F-014

IRT を援用したコンテンツ推薦に関する考察
A study of contents recommendation method using IRT

飯田 委哉[1)]　野口 和久[2)]　伊東 栄典[3)]
Tomoya Iida　Kazuhisa Noguchi　Eisuke Ito

1　はじめに

利用者が動画や小説，画像などのコンテンツを投稿するサービス（CGM, Consumer Generated Media) が人気である．動画 CGM サイトである YouTube やニコニコ動画には毎日多数の動画が投稿されており，膨大な利用者が動画を閲覧している．多くの CGM サイトでは検索だけでなく推薦も提供される．利用者の嗜好に合うコンテンツを推薦できると望ましい．

現在よく知られている推薦手法は，大手 EC サイト Amazon.com が主におこなっている協調フィルタリング [1] である．協調フィルタリングではアイテム群全体で人気のアイテムが推薦されやすい傾向がある．推薦対象の利用者にとって人気アイテムは既知である可能性が高いため，推薦候補が利用者に有益でない可能性がある．我々は，推薦対象に未知であるけれども興味を引くアイテムを推薦する方がより有益であると考える．

本研究では，小説・楽曲・映画などのコンテンツを対象に，セレンディピティの向上を目的とするコンテンツの推薦を研究する．セレンディピティ (serendipity) とは，偶然に大発見をする幸運や，その才能という意味である．推薦されたものが，出会って初めて興味が有ると分かるものであればセレンディピティが有るといえる．

そのために以下の手法を提案する．まず多数のコンテンツ群にクラスタリングを適用し，それほど多くない数のクラスタ（同一ジャンルの集合）に分割する．次に，ある利用者に対し，セレンディピティなコンテンツを推薦するため，利用者のコンテンツ利用履歴から，利用者とクラスタの距離を計算する．その利用者が好むジャンルのクラスタは距離が近くなり，好まないジャンルのクラスタは距離が遠くなる．普段好まないジャンル（クラスタ）の高品質コンテンツを推薦すれば，セレンディピティが向上する．ただし推薦対象となるコンテンツの品質を算出する必要がある．

コンテンツの品質評価に，学力試験の理論である IRT を援用する．現在，TOEFL [2] などの大規模学力試験で項目反応理論 (IRT, Item Response Theory) [3, 4] による問題分析および受験者の評価が行われている．IRT では，試験問題への正答・誤答のデータを用いて，受験者の能力を測ると同時に，試験問題の難易度も測る．同じ正解数でも解答した問題の難易度により異なる成績が出るため加点方式より細かく能力推定ができる．IRT が対象とする受験者と試験問題の関係を，利用者とコンテンツとの関係と捉えると，ある分野に対する利用者の目利き度合い（能力）と，コンテンツの良し悪しを測定できるであろう．

コンテンツ評価に IRT を用いるためには，評価の空白への対処が必要である．IRT が対象とする学力試験では全受験者は良い点数を得ようとするため，空白（回答なし）は誤答と扱われる．一方，コンテンツの場合，利用者は全てを閲覧していないし，閲覧しても評価値を与えるわけではない．評価が無い空白コンテンツに対し，適切な値を設定しなければ IRT を適用出来ない．今回は空白値を埋めるため，協調フィルタリング手法での値推定を検討する．

1)　九州大学大学院システム情報科学府
2)　九州大学大学院統合新領域学府
　　ライブラリサイエンス専攻
3)　九州大学情報基盤研究開発センター

2　関連研究

商品（アイテム）の推薦は 20 年以上前から様々な研究が行われており，実用化も進んでいる．推薦には商品内容の情報を用いる内容ベースの推薦と，利用者の商品評価や購入履歴を用いるソーシャルベースの推薦がある．ソーシャルベースの推薦手法で最も有名なのは協調フィルタリング [5] であろう．協調フィルタリングは多く

過去 20 年間における Amazon での推薦手法に関する報告 [1] によると，当初 Amazon では利用者 (user) ベースの協調フィルタリング手法で推薦していたものの，近年では商品 (item) ベースの推薦に遷移したと報告している．デジタルカメラからメモリカードは推薦できるけれど逆は推薦しない，という依存関係や一方向性も検討されている．商品推薦においては，商品のジャンルや，価格が重要との報告もあり，安い商品は推薦で購入されやすいが，高い商品はそうではないと述べている．

協調フィルタリングから派生した推薦手法としては，SVD (Singular Value Decomposition) や Matrix Factorization などが有る．Steffen Rendle は Factorization Machines (FM) を提案し [6]，それを推薦に用いる方法も提供している．FM は，協調フィルタリングにおける次元を削減することで良い推薦を行う手法である．

Mouzhi らの研究 [7] は，予測精度に加えてセレンディピティが推薦に重要であると述べている．Himan らの研究 [8] では，人気がロングテール型の分布に従うアイテムの中で，下位 80%以下のロングテール部に属したアイテムに着目し，人気の低いアイテムを推薦することでセレンディピティを上げる手法を提案している．しかしながら，下位 80%ロングテール部に属するアイテムは，コールドスタートと言われる品質は良いものの知られてないため人気が低いアイテムと，そもそも低品質なため人気が出ないアイテムの 2 種類が存在する．そのため Himan らの手法では必ずしも推薦の満足度が向上するとは言えない．

3　クラスタリングによる同種集合への分割

本研究では，セレンディピティを高めた推薦を目標としている．利用者には嗜好がある．その嗜好と異なるコンテンツを推薦すればセレンディピティが高まるであろう．

小説や音楽などのコンテンツは多数かつ多様である．それらを，多くない数の，同一ジャンルの集合に分割す

る．そのためにクラスタリングを適用する．

3.1 Word2Vec, Doc2Vec によるベクトル化

クラスタリング手法は，階層的手法と，非階層的手法の 2 種類がある．どちらの手法でも，適用対象を数値のベクトルで表現する必要がある．小説や音楽などのコンテンツをベクトル化の対象として，題名，作者名，キーワード，説明などのメタデータを利用する．コンテンツの内容データではなく，メタデータだけをベクトル化の対象とすることで，小説・漫画・動画・楽曲などの全コンテンツに適用できる．

メタデータのベクトル化に，Word2Vec および Doc2Vec を用いる．Word2Vec は Tomas Mikolov らの開発した分散表現を生成する手法で，各単語を高次元のベクトルで表現する [9]．Word2Vec では，文章中に含まれる単語の出現数を利用する Countinuous Bag-of-Words モデルと，文章中に含まれる単語の並びから単語の出現確率を利用する Skip-gram モデルの両方の学習モデルを用いて，Hierarchical Softmax 及び Negative Sampling によって高速化を行っている．同様の手法を文章について使用したものに Doc2Vec [10] が存在する．Doc2Vec は文書の分散表現を生成できるため，文章をベクトル化できる．

3.2 クラスタリング手法

クラスタリング手法は，階層的手法と，非階層的手法の 2 種類がある．K-means などの非階層的手法の方が計算が早いため，コンテンツ数が多い場合には適している．しかしながら，適切なクラスタ数が一意に決まらない．

階層化手法は，計算が遅い（n 個のとき $O(n^2)$）ため，コンテンツ数が多い時は適していない．しかしながら，クラスタ数を次善に決める必要が無い．また階層の段数を指定することで，クラスタの要素数を柔軟に変更できる利点がある．

推薦対象とするコンテンツの数に応じて適切なクラスタリング手法を適用したい．クラスタリングの際には，コンテンツ数だけでなく，そのコンテンツの利用者数も検討したい．例えば，利用者のコンテンツへのブックマーク数が，どのクラスタもだいたい均等になるようにしたい．

4 IRT

本研究では，コンテンツの品質評価に IRT の援用を検討する．項目反応理論 (IRT, Item Responce Theory) は，TOEFL などの資格試験で成績評価に採用されているテスト理論である．従来テストの評価法では加点方式が用いられてきた．項目反応理論は同じ正解数でも解答した問題の難易度により異なる成績が出るため加点方式より細かく能力推定ができる．しかし，項目反応理論は統計手法を用いて能力推定を行うため計算が複雑で加点方式より使用が難しい．

4.1 1PLM IRT

パラメタが 1 つのモデル，1PLM (one parameter logistic model) の IRT を考える．項目パラメータである困難度 b_j と，受験者の能力値パラメータ θ_i を持つロジスティックモデルを考え，このモデルに正答確率 $P_j(\theta_i)$ が従うと考える．ロジスティックモデルでは，あるテスト受験者 i が問題 j に正解する確率 $P_j(\theta_i)$ とし，問題特性のモデルを式 1 で表現する．

$$\mathrm{P}_j(\theta_i) = \frac{1}{1+\exp(-D(\theta_i - b_j))} \tag{1}$$

右辺の D は尺度因子と呼ばれ $D = 1.7$ とする．θ_i は受験者 i の能力値パラメータと呼ばれており，この値が大きいほど受験者の能力 (成績) が高いことを示す．b_j は問題 j の困難度と呼ばれており，この値が大きいほどその問題は難問となり全ての受験者の正答確率が低くなる．項目パラメータは他にも項目 j の識別力 a_j や当て推量 c_j を含むモデルがある．本論文では項目パラメータは困難度 b_j のみの 1PLM のみ扱う．

4.2 IRT での能力推定・困難度推定

次に尤度推定による IRT のパラメタ値推定を述べる．尤度推定により能力値パラメータの推定が行われる．

受験者 $i = 1, 2, \cdots, I$ が，問題数 $j = 1, 2, \cdots, J$ のテストを受験したとき，テストの成績に対する実現確率 (尤度) は以下で与えられる．

$$\mathrm{L}(\theta_i, b_j|\delta_{ij}) = \prod_{i=1}^{I}\prod_{j=1}^{J} \mathrm{P}_j(\theta_i)^{\delta_{ij}}(1-\mathrm{P}_j(\theta_i))^{1-\delta_{ij}} \tag{2}$$

ここで θ_i, b_j はすべて未知である．この尤度 $L(\theta_i)$ を最大にする項目パラメータ b_j と能力パラメータ θ_i を求める．この方法を最尤推定法という．このとき，δ_{ij} は受験者 i が問題 j に解答した正誤 (反応) である．

$$\delta_{ij} = \begin{cases} 1 & (Correct) \\ 0 & (False) \end{cases} \tag{3}$$

項目反応理論における最尤推定法は尤度 $L(\theta_i)$ の最大化問題と考えられる．以下の式の最大化問題を解くことでパラメータ推定が行われる．

$$\max \quad \mathrm{L}(\theta_i, b_j|\delta_{ij}) \tag{4}$$

$$s.t. \quad \theta_i, b_j \in \mathbf{R}^n \tag{5}$$

パラメータ推定では，尤度関数 L の対数である対数尤度関数 lnL を最大にする値を求める．つまり，以下を満たすパラメータの最尤推定値 $\hat{\theta_i}, \hat{b_j}$ を求める．

$$\begin{cases} \dfrac{\partial \ln \mathrm{L}(\theta_i, b_j|\delta_{i,j})}{\partial \theta_i} = 0, \ i = 1, \ldots, I \\ \dfrac{\partial \ln \mathrm{L}(\theta_i, b_j|\delta_{i,j})}{\partial b_j} = 0, \ j = 1, \ldots, J. \end{cases} \tag{6}$$

5 IRT によるコンテンツ品質推定

IRT を用いたコンテンツ品質を定量化について述べる．学力試験における正答・誤答を表す行列を，コンテンツ推薦の場合，コンテンツと読者の行列で考える．

利用者 i が，コンテンツ j をブックマークしているか，評価値を与えている場合，$\delta_{ij} = 1$ とする行列を考える．この行列に 1 パラメータの IRT として計算すると，以下のパラメータを得る．

- 読者 i の能力 θ_i
- コンテンツ j の理解困難度 b_j

推薦手法において，読者の能力はそのコンテンツが属するジャンルに精通しているかどうかを表し，コンテン

ツの理解困難度はコンテンツが属するジャンル特徴の表出度合いを表す．高い能力を持つ読者が評価するコンテンツや理解困難度が高いコンテンツはよりそのジャンルらしいコンテンツであると言える．

5.1 空白値の対応

IRT を能力の推定に適用する場合，ユーザとアイテムは空白値ができるだけ少ないことが望ましい．一方で，コンテンツ推薦に用いられるアイテムとユーザの評価行列は非常に疎なデータであることが多い．そこで本研究では空白値を埋めるため，協調フィルタリング手法での値推定を用いる．

協調フィルタリングは類似度の高いユーザを探し，類似度の高いユーザが評価しているアイテムを推薦する手法である．類似度はコサイン類似度を用いて計算する．ユーザ A とユーザ B のコサイン類似度は以下の式で計算される．

$$similarity(\vec{A},\vec{B}) = cos(\vec{A},\vec{B}) = \frac{\vec{A}\cdot\vec{B}}{\|\vec{A}\| * \|\vec{B}\|} \tag{7}$$

このとき A と B は各ユーザのアイテムの評価に関する 1 次元行列である．これをユーザ・アイテム行列全体に適用し，ユーザの類似度を求める．この類似度を重み w として，利用者 a がコンテンツ y に与える評価値 $\hat{r}_{ay}$ は以下の式で推定することができる．

$$\hat{r}_{ay} = \bar{r}_a + \frac{\sum_{x\in X_y} w_{ax}(r_{xy} - \bar{r'}_x)}{\sum_{x\in X_y} w_{ax}} \tag{8}$$

このとき，X_y はアイテム y を評価済みの利用者の集合，r_{xy} はユーザ x がアイテム y に与えた評価値，$\bar{r}_a$ はユーザ a が与えた評価値の平均値，$\bar{r'}_x$ はユーザ x の評価値の平均値である．

5.2 IRT による利用者と小説への数値付け

IRT は Python モジュールとして提供されている pyirt を利用する．本モジュールは計算量の関係上，ある程度データを小さくして実行させる必要がある．各クラスタごとに，小説とユーザのブックマーク関係行列を行，列ともにブックマーク数で降順に並べ替える．その行列からブックマーク数が 10 以上の利用者を含む行列を抽出し，各小説に対して項目困難度 b を，各利用者に能力パラメタ θ を算出する．

次に算出された能力パラメタ θ と項目困難度 b を利用し，各データに重み付けを行う．小説 i の項目困難度 b_i と，利用者 j の能力パラメタ θ_j をブックマークの有無を表す δ_{ij} にかけた $b_i\theta_j\delta_{ij}$ を，小説 i に対する利用者 j の評価値 r_{ij} としたい．

5.3 利用者とクラスタの距離

セレンディピティの高いコンテンツを推薦するため，利用者の嗜好と異なるクラスタのコンテンツを推薦したい．そのために利用者とクラスタの距離または類似度を定義する必要がある．

最も簡単な距離としては，利用者 u のブックマークが利用できる．クラスタが含むのコンテンツのうち，利用者 u がブックマークしているコンテンツ数を距離とする．式で書くと，利用者 i とクラスタ C の距離 $d(i,C)$ は以下となる．

$$d(i,C) = \sum_{j\in C} \delta_{i,j}$$

他にも，単語を使う方法もある．利用者がブックマークしている小説のメタデータをあつめ，その利用者が重要視する単語を TFIDF 値などで算出する．クラスタを 1 つの文書として扱えば，クラスタの特徴語を TFIDF などで算出できる．クラスタ C の文書（メタデータ）と，それ以外の文書を，単語で区別する SVM を作れば，重みの大きな単語がクラスタの特徴語として扱うことも可能である．利用者が重要視する単語と，クラスタの特徴語が分かれば，距離や類似度を計算できる．

$d(i,C)$ を算出することで，利用者から遠いクラスタを決めることができる．利用者から遠いクラスタ内のコンテンツから，評価値の高いものを推薦する．

6 「小説家になろう」での実験

推薦対象の CGM コンテンツとして「小説家になろう」(`http://syosetu.com/`) の小説群を考える．「小説家になろう」は，株式会社ヒナプロジェクトが提供する小説投稿サイトである．利用者登録の後，無料で小説をサイトで公開できる．2004 年のサイト開設当初は個人サイトとしての運営されていた．その後のアクセス増加により，2008 年からグループによる運営に移行し，2010 年に正式に法人化した．Wikipedia [11] によると，2014 年 12 月時点のアクセス数は月間約 9 億 5000 万 PV，ユニーク利用者数は 400 万人である．また 2018 年 1 月 31 日，登録者数が 1,185,453 人，掲載小説数は 542,291 作品である．このサイトの小説は「なろう小説」と呼ばれている．「なろう小説」の一部人気が出たものは，紙の小説として出版されたり，マンガやアニメの原作になることもある．

6.1 小説メタデータとブックマーク収集

サイトから小説メタデータと，利用者の小説ブックマークを集めた．

サイトの運営社は，小説データを取得するための Web API（なろう API）を提供している．この Web API を用いて，全小説のメタデータを取得するクローラーを python 言語で作成した．小説のメタデータは JSON または YAML 形式提供されている．集めたデータは，整形後，DB に格納して分析に利用する．

次に利用者のブックマーク収集した．「小説家になろう」では，利用者は自分のブックマークした小説リストを公開可能である．そこで，全利用者のブックマークページを集める．100 万人以上の全利用者のブックマークページの収集は困難であるため，1 つでも小説を投稿している作者である利用者のブックマークを集めた．作者である ID 数は約 8 万である．利用者 ID を指定することで，利用者 ID のブックマーク集の HTML ページを閲覧できる．そこで，利用者 ID に対するブックマークページを集めるクローラーを python 言語で作成した．

また，集めたデータを整形し，利用者と小説のブックマーク関係の行列を抽出した．利用者 i が小説 j をブックマークしているかを δ_{ij} とすると，ブックマークしている場合 $\delta_{ij} = 1$，そうでない場合は $\delta_{ij} = 0$ である．

6.1.1 小説メタデータのベクトル化

Word2Vec や Doc2Vec を用いる場合，単語を適切なベクトルで表現するための学習データが必要である．小説のあらすじから改行を除いて一行の文章とし，Doc2Vec に適用する学習データ（コーパス）とした [12]．Python 用の自然言語処理及び機械学習モジュール群である

gensim [13] の Doc2Vec を使い，学習用データから各あらすじの分散表現 (100 次元ベクトル) を生成する．

6.1.2 Ward 法によるクラスタリング

階層的クラスタリングでは，クラスタリングされていない N 個のデータから，類似度の高い順に融合して次第に大きなクラスタを作り，最終的には N 個のデータを一つのクラスタに統合する．統合過程は，樹状図（デンドログラム）と呼ばれる木の形で表現できる．デンドログラムの階層構造を見ることで，まとまりの良いクラスタに分割できる．

Ward 法は，階層的クラスタリングにおける類似度の定義の一つである．クラスタ A と B の距離を，それらを融合した時のクラスタ内の変動の増加分 $D(A,B)$ を以下で定義し，距離の小さなクラスタから統合していく．

$$D(A,B) = \sum_{x \in A,B} d(x,\mu_{AB})^2 - \left(\sum_{x \in A} d(x,\mu_A)^2 + \sum_{x \in_B} d(x,\mu_B)^2 \right) = S_{AB} - (S_A + S_B) \quad (9)$$

$d(x,y)$ はユークリッド距離，μ_{AB} はクラスタ A と B を融合したクラスタの平均ベクトル，μ_A と μ_B はクラスタ A と B それぞれの平均ベクトルである．S は平均からの距離（偏差）の 2 乗和，つまり変動である．

Python のモジュール Scipy [14] で Ward 法クラスタリングが可能である．クラスタ数が 2 つの時の距離の 15% を閾値として，クラスタ間の距離が閾値を超えるまでを 1 つのクラスタとする予定である．

7 おわりに

本論文では，セレンディピティ性を配慮したコンテンツ推薦を行うための手法について検討した．多数のコンテンツ群を，同一種類（ジャンル）の小集合に分割する．ある利用者の嗜好から遠いクラスタ内の，高品質コンテンツを推薦することで，セレンディピティが高い推薦が可能であると考えている．

コンテンツ群を，同一種類（ジャンル）の小集合に分割するためのクラスタリング手法について述べた．次に，クラスタ内のコンテンツの品質評価方法として IRT (項目反応理論）の援用を考えた．クラスタ内のコンテンツについて IRT による小説の項目困難度と，利用者の能力パラメタを算出する手法を示した．また，IRT の算出結果を用いて，各コンテンツの品質の評価値を定義した．利用者の嗜好を測るため，利用者とクラスタとの距離や類似度を検討した．

実験として，「小説家になろう」の小説と利用者を用いることを検討した．Python 言語で自作したクローラーで小説メタデータと，利用者のブックマークを収集した．収集データを，後の分析のために整形した．小説メタデータを Doc2Vec を用いてベクトル化し，Ward 法を用いてベクトルをクラスタリングする手法を検討した．

今後はまず，クラスタリングの結果を評価する．同一種類（ジャンル）に分割されているかを調査する．IRT によるコンテンツの品質評価手法は提案したものの，実際のデータには適用していない．提案手法によるコンテンツ品質についても調査する．これらの結果から，提案した手法でセレンディピティの度合いが高いコンテンツが推薦されるのかも調査する．他にも，多くの研究者に使われているデータセットを用いて，他の推薦手法と比較し推薦の多様性や精度など品質面の評価を行いたい．

参考文献

[1] Smith, B. and Linden, G.: Two Decades of Recommender Systems at Amazon.com, *IEEE Internet Computing*, Vol. 21, No. 3, pp. 12–18 (2017).

[2] Jang, E. E. and Roussos, L.: An investigation into the dimensionality of TOEFL using conditional covariance-based nonparametric approach, *Journal of Educational Measuremen*, Vol. 44, No. 1, pp. 1–21 (2007).

[3] Baker, F. B. and Kim, S.-H.: *Item response theory: Parameter estimation techniques*, CRC Press (2004).

[4] 加藤健太郎，山田剛史，川端一光：R による項目反応理論，オーム社 (2014).

[5] Resnick, P. and Varian, H. R.: Recommender systems, *Communications of the ACM*, Vol. 40, No. 3, pp. 56–58 (1997).

[6] Rendle, S.: Factorization Machines, *ICDM '10*, pp. 995–1000 (2010).

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

[8] Abdollahpouri, H., Burke, R. and Mobasher, B.: Controlling Popularity Bias in Learning to Rank Recommendation, *ACM RecSys17 (Proceedings of the Eleventh ACM Conference on Recommender Systems)*, RecSys '17, pp. 42–46 (2017).

[9] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. and Dean, J.: Distributed Representations of Words and Phrases and Their Compositionality, *Proceedings of the 26th International Conference on Neural Information Processing Systems*, NIPS'13, Vol. 2, USA, Curran Associates Inc., pp. 3111–3119 (2013).

[10] Le, Q. and Mikolov, T.: Distributed Representations of Sentences and Documents, *Proceedings of the 31st International Conference on Machine*, pp. 1188–1196 (2014).

[11] Wikipedia: 小説家になろう in Wikipedia, https://ja.wikipedia.org/wiki/%E5%B0%8F%E8%AA%AC%E5%AE%B6%E3%81%AB%E3%81%AA%E3%82%8D%E3%81%86.

[12] 佐嘉田悠樹，伊東栄典：CGM 百科辞典を用いた利用者投稿動画クラスタリング，平成 29 年度電気・情報関係学会九州支部連合大会，pp. 544–545 (2017).

[13] Gensim: gensim topic modeling for humans, https://radimrehurek.com/gensim/.

[14] Jones, E., Oliphant, T., Peterson, P. and et al.: SciPy: Open source scientific tools for Python.