲線を描いた．これと同じように適合率-多様性曲線を描くこともできる．

§6 Subtopic retrieval 指標

既存のカテゴリのうち，推薦リストに提示されたアイテムがどれほど多くのカテゴリを網羅していたかを示す指標に subtopic retrieval 指標がある [Zhai 03]．subtopic retrieval 指標はいくつかの派生バージョンがあるが，その基本形である subtopic recall について説明する．subtopic recall ($S\text{-}recall$) は，既存のカテゴリの総数 n に対する，推薦リスト L_i 中のアイテム s_j のカテゴリの種類数の割合で計算される．

$$S\text{-}recall = \frac{|\bigcup_{s_j \in \Im L_i} subtopics(s_j)|}{n}$$

ここで，$subtopics(s_j)$ は，アイテム s_j の属するサブトピックの集合を返す関数である．指標の名前に subtopic とあるのは，これはもともと情報検索の分野で提案された指標で，検索クエリそのものが持つトピックに対して，多くの種類のサブトピックを返すことが重要であるという考えに基づいているからである．多くの種類のトピック（カテゴリ）を含むべきであるという考え方は，推薦システムにおいても有効であると考えられる．また，高い順位において，早く $S\text{-}recall$ が大きくなることを理想のランキングとしている．

§7 MMR (Maximal Marginal Relevance)

推薦リストの各順位において，さらに上位の順位において似たアイテムが推薦されていないかを考慮して多様性を算出する指標に，MMR (Maximal Marginal Relevance)[Carbonell 98] がある．この指標も情報検索の分野で開発されたものである．実は，この指標は検索結果のリストを評価するためのものでなく，検索結果のリストを作成する段階で，リストの各順位にどのアイテムを配置するかを決めるために，アイテムを評価するためのものである．しかし，その基本的な考え方は作成されたリストを評価するのにも使える．以下，筆者が推薦システムの課題に適用した式を紹介する．推薦リストを評価するための MMR MMR_{List} は以下の式で計算される．

$$MMR_{List} = \sum_{b_j \in \Im L_i} \{\lambda rel(b_j) - (1 - \lambda) \max_{b_k \in \Im L_i^{j-1}} sim(b_j, b_k)\}$$

ここで，L_i は推薦リスト，L_i^{j-1} は推薦リストの上から $j-1$ 位までのリスト，b_j は推薦リスト中のアイテム，$rel(b_j)$ はアイテム b_j がユーザの興味に適合していたかどうかを返す関数，$sim(b_j, b_k) : B \times B \longrightarrow [-1, +1]$ は 2 つのアイテム b_j, b_k の類似度を返す関数である．$sim(b_j, b_k)$ 関数は，コンテンツベースフィルタリングでは特徴量のベクトルの類似度が，カテゴリ情報を持っている場合はそのカテゴリ間の距離の逆数などが使える．

§8 α-nDCG

推薦リストの正確性を求める指標として，3·4·3 節で nDCG という指標を紹介した．これを推薦リストの多様性を計算するために改良した指標として α-nDCG がある（もともとは情報検索の検索結果の多様化を目指した指標

である）[Clarke 08]．基本的な考え方としては，nDCG ではアイテムに付与された実際のユーザの評価である rel の総和を取り推薦リストの正確性を表現していたが，α-nDCG では，アイテムのカテゴリが今まで（上位リストのアイテム）とは異なるカテゴリであるかどうかの総和を取り，多様性を表現している．α-nDCG は，nDCG と同様の求め方で求めるが，DCG を算出する際の rel を多様性を表す式で置き換える．具体的な算出式は以下のようである．まずは，α-DCG を以下の 2 つ式のいずれかで算出する．ここで，N は推薦リストの長さ，i は推薦リスト中の順位を示す．

$$\alpha\text{-}DCG = rel_1 + \sum_{i=2}^{N} \frac{rel_i}{log_2(i)}$$

$$\alpha\text{-}DCG = \sum_{i=1}^{N} \frac{2^{rel_i} - 1}{log_2(i+1)}$$

ここで，rel_i は以下の式で算出する．

$$rel_i = \sum_{k=1}^{K} J(b_i, k)(1-\alpha)^{r_{k,i-1}}$$

ここで，k はカテゴリの番号を表す（文献 [Clarke 08] では nugget と呼んでいる）．K はカテゴリの総数である．b_i は推薦リストの i 位にあるアイテムを指す．関数 $J(b_i, k)$ は，アイテム b_i がカテゴリ k を含むか否か（1 または 0）を返す．$r_{k,i-1}$ は，推薦リストの $i-1$ 位まででカテゴリ k が出現した回数を表す．このように，これまで数多く出現したカテゴリのアイテムは，rel に大きな値を与えないようになっている．最後は，以下の式で理想のランキングにおける α-DCG（α-IDCG と呼ぶ）との比を求める．

$$\alpha\text{-}nDNCG = \frac{\alpha\text{-}DCG}{\alpha\text{-}IDCG}$$

§9 情報検索分野におけるその他指標

本稿では，情報検索分野における評価指標でも，推薦システムの評価に用いることができるものは積極的に紹介してきた．しかし，情報検索分野における多様性に関する指標の中には，推薦システムの評価に用いることが困難なものもある．ここではそれらについて紹介する．

情報検索分野でよく用いられる指標に，nDCG-IA, MRR-IA, MAP-IA がある [Agrawal 09]．また，これらと同様の考え方を ERR に適用した ERR-IA もある [Chandar 11]．名称末尾の “IA” は，“Intent Aware” を表している．これらは，ある検索クエリで文書検索を行った際，検索結果の有用性を文書と検索クエリの適合性から評価するのではなく，あるカテゴリ（ユーザの検索意図をカテゴリで表せるとみなしている）における検索クエリと文書の適合性と，その検索クエリが与えられた時にユーザの検索意図がそのカテゴリに適合する確率から評価するものである．具体的には，以下の式で算出される．

$$nDCG\text{-}IA(L_i) = \sum_c P(c|q) nDCG(L_i|c)$$

$$MRR\text{-}IA(L_i) = \sum_c P(c|q) MRR(L_i|c)$$

$$MAP\text{-}IA(L_i) = \sum_c P(c|q) MAP(L_i|c)$$

$$ERR\text{-}IA(L_i) = \sum_c P(c|q) ERR(L_i|c)$$

ここで，L_i は推薦リスト ($|\Im L_i| = N$)，c はカテゴリ，q は検索クエリである．$P(c|q)$ は検索クエリ q が与えられた時に，検索意図を表すカテゴリが c となる確率である．$nDCG(L_i|c), MRR(L_i|c), MAP(L_i|c)$ は，検索意図 c を考慮して文書と検索クエリ q の適合性を判断する関数である．カテゴリ c の状況下で，それぞれ nDCG, MRR, MAP, ERR を 3·4·3 節, 3·4·1 節, 3·3·2 節, 3·4·6 節の式で算出する．

ただし，これらの評価指標は，一般的なユーザ集合に向けた利得の最大化を図ることを指標の本質としているため，個々のユーザに向けて結果を最適化する推薦システムの評価には適切ではない．また，クエリは状況によって検索意図が大きく異なることを前提としているが，これはユーザプロファイルには当てはまらない．クエリと検索意図との適合度を過去の事例より確率的に求めているが，推薦システムでは過去に選択していないカテゴリも価値を持つ可能性がある．これらの点から，上記指標を推薦システムの評価に用いるのは適さないと思われる．これらは，情報検索のタスクと情報推薦のタスクの本質的な違いを表していると考えられる．

4·4 新規性 (Novelty) に関する指標

ここでは，新規性 (novelty) に関連する指標を紹介する．多様性と違って，新規性の評価にはユーザがあるアイテムを知っていたか否かを考慮しないといけない．あるいは，そのアイテムを知らない可能性が極めて高いかどうかを考慮しないといけない．

前者の情報を得るには，評価値行列以外のデータセットを用いることになる．そのようなデータセットとして，ユーザがそのアイテムを好むかどうかにはかかわらず，そのアイテムを知っていたか否かを尋ねたものがある [Hijikata 09]．これを既知/不既知の評価 (rating of acquaintace) と呼ぶ．評価値行列と同じように表現できるが，要素の値は 0（知らなかった）か 1（知っている）かのバイナリ値である．これと，興味や嗜好に関する評価値の行列を用いれば，好きかつ知らないという novelty を表す指標が算出できる．なお，オンライン評価での試みではあるが，ユーザに音楽を推薦し（音楽のみ提示．アーティスト名や曲名などのメタデータは提示しない），各曲を視聴してもらった後に，各アイテムに 1) 全く知らない，

2) アーティスト名のみ思い出せる（既知），3) アーティスト名と曲名の両方を思い出せる（既知），の 3 種類で問う方法（音楽ドメイン対象）もある [Celma 08]．アイテムの提示方法により質問の仕方が変わってくると思われる．

後者の情報を得るには，他のユーザからの人気度を算出したり，アイテム間の類似度を算出したりする必要がある [Zhang 08, Celma 08, Shani 08, Meyer 12]．人気度は，評価値行列のみからでも計算できる．アイテム間の類似度は，アイテムの特徴量を抽出しておくか，ジャンルやカテゴリーなどのオントロジーを必要とする．前者の情報よりは取得が容易であるため，これを用いた実用的な評価方法が考案されている．

§1 発見率 (Discovery ratio)

発見率 (discovery ratio) は，既知/不既知の評価値のみを用いて，推薦リストに挙げられたアイテムのうち，どれだけのアイテムを知らなかったかを表す指標である [Hijikata 09]．ここでは，ユーザの興味や嗜好に適合しているかどうかは問わない．推薦システムが未知のアイテムを推薦する能力があるかどうかだけを測定する指標である．発見率 $Discovery$ は以下の式で計算される．

$$Discovery = \frac{|D_i \cap \Im L_i|}{|\Im L_i|}$$

ここで D_i は，ユーザ a_i の知らないアイテムの集合で，L_i はユーザ a_i への推薦リストである．発見率は，知らないアイテムを推薦できたかどうかのみ見ているため，次に説明する novelty の適合率と合わせて考慮したい指標である．

§2 Novelty の適合率 (Precision of novelty)

novelty の適合率 (precision of novelty) は，既知/不既知の評価値と嗜好の評価値を用いて，推薦リストに挙げられたアイテムのうち，どれだけのアイテムが好きかつ知らないものであったかを表す指標である [Hijikata 09]．novelty の適合率 $Prec_{novelty}$ は以下の式で計算される．

$$Prec_{novelty} = \frac{|D_i \cap F_i \cap \Im L_i|}{|\Im L_i|}$$

また，興味・嗜好に対する正確性と同じように，再現率 (recall of novelty) も計算できる．novelty の再現率 $Recall_{novelty}$ は以下の式で計算される．

$$Recall_{novelty} = \frac{|D_i \cap F_i \cap \Im L_i|}{|D_i \cap F_i|}$$

ここで，F_i と D_i はそれぞれ，ユーザ a_i が好むアイテムの集合と知らないアイテムの集合である

novelty の適合率は，数ある発見性に関連する指標の中でも，最も信頼できる指標である．直接にユーザに既知/不既知の情報を尋ねているので，知らないアイテムを正確に指標に反映できているためである．また，アイテムを知っているか否かは，評価の際の揺らぎが非常に少ない点も利点である．しかし，既知/不既知の評価値をユーザに尋ねている推薦システムは少ないため，適用できるケースが少ない点が問題である．小規模でも良いので，特定のユーザ群から既知/不既知の評価値を聞き出し，上記の指標を算出しておくと参考になると思われる．

§3 アイテム新規性 (Item novelty)

推薦の新規性の評価を行うに当たって，推薦されたアイテム単位で新規性の測定を行う指標をアイテム新規性 (item novelty) [Zhang 08] と言う．アイテム新規性 Nov_{L_i,b_j} は，リストの多様性を利用して（算出式は 4·3·5 節を参照），以下のように算出される．

$$\begin{aligned} Nov_{L_i,b_j} &= N \cdot (Div(\Im L_i) - Div(\Im L_i - b_j)) \\ &= \frac{1}{N-1} \sum_{b_k \in \Im L_i} d(b_j, b_k) \end{aligned}$$

L_i は推薦リスト ($|L_i| = N$)，b_j は前記推薦リスト中のアイテムである．また，$d(b_j, b_k)$ はアイテム間の距離関数，$Div(\Im L_i)$ はリスト L_i 中のアイテム集合の多様性を返す関数である．測定対象のアイテムがリストに加わった時に，大きく多様性が増す時に，そのアイテムの新規性を高く算出する．

§4 時間的新規性 (Temporal novelty)

4·3·4 節では，時間の推移によって，システムが異なる推薦リストを提供しているかどうかを考えた．この考え方は，推薦リストの新規性の評価にも拡張できる．Lathia らは，過去に受け取った全ての推薦リスト中のアイテム群 S_{past} を考慮した時に，今回受け取った推薦リスト L_i ($|L_i| = N$) に，どれほど新しいアイテムが含まれていたのかを示す指標として，時間的新規性 (temporal novelty) を提案している [Lathia 10]．時間的新規性 Nov_{tmp} は，以下の式で計算される．

$$S_{past} = \bigcup_{j=1}^{i-1} \Im L_j$$

$$Nov_{tmp} = \frac{|\Im L_i \backslash S_{past}|}{N}$$

過去に推薦したものを再度推薦することに，ユーザの利便性はないため，重要な評価指標である．また，評価値行列のみで算出可能である．ただし，過去に推薦したものをリスト化しておき，今回の推薦リストから除外する仕組みを導入すれば，上記の値は簡単に 1 にすることができる．そのため，時間的新規性が低くとも，その問題は簡単に解決できると言える．また，ユーザにとって本当に新規であったかどうかを考慮していない点が，この指標の欠点である．

§5 HLU に基づく新規性 (Novelty based on HLU)

推薦リストの順位を考慮した正確性に関する指標として，3·4·4 節にて Half-life Utility Metric (HLU) を紹介した．これに人気度の高いアイテムを予測するほどペナ

ルティを大きくする指標として，HLU に基づく新規性 (NHLU: Novelty based on HLU) [Shani 08] がある．

HLU の式を再掲すると，以下のようになる．

$$HLU_u = \sum_i^N \frac{max(rel_{u,i} - d, 0)}{2^{(i-1)/(\alpha-1)}}$$

NHLU では，人気度の低いアイテムほど重みを大きくするため，以下の関数を導入している．

$$f(i) = log_2\left(\frac{n}{n_i}\right)$$

n がデータセット中のユーザ数で，n_i がアイテム i を好んだユーザ数である．これを用いて，NHLU を以下のように算出している．

$$NHLU_u = \sum_i^N f(i)\frac{max(rel_{u,i} - d, 0)}{2^{(i-1)/(\alpha-1)}}$$

各アイテムにおいて元の HLU の値に人気のなさの程度をかけていることが分かる．これも，評価値行列からのみ計算できるが，本当にユーザにとって新規性があったのかどうかまでは評価していない．

§6 ロングテールに基づく指標 (Metrics based on long tail)

単一の式で表現できる指標ではないが，Celma と Herrera はロングテール (long-tail) の現象を使って推薦システムが新規性のあるアイテムを推薦する能力があるかどうかを評価する方法を提案している [Celma 08]．彼らは音楽のドメインにおいて，各曲の再生回数に注目している．アーティストごとにその曲の再生回数を縦軸にとり，再生回数の多い順にアーティストを横軸に並べれば，図 17 の左のようなグラフが描ける（横軸は対数を取っている）．これは，再生回数の多いアーティストは少数で，大半はあまり（ほとんど）再生されない曲ばかりをリリースしているアーティストばかりということを示している．なお，図は実際のデータで *last.fm* における 2007 年 7 月の再生回数のデータである（図は [Celma 08] より許諾を取り転載）．

これを，縦軸に累積再生回数の割合 (cumulative percentage of play counts) を取ると，図 17 の右のようなグラフが描ける．このようなロングテールの現象を表すのに，x 位にあるオブジェクトの累積量の割合を返す関数 $F(x)$ で表現されるモデル [Kilkki 07] を用いている（以下の式参照）．

$$F(x) = \frac{\beta}{\left(\frac{N_{50}}{x}\right)^{\alpha} + 1}$$

ここで，α は関数の S 字の形態を定義する要素，β は総累積量，N_{50} は総累積量が 50%になる時の順位である．

この式を用いて，彼らはこのロングテールの曲線を，$head, mid, tail$ の 3 つの部分に分割している．分割の閾値は以下のように式で決定している．

$$Xhead \rightarrow mid = N_{50}^{2/3}$$

$$Xmid \rightarrow tail = N_{50}^{4/3}$$

これにより，図 17 の右のような分割になる．この図では，N_{50}=737 であり，これより $Xhead \rightarrow mid = 82$, $Xmid \rightarrow tail = 6655$ となる．また，$\alpha = 0.73$, $\beta = 1.02$ としている（設定の詳細は [Celma 08] を参照）．

表 3 The table representing the percentage of translation among the three parts (Head, Mid and Tail) of item pairs including in the recommendation list (The values are those reprinted from [Celma 08].)

$a_i \rightarrow a_j$	Head	Mid	Tail
Head	45.32%	54.68%	0%
Mid	5.43%	71.75%	22.82%
Tail	0.24%	17.16%	82.60%

推薦システムにより，上位 N 個 (Top-N) のアイテムから成る推薦リストを作成したとする．このうち任意の 2 つのアイテム a_i と a_j を取り出したときに，これらのアイテムが上記 3 つの区分のうち，どの区分とどの区分に当てはまるかの割合を取ったものを新規性の評価とする．これは，表 3 のような 9 つの要素からなる表で表現できる．割合は行ごとに計算し，各行の値は合計すると 100% になるようにする．$head \rightarrow head$ が非常に多いと，システムが新規性のあるアイテムを推薦する能力は低いことになる．表 3 には，文献 [Celma 08] において協調フィルタリングの上位 20 位までの推薦リストの結果も合わせて示している．$head \rightarrow head$ の割合が 45.32%とあり，新規性の低い推薦を行っていることが分かる．

§7 新規性モデルの一般化 (Generalized novelty model)

ジニ係数を用いた多様性の評価（4·3·3 節参照）やロングテールに基づく新規性の評価（4·4·6 節参照）では，アイテムがどれだけ多くのユーザから評価されているか（人気度）を用いた．また，リスト内類似度や MMR では，アイテム間の類似度を用いた（4·3·5 節, 4·3·7 節参照）．人気度や類似度が低いアイテムは，ユーザが知らない可能性が高く [Meyer 12]，これを新規性の評価に用いることができる．Vargas と Castells は，これらの考え方を基に，アイテムの新規性と推薦リストの新規性を評価するための一般的なモデルを提案している [Vargas 11]．

彼らは，アイテムが選択されたか ($choose$)，アイテムが推薦リスト中にあることを確認したか ($seen$)，ユーザはアイテムに興味を持ったか (rel) に分けて，それぞれをバイナリ値で表現している．ユーザの選択行動は，次のようにモデル化できるとしている．

$$p(choose) \sim p(seen)p(rel)$$

また，アイテム i の新規性 $nov(i|\theta)$ を測定する二つの指標を提案している．一つは人気に基づく新規性 (popularity-

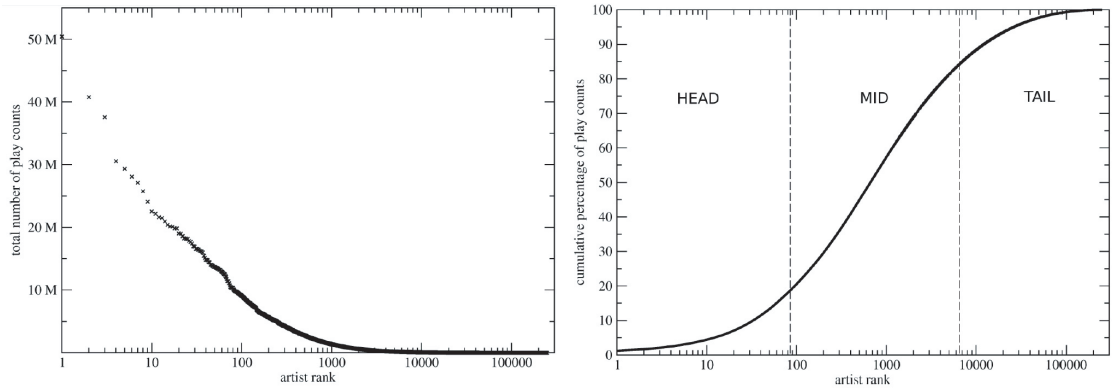

図 17 A graph representing long tail on play counts of each artist in last.fm. (Reprinted from the figure in [Celma 08] with permission)

based item novelty) で，もう一つは距離に基づく新規性 (distance-based item novelty) である．

人気に基づく新規性では，情報量の概念を用いて，下記のように定義している．

$$nov(i|\theta) = -log_2 p(seen|i,\theta)$$

ここで θ は，新規性を表す考え方を意味しており，例えば個人にとって新規であるのか，グループ（ユーザ全体）にとって新規であるのか，時間経過後に新規であるのかなどを表している．このように新規性の概念が変わっても，一つのモデルで新規性を表現できるようにしている．いずれの概念を用いたとしても，基本的には過去のデータから当該アイテムがどれほど提示されやすいかを計算し，それを用いて選択情報量を計算したものである．この考え方は，4·3·2 節で説明したリストの個別化度合いでも見られる．この指標の解釈の仕方であるが，例えば θ をデータセット中の全てのユーザとすると，より多くのユーザによって評価されているアイテムは新規性が低く，ほとんど誰にも評価されていないアイテムは新規性が高いと解釈できる．

距離に基づく新規性は，距離関数を用いて，以下のように定義している．

$$nov(i|\theta) = \sum_{j \in \theta} p(j|choose,\theta,i) d(i,j)$$

ここで，θ は，過去に選択したアイテムの集合を表す．$p(j|choose,\theta,i)$ は，アイテム i を選択していて，アイテム j を選択する確率を表す．$d(i,j)$ は距離関数で，$d(i,j) = 1 - sim(i,j)$ である．$sim(i,j)$ は，コサイン類似度やピアソン相関などの任意の類似度関数が使える．コンテンツに基づくフィルタリングでは，アイテムの特徴量に基づく類似度となる．協調フィルタリングでは，ユーザ群があるアイテムに評価付けしたことを記録したアイテムベクトルに対する類似度となる．つまり，過去に選択したアイテムから距離が遠いにもかかわらず，選択されやすいアイテムが新規性が高いとしている．ここで,「選択されやすい」という概念を入れるべきかどうかについては議論の余地があるが，実用的には $p(j|choose,\theta,i)$ は 1 として計算して構わないとしている．

上記のようにアイテムそのものの新規性が定義できれば，推薦リストが与えられた時の推薦リストの新規性を評価できる．このモデルも一般化して表現している．まずは，ユーザの推薦リストの閲覧モデルを提案している．

$$p(choose|i,u,L) = p(seen|i,u,L) p(rel|i,u)$$

ここで，L は推薦リストを表す．$p(rel|i,u)$ は，ユーザ u がアイテム i を好む確率である．$p(seen|i,u,L)$ は，ユーザが推薦リスト L 中でアイテム i を見る確率である．これが意味するところは，見られないアイテムは選択されないし，嫌いなアイテムも選択されないということである．また，これらの確率は以下の式で計算される．

$$p(seen|i_k,u,L) = \prod_{l=1}^{k-1} p(cont|l,u,L)$$

ここで，$p(cont|l,u,L)$ は，推薦リスト L 中の l 位のアイテムを見た後に，次のランクのアイテムを続けて見る確率を表す．簡単化する場合は，$p(cont|l,u,L) = p_0 (0 < p_0 < 1)$ として計算する．これは，3·4·5 節の RBP の考え方を発展させたものと解釈できる．また，3·4·6 節で説明した ERR のように，適合するアイテムを発見次第，推薦リストの閲覧を中断するモデルを組み込む場合は，

$$p(seen|i_k,u,L) = \prod_{l=1}^{k-1} (1 - p(rel|i_l,u))$$

とする．最後に $p(rel|i,u)$ は，簡単には正解データの値をそのまま用いる．

推薦リスト L の新規性 $nov(L|\theta)$ は，下記の式で計算される．

$$nov(L|\theta) = C \sum_{i \in \complement L} p(choose|i,u,L) nov(i|\theta)$$

これは，ユーザが推薦リスト L のあるアイテムを見て，なおかつそのアイテムに新規性がある場合にのみ，推薦

リストの新規性に加算するというモデルである．C は正規化のための定数である．

アイテムの新規性についても，推薦リストの新規性についても，条件の設定しだいで評価値行列のみから計算可能である点が利点である．また，ユーザが推薦リストを閲覧する詳細なモデルを利用できれば，よりユーザの閲覧行動に適した新規性を計算できるのも利点である．

4·5 意外性 (Serendipity) に関する指標

ここでは，意外性 (serendipity) に関連する指標を紹介する．新規性と同様，意外性の評価においても，ユーザから推薦されたアイテムが本当に意外なものであったかを取得する必要がある．オフライン評価ではないが，ユーザの意外性に関する情報の取得に関しては，村上らによって推薦リストの各番組（テレビの番組推薦）に，意外であるかどうかを尋ねた試みがある [村上 09]．意外の定義としては，過去に番組名を聞いたことがないが興味を持った番組，あるいは，番組名を知ってはいたが，見たことがなく興味のある番組としている．この情報を用いれば，意外性の一つの解釈として定量的な評価を示せる．

しかし，意外であるかどうかの情報は，既知/不既知の評価以上にユーザから引き出すことが難しい．なぜなら，意外であるかどうかは,「推薦の過程において」というコンテキストを前提としているため，オフラインでのアンケートが極めて困難であるからである．そのため，一般の推薦システムが推薦することが難しいアイテムを意外なアイテムとみなす評価指標も提案されている．ここでは，まずユーザに意外性について尋ねない評価指標を紹介し，その後村上らのユーザに意外性を尋ねる評価指標を紹介する．

§1 予測不可能性 (Unexpectedness)

推薦システムが，serendipity なアイテムを推薦するかどうかを手軽に評価できる指標として，予測不可能性 (unexpectedness) がある [Murakami 08]．これは，ごく基本的な推薦アルゴリズムを用いたシステムをプリミティブシステムとして，プリミティブシステムが出力する予測評価値と，評価対象の推薦システムが出力する予測評価値とを比較し，その差が大きいほど serendipity が高いとみなす指標である（図 18 参照）．

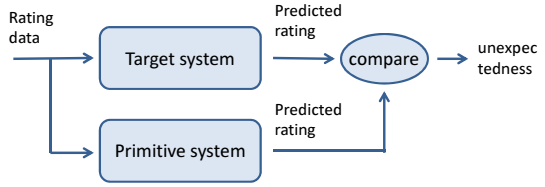

図 18 Basic idea of unexpectedness calculation (Modified from the figure in [Murakami 08])

推薦リスト L_i の予測不可能性 $UNEXP$ は，以下の式で計算される．

$$UNEXP = \frac{1}{|\Im L_i|} \sum_{s_j \in \Im L_i} \{max(P(s_j) - prim(s_j), 0) \cdot rel(s_j)\}$$

推薦リスト中の第 j 位にあるアイテムを s_j と表している．関数 $P(s_j)$ は評価対象のシステムが出力したアイテム s_j の予測評価値を返す．関数 $prim(s_j)$ は，プリミティブシステムが出力したアイテム s_j の予測評価値を返す．関数 $rel(s_j)$ は，アイテム s_j に対するユーザの実際の評価値を返す．

また，DCG や Half-life Utility Metric のように，順位による重みづけをした $unexpectedness_r$ も考案されている．i 位より上位にランク付けされたアイテムのうちユーザの嗜好に適合したアイテムの件数を $count(i)$ として，$unexpectedness_r$ ($UNEXP_r$) は，以下の式で計算される．

$$UNEXP_r = \frac{1}{|\Im L_i|} \sum_{s_j \in \Im L_i} \{max(P(s_j) - prim(s_j), 0) \cdot rel(s_j) \cdot \frac{count(j)}{j}\}$$

なお，同様の考え方は Ge らや Adamopoulos らによっても別の数学的モデルで表現されている [Ge 10, Adamopoulos 13]．彼らは，RS を推薦システムが推薦したアイテム集合，PM がプリミティブシステムが推薦したアイテム集合として，意外性のあるアイテム集合 $UNEXP$ を以下のように算出している．

$$UNEXP = RS \backslash PM$$

また，意外性の評価指標 SRDP を，ユーザに役立つものであったかどうかを考慮して，以下のように定義している．

$$SRDP = \frac{|UNEXP \bigcap USEFUL|}{N}$$

$USEFUL$ は，ユーザに役立ったアイテムの集合であり，N は推薦リスト長である．Murakami らの指標 [Murakami 08] との違いは，ユーザの興味に適合していたかどうかかユーザに役立つものであったかにあるが，実際にはほとんど違いはないと言える．ただし，Adamopoulos らの UNEXP の集合を拡大解釈しており，過去にユーザが閲覧したアイテム群や典型的なアイテム群（誰にでも発見できるアイテム群），さらにこれらと似たアイテム群も含めるとしている [Adamopoulos 13].

この指標では，プリミティブシステムに何を用いるかがポイントとなる．一般に，アイテムベースの協調フィルタリングは，良く似たアイテムを出力し，目新しさに欠けるアイテムを多く推薦すると言われている [McNee 06b]．アイテムベースの協調フィルタリングを用いることができる環境であれば，これをプリミティブシステム

とするのが良いであろう．ただし，この指標は対象のシステムがプリミティブシステムと違う予測をしたかどうかを計測しているに過ぎない．そのため，本当にユーザにとってそのアイテムが意外であったのかどうかは分からない．

§2 情報量に基づく多様性 (Entropy-based diversity)

予測不可能性は，良く似たアイテムばかり出力しがちなプリミティブシステムを用いて，それとの差分を計測する指標であった．しかし，前節で示したようにどのシステムをプリミティブシステムに採用するかは難しい問題である．これをより多くの推薦システムを用いて評価することを考えたのが，情報量に基づく多様性 (entropy-based diversity) である（以下，EBD と略す）[Bellogin 10]．EBD の基本的な考え方は，ある推薦システムがあるユーザに提供した推薦リストがあったときに，その推薦リスト中のアイテムが他の推薦システムでは出力されなかったのかどうかを見る．

推薦システムの集合を A，評価対象の推薦システムを $a \in A$，システム a のユーザ u への推薦リストを $L_{a,u}$，ユーザ u への適合アイテム集合を R_u としたとき，EBD $div_{a,u}$ は以下の式で算出される．

$$div_{a,u} = -\sum_{i \in \Im L_{a,u} \cap R_u} p_{u,i} \cdot log_2 p_{u,i}$$

$$p_{u,i} = \frac{\sum_{a \in A} \delta(a,u,i)}{|A|}$$

ここで，関数 $\delta()$ は，$\delta(a,u,i) = 1$ if $i \in \Im L_{a,u} \cap R_u$ and 0 otherwise となる．$div_{a,u}$ は，システム a がユーザ u に提供した推薦リスト $L_{a,u}$ 内のアイテムに関する平均情報量 (expected value of self information)（またはエントロピー (entropy)）に相当する．情報量を使った考え方は，4·3·2 節の個別化度合いや，4·4·7 節の新規性の一般化でも用いられている．ここでの違いは，アイテムの選択されやすさを，ユーザ集合から求めているのではなく，システム集合から求めている点にある．

この評価指標は，文献 [Bellogin 10] では多様性を計算する方法の一つとして提案されている．しかし，Murakami らの提案 [Murakami 08] の通り，他の多くのシステムでは推薦できなかったアイテムは，ユーザにとって意外なものになるかもしれない．その点で，serendipity を測る指標の一つとみなすこともできる．しかし，この指標の算出はかなり困難なものとなる．まず，確率 $p_{u,i}$ を信頼できる値にするためには，多くの推薦システムを用意しないといけない．しかも，ここで用意する推薦システムは，広範囲の種類のものが必要となる（例えば，ユーザベースの協調フィルタリングの派生バージョンばかりでは，指標の信頼性を上げることができない）．

例えば，文献 [Bellogin 10] では，コンテンツに基づくフィルタリングのアルゴリズムとして，df (document frequency) を用いたもの，BM25 を用いたもの，TF-IDF のコサイン類似度に基づくもの，BM25 のコサイン類似度に基づくものを用いている．また，協調フィルタリングのアルゴリズムとして，ユーザベースのものと，アイテムベースのものを用いている．さらに，社会関係を用いたものとして，ユーザベースの協調フィルタリングの近接ユーザを社会的な友人としたもの，近接ユーザをアイテム評価値の近いユーザと社会的な友人の両方にしたものを用いている．すなわち，合計 8 種類用いている．このように，大きく推薦方式の異なるものを数多く利用することで，確率の信頼性を上げることができる．

§3 非意外性 (Unserendipity)

Zhang らは，ユーザのこれまでの視聴履歴（購買履歴）中のアイテムとの内容の近さを測ることで，アイテムの非意外性を算出している [Zhang 12]．さらに，多数のユーザの推薦リスト中のアイテムに対する値の総和を取ることで，推薦システムが出力する推薦の非意外性を計測している．具体的には，非意外性 $Unseren$ は，以下の式で計算される．

$$Unseren = \sum_{u \in U} \frac{1}{|U||H_u|} \sum_{h \in H_u} \sum_{i \in \Im L_u} \frac{sim(i,h)}{|\Im L_u|}$$

ここで，U はユーザ集合，H_u はユーザ u の閲覧（視聴）履歴中のアイテム集合，L_u はユーザ u への推薦リスト，$sim(i,h)$ はアイテム間の類似度関数である．多くの商用 Web サイトでは，ユーザの視聴履歴（購買履歴）を保持しているため，非常に有用な指標である．

§4 意外性の half-life utility (Half-life utility of serendipity)

村上らは，推薦システムの意外性の評価を，ユーザに提示されたアイテムへの意外性に関する評価値のデータを用い，Half-life Utility Metric を改良した式で評価している [村上 09]．具体的には，以下の式で意外性の Half-life Utility (Half-life utility of serendipity) $HLU_{serendipity}$ を計算している．

$$HLU_{serendipity} = \frac{R^{ser}}{R^{rel}_{max}}$$

$$R^{ser} = \sum_{s_j \in \Im L_i \cap Ser} \frac{1}{2^{(j-1)(\beta-1)}}$$

$$R^{ser}_{max} = \sum_{s_j \in Ser} \frac{1}{2^{(j-1)(\beta-1)}}$$

ここで，B を全体のアイテム ($|B| = M$)，B のうち対象ユーザ a_i が意外であるとしたアイテムの集合を Ser ($Ser \in B$)，対象アルゴリズム i での推薦リストを L_i ($|L_i| = N$) としている．s_j はアイテムを表し，j はアイテムの順位を表す．村上らの実験では，アルゴリズムの異なる推薦システムでアイテムを推薦し，それぞれの推薦アイテム集合を合わせたものを全体のアイテム集合としている [村上 09]．また，この実験では N は 10，M はその 2〜6 倍程度の大きさ，$\beta = 50$ としている．

この指標は，ユーザに直接に意外であるかどうかを尋ねているため，serendipity を評価する指標の中では信頼できる指標と言える．一方，オフライン評価が困難な点と，意外さの評価には揺らぎが大きくなると思われる点が，欠点である．

4·6 発見性に関する指標の利用

この章では，ユーザにとって推薦結果が目新しいものかどうかを評価することを考えてきた．最初に説明したように，提示された推薦リストのほとんど全てが知っているもので，推薦されなくても買うかどうかを決めていたものだとすると，その推薦リストへの満足度は低くなると思われる．しかし，逆に目新しいアイテムばかりが並んでいたとしても，それは必ずしもユーザの満足度向上につながるとは限らないのも事実である．

Swearingen と Sinha の実験では，ユーザはすでに馴染みのあるアイテムの推薦を好むという発見をしている [Sinha 01]．また，Ekstrand が行った新規性と嗜好の適合率，ユーザ満足度との関連を調査した実験でも，ユーザ満足度には嗜好の適合率が貢献しているものの，新規性はユーザ満足度に悪影響を与えることを示している [Ekstrand 14]．この実験は，一般的な協調フィルタリングのアルゴリズムを用いており，特にユーザを限定せずに行ったものである．そのため，この実験結果はユーザの推薦結果に対する基本性質を表していると思われる．

推薦の新規性や意外性を上げるには，どのユーザに対して行うのか，そのユーザは推薦システムを使い始めてからどれぐらい経っているのかなど，適用すべきかどうかを十分に検討してからの方が良いであろう．また，新規性や意外性を上げた推薦を行っていることを，前もって伝えておいたり，推薦結果に説明付けしたりするのも良いだろう．このようなユーザへの配慮が必要である．特にユーザが推薦システムを使い始めた段階，すなわちユーザが推薦システムが信頼に足る存在かどうかを評価している段階では，慎重になる必要がある．また，推薦システムをどのようなドメインに適用するのか，どのようなコンテキストで推薦するのか，ユーザにコスト（お金）はかかるのかなどによって，目新しさをどの程度重要視すれば良いかも考慮する必要があるであろう．

5. ま と め

本稿では，推薦システムのオフライン評価に利用できる評価指標を紹介した．推薦システムの評価は，オフライン評価とオンライン評価に大別できるが，まずはオフライン評価の利点と欠点を 9 の観点からまとめた．また，データセットにおける注意すべき特徴をまとめ，交差検定の方法を説明した．次に，推薦システムの正確性に関する評価指標を紹介した．特に，予測評価値の正確性を示すもの，ユーザの嗜好に適合する順序の正しさを示すもの，一定の大きさのリストの正確性を示すもの，順位を考慮した正確性を示すものの 4 つに分けて紹介した．嗜好の正確性に関する評価指標は，情報検索の分野で古くから実践されていたため，かなり確立したものとなっている．評価したいシステムやアルゴリズムの特徴を考慮して，評価指標を選択すればよいと思われる．

その次に，推薦の発見性に関する評価指標を紹介した．このような指標は 2000 年前後から出現し始めたもので，今でも新しい指標が提案され続けている．発見性を，多様性，新規性，意外性に分けて，それぞれで提案されている指標を紹介した．多くの指標は，評価値行列やオントロジー情報のみを用いて算出可能なものである．しかし，真に新規性や意外性の評価を行うには，本当に知らないアイテムであったのか，推薦は驚きをもたらすものであったのかを聞かなければならない．これらのデータが付与されると，多くの評価指標が提案されるようになると思われる．

本稿では，推薦システムのオフライン評価に用いられる評価指標を網羅的に紹介してきた．それぞれの評価指標は，評価の目的や評価に必要な情報，評価の単位などが異なる．評価したい推薦システムの特徴や，用いるデータセットの特徴，さらにはユーザへの推薦結果の提示方式などを考慮して，適切な評価指標を選択していただければと思う．たとえ提案手法がちょっとした工夫でしかなかったり，革新的なアイディアであるとは言えなかったりしたとしても，適切な評価指標を用い信頼に足るデータ規模で実験を行えば，推薦システムの研究分野にもたらす価値はより大きくなると思われる．本稿が，本分野の多くの研究者と，新しいビジネスやサービスを切り拓くエンジニアにとって，少しでも役に立てば幸いである．

謝 辞

本稿の執筆に当たり，オフライン評価とオンライン評価の違いを考察するための観点については，テキサス州立大学の Michael Ekstrand 氏にご助言をいただいた．また，サムスン日本研究所の倉本秀治氏には，推薦システム評価に関して企業で直面している問題についてご教示いただいた．ここに感謝の意を表したい．最後に，本稿は筆者がミネソタ大学 GroupLens Research に赴任中に執筆したものである．新たな環境に挑戦するチャンスを与えてくださった，ミネソタ大学 GroupLens Research の Joe Konstan 先生に感謝したい．

◇ 参 考 文 献 ◇

[Adamopoulos 13] Adamopoulos, P. and Tuzhilin, A.: On Unexpectedness in Recommender Systems: Or How to Better Expect the Unexpected, *ACM Transactions on Intelligent Systems and Technology*, Vol. 1, No. 1, pp. 1–51 (2013)

[Adomavicius 05] Adomavicius, G. and Tuzhilin, A.: Toward the Next Generation of Recommender Systems: a Survey of the Stateof-the-art and Possible Extensions, *IEEE Transactions on Knowledge*

and Data Engineering, Vol. 17, No. 6, pp. 734–749 (2005)

[Adomavicius 07] Adomavicius, G. and Kwon, Y.: New Recommendation Techniques for Multicriteria Rating Systems, *IEEE Intelligent Systems*, Vol. 22, No. 3, pp. 48–55 (2007)

[Adomavicius 12] Adomavicius, G. and Kwon, Y.: Improving Aggregate Recommendation Diversity Using Ranking-Based Techniques, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 24, No. 5, pp. 896–911 (2012)

[Agrawal 09] Agrawal, R., Gollapudi, S., Halverson, A., and Ieong, S.: Diversifying Search Results, in *Proc. of the Second ACM International Conference on Web Search and Data Mining (WSDM'09)*, pp. 5–14 (2009)

[Amatriain 09] Amatriain, X., Pujol, J. M., and Oliver, N.: I Like It... I Like It Not: Evaluating User Ratings Noise in Recommender Systems, *User Modeling, Adaptation, and Personalization (Lecture Notes in Computer Science)*, Vol. 5535, pp. 247–258 (2009)

[Baeza-Yates 11] Baeza-Yates, R. and Ribeiro-Neto, B.: *Modern Information Retrieval: The Concepts and Technology behind Search (2nd Edition)*, Addison-Wesley Professional (2011)

[Bell 07] Bell, R. M., Koren, Y., and Volinsky, C.: Modeling Relationships at Multiple Scales to Improve Accuracy of Large Recommender Systems, in *Proc. of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (ACM KDD'07)*, pp. 95–104 (2007)

[Bellogin 10] Bellogin, A., Cantador, I., and Castells, P.: A study of Heterogeneity in Recommendations for a Social Music Service, in *Proc. of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec'10)*, pp. 1–8 (2010)

[Bollen 10] Bollen, D., Knijnenburg, B. P., Willemsen, M. C., and Graus, M.: Understanding Choice Overload in Recommender Systems, in *Proc. of the 2010 ACM Conference on Recommender Systems (ACM RecSys'10)*, pp. 63–70 (2010)

[Bonhard 07] Bonhard, P., Harries, C., McCarthy, J., and Sasse, M. A.: Accounting for Taste: Using Profile Similarity to Improve Recommender Systems, in *Proc. of the SIGCHI Conference on Human Factors in Computing Systems (ACM CHI'07)*, pp. 1057–1066 (2007)

[Breese 98] Breese, J. S., Heckerman, D., and Kadie, C.: Empirical Analysis of Predictive Algorithms for Collaborative Filtering, in *Proc. of Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI'98)*, pp. 43–52 (1998)

[Buckley 05] Buckley, C. and Voorhees, E. M.: *Retrieval System Evaluation in TREC: Experiment and Evaluation in Information Retrieval*, MIT Press (2005)

[Burke 02] Burke, R.: Hybrid Recommender Systems: Survey and Experiments, *User Modeling and User-Adapted Interaction*, Vol. 12, No. 4, pp. 331–370 (2002)

[Büttcher 07] Büttcher, S., Clarke, C. L. A., Yeung, P. C. K., and Soboroff, I.: Reliable Information Retrieval Evaluation with Incomplete and Biased Judgements, in *Proc. of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (ACM SIGIR'07)*, pp. 63–70 (2007)

[Carbonell 98] Carbonell, J. and Goldstein, J.: The Use of MMR, Diversity-based Reranking for Reordering Documents and Producing Summaries, in *Proc. of 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (ACM SIGIR'98)*, pp. 335–336 (1998)

[Carterette 11] Carterette, B.: System Effectiveness, User Models, and User Utility: a Conceptual Framework for Investigation, in *Proc. of 34th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (ACM SIGIR'11)*, pp. 903–912 (2011)

[Celma 08] Celma, O. and Herrera, P.: A New Approach to Evaluating Novel Recommendations, in *Proc. of the 2008 ACM Conference on Recommender Systems (ACM RecSys'08)*, pp. 179–186 (2008)

[Chandar 11] Chandar, P. and Carterette, B.: Analysis of Various Evaluation Measures for Diversity, in *Proc. of the Workshop on Diversity in Document Retrieval (DDR'11)* (2011)

[Chapelle 09] Chapelle, O., Metlzer, D., Zhang, Y., and Grinspan, P.: Expected Reciprocal Rank for Graded Relevance, in *Proc. of the 18th ACM Conference on Information and Knowledge Management (ACM CIKM'09)*, pp. 621–630 (2009)

[Clarke 08] Clarke, C. L., Kolla, M., Cormack, G. V., Vechtomova, O., Ashkan, A., Büttcher, S., and MacKinnon, I.: Novelty and Diversity in Information Retrieval Evaluation, in *Proc. of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (ACM SIGIR'08)*, pp. 659–666 (2008)

[Cosley 03] Cosley, D., Lam, S. K., Albert, I., Konstan, J. A., and Riedl, J.: Is Seeing Believing? How Recommender Interfaces Affect Users' Opnions, in *Proc. of the Conf. on Human Factors in Computing Systems (ACM CHI'03)*, pp. 585–592 (2003)

[Cramer 08] Cramer, H., Evers, V., Ramlal, S., Someren, van M., Rutledge, L., Stash, N., Aroyo, L., and Wielinga, B.: The Effects of Transparency on Trust in and Acceptance of a Content-based Art Recommender, *User Modeling and User-Adapted Interaction*, Vol. 18, No. 5, pp. 455–496 (2008)

[Craswell 08] Craswell, N., Zoeter, O., Taylor, M., and Ramsey, B.: An Experimental Comparison of Click Position-bias Models, in *Proc. of International Conference on Web Search and Data Mining (ACM WSDM'08)*, pp. 87–94 (2008)

[Deshpande 04] Deshpande, M. and Karypis, G.: Item-based Top-N Recommendation Algorithms, *ACM Transactions on Information Systems (TOIS)*, Vol. 22, No. 1, pp. 143–177 (2004)

[Drenner 08] Drenner, S., Sen, S., and Terveen, L.: Crafting the Initial User Experience to Achieve Community Goals, in *Proc. of the ACM Conf. on Recommender Systems (ACM RecSys'08)*, pp. 187–194 (2008)

[Ekstrand 14] Ekstrand, M., Harper, F. M., Willemsen, M., and Konstan, J.: User Perception of Differences in Movie Recommendation Algorithms, in *Proc. of the fourth ACM Conference on Recommender Systems (ACM RecSys'10)* (2014)

[Fleder 10] Fleder, D. M. and Hosanagar, K.: Recommender Systems and their Impact on Sales Diversity, in *Proc. of the 8th ACM conference on Electronic Commerce (EC'10)*, pp. 192–199 (2010)

[Garner 60] Garner, W. R.: Rating Scales, Discriminability, and Information Transmission, *Psychological Review*, pp. 343–352 (1960)

[Ge 10] Ge, M., Delgado-Battenfeld, C., and Jannach, D.: Beyond Accuracy: Evaluating Recommender Systems by Coverage and Serendipity, in *Proc. of the of the fourth ACM Conference on Recommender Systems (RecSys'10)*, pp. 257–260 (2010)

[Gunawardana 09] Gunawardana, A. and Shani, G.: A Survey of Accuracy Evaluation Metrics of Recommendation Tasks, *The Journal of Machine Learning Research*, Vol. 10, pp. 2935–2962 (2009)

[Herlocker 04] Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T.: Evaluating Collaborative Filtering Recommender Systems, *ACM Transactions on Information Systems (TOIS)*, Vol. 22, No. 1, pp. 5–53 (2004)

[土方 04] 土方 嘉徳：情報推薦・情報フィルタリングのためのユーザプロファイリング技術, 人工知能学会誌, Vol. 19, No. 3, pp. 365–372 (2004)

[土方 07] 土方 嘉徳：嗜好抽出と情報推薦技術, 情報処理学会誌, Vol. 48, No. 9, pp. 957–965 (2007)

[Hijikata 09] Hijikata, Y., Shimizu, T., and Nishida, S.: Discovery-oriented Collaborative Filtering for Improving User Satisfaction, in *Proc. of the International Conference on Intelligent User Interfaces (ACM IUI'09)*, pp. 67–76 (2009)

[Hijikata 12] Hijikata, Y., Kai, Y., and Nishida, S.: The Relation between User Intervention and User Satisfaction for Information Recommendation, in *Proc. of the 27th Annual ACM Symposium on Applied Computing (ACM SAC 2012)*, pp. 2002–2007 (2012)

[Hill 95] Hill, W., Stead, L., Rosenstein, M., and Furnas, G.: Recommending and Evaluating Choices in a Virtual Community of Use, in *Proc. of the Conf. on Human Factors in Computing Systems (ACM CHI'95)*, pp. 194–201 (1995)

[Hosanagar 05] Hosanagar, K.: A Utility Theoretic Approach to Determining Optimal Wait Times in Distributed Information Retrieval, in *Proc. of 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (ACM SIGIR'05)*, pp. 91–97 (2005)

[Järvelin 02] Järvelin, K. and Kekäläinen, J.: Cumulated Gain-based Evaluation of IR Techniques, *ACM Transactions on Information Sys-*

tems (TOIS), Vol. 20, No. 4, pp. 422–446 (2002)

[Joachims 05] Joachims, T., Granka, L., Pan, B., Hembrooke, H., and Gay, G.: Accurately Interpreting Click-through Data as Implicit Feedback, in *Proc. of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (ACM SIGIR'05)*, pp. 154–161 (2005)

[Jones 07] Jones, N. and Pu, P.: User Technology Adoption Issues in Recommender Systems, in *Proc. of the 2007 Networking and Electronic Commerce Research Conference (NAEC'07)*, pp. 379–394 (2007)

[神嶌 07] 神嶌 敏弘：推薦システムのアルゴリズム (1), 人工知能学会誌, Vol. 22, No. 6, pp. 826–837 (2007)

[神嶌 08a] 神嶌 敏弘：推薦システムのアルゴリズム (2), 人工知能学会誌, Vol. 23, No. 1, pp. 89–103 (2008)

[神嶌 08b] 神嶌 敏弘：推薦システムのアルゴリズム (3), 人工知能学会誌, Vol. 23, No. 2, pp. 248–263 (2008)

[Kawamae 10] Kawamae, N.: Serendipitous Recommendations via Innovators, in *Proc. of the 33rd International ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR'10)*, pp. 218–225 (2010)

[Kilkki 07] Kilkki, K.: A Practical Model for Analyzing Long Tails, *First Monday*, Vol. 12, No. 5 (2007)

[Knijnenburg 12] Knijnenburg, B. P., Willemsen, M. C., Gantner, Z., Soncu, H., and Newell, C.: Explaining the User Experience of Recommender Systems, *User Modeling and User-Adapted Interaction*, Vol. 22, No. 4-5, pp. 441–504 (2012)

[Koychev 00] Koychev, I. and Schwab, I.: Adaptation to Drifting User's Interests, in *Proc. of ECML2000 Workshop: Machine Learning in New Information Age*, pp. 39–46 (2000)

[Lam 06] Lam, S. K., Frankowski, D., and Riedl, J.: Do You Trust Your Recommendations? An Exploration Of Security and Privacy Issues in Recommender Systems, *Emerging Trends in Information and Communication Security: Lecture Notes in Computer Science*, Vol. 3995, pp. 14–29 (2006)

[Lathia 10] Lathia, N., Hailes, S., Capra, L., and Amatriain, X.: Temporal Diversity in Recommender Systems, in *Proc. of the 33rd international ACM SIGIR conference on Research and development in information retrieval (ACM SIGIR'10)*, pp. 210–217 (2010)

[Loeb 92] Loeb, S. and Terry, D.: Information Filtering, *Comm. of the ACM*, Vol. 35, No. 12, pp. 26–81 (1992)

[Lops 11] Lops, P., Gemmis, de M., and Semeraro, G.: Content-based Recommender Systems: State of the Art and Trends, *Recommender Systems Handbook*, pp. 73–105 (2011)

[Manning 08] Manning, C. D., Raghavan, P., and Schütze, H.: *Introduction to Information Retrieval*, Cambridge University Press (2008)

[McNee 06a] McNee, S. M., Riedl, J., and Konstan, J. A.: Being Accurate is Not Enough: How Accuracy Metrics Have Hurt Recommender Systems, in *Proc. of Extended Abstracts on Human Factors in Computing Systems (ACM CHI'06)*, pp. 1097–1101 (2006)

[McNee 06b] McNee, S. M., Riedl, J., and Konstan, J. A.: Making Recommendations Better: An Analytic Model for Human-Recommender Interaction, in *Proc. of Extended Abstracts on Human Factors in Computing Systems (ACM CHI'06)*, pp. 1103–1108 (2006)

[Meyer 12] Meyer, F., Fessant, F., Clérot, F., and Gaussier, E.: Toward a New Protocol to Evaluate Recommender Systems, in *Proc. of Workshop on Recommendation Utility Evaluation: Beyond RMSE (RUE 2012)*, pp. 9–14 (2012)

[Moffat 08] Moffat, A. and Zobel, J.: Rank-biased Precision for Measurement of Retrieval Effectiveness, *ACM Transactions on Information Systems (TOIS)*, Vol. 27, No. 1 (2008)

[Murakami 08] Murakami, T., Mori, K., and Orihara, R.: Metrics for Evaluating the Serendipity of Recommendation Lists, *New Frontiers in Artificial Intelligence, Lecture Notes in Computer Science, Springer*, Vol. 4914, pp. 40–46 (2008)

[村上 09] 村上 知子, 森 紘一郎, 折原 良平：推薦の意外性向上のための手法とその評価, 人工知能学会論文誌, Vol. 24, No. 5, pp. 428–436 (2009)

[O'Donovan 05] O'Donovan, J. and Smyth, B.: Trust in Recommender Systems, in *Proc. of the 10th International Conference on Intelligent User Interfaces (ACM IUI'05)*, pp. 167–174 (2005)

[奥 13] 奥 健太：私のブックマーク：情報推薦システム, 人工知能学会誌, Vol. 28, No. 6, pp. 1015–1018 (2013)

[Olmo 08] Olmo, del F. H. and Gaudioso, E.: Evaluation of Recommender Systems: A New Approach, *Expert Systems with Applications*, Vol. 35, pp. 790–804 (2008)

[O'Sullivan 04] O'Sullivan, D., Wilson, D. C., and Smyth, B.: Improving the Quality of the Personalized Electronic Program Guide, *User Modeling and User-Adapted Interaction*, Vol. 14, No. 1, pp. 5–36 (2004)

[Rashid 02] Rashid, A. M., Albert, I., Cosley, D., Lam, S. K., McNee, S. M., Konstan, J. A., and Riedl, J.: Getting to Know You: Learning New User Preferences in Recommender Systems, in *Proc. of the 7th International Conference on Intelligent User Interfaces (ACM IUI'02)*, pp. 127–134 (2002)

[Resnick 94] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J.: GroupLens : An Open Architecture for Collaborative Filtering of Netnews, in *Proc. of the ACM Conference on Computer Supported Cooperative Work (ACM CSCW'94)*, pp. 175–186 (1994)

[Resnick 97] Resnick, P. and Varian, H. R.: Recommender Systems, *Comm. of the ACM*, Vol. 40, No. 3, pp. 56–89 (1997)

[Richardson 07] Richardson, M., Dominowska, E., and Ragno, R.: Predicting Clicks: Estimating the Click-through Rate for New Ads, in *Proc. of the 16th International Conference on World Wide Web (ACM WWW'07)*, pp. 521–530 (2007)

[Riecken 00] Riecken, D.: Personalized Views of Personalization, *Comm. of the ACM*, Vol. 43, No. 8, pp. 26–158 (2000)

[Robertson 97] Robertson, S. E.: The Probability Ranking Principle in IR, *Journal of Documentation*, Vol. 33, pp. 294–304 (1997)

[Sarwar 98] Sarwar, B. M., Konstan, J. A., Borchers, A., Herlocker, J., Miller, B., and Riedl, J.: Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collaborative Filtering System, in *Proc. of the 1998 ACM Conference on Computer Supported Cooperative Work (ACM CSCW'98)*, pp. 345–354 (1998)

[Sarwar 00] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J.: Analysis of Recommendation Algorithms for e-commerce, in *Proc. of the 2nd ACM Conference on Electronic Commerce (ACM EC'00)*, pp. 158–167 (2000)

[Sarwar 01] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J.: Item-based Collaborative Filtering Recommendation Algorithms, in *Proc. of ACM the 10th International Conference on World Wide Web (ACM WWW'01)*, pp. 285–295 (2001)

[Schein 02] Schein, A. I., Popescul, A., Ungar, L. H., and Pennock, D. M.: Methods and Metrics for Cold-start Recommendations, in *Proc. of of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (ACM SIGIR'02)*, pp. 253–260 (2002)

[Shani 08] Shani, G., Chickering, M., and Meek, C.: Mining Recommendations From The Web, in *Proc. of the 2008 ACM Conference on Recommender Systems (ACM RecSys'08)*, pp. 35–42 (2008)

[Sinha 01] Sinha, S., Rashmi, K. S., and Sinha, R.: Beyond Algorithms: A HCI Perspective on Recommender Systems, in *Proc. of SIGIR 2001 Workshop on Recommender Systems* (2001)

[Sinha 02] Sinha, R. and Swearingen, K.: The Role of Transparency in Recommender Systems, in *Proc. of the Extended Abstracts on Human Factors in Computing Systems (ACM CHI'02)*, pp. 830–831 (2002)

[Sparling 11] Sparling, I. and Sen, S.: Rating: How Difficult is It?, in *Proc. of the 2008 ACM Conference on Recommender Systems (ACM RecSys'11)*, pp. 149–156 (2011)

[Stone 74] Stone, M.: Cross-Validatory Choice and Assessment of Statistical Predictions, *Journal of the Royal Statistical Society B*, Vol. 36, No. 1, pp. 111–147 (1974)

[Tintarev 07] Tintarev, N. and Masthoff, J.: A Survey of Explanations in Recommender Systems, in *Proc. of the 2007 IEEE 23rd International Conference on Data Engineering Workshop*, pp. 801–810 (2007)

[Vargas 11] Vargas, S. and Castells, P.: Rank and Relevance in Novelty and Diversity Metrics for Recommender Systems, in *Proc. of the fifth ACM Conference on Recommender Systems (ACM RecSys'11)*, pp. 109–116 (2011)

[Yang 01] Yang, Y. and Padmanabhan, B.: On Evaluating Online Personalization, in *Proc. of the Workshop on Information Technology*

and Systems, pp. 35–41 (2001)
[Yao 95] Yao, Y.: Measuring Retrieval Effectiveness Based on User Preference of Documents, *Journal of the American Society for Information Science*, Vol. 46, pp. 133–145 (1995)
[Zhai 03] Zhai, C. X., Cohen, W. W., and Lafferty, J.: Beyond Independent Relevance: Methods and Evaluation Metrics for Subtopic Retrieval, in *Proc. of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval (ACM SIGIR'03)*, pp. 10–17 (2003)
[Zhang 08] Zhang, M. and Hurley, N.: Avoiding Monotony: Improving the Diversity of Recommendation Lists, in *Proc. of the second ACM Conference on Recommender Systems (ACM RecSys'08)*, pp. 123–130 (2008)
[Zhang 12] Zhang, Y. C., Séaghdha, D. Ó., Quercia, D., and Jambor, T.: Auralist: Introducing Serendipity into Music Recommendation, in *Proc. of the fifth ACM international conference on Web search and data mining (WSDM'12)*, pp. 13–22 (2012)
[Zhou 10] Zhou, T., Kuscsik, Z., Liu, J.-G., Medo, M., Wakeling, J. R., and Zhang, Y.-C.: Solving the Apparent Diversity-accuracy Dilemma of Recommender Systems, in *Proc. of the National Academy of Sciences*, pp. 4511–4515 (2010)
[Ziegler 04] Ziegler, C.-N., Lausen, G., and Schmidt-Thieme, L.: Taxonomy-driven Computation of Product Recommendations, in *Proc. of the thirteenth ACM International Conference on Information and Knowledge Management*, pp. 406–415 (2004)
[Ziegler 05] Ziegler, C.-N., McNee, S. M., Konstan, J. A., and Lausen, G.: Improving Recommendation Lists through Topic Diversification, in *Proc. of the 14th International Conference on World Wide Web (ACM WWW'05)*, pp. 22–32 (2005)

著 者 紹 介

土方 嘉徳（正会員）

1996 年大阪大学基礎工学部システム工学科卒業．1998 年同大学大学院修士課程修了．同年日本アイ・ビー・エム (株) 東京基礎研究所入社．2002 年より大阪大学大学院基礎工学研究科システム創成専攻助手．2009 年より同准教授．2014 年ミネソタ大学 GroupLens Research 客員研究員．2005 年インタラクション 2005 ベストペーパー賞，2006 年 ACM IUI Best Paper Award，DEWS2006 優秀論文賞，2011 年 WebDB フォーラム 2011 最優秀論文賞，2012 年 WebDB フォーラム 2012 優秀論文賞，2013 年インタラクション 2013 ベストペーパー賞，WebDB フォーラム 2013 優秀論文賞，情報処理学会 山下記念研究賞，各受賞．ソーシャルコンピューティング，情報推薦，テキストマイニングの研究に従事．情報処理学会，人工知能学会，ヒューマンインタフェース学会，日本データベース学会ほか会員．電子情報通信学会シニア会員．博士（工学）．