セレンディピティを考慮したCGM小説推薦

飯田, 委哉
九州大学

伊東, 栄典
九州大学

出版情報：2018-11-23. The Japanese Society for Artificial Intelligence
バージョン：
権利関係：

セレンディピティを考慮したCGM小説推薦

Recommendation of CGM novels considering serendipity.

飯田 委哉[1*]　伊東栄典[1]

[1] 九州大学
[1] Kyushu University

Abstract: Recently, CGM (Consumer Generated Media) services become popular. Although a huge amount of contents have been posted to CGM site, but only a few contents are selected and viewed by users. This phenomenon are called as very short head and very long tails. Some researcher mentioned that recommendation algorithms made influence to user's content selection strongly. Serendipitous recommendation is necessary for healthy cultural glowth of CGM contents. In this paper, we propose a new serendipitious recommendation method for CGM novels. We apply clustering to novels to divide into middle size clusters, and contents are similar in the same cluster. For clustering, each content is represented into a vector using doc2vec. Next, we calculate the distance from the user ' s preference (bookmark) to each clusters. We believe that contents in near but not nearest cluster are better for serendipitious recommendation. We apply our method to web CGM novels in syosetu.com, and we also construct a web based recommendation system.

1　はじめに

動画や小説，イラストなどのコンテンツを利用者が投稿するサービス（CGM, Consumer Generated Media）が人気である．動画 CGM では Youtube やニコニコ動画，小説であればカクヨムや小説家になろう，イラストでは pixiv などが代表的である．これらのサイトはいずれも 2000 年代中頃にリリースされてから現在に至るまで多数のコンテンツが投稿されて続けている．また，日別作品投稿数はサイトの人気，成熟と共に増加している．一方で，利用者が 1 日にサイトに利用できる時間には大きな変化がない．時間あたりの満足感を最大化するには，自分が満足できる期待値が大きいと考えられるコンテンツしか触れられなくなる．具体的には，ランキングページに掲載されるような多数の利用者に人気を得ているコンテンツもしくは，自分が好んでいると自覚しているジャンル・カテゴリのコンテンツに利用する時間が多数を占め，それ以外のコンテンツに利用する時間が縮小してしまう．多種多様なコンテンツが投稿されることが CGM において，未開拓の分野を生み出してしまうのは機会の損失であると考える．そこで一定期間 CGM を利用し，閲覧コンテンツが固定化されてきている利用者を対象に，新規コンテンツに触れる機会を増加させることを考える．

新たなコンテンツとの出会いを促す方法として推薦が考えられる．こちらから”気の利いた”推薦を行い興味を引くコンテンツを提示できれば，利用者がコンテンツを楽しむ時間を奪うことなく新規コンテンツに触れる機会を生み出すことが可能である．

そこで本研究では，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手法の確立を目標とする．推薦手法として以下の手法を提案する．

近い種類のコンテンツを集めたクラスタに分割する．そのためにコンテンツをベクトルで表現し，ベクトルを用いてクラスタリングを行う．利用者の興味・関心データと分割したクラスタとの距離を計算し，利用者の興味・関心に近いクラスタと遠いクラスタを決定する．利用者が普段触れないものの，利用者の興味・関心に近いコンテンツを多く含む，利用者の興味・関心から少し離れたクラスタのコンテンツを推薦する．本研究では，Web 上の CGM 小説サイト「小説家になろう」の小説を対象に推薦手法および推薦システムの構築を目指す．

本論文の構成を述べる．2 章では関連研究について述べる．3 章では対象にした小説家になろうとそのメタデータの収集について述べる．4 章では推薦のおおまかな方針について述べる．5 章ではクラスタリングの手法の検討について述べる．6 章では構築した推薦システムとその評価について述べる．最後に 7 章でまとめと今後の課題について述べる．

*連絡先：九州大学
t.iida.630@s.kyushu-u.ac.jp

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 [3] であろう．

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 [4]，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 [5] 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 [2] 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　小説家になろう

本研究では Web 上の CGM 小説サイト「小説家になろう」の小説群を推薦対象とする．以下でサイトの概要とデータの収集について述べる．

3.1　小説家になろう

「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 (`http://syosetu.com/`) 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 [6] 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．

3.2　なろう API

利用者が小説を閲覧するか否か判断する場合，小説の内容（文章）で判断することは少なく，小説を説明するメタデータで判断することが多い．メタデータには題名，作者名，あらすじ，ブックマーク数などの情報が含まれる．本研究でも，小説の推薦や自動分類に，小説のメタデータを利用する．

「小説家になろう」のサイトを運営しているヒナプロジェクト社は，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．2018 年 6 月 15 日時点に収集した小説数は 574,260 件である．

各小説のメタデータは JSON または YAML 形式で記述されており, メタデータは構造を持っている．取得したメタデータの項目および説明を表 1 に示す．後の分析に利用するため，収集した JSON 形式のメタデータは，整形して SQLite3 の DB として格納した．

3.3　ブックマークデータの収集

「小説家になろう」では利用者アカウントを登録することで，自分が気に入った小説のブックマークや，小説への評点付与，小説への感想投稿を行うことができる．利用者が公開したブックマークはオンラインで閲覧可能である．一つのブックマークには 400 件までの小説を保持でき，一人の利用者は最大 10 個のブックマークを保持できる．ブックマーク毎に他者への閲覧可否を設定できる．

推薦のために利用者のブックマークを収集する．なろう API にはブックマーク取得の機能が無い．そのため HTML で書かれたブックマークデータを Web サイトから順次取得するクローラーを Python 言語で作成した．

利用者のブックマークページの URL を機械的に作成して総当たりアクセスし，利用者 ID と小説 ID である ncode との組みを収集した．ただし最大 10 個ある

表 1: メタデータ一覧

項目名	説明
ncode	小説固有の ID
title	タイトル
writerID	著者の ID
writer	著者名
story	あらすじ
big_genre	大ジャンル
genre	小ジャンル
keyword	著者のみが設定できる小説タグ
general_firstup	初投稿日時
general_lastup	最新話投稿日時
novel_type	長編か短編か
isend	完結済みかどうかのフラグ
general_point	総合得点
fav_novel_cnt	小説のブックマーク数
review_cnt	レビューの総数
all_point	評価点
all_hyoka_cnt	評価者数
sasie_cnt	本文中の挿絵の数
taken_data	データの取得日時

ブックマークのうち，最初のブックマークのみ収集した．各利用者の最大ブックマーク小説数は 400 件になる．2018 年 6 月 1 日時点の利用者数 623,123 人，ブックマーク 29,768,817 件分のデータを収集した．集めたブックマークデータも SQLite3 の DB に格納した．

4 推薦の方針

本研究では web 小説投稿サイト「小説家になろう」を対象に推薦を行う．この推薦について方針を述べる．web 小説をよく読む利用者を推薦の対象とする．具体的には，ランキングを頻繁に利用し自分の好きなジャンルがある程度決まってきている利用者である．これらの利用者に，利用者にとって未知でありかつ興味があるジャンルの小説を推薦する．

手法は以下の通りである．近い種類のコンテンツを集めたクラスタに分割する．そのためにコンテンツをベクトルで表現し，ベクトルを用いてクラスタリングを行う．利用者の興味・関心データと分割したクラスタとの距離を計算し，利用者の興味・関心に近いクラスタと遠いクラスタを決定する．利用者が普段触れないものの，利用者の興味・関心に近いコンテンツを多く含む，利用者の興味・関心から少し離れたクラスタのコンテンツを推薦する．

5 クラスタリングによる同種集合への分割

利用者に新たなコンテンツとの出会いを提供し興味を広げることを目的とした場合，利用者がよく知っている分野とそうでない分野を明確に区別する必要がある．この時，CGM が設定しているカテゴリやジャンルを利用することが考えられる．しかし，CGM サイトが設定したカテゴリやジャンルはそもそも複合し，その中に明確な小カテゴリを内包しているものもある．複数のカテゴリやジャンルにまたがって存在しているような分野も存在するため，既成のジャンルやカテゴリでは不足した部分がある．また，正確にコンテンツを分割できれば利用者の興味があるものとその周辺がはっきりと分かるようになる．よって，本研究ではコンテンツ集合をクラスタに分割し，その後利用者の興味・関心にマッチしたクラスタからコンテンツを推薦する．

収集したデータを用いて小説をクラスタリングを行う．コンテンツをクラスタリングするために，何らかの形でコンテンツをベクトルで表現する必要がある．ベクトル化には大きく分けて内容ベースとソーシャルベースの 2 通りがある．内容ベースはコンテンツの内容そのものをベクトル化する手法である．小説はジャンルやあらすじ，タイトルなど本文に基づくもの，イラストなどの画像は付随するタイトルやキャプションの他，画像そのもの，動画は再生時間やサイズ，各フレームごとに表示される画像を利用する．ソーシャルベースは利用者の評価から間接的にコンテンツを表現する手法である．例えば，あるジャンル A を高く評価する人によって高い評価を受けているコンテンツはジャンル A の要素を含んでいるだろうと類推する形である．ニコニコ生放送では過去に閲覧した放送を元にソーシャルベースで生放送をラベル付けし，推薦を行なっている．[7]

本研究ではまず内容ベースに焦点を当ててベクトル化を行う．理由は以下の 2 点である．一つは対象コンテンツが web 小説であるため，動画などよりは内容ベースでもベクトル化が容易であることである．もう一つはサイトのシステム上，無関心なのか読んだ上で自分に合わないとしているかの判別がつきづらいことである．

そこで今回はなろう API を用いて収集したメタデータ，特に本文に実際関わりのある文字情報，具体的にはあらすじやキーワードを利用してベクトル化を行う．ベクトル化には Doc2Vec を利用した．

次にベクトル化したデータを用いてクラスタリングを行う．本研究では Ward 法を利用する．Ward 法はデンドログラムで表現できるように，階層的にクラスタリングを行うことでクラスタ間の距離を掴みやすいためである．

5.1 Doc2Vec によるベクトル化

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 [8]．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 [9] 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．

5.2 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．

$$\begin{aligned} D(A,B) &= \sum_{x\in A,B} d(x,\mu_{AB})^2 - \left(\sum_{x\in A} d(x,\mu_A)^2 + \sum_{x\in B} d(x,\mu_B)^2\right) \\ &= S_{AB} - (S_A + S_B) \end{aligned} \tag{1}$$

$d(x,y)$ 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．

5.3 クラスタリングの評価

クラスタリングの概要について述べた．利用者の興味に合わせた推薦を行うため，できる限り正確にコンテンツを分割する必要がある．一方で本クラスタリングは正例がないため，クラスタリングが正しく行われたかを評価することが難しい．そこでクローリングした小説からテストデータとして小説を抽出した．抽出したテストデータについて，ジャンル・あらすじ・キーワードを基に，筆者が手作業でクラスタリングを行い正例を作成した．手作業でのクラスタリングのための Web CGI も作成した．その際，各クラスタの要素数ができる限り均等になるように配慮した．手作業で行った結果のクラスタ数は 72 個となった．

手作業で作成した正例を用いて機械によるクラスタリング結果を評価する．評価にはクラシフィケーションエラー E_c を用いる．クラシフィケーションエラー E_c は入力データ要素全体に対して誤って分類された要素数の割合である．クラスタリングアルゴリズムが要素数 $s_1,\ldots,s_i,\ldots,s_k$ の k 個のクラスタを出力し，クラスタ C_i の最多共通ラベルを主クラスラベルとする．また，クラスタ C_i の主クラスラベルを持つデータ要素数が m_i であるとき，クラシフィケーションエラー E_c は以下の式によって定義される．ここで N は入力データ要素数である．

$$E_c = \frac{\sum_{i=1}^{k}(s_i - m_i)}{\sum_{i=1}^{k} s_i} = \frac{\sum_{i=1}^{k}(s_i - m_i)}{N} \tag{2}$$

テストデータは以下の条件で 1009 件の小説を抽出した．

- 第 1 話が 2010 年 1 月 1 日から 2014 年 12 月 31 日の間に投稿された小説
- 運営が設定している 21 ジャンルそれぞれについて各年の人気上位 10 作品

このテストデータを利用してベクトル化の際に最も有効な単語の抜き出し方を考察する．

5.4 ベクトル化

クラスタリングを行う前に，Doc2Vec を用いてベクトル化が必要である．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．全小説 574260 件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 [10]．また，先ほどのコーパスにキーワードを加えたものも学習データとして作成し比較を行う．このとき，ストップワードの除去を行う．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 [11] 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．その後，Python のモジュール Scipy [12] で Ward 法クラスタリングを行う．クラスタ数が 2 つの時の距離との比を閾値とし，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比の値はクラスタ数が 65 ～ 74 件に収まるように設定した．

コーパスは以下のように作成した．小説のあらすじのみあるいはあらすじとキーワードを利用する．そのそれぞれについて，カタカナは全角にし，英数字をすべて半角・小文字に変換する．形態素解析エンジン MeCab を利用し，動詞，形容詞，名詞，副詞を取り出し，ス

トップワードの除去を行った．あらすじのみ，キーワードとあらすじの２種類に含まれる単語をベクトル化しクラスタリングを行う．これに対し，代名詞や非自立語，数詞や人名などのストップワードを除去する処理と頻度の少ない単語を取り除く処理を適応する．適応には，どちらも行わない，片方のみ行う，両方行うの各組み合わせでクラスタリングを行い，E_c で比較を行った．結果どの場合においても，E_c の値は 0.865 ～ 0.880 の間で大きく変化しなかった．

クラスタ数を 200 程度とすると，E_c が大きく下がることから小説ペアもしくは 4 つ以下の組み合わせの場合だと正例に近いクラスタができていると考えられる．正例の各クラスタが特定のジャンルに偏っているため，ジャンルの情報もベクトル化の際に必要であるだろう．

一方で，作成されたクラスタは一見見当違いに見えても何らかの特徴によって距離が近い小説の集合となっている．よって，その中に利用者の好む小説，具体的にはブックマーク済みの小説が含まれていれば，それ以外の小説には「利用者の興味から近すぎず遠すぎない小説」が存在していると言える．そこで，実際に推薦を行い，推薦に有効なクラスタリングであるかの評価を行う．

6　推薦システムの構築

6.1　推薦システムの構成

利用者のブックマークデータから各クラスタとの類似度を計算し，一定の距離にあるクラスタの作品を推薦するシステムを構築する．推薦システムは以下の図 1 のように設計する．

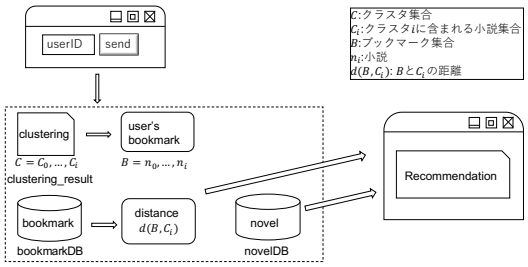

図 1: 推薦システムの構成

まず事前に小説を上記の手法を用いてクラスタリングを行い，作成された k 個のクラスタと小説の ncode の対応を csv 形式で出力しておく．利用者はフォームから小説家になろうの利用者 ID を入力する．入力された利用者 ID を用いてクローリングしたブックマークデータから参照しブックマークしている小説の ncode を取り出す．この時，ncode のブックマーク集合を B とする．それぞれのクラスタ $C_0, C_1, ..., C_k$ と B について $|B \cap C_k|$ の値を算出し，これを類似度とする．最後に類似度が高い順でかつ総合得点の高い順に $C_k \cap \bar{B}$ を表示する．

6.2　推薦システムの評価

推薦システムの評価は以下のように行うである．何人かの利用者に協力してもらい，実際に利用してもらう．その時の推薦結果に対して，読んだことがあるものはあるか，満足いく推薦がいくかなどをアンケートを行う．その結果をもとにシステムの評価を行う．

7　おわりに

現在 CGM コンテンツの人気に伴い，利用者の体験は自身が満足できる期待値が大きいと考えられるコンテンツに偏っている．コンテンツ推薦によって，利用者の時間を奪うことなく，CGM コンテンツ体験の多様性を広げることができると考える．ただし，その推薦によって，利用者が興味はあるけれども未知であるコンテンツが推薦されることが必要である．クラスタリングを利用しコンテンツを分割することで利用者が興味を持つ分野，そうでない分野を明確に割り出すことができると考える．そして，各クラスタと利用者の興味との距離を測定しすることで利用者が満足するコンテンツを推薦できると考える．

今回，web 小説投稿サイト，小説家になろうを対象に推薦手法の提案を行った．内容ベースでのクラスタリングを行い，ブックマークと各クラスタの積集合を取ることで距離とし推薦対象を決定した．現状の内容ベースでベクトル化を行い Ward 法でクラスタリングを行った．この手法ではクラシフィケーションエラーが非常に悪く，より改良が必要であることがわかった．また，ブックマークとクラスタの積集合の個数を距離とする方法では，ほとんどのクラスタの距離が 0 となってしまい，興味があるが未知であるコンテンツを多く含むクラスタを見逃している可能性が高い．

今後の課題として，クラスタリング手法の改良また，ソーシャルベースでのベクトル化も検討したい．また利用者の興味集合と各クラスタの距離の表現もよりよいものを検討していく．

参考文献

[1] Brent Smith and Greg Linden. Two decades of recommender systems at amazon.com. *IEEE Internet Computing*, Vol. 21, No. 3, pp. 12–18, May-June 2017.

[2] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. Controlling popularity bias in learning to rank recommendation. In *ACM RecSys17 (Proceedings of the Eleventh ACM Conference on Recommender Systems)*, RecSys '17, pp. 42–46, August 2017.

[3] Paul Resnick and Hal R. Varian. Recommender systems. *Communications of the ACM*, Vol. 40, No. 3, pp. 56–58, March 1997.

[4] Steffen Rendle. Factorization machines. In *ICDM '10 (Proceedings of the 2010 IEEE International Conference on Data Mining)*, pp. 995–1000, December 2010.

[5] Mouzhi Ge, Carla Delgado-Battenfeld, and Dietmar Jannach. Beyond accuracy: Evaluating recommender systems by coverage and serendipity. In *ACM RecSys '10 (Proceedings of the fourth ACM conference on Recommender systems)*, pp. 257–260, September 2010.

[6] 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`.

[7] 大元司. niconico におけるコンテンツレコメンドの取り組み. `https://niconare.nicovideo.jp/watch/kn3440`, 2019.

[8] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, Vol. 2 of *NIPS'13*, pp. 3111–3119, USA, 2013. Curran Associates Inc.

[9] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine*, pp. 1188–1196, 2014.

[10] 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.

[11] gensim topic modeling for humans. `https://radimrehurek.com/gensim/`.

[12] E. Jones, T. Oliphant, P. Peterson, and et al. Scipy: Open source scientific tools for python.