

Storyline を適用した時系列データの可視化

八木佐也香[†] 伊藤 貴之[†] 高塚 正浩^{††}

[†] お茶の水女子大学大学院人間文化創成科学研究科 〒112-8610 東京都文京区大塚 2-1-1

^{††} Faculty of Engineering & IT, The University of Sydney School of Information Technologies, J12,
University of Sydney, NSW 2006 Australia

E-mail: †{sayaka,itot}@itolab.is.ocha.ac.jp, ††masa.takatsuka@sydney.edu.au

あらまし 我々の身の回りには数多くの時系列データが存在しており、その分析において、可視化は有効である。代表的な時系列データの可視化手法として、折れ線グラフとヒートマップがあるが、いずれの手法もデータの部分的な特徴と全体のクラスタの変遷の両方を同時に観察するのは難しい。本論文では時系列数値情報を対象に、要素間の交流や共起性の可視化手法である storyline を用いた可視化手法を提案する。本手法ではある期間ごとに時系列数値情報の類似度を算出し、類似した要素同士が画面内で近接するような storyline を描く。さらに各線分の変化に着目し、storyline 上に視覚変数として与えることで、クラスタの切り替わる時間と変化量を直感的に読み取ることが可能になる。本手法を用いてクラスタの時間的変遷を観察することで、長時間にわたって類似した値/外れ値を有する要素だけでなく、所属するクラスタが頻繁に切り替わるような要素に着目しやすくなる。さらに元の数値情報やメタ情報を参照することで、クラスタの切り替えの要因が分析しやすくなると考えられる。

キーワード 時系列データ, 情報可視化, Storyline

1. はじめに

ビッグデータ時代の到来に伴い、多種多様なデータベースに時系列性を有する情報が蓄積されるようになった。これらの時系列データの分析・予測はデータマイニングや知覚情報処理等の分野においては既に数多くの研究が進められている。一方で、時系列データの観察と理解、またそれを扱う業務での能動的な意思決定を支援する道具として、情報可視化が有用である。情報可視化には、目で見て全体像を把握し、対話操作によってデータ中の必要な部分を能動的に選択できる点に加え、特に時系列データにおいてはリアルタイムな観察のための瞬時な理解においても有効である。一般的に情報可視化手法は、一次元、二次元、三次元、多次元、木構造、グラフ、時系列の7種類のデータ構造に大別されると言われている [1]。この7種類の中でも時系列性の観察を中心とした可視化手法の研究は、他のデータ構造に着目した研究に比べて発表件数が少なかった [2]。この理由として、以下が考えられる。

- 時系列データ可視化手法には、画面上の一軸を時刻とする強い制約があり、視覚表現のバリエーションを生みにくい。
- 時系列データは数値情報に加え、メタ情報や他のデータ構造などと結びついた複合的なデータである場合が多いため、時系列以外のデータ構造（多次元や木構造など）が優先的に着目されることが多い。

そこで我々はこれまで、時系列データの可視化に関する新しい手法の開発に取り組んできた [3]。

多量の実数値で構成される時系列データの可視化では、データを構成する実数値の大小そのものだけでなく、実数値どうしの関連性や変化量などを観察することも重要である。例えば多数の商品の売上を一定時刻ごとに集計して作成される時系列

データでは、商品単体の売上の推移だけでなく、どの商品と同時期に売れるか、季節や時間帯との相関はあるか、といった関連性を観察することによって、より深い分析が可能になり、売上予測や予兆発見などに貢献できると考えられる。

時系列データの可視化では折れ線グラフとヒートマップの2種類の視覚表現が特に多用されている [2]。しかしいずれの視覚表現においても、大量のデータ要素（以下「要素」と称する）の相互関係を全て表現するのは難しい。そこでクラスタリングなどを適用して要素を構造化した上で可視化することが有効である。ここで時系列データにおいては要素間の相互関係も時間とともに変遷するため、クラスタ構造の時間的変遷を表現することも重要となる。一方でこれらの可視化手法において、データの部分的な特徴とクラスタ構造の時間的変遷を同時に表現するのは簡単ではない。

実数値で構成される時系列データに限定しなければ、クラスタ構造の時間的遷移の表現に向けた可視化手法として、Sankey Diagram と storyline がよく利用されている。Sankey Diagram はクラスタ間の変化の総量を観察するのに適している。一方、storyline は個々の要素についてのクラスタから別のクラスタへの移動を観察するのに向いている。Storyline はもともと XKCD というウェブコミックに手書きイラスト “Movie Narrative Charts” として紹介されたものであり [4]、物語の登場人物の共起行動を時系列のダイアグラムで表現している。図 1 にあるように、x 軸が時刻、各曲線が映画の各登場人物を表しており、複数の登場人物が同一のシーンに登場した場合に線が近接するような可視化結果となる。

以上を踏まえて我々は、storyline を用いて実数値で構成される時系列データを表現する可視化手法を提案する。本手法では、ある期間ごとに時系列数値情報の類似度を算出し、類似し

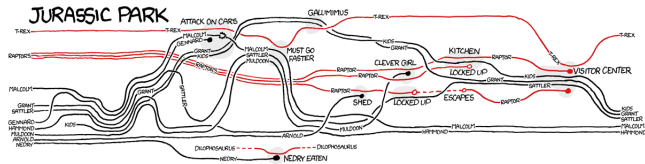


図1 手書きによる storyline の例 ([4] より転載)。

た要素同士が画面内で近接するような storyline を描く。ここで storyline 上を構成する曲線群の描画において、対応する要素が有する数値の変化も同時に表現することで、クラスタの移動の要因となる数値の変化量も直感的に読み取ることが可能になる。さらに本手法では、単にクラスタの移動を伴う要素を発見しやすくするだけでなく、このような変化を有する要素の数値情報やメタ情報を参照できる機能を有する。これを用いることで、要素がクラスタを移動する要因を考察できる。

本手法を用いてクラスタの時間的変遷を観察することで、長時間にわたって類似した値/外れ値を有する要素の発見が容易になる。また、ある時には特定のクラスタに属し、またある時には別のクラスタに属する、というようなふらふらとした時間的遷移を有する要素に着目しやすくなる。例えば、夏には商品 A と B の売上の時間的推移が類似して、冬には商品 A と C の売上の時間的推移が類似している、という商品データがあるとす。本手法でこれを可視化することで、商品 A が夏から冬にかけて別のクラスタに移動していることが storyline 上で明確に表現できる。これによって、商品 A に対する店舗での陳列には注意が必要である、というような知識の発見につながる。

2. 関連研究

要素間の共通性を示す storyline のメタファを用いた時系列データ可視化手法が近年数多く提案されている [5]-[10] が、その多くは XKCD の手描きイラスト [4] に着想を得たものである。中でも以下の 2 つの手法は、XKCD 風の storyline の自動生成を目標としている。

Ogawa らはソフトウェア開発のリポジトリデータを適用し、storyline のレイアウトを自動生成する手法を提案した [5]。Ogawa らの手法は、地下鉄の路線図風の太い線分を用いて storyline を描画するものとし、以下の定義にもとづいて線分の配置を最適化している。

- 相互作用している線分（開発者）は近接した track (interaction session) に配置する
 - 異なるクラスタどうしは離れるように配置する。
 - 存在し続けている線分はあまり位置が変わらないように配置する。

この手法により、小～中規模のプロジェクトにおいて、ソフトウェア開発の構造がどのように変化しているかを観察することが可能になる。また、クリック操作によって線分をハイライトする機能も実装されている。しかし、Ogawa らの可視化結果には線分の小刻みな蛇行や線分同士の重なりが数多く含まれており、時間軸に沿って個々の線分の動きを追うことは困難である。

Tanahashi らは Ogawa らの手法 [5] を拡張し、遺伝的アル

ゴリズムを用いて線分の配置を最適化する手法を提案した [6]。Tanahashi らの手法では、Ogawa らの定義に加え、他の線分に収束・分岐しない限り、1本の線分はなるべく曲がらないという制約を付与している。さらに、線分のずれや重なり、画面上の空白を制御することで、線分同士の共起性に着目しやすい結果が得られている。いずれの手法についても、スケーラビリティと計算時間に課題がある。データ数が数百以上の規模になった場合、線分同士の重なりがひどくなり、線分の観察が困難になると考えられる。

この問題を解決するため、StoryFlow は各クラスタに詳細度制御を適用し、storyline 全体の特徴を捉えやすくすることを可能にした [7]。StoryFlow では各要素の階層構造を考慮した配置に加え、インタラクティブな線分の並べ替え、バンドリングなどの機能も実現している。また、Reda らは storyline のメタファを用いながらも、数百規模の要素の描画が可能な手法を提案している [8]。Reda らの手法では、y 軸上にコミュニティを示すスロットを固定し、各スロット内に線分を配置している。このアプローチはユーザが特定のクラスタ間の変化に着目したい場合、有効であるといえる。

しかしこれらの既存手法の中で、storyline を構成する曲線の移動量に着目した手法はほとんどない。また、既存手法のほとんどが既知の関係性やキーワード/トピックをもとに算出した類似度をクラスタリングに用いており、実数値で構成される時系列データから要素間の類似度を算出するという事例は我々が調査する限り見当たらない。

3. 従来の storyline の問題点

storyline には 1 章で論じたように、どのクラスタがどの要素群で構成されているか、どの要素がいつクラスタを移動したか、といった現象を観察するのに向いている。しかし storyline を用いて実数値の時系列データを可視化するにあたり、以下の問題点が考えられる。storyline では多くの場合において、画面上で大きく移動する曲線が目立って見えるため、この曲線に対応する要素が重要な要素、あるいは大きな実質的变化を有する要素であると連想しがちである。一方で storyline を構成する曲線の移動量は、個々のクラスタに含まれる曲線の本数や、クラスタの配置などによって決定される。よって実数値の時系列データを storyline で可視化すると、実数値の変化の小さい要素に対応する曲線が画面上で大きく移動して目立ってしまう、というような直感に合わない可視化結果を生むことがある。例えば図 2 において、左下部の赤の曲線は短時間で急激な変化を有しているように見える。その他にも画面中央から右側にかけて、一番下のクラスタから画面上部のクラスタへと大きく移動する曲線が複数見受けられる。これらの曲線は画面上での移動量が大きいため、直感的にユーザの目に留まりやすい。一方で、クラスタの配置やクラスタ内での線分の配置を変えることにより、これらの移動量は小さくもなり得る。実数値の時系列データにこのような可視化手法を適用した場合に、曲線の画面上での移動量が必ずしも実数値の変化量に比例するとは限らないため、誤解を生むような可視化結果を生じることが懸念される。

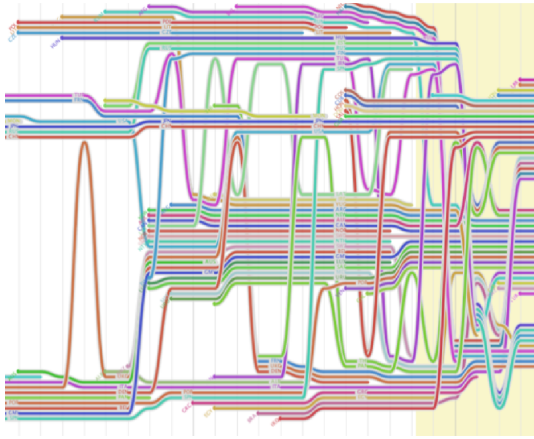


図2 画面上での変化量が大きく見える storyline の例 (文献 [6] より転載)。

もう1つの問題点として、各要素の実数値間の類似度を算出する単位となる時間帯 (ウィンドウサイズ) の設定方法が挙げられる。前章で述べたように、従来の手法ではソフトウェア開発のリポジトリログや議会の投票記録、特定のキーワードに関するコミュニティの変遷などを可視化しており、それらの interaction session は固定であった。数値データの類似度を算出する場合、類似度算出に用いるウィンドウサイズが出てくる切り替えの数や時間、クラスタ等に大きく影響する。図3は同じ時系列数値データに対し、ウィンドウサイズを2パターン適用した結果、描画される storyline の違いを示す。図3(上)では、 dw_0 , dw_1 において赤の要素は他の2要素 (黄・青) とは別のクラスタに配置されている。しかし、図3(下)のようにウィンドウサイズを変えることで、 dw_1 においては3つの要素すべてが同一のクラスタに属することが観察でき、こちらの方がもとの折れ線グラフの特徴に近いといえる。このように、ユーザが元の数値データをどのように分割するのが適切であるかの知見を有していない場合、ウィンドウサイズを可変にすることで、目で見えて判断できるようになると考えられる。

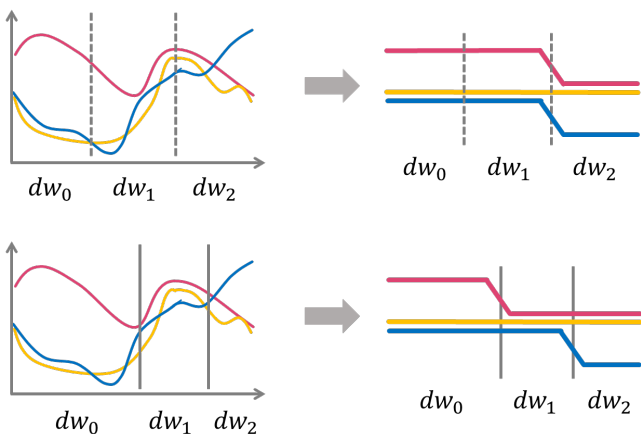


図3 類似度算出用のウィンドウサイズを変えることにより、描画される storyline に違いが生じる。

4. 提案手法

実数値で構成される時系列データの各要素にクラスタリング

を適用し、その結果を storyline で可視化することで、主に以下の3種類の性質をもつ要素を視覚的に観察できると考えられる。

- (1) 同一のクラスタにとどまる (相互作用している) 要素
- (2) クラスタの移動が激しい (ふらふらしている) 要素
- (3) クラスタに属さない (他の要素と作用しない) 要素

既存手法では要素同士の相互作用を可視化することを重視しており (結果として (3) のような要素も観察できる)、必ずしも (2) に着目しやすい結果になるとは言えなかった。しかし我々は、頻繁にクラスタを移動する (2) の要素に着目することも重要と考える。

Marr の理論によると、人間の目は知覚的に変化の激しい部分に対して反応する [11]。我々は storyline 上での要素のクラスタ移動を強調表示することで、人間の知覚に近い可視化結果を得られると考える。具体的には、各曲線におけるクラスタ移動時の実数値の微分値を算出し、その値を視覚的に storyline に反映させる。反映方法として以下の2つのアプローチが考えられる。

- 視覚変数として storyline 上に載せる
- 線分の並べ替えに反映させる (線の配置の最適化の指標として使用する)

本論文では前者を採用した手法を提案する。次節にて、視覚変数の種類と提案手法における各変数の有効性について述べる。

4.1 視覚変数の選択

視覚変数とは Bertin [12] によって提案された、データを視覚記号に変換するために用いる変数集合である。Bertin の提示した視覚変数は、position, size, shape, (color) value, color (hue), orientation, texture (grain) の7種類である。MacEachen はこれらの変数に加え、明瞭さに関する3つの変数 (crispness, resolution, transparency) を挙げている [13]。この他にも多くの先行研究が Bertin の視覚変数を拡張している [14] [15] [16] が、本節では上記の10個の変数について、有効性を検証する。

本手法において、以下の変数は曲線に対応する要素の実数値変化を表すためには不適と考える。その理由は下記の通りである。

- Position: 既に曲線間の相互作用を表現するために使用している
- Size (曲線の太さ): 描画する曲線の本数が多いときに、曲線同士の重なりが発生する
- Shape, orientation, texture: 曲線上の細かい領域に適用するのが難しい
- Color value, color hue: 既に曲線の属性の識別用に使っているため、さらに色を使うと情報が読み取りづらい
- Crispness, resolution: 値の順序を読み取るのに効果的でない [14]

また、既存の時系列データ可視化の研究、特にグラフ構造の変化や地図上の時系列性に着目した手法においては、その多くがアニメーションを時系列性を示す変数として利用している。しかし、アニメーションを使用した場合、複数の時刻の結果を比較することが困難である。さらに、本手法では対象の期間すべての特徴を捉えることを目標としているので、視覚変数とし

てアニメーションを用いるのは不適切であるといえる。

MacEachen は著書の中で、画面内の同一座標上に複数の情報を載せる場合、透明度を用いるのが有効であると述べている [13]。透明度から定量的な差異を読み取ることは困難であるものの、ユーザは直感的に濃さを見て変化量が同程度か判断できると考えられる。以上を踏まえ、本手法では視覚変数として透明度を採用する。具体的には、曲線に対応する要素の実数値変化が大きい部分は透明度を低く、実数値変化が小さい部分は透明度を高くすることで、既に用いられている他の視覚変数に干渉することなく、ユーザの注意を実数値変化の大きい部分に向けられると考える。

4.2 処理手順

本手法の処理の流れは下記の通りである。

- (1) 各ウィンドウ内の時系列数値データのクラスタリング
- (2) クラスタの画面配置
- (3) storyline の透明度算出
- (4) storyline の描画

以下で各手順について詳しく述べる。

4.2.1 各ウィンドウ内の時系列数値データのクラスタリング

まず与えられた時系列データを任意の幅で区切り、各ウィンドウに対してクラスタリングを適用する。標本時間内に n 個のタイムステップが含まれている場合、グループ内の要素の各時点における y 座標値をもった n 次元ベクトルを生成する。各ウィンドウについて、ベクトルの類似度を算出することで、要素同士の相互作用を求める。同じクラスタに属する要素を storyline 上での interaction session とみなす。本論文ではクラスタリング手法として非階層型クラスタリングを適用しているが、階層型を含め他のクラスタリング手法を適用することも可能である。ここで、クラスタの類似度を求めるためのウィンドウの幅を可変にすることで、「あるウィンドウサイズでは同じクラスタに含まれる要素が、ウィンドウサイズを小さくすることによって別のクラスタに移る要素が出てくる」といった変わり目を観察できると考えられる。

4.2.2 クラスタの画面配置

一般的な storyline の描画手法として、以下の 2 方法が考えられる。

- Tanahashi らの手法 [6] のように、線の配置を最適化する方法
- Reda らの手法 [8] のように、クラスタの位置を y 軸上で固定する方法

後者を適用する場合、従来手法では既知のクラスタをもとに配置を行っているが、数値データから類似度を算出する場合、隣り合うウィンドウ同士でクラスタのマッチングを取る必要がある。本手法では、以下の手順でクラスタの共通度を求める。

元の時系列データを任意の幅で n 個に区切ったウィンドウのセットを $W = \{w_1, \dots, w_n\}$ とすると、 i 番目のウィンドウ w_i は以下で表される：

$$w_i = \{t_{start}, t_{end}, C_{ij}\}, \text{ ただし } C_{ij} = \{c_{i1}, c_{i2}, \dots, c_{ik}\}.$$

ここで、 t_{start} , t_{end} は w_i の開始時間および終了時間であり、

k はウィンドウ w_i におけるクラスタ数、 c_{ij} は j 番目のクラスタを示す。

本手法では、隣接ウィンドウ内に含まれるクラスタ C_{i1j_1} , C_{i2j_2} の共通度を、以下の式 (1) が最大となる重み α を求める最大マッチング問題として解く。

$$\sum \left(\alpha \frac{n(C_{i1j_1} \cap C_{i2j_2})}{n_{i1j_1} + n_{i2j_2}} + (1 - \alpha) \frac{1}{|a_{i1j_1} - a_{i2j_2}|} \right) \quad (1)$$

ここで、 n_{ij} は C_{ij} に含まれる曲線の本数であり、 a_{ij} は共通の線の値の平均を示す。式 (1) が最大となるクラスタを隣接ウィンドウ間で一致するクラスタとみなし、共通のクラスタ番号を与える。現在の実装においては、 y 軸上のクラスタの並びは以下のように決定している。

- (1) 画面左端のウィンドウ内部にてクラスタを番号順に配置、
- (2) 画面の左から 2 番目以降のウィンドウ内部ではクラスタの共通度が高くなるように配置。

将来的には、(1) の初期配置を SOM (self-organizing maps) によって多次元データを 1 次元に変換し決定することで、1 次元の数値情報だけでなく、メタ情報を含めた多次元の情報をクラスタの配置に反映することも可能であると考えられる。

4.2.3 storyline の透明度算出

現時点での我々の実装では、要素の実数値の微分を storyline の透明度に採用している。隣接ウィンドウ間でクラスタを移動する曲線について、対応する要素の実数値の微分を以下の式 (2) で求める。

$$\frac{d_{ab}}{|dt|} \quad (2)$$

ここで、 d_{ab} はクラスタ AB 間の距離であり、 $|dt|$ は微分値を計算する時間幅を示す。図 4 に見られるように、storyline の描画では曲線同士の重なりを減らすために、クラスタの移動が発生する時間帯を横方向に拡大して表示することが一般的である。現在の実装では、この拡大表示を適用した一定時間を $|dt|$ としているが、特に長期間における変化量を算出したい場合には、任意の時間幅を $|dt|$ に適用することが望ましい。

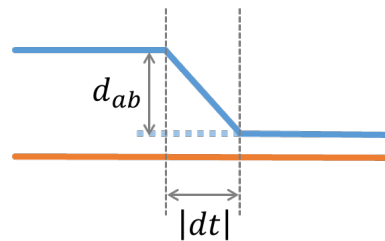


図 4 Storyline 上でクラスタの移動が発生するとき、拡大表示した時間帯を $|dt|$ とする。

4.2.4 storyline の描画

描画時には、(2) で求めた y 軸上のクラスタの配置をもとに、各クラスタ内に任意の順で曲線を配置する。このとき、曲線の 1 次元の属性を色として付与することが可能である。さらに、(3) で求めた微分値をもとに、各曲線のクラスタ移動部分に透

明度を与える。微分値が大きいほど透明度を低く描画することで、観察者の目を実数値の変化が大きい部分に向けることができる。

また、各要素のクラスタ移動の要因を分析するために、実数値を直接可視化した結果と storyline を比較することが有効であると考える。これを視野に入れて将来的には、本手法と折れ線グラフベースの複合型時系列データ可視化手法 [3] と連携操作可能な状態で併用することを考えている。

5. 実行結果

我々は、全国 86 箇所の観測所について、3 時間ごとに観測された 2010 年 1 月の 1 カ月分のアメダス気温データを適用した。x 軸を時刻、1 本の曲線を 1 観測所とし、近接する曲線が同じクラスタに属していることを示す。この事例ではクラスタ数を 4、ウィンドウサイズを 12 時間ごと、各クラスタ内の曲線の並びは観測所 ID 順とし、曲線を地方ごとに 8 色に色分けした。地方の分類は観測所 ID に基づいて決定した。図 5 に色と地方の対応を示す。なお色については、区分を明瞭にするため、ColorBrewer2 [17] の配色に従って決定した。

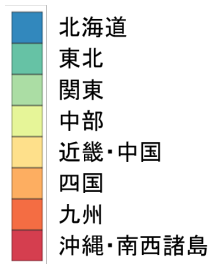


図 5 曲線の色と地方の対応。

図 6 は 1 月上旬の 4~5 日分を表す storyline である。全体的に、青で示された北海道や深い赤で示された沖縄・南西諸島がそれぞれ単独のクラスタで推移していることが観察できる。その他の地方については、図の下部に配置されたクラスタに属する観測所が多いが、後半の 1 日分 (2 ウィンドウ分) については、北海道や東北、九州にかけてがすべて同じクラスタに属しており、全国的に同じような気温の変化をしていたと推察できる。

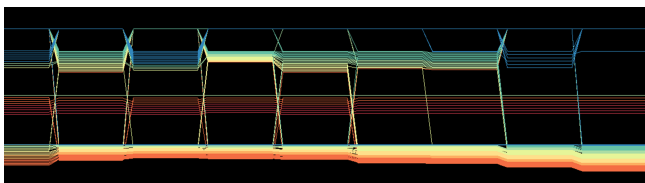


図 6 2010 年 1 月上旬の 4~5 日分を表した storyline。

同じデータを折れ線ベースの可視化手法 [3] で描画したものを図 7 に示す。横軸が時刻、縦軸が気温、各折れ線が各観測所を示す。なお、こちらの手法では赤を快晴、黄を晴れ、緑を曇り、雨を青、雪をシアンというように、天気の色を割り当てている。破線で囲んだ部分が前述の storyline での後半 1 日分に相

当する。上部に描画された高い気温の折れ線を除き、折れ線群の多くが似たような気温の変化を示しており、気温差も他の時刻に比べて小さいことが観察できる。この結果から、storyline でクラスタの変化の傾向を観察すると同時に、特徴的な気温の変化を見つけることが可能であるといえる。

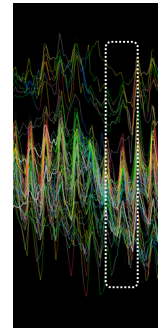


図 7 2010 年 1 月上旬の 4~5 日分を折れ線ベースの可視化手法 [3] で描画した結果。

6. まとめと今後の課題

本論文では、storyline を用いた時系列数値データの可視化手法を提案した。本手法ではある期間ごとに時系列数値情報の類似度を算出し、類似した要素同士が画面内で近接するような storyline を描く。

今後の課題としては、以下が考えられる。

- Storyline 表示画面左端のウィンドウのクラスタ配置の最適化
- クラスタ内の曲線の並び順の検討
- 折れ線グラフを用いた可視化手法 [3] との連携機能の実装

文 献

- [1] B. Shneiderman, The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations, IEEE Symposium on Visual Languages, 336-343, 1996.
- [2] 伊藤, 視覚協創学 (5): マッピング技術 特に時系列データ可視化技術の体系化に向けて, 第 17 回計算工学講演会, 2012.
- [3] S. Yagi, Y. Uchida and T. Itoh, A Polyline-Based Visualization Technique for Tagged Time-Varying Data, 16th International Conference on Information Visualisation (IV2012), pp. 106-111, 2012.
- [4] R. Munroe, xkcd # 657: "Movie narrative charts", <http://xkcd.com/657/>, Accessed January 2015.
- [5] M. Ogawa and K.-L. Ma, Software evolution storylines, In Proceedings of the 5th international symposium on Software visualization, 35-42, 2010.
- [6] Y. Tanahashi and K.-L. Ma, Design considerations for optimizing storyline visualizations, IEEE Transactions on Visualization and Computer Graphics, 18(12) 2679-2688, 2012.
- [7] S. Liu, Y. Wu, E. Wei, M. Liu and Y. Liu, Storyflow: Tracking the evolution of stories. Visualization and Computer Graphics, IEEE Transactions on, 19(12), 2436-2445, 2013.
- [8] K. Reda, C. Tantipathananandh, A. Johnson, J. Leigh and T. Berger-Wolf, Visualizing the evolution of community structures in dynamic social networks, Comp. Graphics Forum, 30(3) 1061-1070, 2011.
- [9] N. W. Kim, S. K. Card and J. Heer, Tracing genealogical data with timenets, In Proceedings of the International

Conference on Advanced Visual Interfaces, ACM, 241-248, 2010.

- [10] C. W. Muedler, T. Crnovrsanin, A. Sallaberry, K.-L. Ma, *Egocentric storylines for visual analysis of large dynamic graphs*, Big Data, 2013 IEEE International Conference on, 56-62, 2013.
- [11] D. Marr, *Vision: A computational investigation into the human representation and processing of visual information*, Henry Holt and Co. Inc., New York, 2-46, 1982.
- [12] J. Bertin, *Semiology of graphics: Diagrams, networks, maps (WJ Berg, Trans.)*, Madison, WI: The University of Wisconsin Press, Ltd, 1983.
- [13] A. M. MacEachren, *How Maps Work: Representation, Visualization, and Design*, The Guilford Press, 1995.
- [14] L. Halik, *The analysis of visual variables for use in the cartographic design of point symbols for mobile Augmented Reality applications*, Geodesy and Cartography 61(1), 19-30, 2012.
- [15] J. L. Morrison, *A theoretical framework for cartographic generalization with the emphasis on the process of symbolization*, International Yearbook of Cartography, 14, 115-127, 1974.
- [16] J. L. Caivano, *Visual texture as a semiotic system*, Semiotica, 80(3-4), 239-252, 1990.
- [17] C. A. Brewer and M. Harrower, "ColorBrewer 2.0.", <http://colorbrewer2.org/>, Accessed February 2015.