Wikipedia 協調フィルタリング法

竹内皓紀 [1] 林克彦 [2]
[1] 群馬大学 [2] 北海道大学
[2] katsuhiko-h@ist.hokudai.ac.jp

概要

Wikipedia は様々な物事（ここでは「エンティティ」と呼ぶ）について質の高い記事が存在し，多様な研究領域において利用されてきた．従来の研究では，Wikipedia の概要文やハイパーリンクなどのコンテンツ情報を利用することが一般的であったが，Wikipedia のコンテンツ情報はユーザの主観を排して編集されるため，評論やレビュー文とは異なり，エンティティに関する表層的な属性情報しか考慮することができない．この課題を解決するため，本稿では Wikipedia の編集者情報を利用した協調フィルタリング法を提案する．提案手法をエンティティ間の類似度推定に利用し，推薦タスクで評価を行った結果，その有効性を確認した．

1 はじめに

Wikipedia は誰でも編集できるオンライン百科事典であり，編集の容易さや編集人数の多さから様々な物事（ここでは「エンティティ」と呼ぶ）について質の高い記事が存在する．そのため，Wikipedia から得た情報は様々な研究領域において利用されてきた．その中でも，エンティティ間の類似度推定は，推薦，検索や自然言語処理など多くの応用先があり，単語埋め込みなどの手法とも関係性がある研究課題である．エンティティ間の類似度を推定する際には，まずエンティティの特徴量を抽出する必要がある．その情報源として図 1 に示すような概要文やハイパーリンクなどの Wikipedia のコンテンツ情報を活用することが一般的である [1, 2, 3].

しかし，Wikipedia は百科事典であり，そのコンテンツに関しては「中立的な観点」を基本方針の 1 つとしている．「中立的な観点」とは，信頼できる情報源を慎重に分析し，可能な限り編集上の偏向なく読者に伝えることを指す[1)]．そのため，Wikipedia

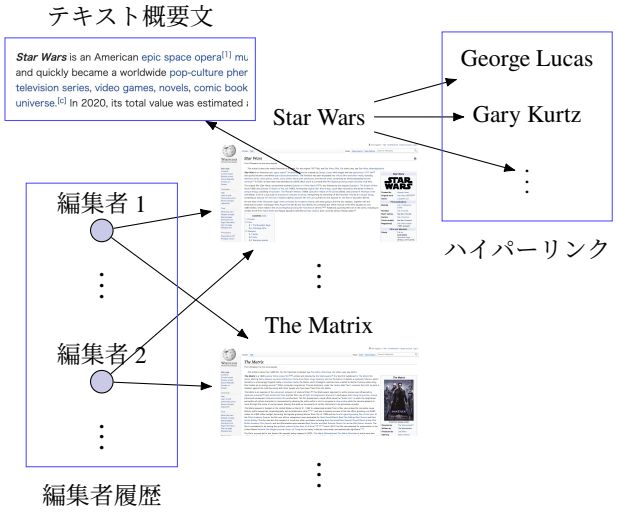

図 1: Wikipedia から取得できる各種情報．画像は英語版 Wikipedia から引用.

のコンテンツ情報は，編集者の個人的な意見や嗜好が反映されにくく，表層的な属性情報に限られる．よって，従来のコンテンツベースによるエンティティ類似度推定手法の欠点の 1 つとして，人間の嗜好性などに内在するエンティティ間の複雑な類似性を捉えることが本質的に難しい点が挙げられる．

一方，コンテンツベースとは異なる考え方として，協調フィルタリングと呼ばれる方法論がある [4]．協調フィルタリングはユーザの嗜好情報で推論を行う方法論であり，主に推薦に関する研究分野で発展してきた．有名な応用例としては，Amazon の商品推薦，Netflix の動画推薦システムなどが知られている．このようなシステムを構築するにはユーザの嗜好が含まれた購買履歴などのプロファイル情報を入手する必要があり，協調フィルタリングを適用できるドメインは一般に限られるが，商品や動画などのエンティティ間に内在する複雑な類似性を捉えることが可能となる．

このような背景から，本研究では協調フィルタ

1) https://en.wikipedia.org/wiki/Wikipedia:Neutral_point_of_view (参照 2022-10-25)

リングを参考にして，Wikipedia におけるエンティティ間の類似度を推定する新しい手法を提案する．図 1 で示すように，Wikipedia には各記事を誰が編集したかという履歴が残されており，提案手法ではこのような記事の編集者情報を利用する．編集者は一般に関心がある記事を編集するため，協調フィルタリングの考え方から，同一編集者に編集された記事に対応するエンティティ同士は類似すると仮定できる．そのため，提案手法では，客観的な属性情報では捉えることが難しかったエンティティ間の複雑な類似性を推定できることが期待される．

本研究では推薦タスクを用いて提案手法の有効性を検証した．実験結果からは Wikipedia の概要文やハイパーリンクに基づくコンテンツベースの手法に対して，編集者履歴を使った提案手法の有効性が確認されたのでこれを報告する．

2 関連研究

2.1 Wikipedia からの特徴量抽出

推薦・情報検索・自然言語処理分野において，Wikipedia から単語間や文書間の意味的な類似性を捉えた特徴ベクトルを学習することは重要な課題である．このとき，Wikipedia のコンテンツ情報であるテキストやハイパーリンクに対して，Explicit semantic analysis（ESA） [2]，Word2Vec [5]，Wikipedia2Vec [1]，BERT [6]，潜在的意味解析 [7] や Paragraph Vector [8] などのモデルやツールを利用してベクトルを学習することが一般的となっている．

このような特徴ベクトルの応用先の 1 つとして推薦タスクが考えられる．文献 [3] では，Wikipedia 記事のテキストを対象に文書ベクトルを推定し，その類似度を使って映画や書籍の推薦を実現している．しかし，推薦のような応用先を考える場合，コンテンツ情報では人の嗜好性を捉えることが難しく，文献 [3] の手法では十分な推薦精度を達成できていない．本稿で提案する Wikipedia の編集者情報を利用するアプローチは，このような従来手法の課題を解決できる可能性を秘めた新しい試みとなっている．

2.2 協調フィルタリングを用いた推薦

協調フィルタリングを用いた推薦手法として，行列分解による手法 [9] や，それを一般化させた深層学習による手法 [10] がある．しかし，深層学習モデルの隠れ層を増加させても，推薦性能に大きな違いが見られないという報告もある [11, 12, 10, 13, 14]．一方で，協調フィルタリングを用いた古典的な推薦法として，近傍探索に基づく手法 [15, 16] があり，現在でも商用のシステムで広く利用されている．

アイテムベースの近傍探索モデルではアイテム間の類似度を推定する必要があり，近年では回帰に基づく手法が主流である [11, 17]．特に EASE モデル [11] では，類似度行列の推定をリッジ回帰問題として定式化する．これは閉形式で解を推定できるため，最適化が容易であり，安定的に高い推薦性能を実現できることが報告されている．

3 EASE による類似度推定

本稿ではエンティティ間の類似度を推定するためのモデルとして，EASE [11] を採用する．

N 件の Wikipedia 記事 $(D_1, D_2, \cdots, D_N)$ が与えられたとき，D_i は M 種類の素性 (特徴量) を基底としたベクトル $[f_{i1}, f_{i2}, \cdots, f_{iM}]^{\mathrm{T}}$ として表せる (Wikipedia 記事はエンティティに対応する)．これは Wikipedia 記事 D_i に素性 w_j が出現すれば，$f_{ij} = 1$，出現しなければ，$f_{ij} = 0$ となるベクトルとする[2]．このような N 個のベクトルを行に並べた Wikipedia 記事行列:

$$\mathbf{F} = \begin{pmatrix} f_{11} & f_{12} & \cdots & f_{1M} \\ f_{21} & f_{22} & \cdots & f_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ f_{N1} & f_{N2} & \cdots & f_{NM} \end{pmatrix} \in \{0, 1\}^{N \times M}$$

を定義する．Wikipedia 記事の素性としては，概要文に含まれる 1-gram や 2-gram，ハイパーリンク，編集者履歴などを利用することができる．

EASE によって類似度行列を推定する場合，Wikipedia 記事 D_i に素性 w_j が出現することを回帰で予測する問題として定式化する．回帰で用いる説明変数として，素性 w_j を除いた残りの $M - 1$ 個の素性を用いることを考える．これを行列形式で定義すると以下のようになる．

$$\widehat{\mathbf{B}} = \arg\min_{\mathbf{B}} \left\{ ||\mathbf{F} - \mathbf{F}\mathbf{B}||_F^2 + \lambda ||\mathbf{B}||_F^2 \right\} \qquad (1)$$
$$\text{s.t.} \quad \mathrm{diag}\,(\mathbf{B}) = \mathbf{0}.$$

これは L2 正則化付きの二乗誤差を最小化することで，行列 $\mathbf{F}$ を自己復元する重み表現 $\mathbf{B}$ を獲得することが目的となる．ただし，$\mathbf{B} = \mathbf{I}$ とすれば，自明な形

2) f_{ij} には Wikipedia 記事 D_i 中に素性 w_j が出現した回数を考えることもできる．

表 1: データセットの統計情報: 各特徴量の種類数.

	ML-20M	Last.fm	LT
エンティティ数	18,148	9,176	9,545
ユーザ数	126,596	1,883	7,223
概要文（1-gram）	75,433	57,526	42,635
概要文（2-gram）	631,726	418,278	340,522
ハイパーリンク	421,742	379,827	191,635
カテゴリ	31,621	25,788	15,742
編集者 (英語)	1,762,411	2,287,302	637,015
編集者 (多言語)	3,182,240	4,435,536	1,048,468

で式 (1) の最小化が達成されてしまうため，制約条件として $\mathrm{diag}(\mathbf{B}) = \mathbf{0}$ を課している．これは $\mathbf{B}$ の対角成分を全て 0 とすることを意味する．

式 (1) はラグランジュの未定乗数法により，

$$||\mathbf{F} - \mathbf{FB}||_F^2 + \lambda||\mathbf{B}||_F^2 + 2\boldsymbol{\alpha}^{\mathrm{T}}\,\mathrm{diag}(\mathbf{B})$$

を最小化する問題に帰着される．ここで $\boldsymbol{\alpha}$ はラグランジュ乗数のベクトルを表す．そして，この解は下記のような閉形式を持ち，

$$\widehat{\mathbf{B}} = \mathbf{I} - \mathbf{P}\,\mathrm{mat}(\mathbf{1} \oslash \mathrm{diag}(\mathbf{P})) \tag{2}$$

$\mathbf{P} = (\mathbf{F}^{\mathrm{T}}\mathbf{F} + \lambda\mathbf{I})^{-1}$ であり，各要素は

$$\widehat{\mathbf{B}}_{jk} = \begin{cases} 0 & (j = k) \\ -\frac{\mathbf{P}_{jk}}{\mathbf{P}_{kk}} & (j \neq k) \end{cases} \tag{3}$$

として推定できる．$\widehat{\mathbf{B}}$ は Wikipedia 記事間の類似度行列として用いることができる．

4 実験

4.1 評価用データセットの整備

実験ではデータセットとして MovieLens-20M（ML-20M）[3]，Last.fm hetrec-2011（Last.fm） [18] と LibraryThing（LT）[4] を用いた．それぞれ，映画，音楽アーティストと書籍のドメインに関する推薦評価用データセットであり，ユーザの評価値を 2 値として扱っている．また，エンティティと Wikipedia 記事の対応付けについて，ML-20M は映画タイトルと編集距離が近いタイトルの Wikipedia 記事を用いた．Last.fm と LT については，エンティティと Wikipedia 記事の対応付けデータ [19, 20] を参考にした．

各推薦データのエンティティ（映画，音楽アーティストや書籍）に対応付けた Wikipedia 記事から，英語 Wikipedia の編集者，多言語 Wikipedia の編集者，概要文，ハイパーリンクとカテゴリに関する情報を取得した．ユーザ数や編集者数，語彙数などの統計情報は表 1 に示す．また，図 2 には ML-20M, Last.fm と LT に対して，記事数が多い上位 25 カ国の記事数を示した．英語版 Wikipedia については全エンティティに対する記事が存在する．

3) https://grouplens.org/datasets/movielens/20m/
4) https://github.com/sisinflab/LinkedDatasets

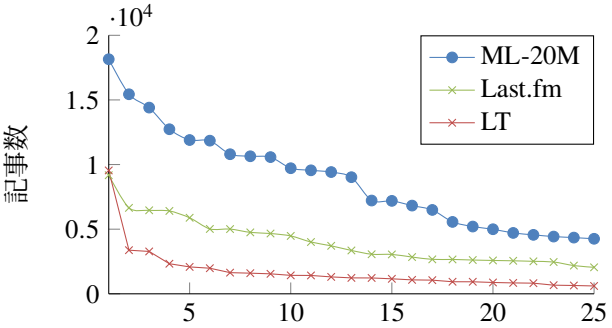

図 2: 多言語 Wikipedia の記事数 (上位 25 カ国).

4.2 推薦による評価

本実験では推薦データに内在するユーザの嗜好性を Wikipedia 情報からどの程度捉えることができるのか調査する．具体的には，Wikipedia 情報から推定したエンティティの類似度行列を用いて推薦タスクの性能評価を行う．よって，以下ではまず，推薦タスクの手順について説明する．

評価手順　推薦による評価を行うには，まず以下 3 つのデータが必要となる．

- 訓練用データ
- 評価用履歴データ
- 評価用解答データ

訓練用データはエンティティの類似度行列 $\widehat{\mathbf{B}} \in \mathbb{R}^{N \times N}$ を推定するのに利用するデータである．本実験では，Wikipedia から抽出した情報を訓練用データとして扱う．

評価用履歴データは推薦データに含まれるユーザの過去の嗜好プロファイル情報である．評価用解答データは，評価用履歴データと同じユーザに対する嗜好プロファイル情報であるが，それぞれのデータにおいて，あるユーザ u が関心を持ったエンティティに重なりはないように分割されている．ML-20M，Last.fm と LT の各推薦データに対して，ユーザをランダムに 5 分割し，各分割のユーザに対して，評価用履歴データと評価用解答データを大凡 80%と 20%の割合で分割した．

評価用データについて，推薦タスクにおける具体的な役割を説明する．評価用履歴データに含まれる

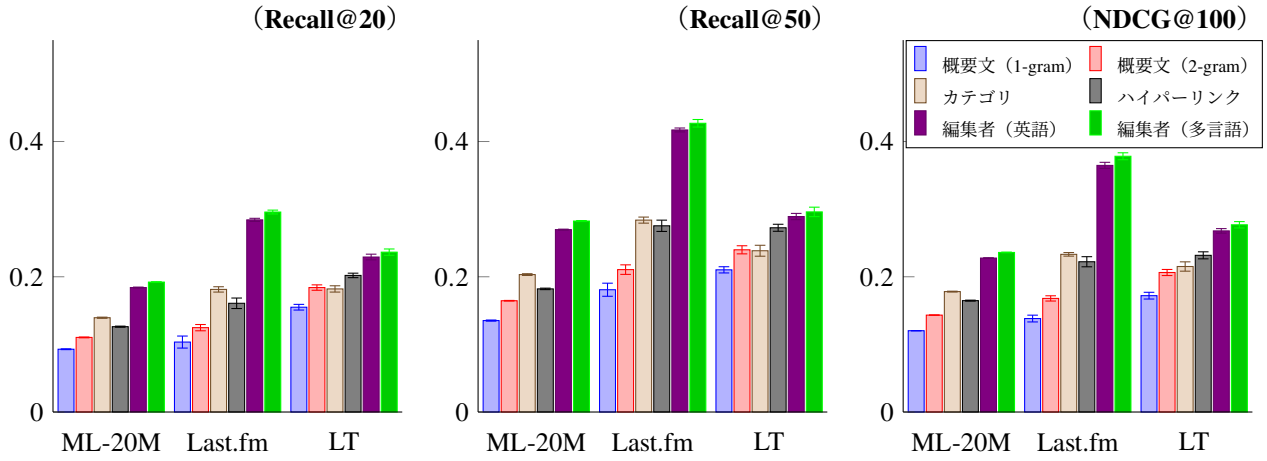

図 3: ML-20，Last.fm と LT における 5 分割した各結果の推薦性能に対する平均と標準偏差．

ユーザ u の履歴情報は $\mathbf{x} \in \{0,1\}^N$ というエンティティ数 N 次元のベクトルで表され，このベクトルは u が過去に関心を持ったエンティティに対応する次元の要素が 1 となり，それ以外は 0 となる．推薦はこのユーザ u が関心を持つ可能性のあるエンティティを予測するタスクである．この予測は $\mathbf{x}^{\mathrm{T}}\widehat{\mathbf{B}}$ で計算され，その結果はユーザ u の各エンティティに対するスコア（関心）を表す．そして，このスコアの高いエンティティが評価用解答データに含まれていれば良い推薦として評価される．

評価指標　本実験では Recall@R と NDCG@R という 2 つの評価指標を用いて，推薦性能を評価する．R とは推薦したエンティティの数を表す．また，本稿では $\omega(r)$ を R 個中の上位 r 番目のエンティティ，$\mathbb{I}[\cdot]$ を指示関数，$\mathcal{I}_u$ をユーザ u の評価用解答データとして定義する．これらをふまえて，あるユーザ u に対する Recall@R は以下のように定義される．

$$\mathrm{Recall@}R := \sum_{r=1}^{R} \frac{\mathbb{I}[\omega(r) \in \mathcal{I}_u]}{\min(R, |\mathcal{I}_u|)}.$$

また，DCG@R は以下のように定義される．

$$\mathrm{DCG@}R := \sum_{r=1}^{R} \frac{2^{\mathbb{I}[\omega(r) \in \mathcal{I}_u]} - 1}{\log(r+1)}.$$

NDCG@R は，推定した DCG@R を DCG@R の理想値で割ることにより計算される．これらの詳細は文献 [10] などを参照されたい．

性能評価　結果を図 3 に示す．これらは 5 分割した各結果の推薦性能に対する平均と標準偏差を示している．Wikipedia のコンテンツ情報と編集者情報を比較すると ML-20M，Last.fm と LT の各データにおいて，すべての評価指標で編集者情報を用いたシステムの方が高い性能を示している．

分析　Wikipedia 情報で推定した類似度行列に対する事例分析を行うことで，提案手法がコンテンツ情報を用いる従来法よりも優れた推薦性能を達成できた要因を探る．ML-20M の Fight Club という映画に対して，各情報から推定した類似度が高い上位 10 映画を抽出し，定性的分析を行った．コンテンツ情報から推定した類似度上位 10 映画について，ハイパーリンクは Fight Club と同監督の映画を多く挙げていた．一方で，編集者情報から推定した類似度上位 10 映画においては，Pulp Fiction や Memento（film）など，Fight Club と人の嗜好性に基づく関係性のある映画が挙げられた．つまり，従来法では属性情報を捉えているのに対し，提案手法では嗜好情報を捉えており，こうした違いが推薦性能の差に現れたのだと考えられる．詳細は付録 A に付す．

5　まとめ

本稿では，Wikipedia の編集者情報を利用した協調フィルタリング法を提案し，エンティティの類似度推定に応用した．従来の概要文やハイパーリンクなどのコンテンツ情報を利用した推定手法とは異なり，提案法はユーザの嗜好性を反映した類似度の推定が可能になると期待され，実際に，映画，音楽と書籍ドメインの推薦データを用いた評価実験から提案法の有効性を確認することができた．

今後は，推薦以外のタスクへの応用も検討し，提案法の有効性をさらに検証したい．その際，類似度としての活用ではなく，Wikipedia 記事行列の行列分解などを用いて，エンティティの低次元ベクトル表現を構築することで，より応用性の高い形式で結果を活用することも検討したい．

謝辞

本研究は JSPS 科研費 JP 21H03491 の助成を受けた.

参考文献

[1] Ikuya Yamada, Akari Asai, Jin Sakuma, Hiroyuki Shindo, Hideaki Takeda, Yoshiyasu Takefuji, and Yuji Matsumoto. Wikipedia2vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from wikipedia. **arXiv preprint arXiv:1812.06280**, 2018.

[2] Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In **Proc. of IJCAI**, pp. 1606–1611, 2007.

[3] Cataldo Musto, Giovanni Semeraro, Marco de Gemmis, and Pasquale Lops. Learning word embeddings from wikipedia for content-based recommender systems. In **Proc. of ECIR**, pp. 729–734, 2016.

[4] David Goldberg, David Nichols, Brian M Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. **Communications of the ACM**, Vol. 35, No. 12, pp. 61–70, 1992.

[5] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In **Proc. of NIPS Conference**, 2013.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. **arXiv preprint arXiv:1810.04805**, 2018.

[7] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. **Journal of the American society for information science**, Vol. 41, No. 6, pp. 391–407, 1990.

[8] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In **Proc. of ICML**, pp. 1188–1196, 2014.

[9] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Application of dimensionality reduction in recommender system-a case study. Technical report, 2000.

[10] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. Variational autoencoders for collaborative filtering. In **Proc. of WWW Conference**, pp. 689–698, 2018.

[11] Harald Steck. Embarrassingly shallow autoencoders for sparse data. In **Proc. of WWW Conference**, pp. 3251–3257, 2019.

[12] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In **Proc. of WWW Conference**, pp. 173–182, 2017.

[13] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. Autorec: Autoencoders meet collaborative filtering. In **Proc. of WWW Conference**, pp. 111–112, 2015.

[14] Yin Zheng, Bangsheng Tang, Wenkui Ding, and Hanning Zhou. A neural autoregressive approach to collaborative filtering. In **Proc. of ICML**, pp. 764–773, 2016.

[15] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In **Proc. of WWW Conference**, pp. 285–295, 2001.

[16] Katsuhiko Hayashi. Rethinking correlation-based item-item similarities for recommender systems. In **Proc. of SIGIR Conference**, pp. 2287–2291, 2022.

[17] Xia Ning and George Karypis. Slim: Sparse linear methods for top-n recommender systems. In **Proc. of ICDM**, pp. 497–506, 2011.

[18] Iván Cantador, Peter Brusilovsky, and Tsvi Kuflik. Second workshop on information heterogeneity and fusion in recommender systems (hetrec2011). In **Proc. of RecSys Conference**, pp. 387–388, 2011.

[19] Ignacio Fernández-Tobías, Paolo Tomeo, Iván Cantador, Tommaso Di Noia, and Eugenio Di Sciascio. Accuracy and diversity in cross-domain recommendations for cold-start users with positive-only feedback. In **Proc. of RecSys Conference**, pp. 119–122, 2016.

[20] Tommaso Di Noia, Vito Claudio Ostuni, Paolo Tomeo, and Eugenio Di Sciascio. Sprank: Semantic path-based ranking for top-n recommendations using linked open data. **ACM Transactions on Intelligent Systems and Technology (TIST)**, Vol. 8, No. 1, pp. 1–34, 2016.

A 参考情報

表 2: ML-20M に登録されている映画 Fight Club と類似度が高い上位 10 映画: 下線は Fight Club と同監督の映画を，太字は Fight Club と嗜好性に基づく関係性がある映画を表す．

	Fight Club
概要文	Escape from L.A. Last Action Hero Life (1999 film) Rebel Without a Cause Panic Room Great Expectations (2012 film) Vampire in Brooklyn The Curious Case of Benjamin Button (film) Never Back Down Who Am I (2014 film)
カテゴリ	Panic Room Rashomon Barton Fink Eyes Wide Shut Die Hard Tokyo Story Django Unchained Storm Catcher The Skulls (film) Pushing Tin
ハイパーリンク	Choke (2008 film) Panic Room The Game (1997 film) Alien 3 Zodiac (film) The Girl with the Dragon Tattoo (2011 film) The Curious Case of Benjamin Button (film) **Seven (1995 film)** The Social Network Tron: Legacy
編集者	**Seven (1995 film)** **Pulp Fiction** **The Matrix** Reservoir Dogs V for Vendetta (film) The Godfather **Memento (film)** A Clockwork Orange (film) Zodiac (film) Apocalypse Now

表 2 には，Wikipedia 情報で推定した類似度行列に対する事例分析の結果を記す．具体的には，ML-20M に登録されている Fight Club という映画に対して，各情報から推定した類似度が高い上位 10 映画を示した．下線は Fight Club と同監督の映画を，太字は Figth Club と嗜好性に基づく関係性があると思われる映画を表す．この結果から，コンテンツ情報を用いる従来法は表層的な属性情報を，提案手法は人の嗜好性を捉えていることがわかる．