

再圧縮処理に基づく多次元データの 高効率変化検知・復元方式の検討

古澤 宏樹[†] 笠井 裕之[†]

大量の多次元データが時々刻々と変化する状況において、これまで容易に検知することが困難であった重要な変化や情報を取得することへの期待は高まっている。また、それらの大量データを効率に蓄積するとともに、必要に応じて抽出できることが要求される。これまで、多次元データに対して圧縮センシング技術を用いて主要データを抽出することで、効率的な変化検知や効率的なデータ蓄積、また復元する方式が提案されてきた。本稿では、圧縮されたデータをさらに再圧縮方式を導入することで、変化検知時に影響を与えない形で、効率的なデータ復元処理が可能となる方式を提案する。シミュレーション用に生成した実験用データと実データを利用して評価実験を行い、提案方式の有効性を示す。

1. はじめに

Twitter や Facebook 等のソーシャルメディアの普及により、社会への大量の多次元データ流通とその活用が始まっている。このような大量の多次元データが時々刻々と変化する状況において、その変化を効率的に検知することで、これまで得ることが困難であった重要な変化や情報を取得することが可能となる。しかしながら、対象データの量的側面、多次元性に起因する複雑性に起因して、効率的にそのような変化を検知することは技術的課題が多く存在する。このような課題に対して、対象多次元データのテンソル[1]表現に着目し、テンソル分解を行うことにより得られる特徴スペクトルからコアテンソルを取り出すことで、効率的に変化検知を行う方式が提案されてきた[2]。本方式は、テンソル分解を行うことで、大量データの中の個々のデータの変化にとらわれず、特徴スペクトルの変化のみを検知することが可能であるため、変化検知の精度は向上する。しかしながら、テンソル分解によるコアテンソルの導出計算コストが高いという欠点がある。そこで文献[3]では、テンソル分解を随時更新していく Dynamic Tensor Analysis (DTA)及び近似手法を用いた高速な Streaming Tensor Analysis (STA)を提案している。しかしながら、これらのインクリメンタルテンソル分解手法は、尚、計算コストが高いことが課題である。そこで、本課題を回避することを目的として、コアテンソルに圧縮センシング [5][6] (Compressive Sensing; CS) を利用し、センシングベクトルをコアテンソルの変化と同等に見なすことにより計算コストを抑制する方式が提案された[4]。

しかしながら、従来方式では、復元処理量を削減するためには圧縮率を高める必要があるが、圧縮率を高めると変化検知及び復元品質が低下する可能性がある。従って品質を担保するには、圧縮率を低く (圧縮後の次元を大きく) する必要があり、その場合、復元処理量が増大する。そこで最適な圧縮率を事前に決定したいが、未来のデータ特性は時々刻々と変化するため予測できず、システム開始時に、適切な圧縮率 (センシング行列サイズ) を決定することは不可能である。

そこで本稿では、復元品質に影響を与えない十分な大きさのサイズのセンシングベクトルを変化検知処理で利用し、一方で、再圧縮により復元対象のベクトルを最小サイズまで縮小し、復元処理量を削減することが可能な方式を提案する。提案方式のさらなる利点としては、従来方式では、圧縮処理に使用するセンシング行列のサイズを設定する際、復元品質と復元処理量増大を見越して適切なサイズを求め選択する必要があるが、提案方式では、復元処理量増大は再圧縮処理により回避可能なため、ある程度の大きさのセンシング行列サイズを設定することが可能となる。

以降、2 節で圧縮センシングに基づく多次元データ変化検知・復元に関わる従来方式を説明する。次に3 節では、本稿で対象とする問題設定を行い、そのための仮説を予備実験により検証する。4 節では再圧縮処理に基づく変化検知・復元方式を提案する。最後に、5 節で比較実験を行い、6 節でまとめと今後の展望について述べる。

2. 圧縮センシングに基づく多次元データ変化検知・復元に関わる従来方式

本節では、多次元データをテンソルストリームとして表現し、それに対してテンソル分解を用いて変化検知・復元を行なう方式を説明する。次に、圧縮センシングを用いることで、毎回のテンソル分解を回避し計算コストを削減する方式について説明する。

2.1 テンソル分解に基づく変化検知・復元方式

時々刻々と変化する大量の多次元データの特徴を効率的に表現する方法の1つとして、テンソルストリーム表現がある。そのように表現された多次元データのテンソルを解析するためのテンソル分解には様々な方法が提案されてきた[1]。テンソル分解の方法の一つに行列データを解析する方法の一つである特異値分解 (Singular Value Decomposition; SVD) をテンソルに拡張した方法が提案されており、本稿ではその中の高次特異値分解 (Higher-Order SVD; HOSVD) [9] を利用しテンソル解析をする。 $\mathcal{X} \in R^{I_1 \times \dots \times I_M}$ を元信号、 \mathcal{Z} をコアテンソルとし、さらに \mathcal{X} をモードで展開し SVD を行った際に得られる左特異ベクトルを成分に持つ直交行列を $U_m |_{m=1}^M \in R^{I_m \times R_m}$ と定義すると、 \mathcal{X} は式(1)で展開される。

$$\mathcal{X} \approx \mathcal{Z} \prod_{m=1}^M \times_m U_m^T \quad (1)$$

式(1)においては、記号「 \times_n 」は n -モード積 (n-mode product) と呼ばれる。この分解は全てのモードでモード展開を行い、行列化し、その一つ一つに対し SVD を行うことで左特異値を得る。コアテンソルのサイズは元信号のランクであるものを得ることが出来る。本方法で得られたコアテンソル \mathcal{Z} は、ノイズをなくしてテンソルの重要な要素だけを取り出した表現である。このコアテンソル \mathcal{Z} が入力多次元データの重要な特徴を示すデータであることに着目し、入力多次元データの重要な変化を検知する方式が提案されている[4]。この手法では高い確率で変化検知を行うことができ、更に元のデータに復元することもテンソル分解を利用することで可能である。しかし、毎回テンソル分解を行

[†]電気通信大学 情報システム学研究科

うため、この変化検知方法では計算コストが増加するという欠点がある。

2.2 圧縮センシングに基づく変化検知・復元方式 (SCENT)

前節で説明したテンソル分解を用いた変化検知は、テンソル分解において計算処理量が増大する。よって、テンソル分解で得られるコアテンソルの代わりに、元データとセンシング行列より得られるセンシングベクトルを利用する方式 (SCENT) が提案された。これにより、変化検知時の計算処理量を削減する。まず、SCENT 方式で利用する圧縮センシングについて簡単に述べ、テンソル分解との等価的扱いができることを示す。

2.2.1 圧縮センシングとテンソル分解

圧縮センシングは、元信号がスパースであるという条件の下で信号を圧縮し、従来必要な観測サンプル数より小さな数で元信号の復元を行う方法である[5]。具体的には、 v を n 次元の元信号とし、この元信号に、ある行列との積を利用することでサンプリングする。具体的には、 $k \times n$ のセンシングマトリクス Φ を用いて、 k 次元のセンシングベクトル y は $y = \Phi v$ と表現される。よって、これは n 次元の元信号から k 次元のセンシングベクトル y をサンプルしたことになる。ここで元信号 v にはスパースであるという前提条件があると仮定する。仮に S スパース、 S 個比零の数があるとすると、この条件と Φ が制限等長性[10]の条件を満たすことで、式(2)のように記述することができる。

$$\min \|\tilde{v}\|_1 \quad \text{subject to} \quad y = \Phi \tilde{v} \quad (2)$$

\tilde{v} は再構成する元の信号を示す。ここで、 $\|\tilde{v}\|_1 = \sum_i |v_i|$ であり \tilde{v} の L1 ノルムを表す。 Φ が制限等長性 (restricted isometry property; RIP) の条件を満たしているため L1 ノルムの最小化 (以後、L1 最小化と呼ぶ) を行うことで、高い確率で元データの v と \tilde{v} の値が一致し復元が可能となる。

次に、圧縮センシングをテンソルに拡張する。テンソル表現された元の信号 X をベクトル化することにより、センシングベクトル y を得る。ここでテンソル分解の表現を使うことにより式は $y = \Phi \Psi z$ と書き換えることができる。 Ψ はテンソル分解をした時に得られた左特異値の積である。 Φ と Ψ の積が一定で制限等長性を満たしていれば前述の式と同じよう議論することができる。

2.2.2 変化検知方式と元信号復元方式

これまでに説明した圧縮センシングとテンソル分解を利用した多次元データの変化検知方法について説明する。変化検知をテンソルストリームから行うということは、時刻 t と時刻 $t-1$ を比較し、コアテンソルの変化をみることに相当する。ここで文献[3]よりコアテンソルの変化をみることは、センシングベクトルの変化をみることと等価であることから、 $\|\Delta z\| \propto \|\Delta y\|$ がいえる。よって、以下の式により変化検知が可能となる。

$$\delta = \|y_t - y_{t-1}\| / \|y_t\| > \tau \quad (3)$$

このようにセンシングベクトルの y の変化をみることで変化検知を行うことができる。次に変化が検知された際の元信号の復元方式について述べる。ここで、センシングベクトル y 、センシング行列と左特異値の積からなる $P = \Phi \Psi$ 、元データの非零の数 S がわかっており、ここからコアテンソルを得ることが目的であり、その後、得られたコアテンソルを利用し元のデータを復元する。コアテンソルを得る方法として先ほど紹介した圧縮センシングを利用する。 $P = \Phi \Psi$ とすると $K \times S$ 行列となり、式(4)のように表現できる。

$$\min \|\tilde{z}\|_1 \quad \text{subject to} \quad y = P\tilde{z} \quad (4)$$

ここで、 P も制限等長性の条件を満たしていることを利用すると、この式(4)、L1 ノルム最小化を解くことで \tilde{z} を得ることができる。これは圧縮センシングの原理より、ここで得ることができた \tilde{z} は求めたコアテンソルの係数 z と一致する。まとめるとこの方式の流れとしては元データとセンシング行列との積からセンシングベクトルを得ることができ、そこから変化検知を行い変化が検知された場合は圧縮センシングを利用しコアテンソルを得て、最終的に元のデータに復元するという流れである。この方式のフロー図を図1に示す。

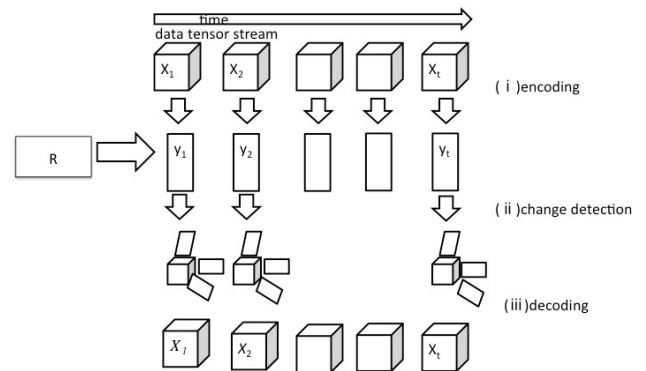


図1: 圧縮処理, 変化検知, 復元処理の手順

3. 課題定義と予備検証実験

3.1 課題定義

SCENT 方式では、変化検知のため入力データにセンシング行列を乗じることにより圧縮データであるセンシングベクトルを導出し、前後の時刻におけるセンシングベクトルを比較することで変化検知を行なう。次に、データ復元においては、圧縮されたセンシングベクトルに対して L1 最小化アルゴリズムを用いて元の入力データの復元を行なう。そこで、従来方式における制約条件について整理する。

- 圧縮後のセンシングベクトルのサイズの同一要件
 変化検知においては、前後の時刻の 2 つのセンシングベクトルを比較することから、圧縮後のデータサイズ (次元) を同一にする必要がある。
- 復元品質と復元処理量のトレードオフ
 データ特性 (非ゼロ数) に応じて、あるしきい値以下のサイズの場合、正確な復元が不可能である。一方、センシングベクトルのサイズが小さいほど、復元にかかる L1 最小化処理量が小さくなる。このように両指標間でトレードオフの関係がある。
- 将来データの未確定性
 第 1 の条件から、圧縮後のサイズは常に一定であることが求められるが、システム稼働開始時に、将来発生するデータの特性は不明である。

以上から、提案方式における問題点を整理する。

- 従来方式では、復元処理量を削減するためには圧縮率を高める必要があるが、圧縮率を高めると変化検知及び復元品質が低下する可能性がある。
- 従って品質を担保するには、圧縮率を低く (圧縮後の次元を大きく) する必要があり、その場合、復元処理量が増大する。
- 最適な圧縮率を事前に決定したいが、未来のデータ特性は時々刻々変化するため予測できず、システム開始時に、適切な圧縮率 (センシング行列サイズ) を決定

することは不可能である。

- 従って、復元品質に影響を及ぼさない余裕のあるサイズを予め設定しておきたいが、その場合、復元率の観点で冗長となり、復元処理量が增大する。

しかし、上記の問題点の指摘においては2つの点を仮定している。第1点目は、圧縮センシングにより圧縮し出力されるセンシングベクトルを復元する際、L1最適化による処理量が、センシングベクトルのサイズ(サンプル数)の増加により増加する点である。一方、2点目は、サンプル数の増加に伴い復元精度が増加するという点である。これより、以下では、この2点について予備実験を通して検証する。

3.2 L1最小化に関する予備検証実験

L1ノルムを最小化する解を求めるL1最小化問題は、不定線形システム $b=Ax$ の解を求める問題として、近年大きな注目を浴びている。内点法(Interior-point method)に代表される汎用的な従来方式は、大規模な現実問題に対して十分な能力を有しないことから、近年、数多くの高速アルゴリズムが提案されている。貪欲法として Orthogonal matching pursuit (MP) や Least angle regression (LARS) が提案されているが、解が極めてスパースな場合に限り良い性能を示すが、非ゼロ数が増加すると大域的収束性に対する強固な理論的保証が得られないことが知られている[9]。Iterative Shrinkage-Thresholding (IST)等、近似法に基づく様々な高速化手法が提案されている。しかし、いずれにおいてもL1最小化処理は処理量が極めて高いことが知られている。前述の仮定をこのようなL1最小化を用いて復元した際に成り立つのかを予備実験を行うことで示した。

3.2.1 サンプル数と処理量の関係(予備実験 I)

まず、サンプル数と処理量の関係についての実験を行った。実験内容としては、復元対象データのサンプル数を変化させ、復元を行う場合、L1最小化問題を解くために計算時間にどのような影響があるのかを検証した。この実験より、前節で仮定したサンプル数の削減が計算時間の削減につながるのかを確認することができる。本実験では、双対内点法(Primal-dual interior-point method)の実装ソルバーである *ll magic*[11]を用いた。実験は信号の数を信号数が1000、非零の数が10、50のもの、信号数が2500、非零の数25、125の4種類を用意し、これに、ランダムに行列をかけ、圧縮した後に復元を行った時にかかる時間を観察した。信号は50個ずつ用意し、計算時間は50個の平均の値である。結果は図2、3に示す。

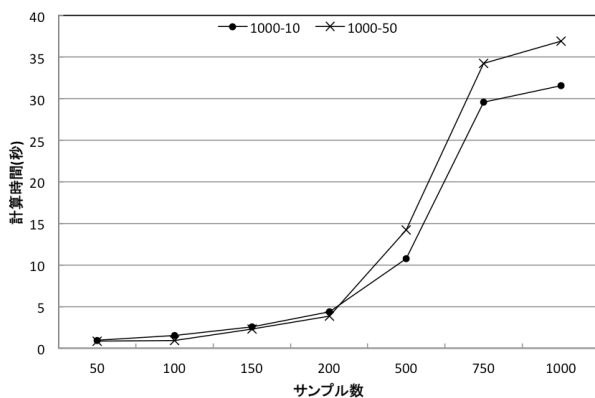


図2：サンプル数と処理量の関係(信号数1000)

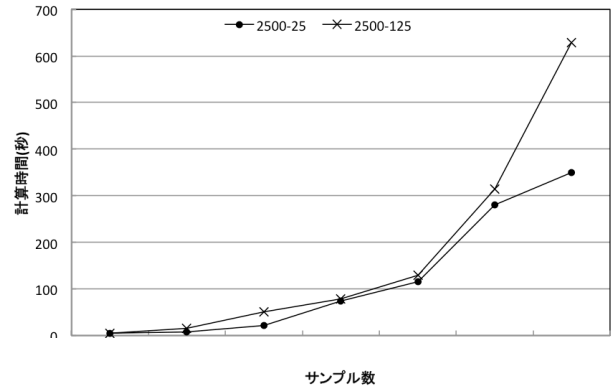


図3：サンプル数と処理量の関係(信号数2500)

図2、3より実験で試した4種類全ての信号でサンプル数が増えると復元時、L1最小化を行う際に計算時間が長くなることを見て取ることができる。これより前節で仮定した1点目が正しいことを示すことが出来た。

3.2.2 サンプル数と復元精度(予備実験 II)

次に、サンプル数と復元精度の関係についての実験を行った。この実験より、前節で仮定したサンプル数の削減が計算時間の削減につながるのかを確認することができる。実験は信号を2種類、信号数1000で非零の数10と信号数2500で非零の数50を用意した。これらの信号に、ランダムに行列をかけた後に復元を行った。復元時の評価には式(5)を利用し復元率という形で実験結果を出した。具体的には、

$$e_T = 1/T \times \sum_{t=1}^T \|x_t - \tilde{x}_t\| \quad (5)$$

T は入ってきたテンソルの数、 x_t は元の信号、 \tilde{x}_t は再構成した信号を指す。結果は図4、5に示す。

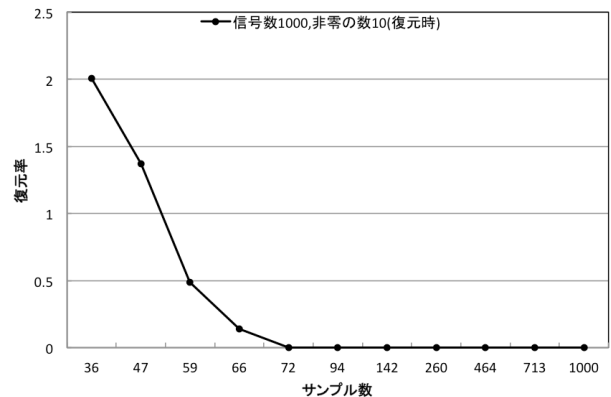


図4：サンプル数と復元精度(信号数1000)

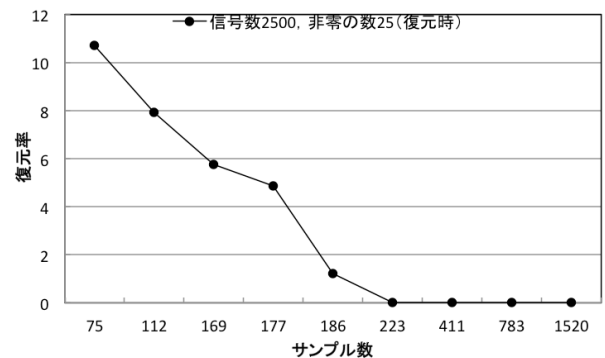


図5：サンプル数と復元精度(信号数2500)

図4および図5より実験で試した、2種類の信号でサンプル数の増加に伴い復元精度も増加していることが見て取ることができる。また、復元精度が突然悪化する部分が存在することも見て取ることができる。これより前節で仮定した2点目も正しいことが示すことができた。ここまで予備実験で示すことができた2点のことと課題定義した条件を利用して次章、提案方式を説明する。

4. 再圧縮処理に基づく変化検知・復元方式

従来方式では変化検知時と復元時において同一サイズの圧縮データを用いている。しかしながら、入力データ特性によっては、復元処理に際してサンプル数が冗長である可能性がある。つまり、変化検知時の圧縮データであるセンシングベクトルのサイズが、復元精度の観点で復元処理に必要なサイズよりも大きい場合がある。この場合、復元対象となるセンシングベクトルをさらに小さくすることにより、復元処理にかかる計算処理を削減することが可能であると考えられる。従って、従来方式の復元品質を保持しながら、方式全体の計算処理量を削減することが可能となる。そこで、変化検知時で使うセンシングベクトルと比べ、復元が可能より小さなサイズ、つまり最低限のサンプル数まで再圧縮したセンシングベクトルを作成し、それに対し復元処理を実行することにより、復元処理量を削減することができると思われる。

4.1 提案方式

本稿では、復元品質に影響を与えない十分な大きさのサイズのセンシング行列を変化検知処理で利用してセンシングベクトルを算出し、一方で、変化検知後にセンシングベクトルを再圧縮し最小サイズまで縮小することで、元データの復元処理量を削減することが可能な方式を提案する。提案方式の一つの利点は、従来方式では、圧縮処理に使用するセンシング行列のサイズを設定する際、復元品質と復元処理量増大を見越して適切なサイズを選択する必要があるが、提案方式では、復元処理量増大は再圧縮処理により回避可能なため、ある程度の大きさのサイズを、余裕を持って設定することが可能となる点である。これは、将来発生するデータ特性が不明という条件下において大きな利点となる。ここで、センシングベクトル y を再圧縮することで得られるベクトルを最小センシングベクトル y_{min} と呼ぶことにする。また、変化検知時に必要なセンシングベクトル y を生成する時に使用したセンシング行列を R_f とし、再圧縮時に利用するセンシング行列を R_s と定義する。本提案手法のアルゴリズムを表1に示し、処理フローを図6に示す。

表1：提案手法アルゴリズム

Change-Detection: Given data tensor $\mathcal{X} \in R^{I_1 \times \dots \times I_M}$
1. Sense y_t from $y_{k,t} = \langle \mathcal{X}_t, R_f \rangle$
2. Compute $\delta = \ y_t - y_{t-1}\ / \ y_{t-1}\ $. If $\delta > \tau$, output change
3. Construct R_s from R_{direct}, R_f
3. Compute minimum sensing vector y_{min} from y, R_s and keep it
Decoding:
1. Use L1-minimization for y_{min}, R_s, R_f
2. Construct the matrix P from R_s, R_f
3. Compute core tensor coefficient $z = \text{vec}(Z)$ from $z = Py_{min}$

尚、提案方式においては、再圧縮時に再圧縮用センシング行列 R_s の作成処理と、復元時に最初の行列 R_f と R_s の行列積の演算処理が追加で必要となることから、これらの追加処理の全体処理量への影響が懸念される。しかしながら、先に述べたように、復元処理における L1 最小化処理にかかる時間は非常に大きい。これより、再圧縮用行列作成、行列演算による再圧縮処理は、L1 最小化による復元処理を含む全体としての計算時間に大きな影響を与えることはない。具体的には行列作成には復元処理のおよそ 1/1000 程度の処理量であることを予備検証実験から得ている。

4.2 再圧縮処理と再圧縮用センシング行列 R_s 生成

変化検知処理後、圧縮データであるセンシングベクトルを再圧縮し、復元時に利用する必要最低センシングベクトルまで再圧縮する。ここで、最小センシングベクトル y_{min} は以下のように得られる。

$$y_{min} = R_s R_f \chi = R_s y \quad (6)$$

ここで、再圧縮用センシング行列 R_{sec} の作成方法について説明する。まず、圧縮センシングでは RIP 特性に基づくことで、復元性能を保証するセンシング行列を生成することが可能となる[6]。本稿では、[7]により提案されているガウスランダム行列によるスパースアンサンブル行列を採用する。これは、RIP 性を満たし、行列が $K \times N$ のとき $K \geq C \cdot s \cdot \log(N/s)$ の下で s スパースなベクトルを高い確率で復元することが可能である。ここで C は正の定数である。このアンサンブル行列の利点は、1/3 のみ非ゼロ係数であることである。

$$R_s = \sqrt{\frac{3}{n}} \begin{cases} +1 & \text{with probability } 1/6 \\ 0 & \text{with probability } 2/3 \\ -1 & \text{with probability } 1/6 \end{cases} \quad (7)$$

式(10)に示すセンシング行列を使用して、再圧縮用 R_s を生成する。次に、最終的に再圧縮により得られるセンシングベクトルを直接、入力信号 X から得るための行列 R_{direct} を同様に式(8)から生成する。以上から、 R_s は以下の一般化逆行列を用いて導くことができる。ここで、 A^+ は行列 A の一般化逆行列を示す。

$$R_s = R_{direct} R_f^+ \quad (8)$$

4.3 再圧縮センシング行列サイズの導出

ここで、再圧縮後のセンシング行列のサイズについて検討する。圧縮後のセンシングベクトルを復元するための条件については文献[8]で説明されており、具体的には以下の

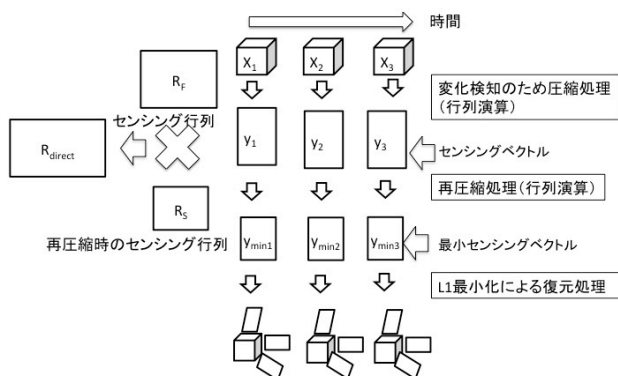


図6：提案方式フロー図

不等式で示されている.

$$\frac{1}{2^{0.25}-1} \left(\sqrt{2Sa} + \beta \sqrt{S \log \left(\frac{m}{s} \right)} \right) < \sqrt{p} \quad (9)$$

変数の p は必要なサンプル数, m は元データの信号の次元, S は非零の数, β は正の数である. よって, β の値を制御することでサンプル数 p を定めることができる. このサンプル数 p が最小センシングベクトルの次元数である. ここで得られたセンシングベクトルを用いて復元を行う. 復元時には2つの行列の積を用いて圧縮センシングを行うことでコアテンソルを得て, そこから元データに復元する.

しかしながら, 式(9)に含まれる β を一意に定める指針は明確でないため, 本稿では, 数値シミュレーションから β の特性を分析し, 後続の実験で使用することとした. そこで, 提案する再圧縮方式で言及した β について検証実験を行なう. 復元時に必要なサンプル数を β の値を変化させていくことにより変化させ復元率との関係を実験で確かめた. 実験は信号の数を信号数が 1000, 非零の数が 10, 50 のものと, 信号数が 2500, 非零の数 25, 125 の 4 種類を用いし β の値, サンプル数を変化させながら圧縮, 復元を行った. この実験の結果は図 9, 図 10 に示した.

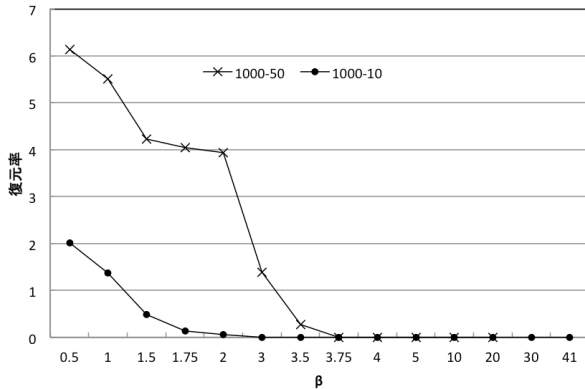


図 7: β の性能評価 (信号数 1000)

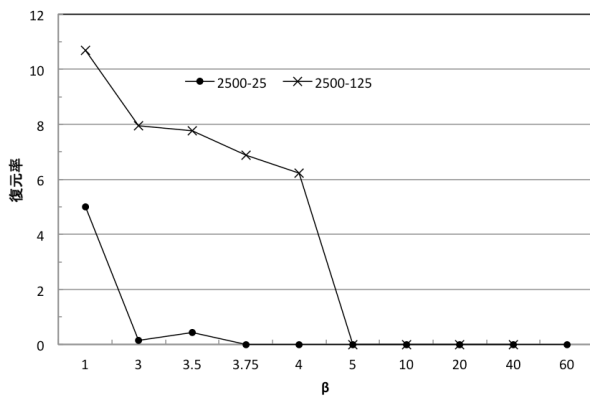


図 8: β の性能評価 (信号数 2500)

この結果から, 信号の特性により違いはあるが, 実験 4.1.3 と同様に急激に復元率が悪くなる境界があるところが見て取れた. ここに圧縮センシングの完全十分条件の境界があることがわかる. この結果を利用して, R_{direct} のサイズを決定し, 圧縮センシング行列 R_s を導出する. 以降では, この値を用いて評価実験を行う.

5. 評価実験

実際の適用場面では様々な状況が想定される. 例えば, 検知後にすぐにデータを利用するだけではなく, その後時間が経過後, 蓄積データが必要になった時に復元を行う状況. あるいは事前に未来のデータというものがどのような性質を持っているのかわからないため, 圧縮し過ぎて復元が出来なくなってしまうこと, あるいはその逆にサンプル数を多くとり過ぎてしまい計算時間を増加させてしまう状況等が考えられる.

そこで提案手法を使うことで従来手法では対応することが難しかった状況, あるいは計算量を増加させてしまっていた状況に対応することが可能となったことを比較実験を行うことで確かめた. 計算時間と復元時の元信号との差を数値として評価した. 計算時間は一番先頭のデータが入力されてから, 元のテンソルに復元するまでの時間である. また, 元信号との差は直感的にどのくらい復元したデータと違っているのかわかりやすいように設定した. 具体的には入力信号のデータの中心を 0, 1 以外とらないようにした. そして, 復元したデータは中心のデータそれぞれの成分の値を 1 以下の場合は小数点第 1 位で四捨五入を行い, 1 以上に復元された場合を 1 にすることで復元データも 0, 1 にすることで何個, 元データ成分同士を比べることにより元データとの差を調べた. この値が 5 と出ているときは一つのテンソルに平均で 5 個の成分が間違っていて復元されているとみることができる. 25 個のテンソルストリームを 3 パターン入力信号として作成した. 詳しいデータセットは表 2 に示した. サンプル数は 23 パターン用意した. 提案手法では, 初めに圧縮した際のサンプル数 300, 600, 900 からそれぞれその数以下のサンプル数で再圧縮をした. それぞれが再圧縮したサンプル数が同じなら, ほぼ同じ挙動を計算時間, 復元したデータとの差をとったため図は始めの圧縮時に, サンプル数が 900 のもののみを記載した. それぞれ結果は図 9 から 12 までに示した.

表 2: 入力データのデータセット

データセット	入力信号のテンソル
パターン1	20*20*20 非零数 20
パターン2	20*20*20 非零数 40
パターン3	20*20*20 非零数 80

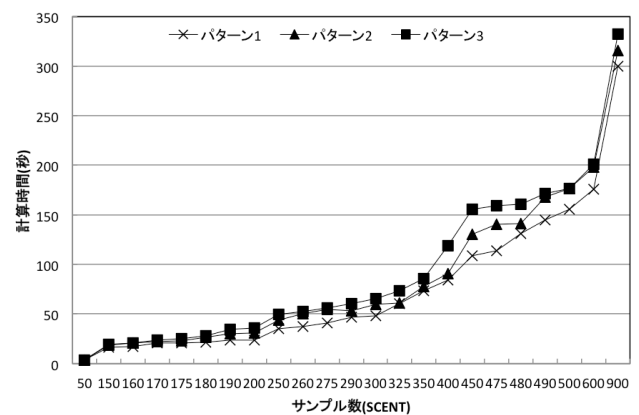


図 9: サンプル数と計算時間 (SCENT)

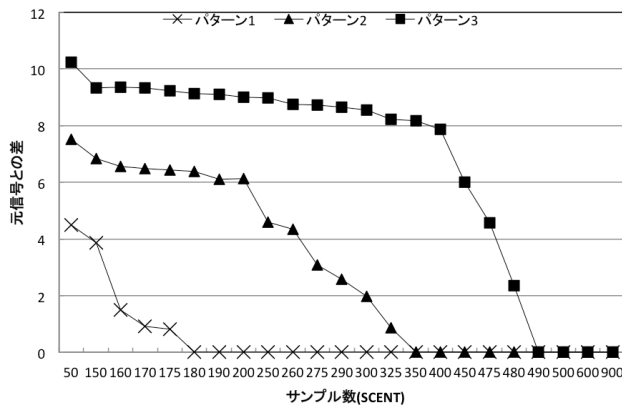


図 10 : サンプル数と計算時間 (SCENT)

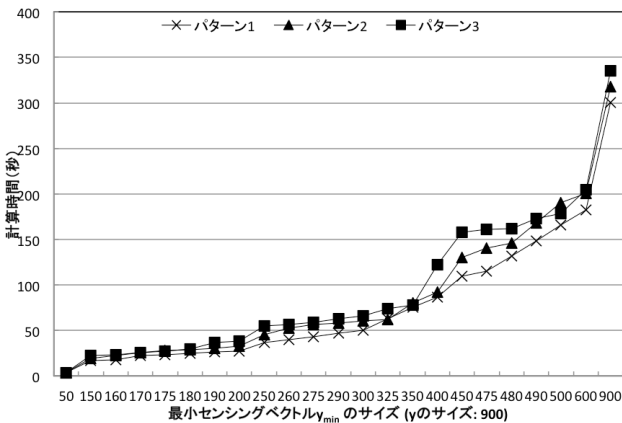


図 11 : y_{min} と計算時間 ($y=900$ から再圧縮)

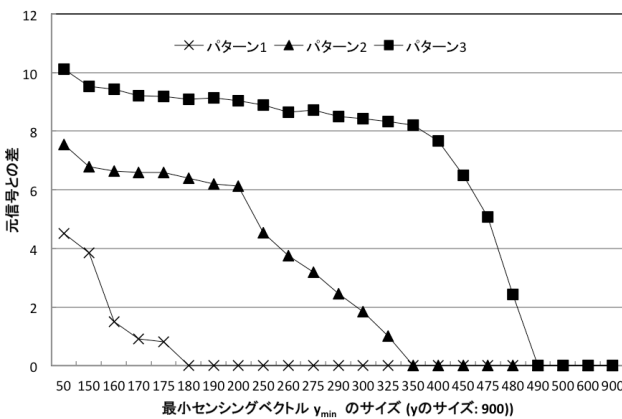


図 12 : y_{min} と元信号との差 ($y=900$ から再圧縮)

この比較実験から変化検知の段階ではデータの特性がわからなかったため、サンプル数を多めにとっていった状況から、従来手法では変化検知時と同じサンプル数で復元を行わなくてはならなかったため、余分な計算時間がかかっていた可能性がある。しかし、提案手法では変化検知を行い、どのような信号であるか、わかった後に再圧縮をすることが可能なため、復元時に必要最低限のサンプル数さえ所持していれば、計算時間を削減することが出来る可能性がある。例えば、パターン 1 を従来手法で復元を行った場合はまず、データの特性がわかっていないため復元ができない大きさにしてしまうこと可能性があるので大きめにサンプル数をとる必要がある例えばサンプル数を 300 でとった

場合、復元時にもその値で復元をする必要がある。しかし、提案手法では変化検知の時にサンプル数 300 で圧縮をして、検知が終わり入力データの特性がわかったところでさらに再圧縮を行うことができるので例えば元信号との差がサンプル数 300 とは変わらない境界のサンプル数であるサンプル数 180 まで下げて復元することが出来る。計算時間として 50%ほど削減することが可能である。さらに 2つのグラフのサンプル数が同じ箇所をみるとほとんど計算時間がかかっていないため逆行列から再圧縮データを作るためにかかる時間は少なく L1 最小化によって復元を行う箇所に時間がかかっていることも見て取ることが出来る。

6. まとめと今後の展望

今回、L1 最小化におけるサンプル数と計算時間と復元時のデータの関係から変化検知時と復元時でサンプル数を変えて再圧縮を行うことで計算時間を削減することが出来る可能性を示した。また、データが蓄積していくような状況を想定すると保持しておく必要があるデータ量も削減することが出来る可能性がある。また、比較実験を行うことにより従来手法では対応できない状況でも柔軟に対応できることを示した。しかし、今回サンプル数を導出する際には制御パラメータ β は実験から求めたものを利用しただけである。したがって、今後は計算式や、センシング行列、元データに条件付けをすることで、必要なサンプル数を理論的に一意に導出できるよう考えたい。また、今後は変化検知時にも着目したいと考えている。

参考文献

- [1] Tamara G. Kolda, and Brett W. Bader, "Tensor decompositions and applications," SIAM Review, vol. 51, issue. 3, pp. 455-500, 2009.
- [2] Y. Chi, B. L. Tseng, and J. Tatemura, "Eigen-trend: trend analysis in the blogosphere based on singular value decompositions," in Proc. of the 15th ACM international conference on Information and knowledge management (CIKM'06), pp. 68-77, 2006
- [3] J. Sun, D. Tao, and C. Faloutsos, "Beyond streams and graphs: Dynamic tensor analysis," In Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 374-383, 2006.
- [4] Yu-Ru Lin, K.S.Candan, H. Sundaram, and L. Xie, "SCENT: Scalable Compressed Monitoring of Evolving Multirelational Social Networks," ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP), vol. 7S, issue. 1, 2001.
- [5] E. Candes, and J. Romberg, "Sparsity and incoherence in compressive sampling," Inv. Probl. 23, 969-985, 2006.
- [6] E. Candes, and T. TAO, "Near-optimal signal recovery from random projections: Universal encoding strategies?" IEEE Trans. on Inform. Theory, vol. 52, no. 12, pp. 5406-5425, 2007.
- [7] R. Baraniuk, M. Davenport, R. Devore, and M. Wakin, "A simple proof of the restricted isometry property for random matrices," Constructive Approximation, vol. 28, issue 3, pp. 253-263, 2007.
- [8] Rauhut, H., Schass, K. and Vandergheynst, P. (2008): Compressed sensing and redundant dictionaries, IEEE Trans. on Information Theory, Vol.54, No.5, pp.2210-2219
- [9] M. Plumbley, "Recovery of sparse representations by polytope faces pursuit," in Proceedings of International Conference on Independent Component Analysis and Blind Source Separation, pp. 206-213, 2006.
- [10] Allen Y. Yang, Zihan Zhou, Arvind Ganesh, S. Shankar Sastry, and Yi Ma, "Fast l1-Minimization Algorithms for Robust Face Recognition," IEEE Transactions on Image Processing, vol. 22, issue 8, pp. 3234-3246, 2013.
- [11] l1 magic, <http://www.acm.caltech.edu/l1magic/>.