

ユーザ情報を利用したレビューの評価値推定

松本 健太郎^{1,a)} 嶋田 和孝^{2,b)}

概要: 本論文ではユーザ情報としてユーザの評価の傾向に着目し、テキスト情報にユーザの評価の傾向を加えたレビューの評価値推定モデルを構築する。本研究においては評価の傾向を表す尺度としてユーザの評価値ごとのレビューの割合（以降、評価値分布）を用いる。提案手法ではテキスト情報に単語 N-gram ($N = 1, 2, 3$), Paragraph-Vector, Word2Vec を用い、ユーザの評価値分布を加えてモデルを構築する。実験においては、評価値分布を用いない手法と提案手法を正解率、平均 2 乗誤差平方根により比較することで提案手法の有効性を検証する。

キーワード: 評価値推定, 評判分析, ユーザ情報

Review Rating Prediction with User Information

KENTARO MATSUMOTO^{1,a)} KAZUTAKA SHIMADA^{2,b)}

Abstract: In this paper, we propose a model of review rating prediction with user evaluation tendency. In this research, we use the percentage of the rating values in user's reviews as the tendency of evaluation. In the proposed method, words N-gram ($N = 1, 2, 3$), Paragraph-Vector, Word2Vec are used for text information, and the model is constructed by adding the feature about the user's evaluation tendency. In this experiment, we verify the effectiveness of the proposed method as compared with a baseline method which does not use the user evaluation tendency in terms of the accuracy and root mean squared error.

Keywords: Rating Prediction, Sentiment Analysis, User Information

1. はじめに

近年、インターネットの普及により Web 上には多くの口コミやレビューなどの評判情報が蓄積されており、評判分析の需要が増している。評判分析のタスクの一つとしてレビューの評価値推定がある。レビューの評価値推定はレビューに対して付与されたレビュアーの肯定、否定の評価の度合いを表す評価値を推定するタスクである。ある文書が肯定、否定かの 2 値を判定する極性判定のタスクに比べ評価値 1~5 のような多値を分類することで、肯定的か否

定的かの情報だけでなく肯定、否定の度合いを獲得できる。そのため、評価値を推定することは商品のランキングや複数のレビュアーのレビューの比較などに役立ち、これまでも評価値推定の研究が進められている [1]。

これまでの評価値推定においてはレビュー中のテキスト情報のみに着目している場合が多く、レビューを書いたユーザに関して考慮されていないことが多い。テキスト情報として Baccianella ら [2] は N-gram, Qu ら [3] は Bag-of-Opinion などの素性を用いているが両研究においてユーザ情報は考慮されていない。しかし、ユーザは各々が異なる評価の基準を持ち、評価の傾向が異なる。たとえば、評価に寛容なユーザはどのような商品に対しても高い評価を与える傾向があり、厳格なユーザは商品の悪い部分を指摘し、低い評価を与える傾向がある。このように評価の傾向はユーザにより異なるため評価値を推定する際にユーザ

¹ 九州工業大学大学院 情報工学府 先端情報工学専攻
Kyushu Institute of Technology, Graduate School of Computer Science and Systems Engineering

² 九州工業大学 大学院情報工学研究院 知能情報工学研究系
Department of Artificial Intelligence, Kyushu Institute of Technology

a) k_matsumoto@pluto.ai.kyutech.ac.jp

b) shimada@pluto.ai.kyutech.ac.jp

の情報を考慮することでユーザごとの特徴を反映でき精度の向上が期待できる。

本研究ではユーザにより異なる評価の傾向を考慮することを目的として、ユーザの評価の傾向に基づく評価値推定モデルを構築する。具体的には、既存の研究で良く用いられる3種類のテキスト情報（単語 N-gram, Paragraph-Vector, Word2Vec）にユーザの評価の傾向を表す素性を加えた評価値推定モデルを構築する。単語 N-gram は Bag-of-Ngram の形式に、Paragraph-Vector に関してはレビュー文を固定次元のベクトル表現に変換し、Support-Vector-Machine (SVM) を用いてモデルを構築する。Word2Vec においてはレビュー中の単語を固定次元のベクトル表現に変換し、Convolutional-Neural-Network (CNN) により単語のベクトルを畳み込んで作成したベクトル表現をもとにモデルを構築する。ユーザの評価の傾向に関してはユーザの評価履歴から求めた評価値ごとのレビューの割合を用い、テキスト情報を表すレビューのベクトル表現に加えることでユーザの評価の傾向を反映させる。この3つのモデルに対し、楽天市場のレビューデータ（ユーザ数：1000, レビュー数：約 70000 件）を適用し、提案する評価の傾向に関する素性の有無についてその有効性を検証する。さらに、どの程度の量のユーザ情報が必要かについて実験的に考察する。

2. 関連研究

レビューの評価値推定では Pang ら [1] の手法に従い、古くからのアプローチとして複数クラスのカテゴリまたは回帰のタスクとして機械学習を利用し問題を解いている。機械学習のパフォーマンスは素性に依存するため、有効な素性の選択が求められ、テキスト情報として Baccianella ら [2] は N-gram, Qu ら [3] は Bag-of-Opinion などの素性を用いている。しかし、これらの研究ではテキスト情報だけに焦点を当てているためレビューを書いたユーザに関する情報が無視されている。この問題に対し近年ではユーザ情報を考慮した評価値推定の手法が提案されている。

ユーザ情報を考慮した評価値推定として Li ら [4] はユーザ固有または製品固有の N-gram を用いることでユーザと商品を考慮した評価値推定の手法を提案している。Tang ら [5] は単語のベクトルに対してユーザ、製品に基づき固有の修正を加えるようなニューラルネットワークのモデルによりユーザと商品の質を考慮した評価値推定の手法を提案している。Gao ら [6] は評価値 1~5 のような 5 値ではなく肯定、否定の 2 値のカテゴリにおいてユーザの寛容さと製品の人気度を考慮した手法を提案している。Gao らは異なるユーザは異なる評価基準を持ち、異なる製品は異なる評価の傾向を受けることに着目している。それに対し、ユーザの評価がどの程度寛容であるかを表す寛容度と製品の人気度を用いたモデルを構築している。

本研究ではユーザの評価が高評価や低評価に偏っている

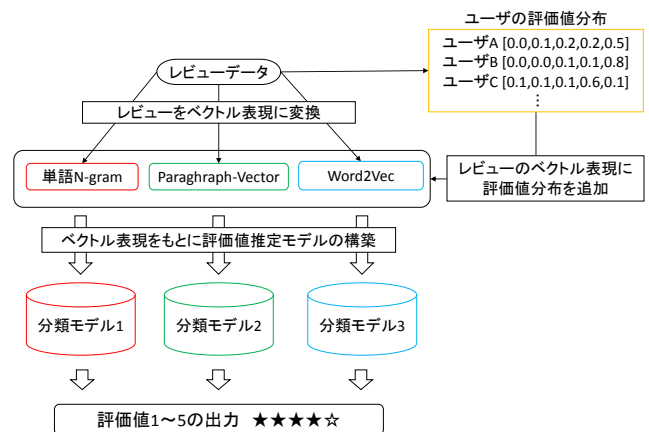


図 1 提案手法の概要図

といった評価の傾向に着目する。あるユーザにおいて、その単語がどの程度肯定的であるかのような単語の重みに着目した Li らや Tang らの手法に対して本研究では、ユーザが高評価を付けやすいといった評価の偏りに着目しており、ユーザ情報として考慮する対象が異なる。Gao らは批判的なユーザは否定的な評価を与えやすい、というようなユーザの肯定、否定の評価に偏りがあることに着目している。そして本研究でも同様にあるユーザが高評価や低評価をつけやすいといった評価の偏りに着目している。Gao らの手法においてはユーザの肯定、否定の評価の偏りを肯定、否定のレビューの割合から算出している。それに対し、本研究では評価値 1~5 の 5 値を推定するタスクであるため、ユーザの評価履歴から評価値ごとのレビューの割合を算出し、その値をユーザの評価の傾向を表す素性としてテキスト情報に加える。

3. 提案手法

本手法では、ユーザ情報としてユーザの評価の傾向を考慮し、評価値を推定する。評価値の推定においてはレビューデータに付与された評価値 1~5 の 5 値分類の問題として扱う。提案手法の流れを以下の図 1 に示す。

提案手法では、まずレビューをベクトル表現に変換する。ベクトル表現にはテキスト情報を表す素性として単語 N-gram ($N = 1, 2, 3$), Paragraph-Vector, Word2Vec を用いる。作成したベクトル表現には、レビューを書いたユーザに対応する評価の傾向を表す尺度（以降、評価値分布）を追加する。そして、各ベクトル表現をもとに評価値推定モデルを構築する。次に、テキスト情報を表す各素性と素性ごとの評価値推定モデルの作成方法、ユーザの評価値分布に関して説明する。

3.1 単語 N-gram

単語 N-gram は、任意の単語が n 個連続した文字列であり、 $n = 1$ で unigram, $n = 2$ で bigram, $n = 3$ で tri-

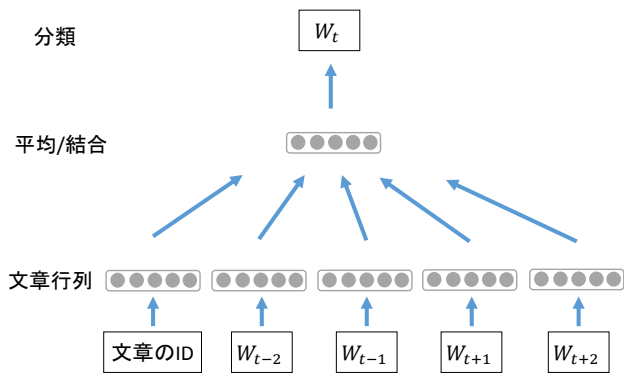


図 2 PV-DM

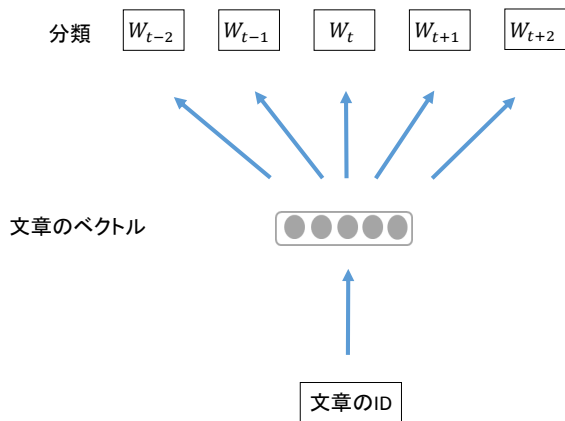


図 3 PV-DBOW

gram である。ベースラインにおいては unigram, bigram, trigram の 3 つにおいてモデルを構築する。モデルの作成においては、まず出現頻度 5 回以下を除いた unigram, bigram, trigram によってレビューをベクトル表現に変換する。レビューのベクトル表現には N-gram の有無で表現する Bag-of-Ngram を用いる。そして、作成したベクトル表現にユーザの評価値分布を表すベクトルを追加し、SVM により評価値推定モデルを構築する。

3.2 Paragraph Vector

Paragraph-Vector は Le ら [7] の提案した文をベクトル表現にする手法である。Paragraph-Vector には Paragraph Vector with Distributed Memory (PV-DM) と Distributed Bag of Words version of Paragraph Vector (PV-DBOW) の 2 種類のモデルがある。PV-DM は図 2 に、PV-DBOW は図 3 にモデルの図を示す。

PV-DM はある文章において、周辺の単語の情報から中心の単語を推定するように学習させたニューラルネットワークにおいて入力層に文章の ID を加えたモデルである。

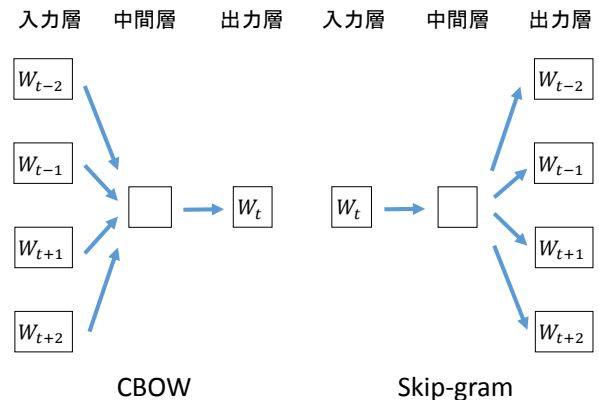


図 4 CBOw, Skip-gram

一方で PV-DBOW は文章の ID から文章に含まれる単語を推定するように学習させたニューラルネットワークのモデルである。評価値推定モデルの作成においては、まず実験に使用するデータに対して PV-DM, PV-DBOW の 2 種類のモデルでレビューをベクトル表現に変換するモデルを学習する。次に学習したモデルを用いてレビュー文を固定次元のベクトル表現 (本研究においては 100 次元) に変換する。最後に作成したレビューのベクトル表現にユーザの評価値分布を表すベクトルを追加し、SVM により評価値推定モデルを構築する。

3.3 Word2Vec・CNN

本研究ではレビュー中の単語をベクトル表現に変換する素性の一つとして Word2Vec を用いる。そして、Word2Vec で固定次元に変換した単語のベクトル表現に CNN を用いて評価値推定モデル構築する (以降、Word2Vec と CNN により構築したモデルは Word2Vec+CNN と表現)。以降では Word2Vec, CNN に関して説明する。

3.3.1 Word2vec

Word2Vec は Mikolov ら [8] によって提案された単語をベクトル表現にする手法である。Word2Vec は Continuous Bag-of-Words (CBOw) と Skip-gram により構成される。それぞれのモデルを以下の図 4 に示す。

CBOw は周辺単語から中心の単語を推定するように学習されたニューラルネットワークであり、逆に Skip-gram は文章のある単語からその周辺単語を推定するように学習されたニューラルネットワークである。本研究における Word2Vec のモデルは Wikipedia の記事に対して Skip-gram を用いて学習したモデルを使用し、単語の次元数は 300 次元とする。

3.3.2 Convolutional-Neural-Network(CNN)

Convolutional-Neural-Network (CNN) は畳み込み処理やプーリング処理を多層に渡って行う畳み込みニューラルネットワークである。本研究では Word2Vec で固定次元に

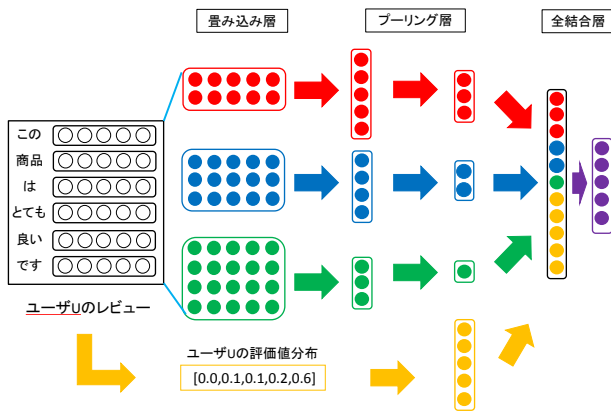


図 5 CNN の概要図

変換した単語のベクトル表現をもとに CNN を用いて評価値推定モデル構築する。CNN の処理の概要となる図を以下の図 5 に示す。

処理の流れはまず、レビュー中の各単語を Word2Vec を用いて固定次元のベクトル表現に変換する。次に畳み込み層においては各単語のベクトルに対して複数のフィルタで畳み込み処理をする。畳み込み処理後のベクトルはプーリング層において max-pooling 処理をし、それぞれのフィルタで作成されたベクトルを結合する。そして、作成されたベクトルにユーザの評価値分布を表すベクトルを追加し、全結合層において出力の次元数に変換する。出力のベクトルに関しては Softmax 関数を適用し評価値ごとの確率値を得る。CNN のモデルのパラメータに関しては窓幅 2, 3, 4 のフィルタを使用し、出力は評価値 1~5 の 5 次元とし学習は 50 epoch 実施する。

3.4 ユーザの評価値分布

本研究ではユーザ情報としてユーザの評価の傾向を用いる。評価の傾向とは、評価に寛容なユーザがどのような商品に対しても高い評価を与える傾向や、厳格なユーザが商品の悪い部分を指摘し、低い評価を与える傾向などである。本研究においては評価の傾向を表す尺度（評価値分布）としてユーザの評価値ごとのレビューの割合がユーザの評価の傾向を表すと仮定して用いる。このような仮定をする理由は、ユーザの評価値 5 のレビューが多いという状況は単純に質の良い商品を買っているだけでユーザが評価値 5 を付けやすいことと異なる場合があるためである。ユーザ U の評価値 i における評価値分布 $Evaldist_{U_i}$ は以下の式により算出する。

$$Evaldist_{U_i} = \frac{N_{U_i}}{N_U}$$

$Evaldist_{U_i}$ はユーザ U のレビュー数 N_U のうち評価値 i のレビュー数 N_{U_i} の割合を表す。以下に評価値 i における

ユーザ U のレビュー数 N_{U_i} と評価値分布 $Evaldist_{U_i}$ の例を示す。

ユーザ A

評価値 i のレビュー数 $N_{A_{1\sim5}}$ [2, 2, 6, 10, 50]

評価値分布 $Evaldist_{A_{1\sim5}}$ [0.02, 0.02, 0.08, 0.14, 0.71]

ユーザ B

評価値 i のレビュー数 $N_{B_{1\sim5}}$ [33, 10, 3, 2, 2]

評価値分布 $Evaldist_{B_{1\sim5}}$ [0.66, 0.20, 0.06, 0.04, 0.04]

ユーザ C

評価値 i のレビュー数 $N_{C_{1\sim5}}$ [12, 11, 13, 11, 13]

評価値分布 $Evaldist_{C_{1\sim5}}$ [0.20, 0.18, 0.21, 0.18, 0.21]

例より、ユーザ A のような評価値 5 のレビューが多いユーザであれば評価値分布の評価値 5 の値が大きくなり、ユーザ B のように評価値 1 のレビューが多いユーザであれば評価値 1 の値が大きくなる。また、ユーザ C のようにどの評価値においても同程度のレビュー数である評価に偏りがないユーザに関しても表現が可能である。このような評価値分布を用いることで、ユーザの評価の傾向を表現する。提案手法においてはテキスト情報を表すベクトル表現に対し、予めレビューデータから算出しておいたユーザの評価値分布をレビュー書いたユーザをもとに追加する。

4. 実験

本実験ではユーザの評価の傾向として評価値分布をテキスト情報に加えた評価値推定モデルの有効性を検証する。提案手法に対する比較対象としてテキスト情報を表す素性のみを用いて評価値推定モデルを構築し、評価値分布の有無における正解率、平均 2 乗誤差平方根 (RMSE) を比較した。素性のうち単語 N-gram, Paragraph-Vector を用いて作成したモデルに関しては 10 分割交差検定により正解率、RMSE を求め、Word2Vec に関しては CNN を用いた 50 epoch の学習の中で最も良かった時の正解率、RMSE を求めた。

実験のデータには楽天市場のレビューデータ*1 からユーザ 1000 人分の 71337 件のレビュー (1 ユーザあたり 50 件~100 件程度のレビュー) を使用し、9 割を訓練データ、1 割をテストデータとして使用する。レビューデータの評価値 1~5 のレビューの内訳はそれぞれ 1084, 1421, 6242, 19589, 43001 件ずつである。

4.1 正解率・平均 2 乗誤差平方根 (RMSE)

本実験においては提案手法の有効性を評価値推定の正解率、平均 2 乗誤差平方根 (RMSE) により比較し検証する。正解率を表す *Accuracy* と平均 2 乗誤差平方根を表す *RMSE* を以下の式により計算する。

$$Accuracy = \frac{\text{評価値を正しく推定できたレビュー数}}{\text{テストデータのレビュー数}}$$

*1 <http://rit.rakuten.co.jp/opendataj.html>

表 1 提案手法の評価値分布の有無における正解率, RMSE

素性	評価値分布無し		評価値分布有り	
	正解率	RMSE	正解率	RMSE
unigram	0.609	0.876	0.663	0.813
bigram	0.570	0.886	0.630	0.826
trigram	0.570	0.894	0.630	0.830
PV-DM	0.615	0.966	0.670	0.875
PV-DBOW	0.619	0.951	0.673	0.866
Word2vec+CNN	0.632	0.840	0.644	0.790
平均値	0.602	0.902	0.652	0.833

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - t_i)^2}$$

正解率を表す *Accuracy* はテストデータに使用するレビューのうち正解したレビュー数の割合を表し、値が大きいほど良い。一方で *RMSE* はテストデータのレビュー数 N 個分の i 番目のレビューに対するモデルの予測値 y_i と訓練データの正解値 t_i の差の平均値であり、値が小さいほど予測値と正解値の誤差が小さく良い。

4.2 実験結果・考察

提案手法の各素性の評価値分布の有無における正解率, RMSE を表 1 に示す。

表 1 より、評価値分布無しの場合と評価値分布有りの場合とで正解率, RMSE を比較すると、どの素性においても評価値分布有りの場合の正解率が大きく、RMSE が小さかった。

各素性ごとに評価値分布の有無における正解数の変化を分析した。テストデータに用いたレビューデータの評価値 1~5 のレビューの内訳はそれぞれ 109, 143, 625, 1959, 4301 件ずつである。以下にそれぞれの素性における評価値ごとの正解数の変化の例を示す。単語 N-gram と Paragraph-Vector に関しては、それぞれ正解率, RMSE が良かった unigram, PV-DBOW の例を示す。

単語 N-gram (unigram)

正解数 (評価値分布無し) [28, 16, 114, 610, 3570]

正解数 (評価値分布有り) [31, 20, 156, 908, 3658]

Paragraph-Vector (PV-DBOW)

正解数 (評価値分布無し) [0, 0, 0, 359, 4066]

正解数 (評価値分布有り) [0, 0, 29, 917, 3835]

Word2Vec+CNN

正解数 (評価値分布無し) [0, 1, 63, 426, 4010]

正解数 (評価値分布有り) [0, 1, 82, 670, 3828]

単語 N-gram (unigram) の評価値分布無しの場合にはデータ数の多い評価値 4 や 5 に正解数の偏りはあるが、どの評価値に関しても満遍なく正解していた。評価値分布有りの場合には全ての評価値において正解数が増加している。

Paragraph-Vector は評価値分布無しの場合にはデータ数の多い評価値 4 と 5 しか正解していなかった。しかし、

評価値分布有りの場合にはデータ数が最も多い評価値 5 の正解数が減り、評価値 3, 4 に関する正解数が増加している。PV-DBOW の評価値分布有りの場合は正解率が全素性の中で最も良かった。

Word2Vec+CNN は評価値分布無しの場合には Paragraph-Vector と同様に正解の大半は評価値 4 や 5 であった。しかし、評価値分布有りの場合にはデータ数が最も多い評価値 5 の正解数が減り、評価値 3, 4 の正解数が増加していた。RMSE に関しては評価値分布無し、有りの場合両方において全素性の中で最も良かった。

これらの結果や全体の平均値からみても評価値分布を考慮することでどの素性においても評価値推定の精度が向上しておりテキスト情報にユーザの評価の傾向を考慮した提案手法の有効性が確認できた。

次に精度が向上した要因として、どのようなユーザにおいて正解数が増加しているかを分析した。分析には評価値分布を用いた時に正解率, RMSE がともに平均値より良かった単語 N-gram (unigram) のモデルを用いた。ユーザの評価値分布を加えたモデルにおける評価値ごとの正解数の変化に関して以下に例を示す。

ユーザ D

評価値分布 [0.00, 0.01, 0.13, 0.68, 0.16]

正解数 (評価値分布無し) [0, 0, 2, 15, 11]

正解数 (評価値分布有り) [0, 0, 1, 44, 3]

ユーザ E

評価値分布 [0.09, 0.05, 0.11, 0.30, 0.42]

正解数 (評価値分布無し) [1, 0, 2, 3, 17]

正解数 (評価値分布有り) [2, 0, 1, 5, 15]

例のように評価に偏りがあるユーザ D (評価値 4 のレビューが約 7 割) のようなユーザにおいてはその偏りに応じて正解数が増加しており (評価値 4 について 15 → 44), このように偏りが大きいユーザであるほど正解数が増加する傾向があった。一方で、比較的に評価の偏りが小さいユーザ E のようなユーザにおいては正解数があまり変化していなかった。

実験に使用したレビューデータが評価値 4 や 5 に偏っており、評価値分布無しの場合にはデータ数が多い評価値 5 の予測が大半である中、評価値分布有りの場合では例のユーザ D のように評価値分布に基づいて評価値 4 の正解数が増加していた。これはユーザの評価値分布に基づいて偏りのある評価値をうまく予測できていることを示していると考えられる。これらの結果から本手法が評価の偏りが大きいユーザであるほど有効な手法であることが確認できた。

4.3 分析：データ量を変化させた時の精度の変化

提案手法がどの程度のデータ量で有効かを確認するためにユーザの評価値分布を算出する際に使用するレビュー数を変化させ、精度の変化を分析した。ユーザの評価値分布

表 2 ユーザあたりのレビュー数を変化させた時の正解率, RMSE

レビュー数	0	10	30	50	全て
正解率	0.6091	0.6630	0.6633	0.6634	0.6631
RMSE	0.8762	0.8123	0.8124	0.8117	0.8129

はレビュー数が 10, 30, 50, 全ての段階に分けて算出し, 正解率, RMSE を比較した. また, 分析には 4.2 節における精度向上の要因分析と同様に単語 N-gram (unigram) の評価値推定モデル (評価値分布有り) を用いた.

ユーザあたりのレビュー数を変化させた時の正解率, RMSE を表 2 に示す. またレビュー数 0 のときはユーザの評価値分布を使用せずに unigram の素性のみを用いて構築したモデルの結果 (表 1 の unigram と同じ) である.

表 2 より評価値分布を算出する際に用いるレビュー数を変化させたがユーザのレビュー全てを使用して評価値分布を求めた場合とあまり差がなかった. 提案手法では 1 ユーザあたり 50 件~100 件程度のレビューを用いているが 10 件のレビューを用いて算出した評価値分布であっても全てのレビューを用いた場合と同等の精度が得られた. これは, 少ないデータ数であってもユーザの評価値分布を再現できているからであると考えられる. このような結果が得られた要因を分析すると評価値 5 の割合が 5 割以上のユーザが 700 人程度, 評価値 4 が 5 割以上のユーザが 100 人程度存在し, 実験に用いたユーザに評価の偏りが大きいユーザが多く含まれていたことが要因であると考えられる. 4.2 節において評価の偏りが大きいユーザである程正解数が増加していたこともふまえると評価の偏りが大きいユーザであれば少ないデータにおいても評価値分布がうまく再現でき評価値推定の精度の向上につながると考えられる.

少ないデータにおいてユーザの評価値分布が求められるならば, 高評価をつけやすいユーザや低評価をつけやすいユーザなど特徴的なユーザを容易に特定できる. そのため, 高評価をよくつけるユーザにおける低評価のレビューであったり, 低評価をよく付けるユーザにおける高評価のレビューなどの価値のある情報を収集するなどデータマイニング的な分析の場面で今回得られた知見が有効利用可能であると考えられる.

5. おわりに

本研究では, テキスト情報にユーザの評価の傾向として評価値分布を加えた評価値推定モデルを構築した. 提案手法ではテキスト情報のみを用いたレビューのベクトル表現にユーザの評価値分布を加え評価値推定モデルを構築し, 実験の結果からユーザの評価値分布を加えることでテキスト情報のみを用いる場合よりも精度が向上した. 評価値推定モデルの正解数の分析からは, 本手法がユーザの評価の偏りが大きい程有効な手法であることが確認できた. 評価値分布の算出に用いるデータ数を変化させた実験からは, 少ないデータ数においてもユーザの評価値分布を求めること

ができ, 評価の偏りが大きいユーザである程ユーザの評価値分布を考慮することの有効性が確認できた.

一方で本手法はユーザごとにレビューを確保できていることを前提としたデータ依存の手法であるため訓練データにはない新規のユーザに対する評価値の推定が課題である. Gao ら [6] らの手法ではこのような問題に対して 2 段階のプロセスを経て文の極性を判定している. 具体的には, 1 段階目で訓練データにはないユーザのデータが含まれる場合に一度テキスト情報のみで極性判定をして, 2 段階目では予測値を用いて新規ユーザの寛容度を算出し, テキスト情報に寛容度を加え再度文の極性判定をしている. これにより訓練データにはないユーザに対しても文の極性判定が可能になる. 本手法においても同様な処理をすることで訓練データにはない新規のユーザにも対応できると考えられる.

本研究においてはユーザの評価値分布がユーザの評価の傾向を表すという仮定に基づいている. そのため, 3.4 節で言及したように評価値分布が評価の傾向を表しておらず商品に依存しているような場合がある. すなわちユーザが評価値 5 を付けやすいのではなく評価値 5 をつけるような質のよい商品をよく購入し, それに対してレビューをしている場合がある. この問題に関しては, Tang ら [5] のようにレビューの書かれた商品の質を考慮することで評価値の分布がユーザの評価の傾向を表しているかどうか分かると考えられる. 本研究の今後の課題は製品の質と評価の傾向を考慮した評価値推定モデルを構築することである.

参考文献

- [1] Bo Pang, Lillian Lee. Seeing stars : Exploiting class relationships for sentiment categorization with respect to rating scales. pp 115-124, ACL(2005).
- [2] Stefano Baccianella, Andrea Esuli, Fabrizio Sebastiani. Multi-facet Rating of Product Reviews. pp461-472, ECIR(2009).
- [3] Lizhen Qu, Georgiana Ifrim, and Gerhard Weikum. The Bag-of-Opinions Method for Review Rating Prediction from Sparse Text Patterns. pp 913-921, COLING(2010).
- [4] Fangtao Li, Nathan Liu, Hongwei Jin, Kai Zhao, Qiang Yang, Xiaoyan Zhu. Incorporating Reviewer and Product Information for Review Rating Prediction. pp 1820-1825, IJCAI(2011).
- [5] Duyu Tang, Bing Qin, Ting Liu, Yuekui Yang. User Modeling with Neural Network for Review Rating Prediction. pp 115-124, IJCAI(2015).
- [6] Wenliang Gao, Naoki Yoshinaga, Nobuhiro Kaji, Masaru Kitsuregawa. Modeling User Leniency and Product Popularity for Sentiment Classification. pp 1107-1111, IJCNLP(2013).
- [7] Quoc Le, Tomas Mikolov. Distributed Representations of Sentences and Documents. pp 1188-1196, ICML(2014).
- [8] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. arXiv:1301.3781(2013).