1E4-4in2

クタスタリングとヒートマップによる高次元データ可視化

Time-Varying Data Visualization Using Clustered Heatmap and Dual Scatterplots

熊谷 沙津希, 伊藤 貴之[*1]　本橋 洋介, 梅津 圭介[*2]　高塚 正浩

Satsuki Kumatani, Takayuki Itoh　Yousuke Motohashi, Keisuke Umezu　Masahiro Takatsuka

[*1] お茶の水女子大学大学院 Ochanomizu University

[*2] 日本電気株式会社 NEC Corporation

[*3] The University of Sydney

Heatmap is one of the effective representations for time-varying data visualization. We may often want mechanisms to interactively filter non-important data items or time steps, so that we can form appropriate sizes of heatmaps and focus on important data items or time steps. This paper presents a heatmap-based time-varying data visualization technique featuring an interactive mechanism to display meaningful data items and time steps. This technique firstly calculates distances between arbitrary pairs of data items, and constructs a dendrogram consisting the data items. It then generates clusters of the data items and displays the data items belonging to the specified sizes of clusters in the heatmap, so that we can focus on groups of similar or correlated data items. It applies a similar mechanism to a set of time steps so that we can remove outlier time steps from the heatmap. Our implementation features two scatterplots, which represent distribution of data items and time steps respectively, and slider widgets to interactively adjust the thresholds of the clustering process.

1. 概要

ヒートマップは，時系列データ可視化のための効果的な表現の一つであり，折れ線グラフと同様に広く用いられている手法である．しかしながら，入力データが変数や時刻を多数含むとき，表示のための画面領域を大幅に必要とする場合がある．我々は，時系列データ可視化においてヒートマップを適切に形成するとともに，重要ではない標本や時刻を対話的にフィルタリングすることが重要であると考える．本報告では，変数と時刻の有意義な表示を可能にし，ユーザが対話的に操作できる時系列データのためのヒートマップベースな可視化手法について議論する．

我々が開発しているヒートマップベースの可視化手法の画面キャプチャを図 1 に示す．本手法では 2 標本間の非類似度，および 2 時刻間の非類似度を算出し，その位置関係を 2 つの散布図で表示する．それと同時に，非類似度にもとづいて標本群の系統樹を生成しヒートマップを生成する．図 1 において画面の左半分にはヒートマップが表示され，画面の右半分には 2 つの散布図が表示されている．画面の右端には 2 つのスライダー部品が搭載されており，これを操作することで重要でない標本，または重要でない時刻をヒートマップから割愛するための閾値を調節することができる．

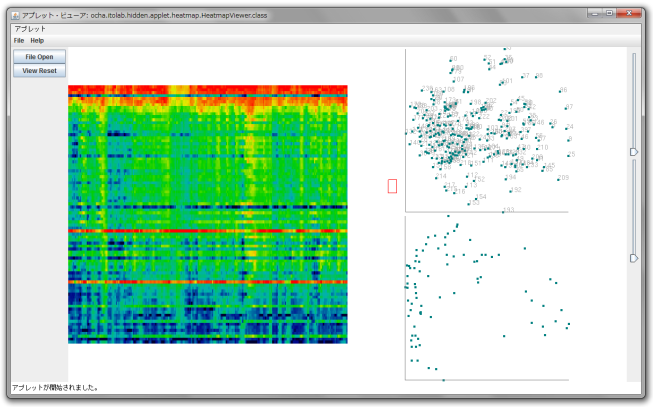

図 1 提案手法の画面キャプチャ例

2. 関連研究

時系列データの可視化には，一般的に折れ線グラフやヒートマップを用いた可視化が利用されている．

折れ線グラフはヒートマップに比べて，各標本における正確な数値を読み取りやすい，数値の上昇や下降を読み取りやすい，などの利点がある．しかし，データを構成する標本数の増加に伴い，折れ線どうしの絡み合いも増加し，視認性低下の原因となる．これらの問題を解決するために，折れ線の表示数を対話的に調節する手法が提案されている[1]．また，数値の範囲が大きく，かつ数値分布が不均一である場合，画面領域を浪費するような可視化結果になることが多い，という問題もある．

ヒートマップは値の大きさを色で表示する可視化手法である．ヒートマップを用いた時系列データの可視化には，データを構成する標本を縦軸に沿って一列ずつに並べ，横軸

連絡先：[*1]{satsuki@itolab.is.ocha.ac.jp, itot@is.ocha.ac.jp},
[*2]{y-motohashi@bk.jp.nec.com, k-umezu@ak.jp.nec.com},
[*3] masa.takatsuka@sydneyu.edu.au

に何らかの変数(時系列データの場合には時刻)を割り当てて，両軸を分割して得られる各領域に色付ける．ヒートマップは標本数が大きなデータにおいても，数値を表現する形状が画面上で重なり絡み合うことがないため，視認性の維持が容易である．さらに折れ線グラフと比較して，数値の範囲や分布により表示領域を浪費するような可視化結果を生じることもない．

ヒートマップを用いた可視化では，標本や変数を縦軸に沿って並べる順番によって，その効果が大きく変わる．ここで井元らはヒートマップの時間的に変化するデータの興味深い結果を抽出する手法を提示した[2]．さらに，標本の画面上での配置順を決定するための一手段として，類似性にもとづいて標本の分類や順列化を適用することが考えられる[3,4]．しかしこれらの手法では，k-means 法を単純に適用して標本をクラスタリングしているため，類似性は低いが関係性の高い標本を視覚的に比較することが容易ではない．例えば負の相関を有する変数が同一のクラスタに属することがないため，このような変数を視覚的に比較することが容易ではない．

3. 変数間距離に基づくヒートマップ表示

本章では提案手法が前提とするデータ構造を定義し，全体的な処理手順を述べたのちに，具体的な実装方法について論じる．

3.1 データ構造と処理手順

入力情報となる時系列データがm標本およびn時刻を有するデータであるとする．このとき本報告では，標本群Aおよび時刻群Tを以下のように表現する．ここでv_{ij}はi番目の標本のi番目の時刻における実数値である．

$$A = \{a_1, \dots, a_m\}$$
$$a_i = \{v_{i1}, \dots, v_{in}\}$$
$$T = \{t_1, \dots, t_n\}$$
$$t_j = \{v_{1j}, \dots, v_{mj}\}$$

以上の定義により，各標本および各時刻はベクタとして表現される．図 2(1)の赤線が 1 個の標本を，図 2(1)の青線が 1 個の時刻に対応する．

提案手法では任意の時刻間の非類似度(以下距離と称する)を算出して，それらの距離関係を表す散布図を図 2(2)のように表示する．同様に提案手法では，任意の標本間の距離を算出して，それらの距離関係を表す散布図を図 2(3)のように表示する．さらに提案手法では，時刻群と標本群にクラスタリングを適用することで，ヒートマップ上での表示対象となる時刻群と標本群を特定する(図 2(4))．

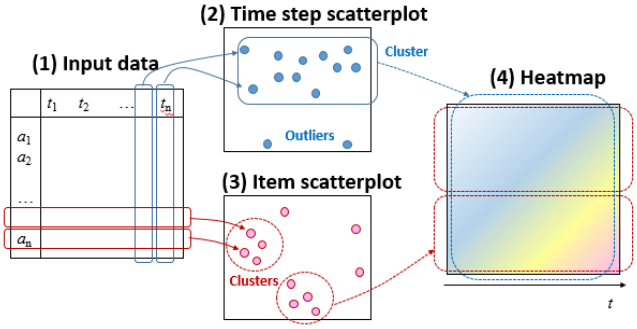

図 2 提案手法の処理手順

3.2 標本間の距離算出

2 章で論じたヒートマップ型の時系列データ可視化手法の問題点を解決するために，変数間の類似性に限定せずに任意の変数間距離を算出し，それに基づいて生成された距離行列を用い，変数をクラスタリングすることを考える．

我々は正と負の相関において変数間の関係を観察したいと考え，相関係数にもとづいて変数間距離を算出することにした．ここで高次元データを構成する変数$v = \{v_1 \dots v_n\}$とし，i番目の変数は数値$v_i = \{v_{i1} \dots v_{im}\}$を持つとする．現時点の我々の実装ではケンドール順位相関係数を適用して，式(1)からi番目とj番目の変数間の相関係数$d(i,j)$を算出する．また，このときPは変数iにおける任意の隣り合う 2 つの時刻と，変数jのこれに対応する 2 つの時刻について，それぞれの組の大小関係が一致するとき 1 を加算し，不一致のとき-1 を加算した和とする．

$$d(i,j) = \frac{4P}{n(n-1)} - 1$$

ここで得られた値を式(2)に適用し，i番目とj番目の変数間の距離$L_{i,j}$を得る．

$$L_{i,j} = 1 - |d(i,j)|$$

上記処理を全ての 2 変数のペアに適用し，距離行列を得る．続いてこの距離行列に，以下のクラスタリングを適用する．

3.3 階層型クラスタリングとヒートマップデザイン

階層型クラスタリングとは，近い距離にある標本を 1 個ずつ階層的に連結させ，標本を一続きにつなげた系統樹を形成し，クラスタをボトムアップに形成する手法である．提案手法では階層型クラスタリング手法により，全ての標本を連結する系統樹を構築し，その隣接順に沿ってヒートマップ上での標本の並び順を決定する．また距離の閾値を与えることで，標本群のクラスタを形成する．提案手法におけるヒートマップデザインの考え方を図 3 に示す．

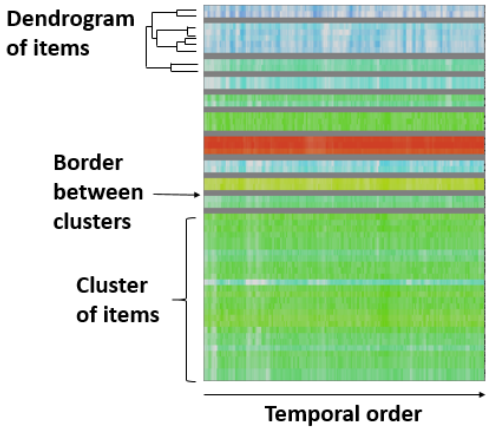

図 3　系統樹にもとづくヒートマップデザイン

提案手法が表示するヒートマップでは，横軸が時刻を表し，標本群が画面の縦軸に沿って上下に並ぶ．同一クラスタに属する標本群は間隔を空けずにひと続きに並べて表示し，異なるクラスタに属する標本間には境界線を表す灰色の帯をはさんで表示する．

また我々の実装では，表示するクラスタの構成標本数の下限値を 2 に設定し，それよりも標本数の少ないクラスタは表示しない，という制約を設けている．いずれの他の標本とも強い相関を有さない標本は標本数 1 のクラスタを構成するため，そのような標本は可視化する意義がないとし，ヒートマップから割愛する．

系統樹を構築するための提案手法での具体的な実装について述べる．木構造には各変数に相当するリーフ部分と，任意の変数同士を連結させる際の分岐に当たるブランチ部分がある．構築された系統樹を保存し，このブランチ部分に番号を振り分ける．より下位層にリーフを所有するブランチに若い番号を付与し，各階層において同様にブランチに番号を付与する．ヒートマップ表示の際にはこの番号が若い順に下から表示するため，ユーザ操作によって変わるクラスタリングの表示結果の前後で，各クラスタの位置，およびクラスタ内の各標本の位置関係が大きく入れ替わることがない．これにより，ユーザのメンタルマップを保持しやすい可視化を実現できる．同一の入力データに対してクラスタリングの閾値を調節して可視化結果の例を図 4 に示す．

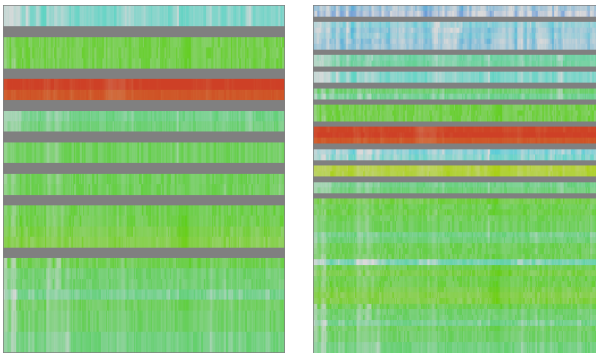

図 4　クラスタリング結果とヒートマップ表示
(左)閾値 0.2　(右)閾値 0.25

なお我々の実装では，時刻群にも同様に階層型クラスタリングを適用し，表示するクラスタの構成時刻数の下限値を 2 と設定している．これにより，他のいずれの時刻とも数値分布が全く異なるノイズのような時刻をヒートマップから割愛することができる．

3.4 カラーマップ

我々の実装ではヒートマップの色計算に以下の 2 種類のカラーマップを採用している．なお色計算には HSB 表色系を採用し，入力値にはv_{ij}を[0,1]の区間に正規化した値v'_{ij}を使用するものとする．

1)　色相のみを変化させる汎用的なカラーマップ

圧倒的に多数の可視化ソフトウェアが採用している色計算．色相をv'_{ij}の関数とし，彩度と明度を定数とする．

2)　平均値と相対値に着目したカラーマップ

HSB 値を以下の式で算出する．

$$\mathrm{H} = \mathrm{hue}(\overline{v'_{ij}}), S = saturation(v''_{ij}), B = const$$

ここで hue および saturation は色相と彩度を算出する関数であり，$\overline{v'_{ij}}$ は 1 つの標本におけるv'_{ij}の平均値であるとする．また$v''_{ij} = v'_{ij} - \overline{v'_{ij}}$ とする．この表現により，色相によって標本全体の数値の大小を表現し，彩度によって単一の標本における数値の変化を表現する．

3.5 散布図表示とユーザインタフェース

我々の実装では図 1 に示す通り，クラスタリングのための閾値を設定するスライダーを搭載している．スライダー操作を停止するたびに提案手法では，スライダーの位置によって指定される閾値にもとづいて，標本または時刻のクラスタを形成する．

また我々の実装では，系統樹が構築される様子を散布図上で可視化する．画面右側の 2 つの散布図では，標本群または時刻群の距離行列に対して MDS(多次元尺度構成法)を適用することで各標本または各時刻の散布図上の位置を算出している．この散布図ではスライダー操作に連動して，同一クラスタに所属する 2 標本または 2 時刻を連結する線分を描画する．この散布図表示により，標本間または時刻間の関係を把握しやすくなる．

4. 応用:異常値を含む標本間の相関の可視化

本報告の提案手法の応用として我々は現在，時系列データ中で異常値を含む標本を抽出し，その標本間の相関を可視化するツールを開発している．

3.1 節と同様に，入力情報となる時系列データがm標本およびn時刻を有するものとする．このとき我々の実装では，標本群$A = \{a_1, \ldots, a_m\}$ の中から異常値を含む上位m'個

の標本を抽出する．以降これを

$$A' = \{a_1, \ldots, a_{m'}\}$$

と記述する．我々の現在の実装では単純に，i 番目の標本を構成する各時刻の実数値に対して

$$(v_{ij} - \mu_i)/\sigma_i$$

を算出し，その最大値が上位である m’個の標本を抽出している．ここで μ_i は i 番目の標本における実数値の平均値，σ_i は i 番目の標本における実数値の標準偏差である．これ以外の異常値検出方法を適用することも可能である．

以上の処理によって抽出された標本群 A' に提案手法を適用することで，異常値を含む標本間の相関を可視化できる．このような可視化は例えば，

「同じ時期に異常によく売れる商品間の
売上の相関を可視化する」
「同じ時期に異常によくアクセスされる
ウェブページ間のアクセス数の相関を可視化する」

といった用途に有効である．

図 5 にその例を示す．なお，この応用例に関する実装では，散布図の表示を割愛している．かわりにこの実装では，ユーザによるヒートマップ上でのマウスクリック操作に対応する箇所の詳細情報を表示するテキスト表示欄をウィンドウ右下部に表示している．

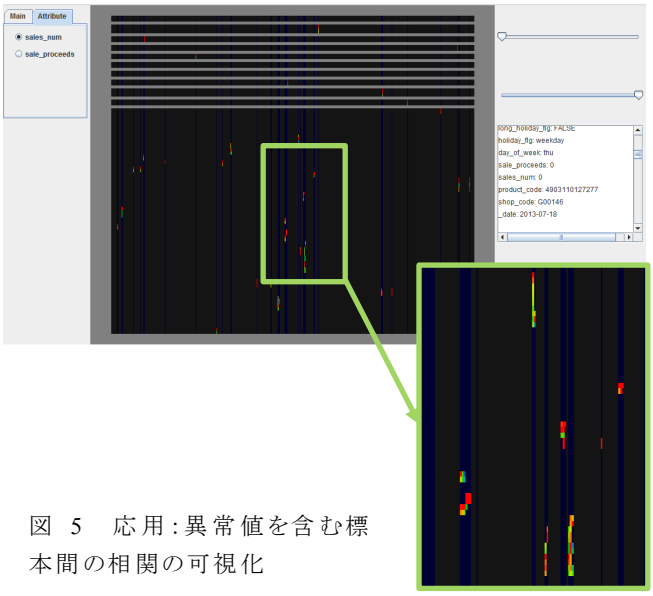

図 5　応用：異常値を含む標本間の相関の可視化

5. まとめと今後の課題

本報告では，ヒートマップを用いた時系列データ可視化の一手法を提案した．提案手法では時間軸を横軸として標本群を縦軸に並べる一般的なヒートマップとは別に，時刻群と標本群の位置関係を表示する 2 つの散布図を搭載している．時刻群と標本群はそれぞれ系統樹によって構造化され，対話操作による閾値設定とともに階層型クラスタリングが適用され，そのクラスタリング結果に沿ってヒートマップが更新される仕組みとなっている．またクラスタ形成の過程は散布図上にも表示される．

現在の実装では相関係数から算出した距離にもとづくクラスタリングを適用することで，ノイズとなる時刻を割愛し，相関のある標本群をクラスタとして表示している．今後の課題として，それ以外の距離やクラスタリング手法の適用によって，さらに多彩な数値特性を可視化できるようにしたい．特に，異常値の見られる標本群どうしにどのような相関が見られるか，あるいはどの時刻が重要であるか，といった点についてさらに考察を深めたい．

参　考　文　献

[1] [八木　2012]　八木，内田，伊藤: Polyline-Based Visualization Technique for Tagged Time-Varying Data，16th International Conference on Information Visualisation (IV2012)，106-111，2012.

[2] [井元　2010]　井元，伊藤: A 3D Visualization Technique for Large Scale Time-Varying Data，14th International Conference on Information Visualisation (IV2010)，17-22，2010.

[3] [林　2013]　林，伊藤，中村: A Visual Analytics Tool for System Logs Adopting Variable Recommendation and Feature-Based Filtering，17th International Conference on Information Visualisation (IV2013)，1-10，2013.

[4] [末松　2014]　末松，八木，伊藤，本橋，青木，森永: A Heatmap-Based Time-Varying Multi-Variate Data Visualization Unifying Numeric and Categorical Variables，18th International Conference on Information Visualisation (IV2014)，84-87，2014.