2006 年情報論的学習理論ワークショップ
2006 Workshop on Information-Based Induction Sciences (IBIS2006)
Osaka, Japan, October 31 - November 2, 2006.

行列の圧縮による変化点検出の高速化

Speeding up Change-Point Detection using Matrix Compression

井手剛*
Tsuyoshi Idé

Abstract: We present an approach to speed up a change-point detection algorithm called singular-spectrum transformation (SST). SST computes the change-point score at each time by (1) performing feature extraction from time-series subsequences and (2) evaluating the discrepancy between the feature vectors. In this paper, we show that the feature extraction step can be done implicitly by utilizing mathematical properties of tridiagonal decomposition of matrices. We experimentally show that this implicit algorithm dramatically reduces the computational cost of SST.

Keywords: time series, change point, Krylov subspace

1 はじめに

時系列データの変化点とは、抽象的にはデータ生成機構にある変化が生じた時点と定義され、実数値データについて言えば、折れ曲がり、段差、周波数変動、振幅変動、分散の変化、等々、さまざまな現れ方をする。時系列データの変化は、工学的に重要な情報を含んでいることが多い。しかもその検出が一筋縄ではいかないことから、知識発見工学の分野でのひとつの重要な研究領域になっている。

変化点検出問題自体は、統計学的には大変古い問題である [1]。それが近年再び注目を浴びている背景には、産業の多くの分野で大量のデータを容易に入手できるようになったこと、そしてそこからの知識発見に取り組む中で、古典的変化点検出手法の限界が認識されるようになったことがある。

変化点検出において重要なことは、変化の多様性に対応することである。例えば、ある波長を持つ正弦波が、ある時点から別の波長に遷移したとする。このような変化は、入力が正弦波であるとわかっていれば、フーリエ係数を監視することで容易に検出することができよう。しかし工学的に興味ある多くの場合、不完全な事前知識に基づいて、しかも非定常な時系列を扱う必要がある。古典的な変化点検出手法 [1, 7] のほとんどは、ある程度「おとなしい」振る舞いをする入力が陰に陽に想定されており、実用性にやや限界があった。

Yamanishi-Takeuchi（YT）はこれに対し、混合正規分布の逐次更新型 EM (expectation-minimization) 学習と自己回帰モデルを組み合わせて、かなり広い範囲の変化点を、しかも非定常な入力に対して可能にする枠組みを提案した [9, 10]。これは現在のところ、実用的に最も成功しているモデルと言えるが、混合正規分布および自己回帰モデルという特定のパラメトリックモデルに基づいているため、たとえば混合数の選択や、局所最適性の問題など、モデル特有の制約から自由というわけではない。

本論文では、特異スペクトル変換（singular spectrum transformation; SST）と呼ばれる変化点検出手法 [5] を考える。YT モデルと対比的に言えば、SST はノンパラメトリックな変化点検出手法である。SST は、各時刻において、過去側と未来側の部分系列からの特徴抽出を行い、抽出された特徴ベクトル同士の食い違いをもって変化度スコアとする。SST という名前は、特徴ベクトル抽出に、特異値分解（singular value decomposition; SVD）を使うことに由来する。SST は特定の確率モデルを仮定しないので、入力時系列の多様性に比較的頑強であり、局所解の心配もない。しかしよく知られているように、SVD は計算量的に非常に重い操作であり、この欠点が多くの利点を消し去るに余りあった。

本論文では、SST の高速化手法について考える。変化点検出問題を、特徴ベクトル同士のカーネル（内積）の計算問題に帰着させ、そのカーネルを注意深い近似手法を用いて高速に計算する。以下、SST の説明（2 節）、

*IBM 東京基礎研究所, 242-8502 神奈川県大和市下鶴間 1623-14, e-mail: goodidea@jp.ibm.com.
IBM Research, Tokyo Research Labotarory, 1623-14 Shimo-Tsuruma, Yamato-shi, Kanagawa 242-8502, Japan.

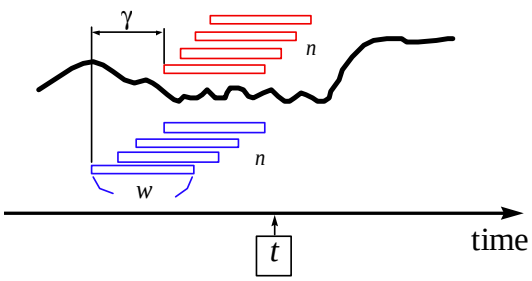

図 1: SST におけるパラメターの定義。

SVD の諸手法のスケッチ（3 節）、提案手法の導出（4 および 5 節）、実験結果（6 節）、まとめ（7 節）と続く。

2 特異スペクトル変換

等間隔の時刻で観測された時系列データ $\{x_t \in \mathbb{R} \mid t = 1, 2, ...\}$ に対して、長さ w の部分系列を、$\mathbb{R}^w$ の列ベクトルとして

$$\boldsymbol{s}(t) \equiv (x_{t-w+1}, ..., x_{t-1}, x_t)^\top$$

のように定義する。上付きの $\top$ は転置を表す。ある時刻 t において、部分系列を n 本ならべた行列 H_1、H_2 を次のように定義する[1]。

$$\begin{aligned} \mathsf{H}_1(t) &\equiv [\boldsymbol{s}(t-n), ..., \boldsymbol{s}(t-2), \boldsymbol{s}(t-1)] \\ \mathsf{H}_2(t) &\equiv [\boldsymbol{s}(t-n+\gamma), ..., \boldsymbol{s}(t-1+\gamma)] \end{aligned}$$

ただし γ はある正整数である。参考のため、図 1 では、過去側、未来側（もしくは現時刻近傍）共に 3 本の部分系列を取り出す様子が描かれている。

$\mathsf{H}_1(t)$ と $\mathsf{H}_2(t)$ の列空間はそれぞれ、時刻 t をまたいで過去側と未来側の近傍における特徴的なパターンを含んでいるはずである。SST では、これらの行列から特徴抽出を SVD により行う。$\mathsf{H}_1(t)$ において特異値の大きい順に左特異ベクトルを $r(< w, n)$ 個求め、それを $\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}, ..., \boldsymbol{u}^{(r)}$ と表す。以下、誤解のない限り引数 t を省略する。特異ベクトルは L_2 ノルムが 1 に規格化されているとする。$\mathcal{H}_r \equiv \mathrm{span}\{\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}, ..., \boldsymbol{u}^{(r)}\}$ とおき、$\mathcal{H}_r$ を履歴空間と呼ぶ。また、H_2 から最大左特異ベクトル $\boldsymbol{\mu}$ を求める[2]。

文献 [5] に従えば、t における変化度は、

$$1 - \boldsymbol{\mu}^\top \mathsf{U}^\top \boldsymbol{\mu} / ||\mathsf{U}^\top \boldsymbol{\mu}|| \qquad (1)$$

で計算される。ただし、$\mathsf{U} \equiv [\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}, ..., \boldsymbol{u}^{(r)}]$ である。第 2 項は、超平面 $\mathcal{H}_r$ と $\boldsymbol{\mu}$ との角度の余弦であり、両者の角度をもって変化の度合いとしていることになる。

経験的に変化度は n と γ の選択にさほど敏感ではないことがわかっているので [5]、$n = w$、$\gamma = n/2$ と固定する（奇数なら切捨てる）。すると与えるべきパラメターは、w と r の二つである。下記に、SST のアルゴリズムをまとめた。

Algorithm 1 *(SST [5])* ある w と r を与える。各 t において以下を行う。

1. H_1 の左特異ベクトルの特異値が大きいものから r 個求める。
2. H_2 の最大左特異ベクトル $\boldsymbol{\mu}$ を求める。
3. 式 *(1)* から変化度を求める。

3 関連技術

SST の明らかな問題は、計算が遅いことである。悪いことに、H_1 と H_2 は通常、すべての要素に有限の値を持つ**密行列**である。疎行列の場合なら、行列のサイズではなく、非ゼロの要素数に計算量が連動するような工夫がさまざま可能で、実際に情報検索の分野で広く使われている。しかし密行列には妙手はなく、通常、行列サイズ分の計算量を甘受せざるを得ない。

例えば、いわゆる thin-QR 分解 [3] で SVD を行う場合、$w = n$ の状況では、ある t において変化度を求めるためには $\mathrm{O}(rw^2)$ の計算量が必要である。機械学習の分野で知られている最速の PCA (principal component analysis) 手法は、おそらく EM-PCA [8] であるが、これを SVD に流用したとしても計算量のオーダー自体は変わらない。収束速度が多少改善される程度である。

ひとつ考えられるのは、$t-1$ での計算結果を、t での計算に利用することである。SVD のオンライン更新法については情報検索の分野でよく調べられている。代表的なものは Zha-Simon [11] のアルゴリズムである。これは、協調フィルタリング [2] や、視覚追跡 [6] にも応用されている。本論文の記号を使えば、これは、$\mathsf{H}_1(t)$ が $\mathsf{H}_1(t-1)$ の階数 1 更新と見なせることを利用し、問題を、階数 1 更新された対角行列の SVD に帰着させるものである。

しかし Zha-Simon の方法は、$\mathcal{H}_r(t)$ の構成のために $\mathcal{H}_r(t-1)$ を直接使う近似アルゴリズムである。low-rank-plus-shift と呼ばれる特殊な構造を持つ行列には近似誤差が非常に小さくなることが示されているものの [12]、この条件は SST においては通常満たされない。より直

[1] この定義だと $\mathsf{H}_2(t)$ に未来のデータが含まれるので、実時間計算の時は、過去のデータだけで計算できるように適当に定義を読み替える必要がある。

[2] $\{x_t\}$ がおおむね正の値を取るようにデータの前処理を行うものとする。これにより、少なくとも最大特異値における縮退を避けることができて、数値計算が安定化される。

感的に言えば、Zha-Simon のアルゴリズムは、SVD で取り出すべき latent structure が大きく時間変化しないことを前提にしている。変化点検出アルゴリズムに使う近似手法としては、原理的な困難を抱えていると言えよう。

4 Implicit kernel 近似

まず変化度を、式 (1) からやや変更して、次のように書き表す。

$$z = 1 - \sum_{i=1}^{r} K(i, \boldsymbol{\mu})^2 \tag{2}$$

ただし、K は

$$K(i, \boldsymbol{\mu}) \equiv \boldsymbol{\mu}^\top \boldsymbol{u}^{(i)} \tag{3}$$

で定義される内積（もしくはカーネル）である。これは、$\boldsymbol{\mu}$ で張られる空間と履歴空間の距離（の 2 乗）という意味を持つ（[3], Section 2.5)。密行列であっても、最大特異ベクトルは数値的に高速に求められるから、問題は、$\boldsymbol{\mu}$ が与えられた時に、いかに効率よく $K(i, \boldsymbol{\mu})$ を計算するかである。

4.1 Krylov 部分空間の導入

$\rho \equiv \mathsf{H}_1 \mathsf{H}_1{}^\top$ とした時、H_1 の最大左特異ベクトル $\boldsymbol{u}^{(1)}$ は、次の変分方程式の解として求められる。

$$\boldsymbol{u}^{(1)} = \arg\max_{\boldsymbol{u}} R(\boldsymbol{u}) \ \text{ ただし } \ R(\boldsymbol{u}) = \frac{\boldsymbol{u}^\top \rho \boldsymbol{u}}{\boldsymbol{u}^\top \boldsymbol{u}}$$

$R(\boldsymbol{u})$ はレイリー商と呼ばれる量である。ここで次の問題を考える。

> w より小さい任意の正整数 s に対し、$s+1$ 次元の近似空間が、s 次元の近似空間よりもより「濃く」最大固有状態を含むような近似空間を構成せよ。

$\boldsymbol{\mu}$ で張られる 1 次元空間から出発する。上記の要請を満たすためには、span$\{\boldsymbol{\mu}\}$ に基底を付け加えて span$\{\boldsymbol{\mu}, \boldsymbol{\Delta}\}$ のような 2 次元空間を構成した時に、これが R の最急勾配方向を含む必要がある。R の勾配は

$$\left.\frac{d}{d\boldsymbol{u}} R(\boldsymbol{u})\right|_{\boldsymbol{u}=\boldsymbol{\mu}} = \frac{-2}{\boldsymbol{u}^\top \boldsymbol{u}} \left[R(\boldsymbol{\mu})\boldsymbol{\mu} - \rho\boldsymbol{\mu}\right]$$

のように計算できる。これより、$\boldsymbol{\Delta}$ を $\rho\boldsymbol{\mu}$ と選べば、span$\{\boldsymbol{\mu}, \rho\boldsymbol{\mu}\}$ は、上記最急勾配方向を含むことがわかる。

この論法を続けていけば、k 次元空間

$$\mathcal{K}_k(\boldsymbol{\mu}, \rho) \equiv \text{span}\{\boldsymbol{\mu}, \rho\boldsymbol{\mu}, ..., \rho^{k-1}\boldsymbol{\mu}\}$$

は、R の最大化の観点から最善の k 次元部分空間であることがわかる。言い換えると、履歴行列の列空間（w 次元）の部分空間としての k 次元空間のとり方は多くありえるが、その多くの選択肢のうちで、$\boldsymbol{\mu}$ をひとつの基底とする制約の下で、ρ の最大固有状態を最も「濃く」含むものが $\mathcal{K}_k(\boldsymbol{\mu}, \rho)$ である。最大固有状態ばかりではなく、第 2、第 3 固有状態についても同様のことが言える。この部分空間は、数学では、（$\boldsymbol{\mu}$ と ρ から誘導される）Krylov 部分空間と呼ばれる。

4.2 固有値問題の圧縮

$\mathcal{K}_k(\boldsymbol{\mu}, \rho)$ を張る正規直交基底を $\{\boldsymbol{q}_1, ..., \boldsymbol{q}_k\}$ と表し、これらを列ベクトルとして持つ $w \times k$ 行列を

$$\mathsf{Q} \equiv [\boldsymbol{q}_1, ..., \boldsymbol{q}_k]$$

と置く。一般には Q は一意には決まらないが、Krylov 行列 $\left[\boldsymbol{\mu}, \rho\boldsymbol{\mu}, ..., \rho^{k-1}\boldsymbol{\mu}\right]$ の QR 分解で求めることにすれば、Q は一意的に決まる（QR 分解の一意性）。また、$\boldsymbol{q}_1 = \boldsymbol{\mu}$ となる。

すると、ρ についての固有値方程式 $\rho\boldsymbol{u} = \lambda\boldsymbol{u}$ は、$\mathcal{K}_k(\boldsymbol{\mu}, \rho)$ への $\boldsymbol{u}$ の射影 $\boldsymbol{x} \equiv \mathsf{Q}^\top \boldsymbol{u}$ を用いることで

$$\mathsf{Q}^\top \rho \mathsf{Q} \boldsymbol{x} = \lambda \boldsymbol{x} \tag{4}$$

に（近似的に）変換される。つまり問題は、$k \times k$ 行列 $\mathsf{Q}^\top \rho \mathsf{Q}$ の対角化問題に帰着される。しかも、Krylov 行列と QR 分解の定義から、

$$\mathsf{T} \equiv \mathsf{Q}^\top \rho \mathsf{Q} = \text{(対称 3 重対角行列)}$$

となっていることを示せる [3]。$w \times w$ 行列 ρ に含まれていた情報が、T の中のわずか $2k-1$ 個の要素にいわば圧縮されたわけである。その対角要素 $\alpha_1, .., \alpha_k$ と、副対角要素 $\beta_1, .., \beta_{k-1}$ は、Lanczos 反復と呼ばれるアルゴリズム（Algorithm 2）で容易に求められることが知られている [3]。それからわかるように、係数 α_s と β_s を計算するためには、$\boldsymbol{q}_s$ と $\boldsymbol{q}_{s-1}$ のみがあればよい。後で述べるように、Q を陽に求める必要はないので、各ステップで不要なものは破棄していけばよい。

Algorithm 2 *(Lanczos iteration)* $\boldsymbol{r}_0 = \boldsymbol{\mu}$, $\beta_0 = 1$, $\boldsymbol{q}_0 = 0$, $s = 0$ と初期化する。以下を、$s < k$ である間繰り返す。

$$\begin{aligned}
\boldsymbol{q}_{s+1} &= \boldsymbol{r}_s / \beta_s \\
s &\leftarrow s+1 \\
\alpha_s &= \boldsymbol{q}_s{}^\top \rho \boldsymbol{q}_s \\
\boldsymbol{r}_s &= \text{prod}(\rho, \boldsymbol{q}_s) - \alpha_s \boldsymbol{q}_s - \beta_{s-1}\boldsymbol{q}_{s-1} \\
\beta_s &= \sqrt{\boldsymbol{r}_s{}^\top \boldsymbol{r}_s}
\end{aligned}$$

ここで $\mathrm{prod}(\rho, \boldsymbol{q}_s)$ は、ρ と $\boldsymbol{q}_s$ の積を与える関数である。ρ は 1 本の時系列から滑走窓式に取られた時系列から作られるから、あらわに行列を保持しなくても積を計算することは簡単にできる。

4.3 カーネルの陰な計算

3 重対角行列の固有値問題は数値的にきわめて効率よく解くことができる [3]。式 (4) で求められた固有ベクトルを $\{\boldsymbol{x}^{(1)}, ..., \boldsymbol{x}^{(k)}\}$ とすると、H_1 の固有ベクトルは、

$$\boldsymbol{u}^{(\alpha)} = \sum_{i=1}^{k} \boldsymbol{q}_i x_i^{(\alpha)} \tag{5}$$

のようにして求められる。ただし $x_i^{(\alpha)}$ は k 次元ベクトル $\boldsymbol{x}^{(\alpha)}$ の第 i 成分を表す。$K(\alpha, \boldsymbol{\mu})$ はこれに $\boldsymbol{\mu}^\top$ を掛けたものであるが、$\boldsymbol{q}_1 = \boldsymbol{\mu}$ かつ $\boldsymbol{q}_i^\top \boldsymbol{q}_j = \delta_{i,j}$ であるから (δ はクロネッカーのデルタ)、結局カーネルは、

$$K(\alpha, \boldsymbol{\mu}) = x_1^{(\alpha)} \tag{6}$$

となる。すなわち、変化度を計算するにあたり、あらわに H_1 の特異ベクトル $\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}, ..., \boldsymbol{u}^{(r)}$ を求める必要はなく、$k \times k$ の固有値問題 (4) を解いて固有ベクトルを求め、その固有ベクトルの第 1 成分を取れば、それで変化度を

$$z = 1 - \sum_{i=1}^{r} \left[x_1^{(i)}\right]^2 \tag{7}$$

のように計算できるということである。これを、Implicit kernel 法と呼んでおく。

5 高速な変化点検出手法

5.1 アルゴリズム

Implicit kernel 法でカーネルを計算するためには、事前に $\boldsymbol{\mu}$ を求めておく必要がある。$\boldsymbol{\mu}$ は H_2 の最大左特異ベクトルであるから、密行列であっても数値的に効率よく計算できる。ここでは、べき乗法などの反復法に Zha-Simon 流のオンライン更新の考えを入れた一般的な枠組みとして、以下のものを用いる。

Algorithm 3 *(feedback iterative method)* ϵ をある小さい定数とする。$\boldsymbol{a}$ を正規化されたランダムベクトル $\tilde{\boldsymbol{a}}$ に初期化する。各 t において以下を行う。

1. $\mathsf{H}_2^\top \mathsf{H}_2$ に対する反復法を $\boldsymbol{a}$ を初期ベクトルにして行い、最大固有ベクトルとして $\boldsymbol{\mu}$ を求める
2. $\boldsymbol{a} = \boldsymbol{\mu} + \epsilon\tilde{\boldsymbol{a}}$ とおき、正規化する。

反復法としては、べき乗法、EM-PCA [8]、あるいは Lanczos 3 重対角化と QR 反復の組みあわせ（以下では LQR と略す)、など、初期ベクトルを解に近づけていくタイプの算法を利用することができる。Zha-Simon 流の逐次更新アルゴリズムと違い、これらの方法では、初期ベクトルが解空間と直交していない限り、厳密な解に到達することが保証されていることに注意されたい。

以上、われわれの新しい SST アルゴリズムは、Implicit kernel 近似と、$\boldsymbol{\mu}$ のフィードバック計算からなる。これを便宜上、FELIX (FEedback impLIcit kernel approXimation) 法と呼んでおく。FELIX-SST アルゴリズムを下記にまとめる。

Algorithm 4 *(FELIX-SST)* 各時刻 t において、下記の計算を行う。

1. Algorithm *3* で $\boldsymbol{\mu}$ を更新する。
2. $\boldsymbol{r}_0$ を $\boldsymbol{\mu}$ に初期化して Algorithm *2* を実行し、$\alpha_1, .., \alpha_k$ と $\beta_1, .., \beta_{k-1}$ を求める。
3. $\{\alpha_i\}$ を対角要素、$\{\beta_i\}$ を副対角要素とする *3* 重対角行列の固有ベクトルの上位 r 本を求める（例えば *QL* 反復を利用できる)。
4. 式 *(7)* にしたがって変化度を求める。

5.2 パラメターの選択

ここで、FELIX-SST におけるパラメターの選択についてもまとめておく。まず、窓幅 w に関しては、「1 秒間以下で生じる細かい振動は無視する」、など、捉えたい変化の時間スケールから決めることができる。r に関しては、部分系列のパワースペクトルの解析から（ほとんど w に依存せずに)、相当広い範囲の時系列データで、数個程度でかまわないことがわかっているので [4]、特に明確な周期性のない時系列データについては w によらず 3 から 4 程度、振動的な成分を含む時系列データについてはそれよりも若干増やして 5 前後でよい。FELIX 法ではこれらに加えて、$\mathcal{K}_k(\boldsymbol{\mu}, \rho)$ の次元 k を与える必要があるが、Krylov 部分空間が、実は最大固有状態に加えて最小固有状態についてもよい近似空間になっていること [3] から考えて、

$$k = \begin{cases} 2r & r \in \text{even} \\ 2r - 1 & r \in \text{odd} \end{cases} \tag{8}$$

のように選ぶのが合理的である。以上、結局のところ、事前に入力すべきパラメターは、事実上 w のみである。この単純さが、SST のひとつの利点である。

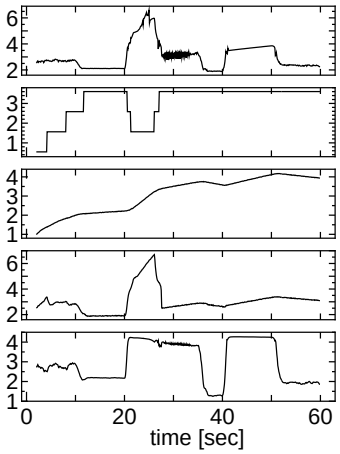

図 2: ベンチマーク用データ。

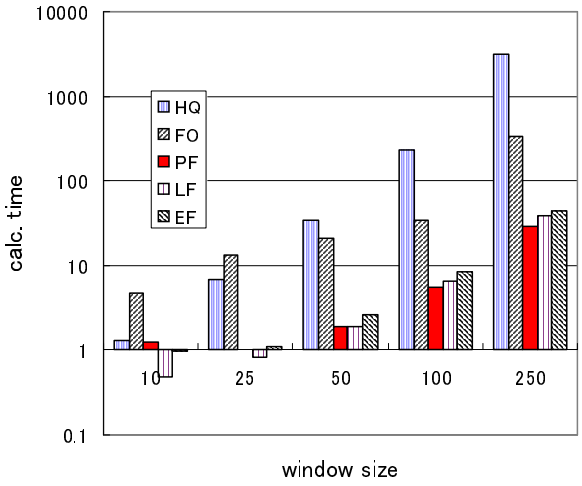

図 3: 変化度計算時間の比較。HQ が従来技術。FO が改良版の従来技術。他が FELIX-SST。

6 実験

図 2 はある機械システムから採取した時系列データである。それぞれ 100ms 刻みで 580 個のデータ点を含む。実際の計測値は、いずれも分散が 1 で平均値が 3 になるように前処理されている。このデータに対して、いくつかの SST アルゴリズムで変化度を計算し、さまざまな w に対して 5 本の時系列に対する合計の計算時間を比較した。時系列データには周期的境界条件を課し（つまり 581 番目のデータ点が第 1 番目のデータ点になる）、w が変化しても z を計算する回数が変化しないようにした。パラメターは $(r, k) = (3, 5)$ とした。

結果を図 3 に示す（Java 1.4、Pentium M 1.8GHz）。比較した手法は 5 つあり、それぞれを省略記号で表している。HQ と FO が、Implicit kernel 近似を用いず、式 (3) により変化度を計算したもので、PF、LF、EF が FELIX-SST に基づく。

HQ と FO については、H_1 および H_2 の SVD を、(1) HQ は Householder 法＋ QR 反復で、(2) FO は直交反復法で、で行った。FO には Algorithm 3 と同様のフィードバックを用いている。

また、PF、LF、EF については、相違は $\boldsymbol{\mu}$ の計算方法だけにあり、それぞれ、Algorithm 3 の反復法を (3) PF はべき乗法で、(4) LF は LQR で、(5) EF は EM-PCA で、行った。これらのいずれも、T の対角化には、QL 反復を用いた。

まず、HQ と FO を比べると、w が大きい時にはフィードバックによる加速は効果的だが、w が 50 くらいまではほとんど効果がないことがわかる。素朴に SVD に逐次更新のアイディアを取り入れても限界があるということである。

次に、これらと、FELIX-SST の結果を比べると、Implicit kernel 近似による計算時間の減少は劇的であることがわかる。たとえば、$w = 100$ においては PF は、HQ に比べて約 1/52、FO に比べても 1/7.8 に計算時間が短縮されている。$w = 250$ に至ってはそれぞれ、1/130、1/14 と更に差が開く。その結果、従来手法では w が 50 程度で実時間計算が苦しくなったにもかかわらず、PF では、$w = 250$ でもまだ実時間計算が可能である。

図 3 ではまた、FELIX-SST 同士の比較がなされている。図によれば、PF、LF、EF の計算時間の差はわずかであるが、w が数 10 を越えると PF が最も優れた成績を挙げていることがわかる。

Implicit kernel 近似に伴い、計算される変化度にはある程度の誤差が含まれるはずである。それを吟味するために、周波数変動問題における変化度の様子を調べた[3]。結果を図 4 に示す。これは、$w = 25$、$(r, k) = (3, 5)$ に対して、HQ と PF を比較した結果である。図に示すとおり、近似に伴う誤差はほとんど無視できることがわかる。他の w についても同様のことが言える。

7 まとめ

SST の欠点であった計算速度の問題を大幅に改善する近似アルゴリズムを提案した。まず変化度の計算をカーネルの計算に帰着させ、Lanczos 法により有効な情報を濃縮した上で、陰にそのカーネルを計算する。しかも本手法は、既存のオンライン SVD アルゴリズムを直接適用した時のような原理的困難を持たない（3 節参照）。

[3]用いたデータは、$a\sin(2\pi t/\lambda)$ の振幅 a に $\pm 7.5\%$、波長 λ に $\pm 0.5\%$ の割合でランダムノイズを付加して生成したもの。ただし、$t = 150, 300$ において、波長の平均値 λ が $\sqrt{80}$ から $\sqrt{120}$、$\sqrt{120}$ から $\sqrt{70}$ に遷移する。

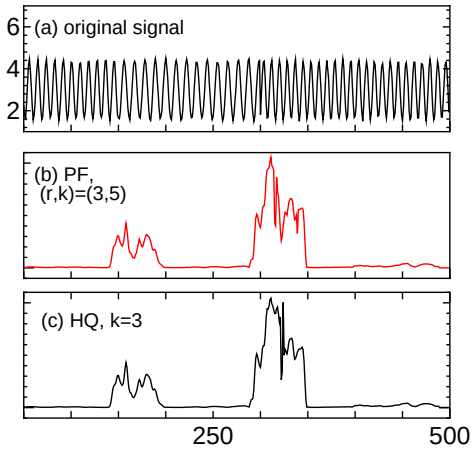

図 4: Implicit kernel 近似による誤差の吟味。(a) 原時系列信号。(b) FELIX-SST 法による変化度。(c) Householder + QR 反復による従来手法。

最後に、提案手法は、高速化に加えて数値的安定化にも有効であることを述べておく。一般に、次元が数百程度になると、密行列の複数の特異ベクトルを反復法で求めるのは困難になる。求めたい固有ベクトルが増えるにつれ急速に計算速度が低下し、また、偽固有値や固有ベクトルの非直交性などの数値的不安定性に悩まされることになる（たとえば文献 [3] の 9.2 節参照)。実際、もし、Implicit kernel 法を使わず、例えばランダムベクトルから出発して式 (5) により陽に特徴ベクトルを求めたとすると、しばしば特異ベクトルに偽縮退が現れる。このようなことがおこると、履歴空間の基底がでたらめになってしまい、結果として z の値が非常に不安定になる。

Implicit kernel 近似は、密行列で起こりがちなこのような困難を巧妙に回避する方法を与えている。密行列の情報を圧縮することにより、数値的に扱いやすい低次元空間ですべての計算を済ませる。われわれが求めたいのは z であって、$\mathcal{H}_r$ が直接必要なわけではないから、危険の多い $\{\boldsymbol{u}^{(i)}\}$ の計算を回避できればそれに越したことはないのである。

参考文献

[1] M. Basseville and I. V. Nikiforov. *Detection of Abrupt Changes: Theory and Application*. Prentice Hall, 1993.

[2] M. Brand. Fast online SVD revisions for lightweight recommender systems. In *Proc. SIAM Intl. Conf. Data Mining*, 2003.

[3] G. H. Golub and C. F. V. Loan. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, Baltimore, MD, 1996.

[4] T. Idé. Why does subsequence time-series clustering produce sine waves? In *Proc. European Conf. Principles and Practice of Knowledge Discovery in Databases (PKDD), Lecture Notes in Artificial Intelligence*, volume 4213, pages 311–322. Springer, 2006.

[5] T. Idé and K. Inoue. Knowledge discovery from heterogeneous dynamic systems using change-point correlations. In *Proc. SIAM Intl. Conf. Data Mining*, pages 571–575, 2005.

[6] J. Lim, D. A. Ross, R.-S. Lin, and M.-H. Yang. Incremental learning for visual tracking. In *Advances in Neural Information Processing Systems*, volume 17, pages 793–800, 2005.

[7] S. G. Mallat and W. L. Hwang. Singularity detection and processing with wavelets. *IEEE Trans. Information Theory*, 38(2):617–643, 1992.

[8] S. Roweis. EM algorithms for PCA and SPCA. In *Advances in Neural Information Processing Systems*, volume 10, 1998.

[9] J. Takeuchi and K. Yamanishi. A unifying framework for detecting outliers and change points from time series. *IEEE Trans. Knowledge and Data Engineering*, 18(4):482–492, 2006.

[10] K. Yamanishi, J. Takeuchi, G. Williams, and P. Milne. On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. *Data Mining and Knowledge Discovery*, 8:275–300, 2004.

[11] H. Zha and H. D. Simon. On updating problems in latent semantic indexing. *SIAM Journal on Scientific Computing*, 21(2):782–791, 1999.

[12] H. Zha and Z. Zhang. On matrices with low-rank-plus-shift structures: Partial SVD and latent semantic indexing. *SIAM Journal of Matrix Analysis and Applications*, 21:522–536, 1999.