

階層的センサネットワークのための 特異値分解を用いたデータ圧縮手法

岸野 泰 恵^{†1} 櫻井 保 志^{†1}
前川 卓 也^{†1} 須山 敬 之^{†1}

本稿では、階層的な木構造のトポロジのセンサネットワークにおいて、特異値分解を用いてデータを圧縮して効率よくデータを収集する手法を提案する。提案手法では、各センサノードで特異値分解によってセンサデータ系列を基底系列とそれに対する重みに分解することで、データ量を圧縮する。さらに計算能力の限られるセンサノードでも十分に処理ができるよう、逐次処理的な特異値分解によって圧縮の計算を軽量化する。この処理を各階層で繰り返すことで大幅にデータ量を削減できる。評価実験では、提案手法によって階層が増加してもデータ量や誤差が低く抑えられること、および単純に特異値分解を用いてデータを集約する手法よりも計算量が低減することを示した。

SVD-based data compression for Hierarchical Sensor Networks

YASUE KISHINO,^{†1} YASUSHI SAKURAI,^{†1}
TAKUYA MAEKAWA^{†1} and TAKAYUKI SUYAMA^{†1}

In this paper we propose a data compression method for efficient data gathering in hierarchical sensor networks. The proposed method compresses sensor data sequences by decomposing the data sequences to base sequences and weights using Singular Value Decomposition (SVD). Moreover, we also reduced the computation time of the compression by adopting incremental SVD. We can reduce the amount of sensor data dynamically by repeating the data compression hierarchically. Our experimental result shows the proposed method can reduce the amount of data and error when the level of network topology is high and computational time.

1. はじめに

近年、多数のセンサノードからデータを収集するセンサネットワークに関する研究が盛んに行われ^{5),9)}、環境モニタリング、遠隔監視、科学実験、ライフログなどさまざまな応用が実現可能になりつつある。このような状況では、センサノードはさまざまな場所に埋め込まれ、光、温度、湿度、加速度、照度、ガス濃度、風力などさまざまな現象をセンシングする^{1),10)}。その結果は基地局へ向けて送信され、基地局はアプリケーションの要求に応じて収集したデータの分析や処理を行う。

さらに最近では、多数のセンサノードを同時に用いて、高精細な監視や、詳細な分析などを行うより高度なアプリケーションに関心が高まっている^{11),14)}。こういったアプリケーションを実現するためには、効率的なデータ収集手法が必要となる。本研究では、(1) 高速にデータを収集でき、(2) 収集したデータは正確であり、(3) センサノードが低消費エネルギーで稼働できるような手法を効率的と考える。逆に、(1) データ収集の遅延が大きく、(2) 収集したデータが大きな欠損や誤差を含み、(3) 無線通信において冗長なデータの送受信やデータの再送などによって電池を大きく消耗してしまうようなデータ収集手法は効率が悪いと言える。

多数のセンサノードからなるセンサネットワークでは、階層的なクラスタを用いた木構造のトポロジを用いると、無線通信帯域を効率的に利用でき、木構造のトポロジによってセンサノードの数が増加したときにスケラビリティが高いため、効率的にデータを収集できる。一般的にセンサノードにおいては無線通信によるデータ転送に最も電力を消費するため、階層的にデータを集約してデータ量を大きく削減できれば、効率のよいデータ収集を実現できる。データ処理の計算に必要な電力は無線通信と比べれば少ないが、できるだけ軽量のデータ処理方法が望ましい。

センサノードが計測して取得するセンサデータには以下のような特徴がある。

センサノード間のセンサデータの類似性: 近接して設置されたセンサノード間では、発生したセンサデータの系列は高い類似性をもつ。例えば温度やガスは壁などで空間が区切られない限りはセンサデータはほぼ一様に分布し、物の動きを加速度センサで測定する場合には同じ物についているセンサからは相関のあるデータが発生する。複数のセンサ

^{†1} NTT コミュニケーション科学基礎研究所
NTT Communication Science Laboratories

ノードからのデータに類似性があるのであれば、類似の部分をもとめることでデータ量を大幅に削減できる。

センサデータの周期性: センサデータには周期性がある。温度や湿度は一日の周期で変化し、環境に設置したセンサで人の動作を計測するときには人が歩くなど同じ動作を繰り返せば周期性のある変化をする。周期的にデータが繰り返されるのであれば、繰り返される部分をもとめて圧縮することで効率のよい通信が実現できる。

このように、センサデータの特徴を活用してデータの集約が行えればセンサデータの量を劇的に削減できる可能性がある。

筆者らは、センサデータの周期性とセンサノード間の類似性、階層的ネットワークの特徴を活かした効率的なデータ収集手法に関して研究を進めている⁷⁾。データを圧縮する手法としては特異値分解を用いる。後述するように特異値分解を用いると、センサの時系列データを代表的なデータ系列（基底系列）とそれに対する重みの係数値に分解できる。周期性のあるセンサデータでは、類似したデータ系列が周期的に発生するため、少ない数の基底系列でも十分にデータを近似できる。また、階層的なネットワークトポロジでは、近隣のセンサノードで類似したデータ系列が発生しているため、クラスタ内でデータを集約する際に、効率よくデータ量を削減できる。このような特異値分解によるデータ集約を階層的に繰り返すと、最終的に基地局には全センサノードで共通の基底系列とそれに対する各センサノードのデータを復元するための係数値が集まり、データ量は大幅に減少する。一方で特異値分解は一般的に計算量が多く、センサノードで特異値分解を用いたデータ系列を圧縮すると、計算に時間がかかるためにデータの送信に遅延が発生したり、電池を余分に消費する可能性がある。そこで本稿では、計算量を抑えながら、効率よくデータを圧縮して収集する手法について述べる。

2. 研究の背景

2.1 階層的センサネットワーク

階層的なネットワークトポロジのセンサネットワークでは、ネットワークのトポロジが木構造であるため、センサノードの数が増えても効率的にデータを収集できる。図1に階層的センサネットワークのイメージ図を示す。まず最下位の階層では、各ノードは近隣に配置された他のセンサノードとクラスタを生成し、その中の1台（親ノード）へデータを集める。クラスタ内のその他のセンサノードは子ノードとなる。親ノードは他の親ノードと1段上位階層のクラスタを生成する。このクラスタでさらにデータを集約し、さらに上位階層のクラ

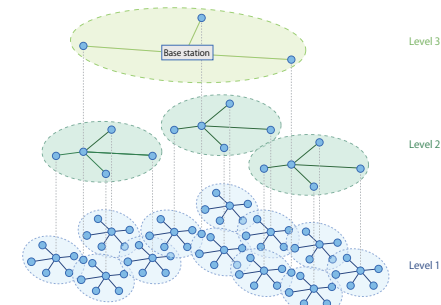


図1 階層的センサネットワーク

スタでデータを集約することを繰り返し、最終的には最上位の階層にある基地局にすべてのデータが集約される。階層の高さはセンサノードの数を N とするとき、 $O(\log N)$ となり、センサノードが多数ある場合でもそれほど多段のネットワーク構成になるわけではない。

ネットワークのトポロジは、基地局が無線通信の通信範囲や衝突、電力消費などを考慮してあらかじめ決定するものとする。提案手法は、基地局にネットワーク上の全センサノードからデータを収集する際に、センサデータの特徴を考慮して効率のよいデータ収集を実現するものであるため、センサノードの通信方式に関わらず効果のある手法である。センサデータを集約していく上では、基地局はデータを収集する木構造のネットワークの構成を把握し、その情報をもとに各センサノードで発生したデータを復元する。センサノードは、基地局から指定された自身の親ノードと子ノードの情報を持ち、それによってデータを送受信する。

2.2 関連研究

階層的センサネットワークにおけるデータ収集に関しては、これまでもさまざまなアプローチの研究がおこなわれている。例えば、文献⁶⁾では、センサデータに関連性があるセンサノードのデータが同じ経路上で収集されるようにネットワークトポロジを構成する手法が提案され、文献²⁾では、階層的なネットワーク上で、データを SUM, AVG, COUNT, MIN, MIX といった関数に要約する手法が提案されている。

また、センサデータの特徴を活かしたデータ処理に関しては、文献³⁾では、複数のセンサデータ間の類似性を活かして、データ量を削減する手法が提案され、回帰分析の技術を用いて、基底となる信号に対する係数値で新たに発生したデータを表現し、データの大幅な圧縮を図っている。文献⁴⁾では、適応フィルタを利用してデータ系列の変化を予測しながらデー

タを収集する手法が提案されている。

これに対して本研究では、センサデータの周期性とセンサノード間の類似性という特徴を活かし、階層的ネットワークにおいて効率的にデータを収集するための圧縮手法について研究を進めている。

3. 特異値分解を用いたセンサデータ系列の圧縮手法

3.1 センサネットワークにおけるデータ系列の圧縮

提案手法では、特異値分解を用いてデータの集約を行う。特異値分解を用いると、センサデータ系列の代表的な変化を基底系列とし、新たなデータ系列を基底系列に対するわずかな数の重みの係数で近似でき、少ない誤差でデータを圧縮できる。フーリエ変換を使っても同じように三角関数に対する重みの係数でデータ系列を表現できるが、特異値分解を用いた方がセンサデータの特徴をとらえた系列を基底系列とすることができ、効率的である。

単体のセンサノードでは、一定時間毎にデータ系列を区切り、複数個のデータ系列をまとめて特異値分解によって基底系列とそれに対する重みの係数値に分解する。周期的な繰り返しのあるデータ系列では代表的な変化が基底の系列に現れ、効率よく近似できる。さらに本論文では、階層的に特異値分解を繰り返し、データを収集する手法を提案する。末端のセンサノードでは、発生したセンサデータを基底系列と係数値に分解し、親ノードに送信する。親ノードでは、受信した複数の子ノードからの基底系列をまとめ、特異値分解によって代表的な基底系列と係数値に分解する。この操作を階層的に繰り返すと、上位の階層では、センサノード間で共通の基底系列に集約される。階層的な特異値分解を用いた研究例としては、周期性のある時系列データから代表的なパターンを取り出す手法が提案されている¹³⁾。筆者らの手法では、この研究例を参考に、センサネットワークにおけるセンサデータの集約に対して階層的に特異値分解を適用している。

3.2 特異値分解

本稿では、 n 個の要素からなるベクトルは $\mathbf{v} \equiv [v_1 v_2 \cdots v_n]^T$ と小文字のボールド体で表記し、 $m \times n$ の行列を \mathbf{A} と大文字のボールド体で表記する。行列 \mathbf{A} の列ベクトルは $\mathbf{a}_{(i)}$ 、行ベクトルは \mathbf{a}_j とする。つまり、 $\mathbf{A} \equiv [\mathbf{a}_{(1)} \mathbf{a}_{(2)} \cdots \mathbf{a}_{(n)}] \equiv [\mathbf{a}_1 \mathbf{a}_2 \cdots \mathbf{a}_m]^T$ となる。

特異値分解は線形変換の一種であり、任意の $m \times n$ の行列 \mathbf{A} を、

$$\mathbf{A} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T \quad (1)$$

と分解できる。ここで \mathbf{A} のランクを r とすると、 \mathbf{U} は、 $m \times r$ のユニタリ行列、 \mathbf{V} は $n \times r$

のユニタリ行列、 $\mathbf{\Sigma}$ は $r \times r$ の対角行列である。 $\mathbf{\Sigma} \equiv \text{diag}[\sigma_1 \sigma_2 \cdots \sigma_r]$ は正の値 σ_i からなる対角行列であり、 \mathbf{A} の特異値を指す。対角成分 σ_i は降順に並べられている。 \mathbf{U} 、 \mathbf{V} はユニタリ行列であるため、各成分 $v_{(i)}$ 、 $u_{(i)}$ は単位ベクトルとなる。

特異値分解の適用例の一つとして、行列の近似が挙げられる。降順に並んだ特異値の上位 k 個のみで行列を表現すると、 \mathbf{A} と \mathbf{A} を近似した行列 $\tilde{\mathbf{A}}$ の間の二乗誤差が最小となる。

$$\tilde{\mathbf{A}} = \tilde{\mathbf{U}} \tilde{\mathbf{\Sigma}} \tilde{\mathbf{V}}^T \quad (2)$$

ここで、 $\tilde{\mathbf{U}} := [\mathbf{u}_{(1)} \mathbf{u}_{(2)} \cdots \mathbf{u}_{(k)}]$ 、 $\tilde{\mathbf{V}} := [\mathbf{v}_{(1)} \mathbf{v}_{(2)} \cdots \mathbf{v}_{(k)}]$ 、 $\tilde{\mathbf{\Sigma}} = \text{diag}[\sigma_1 \sigma_2 \cdots \sigma_k]$ とする。

(2) 式の変換において、 $\tilde{\mathbf{P}} = \tilde{\mathbf{U}} \tilde{\mathbf{\Sigma}}$ とすれば、

$$\tilde{\mathbf{A}} = \tilde{\mathbf{P}} \tilde{\mathbf{V}}^T \quad (3)$$

と変形でき、このように $\tilde{\mathbf{A}}$ は、 $\tilde{\mathbf{P}}$ と $\tilde{\mathbf{V}}$ を用いて近似できる。 $m > k$ かつ $n > k$ であるため、センサノードは $m \times n$ の \mathbf{A} の代わりに、より少ないデータ量の $m \times k$ の行列 $\tilde{\mathbf{P}}$ と $n \times k$ の行列 $\tilde{\mathbf{V}}$ を送信するだけで、基地局ではセンサデータを復元できる。しかも、特異値分解の特性上、任意の k に対して二乗誤差が最小となる。

3.3 データの圧縮

センサノードでは、連続したセンサデータ系列を長さ l のウィンドウに区切り、 c 個の部分系列を \mathbf{A} として保持する ($l > c$ とする)。センサデータで発生したデータ系列を $\{x_1, x_2, \dots, x_{cl}\}$ とすれば、 \mathbf{A} の i 番目の行 a_i に部分データ系列 $\{x_{(i-1)l+1}, x_{(i-1)l+2}, \dots, x_{il}\}$ を保存することになる。この際に部分データ系列の平均と分散を算出して \bar{a}_i と s_i とし、正規化を行った上で保存する ($a_{ij} = (x_{(i-1)l+j} - \bar{a}_i) / s_i$)。 c 個のデータ系列に対して、(3) 式のように \mathbf{A} を特異値分解によって、 \mathbf{P} と \mathbf{V} に分解すれば、 \mathbf{V} から上位 k 個の $v_{(i)}$ を基底となる系列と、これに対する係数行列 $\tilde{\mathbf{P}}$ に \mathbf{A} を圧縮できる。基地局では、センサノードから受け取った $\tilde{\mathbf{V}}$ と $\tilde{\mathbf{P}}$ を掛け合わせて元のデータ系列 $\tilde{\mathbf{A}}$ を復元する。

図 2 に特異値分解を用いたデータ圧縮の例を示す。図 2 では、ある地点の温度センサのデータを例とし、一日毎にウィンドウを区切っている。4 つの部分系列 $\mathbf{a}_1 \dots \mathbf{a}_4$ が、特異値分解によって基底系列 $\mathbf{v}_{(1)}$ 、 $\mathbf{v}_{(2)}$ と 8 つの係数値 p に圧縮されている。この例に限らず、多くのセンサデータには周期性があり、適切なウィンドウ幅 l を用いれば、 d に小さな値を設定してデータ量を抑えても、少ない誤差でデータを圧縮できる。周期が前もってわからない場合には、文献¹³⁾などで提案されている自動的にデータの周期を解析する方法を用いれば適切な値を決定できる。実際には、 l 、 c 、 d といったパラメータは無線通信のパケットサイ

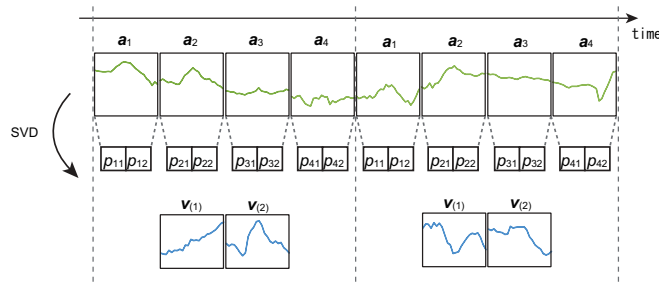


図2 特異値分解によるデータの圧縮 ($c = 4, d = 0.5$)

ズ、センサノードのメモリ量や処理能力も考慮した上で決定する必要がある。

実際のセンサデータでは、周囲の状況に変化がなければ同じようなデータの変化を繰り返すため、データが c 個発生するたびにデータの処理を行っても、基底系列はほぼ同じものが選択されると予想できる。このため、センサデータの傾向が変化するまで、同じ \tilde{V} を繰り返し使用し、傾向が変化したときにのみ \tilde{V} を送信し直しても、基地局ではデータをよく近似できる。センサノードは、データの傾向に変化がなければ、大きなデータ量の \tilde{V} は低い頻度で送信し、少ないデータ量の p_i のみを送信すればよいため、大きく圧縮の効率を上げられる。

すでにある基底の系列 V に対する係数行列 P は以下のようにして計算する。式 (4) は、 V がユニタリ行列であるため、

$$P = AV \quad (4)$$

と変形できる。この式から、 \tilde{P} は

$$\tilde{P} \approx \tilde{A}\tilde{V}. \quad (5)$$

によって近似できると考えられる。そこで、 a_i が更新されたときに、この a_i を基底の系列の和で表現するための重みを、

$$\tilde{p}_i = a_i \tilde{V} \quad (6)$$

と算出する。データを復元する際には、

$$\tilde{a}_i = \tilde{p}_i \tilde{V}^T \quad (7)$$

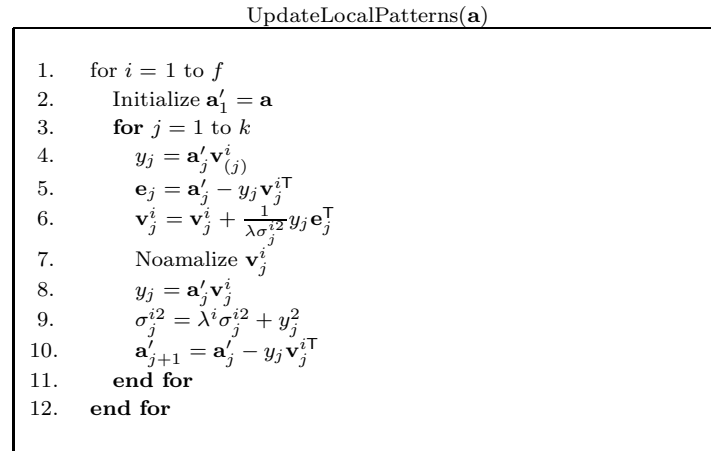
と計算する。新たなデータ系列 a_i がこれまでと同じ傾向であればこのような方法で算出した \tilde{p}_i を用いて少ない誤差で近似できるが、データ系列の傾向に大きな変化があった場合に

は、上式のような算出方法では誤差が大きくなるため、誤差が閾値 ϵ を超えれば基底系列を再度送信するものとする。誤差の量は単純にはデータ a_i と (7) 式で復元されたデータ系列 \tilde{a}_i を比較すればよいが、(7) 式の計算には多数の乗算が発生するため、 \tilde{p}_i から \tilde{V} を用いて射影できたセンサデータ系列のエネルギー ($|\tilde{p}_i|^2$) を計算して誤差を見積もる。

誤差の大きさを検討する際には、新しく発生したデータ系列のみの誤差を計算するのではなく、最近の c 個のデータ系列から誤差を計算している。一時的にデータ系列の傾向が変化する度に新たな \tilde{V} を送信すると効率が悪いので、最近の c 個で誤差が大きくなることを確認してから基底系列を送信する。一時的に傾向の変化があった際に誤差が増加する可能性はあるが、その後、元の傾向に戻る可能性を考慮してこのような手法としている。また、センサノードでは、長さ l のデータが発生するとすぐに、 \tilde{V} を用いて \tilde{p}_i を計算してデータを送信できるため、データ収集の遅延低減にもつながる。

このようにデータの傾向が変化するまで一つの基底系列を使用し続けられれば、大幅に大幅に少ないデータ量でセンサデータを収集できるようになるが、基底系列を再送する際には特異値分解を行う必要がある。特異値分解の計算には一般的に計算量が必要であり、この計算自体がセンサノードの電力を消費する可能性がある。軽量な特異値分解の計算方法はこれまでにさまざまなものが提案されているが、本稿では、ストリームデータ処理の分野で実績のある文献⁽¹²⁾で提案されているインクリメンタル特異値分解、すなわち逐次的に特異値分解の更新を行うアルゴリズムを用いて計算量を抑える。この手法では、忘却変数 (λ) と呼ぶ変数を用いて現在のデータの傾向と過去のデータの傾向を重視する割合を調整できる。本研究はできるだけデータ量を抑え、その中で少ない誤差でデータを収集することを目的としているが、最適な λ の値は状況によって異なるため、 λ の値の異なる基底系列を複数用意する。データが発生する度にすべての基底系列の更新を行い、基底系列を上位ノードに送信する際には、誤差が最小になると予想されるものを選択する。インクリメンタル特異値分解は計算量が軽いため、複数の基底系列の更新を行ってもそれほどセンサノードの負担にはならない。誤差の予想には、前述の基底系列再送を判断する際と同様に射影できたデータ系列のエネルギーを用いる。基底系列を選択する際にも過去の複数のデータ系列も考慮した上で最適なものを選択することが望ましいが、特に計算量の限られるデバイスでは一部のデータ系列のみのエネルギーから基底系列を選択してもよいものとする。

図3にアルゴリズムの詳細を示す。新たなデータ系列 a で基底系列 V を更新する際には、初期化を行った後、基底系列の各列ベクトルに対して4行目から10行目の処理を繰り返す。まず、基底系列の j 番目のベクトルで a_j^t を射影し、その結果を y_j とする (4行目)。得ら



$\mathbf{V}^i, \sigma^i, \lambda^i$: i 個目の基底系列とそれに対応する特異値と忘却変数。 \mathbf{V}^i の初期値には任意の単位ベクトル, 特異値 σ^i には小さな値を設定する。
 f : 忘却変数の個数. y_j : \mathbf{a}'_j を計算中の $\mathbf{v}_{(j)}^i$ に射影した結果.
 \mathbf{e}_j は射影による誤差.

図 3 インクリメンタル特異値分解のアルゴリズム

れた y_j を用いて \mathbf{a}'_j で射影しきれなかった誤差ベクトル \mathbf{e}_j を算出し (5 行目), 基底ベクトル $\mathbf{v}_{(j)}^i$ を忘却変数 λ を考慮した上で, 特異値 σ_j^i と y_j に応じて誤差ベクトルの方へ更新する (6 行目). その後, 基底ベクトル $\mathbf{v}_{(j)}^i$ を正規化し (7 行目), 正規化後の基底ベクトルを用いて再度 y_j を算出する (8 行目). y_j を用いて σ_j^i を更新し (9 行目), $\mathbf{v}_{(j)}^i$ を用いて射影できたデータを \mathbf{a}'_j から差し引いたものを \mathbf{a}'_{j+1} とする (10 行目). 以上の処理を送信する基底系列の個数分 (k 回) 繰り返す. また, この処理が終了する毎に, 特異値分解と同様に基底系列を特異値の大きさの順に並べなおす. なお, 更新の閾値 ϵ と比較して十分なエネルギーのデータが射影できた際には, 計算を打ち切り, 以降の重み p は 0 とする.

3.4 階層的なデータの集約

次に, 前節で述べた単体のセンサノードでのデータ圧縮手法を階層的に適用し, センサネットワーク全体からデータを集約する手法について述べる. 図 4 に階層的に特異値分解を適用したデータの集約手法のイメージ図を示す. 末端のノードでの圧縮処理も中間のノードで子ノードからのデータを取りまとめて圧縮する処理も同様な処理を行う. 以降では, h 階層目のデータ系列を $\mathbf{A}^{(h)}$, 基底系列を $\mathbf{V}^{(h)}$ などとし, センサノード i を $N_i^{(h)}$ と表記す

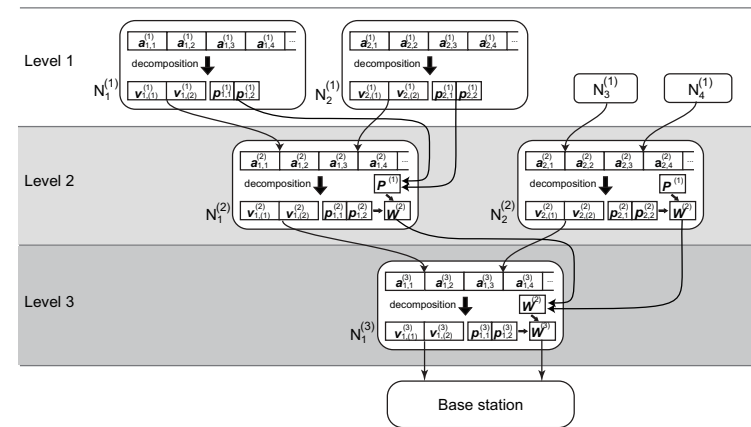


図 4 階層的特異値分解によるデータの圧縮 (2 分木のトポロジ, $c = 4, d = 0.5, h = 3$)

る. また, 特にセンサノード i を区別して各変数を記述する時には, h 階層目の i 番目のセンサノードの基底系列を $\mathbf{V}_{i,1}^{(h)}$, 同じセンサノードの 1 番目のデータ系列を $\mathbf{a}_{i,1}^{(h)}$ などと表記する. ここでは最下位層のセンサノードで発生したデータを集約する手法について述べるが, 途中の階層のノードで発生したセンサデータについても各階層で並行して処理を行ってデータを収集する. 例えば 2 階層目のセンサノードでは, 1 階層目で発生したデータを中継する処理だけでなく, 自ノードで発生したデータを圧縮して送信する処理も行う.

1 階層目で発生したセンサデータはこれまでに述べた方法によって基底系列 $\tilde{\mathbf{V}}^{(1)}$ とこれに対する重みの係数値行列 $\tilde{\mathbf{P}}^{(1)}$ に分解され, 親ノードである 2 階層目のノードへ送信される. $\mathbf{A}^{(1)}$ の平均と分散は, そのまま基地局まで他のデータと共に転送される. 2 階層目のノードでは, 受け取った基底系列を次のように $\mathbf{A}^{(2)}$ に保存し,

$$\mathbf{A}^{(2)} = \begin{bmatrix} \tilde{\mathbf{V}}_1^{(1)} \\ \tilde{\mathbf{V}}_2^{(1)} \\ \vdots \\ \tilde{\mathbf{V}}_n^{(1)} \end{bmatrix} = \mathbf{P}^{(2)} \times \mathbf{V}^{(2)T}. \quad (8)$$

子ノード 1 から n の基底系列をまとめて特異値分解する. また, 1 階層目ですでに正規化は行っているため, 2 階層目以降では正規化は行わない. この階層でのデータ処理によって得られた $\tilde{\mathbf{V}}^{(2)}$ を自身の親ノードである 3 階層目のセンサノードへ送信する.

重みの係数値行列 $\tilde{\mathbf{P}}^{(1)}$ は、基底系列 $\tilde{\mathbf{V}}^{(1)}$ に比べればデータ量が少なく、このままデータを基地局まで転送してもよいが、最終的に基地局では各階層で発生した $\tilde{\mathbf{P}}^{(h)}$ を各ノード毎に掛け合わせてデータを復元するため、中継のセンサノードであらかじめ掛け合わせながらデータを転送する。各階層で掛けあわせた結果を $\tilde{\mathbf{W}}_i^{(h)}$ とすれば、基地局では、

$$\tilde{\mathbf{A}}_i^{(1)} = \tilde{\mathbf{W}}_i^{(h)} \times \tilde{\mathbf{V}}^{(h)\top} \quad (9)$$

とデータ系列が復元できる。

これを階層的に繰り返す。最終的に基地局には、最上位の階層で計算された基底系列 $\tilde{\mathbf{V}}^{(h_{max})}$ 、各センサノードで発生したデータ系列の $\tilde{\mathbf{V}}^{(h_{max})}$ に対する重みの係数値行列 $\tilde{\mathbf{W}}^{(h_{max})}$ と、1階層目で発生した平均と分散が収集される。基地局は式(9)のように $\tilde{\mathbf{W}}^{(h_{max})}$ と $\tilde{\mathbf{V}}^{(h_{max})}$ を掛け合わせてデータ系列 $\tilde{\mathbf{A}}^{(1)}$ を復元する。なお、ここでは説明の都合上、全階層の全センサノードで同じ k を用いる場合について説明したが、階層やセンサノード毎に異なる k を用いても構わない。

図5に以上をまとめたデータ収集手法のアルゴリズムを示す。図5に示したのは、中間の階層 h のあるノードにおいて、 $h-1$ 階層の i 番目の子ノードから送信された基底系列 $\tilde{\mathbf{V}}_i^{(h-1)}$ と、 $h-1$ 階層で計算された係数値行列 $\mathbf{w}^{(h-1)}$ を受信した際のデータ処理である。自ノードのIDの添え字は省略している。末端のセンサノードの場合、受信したデータではなく、発生したセンサデータ系列を処理することになる。中間ノードでは、受信した $\tilde{\mathbf{V}}_i^{(h-1)}$ を $\mathbf{A}^{(h)}$ に保存し、基底系列の更新を行う(1-4行目)。最後に送信した基底 $\mathbf{V}^{(h)}$ を用いて近似した際の誤差 e を計算して(5-6行目)、誤差が閾値 ϵ よりも大きければ、傾向が変化したと判断し、最適な忘却変数の基底系列を取得する(8行目)。この際、新たな基底系列によって誤差が小さくなることを確認し(9-11行目)、誤差が小さくなる場合にのみ、基底系列を $\tilde{\mathbf{V}}^{(h)}$ を自身の親ノードへ送信する(11-14行目)。最後に、 $\tilde{\mathbf{P}}^{(h)}$ と $\mathbf{W}^{(h-1)}$ から $\mathbf{W}^{(h)}$ を算出して上位階層へ転送する(16行目)。 \mathbf{W} のみを受信したときには、最新の係数行列 \mathbf{P} を用いて16行目の処理のみを行う。また、図5では行列全体の乗算をしているように記述している部分があるが、実際には更新部分のみを計算する。

4. 評価実験

シミュレータを用いた評価実験により提案手法の有効性を検証し、さらに提案手法を実機のセンサノードに実装した結果について報告する。

```

1. for  $j = 1$  to  $k$ 
2.    $\mathbf{a}_{(i-1)k+j}^{(h)} = \tilde{\mathbf{v}}_{i,(j)}^{(h-1)}$ ;
3.   UpdateLocalPattern( $\mathbf{a}_{(i-1)k+j}^{(h)}$ ); // Figure 3
4. end for
5.  $\tilde{\mathbf{P}}^{(h)} = \mathbf{A}^{(h)} \tilde{\mathbf{V}}^{(h)}$ ;
6.  $e = \sum_{i=0}^c (|\mathbf{p}_i^{(h)}|^2 - \text{power of } \mathbf{a}_i^{(h)})$ ;
7. if  $e > \epsilon$  then
8.   GetLocalPattern( $\tilde{\mathbf{V}}^{(h)}$ );
9.    $\tilde{\mathbf{P}}^{(h)} = \mathbf{A}^{(h)} \tilde{\mathbf{V}}^{(h)}$ ;
10.   $e' = \sum_{i=0}^c (|\mathbf{p}_i^{(h')}|^2 - \text{power of } \mathbf{a}_i^{(h)})$ ;
11.  if  $e' < e$  then
12.     $\tilde{\mathbf{V}}^{(h)} = \tilde{\mathbf{V}}^{(h')}$ ,  $\tilde{\mathbf{P}}^{(h)} = \tilde{\mathbf{P}}^{(h')}$ ;
13.    Send  $\tilde{\mathbf{V}}^{(h)}$ ;
14.  end if
15. end if
16. Send  $\mathbf{W}^{(h)}$  computed from  $\mathbf{W}^{(h-1)}$  and  $\tilde{\mathbf{P}}^{(h)}$ 

```

$\tilde{\mathbf{V}}^{(h)}$: ここでは、最後に送信した基底系列。

$\tilde{\mathbf{P}}^{(h)}$, $\mathbf{V}^{(h)}$, $\mathbf{A}^{(h)}$, e' : 一時変数。

GetLocalPattern(\mathbf{V}): 最適な忘却変数の基底系列を \mathbf{V} に代入する関数。

図5 階層的オンライン処理アルゴリズム

4.1 シミュレーション実験

シミュレーションでは、完全4分木のネットワークを想定し、末端の各センサノードで発生したセンサデータを1台の基地局に収集する際のデータの圧縮率と誤差の割合を評価した。実験で用いたデータセットの詳細を表1に示す。AMEDASの2007年7月から2008年6月の気象データから欠損の少ない地点の温度のデータと、屋内の気温と湿度を約10日間計測したデータを用いた。実験では $c = 32$, $d = 0.125$ とし、忘却変数 λ は 0.95, 0.98, 1.0 の3種類とした。

実験では、 c 個のデータが発生する度に単純に特異値分解によってデータを圧縮するバッチ手法と提案手法を比較する。階層的にデータの圧縮を行っているのは基底系列 \mathbf{V} に関しのみであり、係数行列 \mathbf{W} は両手法で同じように上位階層へ転送しているため、圧縮率に関する評価は基底系列のみを対象とした。圧縮率は、全センサノードで発生したセンサデータ系列のデータ量に対する、基地局が直接受け取った基底系列のデータ量の割合である。

表 1 実験データ

データの種類	AMEDAS	屋内の気温, 湿度
データ長	87600 (10 分間隔, 1 年間)	30000 (30 秒間隔)
データの周期	144 (1 日)	2880 (1 日)
ウィンドウサイズ l	72	180

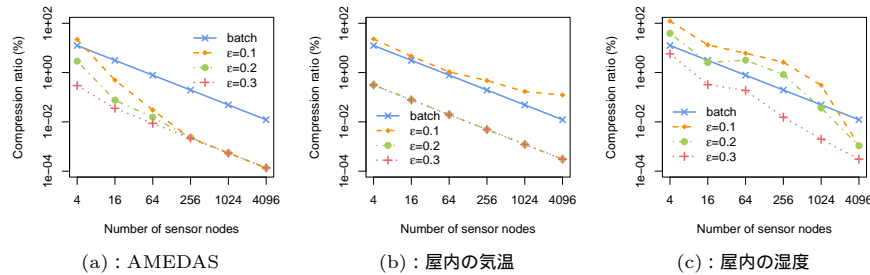


図 6 実験結果 (圧縮率)

図 6 と図 7 に実験結果を示す．実験では，階層数 $h = 1..6$ の 4 分木のトポロジを用意し，各階層において 4^{6-h} 回実験を行いその平均を求めた．図 6 では，横軸はセンサノード数を対数で，縦軸は圧縮率（百分率）を対数で表示し，図 7 では横軸はセンサノード数を対数で，縦軸は正規化したデータにおける元のデータ系列と復元したデータ系列の間の二乗誤差の平均を表示している．

どのデータの場合も，全体的にバッチ手法よりも提案手法の方が誤差は多いものの，圧縮の効率が良いことが確認できる． ϵ については，値を小さくすると誤差が減少するが，データ量は全体的に増加する．また，湿度のデータは，日による傾向の差が大きく気温を用いた他のデータセットより効率が悪くなっているが，AMEDAS の気温データのように地域毎の温度の高低があっても大まかな傾向が全体で同一であれば効率よくデータを収集できることが確認できた．また，圧縮率は階層数が増えるにつれて向上するが，誤差はどのデータでもノード数の増加と比較してわずかである．このことから提案手法は，多数のノードからなるセンサネットワークにおいて，センサノードの数が増加したとしても効率よくデータを収集できる手法であることが明らかになった．

表 2 にインクリメンタル特異値分解を用いたオンライン手法の計算時間を示す．時間の計測は Windows の PC (CPU: Intel Core 2 1.88GHz) で行い，表にはバッチ処理手法の計算時間に対する割合を示している．表より，いずれのデータにおいても，提案手法の方が

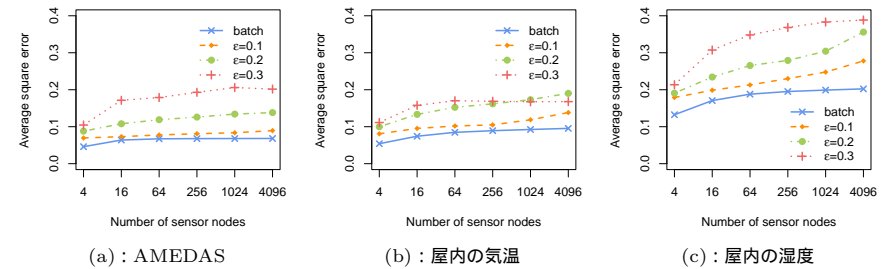


図 7 実験結果 (誤差)

表 2 オンライン手法の計算時間 (バッチ処理手法の計算時間に対する比)

データの種類	AMEDAS	屋内の気温
計算時間 (1 階層)	0.73	0.44
計算時間 (6 階層)	0.66	0.40

高速に計算できることがわかった．屋内の気温のデータの方がより高速になっているのは，データ系列が長いためである．これより，単純に特異値分解を定期的に行うバッチ処理手法よりも，提案手法は圧縮率の点で効率的なだけでなく，計算量の点でも短時間で計算を終えられるため，センサノードの電池の消費を抑えられることが示された．

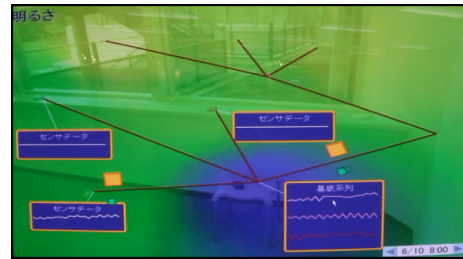
4.2 実機のセンサノードでの動作確認

提案する手法を IEEE 802.15.4 の無線通信モジュールを搭載したセンサノード (図 8(a)) 実装し，実際にセンサデータを収集するシステムを試作して提案手法の動作を確認した．無線通信モジュールには TWE-001 (東京コスモス電機) を用い，照度センサの値を収集した．TWE-001 は 32bit の RISC 型 MPU であり，筆者らがこれまでに提案した CIL 仮想マシン⁸⁾ を載せ，その上に Visual C# でデータ収集のソフトウェアを実装した．センサノードは 16MHz で動作し，実行速度やメモリ量を考慮し， $l=24$ ， $c=12$ とし，500 ミリ秒間隔で照度データを測定することとした．

実際に 8 台のセンサノードからデータを収集している様子を図 8(b) に示す．この図では，センサを配置した範囲をカメラで撮影し，撮影した画像の上に得られた照度を明るければ黄色，暗ければ紺色を塗って表示した．さらに，動作を確認するため，中間のノードでの基底系列と，復元した末端のセンサノードでのデータ系列を表示している．実際にデータ収集を行ったところ，各センサノードからのデータがおおよそ 12 秒に 1 回到着することと，あるセンサノードの明るさの傾向が変化すると基底系列の再送が行われることを確認した．



(a) : センサノード (無線通信モジュール, 照度センサの他に温湿度センサ, 加速度センサ, 有機 EL ディスプレイを搭載.)



(b) : 照度可視化ソフトのスクリーンショット

図 8 実機のセンサノードを用いたデータ収集

5. ま と め

本稿では, 階層的センサネットワークにおける効率的なデータ収集のための, 特異値分解を用いたデータ圧縮手法について述べた. 提案手法は, センサデータの周期性とセンサノード間の類似性という特徴を利用し, 特異値分解によってセンサデータに共通の基底系列とそれに対する重みにセンサデータ系列を分解することで, 大幅にデータ量を削減できる. また, その計算もインクリメンタル特異値分解によって高速であり, 実機のセンサノードにも搭載可能な程軽量であることを示した. 評価実験の結果, 提案手法でデータを圧縮して収集すると, センサノードの数が増加しても誤差はそれほど増加せずにデータを収集でき, しかもそのデータ量も大幅に低く抑えられることを示した. 提案手法により, 無線通信のデータ量と圧縮の計算量を低減できるため, 電池駆動のセンサノードの電力消費を抑えられる. これにより, センサネットワーク全体の駆動時間を延長することにも効果があると考えられる.

提案手法はセンサデータ系列の傾向をまとめながらセンサデータを集約し, 傾向の変化を基底系列の再送によって通知している, と解釈することもできるため, この特徴を積極的に利用したデータ解析について今後検討を進める予定である. また類似性の高いデータが発生するセンサノードが近隣のトポロジに配置されればより効率よくデータを収集できるため, データの傾向を考慮したトポロジ発見手法などにも取り組みたいと考えている.

参 考 文 献

1) Beigl, M. and Gellersen, H.: Smart-Its: An Embedded Platform for Smart Objects, Smart Objects Conference (sOc) 2003 (2003).

2) Deligiannakis, A., Kotidis, Y., and Roussopoulos N.: Hierarchical In-Network Data Aggregation with Quality, Proc. of Intl. Conf. on Extending Database Technology, pp.577-587 (2004).

3) Deligiannakis, A., Kotidis, Y., and Roussopoulos, N.: Dissemination of compressed historical information in sensor networks, The VLDB Journal, Vol.16, No.4, pp.439-461 (2007).

4) Edara, P., Limaye, A., and Ramamritham K.: Asynchronous in-network prediction: Efficient aggregation in sensor networks, ACM Transaction of Sensor Networks, Vol.4, Issues 4, pp.25:1-25:34 (2008).

5) Fan, K.-W., Liu, S., and Sinha P.: Scalable Data Aggregation for Dynamic Events in Sensor Networks, Proc. of Sensys 2006, pp.181-194 (2006).

6) Gupta, H., Navda, V., Das, S. R., Chowdhary, V.: Efficient gathering of correlated data in sensor networks, Proc. of MobiHoc 2005, pp. 402-413 (2005).

7) 岸野泰恵, 櫻井保志, 亀井剛次, 前川卓也, 柳沢豊, 岡留剛: 階層的センサネットワークのための効率的なデータ収集手法, 情報処理学会論文誌 データベース, Vol.3, No.4, pp.82-93 (2010).

8) 岸野泰恵, 柳沢豊, 寺田努, 塚本昌彦, 須山敬之: 小型無線デバイスのためのプログラム配布機能を備えた CIL 仮想マシン, 情報処理学会 マルチメディア, 分散, 協調とモバイルシンポジウム論文集, pp.1486-1494 (2010).

9) Madeen, S., Franklin, M.J., Hellerstein, J.M., and Hong, W.: TAG: a Tiny Aggregation Service for Ad-hoc Sensor Networks, Proc. of Symposium on Operating Systems Design and Implementation, pp.131-146 (2002).

10) Madden, S., Szewczyk, R., Franklin, M.J., and Culler, D.: Supporting Aggregate Queries Over Ad-hoc Wireless Sensor Networks, Proc. of IEEE Workshop on Mobile Computing Systems and Applications, pp.49-58 (2002).

11) Munguia, E., Stephen, T., and Larson, I. K.: Activity Recognition in the Home Using Simple and Ubiquitous Sensors, Proc. of Pervasive 2004, pp.158-175 (2004).

12) Papadimitriou, S., Sun, J., and Faloutsos, C.: Streaming pattern discovery in multiple time-series, Proc. of VLDB '05, pp.697-708 (2005).

13) Papadimitriou, S. and Yu, P.: Optimal multi-scale patterns in time series streams, Proc. of ACM SIGMOD 2006, pp.647-658 (2006).

14) Wilson, D. and Atkeson, C.: Simultaneous Tracking & Activity Recognition (STAR) Using Many Anonymous, Binary Sensors, Proc. of Pervasive 2005, pp.62-79 (2005).