F-025

局所改善法によるレビュー変化点検出法
Detecting Change-Points by Local Improvement in Time Series Data of Reviews

山岸祐己 † 斉藤和巳 † 大久保誠也 †
Yuki Yamagishi Kazumi Saito Seiya Okubo

1. はじめに

オンラインレビューサイトとは，商品やサービスについてのレビューを投稿することができるウェブサイトの総称である．レビューは点数・文章・画像から成ることが多く，レビュー点数の平均点が一般的な評価指標として扱われている．オンラインレビューサイトについては，既に多様な分析や研究が展開されている [1]．近年，オンラインレビューサイトにおけるユーザーのレビュー行動が非常に活発であり，サイトそのものが商品やサービスのプロモーションを左右する重要なメディアになりつつある．

しかし，殆どのレビューサイトが投稿回数制限やレビュー内容の吟味を行なっていないため，自己主張の激しいユーザーによる極端な評価が書かれたレビューが執拗に飛び交っているのが現状である．あまりにも肯定的なレビューは，商品の製造会社や関係会社が意図的に書いたのではないかと疑われ，あまりにも否定的なレビューは，競合他社や個人の嫌がらせとして見做される場合もある．結果的に，商品やサービスに対して付けられる「平均得点」という評価指標の価値が希薄化してしまい，さらには，金銭を受け取って好意的なレビューを書いたり書かせたりする「さくら」や「やらせ業者」の特定も相次いでいるため，オンラインレビューサイトに対する不信感は増すばかりである．上記のような意図的に評価を調整しようとするレビューやユーザーは，そうでないレビューやユーザーとは区別する必要があるため，オンラインレビューサイトにおけるレビューの変化点検出は重要な研究課題と言える．

本論文では，Swan と Allan [2] や Kleinberg [3] と同様に，回顧的 (Retrospective) な立場で異常を検出する新たな手法を提案する．我々は既に，ユーザーの基本評点行動として多項分布モデルを仮定し，尤度比検定により異常期間を検出することを特徴とする単一区間抽出法を提案している [4]．また，この手法を拡張し，複数区間の抽出を可能とする手法も提案している [5]．本稿では，このような状況を複数の変化点の検出問題としてとらえ，局所改善を導入することで，解品質の向上度合いを現実データを用いて評価する．

2. 変化点検出法

2.1. 問題設定

評点の時系列データを $\mathcal{D} = \{(a_0, t_0), \cdots, (a_N, t_N)\}$ とする．ここで各評点は，1 から J の整数値で与えられるとする．即ち，$a_n \in \{1, \cdots, J\}$ となる．t_n はそれぞれの評点が与えられた時刻を指す．評点付けの基本モデルとして多項分布を仮定し，評点 j が与えられる確率 p_j に従うとする．ただし，評点付けモデルには複数の変化点が存在するとする．いま，その変化点を T_k とする．ここで，$t_0 < T_k < t_N$ である．このとき，評点 j が与えられる確率については，変化点 T_k の直前までは $p_{k,j}$，そして変化点 T_k 直後では $p_{k+1,j}$ というパラメータの多項分布に従うとする．いま，K 個の時刻から構成される変化点集合を $\mathcal{S}_K = \{T_1, \cdots, T_K\}$ とし，便宜上 $T_0 = t_0$ かつ $T_{K+1} = t_N$ と設定しておく．また，$T_{k-1} < T_k$ であるとし，$\mathcal{S}_K$ による $\mathcal{D}$ の分割を $\mathcal{D}_k = \{(a_n, t_n); T_{k-1} < t_n \leq T_k\}$ で定義する．すなわち，$\mathcal{D} = \mathcal{D}_1 \cup \cdots \cup \mathcal{D}_{K+1}$ となり，$|\mathcal{D}_k|$ は区間 $(T_{k-1}, T_k]$ に含まれる観測時刻数を表す．ここで，任意の $k \in \{1, \cdots, K+1\}$ に対して，$|\mathcal{D}_k| \neq 0$ とする．一方，パラメータのベクトルを $\mathbf{P}_K = (p_{1,1}, \cdots, p_{1,J}, \cdots, p_{K,1}, \cdots, p_{K,J})$ で定義すれば，変化点集合 $\mathcal{S}_K$ が与えられたときの観測データ $\mathcal{D}$ に対する対数尤度は次式で計算できる．

$$L(\mathcal{D}; \mathbf{P}_{K+1}, \mathcal{S}_K) = \sum_{k=1}^{K+1} \sum_{(a_n, t_n) \in \mathcal{D}_k} \log p_{k,a_n} \qquad (1)$$

よって，式 (1) の尤度を最大にするパラメータの最尤推定値を $\hat{\mathbf{P}}_{K+1}$ とすれば，我々の変化点検出問題は，$L(\mathcal{D}; \hat{\mathbf{P}}_{K+1}, \mathcal{S}_K)$ を最大化する変化点集合 $\mathcal{S}_K$ を求める問題となる．ただし，変化点集合 $\mathcal{S}_K$ の導入効果を直接評価するため，この問題の別表現として，尤度比検定の目的関数として我々の変化点検出問題を定式化する．つまり，変化点が存在しないとした $\mathcal{S}_0 = \emptyset$ とし，変化点が K 個存在するとしたときと，存在しないとしたときの尤度比の対数を次式で定義する．

$$LR(\mathcal{S}_K) = L(\mathcal{D}; \hat{\mathbf{P}}_{K+1}, \mathcal{S}_K) - L(\mathcal{D}; \hat{\mathbf{P}}_1, \mathcal{S}_0) \qquad (2)$$

本論文では，式 (2) で定義した $LR(\mathcal{S}_K)$ を最大化する変化点集合 $\mathcal{S}_K$ を求める問題を考える．以下では，その解法としてシンプル法と提案法について述べる．

2.2. シンプル法

既に選定した $(k-1)$ 個の変化点集合 $\mathcal{S}_{k-1}$ を固定し，新たに付加するとして最適な変化点 T_k を求め $\mathcal{S}_{k-1}$ に加えることを，$k = 1$ から K まで繰り返す．すなわち，そのアルゴリズムは以下となる．まず，$k = 1$, $\mathcal{S}_0 = \emptyset$ と初期化する．次に，$T_k = \arg\max_{t_n} \{LR(\mathcal{S}_{k-1} \cup \{t_n\})\}$ を求め，$\mathcal{S}_k = \mathcal{S}_{k-1} \cup \{T_k\}$ と更新する．もし $k = K$ なら $\mathcal{S}_K$ を出力して終了し，さもなければ $k = k+1$ とし，処理を繰り返す．ただし，変化点集合 $\mathcal{S}_k$ の要素は $T_{i-1} < T_i$ となるようインデックスを更新するとする．ここで，$i = 2, \cdots k$ である．明らかに，シンプル法の計算量は $O(NK)$ となる．すなわち，ある程度 N が大きくなっても，任意の K に対して，実用的な時間で結果を得ることができる．しかしながら，シンプル法は貪欲法に基づく手法であるため，比較的プア一な局所解にトラップされるケースも危惧される．

†静岡県立大学, University of Shizuoka

2.3. 提案法

前述のシンプル法と同程度の計算量で，解品質の向上を目的とした提案法について述べる．この方法では，シンプル法で求めた変化点集合 $\mathcal{S}_K$ を出発点として，既に選定した一つの変化点 T_k を選び，これ以外の変化点集合 $\mathcal{S}_K \setminus \{T_k\}$ を固定し，より望ましい別の変化点 T'_k に置き換えることを繰り返す．ここで，$\cdot \setminus \cdot$ は集合差を表す．明らかに，$k = 1, \cdots K$ のすべてで置き換えできなければ，すなわち，どの k でも $T'_k = T_k$ となれば，この方法でさらなる改善はできないので反復を終了させるとする．すなわち，そのアルゴリズムは以下となる．まず， $\mathcal{S}_K$ をシンプル法で求め，$k = 1, h = 0$ と初期化する．次に，$T'_k = \arg\max_{t_n}\{LR(\mathcal{S}_k \setminus \{T_k\} \cup \{t_n\})\}$ を求め，$T'_k = T_k$ なら $h = h + 1$, さもなければ $h = 0$ とし，$\mathcal{S}_K = \mathcal{S}_K \setminus \{T_k\} \cup \{T'_k\}$ と更新する．もし $h = K$ なら $\mathcal{S}_K$ を出力して終了し，さもなければ $h = h + 1$ とし，処理を繰り返す．明らかに，シンプル法と比較して，一般に，提案法は数倍の計算量が必要となる．ただし，計算量がどの程度増加するかとともに，解品質がどの程度改善するかは問題に依存する．

3. 実験による評価

3.1. データセット

今回使用するデータセットは，食ベログ[1]のレビューデータ及び価格.com[2]のレビューデータである．食ベログとは，カカクコムグループ[3]が運営するグルメサイトであり，2005 年 3 月にサービスが開始された．このデータセットは，2012 年の 2 月に食ベログをクロールして取得したものであり，449447 レストラン，301086 ユーザー，3114507 レビューを有する．レビュー点数は，1〜5 まで 0.5 点刻みで付けることができる．

価格.com とは，同じくカカクコムグループが運営する価格比較サイトであり，サイトの原形が創設されたのは 1997 年 5 月である．このデータセットは，2012 年の 6 月に価格.com をクロールして取得したものであり，79586 アイテム，171806 ユーザー，482452 レビューを有する．レビュー点数は 1〜5 までの整数値である．

3.2. 実験結果

図 1 と図 2 に，食ベログと価格.com のレビューデータのアイテムに対し，各変化点の個数 $K \in \{2, 3, 4, 5\}$ でのナイーブ法と提案法の性能差を示す．ここでは，それぞれのデータのアイテムにおいて，レビュー数の多い順に ID を 1 から順に付与し，上位 100 件のアイテムで評価結果である．図より，どちらのデータでも，アイテムと K の設定のペア毎に，性能差は大幅に変化することが見て取れる．一方，シンプル法と比較した提案法の改善度は，$K = 2, 3, 4, 5$ のそれぞれで，食ベログでは $0.5637, 1.1714, 1.4971, 1.9113$ となり，価格.com では $0.3425, 0.7790, 0.8539, 1.0025$ となった．明らかに，K の設定を大きくすると，性能差が増大する傾向が観測された．これらの結果より，シンプル法と比較して提案法の有用性が示唆されたと考える．

[1]http://tabelog.com/
[2]http://kakaku.com
[3]http://corporate.kakaku.com/

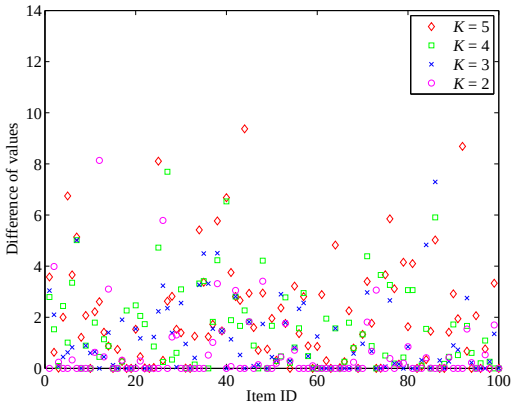

図 1: 食ベログのレビューデータでの性能差

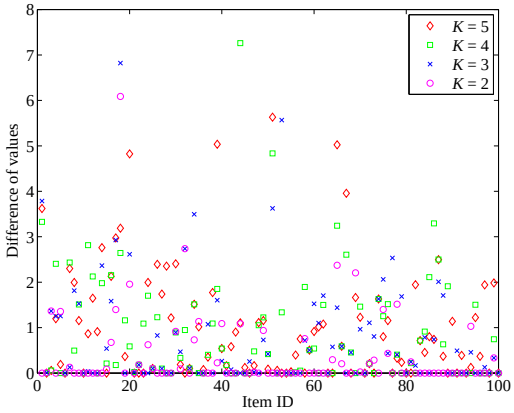

図 2: 価格.com のレビューデータでの性能差

謝辞

本研究は，NTT 未来ねっと研究所との共同研究，及び，科学研究費補助基金基盤研究 (C)(No. 23500312) の支援を受けて行ったものである．

参考文献

[1] M.J.Salganik, P.S.Dodds, and D.J.Watts, ”Experimental Study of Inequality and Unpredictability in an Artificial Cultural Market”, Science 10, pp.854-856, February 2006.

[2] R.Swan and J.Allan, ”Automatic Generation of Overview Timelines”, SIGIR 2000, pp.49-56, 2000.

[3] J.Kleinberg, ”Bursty and Hierarchical Structure in Streams”, KDD 2002, pp.91-101, 2002.

[4] 山岸 祐己，斉藤 和巳，大久保 誠也，”オンラインレビューサイトの評点時系列データからの異常検出”，第 4 回データ工学と情報マネジメントに関するフォーラム（DEIM2012），2012.

[5] 山岸 祐己，斉藤 和巳，大久保 誠也，”レビュー時系列データからの分割統治による変化点検出法”，第 24 回人工知能学会全国大会（JSAI2012），2012.