

動画のコメントデータを対象とした時系列感性変化計量システム

尾崎遼太郎[†] 佐々木史織[‡] 清木康[†]

[†]慶應義塾大学環境情報学部 〒252-0011 神奈川県藤沢市遠藤 5322

[‡]慶應義塾大学政策・メディア研究科 〒252-0011 神奈川県藤沢市遠藤 5322

E-mail: [†] {t15188ro, kiyoki} @sfc.keio.ac.jp, [‡] {sashiori} @sfc.keio.ac.jp

あらまし 本稿では、動画共有サービスにおいて動画について投稿されたコメント（以下コメントデータと呼ぶ）とその時系列変化分析により、動画の感性変化を計量するシステムを提案する。近年、動画共有を行うウェブサービスの利用者は増加しており、動画へのコメント投稿が活発である。動画についての印象に関する分析において、動画に付与されたコメントデータは、動画を見た人々の感性が直接的に反映されたデータであり、動画の感性分析において有効である。本システムは、個人の設定した感性データを対象とし、動画の時系列に沿ってコメントデータを分析し、動画を時系列感性ベクトルとして表現する。本システムは、このコメントデータの時系列感性ベクトルを対象とした相関量計量により、動画同士の距離計算やユーザーへの最適な動画の推薦を行う。

キーワード 時系列データ、テキストデータ、マルチメディアデータベース

1.はじめに

近年、動画共有サイト等の発達により、大量の動画データがインターネット上で共有、閲覧可能になっている。しかしながら、これらの一般的な共有サイト上では動画の内容分析はあまり行われておらず、動画検索はキーワードによる検索に頼られているという現状がある。動画から受ける感性をもとにした検索や閲覧には大きな需要があり、動画や音楽といった時間的連続性を持つメディアデータを色彩心理学や音楽心理学の知識をベースとして感性的に分析する試みは数多くある[1]-[7]。ただし、動画は大量の画像データと音声データからなるメディアデータであり、これらを制限時間内に、機械的かつ人間の感性に合致するように処理することには限界があると考えられる。そこで本稿では、メディアデータを多次元空間上にマッピングする手法[8]、テキストデータの時系列類似度の計量手法[9][10]にヒントを得て、現在動画共有サイトにおいて一般的に活用されているコメントデータを活用して動画の感性計量を行う手法を提案する。

本システムは、動画共有サービスにおいて動画について投稿されたコメント（以下、コメントデータと呼ぶ）をもとに動画の感性を時系列に沿って計量する。また、計量された感性を元に動画同士の距離を計算し、ある動画に対して感性的に近い動画を推薦する。

コメントデータを活用することには、大きく2つのメリットがある。

1 点目として、コメントデータはテキストデータであり、収集や分析がメディアデータに比べて容易であることが挙げられる。動画の特徴を十分表せるだけのデータを集めたとしても、軽量のテキストファイルとして扱うことが可能である。

2 点目として、コメントデータは動画共有サイトを利用するユーザーの感性が反映されたデータであることが挙げられ

る。コメントは、動画共有サイトを利用するユーザーが、動画から受けた感性を直接的に反映したものであり、この分析は感性分析手法として有効であると言える。

本稿では、時系列感性ベクトルを、時系列を m 分割、感性 n 種類という設定の下で作成される、 m 行 n 列かつ長さ 1 の行列であると定義し、次に述べる基本方式によってベクトル間の相関量計量による時系列感性変化計量システムの実現方式について述べる。

2.基本方式

本システムは、(1)ユーザーが自身の感性を反映させるプロセス、(2)コメントデータから時系列感性ベクトルを生成するプロセス、(3)時系列感性ベクトルをもとに動画同士の感性的な距離を計算するプロセス、(4)計算結果を可視化するプロセスという、四つのプロセスから構成される。システム構成を図1に示す。

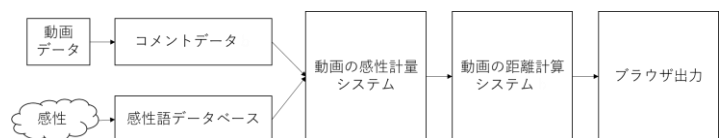


図1 システム構成図

2.1. ユーザーが自身の感性を反映させるプロセス

本システムを利用する際、ユーザーは自身の感性を反映させるために感性語データベース、時系列の分割粒度という、2つの情報を入力する必要がある。

感性語データベースは動画のコメントデータにおいて感性を表現する単語（以下感性語）と、その感性語のもつ感性ベクトルからなる。感性の種類や感性ベクトルはユーザーが自身の感性を元に作成する。感性の種類数を n 個とした場合、

単語の感性ベクトルは全て n 次元のベクトルである。

時系列の分割粒度は、動画を時系列分析する際に、単位となる大きさである。本システムは対象の動画を時間で等分割し、いくつかのブロックの集合体とすることで時系列を扱う。ユーザーは、分析対象となる動画の完成変化を反映すると考える粒度を自身で設定し、システムに入力する。

2.2.時系列感性ベクトルの生成プロセス

本システムは設定された感性および時系列粒度に基いて時系列感性ベクトルを作成する。

時系列感性ベクトルは m 行 n 列の行列である。このとき、 m は時系列を分割したブロック数であり、 n は感性の種類数である。

$$\begin{array}{c} \text{感性} \\ f_1 \quad \dots \quad f_n \\ \text{時系列分割} \begin{array}{c} t_1 \\ \vdots \\ t_m \end{array} \begin{pmatrix} S_{11} & \dots & S_{1n} \\ \vdots & \ddots & \vdots \\ S_{m1} & \dots & S_{mn} \end{pmatrix} \end{array}$$

図2 時系列感性ベクトル

時系列感性ベクトルは、コメントごとに感性ベクトルを作成し、コメントの感性ベクトル(VC)を時系列に沿って計量することによって作成される。

コメントの感性ベクトル(VC)は、コメントに含まれる単語の感性ベクトル(VW)を合計したのち、2ノルム正規化を行ったものである。すなわち、感性の種類数を n とした場合、コメントの感性ベクトルは n 次元のベクトルであり、その長さは1である。

$$VC = (\sum_{k=0}^n VW_k) \div |VC|$$

図3 コメントの感性ベクトルの算出方法

このようにして作成したコメントの感性ベクトル(VC)を、そのコメントが属する時系列ブロックごとに合計し、時系列順に並べたものを用意し、最後にこの行列を2ノルム正規化し、動画全体の感性量を1に正規化したものが、その動画の時系列感性変化ベクトルである。

2.3.時系列感性ベクトルをもとに動画同士の感性的な距離を計算するプロセス

本システムにおいて、ブロック数である m 、および n 種類の感性が対応している m 行 n 列の時系列感性ベクトル同士は、感性的な距離を計算し、感性的に近い動画を推薦することが可能である。

本システムにおける動画同士の距離は、0以上1以下の実数によって表されている。

まず、計算を行う二つの時系列感性ベクトルを感性ごとに分割し、 n 次元の時系列ベクトルを m 個ずつ用意する。そして、それぞれの感性ごとにベクトル同士のコサイン類似度を計算し、0以上1以下の実数が m 個求められる。この m 個の

実数の平均値を求めたものが、動画の感性的な距離の定義である。

2.4.計算結果を可視化するプロセス

最後に、計算結果である動画の時系列感性ベクトルと、感性的に近い動画を、可視化して出力する。

3.実現方式

python を用いてシステムを実現した。

動画共有サイトとして、今回はニコニコ動画を利用した[11]。理由としては、利用者が多くデータが豊富であること、時系列に沿った形でコメントされるシステムであること、データが配布されており利用がしやすいことがあげられる。

データには、国立情報学研究所にて提供されているニコニコデータセットを利用した[12]。

このデータは、動画情報およびコメントデータが json 形式で表現されたデータである。

本システムではコメントデータの少ない動画を対象とした場合、あるいは著しく長いもしくは短い動画を対象とした場合に、計量が適切に行われないことが想定される為、それぞれ閾値を指定し、コメントデータが閾値を超え、動画の長さが範囲内に存在する動画のみをリストアップするプログラムを作成した。

3.1.ユーザーが自身の感性を反映させるプロセス

感性語データベースは csv 形式で作成し、その処理には python のライブラリである pandas を利用した[13]。

時系列分割のブロック数はプログラム内で変数として与えられる。

プログラム内において感性語の数、感性の種類数、時系列ブロック数などは全て変数として管理されており、可変的である。

3.2.時系列感性ベクトルの生成プロセス

行列計算は python のライブラリである numpy を用いて実装した[14]。

時系列感性ベクトルは、それぞれの動画において numpy のデータ保存形式である npy ファイルとして保存される。

3.3.時系列感性ベクトルを元に動画同士の感性的な距離を計算するプロセス

動画同士の距離計算プログラムは、時系列感性ベクトルの生成プロセスとは分けて作られており、保存された時系列感性ベクトルを再度読み込む形で実装されている。これにより、別の場所や方法で作成されていても、形の統一された時系列感性ベクトル同士であれば計算が可能になることを示している。計算結果として、各動画に対して、動画番号と距離が算出される。システムはこの結果を、距離に近い順に csv 形式で出力する。

3.4. 計算結果を可視化するプロセス

計算結果の出力は、python の web フレームワークである Django を用いて、ブラウザで確認できる形で実装した[15]。

各動画の時系列感性ベクトルは npy データから読み込みグラフとして可視化し、動画の距離は csv データから読み込み距離の近いいくつかの動画と合わせて画面に出力を行う。

計算自体はすでに行われており、出力に際しては計算を行わない。そのため、出力は高速に行うことが可能である。

4. 実験

ニコニコ動画に存在する動画全てを対象とした場合、動画数が多くなりすぎること、使用される言葉や頻出する固有名詞が時代によって異なることなどから、計量が正しく行われないことが想定されるため、データセットの中で比較的新しい動画群を対象として計量を行うこととした。

まず、自分が動画に対してもつ感性として、「笑い (fun) ・興奮 (excitement) ・驚き (surprise) ・恐怖 (fear) ・悲しみ (sadness) ・怒り (anger) ・不安 (worry) ・幸福 (happy)」という 8 種類の感性 (f1, f2, ... f8) を設定した。

感性語はニコニコ動画のコメントにおいて頻出する単語の中で、感性を表すと考えられるものを 20 個リストアップし、それぞれに対し自分の感性を 0 から 10 までの 10 段階で反映させて感性語データベースを構築した。

この際、頻出する単語を数えるために、形態素解析エンジンである mecab を利用した[16]。

表 1 感性語データベースの一部

wid	f1	f2	f3	f4	f5	f6	f7	f8
0	8	0	4	0	0	0	0	4
1	0	4	8	4	0	4	0	4
2	4	0	4	0	0	0	8	0
3	8	4	8	4	4	4	0	0
4	4	4	8	4	4	0	0	0
5	4	4	8	4	4	0	0	0

コメントの閾値として 500 個、動画の長さの閾値として 5 分以上 30 分未満という設定を行い、リストアップを行った。今回の実験では 62 個の動画がこれに該当した。

```

0 sm29560079
【ボイスロイド実況】 茜のカービィボウルをプレイするで！ part10
1 sm29560199
【MUGEN】 春闘下 12P 前後 狂下位ランセル！ 台パンの向こう側へ！
Part40
2 sm29560445
【ゆっくり実況】 UNDERTALE Part19 (Genocide End)
3 sm29560591
【実況】 いまだかつてないほど初見すぎる Minecraft Part70
4 sm29560791
週刊 VOCALOID と UTAU ランキング #465・407
5 sm29560898
「サタン」と「ガブリエル」の配布に物申す。【シャドウバース】
6 sm29561313
【ゆっくり TRPG】 超人高校生+おまけのやりたい放題クトゥルフ 59 話
7 so29560248
【ベカる TV】 群馬の優良店で 10 万円取り戻したいんだの巻 (牙狼金色・
魔戒・沖ドキ等) 【それ行け養分騎士 vol. 17】
    
```

図 4 対象となった動画の一部

時系列分割ブロック数は、5 分・30 分という長さを踏まえ、20 ブロックと設定した。

以上を設定した上で実際に計量を行った。

各動画の時系列感性ベクトルの例を、図 5, 図 6 に示す。

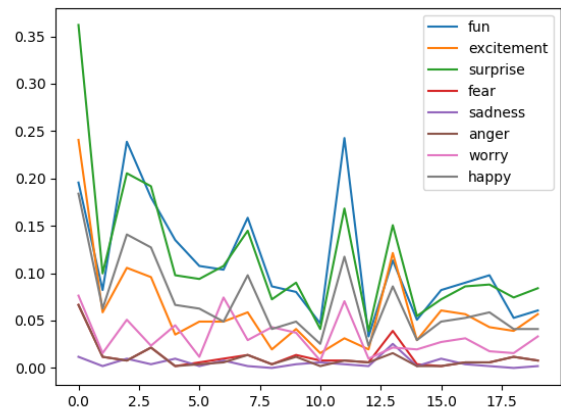


図 5 動画 0 (sm29560079) の時系列感性ベクトル

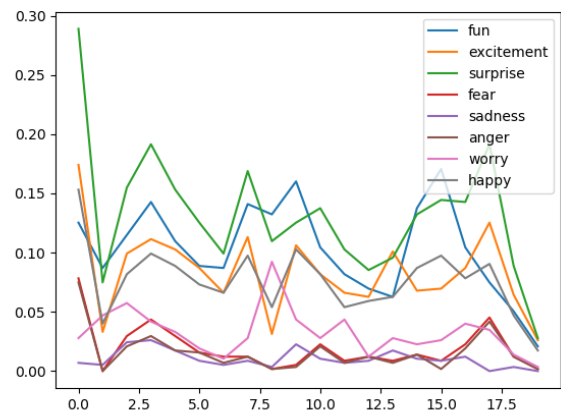


図 6 動画 1 (sm29560199) の時系列感性ベクトル

また、動画同士の距離計算結果の例を図7に示す。

[[0.00000000 0.44833209 0.41553803 ... 0.14251599 0.26084857
0.30974781]
[0.44833209 0.00000000 0.44596172 ... 0.40949881 0.3110088
0.36383379]
[0.41553803 0.44596172 0.00000000 ... 0.38962032 0.3574949
0.41766504]
...
[0.14251599 0.40949881 0.38962032 ... 0.00000000 0.30715718
0.36524598]
[0.26084857 0.3110088 0.3574949 ... 0.30715718 0.00000000
0.36866828]

図7 動画同士の距離行列

さらに、各動画に対して近い動画を順にソートした結果を表2に示す。

表2 近い動画のリストアップ (一部)

動画	一番近い	距離	次に近い	距離
sm29550166	sm29560898	0.142515993	sm29558082	0.157062021
sm29550318	sm29555279	0.242939855	sm29554647	0.283876819
sm29550423	sm29554555	0.204341597	sm29552610	0.229934798
sm29550517	sm29552921	0.06001082	sm29558082	0.084709937

以上をもとに、実際にブラウザ上に出力した結果を図8に示す。

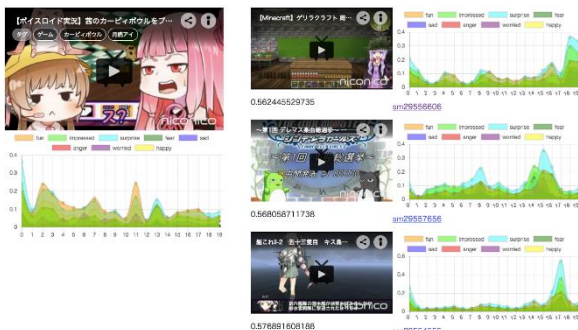


図8 ブラウザ上での出力結果例

5. 考察

今回の実験において、時系列感性ベクトルは概ね動画の感性を表現していると言えるものになっていた。一方で、いくつか実際の感性とは異なった計量となっているように感じられるものもあり、その理由として次の二点が挙げられる。

まず一点目として、本システムが典型的なコメントに強く影響されている点が挙げられる。ニコニコ動画においては「うぼつ」「888」など、動画の特定のタイミングにおいて典型的に用いられるコメントが多くある。こういったコメントは本来大きな感性を持たないと考えられるものであるが、本システムはコメントごとに正規化を行なっている特性上、これらも他のコメントと同様に感性量を持っているものとして計量される。そのため、定型文として多く投稿されて

いる定型文コメント群を、強い感性を持った場所として認識してしまっている。

二点目として、感性語を含まないコメントを感性量0として計量している点が挙げられる。本システムにおいて各コメントは、2ノルム正規化によって長さ1に揃えられているが、感性語を一つも含まないコメントに関しては、感性量0として扱っている。そのため、コメント数が多い部分でも、感性語が含まれない場合には感性がないものとして扱われてしまっている。

これら二つの問題の対処法として、コメントごとに重み付けを行うことが有効だと考えられる。典型的なコメントは感性量を少なく、感性を強く表していると考えられるコメントについては、強い感性量を強く計量できるような重み付けを行うことにより、感性計量の精度を高められる。

また、動画同士の感性的な距離の計算に関しては、概ね良い結果が出ているとはいえ、まだ精度の向上が可能と考えられる。現在のアルゴリズムにおいては、動画同士の距離をそれぞれの感性ごとに計算しており、任意の時間軸における感性の複合的な状況に関してはほとんど考慮されていない。システムの構成としては、異なったアルゴリズムを実装することは容易であるため、今後は複数のアルゴリズムによる計算結果を別々に表示することで、ユーザーに対し適切なリコメントができるようなシステムの構築を目指す。

6. 結論および今後の展望

本稿では、動画を感性的に計量する手法として、動画共有サイトのコメントデータを活用すること、および、時系列感性ベクトルを利用した実際の計算手法と実現方式を示した。

現在の課題は、個人の感性を適切かつ効率的にシステムに反映させる手法が挙げられる。単語と感性量を逐次手動で入力していく手法は手間がかかる上正確さに疑問がある。今後は、言語辞書や機械学習等を活用して、感性語の選定やベクトルの生成を自動化、もしくは半自動化できる手法を検証したい。

また、本稿における時系列感性ベクトルは、時系列に沿ったコメントデータを感性分析可能な形に計量する手法であり、動画に限らず、様々なコメント付きマルチメディア（テレビの実況中継やSNSデータ等を含む）に対しての適用が可能である。今後はこれらの応用についての展開も視野に、スケーラビリティを検討していく。

参考文献

1. 清木康, 金子昌史, 北川高嗣, “意味の数学モデルによる画像データベース探索方式とその学習機構,” 電子情報通信学会論文誌, Vol. J79-D-II, No. 4, pp. 509-519, Apr. 1996.
2. S. Dubnov, Y. Kiyoki, “Opera of Meaning: film and music performance with semantic associative search”, Frontiers in Artificial Intelligence and Applications, Information Modelling and Knowledge Bases XX, Volume 190, pp. 384-391, 2009.
3. Kurabayashi, S., Ueno, T., Kiyoki, Y.: A Context-Based Whole

Video Retrieval System with Dynamic Video Stream Analysis Mechanisms. In: 11th IEEE International Symposium on Multimedia, San Diego, USA, pp. 505-510 (2009).

4. Kurabayashi S., Kiyoki Y. (2010) MediaMatrix: A Video Stream Retrieval System with Mechanisms for Mining Contexts of Query Examples. In: Kitagawa H., Ishikawa Y., Li Q., Watanabe C. (eds) Database Systems for Advanced Applications. DASFAA 2010. Lecture Notes in Computer Science, vol 5982. Springer, Berlin, Heidelberg.
5. Ijichi, A. and Kiyoki, Y.: "A Kansei Metadata Generation Method for Music Data Dealing with Dramatic Interpretation," Information Modelling and Knowledge Bases, Vol.XVI, IOS Press, pp. 170--182, May, 2005.
6. Jeremy K. Hall, Shuichi Kurabayashi, Yasushi Kiyoki, Multimedia Data Analysis on a Massively Distributed Parallelization Network of Anonymous Web Clients, Advanced Information Networking and Applications Workshops (WAINA), 2010 IEEE 24th International Conference, 20-23 April 2010.
7. Munehiro Fukuda, Shuichi Kurabayashi, Jeremy K. Hall, and Yasushi Kiyoki, "Morphing Parallelization Strategy to Support On-the-Fly Video Analysis," In Proceedings of the 2010 International Conference on Parallel and Distributed Processing Techniques and Applications - PDPTA 2010, pp.541-547, Las Vegas, USA, July 12-15, 2010.
8. 伊藤 正彦, 豊田 正史, 喜連川 優 「多メディア間の話題探索のための時系列画像3次元可視化システム」 情報処理学会論文誌データベース (TOD) 8巻 1号 ページ 27 - 44 発行 2015-03-30.
9. 高橋 勝稔, *濱田 賢吾, 馬野 元秀 「言葉による表現を用いた類似時系列検索の診断への応用」 第31回ファジィシステムシンポジウム セッション ID: TD4-2 2016/02/26.
10. 廣安 知之, 福島 亜梨花, 山本 詩子, 横内 久猛 「2本の時系列データの類似部分自動抽出法の提案 —fNIRS 時系列データに対する検討」 情報処理学会論文誌数理モデル化と応用 (TOM) 7巻 2号 ページ 64-73 発行年 2014-11-27.
11. ニコニコ動画 <http://www.nicovideo.jp>
12. 国立情報学研究所 ニコニコデータセット <https://www.nii.ac.jp/dsc/idr/nico/nico.html>
13. Pandas <https://pandas.pydata.org/>
14. Numpy <http://www.numpy.org/>
15. Django <https://www.djangoproject.com/>
16. MeCab <http://taku910.github.io/mecab/>