WIP 2020F

ベンフォードの法則を用いたフェイクレビュー検出の検討

NECO B2 jonah

今学期取り組んだこと

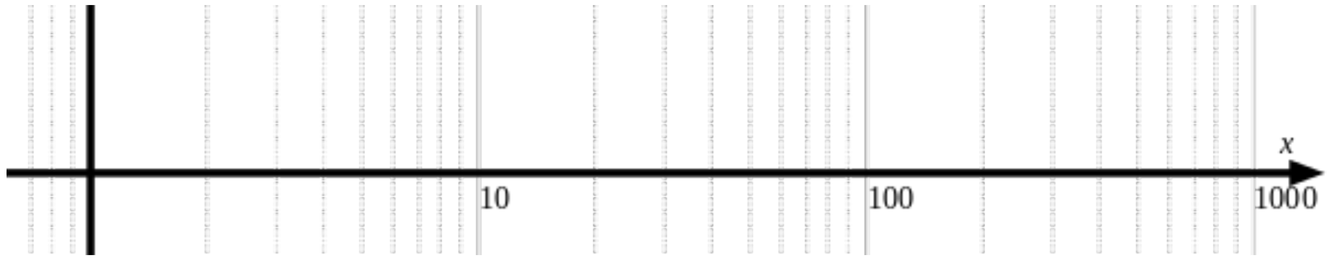
- 実用的でないPythonプログラミング
- ゼロから作るDeep Learning
- Discordでbot作成
- 競技プログラミング
- **ベンフォードの法則を用いたフェイクレビュー検出の検討**

背景

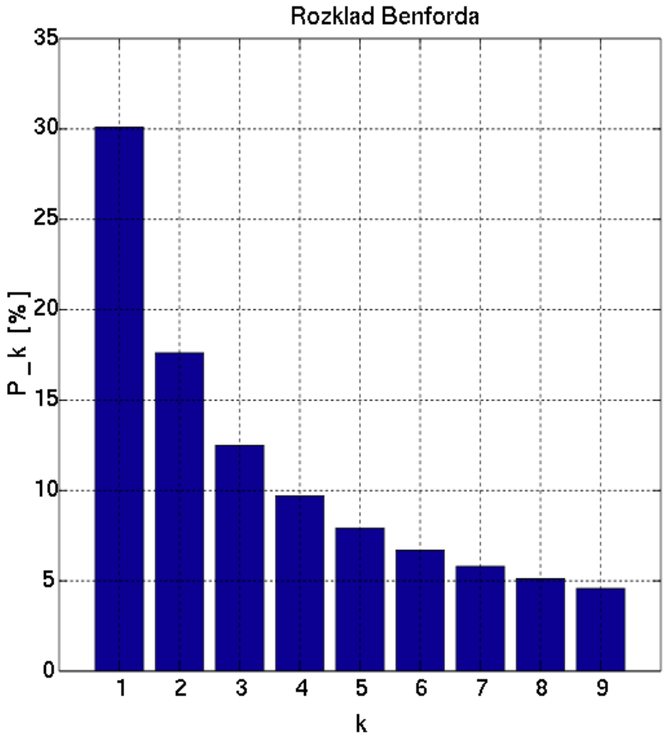
- 通販サイトでのフェイクレビュー問題
- Amazonのフェイクレビューを見抜くサービスしか存在しない
- 商品にフェイクレビューが含まれているかどうかを判定するのは難しい
- **ベンフォードの法則を用いてレビューの文字の出現頻度という性質のみから判断したい**

背景
ベンフォードの法則について

- 自然界に発生する数の分布における**先頭**の数字の出現頻度は特定の分布になっている
- 人口，川の長さ，物理定数など
- 対数スケールなどによって説明される



対数スケールのグラフにランダムに点を取ると
最初の桁が１になる確率がおよそ30%



ベンフォードの法則に従った場合の分布

背景
仮説

- フェイクレビュアーは短時間で多くのレビューを書く必要がある.
- フェイクレビュアーは実際に商品を持っていない.
- フェイクレビュアーは他のレビューを参考に書くと考えられ，その結果文字の出現頻度の最上位桁の分布は自然に書かれたレビューの分布と異なると考えた.

既存方式

- サクラチェッカーでは価格，レビュー，投稿日時やショップの評価など8つの項目を，機械学習などを用いて分析している．

  課題：サクラチェッカーはAmazonのみの対応で，評価項目が多いため他サイトへ応用しづらい．

提案方式
1/2

- サクラチェッカーによってサクラ度が高い商品と低い商品*のレビューの文字の頻度を分析する。

*サクラチェッカーの判定でサクラ度が20%以下のものをサクラ度が低い（安全な商品）とし，サクラ度が80%以上のものをサクラ度が高い（危険な商品）とした.

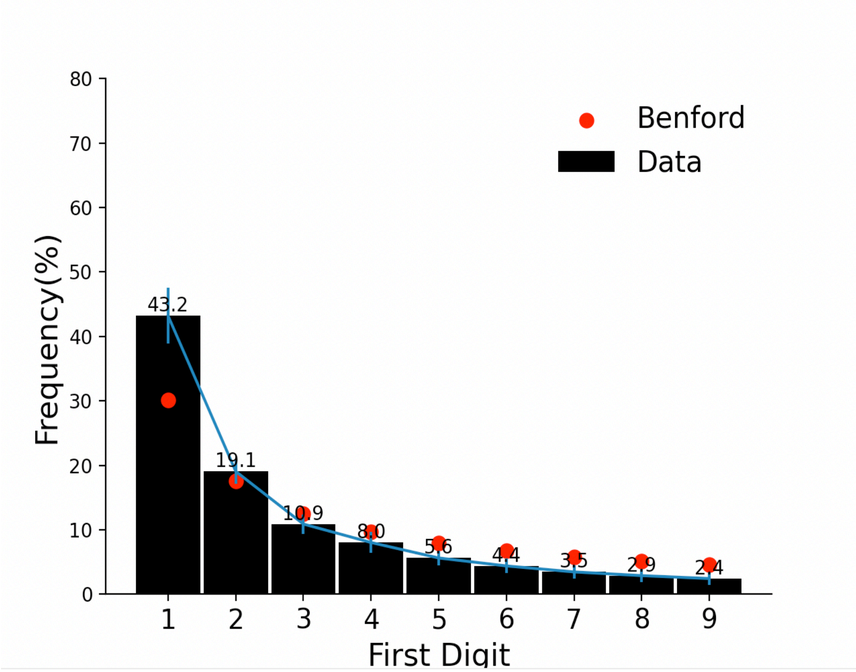
提案方式
2/2

- 文字の出現頻度の最上位桁の割合をもとにサクラ度が高いレビュー群と低い群の素性を分析する.

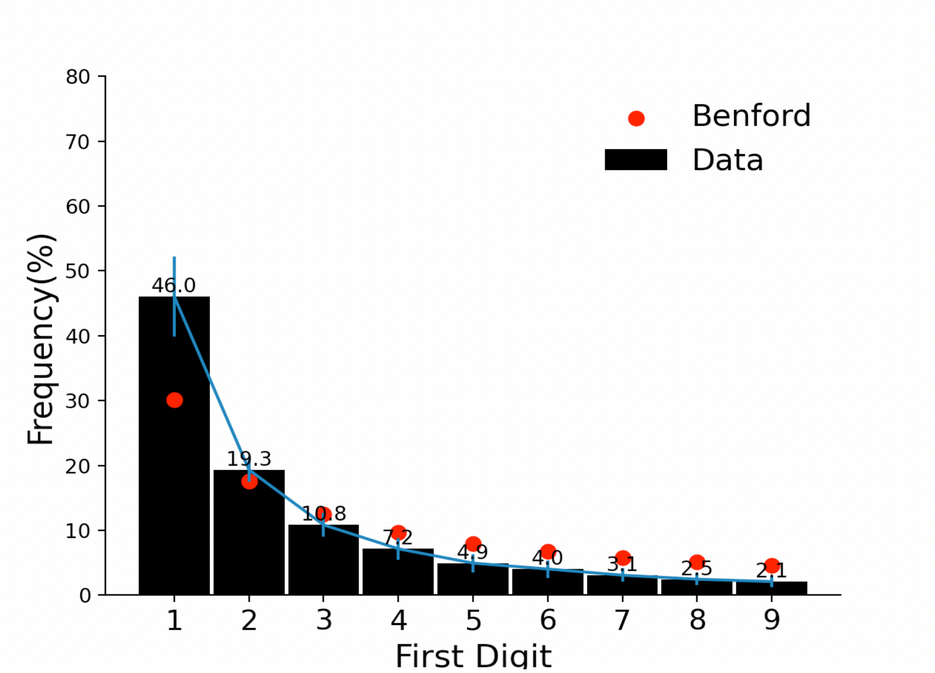
- 結果をもとに、ある商品のレビュー集合を入力としてサクラチェッカーでのサクラ度が高いか，低いかを判別し出力する.

結果
1/2

安全な商品と危険な商品でそれぞれ200商品のレビューを分析した.



安全な商品



危険な商品

結果
2/2

- 危険な商品と安全な商品のレビューの結果に差はほとんど見られなかった
- どちらのグループもベンフォードの法則の分布と比べ１の位の頻度が高かった
- ２つのグループに差が見られなかったため，商品の検証に至らなかった

考察
うまくいかなかった原因

1. フェイクレビュアーが他のレビューを参考にして似た文章を書いているという仮説が正しくない

2. 似た文章が多い場合でも，文字頻度の最上位桁の分布が変わらない可能性がある

3. 一商品あたりのレビューが少ないと文字数が足りずに正しい分布が得られない

4. もっと多くの商品のレビューを集めたかったが，サクラチェッカーがスクレイピング禁止で困難だった

学べたこと

- SeleniumによるWebスクレイピング
- Numpyやmatplotlibを用いた分析とチャートの作成
- janomeを用いたPythonによる形態素解析
- LaTeXによる文章作成

今後の展望

- ゼロから作るDeep Learningを一通り読んだ
- 機械学習の勉強を進めて機械学習によるフェイクレビューの検証や，Kaggleにも挑戦してみたい
- 競技プログラミングをもっと真剣に取り組みたい

参考

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】/ (参照 2021/1/27)

⑥ うぇぶのきわみ, 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)