

ベンフォードの法則によるフェイクレビュー検出手法の検討

NECO B2 江頭叙那 (jonah)

親 ks91

2021 年 1 月 27 日

1 はじめに

ネットショッピングが普及し、多くの人々がオンラインで買い物をするようになった。商品のレビューはユーザーが購入を検討する際に大きな要因となるが、企業が金銭を見返りに偽のレビューを書かせるフェイクレビューが問題になっている。

2 目的

ベンフォードの法則を応用した素性を利用して、既存方式よりも少ない情報から、商品にフェイクレビューが含まれているかどうかを検出することを目指す。

3 ベンフォードの法則の概要

ベンフォードの法則とは、電気料金の請求書、住所の番地、株価、人口、革の長さ、物理・数学定数などの自然界に現れる数値集合の各数値における最上位桁の数値の出現確率には偏りがあるという法則である。この分布から大きく離れている場合、何らかの人手による操作などが行われた可能性がある。[1]

4 仮説

偽のレビューを書くレビュアーは実際に商品を使っておらず、さらに報酬を得るために短期間で多くのレビューを書く必要があるため、他のレビューを参考に似た文章を書くケースが多いのではないかと考える。よって、偽のレビューを含んでいない商品と比べて使用される文字の頻度の分布が異なるのではないかと考えた。

5 既存方式

Amazon のフェイクレビューを見抜くサクラチェッカー [2] では、価格・製品、レビュー分布やショッピングレビューなど計 8 項目の情報を、独自のロジックや機械学習を用いて分析している。

6 既存方式の課題

サクラチェッカーでは Amazon のレビューしか判定することができず、他サイトのレビュー判定ができない。また、分析項目が多く他サイトに応用しづらい点が課題である。

7 提案方式

1. サクラチェッカーでサクラ度が 20% 以下の商品を安全な商品、80% 以上の商品を危険な商品とし、2 グループそれぞれ 200 商品のレビューを Selenium によって取得する。[5]
2. 取得したレビューの文字の出現頻度をカウントし、頻度の最上位桁の割合を計算する。
3. 安全な商品と危険な商品のグループで、最上位桁の分布がどのように異なるかを分析する。
4. 分析結果をもとに商品にフェイクレビューが含まれているかどうかを判定する

8 結果

サクラチェッカーによる安全な商品と危険な商品のレビューをそれぞれ 200 商品ずつ取得し、出現する文字の頻度の最上位桁を取得した結果が以下のグラフである。

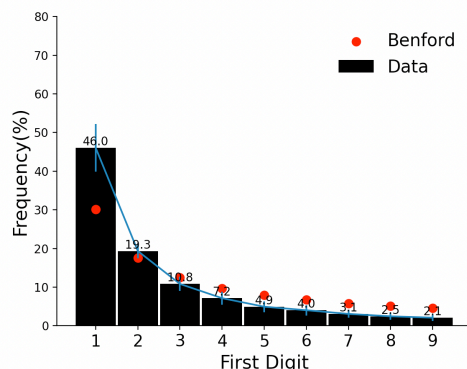


図1 危険な商品の文字頻度の分布

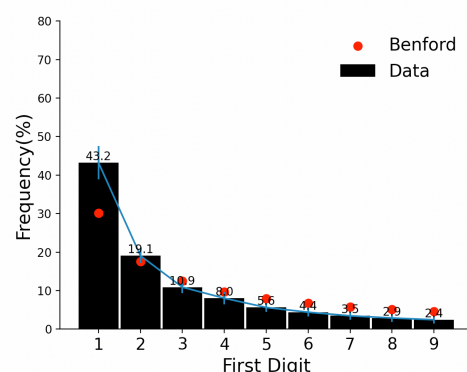


図2 安全な商品の文字頻度の分布

1の位において、危険な商品のほうが出現頻度が高く標準偏差も大きかったが、差は僅かであった。そのため、レビューの文字出現頻度からフェイクレビューが含まれるかを判定することはできなかった。また、janomeによる形態素解析によって単語に分解したものの頻度を分析しても、2つの商品グループに僅かな差しか認められなかった。

9 考察

安全な商品と危険な商品の文字の出現頻度の最上位桁の割合に差がほとんどなかった理由は以下の3つであると考えた。

1. フェイクレビューが他のレビューを参考に似た文を書いているという仮説が正しくなかった。
2. 商品あたりのレビューが少ない場合に、頻度が

偏ってしまった。

10 今後の展望

ベンフォードの法則を用いてレビューのみからフェイクレビューを検出することは困難であると考えた。今後は機械学習の学習を進めて様々な観点からのフェイクレビュー検出を検討していきたい。

参考文献

- [1] ベンフォードの法則
<https://ja.wikipedia.org/wiki/ベンフォードの法則/> (参照 2021/1/27)
- [2] サクラチェッカー
<https://sakura-checker.jp/> (参照 2021/1/27)
- [3] Lee Vaughan, 高島亮祐訳, “実用的でないPythonプログラミング”, 第1版, 共立出版(2020)
- [4] 蔵内 雄貴 他, “ベンフォードの法則を応用したbotアカウント検出”, 日本電信電話株式会社 NTT サービスエボリューション研究所 (2013)
- [5] ジコログ “Amazon のスクレイピング対策を攻略する”
[https://self-development.info/amazon のスクレイピング対策を攻略する【selenium 最強説】](https://self-development.info/amazonのスクレイピング対策を攻略する【selenium最強説】/) / (参照 2021/1/27)
- [6] うえぶのきわみ, IkeSei, “python で日本語の記事に登場する単語の出現数を調べる方法”
<https://web-kiwami.com/count-words-article-python.html> (参照 2021/1/27)